

User's Guide

Scyld ClusterWare Release 7.3.3-733g0000

April 28, 2017

User's Guide: Scyld ClusterWare Release 7.3.3-733g0000; April 28, 2017

Revised Edition

Published April 28, 2017

Copyright © 1999 - 2017 Penguin Computing, Inc.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording or otherwise) without the prior written permission of Penguin Computing, Inc..

The software described in this document is "commercial computer software" provided with restricted rights (except as to included open/free source). Use beyond license provisions is a violation of worldwide intellectual property laws, treaties, and conventions.

Scyld ClusterWare, the Highly Scyld logo, and the Penguin Computing logo are trademarks of Penguin Computing, Inc.. Intel is a registered trademark of Intel Corporation or its subsidiaries in the United States and other countries. Infiniband is a trademark of the InfiniBand Trade Association. Linux is a registered trademark of Linus Torvalds. Red Hat and all Red Hat-based trademarks are trademarks or registered trademarks of Red Hat, Inc. in the United States and other countries. All other trademarks and copyrights referred to are the property of their respective owners.



Table of Contents

Preface	v
Feedback	v
1. Scyld ClusterWare Overview	1
What Is a Beowulf Cluster?	1
A Brief History of the Beowulf	1
First-Generation Beowulf Clusters	2
Scyld ClusterWare: A New Generation of Beowulf	3
Scyld ClusterWare Technical Summary	3
Top-Level Features of Scyld ClusterWare	3
Process Space Migration Technology	5
Compute Node Provisioning	5
Compute Node Categories	5
Compute Node States	5
Major Software Components	6
Typical Applications of Scyld ClusterWare	7
2. Interacting With the System	9
Verifying the Availability of Nodes	9
Monitoring Node Status	9
The BeoStatus GUI Tool	9
BeoStatus Node Information	10
BeoStatus Update Intervals	10
BeoStatus in Text Mode	11
The bpstat Command Line Tool	11
The beostat Command Line Tool	12
Issuing Commands	14
Commands on the Master Node	14
Commands on the Compute Node	14
Examples for Using bpsb	14
Formatting bpsb Output	15
bpsb and Shell Interaction	16
Copying Data to the Compute Nodes	17
Sharing Data via NFS	17
Copying Data via bpcp	17
Programmatic Data Transfer	18
Data Transfer by Migration	18
Monitoring and Controlling Processes	18
3. Running Programs	21
Program Execution Concepts	21
Stand-Alone Computer vs. Scyld Cluster	21
Traditional Beowulf Cluster vs. Scyld Cluster	21
Program Execution Examples	22
Environment Modules	24
Running Programs That Are Not Parallelized	24
Starting and Migrating Programs to Compute Nodes (bpsb)	25
Copying Information to Compute Nodes (bpcp)	25
Running Parallel Programs	26
An Introduction to Parallel Programming APIs	26

MPI.....	27
PVM	27
Custom APIs.....	28
Mapping Jobs to Compute Nodes	28
Running OpenMPI Programs.....	29
Pre-Requisites to Running OpenMPI	29
Using OpenMPI.....	29
Running MPICH2 and MVAPICH2 Programs	30
Pre-Requisites to Running MPICH2/MVAPICH2	30
Using MPICH2.....	30
Using MVAPICH2.....	30
Running PVM-Aware Programs	30
Porting Other Parallelized Programs.....	31
Running Serial Programs in Parallel.....	31
mpprun	31
Options	32
Examples	32
beorun.....	32
Options	32
Examples	33
Job Batching	33
Job Batching Options for ClusterWare	33
Job Batching with TORQUE	34
Running a Job	34
Checking Job Status	35
Finding Out Which Nodes Are Running a Job.....	36
Finding Job Output	36
Job Batching with POD Tools.....	36
Using Singularity	37
File Systems.....	37
A. Glossary of Parallel Computing Terms	39
B. TORQUE and Maui Release Information.....	43
C. OpenMPI Release Information	45
D. MPICH2 Release Information.....	111
E. MVAPICH2 Release Information.....	117
F. MPICH-3 Release Information.....	149
CHANGELOG	149
Release Notes.....	180
G. SLURM Release Information.....	183

Preface

Welcome to the Scyld ClusterWare HPC User's Guide. This manual is for those who will use ClusterWare to run applications, so it presents the basics of ClusterWare parallel computing — what ClusterWare is, what you can do with it, and how you can use it. The manual covers the ClusterWare architecture and discusses the unique features of Scyld ClusterWare HPC. It will show you how to navigate the ClusterWare environment, how to run programs, and how to monitor their performance.

Because this manual is for the user accessing a ClusterWare system that has already been configured, it does *not* cover how to install, configure, or administer your Scyld cluster. You should refer to other parts of the Scyld documentation set for additional information, specifically:

- Visit the Penguin Computing Support Portal at <http://www.penguincomputing.com/support/documentation> to find the latest documentation.
- If you have not yet built your cluster or installed Scyld ClusterWare HPC, refer to the latest *Release Notes* and the *Installation Guide*.
- If you are looking for information on how to administer your cluster, refer to the *Administrator's Guide*.
- If you plan to write programs to use on your Scyld cluster, refer to the *Programmer's Guide*.

Also not covered is use of the Linux operating system, on which Scyld ClusterWare is based. Some of the basics are presented here, but if you have not used Linux or Unix before, a book or online resource will be helpful. Books by *O'Reilly and Associates*² are good sources of information.

This manual will provide you with information about the basic functionality of the utilities needed to start being productive with Scyld ClusterWare.

Feedback

We welcome any reports on errors or difficulties that you may find. We also would like your suggestions on improving this document. Please direct all comments and problems to support@penguincomputing.com.

When writing your email, please be as specific as possible, especially with errors in the text. Please include the chapter and section information. Also, please mention in which version of the manual you found the error. This version is *Scyld ClusterWare HPC, Revised Edition*, published April 28, 2017.

Notes

1. <http://www.penguincomputing.com/support/documentation>
2. <http://www.oreilly.com>

Preface

Chapter 1. Scyld ClusterWare Overview

Scyld ClusterWare is a Linux-based high-performance computing system. It solves many of the problems long associated with Linux Beowulf-class cluster computing, while simultaneously reducing the costs of system installation, administration, and maintenance. With Scyld ClusterWare, the cluster is presented to the user as a single, large-scale parallel computer.

This chapter presents a high-level overview of Scyld ClusterWare. It begins with a brief history of Beowulf clusters, and discusses the differences between the first-generation Beowulf clusters and a Scyld cluster. A high-level technical summary of Scyld ClusterWare is then presented, covering the top-level features and major software components of Scyld. Finally, typical applications of Scyld ClusterWare are discussed.

Additional details are provided throughout the Scyld ClusterWare HPC documentation set.

What Is a Beowulf Cluster?

The term "Beowulf" refers to a multi-computer architecture designed for executing parallel computations. A "Beowulf cluster" is a parallel computer system conforming to the Beowulf architecture, which consists of a collection of commodity off-the-shelf computers (*COTS*) (referred to as "nodes"), connected via a private network running an open-source operating system. Each node, typically running Linux, has its own processor(s), memory storage, and I/O interfaces. The nodes communicate with each other through a private network, such as Ethernet or Infiniband, using standard network adapters. The nodes usually do not contain any custom hardware components, and are trivially reproducible.

One of these nodes, designated as the "master node", is usually attached to both the private and public networks, and is the cluster's administration console. The remaining nodes are commonly referred to as "compute nodes". The master node is responsible for controlling the entire cluster and for serving parallel jobs and their required files to the compute nodes. In most cases, the compute nodes are configured and controlled by the master node. Typically, the compute nodes require neither keyboards nor monitors; they are accessed solely through the master node. From the viewpoint of the master node, the compute nodes are simply additional processor and memory resources.

In conclusion, Beowulf is a technology of networking Linux computers together to create a parallel, virtual supercomputer. The collection as a whole is known as a "Beowulf cluster". While early Linux-based Beowulf clusters provided a cost-effective hardware alternative to the supercomputers of the day, allowing users to execute high-performance computing applications, the original software implementations were not without their problems. Scyld ClusterWare addresses — and solves — many of these problems.

A Brief History of the Beowulf

Cluster computer architectures have a long history. The early network-of-workstations (*NOW*) architecture used a group of standalone processors connected through a typical office network, their idle cycles harnessed by a small piece of special software, as shown below.

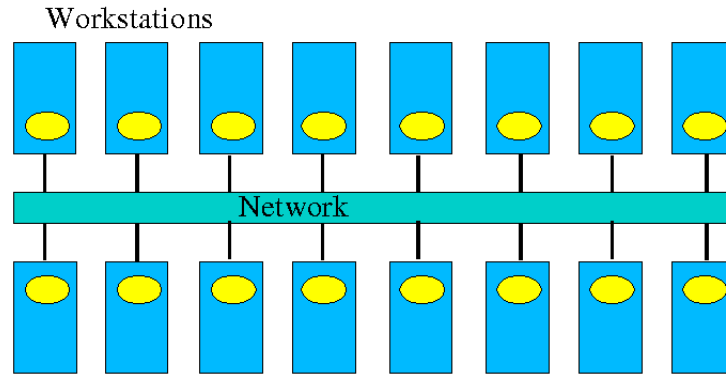


Figure 1-1. Network-of-Workstations Architecture

The *NOW* concept evolved to the Pile-of-PCs architecture, with one master PC connected to the public network, and the remaining PCs in the cluster connected to each other and to the master through a private network as shown in the following figure. Over time, this concept solidified into the Beowulf architecture.

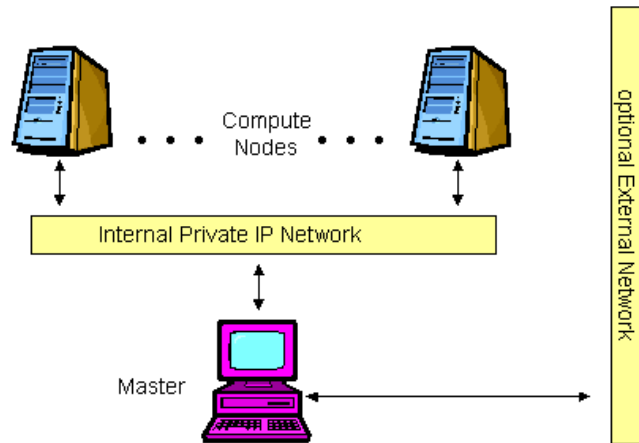


Figure 1-2. A Basic Beowulf Cluster

For a cluster to be properly termed a "Beowulf", it must adhere to the "Beowulf philosophy", which requires:

- Scalable performance
- The use of commodity off-the-shelf (*COTS*) hardware
- The use of an open-source operating system, typically Linux

Use of commodity hardware allows Beowulf clusters to take advantage of the economies of scale in the larger computing markets. In this way, Beowulf clusters can always take advantage of the fastest processors developed for high-end workstations, the fastest networks developed for backbone network providers, and so on. The progress of Beowulf clustering technology is not governed by any one company's development decisions, resources, or schedule.

First-Generation Beowulf Clusters

The original Beowulf software environments were implemented as downloadable add-ons to commercially-available Linux distributions. These distributions included all of the software needed for a networked workstation: the kernel, various utilities, and many add-on packages. The downloadable Beowulf add-ons included several programming environments and development libraries as individually-installable packages.

With this first-generation Beowulf scheme, every node in the cluster required a full Linux installation and was responsible for running its own copy of the kernel. This requirement created many administrative headaches for the maintainers of Beowulf-class clusters. For this reason, early Beowulf systems tended to be deployed by the software application developers themselves (and required detailed knowledge to install and use). Scyld ClusterWare reduces and/or eliminates these and other problems associated with the original Beowulf-class clusters.

Scyld ClusterWare: A New Generation of Beowulf

Scyld ClusterWare streamlines the process of configuring, administering, running, and maintaining a Beowulf-class cluster computer. It was developed with the goal of providing the software infrastructure for commercial production cluster solutions.

Scyld ClusterWare was designed with the differences between master and compute nodes in mind; it runs only the appropriate software components on each compute node. Instead of having a collection of computers each running its own fully-installed operating system, Scyld creates one large distributed computer. The user of a Scyld cluster will never log into one of the compute nodes nor worry about which compute node is which. To the user, the master node *is* the computer, and the compute nodes appear merely as attached processors capable of providing computing resources.

With Scyld ClusterWare, the cluster appears to the user as a single computer. Specifically,

- The compute nodes appear as attached processor and memory resources
- All jobs start on the master node, and are migrated to the compute nodes at runtime
- All compute nodes are managed and administered collectively via the master node

The Scyld ClusterWare architecture simplifies cluster setup and node integration, requires minimal system administration, provides tools for easy administration where necessary, and increases cluster reliability through seamless scalability. In addition to its technical advances, Scyld ClusterWare provides a standard, stable, commercially-supported platform for deploying advanced clustering systems. See the next section for a technical summary of Scyld ClusterWare.

Scyld ClusterWare Technical Summary

Scyld ClusterWare presents a more uniform system view of the entire cluster to both users and applications through extensions to the kernel. A guiding principle of these extensions is to have little increase in both kernel size and complexity and, more importantly, negligible impact on individual processor performance.

In addition to its enhanced Linux kernel, Scyld ClusterWare includes libraries and utilities specifically improved for high-performance computing applications. For information on the Scyld libraries, see the *Reference Guide*. Information on using the Scyld utilities to run and monitor jobs is provided in Chapter 2 and Chapter 3. If you need to use the Scyld utilities to configure and administer your cluster, see the *Administrator's Guide*.

Top-Level Features of Scyld ClusterWare

The following list summarizes the top-level features of Scyld ClusterWare.

Security and Authentication

With Scyld ClusterWare, the master node is a single point of security administration and authentication. The authentication envelope is drawn around the entire cluster and its private network. This obviates the need to manage copies or caches of credentials on compute nodes or to add the overhead of networked authentication. Scyld ClusterWare provides simple permissions on compute nodes, similar to Unix file permissions, allowing their use to be administered without additional overhead.

Easy Installation

Scyld ClusterWare is designed to augment a full Linux distribution, such as Red Hat Enterprise Linux (RHEL) or CentOS. The installer used to initiate the installation on the master node is provided on an auto-run CD-ROM. You can install from scratch and have a running Linux HPC cluster in less than an hour. See the *Installation Guide* for full details.

Install Once, Execute Everywhere

A full installation of Scyld ClusterWare is required only on the master node. Compute nodes are provisioned from the master node during their boot process, and they dynamically cache any additional parts of the system during process migration or at first reference.

Single System Image

Scyld ClusterWare makes a cluster appear as a multi-processor parallel computer. The master node maintains (and presents to the user) a single process space for the entire cluster, known as the **BProc** Distributed Process Space. **BProc** is described briefly later in this chapter, and more details are provided in the *Administrator's Guide*.

Execution Time Process Migration

Scyld ClusterWare stores applications on the master node. At execution time, **BProc** migrates processes from the master to the compute nodes. This approach virtually eliminates both the risk of *version skew* and the need for hard disks on the compute nodes. More information is provided in the section on process space migration later in this chapter. Also refer to the **BProc** discussion in the *Administrator's Guide*.

Seamless Cluster Scalability

Scyld ClusterWare seamlessly supports the dynamic addition and deletion of compute nodes without modification to existing source code or configuration files. See the chapter on the **BeoSetup** utility in the *Administrator's Guide*.

Administration Tools

Scyld ClusterWare includes simplified tools for performing cluster administration and maintenance. Both graphical user interface (GUI) and command line interface (CLI) tools are supplied. See the *Administrator's Guide* for more information.

Web-Based Administration Tools

Scyld ClusterWare includes web-based tools for remote administration, job execution, and monitoring of the cluster. See the *Administrator's Guide* for more information.

Additional Features

Additional features of Scyld ClusterWare include support for cluster power management (IPMI and Wake-on-LAN, easily extensible to other out-of-band management protocols); runtime and development support for MPI and PVM; and support for the LFS and NFS3 file systems.

Fully-Supported

Scyld ClusterWare is fully-supported by Penguin Computing, Inc.

Process Space Migration Technology

Scyld ClusterWare is able to provide a single system image through its use of the **BProc** Distributed Process Space, the Beowulf process space management kernel enhancement. **BProc** enables the processes running on compute nodes to be visible and managed on the master node. All processes appear in the master node's process table, from which they are migrated to the appropriate compute node by **BProc**. Both process parent-child relationships and Unix job-control information are maintained with the migrated jobs. The `stdout` and `stderr` streams are redirected to the user's `ssh` or terminal session on the master node across the network.

The **BProc** mechanism is one of the primary features that makes Scyld ClusterWare different from traditional Beowulf clusters. For more information, see the system design description in the *Administrator's Guide*.

Compute Node Provisioning

Scyld ClusterWare utilizes light-weight provisioning of compute nodes from the master node's kernel and Linux distribution. For Scyld Series 30 and Scyld ClusterWare HPC, PXE is the supported method for booting nodes into the cluster; the 2-phase boot sequence of earlier Scyld distributions is no longer used.

The master node is the DHCP server serving the cluster private network. PXE booting across the private network ensures that the compute node boot package is version-synchronized for all nodes within the cluster. This boot package consists of the kernel, `initrd`, and `rootfs`. If desired, the boot package can be customized per node in the Beowulf configuration file `/etc/beowulf/config`, which also includes the kernel command line parameters for the boot package.

For a detailed description of the compute node boot procedure, see the system design description in the *Administrator's Guide*. Also refer to the chapter on compute node boot options in that document.

Compute Node Categories

Compute nodes seen by the master over the private network are classified into one of three categories by the master node, as follows:

- *Unknown* — A node not formally recognized by the cluster as being either a *Configured* or *Ignored* node. When bringing a new compute node online, or after replacing an existing node's network interface card, the node will be classified as *unknown*.
- *Ignored* — Nodes which, for one reason or another, you'd like the master node to ignore. These are not considered part of the cluster, nor will they receive a response from the master node during their boot process.
- *Configured* — Those nodes listed in the cluster configuration file using the "node" tag. These are formally part of the cluster, recognized as such by the master node, and used as computational resources by the cluster.

For more information on compute node categories, see the system design description in the *Administrator's Guide*.

Compute Node States

BProc maintains the current condition or "node state" of each configured compute node in the cluster. The compute node states are defined as follows:

- *down* — Not communicating with the master, and its previous state was either *down*, *up*, *error*, *unavailable*, or *boot*.
- *unavailable* — Node has been marked *unavailable* or "off-line" by the cluster administrator; typically used when performing maintenance activities. The node is useable only by the user *root*.
- *error* — Node encountered an error during its initialization; this state may also be set manually by the cluster administrator. The node is useable only by the user *root*.
- *up* — Node completed its initialization without error; node is online and operating normally. This is the only state in which non-*root* users may access the node.
- *reboot* — Node has been commanded to reboot itself; node will remain in this state until it reaches the *boot* state, as described below.
- *halt* — Node has been commanded to halt itself; node will remain in this state until it is reset (or powered back on) and reaches the *boot* state, as described below.
- *poweroff* — Node has been commanded to power itself off; node will remain in this state until it is powered back on and reaches the *boot* state, as described below.
- *boot* — Node has completed its *stage 2* boot but is still initializing. After the node finishes booting, its next state will be either *up* or *error*.

For more information on compute node states, see the system design description in the *Administrator's Guide*.

Major Software Components

The following is a list of the major software components included with Scyld ClusterWare HPC. For more information, see the relevant sections of the Scyld ClusterWare HPC documentation set, including the *Installation Guide*, *Administrator's Guide*, *User's Guide*, *Reference Guide*, and *Programmer's Guide*.

- **BProc** — The process migration technology; an integral part of Scyld ClusterWare.
- **BeoSetup** — A GUI for configuring the cluster.
- **BeoStatus** — A GUI for monitoring cluster status.
- **beostat** — A text-based tool for monitoring cluster status.
- **beoboot** — A set of utilities for booting the compute nodes.
- **beofdisk** — A utility for remote partitioning of hard disks on the compute nodes.
- **beoserv** — The cluster's DHCP, PXE and dynamic provisioning server; it responds to compute nodes and serves the boot image.
- **BPmaster** — The **BProc** master daemon; it runs on the master node.
- **BPslave** — The **BProc** compute daemon; it runs on each of the compute nodes.
- **bpstat** — A **BProc** utility that reports status information for all nodes in the cluster.
- **bpctl** — A **BProc** command line interface for controlling the nodes.
- **bpsh** — A **BProc** utility intended as a replacement for **rsh** (remote shell).
- **bpcp** — A **BProc** utility for copying files between nodes, similar to **rcp** (remote copy).

- **MPI** — The Message Passing Interface, optimized for use with Scyld ClusterWare.
- **PVM** — The Parallel Virtual Machine, optimized for use with Scyld ClusterWare.
- **mpprun** — A parallel job-creation package for Scyld ClusterWare.

Typical Applications of Scyld ClusterWare

Scyld clustering provides a facile solution for anyone executing jobs that involve either a large number of computations or large amounts of data (or both). It is ideal for both large, monolithic, parallel jobs and for many normal-sized jobs run many times (such as Monte Carlo type analysis).

The increased computational resource needs of modern applications are frequently being met by Scyld clusters in a number of domains, including:

- *Computationally-Intensive Activities* — Optimization problems, stock trend analysis, financial analysis, complex pattern matching, medical research, genetics research, image rendering
- *Scientific Computing / Research* — Engineering simulations, 3D-modeling, finite element analysis, computational fluid dynamics, computational drug development, seismic data analysis, PCB / ASIC routing
- *Large-Scale Data Processing* — Data mining, complex data searches and results generation, manipulating large amounts of data, data archival and sorting
- *Web / Internet Uses* — Web farms, application serving, transaction serving, data serving

These types of jobs can be performed many times faster on a Scyld cluster than on a single computer. Increased speed depends on the application code, the number of nodes in the cluster, and the type of equipment used in the cluster. All of these can be easily tailored and optimized to suit the needs of your applications.

Chapter 2. Interacting With the System

This chapter discusses how to verify the availability of the nodes in your cluster, how to monitor node status, how to issue commands and copy data to the compute nodes, and how to monitor and control processes. For information on running programs across the cluster, see Chapter 3.

Verifying the Availability of Nodes

In order to use a Scyld cluster for computation, at least one node must be available or *up*. Thus, the first priority when interacting with a cluster is ascertaining the availability of nodes. Unlike traditional Beowulf clusters, Scyld ClusterWare provides rich reporting about the availability of the nodes.

You can use either the **BeoStatus** GUI tool or the **bpstat** command to determine the availability of nodes in your cluster. These tools, which can also be used to monitor node status, are described in the next section.

If fewer nodes are *up* than you think should be, or some nodes report an error, check with your Cluster Administrator.

Monitoring Node Status

You can monitor the status of nodes in your cluster with the **BeoStatus** GUI tool or with either of two command line tools, **bpstat** and **beostat**. These tools are described in the sections that follow. Also see the *Reference Guide* for information on the various options and flags supported for these tools.

The BeoStatus GUI Tool

The **BeoStatus** graphical user interface (GUI) tool is the best way to check the status of the cluster, including which nodes are available or *up*. There are two ways to open the **BeoStatus** GUI as a Gnome X window, as follows.

Click the **BeoStatus** icon in the tool tray or in the applications pulldown.



Alternatively, type the command **beostat** in a terminal window on the master node; you do not need to be a privileged user to use this command.

The default **BeoStatus** GUI mode is a tabular format known as the "Classic" display (shown in the following figure). You can select different display options from the **Mode** menu.

Node	Up	State	CPU 0	CPU 1	Memory	Swap	Disk	Network
-1	✓	up	0%	0%	532/4022 MB (13%)	0/1992 MB (0%)	25806/179829 MB (14%)	5 kBps
0	✓	up	0%	0%	24/4021 MB (0%)	None	58/2010 MB (2%)	0 kBps
1	✓	up	0%	0%	26/4021 MB (0%)	None	58/2010 MB (2%)	0 kBps
2	✓	up	0%	0%	41/4021 MB (1%)	None	57/2010 MB (2%)	0 kBps
3	✓	up	0%	0%	19/4021 MB (0%)	None	57/2010 MB (2%)	0 kBps
4	✓	up	0%	0%	48/4021 MB (1%)	None	58/2010 MB (2%)	0 kBps
5	✓	up	0%	0%	49/4021 MB (1%)	None	58/2010 MB (2%)	0 kBps
6	✗	down	53%	80%	59/4021 MB (1%)	None	58/2010 MB (2%)	4854 kBps

Figure 2-1. BeoStatus in the "Classic" Display Mode

BeoStatus Node Information

Each row in the **BeoStatus** display reports information for a single node, including the following:

- *Node* — The node's assigned node number, starting at zero. Node -1, if shown, is the master node. The total number of node entries shown is set by the "iprange" or "nodes" keywords in the file `/etc/beowulf/config`, rather than the number of detected nodes. The entry for an inactive node displays the last reported data in a grayed-out row.
- *Up* — A graphical representation of the node's status. A green checkmark is shown if the node is up and available. Otherwise, a red "X" is shown.
- *State* — The node's last known state. This should agree with the state reported by both the **bpstat** command and in the **BeoSetup** window.
- *CPU "X"* — The CPU loads for the node's processors; at minimum, this indicates the CPU load for the first processor in each node. Since it is possible to mix uni-processor and multi-processor machines in a Scyld cluster, the number of CPU load columns is equal to the maximum number of processors for any node in your cluster. The label "N/A" will be shown for nodes with less than the maximum number of processors.
- *Memory* — The node's current memory usage.
- *Swap* — The node's current swap space (virtual memory) usage.
- *Disk* — The node's hard disk usage. If a RAM disk is used, the maximum value shown is one-half the amount of physical memory. As the RAM disk competes with the kernel and application processes for memory, not all the RAM may be available.
- *Network* — The node's network bandwidth usage. The total amount of bandwidth available is the sum of all network interfaces for that node.

BeoStatus Update Intervals

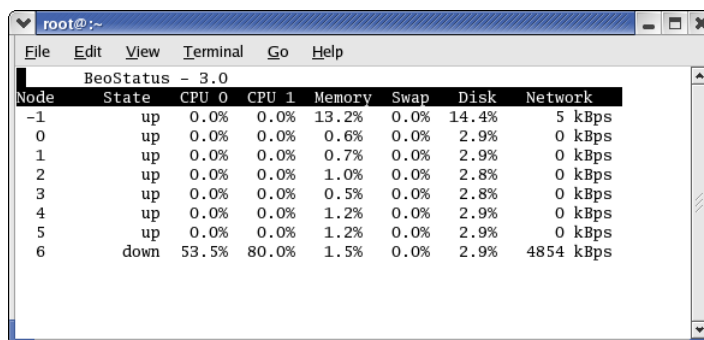
Once running, **BeoStatus** is non-interactive; the user simply monitors the reported information. The display is updated at 4-second intervals by default. You can modify this default using the command **beostatus -u secs** (where *secs* is the number of seconds) in a terminal window or an **ssh** session to the master node with X-forwarding enabled.

Tip: Each update places load on the master and compute nodes, as well as the interconnection network. Too-frequent updates can degrade the overall system performance.

BeoStatus in Text Mode

In environments where use of the Gnome X window system is undesirable or impractical, such as when accessing the master node through a slow remote network connection, you can view the status of the cluster as curses text output (shown in the following figure). To do this, enter the command **beostatus -c** in a terminal window on the master node or an **ssh** session to the master node.

BeoStatus in text mode reports the same node information as reported by the "Classic" display, except for the graphical indicator of node *up* (green checkmark) or node *down* (red X). The data in the text display is updated at 4-second intervals by default.



Node	State	CPU 0	CPU 1	Memory	Swap	Disk	Network
-1	up	0.0%	0.0%	13.2%	0.0%	14.4%	5 kBps
0	up	0.0%	0.0%	0.6%	0.0%	2.9%	0 kBps
1	up	0.0%	0.0%	0.7%	0.0%	2.9%	0 kBps
2	up	0.0%	0.0%	1.0%	0.0%	2.8%	0 kBps
3	up	0.0%	0.0%	0.5%	0.0%	2.8%	0 kBps
4	up	0.0%	0.0%	1.2%	0.0%	2.9%	0 kBps
5	up	0.0%	0.0%	1.2%	0.0%	2.9%	0 kBps
6	down	53.5%	80.0%	1.5%	0.0%	2.9%	4854 kBps

Figure 2-2. BeoStatus in Text Mode

The bpstat Command Line Tool

You can also check node status with the **bpstat** command. When run at a shell prompt on the master node without options, **bpstat** prints out a listing of all nodes in the cluster and their current status. You do not need to be a privileged user to use this command.

Following is an example of the outputs from **bpstat** for a cluster with 10 compute nodes.

```
[user@cluster user] $ bpstat
Node(s)      Status      Mode          User          Group
5-9          down       -----      root          root
4            up         ---x---x---x any           any
0-3          up         ---x---x---x root          root
```

bpstat will show one of the following indicators in the "Status" column:

- A node marked *up* is available to run jobs. This status is the equivalent of the green checkmark in the **BeoStatus** GUI.

- Nodes that have not yet been configured are marked as *down*. This status is the equivalent of the red X in the **BeoStatus** GUI.
- Nodes currently booting are temporarily shown with a status of *boot*. Wait 10-15 seconds and try again.
- The "error" status indicates a node initialization problem. Check with your Cluster Administrator.

For additional information on **bpstat**, see the section on monitoring and controlling processes later in this chapter. Also see the *Reference Guide* for details on using **bpstat** and its command line options.

The beostat Command Line Tool

You can use the **beostat** command to display raw status data for cluster nodes. When run at a shell prompt on the master node without options, **beostat** prints out a listing of stats for all nodes in the cluster, including the master node. You do not need to be a privileged user to use this command.

The following example shows the **beostat** output for the master node and one compute node:

```
[user@cluster user] $ beostat
model          : 5
model name     : AMD Opteron(tm) Processor 248
stepping      : 10
cpu MHz       : 2211.352
cache size    : 1024 KB
fdiv_bug     : no
hlt_bug      : no
sep_bug      : no
f00f_bug     : no
coma_bug     : no
fpu          : yes
fpu_exception : yes
cpuid level   : 1
wp           : yes
bogomips     : 4422.05

*** /proc/meminfo *** Sun Sep 17 10:46:33 2006
      total:      used:      free:  shared: buffers:  cached:
Mem:  4217454592 318734336 3898720256      0 60628992      0
Swap: 2089209856      0 2089209856
MemTotal:  4118608 kB
MemFree:   3807344 kB
MemShared:      0 kB
Buffers:    59208 kB
Cached:      0 kB
SwapTotal: 2040244 kB
SwapFree:   2040244 kB

*** /proc/loadavg *** Sun Sep 17 10:46:33 2006
3.00 2.28 1.09 178/178 0

*** /proc/net/dev *** Sun Sep 17 10:46:33 2006
Inter-|   Receive                                     |   Transmit
face |bytes  packets errs drop fifo frame compressed multicast|bytes  packets errs drop fifo co
eth0:85209660 615362      0      0      0      0      0      0      0      0 703311290 559376
eth1:4576500575 13507271      0      0      0      0      0      0      0      0 9430333982 13220730
```

```

sit0:      0      0      0      0      0      0      0      0      0      0      0

*** /proc/stat ***
cpu0 15040 0 466102 25629625      Sun Sep 17 10:46:33 2006
cpu1 17404 0 1328475 24751544      Sun Sep 17 10:46:33 2006

*** statfs ("/") *** Sun Sep 17 10:46:33 2006
path:      /
f_type:    0xef53
f_bsize:   4096
f_blocks:  48500104
f_bfree:   41439879
f_bavail:  38976212
f_files:   24641536
f_ffree:   24191647
f_fsid:    000000 000000
f_namelen: 255

===== Node: .0 (index 0) =====

*** /proc/cpuinfo *** Sun Sep 17 10:46:34 2006
num processors : 2
vendor_id      : AuthenticAMD
cpu family    : 15
model         : 5
model name    : AMD Opteron(tm) Processor 248
stepping     : 10
cpu MHz      : 2211.386
cache size   : 1024 KB
fdiv_bug     : no
hlt_bug      : no
sep_bug      : no
f00f_bug    : no
coma_bug     : no
fpu          : yes
fpu_exception : yes
cpuid level  : 1
wp           : yes
bogomips     : 4422.04

*** /proc/meminfo *** Sun Sep 17 10:46:34 2006
      total:      used:      free:  shared: buffers:  cached:
Mem:  4216762368 99139584 4117622784      0      0      0
Swap:      0      0      0
MemTotal:  4117932 kB
MemFree:   4021116 kB
MemShared:      0 kB
Buffers:    0 kB
Cached:     0 kB
SwapTotal:   0 kB
SwapFree:   0 kB

*** /proc/loadavg *** Sun Sep 17 10:46:34 2006
0.99 0.75 0.54 36/36 0

```

```
*** /proc/net/dev *** Sun Sep 17 10:46:34 2006
Inter-|   Receive                                          |   Transmit
face |bytes    packets errs drop fifo frame compressed multicast|bytes    packets errs drop fifo co
eth0:312353878  430256      0      0      0      0      0      0      0      0 246128779  541105
eth1:      0      0      0      0      0      0      0      0      0      0      0      0

*** /proc/stat ***
cpu0 29984 0 1629 15340009          Sun Sep 17 10:46:34 2006
cpu1 189495 0 11131 15170565       Sun Sep 17 10:46:34 2006

*** statfs ("/") *** Sun Sep 17 10:46:34 2006
path:      /
f_type:    0x1021994
f_bsize:   4096
f_blocks:  514741
f_bfree:   492803
f_bavail:  492803
f_files:   514741
f_ffree:   514588
f_fsid:    000000 000000
f_namelen: 255
```

The *Reference Guide* provides details for using **beostat** and its command line options.

Issuing Commands

Commands on the Master Node

When you log into the cluster, you are actually logging into the master node, and the commands you enter on the command line will execute on the master node. The only exception is when you use special commands for interacting with the compute nodes, as described in the next section.

Commands on the Compute Node

Scyld ClusterWare provides the **bpsh** command for running jobs on the compute nodes. **bpsh** is a replacement for the traditional Unix utility **rsh**, used to run a job on a remote computer. Like **rsh**, the **bpsh** arguments are the node on which to run the command and the command. **bpsh** allows you to run a command on more than one node without having to type the command once for each node, but it doesn't provide an interactive shell on the remote node like **rsh** does.

bpsh is primarily intended for running utilities and maintenance tasks on a single node or a range of nodes, rather than for running parallel programs. For information on running parallel programs with Scyld ClusterWare, see Chapter 3.

bpsh provides a convenient yet powerful interface for manipulating all (or a subset of) the cluster's nodes simultaneously. **bpsh** provides you the flexibility to access a compute node individually, but removes the requirement to access each node individually when a collective operation is desired. A number of examples and options are discussed in the sections that follow. For a complete reference to all the options available for **bpsh**, see the *Reference Guide*.

Examples for Using `bpsh`

Example 2-1. Checking for a File

You can use `bpsh` to check for specific files on a compute node. For example, to check for a file named `output` in the `/tmp` directory of node 3, you would run the following command on the master node:

```
[user@cluster user] $ bpsh 3 ls /tmp/output
```

The command output would appear on the master node terminal where you issued the command.

Example 2-2. Running a Command on a Range of Nodes

You can run the same command on a range of nodes using `bpsh`. For example, to check for a file named `output` in the `/tmp` directory of nodes 3 through 5, you would run the following command on the master node:

```
[user@cluster user] $ bpsh 3,4,5 ls /tmp/output
```

Example 2-3. Running a Command on All Available Nodes

Use the `-a` flag to indicate to `bpsh` that you wish to run a command on all available nodes. For example, to check for a file named `output` in the `/tmp` directory of all nodes currently active in your cluster, you would run the following command on the master node:

```
[user@cluster user] $ bpsh -a ls /tmp/output
```

Note that when using the `-a` flag, the results are sorted by the response speed of the compute nodes, and are returned without node identifiers. Because this command will produce output for every currently active node, the output may be hard to read if you have a large cluster. For example, if you ran the above command on a 64-node cluster in which half of the nodes have the file being requested, the results returned would be 32 lines of `/tmp/output` and another 32 lines of `ls: /tmp/output: no such file or directory`. Without node identifiers, it is impossible to ascertain the existence of the target file on a particular node.

See the next section for `bpsh` options that enable you to format the results for easier reading.

Formatting `bpsh` Output

The `bpsh` command has a number of options for formatting its output to make it more useful for the user, including the following:

- The `-L` option makes `bpsh` wait for a full line from a compute node before it prints out the line. Without this option, the output from your command could include half a line from node 0 with a line from node 1 tacked onto the end, then followed by the rest of the line from node 0.
- The `-p` option prefixes each line of output with the node number of the compute node that produced it. This option causes the functionality for `-L` to be used, even if not explicitly specified.

- The **-s** option forces the output of each compute node to be printed in sorted numerical order, rather than by the response speed of the compute nodes. With this option, all the output for node 0 will appear before any of the output for node 1. To add a divider between the output from each node, use the **-d** option.
- Using **-d** generates a divider between the output from each node. This option causes the functionality for **-s** to be used, even if not explicitly specified.

For example, if you run the command **bpsh -a -d -p ls /tmp/output** on an 8-node cluster, the output would make it clear which nodes do and do not have the file `output` in the `/tmp` directory, for example:

```
0 -----
  /tmp/output
1 -----
1: ls: /tmp/output: No such file or directory
2 -----
2: ls: /tmp/output: No such file or directory
3 -----
3: /tmp/output
4 -----
4: /tmp/output
5 -----
5: /tmp/output
6 -----
6: ls: /tmp/output: No such file or directory
7 -----
7: ls: /tmp/output: No such file or directory
```

bpsh and Shell Interaction

Special shell features, such as piping and input/output redirection, are available to advanced users. This section provides several examples of shell interaction, using the following conventions:

- The command running will be **cmda**.
- If it is piped to anything, it will be piped to **cmdb**.
- If an input file is used, it will be `/tmp/input`.
- If an output file is used, it will be `/tmp/output`.
- The node used will always be node 0.

Example 2-4. Command on Compute Node, Output on Master Node

The easiest case is running a command on a compute node and doing something with its output on the master node, or giving it input from the master. Following are a few examples:

```
[user@cluster user] $ bpsh 0 cmda | cmdb
[user@cluster user] $ bpsh 0 cmda > /tmp/output
[user@cluster user] $ bpsh 0 cmda < /tmp/input
```

Example 2-5. Command on Compute Node, Output on Compute Node

A bit more complex situation is to run the command on the compute node and do something with its input (or output) on that same compute node. There are two ways to accomplish this.

The first solution requires that all the programs you run be on the compute node. For this to work, you must first copy the **cmda** and **cmdb** executable binaries to the compute node. Then you would use the following commands:

```
[user@cluster user] $ bpsb 0 sh -c "cmda | cmdb"
[user@cluster user] $ bpsb 0 sh -c "cmda > /tmp/output"
[user@cluster user] $ bpsb 0 sh -c "cmda < /tmp/input"
```

The second solution doesn't require any of the programs to be on the compute node. However, it uses a lot of network bandwidth as it takes the output and sends it to the master node, then sends it right back to the compute node. The appropriate commands are as follows:

```
[user@cluster user] $ bpsb 0 cmda | bpsb 0 cmdb
[user@cluster user] $ bpsb 0 cmda | bpsb 0 dd of=/tmp/output
[user@cluster user] $ bpsb 0 cat /tmp/input | bpsb 0 cmda
```

Example 2-6. Command on Master Node, Output on Compute Node

You can also run a command on the master node and do something with its input or output on the compute nodes. The appropriate commands are as follows:

```
[user@cluster user] $ cmda | bpsb 0 cmdb
[user@cluster user] $ cmda | bpsb 0 dd of=/tmp/output
[user@cluster user] $ bpsb 0 cat /tmp/input | cmda
```

Copying Data to the Compute Nodes

There are several ways to get data from the master node to the compute nodes. This section describes using NFS to share data, using the Scyld ClusterWare command **bpcp** to copy data, and using programmatic methods for data transfer.

Sharing Data via NFS

The easiest way to transfer data to the compute nodes is via NFS. All files in your `/home` directory are shared by default to all compute nodes via NFS. Opening an NFS-shared file on a compute node will, in fact, open the file on the master node; no actual copying takes place.

Copying Data via **bpcp**

To copy a file, rather than changing the original across the network, you can use the **bpcp** command. This works much like the standard Unix file-copying command **cp**, in that you pass it a file to copy as one argument and the destination as the next argument. Like the Unix **scp**, the file paths may be qualified by a computer host name.

With **bpcp**, you can indicate the node number for the source file, destination file, or both. To do this, prepend the node number with a colon before the file name, to specify that the file is on that node or should be copied to that node. For example, to copy the file `/tmp/foo` to the same location on node 1, you would use the following command:

```
[user@cluster user] $ bpcp /tmp/foo 1:/tmp/foo
```

Programmatic Data Transfer

The third method for transferring data is to do it programmatically. This is a bit more complex than the methods described in the previous section, and will only be described here only conceptually.

If you are using an MPI job, you can have your Rank 0 process on the master node read in the data, then use MPI's message passing capabilities to send the data over to a compute node.

If you are writing a program that uses **BProc** functions directly, you can have the process first read the data while it is on the master node. When the process is moved over to the compute node, it should still be able to access the data read in while on the master node.

Data Transfer by Migration

Another programmatic method for file transfer is to read a file into memory prior to calling **BProc** to migrate the process to another node. This technique is especially useful for parameter and configuration files, or files containing the intermediate state of a computation. See the *Reference Guide* for a description of the **BProc** system calls.

Monitoring and Controlling Processes

One of the features of Scyld ClusterWare that isn't provided in traditional Beowulf clusters is the **BProc** Distributed Process Space. **BProc** presents a single unified process space for the entire cluster, run from the master node, where you can see and control jobs running on the compute nodes. This process space allows you to use standard Unix tools, such as **top**, **ps**, and **kill**. See the *Administrator's Guide* for more details on **BProc**.

Scyld ClusterWare also includes a tool called **bpstat** that can be used to determine which node is running a process. Using the command option **bpstat -p** will list all processes currently running by processID (PID), with the number of the node running each process. The following output is an example:

```
[user@cluster user] $ bpstat -p
  PID      Node
  6301      0
  6302      1
  6303      0
  6304      2
  6305      1
  6313      2
  6314      3
  6321      3
```


Using the command option **bpstat -P** (with an uppercase "P" instead of a lowercase "p") tells **bpstat** to take the output of the **ps** and reformat it, pre-pending a column showing the node number. The following two examples show the difference in the outputs from **ps** and from **bpstat -P**.

Example output from **ps**:

```
[user@cluster user] $ ps xf
PID  TTY      STAT   TIME COMMAND
6503 pts/2    S       0:00 bash
6665 pts/2    R       0:00 ps xf
6471 pts/3    S       0:00 bash
6538 pts/3    S       0:00 /bin/sh /usr/bin/linpack
6553 pts/3    S       0:00  \_ /bin/sh /usr/bin/mpirun -np 5 /tmp/xhpl
6654 pts/3    R       0:03    \_ /tmp/xhpl -p4pg /tmp/PI6553 -p4wd /tmp
6655 pts/3    S       0:00        \_ /tmp/xhpl -p4pg /tmp/PI6553 -p4wd /tmp
6656 pts/3    RW      0:01        \_ [xhpl]
6658 pts/3    SW      0:00        |  \_ [xhpl]
6657 pts/3    RW      0:01        \_ [xhpl]
6660 pts/3    SW      0:00        |  \_ [xhpl]
6659 pts/3    RW      0:01        \_ [xhpl]
6662 pts/3    SW      0:00        |  \_ [xhpl]
6661 pts/3    SW      0:00        \_ [xhpl]
6663 pts/3    SW      0:00        \_ [xhpl]
```

Example of the same **ps** output when run through **bpstat -P** instead:

```
[user@cluster user] $ ps xf | bpstat -P
NODE  PID  TTY      STAT   TIME COMMAND
      6503 pts/2    S       0:00 bash
      6666 pts/2    R       0:00 ps xf
      6667 pts/2    R       0:00 bpstat -P
      6471 pts/3    S       0:00 bash
      6538 pts/3    S       0:00 /bin/sh /usr/bin/linpack
      6553 pts/3    S       0:00  \_ /bin/sh /usr/bin/mpirun -np 5 /tmp/xhpl
      6654 pts/3    R       0:06    \_ /tmp/xhpl -p4pg /tmp/PI6553 -p4wd /tmp
      6655 pts/3    S       0:00        \_ /tmp/xhpl -p4pg /tmp/PI6553 -p4wd /tmp
0     6656 pts/3    RW      0:06        \_ [xhpl]
0     6658 pts/3    SW      0:00        |  \_ [xhpl]
1     6657 pts/3    RW      0:06        \_ [xhpl]
1     6660 pts/3    SW      0:00        |  \_ [xhpl]
2     6659 pts/3    RW      0:06        \_ [xhpl]
2     6662 pts/3    SW      0:00        |  \_ [xhpl]
3     6661 pts/3    SW      0:00        \_ [xhpl]
3     6663 pts/3    SW      0:00        \_ [xhpl]
```

For additional information on **bpstat**, see the section on monitoring node status earlier in this chapter. For information on the **bpstat** command line options, see the *Reference Guide*.

Chapter 3. Running Programs

This chapter describes how to run both serial and parallel jobs with Scyld ClusterWare, and how to monitor the status of the cluster once your applications are running. It begins with a brief discussion of program execution concepts, including some examples. The discussion then covers running programs that aren't parallelized, running parallel programs (including MPI-aware and PVM-aware programs), running serial programs in parallel, job batching, and file systems.

Program Execution Concepts

This section compares program execution on a stand-alone computer and a Scyld cluster. It also discusses the differences between running programs on a traditional Beowulf cluster and a Scyld cluster. Finally, it provides some examples of program execution on a Scyld cluster.

Stand-Alone Computer vs. Scyld Cluster

On a stand-alone computer running Linux, Unix, and most other operating systems, executing a program is a very simple process. For example, to generate a list of the files in the current working directory, you open a terminal window and type the command `ls` followed by the `[return]` key. Typing the `[return]` key causes the command shell — a program that listens to and interprets commands entered in the terminal window — to start the `ls` program (stored at `/bin/ls`). The output is captured and directed to the standard output stream, which also appears in the same window where you typed the command.

A Scyld cluster isn't simply a group of networked stand-alone computers. Only the master node resembles the computing system with which you are familiar. The compute nodes have only the minimal software components necessary to support an application initiated from the master node. So for instance, running the `ls` command on the master node causes the same series of actions as described above for a stand-alone computer, and the output is for the master node only.

However, running `ls` on a compute node involves a very different series of actions. Remember that a Scyld cluster has no resident applications on the compute nodes; applications reside only on the master node. So for instance, to run the `ls` command on compute node 1, you would enter the command `bpsh 1 ls` on the master node. This command sends `ls` to compute node 1 via Scyld's **BProc** software, and the output stream is directed to the terminal window on the master node, where you typed the command.

Some brief examples of program execution are provided in the last section of this chapter. Both **BProc** and `bpsh` are covered in more detail in the *Administrator's Guide*.

Traditional Beowulf Cluster vs. Scyld Cluster

A job on a Beowulf cluster is actually a collection of processes running on the compute nodes. In traditional clusters of computers, and even on earlier Beowulf clusters, getting these processes started and running together was a complicated task. Typically, the cluster administrator would need to do all of the following:

- Ensure that the user had an account on all the target nodes, either manually or via a script.
- Ensure that the user could spawn jobs on all the target nodes. This typically entailed configuring a `hosts.allow` file on each machine, creating a specialized PAM module (a Linux authentication mechanism), or creating a server daemon on each node to spawn jobs on the user's behalf.
- Copy the program binary to each node, either manually, with a script, or through a network file system.
- Ensure that each node had available identical copies of all the dependencies (such as libraries) needed to run the program.

- Provide knowledge of the state of the system to the application manually, through a configuration file, or through some add-on scheduling software.

With Scyld ClusterWare, most of these steps are removed. Jobs are started on the master node and are migrated out to the compute nodes via **BProc**. A cluster architecture where jobs may be initiated only from the master node via **BProc** provides the following advantages:

- Users no longer need accounts on remote nodes.
- Users no longer need authorization to spawn jobs on remote nodes.
- Neither binaries nor libraries need to be available on the remote nodes.
- The **BProc** system provides a consistent view of all jobs running on the system.

With all these complications removed, program execution on the compute nodes becomes a simple matter of letting **BProc** know about your job when you start it. The method for doing so depends on whether you are launching a parallel program (for example, an MPI job or PVM job) or any other kind of program. See the sections on running parallel programs and running non-parallelized programs later in this chapter.

Program Execution Examples

This section provides a few examples of program execution with Scyld ClusterWare. Additional examples are provided in the sections on running parallel programs and running non-parallelized programs later in this chapter.

Example 3-1. Directed Execution with **bpsh**

In the directed execution mode, the user explicitly defines which node (or nodes) will run a particular job. This mode is invoked using the **bpsh** command, the ClusterWare shell command analogous in functionality to both the **rsh** (remote shell) and **ssh** (secure shell) commands. Following are two examples of using **bpsh**.

The first example runs **hostname** on compute node 0 and writes the output back from the node to the user's screen:

```
[user@cluster user] $ bpsh 0 /bin/hostname
n0
```

If **/bin** is in the user's **\$PATH**, then the **bpsh** does not need the full pathname:

```
[user@cluster user] $ bpsh 0 hostname
n0
```

The second example runs the **/usr/bin/uptime** utility on node 1. Assuming **/usr/bin** is in the user's **\$PATH**:

```
[user@cluster user] $ bpsh 1 uptime
12:56:44 up 4:57, 5 users, load average: 0.06, 0.09, 0.03
```

Example 3-2. Dynamic Execution with **beorun** and **mpprun**

In the dynamic execution mode, Scyld decides which node is the most capable of executing the job at that moment in time. Scyld includes two parallel execution tools that dynamically select nodes: **beorun** and **mpprun**. They differ only in that **beorun** runs the job concurrently on the selected nodes, while **mpprun** runs the job sequentially on one node at a time.

The following example shows the difference in the elapsed time to run a command with **beorun** vs. **mpprun**:

```
[user@cluster user] $ date;beorun -np 8 sleep 1;date
```

```

Fri Aug 18 11:48:30 PDT 2006
Fri Aug 18 11:48:31 PDT 2006
[user@cluster user] $ date;mpprun -np 8 sleep 1;date
Fri Aug 18 11:48:46 PDT 2006
Fri Aug 18 11:48:54 PDT 2006

```

Example 3-3. Binary Pre-Staged on Compute Node

A needed binary can be "pre-staged" by copying it to a compute node prior to execution of a shell script. In the following example, the shell script is in a file called `test.sh`:

```

#####
#! /bin/bash
hostname.local
#####

[user@cluster user] $ bpsb 1 mkdir -p /usr/local/bin
[user@cluster user] $ bpcp /bin/hostname 1:/usr/local/bin/hostname.local
[user@cluster user] $ bpsb 1 ./test.sh
nl

```

This makes the **hostname** binary available on compute node 1 as `/usr/local/bin/hostname.local` before the script is executed. The shell's `$PATH` contains `/usr/local/bin`, so the compute node searches locally for **hostname.local** in `$PATH`, finds it, and executes it.

Note that copying files to a compute node generally puts the files into the RAM filesystem on the node, thus reducing main memory that might otherwise be available for programs, libraries, and data on the node.

Example 3-4. Binary Migrated to Compute Node

If a binary is not "pre-staged" on a compute node, the full path to the binary must be included in the script in order to execute properly. In the following example, the master node starts the process (in this case, a shell) and moves it to node 1, then continues execution of the script. However, when it comes to the **hostname.local2** command, the process fails:

```

#####
#! /bin/bash
hostname.local2
#####

[user@cluster user] $ bpsb 1 ./test.sh
./test.sh: line 2: hostname.local2: command not found

```

Since the compute node does not have **hostname.local2** locally, the shell attempts to resolve the binary by asking for the binary from the master. The problem is that the master has no idea which binary to give back to the node, hence the failure.

Because there is no way for **Bproc** to know which binaries may be needed by the shell, **hostname.local2** is not migrated along with the shell during the initial startup. Therefore, it is important to provide the compute node with a full path to the binary:

```

#####
#! /bin/bash
/tmp/hostname.local2
#####

```

```
[user@cluster user] $ cp /bin/hostname /tmp/hostname.local2
[user@cluster user] $ bpsch 1 ./test.sh
n1
```

With a full path to the binary, the compute node can construct a proper request for the master, and the master knows which exact binary to return to the compute node for proper execution.

Example 3-5. Process Data Files

Files that are opened by a process (including files on disk, sockets, or named pipes) are not automatically migrated to compute nodes. Suppose the application BOB needs the data file `1.dat`:

```
[user@cluster user] $ bpsch 1 /usr/local/BOB/bin/BOB 1.dat
```

`1.dat` must be either pre-staged to the compute node, e.g., using **bpcp** to copy it there; or else the data files must be accessible on an NFS-mounted file system. The file `/etc/beowulf/fstab` (or a node-specific `fstab.nodeNumber`) specifies which filesystems are NFS-mounted on each compute node by default.

Example 3-6. Installing Commercial Applications

Through the course of its execution, the application BOB in the example above does some work with the data file `1.dat`, and then later attempts to call `/usr/local/BOB/bin/BOB.helper.bin` and `/usr/local/BOB/bin/BOB.cleanup.bin`.

If these binaries are not in the memory space of the process during migration, the calls to these binaries will fail. Therefore, `/usr/local/BOB` should be NFS-mounted to all of the compute nodes, or the binaries should be pre-staged using **bpcp** to copy them by hand to the compute nodes. The binaries will stay on each compute node until that node is rebooted.

Generally for commercial applications, the administrator should have `$APP_HOME` NFS-mounted on the compute nodes that will be involved in execution. A general best practice is to mount a general directory such as `/opt`, and install all of the applications into `/opt`.

Environment Modules

The ClusterWare **env-modules** environment-modules package provides for the dynamic modification of a user's environment via modulefiles. Each modulefile contains the information needed to configure the shell for an application, allowing a user to easily switch between applications with a simple **module switch** command that resets environment variables like `PATH` and `LD_LIBRARY_PATH`. A number of modules are already installed that configure application builds and execution with OpenMPI, MPICH2, and MVAPICH2. Execute the command **module avail** to see a list of available modules. See specific sections, below, for examples of how to use modules.

For more information about creating your own modules, see <http://modules.sourceforge.net>, or view the manpages **man module** and **man modulefile**.

Running Programs That Are Not Parallelized

Starting and Migrating Programs to Compute Nodes (bpsh)

There are no executable programs (binaries) on the file system of the compute nodes. This means that there is no **getty**, no **login**, nor any shells on the compute nodes.

Instead of the remote shell (**rsh**) and secure shell (**ssh**) commands that are available on networked stand-alone computers (each of which has its own collection of binaries), Scyld ClusterWare has the **bpsh** command. The following example shows the standard **ls** command running on node 2 using **bpsh**:

```
[user@cluster user] $ bpsh 2 ls -FC /
  bin/   dev/   home/  lib64/  proc/   sys/   usr/
  bpfs/  etc/   lib/   opt/    sbin/   tmp/   var/
```

At startup time, by default Scyld ClusterWare exports various directories, e.g., `/bin` and `/usr/bin`, on the master node, and those directories are NFS-mounted by compute nodes.

However, an NFS-accessible `/bin/ls` is not a requirement for **bpsh 2 ls** to work. Note that the `/sbin` directory also exists on the compute node. It is not exported by the master node by default, and thus it exists locally on a compute node in the RAM-based filesystem. **bpsh 2 ls /sbin** usually shows an empty directory. Nonetheless, **bpsh 2 modprobe bproc** executes successfully, even though **which modprobe** shows the command resides in `/sbin/modprobe` and **bpsh 2 which modprobe** fails to find the command on the compute node because its `/sbin` does not contain **modprobe**.

bpsh 2 modprobe bproc works because the **bpsh** initiates a **modprobe** process on the master node, then forms a process memory image that includes the command's binary and references to all its dynamically linked libraries. This process memory image is then copied (migrated) to the compute node, and there the references to dynamic libraries are remapped in the process address space. Only then does the **modprobe** command begin real execution.

bpsh is not a special version of **sh**, but a special way of handling execution. This process works with any program. Be aware of the following:

- All three standard I/O streams — `stdin`, `stdout`, and `stderr` — are forwarded to the master node. Since some programs need to read standard input and will stop working if they're run in the background, be sure to close standard input at invocation by using the **bpsh -n** flag when you run a program in the background on a compute node.
- Because shell scripts expect executables to be present, and because compute nodes don't meet this requirement, shell scripts should be modified to include the **bpsh** commands required to affect the compute nodes and run on the master node.
- The dynamic libraries are cached separately from the process memory image, and are copied to the compute node only if they are not already there. This saves time and network bandwidth. After the process completes, the dynamic libraries are unloaded from memory, but they remain in the local cache on the compute node, so they won't need to be copied if needed again.

For additional information on the **BProc** Distributed Process Space and how processes are migrated to compute nodes, see the *Administrator's Guide*.

Copying Information to Compute Nodes (bpcp)

Just as traditional Unix has copy (**cp**), remote copy (**rcp**), and secure copy (**scp**) to move files to and from networked machines, Scyld ClusterWare has the **bpcp** command.

Although the default sharing of the master node's home directories via NFS is useful for sharing small files, it is not a good solution for large data files. Having the compute nodes read large data files served via NFS from the master node will result in major network congestion, or even an overload and shutdown of the NFS server. In these cases, staging data files on compute nodes using the **bpcp** command is an alternate solution. Other solutions include using dedicated NFS servers or NAS appliances, and using cluster file systems.

Following are some examples of using **bpcp**.

This example shows the use of **bpcp** to copy a data file named `foo2.dat` from the current directory to the `/tmp` directory on node 6:

```
[user@cluster user] $ bpcp foo2.dat 6:/tmp
```

The default directory on the compute node is the current directory on the master node. The current directory on the compute node may already be NFS-mounted from the master node, but it may not exist. The example above works, since `/tmp` exists on the compute node, but will fail if the destination does not exist. To avoid this problem, you can create the necessary destination directory on the compute node before copying the file, as shown in the next example.

In this example, we change to the `/tmp/foo` directory on the master, use **bpsh** to create the same directory on the node 6, then copy `foo2.dat` to the node:

```
[user@cluster user] $ cd /tmp/foo
[user@cluster user] $ bpsh 6 mkdir /tmp/foo
[user@cluster user] $ bpcp foo2.dat 6:
```

This example copies `foo2.dat` from node 2 to node 3 directly, without the data being stored on the master node. As in the first example, this works because `/tmp` exists:

```
[user@cluster user] $ bpcp 2:/tmp/foo2.dat 3:/tmp
```

Running Parallel Programs

An Introduction to Parallel Programming APIs

Programmers are generally familiar with serial, or sequential, programs. Simple programs — like "Hello World" and the basic suite of searching and sorting programs — are typical of sequential programs. They have a beginning, an execution sequence, and an end; at any time during the run, the program is executing only at a single point.

A thread is similar to a sequential program, in that it also has a beginning, an execution sequence, and an end. At any time while a thread is running, there is a single point of execution. A thread differs in that it isn't a stand-alone program; it runs within a program. The concept of threads becomes important when a program has multiple threads running at the same time and performing different tasks.

To run in parallel means that more than one thread of execution is running at the same time, often on different processors of one computer; in the case of a cluster, the threads are running on different computers. A few things are required to make parallelism work and be useful: The program must migrate to another computer or computers and get started; at some point, the data upon which the program is working must be exchanged between the processes.

The simplest case is when the same single-process program is run with different input parameters on all the nodes, and the results are gathered at the end of the run. Using a cluster to get faster results of the same non-parallel program with different inputs is called *parametric* execution.

A much more complicated example is a simulation, where each process represents some number of elements in the system. Every few time steps, all the elements need to exchange data across boundaries to synchronize the simulation. This situation requires a *message passing interface* or MPI.

To solve these two problems — program startup and message passing — you can develop your own code using POSIX interfaces. Alternatively, you could utilize an existing parallel application programming interface (API), such as the Message Passing Interface (MPI) or the Parallel Virtual Machine (PVM). These are discussed in the sections that follow.

MPI

The Message Passing Interface (MPI) application programming interface is currently the most popular choice for writing parallel programs. The MPI standard leaves implementation details to the system vendors (like Scyld). This is useful because they can make appropriate implementation choices without adversely affecting the output of the program.

A program that uses MPI is automatically started a number of times and is allowed to ask two questions: How many of us (size) are there, and which one am I (rank)? Then a number of conditionals are evaluated to determine the actions of each process. Messages may be sent and received between processes.

The advantages of MPI are that the programmer:

- Doesn't have to worry about how the program gets started on all the machines
- Has a simplified interface for inter-process messages
- Doesn't have to worry about mapping processes to nodes
- Abstracts the network details, resulting in more portable hardware-agnostic software

Also see the section on running MPI-aware programs later in this chapter. Scyld ClusterWare includes several implementations of MPI:

MPICH2

Scyld ClusterWare includes MPICH2, a second generation MPICH. Visit <http://www.mpich.org> for more information. Scyld MPICH2 is customized to use environment modules. See the Section called *Running MPICH2 and MVAPICH2 Programs* for details.

MVAPICH2

MVAPICH2 is second generation MVAPICH. Visit <http://mvapich.cse.ohio-state.edu/> for more information. Scyld MVAPICH2 is customized to use environment modules. See the Section called *Running MPICH2 and MVAPICH2 Programs* for details.

OpenMPI

OpenMPI is an open-source implementation of the Message Passing Interface 2 (MPI-2) specification. The OpenMPI implementation is an optimized combination of several other MPI implementations. Visit <http://www.open-mpi.org/> for more information. Also see the Section called *Running OpenMPI Programs* for details.

Other MPI Implementations

Various commercial MPI implementations run on Scyld ClusterWare. Visit the Penguin Computing Support Portal at <http://www.penguincomputing.com/support> for more information. You can also download and build your own version of MPI, and configure it to run on Scyld ClusterWare.

PVM

Parallel Virtual Machine (PVM) was an earlier parallel programming interface. Unlike MPI, it is not a specification but a single set of source code distributed on the Internet. PVM reveals much more about the details of starting your job on remote nodes. However, it fails to abstract implementation details as well as MPI does.

PVM is deprecated, but is still in use by legacy code. We generally advise against writing new programs in PVM, but some of the unique features of PVM may suggest its use.

Also see the section on running PVM-aware programs later in this chapter.

Custom APIs

As mentioned earlier, you can develop your own parallel API by using various Unix and TCP/IP standards. In terms of starting a remote program, there are programs written:

- Using the **rexec** function call
- To use the **rexec** or **rsh** program to invoke a sub-program
- To use Remote Procedure Call (RPC)
- To invoke another sub-program using the **inetd** super server

These solutions come with their own problems, particularly in the implementation details. What are the network addresses? What is the path to the program? What is the account name on each of the computers? How is one going to load-balance the cluster?

Scyld ClusterWare, which doesn't have binaries installed on the cluster nodes, may not lend itself to these techniques. We recommend you write your parallel code in MPI. That having been said, we can say that Scyld has some experience with getting **rexec()** calls to work, and that one can simply substitute calls to **rsh** with the more cluster-friendly **bpsh**.

Mapping Jobs to Compute Nodes

Running programs specifically designed to execute in parallel across a cluster requires at least the knowledge of the number of processes to be used. Scyld ClusterWare uses the **NP** environment variable to determine this. The following example will use 4 processes to run an MPI-aware program called **a.out**, which is located in the current directory.

```
[user@cluster user] $ NP=4 ./a.out
```

Note that each kind of shell has its own syntax for setting environment variables; the example above uses the syntax of the Bourne shell (**/bin/sh** or **/bin/bash**).

What the example above does not specify is which specific nodes will execute the processes; this is the job of the *mapper*. Mapping determines which node will execute each process. While this seems simple, it can get complex as various requirements are added. The mapper scans available resources at the time of job submission to decide which processors to use.

Scyld ClusterWare includes **beomap**, a mapping API (documented in the *Programmer's Guide* with details for writing your own mapper). The mapper's default behavior is controlled by the following environment variables:

- **NP** — The number of processes requested, but not the number of processors. As in the example earlier in this section, **NP=4 ./a.out** will run the MPI program **a.out** with 4 processes.

- *ALL_CPUS* — Set the number of processes to the number of CPUs available to the current user. Similar to the example above, `--all-cpus=1 ./a.out` would run the MPI program `a.out` on all available CPUs.
- *ALL_NODES* — Set the number of processes to the number of nodes available to the current user. Similar to the *ALL_CPUS* variable, but you get a maximum of one CPU per node. This is useful for running a job per node instead of per CPU.
- *ALL_LOCAL* — Run every process on the master node; used for debugging purposes.
- *NO_LOCAL* — Don't run any processes on the master node.
- *EXCLUDE* — A colon-delimited list of nodes to be avoided during node assignment.
- *BEOWULF_JOB_MAP* — A colon-delimited list of nodes. The first node listed will be the first process (MPI Rank 0) and so on.

You can use the `beomap` program to display the current mapping for the current user in the current environment with the current resources at the current time. See the *Reference Guide* for a detailed description of `beomap` and its options, as well as examples for using it.

Running OpenMPI Programs

OpenMPI programs are those written to the MPI-2 specification. This section provides information needed to use programs with OpenMPI as implemented in Scyld ClusterWare.

Pre-Requisites to Running OpenMPI

A number of commands, such as `mpirun`, are duplicated between OpenMPI and other MPI implementations. Scyld ClusterWare provides the `env-modules` package which gives users a convenient way to switch between the various implementations. Be sure to load an OpenMPI module to favor OpenMPI, located in `/opt/scyld/openmpi/`. Each module bundles together various compiler-specific environment variables to configure your shell for building and running your application, and for accessing compiler-specific manpages. Be sure that you are loading the proper module to match the compiler that built the application you wish to run. For example, to load the OpenMPI module for use with the Intel compiler, do the following:

```
[user@cluster user] $ module load openmpi/intel
```

Currently, there are modules for the GNU, Intel, and PGI compilers. To see a list of all of the available modules:

```
[user@cluster user] $ module avail openmpi
----- /opt/modulefiles -----
openmpi/gnu/1.5.3   openmpi/intel/1.5.3 openmpi/pgi/1.5.3
```

For more information about creating your own modules, see <http://modules.sourceforge.net> and the manpages `man module` and `man modulefile`.

Using OpenMPI

OpenMPI does not honor the Scyld Beowulf job mapping environment variables. You must either specify the list of hosts on the command line, or inside of a hostfile. To specify the list of hosts on the command line, use the `-H` option. The argument following `-H` is a comma separated list of hostnames, not node numbers. For example, to run a two process job, with one process running on node 0 and one on node 1:

```
[user@cluster user] $ mpirun -H n0,n1 -np 2 ./mpiprogram
```

Support for running jobs over Infiniband using the OpenIB transport is included with OpenMPI distributed with Scyld ClusterWare. One must specifically request the use of OpenIB. For example:

```
[user@cluster user] $ mpirun --mca btl openib,sm,self -H n0,n1 -np 2 ./myprog
```

Read the OpenMPI **mpirun** man page for more information about, using a hostfile, and using other tunable options available through **mpirun**.

Running MPICH2 and MVAPICH2 Programs

MPICH2 and MVAPICH2 programs are those written to the MPI-2 specification. This section provides information needed to use programs with MPICH2 or MVAPICH2 as implemented in Scyld ClusterWare.

Pre-Requisites to Running MPICH2/MVAPICH2

As with Scyld OpenMPI, the Scyld MPICH2 and MVAPICH2 distributions are repackaged Open Source MPICH2 and MVAPICH2 that utilize environment modules to build and to execute applications. Each module bundles together various compiler-specific environment variables to configure your shell for building and running your application, and for accessing implementation- and compiler-specific manpages. You must use the same module to both build the application and to execute it. For example, to load the MPICH2 module for use with the Intel compiler, do the following:

```
[user@cluster user] $ module load mpich2/intel
```

Currently, there are modules for the GNU, Intel, and PGI compilers. To see a list of all of the available modules:

```
[user@cluster user] $ module avail mpich2 mvapich2
----- /opt/modulefiles -----
mpich2/gnu/1.3.2   mpich2/intel/1.3.2 mpich2/pgi/1.3.2
----- /opt/modulefiles -----
mvapich2/gnu/1.6   mvapich2/intel/1.6 mvapich2/pgi/1.6
```

For more information about creating your own modules, see <http://modules.sourceforge.net> and the manpages **man module** and **man modulefile**.

Using MPICH2

MPICH2 does not honor the Scyld Beowulf job mapping environment variables. Use **mpiexec** to execute MPICH2 applications. After loading an mpich2 module, see the **man mpiexec** manpage for specifics, and visit <http://www.mpich.org> for full documentation.

Using MVAPICH2

MVAPICH2 does not honor the Scyld Beowulf job mapping environment variables. Use **mpirun_rsh** to execute MVAPICH2 applications. After loading an mvapich2 module, use **mpirun_rsh --help** to see specifics, and visit <http://mvapich.cse.ohio-state.edu/> for full documentation.

Running PVM-Aware Programs

Parallel Virtual Machine (PVM) is an application programming interface for writing parallel applications, enabling a collection of heterogeneous computers to be used as a coherent and flexible concurrent computational resource. Scyld has developed the Scyld PVM library, specifically tailored to allow PVM to take advantage of the technologies used in Scyld ClusterWare. A PVM-aware program is one that has been written to the PVM specification and linked against the Scyld PVM library.

A complete discussion of cluster configuration for PVM is beyond the scope of this document. However, a brief introduction is provided here, with the assumption that the reader has some background knowledge on using PVM.

You can start the master PVM daemon on the master node using the PVM console, **pvm**. To add a compute node to the virtual machine, issue an **add .#** command, where # is replaced by a node's assigned number in the cluster.

Tip: You can generate a list of node numbers using **bpstat** command.

Alternately, you can start the PVM console with a hostfile filename on the command line. The hostfile should contain a **.#** for each compute node you want as part of the virtual machine. As with standard PVM, this method automatically spawns PVM slave daemons to the specified compute nodes in the cluster. From within the PVM console, use the **conf** command to list your virtual machine's configuration; the output will include a separate line for each node being used. Once your virtual machine has been configured, you can run your PVM applications as you normally would.

Porting Other Parallelized Programs

Programs written for use on other types of clusters may require various levels of change to function with Scyld ClusterWare. For instance:

- Scripts or programs that invoke **rsh** can instead call **bpsh**.
- Scripts or programs that invoke **rcp** can instead call **bpcp**.
- **beomap** can be used with any script to load balance programs that are to be dispatched to the compute nodes.

For more information on porting applications, see the *Programmer's Guide*

Running Serial Programs in Parallel

For jobs that are not "MPI-aware" or "PVM-aware", but need to be started in parallel, Scyld ClusterWare provides the parallel execution utilities **mpprun** and **beorun**. These utilities are more sophisticated than **bpsh**, in that they can automatically select ranges of nodes on which to start your program, run tasks on the master node, determine the number of CPUs on a node, and start a copy on each CPU. Thus, **mpprun** and **beorun** provide you with true "dynamic execution" capabilities, whereas **bpsh** provides "directed execution" only.

mpprun and **beorun** are very similar, and have similar parameters. They differ only in that **mpprun** runs jobs sequentially on the selected processors, while **beorun** runs jobs concurrently on the selected processors.

mpprun

mpprun is intended for applications rather than utilities, and runs them sequentially on the selected nodes. The basic syntax of **mpprun** is as follows:

```
[user@cluster user] $ mpprun [options] app arg1 arg2...
```

where *app* is the application program you wish to run; it need not be a parallel program. The *arg* arguments are the values passed to each copy of the program being run.

Options

mpprun includes options for controlling various aspects of the job, including the ability to:

- Specify the number of processors on which to start copies of the program
- Start one copy on each node in the cluster
- Start one copy on each CPU in the cluster
- Force all jobs to run on the master node
- Prevent any jobs from running on the master node

The most interesting of the options is the **--map** option, which lets the user specify which nodes will run copies of a program; an example is provided in the next section. This argument, if specified, overrides the mapper's selection of resources that it would otherwise use.

See the *Reference Guide* for a complete list of options for **mpprun**.

Examples

Run 16 tasks of program *app*:

```
[user@cluster user] $ mpprun -np 16 app infile outfile
```

Run 16 tasks of program *app* on any available nodes except nodes 2 and 3:

```
[user@cluster user] $ mpprun -np 16 --exclude 2:3 app infile outfile
```

Run 4 tasks of program *app* with task 0 on node 4, task 1 on node 2, task 2 on node 1, and task 3 on node 5:

```
[user@cluster user] $ mpprun --map 4:2:1:5 app infile outfile
```

beorun

beorun is intended for applications rather than utilities, and runs them concurrently on the selected nodes. The basic syntax of **beorun** is as follows:

```
[user@cluster user] $ beorun [options] app arg1 arg2...
```

where *app* is the application program you wish to run; it need not be a parallel program. The *arg* arguments are the values passed to each copy of the program being run.

Options

beorun includes options for controlling various aspects of the job, including the ability to:

- Specify the number of processors on which to start copies of the program
- Start one copy on each node in the cluster
- Start one copy on each CPU in the cluster
- Force all jobs to run on the master node
- Prevent any jobs from running on the master node

The most interesting of the options is the **--map** option, which lets the user specify which nodes will run copies of a program; an example is provided in the next section. This argument, if specified, overrides the mapper's selection of resources that it would otherwise use.

See the *Reference Guide* for a complete list of options for **beorun**.

Examples

Run 16 tasks of program *app*:

```
[user@cluster user] $ beorun -np 16 app infile outfile
```

Run 16 tasks of program *app* on any available nodes except nodes 2 and 3:

```
[user@cluster user] $ beorun -np 16 --exclude 2:3 app infile outfile
```

Run 4 tasks of program *app* with task 0 on node 4, task 1 on node 2, task 2 on node 1, and task 3 on node 5:

```
[user@cluster user] $ beorun --map 4:2:1:5 app infile outfile
```

Job Batching

Job Batching Options for ClusterWare

For Scyld ClusterWare HPC, the default installation includes both the TORQUE resource manager and the Slurm workload manager, each providing users an intuitive interface for remotely initiating and managing batch jobs on distributed compute nodes. TORQUE is an open source tool based on standard OpenPBS. Slurm is another open source tool, employing the open source Munge for authentication and mariadb for managing a database. Basic instructions for using TORQUE are provided in the next section. For more general product information, see <http://www.adaptivecomputing.com/> for Adaptive Computing's TORQUE information and <http://slurm.schedmd.com/> for Slurm information.

Only one job manager can be enabled at any one time. See the Scyld ClusterWare *Administrator's Guide* for details about how to enable either TORQUE or Slurm. If Slurm is the chosen job manager, then users must setup the `PATH` and `LD_LIBRARY_PATH` environment variables to properly access the Slurm commands. This is done automatically for users who login when the *slurm* service is running and the *pbs_server* is not running, via the `/etc/profile.d/scyld.slurm.sh` script. Alternatively, each Slurm user can manually execute **module load slurm** or can add that command line to (for example) the user's `.bash_profile`.

The <http://slurm.schedmd.com/> Slurm website also provides an optional TORQUE wrapper to minimize the syntactic differences between TORQUE and Slurm commands and scripts. See <http://slurm.schedmd.com/rosetta.pdf> for a discussion of the differences between TORQUE and Slurm, and <http://slurm.schedmd.com/faq.html#torque> provides useful information about how to switch from PBS or TORQUE to Slurm.

Scyld also redistributes the Scyld Maui job scheduler, also derived from Adaptive Computing, that functions in conjunction with the TORQUE job manager. The alternative Moab job scheduler is also available from Adaptive Computing with a separate license, giving customers additional job scheduling, reporting, and monitoring capabilities.

In addition, Scyld provides support for most popular open source and commercial schedulers and resource managers, including SGE, LSF, and PBSPro. For the latest information, visit the Penguin Computing Support Portal at <http://www.penguincomputing.com/support>.

Job Batching with TORQUE

The default installation is configured as a simple job serializer with a single queue named batch.

You can use the TORQUE resource manager to run jobs, check job status, find out which nodes are running your job, and find job output.

Running a Job

To run a job with TORQUE, you can put the commands you would normally use into a job script, and then submit the job script to the cluster using **qsub**. The **qsub** program has a number of options that may be supplied on the command line or as special directives inside the job script. For the most part, these options should behave exactly the same in a job script or via the command line, but job scripts make it easier to manage your actions and their results.

Following are some examples of running a job using **qsub**. For more detailed information on **qsub**, see the **qsub** man page.

Example 3-7. Starting a Job with a Job Script Using One Node

The following script declares a job with the name "myjob", to be run using one node. The script uses the PBS -N directive, launches the job, and finally sends the current date and working directory to standard output.

```
#!/bin/sh

## Set the job name
#PBS -N myjob
#PBS -l nodes=1

# Run my job
/path/to/myjob

echo Date: $(date)
echo Dir: $PWD
```

You would submit "myjob" as follows:

```
[bjosh@iceberg]$ qsub -l nodes=1 myjob
15.iceberg
```


Example 3-8. Starting a Job from the Command Line

This example provides the command line equivalent of the job run in the example above. We enter all of the **qsub** options on the initial command line. Then **qsub** reads the job commands line-by-line until we type **^D**, the end-of-file character. At that point, **qsub** queues the job and returns the Job ID.

```
[bjosh@iceberg]$ qsub -N myjob -l nodes=1:ppn=1 -j oe
cd $PBS_0_WORKDIR
echo Date: $(date)
echo Dir: $PWD
^D
16.iceberg
```

Example 3-9. Starting an MPI Job with a Job Script

The following script declares an MPI job named "mpijob". The script uses the **PBS -N** directive, prints out the nodes that will run the job, launches the job using **mpirun**, and finally prints out the current date and working directory. When submitting MPI jobs using TORQUE, it is recommended to simply call **mpirun** without any arguments. **mpirun** will detect that it is being launched from within TORQUE and assure that the job will be properly started on the nodes TORQUE has assigned to the job. In this case, TORQUE will properly manage and track resources used by the job.

```
## Set the job name
#PBS -N mpijob

# RUN my job
mpirun /path/to/mpijob

echo Date: $(date)
echo Dir: $PWD
```

To request 8 total processors to run "mpijob", you would submit the job as follows:

```
[bjosh@iceberg]$ qsub -l nodes=8 mpijob
17.iceberg
```

To request 8 total processors, using 4 nodes, each with 2 processors per node, you would submit the job as follows:

```
[bjosh@iceberg]$ qsub -l nodes=4:ppn=2 mpijob
18.iceberg
```

Checking Job Status

You can check the status of your job using **qstat**. The command line option **qstat -n** will display the status of queued jobs. To watch the progression of events, use the **watch** command to execute **qstat -n** every 2 seconds by default; type **[CTRL]-C** to interrupt **watch** when needed.

Example 3-10. Checking Job Status

This example shows how to check the status of the job named "myjob", which we ran on 1 node in the first example above, using the option to watch the progression of events.

```
[bjosh@iceberg]$ qsub myjob && watch qstat -n
iceberg:
```

```
JobID Username Queue Jobname SessID NDS TSK ReqdMemory ReqdTime S ElapTime
```

```
15.iceberg bjosh default myjob -- 1 -- -- 00:01 Q --
```

Table 3-1. Useful Job Status Commands

Command	Purpose
<code>ps -ef bpstat -P</code>	Display all running jobs, with node number for each
<code>qstat -Q</code>	Display status of all queues
<code>qstat -n</code>	Display status of queued jobs
<code>qstat -f JOBID</code>	Display very detailed information about Job ID
<code>pbsnodes -a</code>	Display status of all nodes

Finding Out Which Nodes Are Running a Job

To find out which nodes are running your job, use the following commands:

- To find your Job Ids: `qstat -an`
- To find the Process IDs of your jobs: `qstat -f <jobid>`
- To find the number of the node running your job: `ps -ef | bpstat -P | grep <yourname>`

The number of the node running your job will be displayed in the first column of output.

Finding Job Output

When your job terminates, TORQUE will store its output and error streams in files in the script's working directory.

- Default output file: `<jobname>.o<jobid>`
You can override the default using `qsub` with the `-o <path>` option on the command line, or use the `#PBS -o <path>` directive in your job script.
- Default error file: `<jobname>.e<jobid>`
You can override the default using `qsub` with the `-e <path>` option on the command line, or use the `#PBS -e <path>` directive in your job script.
- To join the output and error streams into a single file, use `qsub` with the `-j oe` option on the command line, or use the `#PBS -j oe` directive in your job script.

Job Batching with POD Tools

POD Tools is a collection of tools for submitting TORQUE jobs to a remote cluster and for monitoring them. POD Tools is useful for, but not limited to, submitting and monitoring jobs to a remote Penguin On Demand cluster. POD Tools executes on both Scyld and non-Scyld client machines, and the Tools communicate with the **beoweb** service that must be executing on the target cluster.

The primary tool in POD Tools is **POD Shell (podsh)**, which is a command-line interface that allows for remote job submission and monitoring. POD Shell is largely self-documented. Enter `podsh --help` for a list of possible commands and their formats.

The general usage is `podsh <action> [OPTIONS] [FILE/ID]`. The *action* specifies what type of action to perform, such as *submit* (for submitting a new job) or *status* (for collecting status on all jobs or a specific job).

POD Shell can upload a TORQUE job script to the target cluster, where it will be added to the job queue. Additionally, POD Shell can be used to stage data in and out of the target cluster. Staging data in (i.e. copying data to the cluster) is performed across an unencrypted TCP socket. Staging data out (i.e. from the cluster back to the client machine) is performed using `scp` from the cluster to the client. In order for this transfer to be successful, password-less authentication must be in place using SSH keys between the cluster's master node and the client.

POD Shell uses a configuration file that supports both site-wide and user-local values. Site-wide values are stored in entries in `/etc/podtools.conf`. These settings can be overridden by values in a user's `~/podtools/podtools.conf` file. These values can again be overridden by command-line arguments passed to `podsh`. The template for `podtools.conf` is found at `/opt/scyld/podtools/podtools.conf.template`.

Using Singularity

Scyld ClusterWare 7 distributes Singularity, a powerful Linux container platform designed by Lawrence Berkeley National Laboratory.

Singularity enables users to have full control of their environment, allowing a non-privileged user to "swap out" the operating system on the host by executing a lightweight Singularity container environment and an application that executes within that environment. For example, Singularity can provide a user with the ability to create an Ubuntu image of their application, and run the containerized application on a RHEL7 or CentOS7 ClusterWare system in its native Ubuntu environment.

Refer to the Lawrence Berkeley Lab's Singularity documentation at <http://singularity.lbl.gov> for instructions on how to create and use Singularity containers.

When running MPI-enabled applications with Singularity on Scyld ClusterWare, follow these additional instructions:

- Always compile MPI applications inside a container image with the same MPI implementation and version you plan to use on your Scyld ClusterWare system. Refer to the Singularity documentation for currently supported MPI implementations.
- Be aware of the MPI transports which are compatible with your containerized binary, and ensure that you use the same MPI transport when executing MPI applications through Singularity. For example, Scyld ClusterWare's OpenMPI packages support TCP, Verbs, PSM and PSM2 MPI transports, but not all operating systems will support this gamut of options. Adjust your `mpirun` accordingly on Scyld ClusterWare to use the MPI transport supported by your containerized application.

For example, after building a container image and an OpenMPI executable binary that was built for that image::

```
module load singularity
module load openmpi/gnu/2.0.2
mpirun -np 4 -H n0,n1,n2,n3 singularity exec <container.img> <container mpi binary>
```

File Systems

Data files used by the applications processed on the cluster may be stored in a variety of locations, including:

- On the local disk of each node
- On the master node's disk, shared with the nodes through a network file system

- On disks on multiple nodes, shared with all nodes through the use of a parallel file system

The simplest approach is to store all files on the master node, as with the standard Network File System. Any files in your `/home` directory are shared via NFS with all the nodes in your cluster. This makes management of the files very simple, but in larger clusters the performance of NFS on the master node can become a bottleneck for I/O-intensive applications. If you are planning a large cluster, you should include disk drives that are separate from the system disk to contain your shared files; for example, place `/home` on a separate pair of RAID1 disks in the master node. A more scalable solution is to utilize a dedicated NFS server with a properly configured storage system for all shared files and programs, or a high performance NAS appliance.

Storing files on the local disk of each node removes the performance problem, but makes it difficult to share data between tasks on different nodes. Input files for programs must be distributed manually to each of the nodes, and output files from the nodes must be manually collected back on the master node. This mode of operation can still be useful for temporary files created by a process and then later reused on that same node.

Notes

1. <http://modules.sourceforge.net>
2. <http://www.mpich.org>
3. <http://mvapich.cse.ohio-state.edu/>
4. <http://www.open-mpi.org/>
5. <http://www.penguincomputing.com/support>
6. <http://modules.sourceforge.net>
7. <http://modules.sourceforge.net>
8. <http://www.mpich.org>
9. <http://mvapich.cse.ohio-state.edu/>
10. <http://www.adaptivecomputing.com/>
11. <http://slurm.schedmd.com/>
12. <http://slurm.schedmd.com/>
13. <http://slurm.schedmd.com/rosetta.pdf>
14. <http://slurm.schedmd.com/faq.html#torque>
15. <http://www.penguincomputing.com/support>
16. <http://singularity.lbl.gov>

Appendix A. Glossary of Parallel Computing Terms

Bandwidth

A measure of the total amount of information delivered by a network. This metric is typically expressed in millions of bits per second (Mbps) for data rate on the physical communication media or megabytes per second (MBps) for the performance seen by the application.

Backplane Bandwidth

The total amount of data that a switch can move through it in a given time, typically much higher than the bandwidth delivered to a single node.

Bisection Bandwidth

The amount of data that can be delivered from one half of a network to the other half in a given time, through the least favorable halving of the network fabric.

Boot Image

The file system and kernel seen by a compute node at boot time; contains enough drivers and information to get the system up and running on the network.

Cluster

A collection of nodes, usually dedicated to a single purpose.

Compute Node

Nodes attached to the master through an interconnection network, used as dedicated attached processors. With Scyld, users never need to directly log into compute nodes.

Data Parallel

A style of programming in which multiple copies of a single program run on each node, performing the same instructions while operating on different data.

Efficiency

The ratio of a program's actual speed-up to its theoretical maximum.

FLOPS

Floating-point operations per second, a key measure of performance for many scientific and numerical applications.

Grain Size, Granularity

A measure of the amount of computation a node can perform in a given problem between communications with other nodes, typically defined as "coarse" (large amount of computation) or "fine" (small amount of computation). Granularity is a key in determining the performance of a particular process on a particular cluster.

High Availability

Refers to level of reliability; usually implies some level of fault tolerance (ability to operate in the presence of a hardware failure).

Hub

A device for connecting the NICs in an interconnection network. Only one pair of ports (a bus) can be active at any time. Modern interconnections utilize switches, not hubs.

Isoefficiency

The ability of a process to maintain a constant efficiency if the size of the process scales with the size of the machine.

Jobs

In traditional computing, a job is a single task. A parallel job can be a collection of tasks, all working on the same problem but running on different nodes.

Kernel

The core of the operating system, the kernel is responsible for processing all system calls and managing the system's physical resources.

Latency

The length of time from when a bit is sent across the network until the same bit is received. Can be measured for just the network hardware (wire latency) or application-to-application (includes software overhead).

Local Area Network (LAN)

An interconnection scheme designed for short physical distances and high bandwidth, usually self-contained behind a single router.

MAC Address

On an Ethernet NIC, the hardware address of the card. MAC addresses are unique to the specific NIC, and are useful for identifying specific nodes.

Master Node

Node responsible for interacting with users, connected to both the public network and interconnection network. The master node controls the compute nodes.

Message Passing

Exchanging information between processes, frequently on separate nodes.

Middleware

A layer of software between the user's application and the operating system.

MPI

The Message Passing Interface, the standard for producing message passing libraries.

Network Interface Card (NIC)

The device through which a node connects to the interconnection network. The performance of the NIC and the network it attaches to limit the amount of communication that can be done by a parallel program.

Node

A single computer system (motherboard, one or more processors, memory, possibly a disk, network interface).

Parallel Programming

The art of writing programs that are capable of being executed on many processors simultaneously.

Process

An instance of a running program.

Process Migration

Moving a process from one computer to another after the process begins execution.

PVM

The Parallel Virtual Machine, a common message passing library that predates MPI.

Scalability

The ability of a process to maintain efficiency as the number of processors in the parallel machine increases.

Single System Image

All nodes in the system see identical system files, including the same kernel, libraries, header files, etc. This guarantees that a program that will run on one node will run on all nodes.

Socket

A low-level construct for creating a connection between processes on a remote system.

Speedup

A measure of the improvement in the execution time of a program on a parallel computer vs. a serial computer.

Switch

A device for connecting the NICs in an interconnection network so that all pairs of ports can communicate simultaneously.

Version Skew

The problem of having more than one version of software or files (kernel, tools, shared libraries, header files) on different nodes.

Appendix B. TORQUE and Maui Release Information

TORQUE software downloads from Adaptive Computing: <http://www.adaptivecomputing.com/support/download-center/torque-download/>

Maui software downloads from: <http://www.adaptivecomputing.com/support/download-center/maui-cluster-scheduler/> and documentation is found at: <http://docs.adaptivecomputing.com/maui/index.php>

Adaptive Computing's TORQUE release notes are found at <http://www.adaptivecomputing.com/support/documentation-index/torque-resource-manager-documentation/>

Notes

1. <http://www.adaptivecomputing.com/support/download-center/torque-download/>
2. <http://www.adaptivecomputing.com/support/download-center/maui-cluster-scheduler/>
3. <http://docs.adaptivecomputing.com/maui/index.php>
4. <http://www.adaptivecomputing.com/support/documentation-index/torque-resource-manager-documentation/>

Appendix C. OpenMPI Release Information

The following is reproduced essentially verbatim from files contained within the OpenMPI tarball downloaded from <http://www.open-mpi.org/>

Copyright (c) 2004-2010 The Trustees of Indiana University and Indiana University Research and Technology Corporation. All rights reserved.

Copyright (c) 2004-2006 The University of Tennessee and The University of Tennessee Research Foundation. All rights reserved.

Copyright (c) 2004-2008 High Performance Computing Center Stuttgart, University of Stuttgart. All rights reserved.

Copyright (c) 2004-2006 The Regents of the University of California. All rights reserved.

Copyright (c) 2006-2017 Cisco Systems, Inc. All rights reserved.

Copyright (c) 2006 Voltaire, Inc. All rights reserved.

Copyright (c) 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

Copyright (c) 2006-2017 Los Alamos National Security, LLC. All rights reserved.

Copyright (c) 2010-2012 IBM Corporation. All rights reserved.

Copyright (c) 2012 Oak Ridge National Labs. All rights reserved.

Copyright (c) 2012-2017 Sandia National Laboratories. All rights reserved.

Copyright (c) 2012 University of Houston. All rights reserved.

Copyright (c) 2013 NVIDIA Corporation. All rights reserved.

Copyright (c) 2013-2016 Intel, Inc. All rights reserved.

Additional copyrights may follow.

As more fully described in the "Software Version Number" section in the README file, Open MPI typically releases two separate version series simultaneously. Since these series have different goals and are semi-independent of each other, a single NEWS-worthy item may be introduced into different series at different times. For example, feature F was introduced in the vA.B series at version vA.B.C, and was later introduced into the vX.Y series at vX.Y.Z.

The first time feature F is released, the item will be listed in the vA.B.C section, denoted as:

```
(** also to appear: X.Y.Z) -- indicating that this item is also
                           likely to be included in future release
                           version vX.Y.Z.
```

When vX.Y.Z is later released, the same NEWS-worthy item will also be included in the vX.Y.Z section and be denoted as:

```
(** also appeared: A.B.C) -- indicating that this item was previously
                           included in release version vA.B.C.
```

2.1.0 -- March, 2017

Major new features:

Appendix C. OpenMPI Release Information

- The main focus of the Open MPI v2.1.0 release was to update to PMIx v1.2.1. When using PMIx (e.g., via mpirun-based launches, or via direct launches with recent versions of popular resource managers), launch time scalability is improved, and the run time memory footprint is greatly decreased when launching large numbers of MPI / OpenSHMEM processes.
- Update OpenSHMEM API conformance to v1.3.
- The usnic BTL now supports MPI_THREAD_MULTIPLE.
- General/overall performance improvements to MPI_THREAD_MULTIPLE.
- Add a summary message at the bottom of configure that tells you many of the configuration options specified and/or discovered by Open MPI.

Changes in behavior compared to prior versions:

- None.

Removed legacy support:

- The ptmalloc2 hooks have been removed from the Open MPI code base. This is not really a user-noticable change; it is only mentioned here because there was much rejoicing in the Open MPI developer community.

Bug fixes/minor improvements:

- New MCA parameters:
 - `iof_base_redirect_app_stderr_to_stdout`: as its name implies, it combines MPI / OpenSHMEM applications' stderr into its stdout stream.
 - `opal_event_include`: allow the user to specify which FD selection mechanism is used by the underlying event engine.
 - `opal_stacktrace_output`: indicate where stacktraces should be sent upon MPI / OpenSHMEM process crashes ("none", "stdout", "stderr", "file:filename").
 - `orte_timeout_for_stack_trace`: number of seconds to wait for stack traces to be reported (or ≤ 0 to wait forever).
 - `mtl_ofi_control_prog_type/mtl_ofi_data_prog_type`: specify libfabric progress model to be used for control and data.
- Fix MPI_WTICK regression where the time reported may be inaccurate on systems with processor frequency scaling enabled.
- Fix regression that lowered the memory maximum message bandwidth for large messages on some BTL network transports, such as openib, sm, and vader.
- Fix a name collision in the shared file pointer MPI IO file locking scheme. Thanks to Nicolas Joly for reporting the issue.
- Fix datatype extent/offset errors in MPI_PUT and MPI_RACCUMULATE when using the Portals 4 one-sided component.
- Add support for non-contiguous datatypes to the Portals 4 one-sided component.
- Various updates for the UCX PML.
- Updates to the following man pages:
 - `mpirun(1)`
 - `MPI_COMM_CONNECT(3)`

- MPI_WIN_GET_NAME(3). Thanks to Nicolas Joly for reporting the typo.
- MPI_INFO_GET_[NKEYS|NTHKEY](3). Thanks to Nicolas Joly for reporting the typo.
- Fixed a problem in the TCP BTL when using MPI_THREAD_MULTIPLE. Thanks to Evgueni Petrov for reporting.
- Fixed external32 representation in the romio314 module. Note that for now, external32 representation is not correctly supported by the ompio module. Thanks to Thomas Gastine for bringing this to our attention.
- Add note how to disable a warning message about when a high-speed MPI transport is not found. Thanks to Susan Schwarz for reporting the issue.
- Ensure that sending SIGINT when using the rsh/ssh launcher does not orphan children nodes in the launch tree.
- Fix the help message when showing deprecated MCA param names to show the correct (i.e., deprecated) name.
- Enable support for the openib BTL to use multiple different InfiniBand subnets.
- Fix a minor error in MPI_AINT_DIFF.
- Fix bugs with MPI_IN_PLACE handling in:
 - MPI_ALLGATHER[V]
 - MPI_[I][GATHER|SCATTER][V]
 - MPI_IREDUCE[_SCATTER]
- Thanks to all the users who helped diagnose these issues.
- Allow qrsh to tree spawn (if the back-end system supports it).
- Fix MPI_T_PVAR_GET_INDEX to return the correct index.
- Correctly position the shared file pointer in append mode in the OMPIO component.
- Add some deprecated names into shmem.h for backwards compatibility with legacy codes.
- Fix MPI_MODE_NOCHECK support.
- Fix a regression in PowerPC atomics support. Thanks to Orion Poplawski for reporting the issue.
- Fixes for assembly code with aggressively-optimized compilers on x86_64/AMD64 platforms.
- Fix one more place where configure was mangling custom CFLAGS. Thanks to Phil Tooley (@Telemin) for reporting the issue.
- Better handle builds with external installations of hwloc.
- Fixed a hang with MPI_PUT and MPI_WIN_LOCK_ALL.
- Fixed a bug when using MPI_GET on non-contiguous datatypes and MPI_LOCK/MPI_UNLOCK.
- Fixed a bug when using POST/START/COMPLETE/WAIT after a fence.
- Fix configure portability by cleaning up a few uses of "==" with "test". Thanks to Kevin Buckley for pointing out the issue.
- Fix bug when using darrays with lib and extent of darray datatypes.
- Updates to make Open MPI binary builds more bit-for-bit reproducible. Thanks to Alastair McKinstry for the suggestion.
- Fix issues regarding persistent request handling.
- Ensure that shmemx.h is a standalone OpenSHMEM header file. Thanks to Nick Park (@nspark) for the report.
- Ensure that we always send SIGTERM prior to SIGKILL. Thanks to Noel Rycroft for the report.
- Added ConnectX-5 and Chelsio T6 device defaults for the openib BTL.
- OpenSHMEM no longer supports MXM less than v2.0.

Appendix C. OpenMPI Release Information

- Plug a memory leak in `ompi_osc_sm_free`. Thanks to Joseph Schuchart for the report.
- The "self" BTL now uses less memory.
- The vader BTL is now more efficient in terms of memory usage when using XPMEM.
- Removed the `--enable-openib-failover` configure option. This is not considered backwards-incompatible because this option was stale and had long-since stopped working, anyway.
- Allow jobs launched under Cray aprun to use hyperthreads if `opal_hwloc_base_hwthreads_as_cpus` MCA parameter is set.
- Add support for 32-bit and floating point Cray Aries atomic operations.
- Add support for network AMOs for `MPI_ACCUMULATE`, `MPI_FETCH_AND_OP`, and `MPI_COMPARE_AND_SWAP` if the "ompi_single_intrinsic" info key is set on the window or the "acc_single_intrinsic" MCA param is set.
- Automatically disqualify RDMA CM support in the openib BTL if `MPI_THREAD_MULTIPLE` is used.
- Make configure smarter/better about auto-detecting Linux CMA support.
- Improve the scalability of `MPI_COMM_SPLIT_TYPE`.
- Fix the mixing of C99 and C++ header files with the MPI C++ bindings. Thanks to Alastair McKinstry for the bug report.
- Add support for ARM v8.
- Several MCA parameters now directly support MPI_T enumerator semantics (i.e., they accept a limited set of values -- e.g., MCA parameters that accept boolean values).
- Added `--with-libmpi-name=STRING` configure option for vendor releases of Open MPI. See the README for more detail.
- Fix a problem with Open MPI's internal memory checker. Thanks to Yvan Fournier for reporting.
- Fix a multi-threaded issue with `MPI_WAIT`. Thanks to Pascal Deveze for reporting.

Known issues (to be addressed in v2.1.1):

- See the list of fixes slated for v2.1.1 here:
<https://github.com/open-mpi/ompi/milestone/26>

2.0.2 -- 26 January 2017

Bug fixes/minor improvements:

- Fix a problem with `MPI_FILE_WRITE_SHARED` when using `MPI_MODE_APPEND` and Open MPI's native MPI-IO implementation. Thanks to Nicolas Joly for reporting.
- Fix a typo in the `MPI_WIN_GET_NAME` man page. Thanks to Nicolas Joly for reporting.
- Fix a race condition with ORTE's session directory setup. Thanks to @tbj900 for reporting this issue.
- Fix a deadlock issue arising from Open MPI's approach to catching calls to `munmap`. Thanks to Paul Hargrove for reporting and helping to analyze this problem.
- Fix a problem with PPC atomics which caused `make check` to fail unless builtin

- atomics configure option was enabled. Thanks to Orion Poplawski for reporting.
- Fix a problem with use of x86_64 cpuid instruction which led to segmentation faults when Open MPI was configured with -O3 optimization. Thanks to Mark Santcroos for reporting this problem.
 - Fix a problem when using built in atomics configure options on PPC platforms when building 32 bit applications. Thanks to Paul Hargrove for reporting.
 - Fix a problem with building Open MPI against an external hwloc installation. Thanks to Orion Poplawski for reporting this issue.
 - Remove use of DATE in the message queue version string reported to debuggers to insure bit-wise reproducibility of binaries. Thanks to Alastair McKinstry for help in fixing this problem.
 - Fix a problem with early exit of a MPI process without calling MPI_FINALIZE or MPI_ABORT that could lead to job hangs. Thanks to Christof Koehler for reporting.
 - Fix a problem with forwarding of SIGTERM signal from mpirun to MPI processes in a job. Thanks to Noel Rycroft for reporting this problem
 - Plug some memory leaks in MPI_WIN_FREE discovered using Valgrind. Thanks to Joseph Schuchart for reporting.
 - Fix a problems MPI_NEIGHOR_ALLTOALL when using a communicator with an empty topology graph. Thanks to Daniel Ibanez for reporting.
 - Fix a typo in a PMIx component help file. Thanks to @njoly for reporting this.
 - Fix a problem with Valgrind false positives when using Open MPI's internal memchecker. Thanks to Yvan Fournier for reporting.
 - Fix a problem with MPI_FILE_DELETE returning MPI_SUCCESS when deleting a non-existent file. Thanks to Wei-keng Liao for reporting.
 - Fix a problem with MPI_IMPROBE that could lead to hangs in subsequent MPI point to point or collective calls. Thanks to Chris Pattison for reporting.
 - Fix a problem when configure Open MPI for powerpc with --enable-mpi-cxx enabled. Thanks to Alastair McKinstry for reporting.
 - Fix a problem using MPI_IALLTOALL with MPI_IN_PLACE argument. Thanks to Chris Ward for reporting.
 - Fix a problem using MPI_RACCUMULATE with the Portals4 transport. Thanks to @PDeveze for reporting.
 - Fix an issue with static linking and duplicate symbols arising from PMIx Slurm components. Thanks to Limin Gu for reporting.
 - Fix a problem when using MPI dynamic memory windows. Thanks to Christoph Niethammer for reporting.
 - Fix a problem with Open MPI's pkgconfig files. Thanks to Alastair McKinstry for reporting.
 - Fix a problem with MPI_IREDUCE when the same buffer is supplied for the send and recv buffer arguments. Thanks to Valentin Petrov for reporting.
 - Fix a problem with atomic operations on PowerPC. Thanks to Paul Hargrove for reporting.

Known issues (to be addressed in v2.0.3):

- See the list of fixes slated for v2.0.3 here:
<https://github.com/open-mpi/mpi/milestone/23>

2.0.1 -- 2 September 2016

Bug fixes/minor improvements:

Appendix C. OpenMPI Release Information

- Short message latency and message rate performance improvements for all transports.
- Fix shared memory performance when using RDMA-capable networks. Thanks to Tetsuya Mishima and Christoph Niethammer for reporting.
- Fix bandwidth performance degradation in the yalla (MXM) PML. Thanks to Andreas Kempf for reporting the issue.
- Fix OpenSHMEM crash when running on non-Mellanox MXM-based networks. Thanks to Debendra Das for reporting the issue.
- Fix a crash occurring after repeated calls to `MPI_FILE_SET_VIEW` with predefined datatypes. Thanks to Eric Chamberland and Matthew Knepley for reporting and helping chase down this issue.
- Fix stdin propagation to MPI processes. Thanks to Jingchao Zhang for reporting the issue.
- Fix various runtime and portability issues by updating the PMIx internal component to v1.1.5.
- Fix process startup failures on Intel MIC platforms due to very large entries in `/proc/mounts`.
- Fix a problem with use of relative path for specifying executables to `mpirun/oshrun`. Thanks to David Schneider for reporting.
- Various improvements when running over portals-based networks.
- Fix thread-based race conditions with GNI-based networks.
- Fix a problem with `MPI_FILE_CLOSE` and `MPI_FILE_SET_SIZE`. Thanks to Cihan Altinay for reporting.
- Remove all use of `rand(3)` from within Open MPI so as not to perturb applications use of it. Thanks to Matias Cabral and Noel Rycroft for reporting.
- Fix crash in `MPI_COMM_SPAWN`.
- Fix types for `MPI_UNWEIGHTED` and `MPI_WEIGHTS_EMPTY`. Thanks to Lisandro Dalcin for reporting.
- Correctly report the name of `MPI_INTEGER16`.
- Add some missing MPI constants to the Fortran bindings.
- Fixed compile error when configuring Open MPI with `--enable-timing`.
- Correctly set the shared library version of `libompitrace.so`. Thanks to Alastair McKinstry for reporting.
- Fix errors in the `MPI_RPUT`, `MPI_RGET`, `MPI_RACCUMULATE`, and `MPI_RGET_ACCUMULATE` Fortran bindings. Thanks to Alfio Lazzaro and Joost VandeVondele for tracking this down.
- Fix problems with use of derived datatypes in non-blocking collectives. Thanks to Yuki Matsumoto for reporting.
- Fix problems with OpenSHMEM header files when using CMake. Thanks to Paul Kapinos for reporting the issue.
- Fix problem with use use of non-zero lower bound datatypes in collectives. Thanks to Hristo Iliev for reporting.
- Fix a problem with memory allocation within `MPI_GROUP_INTERSECTION`. Thanks to Lisandro Dalcin for reporting.
- Fix an issue with `MPI_ALLGATHER` for communicators that don't consist of two ranks. Thanks to David Love for reporting.
- Various fixes for collectives when used with esoteric MPI datatypes.
- Fixed corner cases of handling `DARRAY` and `HINDEXED_BLOCK` datatypes.
- Fix a problem with filesystem type check for OpenBSD. Thanks to Paul Hargrove for reporting.
- Fix some debug input within Open MPI internal functions. Thanks to Durga Choudhury for reporting.
- Fix a typo in a configury help message. Thanks to Paul Hargrove for reporting.

- Correctly support MPI_IN_PLACE in MPI_[I]ALLTOALL[V|W] and MPI_[I]EXSCAN.
- Fix alignment issues on SPARC platforms.

Known issues (to be addressed in v2.0.2):

- See the list of fixes slated for v2.0.2 here: <https://github.com/open-mpi/ompi/milestone/20>, and <https://github.com/open-mpi/ompi-release/milestone/19> (note that the "ompi-release" Github repo will be folded/absorbed into the "ompi" Github repo at some point in the future)

2.0.0 -- 12 July 2016

```
*****
*   Open MPI is now fully MPI-3.1 compliant
*****
```

Major new features:

- Many enhancements to MPI RMA. Open MPI now maps MPI RMA operations on to native RMA operations for those networks which support this capability.
- Greatly improved support for MPI_THREAD_MULTIPLE (when configured with `--enable-mpi-thread-multiple`).
- Enhancements to reduce the memory footprint for jobs at scale. A new MCA parameter, "mpi_add_procs_cutoff", is available to set the threshold for using this feature.
- Completely revamped support for memory registration hooks when using OS-bypass network transports.
- Significant OMPIO performance improvements and many bug fixes.
- Add support for PMIx - Process Management Interface for Exascale. Version 1.1.2 of PMIx is included internally in this release.
- Add support for PLFS file systems in Open MPI I/O.
- Add support for UCX transport.
- Simplify build process for Cray XC systems. Add support for using native SLURM.
- Add a `--tune mpirun` command line option to simplify setting many environment variables and MCA parameters.
- Add a new MCA parameter "orte_default_dash_host" to offer an analogue to the existing "orte_default_hostfile" MCA parameter.
- Add the ability to specify the number of desired slots in the mpirun `--host` option.

Changes in behavior compared to prior versions:

- In environments where mpirun cannot automatically determine the number of slots available (e.g., when using a hostfile that does not specify "slots", or when using `--host` without specifying a ":N" suffix to hostnames), mpirun now requires the use of "`-np N`" to specify how many MPI processes to launch.
- The MPI C++ bindings -- which were removed from the MPI standard in v3.0 -- are no longer built by default and will be removed in some

Appendix C. OpenMPI Release Information

- future version of Open MPI. Use the `--enable-mpi-cxx-bindings` configure option to build the deprecated/removed MPI C++ bindings.
- `ompi_info` now shows all components, even if they do not have MCA parameters. The `prettyprint` output now separates groups with a dashed line.
- OMPIO is now the default implementation of parallel I/O, with the exception for Lustre parallel filesystems (where ROMIO is still the default). The default selection of OMPIO vs. ROMIO can be controlled via the `"--mca io ompi|romio"` command line switch to `mpirun`.
- Per Open MPI's versioning scheme (see the README), increasing the major version number to 2 indicates that this version is not ABI-compatible with prior versions of Open MPI. You will need to recompile MPI and OpenSHMEM applications to work with this version of Open MPI.
- Removed checkpoint/restart code due to loss of maintainer. :-)
- Change the behavior for handling certain signals when using PSM and PSM2 libraries. Previously, the PSM and PSM2 libraries would trap certain signals in order to generate tracebacks. The mechanism was found to cause issues with Open MPI's own error reporting mechanism. If not already set, Open MPI now sets the `IPATH_NO_BACKTRACE` and `HFI_NO_BACKTRACE` environment variables to disable PSM/PSM2's handling these signals.

Removed legacy support:

- Removed support for OS X Leopard.
- Removed support for Cray XT systems.
- Removed VampirTrace.
- Removed support for Myrinet/MX.
- Removed legacy collective module:ML.
- Removed support for Alpha processors.
- Removed `--enable-mpi-profiling` configure option.

Known issues (to be addressed in v2.0.1):

- See the list of fixes slated for v2.0.1 here:
<https://github.com/open-mpi/ompi/milestone/16>, and
<https://github.com/open-mpi/ompi-release/milestone/16>
(note that the "ompi-release" Github repo will be folded/absorbed into the "ompi" Github repo at some point in the future)
- `ompi-release#986`: Fix data size counter for large ops with `fcoll/static`
- `ompi-release#987`: Fix OMPIO performance on Lustre
- `ompi-release#1013`: Fix potential inconsistency in `btl/openib` default settings
- `ompi-release#1014`: Do not return `MPI_ERR_PENDING` from collectives
- `ompi-release#1056`: Remove dead profile code from `oshmem`
- `ompi-release#1081`: Fix `MPI_IN_PLACE` checking for `IALLTOALL{V|W}`
- `ompi-release#1081`: Fix memchecker in `MPI_IALLTOALLW`
- `ompi-release#1081`: Support `MPI_IN_PLACE` in `MPI_(I)ALLTOALLW` and `MPI_(I)EXSCAN`
- `ompi-release#1107`: Allow future PMIx support for RM spawn limits
- `ompi-release#1108`: Fix sparse group process reference counting
- `ompi-release#1109`: If specified to be oversubscribed, disable binding
- `ompi-release#1122`: Allow NULL arrays for empty datatypes
- `ompi-release#1123`: Fix signed vs. unsigned compiler warnings
- `ompi-release#1123`: Make max hostname length uniform across code base

- ompi-release#1127: Fix MPI_Compare_and_swap
- ompi-release#1127: Fix MPI_Win_lock when used with MPI_Win_fence
- ompi-release#1132: Fix typo in help message for --enable-mca-no-build
- ompi-release#1154: Ensure pairwise coll algorithms disqualify themselves properly
- ompi-release#1165: Fix typos in debugging/verbose message output
- ompi-release#1178: Fix ROMIO filesystem check on OpenBSD 5.7
- ompi-release#1197: Fix Fortran pthread configure check
- ompi-release#1205: Allow using external PMIx 1.1.4 and 2.0
- ompi-release#1215: Fix configure to support the NAG Fortran compiler
- ompi-release#1220: Fix combiner args for MPI_HINDEXED_BLOCK
- ompi-release#1225: Fix combiner args for MPI_DARRAY
- ompi-release#1226: Disable old memory hooks with recent gcc versions
- ompi-release#1231: Fix new "patcher" support for some XLC platforms
- ompi-release#1244: Fix Java error handling
- ompi-release#1250: Ensure TCP is not selected for RDMA operations
- ompi-release#1252: Fix verbose output in coll selection
- ompi-release#1253: Set a default name for user-defined MPI_Op
- ompi-release#1254: Add count==0 checks in some non-blocking colls
- ompi-release#1258: Fix "make distclean" when using external pmix/hwloc/libevent
- ompi-release#1260: Clean up/uniform mca/coll/base memory management
- ompi-release#1261: Remove "patcher" warning message for static builds
- ompi-release#1263: Fix IO MPI_Request for 0-size read/write
- ompi-release#1264: Add blocking fence for SLURM operations

Bug fixes / minor enhancements:

- Updated internal/embedded copies of third-party software:
 - Update the internal copy of ROMIO to that which shipped in MPICH 3.1.4.
 - Update internal copy of libevent to v2.0.22.
 - Update internal copy of hwloc to v1.11.2.
- Notable new MCA parameters:
 - opal_progress_lp_call_ration: Control how often low-priority callbacks are made during Open MPI's main progress loop.
 - opal_common_verbs_want_fork_support: This replaces the btl_openib_want_fork_support parameter.
- Add --with-platform-patches-dir configure option.
- Add --with-pmi-libdir configure option for environments that install PMI libs in a non-default location.
- Various configure-related compatibility updates for newer versions of libibverbs and OFED.
- Numerous fixes/improvements to orte-dvm. Special thanks to Mark Santcroos for his help.
- Fix a problem with timer code on ia32 platforms. Thanks to Paul Hargrove for reporting this and providing a patch.
- Fix a problem with use of a 64 bit atomic counter. Thanks to Paul Hargrove for reporting.
- Fix a problem with singleton job launching. Thanks to Lisandro Dalcin for reporting.
- Fix a problem with use of MPI_UNDEFINED with MPI_COMM_SPLIT_TYPE. Thanks to Lisandro Dalcin for reporting.
- Silence a compiler warning in PSM MTL. Thanks to Adrian Reber for reporting this.
- Properly detect Intel TrueScale and OmniPath devices in the ACTIVE state. Thanks to Durga Choudhury for reporting the issue.

Appendix C. OpenMPI Release Information

- Fix detection and use of Solaris Studio 12.5 (beta) compilers. Thanks to Paul Hargrove for reporting and debugging.
- Fix various small memory leaks.
- Allow NULL arrays when creating empty MPI datatypes.
- Replace use of `alloca` with `malloc` for certain datatype creation functions. Thanks to Bogdan Sataric for reporting this.
- Fix use of `MPI_LB` and `MPI_UB` in creation of of certain MPI datatypes. Thanks to Gus Correa for helping to fix this.
- Implement a workaround for a GNU Libtool problem. Thanks to Eric Schnetter for reporting and fixing.
- Improve `hcoll` library detection in `configure`. Thanks to David Shrader and Ake Sandgren for reporting this.
- Miscellaneous minor bug fixes in the `hcoll` component.
- Miscellaneous minor bug fixes in the `ugni` component.
- Fix problems with XRC detection in OFED 3.12 and older releases. Thanks to Paul Hargrove for his analysis of this problem.
- Update (non-standard/experimental) Java MPI interfaces to support MPI-3.1 functionality.
- Fix an issue with MCA parameters for Java bindings. Thanks to Takahiro Kawashima and Siegmur Gross for reporting this issue.
- Fix a problem when using persistent requests in the Java bindings. Thanks to Nate Chambers for reporting.
- Fix problem with Java bindings on OX X 10.11. Thanks to Alexander Daryin for reporting this issue.
- Fix a performance problem for large messages for Cray XC systems. Thanks to Jerome Vienne for reporting this.
- Fix an issue with `MPI_WIN_LOCK_ALL`. Thanks to Thomas Jahns for reporting.
- Fix an issue with passing a parameter to `configure` multiple times. Thanks to QuesarVII for reporting and supplying a fix.
- Add support for ALPS resource allocation system on Cray CLE 5.2 and later. Thanks to Mark Santcross.
- Corrections to the `HACKING` file. Thanks to Maximilien Levesque.
- Fix an issue with user supplied reduction operator functions. Thanks to Rupert Nash for reporting this.
- Fix an issue with an internal list management function. Thanks to Adrian Reber for reporting this.
- Fix a problem with MPI-RMA PSCW epochs. Thanks to Berk Hess for reporting this.
- Fix a problem in neighborhood collectives. Thanks to Lisandro Dalcin for reporting.
- Fix `MPI_IREDUCE_SCATTER_BLOCK` for a one-process communicator. Thanks to Lisandro Dalcin for reporting.
- Add (Open MPI-specific) additional flavors to `MPI_COMM_SPLIT_TYPE`. See `MPI_Comm_split_type(3)` for details. Thanks to Nick Andersen for supplying this enhancement.
- Improve closing of file descriptors during the job launch phase. Thanks to Piotr Lesnicki for reporting and providing this enhancement.
- Fix a problem in `MPI_GET_ACCUMULATE` and `MPI_RGET_ACCUMULATE` when using `Portals4`. Thanks to Nicolas Chevalier for reporting.
- Use correct include file for `lstat` prototype in `ROMIO`. Thanks to William Throwe for finding and providing a fix.
- Add missing Fortran bindings for `MPI_WIN_ALLOCATE`. Thanks to Christoph Niethammer for reporting and fixing.

- Fortran related fixes to handle Intel 2016 compiler. Thanks to Fabrice Roy for reporting this.
- Fix a Fortran linkage issue. Thanks to Macro Atzeri for finding and suggesting a fix.
- Fix problem with using BIND(C) for Fortran bindings with logical parameters. Thanks to Paul Romano for reporting.
- Fix an issue with use of DL-related macros in opal library. Thanks to Scott Atchley for finding this.
- Fix an issue with parsing mpirun command line options which contain colons. Thanks to Lev Given for reporting.
- Fix a problem with Open MPI's package configury files. Thanks to Christoph Junghans for reporting.
- Fix a typo in the MPI_INTERCOMM_MERGE man page. Thanks To Harald Servat for reporting and correcting.
- Update man pages for non-blocking sends per MPI 3.1 standard. Thanks to Alexander Pozdneev for reporting.
- Fix problem when compiling against PVFS2. Thanks to Dave Love for reporting.
- Fix problems with MPI_NEIGHBOR_ALLTOALL{V,W}. Thanks to Willem Vermin for reporting this issue.
- Fix various compilation problems on Cygwin. Thanks to Marco Atzeri for supplying these fixes.
- Fix problem with resizing of subarray and darray data types. Thanks to Keith Bennett and Dan Garmann for reporting.
- Fix a problem with MPI_COMBINER_RESIZED. Thanks to James Ramsey for the report.
- Fix an hwloc binding issue. Thanks to Ben Menadue for reporting.
- Fix a problem with the shared memory (sm) BTL. Thanks to Peter Wind for the report.
- Fixes for heterogeneous support. Thanks to Siegmur Gross for reporting.
- Fix a problem with memchecker. Thanks to Clinton Simpson for reporting.
- Fix a problem with MPI_UNWEIGHTED in topology functions. Thanks to Jun Kudo for reporting.
- Fix problem with a MCA parameter base filesystem types. Thanks to Siegmur Gross for reporting.
- Fix a problem with some windows info argument types. Thanks to Alastair McKinstry for reporting.

1.10.6 - 17 Feb 2017

- Fix bug in timer code that caused problems at optimization settings greater than 2
- OSHMEM: make mmap allocator the default instead of sysv or verbs
- Support MPI_Dims_create with dimension zero
- Update USNIC support
- Prevent 64-bit overflow on timer counter
- Add support for forwarding signals
- Fix bug that caused truncated messages on large sends over TCP BTL
- Fix potential infinite loop when printing a stacktrace

1.10.5 - 19 Dec 2016

- Update UCX APIs

Appendix C. OpenMPI Release Information

- Fix bug in darray that caused MPI/IO failures
- Use a `MPI_Get_library_version()` like string to tag the debugger DLL. Thanks to Alastair McKinstry for the report
- Fix multi-threaded race condition in `coll/libnbc`
- Several fixes to OSHMEM
- Fix bug in UCX support due to uninitialized field
- Fix `MPI_Ialltoallv` with `MPI_IN_PLACE` and without MPI param check
- Correctly reset receive request type before init. Thanks Chris Pattison for the report and test case.
- Fix bug in `iallgather[v]`
- Fix concurrency issue with `MPI_Comm_accept`. Thanks to Pieter Noordhuis for the patch
- Fix `ompi_coll_base_{gather,scatter}_intra_binomial`
- Fixed an issue with `MPI_Type_get_extent` returning the wrong extent for distributed array datatypes.
- Re-enable use of `rtdtsc` instruction as a monotonic clock source if the processor has a core-invariant tsc. This is a partial fix for a performance regression introduced in Open MPI v1.10.3.

1.10.4 - 01 Sept 2016

- Fix assembler support for MIPS
- Improve memory handling for temp buffers in collectives
- Fix `[all]reduce` with non-zero lower bound datatypes
Thanks Hristo Iliev for the report
- Fix non-standard ddt handling. Thanks Yuki Matsumoto for the report
- Various `libnbc` fixes. Thanks Yuki Matsumoto for the report
- Fix typos in request RMA bindings for Fortran. Thanks to @alazzaro and @vondele for the assist
- Various bug fixes and enhancements to collective support
- Fix predefined types mapping in `hcoll`
- Revive the `coll/sync` component to resolve unexpected message issues during tight loops across collectives
- Fix typo in wrapper compiler for Fortran static builds

1.10.3 - 15 June 2016

- Fix zero-length datatypes. Thanks to Wei-keng Liao for reporting the issue.
- Minor manpage cleanups
- Implement atomic support in OSHMEM/UCX
- Fix support of `MPI_COMBINER_RESIZED`. Thanks to James Ramsey for the report
- Fix computation of `#cpus` when `--use-hwthread-cpus` is used
- Add entry points for `Allgatherv`, `iAllgatherv`, `Reduce`, and `iReduce` for the `HCOLL` library
- Fix an `HCOLL` integration bug that could signal completion of request while still being worked
- Fix computation of `cores` when SMT is enabled. Thanks to Ben Menadue for the report
- Various USNIC fixes

- Create a datafile in the per-proc directory in order to make it unique per communicator. Thanks to Peter Wind for the report
- Fix zero-size malloc in one-sided pt-to-pt code. Thanks to Lisandro Dalcin for the report
- Fix MPI_Get_address when passed MPI_BOTTOM to not return an error. Thanks to Lisandro Dalcin for the report
- Fix MPI_TYPE_SET_ATTR with NULL value. Thanks to Lisandro Dalcin for the report
- Fix various Fortran08 binding issues
- Fix memchecker no-data case. Thanks to Clinton Stimpson for the report
- Fix CUDA support under OS-X
- Fix various OFI/MTL integration issues
- Add MPI_T man pages
- Fix one-sided pt-to-pt issue by preventing communication from happening before a target enters a fence, even in the no-precede case
- Fix a bug that disabled Totalview for MPMD use-case
- Correctly support MPI_UNWEIGHTED in topo-graph-neighbors. Thanks to Jun Kudo for the report
- Fix singleton operations under SLURM when PMI2 is enabled
- Do not use MPI_IN_PLACE in neighborhood collectives for non-blocking collectives (libnbc). Thanks to Jun Kudo for the report
- Silence autogen deprecation warnings for newer versions of Perl
- Do not return MPI_ERR_PENDING from collectives
- Use type int* for MPI_WIN_DISP_UNIT, MPI_WIN_CREATE_FLAVOR, and MPI_WIN_MODEL. Thanks to Alastair McKinstry for the report
- Fix register_datarep stub function in IO/OMPIO. Thanks to Eric Chamberland for the report
- Fix a bus error on MPI_WIN_[POST,START] in the shared memory one-sided component
- Add several missing MPI_WIN_FLAVOR constants to the Fortran support
- Enable connecting processes from different subnets using the openib BTL
- Fix bug in basic/barrier algorithm in OSHMEM
- Correct process binding for the --map-by node case
- Include support for subnet-to-subnet routing over InfiniBand networks
- Fix usnic resource check
- AUTHORS: Fix an errant reference to Subversion IDs
- Fix affinity for MPMD jobs running under LSF
- Fix many Fortran binding bugs
- Fix 'MPI_IN_PLACE'-related bugs
- Fix PSM/PSM2 support for singleton operations
- Ensure MPI transports continue to progress during RTE barriers
- Update HWLOC to 1.9.1 end-of-series
- Fix a bug in the Java command line parser when the -Djava.library.path options was given by the user
- Update the MTL/OFI provider selection behavior
- Add support for clock_gettime on Linux.
- Correctly detect and configure for Solaris Studio 12.5 beta compilers
- Correctly compute #slots when -host is used for MPMD case
- Fix a bug in the hcoll collectives due to an uninitialized field
- Do not set a binding policy when oversubscribing a node
- Fix hang in intercommunicator operations when oversubscribed
- Speed up process termination during MPI_Abort
- Disable backtrace support by default in the PSM/PSM2 libraries to prevent unintentional conflicting behavior.

Appendix C. OpenMPI Release Information

1.10.2: 26 Jan 2016

```
*****
*   OSHMEM is now 1.2 compliant
*****
```

- Fix NBC_Copy for legitimate zero-size messages
- Fix multiple bugs in OSHMEM
- Correctly handle mpirun --host <user>@<ip-address>
- Centralize two MCA params to avoid duplication between OMPI and OSHMEM layers: opal_abort_delay and opal_abort_print_stack
- Add support for Fujitsu compilers
- Add UCX support for OMPI and OSHMEM
- Correctly handle oversubscription when not given directives to permit it. Thanks to @ammorel for reporting it
- Fix rpm spec file to not include the /usr directory
- Add Intel HFI1 default parameters for the openib BTL
- Resolve symbol conflicts in the PSM2 library
- Add ability to empty the rgpusm cache when full if requested
- Fix another libtool bug when -L requires a space between it and the path. Thanks to Eric Schnetter for the patch.
- Add support for OSHMEM v1.2 APIs
- Improve efficiency of oshmem_preconnect_all algorithm
- Fix bug in buffered sends support
- Fix double free in edge case of mpirun. Thanks to @jsharpe for the patch
- Multiple one-sided support fixes
- Fix integer overflow in the tuned "reduce" collective when using buffers larger than INT_MAX in size
- Fix parse of user environment variables in mpirun. Thanks to Stefano Garzarella for the patch
- Performance improvements in PSM2 support
- Fix NBS iBarrier for inter-communicators
- Fix bug in vader BTL during finalize
- Improved configure support for Fortran compilers
- Fix rank_file mapper to support default --slot-set. Thanks to Matt Thompson for reporting it
- Update MPI_Testsome man page. Thanks to Eric Schnetter for the suggestion
- Fix missing resize of the returned type for subarray and darray types. Thanks to Keith Bennett and Dan Garmann for reporting it
- Fix Java support on OSX 10.11. Thanks to Alexander Daryin for reporting the problem
- Fix some compilation issues on Solaris 11.2. Thanks to Paul Hargrove for his continued help in such areas

1.10.1: 4 Nov 2015

- Workaround an optimization problem with gcc compilers >= 4.9.2 that causes problems with memory registration, and forced

- mpi_leave_pinned to default to 0 (i.e., off). Thanks to @oere for the fix.
- Fix use of MPI_LB and MPI_UB in subarray and darray datatypes. Thanks to Gus Correa and Dimitar Pashov for pointing out the issue.
 - Minor updates to mpi_show_mpi_alloc_mem_leaks and omp_debug_show_handle_leaks functionality.
 - Fix segv when invoking non-blocking reductions with a user-defined operation. Thanks to Rupert Nash and Georg Geiser for identifying the issue.
 - No longer probe for PCI topology on Solaris (unless running as root).
 - Fix for Intel Parallel Studio 2016 ifort partial support of the !GCC\$ pragma. Thanks to Fabrice Roy for reporting the problem.
 - Bunches of Coverity / static analysis fixes.
 - Fixed ROMIO to look for lstat in <sys/stat.h>. Thanks to William Thrope for submitting the patch both upstream and to Open MPI.
 - Fixed minor memory leak when attempting to open plugins.
 - Fixed type in MPI_IBARRIER C prototype. Thanks to Harald Servat for reporting the issue.
 - Add missing man pages for MPI_WIN_CREATE_DYNAMIC, MPI_WIN_ATTACH, MPI_WIN_DETACH, MPI_WIN_ALLOCATE, MPI_WIN_ALLOCATE_SHARED.
 - When mpirun-launching new applications, only close file descriptors that are actually open (resulting in a faster launch in some environments).
 - Fix "test ==" issues in Open MPI's configure script. Thank to Kevin Buckley for pointing out the issue.
 - Fix performance issue in usnic BTL: ensure progress thread is throttled back to not aggressively steal CPU cycles.
 - Fix cache line size detection on POWER architectures.
 - Add missing #include in a few places. Thanks to Orion Poplawski for supplying the patch.
 - When OpenSHMEM building is disabled, no longer install its header files, help files, or man pages. Add man pages for oshrun, oshcc, and oshfort.
 - Fix mpi_f08 implementations of MPI_COMM_SET_INFO, and profiling versions of MPI_BUFFER_DETACH, MPI_WIN_ALLOCATE, MPI_WIN_ALLOCATE_SHARED, MPI_WTICK, and MPI_WTIME.
 - Add orte_rmaps_dist_device MCA param, allowing users to map near a specific device.
 - Various updates/fixes to the openib BTL.
 - Add missing defaults for the Mellanox ConnectX 3 card to the openib BTL.
 - Minor bug fixes in the OFI MTL.
 - Various updates to Mellanox's MXM, hcoll, and FCA components.
 - Add OpenSHMEM man pages. Thanks to Tony Curtis for sharing the man pages files from openshmem.org.
 - Add missing "const" attributes to MPI_COMPARE_AND_SWAP, MPI_FETCH_AND_OP, MPI_RACCUMULATE, and MPI_WIN_DETACH prototypes. Thanks to Michael Knobloch and Takahiro Kawashima for bringing this to our attention.
 - Fix linking issues on some platforms (e.g., SLES 12).
 - Fix hang on some corner cases when MPI applications abort.
 - Add missing options to mpirun man page. Thanks to Daniel Letai for bringing this to our attention.
 - Add new --with-platform-patches-dir configure option
 - Adjust relative selection priorities to ensure that MTL support is favored over BTL support when both are available

Appendix C. OpenMPI Release Information

- Use CUDA IPC for all sized messages for performance

1.10.0: 25 Aug 2015

- ** NOTE: The v1.10.0 release marks the transition to Open MPI's new version numbering scheme. The v1.10.x release series is based on the v1.8.x series, but with a few new features. v2.x will be the next series after the v1.10.x series, and complete the transition to the new version numbering scheme. See README for more details on the new versioning scheme.
- **
- ** NOTE: In accordance with OMPI version numbering, the v1.10 is *not* API compatible with the v1.8 release series.
- Added libfabric support (see README for more details):
 - usNIC BTL updated to use libfabric.
 - Added OFI MTL (usable with PSM in libfabric v1.1.0).
- Added Intel Omni-Path support via new PSM2 MTL.
- Added "yalla" PML for faster MXM support.
- Removed support for MX
- Added persistent distributed virtual machine (pDVM) support for fast workflow executions.
- Fixed typo in GCC inline assembly introduced in Open MPI v1.8.8. Thanks to Paul Hargrove for pointing out the issue.
- Add missing man pages for MPI_Win_get|set_info(3).
- Ensure that session directories are cleaned up at the end of a run.
- Fixed linking issues on some OSs where symbols of dependent libraries are not automatically publicly available.
- Improve hcoll and fca configury library detection. Thanks to David Shrader for helping track down the issue.
- Removed the LAMA mapper (for use in setting affinity). Its functionality has been largely superseded by other mpirun CLI options.
- CUDA: Made the asynchronous copy mode be the default.
- Fix a malloc(0) warning in MPI_IREDUCE_SCATTER_BLOCK. Thanks to Lisandro Dalcin for reporting the issue.
- Fix typo in MPI_Scatter(3) man page. Thanks to Akshay Venkatesh for noticing the mistake.
- Add rudimentary protection from TCP port scanners.
- Fix typo in Open MPI error handling. Thanks to Åke Sandgren for pointing out the error.
- Increased the performance of the CM PML (i.e., the Portals, PSM, PSM2, MXM, and OFI transports).
- Restored visibility of blocking send requests in message queue debuggers (e.g., TotalView, DDT).
- Fixed obscure IPv6-related bug in the TCP BTL.
- Add support for the "no_locks" MPI_Info key for one-sided functionality.
- Fixed ibv_fork support for verbs-based networks.
- Fixed a variety of small bugs in OpenSHMEM.
- Fixed MXM configure with additional CPPFLAGS and LDFLAGS. Thanks to David Shrader for the patch.
- Fixed incorrect memalign threshold in the openib BTL. Thanks to

Xavier Besseron for pointing out the issue.

1.8.8: 5 Aug 2015

- Fix a segfault in MPI_FINALIZE with the PSM MTL.
- Fix mpi_f08 sentinels (e.g., MPI_STATUS_IGNORE) handling.
- Set some additional MXM default values for OSHMEM.
- Fix an invalid memory access in MPI_MRECV and MPI_IMRECV.
- Include two fixes that were mistakenly left out of the official v1.8.7 tarball:
 - Fixed MPI_WIN_POST and MPI_WIN_START for zero-size messages
 - Protect the OOB TCP ports from segfaulting when accessed by port scanners

1.8.7: 15 Jul 2015

- ** NOTE: v1.8.7 technically breaks ABI with prior versions
- ** in the 1.8 series because it repairs two incorrect API
- ** signatures. However, users will only need to recompile
- ** if they were using those functions - which they couldn't
- ** have been, because the signatures were wrong :-)
- Plugged a memory leak that impacted blocking sends
- Fixed incorrect declaration for MPI_T_pvar_get_index and added missing return code MPI_T_INVALID_NAME.
- Fixed an uninitialized variable in PMI2 support
- Added new vendor part id for Mellanox ConnectX4-LX
- Fixed NBC_Copy for legitimate zero-size messages
- Fixed MPI_Win_post and MPI_Win_start for zero-size messages
- Protect the OOB ports from segfaulting when accessed by port scanners
- Fixed several Fortran typos
- Fixed configure detection of XRC support
- Fixed support for highly heterogeneous systems to avoid memory corruption when printing out the bindings

1.8.6: 17 Jun 2015

- Fixed memory leak on Mac OS-X exposed by TCP keepalive
- Fixed keepalive support to ensure that daemon/node failure results in complete job cleanup
- Update Java binding support
- Fixed MPI_THREAD_MULTIPLE bug in vader shared memory BTL
- Fixed issue during shutdown when CUDA initialization wasn't complete
- Fixed orted environment when no prefix given
- Fixed trivial typo in MPI_Neighbor_allgather manpage
- Fixed tree-spawn support for sh and ksh shells
- Several data type fixes
- Fixed IPv6 support bug
- Cleaned up an unlikely build issue

Appendix C. OpenMPI Release Information

- Fixed PMI2 process map parsing for cyclic mappings
- Fixed memalign threshold in openib BTL
- Fixed debugger access to message queues for blocking send/recv

1.8.5

- Fixed configure problems in some cases when using an external hwloc installation. Thanks to Erick Schnetter for reporting the error and helping track down the source of the problem.
- Fixed linker error on OS X when using the clang compiler. Thanks to Erick Schnetter for reporting the error and helping track down the source of the problem.
- Fixed MPI_THREAD_MULTIPLE deadlock error in the vader BTL. Thanks to Thomas Klimpel for reporting the issue.
- Fixed several Valgrind warnings. Thanks for Lisandro Dalcin for contributing a patch fixing some one-sided code paths.
- Fixed version compatibility test in OOB that broke ABI within the 1.8 series. NOTE: this will not resolve the problem between pre-1.8.5 versions, but will fix it going forward.
- Fix some issues related to running on Intel Xeon Phi coprocessors.
- Opportunistically switch away from using GNU Libtool's libltdl library when possible (by default).
- Fix some VampirTrace errors. Thanks to Paul Hargrove for reporting the issues.
- Correct default binding patterns when --use-hwthread-cpus was specified and nprocs <= 2.
- Fix warnings about -finline-functions when compiling with clang.
- Updated the embedded hwloc with several bug fixes, including the "duplicate Lhwloc1 symbol" that multiple users reported on some platforms.
- Do not error when mpirun is invoked with with default bindings (i.e., no binding was specified), and one or more nodes do not support bindings. Thanks to Annu Desari for pointing out the problem.
- Let root invoke "mpirun --version" to check the version without printing the "Don't run as root!" warnings. Thanks to Robert McLay for the suggestion.
- Fixed several bugs in OpenSHMEM support.
- Extended vader shared memory support to 32-bit architectures.
- Fix handling of very large datatypes. Thanks to Bogdan Sataric for the bug report.
- Fixed a bug in handling subarray MPI datatypes, and a bug when using MPI_LB and MPI_UB. Thanks to Gus Correa for pointing out the issue.
- Restore user-settable bandwidth and latency PML MCA variables.
- Multiple bug fixes for cleanup during MPI_FINALIZE in unusual situations.
- Added support for TCP keepalive signals to ensure timely termination when sockets between daemons cannot be created (e.g., due to a firewall).
- Added MCA parameter to allow full use of a SLURM allocation when started from a tool (supports LLNL debugger).
- Fixed several bugs in the configure logic for PMI and hwloc.
- Fixed incorrect interface index in TCP communications setup. Thanks

- to Mark Kettenis for spotting the problem and providing a patch.
- Fixed MPI_IREDUCE_SCATTER with single-process communicators when MPI_IN_PLACE was not used.
- Added XRC support for OFED v3.12 and higher.
- Various updates and bug fixes to the Mellanox hcoll collective support.
- Fix problems with Fortran compilers that did not support REAL*16/COMPLEX*32 types. Thanks to Orion Poplawski for identifying the issue.
- Fixed problem with rpath/runpath support in pkg-config files. Thanks to Christoph Junghans for notifying us of the issue.
- Man page fixes:
 - Removed erroneous "color" discussion from MPI_COMM_SPLIT_TYPE. Thanks to Erick Schnetter for spotting the outdated text.
 - Fixed prototypes for MPI_IBARRIER. Thanks to Maximilian for finding the issue.
 - Updated docs about buffer usage in non-blocking communications. Thanks to Alexander Pozdnev for citing the outdated text.
 - Added documentation about the 'ompi_unique' MPI_Info key with MPI_PUBLISH_NAME.
 - Fixed typo in MPI_INTERCOMM_MERGE. Thanks to Harald Servat for noticing and sending a patch.
 - Updated configure paths in HACKING. Thanks to Maximilien Levesque for the fix.
 - Fixed Fortran typo in MPI_WIN_LOCK_ALL. Thanks to Thomas Jahns for pointing out the issue.
- Fixed a number of MPI one-sided bugs.
- Fixed MPI_COMM_SPAWN when invoked from a singleton job.
- Fixed a number of minor issues with CUDA support, including registering of shared memory and supporting reduction support for GPU buffers.
- Improved support for building OMPI on Cray platforms.
- Fixed performance regression introduced by the inadvertent default enabling of MPI_THREAD_MULTIPLE support.

1.8.4

- Fix MPI_SIZEOF; now available in mpif.h for modern Fortran compilers (see README for more details). Also fixed various compiler/linker errors.
- Fixed inadvertant Fortran ABI break between v1.8.1 and v1.8.2 in the mpi interface module when compiled with gfortran >= v4.9.
- Fix various MPI_THREAD_MULTIPLE issues in the TCP BTL.
- mpirun no longer requires the --hetero-nodes switch; it will automatically detect when running in heterogeneous scenarios.
- Update LSF support, to include revamped affinity functionality.
- Update embedded hwloc to v1.9.1.
- Fixed max registerable memory computation in the openib BTL.
- Updated error message when debuggers are unable to find various symbols/types to be more clear. Thanks to Dave Love for raising the issue.
- Added proper support for LSF and PBS/Torque libraries in static builds.
- Rankfiles now support physical processor IDs.

Appendix C. OpenMPI Release Information

- Fixed potential hang in MPI_ABORT.
- Fixed problems with the PSM MTL and "re-connect" scenarios, such as MPI_INTERCOMM_CREATE.
- Fix MPI_IREDUCE_SCATTER with a single process.
- Fix (rare) race condition in stdout/stderr funneling to mpirun where some trailing output could get lost when a process terminated.
- Removed inadvertent change that set --enable-mpi-thread-multiple "on" by default, thus impacting performance for non-threaded apps.
- Significantly reduced startup time by optimizing internal hash table implementation.
- Fixed OS X linking with the Fortran mpi module when used with gfortran >= 4.9. Thanks to Github user yafshar for raising the issue.
- Fixed memory leak on Cygwin platforms. Thanks for Marco Atzeri for reporting the issue.
- Fixed seg fault in neighborhood collectives when the degree of the topology is higher than the communicator size. Thanks to Lisandro Dalcin for reporting the issue.
- Fixed segfault in neighborhood collectives under certain use-cases.
- Fixed various issues regarding Solaris support. Thanks to Siegmur Gross for patiently identifying all the issues.
- Fixed PMI configure tests for certain Slurm installation patterns.
- Fixed param registration issue in Java bindings. Thanks to Takahiro Kawashima and Siegmur Gross for identifying the issue.
- Several man page fixes.
- Silence several warnings and close some memory leaks (more remain, but it's better than it was).
- Re-enabled the use of CMA and knem in the shared memory BTL.
- Updated mpirun manpage to correctly explain new map/rank/binding options.
- Fixed MPI_IALLGATHER problem with intercommunicators. Thanks for Takahiro Kawashima for the patch.
- Numerous updates and performance improvements to OpenSHMEM.
- Turned off message coalescing in the openib BTL until a proper fix for that capability can be provided (tentatively expected for 1.8.5)
- Fix a bug in iof output that dates back to the dinosaurs which would output extra bytes if the system was very heavily loaded
- Fix a bug where specifying mca_component_show_load_errors=0 could cause ompi_info to segfault
- Updated valgrind suppression file

1.8.3

- Fixed application abort bug to ensure that MPI_Abort exits appropriately and returns the provided exit status
- Fixed some alignment (not all) issues identified by Clang
- Allow CUDA-aware to work with nonblocking collectives. Forces packing to happen when using GPU buffers.
- Fixed configure test issue with Intel 2015 Fortran compiler
- Fixed some PGI-related errors
- Provide better help message when encountering a firewall
- Fixed MCA parameter quoting to protect multi-word params and params that contain special characters
- Improved the bind-to help message to clarify the defaults

- Add new MPI-3.1 tools interface
- Several performance optimizations and memory leak cleanups
- Turn off the coll/ml plugin unless specifically requested as it remains in an experimental state
- Fix LSF support by adding required libraries for the latest LSF releases. Thanks to Joshua Randal for supplying the initial patches.

1.8.2

- Fix auto-wireup of OOB, allowing ORTE to automatically test all available NICs
- "Un-deprecate" pernode, npernode, and npersocket options by popular demand
- Add missing Fortran bindings for MPI_WIN_LOCK_ALL, MPI_WIN_UNLOCK_ALL, and MPI_WIN_SYNC.
- Fix cascading/over-quoting in some cases with the rsh/ssh-based launcher. Thanks to multiple users for raising the issue.
- Properly add support for gfortran 4.9 ignore TKR pragma (it was erroneously only partially added in v1.7.5). Thanks to Marcus Daniels for raising the issue.
- Update/improve help messages in the usnic BTL.
- Resolve a race condition in MPI_Abort.
- Fix obscure cases where static linking from wrapper compilers would fail.
- Clarify the configure --help message about when OpenSHMEM is enabled/disabled by default. Thanks to Paul Hargrove for the suggestion.
- Align pages properly where relevant. Thanks to Paul Hargrove for identifying the issue.
- Various compiler warning and minor fixes for OpenBSD, FreeBSD, and Solaris/SPARC. Thanks to Paul Hargrove for the patches.
- Properly pass function pointers from Fortran to C in the mpi_f08 module, thereby now supporting gfortran 4.9. Thanks to Tobias Burnus for assistance and testing with this issue.
- Improve support for Cray CLE 5.
- Fix mpirun regression: ensure exit status is non-zero if mpirun is terminated due to signal.
- Improved CUDA efficiency of asynchronous copies.
- Fix to parameter type in MPI_Type_indexed.3. Thanks to Bastian Beischer for reporting the mistake.
- Fix NUMA distance calculations in the openib BTL.
- Decrease time required to shut down mpirun at the end of a job.
- More RMA fixes.
- More hostfile fixes from Tetsuya Mishima.
- Fix darray issue where UB was not computed correctly.
- Fix mpi_f08 parameter name for MPI_GET_LIBRARY_VERSION. Thanks to Junchao Zhang for pointing out the issue.
- Ensure mpirun aborts properly when unable to map processes in scheduled environments.
- Ensure that MPI RMA error codes show up properly. Thanks to Lisandro Dalcin for reporting the issue.
- Minor bug fixes and improvements to the bash and zsh mpirun

Appendix C. OpenMPI Release Information

- autocompletion scripts.
- Fix sequential mpirun process mapper. Thanks to Bill Chen for reporting the issue.
- Correct SLURM stdout/stderr redirection.
- Added missing portals 4 files.
- Performance improvements for blocking sends and receives.
- Lots of cleanup to the ml collective component
- Added new Java methods to provide full MPI coverage
- Many OSHMEM cleanups
- Prevent comm_spawn from automatically launching a VM across all available nodes
- Close many memory leaks to achieve valgrind-clean operation
- Better handling of TCP connection discovery for mismatched networks where we don't have a direct 1:1 subnet match between nodes
- Prevent segfault when OMPI info tools are used in pipes and user exits one step of that pipe before completing output

1.8.1

- Fix for critical bug: mpirun removed files (but not directories) from / when run as root. Thanks to Jay Fenlason and Orion Poplawski for bringing the issue to our attention and helping identify the fix.

1.8

- Commit upstream ROMIO fix for mixed NFS+local filesystem environments.
- Several fixes for MPI-3 one-sided support. For example, arbitrary-length datatypes are now supported.
- Add config support for the Mellanox ConnectX 4 card.
- Add missing MPI_COMM_GET|SET_INFO functions, and missing MPI_WEIGHTS_EMPTY and MPI_ERR_RMA_SHARED constants. Thanks to Lisandro Dalcin for pointing out the issue.
- Update some help messages in OSHMEM, the usnic BTL, the TCP BTL, and ORTE, and update documentation about ompi_info's --level option.
- Fix some compiler warnings.
- Ensure that ORTE daemons are not bound to a single processor if TaskAffinity is set on by default in Slurm. Thanks to Artem Polyakov for identifying the problem and providing a patch

1.7.5 20 Mar 2014

- *****
* Open MPI is now fully MPI-3.0 compliant

- Add Linux OpenSHMEM support built on top of Open MPI's MPI layer. Thanks to Mellanox for contributing this new feature.
- Allow restricting ORTE daemons to specific cores using the orte_daemon_cores MCA param.
- Ensure to properly set "locality" flags for processes launched via

- MPI dynamic functions such as MPI_COMM_SPAWN.
- Fix MPI_GRAPH_CREATE when nnodes is smaller than the size of the old communicator.
- usnic BTL now supports underlying UDP transport.
- usnic BTL now checks for common connectivity errors at first send to a remote server.
- Minor scalability improvements in the usnic BTL.
- ompi_info now lists whether the Java MPI bindings are available or not.
- MPI-3: mpi.h and the Fortran interfaces now report MPI_VERSION==3 and MPI_SUBVERSION==0.
- MPI-3: Added support for new RMA functions and functionality.
- Fix MPI_Info "const buglet. Thanks to Orion Poplawski for identifying the issue.
- Multiple fixes to mapping/binding options. Thanks to Tetsuya Mishima for his assistance.
- Multiple fixes for normal and abnormal process termination, including singleton MPI_Abort and ensuring to kill entire process groups when abnormally terminating a job.
- Fix DESTDIR install for javadocs. Thanks to Orion Poplawski for pointing out the issue.
- Various performance improvements for the MPI Java bindings.
- OMPI now uses its own internal random number generator and will not perturb srand() and friends.
- Some cleanups for Cygwin builds. Thanks to Marco Atzeri for the patches.
- Add a new collective component (coll/ml) that provides substantially improved performance. It is still experimental, and requires setting coll_ml_priority > 0 to become active.
- Add version check during startup to ensure you are using the same version of Open MPI on all nodes in a job.
- Significantly improved the performance of MPI_DIMS_CREATE for large values. Thanks to Andreas Schäfer for the contribution.
- Removed ASYNCHRONOUS keyword from the "ignore TKR" mpi_f08 module.
- Deprecated the following mpirun options:
 - bynode, --bycore, --byslot: replaced with --map-by node|core|slot.
 - npernode, --npersocket: replaced with --map-by ppr:N:node and --map-by ppr:N:socket, respectively
- Pick NFS "infinitely stale" fix from ROMIO upstream.
- Various PMI2 fixes and extension to support broader range of mappings.
- Improve launch performance at large scale.
- Add support for PBS/Torque environments that set environment variables to indicate the number of slots available on each nodes. Set the ras_tm_smp MCA parameter to "1" to enable this mode.
- Add new, more scalable endpoint exchange (commonly called "modex") method that only exchanges endpoint data on a per-peer basis on first message. Not all transports have been updated to use this feature. Set the rte_orte_direct_modex parameter to "1" to enable this mode.

1.7.4

* CRITICAL CHANGE

*

Appendix C. OpenMPI Release Information

```
* As of release 1.7.4, OpenMPI's default mapping, ranking, and binding
* settings have changed:
*
* Mapping:
*   if #procs <= 2, default to map-by core
*   if #procs > 2, default to map-by socket
* Ranking:
*   if default mapping is used, then default to rank-by slot
*   if map-by <obj> is given, then default to rank-by <obj>,
*     where <obj> is whatever object we mapped against
* Binding:
*   default to bind-to core
*
* Users can override any of these settings individually using the
* corresponding MCA parameter. Note that multi-threaded applications
* in particular may want to override at least the binding default
* to allow threads to use multiple cores.
*****
- Restore version number output in "ompi_info --all".
- Various bug fixes for the mpi_f08 Fortran bindings.
- Fix ROMIO compile error with Lustre 2.4. Thanks to Adam Moody for
  reporting the issue.
- Various fixes for 32 bit platforms.
- Add ability to selectively disable building the mpi or mpi_f08
  module. See the README file for details.
- Fix MX MTL finalization issue.
- Fix ROMIO issue when opening a file with MPI_MODE_EXCL.
- Fix PowerPC and MIPS assembly issues.
- Various fixes to the hcoll and FCA collective offload modules.
- Prevent integer overflow when creating datatypes. Thanks to
  original patch from Gilles Gouaillardet.
- Port some upstream hwloc fixes to Open MPI's embedded copy for
  working around buggy NUMA node cpusets and including missing header
  files. Thanks to Jeff Becker and Paul Hargrove for reporting the
  issues.
- Fix recursive invocation issues in the MXM MTL.
- Various bug fixes to the new MCA parameter back-end system.
- Have the posix fbt module link against -laio on NetBSD platforms.
  Thanks to Paul Hargrove for noticing the issue.
- Various updates and fixes to network filesystem detection to support
  more operating systems.
- Add gfortran v4.9 "ignore TKR" syntax to the mpi Fortran module.
- Various compiler fixes for several BSD-based platforms. Thanks to
  Paul Hargrove for reporting the issues.
- Fix when MPI_COMM_SPAWN[_MULTIPLE] is used on oversubscribed
  systems.
- Change the output from --report bindings to simply state that a
  process is not bound, instead of reporting that it is bound to all
  processors.
- Per MPI-3.0 guidance, remove support for all MPI subroutines with
  choice buffers from the TKR-based mpi Fortran module. Thanks to Jed
  Brown for raising the issue.
- Only allow the usnic BTL to build on 64 bit platforms.
- Various bug fixes to SLURM support, to include ensuring proper
  exiting on abnormal termination.
```

- Ensure that `MPI_COMM_SPAWN[_MULTIPLE]` jobs get the same mapping directives that were used with `mpirun`.
- Fixed the application of `TCP_NODELAY`.
- Change the TCP BTL to not warn if a non-existent interface is ignored.
- Restored the `"--bycore"` `mpirun` option for backwards compatibility.
- Fixed debugger attach functionality. Thanks to Ashley Pittman for reporting the issue and suggesting the fix.
- Fixed faulty `MPI_IBCAST` when invoked on a communicator with only one process.
- Add new Mellanox device IDs to the `openib` BTL.
- Progress towards cleaning up various internal memory leaks as reported by Valgrind.
- Fixed some annoying flex-generated warnings that have been there for years. Thanks to Tom Fogal for the initial patch.
- Support user-provided environment variables via the `"env"` info key to `MPI_COMM_SPAWN[_MULTIPLE]`. Thanks to Tom Fogal for the feature request.
- Fix uninitialized variable in `MPI_DIST_GRAPH_CREATE`.
- Fix a variety of memory errors on SPARC platforms. Thanks to Siegmur Gross for reporting and testing all the issues.
- Remove Solaris threads support. When building on Solaris, `pthread`s will be used.
- Correctly handle the convertor internal stack for persistent receives. Thanks to Guillaume Gouaillardet for identifying the problem.
- Add support for using an external `libevent` via `--with-libevent`. See the README for more details.
- Various OMPIO updates and fixes.
- Add support for the `MPIEXEC_TIMEOUT` environment variable. If set, `mpirun` will terminate the job after this many seconds.
- Update the internal copy of ROMIO to that which shipped in MPICH 3.0.4.
- Various performance tweaks and improvements in the `usnic` BTL, including now reporting `MPI_T` performance variables for each `usnic` device.
- Fix to not access send datatypes for non-root processes with `MPI_ISCATTER[V]` and `MPI_IGATHER[V]`. Thanks to Pierre Jolivet for supplying the initial patch.
- Update `VampirTrace` to 5.14.4.9.
- Fix `ptmalloc2` hook disable when used with `ummunotify`.
- Change the default connection manager for the `openib` BTL to be based on UD verbs data exchanges instead of ORTE OOB data exchanges.
- Fix Fortran compile error when compiling with 8-byte `INTEGERS` and 4-byte ints.
- Fix C++11 issue identified by Jeremiah Willcock.
- Many changes, updates, and bug fixes to the ORTE run-time layer.
- Correctly handle `MPI_REDUCE_SCATTER` with `recvcounts` of 0.
- Update man pages for MPI-3, and add some missing man pages for MPI-2.x functions.
- Updated `mpi_f08` module in accordance with post-MPI-3.0 errata which basically removed `BIND(C)` from all interfaces.
- Fixed `MPI_IN_PLACE` detection for `MPI_SCATTER[V]` in Fortran routines. Thanks to Charles Gerlach for identifying the issue.
- Added support for routable RoCE to the `openib` BTL.

Appendix C. OpenMPI Release Information

- Update embedded hwloc to v1.7.2.
- ErrMgr framework redesigned to better support fault tolerance development activities. See the following RFC for details:
<http://www.open-mpi.org/community/lists/devel/2010/03/7589.php>
- Added database framework to OPAL and changed all modex operations to flow thru it, also included additional system info in the available data
- Added staged state machine to support sequential work flows
- Added distributed file system support for accessing files across nodes that do not have networked file systems
- Extended filem framework to support scalable pre-positioning of files for use by applications, adding new "raw" component that transmits files across the daemon network
- Native Windows support has been removed. A cygwin package is available from that group for Windows-based use.
- Added new MPI Java bindings. See the Javadocs for more details on the API.
- Wrapper compilers now add rpath support by default to generated executables on systems that support it. This behavior can be disabled via `--disable-wrapper-rpath`. See note in README about ABI issues when using rpath in MPI applications.
- Added a new parallel I/O component and multiple new frameworks to support parallel I/O operations.
- Fixed MPI_STATUS_SIZE Fortran issue when used with 8-byte Fortran INTEGERS and 4-byte C ints. Since this issue affects ABI, it is only enabled if Open MPI is configured with `--enable-abi-breaking-fortran-status-i8-fix`. Thanks to Jim Parker for supplying the initial patch.
- Add support for Intel Phi SCIF transport.
- For CUDA-aware MPI configured with CUDA 6.0, use new pointer attribute to avoid extra synchronization in stream 0 when using CUDA IPC between GPUs on the same node.
- For CUDA-aware MPI configured with CUDA 6.0, compile in support of GPU Direct RDMA in openib BTL to improve small message latency.
- Updated ROMIO from MPICH v3.0.4.
- MPI-3: Added support for remaining non-blocking collectives.
- MPI-3: Added support for neighborhood collectives.
- MPI-3: Updated C bindings with consistent use of [].
- MPI-3: Added the const keyword to read-only buffers.
- MPI-3: Added support for non-blocking communicator duplication.
- MPI-3: Added support for non-collective communicator creation.

1.7.3

- Make CUDA-aware support dynamically load libcuda.so so CUDA-aware MPI library can run on systems without CUDA software.
- Fix various issues with dynamic processes and intercommunicator operations under Torque. Thanks to Suraj Prabhakaran for reporting the problem.
- Enable support for the Mellanox MXM2 library by default.
- Improve support for Portals 4.
- Various Solaris fixes. Many thanks to Siegmur Gross for his incredible patience in reporting all the issues.

- MPI-2.2: Add reduction support for MPI_C_*COMPLEX and MPI::*COMPLEX.
- Fixed internal accounting when openpty() fails. Thanks to Michal Peclo for reporting the issue and providing a patch.
- Fixed too-large memory consumption in XRC mode of the openib BTL. Thanks to Alexey Ryzhikh for the patch.
- Add bozo check for negative np values to mpirun to prevent a deadlock. Thanks to Upinder Malhi for identifying the issue.
- Fixed MPI_IS_THREAD_MAIN behavior. Thanks to Lisandro Dalcin for pointing out the problem.
- Various rankfile fixes.
- Fix functionality over iWARP devices.
- Various memory and performance optimizations and tweaks.
- Fix MPI_Cancel issue identified by Fujitsu.
- Add missing support for MPI_Get_address in the "use mpi" TKR implementation. Thanks to Hugo Gagnon for identifying the issue.
- MPI-3: Add support for MPI_Count.
- MPI-2.2: Add missing MPI_IN_PLACE support for MPI_ALLTOALL.
- Added new usnic BTL to support the Cisco usNIC device.
- Minor VampirTrace update to 5.14.4.4.
- Removed support for ancient OS X systems (i.e., prior to 10.5).
- Fixed obscure packing/unpacking datatype bug. Thanks to Takahiro Kawashima for identifying the issue.
- Add run-time support for PMI2 environments.
- Update openib BTL default parameters to include support for Mellanox ConnectX3-Pro devices.
- Update libevent to v2.0.21.
- "ompi_info --param TYPE PLUGIN" now only shows a small number of MCA parameters by default. Add "--level 9" or "--all" to see *all* MCA parameters. See README for more details.
- Add support for asynchronous CUDA-aware copies.
- Add support for Mellanox MPI collective operation offload via the "hcoll" library.
- MPI-3: Add support for the MPI_T interface. Open MPI's MCA parameters are now accessible via the MPI_T control variable interface. Support has been added for a small number of MPI_T performance variables.
- Add Gentoo memory hooks override. Thanks to Justin Bronder for the patch.
- Added new "mindist" process mapper, allowing placement of processes via PCI locality information reported by the BIOS.
- MPI-2.2: Add support for MPI_Dist_graph functionality.
- Enable generic, client-side support for PMI2 implementations. Can be leveraged by any resource manager that implements PMI2; e.g. SLURM, versions 2.6 and higher.

1.7.2

- Major VampirTrace update to 5.14.4.2.
(** also appeared: 1.6.5)
- Fix to set flag==1 when MPI_IPROBE is called with MPI_PROC_NULL.
(** also appeared: 1.6.5)
- Set the Intel Phi device to be ignored by default by the openib BTL.
(** also appeared: 1.6.5)

Appendix C. OpenMPI Release Information

- Decrease the internal memory storage used by intrinsic MPI datatypes for Fortran types. Thanks to Takahiro Kawashima for the initial patch.
(** also appeared: 1.6.5)
- Fix total registered memory calculation for Mellanox ConnectIB and OFED 2.0.
(** also appeared: 1.6.5)
- Fix possible data corruption in the MXM MTL component.
(** also appeared: 1.6.5)
- Remove extraneous -L from hwloc's embedding. Thanks to Stefan Friedel for reporting the issue.
(** also appeared: 1.6.5)
- Fix contiguous datatype memory check. Thanks to Eric Chamberland for reporting the issue.
(** also appeared: 1.6.5)
- Make the openib BTL more friendly to ignoring verbs devices that are not RC-capable.
(** also appeared: 1.6.5)
- Fix some MPI datatype engine issues. Thanks to Thomas Jahns for reporting the issue.
(** also appeared: 1.6.5)
- Add INI information for Chelsio T5 device.
(** also appeared: 1.6.5)
- Integrate MXM STREAM support for MPI_ISEND and MPI_Irecv, and other minor MXM fixes.
(** also appeared: 1.6.5)
- Fix to not show amorphous "MPI was already finalized" error when failing to MPI_File_close an open file. Thanks to Brian Smith for reporting the issue.
(** also appeared: 1.6.5)
- Add a distance-based mapping component to find the socket "closest" to the PCI bus.
- Fix an error that caused epoll to automatically be disabled in libevent.
- Upgrade hwloc to 1.5.2.
- *Really* fixed XRC compile issue in Open Fabrics support.
- Fix MXM connection establishment flow.
- Fixed parallel debugger ability to attach to MPI jobs.
- Fixed some minor memory leaks.
- Fixed datatype corruption issue when combining datatypes of specific formats.
- Added Location Aware Mapping Algorithm (LAMA) mapping component.
- Fixes for MPI_STATUS handling in corner cases.
- Add a distance-based mapping component to find the socket "closest" to the PCI bus.

1.7.1

- Fixed compile error when --without-memory-manager was specified on Linux
- Fixed XRC compile issue in Open Fabrics support.

1.7

- Added MPI-3 functionality:
 - MPI_GET_LIBRARY_VERSION
 - Matched probe
 - MPI_TYPE_CREATE_HINDEXED_BLOCK
 - Non-blocking collectives
 - MPI_INFO_ENV support
 - Fortran '08 bindings (see below)
- Dropped support for checkpoint/restart due to loss of maintainer :-)
- Enabled compile-time warning of deprecated MPI functions by default (in supported compilers).
- Revamped Fortran MPI bindings (see the README for details):
 - "mpifort" is now the preferred wrapper compiler for Fortran
 - Added "use mpi_f08" bindings (for compilers that support it)
 - Added better "use mpi" support (for compilers that support it)
 - Removed incorrect MPI_SCATTERV interface from "mpi" module that was added in the 1.5.x series for ABI reasons.
- Lots of VampirTrace upgrades and fixes; upgrade to v5.14.3.
- Modified process affinity system to provide warning when bindings result in being "bound to all", which is equivalent to not being bound.
- Removed maffinity, paffinity, and carto frameworks (and associated MCA params).
- Upgraded to hwloc v1.5.1.
- Added performance improvements to the OpenIB (OpenFabrics) BTL.
- Made malloc hooks more friendly to IO interposers. Thanks to the bug report and suggested fix from Darshan maintainer Phil Carns.
- Added support for the DMTCP checkpoint/restart system.
- Added support for the Cray uGNI interconnect.
- Fixed header file problems on OpenBSD.
- Fixed issue with MPI_TYPE_CREATE_F90_REAL.
- Wrapper compilers now explicitly list/link all Open MPI libraries if they detect static linking CLI arguments.
- Open MPI now requires a C99 compiler to build. Please upgrade your C compiler if you do not have a C99-compliant compiler.
- Fix MPI_GET_PROCESSOR_NAME Fortran binding to set ierr properly. Thanks to LANL for spotting the error.
- Many MXM and FCA updates.
- Fixed erroneous free of putenv'ed string that showed up in Valgrind reports.
- Fixed MPI_IN_PLACE case for MPI_ALLGATHER.
- Fixed a bug that prevented MCA params from being forwarded to daemons upon launch.
- Fixed issues with VT and CUDA --with-cuda[-libdir] configuration CLI parameters.
- Entirely new implementation of many MPI collective routines focused on better performance.
- Revamped autogen / build system.
- Add new sensor framework to ORTE that includes modules for detecting stalled applications and processes that consume too much memory.
- Added new state machine framework to ORTE that converts ORTE into an event-driven state machine using the event library.
- Added a new MCA parameter (ess_base_stream_buffering) that allows the user to override the system default for buffering of stdout/stderr streams

Appendix C. OpenMPI Release Information

- (via setvbuf). Parameter is not visible via ompi_info.
- Revamped the launch system to allow consideration of node hardware in assigning process locations and bindings.
- Added the `-novm` option to preserve the prior launch behavior.
- Revamped the process mapping system to utilize node hardware by adding new `map-by`, `rank-by`, and `bind-to` cmd line options.
- Added new MCA parameter to provide protection against IO forwarding backlog.
- Dropped support for native Windows due to loss of maintainers. :-)
- Added a new parallel I/O component and multiple new frameworks to support parallel I/O operations.
- Fix typo in `orte_setup_hadoop.m4`. Thanks to Aleksej Saushev for reporting it
- Fix a very old error in `opal_path_access()`. Thanks to Marco Atzeri for chasing it down.

1.6.6: Not released

- Prevent integer overflow in datatype creation. Thanks to Gilles Gouaillardet for identifying the problem and providing a preliminary version of the patch.
- Ensure `help-opal-hwloc-base.txt` is included in distribution tarballs. Thanks to Gilles Gouaillardet for supplying the patch.
- Correctly handle the invalid status for NULL and inactive requests. Thanks to KAWASHIMA Takahiro for submitting the initial patch.
- Fixed `MPI_STATUS_SIZE` Fortran issue when used with 8-byte Fortran INTEGERS and 4-byte C ints. Since this issue affects ABI, it is only enabled if Open MPI is configured with `--enable-abi-breaking-fortran-status-i8-fix`. Thanks to Jim Parker for supplying the initial patch.
- Fix datatype issue for sending from the middle of non-contiguous data.
- Fixed failure error with pty support. Thanks to Michal Pecio for the patch.
- Fixed debugger support for direct-launched jobs.
- Fix `MPI_IS_THREAD_MAIN` to return the correct value. Thanks to Lisandro Dalcin for pointing out the issue.
- Update VT to 5.14.4.4:
 - Fix C++-11 issue.
 - Fix support for building RPMs on Fedora with CUDA libraries.
- Add `openib` part number for ConnectX3-Pro HCA.
- Ensure to check that all resolved IP addresses are local.
- Fix `MPI_COMM_SPAWN` via `rsh` when `mpirun` is on a different server.
- Add Gentoo "sandbox" memory hooks override.

1.6.5

- Updated default SRQ parameters for the `openib` BTL.
(** also to appear: 1.7.2)
- Major `VampirTrace` update to 5.14.4.2.
(** also to appear: 1.7.2)

- Fix to set flag==1 when MPI_IPROBE is called with MPI_PROC_NULL.
(** also to appear: 1.7.2)
- Set the Intel Phi device to be ignored by default by the openib BTL.
(** also to appear: 1.7.2)
- Decrease the internal memory storage used by intrinsic MPI datatypes for Fortran types. Thanks to Takahiro Kawashima for the initial patch.
(** also to appear: 1.7.2)
- Fix total registered memory calculation for Mellanox ConnectIB and OFED 2.0.
(** also to appear: 1.7.2)
- Fix possible data corruption in the MXM MTL component.
(** also to appear: 1.7.2)
- Remove extraneous -L from hwloc's embedding. Thanks to Stefan Friedel for reporting the issue.
(** also to appear: 1.7.2)
- Fix contiguous datatype memory check. Thanks to Eric Chamberland for reporting the issue.
(** also to appear: 1.7.2)
- Make the openib BTL more friendly to ignoring verbs devices that are not RC-capable.
(** also to appear: 1.7.2)
- Fix some MPI datatype engine issues. Thanks to Thomas Jahns for reporting the issue.
(** also to appear: 1.7.2)
- Add INI information for Chelsio T5 device.
(** also to appear: 1.7.2)
- Integrate MXM STREAM support for MPI_ISEND and MPI_Irecv, and other minor MXM fixes.
(** also to appear: 1.7.2)
- Improved alignment for OpenFabrics buffers.
- Fix to not show amorphous "MPI was already finalized" error when failing to MPI_File_close an open file. Thanks to Brian Smith for reporting the issue.
(** also to appear: 1.7.2)

1.6.4

- Fix Cygwin shared memory and debugger plugin support. Thanks to Marco Atzeri for reporting the issue and providing initial patches.
- Fix to obtaining the correct available nodes when a rankfile is providing the allocation. Thanks to Siegmund Gross for reporting the problem.
- Fix process binding issue on Solaris. Thanks to Siegmund Gross for reporting the problem.
- Updates for MXM 2.0.
- Major VT update to 5.14.2.3.
- Fixed F77 constants for Cygwin/Cmake build.
- Fix a linker error when configuring --without-hwloc.
- Automatically provide compiler flags that compile properly on some types of ARM systems.
- Fix slot_list behavior when multiple sockets are specified. Thanks to Siegmund Gross for reporting the problem.

Appendix C. OpenMPI Release Information

- Fixed memory leak in one-sided operations. Thanks to Victor Vysotskiy for letting us know about this one.
- Added performance improvements to the OpenIB (OpenFabrics) BTL.
- Improved error message when process affinity fails.
- Fixed MPI_MINLOC on man pages for MPI_REDUCE(_LOCAL). Thanks to Jed Brown for noticing the problem and supplying a fix.
- Made malloc hooks more friendly to IO interproser. Thanks to the bug report and suggested fix from Darshan maintainer Phil Carns.
- Restored ability to direct launch under SLURM without PMI support.
- Fixed MPI datatype issues on OpenBSD.
- Major VT update to 5.14.2.3.
- Support FCA v3.0+.
- Fixed header file problems on OpenBSD.
- Fixed issue with MPI_TYPE_CREATE_F90_REAL.
- Fix an issue with using external libltdl installations. Thanks to opolawski for identifying the problem.
- Fixed MPI_IN_PLACE case for MPI_ALLGATHER for FCA.
- Allow SLURM PMI support to look in lib64 directories. Thanks to Guillaume Papaure for the patch.
- Restore "use mpi" ABI compatibility with the rest of the 1.5/1.6 series (except for v1.6.3, where it was accidentally broken).
- Fix a very old error in opal_path_access(). Thanks to Marco Atzeri for chasing it down.

1.6.3

- Fix mpirun --launch-agent behavior when a prefix is specified. Thanks to Reuti for identifying the issue.
- Fixed memchecker configury.
- Brought over some compiler warning squashes from the development trunk.
- Fix spawning from a singleton to multiple hosts when the "add-host" MPI_Info key is used. Thanks to Brian Budge for pointing out the problem.
- Add Mellanox ConnexIB IDs and max inline value.
- Fix rankfile when no -np is given.
- FreeBSD detection improvement. Thanks to Brooks Davis for the patch.
- Removed TCP warnings on Windows.
- Improved collective algorithm selection for very large messages.
- Fix PSM MTL affinity settings.
- Fix issue with MPI_OP_COMMUTATIVE in the mpif.h bindings. Thanks to Ake Sandgren for providing a patch to fix the issue.
- Fix issue with MPI_SIZEOF when using CHARACTER and LOGICAL types in the mpi module. Thanks to Ake Sandgren for providing a patch to fix the issue.

1.6.2

- Fix issue with MX MTL. Thanks to Doug Eadline for raising the issue.
- Fix singleton MPI_COMM_SPAWN when the result job spans multiple nodes.
- Fix MXM hang, and update for latest version of MXM.

- Update to support Mellanox FCA 2.5.
- Fix startup hang for large jobs.
- Ensure MPI_TESTANY / MPI_WAITANY properly set the empty status when count==0.
- Fix MPI_CART_SUB behavior of not copying periods to the new communicator properly. Thanks to John Craske for the bug report.
- Add btl_openib_abort_not_enough_reg_mem MCA parameter to cause Open MPI to abort MPI jobs if there is not enough registered memory available on the system (vs. just printing a warning). Thanks to Brock Palen for raising the issue.
- Minor fix to Fortran MPI_INFO_GET: only copy a value back to the user's buffer if the flag is .TRUE.
- Fix VampirTrace compilation issue with the PGI compiler suite.

1.6.1

- A bunch of changes to eliminate hangs on OpenFabrics-based networks. Users with Mellanox hardware are *****STRONGLY ENCOURAGED***** to check their registered memory kernel module settings to ensure that the OS will allow registering more than 8GB of memory. See this FAQ item for details:

<http://www.open-mpi.org/faq/?category=openfabrics#ib-low-reg-mem>

- Fall back to send/receive semantics if registered memory is unavailable for RDMA.
- Fix two fragment leaks when registered memory is exhausted.
- Heuristically determine how much registered memory is available and warn if it's significantly less than all of RAM.
- Artificially limit the amount of registered memory each MPI process can use to about 1/Nth to total registered memory available.
- Improve error messages when events occur that are likely due to unexpected registered memory exhaustion.
- Fix double semicolon error in the C++ in <mpi.h>. Thanks to John Foster for pointing out the issue.
- Allow -Xclang to be specified multiple times in CFLAGS. Thanks to P. Martin for raising the issue.
- Break up a giant "print *" statement in the ABI-preserving incorrect MPI_SCATTER interface in the "large" Fortran "mpi" module. Thanks to Juan Escobar for the initial patch.
- Switch the MPI_ALLTOALLV default algorithm to a pairwise exchange.
- Increase the openib BTL default CQ length to handle more types of OpenFabrics devices.
- Lots of VampirTrace fixes; upgrade to v5.13.0.4.
- Map MPI_2INTEGER to underlying MPI_INTEGERs, not MPI_INTs.
- Ensure that the OMPI version number is tolerant of handling spaces. Thanks to dragonboy for identifying the issue.
- Fixed IN parameter marking on Fortran "mpi" module MPI_COMM_TEST_INTER interface.
- Various MXM improvements.
- Make the output of "mpirun --report-bindings" much more friendly / human-readable.

Appendix C. OpenMPI Release Information

- Properly handle MPI_COMPLEX8|16|32.
- More fixes for mpirun's processor affinity options (--bind-to-core and friends).
- Use aligned memory for OpenFabrics registered memory.
- Multiple fixes for parameter checking in MPI_ALLGATHERV, MPI_REDUCE_SCATTER, MPI_SCATTERV, and MPI_GATHERV. Thanks to the mpi4py community (Bennet Fauber, Lisandro Dalcin, Jonathan Dursi).
- Fixed file positioning overflows in MPI_FILE_GET_POSITION, MPI_FILE_GET_POSITION_SHARED, FILE_GET_SIZE, FILE_GET_VIEW.
- Removed the broken --cpu-set mpirun option.
- Fix cleanup of MPI errorcodes. Thanks to Alexey Bayduraev for the patch.
- Fix default hostfile location. Thanks to Götz Waschke for noticing the issue.
- Improve several error messages.

1.6

- Fix some process affinity issues. When binding a process, Open MPI will now bind to all available hyperthreads in a core (or socket, depending on the binding options specified).
 - > Note that "mpirun --bind-to-socket ..." does not work on POWER6- and POWER7-based systems with some Linux kernel versions. See the FAQ on the Open MPI web site for more information.
- Add support for ARM5 and ARM6 (in addition to the existing ARM7 support). Thanks to Evan Clinton for the patch.
- Minor Mellanox MXM fixes.
- Properly detect FDR10, FDR, and EDR OpenFabrics devices.
- Minor fixes to the mpirun(1) and MPI_Comm_create(3) man pages.
- Prevent segv if COMM_SPAWN_MULTIPLE fails. Thanks to Fujitsu for the patch.
- Disable interposed memory management in fakeroot environments. This fixes a problem in some build environments.
- Minor hwloc updates.
- Array versions of MPI_TEST and MPI_WAIT with a count==0 will now return immediately with MPI_SUCCESS. Thanks to Jeremiah Willcock for the suggestion.
- Update VampirTrace to v5.12.2.
- Properly handle forwarding stdin to all processes when "mpirun --stdin all" is used.
- Workaround XLC assembly bug.
- OS X Tiger (10.4) has not been supported for a while, so forcibly abort configure if we detect it.
- Fix segv in the openib BTL when running on SPARC 64 systems.
- Fix some include file ordering issues on some BSD-based platforms. Thanks to Paul Hargrove for this (and many, many other) fixes.
- Properly handle .FALSE. return parameter value to attribute copy callback functions.
- Fix a bunch of minor C++ API issues; thanks to Fujitsu for the patch.
- Fixed the default hostfile MCA parameter behavior.
- Per the MPI spec, ensure not to touch the port_name parameter to MPI_CLOSE_PORT (it's an IN parameter).

1.5.5

- Many, many portability configure/build fixes courtesy of Paul Hargrove. Thanks, Paul!
- Fixed shared memory fault tolerance support compiler errors.
- Removed not-production-quality rshd and tmd PLM launchers.
- Minor updates to the Open MPI SRPM spec file.
- Fixed mpirun's --bind-to-socket option.
- A few MPI_THREAD_MULTIPLE fixes in the shared memory BTL.
- Upgrade the GNU Autotools used to bootstrap the 1.5/1.6 series to all the latest versions at the time of this release.
- Categorically state in the README that if you're having a problem with Open MPI with the Linux Intel 12.1 compilers, *upgrade your Intel Compiler Suite to the latest patch version*, and the problems will go away. :-)
- Fix the --without-memory-manager configure option.
- Fixes for Totalview/DDT MPI-capable debuggers.
- Update rsh/ssh support to properly handle the Mac OS X library path (i.e., DYLD_LIBRARY_PATH).
- Make warning about shared memory backing files on a networked file system be optional (i.e., can be disabled via MCA parameter).
- Several fixes to processor and memory affinity.
- Various shared memory infrastructure improvements.
- Various checkpoint/restart fixes.
- Fix MPI_IN_PLACE (and other MPI sentinel values) on OS X. Thanks to Dave Goodell for providing the magic OS X gcc linker flags necessary.
- Various man page corrections and typo fixes. Thanks to Fujitsu for the patch.
- Updated wrapper compiler man pages to list the various --showme options that are available.
- Add PMI direct-launch support (e.g., "srun mpi_application" under SLURM).
- Correctly compute the aligned address when packing the datatype description. Thanks to Fujitsu for the patch.
- Fix MPI obscure corner case handling in packing MPI datatypes. Thanks to Fujitsu for providing the patch.
- Workaround an Intel compiler v12.1.0 2011.6.233 vector optimization bug.
- Output the MPI API in ompi_info output.
- Major VT update to 5.12.1.4.
- Upgrade embedded Hardware Locality (hwloc) v1.3.2, plus some post-1.3.2-release bug fixes. All processor and memory binding is now done through hwloc. Woo hoo! Note that this fixes core binding on AMD Opteron 6200 and 4200 series-based systems (sometimes known as Interlagos, Valencia, or other Bulldozer-based chips).
- New MCA parameters to control process-wide memory binding policy: hwloc_base_mem_alloc_policy, hwloc_base_mem_bind_failure_action (see ompi_info --param hwloc base).
- Removed direct support for libnuma. Libnuma support may now be picked up through hwloc.
- Added MPI_IN_PLACE support to MPI_EXSCAN.
- Various fixes for building on Windows, including MinGW support.
- Removed support for the OpenFabrics IBCM connection manager.

Appendix C. OpenMPI Release Information

- Updated Chelsio T4 and Intel NE OpenFabrics default buffer settings.
- Increased the default RDMA CM timeout to 30 seconds.
- Issue a warning if both `btl_tcp_if_include` and `btl_tcp_if_exclude` are specified.
- Many fixes to the Mellanox MXM transport.

1.5.4

- Add support for the (as yet unreleased) Mellanox MXM transport.
- Add support for dynamic service levels (SLs) in the openib BTL.
- Fixed C++ bindings cosmetic/warnings issue with `MPI::Comm::NULL_COPY_FN` and `MPI::Comm::NULL_DELETE_FN`. Thanks to Julio Hoffmann for identifying the issues.
- Also allow the word "slots" in rankfiles (i.e., not just "slot"). (** also to appear in 1.4.4)
- Add Mellanox ConnectX 3 device IDs to the openib BTL defaults. (** also to appear in 1.4.4)
- Various FCA updates.
- Fix 32 bit SIGBUS errors on Solaris SPARC platforms.
- Add missing ARM assembly code files.
- Update to allow more than 128 entries in an appfile. (** also to appear in 1.4.4)
- Various VT updates and bug fixes.
- Update description of `btl_openib_cq_size` to be more accurate. (** also to appear in 1.4.4)
- Various assembly "clobber" fixes.
- Fix a hang in carto selection in obscure situations.
- Guard the inclusion of `execinfo.h` since not all platforms have it. Thanks to Aleksey Saushev for identifying this issue. (** also to appear in 1.4.4)
- Support Solaris legacy `munmap` prototype changes. (** also to appear in 1.4.4)
- Updated to Automake 1.11.1 per <http://www.open-mpi.org/community/lists/devel/2011/07/9492.php>.
- Fix compilation of LSF support.
- Update `MPI_Comm_spawn_multiple.3` man page to reflect what it actually does.
- Fix for possible corruption of the environment. Thanks to Peter Thompson for the suggestion. (** also to appear in 1.4.4)
- Enable use of PSM on direct-launch SLURM jobs.
- Update `paffinity hwloc` to v1.2, and to fix minor bugs affinity assignment bugs on PPC64/Linux platforms.
- Let the openib BTL auto-detect its bandwidth.
- Support new MPI-2.2 datatypes.
- Updates to support more datatypes in MPI one-sided communication.
- Fix recursive locking bug when MPI-IO was used with `MPI_THREAD_MULTIPLE`. (** also to appear in 1.4.4)
- Fix `mpirun` handling of prefix conflicts.
- Ensure `mpirun's --xterm` options leaves sessions attached. (** also to appear in 1.4.4)
- Fixed type of `sendcounts` and `displs` in the "use mpi" F90 module. ABI is preserved, but applications may well be broken. See the README for more details. Thanks to Stanislav Sazykin for

- identifying the issue. (** also to appear in 1.4.4)
- Fix indexed datatype leaks. Thanks to Pascal Deveze for supplying the initial patch. (** also to appear in 1.4.4)
- Fix debugger mapping when mpirun's -npernode option is used.
- Fixed support for configure's --disable-dlopen option when used with "make distclean".
- Fix segv associated with MPI_Comm_create with MPI_GROUP_EMPTY. Thanks to Dominik Goedeke for finding this. (** also to appear in 1.4.4)
- Improved LoadLeveler ORTE support.
- Add new WinVerbs BTL plugin, supporting native OpenFabrics verbs on Windows (the "wv" BTL).
- Add new btl_openib_gid_index MCA parameter to allow selecting which GID to use on an OpenFabrics device's GID table.
- Add support for PCI relaxed ordering in the OpenFabrics BTL (when available).
- Update rsh logic to allow correct SGE operation.
- Ensure that the mca_paffinity_alone MCA parameter only appears once in the ompi_info output. Thanks to Gus Correa for identifying the issue.
- Fixed return codes from MPI_PROBE and MPI_IPROBE. (** also to appear in 1.4.4)
- Remove --enable-progress-thread configure option; it doesn't work on the v1.5 branch. Rename --enable-mpi-threads to --enable-mpi-thread-multiple. Add new --enable-opal-multi-threads option.
- Updates for Intel Fortran compiler version 12.
- Remove bproc support. Farewell bproc!
- If something goes wrong during MPI_INIT, fix the error message to say that it's illegal to invoke MPI_INIT before MPI_INIT.

1.5.3

- Add missing "affinity" MPI extension (i.e., the OMPI_Affinity_str() API) that was accidentally left out of the 1.5.2 release.

1.5.2

- Replaced all custom topology / affinity code with initial support for hwloc v1.1.1 (PLPA has been removed -- long live hwloc!). Note that hwloc is bundled with Open MPI, but an external hwloc can be used, if desired. See README for more details.
- Many CMake updates for Windows builds.
- Updated opal_cr_thread_sleep_wait MCA param default value to make it less aggressive.
- Updated debugger support to allow Totalview attaching from jobs launched directly via srun (not mpirun). Thanks to Nikolay Piskun for the patch.
- Added more FTB/CIFTS support.
- Fixed compile error with the PGI compiler.

Appendix C. OpenMPI Release Information

- Portability fixes to allow the openib BTL to run on the Solaris verbs stack.
- Fixed multi-token command-line issues when using the mpirun --debug switch. For example:
 mpirun --debug -np 2 a.out "foo bar"
Thanks to Gabriele Fatigati for reporting the issue.
- Added ARM support.
- Added the MPI_ROOT environment variable in the Open MPI Linux SRPM for customers who use the BPS and LSF batch managers.
- Updated ROMIO from MPICH v1.3.1 (plus one additional patch).
- Fixed some deprecated MPI API function notification messages.
- Added new "bfo" PML that provides failover on OpenFabrics networks.
- Fixed some buffer memcheck issues in MPI*_init.
- Added Solaris-specific chip detection and performance improvements.
- Fix some compile errors on Solaris.
- Updated the "rmcast" framework with bug fixes, new functionality.
- Updated the Voltaire FCA component with bug fixes, new functionality. Support for FCA version 2.1.
- Fix gcc 4.4.x and 4.5.x over-aggressive warning notifications on possibly freeing stack variables. Thanks to the Gentoo packagers for reporting the issue.
- Make the openib component be verbose when it disqualifies itself due to MPI_THREAD_MULTIPLE.
- Minor man page fixes.
- Various checkpoint / restart fixes.
- Fix race condition in the one-sided unlock code. Thanks to Guillaume Thouvenin for finding the issue.
- Improve help message aggregation.
- Add OMPI_Affinity_str() optional user-level API function (i.e., the "affinity" MPI extension). See README for more details.
- Added btl_tcp_if_seq MCA parameter to select a different ethernet interface for each MPI process on a node. This parameter is only useful when used with virtual ethernet interfaces on a single network card (e.g., when using virtual interfaces give dedicated hardware resources on the NIC to each process).
- Changed behavior of mpirun to terminate if it receives 10 (or more) SIGPIPEs.
- Fixed oversubscription detection.
- Added new mtl_mx_board and mtl_mx_endpoint MCA parameters.
- Added ummunotify support for OpenFabrics-based transports. See the README for more details.

1.5.1

- Fixes for the Oracle Studio 12.2 Fortran compiler.
- Fix SPARC and SPARCv9 atomics. Thanks to Nicola Stange for the initial patch.
- Fix Libtool issues with the IBM XL compiler in 64-bit mode.
- Restore the reset of the libevent progress counter to avoid over-sampling the event library.
- Update memory barrier support.
- Use memmove (instead of memcpy) when necessary (e.g., source and destination overlap).

- Fixed ompi-top crash.
- Fix to handle Autoconf --program-transforms properly and other m4/configury updates. Thanks to the GASNet project for the --program transforms fix.
- Allow hostfiles to specify usernames on a per-host basis.
- Update wrapper compiler scripts to search for perl during configure, per request from the BSD maintainers.
- Minor man page fixes.
- Added --with-libltdl option to allow building Open MPI with an external installation of libltdl.
- Fixed various issues with -D_FORTIFY_SOURCE=2.
- Various VT fixes and updates.

1.5

- Added "knem" support: direct process-to-process copying for shared memory message passing. See <http://runtime.bordeaux.inria.fr/knem/> and the README file for more details.
- Updated shared library versioning scheme and linking style of MPI applications. The MPI application ABI has been broken from the v1.3/v1.4 series. MPI applications compiled against any prior version of Open MPI will need to, at a minimum, re-link. See the README file for more details.
- Added "fca" collective component, enabling MPI collective offload support for Voltaire switches.
- Fixed MPI one-sided operations with large target displacements. Thanks to Brian Price and Jed Brown for reporting the issue.
- Fixed MPI_GET_COUNT when used with large counts. Thanks to Jed Brown for reporting the issue.
- Made the openib BTL safer if extremely low SRQ settings are used.
- Fixed handling of the array_of_argv parameter in the Fortran binding of MPI_COMM_SPAWN_MULTIPLE (** also to appear: 1.4.3).
- Fixed malloc(0) warnings in some collectives.
- Fixed a problem with the Fortran binding for MPI_FILE_CREATE_ERRHANDLER. Thanks to Secretan Yves for identifying the issue (** also to appear: 1.4.3).
- Updates to the LSF PLM to ensure that the path is correctly passed. Thanks to Teng Lin for the patch (** also to appear: 1.4.3).
- Fixes for the F90 MPI_COMM_SET_ERRHANDLER and MPI_WIN_SET_ERRHANDLER bindings. Thanks to Paul Kapinos for pointing out the issue (** also to appear: 1.4.3).
- Fixed extra_state parameter types in F90 prototypes for MPI_COMM_CREATE_KEYVAL, MPI_GREQUEST_START, MPI_REGISTER_DATAREP, MPI_TYPE_CREATE_KEYVAL, and MPI_WIN_CREATE_KEYVAL.
- Fixes for Solaris oversubscription detection.
- If the PML determines it can't reach a peer process, print a slightly more helpful message. Thanks to Nick Edmonds for the suggestion.
- Make btl_openib_if_include/exclude function the same way btl_tcp_if_include/exclude works (i.e., supplying an _include list overrides supplying an _exclude list).
- Apply more scalable reachability algorithm on platforms with more than 8 TCP interfaces.

Appendix C. OpenMPI Release Information

- Various assembly code updates for more modern platforms / compilers.
- Relax restrictions on using certain kinds of MPI datatypes with one-sided operations. Users beware; not all MPI datatypes are valid for use with one-sided operations!
- Improve behavior of MPI_COMM_SPAWN with regards to --bynode.
- Various threading fixes in the openib BTL and other core pieces of Open MPI.
- Various help file and man pages updates.
- Various FreeBSD and NetBSD updates and fixes. Thanks to Kevin Buckley and Aleksey Saushev for their work.
- Fix case where freeing communicators in MPI_FINALIZE could cause process failures.
- Print warnings if shared memory state files are opened on what look like networked filesystems.
- Update libevent to v1.4.13.
- Allow propagating signals to processes that call fork().
- Fix bug where MPI_GATHER was sometimes incorrectly examining the datatype on non-root processes. Thanks to Michael Hofmann for investigating the issue.
- Various Microsoft Windows fixes.
- Various Catamount fixes.
- Various checkpoint / restart fixes.
- Xgrid support has been removed until it can be fixed (patches would be welcome).
- Added simplistic "libompitrace" contrib package. Using the MPI profiling interface, it essentially prints out to stderr when select MPI functions are invoked.
- Update bundled VampirTrace to v5.8.2.
- Add pkg-config(1) configuration files for ompi, ompi-c, ompi-cxx, ompi-f77, ompi-f90. See the README for more details.
- Removed the libopenmpi_malloc library (added in the v1.3 series) since it is no longer necessary
- Add several notifier plugins (generally used when Open MPI detects system/network administrator-worthy problems); each have their own MCA parameters to govern their usage. See "ompi_info --param notifier <name>" for more details.
 - command to execute arbitrary commands (e.g., run a script).
 - file to send output to a file.
 - ftb to send output to the Fault Tolerant Backplane (see <http://wiki.mcs.anl.gov/cifts/index.php/CIFTS>)
 - hnp to send the output to mpirun.
 - smtp (requires libesmtplib) to send an email.

1.4.5

- Fixed the --disable-memory-manager configure switch.
(** also to appear in 1.5.5)
- Fix typos in code and man pages. Thanks to Fujitsu for these fixes.
(** also to appear in 1.5.5)
- Improve management of the registration cache; when full, try freeing old entries and attempt to re-register.
- Fixed a data packing pointer alignment issue. Thanks to Fujitsu for the patch.

- (** also to appear in 1.5.5)
- Add ability to turn off warning about having the shared memory backing store over a networked filesystem. Thanks to Chris Samuel for this suggestion.
- (** also to appear in 1.5.5)
- Removed an unnecessary memmove() and plugged a couple of small memory leaks in the openib OOB connection setup code.
- Fixed some QLogic bugs. Thanks to Mark Debbage from QLogic for the patches.
- Fixed problem with MPI_IN_PLACE and other sentinel Fortran constants on OS X.
- (** also to appear in 1.5.5)
- Fix SLURM cpus-per-task allocation.
- (** also to appear in 1.5.5)
- Fix the datatype engine for when data left over from the previous pack was larger than the allowed space in the pack buffer. Thanks to Yuki Matsumoto and Takahiro Kawashima for the bug report and the patch.
- Fix Fortran value for MPI_MAX_PORT_NAME. Thanks to Enzo Dari for raising the issue.
- Workaround an Intel compiler v12.1.0 2011.6.233 vector optimization bug.
- Fix issues on Solaris with the openib BTL.
- Fixes for the Oracle Studio 12.2 Fortran compiler.
- Update iWARP parameters for the Intel NICs.
- (** also to appear in 1.5.5)
- Fix obscure cases where MPI_ALLGATHER could crash. Thanks to Andrew Senin for reporting the problem.
- (** also to appear in 1.5.5)

1.4.4

- Modified a memcpy() call in the openib btl connection setup to use memmove() instead because of the possibility of an overlapping copy (as identified by valgrind).
- Changed use of sys_timer_get_cycles() to the more appropriate wrapper: opal_timer_base_get_cycles(). Thanks to Jani Monoses for this fix.
- Corrected the reported default value of btl_openib_ib_timeout in the "IB retries exceeded" error message. Thanks to Kevin Buckley for this correction.
- Increased rdmacm address resolution timeout from 1s to 30s & updated Chelsio T4 openib BTL defaults. Thanks to Steve Wise for these updates.
- (** also to appear in 1.5.5)
- Ensure that MPI_Accumulate error return in 1.4 is consistent with 1.5.x and trunk.
- Allow the word "slots" in rankfiles (i.e., not just "slot").
- (** also appeared in 1.5.4)
- Add Mellanox ConnectX 3 device IDs to the openib BTL defaults.
- (** also appeared in 1.5.4)
- Update description of btl_openib_cq_size to be more accurate.
- Ensure mpirun's --xterm options leaves sessions attached.
- (** also appeared in 1.5.4)

Appendix C. OpenMPI Release Information

- Update to allow more than 128 entries in an appfile.
(** also appeared in 1.5.4)
- Update description of btl_openib_cq_size to be more accurate.
(** also appeared in 1.5.4)
- Fix for deadlock when handling recursive attribute keyval deletions
(e.g., when using ROMIO with MPI_THREAD_MULTIPLE).
- Fix indexed datatype leaks. Thanks to Pascal Deveze for supplying
the initial patch. (** also appeared in 1.5.4)
- Fixed the F90 types of the sendcounts and displs parameters to
MPI_SCATTERV. Thanks to Stanislav Sazykin for identifying the issue.
(** also appeared in 1.5.4)
- Exclude opal/libltdl from "make distclean" when --disable-dlopen is
used. Thanks to David Gunter for reporting the issue.
- Fixed a segv in MPI_Comm_create when called with GROUP_EMPTY.
Thanks to Dominik Goedeke for finding this.
(** also appeared in 1.5.4)
- Fixed return codes from MPI_PROBE and MPI_IPROBE.
(** also appeared in 1.5.4)
- Fixed undefined symbol error when using the vtf90 profiling tool.
- Fix for referencing an uninitialized variable in DPM ORTE. Thanks
to Avinash Malik for reporting the issue.
- Fix for correctly handling multi-token args when using debuggers.
- Eliminated the unneeded u_int*_t datatype definitions.
- Change in ORTE DPM to get around gcc 4.[45].x compiler warnings
about possibly calling free() on a non-heap variable, even though it
will never happen because the refcount will never go to zero.
- Fixed incorrect text in MPI_File_set_view man page.
- Fix in MPI_Init_thread for checkpoint/restart.
- Fix for libtool issue when using pgcc to compile ompi in conjunction
with the -tp option.
- Fixed a race condition in osc_rdma_sync. Thanks to Guillaume
Thouvenin for finding this issue.
- Clarification of MPI_Init_thread man page.
- Fixed an indexing problem in precondition_transports.
- Fixed a problem in which duplicated libs were being specified for
linking. Thanks to Hicham Mouline for noticing it.
- Various autogen.sh fixes.
- Fix for memchecking buffers during MPI_*INIT.
- Man page cleanups. Thanks to Jeremiah Willcock and Jed Brown.
- Fix for VT rpmbuild on RHEL5.
- Support Solaris legacy munmap prototype changes.
(** also appeared in 1.5.4)
- Expands app_idx to int32_t to allow more than 127 app_contexts.
- Guard the inclusion of execinfo.h since not all platforms have it. Thanks
to Aleksey Saushev for identifying this issue.
(** also appeared in 1.5.4)
- Fix to avoid possible environment corruption. Thanks to Peter Thompson
for identifying the issue and supplying a patch.
(** also appeared in 1.5.4)
- Fixed paffinity base MCA duplicate registrations. Thanks to Gus
Correa for bringing this to our attention.
- Fix recursive locking bug when MPI-IO was used with
MPI_THREAD_MULTIPLE. (** also appeared in 1.5.4)
- F90 MPI API fixes.
- Fixed a misleading MPI_Bcast error message. Thanks to Jeremiah

- Willcock for reporting this.
- Added `<sys/stat.h>` to `ptmalloc's hooks.c` (it's not always included by default on some systems).
 - Libtool patch to get around a build problem when using the IBM XL compilers.
 - Fix to detect and avoid overlapping `memcpy()`. Thanks to Francis Pellegrini for identifying the issue.
 - Fix to allow `ompi` to work on top of RoCE vLANs.
 - Restored a missing debugger flag to support TotalView. Thanks to David Turner and the TV folks for supplying the fix.
 - Updated SLURM support to 1.5.1.
 - Removed an extraneous `#include` from the TCP BTL.
 - When specifying OOB ports, fix to convert the ports into network byte order before binding.
 - Fixed use of memory barriers in the SM BTL. This fixed `segv's` when compiling with Intel 10.0.025 or PGI 9.0-3.
 - Fix to prevent the SM BTL from creating its `mmap'd` file in directories that are remotely mounted.

1.4.3

- Fixed handling of the `array_of_argv` parameter in the Fortran binding of `MPI_COMM_SPAWN_MULTIPLE` (** also to appear: 1.5).
- Fixed a problem with the Fortran binding for `MPI_FILE_CREATE_ERRHANDLER`. Thanks to Secretan Yves for identifying the issue (** also to appear: 1.5).
- Updates to the LSF PLM to ensure that the path is correctly passed. Thanks to Teng Lin for the patch (** also to appear: 1.5).
- Fixes for the F90 `MPI_COMM_SET_ERRHANDLER` and `MPI_WIN_SET_ERRHANDLER` bindings. Thanks to Paul Kapinos for pointing out the issue. (** also to appear: 1.5).
- Fixed various `MPI_THREAD_MULTIPLE` race conditions.
- Fixed an issue with an undeclared variable from `ptmalloc2 munmap` on BSD systems.
- Fixes for BSD interface detection.
- Various other BSD fixes. Thanks to Kevin Buckley helping to track all of this down.
- Fixed issues with the use of the `-nper*` `mpirun` command line arguments.
- Fixed an issue with `coll` tuned dynamic rules.
- Fixed an issue with the use of `OPAL_DESTDIR` being applied too aggressively.
- Fixed an issue with one-sided xfers when the displacement exceeds 2GBytes.
- Change to ensure TotalView works properly on Darwin.
- Added support for Visual Studio 2010.
- Fix to ensure proper placement of `VampirTrace` header files.
- Needed to add `volatile` keyword to a variable used in debugging (`MPIR_being_debugged`).
- Fixed a bug in `inter-allgather`.
- Fixed `malloc(0)` warnings.
- Corrected a typo the `MPI_Comm_size` man page (`intra -> inter`). Thanks to Simon number.cruncher for pointing this out.
- Fixed a `SegV` in `orted` when given more than 127 `app_contexts`.
- Removed `xgrid` source code from the 1.4 branch since it is no longer supported in the 1.4 series.

Appendix C. OpenMPI Release Information

- Removed the `--enable-opal-progress-threads` config option since opal progress thread support does not work in 1.4.x.
- Fixed a defect in VampirTrace's `vtfilter`.
- Fixed wrong Windows path in `hnp_contact`.
- Removed the requirement for a paffinity component.
- Removed a hardcoded limit of 64 interconnected jobs.
- Fix to allow singletons to use `ompi-server` for rendezvous.
- Fixed bug in `output-filename` option.
- Fix to correctly handle failures in `mx_init()`.
- Fixed a potential Fortran memory leak.
- Fixed an incorrect branch in some ppc32 assembly code. Thanks to Matthew Clark for this fix.
- Remove use of undocumented `AS_VAR_GET` macro during configuration.
- Fixed an issue with VampirTrace's wrapper for `MPI_init_thread`.
- Updated `mca-btl-openib-device-params.ini` file with various new vendor id's.
- Configuration fixes to ensure `CPPFLAGS` in handled properly if a non-standard `valgrind` location was specified.
- Various man page updates

1.4.2

- Fixed problem when running in heterogeneous environments. Thanks to Timur Magomedov for helping to track down this issue.
- Update LSF support to ensure that the path is passed correctly. Thanks to Teng Lin for submitting a patch.
- Fixed some miscellaneous oversubscription detection bugs.
- IBM re-licensed its LoadLeveler code to be BSD-compliant.
- Various OpenBSD and NetBSD build and run-time fixes. Many thanks to the OpenBSD community for their time, expertise, and patience getting these fixes incorporated into Open MPI's main line.
- Various fixes for multithreading deadlocks, race conditions, and other nefarious things.
- Fixed ROMIO's handling of "nearly" contiguous issues (e.g., with non-zero `true_lb`). Thanks for Pascal Deveze for the patch.
- Bunches of Windows build fixes. Many thanks to several Windows users for their help in improving our support on Windows.
- Now allow the graceful failover from MTLs to BTLs if no MTLs can initialize successfully.
- Added "clobber" information to various atomic operations, fixing erroneous behavior in some newer versions of the GNU compiler suite.
- Update various iWARP and InfiniBand device specifications in the `OpenFabrics .ini` support file.
- Fix the use of hostfiles when a username is supplied.
- Various fixes for rankfile support.
- Updated the internal version of VampirTrace to 5.4.12.
- Fixed OS X TCP wireup issues having to do with IPv4/IPv6 confusion (see <https://svn.open-mpi.org/trac/ompi/changeset/22788> for more details).
- Fixed some problems in processor affinity support, including when there are "holes" in the processor namespace (e.g., offline processors).
- Ensure that Open MPI's "session directory" (usually located in `/tmp`) is cleaned up after process termination.

- Fixed some problems with the collective "hierarch" implementation that could occur in some obscure conditions.
- Various MPI_REQUEST_NULL, API parameter checking, and attribute error handling fixes. Thanks to Lisandro Dalcin for reporting the issues.
- Fix case where MPI_GATHER erroneously used datatypes on non-root nodes. Thanks to Michael Hofmann for investigating the issue.
- Patched ROMIO support for PVFS2 > v2.7 (patch taken from MPICH2 version of ROMIO).
- Fixed "mpirun --report-bindings" behavior when used with mpi_paffinity_alone=1. Also fixed mpi_paffinity_alone=1 behavior with non-MPI applications. Thanks to Brice Goglin for noticing the problem.
- Ensure that all OpenFabrics devices have compatible receive_queues specifications before allowing them to communicate. See the lengthy comment in <https://svn.open-mpi.org/trac/ompi/changeset/22592> for more details.
- Fix some issues with checkpoint/restart.
- Improve the pre-MPI_INIT/post-MPI_FINALIZE error messages.
- Ensure that loopback addresses are never advertised to peer processes for RDMA/OpenFabrics support.
- Fixed a CSUM PML false positive.
- Various fixes for Catamount support.
- Minor update to wrapper compilers in how user-specific argv is ordered on the final command line. Thanks to Jed Brown for the suggestions.
- Removed flex.exe binary from Open MPI tarballs; now generate flex code from a newer (Windows-friendly) flex when we make official tarballs.

1.4.1

- Update to PLPA v1.3.2, addressing a licensing issue identified by the Fedora project. See <https://svn.open-mpi.org/trac/plpa/changeset/262> for details.
- Add check for malformed checkpoint metadata files (Ticket #2141).
- Fix error path in ompi-checkpoint when not able to checkpoint (Ticket #2138).
- Cleanup component release logic when selecting checkpoint/restart enabled components (Ticket #2135).
- Fixed VT node name detection for Cray XT platforms, and fixed some broken VT documentation files.
- Fix a possible race condition in tearing down RDMA CM-based connections.
- Relax error checking on MPI_GRAPH_CREATE. Thanks to David Singleton for pointing out the issue.
- Fix a shared memory "hang" problem that occurred on x86/x86_64 platforms when used with the GNU >=4.4.x compiler series.
- Add fix for Libtool 2.2.6b's problems with the PGI 10.x compiler suite. Inspired directly from the upstream Libtool patches that fix the issue (but we need something working before the next Libtool release).

Appendix C. OpenMPI Release Information

1.4

The *only* change in the Open MPI v1.4 release (as compared to v1.3.4) was to update the embedded version of Libtool's libltdl to address a potential security vulnerability. Specifically: Open MPI v1.3.4 was created with GNU Libtool 2.2.6a; Open MPI v1.4 was created with GNU Libtool 2.2.6b. There are no other changes between Open MPI v1.3.4 and v1.4.

1.3.4

- Fix some issues in OMPI's SRPM with regard to `shell_scripts_basename` and its use with `mpi-selector`. Thanks to Bill Johnstone for pointing out the problem.
- Added many new MPI job process affinity options to `mpirun`. See the newly-updated `mpirun(1)` man page for details.
- Several updates to `mpirun`'s XML output.
- Update to fix a few Valgrind warnings with regards to the `ptmalloc2` allocator and Open MPI's use of PLPA.
- Many updates and fixes to the (non-default) "sm" collective component (i.e., native shared memory MPI collective operations).
- Updates and fixes to some `MPI_COMM_SPAWN_MULTIPLE` corner cases.
- Fix some internal copying functions in Open MPI's use of PLPA.
- Correct some SLURM nodelist parsing logic that may have interfered with large jobs. Additionally, per advice from the SLURM team, change the environment variable that we use for obtaining the job's allocation.
- Revert to an older, safer (but slower) communicator ID allocation algorithm.
- Fixed minimum distance finding for OpenFabrics devices in the `openib` BTL.
- Relax the parameter checking `MPI_CART_CREATE` a bit.
- Fix `MPI_COMM_SPAWN[_MULTIPLE]` to only error-check the info arguments on the root process. Thanks to Federico Golfre Andreasi for reporting the problem.
- Fixed some BLCR configure issues.
- Fixed a potential deadlock when the `openib` BTL was used with `MPI_THREAD_MULTIPLE`.
- Fixed dynamic rules selection for the "tuned" coll component.
- Added a launch progress meter to `mpirun` (useful for large jobs; set the `orte_report_launch_progress` MCA parameter to 1 to see it).
- Reduced the number of file descriptors consumed by each MPI process.
- Add new device IDs for Chelsio T3 RNICs to the `openib` BTL config file.
- Fix some CRS self component issues.
- Added some MCA parameters to the PSM MTL to tune its run-time behavior.
- Fix some VT issues with `MPI_BOTTOM/MPI_IN_PLACE`.
- Man page updates from the Debain Open MPI package maintainers.
- Add cycle counter support for the Alpha and Sparc platforms.
- Pass visibility flags to `libltdl`'s configure script, resulting in those symbols being hidden. This appears to mainly solve the

problem of applications attempting to use different versions of libltdl from that used to build Open MPI.

1.3.3

- Fix a number of issues with the openib BTL (OpenFabrics) RDMA CM, including a memory corruption bug, a shutdown deadlock, and a route timeout. Thanks to David McMillen and Hal Rosenstock for help in tracking down the issues.
- Change the behavior of the EXTRA_STATE parameter that is passed to Fortran attribute callback functions: this value is now stored internally in MPI -- it no longer references the original value passed by MPI*_CREATE_KEYVAL.
- Allow the overriding RFC1918 and RFC3330 for the specification of "private" networks, thereby influencing Open MPI's TCP "reachability" computations.
- Improve flow control issues in the sm btl, by both tweaking the shared memory progression rules and by enabling the "sync" collective to barrier every 1,000th collective.
- Various fixes for the IBM XL C/C++ v10.1 compiler.
- Allow explicit disabling of ptmalloc2 hooks at runtime (e.g., enable support for Debian's builtroot system). Thanks to Manuel Prinz and the rest of the Debian crew for helping identify and fix this issue.
- Various minor fixes for the I/O forwarding subsystem.
- Big endian iWARP fixes in the Open Fabrics RDMA CM support.
- Update support for various OpenFabrics devices in the openib BTL's .ini file.
- Fixed undefined symbol issue with Open MPI's parallel debugger message queue support so it can be compiled by Sun Studio compilers.
- Update MPI_SUBVERSION to 1 in the Fortran bindings.
- Fix MPI_GRAPH_CREATE Fortran 90 binding.
- Fix MPI_GROUP_COMPARE behavior with regards to MPI_IDENT. Thanks to Geoffrey Irving for identifying the problem and supplying the fix.
- Silence gcc 4.1 compiler warnings about type punning. Thanks to Number Cruncher for the fix.
- Added more Valgrind and other memory-cleanup fixes. Thanks to various Open MPI users for help with these issues.
- Miscellaneous VampirTrace fixes.
- More fixes for openib credits in heavy-congestion scenarios.
- Slightly decrease the latency in the openib BTL in some conditions (add "send immediate" support to the openib BTL).
- Ensure to allow MPI_REQUEST_GET_STATUS to accept an MPI_STATUS_IGNORE parameter. Thanks to Shaun Jackman for the bug report.
- Added Microsoft Windows support. See README.WINDOWS file for details.

1.3.2

- Fixed a potential infinite loop in the openib BTL that could occur in senders in some frequent-communication scenarios. Thanks to Don

Appendix C. OpenMPI Release Information

- Wood for reporting the problem.
- Add a new checksum PML variation on ob1 (main MPI point-to-point communication engine) to detect memory corruption in node-to-node messages
- Add a new configuration option to add padding to the openib header so the data is aligned
- Add a new configuration option to use an alternative checksum algo when using the checksum PML
- Fixed a problem reported by multiple users on the mailing list that the LSF support would fail to find the appropriate libraries at run-time.
- Allow empty shell designations from `getpwuid()`. Thanks to Sergey Koposov for the bug report.
- Ensure that `mpirun` exits with non-zero status when applications die due to user signal. Thanks to Geoffroy Pignot for suggesting the fix.
- Ensure that `MPI_VERSION / MPI_SUBVERSION` match what is returned by `MPI_GET_VERSION`. Thanks to Rob Egan for reporting the error.
- Updated `MPI_*KEYVAL_CREATE` functions to properly handle Fortran extra state.
- A variety of ob1 (main MPI point-to-point communication engine) bug fixes that could have caused hangs or seg faults.
- Do not install Open MPI's signal handlers in `MPI_INIT` if there are already signal handlers installed. Thanks to Kees Verstoep for bringing the issue to our attention.
- Fix GM support to not seg fault in `MPI_INIT`.
- Various `VampirTrace` fixes.
- Various `PLPA` fixes.
- No longer create BTLs for invalid (TCP) devices.
- Various man page style and lint cleanups.
- Fix critical OpenFabrics-related bug noted here:
<http://www.open-mpi.org/community/lists/announce/2009/03/0029.php>.
Open MPI now uses a much more robust memory intercept scheme that is quite similar to what is used by MX. The use of `"-lopenmpi-malloc"` is no longer necessary, is deprecated, and is expected to disappear in a future release. `-lopenmpi-malloc` will continue to work for the duration of the Open MPI v1.3 and v1.4 series.
- Fix some OpenFabrics shutdown errors, both regarding `iWARP` and `SRQ`.
- Allow the `udapl` BTL to work on Solaris platforms that support relaxed PCI ordering.
- Fix problem where the `mpirun` would sometimes use `rsh/ssh` to launch on the localhost (instead of simply forking).
- Minor `SLURM` `stdin` fixes.
- Fix to run properly under `SGE` jobs.
- Scalability and latency improvements for shared memory jobs: convert to using one message queue instead of N queues.
- Automatically size the shared-memory area (`mmap` file) to match better what is needed; specifically, so that large-`np` jobs will start.
- Use fixed-length MPI predefined handles in order to provide ABI compatibility between Open MPI releases.
- Fix building of the `posix` `paffinity` component to properly get the number of processors in loosely tested environments (e.g., `FreeBSD`). Thanks to Steve Kargl for reporting the issue.
- Fix `--with-libnuma` handling in `configure`. Thanks to Gus Correa for reporting the problem.

1.3.1

- Added "sync" coll component to allow users to synchronize every N collective operations on a given communicator.
- Increased the default values of the IB and RNR timeout MCA parameters.
- Fix a compiler error noted by Mostyn Lewis with the PGI 8.0 compiler.
- Fix an error that prevented stdin from being forwarded if the rsh launcher was in use. Thanks to Branden Moore for pointing out the problem.
- Correct a case where the added datatype is considered as contiguous but has gaps in the beginning.
- Fix an error that limited the number of comm_spawns that could simultaneously be running in some environments
- Correct a corner case in OB1's GET protocol for long messages; the error could sometimes cause MPI jobs using the openib BTL to hang.
- Fix a bunch of bugs in the IO forwarding (IOF) subsystem and add some new options to output to files and redirect output to xterm. Thanks to Jody Weissmann for helping test out many of the new fixes and features.
- Fix SLURM race condition.
- Fix MPI_File_c2f(MPI_FILE_NULL) to return 0, not -1. Thanks to Lisandro Dalcin for the bug report.
- Fix the DSO build of tm PLM.
- Various fixes for size disparity between C int's and Fortran INTEGER's. Thanks to Christoph van Wullen for the bug report.
- Ensure that mpirun exits with a non-zero exit status when daemons or processes abort or fail to launch.
- Various fixes to work around Intel (NetEffect) RNIC behavior.
- Various fixes for mpirun's --preload-files and --preload-binary options.
- Fix the string name in MPI::ERRORS_THROW_EXCEPTIONS.
- Add ability to forward SIFTSTP and SIGCONT to MPI processes if you set the MCA parameter orte_forward_job_control to 1.
- Allow the sm BTL to allocate larger amounts of shared memory if desired (helpful for very large multi-core boxes).
- Fix a few places where we used PATH_MAX instead of OPAL_PATH_MAX, leading to compile problems on some platforms. Thanks to Andrea Iob for the bug report.
- Fix mca_btl_openib_warn_no_device_params_found MCA parameter; it was accidentally being ignored.
- Fix some run-time issues with the sctp BTL.
- Ensure that RTLD_NEXT exists before trying to use it (e.g., it doesn't exist on Cygwin). Thanks to Gustavo Seabra for reporting the issue.
- Various fixes to VampirTrace, including fixing compile errors on some platforms.
- Fixed missing MPI_Comm_accept.3 man page; fixed minor issue in orterun.1 man page. Thanks to Dirk Eddelbuettel for identifying the problem and submitting a patch.
- Implement the XML formatted output of stdout/stderr/stddiag.
- Fixed mpirun's -wdir switch to ensure that working directories for multiple app contexts are properly handled. Thanks to Geoffroy

Appendix C. OpenMPI Release Information

Pignot for reporting the problem.

- Improvements to the MPI C++ integer constants:
 - Allow MPI::SEEK_* constants to be used as constants
 - Allow other MPI C++ constants to be used as array sizes
- Fix minor problem with orte-restart's command line options. See ticket #1761 for details. Thanks to Gregor Dschung for reporting the problem.

1.3

- Extended the OS X 10.5.x (Leopard) workaround for a problem when assembly code is compiled with -g[0-9]. Thanks to Barry Smith for reporting the problem. See ticket #1701.
- Disabled MPI_REAL16 and MPI_COMPLEX32 support on platforms where the bit representation of REAL*16 is different than that of the C type of the same size (usually long double). Thanks to Julien Devriendt for reporting the issue. See ticket #1603.
- Increased the size of MPI_MAX_PORT_NAME to 1024 from 36. See ticket #1533.
- Added "notify debugger on abort" feature. See tickets #1509 and #1510. Thanks to Seppo Sahrakropi for the bug report.
- Upgraded Open MPI tarballs to use Autoconf 2.63, Automake 1.10.1, Libtool 2.2.6a.
- Added missing MPI::Comm::Call_errhandler() function. Thanks to Dave Goodell for bringing this to our attention.
- Increased MPI_SUBVERSION value in mpi.h to 1 (i.e., MPI 2.1).
- Changed behavior of MPI_GRAPH_CREATE, MPI_TOPO_CREATE, and several other topology functions per MPI-2.1.
- Fix the type of the C++ constant MPI::IN_PLACE.
- Various enhancements to the openib BTL:
 - Added btl_openib_if_[in|ex]clude MCA parameters for including/excluding comma-delimited lists of HCAs and ports.
 - Added RDMA CM support, including btl_openib_cpc_[in|ex]clude MCA parameters
 - Added NUMA support to only use "near" network adapters
 - Added "Bucket SRQ" (BSRQ) support to better utilize registered memory, including btl_openib_receive_queues MCA parameter
 - Added ConnectX XRC support (and integrated with BSRQ)
 - Added btl_openib_ib_max_inline_data MCA parameter
 - Added iWARP support
 - Revamped flow control mechanisms to be more efficient
 - "mpi_leave_pinned=1" is now the default when possible, automatically improving performance for large messages when application buffers are re-used
- Eliminated duplicated error messages when multiple MPI processes fail with the same error.
- Added NUMA support to the shared memory BTL.
- Add Valgrind-based memory checking for MPI-semantic checks.
- Add support for some optional Fortran datatypes (MPI_LOGICAL1, MPI_LOGICAL2, MPI_LOGICAL4 and MPI_LOGICAL8).
- Remove the use of the STL from the C++ bindings.
- Added support for Platform/LSF job launchers. Must be Platform LSF v7.0.2 or later.
- Updated ROMIO with the version from MPICH2 1.0.7.

- Added RDMA capable one-sided component (called rdma), which can be used with BTL components that expose a full one-sided interface.
- Added the optional datatype MPI_REAL2. As this is added to the "end of" predefined datatypes in the fortran header files, there will not be any compatibility issues.
- Added Portable Linux Processor Affinity (PLPA) for Linux.
- Addition of a finer symbols export control via the visibility feature offered by some compilers.
- Added checkpoint/restart process fault tolerance support. Initially support a LAM/MPI-like protocol.
- Removed "mvapi" BTL; all InfiniBand support now uses the OpenFabrics driver stacks ("openib" BTL).
- Added more stringent MPI API parameter checking to help user-level debugging.
- The ptmalloc2 memory manager component is now by default built as a standalone library named libopenmpi-malloc. Users wanting to use leave_pinned with ptmalloc2 will now need to link the library into their application explicitly. All other users will use the libc-provided allocator instead of Open MPI's ptmalloc2. This change may be overridden with the configure option enable-ptmalloc2-internal
- The leave_pinned options will now default to using mallopt on Linux in the cases where ptmalloc2 was not linked in. mallopt will also only be available if munmap can be intercepted (the default whenever Open MPI is not compiled with --without-memory-manager).
- Open MPI will now complain and refuse to use leave_pinned if no memory intercept / mallopt option is available.
- Add option of using Perl-based wrapper compilers instead of the C-based wrapper compilers. The Perl-based version does not have the features of the C-based version, but does work better in cross-compile environments.

1.2.9 (unreleased)

- Fix a segfault when using one-sided communications on some forms of derived datatypes. Thanks to Dorian Krause for reporting the bug. See #1715.
- Fix an alignment problem affecting one-sided communications on some architectures (e.g., SPARC64). See #1738.
- Fix compilation on Solaris when thread support is enabled in Open MPI (e.g., when using --with-threads). See #1736.
- Correctly take into account the MTU that an OpenFabrics device port is using. See #1722 and https://bugs.openfabrics.org/show_bug.cgi?id=1369.
- Fix two datatype engine bugs. See #1677. Thanks to Peter Kjellstrom for the bugreport.
- Fix the bml r2 help filename so the help message can be found. See #1623.
- Fix a compilation problem on RHEL4U3 with the PGI 32 bit compiler caused by <infiniband/driver.h>. See ticket #1613.
- Fix the --enable-cxx-exceptions configure option. See ticket #1607.
- Properly handle when the MX BTL cannot open an endpoint. See ticket #1621.
- Fix a double free of events on the tcp_events list. See ticket #1631.
- Fix a buffer overrun in opal_free_list_grow (called by MPI_Init).

Appendix C. OpenMPI Release Information

Thanks to Patrick Farrell for the bugreport and Stephan Kramer for the bugfix. See ticket #1583.

- Fix a problem setting OPAL_PREFIX for remote sh-based shells. See ticket #1580.

1.2.8

- Tweaked one memory barrier in the openib component to be more conservative. May fix a problem observed on PPC machines. See ticket #1532.
- Fix OpenFabrics IB partition support. See ticket #1557.
- Restore v1.1 feature that sourced .profile on remote nodes if the default shell will not do so (e.g. /bin/sh and /bin/ksh). See ticket #1560.
- Fix segfault in MPI_Init_thread() if ompi_mpi_init() fails. See ticket #1562.
- Adjust SLURM support to first look for \$SLURM_JOB_CPUS_PER_NODE instead of the deprecated \$SLURM_TASKS_PER_NODE environment variable. This change may be **required** when using SLURM v1.2 and above. See ticket #1536.
- Fix the MPIR_Proctable to be in process rank order. See ticket #1529.
- Fix a regression introduced in 1.2.6 for the IBM eHCA. See ticket #1526.

1.2.7

- Add some Sun HCA vendor IDs. See ticket #1461.
- Fixed a memory leak in MPI_Alltoallw when called from Fortran. Thanks to Dave Grote for the bugreport. See ticket #1457.
- Only link in libutil when it is needed/desired. Thanks to Brian Barret for diagnosing and fixing the problem. See ticket #1455.
- Update some QLogic HCA vendor IDs. See ticket #1453.
- Fix F90 binding for MPI_CART_GET. Thanks to Scott Beardsley for bringing it to our attention. See ticket #1429.
- Remove a spurious warning message generated in/by ROMIO. See ticket #1421.
- Fix a bug where command-line MCA parameters were not overriding MCA parameters set from environment variables. See ticket #1380.
- Fix a bug in the AMD64 atomics assembly. Thanks to Gabriele Fatigati for the bug report and bugfix. See ticket #1351.
- Fix a gather and scatter bug on intercommunicators when the datatype being moved is 0 bytes. See ticket #1331.
- Some more man page fixes from the Debian maintainers. See tickets #1324 and #1329.
- Have openib BTL (OpenFabrics support) check for the presence of /sys/class/infiniband before allowing itself to be used. This check prevents spurious "OMPI did not find RDMA hardware!" notices on systems that have the software drivers installed, but no corresponding hardware. See tickets #1321 and #1305.
- Added vendor IDs for some ConnectX openib HCAs. See ticket #1311.
- Fix some RPM specfile inconsistencies. See ticket #1308. Thanks to Jim Kusznir for noticing the problem.
- Removed an unused function prototype that caused warnings on some systems (e.g., OS X). See ticket #1274.
- Fix a deadlock in inter-communicator scatter/gather operations. Thanks to Martin Audet for the bug report. See ticket #1268.

1.2.6

- Fix a bug in the inter-allgather for asymmetric inter-communicators. Thanks to Martin Audet for the bug report. See ticket #1247.
- Fix a bug in the openib BTL when setting the CQ depth. Thanks to Jon Mason for the bug report and fix. See ticket #1245.
- On Mac OS X Leopard, the execinfo component will be used for backtraces, making for a more durable solution. See ticket #1246.
- Added vendor IDs for some QLogic DDR openib HCAs. See ticket #1227.
- Updated the URL to get the latest config.guess and config.sub files. Thanks to Ralf Wildenhues for the bug report. See ticket #1226.
- Added shared contexts support to PSM MTL. See ticket #1225.
- Added pml_obl_use_early_completion MCA parameter to allow users to turn off the OBl early completion semantic and avoid "stall" problems seen on InfiniBand in some cases. See ticket #1224.
- Sanitized some #define macros used in mpi.h to avoid compiler warnings caused by MPI programs built with different autoconf versions. Thanks to Ben Allan for reporting the problem, and thanks to Brian Barrett for the fix. See ticket #1220.
- Some man page fixes from the Debian maintainers. See ticket #1219.
- Made the openib BTL a bit more resilient in the face of driver errors. See ticket #1217.
- Fixed F90 interface for MPI_CART_CREATE. See ticket #1208. Thanks to Michal Charemza for reporting the problem.
- Fixed some C++ compiler warnings. See ticket #1203.
- Fixed formatting of the orterun man page. See ticket #1202. Thanks to Peter Breitenlohner for the patch.

1.2.5

- Fixed compile issue with open() on Fedora 8 (and newer) platforms. Thanks to Sebastian Schmitzdorff for noticing the problem.
- Added run-time warnings during MPI_INIT when MPI_THREAD_MULTIPLE and/or progression threads are used (the OMPI v1.2 series does not support these well at all).
- Better handling of ECONNABORTED from connect on Linux. Thanks to Bob Soliday for noticing the problem; thanks to Brian Barrett for submitting a patch.
- Reduce extraneous output from OOB when TCP connections must be retried. Thanks to Brian Barrett for submitting a patch.
- Fix for ConnectX devices and OFED 1.3. See ticket #1190.
- Fixed a configure problem for Fortran 90 on Cray systems. Ticket #1189.
- Fix an uninitialized variable in the error case in opal_init.c. Thanks to Ake Sandgren for pointing out the mistake.
- Fixed a hang in configure if \$USER was not defined. Thanks to Darrell Kresge for noticing the problem. See ticket #900.
- Added support for parallel debuggers even when we have an optimized build. See ticket #1178.
- Worked around a bus error in the Mac OS X 10.5.X (Leopard) linker when compiling Open MPI with -g. See ticket #1179.
- Removed some warnings about 'rm' from Mac OS X 10.5 (Leopard) builds.

Appendix C. OpenMPI Release Information

- Fix the handling of `mx_finalize()`. See ticket #1177.
Thanks to Ake Sandgren for bringing this issue to our attention.
- Fixed minor file descriptor leak in the Altix timer code. Thanks to Paul Hargrove for noticing the problem and supplying the fix.
- Fix a problem when using a different compiler for C and Objective C. See ticket #1153.
- Fix segfault in `MPI_COMM_SPAWN` when the user specified a working directory. Thanks to Murat Knecht for reporting this and suggesting a fix.
- A few manpage fixes from the Debian Open MPI maintainers. Thanks to Tilman Koschnick, Sylvestre Ledru, and Dirk Eddelbuettel.
- Fixed issue with pthread detection when compilers are not all from the same vendor. Thanks to Ake Sandgren for the bug report. See ticket #1150.
- Fixed vector collectives in the self module. See ticket #1166.
- Fixed some data-type engine bugs: an indexing bug, and an alignment bug. See ticket #1165.
- Only set the `MPI_APPNUM` attribute if it is defined. See ticket #1164.

1.2.4

- Really added support for TotalView/DDT parallel debugger message queue debugging (it was mistakenly listed as "added" in the 1.2 release).
- Fixed a build issue with GNU/kFreeBSD. Thanks to Petr Salinger for the patch.
- Added missing `MPI_FILE_NULL` constant in Fortran. Thanks to Bernd Schubert for bringing this to our attention.
- Change such that the UDAPL BTL is now only built in Linux when explicitly specified via the `--with-udapl` configure command line switch.
- Fixed an issue with `umask` not being propagated when using the TM launcher.
- Fixed behavior if number of slots is not the same on all bproc nodes.
- Fixed a hang on systems without GPR support (ex. Cray XT3/4).
- Prevent users of 32-bit MPI apps from requesting ≥ 2 GB of shared memory.
- Added a Portals MTL.
- Fix 0 sized `MPI_ALLOC_MEM` requests. Thanks to Lisandro Dalcin for pointing out the problem.
- Fixed a segfault crash on large SMPs when doing collectives.
- A variety of fixes for Cray XT3/4 class of machines.
- Fixed which error handler is used when `MPI_COMM_SELF` is passed to `MPI_COMM_FREE`. Thanks to Lisandro Dalcini for the bug report.
- Fixed compilation on platforms that don't have `hton/ntoh`.
- Fixed a logic problem in the fortran binding for `MPI_TYPE_MATCH_SIZE`. Thanks to Jeff Dusenberry for pointing out the problem and supplying the fix.
- Fixed a problem with `MPI_BOTTOM` in various places of the f77-interface. Thanks to Daniel Spangberg for bringing this up.
- Fixed problem where MPI-optional Fortran datatypes were not correctly initialized.
- Fixed several problems with `stdin/stdout` forwarding.

- Fixed overflow problems with the sm mpool MCA parameters on large SMPs.
- Added support for the DDT parallel debugger via orterun's --debug command line option.
- Added some sanity/error checks to the openib MCA parameter parsing code.
- Updated the udapl BTL to use RDMA capabilities.
- Allow use of the BProc head node if it was allocated to the user. Thanks to Sean Kelly for reporting the problem and helping debug it.
- Fixed a ROMIO problem where non-blocking I/O errors were not properly reported to the user.
- Made remote process launch check the \$SHELL environment variable if a valid shell was not otherwise found for the user. Thanks to Alf Wachsmann for the bugreport and suggested fix.
- Added/updated some vendor IDs for a few openib HCAs.
- Fixed a couple of failures that could occur when specifying devices for use by the OOB.
- Removed dependency on sysfsutils from the openib BTL for libibverbs >=v1.1 (i.e., OFED 1.2 and beyond).

1.2.3

- Fix a regression in comm_spawn functionality that inadvertently caused the mapping of child processes to always start at the same place. Thanks to Prakash Velayutham for helping discover the problem.
- Fix segfault when a user's home directory is unavailable on a remote node. Thanks to Guillaume Thomas-Collignon for bringing the issue to our attention.
- Fix MPI_IProbe to properly handle MPI_STATUS_IGNORE on mx and psm MTLs. Thanks to Sophia Corwell for finding this and supplying a reproducer.
- Fix some error messages in the tcp BTL.
- Use _NSGetEnviron instead of environ on Mac OS X so that there are no undefined symbols in the shared libraries.
- On OS X, when MACOSX_DEPLOYMENT_TARGET is 10.3 or higher, support building the Fortran 90 bindings as a shared library. Thanks to Jack Howarth for his advice on making this work.
- No longer require extra include flag for the C++ bindings.
- Fix detection of weak symbols support with Intel compilers.
- Fix issue found by Josh England: ompi_info would not show framework MCA parameters set in the environment properly.
- Rename the oob_tcp_include/exclude MCA params to oob_tcp_if_include/exclude so that they match the naming convention of the btl_tcp_if_include/exclude params. The old names are deprecated, but will still work.
- Add -wd as a synonym for the -wdir orterun/mpirun option.
- Fix the mvapi BTL to compile properly with compilers that do not support anonymous unions. Thanks to Luis Kornbluh for reporting the bug.

1.2.2

- Fix regression in 1.2.1 regarding the handling of \$CC with both

Appendix C. OpenMPI Release Information

- absolute and relative path names.
- Fix F90 array of status dimensions. Thanks to Randy Bramley for noticing the problem.
- Add `btl_openib_ib_pkey_value` MCA parameter for controlling IB port selection.
- Fixed a variety of threading/locking bugs.
- Fixed some compiler warnings associated with ROMIO, OS X, and gridengine.
- If `pbs-config` can be found, use it to look for TM support. Thanks to Bas van der Vlies for the inspiration and preliminary work.
- Fixed a deadlock in `orterun` when the `rsh` PLS encounters some errors.

1.2.1

- Fixed a number of connection establishment errors in the TCP out-of-band messaging system.
- Fixed a memory leak when using `mpi_comm` calls. Thanks to Bas van der Vlies for reporting the problem.
- Fixed various memory leaks in OPAL and ORTE.
- Improved launch times when using TM (PBS Pro, Torque, Open PBS).
- Fixed `mpi_leave_pinned` to work for all datatypes.
- Fix functionality allowing users to disable `sbrk()` (the `mpool_base_disable_sbrk` MCA parameter) on platforms that support it.
- Fixed a pair of problems with the TCP "listen_thread" mode for the `oob_tcp_listen_mode` MCA parameter that would cause failures when attempting to launch applications.
- Fixed a segfault if there was a failure opening a BTL MX endpoint.
- Fixed a problem with `mpirun`'s `--nolocal` option introduced in 1.2.
- Re-enabled `MPI_COMM_SPAWN_MULTIPLE` from singletons.
- LoadLeveler and TM configure fixes, Thanks to Martin Audet for the bug report.
- Various C++ MPI attributes fixes.
- Fixed issues with backtrace code on 64 bit Intel & PPC OS X builds.
- Fixed issues with multi-word CC variables and `libtool`. Thanks to Bert Wesarg for the bug reports.
- Fix issue with non-uniform node naming schemes in SLURM.
- Fix file descriptor leak in the Grid Engine/NlGE support.
- Fix compile error on OS X 10.3.x introduced with Open MPI 1.1.5.
- Implement `MPI_TYPE_CREATE_DARRAY` function (was in 1.1.5 but not 1.2).
- Recognize `zsh` shell when using `rsh/ssh` for launching MPI jobs.
- Ability to set the `OPAL_DESTDIR` or `OPAL_PREFIX` environment variables to "re-root" an existing Open MPI installation.
- Always include `-I` for Fortran compiles, even if the prefix is `/usr/local`.
- Support for "fork()" in MPI applications that use the OpenFabrics stack (OFED v1.2 or later).
- Support for setting specific limits on registered memory.

1.2

- Fixed race condition in the shared memory fifo's, which led to orphaned messages.
- Corrected the size of the shared memory file - subtracted out the

- space the header was occupying.
- Add support for MPI_2COMPLEX and MPI_2DOUBLE_COMPLEX.
- Always ensure to create \$(includedir)/openmpi, even if the C++ bindings are disabled so that the wrapper compilers don't point to a directory that doesn't exist. Thanks to Martin Audet for identifying the problem.
- Fixes for endian handling in MPI process startup.
- Openib BTL initialization fixes for cases where MPI processes in the same job has different numbers of active ports on the same physical fabric.
- Print more descriptive information when displaying backtraces on OS's that support this functionality, such as the hostname and PID of the process in question.
- Fixes to properly handle MPI exceptions in C++ on communicators, windows, and files.
- Much more reliable runtime support, particularly with regards to MPI job startup scalability, BProc support, and cleanup in failure scenarios (e.g., MPI_ABORT, MPI processes abnormally terminating, etc.).
- Significant performance improvements for MPI collectives, particularly on high-speed networks.
- Various fixes in the MX BTL component.
- Fix C++ typecast problems with MPI_ERRCODES_IGNORE. Thanks to Satish Balay for bringing this to our attention.
- Allow run-time specification of the maximum amount of registered memory for OpenFabrics and GM.
- Users who utilize the wrapper compilers (e.g., mpicc and mpif77) will not notice, but the underlying library names for ORTE and OPAL have changed to libopen-rte and libopen-pal, respectively (listed here because there are undoubtedly some users who are not using the wrapper compilers).
- Many bug fixes to MPI-2 one-sided support.
- Added support for TotalView message queue debugging.
- Fixes for MPI_STATUS_SET_ELEMENTS.
- Print better error messages when mpirun's "--nolocal" is used when there is only one node available.
- Added man pages for several Open MPI executables and the MPI API functions.
- A number of fixes for Alpha platforms.
- A variety of Fortran API fixes.
- Build the Fortran MPI API as a separate library to allow these functions to be profiled properly.
- Add new --enable-mpirun-prefix-by-default configure option to always imply the --prefix option to mpirun, preventing many rsh/ssh-based users from needing to modify their shell startup files.
- Add a number of missing constants in the C++ bindings.
- Added tight integration with Sun N1 Grid Engine (N1GE) 6 and the open source Grid Engine.
- Allow building the F90 MPI bindings as shared libraries for most compilers / platforms. Explicitly disallow building the F90 bindings as shared libraries on OS X because of complicated situations with Fortran common blocks and lack of support for unresolved common symbols in shared libraries.
- Added stacktrace support for Solaris and Mac OS X.
- Update event library to libevent-1.1b.

Appendix C. OpenMPI Release Information

- Fixed standards conformance issues with MPI_ERR_TRUNCATED and setting MPI_ERROR during MPI_TEST/MPI_WAIT.
- Addition of "cm" PML to better support library-level matching interconnects, with support for Myrinet/MX, and QLogic PSM-based networks.
- Addition of "udapl" BTL for transport across uDAPL interconnects.
- Really check that the \$CXX given to configure is a C++ compiler (not a C compiler that "sorta works" as a C++ compiler).
- Properly check for local host only addresses properly, looking for 127.0.0.0/8, rather than just 127.0.0.1.

1.1.5

- Implement MPI_TYPE_CREATE_DARRAY function.
- Fix race condition in shared memory BTL startup that could cause MPI applications to hang in MPI_INIT.
- Fix syntax error in a corner case of the event library. Thanks to Bert Wesarg for pointing this out.
- Add new MCA parameter (mpi_preconnect_oob) for pre-connecting the "out of band" channels between all MPI processes. Most helpful for MPI applications over InfiniBand where process A sends an initial message to process B, but process B does not enter the MPI library for a long time.
- Fix for a race condition in shared memory locking semantics.
- Add major, minor, and release version number of Open MPI to mpi.h. Thanks to Martin Audet for the suggestion.
- Fix the "restrict" compiler check in configure.
- Fix a problem with argument checking in MPI_TYPE_CREATE_SUBARRAY.
- Fix a problem with compiling the XGrid components with non-gcc compilers.

1.1.4

- Fixed 64-bit alignment issues with TCP interface detection on intel-based OS X machines.
- Adjusted TCP interface selection to automatically ignore Linux channel-bonded slave interfaces.
- Fixed the type of the first parameter to the MPI_F90 binding for MPI_INITIALIZED. Thanks to Tim Campbell for pointing out the problem.
- Fix a bunch of places in the Fortran MPI bindings where (MPI_Fint*) was mistakenly being used instead of (MPI_Aint*).
- Fixes for fortran MPI_STARTALL, which could sometimes return incorrect request values. Thanks to Tim Campbell for pointing out the problem.
- Include both pre- and post-MPI-2 errata bindings for MPI::Win::Get_attr.
- Fix math error on Intel OS X platforms that would greatly increase shared memory latency.
- Fix type casting issue with MPI_ERRCODES_IGNORE that would cause errors when using a C++ compiler. Thanks to Barry Smith for

bringing this to our attention.

- Fix possible segmentation fault during shutdown when using the MX BTL.

1.1.3

- Remove the "hierarch" coll component; it was not intended to be included in stable releases yet.
- Fix a race condition with stdout/stderr not appearing properly from all processes upon termination of an MPI job.
- Fix internal accounting errors with the self BTL.
- Fix typos in the code path for when `sizeof(int) != sizeof(INTEGER)` in the MPI F77 bindings functions. Thanks to Pierre-Matthieu Anglade for bringing this problem to our attention.
- Fix for a memory leak in the derived datatype function `ompi_ddt_duplicate()`. Thanks to Andreas Schafer for reporting, diagnosing, and patching the leak.
- Used better performing basic algorithm for `MPI_ALLGATHERV`.
- Added a workaround for a bug in the Intel 9.1 C++ compiler (all versions up to and including 20060925) in the MPI C++ bindings that caused run-time failures. Thanks to Scott Weitzenkamp for reporting this problem.
- Fix `MPI_SIZEOF` implementation in the F90 bindings for `COMPLEX` variable types.
- Fixes for persistent requests involving `MPI_PROC_NULL`. Thanks to Lisandro Dalcin for reporting the problem.
- Fixes to `MPI_TEST*` and `MPI_WAIT*` for proper MPI exception reporting. Thanks to Lisandro Dalcin for finding the issue.
- Various fixes for MPI generalized request handling; addition of missing `MPI::Grequest` functionality to the C++ bindings.
- Add "mpi_preconnect_all" MCA parameter to force wireup of all MPI connections during `MPI_INIT` (vs. making connections lazily whenever the first MPI communication occurs between a pair of peers).
- Fix a problem for when `$FC` and/or `$F77` were specified as multiple tokens. Thanks to Orion Poplawski for identifying the problem and to Ralf Wildenhues for suggesting the fix.
- Fix several `MPI_*ERRHANDLER*` functions and `MPI_GROUP_TRANSLATE_RANKS` with respect to what arguments they allowed and the behavior that they effected. Thanks to Lisandro Dalcin for reporting the problems.

1.1.2

- Really fix Fortran status handling in `MPI_WAIT SOME` and `MPI_TEST SOME`.
- Various datatype fixes, reported by several users as causing failures in the BLACS testing suite. Thanks to Harald Forbert, Ake Sandgren and, Michael Kluskens for reporting the problem.
- Correctness and performance fixes for heterogeneous environments.
- Fixed a error in command line parsing on some platforms (causing `mpirun` to crash without doing anything).
- Fix for initialization hangs on 64 bit Mac OS X PowerPC systems.

Appendix C. OpenMPI Release Information

- Fixed some memory allocation problems in mpirun that could cause random problems if "-np" was not specified on the command line.
- Add Kerberos authentication support for XGrid.
- Added LoadLeveler support for jobs larger than 128 tasks.
- Fix for large-sized Fortran LOGICAL datatypes.
- Fix various error checking in MPI_INFO_GET_NTHKEY and MPI_GROUP_TRANSLATE_RANKS, and some collective operations (particularly with regards to MPI_IN_PLACE). Thanks to Lisandro Dalcin for reporting the problems.
- Fix receiving messages to buffers allocated by MPI_ALLOC_MEM.
- Fix a number of race conditions with the MPI-2 Onesided interface.
- Fix the "tuned" collective component where some cases where MPI_BCAST could hang.
- Update TCP support to support non-uniform TCP environments.
- Allow the "poe" RAS component to be built on AIX or Linux.
- Only install mpif.h if the rest of the Fortran bindings are installed.
- Fixes for BProc node selection.
- Add some missing Fortran MPI-2 IO constants.

1.1.1

- Fix for Fortran string handling in various MPI API functions.
- Fix for Fortran status handling in MPI_WAITSSOME and MPI_TESTSSOME.
- Various fixes for the XL compilers.
- Automatically disable using malloc() on AIX.
- Memory fixes for 64 bit platforms with registering MCA parameters in the self and MX BTL components.
- Fixes for BProc to support oversubscription and changes to the mapping algorithm so that mapping processes "by slot" works as expected.
- Fixes for various abort cases to not hang and clean up nicely.
- If using the Intel 9.0 v20051201 compiler on an IA64 platform, the ptmalloc2 memory manager component will automatically disable itself. Other versions of the Intel compiler on this platform seem to work fine (e.g., 9.1).
- Added "host" MPI_Info key to MPI_COMM_SPAWN and MPI_COMM_SPAWN_MULTIPLE.
- Add missing C++ methods: MPI::Datatype::Create_indexed_block, MPI::Datatype::Create_resized, MPI::Datatype::Get_true_extent.
- Fix OSX linker issue with Fortran bindings.
- Fixed MPI_COMM_SPAWN to start spawning new processes in slots that (according to Open MPI) are not already in use.
- Added capability to "mpirun a.out" (without specifying -np) that will run on all currently-allocated resources (e.g., within a batch job such as SLURM, Torque, etc.).
- Fix a bug with one particular case of MPI_BCAST. Thanks to Doug Gregor for identifying the problem.
- Ensure that the shared memory mapped file is only created when there is more than one process on a node.
- Fixed problems with BProc stdin forwarding.
- Fixed problem with MPI_TYPE_INDEXED datatypes. Thanks to Yven

- Fournier for identifying this problem.
- Fix some thread safety issues in MPI attributes and the openib BTL.
 - Fix the BProc allocator to not potentially use the same resources across multiple ORTE universes.
 - Fix gm resource leak.
 - More latency reduction throughout the code base.
 - Make the TM PLS (PBS Pro, Torque, Open PBS) more scalable, and fix some latent bugs that crept in v1.1. Thanks to the Thunderbird crew at Sandia National Laboratories and Martin Schaffoner for access to testing facilities to make this happen.
 - Added new command line options to mpirun:
 - nolocal: Do not run any MPI processes on the same node as mpirun (compatibility with the OSC mpiexec launcher)
 - nooversubscribe: Abort if the number of processes requested would cause oversubscription
 - quiet / -q: do not show spurious status messages
 - version / -V: show the version of Open MPI
 - Fix bus error in XGrid process starter. Thanks to Frank from the Open MPI user's list for identifying the problem.
 - Fix data size mismatches that caused memory errors on PPC64 platforms during the startup of the openib BTL.
 - Allow propagation of SIGUSR1 and SIGUSR2 signals from mpirun to back-end MPI processes.
 - Add missing MPI::Is_finalized() function.

1.1

- Various MPI datatype fixes, optimizations.
- Fixed various problems on the SPARC architecture (e.g., not correctly aligning addresses within structs).
- Improvements in various run-time error messages to be more clear about what they mean and where the errors are occurring.
- Various fixes to mpirun's handling of --prefix.
- Updates and fixes for Cray/Red Storm support.
- Major improvements to the Fortran 90 MPI bindings:
 - General improvements in compile/linking time and portability between different F90 compilers.
 - Addition of "trivial", "small" (the default), and "medium" Fortran 90 MPI module sizes (v1.0.x's F90 module was equivalent to "medium"). See the README file for more explanation.
 - Fix various MPI F90 interface functions and constant types to match. Thanks to Michael Kluskens for pointing out the problems to us.
- Allow short messagees to use RDMA (vs. send/receive semantics) to a limited number peers in both the mvapi and openib BTL components. This reduces communication latency over IB channels.
- Numerous performance improvements throughout the entire code base.
- Many minor threading fixes.
- Add a define OMPI_SKIP_CXX to allow the user to skip the mpicxx.h from being included in mpi.h. It allows the user to compile C code with a CXX compiler without including the CXX bindings.
- PERUSE support has been added. In order to activate it add

Appendix C. OpenMPI Release Information

- enable-peruse to the configure options. All events described in the PERUSE 2.0 draft are supported, plus one Open MPI extension. PERUSE_COMM_REQ_XFER_CONTINUE allow to see how the data is segmented internally, using multiple interfaces or the pipeline engine. However, this version only support one event of each type simultaneously attached to a communicator.
- Add support for running jobs in heterogeneous environments. Currently supports environments with different endianness and different representations of C++ bool and Fortran LOGICAL. Mismatched sizes for other datatypes is not supported.
- Open MPI now includes an implementation of the MPI-2 One-Sided Communications specification.
- Open MPI is now configurable in cross-compilation environments. Several Fortran 77 and Fortran 90 tests need to be pre-seeded with results from a config.cache-like file.
- Add --debug option to mpirun to generically invoke a parallel debugger.

1.0.3 (unreleased; all fixes included in 1.1)

-
- Fix a problem noted by Chris Hennes where MPI_INFO_SET incorrectly disallowed long values.
 - Fix a problem in the launch system that could cause inconsistent launch behavior, particularly when launching large jobs.
 - Require that the openib BTL find <sysfs/libsysfs.h>. Thanks to Josh Aune for the suggestion.
 - Include updates to support the upcoming Autoconf 2.60 and Libtool 2.0. Thanks to Ralf Wildenhues for all the work!
 - Fix bug with infinite loop in the "round robin" process mapper. Thanks to Paul Donohue for reporting the problem.
 - Ensure that memory hooks are removed properly during MPI_FINALIZE. Thanks to Neil Ludban for reporting the problem.
 - Various fixes to the included support for ROMIO.
 - Fix to ensure that MPI_LONG_LONG and MPI_LONG_LONG_INT are actually synonyms, as defined by the MPI standard. Thanks to Martin Audet for reporting this.
 - Fix Fortran 90 configure tests to properly utilize LDFLAGS and LIBS. Thanks to Terry Reeves for reporting the problem.
 - Fix shared memory progression in asynchronous progress scenarios. Thanks to Mykael Bouquey for reporting the problem.
 - Fixed back-end operations for predefined MPI_PROD for some datatypes. Thanks to Bert Wesarg for reporting this.
 - Adapted configure to be able to handle Torque 2.1.0p0's (and above) new library name. Thanks to Brock Palen for pointing this out and providing access to a Torque 2.1.0p0 cluster to test with.
 - Fixed situation where mpirun could set a shell pipeline's stdout to non-blocking, causing the shell pipeline to prematurely fail. Thanks to Darrell Kresge for figuring out what was happening.
 - Fixed problems with leave_pinned that could cause Badness with the mvapi BTL.
 - Fixed problems with MPI_FILE_OPEN and non-blocking MPI-2 IO access.
 - Fixed various InfiniBand port matching issues during startup. Thanks to Scott Weitzenkamp for identifying these problems.
 - Fixed various configure, build and run-time issues with ROMIO.

- Thanks to Dries Kimpe for bringing them to our attention.
- Fixed error in MPI_COMM_SPLIT when dealing with intercommunicators. Thanks to Bert Wesarg for identifying the problem.
 - Fixed backwards handling of "high" parameter in MPI_INTERCOMM_MERGE. Thanks to Michael Kluskens for pointing this out to us.
 - Fixed improper handling of string arguments in Fortran bindings for MPI-IO functionality
 - Fixed segmentation fault with 64 bit applications on Solaris when using the shared memory transports.
 - Fixed MPI_COMM_SELF attributes to free properly at the beginning of MPI_FINALIZE. Thanks to Martin Audet for bringing this to our attention.
 - Fixed alignment tests for cross-compiling to not cause errors with recent versions of GCC.

1.0.2

- Fixed assembly race condition on AMD64 platforms.
- Fixed residual .TRUE. issue with copying MPI attributes set from Fortran.
- Remove unnecessary logic from Solaris pty I/O forwarding. Thanks to Francoise Roch for bringing this to our attention.
- Fixed error when count = 0 was given for multiple completion MPI functions (MPI_TESTSOME, MPI_TESTANY, MPI_TESTALL, MPI_WAIT SOME, MPI_WAITANY, MPI_WAITALL).
- Better handling in MPI_ABORT for when peer processes have already died, especially under some resource managers.
- Random updates to README file, to include notes about the Portland compilers.
- Random, small threading fixes to prevent deadlock.
- Fixed a problem with handling long mpirun app files. Thanks to Ravi Manumachu for identifying the problem.
- Fix handling of strings in several of the Fortran 77 bindings.
- Fix LinuxPPC assembly issues. Thanks to Julian Seward for reporting the problem.
- Enable pty support for standard I/O forwarding on platforms that have ptys but do not have openpty(). Thanks to Pierre Valiron for bringing this to our attention.
- Disable inline assembly for PGI compilers to avoid compiler errors. Thanks to Troy Telford for bringing this to our attention.
- Added MPI_UNSIGNED_CHAR and MPI_SIGNED_CHAR to the allowed reduction types.
- Fix a segv in variable-length message displays on Opterons running Solaris. Thanks to Pierre Valiron for reporting the issue.
- Added MPI_BOOL to the intrinsic reduction operations MPI_LAND, MPI_LOR, MPI_LXOR. Thanks to Andy Selle for pointing this out to us.
- Fixed TCP BTL network matching logic during MPI_INIT; in some cases on multi-NIC nodes, a NIC could get paired with a NIC on another network (typically resulting in deadlock). Thanks to Ken Mighell for pointing this out to us.
- Change the behavior of orterun (mpirun, mpirexec) to search for argv[0] and the cwd on the target node (i.e., the node where the executable will be running in all systems except BProc, where the

Appendix C. OpenMPI Release Information

- searches are run on the node where orterun is invoked).
- Fix race condition in shared memory transport that could cause crashes on machines with weak memory consistency models (including POWER/PowerPC machines).
- Fix warnings about setting read-only MCA parameters on bproc systems.
- Change the exit status set by mpirun when an application process is killed by a signal. The exit status is now set to signo + 128, which conforms with the behavior of (almost) all shells.
- Correct a datatype problem with the convertor when partially unpacking data. Now we can position the convertor to any position not only on the predefined types boundaries. Thanks to Yvan Fournier for reporting this to us.
- Fix a number of standard I/O forwarding issues, including the ability to background mpirun and a loss of data issue when redirecting mpirun's standard input from a file.
- Fixed bug in ompi_info where rcache and bml MCA parameters would not be displayed.
- Fixed umask issues in the session directory. Thanks to Glenn Morris for reporting this to us.
- Fixed tcsh-based LD_LIBRARY_PATH issues with --prefix. Thanks to Glen Morris for identifying the problem and suggesting the fix.
- Removed extraneous \n's when setting PATH and LD_LIBRARY_PATH in the rsh startup. Thanks to Glen Morris for finding these typos.
- Fixed missing constants in MPI C++ bindings.
- Fixed some errors caused by threading issues.
- Fixed openib BTL flow control logic to not overrun the number of send wqes available.
- Update to match newest OpenIB user-level library API. Thanks to Roland Dreier for submitting this patch.
- Report errors properly when failing to register memory in the openib BTL.
- Reduce memory footprint of openib BTL.
- Fix parsing problem with mpirun's "-tv" switch. Thanks to Chris Gottbrath for supplying the fix.
- Fix Darwin net/if.h configure warning.
- The GNU assembler unbelievably defaults to making stacks executable. So when using gas, add flags to explicitly tell it to not make stacks executable (lame but necessary).
- Add missing MPI::Request::Get_status() methods. Thanks to Bill Saphir for pointing this out to us.
- Improved error messages on memory registration errors (e.g., when using high-speed networks).
- Open IB support now checks firmware for how many outstanding RDMA requests are supported. Thanks to Mellanox for pointing this out to us.
- Enable printing of stack traces in MPI processes upon SIGBUS, SIGSEGV, and SIGFPE if the platform supports it.
- Fixed F90 compilation support for the Lahey compiler.
- Fixed issues with ROMIO shared library support.
- Fixed internal accounting problems with rsh support.
- Update to GNU Libtool 1.5.22.
- Fix error in configure script when setting CCAS to ias (the Intel assembler).
- Added missing MPI::Intercomm collectives.
- Fixed MPI_IN_PLACE handling for Fortran collectives.

- Fixed some more C++ `const_cast<>` issues. Thanks for Martin Audet (again) for bringing this to our attention.
- Updated ROMIO with the version from MPICH 1.2.7p1, marked as version 2005-06-09.
- Fixes for some cases where the use of `MPI_BOTTOM` could cause problems.
- Properly handle the case where an mVAPI does not have shared receive queue support (such as the one shipped by SilverStorm / Infinicon for OS X).

1.0.1

- Fixed assembly on Solaris AMD platforms. Thanks to Pierre Valiron for bringing this to our attention.
- Fixed long messages in the send-to-self case.
- Ensure that when the "leave_pinned" option is used, the memory hooks are also enabled. Thanks to Gleb Natapov for pointing this out.
- Fixed compile errors for IRIX.
- Allow hostfiles to have integer host names (for BProc clusters).
- Fixed a problem with message matching of out-of-order fragments in multiple network device scenarios.
- Converted all the C++ MPI bindings to use proper `const_cast<>`'s instead of old C-style casts to get rid of const-ness. Thanks to Martin Audet for raising the issue with us.
- Converted `MPI_Offset` to be a typedef instead of a #define because it causes problems for some C++ parsers. Thanks to Martin Audet for bringing this to our attention.
- Improved latency of TCP BTL.
- Fixed index value in `MPI_TESTANY` to be `MPI_UNDEFINED` if some requests were not `MPI_REQUEST_NULL`, but no requests finished.
- Fixed several Fortran MPI API implementations that incorrectly used integers instead of logicals or address-sized integers.
- Fix so that Open MPI correctly handles the Fortran value for `.TRUE.`, regardless of what the Fortran compiler's value for `.TRUE.` is.
- Improved scalability of MX startup.
- Fix datatype offset handling in the coll basic component's `MPI_SCATTERV` implementation.
- Fix EOF handling on stdin.
- Fix missing `MPI_F_STATUS_IGNORE` and `MPI_F_STATUSES_IGNORE` instantiations. Thanks to Anthony Chan for pointing this out.
- Add a missing value for `MPI_WIN_NULL` in `mpif.h`.
- Bring over some fixes for the sm btl that somehow didn't make it over from the trunk before v1.0. Thanks to Beth Tibbitts and Bill Chung for helping identify this issue.
- Bring over some fixes for the iof that somehow didn't make it over from the trunk before v1.0.
- Fix for `--with-wrapper-ldflags` handling. Thanks to Dries Kimpe for pointing this out to us.

1.0

Appendix C. OpenMPI Release Information

Initial public release.

Notes

1. <http://www.open-mpi.org/>

Appendix D. MPICH2 Release Information

The following is reproduced essentially verbatim from files contained within the MPICH2 tarball downloaded from <http://www.mpich.org/downloads/>.

NOTE: MPICH-2 has been effectively deprecated by the Open Source Community in favor of MPICH-3, which Scyld ClusterWare distributes as a set of *mpich-scyld* RPMs. Scyld ClusterWare continues to distribute *mpich2-scyld*, although we encourage users to migrate to MPICH-3, which enjoys active support by the Community.

```
=====
                          Changes in 1.5
=====
```

```
# OVERALL: Nemesis now supports an "--enable-yield=..." configure
option for better performance/behavior when oversubscribing
processes to cores. Some form of this option is enabled by default
on Linux, Darwin, and systems that support sched_yield().

# OVERALL: Added support for Intel Many Integrated Core (MIC)
architecture: shared memory, TCP/IP, and SCIF based communication.

# OVERALL: Added support for IBM BG/Q architecture. Thanks to IBM
for the contribution.

# MPI-3: const support has been added to mpi.h, although it is
disabled by default. It can be enabled on a per-translation unit
basis with "#define MPICH2_CONST const".

# MPI-3: Added support for MPIX_Type_create_hindexed_block.

# MPI-3: The new MPI-3 nonblocking collective functions are now
available as "MPIX_" functions (e.g., "MPIX_Ibcast").

# MPI-3: The new MPI-3 neighborhood collective routines are now available as
"MPIX_" functions (e.g., "MPIX_Neighbor_allgather").

# MPI-3: The new MPI-3 MPI_Comm_split_type function is now available
as an "MPIX_" function.

# MPI-3: The new MPI-3 tools interface is now available as "MPIX_T_"
functions. This is a beta implementation right now with several
limitations, including no support for multithreading. Several
performance variables related to CH3's message matching are exposed
through this interface.

# MPI-3: The new MPI-3 matched probe functionality is supported via
the new routines MPIX_Mprobe, MPIX_Improbe, MPIX_Mrecv, and
MPIX_Imrecv.

# MPI-3: The new MPI-3 nonblocking communicator duplication routine,
MPIX_Comm_idup, is now supported. It will only work for
single-threaded programs at this time.

# MPI-3: MPIX_Comm_reenable_anysource support
```

Appendix D. MPICH2 Release Information

```
# MPI-3: Native MPIX_Comm_create_group support (updated version of
the prior MPIX_Group_comm_create routine).

# MPI-3: MPI_Intercomm_create's internal communication no longer interferes
with point-to-point communication, even if point-to-point operations on the
parent communicator use the same tag or MPI_ANY_TAG.

# MPI-3: Eliminated the possibility of interference between
MPI_Intercomm_create and point-to-point messaging operations.

# Build system: Completely revamped build system to rely fully on
autotools. Parallel builds ("make -j8" and similar) are now supported.

# Build system: rename "./maint/updatefiles" --> "./autogen.sh" and
"configure.in" --> "configure.ac"

# JUMPSHOT: Improvements to Jumpshot to handle thousands of
timelines, including performance improvements to slog2 in such
cases.

# JUMPSHOT: Added navigation support to locate chosen drawable's ends
when viewport has been scrolled far from the drawable.

# PM/PMI: Added support for memory binding policies.

# PM/PMI: Various improvements to the process binding support in
Hydra. Several new pre-defined binding options are provided.

# PM/PMI: Upgraded to hwloc-1.5

# PM/PMI: Several improvements to PBS support to natively use the PBS
launcher.

# Several other minor bug fixes, memory leak fixes, and code cleanup.
A full list of changes is available using:

    svn log -r8478:HEAD https://svn.mcs.anl.gov/repos/mpi/mpich2/tags/release/mpich2-1.5
... or at the following link:

    https://trac.mcs.anl.gov/projects/mpich2/log/mpich2/tags/release/
    mpich2-1.5?action=follow_copy&rev=HEAD&stop_rev=8478&mode=follow_copy

=====
                        Changes in 1.4.1
=====

# OVERALL: Several improvements to the ARMCI API implementation
within MPICH2.

# Build system: Added beta support for DESTDIR while installing
MPICH2.

# PM/PMI: Upgrade hwloc to 1.2.1rc2.
```

```
# PM/PMI: Initial support for the PBS launcher.

# Several other minor bug fixes, memory leak fixes, and code cleanup.
A full list of changes is available using:

svn log -r8675:HEAD \
  https://svn.mcs.anl.gov/repos/mpi/mpich2/tags/release/mpich2-1.4.1

... or at the following link:

https://trac.mcs.anl.gov/projects/mpich2/log/mpich2/tags/release/
mpich2-1.4.1?action=follow_copy&rev=HEAD&stop_rev=8675&mode=follow_copy
```

```
=====
                          Changes in 1.4
=====
```

```
# OVERALL: Improvements to fault tolerance for collective
operations. Thanks to Rui Wang @ ICT for reporting several of these
issues.

# OVERALL: Improvements to the universe size detection. Thanks to
Yauheni Zelenko for reporting this issue.

# OVERALL: Bug fixes for Fortran attributes on some systems. Thanks
to Nicolai Stange for reporting this issue.

# OVERALL: Added new ARMCI API implementation (experimental).

# OVERALL: Added new MPIX_Group_comm_create function to allow
non-collective creation of sub-communicators.

# FORTRAN: Bug fixes in the MPI_DIST_GRAPH_ Fortran bindings.

# PM/PMI: Support for a manual "none" launcher in Hydra to allow for
higher-level tools to be built on top of Hydra. Thanks to Justin
Wozniak for reporting this issue, for providing several patches for
the fix, and testing it.

# PM/PMI: Bug fixes in Hydra to handle non-uniform layouts of hosts
better. Thanks to the MVAPICH group at OSU for reporting this issue
and testing it.

# PM/PMI: Bug fixes in Hydra to handle cases where only a subset of
the available launchers or resource managers are compiled
in. Thanks to Satish Balay @ Argonne for reporting this issue.

# PM/PMI: Support for a different username to be provided for each
host; this only works for launchers that support this (such as
SSH).

# PM/PMI: Bug fixes for using Hydra on AIX machines. Thanks to
Kitrick Sheets @ NCSA for reporting this issue and providing the
first draft of the patch.
```

Appendix D. MPICH2 Release Information

```
# PM/PMI: Bug fixes in memory allocation/management for environment
variables that was showing up on older platforms. Thanks to Steven
Sutphen for reporting the issue and providing detailed analysis to
track down the bug.

# PM/PMI: Added support for providing a configuration file to pick
the default options for Hydra. Thanks to Saurabh T. for reporting
the issues with the current implementation and working with us to
improve this option.

# PM/PMI: Improvements to the error code returned by Hydra.

# PM/PMI: Bug fixes for handling "=" in environment variable values in
hydra.

# PM/PMI: Upgrade the hwloc version to 1.2.

# COLLECTIVES: Performance and memory usage improvements for MPI_Bcast
in certain cases.

# VALGRIND: Fix incorrect Valgrind client request usage when MPICH2 is
built for memory debugging.

# BUILD SYSTEM: "--enable-fast" and "--disable-error-checking" are once
again valid simultaneous options to configure.

# TEST SUITE: Several new tests for MPI RMA operations.

# Several other minor bug fixes, memory leak fixes, and code cleanup.
A full list of changes is available using:

svn log -r7838:HEAD \
  https://svn.mcs.anl.gov/repos/mpi/mpich2/tags/release/mpich2-1.4

... or at the following link:

https://trac.mcs.anl.gov/projects/mpich2/log/mpich2/tags/release/
mpich2-1.4?action=follow_copy&rev=HEAD&stop_rev=7838&mode=follow_copy
```

KNOWN ISSUES

Known runtime failures

- * MPI_Alltoall might fail in some cases because of the newly added fault-tolerance features. If you are seeing this error, try setting the environment variable `MPICH_ENABLE_COLL_FT_RET=0`.

Threads

- * `ch3:sock` does not (and will not) support fine-grained threading.

- * MPI-IO APIs are not currently thread-safe when using fine-grained threading (`--enable-thread-cs=per-object`).
- * `ch3:nemesis:tcp` fine-grained threading is still experimental and may have correctness or performance issues. Known correctness issues include dynamic process support and generalized request support.

Lacking channel-specific features

- * `ch3` does not presently support communication across heterogeneous platforms (e.g., a big-endian machine communicating with a little-endian machine).
- * `ch3:nemesis:mx` does not support dynamic processes at this time.
- * Support for "external32" data representation is incomplete. This affects the `MPI_Pack_external` and `MPI_Unpack_external` routines, as well the external data representation capabilities of ROMIO.
- * `ch3` has known problems in some cases when threading and dynamic processes are used together on communicators of size greater than one.

Build Platforms

- * Builds using the native "make" program on OpenSolaris fail unknown reasons. A workaround is to use GNU Make instead. See the following ticket for more information:

<http://trac.mcs.anl.gov/projects/mpich2/ticket/1122>

- * Build fails with Intel compiler suite 13.0, because of weak symbol issues in the compiler. A workaround is to disable weak symbol support by passing `--disable-weak-symbols` to configure. See the following ticket for more information:

<https://trac.mcs.anl.gov/projects/mpich2/ticket/1659>

- * The `sctp` channel is fully supported for FreeBSD and Mac OS X. As of the time of this release, bugs in the stack currently existed in the Linux kernel, and will hopefully soon be resolved. It is known to not work under Solaris and Windows. For Solaris, the SCTP API available in the kernel of standard Solaris 10 is a subset of the standard API used by the `sctp` channel. Cooperation with the Sun SCTP developers to support `ch3:sctp` under Solaris for future releases is currently ongoing. For Windows, no known kernel-based SCTP stack for Windows currently exists.

Process Managers

- * The MPD process manager can only handle relatively small amounts of data on stdin and may also have problems if there is data on stdin

Appendix D. MPICH2 Release Information

that is not consumed by the program.

- * The SMPD process manager does not work reliably with threaded MPI processes. `MPI_Comm_spawn()` does not currently work for ≥ 256 arguments with `smpd`.

Performance issues

- * SMP-aware collectives do not perform as well, in select cases, as non-SMP-aware collectives, e.g. `MPI_Reduce` with message sizes larger than 64KiB. These can be disabled by the configure option `"--disable-smpcoll"`.
- * `MPI_Irecv` operations that are not explicitly completed before `MPI_Finalize` is called may fail to complete before `MPI_Finalize` returns, and thus never complete. Furthermore, any matching send operations may erroneously fail. By explicitly completed, we mean that the request associated with the operation is completed by one of the `MPI_Test` or `MPI_Wait` routines.

C++ Binding:

- * The MPI datatypes corresponding to Fortran datatypes are not available (e.g., no `MPI::DOUBLE_PRECISION`).
- * The C++ binding does not implement a separate profiling interface, as allowed by the MPI-2 Standard (Section 10.1.10 Profiling).
- * `MPI::ERRORS_RETURN` may still throw exceptions in the event of an error rather than silently returning.

Notes

1. <http://www.mpich.org/downloads/>

Appendix E. MVAPICH2 Release Information

The following is reproduced essentially verbatim from files contained within the MVAPICH2 tarball downloaded from <http://mvapich.cse.ohio-state.edu/>

The MVAPICH2 2.0 User Guide is available at http://mvapich.cse.ohio-state.edu/support/user_guide_mvapich2-2.0rc1.html MVAPICH2-2.1 introduces an algorithm to determine CPU topology on the node, and this new algorithm does not work properly for older Mellanox controllers and firmware, resulting in software threads not spreading out across a node's cores by default.

Prior to updating to MVAPICH2-2.1 or newer, the cluster administrator should determine the potential vulnerability to this problem. For each node that contains an Infiniband controller, execute **ibstat**, and if the first output line is:

```
CA 'mthca0'
```

then that node *may* exhibit the problem. The cluster administrator has two choices: either avoid updating the `mvapich2-scyld` packages (keeping in mind that the `mvapich2-psm-scyld` packages can be updated, as those packages are only used by QLogic Infiniband controllers, which don't have the problem); or update `mvapich2-scyld`, execute tests to determine if the problem exists for those Mellanox *mthca* nodes, and if the problem does exist, then instruct users to employ explicit CPU Mapping. See <http://mvapich.cse.ohio-state.edu/static/media/mvapich/mvapich2-2.1-userguide.html#x1-540006.5> for details.

MVAPICH2 Changelog

This file briefly describes the changes to the MVAPICH2 software package. The logs are arranged in the "most recent first" order.

MVAPICH2 2.2 (09/07/2016)

- * Features and Enhancements (since 2.2rc2):
 - Single node collective tuning for Bridges@PSC, Stampede@TACC and other architectures
 - Enable PSM builds when both PSM and PSM2 libraries are present
 - Thanks to Adam T. Moody@LLNL for the report and patch
 - Add support for HCAs that return result of atomics in big endian notation
 - Establish loopback connections by default if HCA supports atomics
- * Bug Fixes (since 2.2rc2):
 - Fix minor error in use of communicator object in collectives
 - Fix missing `u_int64_t` declaration with PGI compilers
 - Thanks to Adam T. Moody@LLNL for the report and patch
 - Fix memory leak in RMA rendezvous code path
 - Thanks to Min Si@ANL for the report and patch

MVAPICH2 2.2rc2 (08/08/2016)

- * Features and Enhancements (since 2.2rc1):
 - Enhanced performance for `MPI_Comm_split` through new bitonic algorithm
 - Thanks to Adam T. Moody@LLNL for the patch
 - Enable graceful fallback to Shared Memory if LiMIC2 or CMA transfer fails
 - Enable support for multiple MPI initializations
 - Unify process affinity support in Gen2, PSM and PSM2 channels
 - Remove verbs dependency when building the PSM and PSM2 channels
 - Allow processes to request `MPI_THREAD_MULTIPLE` when socket or NUMA node level affinity is specified
 - Point-to-point and collective performance optimization for Intel Knights

Appendix E. MVAPICH2 Release Information

Landing

- Automatic detection and tuning for InfiniBand EDR HCAs
- Warn user to reconfigure library if rank type is not large enough to represent all ranks in job
- Collective tuning for Opal@LLNL, Bridges@PSC, and Stampede-1.5@TACC
- Tuning and architecture detection for Intel Broadwell processors
- Add ability to avoid using --enable-new-dtags with ld
 - Thanks to Adam T. Moody@LLNL for the suggestion
- Add LIBTVMPICH specific CFLAGS and LDFLAGS
 - Thanks to Adam T. Moody@LLNL for the suggestion

* Bug Fixes (since 2.2rc1):

- Disable optimization that removes use of calloc in ptmalloc hook detection code
 - Thanks to Karl W. Schulz@Intel
- Fix weak alias typos (allows successful compilation with CLANG compiler)
 - Thanks to Min Dong@Old Dominion University for the patch
- Fix issues in PSM large message gather operations
 - Thanks to Adam T. Moody@LLNL for the report
- Enhance error checking in collective tuning code
 - Thanks to Jan Bierbaum@Technical University of Dresden for the patch
- Fix issues with UD based communication in RoCE mode
- Fix issues with PMI2 support in singleton mode
- Fix default binding bug in hydra launcher
- Fix issues with Checkpoint Restart when launched with mpirun_rsh
- Fix fortran binding issues with Intel 2016 compilers
- Fix issues with socket/NUMA node level binding
- Disable atomics when using Connect-IB with RDMA_CM
- Fix hang in MPI_Finalize when using hybrid channel
- Fix memory leaks

MVAPICH2 2.2rc1 (03/29/2016)

* Features and Enhancements (since 2.2b):

- Support for OpenPower architecture
 - Optimized inter-node and intra-node communication
- Support for Intel Omni-Path architecture
 - Thanks to Intel for contributing the patch
 - Introduction of a new PSM2 channel for Omni-Path
- Support for RoCEv2
- Architecture detection for PSC Bridges system with Omni-Path
- Enhanced startup performance and reduced memory footprint for storing InfiniBand end-point information with SLURM
 - Support for shared memory based PMI operations
 - Availability of an updated patch from the MVAPICH project website with this support for SLURM installations
- Optimized pt-to-pt and collective tuning for Chameleon InfiniBand systems at TACC/UoC
- Enable affinity by default for TrueScale(PSM) and Omni-Path(PSM2) channels
- Enhanced tuning for shared-memory based MPI_Bcast
- Enhanced debugging support and error messages
- Update to hwloc version 1.11.2

* Bug Fixes (since 2.2b):

- Fix issue in some of the internal algorithms used for MPI_Bcast, MPI_Alltoall and MPI_Reduce
- Fix hang in one of the internal algorithms used for MPI_Scatter
 - Thanks to Ivan Raikov@Stanford for reporting this issue
- Fix issue with rdma_connect operation
- Fix issue with Dynamic Process Management feature
- Fix issue with de-allocating InfiniBand resources in blocking mode
- Fix build errors caused due to improper compile time guards
 - Thanks to Adam Moody@LLNL for the report
- Fix finalize hang when running in hybrid or UD-only mode
 - Thanks to Jerome Vienne@TACC for reporting this issue
- Fix issue in MPI_Win_flush operation
 - Thanks to Nenad Vukicevic for reporting this issue
- Fix out of memory issues with non-blocking collectives code
 - Thanks to Phanisri Pradeep Pratapa and Fang Liu@GaTech for reporting this issue
- Fix fall-through bug in external32 pack
 - Thanks to Adam Moody@LLNL for the report and patch
- Fix issue with on-demand connection establishment and blocking mode
 - Thanks to Maksym Planeta@TU Dresden for the report
- Fix memory leaks in hardware multicast based broadcast code
- Fix memory leaks in TrueScale (PSM) channel
- Fix compilation warnings

MVAPICH2 2.2b (11/12/2015)

* Features and Enhancements (since 2.2a):

- Enhanced performance for small messages
- Enhanced startup performance with SLURM
 - Support for PMIX_Iallgather and PMIX_I fence
- Support to enable affinity with asynchronous progress thread
- Enhanced support for MPIT based performance variables
- Tuned VBUF size for performance
- Improved startup performance for QLogic PSM-CH3 channel
 - Thanks to Maksym Planeta@TU Dresden for the patch

* Bug Fixes (since 2.2a):

- Fix issue with MPI_Get_count in QLogic PSM-CH3 channel with very large messages (>2GB)
- Fix issues with shared memory collectives and checkpoint-restart
- Fix hang with checkpoint-restart
- Fix issue with unlinking shared memory files
- Fix memory leak with MPIT
- Fix minor typos and usage of inline and static keywords
 - Thanks to Maksym Planeta@TU Dresden for the patch and suggestions
- Fix missing MPIDI_FUNC_EXIT
 - Thanks to Maksym Planeta@TU Dresden for the patch
- Remove unused code
 - Thanks to Maksym Planeta@TU Dresden for the patch
- Continue with warning if user asks to enable XRC when the system does not support XRC

MVAPICH2 2.2a (08/17/2015)

* Features and Enhancements (since 2.1 GA):

Appendix E. MVAPICH2 Release Information

- Based on MPICH 3.1.4
- Support for backing on-demand UD CM information with shared memory for minimizing memory footprint
- Reorganized HCA-aware process mapping
- Dynamic identification of maximum read/atomic operations supported by HCA
- Enabling support for intra-node communications in RoCE mode without shared memory
- Updated to hwloc 1.11.0
- Updated to sm_20 kernel optimizations for MPI Datatypes
- Automatic detection and tuning for 24-core Haswell architecture

* Bug Fixes (since 2.1 GA):

- Fix for error with multi-vbuf design for GPU based communication
- Fix bugs with hybrid UD/RC/XRC communications
- Fix for MPICH putfence/getfence for large messages
- Fix for error in collective tuning framework
- Fix validation failure with Alltoall with IN_PLACE option
 - Thanks for Mahidhar Tatineni @SDSC for the report
- Fix bug with MPI_Reduce with IN_PLACE option
 - Thanks to Markus Geimer for the report
- Fix for compilation failures with multicast disabled
 - Thanks to Devesh Sharma @Emulex for the report
- Fix bug with MPI_Bcast
- Fix IPC selection for shared GPU mode systems
- Fix for build time warnings and memory leaks
- Fix issues with Dynamic Process Management
 - Thanks to Neil Spruit for the report
- Fix bug in architecture detection code
 - Thanks to Adam Moody @LLNL for the report

MVAPICH2-2.1 (04/03/2015)

* Features and Enhancements (since 2.1rc2):

- Tuning for EDR adapters
- Optimization of collectives for SDSC Comet system

* Bug-Fixes (since 2.1rc2):

- Relocate reading environment variables in PSM
 - Thanks to Adam Moody@LLNL for the suggestion
- Fix issue with automatic process mapping
- Fix issue with checkpoint restart when full path is not given
- Fix issue with Dynamic Process Management
- Fix issue in CUDA IPC code path
- Fix corner case in CMA runtime detection

MVAPICH2-2.1rc2 (03/12/2015)

* Features and Enhancements (since 2.1rc1):

- Based on MPICH-3.1.4
- Enhanced startup performance with mpirun_rsh
- Checkpoint-Restart Support with DMTCP (Distributed MultiThreaded CheckPointing)
 - Thanks to the DMTCP project team (<http://dmtcp.sourceforge.net/>)

- Support for handling very large messages in RMA
- Optimize size of buffer requested for control messages in large message transfer
- Enhanced automatic detection of atomic support
- Optimized collectives (bcast, reduce, and allreduce) for 4K processes
- Introduce support to sleep for user specified period before aborting
 - Thanks to Adam Moody@LLNL for the suggestion
- Disable PSM from setting CPU affinity
 - Thanks to Adam Moody@LLNL for providing the patch
- Install PSM error handler to print more verbose error messages
 - Thanks to Adam Moody@LLNL for providing the patch
- Introduce retry mechanism to perform psm_ep_open in PSM channel
 - Thanks to Adam Moody@LLNL for providing the patch

* Bug-Fixes (since 2.1rc1):

- Fix failures with shared memory collectives with checkpoint-restart
- Fix failures with checkpoint-restart when using internal communication buffers of different size
- Fix undeclared variable error when --disable-cxx is specified with configure
 - Thanks to Chris Green from FANL for the patch
- Fix segfault seen during connect/accept with dynamic processes
 - Thanks to Neil Spruit for the fix
- Fix errors with large messages pack/unpack operations in PSM channel
- Fix for bcast collective tuning
- Fix assertion errors in one-sided put operations in PSM channel
- Fix issue with code getting stuck in infinite loop inside ptmalloc
 - Thanks to Adam Moody@LLNL for the suggested changes
- Fix assertion error in shared memory large message transfers
 - Thanks to Adam Moody@LLNL for reporting the issue
- Fix compilation warnings

MVAPICH2-2.1rc1 (12/18/2014)

* Features and Enhancements (since 2.1a):

- Based on MPICH-3.1.3
- Flexibility to use internal communication buffers of different size for improved performance and memory footprint
- Improve communication performance by removing locks from critical path
- Enhanced communication performance for small/medium message sizes
- Support for linking Intel Trace Analyzer and Collector
- Increase the number of connect retry attempts with RDMA_CM
- Automatic detection and tuning for Haswell architecture

* Bug-Fixes (since 2.1a):

- Fix automatic detection of support for atomics
- Fix issue with void pointer arithmetic with PGI
- Fix deadlock in ctxidup MPICH test in PSM channel
- Fix compile warnings

MVAPICH2-2.1a (09/21/2014)

* Features and Enhancements (since 2.0):

- Based on MPICH-3.1.2
- Support for PMI-2 based startup with SLURM

Appendix E. MVAPICH2 Release Information

- Enhanced startup performance for Gen2/UD-Hybrid channel
- GPU support for MPI_Scan and MPI_Exscan collective operations
- Optimize creation of 2-level communicator
- Collective optimization for PSM-CH3 channel
- Tuning for IvyBridge architecture
- Add `-export-all` option to `mpirun_rsh`
- Support for additional MPI-T performance variables (PVARs) in the CH3 channel
- Link with `libstdc++` when building with GPU support (required by CUDA 6.5)

* Bug-Fixes (since 2.0):

- Fix error in large message (>2GB) transfers in CMA code path
- Fix memory leaks in OFA-IB-CH3 and OFA-IB-Nemesis channels
- Fix issues with optimizations for broadcast and reduce collectives
- Fix hang at finalize with Gen2-Hybrid/UD channel
- Fix issues for collectives with non power-of-two process counts
 - Thanks to Evren Yurtesen for identifying the issue
- Make ring startup use HCA selected by user
- Increase counter length for shared-memory collectives

MVAPICH2-2.0 (06/20/2014)

* Features and Enhancements (since 2.0rc2):

- Consider CMA in collective tuning framework

* Bug-Fixes (since 2.0rc2):

- Fix bug when disabling registration cache
- Fix shared memory window bug when shared memory collectives are disabled
- Fix `mpirun_rsh` bug when running `mpmd` programs with no arguments

MVAPICH2-2.0rc2 (05/25/2014)

* Features and Enhancements (since 2.0rc1):

- CMA support is now enabled by default
- Optimization of collectives with CMA support
- RMA optimizations for shared memory and atomic operations
- Tuning RGET and Atomics operations
- Tuning RDMA FP-based communication
- MPI-T support for additional performance and control variables
- The `--enable-mpit-pvars=yes` configuration option will now enable only MVAPICH2 specific variables
- Large message transfer support for PSM interface
- Optimization of collectives for PSM interface
- Updated to `hwloc v1.9`

* Bug-Fixes (since 2.0rc1):

- Fix multicast hang when there is a single process on one node and more than one process on other nodes
- Fix non-power-of-two usage of scatter-doubling-allgather algorithm
- Fix for `bcastzero` type hang during finalize
- Enhanced handling of failures in RDMA_CM based connection establishment
- Fix for a hang in finalize when using RDMA_CM
- Finish receive request when RDMA READ completes in RGET protocol

- Always use direct RDMA when flush is used
- Fix compilation error with --enable-g=all in PSM interface
- Fix warnings and memory leaks

MVAPICH2-2.0rc1 (03/24/2014)

* Features and Enhancements (since 2.0b):

- Based on MPICH-3.1
- Enhanced direct RDMA based designs for MPI_Put and MPI_Get operations in OFA-IB-CH3 channel
- Optimized communication when using MPI_Win_allocate for OFA-IB-CH3 channel
- MPI-3 RMA support for CH3-PSM channel
- Multi-rail support for UD-Hybrid channel
- Optimized and tuned blocking and non-blocking collectives for OFA-IB-CH3, OFA-IB-Nemesis, and CH3-PSM channels
- Improved hierarchical job startup performance
- Optimized sub-array data-type processing for GPU-to-GPU communication
- Tuning for Mellanox Connect-IB adapters
- Updated hwloc to version 1.8
- Added options to specify CUDA library paths
- Deprecation of uDAPL-CH3 channel

* Bug-Fixes (since 2.0b):

- Fix issues related to MPI-3 RMA locks
- Fix an issue related to MPI-3 dynamic window
- Fix issues related to MPI_Win_allocate backed by shared memory
- Fix issues related to large message transfers for OFA-IB-CH3 and OFA-IB-Nemesis channels
- Fix warning in job launch, when using DPM
- Fix an issue related to MPI atomic operations on HCAs without atomics support
- Fixed an issue related to selection of compiler. (We prefer the GNU, Intel, PGI, and Ekopath compilers in that order).
 - Thanks to Uday R Bondhugula from IISc for the report
- Fix an issue in message coalescing
- Prevent printing out inter-node runtime parameters for pure intra-node runs
 - Thanks to Jerome Vienne from TACC for the report
- Fix an issue related to ordering of messages for GPU-to-GPU transfers
- Fix a few memory leaks and warnings

MVAPICH2-2.0b (11/08/2013)

* Features and Enhancements (since 2.0a):

- Based on MPICH-3.1b1
- Multi-rail support for GPU communication
- Non-blocking streams in asynchronous CUDA transfers for better overlap
- Initialize GPU resources only when used by MPI transfer
- Extended support for MPI-3 RMA in OFA-IB-CH3, OFA-IWARP-CH3, and OFA-RoCE-CH3
- Additional MPIT counters and performance variables
- Updated compiler wrappers to remove application dependency on network and other extra libraries
 - Thanks to Adam Moody from LLNL for the suggestion

Appendix E. MVAPICH2 Release Information

- Capability to checkpoint CH3 channel using the Hydra process manager
- Optimized support for broadcast, reduce and other collectives
- Tuning for IvyBridge architecture
- Improved launch time for large-scale mpirun_rsh jobs
- Introduced retry mechanism in mpirun_rsh for socket binding
- Updated hwloc to version 1.7.2

* Bug-Fixes (since 2.0a):

- Consider list provided by MV2_IBA_HCA when scanning device list
- Fix issues in Nemesis interface with --with-ch3-rank-bits=32
- Better cleanup of XRC files in corner cases
- Initialize using better defaults for ibv_modify_qp (initial ring)
- Add unconditional check and addition of pthread library
- MPI_Get_library_version updated with proper MVAPICH2 branding
 - Thanks to Jerome Vienne from the TACC for the report

MVAPICH2-2.0a (08/24/2013)

* Features and Enhancements (since 1.9):

- Based on MPICH-3.0.4
- Dynamic CUDA initialization. Support GPU device selection after MPI_Init
- Support for running on heterogeneous clusters with GPU and non-GPU nodes
- Supporting MPI-3 RMA atomic operations and flush operations with CH3-Gen2 interface
- Exposing internal performance variables to MPI-3 Tools information interface (MPIT)
- Enhanced MPI_Bcast performance
- Enhanced performance for large message MPI_Scatter and MPI_Gather
- Enhanced intra-node SMP performance
- Tuned SMP eager threshold parameters
- Reduced memory footprint
- Improved job-startup performance
- Warn and continue when ptmalloc fails to initialize
- Enable hierarchical SSH-based startup with Checkpoint-Restart
- Enable the use of Hydra launcher with Checkpoint-Restart

* Bug-Fixes (since 1.9):

- Fix data validation issue with MPI_Bcast
 - Thanks to Claudio J. Margulis from University of Iowa for the report
- Fix buffer alignment for large message shared memory transfers
- Fix a bug in One-Sided shared memory backed windows
- Fix a flow-control bug in UD transport
 - Thanks to Benjamin M. Auer from NASA for the report
- Fix bugs with MPI-3 RMA in Nemesis IB interface
- Fix issue with very large message (>2GB bytes) MPI_Bcast
 - Thanks to Lu Qiyue for the report
- Handle case where \$HOME is not set during search for MV2 user config file
 - Thanks to Adam Moody from LLNL for the patch
- Fix a hang in connection setup with RDMA-CM

* Features and Enhancements (since 1.9rc1):

- Updated to hwloc v1.7
- Tuned Reduce, AllReduce, Scatter, Reduce-Scatter and Allgatherv Collectives

* Bug-Fixes (since 1.9rc1):

- Fix cuda context issue with async progress thread
 - Thanks to Osuna Escamilla Carlos from env.ethz.ch for the report
- Overwrite pre-existing PSM environment variables
 - Thanks to Adam Moody from LLNL for the patch
- Fix several warnings
 - Thanks to Adam Moody from LLNL for some of the patches

MVAPICH2-1.9RC1 (04/16/2013)

* Features and Enhancements (since 1.9b):

- Based on MPICH-3.0.3
- Updated SCR to version 1.1.8
- Install utility scripts included with SCR
- Support for automatic detection of path to utilities used by mpirun_rsh during configuration
 - Utilities supported: rsh, ssh, xterm, totalview
- Support for launching jobs on heterogeneous networks with mpirun_rsh
- Tuned Bcast, Reduce, Scatter Collectives
- Tuned MPI performance on Kepler GPUs
- Introduced MV2_RDMA_CM_CONF_FILE_PATH parameter which specifies path to mv2.conf

* Bug-Fixes (since 1.9b):

- Fix autoconf issue with LiMIC2 source-code
 - Thanks to Doug Johnson from OH-TECH for the report
- Fix build errors with --enable-thread-cs=per-object and --enable-refcount=lock-free
 - Thanks to Marcin Zalewski from Indiana University for the report
- Fix MPI_Scatter failure with MPI_IN_PLACE
 - Thanks to Mellanox for the report
- Fix MPI_Scatter failure with cyclic host files
- Fix deadlocks in PSM interface for multi-threaded jobs
 - Thanks to Marcin Zalewski from Indiana University for the report
- Fix MPI_Bcast failures in SCALAPACK
 - Thanks to Jerome Vienne from TACC for the report
- Fix build errors with newer Ekopath compiler
- Fix a bug with shmem collectives in PSM interface
- Fix memory corruption when more entries specified in mv2.conf than the requested number of rails
 - Thanks to Akihiro Nomura from Tokyo Institute of Technology for the report
- Fix memory corruption with CR configuration in Nemesis interface

MVAPICH2-1.9b (02/28/2013)

* Features and Enhancements (since 1.9a2):

- Based on MPICH-3.0.2
 - Support for all MPI-3 features
- Support for single copy intra-node communication using Linux supported CMA (Cross Memory Attach)
 - Provides flexibility for intra-node communication: shared memory, LiMIC2, and CMA
- Checkpoint/Restart using LLNL's Scalable Checkpoint/Restart Library (SCR)
 - Support for application-level checkpointing

Appendix E. MVAPICH2 Release Information

- Support for hierarchical system-level checkpointing
- Improved job startup time
 - Provided a new runtime variable MV2_HOMOGENEOUS_CLUSTER for optimized startup on homogeneous clusters
- New version of LiMIC2 (v0.5.6)
 - Provides support for unlocked ioctl calls
- Tuned Reduce, Allgather, Reduce_Scatter, Allgatherv collectives
- Introduced option to export environment variables automatically with mpirun_rsh
- Updated to HWLOC v1.6.1
- Provided option to use CUDA library call instead of CUDA driver to check buffer pointer type
 - Thanks to Christian Robert from Sandia for the suggestion
- Improved debug messages and error reporting

* Bug-Fixes (since 1.9a2):

- Fix page fault with memory access violation with LiMIC2 exposed by newer Linux kernels
 - Thanks to Karl Schulz from TACC for the report
- Fix a failure when lazy memory registration is disabled and CUDA is enabled
 - Thanks to Jens Glaser from University of Minnesota for the report
- Fix an issue with variable initialization related to DPM support
- Rename a few internal variables to avoid name conflicts with external applications
 - Thanks to Adam Moody from LLNL for the report
- Check for libattr during configuration when Checkpoint/Restart and Process Migration are requested
 - Thanks to John Gilmore from Vastech for the report
- Fix build issue with --disable-cxx
- Set intra-node eager threshold correctly when configured with LiMIC2
- Fix an issue with MV2_DEFAULT_PKEY in partitioned InfiniBand network
 - Thanks to Jesper Larsen from FCOO for the report
- Improve makefile rules to use automake macros
 - Thanks to Carmelo Ponti from CSCS for the report
- Fix configure error with automake conditionals
 - Thanks to Evren Yurtesen from Abo Akademi for the report
- Fix a few memory leaks and warnings
- Properly cleanup shared memory files (used by XRC) when applications fail

MVAPICH2-1.9a2 (11/08/2012)

* Features and Enhancements (since 1.9a):

- Based on MPICH2-1.5
- Initial support for MPI-3:
(Available for all interfaces: OFA-IB-CH3, OFA-IWARP-CH3, OFA-RoCE-CH3, uDAPL-CH3, OFA-IB-Nemesis, PSM-CH3)
 - Nonblocking collective functions available as "MPIX_" functions (e.g., "MPIX_Ibcast")
 - Neighborhood collective routines available as "MPIX_" functions (e.g., "MPIX_Neighbor_allgather")
 - MPI_Comm_split_type function available as an "MPIX_" function
 - Support for MPIX_Type_create_hindexed_block
 - Nonblocking communicator duplication routine MPIX_Comm_idup (will only work for single-threaded programs)

- MPIX_Comm_create_group support
- Support for matched probe functionality (e.g., MPIX_Mprobe, MPIX_Improbe, MPIX_Mrecv, and MPIX_Imrecv), (Not Available for PSM)
- Support for "Const" (disabled by default)
- Efficient vector, hindexed datatype processing on GPU buffers
- Tuned alltoall, Scatter and Allreduce collectives
- Support for Mellanox Connect-IB HCA
- Adaptive number of registration cache entries based on job size
- Revamped Build system:
 - Uses automake instead of simplemake,
 - Allows for parallel builds ("make -j8" and similar)

* Bug-Fixes (since 1.9a):

- CPU frequency mismatch warning shown under debug
- Fix issue with MPI_IN_PLACE buffers with CUDA
- Fix ptmalloc initialization issue due to compiler optimization
 - Thanks to Kyle Sheumaker from ACT for the report
- Adjustable MAX_NUM_PORTS at build time to support more than two ports
- Fix issue with MPI_Allreduce with MPI_IN_PLACE send buffer
- Fix memleak in MPI_Cancel with PSM interface
 - Thanks to Andrew Friedley from LLNL for the report

MVAPICH2-1.9a (09/07/2012)

* Features and Enhancements (since 1.8):

- Support for InfiniBand hardware UD-multicast
- UD-multicast-based designs for collectives (Bcast, Allreduce and Scatter)
- Enhanced Bcast and Reduce collectives with pt-to-pt communication
- LiMIC-based design for Gather collective
- Improved performance for shared-memory-aware collectives
- Improved intra-node communication performance with GPU buffers using pipelined design
- Improved inter-node communication performance with GPU buffers with non-blocking CUDA copies
- Improved small message communication performance with GPU buffers using CUDA IPC design
- Improved automatic GPU device selection and CUDA context management
- Optimal communication channel selection for different GPU communication modes (DD, DH and HD) in different configurations (intra-IOH and inter-IOH)
- Removed libibumad dependency for building the library
- Option for selecting non-default gid-index in a loss-less fabric setup in RoCE mode
- Option to disable signal handler setup
- Tuned thresholds for various architectures
- Set DAPL-2.0 as the default version for the uDAPL interface
- Updated to hwloc v1.5
- Option to use IP address as a fallback if hostname cannot be resolved
- Improved error reporting

* Bug-Fixes (since 1.8):

- Fix issue in intra-node knomial bcast

Appendix E. MVAPICH2 Release Information

- Handle gethostbyname return values gracefully
- Fix corner case issue in two-level gather code path
- Fix bug in CUDA events/streams pool management
- Fix ptmalloc initialization issue when MALLOC_CHECK_ is defined in the environment
 - Thanks to Mehmet Belgin from Georgia Institute of Technology for the report
- Fix memory corruption and handle heterogeneous architectures in gather collective
- Fix issue in detecting the correct HCA type
- Fix issue in ring start-up to select correct HCA when MV2_IBA_HCA is specified
- Fix SEGFAULT in MPI_Finalize when IB loop-back is used
- Fix memory corruption on nodes with 64-cores
 - Thanks to M Xie for the report
- Fix hang in MPI_Finalize with Nemesis interface when ptmalloc initialization fails
 - Thanks to Carson Holt from OICR for the report
- Fix memory corruption in shared memory communication
 - Thanks to Craig Tierney from NOAA for the report and testing the patch
- Fix issue in IB ring start-up selection with mpiexec.hydra
- Fix issue in selecting CUDA run-time variables when running on single node in SMP only mode
- Fix few memory leaks and warnings

MVAPICH2-1.8 (04/30/2012)

* Features and Enhancements (since 1.8rc1):

- Introduced a unified run time parameter MV2_USE_ONLY_UD to enable UD only mode
- Enhanced designs for Alltoall and Allgather collective communication from GPU device buffers
- Tuned collective communication from GPU device buffers
- Tuned Gather collective
- Introduced a run time parameter MV2_SHOW_CPU_BINDING to show current CPU bindings
- Updated to hwloc v1.4.1
- Remove dependency on LEX and YACC

* Bug-Fixes (since 1.8rc1):

- Fix hang with multiple GPU configuration
 - Thanks to Jens Glaser from University of Minnesota for the report
- Fix buffer alignment issues to improve intra-node performance
- Fix a DPM multispawn behavior
- Enhanced error reporting in DPM functionality
- Quote environment variables in job startup to protect from shell
- Fix hang when LIMIC is enabled
- Fix hang in environments with heterogeneous HCAs
- Fix issue when using multiple HCA ports in RDMA_CM mode
 - Thanks to Steve Wise from Open Grid Computing for the report
- Fix hang during MPI_Finalize in Nemesis IB netmod
- Fix for a start-up issue in Nemesis with heterogeneous architectures
- Fix few memory leaks and warnings

MVAPICH2-1.8rc1 (03/22/2012)

* Features & Enhancements (since 1.8a2):

- New design for intra-node communication from GPU Device buffers using CUDA IPC for better performance and correctness
 - Thanks to Joel Scherpelz from NVIDIA for his suggestions
- Enabled shared memory communication for host transfers when CUDA is enabled
- Optimized and tuned collectives for GPU device buffers
- Enhanced pipelined inter-node device transfers
- Enhanced shared memory design for GPU device transfers for large messages
- Enhanced support for CPU binding with socket and numanode level granularity
- Support suspend/resume functionality with mpirun_rsh
- Exporting local rank, local size, global rank and global size through environment variables (both mpirun_rsh and hydra)
- Update to hwloc v1.4
- Checkpoint-Restart support in OFA-IB-Nemesis interface
- Enabling run-through stabilization support to handle process failures in OFA-IB-Nemesis interface
- Enhancing OFA-IB-Nemesis interface to handle IB errors gracefully
- Performance tuning on various architecture clusters
- Support for Mellanox IB FDR adapter

* Bug-Fixes (since 1.8a2):

- Fix a hang issue on InfiniHost SDR/DDR cards
 - Thanks to Nirmal Seenu from Fermilab for the report
- Fix an issue with runtime parameter MV2_USE_COALESCE usage
- Fix an issue with LiMIC2 when CUDA is enabled
- Fix an issue with intra-node communication using datatypes and GPU device buffers
- Fix an issue with Dynamic Process Management when launching processes on multiple nodes
 - Thanks to Rutger Hofman from VU Amsterdam for the report
- Fix build issue in hwloc source with mcmmodel=medium flags
 - Thanks to Nirmal Seenu from Fermilab for the report
- Fix a build issue in hwloc with --disable-shared or --disabled-static options
- Use portable stdout and stderr redirection
 - Thanks to Dr. Axel Philipp from *MTU* Aero Engines for the patch
- Fix a build issue with PGI 12.2
 - Thanks to Thomas Rothrock from U.S. Army SMDC for the patch
- Fix an issue with send message queue in OFA-IB-Nemesis interface
- Fix a process cleanup issue in Hydra when MPI_ABORT is called (upstream MPICH2 patch)
- Fix an issue with non-contiguous datatypes in MPI_Gather
- Fix a few memory leaks and warnings

MVAPICH2-1.8a2 (02/02/2012)

* Features and Enhancements (since 1.8a1p1):

- Support for collective communication from GPU buffers
- Non-contiguous datatype support in point-to-point and collective communication from GPU buffers

Appendix E. MVAPICH2 Release Information

- Efficient GPU-GPU transfers within a node using CUDA IPC (for CUDA 4.1)
 - Alternate synchronization mechanism using CUDA Events for pipelined device data transfers
 - Exporting processes local rank in a node through environment variable
 - Adjust shared-memory communication block size at runtime
 - Enable XRC by default at configure time
 - New shared memory design for enhanced intra-node small message performance
 - Tuned inter-node and intra-node performance on different cluster architectures
 - Update to hwloc v1.3.1
 - Support for fallback to R3 rendezvous protocol if RGET fails
 - SLURM integration with mpiexec.mpirun_rsh to use SLURM allocated hosts without specifying a hostfile
 - Support added to automatically use PBS_NODEFILE in Torque and PBS environments
 - Enable signal-triggered (SIGUSR2) migration
- * Bug Fixes (since 1.8alp1):
- Set process affinity independently of SMP enable/disable to control the affinity in loopback mode
 - Report error and exit if user requests MV2_USE_CUDA=1 in non-cuda configuration
 - Fix for data validation error with GPU buffers
 - Updated WRAPPER_CPPFLAGS when using --with-cuda. Users should not have to explicitly specify CPPFLAGS or LDFLAGS to build applications
 - Fix for several compilation warnings
 - Report an error message if user requests MV2_USE_XRC=1 in non-XRC configuration
 - Remove debug prints in regular code path with MV2_USE_BLOCKING=1
 - Thanks to Vaibhav Dutt for the report
 - Handling shared memory collective buffers in a dynamic manner to eliminate static setting of maximum CPU core count
 - Fix for validation issue in MPICH2 strided_get_indexed.c
 - Fix a bug in packetized transfers on heterogeneous clusters
 - Fix for deadlock between psm_ep_connect and PMGR_COLLECTIVE calls on QLogic systems
 - Thanks to Adam T. Moody for the patch
 - Fix a bug in MPI_Allocate_mem when it is called with size 0
 - Thanks to Michele De Stefano for reporting this issue
 - Create vendor for Open64 compilers and add rpath for unknown compilers
 - Thanks to Martin Hilgemen from Dell Inc. for the initial patch
 - Fix issue due to overlapping buffers with sprintf
 - Thanks to Mark Debbage from QLogic for reporting this issue
 - Fallback to using GNU options for unknown f90 compilers
 - Fix hang in PMI_Barrier due to incorrect handling of the socket return values in mpirun_rsh
 - Unify the redundant FTB events used to initiate a migration
 - Fix memory leaks when mpirun_rsh reads hostfiles
 - Fix a bug where library attempts to use in-active rail in multi-rail scenario

MVAPICH2-1.8alp1 (11/14/2011)

- * Bug Fixes (since 1.8al)
- Fix for a data validation issue in GPU transfers

- Thanks to Massimiliano Fatica, NVIDIA, for reporting this issue
- Tuned CUDA block size to 256K for better performance
- Enhanced error checking for CUDA library calls
- Fix for mpirun_rsh issue while launching applications on Linux Kernels (3.x)

MVAPICH2-1.8a1 (11/09/2011)

* Features and Enhancements (since 1.7):

- Support for MPI communication from NVIDIA GPU device memory
 - High performance RDMA-based inter-node point-to-point communication (GPU-GPU, GPU-Host and Host-GPU)
 - High performance intra-node point-to-point communication for multi-GPU adapters/node (GPU-GPU, GPU-Host and Host-GPU)
 - Communication with contiguous datatype
- Reduced memory footprint of the library
- Enhanced one-sided communication design with reduced memory requirement
- Enhancements and tuned collectives (Bcast and Alltoally)
- Update to hwloc v1.3.0
- Flexible HCA selection with Nemesis interface
 - Thanks to Grigori Inozemtsev, Queens University
- Support iWARP interoperability between Intel NE020 and Chelsio T4 Adapters
- RoCE enable environment variable name is changed from MV2_USE_RDMAOE to MV2_USE_RoCE

* Bug Fixes (since 1.7):

- Fix for a bug in mpirun_rsh while doing process clean-up in abort and other error scenarios
- Fixes for code compilation warnings
- Fix for memory leaks in RDMA CM code path

MVAPICH2-1.7 (10/14/2011)

* Features and Enhancements (since 1.7rc2):

- Support SHMEM collectives upto 64 cores/node
- Update to hwloc v1.2.2
- Enhancement and tuned collective (GatherV)

* Bug Fixes:

- Fixes for code compilation warnings
- Fix job clean-up issues with mpirun_rsh
- Fix a hang with RDMA CM

MVAPICH2-1.7rc2 (09/19/2011)

* Features and Enhancements (since 1.7rc1):

- Based on MPICH2-1.4.1p1
- Integrated Hybrid (UD-RC/XRC) design to get best performance on large-scale systems with reduced/constant memory footprint
- Shared memory backed Windows for One-Sided Communication
- Support for truly passive locking for intra-node RMA in shared memory and LIMIC based windows
- Integrated with Portable Hardware Locality (hwloc v1.2.1)
- Integrated with latest OSU Micro-Benchmarks (3.4)
- Enhancements and tuned collectives (Allreduce and Allgatherv)

Appendix E. MVAPICH2 Release Information

- MPI_THREAD_SINGLE provided by default and MPI_THREAD_MULTIPLE as an option
- Enabling Checkpoint/Restart support in pure SMP mode
- Optimization for QDR cards
- On-demand connection management support with IB CM (RoCE interface)
- Optimization to limit number of RDMA Fast Path connections for very large clusters (Nemesis interface)
- Multi-core-aware collective support (QLogic PSM interface)

* Bug Fixes:

- Fixes for code compilation warnings
- Compiler preference lists reordered to avoid mixing GCC and Intel compilers if both are found by configure
- Fix a bug in transferring very large messages (>2GB)
 - Thanks to Tibor Pausz from Univ. of Frankfurt for reporting it
- Fix a hang with One-Sided Put operation
- Fix a bug in ptmalloc integration
- Avoid double-free crash with mpispawn
- Avoid crash and print an error message in mpirun_rsh when the hostfile is empty
- Checking for error codes in PMI design
- Verify programs can link with LiMIC2 at runtime
- Fix for compilation issue when BLCR or FTB installed in non-system paths
- Fix an issue with RDMA-Migration
- Fix for memory leaks
- Fix an issue in supporting RoCE with second port on available on HCA
 - Thanks to Jeffrey Konz from HP for reporting it
- Fix for a hang with passive RMA tests (QLogic PSM interface)

MVAPICH2-1.7rc1 (07/20/2011)

* Features and Enhancements (since 1.7a2)

- Based on MPICH2-1.4
- CH3 shared memory channel for standalone hosts (including laptops) without any InfiniBand adapters
- HugePage support
- Improved on-demand InfiniBand connection setup
- Optimized Fence synchronization (with and without LIMIC2 support)
- Enhanced mpirun_rsh design to avoid race conditions and support for improved debug messages
- Optimized design for collectives (Bcast and Reduce)
- Improved performance for medium size messages for QLogic PSM
- Support for Ekopath Compiler

* Bug Fixes

- Fixes in Dynamic Process Management (DPM) support
- Fixes in Checkpoint/Restart and Migration support
- Fix Restart when using automatic checkpoint
 - Thanks to Alexandr for reporting this
- Compilation warnings fixes
- Handling very large one-sided transfers using RDMA
- Fixes for memory leaks
- Graceful handling of unknown HCAs
- Better handling of shmem file creation errors
- Fix for a hang in intra-node transfer

- Fix for a build error with --disable-weak-symbols
 - Thanks to Peter Willis for reporting this issue
- Fixes for one-sided communication with passive target synchronization
- Proper error reporting when a program is linked with both static and shared MVAPICH2 libraries

MVAPICH2-1.7a2 (06/03/2011)

* Features and Enhancements (Since 1.7a)

- Improved intra-node shared memory communication performance
- Tuned RDMA Fast Path Buffer size to get better performance with less memory footprint (CH3 and Nemesis)
- Fast process migration using RDMA
- Automatic inter-node communication parameter tuning based on platform and adapter detection (Nemesis)
- Automatic intra-node communication parameter tuning based on platform
- Efficient connection set-up for multi-core systems
- Enhancements for collectives (barrier, gather and allgather)
- Compact and shorthand way to specify blocks of processes on the same host with mpirun_rsh
- Support for latest stable version of HWLOC v1.2
- Improved debug message output in process management and fault tolerance functionality
- Better handling of process signals and error management in mpispawn
- Performance tuning for pt-to-pt and several collective operations

* Bug fixes

- Fixes for memory leaks
- Fixes in CR/migration
- Better handling of memory allocation and registration failures
- Fixes for compilation warnings
- Fix a bug that disallows '=' from mpirun_rsh arguments
- Handling of non-contiguous transfer in Nemesis interface
- Bug fix in gather collective when ranks are in cyclic order
- Fix for the ignore_locks bug in MPI-IO with Lustre

MVAPICH2-1.7a (04/19/2011)

* Features and Enhancements

- Based on MPICH2-1.3.2p1
- Integrated with Portable Hardware Locality (hwloc v1.1.1)
- Supporting Large Data transfers (>2GB)
- Integrated with Enhanced LiMIC2 (v0.5.5) to support Intra-node large message (>2GB) transfers
- Optimized and tuned algorithm for AlltoAll
- Enhanced debugging config options to generate core files and back-traces
- Support for Chelsio's T4 Adapter

MVAPICH2-1.6 (03/09/2011)

* Features and Enhancements (since 1.6-RC3)

- Improved configure help for MVAPICH2 features

Appendix E. MVAPICH2 Release Information

- Updated Hydra launcher with MPICH2-1.3.3 Hydra process manager
- Building and installation of OSU micro benchmarks during default MVAPICH2 installation
- Hydra is the default mpiexec process manager

* Bug fixes (since 1.6-RC3)

- Fix hang issues in RMA
- Fix memory leaks
- Fix in RDMA_FP

MVAPICH2-1.6-RC3 (02/15/2011)

* Features and Enhancements

- Support for 3D torus topology with appropriate SL settings
 - For both CH3 and Nemesis interfaces
- Thanks to Jim Schutt, Marcus Epperson and John Nagle from Sandia for the initial patch
- Quality of Service (QoS) support with multiple InfiniBand SL
 - For both CH3 and Nemesis interfaces
- Configuration file support (similar to the one available in MVAPICH). Provides a convenient method for handling all runtime variables through a configuration file.
- Improved job-startup performance on large-scale systems
- Optimization in MPI_Finalize
- Improved pt-to-pt communication performance for small and medium messages
- Optimized and tuned algorithms for Gather and Scatter collective operations
- Optimized thresholds for one-sided RMA operations
- User-friendly configuration options to enable/disable various checkpoint/restart and migration features
- Enabled ROMIO's auto detection scheme for filetypes on Lustre file system
- Improved error checking for system and BLCR calls in checkpoint-restart and migration codepath
- Enhanced OSU Micro-benchmarks suite (version 3.3)

Bug Fixes

- Fix in aggregate ADIO alignment
- Fix for an issue with LiMIC2 header
- XRC connection management
- Fixes in registration cache
- IB card detection with MV2_IBA_HCA runtime option in multi rail design
- Fix for a bug in multi-rail design while opening multiple HCAs
- Fixes for multiple memory leaks
- Fix for a bug in mpirun_rsh
- Checks before enabling aggregation and migration
- Fixing the build errors with --disable-cxx
- Thanks to Bright Yang for reporting this issue
- Fixing the build errors related to "pthread_spinlock_t" seen on RHEL systems

MVAPICH2-1.6-RC2 (12/22/2010)

* Features and Enhancements

- Optimization and enhanced performance for clusters with nVIDIA GPU adapters (with and without GPUDirect technology)
- Enhanced R3 rendezvous protocol
 - For both CH3 and Nemesis interfaces
- Robust RDMA Fast Path setup to avoid memory allocation failures
 - For both CH3 and Nemesis interfaces
- Multiple design enhancements for better performance of medium sized messages
- Enhancements and optimizations for one sided Put and Get operations
- Enhancements and tuning of Allgather for small and medium sized messages
- Optimization of AllReduce
- Enhancements to Multi-rail Design and features including striping of one-sided messages
- Enhancements to mpirun_rsh job start-up scheme
- Enhanced designs for automatic detection of various architectures and adapters

* Bug fixes

- Fix a bug in Post-Wait/Start-Complete path for one-sided operations
- Resolving a hang in mpirun_rsh termination when CR is enabled
- Fixing issue in MPI_Allreduce and Reduce when called with MPI_IN_PLACE
 - Thanks to the initial patch by Alexander Alekhin
- Fix for an issue in rail selection for small RMA messages
- Fix for threading related errors with comm_dup
- Fix for alignment issues in RDMA Fast Path
- Fix for extra memcopy in header caching
- Fix for an issue to use correct HCA when process to rail binding scheme used in combination with XRC.
- Fix for an RMA issue when configured with enable-g=meminit
 - Thanks to James Dinan of Argonne for reporting this issue
- Only set FC and F77 if gfortran is executable

MVAPICH2-1.6RC1 (11/12/2010)

* Features and Enhancements

- Using LiMIC2 for efficient intra-node RMA transfer to avoid extra memory copies
- Upgraded to LiMIC2 version 0.5.4
- Removing the limitation on number of concurrent windows in RMA operations
- Support for InfiniBand Quality of Service (QoS) with multiple lanes
- Enhanced support for multi-threaded applications
- Fast Checkpoint-Restart support with aggregation scheme
- Job Pause-Migration-Restart Framework for Pro-active Fault-Tolerance
- Support for new standardized Fault Tolerant Backplane (FTB) Events for Checkpoint-Restart and Job Pause-Migration-Restart Framework
- Dynamic detection of multiple InfiniBand adapters and using these by default in multi-rail configurations (OLA-IB-CH3, OFA-iWARP-CH3 and OFA-RoCE-CH3 interfaces)
- Support for process-to-rail binding policy (bunch, scatter and

Appendix E. MVAPICH2 Release Information

- user-defined) in multi-rail configurations (OFA-IB-CH3, OFA-iWARP-CH3 and OFA-RoCE-CH3 interfaces)
- Enhanced and optimized algorithms for MPI_Reduce and MPI_AllReduce operations for small and medium message sizes.
- XRC support with Hydra Process Manager
- Improved usability of process to CPU mapping with support of delimiters ('', ' ', '-') in CPU listing
- Thanks to Gilles Civario for the initial patch
 - Use of gfortran as the default F77 compiler
 - Support of Shared-Memory-Nemesis interface on multi-core platforms requiring intra-node communication only (SMP-only systems, laptops, etc.)
- * Bug fixes
 - Fix for memory leak in one-sided code with --enable-g=all --enable-error-messages=all
 - Fix for memory leak in getting the context of intra-communicator
 - Fix for shmat() return code check
 - Fix for issues with inter-communicator collectives in Nemesis
 - KNEM patch for osu_bibw issue with KNEM version 0.9.2
 - Fix for osu_bibw error with Shared-memory-Nemesis interface
 - Fix for Win_test error for one-sided RDMA
 - Fix for a hang in collective when thread level is set to multiple
 - Fix for intel test errors with rsend, bsend and ssend operations in Nemesis
 - Fix for memory free issue when it allocated by scandir
 - Fix for a hang in Finalize
 - Fix for issue with MPIU_Find_local_and_external when it is called from MPIDI_CH3I_comm_create
 - Fix for handling CPPFLGS values with spaces
 - Dynamic Process Management to work with XRC support
 - Fix related to disabling CPU affinity when shared memory is turned off at run time
- MVAPICH2-1.5.1 (09/14/10)
- * Features and Enhancements
 - Significantly reduce memory footprint on some systems by changing the stack size setting for multi-rail configurations
 - Optimization to the number of RDMA Fast Path connections
 - Performance improvements in Scatterv and Gatherv collectives for CH3 interface (Thanks to Dan Kokran and Max Suarez of NASA for identifying the issue)
 - Tuning of Broadcast Collective
 - Support for tuning of eager thresholds based on both adapter and platform type
 - Environment variables for message sizes can now be expressed in short form K=Kilobytes and M=Megabytes (e.g. MV2_IBA_EAGER_THRESHOLD=12K)
 - Ability to selectively use some or all HCAs using colon separated lists. e.g. MV2_IBA_HCA=mlx4_0:mlx4_1
 - Improved Bunch/Scatter mapping for process binding with HWLOC and SMT support (Thanks to Dr. Bernd Kallies of ZIB for ideas and suggestions)
 - Update to Hydra code from MPICH2-1.3b1
 - Auto-detection of various iWARP adapters
 - Specifying MV2_USE_IWARP=1 is no longer needed when using iWARP
 - Changing automatic eager threshold selection and tuning for iWARP adapters based on number of nodes in the system instead of the number of processes

- PSM progress loop optimization for QLogic Adapters (Thanks to Dr. Avneesh Pant of QLogic for the patch)

* Bug fixes

- Fix memory leak in registration cache with --enable-g=all
- Fix memory leak in operations using datatype modules
- Fix for rdma_cross_connect issue for RDMA CM. The server is prevented from initiating a connection.
- Don't fail during build if RDMA CM is unavailable
- Various mpirun_rsh bug fixes for CH3, Nemesis and uDAPL interfaces
- ROMIO panfs build fix
- Update panfs for not-so-new ADIO file function pointers
- Shared libraries can be generated with unknown compilers
- Explicitly link against DL library to prevent build error due to DSO link change in Fedora 13 (introduced with gcc-4.4.3-5.fc13)
- Fix regression that prevents the proper use of our internal HWLOC component
- Remove spurious debug flags when certain options are selected at build time
- Error code added for situation when received eager SMP message is larger than receive buffer
- Fix for Gather and GatherV back-to-back hang problem with LiMIC2
- Fix for packetized send in Nemesis
- Fix related to eager threshold in nemesis ib-netmod
- Fix initialization parameter for Nemesis based on adapter type
- Fix for uDAPL one sided operations (Thanks to Jakub Fedoruk from Intel for reporting this)
- Fix an issue with out-of-order message handling for iWARP
- Fixes for memory leak and Shared context Handling in PSM for QLogic Adapters (Thanks to Dr. Avneesh Pant of QLogic for the patch)

MVAPICH2-1.5 (07/09/10)

* Features and Enhancements (since 1.5-RC2)

- SRQ turned on by default for Nemesis interface
- Performance tuning - adjusted eager thresholds for variety of architectures, vbuf size based on adapter types and vbuf pool sizes
- Tuning for Intel iWARP NE020 adapter, thanks to Harry Cropper of Intel
- Introduction of a retry mechanism for RDMA_CM connection establishment

* Bug fixes (since 1.5-RC2)

- Fix in build process with hwloc (for some Distros)
- Fix for memory leak (Nemesis interface)

MVAPICH2-1.5-RC2 (06/21/10)

* Features and Enhancements (since 1.5-RC1)

- Support for hwloc library (1.0.1) for defining CPU affinity
- Deprecating the PLPA support for defining CPU affinity
- Efficient CPU affinity policies (bunch and scatter) to

Appendix E. MVAPICH2 Release Information

- specify CPU affinity per job for modern multi-core platforms
- New flag in mpirun_rsh to execute tasks with different group IDs
- Enhancement to the design of Win_complete for RMA operations
- Flexibility to support variable number of RMA windows
- Support for Intel iWARP NE020 adapter

* Bug fixes (since 1.5-RC1)

- Compilation issue with the ROMIO adio-lustre driver, thanks to Adam Moody of LLNL for reporting the issue
- Allowing checkpoint-restart for large-scale systems
- Correcting a bug in clear_kvc function. Thanks to T J (Chris) Ward, IBM Research, for reporting and providing the resolving patch
- Shared lock operations with RMA with scatter process distribution. Thanks to Pavan Balaji of Argonne for reporting this issue
- Fix a bug during window creation in uDAPL
- Compilation issue with --enable-alloca, Thanks to E. Borisch, for reporting and providing the patch
- Improved error message for ibv_poll_cq failures
- Fix an issue that prevents mpirun_rsh to execute programs without specifying the path from directories in PATH
- Fix an issue of mpirun_rsh with Dynamic Process Migration (DPM)
- Fix for memory leaks (both CH3 and Nemesis interfaces)
- Updatefiles correctly update LiMIC2
- Several fixes to the registration cache (CH3, Nemesis and uDAPL interfaces)
- Fix to multi-rail communication
- Fix to Shared Memory communication Progress Engine
- Fix to all-to-all collective for large number of processes

MVAPICH2-1.5-RC1 (05/04/10)

* Features and Enhancements

- MPI 2.2 compliant
- Based on MPICH2-1.2.1p1
- OFA-IB-Nemesis interface design
 - OpenFabrics InfiniBand network module support for MPICH2 Nemesis modular design
 - Support for high-performance intra-node shared memory communication provided by the Nemesis design
 - Adaptive RDMA Fastpath with Polling Set for high-performance inter-node communication
 - Shared Receive Queue (SRQ) support with flow control, uses significantly less memory for MPI library
 - Header caching
 - Advanced AVL tree-based Resource-aware registration cache
 - Memory Hook Support provided by integration with ptmalloc2 library. This provides safe release of memory to the Operating System and is expected to benefit the memory usage of applications that heavily use malloc and free

operations.

- Support for TotalView debugger
- Shared Library Support for existing binary MPI application programs to run ROMIO Support for MPI-IO
- Support for additional features (such as hwloc,

- hierarchical collectives, one-sided, multithreading, etc.), as included in the MPICH2 1.2.1p1 Nemesis channel
 - Flexible process manager support
 - mpirun_rsh to work with any of the eight interfaces (CH3 and Nemesis channel-based) including OFA-IB-Nemesis, TCP/IP-CH3 and TCP/IP-Nemesis
 - Hydra process manager to work with any of the eight interfaces (CH3 and Nemesis channel-based) including OFA-IB-CH3, OFA-iWARP-CH3, OFA-RoCE-CH3 and TCP/IP-CH3
 - MPIEXEC_TIMEOUT is honored by mpirun_rsh
- * Bug fixes since 1.4.1
- Fix compilation error when configured with `'--enable-thread-funneled'`
 - Fix MPE functionality, thanks to Anthony Chan for reporting and providing the resolving patch
 - Cleanup after a failure in the init phase is handled better by mpirun_rsh
 - Path determination is correctly handled by mpirun_rsh when DPM is used
 - Shared libraries are correctly built (again)

MVAPICH2-1.4.1

- * Enhancements since mvapich2-1.4
- MPMD launch capability to mpirun_rsh
 - Portable Hardware Locality (hwloc) support, patch suggested by Dr. Bernd Kallies <kallies@zib.de>
 - Multi-port support for iWARP
 - Enhanced iWARP design for scalability to higher process count
 - Ring based startup support for RDMAoE
- * Bug fixes since mvapich2-1.4
- Fixes for MPE and other profiling tools as suggested by Anthony Chan (chan@mcs.anl.gov)
 - Fixes for finalization issue with dynamic process management
 - Removed overrides to PSM_SHAREDCONTEXT, PSM_SHAREDCONTEXTS_MAX variables. Suggested by Ben Truscott <b.s.truscott@bristol.ac.uk>.
 - Fixing the error check for buffer aliasing in MPI_Reduce as suggested by Dr. Rajeev Thakur <thakur@mcs.anl.gov>
 - Fix Totalview integration for RHEL5
 - Update simplemake to handle build timestamp issues
 - Fixes for `--enable-g={mem, meminit}`
 - Improved logic to control the receive and send requests to handle the limitation of CQ Depth on iWARP
 - Fixing assertion failures with IMB-EXT tests
 - VBUF size for very small iWARP clusters bumped up to 33K
 - Replace internal mallocs with MPIU_Malloc uniformly for correct tracing with `--enable-g=mem`
 - Fixing multi-port for iWARP
 - Fix memory leaks
 - Shared-memory reduce fixes for MPI_Reduce invoked with MPI_IN_PLACE
 - Handling RDMA_CM_EVENT_TIMEWAIT_EXIT event
 - Fix for threaded-ctxdup mpich2 test

Appendix E. MVAPICH2 Release Information

- Detecting spawn errors, patch contributed by Dr. Bernd Kallies <kallies@zib.de>
- IMB-EXT fixes reported by Yutaka from Cray Japan
- Fix alltoall assertion error when limic is used

MVAPICH2-1.4

- * Enhancements since mvapich2-1.4rc2
 - Efficient runtime CPU binding
 - Add an environment variable for controlling the use of multiple cq's for iWARP interface.
 - Add environmental variables to disable registration cache for All-to-All on large systems.
 - Performance tune for pt-to-pt Intra-node communication with LiMIC2
 - Performance tune for MPI_Broadcast
- * Bug fixes since mvapich2-1.4rc2
 - Fix the reading error in lock_get_response by adding initialization to req->mrail.protocol
 - Fix mpirun_rsh scalability issue with hierarchical ssh scheme when launching greater than 8K processes.
 - Add mvapich_ prefix to yacc functions. This can avoid some namespace issues when linking with other libraries. Thanks to Manhui Wang <wangm9@cardiff.ac.uk> for contributing the patch.

MVAPICH2-1.4-rc2

- * Enhancements since mvapich2-1.4rc1
 - Added Feature: Check-point Restart with Fault-Tolerant Backplane Support (FTB_CR)
 - Added Feature: Multiple CQ-based design for Chelsio iWARP
 - Distribute LiMIC2-0.5.2 with MVAPICH2. Added flexibility for selecting and using a pre-existing installation of LiMIC2
 - Increase the amount of command line that mpirun_rsh can handle (Thanks for the suggestion by Bill Barth @ TACC)
- * Bug fixes since mvapich2-1.4rc1
 - Fix for hang with packetized send using RDMA Fast path
 - Fix for allowing to use user specified P_Key's (Thanks to Mike Heinz @ QLogic)
 - Fix for allowing mpirun_rsh to accept parameters through the parameters file (Thanks to Mike Heinz @ QLogic)
 - Modify the default value of shmem_bcast_leaders to 4K
 - Fix for one-sided with XRC support
 - Fix hang with XRC
 - Fix to always enabling MVAPICH2_Sync_Checkpoint functionality
 - Fix build error on RHEL 4 systems (Reported by Nathan Baca and Jonathan Atencio)
 - Fix issue with PGI compilation for PSM interface
 - Fix for one-sided accumulate function with user-defined contiguous datatypes
 - Fix linear/hierarchical switching logic and reduce threshold for the enhanced mpirun_rsh framework.
 - Clean up intra-node connection management code for iWARP
 - Fix --enable-g=all issue with uDAPL interface

- Fix one sided operation with on demand CM.
- Fix VPATH build

MVAPICH2-1.4-rc1

* Bugs fixed since MVAPICH2-1.2p1

- Changed parameters for iWARP for increased scalability
- Fix error with derived datatypes and Put and Accumulate operations
Request was being marked complete before data transfer
had actually taken place when MV_RNDV_PROTOCOL=R3 was used
- Unregister stale memory registrations earlier to prevent
malloc failures
- Fix for compilation issues with --enable-g=mem and --enable-g=all
- Change dapl_prepost_noop_extra value from 5 to 8 to prevent
credit flow issues.
- Re-enable RGET (RDMA Read) functionality
- Fix SRQ Finalize error
Make sure that finalize does not hang when the srq_post_cond is
being waited on.
- Fix a multi-rail one-sided error when multiple QPs are used
- PMI Lookup name failure with SLURM
- Port auto-detection failure when the 1st HCA did
not have an active failure
- Change default small message scheduling for multirail
for higher performance
- MPE support for shared memory collectives now available

MVAPICH2-1.2p1 (11/11/2008)

* Changes since MVAPICH2-1.2

- Fix shared-memory communication issue for AMD Barcelona systems.

MVAPICH2-1.2 (11/06/2008)

* Bugs fixed since MVAPICH2-1.2-rc2

- Ignore the last bit of the pkey and remove the pkey_ix option since the
index can be different on different machines. Thanks for Pasha@Mellanox for
the patch.
- Fix data types for memory allocations. Thanks for Dr. Bill Barth from TACC
for the patches.

Appendix E. MVAPICH2 Release Information

- Fix a bug when MV2_NUM_HCAS is larger than the number of active HCAs.
- Allow builds on architectures for which tuning parameters do not exist.
- * Changes related to the mpirun_rsh framework
 - Always build and install mpirun_rsh in addition to the process manager(s) selected through the --with-pm mechanism.
 - Cleaner job abort handling
 - Ability to detect the path to mpispawn if the Linux proc filesystem is available.
 - Added Totalview debugger support
 - Stdin is only available to rank 0. Other ranks get /dev/null.
- * Other miscellaneous changes
 - Add sequence numbers for RPUT and RGET finish packets.
 - Increase the number of allowed nodes for shared memory broadcast to 4K.
 - Use /dev/shm on Linux as the default temporary file path for shared memory communication. Thanks for Doug Johnson@OSC for the patch.
 - MV2_DEFAULT_MAX_WQE has been replaced with MV2_DEFAULT_MAX_SEND_WQE and MV2_DEFAULT_MAX_RECV_WQE for send and recv wqes, respectively.
 - Fix compilation warnings.

MVAPICH2-1.2-RC2 (08/20/2008)

- * Following bugs are fixed in RC2
 - Properly handle the scenario in shared memory broadcast code when the datatypes of different processes taking part in broadcast are different.
 - Fix a bug in Checkpoint-Restart code to determine whether a connection is a shared memory connection or a network connection.
 - Support non-standard path for BLCR header files.
 - Increase the maximum heap size to avoid race condition in realloc().
 - Use int32_t for rank for larger jobs with 32k processes or more.
 - Improve mvapich2-1.2 bandwidth to the same level of mvapich2-1.0.3.
 - An error handling patch for uDAPL interface. Thanks for Nilesh Awate for the patch.
 - Explicitly set some of the EP attributes when on demand connection is used

in uDAPL interface.

MVAPICH2-1.2-RC1 (07/02/08)

* Following features are added for this new mvapich2-1.2 release:

- Based on MPICH2 1.0.7
- Scalable and robust daemon-less job startup
 - Enhanced and robust mpirun_rsh framework (non-MPD-based) to provide scalable job launching on multi-thousand core clusters
 - Available for OpenFabrics (IB and iWARP) and uDAPL interfaces (including Solaris)
- Adding support for intra-node shared memory communication with Checkpoint-restart
 - Allows best performance and scalability with fault-tolerance support
- Enhancement to software installation
 - Change to full autoconf-based configuration
 - Adding an application (mpiname) for querying the MVAPICH2 library version and configuration information
- Enhanced processor affinity using PLPA for multi-core architectures
- Allows user-defined flexible processor affinity
- Enhanced scalability for RDMA-based direct one-sided communication with less communication resource
- Shared memory optimized MPI_Bcast operations
- Optimized and tuned MPI_Alltoall

MVAPICH2-1.0.2 (02/20/08)

- * Change the default MV2_DAPL_PROVIDER to OpenIB-cma
- * Remove extraneous parameter is_blocking from the gen2 interface for MPIDI_CH3I_MRAILI_Get_next_vbuf
- * Explicitly name unions in struct ibv_wr_descriptor and reference the members in the code properly.
- * Change "inline" functions to "static inline" properly.
- * Increase the maximum number of buffer allocations for communication intensive applications
- * Corrections for warnings from the Sun Studio 12 compiler.

Appendix E. MVAPICH2 Release Information

- * If malloc hook initialization fails, then turn off registration cache
- * Add MV_R3_THESHOLD and MV_R3_NOCACHE_THRESHOLD which allows R3 to be used for smaller messages instead of registering the buffer and using a zero-copy protocol.
- * Fixed an error in message coalescing.
- * Setting application initiated checkpoint as default if CR is turned on.

MVAPICH2-1.0.1 (10/29/07)

- * Enhance udapl initializaton, set all ep_attr fields properly. Thanks for Kanoj Sarcar from NetXen for the patch.
- * Fixing a bug that miscalculates the receive size in case of complex datatype is used. Thanks for Patrice Martinez from Bull for reporting this problem.
- * Minor patches for fixing (i) NBO for rdma-cm ports and (ii) rank variable usage in DEBUG_PRINT in rdma-cm.c. Thanks to Steve Wise for reporting these.

MVAPICH2-1.0 (09/14/07)

- * Following features and bug fixes are added in this new MVAPICH2-1.0 release:
 - Message coalescing support to enable reduction of per Queue-pair send queues for reduction in memory requirement on large scale clusters. This design also increases the small message messaging rate significantly. Available for Open Fabrics Gen2-IB.
 - Hot-Spot Avoidance Mechanism (HSAM) for alleviating network congestion in large scale clusters. Available for Open Fabrics Gen2-IB.
 - RDMA CM based on-demand connection management for large scale clusters. Available for OpenFabrics Gen2-IB and Gen2-iWARP.
 - uDAPL on-demand connection management for large scale clusters. Available for uDAPL interface (including Solaris IB implementation).
 - RDMA Read support for increased overlap of computation and communication. Available for OpenFabrics Gen2-IB and Gen2-iWARP.
 - Application-initiated system-level (synchronous) checkpointing in addition to the user-transparent checkpointing. User application can now request a whole program checkpoint synchronously with BLCR by calling special functions within the application. Available for OpenFabrics Gen2-IB.
 - Network-Level fault tolerance with Automatic Path Migration (APM)

for tolerating intermittent network failures over InfiniBand.
Available for OpenFabrics Gen2-IB.

- Integrated multi-rail communication support for OpenFabrics Gen2-iWARP.
- Blocking mode of communication progress. Available for OpenFabrics Gen2-IB.
- Based on MPICH2 1.0.5p4.

- * Fix for hang while using IMB with -multi option.
Thanks to Pasha (Mellanox) for reporting this.
- * Fix for hang in memory allocations $> 2^{31} - 1$.
Thanks to Bryan Putnam (Purdue) for reporting this.
- * Fix for RDMA_CM finalize rdma_destroy_id failure.
Added Timeout env variable for RDMA_CM ARP.
Thanks to Steve Wise for suggesting these.
- * Fix for RDMA_CM invalid event in finalize. Thanks to Steve Wise and Sean Hefty.
- * Fix for shmем memory collectives related memory leaks
- * Updated src/mpi/romio/adio/ad_panfs/Makefile.in include path to find mpi.h.
Contributed by David Gunter, Los Alamos National Laboratory.
- * Fixed header caching error on handling datatype messages with small vector sizes.
- * Change the finalization protocol for UD connection manager.
- * Fix for the "command line too long" problem. Contributed by Xavier Bru <xavier.bru@bull.net> from Bull (<http://www.bull.net/>)
- * Change the CKPT handling to invalidate all unused registration cache.
- * Added ofed 1.2 interface change patch for iwarp/rdma_cm from Steve Wise.
- * Fix for rdma_cm_get_event err in finalize. Reported by Steve Wise.
- * Fix for when MV2_IBA_HCA is used. Contributed by Michael Schwind of Technical Univ. of Chemnitz (Germany).

MVAPICH2-0.9.8 (11/10/06)

- * Following features are added in this new MVAPICH2-0.9.8 release:
 - BLCR based Checkpoint/Restart support
 - iWARP support: tested with Chelsio and Ammasso adapters and OpenFabrics/Gen2 stack

Appendix E. MVAPICH2 Release Information

- RDMA CM connection management support
- Shared memory optimizations for collective communication operations
- uDAPL support for NetEffect 10GigE adapter.

MVAPICH2-0.9.6 (10/22/06)

* Following features and bug fixes are added in this new MVAPICH2-0.9.6 release:

- Added on demand connection management.
- Enhance shared memory communication support.
- Added ptmalloc memory hook support.
- Runtime selection for most configuration options.

MVAPICH2-0.9.5 (08/30/06)

* Following features and bug fixes are added in this new MVAPICH2-0.9.5 release:

- Added multi-rail support for both point to point and direct one side operations.
- Added adaptive RDMA fast path.
- Added shared receive queue support.
- Added TotalView debugger support
- * Optimization of SMP startup information exchange for USE_MPD_RING to enhance performance for SLURM. Thanks to Don and team members from Bull and folks from LLNL for their feedbacks and comments.
- * Added uDAPL build script functionality to set DAPL_DEFAULT_PROVIDER explicitly with default suggestions.
- * Thanks to Harvey Richardson from Sun for suggesting this feature.

MVAPICH2-0.9.3 (05/20/06)

* Following features are added in this new MVAPICH2-0.9.3 release:

- Multi-threading support
- Integrated with MPICH2 1.0.3 stack
- Advanced AVL tree-based Resource-aware registration cache
- Tuning and Optimization of various collective algorithms

- Processor affinity for intra-node shared memory communication
- Auto-detection of InfiniBand adapters for Gen2

MVAPICH2-0.9.2 (01/15/06)

- * Following features are added in this new MVAPICH2-0.9.2 release:
 - InfiniBand support for OpenIB/Gen2
 - High-performance and optimized support for many MPI-2 functionalities (one-sided, collectives, datatype)
 - Support for other MPI-2 functionalities (as provided by MPICH2 1.0.2p1)
 - High-performance and optimized support for all MPI-1 functionalities

MVAPICH2-0.9.0 (11/01/05)

- * Following features are added in this new MVAPICH2-0.9.0 release:
 - Optimized two-sided operations with RDMA support
 - Efficient memory registration/de-registration schemes for RDMA operations
 - Optimized intra-node shared memory support (bus-based and NUMA)
 - Shared library support
 - ROMIO support
 - Support for multiple compilers (gcc, icc, and pgi)

MVAPICH2-0.6.5 (07/02/05)

- * Following features are added in this new MVAPICH2-0.6.5 release:
 - uDAPL support (tested for InfiniBand, Myrinet, and Ammasso GigE)

MVAPICH2-0.6.0 (11/04/04)

- * Following features are added in this new MVAPICH2-0.6.0 release:
 - MPI-2 functionalities (one-sided, collectives, datatype)
 - All MPI-1 functionalities
 - Optimized one-sided operations (Get, Put, and Accumulate)
 - Support for active and passive synchronization

Appendix E. MVAPICH2 Release Information

- Optimized two-sided operations
- Scalable job start-up
- Optimized and tuned for the above platforms and different network interfaces (PCI-X and PCI-Express)
- Memory efficient scaling modes for medium and large clusters

Notes

1. <http://mvapich.cse.ohio-state.edu/>
2. http://mvapich.cse.ohio-state.edu/support/user_guide_mvapich2-2.0rc1.html
3. <http://mvapich.cse.ohio-state.edu/static/media/mvapich/mvapich2-2.1-userguide.html#x1-540006.5>

Appendix F. MPICH-3 Release Information

The following is reproduced essentially verbatim from files contained within the MPICH-3 tarball downloaded from <http://www.mpich.org/>. See <http://www.mpich.org/documentation/guides/> for various user guides.

CHANGELOG

```
=====
                          Changes in 3.2
=====
```

- # Added support for MPI-3.1 features including nonblocking collective I/O, address manipulation routines, thread-safety for MPI initialization, pre-init functionality, and new MPI_T routines to look up variables by name.
- # Fortran 2008 bindings are enabled by default and fully supported.
- # Added support for the Mellanox MXM InfiniBand interface. (thanks to Mellanox for the code contribution).
- # Added support for the Mellanox HCOLL interface for collectives. (thanks to Mellanox for the code contribution).
- # Significant stability improvements to the MPICH/portals4 implementation.
- # Completely revamped RMA infrastructure including several scalability improvements, performance improvements, and bug fixes.
- # Added experimental support for Open Fabrics Interfaces (OFI) version 1.0.0. <https://github.com/ofiwg/libfabric> (thanks to Intel for code contribution)
- # The Myrinet MX network module, which had a life cycle from 1.1 till 3.1.2, has now been deleted.
- # Several other minor bug fixes, memory leak fixes, and code cleanup.

A full list of changes is available at the following link:

<http://git.mpich.org/mpich.git/shortlog/v3.1.3..v3.2rc1>

A full list of bugs that have been fixed is available at the following link:

<https://trac.mpich.org/projects/mpich/query?status=closed&group=resolution&milestone=mpich-3.2>

```
=====
                          Changes in 3.1.4
=====
```

- # Bug fixes to MPI-3 shared memory functionality.

Appendix F. MPICH-3 Release Information

- # Fixed a bug that prevented Fortran programs from being profiled by PMPI libraries written in C.
- # Fixed support for building MPICH on OSX with Intel C/C++ and Fortran compilers.
- # Several bug fixes in ROMIO.
- # Enhancements to the testsuite.
- # Backports support for the Mellanox MXM InfiniBand interface.
- # Backports support for the Mellanox HCOLL interface for collectives.
- # Several other minor bug fixes, memory leak fixes, and code cleanup.

A full list of changes is available at the following link:

<http://git.mpich.org/mpich.git/shortlog/v3.1.3..v3.1.4>

=====
Changes in 3.1.3
=====

- # Several enhancements to Portals4 support.
- # Several enhancements to PAMI (thanks to IBM for the code contribution).
- # Several enhancements to the CH3 RMA implementation.
- # Several enhancements to ROMIO.
- # Fixed deadlock in multi-threaded MPI_Comm_idup.
- # Several other minor bug fixes, memory leak fixes, and code cleanup.

A full list of changes is available at the following link:

<http://git.mpich.org/mpich.git/shortlog/v3.1.2..v3.1.3>

A full list of bugs that have been fixed is available at the following link:

<https://trac.mpich.org/projects/mpich/query?status=closed&group=resolution&milestone=mpich-3.1.3>

=====
Changes in 3.1.2
=====

- # Significant enhancements to the BG/Q device, especially for RMA and shared memory functionality.
- # Several enhancements to ROMIO.

- # Upgraded to hwloc-1.9.
- # Added more Fortran 2008 (F08) tests and fixed a few F08 binding bugs. Now all MPICH F90 tests have been ported to F08.
- # Updated weak alias support to align with gcc-4.x
- # Minor enhancements to the CH3 RMA implementation.
- # Better implementation of MPI_Allreduce for intercommunicator.
- # Added environment variables to control memory tracing overhead.
- # Added flags to enable C99 mode with Solaris compilers.
- # Updated implementation of MPI-T CVARs of type MPI_CHAR, as interpreted in MPI-3.0 Errata.
- # Several other minor bug fixes, memory leak fixes, and code cleanup.

A full list of changes is available at the following link:

<http://git.mpich.org/mpich.git/shortlog/v3.1.1..v3.1.2>

A full list of bugs that have been fixed is available at the following link:

<https://trac.mpich.org/projects/mpich/query?status=closed&group=resolution&milestone=mpich-3.1.2>

=====
Changes in 3.1.1
=====

- # Blue Gene/Q implementation supports MPI-3. This release contains a functional and compliant Blue Gene/Q implementation of the MPI-3 standard. Instructions to build on Blue Gene/Q are on the mpich.org wiki: <http://wiki.mpich.org/mpich/index.php/BGQ>
- # Fortran 2008 bindings (experimental). Build with --enable-fortran=all. Must have a Fortran 2008 + TS 29113 capable compiler.
- # Significant rework of MPICH library management and which symbols go into which libraries. Also updated MPICH library names to make them consistent with Intel MPI, Cray MPI and IBM PE MPI. Backward compatibility links are provided for older mpich-based build systems.
- # The ROMIO "Blue Gene" driver has seen significant rework. We have separated "file system" features from "platform" features, since GPFS shows up in more places than just Blue Gene
- # New ROMIO options for aggregator selection and placement on Blue Gene

Appendix F. MPICH-3 Release Information

- # Optional new ROMIO two-phase algorithm requiring less communication for certain workloads
- # The old ROMIO optimization "deferred open" either stopped working or was disabled on several platforms.
- # Added support for powerpcle compiler. Patched libtool in MPICH to support little-endian powerpc linux host.
- # Fixed the prototype of the Reduce_local C++ binding. The previous prototype was completely incorrect. Thanks to Jeff Squyres for reporting the issue.
- # The mpd process manager, which was deprecated and unsupported for the past four major release series (1.3.x till 3.1), has now been deleted. RIP.
- # Several other minor bug fixes, memory leak fixes, and code cleanup.

A full list of changes is available at the following link:

<http://git.mpich.org/mpich.git/shortlog/v3.1..v3.1.1>

A full list of bugs that have been fixed is available at the following link:

<https://trac.mpich.org/projects/mpich/query?status=closed&group=resolution&milestone=mpich-3.1.1>

=====
Changes in 3.1
=====

- # Implement runtime compatibility with MPICH-derived implementations as per the ABI Compatibility Initiative (see www.mpich.org/abi for more information).
- # Integrated MPICH-PAMI code base for Blue Gene/Q and other IBM platforms.
- # Several improvements to the SCIF netmod. (code contribution from Intel).
- # Major revamp of the MPI_T interface added in MPI-3.
- # Added environment variables to control a lot more capabilities for collectives. See the README.envvar file for more information.
- # Allow non-blocking collectives and fault tolerance at the same time. The option MPIR_PARAM_ENABLE_COLL_FT_RET has been deprecated as it is no longer necessary.
- # Improvements to MPI_WIN_ALLOCATE to internally allocate shared memory between processes on the same node.

- # Performance improvements for MPI RMA operations on shared memory for MPI_WIN_ALLOCATE and MPI_WIN_ALLOCATE_SHARED.
- # Enable shared library builds by default.
- # Upgraded hwloc to 1.8.
- # Several improvements to the Hydra-SLURM integration.
- # Several improvements to the Hydra process binding code. See the Hydra wiki page for more information:
http://wiki.mpich.org/mpich/index.php/Using_the_Hydra_Process_Manager
- # MPICH now supports operations on very large datatypes (those that describe more than 32 bits of data). This work also allows MPICH to fully support MPI-3's introduction of MPI_Count.
- # Several other minor bug fixes, memory leak fixes, and code cleanup.

A full list of changes is available at the following link:

<http://git.mpich.org/mpich.git/shortlog/v3.0.4..v3.1>

A full list of bugs that have been fixed is available at the following link:

<https://trac.mpich.org/projects/mpich/query?status=closed&group=resolution&milestone=mpich-3.1>

=====
Changes in 3.0.4
=====

- # BUILD SYSTEM: Reordered the default compiler search to prefer Intel and PG compilers over GNU compilers because of the performance difference.

WARNING: If you do not explicitly specify the compiler you want through CC and friends, this might break ABI for you relative to the previous 3.0.x release.
- # OVERALL: Added support to manage per-communicator eager-rendezvous thresholds.
- # PM/PMI: Performance improvements to the Hydra process manager on large-scale systems by allowing for key/value caching.
- # Several other minor bug fixes, memory leak fixes, and code cleanup.
A full list of changes is available at the following link:

<http://git.mpich.org/mpich.git/shortlog/v3.0.3..v3.0.4>

=====
Changes in 3.0.3
=====

Appendix F. MPICH-3 Release Information

```
=====
# RMA: Added a new mechanism for piggybacking RMA synchronization operations,
  which improves the performance of several synchronization operations,
  including Flush.

# RMA: Added an optimization to utilize the MPI_MODE_NOCHECK assertion in
  passive target RMA to improve performance by eliminating a lock request
  message.

# RMA: Added a default implementation of shared memory windows to CH3. This
  adds support for this MPI 3.0 feature to the ch3:sock device.

# RMA: Fix a bug that resulted in an error when RMA operation request handles
  where completed outside of a synchronization epoch.

# PM/PMI: Upgraded to hwloc-1.6.2rc1. This version uses libpciaccess
  instead of libpci, to workaround the GPL license used by libpci.

# PM/PMI: Added support for the Cobalt process manager.

# BUILD SYSTEM: allow MPI_LONG_DOUBLE_SUPPORT to be disabled with a configure
  option.

# FORTRAN: fix MPI_WEIGHTS_EMPTY in the Fortran bindings

# MISC: fix a bug in MPI_Get_elements where it could return incorrect values

# Several other minor bug fixes, memory leak fixes, and code cleanup.
  A full list of changes is available at the following link:

  http://git.mpich.org/mpich.git/shortlog/v3.0.2..v3.0.3
=====
```

Changes in 3.0.2

```
=====
# PM/PMI: Upgrade to hwloc-1.6.1

# RMA: Performance enhancements for shared memory windows.

# COMPILER INTEGRATION: minor improvements and fixes to the clang static type
  checking annotation macros.

# MPI-IO (ROMIO): improved error checking for user errors, contributed by IBM.

# MPI-3 TOOLS INTERFACE: new MPI_T performance variables providing information
  about nemesis communication behavior and and CH3 message matching queues.

# TEST SUITE: "make testing" now also outputs a "summary.tap" file that can be
  interpreted with standard TAP consumer libraries and tools. The
  "summary.xml" format remains unchanged.

# GIT: This is the first release built from the new git repository at
=====
```


git.mpich.org. A few build system mechanisms have changed because of this switch.

BUG FIX: resolved a compilation error related to LLONG_MAX that affected several users (ticket #1776).

BUG FIX: nonblocking collectives now properly make progress when MPICH is configured with the ch3:sock channel (ticket #1785).

Several other minor bug fixes, memory leak fixes, and code cleanup. A full list of changes is available at the following link:

<http://git.mpich.org/mpich.git/shortlog/v3.0.1..v3.0.2>

=====
Changes in 3.0.1
=====

PM/PMI: Critical bug-fix in Hydra to work correctly in multi-node tests.

A full list of changes is available using:

`svn log -r10790:HEAD https://svn.mcs.anl.gov/repos/mpi/mpich2/tags/release/mpich-3.0.1`

... or at the following link:

https://trac.mcs.anl.gov/projects/mpich2/log/mpich2/tags/release/mpich-3.0.1? \ action=follow_copy&rev=HEAD&stop_rev=10790&mode=follow_copy

=====
Changes in 3.0
=====

MPI-3: All MPI-3 features are now implemented and the MPI_VERSION bumped up to 3.0.

OVERALL: Added support for ARM-v7 native atomics

MPE: MPE is now separated out of MPICH and can be downloaded/used as a separate package.

PM/PMI: Upgraded to hwloc-1.6

Several other minor bug fixes, memory leak fixes, and code cleanup. A full list of changes is available using:

`svn log -r10344:HEAD https://svn.mcs.anl.gov/repos/mpi/mpich2/tags/release/mpich-3.0`

... or at the following link:

https://trac.mcs.anl.gov/projects/mpich2/log/mpich2/tags/release/mpich-3.0? \ action=follow_copy&rev=HEAD&stop_rev=10344&mode=follow_copy

```
=====
Changes in 1.5
=====
```

```
# OVERALL: Nemesis now supports an "--enable-yield=..." configure
option for better performance/behavior when oversubscribing
processes to cores. Some form of this option is enabled by default
on Linux, Darwin, and systems that support sched_yield().

# OVERALL: Added support for Intel Many Integrated Core (MIC)
architecture: shared memory, TCP/IP, and SCIF based communication.

# OVERALL: Added support for IBM BG/Q architecture. Thanks to IBM
for the contribution.

# MPI-3: const support has been added to mpi.h, although it is
disabled by default. It can be enabled on a per-translation unit
basis with "#define MPICH2_CONST const".

# MPI-3: Added support for MPIX_Type_create_hindexed_block.

# MPI-3: The new MPI-3 nonblocking collective functions are now
available as "MPIX_" functions (e.g., "MPIX_Ibcast").

# MPI-3: The new MPI-3 neighborhood collective routines are now available as
"MPIX_" functions (e.g., "MPIX_Neighbor_allgather").

# MPI-3: The new MPI-3 MPI_Comm_split_type function is now available
as an "MPIX_" function.

# MPI-3: The new MPI-3 tools interface is now available as "MPIX_T_"
functions. This is a beta implementation right now with several
limitations, including no support for multithreading. Several
performance variables related to CH3's message matching are exposed
through this interface.

# MPI-3: The new MPI-3 matched probe functionality is supported via
the new routines MPIX_Mprobe, MPIX_Improbe, MPIX_Mrecv, and
MPIX_Imrecv.

# MPI-3: The new MPI-3 nonblocking communicator duplication routine,
MPIX_Comm_idup, is now supported. It will only work for
single-threaded programs at this time.

# MPI-3: MPIX_Comm_reenable_anysource support

# MPI-3: Native MPIX_Comm_create_group support (updated version of
the prior MPIX_Group_comm_create routine).

# MPI-3: MPI_Intercomm_create's internal communication no longer interferes
with point-to-point communication, even if point-to-point operations on the
parent communicator use the same tag or MPI_ANY_TAG.
```

```
# MPI-3: Eliminated the possibility of interference between
MPI_Intercomm_create and point-to-point messaging operations.

# Build system: Completely revamped build system to rely fully on
autotools. Parallel builds ("make -j8" and similar) are now supported.

# Build system: rename "./maint/updatefiles" --> "./autogen.sh" and
"configure.in" --> "configure.ac"

# JUMPSHOT: Improvements to Jumpshot to handle thousands of
timelines, including performance improvements to slog2 in such
cases.

# JUMPSHOT: Added navigation support to locate chosen drawable's ends
when viewport has been scrolled far from the drawable.

# PM/PMI: Added support for memory binding policies.

# PM/PMI: Various improvements to the process binding support in
Hydra. Several new pre-defined binding options are provided.

# PM/PMI: Upgraded to hwloc-1.5

# PM/PMI: Several improvements to PBS support to natively use the PBS
launcher.

# Several other minor bug fixes, memory leak fixes, and code cleanup.
A full list of changes is available using:

    svn log -r8478:HEAD https://svn.mcs.anl.gov/repos/mpi/mpich2/tags/release/mpich2-1.5

... or at the following link:

    https://trac.mcs.anl.gov/projects/mpich2/log/mpich2/tags/release/mpich2-1.5? \
action=follow_copy&rev=HEAD&stop_rev=8478&mode=follow_copy
```

=====
Changes in 1.4.1
=====

```
# OVERALL: Several improvements to the ARMCI API implementation
within MPICH2.

# Build system: Added beta support for DESTDIR while installing
MPICH2.

# PM/PMI: Upgrade hwloc to 1.2.1rc2.

# PM/PMI: Initial support for the PBS launcher.

# Several other minor bug fixes, memory leak fixes, and code cleanup.
A full list of changes is available using:

    svn log -r8675:HEAD https://svn.mcs.anl.gov/repos/mpi/mpich2/tags/release/mpich2-1.4.1
```

Appendix F. MPICH-3 Release Information

... or at the following link:

https://trac.mcs.anl.gov/projects/mpich2/log/mpich2/tags/release/mpich2-1.4.1? \ action=follow_copy&rev=HEAD&stop_rev=8675&mode=follow_copy

```
=====
                          Changes in 1.4
=====
```

```
# OVERALL: Improvements to fault tolerance for collective
operations. Thanks to Rui Wang @ ICT for reporting several of these
issues.

# OVERALL: Improvements to the universe size detection. Thanks to
Yauheni Zelenko for reporting this issue.

# OVERALL: Bug fixes for Fortran attributes on some systems. Thanks
to Nicolai Stange for reporting this issue.

# OVERALL: Added new ARMCI API implementation (experimental).

# OVERALL: Added new MPIX_Group_comm_create function to allow
non-collective creation of sub-communicators.

# FORTRAN: Bug fixes in the MPI_DIST_GRAPH_ Fortran bindings.

# PM/PMI: Support for a manual "none" launcher in Hydra to allow for
higher-level tools to be built on top of Hydra. Thanks to Justin
Wozniak for reporting this issue, for providing several patches for
the fix, and testing it.

# PM/PMI: Bug fixes in Hydra to handle non-uniform layouts of hosts
better. Thanks to the MVAPICH group at OSU for reporting this issue
and testing it.

# PM/PMI: Bug fixes in Hydra to handle cases where only a subset of
the available launchers or resource managers are compiled
in. Thanks to Satish Balay @ Argonne for reporting this issue.

# PM/PMI: Support for a different username to be provided for each
host; this only works for launchers that support this (such as
SSH).

# PM/PMI: Bug fixes for using Hydra on AIX machines. Thanks to
Kitrick Sheets @ NCSA for reporting this issue and providing the
first draft of the patch.

# PM/PMI: Bug fixes in memory allocation/management for environment
variables that was showing up on older platforms. Thanks to Steven
Sutphen for reporting the issue and providing detailed analysis to
track down the bug.

# PM/PMI: Added support for providing a configuration file to pick
```

the default options for Hydra. Thanks to Saurabh T. for reporting the issues with the current implementation and working with us to improve this option.

- # PM/PMI: Improvements to the error code returned by Hydra.
- # PM/PMI: Bug fixes for handling "=" in environment variable values in hydra.
- # PM/PMI: Upgrade the hwloc version to 1.2.
- # COLLECTIVES: Performance and memory usage improvements for MPI_Bcast in certain cases.
- # VALGRIND: Fix incorrect Valgrind client request usage when MPICH2 is built for memory debugging.
- # BUILD SYSTEM: "--enable-fast" and "--disable-error-checking" are once again valid simultaneous options to configure.
- # TEST SUITE: Several new tests for MPI RMA operations.
- # Several other minor bug fixes, memory leak fixes, and code cleanup. A full list of changes is available using:

svn log -r7838:HEAD <https://svn.mcs.anl.gov/repos/mpi/mpich2/tags/release/mpich2-1.4>

... or at the following link:

[https://trac.mcs.anl.gov/projects/mpich2/log/mpich2/tags/release/mpich2-1.4? \ action=follow_copy&rev=HEAD&stop_rev=7838&mode=follow_copy](https://trac.mcs.anl.gov/projects/mpich2/log/mpich2/tags/release/mpich2-1.4?%5Baction=follow_copy&rev=HEAD&stop_rev=7838&mode=follow_copy)

=====
Changes in 1.3.2
=====

- # OVERALL: MPICH2 now recognizes the OSX mach_absolute_time as a native timer type.
- # OVERALL: Performance improvements to MPI_Comm_split on large systems.
- # OVERALL: Several improvements to error returns capabilities in the presence of faults.
- # PM/PMI: Several fixes and improvements to Hydra's process binding capability.
- # PM/PMI: Upgrade the hwloc version to 1.1.1.
- # PM/PMI: Allow users to sort node lists allocated by resource managers in Hydra.
- # PM/PMI: Improvements to signal handling. Now Hydra respects Ctrl-Z

Appendix F. MPICH-3 Release Information

signals and passes on the signal to the application.

PM/PMI: Improvements to STDOUT/STDERR handling including improved support for rank prepending on output. Improvements to STDIN handling for applications being run in the background.

PM/PMI: Split the bootstrap servers into "launchers" and "resource managers", allowing the user to pick a different resource manager from the launcher. For example, the user can now pick the "SLURM" resource manager and "SSH" as the launcher.

PM/PMI: The MPD process manager is deprecated.

PM/PMI: The PLPA process binding library support is deprecated.

WINDOWS: Adding support for gfortran and 64-bit gcc libs.

Several other minor bug fixes, memory leak fixes, and code cleanup. A full list of changes is available using:

```
svn log -r7457:HEAD https://svn.mcs.anl.gov/repos/mpi/mpich2/tags/release/mpich2-1.3.2
```

... or at the following link:

```
https://trac.mcs.anl.gov/projects/mpich2/log/mpich2/tags/release/mpich2-1.3.2? \
action=follow_copy&rev=HEAD&stop_rev=7457&mode=follow_copy
```

=====
Changes in 1.3.1
=====

OVERALL: MPICH2 is now fully compliant with the CIFS FTB standard MPI events (based on the draft standard).

OVERALL: Major improvements to RMA performance for long lists of RMA operations.

OVERALL: Performance improvements for Group_translate_ranks.

COLLECTIVES: Collective algorithm selection thresholds can now be controlled at runtime via environment variables.

ROMIO: PVFS error codes are now mapped to MPI error codes.

Several other minor bug fixes, memory leak fixes, and code cleanup. A full list of changes is available using:

```
svn log -r7350:HEAD https://svn.mcs.anl.gov/repos/mpi/mpich2/tags/release/mpich2-1.3.1
```

... or at the following link:

```
https://trac.mcs.anl.gov/projects/mpich2/log/mpich2/tags/release/mpich2-1.3.1? \
action=follow_copy&rev=HEAD&stop_rev=7350&mode=follow_copy
```

=====
Changes in 1.3
=====

```
# OVERALL: Initial support for fine-grained threading in
ch3:nemesis:tcp.

# OVERALL: Support for Asynchronous Communication Progress.

# OVERALL: The ssm and shm channels have been removed.

# OVERALL: Checkpoint/restart support using BLCR.

# OVERALL: Improved tolerance to process and communication failures
when error handler is set to MPI_ERRORS_RETURN. If a communication
operation fails (e.g., due to a process failure) MPICH2 will return
an error, and further communication to that process is not
possible. However, communication with other processes will still
proceed normally. Note, however, that the behavior collective
operations on communicators containing the failed process is
undefined, and may give incorrect results or hang some processes.

# OVERALL: Experimental support for inter-library dependencies.

# PM/PMI: Hydra is now the default process management framework
replacing MPD.

# PM/PMI: Added dynamic process support for Hydra.

# PM/PMI: Added support for LSF, SGE and POE in Hydra.

# PM/PMI: Added support for CPU and memory/cache topology aware
process-core binding.

# DEBUGGER: Improved support and bug fixes in the Totalview support.

# Build system: Replaced F90/F90FLAGS by FC/FCFLAGS. F90/F90FLAGS are
not longer supported in the configure.

# Multi-compiler support: On systems where C compiler that is used to
build mpich2 libraries supports multiple weak symbols and multiple aliases,
the Fortran binding built in the mpich2 libraries can handle different
Fortran compilers (than the one used to build mpich2). Details in README.

# Several other minor bug fixes, memory leak fixes, and code cleanup.
A full list of changes is available using:

    svn log -r5762:HEAD https://svn.mcs.anl.gov/repos/mpi/mpich2/tags/release/mpich2-1.3

... or at the following link:

    https://trac.mcs.anl.gov/projects/mpich2/log/mpich2/tags/release/mpich2-1.3? \
action=follow_copy&rev=HEAD&stop_rev=5762&mode=follow_copy
```

Appendix F. MPICH-3 Release Information

```
=====  
Changes in 1.2.1  
=====
```

```
# OVERALL: Improved support for fine-grained multithreading.  
  
# OVERALL: Improved integration with Valgrind for debugging builds of MPICH2.  
  
# PM/PMI: Initial support for hwloc process-core binding library in  
Hydra.  
  
# PM/PMI: Updates to the PMI-2 code to match the PMI-2 API and  
wire-protocol draft.  
  
# Several other minor bug fixes, memory leak fixes, and code cleanup.  
A full list of changes is available using:  
  
    svn log -r5425:HEAD https://svn.mcs.anl.gov/repos/mpi/mpich2/tags/release/mpich2-1.2.1  
  
    ... or at the following link:  
  
    https://trac.mcs.anl.gov/projects/mpich2/log/mpich2/tags/release/mpich2-1.2.1? \  
action=follow_copy&rev=HEAD&stop_rev=5425&mode=follow_copy
```

```
=====  
Changes in 1.2  
=====
```

```
# OVERALL: Support for MPI-2.2  
  
# OVERALL: Several fixes to Nemesis/MX.  
  
# WINDOWS: Performance improvements to Nemesis/windows.  
  
# PM/PMI: Scalability and performance improvements to Hydra using  
PMI-1.1 process-mapping features.  
  
# PM/PMI: Support for process-binding for hyperthreading enabled  
systems in Hydra.  
  
# PM/PMI: Initial support for PBS as a resource management kernel in  
Hydra.  
  
# PM/PMI: PMI2 client code is now officially included in the release.  
  
# TEST SUITE: Support to run the MPICH2 test suite through valgrind.  
  
# Several other minor bug fixes, memory leak fixes, and code cleanup.  
A full list of changes is available using:  
  
    svn log -r5025:HEAD https://svn.mcs.anl.gov/repos/mpi/mpich2/tags/release/mpich2-1.2  
  
    ... or at the following link:
```


https://trac.mcs.anl.gov/projects/mpich2/log/mpich2/tags/release/mpich2-1.2? \ action=follow_copy&rev=HEAD&stop_rev=5025&mode=follow_copy

=====
Changes in 1.1.1p1
=====

- OVERALL: Fixed an invalid read in the dataloop code for zero count types.
- OVERALL: Fixed several bugs in ch3:nemesis:mx (tickets #744,#760; also change r5126).
- BUILD SYSTEM: Several fixes for functionality broken in 1.1.1 release, including MPICH2LIB_xFLAGS and extra libraries living in \$LIBS instead of \$LDLDFLAGS. Also, '-lpthread' should no longer be duplicated in link lines.
- BUILD SYSTEM: MPICH2 shared libraries are now compatible with glibc versioned symbols on Linux, such as those present in the MX shared libraries.
- BUILD SYSTEM: Minor tweaks to improve compilation under the nvcc CUDA compiler.
- PM/PMI: Fix mpd incompatibility with python2.3 introduced in mpich2-1.1.1.
- PM/PMI: Several fixes to hydra, including memory leak fixes and process binding issues.
- TEST SUITE: Correct invalid arguments in the coll2 and coll3 tests.
- Several other minor bug fixes, memory leak fixes, and code cleanup. A full list of changes is available using:

svn log -r5032:HEAD <https://svn.mcs.anl.gov/repos/mpi/mpich2/tags/release/mpich2-1.1.1p1>

... or at the following link:

https://trac.mcs.anl.gov/projects/mpich2/log/mpich2/tags/release/mpich2-1.1.1p1? \ action=follow_copy&rev=HEAD&stop_rev=5032&mode=follow_copy

=====
Changes in 1.1.1
=====

- # OVERALL: Improved support for Boost MPI.
- # PM/PMI: Significantly improved time taken by MPI_Init with Nemesis and MPD on large numbers of processes.
- # PM/PMI: Improved support for hybrid MPI-UPC program launching with Hydra.
- # PM/PMI: Improved support for process-core binding with Hydra.

Appendix F. MPICH-3 Release Information

```
# PM/PMI: Preliminary support for PMI-2. Currently supported only
with Hydra.

# Many other bug fixes, memory leak fixes and code cleanup. A full
list of changes is available using:

svn log -r4655:HEAD https://svn.mcs.anl.gov/repos/mpi/mpich2/tags/release/mpich2-1.1.1

... or at the following link:

https://trac.mcs.anl.gov/projects/mpich2/log/mpich2/tags/release/mpich2-1.1.1? \
action=follow_copy&rev=HEAD&stop_rev=4655&mode=follow_copy
```

```
=====
Changes in 1.1
=====
```

- OVERALL: Added MPI 2.1 support.
- OVERALL: Nemesis is now the default configuration channel with a completely new TCP communication module.
- OVERALL: Windows support for nemesis.
- OVERALL: Added a new Myrinet MX network module for nemesis.
- OVERALL: Initial support for shared-memory aware collective communication operations. Currently MPI_Bcast, MPI_Reduce, MPI_Allreduce, and MPI_Scan.
- OVERALL: Improved handling of MPI Attributes.
- OVERALL: Support for BlueGene/P through the DCMF library (thanks to IBM for the patch).
- OVERALL: Experimental support for fine-grained multithreading
- OVERALL: Added dynamic processes support for Nemesis.
- OVERALL: Added automatic as well as statically runtime configurable receive timeout variation for MPD (thanks to OSU for the patch).
- OVERALL: Improved performance for MPI_Allgatherv, MPI_Gatherv, and MPI_Alltoall.
- PM/PMI: Initial support for the new Hydra process management framework (current support is for ssh, rsh, fork and a preliminary version of slurm).
- ROMIO: Added support for MPI_Type_create_resized and MPI_Type_create_indexed_block datatypes in ROMIO.
- ROMIO: Optimized Lustre ADIO driver (thanks to Weikuan Yu for initial work and Sun for further improvements).

- Many other bug fixes, memory leak fixes and code cleanup. A full list of changes is available using:

```
svn log -r813:HEAD https://svn.mcs.anl.gov/repos/mpi/mpich2/tags/release/mpich2-1.1
```

... or at the following link:

```
https://trac.mcs.anl.gov/projects/mpich2/log/mpich2/tags/release/mpich2-1.1? \
action=follow_copy&rev=HEAD&stop_rev=813&mode=follow_copy
```

```
=====
                          Changes in 1.0.7
=====
```

- OVERALL: Initial ROMIO device for BlueGene/P (the ADI device is also added but is not configurable at this time).
- OVERALL: Major clean up for the propagation of user-defined and other MPICH2 flags throughout the code.
- OVERALL: Support for STI Cell Broadband Engine.
- OVERALL: Added datatype free hooks to be used by devices independently.
- OVERALL: Added device-specific timer support.
- OVERALL: make uninstall works cleanly now.
- ROMIO: Support to take hints from a config file
- ROMIO: more tests and bug fixes for nonblocking I/O
- PM/PMI: Added support to use PMI Clique functionality for process managers that support it.
- PM/PMI: Added SLURM support to configure to make it transparent to users.
- PM/PMI: SMPD Singleton Init support.
- WINDOWS: Fortran 90 support added.
- SCTP: Added MPICH_SCTP_NAGLE_ON support.
- MPE: Updated MPE logging API so that it is thread-safe (through global mutex).
- MPE: Added infrastructure to piggyback argument data to MPI states.
- DOCS: Documentation creation now works correctly for VPATH builds.
- Many other bug fixes, memory leak fixes and code cleanup. A full

Appendix F. MPICH-3 Release Information

list of changes is available using:

```
svn log -r100:HEAD https://svn.mcs.anl.gov/repos/mpi/mpich2/branches/release/MPICH2_1_0_7
```

```
=====
Changes in 1.0.6
=====
```

- Updates to the ch3:nemesis channel including preliminary support for thread safety.
- Preliminary support for dynamic loading of ch3 channels (sock, ssm, shm). See the README file for details.
- Singleton init now works with the MPD process manager.
- Fixes in MPD related to MPI-2 connect-accept.
- Improved support for MPI-2 generalized requests that allows true nonblocking I/O in ROMIO.
- MPE changes:
 - * Enabled thread-safe MPI logging through global mutex.
 - * Enhanced Jumpshot to be more thread friendly
 - + added simple statistics in the Legend windows.
 - * Added backtrace support to MPE on Solaris and glibc based systems, e.g. Linux. This improves the output error message from the Collective/Datatype checking library.
 - * Fixed the CLOG2 format so it can be used in serial (non-MPI) logging.
- Performance improvements for derived datatypes (including packing and communication) through in-built loop-unrolling and buffer alignment.
- Performance improvements for MPI_Gather when non-power-of-two processes are used, and when a non-zero ranked root is performing the gather.
- MPI_Comm_create works for intercommunicators.
- Enabled -O2 and equivalent compiler optimizations for supported compilers by default (including GNU, Intel, Portland, Sun, Absoft, IBM).
- Many other bug fixes, memory leak fixes and code cleanup. A full list of changes is available at www.mcs.anl.gov/mpi/mpich2/mpich2_1_0_6changes.htm.

```
=====
Changes in 1.0.5
=====
```

- An SCTP channel has been added to the CH3 device. This was implemented by Brad Penoff and Mike Tsai, Univ. of British Columbia.

Their group's webpage is located at <http://www.cs.ubc.ca/labs/dsg/mpi-sctp/> .

- Bugs related to dynamic processes have been fixed.
- Performance-related fixes have been added to derived datatypes and collective communication.
- Updates to the Nemesis channel
- Fixes to thread safety for the ch3:sock channel
- Many other bug fixes and code cleanup. A full list of changes is available at www.mcs.anl.gov/mpi/mpich2/mpich2_1_0_5changes.htm .

=====
Changes in 1.0.4
=====

- For the ch3:sock channel, the default build of MPICH2 supports thread safety. A separate build is not needed as before. However, thread safety is enabled only if the user calls `MPI_Init_thread` with `MPI_THREAD_MULTIPLE`. If not, no thread locks are called, so there is no penalty.
- A new low-latency channel called Nemesis has been added. It can be selected by specifying the option `--with-device=ch3:nemesis`. Nemesis uses shared memory for intranode communication and various networks for internode communication. Currently available networks are TCP, GM and MX. Nemesis is still a work in progress. See the README for more information about the channel.
- Support has been added for providing message queues to debuggers. Configure with `--enable-debuginfo` to make this information available. This is still a "beta" test version and has not been extensively tested.
- For systems with firewalls, the environment variable `MPICH_PORT_RANGE` can be used to restrict the range of ports used by MPICH2. See the documentation for more details.
- Withdrew obsolete modules, including the `ib` and `rdma` communication layers. For Infiniband and MPICH2, please see <http://nowlab.cse.ohio-state.edu/projects/mpi-iba/> For other interconnects, please contact us at mpich2-maint@mcs.anl.gov .
- Numerous bug fixes and code cleanup. A full list of changes is available at www.mcs.anl.gov/mpi/mpich2/mpich2_1_0_4changes.htm .
- Numerous new tests in the MPICH2 test suite.
- For developers, the way in which information is passed between the top level `configure` and `configures` in the device, process management, and related modules has been cleaned up. See the comments at the beginning of the `top-level/configure.in` for details. This change makes it easier to interface other modules to MPICH2.

```
=====
Changes in 1.0.3
=====
```

- There are major changes to the ch3 device implementation. Old and unsupported channels (essm, rdma) have been removed. The internal interface between ch3 and the channels has been improved to simplify the process of adding a new channel (sharing existing code where possible) and to improve performance. Further changes in this internal interface are expected.
- Numerous bug fixes and code cleanup
 - Creation of intercommunicators and intracommunicators from the intercommunicators created with Spawn and Connect/Accept
 - The computation of the alignment and padding of items within structures now handles additional cases, including systems where the alignment and padding depends on the type of the first item in the structure
 - MPD recognizes wdir info keyword
 - gforker's mpiexec supports -env and -genv arguments for controlling which environment variables are delivered to created processes
- While not a bug, to aid in the use of memory trace packages, MPICH2 tries to free all allocated data no later than when MPI_Finalize returns.
- Support for DESTDIR in install targets
- Enhancements to SMPD
- In order to support special compiler flags for users that may be different from those used to build MPICH2, the environment variables MPI_CFLAGS, MPI_FFLAGS, MPI_CXXFLAGS, and MPI_F90FLAGS may be used to specify the flags used in mpicc, mpif77, mpicxx, and mpif90 respectively. The flags CFLAGS, FFLAGS, CXXFLAGS, and F90FLAGS are used in the building of MPICH2.
- Many enhancements to MPE
- Enhanced support for features and idiosyncracies of Fortran 77 and Fortran 90 compilers, including gfortran, g95, and xlf
- Enhanced support for C++ compilers that do not fully support abstract base classes
- Additional tests in the mpich2/tests/mpi
- New FAQ included (also available at <http://www.mcs.anl.gov/mpi/mpich2/faq.htm>)

- Man pages for mpiexec and mpif90
- Enhancements for developers, including a more flexible and general mechanism for inserting logging and information messages, controllable with `--mpich-dbg-xxx` command line arguments or `MPICH_DBG_XXX` environment variables.
- Note to developers:
This release contains many changes to the structure of the CH3 device implementation (in `src/mpid/ch3`), including significant reworking of the files (many files have been combined into fewer files representing logical grouping of functions). The next release of MPICH2 will contain even more significant changes to the device structure as we introduce a new communication implementation.

=====
Changes in 1.0.2
=====

- Optimizations to the MPI-2 one-sided communication functions for the `sshm` (scalable shared memory) channel when window memory is allocated with `MPI_Alloc_mem` (for all three synchronization methods).
- Numerous bug fixes and code cleanup.
- Fixed memory leaks.
- Fixed shared library builds.
- Fixed performance problems with `MPI_Type_create_subarray/darray`
- The following changes have been made to MPE2:
 - MPE2 now builds the MPI collective and datatype checking library by default.
 - SLOG-2 format has been upgraded to 2.0.6 which supports event drawables and provides count of real drawables in preview drawables.
 - new `slog2` tools, `slog2filter` and `slog2updater`, which both are logfile format convertors. `slog2filter` removes undesirable categories of drawables as well as alters the `slog2` file structure. `slog2updater` is a `slog2filter` that reads in older logfile format, 2.0.5, and writes out the latest format 2.0.6.
- The following changes have been made to MPD:
 - Nearly all code has been replaced by new code that follows a more object-oriented approach than before. This has not changed any fundamental behavior or interfaces.
 - There is info support in `spawn` and `spawn_multiple` for providing parts of the environment for spawned processes such as `search-path` and `current working directory`. See the Standard for the required

Appendix F. MPICH-3 Release Information

fields.

- mpdcheck has been enhanced to help users debug their cluster and network configurations.
- CPickle has replaced marshal as the source module for dumps and loads.
- The mpigdb command has been replaced by mpiexec -gdb.
- Alternate interfaces can be used. See the Installer's Guide.

```
=====
Changes in 1.0.1
=====
```

- Copyright statements have been added to all code files, clearly identifying that all code in the distribution is covered by the extremely flexible copyright described in the COPYRIGHT file.
- The MPICH2 test suite (mpich2/test) can now be run against any MPI implementation, not just MPICH2.
- The send and receive socket buffers sizes may now be changed by setting MPICH_SOCKET_BUFFER_SIZE. Note: the operating system may impose a maximum socket buffer size that prohibits MPICH2 from increasing the buffers to the desired size. To raise the maximum allowable buffer size, please contact your system administrator.
- Error handling throughout the MPI routines has been improved. The error handling in some internal routines has been simplified as well, making the routines easier to read.
- MPE (Jumpshot and CLOG logging) is now supported on Microsoft Windows.
- C applications built for Microsoft Windows may select the desired channels at runtime.
- A program not started with mpiexec may become an MPI program by calling MPI_Init. It will have an MPI_COMM_WORLD of size one. It may then call other MPI routines, including MPI_COMM_SPAWN, to become a truly parallel program. At present, the use of MPI_COMM_SPAWN and MPI_COMM_SPAWN_MULTIPLE by such a process is only supported by the MPD process manager.
- Memory leaks in communicator allocation and the C++ binding have been fixed.
- Following GNU guidelines, the parts of the install step that checked the installation have been moved to an installcheck target. Much of the installation now supports the DESTDIR prefix.
- Microsoft Visual Studio projects have been added to make it possible to build x86-64 version
- Problems with compilers and linkers that do not support weak symbols, which are used to support the PMPI profiling interface, have been corrected.

- Handling of Fortran 77 and Fortran 90 compilers has been improved, including support for g95.
- The Fortran stdcall interface on Microsoft Windows now supports character*.
- A bug in the OS X implementation of poll() caused the sock channel to hang. A workaround has been put in place.
- Problems with installation under OS/X are now detected and corrected. (Install breaks libraries that are more than 10 seconds old!)
- The following changes have been made to MPD:
 - Sending a SIGINT to mpiexec/mpdrun, such as by typing control-C, now causes SIGINT to be sent to the processes within the job. Previously, SIGKILL was sent to the processes, preventing applications from catching the signal and performing their own signal processing.
 - The process for merging output has been improved.
 - A new option, -ifhn, has been added to the machine file, allowing the user to select the destination interface to be used for TCP communication. See the User's Manual for details.
 - The user may now select, via the "-s" option to mpiexec/mpdrun, which processes receive input through stdin. stdin is immediately closed for all processes not in set receiving input. This prevents processes not in the set from hanging should they attempt to read from stdin.
 - The MPICH2 Installer's Guide now contains an appendix on troubleshooting problems with MPD.
- The following changes have been made to SMPD:
 - On Windows machines, passwordless authentication (via SSPI) can now be used to start processes on machines within a domain. This feature is a recent addition, and should be considered experimental.
 - On Windows machines, the -localroot option was added to mpiexec, allowing processes on the local machines to perform GUI operations on the local desktop.
 - On Windows machines, network drive mapping is now supported via the -map option to mpiexec.
 - Three new GUI tools have been added for Microsoft Windows. These tools are wrappers to the command line tools, mpiexec.exe and smpd.exe. wmpiexec allows the user to run a job much in the way they with mpiexec. wmpiconfig provides a means of setting various global options to the SMPD process manager environment. wmpiregister encrypts the user's credentials and saves them to the Windows Registry.
- The following changes have been made to MPE2:

Appendix F. MPICH-3 Release Information

- MPE2 no longer attempt to compile or link code during 'make install' to validate the installation. Instead, 'make installcheck' may now be used to verify that the MPE installation.
- MPE2 now supports DESTDIR.
- The sock channel now has preliminary support for MPI_THREAD_SERIALIZED and MPI_THREAD_MULTIPLE on both UNIX and Microsoft Windows. We have performed rudimentary testing; and while overall the results were very positive, known issues do exist. ROMIO in particular experiences hangs in several places. We plan to correct that in the next release. As always, please report any difficulties you encounter.
- Another channel capable of communicating with both over sockets and shared memory has been added. Unlike the ssm channel which waits for new data to arrive by continuously polling the system in a busy loop, the essm channel waits by blocking on an operating system event object. This channel is experimental, and is only available for Microsoft Windows.
- The topology routines have been modified to allow the device to override the default implementation. This allows the device to export knowledge of the underlying physical topology to the MPI routines (Dims_create and the reorder == true cases in Cart_create and Graph_create).
- New memory allocation macros, MPIU_CHK[PL]MEM_*(), have been added to help prevent memory leaks. See mpich2/src/include/mpimem.h.
- New error reporting macros, MPIU_ERR_*, have been added to simplify the error handling throughout the code, making the code easier to read. See mpich2/src/include/mpierrs.h.
- Interprocess communication using the Sock interface (sock and ssm channels) may now be bound to a particular destination interface using the environment variable MPICH_INTERFACE_HOSTNAME. The variable needs to be set for each process for which the destination interface is not the default interface. (Other mechanisms for destination interface selection will be provided in future releases.) Both MPD and SMPD provide a more simplistic mechanism for specifying the interface. See the user documentation.
- Too many bug fixes to describe. Much thanks goes to the users who reported bugs. Their patience and understanding as we attempted to recreate the problems and solve them is greatly appreciated.

=====
Changes in 1.0
=====

- MPICH2 now works on Solaris.
- The User's Guide has been expanded considerably. The Installation Guide has been expanded some as well.
- MPI_COMM_JOIN has been implemented; although like the other dynamic process routines, it is only supported by the Sock channel.

- MPI_COMM_CONNECT and MPI_COMM_ACCEPT are now allowed to connect with remote process to which they are already connected.

- Shared libraries can now be built (and used) on IA32 Linux with the GNU compilers (`--enable-sharedlibs=gcc`), and on Solaris with the native Sun Workshop compilers (`--enable-sharedlibs=solaris`). They may also work on other operating systems with GCC, but that has not been tested. Previous restrictions disallowing C++ and Fortran bindings when building shared libraries have been removed.

- The dataloop and datatype contents code has been improved to address alignment issues on all platforms.

- A bug in the datatype code, which handled zero block length cases incorrectly, has been fixed.

- An segmentation fault in the datatype memory management, resulting from freeing memory twice, has been fixed.

- The following changes were made to the MPD process manager:
 - MPI_SPAWN_MULTIPLE now works with MPD.

 - The arguments to the 'mpixec' command supplied by the MPD have changed. First, the `-default` option has been removed. Second, more flexible ways to pass environment variables have been added.

 - The commands 'mpdcheck' and 'testconfig' have been to installations using MPD. These commands test the setup of the machines on which you wish to run MPICH2 jobs. They help to identify misconfiguration, firewall issues, and other communication problems.

 - Support for MPI_APPNUM and MPI_UNIVERSE_SIZE has been added to the Simple implementation of PMI and the MPD process manager.

 - In general, error detection and recovery in MPD has improved.

- A new process manager, gforker, is now available. Like the forker process manager, gforker spawns processes using `fork()`, and thus is quite useful on SMPs machines. However, unlike forker, gforker supports all of the features of a standard mpixec, plus some. Therefore, It should be used in place of the previous forker process manager, which is now deprecated.

- The following changes were made to ROMIO:
 - The amount of duplicated ROMIO code in the `close`, `resize`, `preallocate`, `read`, `write`, asynchronous I/O, and `sync` routines has been substantially reduced.

 - A bug in flattening code, triggered by nested datatypes, has been fixed.

 - Some small memory leaks have been fixed.

 - The error handling has been abstracted allowing different MPI

Appendix F. MPICH-3 Release Information

implementations to handle and report error conditions in their own way. Using this abstraction, the error handling routines have been made consistent with rest of MPICH2.

- AIO support has been cleaned up and unified. It now works correctly on Linux, and is properly detected on old versions of AIX.
- A bug in MPI_File_seek code, and underlying support code, has been fixed.
- Support for PVFS2 has improved.
- Several dead file systems have been removed. Others, including HFS, SFS, PIOFS, and Paragon, have been deprecated.
- MPE and CLOG have been updated to version 2.1. For more details, please see src/mpe2/README.
- New macros for memory management were added to support function local allocations (alloca), to rollback pending allocations when error conditions are detected to avoid memory leaks, and to improve the conciseness of code performing memory allocations.
- New error handling macros were added to make internal error handling code more concise.

=====
Changes in 0.971
=====

- Code restricted by copyrights less flexible than the one described in the COPYRIGHT file has been removed.
- Installation and User Guides have been added.
- The SMPD PMI Wire Protocol Reference Manual has been updated.
- To eliminate portability problems, common blocks in mpif.h that spanned multiple lines were broken up into multiple common blocks each described on a single line.
- A new command, mpich2version, was added to allow the user to obtain information about the MPICH2 installation. This command is currently a simple shell script. We anticipate that the mpich2version command will eventually provide additional information such as the patches applied and the date of the release.
- The following changes were made to MPD2:
 - Support was added for MPI's "singleton init", in which a single process started in the normal way (i.e., not by mpiexec or mpirun) becomes an MPI process with an MPI_COMM_WORLD of size one by calling MPI_Init. After this the process can call other MPI functions, including MPI_Comm_spawn.
 - The format for some of the arguments to mpiexec have changed,

- especially for passing environment variables to MPI processes.
- In addition to miscellaneous hardening, better error checking and messages have been added.
 - The install process has been improved. In particular, configure has been updated to check for a working install program and supply it's own installation script (install.sh) if necessary.
 - A new program, mpdcheck, has been added to help diagnose machine configurations that might be erroneous or at least confusing to mpd.
 - Runtime version checking has been added to insure that the Simple implementation of PMI linked into the application and the MPD process manager being used to run that application are compatible.
 - Minor improvements have been made to mpdboot.
 - Support for the (now deprecated) BNR interface has been added to allow MPICH1 programs to also be run via MPD2.
 - Shared libraries are now supported on Linux systems using the GNU compilers with the caveat that C++ support must be disabled (--disable-cxx).
 - The CH3 interface and device now provide a mechanism for using RDMA (remote direct memory access) to transfer data between processes.
 - Logging capabilities for MPI and internal routines have been readded. See the documentation in doc/logging for details.
 - A "meminit" option was added to --enable-g to force all bytes associated with a structure or union to be initialized prior to use. This prevents programs like Valgrind from complaining about uninitialized accesses.
 - The dist-with-version and snap targets in the top-level Makefile.sm now properly produce mpich2-<ver>/maint/Version instead of mpich2-<ver>/Version. In addition, they now properly update the VERSION variable in Makefile.sm without clobbering the sed line that performs the update.
 - The dist and snap targets in the top-level Makefile.sm now both use the dist-with-version target to avoid inconsistencies.
 - The following changes were made to simplemake:
 - The environment variables DEBUG, DEBUG_DIRS, and DEBUG_CONFDIR can now be used to control debugging output.
 - Many fixes were made to make simplemake so that it would run cleanly with perl -w.
 - Installation of *all* files from a directory is now possible (example, installing all of the man pages).
 - The clean targets now remove the cache files produced by newer versions of

Appendix F. MPICH-3 Release Information

autoconf.

- For files that are created by configure, the determination of the location of that configure has been improved, so that make of those files (e.g., make Makefile) is more likely to work. There is still more to do here.
- Short loops over subdirectories are now unrolled.
- The maintainerclean target has been renamed to maintainer-clean to match GNU guidelines.
- The distclean and maintainer-clean targets have been improved.
- An option was added to perform one ar command per directory instead of one per file when creating the profiling version of routines (needed only for systems that do not support weak symbols).

```
=====  
Changes in 0.97  
=====
```

- MPI-2 one-sided communication has been implemented in the CH3 device.
- mpigdb works as a simple parallel debugger for MPI programs started with mpd. New since MPICH1 is the ability to attach to running parallel programs. See the README in mpich2/src/pm/mpd for details.
- MPI_Type_create_darray() and MPI_Type_create_subarray() implemented including the right contents and envelope data.
- ROMIO flattening code now supports subarray and darray combiners.
- Improve scalability and performance of some ROMIO PVFS and PVFS2 routines
- An error message string parameter was added to MPID_Abort(). If the parameter is non-NULL this string will be used as the message with the abort output. Otherwise, the output message will be base on the error message associated with the mpi_errno parameter.
- MPID_Segment_init() now takes an additional boolean parameter that specifies if the segment processing code is to produce/consume homogeneous (FALSE) or heterogeneous (TRUE) data.
- The definitions of MPID_VCR and MPID_VCRT are now defined by the device.
- The semantics of MPID_Progress_{Start,Wait,End}() have changed. A typical blocking progress loop now looks like the following.

```
if (req->cc != 0)  
{  
    MPID_Progress_state progress_state;  
  
    MPID_Progress_start(&progress_state);
```

```

while (req->cc != 0)
{
    mpi_errno = MPID_Progress_wait(&progress_state);
    if (mpi_errno != MPI_SUCCESS)
    {
        /* --BEGIN ERROR HANDLING-- */
        MPID_Progress_end(&progress_state);
        goto fn_fail;
        /* --END ERROR HANDLING-- */
    }
}
MPID_Progress_end(&progress_state);
}

```

NOTE: each of these routines now takes a single parameter, a pointer to a thread local state variable.

- The CH3 device and interface have been modified to better support `MPI_COMM_{SPAWN, SPAWN_MULTIPLE, CONNECT, ACCEPT, DISCONNECT}`. Channels writers will notice the following. (This is still a work in progress. See the note below.)
 - The introduction of a process group object (`MPIDI_PG_t`) and a new set of routines to manipulate that object.
 - The renaming of the `MPIDI_VC` object to `MPIDI_VC_t` to make it more consistent with the naming of other objects in the device.
 - The process group information in the `MPIDI_VC_t` moved from the channel specific portion to the device layer.
 - `MPIDI_CH3_Connection_terminate()` was added to the CH3 interface to allow the channel to properly shutdown a connection before the device deletes all associated data structures.
 - A new upcall routine, `MPIDI_CH3_Handle_connection()`, was added to allow the device to notify the device when a connection related event has completed. A present the only event is `MPIDI_CH3_VC_EVENT_TERMINATED`, which notify the device that the underlying connection associated with a VC has been properly shutdown. For every call to `MPIDI_CH3_Connection_terminate()` that the device makes, the channel must make a corresponding upcall to `MPIDI_CH3_Handle_connection()`. `MPID_Finalize()` will likely hang if this rule is not followed.
 - `MPIDI_CH3_Get_parent_port()` was added to provide `MPID_Init()` with the port name of the the parent (spawner). This port name is used by `MPID_Init()` and `MPID_Comm_connect()` to create an intercommunicator between the parent (spawner) and child (spawnee). Eventually, `MPID_Comm_spawn_multiple()` will be update to perform the reverse logic; however, the logic is presently still in the sock channel.

Note: the changes noted are relatively fresh and are the beginning to a set of future changes. The goal is to minimize the amount of code required by a channel to support MPI dynamic process functionality. As such, portions of the device will change dramatically in a future release. A few more changes

Appendix F. MPICH-3 Release Information

to the CH3 interface are also quite likely.

- MPIDI_CH3_{iRead,iWrite}() have been removed from the CH3 interface. MPIDI_CH3U_Handle_recv_pkt() now returns a receive request with a populated iovec to receive data associated with the request. MPIDU_CH3U_Handle_{recv,send}_req() reload the iovec in the request and return and set the complete argument to TRUE if more data is to read or written. If data transfer for the request is complete, the complete argument must be set to FALSE.

=====
Changes in 0.96p2
=====

The shm and ssm channels have been added back into the distribution. Officially, these channels are supported only on x86 platforms using the gcc compiler. The necessary assembly instructions to guarantee proper ordering of memory operations are lacking for other platforms and compilers. That said, we have seen a high success rate when testing these channels on unsupported systems.

This patch release also includes a new unsupported channel. The scalable shared memory, or sshm, channel is similar to the shm channel except that it allocates shared memory communication queues only when necessary instead of preallocating N-squared queues.

=====
Changes in 0.96p1
=====

This patch release fixes a problem with building MPICH2 on Microsoft Windows platforms. It also corrects a serious bug in the poll implementation of the Sock interface.

=====
Changes in 0.96
=====

The 0.96 distribution is largely a bug fix release. In addition to the many bug fixes, major improvements have been made to the code that supports the dynamic process management routines (MPI_Comm_{connect,accept,spawn,...}()). Additional changes are still required to support MPI_Comm_disconnect().

We also added an experimental (and thus completely unsupported) rdma device. The internal interface is similar to the CH3 interface except that it contains a couple of extra routines to inform the device about data transfers using the rendezvous protocol. The channel can use this extra information to pin memory and perform a zero-copy transfer. If all goes well, the results will be rolled back into the CH3 device.

Due to last minute difficulties, this release does not contain the shm or ssm channels. These channels will be included in a subsequent patch release.


```
=====
Changes in 0.94
=====
```

Active target one-sided communication is now available for the `ch3:sock` channel. This new functionality has undergone some correctness testing but has not been optimized in terms of performance. Future release will include performance enhancements, passive target communication, and availability in channels other than just `ch3:sock`.

The shared memory channel (`ch3:shm`), which performs communication using shared memory on a single machine, is now complete and has been extensively tested. At present, this channel only supports IA32 based machines (excluding the Pentium Pro which has a memory ordering bug). In addition, this channel must be compiled with `gcc`. Future releases will support additional architectures and compilers.

A new channel has been added that performs inter-node communication using sockets (TCP/IP) and intra-node communication using shared memory. This channel, `ch3:ssm`, is ideal for clusters of SMPs. Like the shared memory channel (`ch3:shm`), this channel only supports IA32 based machines and must be compiled with `gcc`. In future releases, the `ch3:ssm` channel will support additional architectures and compilers.

The two channels that perform commutation using shared memory, `ch3:shm` and `ch3:ssm`, now support the allocation of shared memory using both the POSIX and System V interfaces. The POSIX interface will be used if available; otherwise, the System V interface is used.

In the interest of increasing portability, many enhancements have been made to both the code and the configure scripts.

And, as always, many bugs have been fixed :-).

***** INTERFACE CHANGES *****

The parameters to `MPID_Abort()` have changed. `MPID_Abort()` now takes a pointer to communicator object, an MPI error code, and an exit code.

`MPIDI_CH3_Progress()` has been split into two functions:
`MPIDI_CH3_Progress_wait()` and `MPIDI_CH3_Progress_test()`.

```
=====
Changes in 0.93
=====
```

Version 0.93 has undergone extensive changes to provide better error reporting. Part of these changes involved modifications to the ADI3 and CH3 interfaces. The following routines now return MPI error codes:

`MPID_Cancel_send()`

Appendix F. MPICH-3 Release Information

```
MPID_Cancel_recv()
MPID_Progress_poke()
MPID_Progress_test()
MPID_Progress_wait()
MPIDI_CH3_Cancel_send()
MPIDI_CH3_Progress()
MPIDI_CH3_Progress_poke()
MPIDI_CH3_iRead()
MPIDI_CH3_iSend()
MPIDI_CH3_iSendv()
MPIDI_CH3_iStartmsg()
MPIDI_CH3_iStartmsgv()
MPIDI_CH3_iWrite()
MPIDI_CH3U_Handle_recv_pkt()
MPIDI_CH3U_Handle_recv_req()
MPIDI_CH3U_Handle_send_req()
```

```
*****
Of special note are MPID_Progress_test(), MPID_Progress_wait() and
MPIDI_CH3_Progress() which previously returned an integer value indicating if
one or more requests had completed. They no longer return this value and
instead return an MPI error code (also an integer). The implication being that
while the semantics changed, the type signatures did not.
*****
```

The function used to create error codes, `MPIR_Err_create_code()`, has also changed. It now takes additional parameters, allowing it to create a stack of errors and making it possible for the reporting function to indicate in which function and on which line the error occurred. It also allows an error to be designated as fatal or recoverable. Fatal errors always result in program termination regardless of the error handler installed by the application.

A RDMA channel has been added and includes communication methods for shared memory and shmem. This is recent development and the RDMA interface is still in flux.

Release Notes

KNOWN ISSUES

Fine-grained thread safety

- * `ch3:sock` does not (and will not) support fine-grained threading.
- * MPI-IO APIs are not currently thread-safe when using fine-grained threading (`--enable-thread-cs=per-object`).
- * `ch3:nemesis:tcp` fine-grained threading is still experimental and may have correctness or performance issues. Known correctness issues

include dynamic process support and generalized request support.

Lacking channel-specific features

- * ch3 does not presently support communication across heterogeneous platforms (e.g., a big-endian machine communicating with a little-endian machine).
- * ch3:nemesis:mx does not support dynamic processes at this time.
- * Support for "external32" data representation is incomplete. This affects the MPI_Pack_external and MPI_Unpack_external routines, as well the external data representation capabilities of ROMIO. In particular: noncontiguous user buffers could consume egregious amounts of memory in the MPI library and any types which vary in width between the native representation and the external32 representation will likely cause corruption. The following ticket contains some additional information:

<http://trac.mpich.org/projects/mpich/ticket/1754>

- * ch3 has known problems in some cases when threading and dynamic processes are used together on communicators of size greater than one.

Process Managers

- * Hydra has a bug related to stdin handling:

<https://trac.mpich.org/projects/mpich/ticket/1782>

Performance issues

- * SMP-aware collectives do not perform as well, in select cases, as non-SMP-aware collectives, e.g. MPI_Reduce with message sizes larger than 64KiB. These can be disabled by the configure option "--disable-mpcoll".
- * MPI_Irecv operations that are not explicitly completed before MPI_Finalize is called may fail to complete before MPI_Finalize returns, and thus never complete. Furthermore, any matching send operations may erroneously fail. By explicitly completed, we mean that the request associated with the operation is completed by one of the MPI_Test or MPI_Wait routines.

Notes

1. <http://www.mpich.org/>
2. <http://www.mpich.org/documentation/guides/>

Appendix G. SLURM Release Information

The following is reproduced essentially verbatim from files contained within the SLURM tarball downloaded from <http://slurm.schedmd.com/>

SLURM was produced at Lawrence Livermore National Laboratory in collaboration with various organizations.

Copyright (C) 2012-2013 Los Alamos National Security, LLC.
Copyright (C) 2011 Trinity Centre for High Performance Computing
Copyright (C) 2010-2015 SchedMD LLC
Copyright (C) 2009-2013 CEA/DAM/DIF
Copyright (C) 2009-2011 Centro Svizzero di Calcolo Scientifico (CSCS)
Copyright (C) 2008-2011 Lawrence Livermore National Security
Copyright (C) 2008 Vijay Ramasubramanian
Copyright (C) 2007-2008 Red Hat, Inc.
Copyright (C) 2007-2013 National University of Defense Technology, China
Copyright (C) 2007-2015 Bull
Copyright (C) 2005-2008 Hewlett-Packard Development Company, L.P.
Copyright (C) 2004-2009, Marcus Holland-Moritz
Copyright (C) 2002-2007 The Regents of the University of California
Copyright (C) 2002-2003 Linux NetworX
Copyright (C) 2002 University of Chicago
Copyright (C) 2001, Paul Marquess
Copyright (C) 2000 Markus Friedl
Copyright (C) 1999, Kenneth Albanowski
Copyright (C) 1998 Todd C. Miller <Todd.Miller@courtesan.com>
Copyright (C) 1996-2003 Maximum Entropy Data Consultants Ltd,
Copyright (C) 1995 Tatu Ylonen <ylo@cs.hut.fi>, Espoo, Finland
Copyright (C) 1989-1994, 1996-1999, 2001 Free Software Foundation, Inc.
Many other organizations contributed code and/or documentation without including a copyright notice.

Written by:

Amjad Majid Ali (Colorado State University)
Par Andersson (National Supercomputer Centre, Sweden)
Don Albert (Bull)
Ernest Artiaga (Barcelona Supercomputer Center, Spain)
Danny Auble (LLNL, SchedMD LLC)
Susanne Balle (HP)
Anton Blanchard (Samba)
Janne Blomqvist (Aalto University, Finland)
David Bremer (LLNL)
Jon Bringham (LANL)
Bill Brophy (Bull)
Hongjia Cao (National University of Defense Technology, China)
Daniel Christians (HP)
Gilles Civario (Bull)
Chuck Clouston (Bull)
Joseph Donaghy (LLNL)
Chris Dunlap (LLNL)
Joey Ekstrom (LLNL/Brigham Young University)
Josh England (TGS Management Corporation)
Kent Engstrom (National Supercomputer Centre, Sweden)
Jim Garlick (LLNL)

Appendix G. SLURM Release Information

Didier Gazen (Laboratoire d'Aerologie, France)
Raphael Geissert (Debian)
Yiannis Georgiou (Bull)
Andriy Grytsenko (Massive Solutions Limited, Ukraine)
Mark Grondona (LLNL)
Takao Hatazaki (HP, Japan)
Matthieu Hautreux (CEA, France)
Chris Holmes (HP)
David Hoppner
Nathan Huff (North Dakota State University)
David Jackson (Adaptive Computing)
Morris Jette (LLNL, SchedMD LLC)
Klaus Joas (University Karlsruhe, Germany)
Greg Johnson (LANL)
Jason King (LLNL)
Aaron Knister (Environmental Protection Agency)
Nancy Kritkauskay (Bull)
Roman Kurakin (Institute of Natural Science and Ecology, Russia)
Eric Lin (Bull)
Don Lipari (LLNL)
Puenlap Lee (Bull)
Dennis Leepow
Bernard Li (Genome Sciences Centre, Canada)
Donald Lipari (LLNL)
Steven McDougall (SiCortex)
Donna Mecozzi (LLNL)
Bjorn-Helge Mevik (University of Oslo, Norway)
Chris Morrone (LLNL)
Pere Munt (Barcelona Supercomputer Center, Spain)
Michal Novotny (Masaryk University, Czech Republic)
Bryan O'Sullivan (Pathscale)
Gennaro Oliva (Institute of High Performance Computing and Networking, Italy)
Alejandro Lucero Palau (Barcelona Supercomputer Center, Spain)
Daniel Palermo (HP)
Dan Phung (LLNL/Columbia University)
Ashley Pittman (Quadrics, UK)
Vijay Ramasubramanian (University of Maryland)
Krishnakumar Ravi[KK] (HP)
Petter Reinholdtsen (University of Oslo, Norway)
Gerrit Renker (Swiss National Computer Centre)
Andy Riebs (HP)
Asier Roa (Barcelona Supercomputer Center, Spain)
Miguel Ros (Barcelona Supercomputer Center, Spain)
Beat Rubischon (DALCO AG, Switzerland)
Dan Rusak (Bull)
Eygene Ryabinkin (Kurchatov Institute, Russia)
Federico Sacerdoti (D.E. Shaw)
Rod Schultz (Bull)
Tyler Strickland (University of Florida)
Jeff Squyres (LAM MPI)
Prashanth Tamraparni (HP, India)
Jimmy Tang (Trinity College, Ireland)
Kevin Tew (LLNL/Brigham Young University)
Adam Todorski (Rensselaer Polytechnic Institute)
Nathan Weeks (Iowa State University)

Tim Wickberg (Rensselaer Polytechnic Institute)
Ramiro Brito Willmersdorf (Universidade Federal de Pernambuco, Brazil)
Jay Windley (Linux NetworX)
Anne-Marie Wunderlin (Bull)

CODE-OCEC-09-009. All rights reserved.

This file is part of SLURM, a resource management program.
For details, see <<http://slurm.schedmd.com/>>.
Please also read the supplied file: DISCLAIMER.

SLURM is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

SLURM is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with SLURM; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

OUR NOTICE AND TERMS OF AND CONDITIONS OF THE GNU GENERAL PUBLIC LICENSE

Our Preamble Notice

Auspices

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Disclaimer

This work was sponsored by an agency of the United States government. Neither the United States Government nor Lawrence Livermore National Security, LLC, nor any of their employees, makes any warranty, express or implied, or assumes any liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. References herein to any specific commercial products, process, or services by trade names, trademark, manufacturer or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

The precise terms and conditions for copying, distribution and modification is provided in the file named "COPYING" in this directory.

Appendix G. SLURM Release Information

This file describes changes in recent versions of Slurm. It primarily documents those changes that are of interest to users and administrators.

* Changes in Slurm 17.02.0

=====

- job_submit/lua - Make "immediate" parameter available.
- Fix srun I/O race condition to eliminate an error message that might be generated if the application exits with outstanding stdin.
- Fix regression when purging/archiving jobs/events.
- Add new job state JOB_OOM indicating Out Of Memory condition as detected by task/cgroup plugin.
- If QOS has been added to the system go refigure out Deny/AllowQOS on partitions.
- Deny job with duplicate GRES requested.
- Fix loading super old assoc_mgr usage without segfaulting.
- CRAY systems: Restore TaskPlugins order of task/cray before task/cgroup.
- Task/cray: Treat missing "mems" cgroup with "debug" messages rather than "error" messages. The file may be missing at step termination due to a change in how cgroups are released at job/step end.
- Fix for job constraint specification with counts, --ntasks-per-node value, and no node count.
- Fix ordering of step task allocation to fill in a socket before going into another one.
- Fix configure to not require C++
- job_submit/lua - Remove access to slurmctld internal reservation fields of job_pend_cnt and job_run_cnt.
- Prevent job_time_limit enforcement from blocking other internal operations if a large number of jobs need to be cancelled.
- Add 'preempt_youngest_order' option to preempt/partition_prio plugin.
- Fix controller being able to talk to a pre-released DBD.
- Added ability to override the invoking uid for "scontrol update job" by specifying "--uid=<uid>|-u <uid>".
- Changed file broadcast "offset" from 32 to 64 bits in order to support files over 2 GB.
- slurm.spec - do not install init scripts alongside systemd service files.

* Changes in Slurm 17.02.0rc1

=====

- Add port info to 'sinfo' and 'scontrol show node'.
- Fix errant definition of USE_64BIT_BITSTR which can lead to core dumps.
- Move BatchScript to end of each job's information when using "scontrol -dd show job" to make it more readable.
- Add SchedulerParameters configuration parameter of "default_gbytes", which treats numeric only (no suffix) value for memory and tmp disk space as being in units of Gigabytes. Mostly for compatibility with LSF.
- Fix race condition in srun/sattach logic which would prevent srun from terminating.
- Bitstring operations are now 64bit instead of 32bit.
- Replace hweight() function in bitstring with faster version.
- scancel would treat a non-numeric argument as the name of jobs to be cancelled (a non-documented feature). Cancelling jobs by name now require the "--jobname=" command line argument.
- scancel modified to note that no jobs satisfy the filter options when the --verbose option is used along with one or more job filters (e.g. "--qos=").


```
-- Change _pack_cred to use pack_bit_str_hex instead of pack_bit_fmt for
better scalability and performance.
-- Add BootTime configuration parameter to knl.conf file to optimize resource
allocations with respect to required node reboots.
-- Add node_features_p_boot_time() to node_features plugin to optimize
scheduling with respect to node reboots.
-- Avoid allocating resources to a job in the event that its run time plus boot
time (if needed) extent into an advanced reservation.
-- Burst_buffer/cray - Avoid stage-out operation if job never started.
-- node_features/knl_cray - Add capability to detected Uncorrectable Memory
Errors (UME) and if detected then log the event in all job and step stderr
with a message of the form:
error: *** STEP 1.2 ON tux1 UNCORRECTABLE MEMORY ERROR AT 2016-12-14T09:09:37 ***
Similar logic added to node_features/knl_generic in version 17.02.0pre4.
-- If job is allocated nodes which are powered down, then reset job start time
when the nodes are ready and do not charge the job for power up time.
-- Add the ability to purge transactions from the database.
-- Add support for requeue'ing of federated jobs (BETA).
-- Add support for interactive federated jobs (BETA).
-- Add the ability to purge rolled up usage from the database.
-- Properly set SLURM_JOB_GPUS environment variable for Prolog.
```

* Changes in Slurm 17.02.0pre4

=====

```
-- Add support for per-partition OverTimeLimit configuration.
-- Add --mem_bind option of "sort" to run zonesort on KNL nodes at step start.
-- Add LaunchParameters=mem_sort option to configure running of zonesort
by default at step startup.
-- Add "FreeSpace" information for each pool to the "scontrol show burstbuffer"
output. Required changes to the burst_buffer_info_t data structure.
-- Add new node state flag of NODE_STATE_REBOOT for node reboots triggered by
"scontrol reboot" commands. Previous logic re-used NODE_STATE_MAINT flag,
which could lead to inconsistencies. Add "ASAP" option to "scontrol reboot"
command that will drain a node in order to reboot it as soon as possible,
then return it to service.
-- Allow unit conversion routine to convert 1024M to 1G.
-- switch/cray plugin - change legacy spool directory location.
-- Add new PriorityFlags option of INCR_ONLY, which prevents a job's priority
from being decremented.
-- Make it so we don't purge job start messages until after we purge step
messages. Hopefully this will reduce the number of messages lost when
filling up memory when the database/DBD is down.
-- Added SchedulingParameters option of "bf_job_part_count_reserve". Jobs below
the specified threshold will not have resources reserved for them.
-- If GRES are configured with file IDs, then "scontrol -d show node" will
not only identify the count of currently allocated GRES, but their specific
index numbers (e.g. "GresUsed=gpu:alpha:2(IDX:0,2),gpu:beta:0(IDX:N/A)").
Ditto for job information with "scontrol -d show job".
-- Add new mcs/account plugin.
-- Add "GresEnforceBind=Yes" to "scontrol show job" output if so configured.
-- Add support for SALLOC_CONSTRAINT, SBATCH_CONSTRAINT and SLURM_CONSTRAINT
environment variables to set default constraints for salloc, sbatch and
srun commands respectively.
-- Provide limited support for the MemSpecLimit configuration parameter without
the task/cgroup plugin.
```

Appendix G. SLURM Release Information

- node_features/knl_generic - Add capability to detected Uncorrectable Memory Errors (UME) and if detected then log the event in all job and step stderr with a message of the form:
error: *** STEP 1.2 ON tux1 UNCORRECTABLE MEMORY ERROR AT 2016-12-14T09:09:37 ***
- Add SLURM_JOB_GID to TaskProlog environment.
- burst_buffer/cray - Remove leading zeros from node ID lists passed to dw_wlm_cli program.
- Add "Partitions" field to "scontrol show node" output.
- Remove sched/wiki and sched/wiki2 plugins and associated code.
- Remove SchedulerRootFilter option and slurm_get_root_filter() API call.
- Add SchedulerParameters option of spec_cores_first to select specialized cores from the lowest rather than highest number cores and sockets.
- Add PrologFlags option of Serial to disable concurrent launch of Prolog and Epilog scripts.
- Fix security issue caused by insecure file path handling triggered by the failure of a Prolog script. To exploit this a user needs to anticipate or cause the Prolog to fail for their job. CVE-2016-10030.

* Changes in Slurm 17.02.0pre3

=====

- Add srun host & PID to job step data structures.
- Avoid creating duplicate pending step records for the same srun command.
- Rewrite srun's logic for pending steps for better efficiency (fewer RPCs).
- Added new SchedulerParameters options step_retry_count and step_retry_time to control scheduling behaviour of job steps waiting for resources.
- Optimize resource allocation logic for --spread-job job option.
- Modify cpu_bind and mem_bind map and mask options to accept a repetition count to better support large task count. For example:
"mask_mem:0x0f*2,0xf0*2" is equivalent to "mask_mem:0x0f,0x0f,0xf0,0xf0".
- Add support for --mem_bind=prefer option to prefer, but not restrict memory use to the identified NUMA node.
- Add mechanism to constrain kernel memory allocation using cgroups. New cgroup.conf parameters added: ConstrainKmemSpace, MaxKmemPercent, and MinKmemSpace.
- Correct invocation of man2html, which previously could cause FreeBSD builds to hang.
- MYSQL - Unconditionally remove 'ignore' clause from 'alter ignore'.
- Modify service files to not start Slurm daemons until after Munge has been started.
NOTE: If you are not using Munge, but are using the "service" scripts to start Slurm daemons, then you will need to remove this check from the etc/slurm*service scripts.
- Do not process SALLOC_HINT, SBATCH_HINT or SLURM_HINT environment variables if any of the following salloc, sbatch or srun command line options are specified: -B, --cpu_bind, --hint, --ntasks-per-core, or --threads-per-core.
- burst_buffer/cray: Accept new jobs on backup slurmctld daemon without access to dw_wlm_cli command. No burst buffer actions will take place.
- Do not include SLURM_JOB_DERIVED_EC, SLURM_JOB_EXIT_CODE, or SLURM_JOB_EXIT_CODE in PrologSlurmctld environment (not available yet).
- Cray - set task plugin to fatal() if task/cgroup is not loaded after task/cray in the TaskPlugin settings.
- Remove separate slurm_blcr package. If Slurm is built with BLCR support, the files will now be part of the main Slurm packages.
- Replace sjstat, seff and sjobexit RPM packages with a single "contribs" package.

- Remove long since defunct slurmdb-direct scripts.
- Add SbcastParameters configuration option to control default file destination directory and compression algorithm.
- Add new SchedulerParameter (max_array_tasks) to limit the maximum number of tasks in a job array independently from the maximum task ID (MaxArraySize).
- Fix issue where number of nodes is not properly allocated when sbatch and salloc are requested with -n tasks < hosts from -w hostlist or from -N.
- Add infrastructure for submitting federated jobs.

* Changes in Slurm 17.02.0pre2

=====

- Add new RPC (REQUEST_EVENT_LOG) so that slurmd and slurmstepd can log events through the slurmctld daemon.
- Remove sbatch --bb option. That option was never supported.
- Automatically clean up task/cgroup cpuset and devices cgroups after steps are completed.
- Add federation read/write locks.
- Limit job purge run time to 1 second at a time.
- The database index for jobs is now 64 bits. If you happen to be close to 4 billion jobs in your database you will want to update your slurmctld at the same time as your slurmdbd to prevent roll over of this variable as it is 32 bit previous versions of Slurm.
- Optionally lock slurmstepd in memory for performance reasons and to avoid possible SIGBUS if the daemon is paged out at the time of a Slurm upgrade (changing plugins). Controlled via new LaunchParameters options of slurmstepd_memlock and slurmstepd_memlock_all.
- Add event trigger on burst buffer errors (see strigger man page, --burst_buffer option).
- Add job AdminComment field which can only be set by a Slurm administrator.
- Add salloc, sbatch and srun option of --delay-boot=<time>, which will temporarily delay booting nodes into the desired state for a job in the hope of using nodes already in the proper state which will be available at a later time.
- Add job burst_buffer_state and delay_boot fields to scontrol and queue output. Also add ability to modify delay_boot from scontrol.
- Fix for node's available TRES array getting filled in with configured GRES model types.
- Log if job --bb option contains any unrecognized content.
- Display configured and allocated TRES for nodes in scontrol show nodes.
- Change all memory values (in MB) to uint64_t to accommodate > 2TB per node.
- Add MailDomain configuration parameter to qualify email addresses.
- Refactor the persistent connections within the federation code to use the same logic that was found in the slurmdbd. Now both functionalities share the same code.
- Remove BlueGene/L and BlueGene/P support.
- Add "flag" field to launch_tasks_request_msg. Remove the following fields (moved into flags): multi_prog, task_flags, user_managed_io, pty, buffered_stdio, and labelio.
- Add protocol version to slurmd startup communications for slurmstepd to permit changes in the protocol.

* Changes in Slurm 17.02.0pre1

=====

- burst_buffer/cray - Add support for rounding up the size of a buffer request if the DataWarp configuration "equalize_fragments" is used.

Appendix G. SLURM Release Information

- Remove AIX support.
- Rename "in" to "input" in slurm_step_io_fds data structure defined in slurm.h. This is needed to avoid breaking Python with by using one of its keywords in a Slurm data structure.
- Remove eligible_time from jobcomp/elasticsearch.
- Enable the deletion of a QOS, even if no clusters have been added to the database.
- SlurmDBD - change all timestamps to bigint from int to solve Y2038 problem.
- Add salloc/sbatch/srun --spread-job option to distribute tasks over as many nodes as possible. This also treats the --ntasks-per-node option as a maximum value.
- Add ConstrainKmemSpace to cgroup.conf, defaulting to yes, to allow cgroup Kmem enforcement to be disabled while still using ConstrainRAMSpace.
- Add support for sbatch --bbf option to specify a burst buffer input file.
- Added burst buffer support for job arrays. Add new SchedulerParameters configuration parameter of bb_array_stage_cnt=# to indicate how many pending tasks of a job array should be made available for burst buffer resource allocation.
- Fix small memory leak when a job fails to load from state save.
- Fix invalid read when attempting to delete clusters from database with running jobs.
- Fix small memory leak when deleting clusters from database.
- Add SLURM_ARRAY_TASK_COUNT environment variable. Total number of tasks in a job array (e.g. "--array=2,4,8" will set SLURM_ARRAY_TASK_COUNT=3).
- Add new sacctmgr commands: "shutdown" (shutdown the server), "list stats" (get server statistics) "clear stats" (clear server statistics).
- Restructure job accounting query to use 'id_job in (1, 2, ..)' format instead of logically equivalent 'id_job = 1 || id_job = 2 || ..' .
- Added start_delay field to jobcomp/elasticsearch.
- In order to support federated jobs, the MaxJobID configuration parameter default value has been reduced from 2,147,418,112 to 67,043,328 and its maximum value is now 67,108,863. Upon upgrading, any pre-existing jobs that have a job ID above the new range will continue to run and new jobs will get job IDs in the new range.
- Added infrastructure for setting up federations in database and establishing connections between federation clusters.

* Changes in Slurm 16.05.10

=====

- Record job state as PREEMPTED instead of TIMEOUT when GraceTime is reached.
- task/cgroup - print warnings to stderr when --cpu_bind=verbose is enabled and the requested processor affinity cannot be set.
- power/cray - Disable power cap get and set operations on DOWN nodes.
- Jobs preempted with PreemptMode=REQUEUE were incorrectly recorded as REQUEUED in the accounting.
- PMIX - Use volatile specifier to avoid flag caching and lock the flag to make sure it is protected.
- PMIX/PMI2 - Make it possible to use %n or %h in a spool dir.
- burst_buffer/cray - Support default pool which is not the first pool reported by DataWarp and log in Slurm when pools that are added or removed from DataWarp.
- Insure job does not start running before PrologSlurmctld is complete and node is booted (all nodes for interactive job, at least first node for batch job without burst buffers).
- Fix minor memory leak in the slurmctld when removing a QOS.

```
-- burst_buffer/cray - Do not execute "pre_run" operation until after all nodes
are booted and ready for use.
-- scontrol - return an error when attempting to use the +=/++ syntax to
update a field where this is not appropriate.
-- Fix task/affinity to work correctly with --ntasks-per-socket.
-- Honor --ntasks-per-node and --ntasks option when used with job constraints
that contain node counts.
-- Prevent deadlocked slurmstepd processes due to unsafe use of regcomp with
older glibc versions.
-- Fix squeue when SLURM_BITSTR_LEN=0 is set in the user environment.
-- Fix comments in acct_policy.c to reflect actual variables instead of
old ones.
-- Fix correct variables when validating GrpTresMins on a QOS.
-- Better debug output when a job is being held because of a GrpTRES[Run]Min
limits.
-- Fix correct state reason when job can't run 'safely' because of an
association GrpWall limit.
-- Squeue always loads new data if user_id option specified
-- Fix for possible job ID parsing failure and abort.
-- If node boot in progress when slurmctld daemon is restarted, then allow
sufficient time for reboot to complete and not prematurely DOWN the node as
"Not responding".
-- For job resize, correct logic to build "resize" script with new values.
Previously the scripts were based upon the original job size.
-- Fix squeue to not limit the size of partition, burst_buffer, exec_host, or
reason to 32 chars.
-- Fix potential packing error when packing a NULL slurmdb_clus_res_rec_t.
-- Fix potential packing errors when packing a NULL slurmdb_reservation_cond_t.
```

* Changes in Slurm 16.05.9

=====

```
-- Fix parsing of SBCAST_COMPRESS environment variable in sbcast.
-- Change some debug messages to errors in task/cgroup plugin.
-- backfill scheduler: Stop trying to determine expected start time for a job
after 2 seconds of wall time. This can happen if there are many running jobs
and a pending job can not be started soon.
-- Improve performance of cr_sort_part_rows() in cons_res plugin.
-- CRAY - Fix dealock issue when updating accounting in the slurmctld and
scheduling a Datawarp job.
-- Correct the job state accounting information for jobs requeued due to burst
buffer errors.
-- burst_buffer/cray - Avoid "pre_run" operation if not using buffer (i.e.
just creating or deleting a persistent burst buffer).
-- Fix slurm.spec file support for BlueGene builds.
-- Fix missing TRES read lock in acct_policy_job_runnable_pre_select() code.
-- Fix debug2 message printing value using wrong array index in
_qos_job_runnable_post_select().
-- Prevent job timeout on node power up.
-- MYSQL - Fix minor memory leak when querying steps and the sql fails.
-- Make it so sacctmgr accepts column headers like MaxTRESPU and not MaxTRESP.
-- Only look at SLURM_STEP_KILLED_MSG_NODE_ID on startup, to avoid race
condition later when looking at a steps env.
-- Make backfill scheduler behave like regular scheduler in respect to
'assoc_limit_stop'.
-- Allow a lower version client command to talk to a higher version controller
```

Appendix G. SLURM Release Information

using the multi-cluster options (e.g. `squeue -M<cluster>`).

- `slurmctld/agent` race condition fix: Prevent job launch while `PrologSlurmctld` daemon is running or node boot in progress.
- MySQL - Fix a few other minor memory leaks when uncommon failures occur.
- `burst_buffer/cray` - Fix race condition that could cause multiple batch job launch requests resulting in drained nodes.
- Correct logic to purge old reservations.
- Fix DBD cache restore from previous versions.
- Fix to logic for getting expected start time of existing job ID with explicit begin time that is in the past.
- Clear job's reason of "BeginTime" in a more timely fashion and/or prevents them from being stuck in a PENDING state.
- Make sure acct policy limits imposed on a job are correct after requeue.

* Changes in Slurm 16.05.8

=====

- Remove `StoragePass` from being printed out in the `slurmdbd` log at `debug2` level.
- Defer `PATH` search for task program until launch in `slurmstepd`.
- Modify regression `test1.89` to avoid leaving vestigial job. Also reduce logging to reduce likelihood of `Expect` buffer overflow.
- Do not `PATH` search for multi-prog launches if `LaunchParameters=test_exec` is enabled.
- Fix for possible infinite loop in `select/cons_res` plugin when trying to satisfy a job's `ntasks_per_core` or socket specification.
- If job is held for bad constraints make it so once updated the job doesn't go into `JobAdminHeld`.
- `sched/backfill` - Fix logic to reserve resources for jobs that require a node reboot (i.e. to change KNL mode) in order to start.
- When unpacking a node or `front_end` record from state and the protocol version is lower than the min version, set it to the min.
- Remove redundant lookup for `part_ptr` when updating a reservation's nodes.
- Fix memory and file descriptor leaks in `slurmd` daemon's `sbcst` logic.
- Do not allocate specialized cores to jobs using the `--exclusive` option.
- Cancel interactive job if Prolog failure with `"PrologFlags=contain"` or `"PrologFlags=alloc"` configured. Send new error prolog failure message to the `salloc` or `srun` command as needed.
- Prevent possible out-of-bounds read in `slurmstepd` on an invalid `#!` line.
- Fix check for `PluginDir` within `slurmctld` to work with multiple directories.
- Cancel interactive jobs automatically on communication error to launching `srun/salloc` process.
- Fix security issue caused by insecure file path handling triggered by the failure of a Prolog script. To exploit this a user needs to anticipate or cause the Prolog to fail for their job. CVE-2016-10030.

* Changes in Slurm 16.05.7

=====

- Fix issue in the `priority/multifactor` plugin where on a `slurmctld` restart, where more time is accounted for than should be allowed.
- `cray/busrt_buffer` - If `total_space` in a pool decreases, reset `used_space` rather than trying to account for buffer allocations in progress.
- `cray/busrt_buffer` - Fix for double counting of `used_space` at `slurmctld` startup.
- Fix regression in 16.05.6 where if you request multiple cpus per task (`-c2`) and request `--ntasks-per-core=1` and only 1 task on the node

the slurmd would abort on an infinite loop fatal.

- cray/busrt_buffer - Internally track both allocated and unusable space. The reported UsedSpace in a pool is now the allocated space (previously was unusable space). Base available space on whichever value leaves least free space.
- cray/burst_buffer - Preserve job ID and don't translate to job array ID.
- cray/burst_buffer - Update "instance" parsing to match updated dw_wlm_cli output.
- sched/backfill - Insure we don't try to start a job that was already started and requeued by the main scheduling logic.
- job_submit/lua - add access to the job features field in job_record.
- select/linear plugin modified to better support heterogeneous clusters when topology/none is also configured.
- Permit cancellation of jobs in configuring state.
- acct_gather_energy/rapl - prevent segfault in slurmd from race to gather data at slurmd startup.
- Integrate node_feature/knl_generic with "hbm" GRES information.
- Fix output routines to prevent rounding the TRES values for memory or BB.
- switch/cray plugin - fix use after free error.
- docs - elaborate on how way to clear TRES limits in sacctmgr.
- knl_cray plugin - Avoid abort from backup slurmctld at start time.
- cgroup plugins - fix two minor memory leaks.
- If a node is booting for some job, don't allocate additional jobs to the node until the boot completes.
- testsuite - fix job id output in test17.39.
- Modify backfill algorithm to improve performance with large numbers of running jobs. Group running jobs that end in a "similar" time frame using a time window that grows exponentially rather than linearly. After one second of wall time, simulate the termination of all remaining running jobs in order to respond in a reasonable time frame.
- Fix slurm_job_cpus_allocated_str_on_node_id() API call.
- sched/backfill plugin: Make malloc match data type (defined as uint32_t and allocated as int).
- srun - prevent segfault when terminating job step before step has launched.
- sacctmgr - prevent segfault when trying to reset usage for an invalid account name.
- Make the openssl crypto plugin compile with openssl >= 1.1.
- Fix SuspendExcNodes and SuspendExcParts on slurmctld reconfiguration.
- sbcast - prevent segfault in slurmd due to race condition between file transfers from separate jobs using zlib compression
- cray/burst_buffer - Increase time to synchronize operations between threads from 5 to 60 seconds ("setup" operation time observed over 17 seconds).
- node_features/knl_cray - Fix possible race condition when changing node state that could result in old KNL mode as an active features.
- Make sure if a job can't run because of resources we also check accounting limits after the node selection to make sure it doesn't violate those limits and if it does change the reason for waiting so we don't reserve resources on jobs violating accounting limits.
- NRT - Make it so a system running against IBM's PE will work with PE version 1.3.
- NRT - Make it so protocols pgas and test are allowed to be used.
- NRT - Make it so you can have more than 1 protocol listed in MP_MSG_API.
- cray/burst_buffer - If slurmctld daemon restarts with pending job and burst buffer having unknown file stage-in status, teardown the buffer, defer the job, and start stage-in over again.

Appendix G. SLURM Release Information

- On state restore in the slurmd don't overwrite the mem_spec_limit given from the slurm.conf when using FastSchedule=0.
- Recognize a KNL's proper NUMA count (rather than setting it to the value in slurm.conf) when using FastSchedule=0.
- Fix parsing in regression test1.92 for some prompts.
- sbcast - use slurmd's gid cache rather than a separate lookup.
- slurmd - return error if setgroups() call fails in _drop_privileges().
- Remove error messages about gres counts changing when a job is resized on a slurmd restart or reconfig, as they aren't really error messages.
- Fix possible memory corruption if a job is using GRES and changing size.
- jobcomp/elasticsearch - fix printf format for a value on 32-bit builds.
- task/cgroup - Change error message if CPU binding can not take place to better identify the root cause of the problem.
- Fix issue where task/cgroup would not always honor --cpu_bind=threads.
- Fix race condition in with getgrouplist() in slurmd that can lead to user accounts being granted access to incorrect group memberships during job launch.

* Changes in Slurm 16.05.6

=====

- Docs - the correct default value for GroupUpdateForce is 0.
- mpi/pmix - improve point to point communication performance.
- SlurmDB - include pending jobs in search during 'sacctmgr show runawayjobs'.
- Add client side out-of-range checks to --nice flag.
- Fix support for sbatch "-W" option, previously eeded to use "--wait".
- node_features/knl_cray plugin and capmc_suspend/resume programs modified to sleep and retry capmc operations if the Cray State Manager is down. Added CapmcRetries configuration parameter to knl_cray.conf.
- node_features/knl_cray plugin: Remove any KNL MCDRAM or NUMA features from node's configuration if capmc does NOT report the node as being KNL.
- node_features/knl_cray plugin: drain any node not reported by "capmc node_status" on startup or reconfig.
- node_features/knl_cray plugin: Substantially streamline and speed up logic to load current node state on reconfigure failure or unexpected node boot.
- node_features/knl_cray plugin: Add separate thread to interact with capmc in response to unexpected node reboots.
- node_features plugin - Add "mode" argument to node_features_p_node_xlate() function to fix some bugs updating a node's features using the node update RPC.
- node_features/knl_cray plugin: If the reconfiguration of nodes for an interactive job fails, kill the job (it can't be requeued like a batch job).
- Testsuite - Added srun/salloc/sbatch tests with --use-min-nodes option.
- Fix typo when an error occurs when discovering pmix version on configure.
- Fix configuring pmix support when you have your lib dir symlinked to lib64.
- Fix waiting reason if a job is waiting for a specific limit instead of always just AccountingPolicy.
- Correct SchedulerParameters=bf_busy_nodes logic with respect to the job's minimum node count. Previous logic would not decrement counter in some locations and reject valid job request for not reaching minimum node count.
- Fix FreeBSD-11 build by using llabs() function in place of abs().
- Cray: The slurmd can manipulate the socket/core/thread values reported based upon the configuration. The logic failed to consider select/cray with SelectTypeParameters=other_cons_res as equivalent to select/cons_res.
- If a node's socket or core count are changed at registration time (e.g. a


```

KNL node's NUMA mode is changed), change it's board count to match.
-- Prevent possible divide by zero in select/cons_res if a node's board count
   is higher than it's socket count.
-- Allow an advanced reservation to contain a license count of zero.
-- Preserve non-KNL node features when updating the KNL node features for a
   multi-node job in which the non-KNL node features vary by node.
-- task/affinity plugin: Honor a job's --ntasks-per-socket and
   --ntasks-per-core options in task binding.
-- slurmd - do not print ClusterName when using 'slurmd -C'.
-- Correct a bitmap test function (used only by the select/bluegene plugin).
-- Do not propagate SLURM_UMASK environment variable to batch script.
-- Added node_features/knl_generic plugin for KNL support on non-Cray systems.
-- Cray: Prevent abort in backfill scheduling logic for requeued job that has
   been cancelled while NHC is running.
-- Improve reported estimates of start and end times for pending jobs.
-- pbsnodes: Show OS value as "unknown" for down nodes.
-- BlueGene - correctly scale node counts when enforcing MaxNodes limit take 2.
-- Fix "sbatch --hold" to set JobHeldUser correctly instead of JobHeldAdmin.
-- Cray - print warning that task/cgroup is required, and must be after
   task/cray in the TaskPlugin settings.
-- Document that node Weight takes precedence over load with LLN scheduling.
-- Fix issue where gang scheduling could happen even with OverSubscribe=NO.
-- Expose JOB_SHARED_* values to job_submit/lua plugin.
-- Fix issue where number of nodes is not properly allocated when srun is
   requested with -n tasks < hosts from -w hostlist.
-- Update srun documentation for -N, -w and -m arbitrary.
-- Fix bug that was clearing MAINT mode on nodes scheduled for reboot (bug
   introduced in version 16.05.5 to address bug in overlapping reservations).
-- Add logging of node reboot requests.
-- Docs - remove recommendation for ReleaseAgent setting in cgroup.conf.
-- Make sure a job cleans up completely if it has a node fail. Mostly an
   issue with gang scheduling.

```

* Changes in Slurm 16.05.5

=====

```

-- Fix accounting for jobs requeued after the previous job was finished.
-- slurmstepd modified to pre-load all relevant plugins at startup to avoid
   the possibility of modified plugins later resulting in inconsistent API
   or data structures and a failure of slurmstepd.
-- Export functions from parse_time.c in libslurm.so.
-- Export unit convert functions from slurm_protocol_api.c in libslurm.so.
-- Fix scancel to allow multiple steps from a job to be cancelled at once.
-- Update and expand upgrade guide (in Quick Start Administrator web page).
-- burst_buffer/cray: Requeue, but do not hold a job which fails the pre_run
   operation.
-- Insure reported expected job start time is not in the past for pending jobs.
-- Add support for PMIx v2.
-- mpi/pmix: support for passing TMPDIR path through info key
-- Cray: update slurmfgen_smw.py script to correctly identify service nodes
   versus compute nodes.
-- FreeBSD - fix build issue in knl_cray plugin.
-- Corrections to gres.conf parsing logic.
-- Make partition State independent of EnforcePartLimits value.
-- Fix multipart srun submission with EnforcePartLimits=NO and job violating
   the partition limits.

```

Appendix G. SLURM Release Information

- Fix problem updating job state_reason.
- pmix - Provide HWLOC topology in the job-data if Slurm was configured with hwloc.
- Cray - Fix issue restoring jobs when blade count increases due to hardware reconfiguration.
- burst_buffer/cray - Hold job after 3 failed pre-run operations.
- sched/backfill - Check that a user's QOS is allowed to use a partition before trying to schedule resources on that partition for the job.
- sacctmgr - Fix displaying nodenames when printing out events or reservations.
- Fix mpiexec wrapper to accept task count with more than one digit.
- Add mpiexec man page to the script.
- Add salloc_wait_nodes option to the SchedulerParameters parameter in the slurm.conf file controlling when the salloc command returns in relation to when nodes are ready for use (i.e. booted).
- Handle case when slurmctld daemon restart while compute node reboot in progress. Return node to service rather than setting DOWN.
- Preserve node "RESERVATION" state when one of multiple overlapping reservations ends.
- Restructure srun command locking for task_exit processing logic for improved parallelism.
- Modify srun task completion handling to only build the task/node string for logging purposes if it is needed. Modified for performance purposes.
- Docs - update salloc/sbatch/srun man pages to mention corresponding environment variables for --mem/--mem-per-cpu and allowed suffixes.
- Silence srun warning when overriding the job ntasks-per-node count with a lower task count for the step.
- Docs - assorted spelling fixes.
- node_features/knl_cray: Fix bug where MCDRAM state could be taken from capmc rather than cselect.
- node_features/knl_cray: If a node is rebooted outside of Slurm's direction, update it's active features with current MCDRAM and NUMA mode information.
- Restore ability to manually power down nodes, broken in 15.08.12.
- Don't log error for job end_time being zero if node health check is still running.
- When powering up a node to change it's state (e.g. KNL NUMA or MCDRAM mode) then pass to the ResumeProgram the job ID assigned to the nodes in the SLURM_JOB_ID environment variable.
- Allow a node's PowerUp state flag to be cleared using update_node RPC.
- capmc_suspend/resume - If a request modify NUMA or MCDRAM state on a set of nodes or reboot a set of nodes fails then just requeue the job and abort the entire operation rather than trying to operate on individual nodes.
- node_features/knl_cray plugin: Increase default CapmcTimeout parameter from 10 to 60 seconds.
- Fix squeue filter by job license when a job has requested more than 1 license of a certain type.
- Fix bug in PMIX_Ring in the pmi2 plugin so that it supports singleton mode. It also updates the testpmixring.c test program so it can be used to check singleton runs.
- Automatically cleanup task/cgroup cpuset and devices cgroups after steps are completed.
- Testsuite - Fix test1.83 to handle gaps in node names properly.
- BlueGene - correctly scale node counts when enforcing MaxNodes limit.
- Make sure no attempt is made to schedule a requeued job until all steps are cleaned (Node Health Check completes for all steps on a Cray).

```
-- KNL: Correct task affinity logic for some NUMA modes.
-- Add salloc/sbatch/srun --priority option of "TOP" to set job priority to
  the highest possible value. This option is only available to Slurm operators
  and administrators.
-- Add salloc/sbatch/srun option --use-min-nodes to prefer smaller node counts
  when a range of node counts is specified (e.g. "-N 2-4").
-- Validate salloc/sbatch --wait-all-nodes argument.
-- Add "sbatch_wait_nodes" to SchedulerParameters to control default sbatch
  behaviour with respect to waiting for all allocated nodes to be ready for
  use. Job can override the configuration option using the --wait-all-nodes=#
  option.
-- Prevent partition group access updates from resetting last_part_update when
  no changes have been made. Prevents backfill scheduler from restarting
  mid-cycle unnecessarily.
-- Cray - add NHC_ABSOLUTELY_NO to never run NHC, even on certain edge cases
  that it would otherwise be run on with NHC_NO.
-- Ignore GRES/QOS updates that maintain the same value as before.
-- mpi/pmix - prepare temp directory for application.
-- Fix display for the nice and priority values in sprio/scontrol/squeue.
```

* Changes in Slurm 16.05.4

=====

```
-- Fix potential deadlock if running with message aggregation.
-- Streamline when schedule() is called when running with message aggregation
  on batch script completes.
-- Fix incorrect casting when [un]packing derived_ec on slurmdb_job_rec_t.
-- Document that persistent burst buffers can not be created or destroyed using
  the salloc or srun --bb options.
-- Add support for setting the SLURM_JOB_ACCOUNT, SLURM_JOB_QOS and
  SLURM_JOB_RESERVAION environment variables are set for the salloc command.
  Document the same environment variables for the salloc, sbatch and srun
  commands in their man pages.
-- Fix issue where sacctmgr load cluster.cfg wouldn't load associations
  that had a partition in them.
-- Don't return the extern step from sstat by default.
-- In sstat print 'extern' instead of 4294967295 for the extern step.
-- Make advanced reservations work properly with core specialization.
-- Fix race condition in the account_gather plugin that could result in job
  stuck in COMPLETING state.
-- Regression test fixes if SelectTypePlugin not managing memory and no node
  memory size set (defaults to 1 MB per node).
-- Add missing partition write locks to _slurm_rpc_dump_nodes/node_single to
  prevent a race condition leading to inconsistent sinfo results.
-- Fix task:CPU binding logic for some processors. This bug was introduced
  in version 16.05.1 to address KNL bunding problem.
-- Fix two minor memory leaks in slurmctld.
-- Improve partition-specific limit logging from slurmctld daemon.
-- Fix incorrect access check when using MaxNodes setting on the partition.
-- Fix issue with sacctmgr when specifying a list of clusters to query.
-- Fix issue when calculating future StartTime for a job.
-- Make EnforcePartLimit support logic work with any ordering of partitions
  in job submit request.
-- Prevent restoration of wrong CPU governor and frequency when using
  multiple task plugins.
-- Prevent slurmd abort if hwloc library fails to populate the "children"
```

Appendix G. SLURM Release Information

```
arrays (observed with hwloc version "dev-333-g85ea6e4").
-- burst_buffer/cray: Add "--groupid" to DataWarp "setup" command.
-- Fix lustre profiling putting it in the Filesystem dataset instead of the
  Network dataset.
-- Fix profiling documentation and code to match be consistent with
  Filesystem instead of Lustre.
-- Correct the way watts is calculated in the rapl plugin when using a poll
  frequency other than AcctGatherNodeFreq.
-- Don't about step launch if job reaches expected end time while node is
  configuring/booting (NOTE: The job end time will be adjusted after node
  becomes ready for use).
-- Fix several print routines to respect a custom output delimiter when
  printing NO_VAL or INFINITE.
-- Correct documented configurations where --ntasks-per-core and
  --ntasks-per-socket are supported.
-- task/affinity plugin buffer allocated too small, can corrupt memory.

* Changes in Slurm 16.05.3
=====
-- Make it so the extern step uses a reverse tree when cleaning up.
-- If extern step doesn't get added into the proctrack plugin make sure the
  sleep is killed.
-- Fix areas the slurmctld can segfault if an extern step is in the system
  cleaning up on a restart.
-- Prevent possible incorrect counting of GRES of a given type if a node has
  the multiple "types" of a given GRES "name", which could over-subscribe
  GRES of a given type.
-- Add web links to Slurm Diamond Collectors (from Harvard University) and
  collectd (from EDF).
-- Add job_submit plugin for the "reboot" field.
-- Make some more Slurm constants (INFINITE, NO_VAL64, etc.) available to
  job_submit/lua plugins.
-- Send in a -1 for a taskid into spank_task_post_fork for the extern_step.
-- MYSQL - Slightly better logic if a job completion comes in with an end time
  of 0.
-- task/cgroup plugin is configured with ConstrainRAMSpace=yes, then set soft
  memory limit to allocated memory limit (previously no soft limit was set).
-- Document limitations in burst buffer use by the salloc command (possible
  access problems from a login node).
-- Fix proctrack plugin to only add the pid of a process once
  (regression in 16.05.2).
-- Fix for sstat to print correct info when requesting jobid.batch as part of
  a comma-separated list.
-- CRAY - Fix issue if pid has already been added to another job container.
-- CRAY - Fix add of extern step to AELD.
-- burstbufer/cray: avoid batch submit error condition if waiting for stagein.
-- CRAY - Fix for reporting steps lingering after they are already finished.
-- Testsuite - fix test1.29 / 17.15 for limits with values above 32-bits.
-- CRAY - Simplify when a NHC is called on a step that has unkillable
  processes.
-- CRAY - If trying to kill a step and you have NHC_NO_STEPS set run NHC
  anyway to attempt to log the backtraces of the potential
  unkillable processes.
-- Fix gang scheduling and license release logic if single node job killed on
  bad node.
```

```
-- Make scontrol show steps show the extern step correctly.
-- Do not scheduled powered down nodes in FAILED state.
-- Do not start slurmctld power_save thread until partition information is read
  in order to prevent race condition that can result invalid pointer when
  trying to resolve configured SuspendExcParts.
-- Add SLURM_PENDING_STEP id so it won't be confused with SLURM_EXTERN_CONT.
-- Fix for core selection with job --gres-flags=enforce-binding option.
  Previous logic would in some cases allocate a job zero cores, resulting in
  slurmctld abort.
-- Minimize preempted jobs for configurations with multiple jobs per node.
-- Improve partition AllowGroups caching. Update the table of UIDs permitted to
  use a partition based upon it's AllowGroups configuration parameter as new
  valid UIDs are found rather than looking up that user's group information
  for every job they submit. If the user is now allowed to use the partition,
  then do not check that user's group access again for 5 seconds.
-- Add routing queue information to Slurm FAQ web page.
-- Do not select_g_step_finish() a SLURM_PENDING_STEP step, as nothing has
  been allocated for the step yet.
-- Fixed race condition in PMIx Fence logic.
-- Prevent slurmctld abort if job is killed or requeued while waiting for
  reboot of its allocated compute nodes.
-- Treat invalid user ID in AllowUserBoot option of knl.conf file as error
  rather than fatal (log and do not exit).
-- qsub - When doing the default output files for an array in qsub style
  make them using the master job ID instead of the normal job ID.
-- Create the extern step while creating the job instead of waiting until the
  end of the job to do it.
-- Always report a 0 exit code for the extern step instead of being canceled
  or failed based on the signal that would always be killing it.
-- Fix to allow users to update QOS of pending jobs.
-- CRAY - Fix minor memory leak in switch plugin.
-- CRAY - Change slurmconfgen_smw.py to skip over disabled nodes.
-- Fix eligible_time for elasticsearch as well as add queue_wait
  (difference between start of job and when it was eligible).
```

* Changes in Slurm 16.05.2

=====

```
-- CRAY - Fix issue where the proctrack plugin could hang if the container
  id wasn't able to be made.
-- Move test for job wait reason value of BurstBufferResources and
  BurstBufferStageIn later in the scheduling logic.
-- Document which srun options apply to only job, only step, or job and step
  allocations.
-- Use more compatible function to get thread name (>= 2.6.11).
-- Fix order of job then step id when noting cleaning flag being set.
-- Make it so the extern step sends a message with accounting information
  back to the slurmctld.
-- Make it so the extern step calls the select_g_step_start|finish functions.
-- Don't print error when extern step is canceled because job is ending.
-- Handle a few error codes when dealing with the extern step to make sure
  we have the pids added to the system correctly.
-- Add support for job dependencies with job array expressions. Previous logic
  required listing each task of job array individually.
-- Make sure tres_cnt is set before creating a slurmdb_assoc_usage_t.
-- Prevent backfill scheduler from starting a second "singleton" job if another
```

Appendix G. SLURM Release Information

```
    one started during a backfill sleep.
-- Fix for invalid array pointer when creating advanced reservation when job
    allocations span heterogeneous nodes (differing core or socket counts).
-- Fix hostlist_ranged_string_xmalloc_dims to correctly not put brackets on
    hostlists when brackets == 0.
-- Make sure we don't get brackets when making a range of reserved ports
    for a step.
-- Change fatal to an error if port ranges aren't correct when reading state
    for steps.

* Changes in Slurm 16.05.1
=====
-- Fix __cplusplus macro in spank.h to allow compilation with C++.
-- Fix compile issue with older glibc < 2.12
-- Fix for starting batch step with mpi/pmix plugin.
-- Fix for "scontrol -dd show job" with respect to displaying the specific
    CPUs allocated to a job on each node. Prior logic would only display
    the CPU information for the first node in the job allocation.
-- Print correct return code on failure to update active node features
    through sview.
-- Allow QOS timelimit to override partition timelimit when EnforcePartLimits
    is set to all/any.
-- Make it so qsub will do a "basename" on a wrapped command for the output
    and error files.
-- Fix issue where slurmd could core when running the ipmi energy plugin.
-- Documentation - clean up typos.
-- Add logic so that slurmstepd can be launched under valgrind.
-- Increase buffer size to read /proc/*/stat files.
-- Fix for tracking job resource allocation when slurmctld is reconfigured
    while Cray Node Health Check (NHC) is running. Previous logic would fail to
    record the job's allocation then perform release operation upon NHC
    completion, resulting in underflow error messages.
-- Make "scontrol show daemons" work with long node names.
-- CRAY - Collect energy using a uint64_t instead of uint32_t.
-- Fix incorrect if statements when determining if the user has a default
    account or wckey.
-- Prevent job stuck in configuring state if slurmctld daemon restarted while
    PrologSlurmctld is running. Also re-issue burst_buffer/pre-load operation
    as needed.
-- Correct task affinity support for FreeBSD.
-- Fix for task affinity on KNL in SNC2/Flat mode.
-- Recalculate a job's memory allocation after node reboot if job requests all
    of a node's memory and FastSchedule=0 is configured. Intel KNL memory size
    can change on reboot with various MCDRAM modes.
-- Fix small memory leak when printing HealthCheckNodeState.
-- Eliminate memory leaks when AuthInfo is configured.
-- Improve sdiag output description in man page.
-- Cray/capmc_resume script modify a node's features (as needed) when the
    reinit (reboot) command is issued rather than wait for the nodes to change
    to the "on" state.
-- Correctly print ranges when using step values in job arrays.
-- Allow from file names / paths over 256 characters when launching steps,
    as well as spaces in the executable name.
-- job_submit.license.lua example modified to send message back to user.
-- Document job --mem=0 option means all memory on a node.
```

```
-- Set SLURM_JOB_QOS environment variable to QOS name instead of description.
-- knl_cray.conf file option of CnselectPath added.
-- node_features/knl_cray plugin modified to get current node NUMA and MCDRAM
   modes using cnselect command rather than capmc command.
-- liblua - add SLES12 paths to runtime search list.
-- Fix qsub default output and error files for task arrays.
-- Fix qsub to set job_name correctly when wrapping a script (-b y)
-- Cray - set EnforcePartLimits=any in slurm.conf template.
```

* Changes in Slurm 16.05.0

=====

```
-- Update seff to fix warnings with ncpus, and list slurm-perlapi dependency
   in spec file.
-- Fix testsuite to consistent use /usr/bin/env {bash,expect} construct.
-- Cray - Ensure that step completion messages get to the database.
-- Fix step cpus_per_task calculation for heterogeneous job allocation.
-- Fix --with-json= configure option to use specified path.
-- Add back thread_id to "thread_id" LogTimeFormat to distinguish between
   mutliple threads with the same name. Now displays thread name and id.
-- Change how Slurm determines the NUMA count of a node. Ignore KNL NUMA
   that only include memory.
-- Cray - Fix node list parsing in capmc_suspend/resume programs.
-- Fix sbatch #BSUB parsing for -W and -M options.
-- Fix GRES task layout bug that could cause slurmctld to abort.
-- Fix to --gres-flags=enforce-binding logic when multiple sockets needed.
```

* Changes in Slurm 16.05.0rc2

=====

```
-- Cray node shutdown/reboot scripts, perform operations on all nodes in one
   capmc command. Only if that fails, issue the operations in parallel on
   individual nodes. Required for scalability.
-- Cleanup two minor Coverity warnings.
-- Make it so the tres units in a job's formatted string are converted like
   they are in a step.
-- Correct partition's MaxCPUsPerNode enforcement when nodes are shared by
   multiple partitions.
-- node_feature/knl_cray - Prevent slurmctld GRES errors for "hbm" references.
-- Display thread name instead of thread id and remove process name in stderr
   logging for "thread_id" LogTimeFormat.
-- Log IP address of bad incomming message to slurmctld.
-- If a user requests tasks, nodes and ntasks-per-node and
   tasks-per-node/nodes != tasks print warning and ignore ntasks-per-node.
-- Release CPU "owner" file locks.
-- Fix for job step memory allocation: Reject invalid step at submit time
   rather than leaving it queued.
-- Whenever possible, avoid allocating nodes that require a reboot.
```

* Changes in Slurm 16.05.0rc1

=====

```
-- Remove the SchedulerParameters option of "assoc_limit_continue", making it
   the default value. Add option of "assoc_limit_stop". If "assoc_limit_stop"
   is set and a job cannot start due to association limits, then do not attempt
   to initiate any lower priority jobs in that partition. Setting this can
   decrease system throughput and utlization, but avoid potentially starving
   larger jobs by preventing them from launching indefinitely.
```

Appendix G. SLURM Release Information

- Update a node's socket and cores per socket counts as needed after a node boot to reflect configuration changes which can occur on KNL processors. Note that the node's total core count must not change, only the distribution of cores across varying socket counts (KNL NUMA nodes treated as sockets by Slurm).
- Rename partition configuration from "Shared" to "OverSubscribe". Rename salloc, sbatch, srun option from "--shared" to "--oversubscribe". The old options will continue to function. Output field names also changed in scontrol, sinfo, squeue and svview.
- Add SLURM_UMASK environment variable to user job.
- knl_conf: Added new configuration parameter of CapmcPollFreq.
- squeue: remove errant spaces in column formats for "squeue -o %all".
- Add ARRAY_TASKS mail option to send emails to each task in a job array.
- Change default compression library for sbcast to lz4.
- select/cray - Initiate step node health check at start of step termination rather than after application completely ends so that NHC can capture information about hung (non-killable) processes.
- Add --units=[KMGTP] option to sacct to display values in specific unit type.
- Modify sacct and sacctmgr to display TRES values in converted units.
- Modify sacctmgr to accept TRES values with [KMGTP] suffixes.
- Replace hash function with more modern SipHash functions.
- Add "--with-cray_dir" build/configure option.
- BB- Only send stage_out email when stage_out is set in script.
- Add r/w locking to file_bcast receive functions in slurmd.
- Add TopologyParam option of "TopoOptional" to optimize network topology only for jobs requesting it.
- Fix build on FreeBSD.
- Configuration parameter "CpuFreqDef" used to set default governor for job step not specifying --cpu-freq (previously the parameter was unused).
- Fix sshare -o<format> to correctly display new lengths.
- Update documentation to rename Shared option to OverSubscribe.
- Update documentation to rename partition Priority option to PriorityTier.
- Prevent changing of QOS on running jobs.
- Update accounting when changing QOS on pending jobs.
- Add support to ntasks_per_socket in task/affinity.
- Generate init.d and systemd service scripts in etc/ through Make rather than at configure time to ensure all variable substitutions happen.
- Use TaskPluginParam for default task binding if no user specified CPU binding. User --cpu_bind option takes precedent over default. No longer any error if user --cpu_bind option does not match TaskPluginParam.
- Make sacct and sattach work with older slurmd versions.
- Fix protocol handling between 15.08 and 16.05 for 'scontrol show config'.
- Enable prefixes (e.g. info, debug, etc.) in slurmd debugging.

* Changes in Slurm 16.05.0pre2

=====

- Split partition's "Priority" field into "PriorityTier" (used to order partitions for scheduling and preemption) plus "PriorityJobFactor" (used by priority/multifactor plugin in calculating job priority, which is used to order jobs within a partition for scheduling).
- Revert call to getaddrinfo, restoring gethostbyaddr (introduced in Slurm 16.05.0pre1) which was failing on some systems.
- knl_cray.conf - Added AllowMCDRAM, AllowNUMA and AllowUserBoot configuration options.
- Add node_features_p_user_update() function to node_features plugin.


```

-- Don't print Weight=1 lines in 'scontrol write config' (its the default).
-- Remove PARAMS macro from slurm.h.
-- Remove BEGIN_C_DECLS and END_C_DECLS macros.
-- Check that PowerSave mode configured for node_features/knl_cray plugin.
  It is required to reconfigure and reboot nodes.
-- Update documentation to reflect new cgroup default location change from
  /cgroup to /sys/fs/cgroup.
-- If NodeHealthCheckProgram configured HealthCheckInterval is non-zero, then
  modify slurmd to run it before registering with slurmd.
-- Fix for tasks being packed onto cores when the requested --cpus-per-task is
  greater than the number of threads on a core and --ntasks-per-core is 1.
-- Make it so jobs/steps track ':' named gres/tres, before hand gres/gpu:tesla
  would only track gres/gpu, now it will track both gres/gpu and
  gres/gpu:tesla as separate gres if configured like
  AccountingStorageTRES=gres/gpu,gres/gpu:tesla
-- Added new job dependency type of "aftercorr" which will start a task of a
  job array after the corresponding task of another job array completes.
-- Increase default MaxTasksPerNode configuration parameter from 128 to 512.
-- Enable sbcast data compression logic (compress option previously ignored).
-- Add --compress option to srun command for use with --bcast option.
-- Add TCPTimeout option to slurm[dbd].conf. Decouples MessageTimeout from TCP
  connections.
-- Don't call primary controller for every RPC when backup is in control.
-- Add --gres-flags=enforce-binding option to salloc, sbatch and srun commands.
  If set, the only CPUs available to the job will be those bound to the
  selected GRES (i.e. the CPUs identified in the gres.conf file will be
  strictly enforced rather than advisory).
-- Change how a node's allocated CPU count is calculated to avoid double
  counting CPUs allocated to multiple jobs at the same time.
-- Added SchedulingParameters option of "bf_min_prio_reserve". Jobs below
  the specified threshold will not have resources reserved for them.
-- Added "sacctmgr show lostjobs" to report any orphaned jobs in the database.
-- When a stepd is about to shutdown and send it's response to srun
  make the wait to return data only hit after 500 nodes and configurable
  based on the TcpTimeout value.
-- Add functionality to reset the lft and rgt values of the association table
  with the slurmdbd.
-- Add SchedulerParameter no_env_cache, if set no env cache will be use when
  launching a job, instead the job will fail and drain the node if the env
  isn't loaded normally.

* Changes in Slurm 16.05.0pre1
=====
-- Add sbatch "--wait" option that waits for job completion before exiting.
  Exit code will match that of spawned job.
-- Modify advanced reservation save/restore logic for core reservations to
  support configuration changes (changes in configured nodes or cores counts).
-- Allow ControlMachine, BackupController, DbdHost and DbdBackupHost to be
  either short or long hostname.
-- Job output and error files can now contain "%" character by specifying
  a file name with two consecutive "%" characters. For example,
  "sbatch -o "slurm.%%.%.j" for job ID 123 will generate an output file named
  "slurm.%.123".
-- Pass user name in Prolog RPC from controller to slurmd when using
  PrologFlags=Alloc. Allows SLURM_JOB_USER env variable to be set when using

```

Appendix G. SLURM Release Information

Native Slurm on a Cray.

- Add "NumTasks" to job information visible to Slurm commands.
- Add mail wrapper script "smail" that will include job statistics in email notification messages.
- Remove vestigial "SICP" job option (inter-cluster job option). Completely different logic will be forthcoming.
- Fix case where the primary and backup dbds would both be performing rollup.
- Add an ack reply from slurmd to slurmstepd when job setup is done and the job is ready to be executed.
- Removed support for authd. authd has not been developed and supported since several years.
- Introduce a new parameter requeue_setup_env_fail in SchedulerParameters. A job that fails to setup the environment will be requeued and the node drained.
- Add ValidateTimeout and OtherTimeout to "scontrol show burst" output.
- Increase default sbcast buffer size from 512KB to 8MB.
- Enable the hdf5 profiling of the batch step.
- Eliminate redundant environment and script files for job arrays.
- Stop searching sbatch scripts for #PBS directives after 100 lines of non-comments. Stop parsing #PBS or #SLURM directives after 1024 characters into a line. Required for decent performance with huge scripts.
- Add debug flag for timing Cray portions of the code.
- Remove all *.la files from RPMs.
- Add Multi-Category Security (MCS) infrastructure to permit nodes to be bound to specific users or groups.
- Install the pmi2 unix sockets in slurmd spool directory instead of /tmp.
- Implement the getaddrinfo and getnameinfo instead of gethostbyaddr and gethostbyname.
- Finished PMIx implementation.
- Implemented the --without=package option for configure.
- Fix sshare to show each individual cluster with -M,--clusters option.
- Added --deadline option to salloc, sbatch and srun. Jobs which can not be completed by the user specified deadline will be terminated with a state of "Deadline" or "DL".
- Implemented and documented PMIX protocol which is used to bootstrap an MPI job. PMIX is an alternative to PMI and PMI2.
- Change default CgroupMountpoint (in cgroup.conf) from "/cgroup" to "/sys/fs/cgroup" to match current standard.
- Add #BSUB options to sbatch to read in from the batch script.
- HDF: Change group name of node from nodename to nodeid.
- The partition-specific SelectTypeParameters parameter can now be used to change the memory allocation tracking specification in the global SelectTypeParameters configuration parameter. Supported partition-specific values are CR_Core, CR_Core_Memory, CR_Socket and CR_Socket_Memory. If the global SelectTypeParameters value includes memory allocation management and the partition-specific value does not, then memory allocation management for that partition will NOT be supported (i.e. memory can be over-allocated). Likewise the global SelectTypeParameters might not include memory management while the partition-specific value does.
- Burst buffer/cray - Add support for multiple buffer pools including support for different resource granularity by pool.
- Burst buffer advanced reservation units treated as bytes (per documentation) rather than GB.
- Add an "scontrol top <jobid>" command to re-order the priorities of a user's pending jobs. May be disabled with the "disable_user_top" option in the

```

SchedulerParameters configuration parameter.
-- Modify svview to display negative job nice values.
-- Increase job's nice value field from 16 to 32 bits.
-- Remove deprecated job_submit/cnode plugin.
-- Enhance slurm.conf option EnforcePartLimit to include options like "ANY" and
  "ALL". "Any" is equivalent to "Yes" and "All" will check all partitions
  a job is submitted to and if any partition limit is violated the job will
  be rejected even if it could possibly run on another partition.
-- Add "features_act" field (currently active features) to the node
  information. Output of scontrol, sinfo, and svview changed accordingly.
  The field previously displayed as "Features" is now "AvailableFeatures"
  while the new field is displayed as "ActiveFeatures".
-- Remove Sun Constellation, IBM Federation Switches (replaced by NRT switch
  plugin) and long-defunct Quadrics Elan support.
-- Add -M<clusters> option to sreport.
-- Rework group caching to work better in environments with
  enumeration disabled. Removed CacheGroups config directive, group
  membership lists are now always cached, controlled by
  GroupUpdateTime parameter. GroupUpdateForce parameter default
  value changed to 1.
-- Add reservation flag of "purge_comp" which will purge an advanced
  reservation once it has no more active (pending, suspended or running) jobs.
-- Add new configuration parameter "KNLPlugins" and plugin infrastructure.
-- Add optional job "features" to node reboot RPC.
-- Add slurmd "-b" option to report node rebooted at daemon start time. Used
  for testing purposes.
-- contribs/cray: Add framework for powering nodes up and down.
-- For job constraint, convert comma separator to "&".
-- Add Max*PerAccount options for QOS.
-- Protect slurm_mutex_* calls with abort() on failure.

* Changes in Slurm 15.08.14
=====
-- For job resize, correct logic to build "resize" script with new values.
  Previously the scripts were based upon the original job size.

* Changes in Slurm 15.08.13
=====
-- Fix issue where slurmd could core when running the ipmi energy plugin.
-- Print correct return code on failure to update node features through svview.
-- Documentation - cleanup typos.
-- Add logic so that slurmstepd can be launched under valgrind.
-- Increase buffer size to read /proc/*/stat files.
-- MYSQL - Handle ER_HOST_IS_BLOCKED better by failing when it occurs instead
  of continuously printing the message over and over as the problem will
  most likely not resolve itself.
-- Add --disable-bluegene to configure. This will make it so Slurm
  can work on a BGAS node.
-- Prevent job stuck in configuring state if slurmctld daemon restarted while
  PrologSlurmctld is running.
-- Handle association correctly if using FAIR_TREE as well as shares=Parent
-- Fix race condition when setting priority of a job and the association
  doesn't have a parent.
-- MYSQL - Fix issue with adding a reservation if the name has single quotes in
  it.

```

Appendix G. SLURM Release Information

- Correctly print ranges when using step values in job arrays.
- Fix for invalid array pointer when creating advanced reservation when job allocations span heterogeneous nodes (differing core or socket counts).
- Fix for sstat to print correct info when requesting jobid.batch as part of a comma-separated list.
- Cray - Fix issue restoring jobs when blade count increases due to hardware reconfiguration.
- Ignore warnings about deprecated functions. This is primarily there for new glibc 2.24+ that deprecates readdir_r.
- Fix security issue caused by insecure file path handling triggered by the failure of a Prolog script. To exploit this a user needs to anticipate or cause the Prolog to fail for their job. CVE-2016-10030.

* Changes in Slurm 15.08.12

=====

- Do not attempt to power down a node which has never responded if the slurmd daemon restarts without state.
- Fix for possible slurmd segfault on invalid user ID.
- MySQL - Fix for possible race condition when archiving multiple clusters at the same time.
- Fix compile for when you don't have hwloc.
- Fix issue where daemons would only listen on specific address given in slurm.conf instead of all. If looking for specific addresses use TopologyParam options No*InAddrAny.
- Cray - Better robustness when dealing with the aeld interface.
- job_submit.lua - add array_inx value for job arrays.
- Perlapi - Remove unneeded/undefined mutex.
- Fix issue when TopologyParam=NoInAddrAny is set the responses wouldn't make it to the slurmd when using message aggregation.
- MySQL - Fix potential memory leak when rolling up data.
- Fix issue with clustername file when running on NFS with root_squash.
- Fix race condition with respects to cleaning up the profiling threads when in use.
- Fix issues when building on NetBSD.
- Fix jobcomp/elasticsearch build when libcurl is installed in a non-standard location.
- Fix MemSpecLimit to explicitly require TaskPlugin=task/cgroup and ConstrainRAMSpace set in cgroup.conf.
- MYSQL - Fix order of operations issue where if the database is locked up and the slurmd doesn't wait long enough for the response it would give up leaving the connection open and create a situation where the next message sent could receive the response of the first one.
- Fix CFULL_BLOCK distribution type.
- Prevent sbatch from trying to enable debug messages when using job arrays.
- Prevent sbcast from enabling "--preserve" when specifying a jobid.
- Prevent wrong error message from spank plugin stack on GLOB_NOSPACE error.
- Fix proctrack/lua plugin to prevent possible deadlock.
- Prevent infinite loop in slurmd if execve fails.
- Prevent multiple responses to REQUEST_UPDATE_JOB_STEP message.
- Prevent possible deadlock in acct_gather_filesystem/lustre on error.
- Make it so --mail-type=NONE didn't throw an invalid error.
- If no default account is given for a user when creating (only a list of accounts) no default account is printed, previously NULL was printed.
- Fix for tracking a node's allocated CPUs with gang scheduling.
- Fix Hidden error during _rpc_forward_data call.

- Fix bug resulting from wrong order-of-operations in `_connect_srun_cr()`, and two others that cause incorrect debug messages.
- Fix backwards compatibility with `sreport` going to `<= 14.11` coming from `>= 15.08` for some reports.

* Changes in Slurm 15.08.11

=====

- Fix for job "--contiguous" option that could cause job allocation/launch failure or `slurmctld` crash.
- Fix to setup logs for single-character program names correctly.
- Backfill scheduling performance enhancement with large number of running jobs.
- Reset job's `prolog_running` counter on `slurmctld` restart or reconfigure.
- `burst_buffer/cray` - Update job's `prolog_running` counter if `pre_run` fails.
- MySQL - Make the error message more specific when removing a reservation and it doesn't meet basic requirements.
- `burst_buffer/cray` - Fix for script creating or deleting persistent buffer would fail "paths" operation and hold the job.
- `power/cray` - Prevent possible divide by zero.
- `power/cray` - Fix bug introduced in 15.08.10 `preventin` operation in many cases.
- Prevent deadlock for flow of data to the `slurmdbd` when sending reservation that wasn't set up correctly.
- `burst_buffer/cray` - Don't call `Datawarp "paths"` function if script includes only create or destroy of persistent burst buffer. Some versions of `Datawarp` software return an error for such scripts, causing the job to be held.
- Fix potential issue when adding and removing TRES which could result in the `slurmdbd` segfaulting.
- Add cast to memory limit calculation to prevent integer overflow for very large memory values.
- Bluegene - Fix issue with reservations resizing under the covers on a restart of the `slurmctld`.
- Avoid error message of "Requested `cpu_bind` option requires entire node to be allocated; disabling affinity" being generated in some cases where `task/affinity` and `task/cgroup` plugins used together.
- Fix version issue when packing GRES information between 2 different versions of Slurm.
- Fix for `srun` hanging with OpenMPI and PMIx
- Better initialization of `node_ptr` when dealing with `protocol_version`.
- Fix incorrect type when initializing header of a message.
- MySQL - Fix incorrect usage of `limit` and `union`.
- MySQL - Remove 'ignore' from `alter ignore` when updating a table.
- Documentation - update `prolog_epilog` page to reflect current behavior if the Prolog fails.
- Documentation - clarify behavior of '`srun --export=NONE`' in man page.
- Fix potential gres underflow on restart of `slurmctld`.
- Fix `sacctmgr` to remove a user who has no associations.

* Changes in Slurm 15.08.10

=====

- Fix issue where if a `slurmdbd` rollup lasted longer than 1 hour the rollup would effectively never run again.
- Make error message in the `pmi2` code to debug as the issue can be expected and retries are done making the error message a little misleading.
- `Power/cray`: Don't specify NID list to Cray APIs. If any of those nodes are

Appendix G. SLURM Release Information

```
not in a ready state, the API returned an error for ALL nodes rather than
valid data for nodes in ready state.
-- Fix potential divide by zero when tree_width=1.
-- checkpoint/blcr plugin: Fix memory leak.
-- If using PrologFlags=contain: Don't launch the extern step if a job is
cancelled while launching.
-- Remove duplicates from AccountingStorageTRES
-- Fix backfill scheduler race condition that could cause invalid pointer in
select/cons_res plugin. Bug introduced in 15.08.9.
-- Avoid double calculation on partition QOS if the job is using the same QOS.
-- Do not change a job's time limit when updating unrelated field in a job.
-- Fix situation on a heterogeneous memory cluster where the order of
constraints mattered in a job.

* Changes in Slurm 15.08.9
=====
-- BurstBuffer/cray - Defer job cancellation or time limit while "pre-run"
operation in progress to avoid inconsistent state due to multiple calls
to job termination functions.
-- Fix issue with resizing jobs and limits not be kept track of correctly.
-- BGQ - Remove redeclaration of job_read_lock.
-- BGQ - Tighter locks around structures when nodes/cables change state.
-- Make it possible to change CPUsPerTask with scontrol.
-- Make it so scontrol update part qos= will take away a partition QOS from
a partition.
-- Fix issue where SocketsPerBoard didn't translate to Sockets when CPUS=
was also given.
-- Add note to slurm.conf man page about setting "--cpu_bind=no" as part
of SallocDefaultCommand if a TaskPlugin is in use.
-- Set correct reason when a QOS' MaxTresMins is violated.
-- Insure that a job is completely launched before trying to suspend it.
-- Remove historical presentations and design notes. Only distribute
maintained doc/html and doc/man directories.
-- Remove duplicate xmalloc() in task/cgroup plugin.
-- Backfill scheduler to validate correct job partition for job submitted to
multiple partitions.
-- Force close on exec on first 256 file descriptors when launching a
slurmstepd to close potential open ones.
-- Step GRES value changed from type "int" to "int64_t" to support larger
values.
-- Fix getting reservations to database when database is down.
-- Fix issue with sbcast not doing a correct fanout.
-- Fix issue where steps weren't always getting the gres/tres involved.
-- Fixed double read lock on getting job's gres/tres.
-- Fix display for RoutePlugin parameter to display the correct value.
-- Fix route/topology plugin to prevent segfault in sbcast when in use.
-- Fix Cray slurmconfgen_smw.py script to use nid as nid, not nic.
-- Fix Cray NHC spawning on job requeue. Previous logic would leave nodes
allocated to a requeued job as non-usable on job termination.
-- burst_buffer/cray plugin: Prevent a requeued job from being restarted while
file stage-out is still in progress. Previous logic could restart the job
and not perform a new stage-in.
-- Fix job array formatting to allow return [0-100:2] display for arrays with
step functions rather than [0,2,4,6,8,...] .
-- FreeBSD - replace Linux-specific set_oom_adj to avoid errors in slurmd log.
```

- Add option for TopologyParam=NoInAddrAnyCtld to make the slurmctld listen on only one port like TopologyParam=NoInAddrAny does for everything else.
- Fix burst buffer plugin to prevent corruption of the CPU TRES data when bb is not set as an AccountingStorageTRES type.
- Suppress error messages in acct_gather_energy/ipmi plugin after repeated failures.
- Change burst buffer use completion email message from
"SLURM Job_id=1360353 Name=tmp Staged Out, StageOut time 00:01:47" to
"SLURM Job_id=1360353 Name=tmp StageOut/Teardown time 00:01:47"
- Generate burst buffer use completion email immediately after teardown completes rather than at job purge time (likely minutes later).
- Fix issue when adding a new TRES to AccountingStorageTRES for the first time.
- Update gang scheduling tables when job manually suspended or resumed. Prior logic could mess up job suspend/resume sequencing.
- Update gang scheduling data structures when job changes in size.
- Associations - prevent hash table corruption if uid initially unset for a user, which can cause slurmctld to crash if that user is deleted.
- Avoid possibly aborting srun on SIGSTOP while creating the job step due to threading bug.
- Fix deadlock issue with burst_buffer/cray when a newly created burst buffer is found.
- burst_buffer/cray: Set environment variables just before starting job rather than at job submission time to reflect persistent buffers created or modified while the job is pending.
- Fix check of per-user qos limits on the initial run by a user.
- Fix gang scheduling resource selection bug which could prevent multiple jobs from being allocated the same resources. Bug was introduced in 15.08.6.
- Don't print the Rgt value of an association from the cache as it isn't kept up to date.
- burst_buffer/cray - If the pre-run operation fails then don't issue duplicate job cancel/requeue unless the job is still in run state. Prevents jobs hung in COMPLETING state.
- task/cgroup - Fix bug in task binding to CPUs.

* Changes in Slurm 15.08.8

=====

- Backfill scheduling properly synchronized with Cray Node Health Check. Prior logic could result in highest priority job getting improperly postponed.
- Make it so daemons also support TopologyParam=NoInAddrAny.
- If scancel is operating on large number of jobs and RPC responses from slurmctld daemon are slow then introduce a delay in sending the cancel job requests from scancel in order to reduce load on slurmctld.
- Remove redundant logic when updating a job's task count.
- MySQL - Fix querying jobs with reservations when the id's have rolled.
- Perl - Fix use of uninitialized variable in slurm_job_step_get_pids.
- Launch batch job requesting --reboot after the boot completes.
- Move debug messages like "not the right user" from association manager to debug3 when trying to find the correct association.
- Fix incorrect logic when querying assoc_mgr information.
- Move debug messages to debug3 notifying a gres_bit_alloc was NULL for gres types without a file.
- Sanity Check Patch to setup variables for RAPL if in a race for it.
- GRES - Fix minor typecast issues.

Appendix G. SLURM Release Information

- burst_buffer/cray - Increase size of intermediate variable used to store buffer byte size read from DW instance from 32 to 64-bits to avoid overflow and reporting invalid buffer sizes.
- Allow an existing reservation with running jobs to be modified without Flags=IGNORE_JOBS.
- srun - don't attempt to execve() a directory with a name matching the requested command
- Do not automatically relocate an advanced reservation for individual cores that spans multiple nodes when nodes in that reservation go down (e.g. a 1 core reservation on node "tux1" will be moved if node "tux1" goes down, but a reservation containing 2 cores on node "tux1" and 3 cores on "tux2" will not be moved node "tux1" goes down). Advanced reservations for whole nodes will be moved by default for down nodes.
- Avoid possible double free of memory (and likely abort) for slurmctld in background mode.
- contribs/cray/csm/slurmconfgen_smw.py - avoid including repurposed compute nodes in configs.
- Support AuthInfo in slurmdbd.conf that is different from the value in slurm.conf.
- Fix build on FreeBSD 10.
- Fix hdf5 build on ppc64 by using correct fprintf formatting for types.
- Fix cosmetic printing of NO_VALs in scontrol show assoc_mgr.
- Fix perl api for newer perl versions.
- Fix for jobs requesting cpus-per-task (eg. -c3) that exceed the number of cpus on a core.
- Remove unneeded perl files from the .spec file.
- Flesh out filters for scontrol show assoc_mgr.
- Add function to remove assoc_mgr_info_request_t members without freeing structure.
- Fix build on some non-glibc systems by updating includes.
- Add new PowerParameters options of get_timeout and set_timeout. The default set_timeout was increased from 5 seconds to 30 seconds. Also re-read current power caps periodically or after any failed "set" operation.
- Fix slurmdbd segfault when listing users with blank user condition.
- Save the ClusterName to a file in SaveStateLocation, and use that to verify the state directory belongs to the given cluster at startup to avoid corruption from multiple clusters attempting to share a state directory.
- MYSQL - Fix issue when rerolling monthly data to work off correct time period. This would only hit you if you rerolled a 15.08 prior to this commit.
- If FastSchedule=0 is used make sure TRES are set up correctly in accounting.
- Fix sreport's truncation of columns with large TRES and not using a parsing option.
- Make sure count of boards are restored when slurmctld has option -R.
- When determine if a job can fit into a TRES time limit after resources have been selected set the time limit appropriately if the job didn't request one.
- Fix inadequate locks when updating a partition's TRES.
- Add new assoc_limit_continue flag to SchedulerParameters.
- Avoid race in acct_gather_energy_cray if energy requested before available.
- MYSQL - Avoid having multiple default accounts when a user is added to a new account and making it a default all at once.

* Changes in Slurm 15.08.7

=====


```

-- sched/backfill: If a job can not be started within the configured
   backfill_window, set it's start time to 0 (unknown) rather than the end
   of the backfill_window.
-- Remove the 1024-character limit on lines in batch scripts.
-- burst_buffer/cray: Round up swap size by configured granularity.
-- select/cray: Log repeated aeld reconnects.
-- task/affinity: Disable core-level task binding if more CPUs required than
   available cores.
-- Preemption/gang scheduling: If a job is suspended at slurmctld restart or
   reconfiguration time, then leave it suspended rather than resume+suspend.
-- Don't use lower weight nodes for job allocation when topology/tree used.
-- BGQ - If a cable goes into error state remove the under lying block on
   a dynamic system and mark the block in error on a static/overlap system.
-- BGQ - Fix regression in 9cc4ae8add7f where blocks would be deleted on
   static/overlap systems when some hardware issue happens when restarting
   the slurmctld.
-- Log if CLOUD node configured without a resume/suspend program or suspend
   time.
-- MYSQL - Better locking around g_qos_count which was previously unprotected.
-- Correct size of buffer used for jobid2str to avoid truncation.
-- Fix allocation/distribution of tasks across multiple nodes when
   --hint=nomultithread is requested.
-- If a reservation's nodes value is "all" then track the current nodes in the
   system, even if those nodes change.
-- Fix formatting if using "tree" option with sreport.
-- Make it so sreport prints out a line for non-existent TRES instead of
   error message.
-- Set job's reason to "Priority" when higher priority job in that partition
   (or reservation) can not start rather than leaving the reason set to
   "Resources".
-- Fix memory corruption when a new non-generic TRES is added to the
   DBD for the first time. The corruption is only noticed at shutdown.
-- burst_buffer/cray - Improve tracking of allocated resources to handle race
   condition when reading state while buffer allocation is in progress.
-- If a job is submitted only with -c option and numcpus is updated before
   the job starts update the cpus_per_task appropriately.
-- Update salloc/sbatch/srun documentation to mention time granularity.
-- Fixed memory leak when freeing assoc_mgr_info_msg_t.
-- Prevent possible use of empty reservation core bitmap, causing abort.
-- Remove unneeded pack32's from qos_rec when qos_rec is NULL.
-- Make sacctmgr print MaxJobsPerUser when adding/altering a QOS.
-- Correct dependency formatting to print array task ids if set.
-- Update sacctmgr help with current QOS options.
-- Update slurmstepd to initialize authentication before task launch.
-- burst_cray/cray: Eliminate need for dedicated nodes.
-- If no MsgAggregationParams is set don't set the internal string to
   anything. The slurmd will process things correctly after the fact.
-- Fix output from api when printing job step not found.
-- Don't allow user specified reservation names to disrupt the normal
   reservation sequeuece numbering scheme.
-- Fix scontrol to be able to accept TRES as an option when creating
   a reservation.
-- contrib/torque/qstat.pl - return exit code of zero even with no records
   printed for 'qstat -u'.
-- When a reservation is created or updated, compress user provided node names

```

Appendix G. SLURM Release Information

```
using hostlist functions (e.g. translate user input of "Nodes=tux1,tux2"
into "Nodes=tux[1-2]").
-- Change output routines for scontrol show partition/reservation to handle
unexpectedly large strings.
-- Add more partition fields to "scontrol write config" output file.
-- Backfill scheduling fix: If a job can't be started due to a "group" resource
limit, rather than reserve resources for it when the next job ends, don't
reserve any resources for it.
-- Avoid slurmstepd abort if malloc fails during accounting gather operation.
-- Fix nodes from being overallocated when allocation straddles multiple nodes.
-- Fix memory leak in slurmctld job array logic.
-- Prevent decrementing of TRESRunMins when AccountingStorageEnforce=limits is
not set.
-- Fix backfill scheduling bug which could postpone the scheduling of jobs due
to avoidance of nodes in COMPLETING state.
-- Properly account for memory, CPUs and GRES when slurmctld is reconfigured
while there is a suspended job. Previous logic would add the CPUs, but not
memory or GPUs. This would result in underflow/overflow errors in select
cons_res plugin.
-- Strip flags from a job state in qstat wrapper before evaluating.
-- Add missing job states from the qstat wrapper.
-- Cleanup output routines to reduce number of fixed-sized buffer function
calls and allow for unexpectedly large strings.
```

* Changes in Slurm 15.08.6
=====

```
-- In slurmctld log file, log duplicate job ID found by slurmd. Previously was
being logged as prolog/epilog failure.
-- If a job is requeued while in the process of being launch, remove it's
job ID from slurmd's record of active jobs in order to avoid generating a
duplicate job ID error when launched for the second time (which would
drain the node).
-- Cleanup messages when handling job script and environment variables in
older directory structure formats.
-- Prevent triggering gang scheduling within a partition if configured with
PreemptType=partition_prio and PreemptMode=suspend,gang.
-- Decrease parallelism in job cancel request to prevent denial of service
when cancelling huge numbers of jobs.
-- If all ephemeral ports are in use, try using other port numbers.
-- Revert way lib lua is handled when doing a dlopen, fixing a regression in
15.08.5.
-- Set the debug level of the rmdir message in xcgroup_delete() to debug2.
-- Fix the qstat wrapper when user is removed from the system but still
has running jobs.
-- Log the request to terminate a job at info level if DebugFlags includes
the Steps keyword.
-- Fix potential memory corruption in _slurm_rpc_epilog_complete as well as
_slurm_rpc_complete_job_allocation.
-- Fix cosmetic display of AccountingStorageEnforce option "nosteps" when
in use.
-- If a job can never be started due to unsatisfied job dependencies, report
the full original job dependency specification rather than the dependencies
remaining to be satisfied (typically NULL).
-- Refactor logic to synchronize active batch jobs and their script/environment
files, reducing overhead dramatically for large numbers of active jobs.
```

```

-- Avoid hard-link/copy of script/environment files for job arrays. Use the
  master job record file for all tasks of the job array.
  NOTE: Job arrays submitted to Slurm version 15.08.6 or later will fail if
  the slurmctld daemon is downgraded to an earlier version of Slurm.
-- Move slurmctld mail handler to separate thread for improved performance.
-- Fix containment of adopted processes from pam_slurm_adopt.
-- If a pending job array has multiple reasons for being in a pending state,
  then print all reasons in a comma separated list.

* Changes in Slurm 15.08.5
=====
-- Prevent "scontrol update job" from updating jobs that have already finished.
-- Show requested TRES in "squeue -O tres" when job is pending.
-- Backfill scheduler: Test association and QOS node limits before reserving
  resources for pending job.
-- burst_buffer/cray: If teardown operations fails, sleep and retry.
-- Clean up the external pids when using the PrologFlags=Contain feature
  and the job finishes.
-- burst_buffer/cray: Support file staging when job lacks job-specific buffer
  (i.e. only persistent burst buffers).
-- Added srun option of --bcast to copy executable file to compute nodes.
-- Fix for advanced reservation of burst buffer space.
-- BurstBuffer/cray: Add logic to terminate dw_wlm_cli child processes at
  shutdown.
-- If job can't be launch or requeued, then terminate it.
-- BurstBuffer/cray: Enable clearing of burst buffer string on completed job
  as a means of recovering from a failure mode.
-- Fix wrong memory free when parsing SrunPortRange=0-0 configuration.
-- BurstBuffer/cray: Fix job record purging if cancelled from pending state.
-- BGQ - Handle database throw correctly when syncing users on blocks.
-- MySQL - Make sure we don't have a NULL string returned when not
  requesting any specific association.
-- sched/backfill: If max_rpc_cnt is configured and the backlog of RPCs has
  not cleared after yielding locks, then continue to sleep.
-- Preserve the job dependency description displayed in 'scontrol show job'
  even if the dependee jobs was terminated and cleaned causing the
  dependent to never run because of DependencyNeverSatisfied.
-- Correct job task count calculation if only node count and ntasks-per-node
  options supplied.
-- Make sure the association manager converts any string to be lower case
  as all the associations from the database will be lower case.
-- Sanity check for xcgroup_delete() to verify incoming parameter is valid.
-- Fix formatting for sacct with variables that switched from uint32_t to
  uint64_t.
-- Fix a typo in sacct man page.
-- Set up extern step to track any children of an ssh if it leaves anything
  else behind.
-- Prevent slurmdbd divide by zero if no associations defined at rollup time.
-- Multifactor - Add sanity check to make sure pending jobs are handled
  correctly when PriorityFlags=CALCULATE_RUNNING is set.
-- Add slurmdb_find_tres_count_in_string() to slurm db perl api.
-- Make lua dlopen() conditional on version found at build.
-- sched/backfill - Delay backfill scheduler for completing jobs only if
  CompleteWait configuration parameter is set (make code match documentation).
-- Release a job's allocated licenses only after epilog runs on all nodes

```

Appendix G. SLURM Release Information

- rather than at start of termination process.
- Cray job NHC delayed until after burst buffer released and epilog completes on all allocated nodes.
- Fix abort of srun if using PrologFlags=NoHold
- Let devices step_extern cgroup inherit attributes of job cgroup.
- Add new hook to Task plugin to be able to put adopted processes in the step_extern cgroups.
- Fix AllowUsers documentation in burst_buffer.conf man page. Usernames are comma separated, not colon delimited.
- Fix issue with time limit not being set correctly from a QOS when a job requests no time limit.
- Various CLANG fixes.
- In both sched/basic and backfill: If a job can not be started due to some account/qos limit, then don't start other jobs which could delay jobs. The old logic would skip the job and start other jobs, which could delay the higher priority job.
- select/cray: Prevent NHC from running more than once per job or step.
- Fix fields not properly printed when adding an account through sacctmgr.
- Update LBNL Node Health Check (NHC) link on FAQ.
- Fix multifactor plugin to prevent slurmctld from getting segmentation fault should the tres_alloc_cnt be NULL.
- sbatch/salloc - Move nodelist logic before the time min_nodes is used so we can set it correctly before tasks are set.

* Changes in Slurm 15.08.4

=====

- Fix typo for the "devices" cgroup subsystem in pam_slurm_adopt.c
- Fix TRES_MAX flag to work correctly.
- Improve the systemd startup files.
- Added burst_buffer.conf flag parameter of "TeardownFailure" which will teardown and remove a burst buffer after failed stage-in or stage-out. By default, the buffer will be preserved for analysis and manual teardown.
- Prevent a core dump in srun if the signal handler runs during the job allocation causing the step context to be NULL.
- Don't fail job if multiple prolog operations in progress at slurmctld restart time.
- Burst_buffer/cray: Fix to purge terminated jobs with burst buffer errors.
- Burst_buffer/cray: Don't stall scheduling of other jobs while a stage-in is in progress.
- Make it possible to query 'extern' step with sstat.
- Make 'extern' step show up in the database.
- MYSQL - Quote assoc table name in mysql query.
- Make SLURM_ARRAY_TASK_MIN, SLURM_ARRAY_TASK_MAX, and SLURM_ARRAY_TASK_STEP environment variables available to PrologSlurmctld and EpilogSlurmctld.
- Fix slurmctld bug in which a pending job array could be canceled by a user different from the owner or the administrator.
- Support taking node out of FUTURE state with "scontrol reconfig" command.
- Sched/backfill: Fix to properly enforce SchedulerParameters of bf_max_job_array_resv.
- Enable operator to reset sdiag data.
- jobcomp/elasticsearch plugin: Add array_job_id and array_task_id fields.
- Remove duplicate #define IS_NODE_POWER_UP.
- Added SchedulerParameters option of max_script_size.
- Add REQUEST_ADD_EXTERN_PID option to add pid to the slurmstepd's extern step.

- Add unique identifiers to anchor tags in HTML generated from the man pages.
- Add with_freeipmi option to spec file.
- Minor elasticsearch code improvements

* Changes in Slurm 15.08.3

=====

- Correct Slurm's RPM build if Munge is not installed.
- Job array termination status email ExitCode based upon highest exit code from any task in the job array rather than the last task. Also change the state from "Ended" or "Failed" to "Mixed" where appropriate.
- Squeue recombines pending job array records only if their name and partition are identical.
- Fix some minor leaks in the job info and step info API.
- Export missing QOS id when filling in association with the association manager.
- Fix invalid reference if a lua job_submit plugin references a default qos when a user doesn't exist in the database.
- Use association enforcement in the lua plugin.
- Fix a few spots missing defines of accounting_enforce or acct_db_conn in the plugins.
- Show requested TRES in scontrol show jobs when job is pending.
- Improve sched/backfill support for job features, especially XOR construct.
- Correct scheduling logic for job features option with XOR construct that could delay a job's initiation.
- Remove unneeded frees when creating a tres string.
- Send a tres_alloc_str for the batch step
- Fix incorrect check for slurmdb_find_tres_count_in_string in various places, it needed to check for INFINITE64 instead of zero.
- Don't allow scontrol to create partitions with the name "DEFAULT".
- burst_buffer/cray: Change error from "invalid request" to "permssion denied" if a non-authorized user tries to create/destroy a persistent buffer.
- PrologFlags work: Setting a flag of "Contain" implicitly sets the "Alloc" flag. Fix code path which could prevent execution of the Prolog when the "Alloc" or "Contain" flag were set.
- Fix for acct_gather_energy/craylibmaem to work with missed enum.
- MYSQL - When inserting a job and begin_time is 0 do not set it to submit_time. 0 means the job isn't eligible yet so we need to treat it so.
- MYSQL - Don't display ineligible jobs when querying for a window of time.
- Fix creation of advanced reservation of cores on nodes which are DOWN.
- Return permission denied if regular user tries to release job held by an administrator.
- MYSQL - Fix rollups for multiple jobs running by the same association in an hour counting multiple times.
- Burstbuffer/Cray plugin - Fix for persistent burst buffer use. Don't call paths if no #DW options.
- Modifications to pam_slurm_adopt to work correctly for the "extern" step.
- Alphabetize debugflags when printing them out.
- Fix systemd's slurmd service from killing slurmdsteps on shutdown.
- Fixed counter of not indexed jobs, error_cnt post-increment changed to pre-increment.

* Changes in Slurm 15.08.2

=====

- Fix for tracking node state when jobs that have been allocated exclusive access to nodes (i.e. entire nodes) and later relinquish some nodes. Nodes

Appendix G. SLURM Release Information

would previously appear partly allocated and prevent use by other jobs.

- Correct some cgroup paths ("step_batch" vs. "step_4294967294", "step_exter" vs. "step_extern", and "step_extern" vs. "step_4294967295").
- Fix advanced reservation core selection logic with network topology.
- MYSQL - Remove restriction to have to be at least an operator to query TRES values.
- For pending jobs have sacct print 0 for nnodes instead of the bogus 2.
- Fix for tracking node state when jobs that have been allocated exclusive access to nodes (i.e. entire nodes) and later relinquish some nodes. Nodes would previously appear partly allocated and prevent use by other jobs.
- Fix updating job in db after extending job's timelimit past partition's timelimit.
- Fix srun -I<timeout> from flooding the controller with step create requests.
- Requeue/hold batch job launch request if job already running (possible if node went to DOWN state, but jobs remained active).
- If a job's CPUs/task ratio is increased due to configured MaxMemPerCPU, then increase it's allocated CPU count in order to enforce CPU limits.
- Don't mark powered down node as not responding. This could be triggered by race condition of the node suspend and ping logic, preventing use of the node.
- Don't requeue RPC going out from slurmctld to DOWN nodes (can generate repeating communication errors).
- Propagate sbatch "--dist=plane=#" option to srun.
- Add acct_gather_energy/ibmaem plugin for systems with IBM Systems Director Active Energy Manager.
- Fix spec file to look for mariadb or mysql devel packages for build requirements.
- MySQL - Improve the code with asking for jobs in a suspended state.
- Fix slurmctld allowing root to see job steps using squeues -s.
- Do not send burst buffer stage out email unless the job uses burst buffers.
- Fix sacct to not return all jobs if the -j option is given with a trailing ','.
- Permit job_submit plugin to set a job's priority.
- Fix occasional srun segfault.
- Fix issue with sacct, printing 0_0 for array's that had finished in the database but the start record hadn't made it yet.
- sacctmgr - Don't allow default account associations to be removed from a user.
- Fix sacct -j, (nothing but a comma) to not return all jobs.
- Fixed slurmctld not sending cold-start messages correctly to the database when a cold-start (-c) happens to the slurmctld.
- Fix case where if the backup slurmdbd has existing connections when it gives up control that the it would be killed.
- Fix task/cgroup affinity to work correctly with multi-socket single-threaded cores. A regression caused only 1 socket to be used on this kind of node instead of all that were available.
- MYSQL - Fix minor issue after an index was added to the database it would previously take 2 restarts of the slurmdbd to make it stick correctly.
- Add hv_to_qos_cond() and qos_rec_to_hv() functions to the Perl interface.
- Add new burst_buffer.conf parameters: ValidateTimeout and OtherTimeout. See man page for details.
- Fix burst_buffer/cray support for interactive allocations >4GB.
- Correct backfill scheduling logic for job with INFINITE time limit.
- Fix issue on a scontrol reconfig all available GRES/TRES would be zeroed out.

```
-- Set SLURM_HINT environment variable when --hint is used with sbatch or
salloc.
-- Add scancel -f/--full option to signal all steps including batch script and
all of its child processes.
-- Fix salloc -I to accept an argument.
-- Avoid reporting more allocated CPUs than exist on a node. This can be
triggered by resuming a previously suspended job, resulting in
oversubscription of CPUs.
-- Fix the pty window manager in slurmstepd not to retry IO operation with
srun if it read EOF from the connection with it.
-- sbatch --ntasks option to take precedence over --ntasks-per-node plus node
count, as documented. Set SLURM_NTASKS/SLURM_NPROCS environment variables
accordingly.
-- MYSQL - Make sure suspended time is only subtracted from the CPU TRES
as it is the only TRES that can be given to another job while suspended.
-- Clarify how TRESBillingWeights operates on memory and burst buffers.
```

* Changes in Slurm 15.08.1

=====

```
-- Fix test21.30 and 21.34 to check grpwall better.
-- Add time to the partition QOS the job is running on instead of just the
job QOS.
-- Print usage for GrpJobs, GrpSubmitJobs and GrpWall even if there is no
limit.
-- If AccountingEnforce=safe is set make sure a job can finish before going
over the limit with grpwall on a QOS or association.
-- burst_buffer/cray - Major updates based upon recent Cray changes.
-- Improve clean up logic of pmi2 plugin.
-- Improve job state reason string when required nodes not available.
-- Fix missing else when packing an update partition message
-- Fix srun from inheriting the SLURM_CPU_BIND and SLURM_MEM_BIND environment
variables when running in an existing srun (e.g. an srun within an salloc).
-- Fix missing else when packing an update partition message.
-- Use more flexible mechanism to find json installation.
-- Make sure safe_limits was initialized before processing limits in the
slurmctld.
-- Fix for burst_buffer/cray to parse type option correctly.
-- Fix memory error and version number in the nonstop plugin and reservation
code.
-- When requesting GRES in a step check for correct variable for the count.
-- Fix issue with GRES in steps so that if you have multiple exclusive steps
and you use all the GRES up instead of reporting the configuration isn't
available you hold the requesting step until the GRES is available.
-- MYSQL - Change debug to print out with DebugFlags=DB_Step instead of debug4
-- Simplify code when user is selecting a job/step/array id and removed
anomaly when only asking for 1 (task_id was never set to INFINITE).
-- MYSQL - If user is requesting various task_ids only return requested steps.
-- Fix issue when tres cnt for energy is 0 for total reported.
-- Resolved scalability issues of power adaptive scheduling with layouts.
-- Burst_buffer/cray bug - Fix teardown race condition that can result in
infinite loop.
-- Add support for --mail-type=NONE option.
-- Job "--reboot" option automatically, set's exclusive node mode.
-- Fix memory leak when using PrologFlags=Alloc.
-- Fix truncation of job reason in squeue.
```

Appendix G. SLURM Release Information

- If a node is in DOWN or DRAIN state, leave it unavailable for allocation when powered down.
- Update the slurm.conf man page documenting better nohold_on_prolog_fail variable.
- Don't truncate task ID information in "squeue --array/-r" or "svview".
- Fix a bug which caused scontrol to core dump when releasing or holding a job by name.
- Fix unit conversion bug in slurmd which caused wrong memory calculation for cgroups.
- Fix issue with GRES in steps so that if you have multiple exclusive steps and you use all the GRES up instead of reporting the configuration isn't available you hold the requesting step until the GRES is available.
- Fix slurmdbd backup to use DbdAddr when contacting the primary.
- Fix error in MPI documentation.
- Fix to handle arrays with respect to number of jobs submitted. Previously only 1 job was accounted (against MaxSubmitJob) for when an array was submitted.
- Correct counting for job array limits, job count limit underflow possible when master cancellation of master job record.
- Combine 2 _valid_uid_gid functions into a single function to avoid diversion.
- Pending job array records will be combined into single line by default, even if started and requeued or modified.
- Fix sacct --format=nnodes to print out correct information for pending jobs.
- Make is so 'scontrol update job 1234 qos=' will set the qos back to the default qos for the association.
- Add [Alloc|Req]Nodes to sacct to be more like cpus.
- Fix sacct documentation about [Alloc|Req]TRES
- Put node count in TRES string for steps.
- Fix issue with wrong protocol version when using the srun --no-allocate option.
- Fix TRES counts on GRES on a clean start of the slurmd.
- Add ability to change a job array's maximum running task count:
"scontrol update jobid=# arraytaskthrottle=#"

* Changes in Slurm 15.08.0

=====

- Fix issue with frontend systems (outside ALPs or BlueGene) where srun wouldn't get the correct protocol version to launch a step.
- Fix for message aggregation return rpcs where none of the messages are intended for the head of the tree.
- Fix segfault in sreport when there was no response from the dbd.
- ALPS - Fix compile to not link against -ljob and -lexpat with every lib or binary.
- Fix testing for CR_Memory when CR_Memory and CR_ONE_TASK_PER_CORE are used with select/linear.
- When restarting or reconfiging the slurmd, if job is completing handle accounting correctly to avoid meaningless errors about overflow.
- Add AccountingStorageTRES to scontrol show config
- MySQL - Fix minor memory leak if a connection ever goes away whist using it.
- ALPS - Make it so srun --hint=nomultithread works correctly.
- Make MaxTRESPerUser work in sacctmgr.
- Fix handling of requeued jobs with steps that are still finishing.
- Cleaner copy for PriorityWeightTRES, it also fixes a core dump when trying


```

to free it otherwise.
-- Add environment variables SLURM_ARRAY_TASK_MAX, SLURM_ARRAY_TASK_MIN,
   SLURM_ARRAY_TASK_STEP for job arrays.
-- Fix srun to use the NoInAddrAny TopologyParam option.
-- Change QOS flag name from PartitionQOS to OverPartQOS to be a better
   description.
-- Fix rpmbuild issue on Centos7.

* Changes in Slurm 15.08.0rc1
=====
-- Added power_cpufreq layout.
-- Make complete_batch_script RPC work with message aggregation.
-- Do not count slurmctld threads waiting in a "throttle" lock against the
   daemon's thread limit as they are not contending for resources.
-- Modify slurmctld outgoing RPC logic to support more parallel tasks (up to
   85 RPCs and 256 pthreads; the old logic supported up to 21 RPCs and 256
   threads). This change can dramatically improve performance for RPCs
   operating on small node counts.
-- Increase total backfill scheduler run time in stats_info_response_msg data
   structure from 32 to 64 bits in order to prevent overflow.
-- Add NoInAddrAny option to TopologyParam in the slurm.conf which allows to
   bind to the interface of return of gethostname instead of any address on
   the node which avoid RSIP issues in Cray systems. This is most likely
   useful in other systems as well.
-- Fix memory leak in Slurm::load_jobs perl api call.
-- Added --noconvert option to sacct, sstat, squeue and sinfo which allows
   values to be displayed in their original unit types (e.g. 2048M won't be
   converted to 2G).
-- Fix spelling of node_rescrs to node_resrcs in Perl API.
-- Fix node state race condition, UNKNOWN->IDLE without configuration info.
-- Cray: Disable LDAP references from slurmstepd on job launch due for
   improved scalability.
-- Remove srun "read header error" due to application termination race
   condition.
-- Optimize sacct queries with additional db indexes.
-- Add SLURM_TOPO_LEN env variable for scontrol show topology.
-- Add free_mem to node information.
-- Fix abort of batch launch if prolog is running, wait for prolog instead.
-- Fix case where job would get the wrong cpu count when using
   --ntasks-per-core and --cpus-per-task together.
-- Add TRESBillingWeights to partitions in slurm.conf which allows taking into
   consideration any TRES Type when calculating the usage of a job.
-- Add PriorityWeightTRES slurm.conf option to be able to configure priority
   factors for TRES types.

* Changes in Slurm 15.08.0pre6
=====
-- Add scontrol options to view and modify layouts tables.
-- Add MsgAggregationParams which controls a reverse tree to the slurmctld
   which can be used to aggregate messages to the slurmctld into a single
   message to reduce communication to the slurmctld. Currently only epilog
   complete messages and node registration messages use this logic.
-- Add sacct and squeue options to print trackable resources.
-- Add sacctmgr option to display trackable resources.
-- If an salloc or srun command is executed on a "front-end" configuration,

```

Appendix G. SLURM Release Information

that job will be assigned a slurmd shepherd daemon on the same host as used to execute the command when possible rather than an slurmd daemon on an arbitrary front-end node.

- Add srun --accel-bind option to control how tasks are bound to GPUs and NIC Generic RESources (GRES).
- gres/nic plugin modified to set OMPI_MCA_btl_openib_if_include environment variable based upon allocated devices (usable with OpenMPI and Mellanox).
- Make it so info options for srun/salloc/sbatch print with just 1 -v instead of 4.
- Add "no_backup_scheduling" SchedulerParameter to prevent jobs from being scheduled when the backup takes over. Jobs can be submitted, modified and cancelled while the backup is in control.
- Enable native Slurm backup controller to reside on an external Cray node when the "no_backup_scheduling" SchedulerParameter is used.
- Removed TICKET_BASED fairshare. Consider using the FAIR_TREE algorithm.
- Disable advanced reservation "REPLACE" option on IBM Bluegene systems.
- Add support for control distribution of tasks across cores (in addition to existing support for nodes and sockets, (e.g. "block", "cyclic" or "fcyclic" task distribution at 3 levels in the hardware rather than 2).
- Create db index on <cluster>_assoc_table.acct. Deleting accounts that didn't have jobs in the job table could take a long time.
- The performance of Profiling with HDF5 is improved. In addition, internal structures are changed to make it easier to add new profile types, particularly energy sensors. sh5util will continue to work with either format.
- Add partition information to sshare output if the --partition option is specified on the sshare command line.
- Add sreport -T/--tres option to identify Trackable RESources (TRES) to report.
- Display job in sacct when single step's cpus are different from the job allocation.
- Add association usage information to "scontrol show cache" command output.
- MPI/MVAPICH plugin now requires Munge for authentication.
- job_submit/lua: Add default_qos fields. Add job record qos. Add partition record allow_qos and qos_char fields.

* Changes in Slurm 15.08.0pre5
=====

- Add jobcomp/elasticsearch plugin. Libcurl is required for build. Configure the server as follows: "JobCompLoc=http://YOUR_ELASTICSEARCH_SERVER:9200".
- Scancel logic large re-written to better support job arrays.
- Added a slurm.conf parameter PrologEpilogTimeout to control how long prolog/epilog can run.
- Added TRES (Trackable resources) to track Mem, GRES, license, etc utilization.
- Add re-entrant versions of glibc time functions (e.g. localtime) to Slurm in order to eliminate rare deadlock of slurmd fork and exec calls.
- Constrain kernel memory (if available) in cgroups.
- Add PrologFlags option of "Contain" to create a proctrack container at job resource allocation time.
- Disable the OOM Killer in slurmd and slurmd's memory cgroup when using MemSpecLimit.

* Changes in Slurm 15.08.0pre4
=====

```

-- Burst_buffer/cray - Convert logic to use new commands/API names (e.g.
   "dws_setup" rather than "bbs_setup").
-- Remove the MinJobAge size limitation. It can now exceed 65533 as it
   is represented using an unsigned integer.
-- Verify that all plugin version numbers are identical to the component
   attempting to load them. Without this verification, the plugin can reference
   Slurm functions in the caller which differ (e.g. the underlying function's
   arguments could have changed between Slurm versions).
   NOTE: All plugins (except SPANK) must be built against the identical
   version of Slurm in order to be used by any Slurm command or daemon. This
   should eliminate some very difficult to diagnose problems due to use of old
   plugins.
-- Increase the MAX_PACK_MEM_LEN define to avoid PMI2 failure when fencing
   with large amount of ranks (to 1GB).
-- Requests by normal user to reset a job priority (even to lower it) will
   result in an error saying to change the job's nice value instead.
-- SPANK naming changes: For environment variables set using the
   spank_job_control_setenv() function, the values were available in the
   slurm_spank_job_prolog() and slurm_spank_job_epilog() functions using
   getenv where the name was given a prefix of "SPANK_". That prefix has
   been removed for consistency with the environment variables available in
   the Prolog and Epilog scripts.
-- Major additions to the layouts framework code.
-- Add "TopologyParam" configuration parameter. Optional value of "dragonfly"
   is supported.
-- Optimize resource allocation for systems with dragonfly networks.
-- Add "--thread-spec" option to salloc, sbatch and srun commands. This is
   the count of threads reserved for system use per node.
-- job_submit/lua: Enable reading and writing job environment variables.
   For example: if (job_desc.environment.LANGUAGE == "en_US") then ...
-- Added two new APIs slurm_job_cpus_allocated_str_on_node_id()
   and slurm_job_cpus_allocated_str_on_node() to print the CPUs id
   allocated to a job.
-- Specialized memory (a node's MemSpecLimit configuration parameter) is not
   available for allocation to jobs.
-- Modify scontrol update job to allow jobid specification without
   the = sign. 'scontrol update job=123 ...' and 'scontrol update job 123 ...'
   are both valid syntax.
-- Archive a month at a time when there are lots of records to archive.
-- Introduce new sbatch option '--kill-on-invalid-dep=yes|no' which allows
   users to specify which behavior they want if a job dependency is not
   satisfied.
-- Add Slurmdb::qos_get() interface to perl api.
-- If a job fails to start set the requeue reason to be:
   job requeued in held state.
-- Implemented a new MPI key,value PMIX_RING() exchange algorithm as
   an alternative to PMI2.
-- Remove possible deadlocks in the slurmd when the slurmdbd is busy
   archiving/purging.
-- Add DB_ARCHIVE debug flag for filtering out debug messages in the slurmdbd
   when the slurmdbd is archiving/purging.
-- Fix some power_save mode issues: Parsing of SuspendTime in slurm.conf was
   bad, powered down nodes would get set non-responding if there was an
   in-flight message, and permit nodes to be powered down from any state.
-- Initialize variables in consumable resource plugin to prevent core dump.

```

Appendix G. SLURM Release Information

* Changes in Slurm 15.08.0pre3

=====

- CRAY - addition of acct_gather_energy/cray plugin.
- Add job credential to "Run Prolog" RPC used with a configuration of PrologFlags=alloc. This allows the Prolog to be passed identification of GPUs allocated to the job.
- Add SLURM_JOB_CONSTRAINTS to environment variables available to the Prolog.
- Added "--mail=stage_out" option to job submission commands to notify user when burst buffer state out is complete.
- Require a "Reason" when using scontrol to set a node state to DOWN.
- Mail notifications on job BEGIN, END and FAIL now apply to a job array as a whole rather than generating individual email messages for each task in the job array.
- task/affinity - Fix memory binding to NUMA with cpusets.
- Display job's estimated NodeCount based off of partition's configured resources rather than the whole system's.
- Add AuthInfo option of "cred_expire=#" to specify the lifetime of a job step credential. The default value was changed from 1200 to 120 seconds.
- Set the delay time for job requeue to the job credential lifetime (120 seconds by default). This insures that prolog runs on every node when a job is requeued. (This change will slow down launch of re-queued jobs).
- Add AuthInfo option of "cred_expire=#" to specify the lifetime of a job step credential.
- Remove srun --max-launch-time option. The option has not been functional since Slurm version 2.0.
- Add sockets and cores to TaskPluginParams' autobind option.
- Added LaunchParameters configuration parameter. Have srun command test locally for the executable file if LaunchParameters=test_exec or the environment variable SLURM_TEST_EXEC is set. Without this an invalid command will generate one error message per task launched.
- Fix the slurm /etc/init.d script to return 0 upon stopping the daemons and return 1 in case of failure.
- Add the ability for a compute node to be allocated to multiple jobs, but restricted to a single user. Added "--exclusive=user" option to salloc, sbatch and srun commands. Added "owner" field to node record, visible using the scontrol and sview commands. Added new partition configuration parameter "ExclusiveUser=yes|no".

* Changes in Slurm 15.08.0pre2

=====

- Add the environment variables SLURM_JOB_ACCOUNT, SLURM_JOB_QOS and SLURM_JOB_RESERVATION in the batch/srun jobs.
- Add sview burst buffer display.
- Properly enforce partition Shared=YES option. Previously oversubscribing resources required gang scheduling to be configured.
- Enable per-partition gang scheduling resource resolution (e.g. the partition can have SelectTypeParameters=CR_CORE, while the global value is CR_SOCKET).
- Make it so a newer version of a slurmstepd can talk to an older srun allocation. Nodes could have been added while waiting for an allocation.
- Expanded --cpu-freq parameters to include min-max:governor specifications. --cpu-freq now supported on salloc and sbatch.
- Add support for optimized job allocations with respect to SGI Hypercube topology.
NOTE: Only supported with select/linear plugin.

NOTE: The program contribs/sgi/netloc_to_topology can be used to build Slurm's topology.conf file.

- Remove 64k validation of incoming RPC nodelist size. Validated at 64MB when unpacking.
- In slurmstepd() add the user primary group if it is not part of the groups sent from the client.
- Added BurstBuffer field to advanced reservations.
- For advanced reservation, replace flag "License_only" with flag "Any_Nodes". It can be used to indicate the an advanced reservation resources (licenses and/or burst buffers) can be used with any compute nodes.
- Allow users to specify the srun --resv-ports as 0 in which case no ports will be reserved. The default behaviour is to allocate one port per task.
- Interpret a partition configuration of "Nodes=ALL" in slurm.conf as including all nodes defined in the cluster.
- Added new configuration parameters PowerParameters and PowerPlugin.
- Added power management plugin infrastructure.
- If job already exceeded one of its QOS/Accounting limits do not return error if user modifies QOS unrelated job settings.
- Added DebugFlags value of "Power".
- When caching user ids of AllowGroups use both getgrnam_r() and getgrent_r() then remove eventual duplicate entries.
- Remove rpm dependency between slurm-pam and slurm-devel.
- Remove support for the XCPU (cluster management) package.
- Add Slurmdb::jobs_get() interface to perl api.
- Performance improvement when sending data from srun to steps when processing fencing.
- Add the feature to specify arbitrary field separator when running sacct -p or sacct -P. The command line option is --separator.
- Introduce slurm.conf parameter to use Proportional Set Size (PSS) instead of RSS to determinate the memory footprint of a job.
Add an slurm.conf option not to kill jobs that is over memory limit.
- Add job submission command options: --sicmp (available for inter-cluster dependencies) and --power (specify power management options) to salloc, sbatch, and srun commands.
- Add DebugFlags option of SICP (inter-cluster option logging).
- In order to support inter-cluster job dependencies, the MaxJobID configuration parameter default value has been reduced from 4,294,901,760 to 2,147,418,112 and it's maximum value is now 2,147,463,647.
ANY JOBS WITH A JOB ID ABOVE 2,147,463,647 WILL BE PURGED WHEN SLURM IS UPGRADED FROM AN OLDER VERSION!
- Add QOS name to the output of a partition in squeue/scontrol/sview/smap.

* Changes in Slurm 15.08.0pre1
=====

- Add sbcast support for file transfer to resources allocated to a job step rather than a job allocation.
- Change structures with association in them to assoc to save space.
- Add support for job dependencies jointed with OR operator (e.g. "--depend=afterok:123?afternotok:124").
- Add "--bb" (burst buffer specification) option to salloc, sbatch, and srun.
- Added configuration parameters BurstBufferParameters and BurstBufferType.
- Added burst_buffer plugin infrastructure (needs many more functions).
- Make it so when the fanout logic comes across a node that is down we abandon the tree to avoid worst case scenarios when the entire branch is down and we have to try each serially.

Appendix G. SLURM Release Information

- Add better error reporting of invalid partitions at submission time.
- Move will-run test for multiple clusters from the sbatch code into the API so that it can be used with DRMAA.
- If a non-exclusive allocation requests --hint=nomultithread on a CR_CORE/SOCKET system lay out tasks correctly.
- Avoid including unused CPUs in a job's allocation when cores or sockets are allocated.
- Added new job state of STOPPED indicating processes have been stopped with a SIGSTOP (using scancel or svview), but retain its allocated CPUs. Job state returns to RUNNING when SIGCONT is sent (also using scancel or svview).
- Added EioTimeout parameter to slurm.conf. It is the number of seconds srun waits for slurmstepd to close the TCP/IP connection used to relay data between the user application and srun when the user application terminates.
- Remove slurmctld/dynalloc plugin as the work was never completed, so it is not worth the effort of continued support at this time.
- Remove DynAllocPort configuration parameter.
- Add advance reservation flag of "replace" that causes allocated resources to be replaced with idle resources. This maintains a pool of available resources that maintains a constant size (to the extent possible).
- Added SchedulerParameters option of "bf_busy_nodes". When selecting resources for pending jobs to reserve for future execution (i.e. the job can not be started immediately), then preferentially select nodes that are in use. This will tend to leave currently idle resources available for backfilling longer running jobs, but may result in allocations having less than optimal network topology. This option is currently only supported by the select/cons_res plugin.
- Permit "SuspendTime=NONE" as slurm.conf value rather than only a numeric value to match "scontrol show config" output.
- Add the 'scontrol show cache' command which displays the associations in slurmctld.
- Test more frequently for node boot completion before starting a job. Provides better responsiveness.
- Fix PMI2 singleton initialization.
- Permit PreemptType=qos and PreemptMode=suspend,gang to be used together. A high-priority QOS job will now oversubscribe resources and gang schedule, but only if there are insufficient resources for the job to be started without preemption. NOTE: That with PreemptType=qos, the partition's Shared=FORCE:# configuration option will permit one job more per resource to be run than than specified, but only if started by preemption.
- Remove the CR_ALLOCATE_FULL_SOCKET configuration option. It is now the default.
- Fix a race condition in PMI2 when fencing counters can be out of sync.
- Increase the MAX_PACK_MEM_LEN define to avoid PMI2 failure when fencing with large amount of ranks.
- Add QOS option to a partition. This will allow a partition to have all the limits a QOS has. If a limit is set in both QOS the partition QOS will override the job's QOS unless the job's QOS has the OverPartQOS flag set.
- The task_dist_states variable has been split into "flags" and "base" components. Added SLURM_DIST_PACK_NODES and SLURM_DIST_NO_PACK_NODES values to give user greater control over task distribution. The srun --dist options has been modified to accept a "Pack" and "NoPack" option. These options can be used to override the CR_PACK_NODE configuration option.

* Changes in Slurm 14.11.12

```

=====
-- Correct dependency formatting to print array task ids if set.
-- Fix for configuration of "AuthType=munge" and "AuthInfo=socket=..." with
  alternate munge socket path.
-- BGQ - Remove redeclaration of job_read_lock.
-- BGQ - Tighter locks around structures when nodes/cables change state.
-- Fix job array formatting to allow return [0-100:2] display for arrays with
  step functions rather than [0,2,4,6,8,...] .
-- Associations - prevent hash table corruption if uid initially unset for
  a user, which can cause slurmctld to crash if that user is deleted.
-- Add cast to memory limit calculation to prevent integer overflow for
  very large memory values.
-- Fix test cases to have proper int return signature.

* Changes in Slurm 14.11.11
=====
-- Fix systemd's slurmd service from killing slurmstepds on shutdown.
-- Fix the qstat wrapper when user is removed from the system but still
  has running jobs.
-- Log the request to terminate a job at info level if DebugFlags includes
  the Steps keyword.
-- Fix potential memory corruption in _slurm_rpc_epilog_complete as well as
  _slurm_rpc_complete_job_allocation.
-- Fix incorrectly sized buffer used by jobid2str which will cause buffer
  overflow in slurmctld. (Bug 2295.)

* Changes in Slurm 14.11.10
=====
-- Fix truncation of job reason in squeue.
-- If a node is in DOWN or DRAIN state, leave it unavailable for allocation
  when powered down.
-- Update the slurm.conf man page documenting better nohold_on_prolog_fail
  variable.
-- Don't truncate task ID information in "squeue --array/-r" or "sview".
-- Fix a bug which caused scontrol to core dump when releasing or
  holding a job by name.
-- Fix unit conversion bug in slurmd which caused wrong memory calculation
  for cgroups.
-- Fix issue with GRES in steps so that if you have multiple exclusive steps
  and you use all the GRES up instead of reporting the configuration isn't
  available you hold the requesting step until the GRES is available.
-- Fix slurmdbd backup to use DbdAddr when contacting the primary.
-- Fix error in MPI documentation.
-- Fix to handle arrays with respect to number of jobs submitted. Previously
  only 1 job was accounted (against MaxSubmitJob) for when an array was
  submitted.
-- Correct counting for job array limits, job count limit underflow possible
  when master cancellation of master job record.
-- For pending jobs have sacct print 0 for nnodes instead of the bogus 2.
-- Fix for tracking node state when jobs that have been allocated exclusive
  access to nodes (i.e. entire nodes) and later relinquish some nodes. Nodes
  would previously appear partly allocated and prevent use by other jobs.
-- Fix updating job in db after extending job's timelimit past partition's
  timelimit.
-- Fix srun -I<timeout> from flooding the controller with step create requests.

```

Appendix G. SLURM Release Information

- Requeue/hold batch job launch request if job already running (possible if node went to DOWN state, but jobs remained active).
- If a job's CPUs/task ratio is increased due to configured MaxMemPerCPU, then increase it's allocated CPU count in order to enforce CPU limits.
- Don't mark powered down node as not responding. This could be triggered by race condition of the node suspend and ping logic.
- Don't requeue RPC going out from slurmctld to DOWN nodes (can generate repeating communication errors).
- Propagate sbatch "--dist=plane=#" option to srun.
- Fix sacct to not return all jobs if the -j option is given with a trailing ','.
- Permit job_submit plugin to set a job's priority.
- Fix occasional srun segfault.
- Fix issue with sacct, printing 0_0 for array's that had finished in the database but the start record hadn't made it yet.
- Fix sacct -j, (nothing but a comma) to not return all jobs.
- Prevent slurmstepd from core dumping if /proc/<pid>/stat has unexpected format.

* Changes in Slurm 14.11.9

=====

- Correct "sdiag" backfill cycle time calculation if it yields locks. A microsecond value was being treated as a second value resulting in an overflow in the calculation.
- Fix segfault when updating timelimit on jobarray task.
- Fix to job array update logic that can result in a task ID of 4294967294.
- Fix of job array update, previous logic could fail to update some tasks of a job array for some fields.
- CRAY - Fix seg fault if a blade is replaced and slurmctld is restarted.
- Fix plane distribution to allocate in blocks rather than cyclically.
- squeue - Remove newline from job array ID value printed.
- squeue - Enable filtering for job state SPECIAL_EXIT.
- Prevent job array task ID being inappropriately set to NO_VAL.
- MYSQL - Make it so you don't have to restart the slurmctld to gain the correct limit when a parent account is root and you remove a subaccount's limit which exists on the parent account.
- MYSQL - Close chance of setting the wrong limit on an association when removing a limit from an association on multiple clusters at the same time.
- MYSQL - Fix minor memory leak when modifying an association but no change was made.
- srun command line of either --mem or --mem-per-cpu will override both the SLURM_MEM_PER_CPU and SLURM_MEM_PER_NODE environment variables.
- Prevent slurmctld abort on update of advanced reservation that contains no nodes.
- ALPS - Revert commit 2c95e2d22 which also removes commit 2e2de6a4 allowing cray with the SubAllocate option to work as it did with 2.5.
- Properly parse CPU frequency data on POWER systems.
- Correct sacct.a man pages describing -i option.
- Capture salloc/srun information in sdiag statistics.
- Fix bug in node selection with topology optimization.
- Don't set distribution when srun requests 0 memory.
- Read in correct number of nodes from SLURM_HOSTFILE when specifying nodes and --distribution=arbitrary.
- Fix segfault in Bluegene setups where RebootQOSList is defined in


```

bluegene.conf and accounting is not setup.
-- MYSQL - Update mod_time when updating a start job record or adding one.
-- MYSQL - Fix issue where if an association id ever changes on at least a
portion of a job array is pending after it's initial start in the
database it could create another row for the remain array instead
of using the already existing row.
-- Fix scheduling anomaly with job arrays submitted to multiple partitions,
jobs could be started out of priority order.
-- If a host has suspended jobs do not reboot it. Reboot only hosts
with no jobs in any state.
-- ALPS - Fix issue when using --exclusive flag on srun to do the correct
thing (-F exclusive) instead of -F share.
-- Fix various memory leaks in the Perl API.
-- Fix a bug in the controller which display jobs in CF state as RUNNING.
-- Preserve advanced _core_ reservation when nodes added/removed/resized on
slurmctld restart. Rebuild core_bitmap as needed.
-- Fix for non-standard Munge port location for srun/pmi use.
-- Fix gang scheduling/preemption issue that could cancel job at startup.
-- Fix a bug in squeue which prevented squeue -tPD to print array jobs.
-- Sort job arrays in job queue according to array_task_id when priorities are
equal.
-- Fix segfault in sreport when there was no response from the dbd.
-- ALPS - Fix compile to not link against -ljob and -lexpat with every lib
or binary.
-- Fix testing for CR_Memory when CR_Memory and CR_ONE_TASK_PER_CORE are used
with select/linear.
-- MySQL - Fix minor memory leak if a connection ever goes away whist using it.
-- ALPS - Make it so srun --hint=nomultithread works correctly.
-- Prevent job array task ID from being reported as NO_VAL if last task in the
array gets requeued.
-- Fix some potential deadlock issues when state files don't exist in the
association manager.
-- Correct RebootProgram logic when executed outside of a maintenance
reservation.
-- Requeue job if possible when slurmstepd aborts.

```

* Changes in Slurm 14.11.8

=====

```

-- Eliminate need for user to set user_id on job_update calls.
-- Correct list of unavailable nodes reported in a job's "reason" field when
that job can not start.
-- Map job --mem-per-cpu=0 to --mem=0.
-- Fix squeue -o %m and %d unit conversion to Megabytes.
-- Fix issue with incorrect time calculation in the priority plugin when
a job runs past it's time limit.
-- Prevent users from setting job's partition to an invalid partition.
-- Fix sreport core dump when requesting
'job SizesByAccount grouping=individual'.
-- select/linear: Correct count of CPUs allocated to job on system with
hyperthreads.
-- Fix race condition where last array task might not get updated in the db.
-- CRAY - Remove libpmi from rpm install
-- Fix squeue -o %X output to correctly handle NO_VAL and suffix.
-- When deleting a job from the system set the job_id to 0 to avoid memory
corruption if thread uses the pointer basing validity off the id.

```

Appendix G. SLURM Release Information

- Fix issue where sbatch would set ntasks-per-node to 0 making any srun afterward cause a divide by zero error.
- switch/cray: Refine logic to set PMI_CRAY_NO_SMP_ENV environment variable.
- When sacctmgr loads archives with version less than 14.11 set the array task id to NO_VAL, so sacct can display the job ids correctly.
- When using memory cgroup if a task uses more memory than requested the failures are logged into memory.failcnt count file by cgroup and the user is notified by slurmstepd about it.
- Fix scheduling inconsistency with GRES bound to specific CPUs.
- If user belongs to a group which has split entries in /etc/group search for its username in all groups.
- Do not consider nodes explicitly powered up as DOWN with reason of "Node unexpected rebooted".
- Use correct slurmd spooldir when creating cpu-frequency locks.
- Note that TICKET_BASED fairshare will be deprecated in the future. Consider using the FAIR_TREE algorithm instead.
- Set job's reason to BadConstraints when job can't run on any node.
- Prevent abort on update of reservation with no nodes (licenses only).
- Prevent slurmctld from dumping core if job_resrcs is missing in the job data structure.
- Fix squeue to print array task ids according to man page when SLURM_BITSTR_LEN is defined in the environment.
- In squeue, sort jobs based on array job ID if available.
- Fix the calculation of job energy by not including the NO_VAL values.
- Advanced reservation fixes: enable update of bluegene reservation, avoid abort on multi-core reservations.
- Set the totalview_stepid to the value of the job step instead of NO_VAL.
- Fix slurmdbd core dump if the daemon does not have connection with the database.
- Display error message when attempting to modify priority of a held job.
- Backfill scheduler: The configured backfill_interval value (default 30 seconds) is now interpreted as a maximum run time for the backfill scheduler. Once reached, the scheduler will build a new job queue and start over, even if not all jobs have been tested.
- Backfill scheduler now considers OverTimeLimit and KillWait configuration parameters to estimate when running jobs will exit.
- Correct task layout with CR_Pack_Node option and more than 1 CPU per task.
- Fix the scontrol man page describing the release argument.
- When job QOS is modified, do so before attempting to change partition in order to validate the partition's Allow/DenyQOS parameter.

* Changes in Slurm 14.11.7

=====

- Initialize some variables used with the srun --no-alloc option that may cause random failures.
- Add SchedulerParameters option of sched_min_interval that controls the minimum time interval between any job scheduling action. The default value is zero (disabled).
- Change default SchedulerParameters=max_sched_time from 4 seconds to 2.
- Refactor scancel so that all pending jobs are cancelled before starting cancellation of running jobs. Otherwise they happen in parallel and the pending jobs can be scheduled on resources as the running jobs are being cancelled.
- ALPS - Add new cray.conf variable NoAPIDSignalOnKill. When set to yes this will make it so the slurmctld will not signal the apid's in a batch job.

```

Instead it relies on the rpc coming from the slurmctld to kill the job to
end things correctly.
-- ALPS - Have the slurmstepd running a batch job wait for an ALPS release
before ending the job.
-- Initialize variables in consumable resource plugin to prevent core dump.
-- Fix scancel bug which could return an error on attempt to signal a job step.
-- In slurmctld communication agent, make the thread timeout be the configured
value of MessageTimeout rather than 30 seconds.
-- sshare -U/--Users only flag was used uninitialized.
-- Cray systems, add "plugstack.conf.template" sample SPANK configuration file.
-- BLUEGENE - Set DB2NOEXITLIST when starting the slurmctld daemon to avoid
random crashing in db2 when the slurmctld is exiting.
-- Make full node reservations display correctly the core count instead of
cpu count.
-- Preserve original errno on execve() failure in task plugin.
-- Add SLURM_JOB_NAME env variable to an salloc's environment.
-- Overwrite SLURM_JOB_NAME in an srun when it gets an allocation.
-- Make sure each job has a wckey if that is something that is tracked.
-- Make sure old step data is cleared when job is requeued.
-- Load libtinfo as needed when building ncurses tools.
-- Fix small memory leak in backup controller.
-- Fix segfault when backup controller takes control for second time.
-- Cray - Fix backup controller running native Slurm.
-- Provide prototypes for init_setproctitle()/fini_setproctitle on NetBSD.
-- Add configuration test to find out the full path to su command.
-- preempt/job_prio plugin: Fix for possible infinite loop when identifying
preemptable jobs.
-- preempt/job_prio plugin: Implement the concept of Warm-up Time here. Use
the QoS GraceTime as the amount of time to wait before preempting.
Basically, skip preemption if your time is not up.
-- Make srun wait KillWait time when a task is cancelled.
-- switch/cray: Revert logic added to 14.11.6 that set "PMI_CRAY_NO_SMP_ENV=1"
if CR_PACK_NODES is configured.

```

* Changes in Slurm 14.11.6

=====

```

-- If SchedulerParameters value of bf_min_age_reserve is configured, then
a newly submitted job can start immediately even if there is a higher
priority non-runnable job which has been waiting for less time than
bf_min_age_reserve.
-- qsub wrapper modified to export "all" with -V option
-- RequeueExit and RequeueExitHold configuration parameters modified to accept
numeric ranges. For example "RequeueExit=1,2,3,4" and "RequeueExit=1-4" are
equivalent.
-- Correct the job array specification parser to accept brackets in job array
expression (e.g. "123_[4,7-9]").
-- Fix for misleading job submit failure errors sent to users. Previous error
could indicate why specific nodes could not be used (e.g. too small memory)
when other nodes could be used, but were not for another reason.
-- Fix squeue --array to display correctly the array elements when the
% separator is specified at the array submission time.
-- Fix priority from not being calculated correctly due to memory issues.
-- Fix a transient pending reason 'JobId=job_id has invalid QOS'.
-- A non-administrator change to job priority will not be persistent except
for holding the job. User's wanting to change a job priority on a persistent

```

Appendix G. SLURM Release Information

basis should reset it's "nice" value.

- Print buffer sizes as unsigned values when failed to pack messages.
- Fix race condition where sprio would print factors without weights applied.
- Document the sacct option JobIDRaw which for arrays prints the jobid instead of the arrayTaskId.
- Allow users to modify MinCPUsNode, MinMemoryNode and MinTmpDiskNode of their own jobs.
- Increase the jobid print field in SQUEUE_FORMAT in opt_modulefiles_slurm.in.
- Enable compiling without optimizations and with debugging symbols by default. Disable this by configuring with --disable-debug.
- job_submit/lua plugin: Add mail_type and mail_user fields.
- Correct output message from sshare.
- Use standard statvfs(2) syscall if available, in preference to non-standard statfs.
- Add a new option -U/--Users to sshare to display only users information, parent and ancestors are not printed.
- Purge 50000 records at a time so that locks can be released periodically.
- Fix potentially uninitialized variables
- ALPS - Fix issue where a frontend node could become unresponsive and never added back into the system.
- Gate epilog complete messages as done with other messages
- If we have more than a certain number of agents (50) wait longer when gating rpcs.
- FrontEnd - ping non-responding or down nodes.
- switch/cray: If CR_PACK_NODES is configured, then set the environment variable "PMI_CRAY_NO_SMP_ENV=1"
- Fix invalid memory reference in SlurmDBD when putting a node up.
- Allow opening of plugstack.conf even when a symlink.
- Fix scontrol reboot so that rebooted nodes will not be set down with reason 'Node xyz unexpectedly rebooted' but will be correctly put back to service.
- CRAY - Throttle the post NHC operations as to not hog the job write lock if many steps/jobs finish at once.
- Disable changes to GRES count while jobs are running on the node.
- CRAY - Fix issue with scontrol reconfig.
- slurmd: Remove wrong reporting of "Error reading step ... memory limit". The logic was treating success as an error.
- Eliminate "Node ping apparently hung" error messages.
- Fix average CPU frequency calculation.
- When allocating resources with resolution of sockets, charge the job for all CPUs on allocated sockets rather than just the CPUs on used cores.
- Prevent slurmdbd error if cluster added or removed while rollup in progress. Removing a cluster can cause slurmdbd to abort. Adding a cluster can cause the slurmdbd rollup to hang.
- svview - When right clicking on a tab make sure we don't display the page list, but only the column list.
- FRONTEND - If doing a clean start make sure the nodes are brought up in the database.
- MySQL - Fix issue when using the TrackSlurmctldDown and nodes are down at the same time, don't double bill the down time.
- MySQL - Various memory leak fixes.
- sreport - Fix Energy displays
- Fix node manager logic to keep unexpectedly rebooted node in state NODE_STATE_DOWN even if already down when rebooted.
- Fix for array jobs submitted to multiple partitions not starting.

```
-- CRAY - Enable ALPs mpp compatibility code in sbatch for native Slurm.
-- ALPS - Move basil_inventory to less confusing function.
-- Add SchedulerParameters option of "sched_max_job_start=" to limit the
  number of jobs that can be started in any single execution of the main
  scheduling logic.
-- Fixed compiler warnings generated by gcc version >= 4.6.
-- sbatch to stop parsing script for "#SBATCH" directives after first command,
  which matches the documentation.
-- Overwrite the SLURM_JOB_NAME in sbatch if already exist in the environment
  and use the one specified on the command line --job-name.
-- Remove xmalloc_nz from unpack functions.  If the unpack ever failed the
  free afterwards would not have zeroed out memory on the variables that
  didn't get unpacked.
-- Improve database interaction from controller.
-- Fix for data shift when loading job archives.
-- ALPS - Added new SchedulerParameters=inventory_interval to specify how
  often an inventory request is handled.
-- ALPS - Don't run a release on a reservation on the slurmctld for a batch
  job.  This is already handled on the stepd when the script finishes.
```

* Changes in Slurm 14.11.5

=====

```
-- Correct the squeue command taking into account that a node can
  have NULL name if it is not in DNS but still in slurm.conf.
-- Fix slurmdbd regression which would cause a segfault when a node is set
  down with no reason.
-- BGQ - Fix issue with job arrays not being handled correctly
  in the runjob_mux plugin.
-- Print FAIR_TREE, if configured, in "scontrol show config" output for
  PriorityFlags.
-- Add SLURM_JOB_GPUS environment variable to those available in the Prolog.
-- Load lua-5.2 library if using lua5.2 for lua job submit plugin.
-- GRES logic: Prevent bad node_offset due to not preserving no_consume flag.
-- Fix wrong variables used in the wrapper functions needed for systems that
  don't support strong_alias
-- Fix code for apple computers SOL_TCP is not defined
-- Cray/BASIL - Check for mysql credentials in /root/.my.cnf.
-- Fix sprio showing wrong priority for job arrays until priority is
  recalculated.
-- Account to batch step all CPUs that are allocated to a job not
  just one since the batch step has access to all CPUs like other steps.
-- Fix job getting EligibleTime set before meeting dependency requirements.
-- Correct the initialization of QOS MinCPUs per job limit.
-- Set the debug level of information messages in cgroup plugin to debug2.
-- For job running under a debugger, if the exec of the task fails, then
  cancel its I/O and abort immediately rather than waiting 60 seconds for
  I/O timeout.
-- Fix associations not getting default qos set until after a restart.
-- Set the value of total_cpus not to be zero before invoking
  acct_policy_job_runnable_post_select.
-- MySQL - When requesting cluster resources, only return resources for the
  cluster(s) requested.
-- Add TaskPluginParam=autobind=threads option to set a default binding in the
  case that "auto binding" doesn't find a match.
-- Introduce a new SchedulerParameters variable nohold_on_prolog_fail.
```

Appendix G. SLURM Release Information

```
If configured don't requeue jobs on hold is a Prolog fails.
-- Make it so sched_params isn't read over and over when an epilog complete
  message comes in
-- Fix squeue -L <licenses> not filtering out jobs with licenses.
-- Changed the implementation of xcputinfo_abs_to_mac() be identical
  _abs_to_mac() to fix CPUs allocation using cpuset cgroup.
-- Improve the explanation of the unbuffered feature in the
  srun man page.
-- Make taskplugin=cgroup work for core spec.  needed to have task/cgroup
  before.
-- Fix reports not using the month usage table.
-- BGQ - Sanity check given for translating small blocks into slurm bg_records.
-- Fix bug preventing the requeue/hold or requeue/special_exit of job from the
  completing state.
-- Cray - Fix for launching batch step within an existing job allocation.
-- Cray - Add ALPS_APP_ID_ENV environment variable.
-- Increase maximum MaxArraySize configuration parameter value from 1,000,001
  to 4,000,001.
-- Added new SchedulerParameters value of bf_min_age_reserve. The backfill
  scheduler will not reserve resources for pending jobs until they have
  been pending for at least the specified number of seconds. This can be
  valuable if jobs lack time limits or all time limits have the same value.
-- Fix support for --mem=0 (all memory of a node) with select/cons_res plugin.
-- Fix bug that can permit someone to kill job array belonging to another user.
-- Don't set the default partition on a license only reservation.
-- Show a NodeCnt=0, instead of NO_VAL, in "scontrol show res" for a license
  only reservation.
-- BGQ - When using static small blocks make sure when clearing the job the
  block is set up to it's original state.
-- Start job allocation using lowest numbered sockets for block task
  distribution for consistency with cyclic distribution.
```

* Changes in Slurm 14.11.4

=====

```
-- Make sure assoc_mgr locks are initialized correctly.
-- Correct check of enforcement when filling in an association.
-- Make sacctmgr print out classification correctly for clusters.
-- Add array_task_str to the perlapi job info.
-- Fix for slurmctld abort with GRES types configured and no CPU binding.
-- Fix for GRES scheduling where count > 1 per topology type (or GRES types).
-- Make CR_ONE_TASK_PER_CORE work correctly with task/affinity.
-- job_submit/pbs - Fix possible deadlock.
-- job_submit/lua - Add "alloc_node" to job information available.
-- Fix memory leak in mysql accounting when usage rollup happens.
-- If users specify ALL together with other variables using the
  --export sbatch/srun command line option, propagate the users'
  environ to the execution side.
-- Fix job array scheduling anomaly that can stop scheduling of valid tasks.
-- Fix perl api tests for libslurmdb to work correctly.
-- Remove some misleading logs related to non-consumable GRES.
-- Allow --ignore-pbs to take effect when read as an #SBATCH argument.
-- Fix Slurmdb::clusters_get() in perl api from not returning information.
-- Fix TaskPluginParam=Cpusets from logging error message about not being able
  to remove cpuset dir which was already removed by the release_agent.
-- Fix sorting by time left in squeue.
```

```
-- Fix the file name substitution for job stderr when %A, %a %j and %u
are specified.
-- Remove minor warning when compiling slurmstepd.
-- Fix database resources so they can add new clusters to them after they have
initially been added.
-- Use the slurm_getpwuid_r wrapper of getpwuid_r to handle possible
interrupts.
-- Correct the scontrol man page and command listing which node states can
be set by the command.
-- Stop sacct from printing non-existent stat information for
Front End systems.
-- Correct srun and acct_gather.conf man pages, mention Filesystem instead
of Lustre.
-- When a job using multiple partition starts send to slurmdbd only
the partition in which the job runs.
-- ALPS - Fix depth for MemoryAllocation in BASIL with CLE 5.2.3.
-- Fix assoc_mgr hash to deal with users that don't have a uid yet when making
reservations.
-- When a job uses multiple partition set the environment variable
SLURM_JOB_PARTITION to be the one in which the job started.
-- Print spurious message about the absence of cgroup.conf at log level debug2
instead of info.
-- Enable CUDA v7.0+ use with a Slurm configuration of TaskPlugin=task/cgroup
ConstrainDevices=yes (in cgroup.conf). With that configuration
CUDA_VISIBLE_DEVICES will start at 0 rather than the device number.
-- Fix job array logic that can cause slurmctld to abort.
-- Report job "shared" field properly in scontrol, squeue, and sview.
-- If a job is requeued because of RequeueExit or RequeueExitHold sent event
REQUEUED to slurmdbd.
-- Fix build if hwloc is in non-standard location.
-- Fix slurmctld job recovery logic which could cause the last task in a job
array to be lost.
-- Fix slurmctld initialization problem which could cause requeue of the last
task in a job array to fail if executed prior to the slurmctld loading
the maximum size of a job array into a variable in the job_mgr.c module.
-- Fix fatal in controller when deleting a user association of a user which
had been previously removed from the system.
-- MySQL - If a node state and reason are the same on a node state change
don't insert a new row in the event table.
-- Fix issue with "sreport cluster AccountUtilizationByUser" when using
PrivateData=users.
-- Fix perlapi tests for libslurm perl module.
-- MySQL - Fix potential issue when PrivateData=Usage and a normal user
runs certain sreport reports.
```

* Changes in Slurm 14.11.3

=====

```
-- Prevent vestigial job record when canceling a pending job array record.
-- Fixed squeue core dump.
-- Fix job array hash table bug, could result in slurmctld infinite loop or
invalid memory reference.
-- In srun honor ntasks_per_node before looking at cpu count when the user
doesn't request a number of tasks.
-- Fix ghost job when submitting job after all jobids are exhausted.
-- MySQL - Enhanced coordinator security checks.
```

Appendix G. SLURM Release Information

- Fix for task/affinity if an admin configures a node for having threads but then sets CPUs to only represent the number of cores on the node.
- Make it so previous versions of salloc/srun work with newer versions of Slurm daemons.
- Avoid delay on commit for PMI rank 0 to improve performance with some MPI implementations.
- auth/munge - Correct logic to read old format AccountingStoragePass.
- Reset node "RESERVED" state as appropriate when deleting a maintenance reservation.
- Prevent a job manually suspended from being resumed by gang scheduler once free resources are available.
- Prevent invalid job array task ID value if a task is started using gang scheduling.
- Fixes for clean build on FreeBSD.
- Fix documentation bugs in slurm.conf.5. DenyAccount should be DenyAccounts.
- For backward compatibility with older versions of OMPI not compiled with --with-pmi restore the SLURM_STEP_RESV_PORTS in the job environment.
- Update the html documentation describing the integration with openmpi.
- Fix sacct when searching by nodelist.
- Fix cosmetic info statements when dealing with a job array task instead of a normal job.
- Fix segfault with job arrays.
- Correct the sbatch pbs parser to process -j.
- BGQ - Put print statement under a DebugFlag. This was just an oversight.
- BLUEGENE - Remove check that would erroneously remove the CONFIGURING flag from a job while the job is waiting for a block to boot.
- Fix segfault in slurmstepd when job exceeded memory limit.
- Fix race condition that could start a job that is dependent upon a job array before all tasks of that job array complete.
- PMI2 race condition fix.

* Changes in Slurm 14.11.2

=====

- Fix Centos5 compile errors.
- Fix issue with association hash not getting the correct index which could result in seg fault.
- Fix salloc/sbatch -B segfault.
- Avoid huge malloc if GRES configured with "Type" and huge "Count".
- Fix jobs from starting in overlapping reservations that won't finish before a "maint" reservation begins.
- When node gets drained while in state mixed display its status as draining in sinfo output.
- Allow priority/multifactor to work with sched/wiki(2) if all priorities have no weight. This allows for association and QOS decay limits to work.
- Fix "squeue --start" to override SQUEUE_FORMAT env variable.
- Fix scancel to be able to cancel multiple jobs that are space delimited.
- Log Cray MPI job calling exit() without mpi_fini(), but do not treat it as a fatal error. This partially reverts logic added in version 14.03.9.
- svview - Fix displaying of suspended steps elapsed times.
- Increase number of messages that get cached before throwing them away when the DBD is down.
- Fix jobs from starting in overlapping reservations that won't finish before a "maint" reservation begins.
- Restore GRES functionality with select/linear plugin. It was broken in version 14.03.10.


```
-- Fix bug with GRES having multiple types that can cause slurmctld abort.
-- Fix squeue issue with not recognizing "localhost" in --odelist option.
-- Make sure the bitstrings for a partitions Allow/DenyQOS are up to date
  when running from cache.
-- Add smap support for job arrays and larger job ID values.
-- Fix possible race condition when attempting to use QOS on a system running
  accounting_storage/filetxt.
-- Fix issue with accounting_storage/filetxt and job arrays not being printed
  correctly.
-- In proctrack/linuxproc and proctrack/pgid, check the result of strtol()
  for error condition rather than errno, which might have a vestigial error
  code.
-- Improve information recording for jobs deferred due to advanced
  reservation.
-- Exports eio_new_initial_obj to the plugins and initialize kvs_seq on
  mpi/pmi2 setup to support launching.
```

* Changes in Slurm 14.11.1

=====

```
-- Get libs correct when doing the xtree/xhash make check.
-- Update xhash/tree make check to work correctly with current code.
-- Remove the reference 'experimental' for the jobacct_gather/cgroup
  plugin.
-- Add QOS manipulation examples to the qos.html documentation page.
-- If 'squeue -w node_name' specifies an unknown host name print
  an error message and return 1.
-- Fix race condition in job_submit plugin logic that could cause slurmctld to
  deadlock.
-- Job wait reason of "ReqNodeNotAvail" expanded to identify unavailable nodes
  (e.g. "ReqNodeNotAvail(Unavailable:tux[3-6])").
```

* Changes in Slurm 14.11.0

=====

```
-- ALPS - Fix issue with core_spec warning.
-- Allow multiple partitions to be specified in sinfo -p.
-- Install the service files in /usr/lib/systemd/system.
-- MYSQL - Add id_array_job and id_resv keys to $CLUSTER_job_table.  THIS
  COULD TAKE A WHILE TO CREATE THE KEYS SO BE PATIENT.
-- CRAY - Resize bitmaps on a restart and find we have more blades
  than before.
-- Add new eio API function for removing unused connections.
-- ALPS - Fix issue where batch allocations weren't correctly confirmed or
  released.
-- Define DEFAULT_MAX_TASKS_PER_NODE based on MAX_TASKS_PER_NODE from
  slurm.h as per documentation.
-- Update the FAQ about relocating slurmctld.
-- In the memory cgroup enable memory.use_hierarchy in the cgroup root.
-- Export eio.c functions for use by MPI/PMI2.
-- Add SLURM_CLUSTER_NAME to job environment.
```

* Changes in Slurm 14.11.0rc3

=====

```
-- Allow envs to override autotools binaries in autogen.sh
-- Added system services files.
-- If the jobs pends with DependencyNeverSatisfied keep it pending even after
```

Appendix G. SLURM Release Information

```
the job which it was depending upon was cleaned.
-- Let operators (in addition to user root and SlurmUser) see job script for
other user's jobs.
-- Perl API modified to return node state of MIXED rather than ALLOCATED if
only some CPUs allocated.
-- Double Munge connect retry timeout from 1 to 2 seconds.
-- svview - Remove unneeded code that was resolved globally in commit
98e24b0dedc.
-- Collect and report the accounting of the batch step and its children.
-- Add configure checks for faccessat and eaccess, and make use of one of
them if available.
-- Make configure --enable-developer also set --enable-debug
-- Introduce a SchedulerParameters variable kill_invalid_depend, if set
then jobs pending with invalid dependency are going to be terminated.
-- Move spank_user_task() call in slurmstepd after the task_g_pre_launch()
so that the task affinity information is available to spank.
-- Make /etc/init.d/slurm script return value 3 when the daemon is
not running. This is required by Linux Standard Base Core
Specification 3.1

* Changes in Slurm 14.11.0rc2
=====
-- Logs for jobs which are explicitly requeued will say so rather than saying
that a node in their allocation failed.
-- Updated the documentation about the remote licenses served by
the Slurm database.
-- Insure that slurm_spank_exit() is only called once from srun.
-- Change the signature of net_set_low_water() to use 4 bytes instead of 8.
-- Export working_cluster_rec in libslurmdb.so as well as move some function
definitions needed for drmaa.
-- If using cons_res or serial cause a fatal in the plugin instead of causing
the SelectTypeParameters to magically set to CR_CPU.
-- Enhance task/affinity auto binding to consider tasks * cpus-per-task.
-- Fix regression the priority/multifactor which would cause memory corruption.
Issue is only in rcl.
-- Add PrivateData value of "cloud". If set, powered down nodes in the cloud
will be visible.
-- Sched/backfill - Eliminate clearing start_time of running jobs.
-- Fix various backwards compatibility issues.
-- If failed to launch a batch job, requeue it in hold.

* Changes in Slurm 14.11.0rc1
=====
-- When using cgroup name the batch step as step_batch instead of
batch_4294967294
-- Changed LEVEL_BASED priority to be "Fair_Tree"
-- Port to NetBSD.
-- BGQ - Add cnode based reservations.
-- Alongside totalview_jobid implement totalview_stepid available
to sattach.
-- Add ability to include other files in slurm.conf based upon the ClusterName.
-- Update strlcpy to latest upstream version.
-- Add reservation information in the sacct and sreport output.
-- Add job priority calculation check for overflow and fix memory leak.
-- Add SchedulerParameters option of pack_serial_at_end to put serial jobs at
```

```

the end of the available nodes rather than using a best fit algorithm.
-- Allow regular users to view default sinfo output when
   privatedata=reservations is set.
-- PrivateData=reservation modified to permit users to view the reservations
   which they have access to (rather than preventing them from seeing ANY
   reservation).
-- job_submit/lua: Fix job_desc set field logic

```

* Changes in Slurm 14.11.0pre5

```
=====
```

```

-- Fix sbatch --export=ALL, it was treated by srun as a request to explicitly
   export only the environment variable named "ALL".
-- Improve scheduling of jobs in reservations that overlap other reservations.
-- Modify sgather to make global file systems easier to configure.
-- Added sacctmgr reconfig to reread the slurmdbd.conf in the slurmdbd.
-- Modify scontrol job operations to accept comma delimited list of job IDs.
   Applies to job update, hold, release, suspend, resume, requeue, and
   requeuehold operations.
-- Refactor job_submit/lua interface. LUA FUNCTIONS NEED TO CHANGE! The
   lua script no longer needs to explicitly load meta-tables, but information
   is available directly using names slurm.reservations, slurm.jobs,
   slurm.log_info, etc. Also, the job_submit.lua script is reloaded when
   updated without restarting the slurmctld daemon.
-- Allow users to specify --resv_ports to have value 0.
-- Cray MPMD (Multiple-Program Multiple-Data) support completed.
-- Added ability for "scontrol update" to references jobs by JobName (and
   filtered optionally by UserID).
-- Add support for an advanced reservation start time that remains constant
   relative to the current time. This can be used to prevent the starting of
   longer running jobs on select nodes for maintenance purpose. See the
   reservation flag "TIME_FLOAT" for more information.
-- Enlarge the jobid field to 18 characters in squeue output.
-- Added "scontrol write config" option to save a copy of the current
   configuration in a file containing a time stamp.
-- Eliminate native Cray specific port management. Native Cray systems must
   now use the MpiParams configuration parameter to specify ports to be used
   for communications. When upgrading Native Cray systems from version 14.03,
   all running jobs should be killed and the switch_cray_state file (in
   SaveStateLocation of the nodes where the slurmctld daemon runs) must be
   explicitly deleted.

```

* Changes in Slurm 14.11.0pre4

```
=====
```

```

-- Added job array data structure and removed 64k array size restriction.
-- Added SchedulerParameters options of bf_max_job_array_resv to control how
   many tasks of a job array should have resources reserved for them.
-- Added more validity checking of incoming job submit requests.
-- Added srun --export option to set/export specific environment variables.
-- Scontrol modified to print separate error messages for job arrays with
   different exit codes on the different tasks of the job array. Applies to
   job suspend and resume operations.
-- Fix race condition in CPU frequency set with job preemption.
-- Always call select plugin on step termination, even if the job is also
   complete.
-- Srun executable names beginning with "." will be resolved based upon the

```

Appendix G. SLURM Release Information

```
working directory and path on the compute node rather than the submit node.
-- Add node state string suffix of "$" to identify nodes in maintenance
reservation or scheduled for reboot. This applies to scontrol, sinfo,
and sview commands.
-- Enable scontrol to clear a nodes's scheduled reboot by setting its state
to "RESUME".
-- As per sbatch and srun documentation when the --signal option is used
signal only the steps and unless, in the case, of a batch job B is
specified in which case signal only the batch script.
-- Modify AuthInfo configuration parameter to accept credential lifetime
option.
-- Modify crypto/munge plugin to use socket and timeout specified in AuthInfo.
-- If we have a state for a step on completion put that in the database
instead of guessing off the exit_code.
-- Added squeue -P/--priority option that can be used to display pending jobs
in the same order as used by the Slurm scheduler even if jobs are submitted
to multiple partitions (job is reported once per usable partition).
-- Improve the pending reason description for various QOS limits. For each
QOS limit that causes a job to be pending print its specific reason.
For example if job pends because of GrpCpus the squeue command will
print QOSGrpCpuLimit as pending reason.
-- sched/backfill - Set expected start time of job submitted to multiple
partitions to the earliest start time on any of the partitions.
-- Introduce a MAX_BATCH_QUEUE define that indicates how many times a job
can be requeued upon prolog failure. When the number is reached the job
is put on hold with reason JobHoldMaxRequeue.
-- Add sbatch job array option to limit the number of simultaneously running
tasks from a job array (e.g. "--array=0-15%4").
-- Implemented a new QOS limit MinCPUs. Users running under a QOS must
request a minimum number of CPUs which is at least MinCPUs otherwise
their job will pend.
-- Introduced a new pending reason WAIT_QOS_MIN_CPUS to reflect the new QOS
limit.
-- Job array dependency based upon state is now dependent upon the state of
the array as a whole (e.g. afterok requires ALL tasks to complete
successfully, afternotok is true if ANY tasks does not complete successfully,
and after requires all tasks to at least be started).
-- The srun -u/--unbuffered options set the stdout of the task launched
by srun to be line buffered.
-- The srun options -/--label and -u/--unbuffered can be specified together.
This limitation has been removed.
-- Provide sacct display of gres accounting information per job.
-- Change the node status size from uin16_t to uint32_t.

* Changes in Slurm 14.11.0pre3
=====
-- Move xcputinfo.[c|h] to the slurmd since it isn't needed anywhere else
and will avoid the need for all the daemons to link to libhwloc.
-- Add memory test to job_submit/partition plugin.
-- Added new internal Slurm functions xmalloc_nz() and xrealloc_nz(), which do
not initialize the allocated memory to zero for improved performance.
-- Modify hostlist function to dynamically allocate buffer space for improved
performance.
-- In the job_submit plugin: Remove all slurmd locks prior to job_submit()
being called for improved performance. If any slurmd data structures are
```

```

read or modified, add locks directly in the plugin.
-- Added PriorityFlag LEVEL_BASED described in doc/html/level_based.shtml
-- If Fairshare=parent is set on an account, that account's children will be
effectively reparented for fairshare calculations to the first parent of
their parent that is not Fairshare=parent. Limits remain the same,
only it's fairshare value is affected.

* Changes in Slurm 14.11.0pre2
=====
-- Added AllowSpecResourcesUsage configuration parameter in slurm.conf. This
allows jobs to use specialized resources on nodes allocated to them if the
job designates --core-spec=0.
-- Add new SchedulerParameters option of build_queue_timeout to throttle how
much time can be consumed building the job queue for scheduling.
-- Added HealthCheckNodeState option of "cycle" to cycle through the compute
nodes over the course of HealthCheckInterval rather than running all at
the same time.
-- Add job "reboot" option for Linux clusters. This invokes the configured
RebootProgram to reboot nodes allocated to a job before it begins execution.
-- Added squeue -O/--Format option that makes all job and step fields available
for printing.
-- Improve database slurmctld entry speed dramatically.
-- Add "CPUs" count to output of "scontrol show step".
-- Add support for lua5.2
-- scancel -b signals only the batch step neither any other step nor any
children of the shell script.
-- MySQL - enforce NO_ENGINE_SUBSTITUTION
-- Added CpuFreqDef configuration parameter in slurm.conf to specify the
default CPU frequency and governor to be set at job end.
-- Added support for job email triggers: TIME_LIMIT, TIME_LIMIT_90 (reached
90% of time limit), TIME_LIMIT_80 (reached 80% of time limit), and
TIME_LIMIT_50 (reached 50% of time limit). Applies to salloc, sbatch and
srun commands.
-- In slurm.conf add the parameter SrunPortRange=min-max. If this is configured
then srun will use its dynamic ports only from the configured range.
-- Make debug_flags 64 bit to handle more flags.

* Changes in Slurm 14.11.0pre1
=====
-- Modify etc/cgroup.release_common.example to set specify full path to the
scontrol command. Also find cgroup mount point by reading cgroup.conf file.
-- Improve qsub wrapper support for passing environment variables.
-- Modify sdiag to report Slurm RPC traffic by user, type, count and time
consumed.
-- In select plugins, stop triggering extra logging based upon the debug flag
CPU_Bind and use SelectType instead.
-- Added SchedulerParameters options of bf_yield_interval and bf_yield_sleep
to control how frequently and for how long the backfill scheduler will
relinquish its locks.
-- To support larger numbers of jobs when the StateSaveDirectory is on a
file system that supports a limited number of files in a directory, add a
subdirectory called "hash.#" based upon the last digit of the job ID.
-- More gracefully handle missing batch script file. Just kill the job and do
not drain the compute node.
-- Add support for allocation of GRES by model type for heterogenous systems

```

Appendix G. SLURM Release Information

(e.g. request a Kepler GPU, a Tesla GPU, or a GPU of any type).

- Record and enable display of nodes anticipated to be used for pending jobs.
- Modify `squeue --start` option to print the nodes expected to be used for pending job (in addition to expected start time, etc.).
- Add association hash to the `assoc_mgr`.
- Better logic to handle resized jobs when the DBD is down.
- Introduce `MemLimitEnforce yes|no` in `slurm.conf`. If set no Slurm will not terminate jobs if they exceed requested memory.
- Add support for non-consumable generic resources for resources that are limited, but can be shared between jobs.
- Introduce 5 new Slurm errors in `slurm_errno.h` related to job to better report error conditions.
- Modify `scontrol` to print error message for each array task when updating the entire array.
- Added `gres_drain` and `gres_used` fields to `node_info_t`.
- Added `PriorityParameters` configuration parameter in `slurm.conf`.
- Introduce automatic job requeue policy based on exit value. See `RequeueExit` and `RequeueExitHold` descriptions in `slurm.conf` man page.
- Modify `slurmd` to cache launched job IDs for more responsive job suspend and gang scheduling.
- Permit jobs steps full control over `cpu_bind` options if specialized cores are included in the job allocation.
- Added `ChosLoc` configuration parameter to specify the pathname of the Chroot OS tool.
- Sent `SIGCONT/SIGTERM` when a job is selected for preemption with `GraceTime` configured rather than waiting for `GraceTime` to be reached before notifying the job.
- Do not resume a job with specialized cores on a node running another job with specialized cores (only one can run at a time).
- Add specialized core count to job suspend/resume calls.
- `task/affinity` and `task/cgroup` - Correct specialized core task binding with user supplied invalid CPU mask or map.
- Add `srun --cpu-freq` options to set the CPU governor (`OnDemand`, `Performance`, `PowerSave` or `UserSpace`).
- Add support for a job step's CPU governor and/or frequency to be reset on suspend/resume (or gang scheduling). The default for an idle CPU will now be "ondemand" rather than "userspace" with the lowest frequency (to recover from hard `slurmd` failures and support gang scheduling).
- Added `PriorityFlags` option of `Calculate_Running` to continue recalculating the priority of running jobs.
- Replace round-robin front-end node selection with least-loaded algorithm.
- CRAY - Improve support of XC30 systems when running natively.
- Add new node configuration parameters `CoreSpecCount`, `CPUSpecList` and `MemSpecLimit` which support the reservation of resources for system use with Linux `cgroup`.
- Add `child_forked()` function to the `slurm_acct_gather_profile` plugin to close open files, leaving application with no extra open file descriptors.
- Cray/ALPS system - Enable backup controller to run outside of the Cray to accept new job submissions and most other operations on the pending jobs.
- Have `sacct` print job and task array id's for job arrays.
- Smooth out fanout logic
- If `<sys/prctl.h>` is present name major threads in `slurmctld`, for example `backfill`
thread: `slurmctld_bckfl`, the rpc manager: `slurmctld_rpcmg` etc.
The name can be seen for example using `top -H`.

```
-- sview - Better job_array support.
-- Provide more precise error message when job allocation can not be satisfied
  (e.g. memory, disk, cpu count, etc. rather than just "node configuration
  not available").
-- Create a new DebugFlags named TraceJobs in slurm.conf to print detailed
  information about jobs in slurmctld. The information include job ids, state
  and node count.
-- When a job dependency can never be satisfied do not cancel the job but keep
  pending with reason WAIT_DEP_INVALID (DependencyNeverSatisfied).
```

* Changes in Slurm 14.03.12

=====

```
-- Make it so previous versions of salloc/srun work with newer versions
  of Slurm daemons.
-- PMI2 race condition fix.
-- Avoid delay on commit for PMI rank 0 to improve performance with some
  MPI implementations.
-- Correct the sbatch pbs parser to process -j.
-- Squeue modified to not merge tasks of a job array if their wait reasons
  differ.
-- Use the slurm_getpwuid_r wrapper of getpwuid_r to handle possible
  interrupts.
-- Allow --ignore-pbs to take effect when read as an #SBATCH argument.
-- Do not launch step if job killed while the prolog was running.
```

* Changes in Slurm 14.03.11

=====

```
-- ALPS - Fix depth for Memory items in BASIL with CLE 5.2
  (changed starting in 5.2.3).
-- ALPS - Fix issue when tracking memory on a PerNode basis instead of
  PerCPU.
-- Modify assoc_mgr_fill_in_qos() to allow for a flag to know if the QOS read
  lock was locked outside of the function or not.
-- Give even better estimates on pending node count if no node count
  is requested.
-- Fix jobcomp/mysql plugin for MariaDB 10+/Mysql 5.6+ to work with reserved
  work "partition".
-- If requested (scontrol reboot node_name) reboot a node even if it has
  an maintenance reservation that is not active yet.
-- Fix issue where exclusive allocations wouldn't lay tasks out correctly
  with CR_PACK_NODES.
-- Do not requeue a batch job from slurmd daemon if it is killed while in
  the process of being launched (a race condition introduced in v14.03.9).
-- Do not let srun overwrite SLURM_JOB_NUM_NODES if already in an allocation.
-- Prevent a job's end_time from being too small after a basil reservation
  error.
-- Fix sbatch --ntasks-per-core option from setting invalid
  SLURM_NTASKS_PER_CORE environment value.
-- Prevent scancel abort when no job satisfies filter options.
-- ALPS - Fix --ntasks-per-core option on multiple nodes.
-- Double max string that Slurm can pack from 16MB to 32MB to support
  larger MPI2 configurations.
-- Fix Centos5 compile issues.
-- Log Cray MPI job calling exit() without mpi_fini(), but do not treat it as
  a fatal error. This partially reverts logic added in version 14.03.9.
```

Appendix G. SLURM Release Information

- svview - Fix displaying of suspended steps elapsed times.
- Increase number of messages that get cached before throwing them away when the DBD is down.
- Fix jobs from starting in overlapping reservations that won't finish before a "maint" reservation begins.
- Fix "squeue --start" to override SQUEUE_FORMAT env variable.
- Restore GRES functionality with select/linear plugin. It was broken in version 14.03.10.
- Fix possible race condition when attempting to use QOS on a system running accounting_storage/filetxt.
- Sanity check for Correct QOS on startup.

* Changes in Slurm 14.03.10

=====

- Fix a few sacctmgr error messages.
- Treat non-zero SlurmSchedLogLevel without SlurmSchedLogFile as a fatal error.
- Correct sched_config.html documentation SchedulingParameters should be SchedulerParameters.
- When using gres and cgroup ConstrainDevices set correct access permission for the batch step.
- Fix minor memory leak in jobcomp/mysql on slurmctld reconfig.
- Fix bug that prevented preservation of a job's GRES bitmap on slurmctld restart or reconfigure (bug was introduced in 14.03.5 "Clear record of a job's gres when requeued" and only applies when GRES mapped to specific files).
- BGQ: Fix race condition when job fails due to hardware failure and is requeued. Previous code could result in slurmctld abort with NULL pointer.
- Prevent negative job array index, which could cause slurmctld to crash.
- Fix issue with squeue/scontrol showing correct node_cnt when only tasks are specified.
- Check the status of the database connection before using it.
- ALPS - If an allocation requests -n set the BASIL -N option to the amount of tasks / number of node.
- ALPS - Don't set the env var APRUN_DEFAULT_MEMORY, it is not needed anymore.
- Fix potential buffer overflow.
- Give better estimates on pending node count if no node count is requested.
- BLUEGENE - Fix issue where requeuing jobs could cause an assert.

* Changes in Slurm 14.03.9

=====

- If slurmd fails to stat(2) the configuration print the string describing the error code.
- Fix for mixing core base reservations with whole node based reservations to avoid overlapping erroneously.
- BLUEGENE - Remove references to Base Partition.
- svview - If compiled on a non-bluegene system then used to view a BGQ fix to allow svview to display blocks correctly.
- Fix bug in update reservation. When modifying the reservation the end time was set incorrectly.
- The start time of a reservation that is in ACTIVE state cannot be modified.
- Update the cgroup documentation about release agent for devices.
- MYSQL - fix for setting up preempt list on a QOS for multiple QOS.
- Correct a minor error in the scancel.1 man page related to the --signal option.


```

-- Enhance the scancel.1 man page to document the sequence of signals sent
-- Fix slurmstepd core dump if the cgroup hierarchy is not completed
   when terminating the job.
-- Fix hostlist_shift to be able to give correct node names on names with a
   different number of dimensions than the cluster.
-- BLUEGENE - Fix invalid pointer in corner case in the plugin.
-- Make sure on a reconfigure the select information for a node is preserved.
-- Correct logic to support job GRES specification over 31 bits (problem
   in logic converting int to uint32_t).
-- Remove logic that was creating GRES bitmap for node when not needed (only
   needed when GRES mapped to specific files).
-- BLUEGENE - Fix sinfo -tr before it would only print idle nodes correctly.
-- BLUEGENE - Fix for licenses_only reservation on bluegene systems.
-- svview - Verify pointer before using strchr.
-- -M option on tools talking to a Cray from a non-Cray fixed.
-- CRAY - Fix rpmbuild issue for missing file slurm.conf.template.
-- Fix race condition when dealing with removing many associations at
   different times when reservations are using the associations that are
   being deleted.
-- When a node's state is set to power_down/power_up, then execute
   SuspendProgram/ResumeProgram even if previously executed for that node.
-- Fix logic determining when job configuration (i.e. running node power up
   logic) is complete.
-- Setting the state of a node in powered down state node to "resume" will
   no longer cause it to reboot, but only clear the "drain" state flag.
-- Fix srun documentation to remove SLURM_NODELIST being equivalent as the -w
   option (since it isn't).
-- Fix issue with --hint=nomultithread and allocations with steps running
   arbitrary layouts (test1.59).
-- PrivateData=reservation modified to permit users to view the reservations
   which they have access to (rather than preventing them from seeing ANY
   reservation). Backport from 14.11 commit 77c2bd25c.
-- Fix PrivateData=reservation when using associations to give privileges to
   a reservation.
-- Better checking to see if select plugin is linear or not.
-- Add support for time specification of "fika" (3 PM).
-- Standardize qstat wrapper more.
-- Provide better estimate of minimum node count for pending jobs using more
   job parameters.
-- ALPS - Add SubAllocate to cray.conf file for those who like the way <=2.5
   did the ALPS reservation.
-- Safer check to avoid invalid reads when shutting down the slurmctld with
   lots of jobs.
-- Fix minor memory leak in the backfill scheduler when shutting down.
-- Add ArchiveResvs to the output of sacctmgr show config and init the variable
   on slurmdbd startup.
-- SLURMDBD - Only set the archive flag if purging the object
   (i.e ArchiveJobs PurgeJobs). This is only a cosmetic change.
-- Fix for job step memory allocation logic if step requests GRES and memory
   is not allocations are not managed.
-- Fix sinfo to display mixed nodes as allocated in '%F' output.
-- Svview - Fix cpu and node counts for partitions.
-- Ignore NO_VAL in SLURMDBD_PURGE_* macros.
-- ALPS - Don't drain nodes if epilog fails. It leaves them in drain state
   with no way to get them out.

```

Appendix G. SLURM Release Information

- Fix issue with task/affinity oversubscribing cpus erroneously when using --ntasks-per-node.
- MYSQL - Fix load of archive files.
- Treat Cray MPI job calling exit() without mpi_fini() as fatal error for that specific task and let srun handle all timeout logic.
- Fix small memory leak in jobcomp/mysql.
- Correct tracking of licenses for suspended jobs on slurmctld reconfigure or restart.
- If failed to launch a batch job requeue it in hold.

* Changes in Slurm 14.03.8

=====

- Fix minor memory leak when Job doesn't have nodes on it (Meaning the job has finished)
- Fix sinfo/sview to be able to query against nodes in reserved and other states.
- Make sbatch/salloc read in (SLURM|(SBATCH|SALLOC))_HINT in order to handle sruns in the script that will use it.
- srun properly interprets a leading "." in the executable name based upon the working directory of the compute node rather than the submit host.
- Fix Lustre misspellings in hdf5 guide
- Fix wrong reference in slurm.conf man page to what --profile option should be used for AcctGatherFilesystemType.
- Update HDF5 document to point out the SlurmdUser is who creates the ProfileHDF5Dir directory as well as all it's sub-directories and files.
- CRAY NATIVE - Remove error message for srun's ran inside an salloc that had --network= specified.
- Defer job step initiation of required GRES are in use by other steps rather than immediately returning an error.
- Deprecate --cpu_bind from sbatch and salloc. These never worked correctly and only caused confusion since the cpu_bind options mostly refer to a step we opted to only allow srun to set them in future versions.
- Modify sgather to work if Nodename and NodeHostname differ.
- Changed use of JobContainerPlugin where it should be JobContainerType.
- Fix for possible error if job has GRES, but the step explicitly requests a GRES count of zero.
- Make "srun --gres=none ..." work when executed without a job allocation.
- Change the global eio_shutdown_time to a field in eio handle.
- Advanced reservation fixes for heterogeneous systems, especially when reserving cores.
- If --hint=nomultithread is used in a job allocation make sure any srun's ran inside the allocation can read the environment correctly.
- If batchdir can't be made set errno correctly so the slurmctld is notified correctly.
- Remove repeated batch complete if batch directory isn't able to be made since the slurmd will send the same message.
- sacctmgr fix default format for list transactions.
- BLUEGENE - Fix backfill issue with backfilling jobs on blocks already reserved for higher priority jobs.
- When creating job arrays the job specification files for each elements are hard links to the first element specification files. If the controller fails to make the links the files are copied instead.
- Fix error handling for job array create failure due to inability to copy job files (script and environment).
- Added patch in the contribs directory for integrating make version 4.0 with

```

Slurm and renamed the previous patch "make-3.81.slurm.patch".
-- Don't wait for an update message from the DBD to finish before sending rc
message back. In slow systems with many associations this could speed
responsiveness in sacctmgr after adding associations.
-- Eliminate race condition in enforcement of MaxJobCount limit for job arrays.
-- Fix anomaly allocating cores for GRES with specific device/CPU mapping.
-- cons_res - When requesting exclusive access make sure we set the number
of cpus in the job_resources_t structure so as nodes finish the correct
cpu count is displayed in the user tools.
-- If the job_submit plugin calls take longer than 1 second to run, print a
warning.
-- Make sure transfer_s_p_options transfers all the portions of the
s_p_options_t struct.
-- Correct the srun man page, the SLURM_CPU_BIND_VERBOSE, SLURM_CPU_BIND_TYPE
SLURM_CPU_BIND_LIST environment variable are set only when task/affinity
plugin is configured.
-- sacct - Initialize variables correctly to avoid incorrect structure
reference.
-- Performance adjustment to avoid calling a function multiple times when it
only needs to be called once.
-- Give more correct waiting reason if job is waiting on association/QOS
MaxNode limit.
-- DB - When sending lft updates to the slurmctld only send non-deleted lfts.
-- BLUEGENE - Fix documentation on how to build a reservation less than
a midplane.
-- If Slurmctld fails to read the job environment consider it an error
and abort the job.
-- Add the name of the node a job is running on to the message printed by
slurmstepd when terminating a job.
-- Remove unsupported options from sacctmgr help and the dump function.
-- Update sacctmgr man page removing reference to obsolete parameter
MaxProcSecondsPerJob.
-- Added more validity checking of incoming job submit requests.

* Changes in Slurm 14.03.7
=====
-- Correct typos in man pages.
-- Add note to MaxNodesPerUser and multiple jobs running on the same node
counting as multiple nodes.
-- PerlAPI - fix renamed call from slurm_api_set_conf_file to
slurm_conf_reinit.
-- Fix gres race condition that could result in job deallocation error message.
-- Correct NumCPUs count for jobs with --exclusive option.
-- When creating reservation with CoreCnt, check that Slurm uses
SelectType=select/cons_res, otherwise don't send the request to slurmctld
and return an error.
-- Save the state of scheduled node reboots so they will not be lost should the
slurmctld restart.
-- In select/cons_res plugin - Insure the node count does not exceed the task
count.
-- switch/nrt - Do not explicitly unload windows for a job on termination,
only unload its table (which automatically unloads its windows).
-- When HealthCheckNodeState is configured as IDLE don't run the
HealthCheckProgram for nodes in any other states than IDLE.
-- Remove all slurmctld locks prior to job_submit() being called in plugins.

```

Appendix G. SLURM Release Information

If any slurmctld data structures are read or modified, add locks directly in the plugin.

- Minor sanity check to verify the string sent in isn't NULL when using bit_unfmt.
- CRAY NATIVE - Fix issue on heavy systems to only run the NHC once per job/step completion.
- Remove unneeded step cleanup for pending steps.
- Fix issue where if a batch job was manually requeued the batch step information wasn't stored in accounting.
- When job is release from a requeue hold state clean up its previous exit code.
- Correct the srun man page about how the output from the user application is sent to srun.
- Increase the timeout of the main thread while waiting for the i/o thread. Allow up to 180 seconds for the i/o thread to complete.
- When using sacct -c to read the job completion data compute the correct job elapsed time.
- Perl package: Define some missing node states.
- When using AccountingStorageType=accounting_storage/mysql zero out the database index for the array elements avoiding duplicate database values.
- Reword the explanation of cputime and cputimeraw in the sacct man page.
- JobCompType allows "jobcomp/mysql" as valid name but the code used "job_comp/mysql" setting an incorrect default database.
- Try to load libslurm.so only when necessary.
- When nodes scheduled for reboot, set state to DOWN rather than FUTURE so they are still visible to sinfo. State set to IDLE after reboot completes.
- Apply BatchStartTimeout configuration to task launch and avoid aborting srun commands due to long running Prolog scripts.
- Fix minor memory leaks when freeing node_info_t structure.
- Fix various memory leaks in sview
- If a batch script is requeued and running steps get correct exit code/signal previous it was always -2.
- If step exitcode hasn't been set display with sacct the -2 instead of acting like it is a signal and exitcode.
- Send calculated step_rc for batch step instead of raw status as done for normal steps.
- If a job times out, set the exit code in accounting to 1 instead of the signal 1.
- Update the acct_gather.conf.5 man page removing the reference to InfinibandOFEDFrequency.
- Fix gang scheduling for jobs submitted to multiple partitions.
- Enable srun to submit job to multiple partitions.
- Update slurm.conf man page. When Epilog or Prolog fail the node state is set ro DRAIN.
- Start a job in the highest priority partition possible, even if it requires preempting other jobs and delaying initiation, rather than using a lower priority partition. Previous logic would preempt lower priority jobs, but then might start the job in a lower priority partition and not use the resources released by the preempted jobs.
- Fix SelectTypeParameters=CR_PACK_NODES for srun making both job and step resource allocation.
- BGQ - Make it possible to pack multiple tasks on a core when not using the entire cnode.
- MYSQL - if unable to connect to mysqld close connection that was initied.
- DBD - when connecting make sure we wait MessageTimeout + 5 since the

timeout when talking to the Database is the same timeout so a race condition could occur in the requesting client when receiving the response if the database is unresponsive.

* Changes in Slurm 14.03.6

=====

- Added examples to demonstrate the use of the sacct -T option to the man page.
- Fix for regression in 14.03.5 with sacctmgr load when Parent has "" around it.
- Update comments in sacctmgr dump header.
- Fix for possible abort on change in GRES configuration.
- CRAY - fix modules file, (backport from 14.11 commit 78fe86192b).
- Fix race condition which could result in requeue if batch job exit and node registration occur at the same time.
- switch/nrt - Unload job tables (in addition to windows) in user space mode.
- Differentiate between two identical debug messages about purging vestigial job scripts.
- If the socket used by slurmd to communicate with slurmd exist when slurmd attempts to create it, for example left over from a previous requeue or crash, delete it and recreate it.

* Changes in Slurm 14.03.5

=====

- If a srun runs in an exclusive allocation and doesn't use the entire allocation and CR_PACK_NODES is set layout tasks appropriately.
- Correct Shared field in job state information seen by scontrol, svview, etc.
- Print Slurm error string in scontrol update job and reset the Slurm errno before each call to the API.
- Fix task/cgroup to handle -mblock:fcyclic correctly
- Fix for core-based advanced reservations where the distribution of cores across nodes is not even.
- Fix issue where association maxnodes wouldn't be evaluated correctly if a QOS had a GrpNodes set.
- GRES fix with multiple files defined per line in gres.conf.
- When a job is requeued make sure accounting marks it as such.
- Print the state of requeued job as REQUEUED.
- Fix if a job's partition was taken away from it don't allow a requeue.
- Make sure we lock on the conf when sending slurmd's conf to the slurmd.
- Fix issue with sacctmgr 'load' not able to gracefully handle bad formatted file.
- sched/backfill: Correct job start time estimate with advanced reservations.
- Error message added when in proctrack/cgroup the step freezer path isn't able to be destroyed for debug.
- Added extra index's into the database for better performance when deleting users.
- Fix issue with wckey when tracking wckey, but not enforcing them, you could get multiple '*' wckey.
- Fix bug which could report to squeue the wrong partition for a running job that is submitted to multiple partitions.
- Report correct CPU count allocated to job when allocated whole node even if not using all CPUs.
- If job's constraints cannot be satisfied put it in pending state with reason BadConstraints and don't remove it.
- sched/backfill - If job started with infinite time limit, set its end_time

Appendix G. SLURM Release Information

```
one year in the future.
-- Clear record of a job's gres when requeued.
-- Clear QOS GrpUsedCPUs when resetting raw usage if QOS is not using any cpus.
-- Remove log message left over from debugging.
-- When using CR_PACK_NODES fix make --ntasks-per-node work correctly.
-- Report correct partition associated with a step if the job is submitted to
multiple partitions.
-- Fix to allow removing of preemption from a QOS
-- If the proctrack plugins fail to destroy the job container print an error
message and avoid to loop forever, give up after 120 seconds.
-- Make srun obey POSIX convention and increase the exit code by 128 when the
process terminated by a signal.
-- Sanity check for acct_gather_energy/rapl
-- If the proctrack plugins fail to destroy the job container print an error
message and avoid to loop forever, give up after 120 seconds.
-- If the sbatch command specifies the option --signal=B:signum sent the signal
to the batch script only.
-- If we cancel a task and we have no other exit code send the signal and
exit code.
-- Added note about InnoDB storage engine being used with MySQL.
-- Set the job exit code when the job is signaled and set the log level to
debug2() when processing an already completed job.
-- Reset diagnostics time stamp when "sdiag --reset" is called.
-- squeue and scontrol to report a job's "shared" value based upon partition
options rather than reporting "unknown" if job submission does not use
--exclusive or --shared option.
-- task/cgroup - Fix cpuset binding for batch script.
-- sched/backfill - Fix anomaly that could result in jobs being scheduled out
of order.
-- Expand pseudo-terminal size data structure field sizes from 8 to 16 bits.
-- Set the job exit code when the job is signaled and set the log level to
debug2() when processing an already completed job.
-- Distinguish between two identical error messages.
-- If using accounting_storage/mysql directly without a DBD fix issue with
start of requeued jobs.
-- If a job fails because of batch node failure and the job is requeued and an
epilog complete message comes from that node do not process the batch step
information since the job has already been requeued because the epilog
script running isn't guaranteed in this situation.
-- Change message to note a NO_VAL for return code could of come from node
failure as well as interactive user.
-- Modify test4.5 to only look at one partition instead of all of them.
-- Fix sh5util -u to accept username different from the user that runs the
command.
-- Corrections to man pages:salloc.1 sbatch.1 srun.1 nonstop.conf.5
slurm.conf.5.
-- Restore srun --pty resize ability.
-- Have sacctmgr dump cluster handle situations where users or such have
special characters in their names like ':'
-- Add more debugging for information should the job ran on wrong node
and should there be problems accessing the state files.

* Changes in Slurm 14.03.4
=====
-- Fix issue where not enforcing QOS but a partition either allows or denies
```

```

them.
-- CRAY - Make switch/cray default when running on a Cray natively.
-- CRAY - Make job_container/cncu default when running on a Cray natively.
-- Disable job time limit change if it's preemption is in progress.
-- Correct logic to properly enforce job preemption GraceTime.
-- Fix sinfo -R to print each down/draind node once, rather than once per
partition.
-- If a job has non-responding node, retry job step create rather than
returning with DOWN node error.
-- Support SLURM_CONF path which does not have "slurm.conf" as the file name.
-- CRAY - make job_container/cncu default when running on a Cray natively
-- Fix issue where batch cpuset wasn't looked at correctly in
jobacct_gather/cgroup.
-- Correct squeue's job node and CPU counts for requeued jobs.
-- Correct SelectTypeParameters=CR_LLN with job selection of specific nodes.
-- Only if ALL of their partitions are hidden will a job be hidden by default.
-- Run EpilogSlurmctld for a job is killed during slurmctld reconfiguration.
-- Close window with srun if waiting for an allocation and while printing
something you also get a signal which would produce deadlock.
-- Add SelectTypeParameters option of CR_PACK_NODES to pack a job's tasks
tightly on its allocated nodes rather than distributing them evenly across
the allocated nodes.
-- cpus-per-task support: Try to pack all CPUs of each tasks onto one socket.
Previous logic could spread the tasks CPUs across multiple sockets.
-- Add new distribution method fcyclic so when a task is using multiple cpus
it can bind cyclically across sockets.
-- task/affinity - When using --hint=nomultithread only bind to the first
thread in a core.
-- Make cgroup task layout (block | cyclic) method mirror that of
task/affinity.
-- If TaskProlog sets SLURM_PROLOG_CPU_MASK reset affinity for that task
based on the mask given.
-- Keep supporting 'srun -N x --pty bash' for historical reasons.
-- If EnforcePartLimits=Yes and QOS job is using can override limits, allow
it.
-- Fix issues if partition allows or denies account's or QOS' and either are
not set.
-- If a job requests a partition and it doesn't allow a QOS or account the
job is requesting pend unless EnforcePartLimits=Yes. Before it would
always kill the job at submit.
-- Fix format output of scontrol command when printing node state.
-- Improve the clean up of cgroup hierarchy when using the
jobacct_gather/cgroup plugin.
-- Added SchedulerParameters value of Ignore_NUMA.
-- Fix issues with code when using automake 1.14.1
-- select/cons_res plugin: Fix memory leak related to job preemption.
-- After reconfig rebuild the job node counters only for jobs that have
not finished yet, otherwise if requeued the job may enter an invalid
COMPLETING state.
-- Do not purge the script and environment files for completed jobs on
slurmctld reconfiguration or restart (they might be later requeued).
-- scontrol now accepts the option job=xxx or jobid=xxx for the requeue,
requeuehold and release operations.
-- task/cgroup - fix to bind batch job in the proper CPUs.
-- Added strigger option of -N, --noheader to not print the header when

```

Appendix G. SLURM Release Information

```
displaying a list of triggers.
-- Modify strigger to accept arguments to the program to execute when an
event trigger occurs.
-- Attempt to create duplicate event trigger now generates ESLURM_TRIGGER_DUP
("Duplicate event trigger").
-- Treat special characters like %A, %s etc. literally in the file names
when specified escaped e.g. sbatch -o /home/zebra\\%s will not expand
%s as the stepid of the running job.
-- CRAYALPS - Add better support for CLE 5.2 when running Slurm over ALPS.
-- Test time when job_state file was written to detect multiple primary
slurmctld daemons (e.g. both backup and primary are functioning as
primary and there is a split brain problem).
-- Fix scontrol to accept update jobid=# numtasks=#
-- If the backup slurmctld assumes primary status, then do NOT purge any
job state files (batch script and environment files) and do not re-use them.
This may indicate that multiple primary slurmctld daemons are active (e.g.
both backup and primary are functioning as primary and there is a split
brain problem).
-- Set correct error code when requeuing a completing/pending job
-- When checking for if dependency of type afterany, afterok and afternotok
don't clear the dependency if the job is completing.
-- Cleanup the JOB_COMPLETING flag and eventually requeue the job when the
last epilog completes, either slurmd epilog or slurmctld epilog, whichever
comes last.
-- When attempting to requeue a job distinguish the case in which the job is
JOB_COMPLETING or already pending.
-- When reconfiguring the controller don't restart the slurmctld epilog if it
is already running.
-- Email messages for job array events print now use the job ID using the
format "#_# (#)" rather than just the internal job ID.
-- Set the number of free licenses to be 0 if the global license count
decreases and total is less than in use.
-- Add DebugFlag of BackfillMap. Previously a DebugFlag value of Backfill
logged information about what it was doing plus a map of expected resource
use in the future. Now that very verbose resource use map is only logged
with a DebugFlag value of BackfillMap
-- Fix slurmstepd core dump.
-- Modify the description of -E and -S option of sacct command as point in time
'before' or 'after' the database records are returned.
-- Correct support for partition with Shared=YES configuration.
-- If job requests --exclusive then do not use nodes which have any cores in an
advanced reservation. Also prevents case where nodes can be shared by other
jobs.
-- For "scontrol --details show job" report the correct CPU_IDs when there are
multiple threads per core (we are translating a core bitmap to CPU IDs).
-- If DebugFlags=Protocol is configured in slurm.conf print details of the
connection, ip address and port accepted by the controller.
-- Fix minor memory leak when reading in incomplete node data checkpoint file.
-- Enlarge the width specifier when printing partition SHARE to display larger
sharing values.
-- sinfo locks added to prevent possibly duplicate record printing for
resources in multiple partitions.
```

* Changes in Slurm 14.03.3-2

=====


```

-- BGQ - Fix issue with uninitialized variable.

* Changes in Slurm 14.03.3
=====
-- Correction to default batch output file name. In version 14.03.2 was using
  "slurm_<jobid>_4294967294.out" due to error in job array logic.
-- In slurm.spec file, replace "Requires cray-MySQL-devel-enterprise" with
  "Requires mysql-devel".

* Changes in Slurm 14.03.2
=====
-- Fix race condition if PrologFlags=Alloc,NoHold is used.
-- Cray - Make NPC only limit running other NPC jobs on shared blades instead
  of limited non NPC jobs.
-- Fix for sbatch #PBS -m (mail) option parsing.
-- Fix job dependency bug. Jobs dependent upon multiple other jobs may start
  prematurely.
-- Set "Reason" field for all elements of a job array on short-circuited
  scheduling for job arrays.
-- Allow -D option of salloc/srun/sbatch to specify relative path.
-- Added SchedulerParameter of batch_sched_delay to permit many batch jobs
  to be submitted between each scheduling attempt to reduce overhead of
  scheduling logic.
-- Added job reason of "SchedTimeout" if the scheduler was not able to reach
  the job to attempt scheduling it.
-- Add job's exit state and exit code to email message.
-- scontrol hold/release accepts job name option (in addition to job ID).
-- Handle when trying to cancel a step that hasn't started yet better.
-- Handle Max/GrpCPU limits better
-- Add --priority option to salloc, sbatch and srun commands.
-- Honor partition priorities over job priorities.
-- Fix sacct -c when using jobcomp/filetxt to read newer variables
-- Fix segfault of sacct -c if spaces are in the variables.
-- Release held job only with "scontrol release <jobid>" and not by resetting
  the job's priority. This is needed to support job arrays better.
-- Correct squeue command not to merge jobs with state pending and completing
  together.
-- Fix issue where user is requesting --acctg-freq=0 and no memory limits.
-- Fix issue with GrpCPURunMins if a job's timelimit is altered while the job
  is running.
-- Temporary fix for handling our typemap for the perl api with newer perl.
-- Fix allowgroup on bad group seg fault with the controller.
-- Handle node ranges better when dealing with accounting max node limits.

* Changes in Slurm 14.03.1-2
=====
-- Update configure to set correct version without having to run autogen.sh

* Changes in Slurm 14.03.1
=====
-- Add support for job std_in, std_out and std_err fields in Perl API.
-- Add "Scheduling Configuration Guide" web page.
-- BGQ - fix check for jobinfo when it is NULL
-- Do not check cleaning on "pending" steps.
-- task/cgroup plugin - Fix for building on older hwloc (v1.0.2).

```

Appendix G. SLURM Release Information

- In the PMI implementation by default don't check for duplicate keys. Set the SLURM_PMI_KVS_DUP_KEYS if you want the code to check for duplicate keys.
- Add job submission time to queue.
- Permit user root to propagate resource limits higher than the hard limit slurmd has on that compute node has (i.e. raise both current and maximum limits).
- Fix issue with license used count when doing an scontrol reconfig.
- Fix the PMI iterator to not report duplicated keys.
- Fix issue with sinfo when -o is used without the %P option.
- Rather than immediately invoking an execution of the scheduling logic on every event type that can enable the execution of a new job, queue its execution. This permits faster execution of some operations, such as modifying large counts of jobs, by executing the scheduling logic less frequently, but still in a timely fashion.
- If the environment variable is greater than MAX_ENV_STRLEN don't set it in the job env otherwise the exec() fails.
- Optimize scontrol hold/release logic for job arrays.
- Modify srun to report an exit code of zero rather than nine if some tasks exit with a return code of zero and others are killed with SIGKILL. Only an exit code of zero did this.
- Fix a typo in scontrol man page.
- Avoid slurmd crash getting job info if detail_ptr is NULL.
- Fix sacctmgr add user where both defaultaccount and accounts are specified.
- Added SchedulerParameters option of max_sched_time to limit how long the main scheduling loop can execute for.
- Added SchedulerParameters option of sched_interval to control how frequently the main scheduling loop will execute.
- Move start time of main scheduling loop timeout after locks are aquired.
- Add queue job format option of "%y" to print a job's nice value.
- Update scontrol update jobID logic to operate on entire job arrays.
- Fix PrologFlags=Alloc to run the prolog on each of the nodes in the allocation instead of just the first.
- Fix race condition if a step is starting while the slurmd is being restarted.
- Make sure a job's prolog has ran before starting a step.
- BGQ - Fix invalid memory read when using DefaultConnType in the bluegene.conf
- Make sure we send node state to the DBD on clean start of controller.
- Fix some sinfo and queue sorting anomalies due to differences in data types.
- Only send message back to slurmd when PrologFlags=Alloc is used on a Cray/ALPS system, otherwise use the slurmd to wait on the prolog to gate the start of the step.
- Remove need to check PrologFlags=Alloc in slurmd since we can tell if prolog has ran yet or not.
- Fix queue to use a correct macro to check job state.
- BGQ - Fix incorrect logic issues if MaxBlockInError=0 in the bluegene.conf.
- priority/basic - Insure job priorities continue to decrease when jobs are submitted with the --nice option.
- Make the PrologFlag=Alloc work on batch scripts
- Make PrologFlag=NoHold (automatically sets PrologFlag=Alloc) not hold in salloc/srun, instead wait in the slurmd when a step hits a node and the prolog is still running.
- Added --cpu-freq=highml (high minus one) option.

```
-- Expand StdIn/Out/Err string length output by "scontrol show job" from 128
to 1024 bytes.
-- squeue %F format will now print the job ID for non-array jobs.
-- Use quicksort for all priority based job sorting, which improves performance
significantly with large job counts.
-- If a job has already been released from a held state ignore successive
release requests.
-- Fix srun/salloc/sbatch man pages for the --no-kill option.
-- Add squeue -L/--licenses option to filter jobs by license names.
-- Handle abort job on node on front end systems without core dumping.
-- Fix dependency support for job arrays.
-- When updating jobs verify the update request is not identical to
the current settings.
-- When sorting jobs and priorities are equal sort by job_id.
-- Do not overwrite existing reason for node being down or drained.
-- Requeue batch job if Munge is down and credential can not be created.
-- Make _slurm_init_msg_engine() tolerate bug in bind() returning a busy
ephemeral port.
-- Don't block scheduling of entire job array if it could run in multiple
partitions.
-- Introduce a new debug flag Protocol to print protocol requests received
together with the remote IP address and port.
-- CRAY - Set up the network even when only using 1 node.
-- CRAY - Greatly reduce the number of error messages produced from the task
plugin and provide more information in the message.
```

* Changes in Slurm 14.03.0

=====

```
-- job_submit/lua: Fix invalid memory reference if script returns error message
for user.
-- Add logic to sleep and retry if slurm.conf can't be read.
-- Reset a node's CpuLoad value at least once each SlurmdTimeout seconds.
-- Scheduler enhancements for reservations: When a job needs to run in
reservation, but can not due to busy resources, then do not block all jobs
in that partition from being scheduled, but only the jobs in that
reservation.
-- Export "SLURM*" environment variables from sbatch even if --export=NONE.
-- When recovering node state if the Slurm version is 2.6 or 2.5 set the
protocol version to be SLURM_2_5_PROTOCOL_VERSION which is the minimum
supported version.
-- Update the scancel man page documenting the -s option.
-- Update sacctmgr man page documenting how to modify account's QOS.
-- Fix for sjstat which currently does not print >1TB memory values correctly.
-- Change xmalloc()/xfree() to malloc()/free() in hostlist.c for better
performance.
-- Update squeue.1 man page describing the SPECIAL_EXIT state.
-- Added scontrol option of errnumstr to return error message given a slurm
error number.
-- If srun invoked with the --multi-prog option, but no task count, then use
the task count provided in the MPMD configuration file.
-- Prevent svview abort on some systems when adding or removing columns to the
display for nodes, jobs, partitions, etc.
-- Add job array hash table for improved performance.
-- Make AccountingStorageEnforce=all not include nojobs or nosteps.
-- Added sacctmgr mod qos set RawUsage=0.
```

Appendix G. SLURM Release Information

```
-- Modify hostlist functions to accept more than two numeric ranges (e.g.
   "row[1-3]rack[0-8]slot[0-63]")

* Changes in Slurm 14.03.0rc1
=====
-- Fixed typos in srun_cr man page.
-- Run job scheduling logic immediately when nodes enter service.
-- Added sbatch '--parsable' option to output only the job id number and the
   cluster name separated by a semicolon. Errors will still be displayed.
-- Added failure management "slurmctld/nonstop" plugin.
-- Prevent jobs being killed when a checkpoint plugin is enabled or disabled.
-- Update the documentation about SLURM_PMI_KVS_NO_DUP_KEYS environment
   variable.
-- select/cons_res bug fix for range of node counts with --cpus-per-task
   option (e.g. "srun -N2-3 -c2 hostname" would allocate 2 CPUs on the first
   node and 0 CPUs on the second node).
-- Change reservation flags field from 16 to 32-bits.
-- Add reservation flag value of "FIRST_CORES".
-- Added the idea of Resources to the database. Framework for handling
   license servers outside of Slurm.
-- When starting the slurmctld only send past job/node state information to
   accounting if running for the first time (should speed up startup
   dramatically on systems with lots of nodes or lots of jobs).
-- Compile and run on FreeBSD 8.4.
-- Make job array expressions more flexible to accept multiple step counts in
   the expression (e.g. "--array=1-10:2,50-60:5,123").
-- switch/cray - add state save/restore logic tracking allocated ports.
-- SchedulerParameters - Replace max_job_bf with bf_max_job_start (both will
   work for now).
-- Add SchedulerParameters options of preempt_reorder_count and
   preempt_strict_order.
-- Make memory types in acct_gather uint64_t to handle systems with more than
   4TB of memory on them.
-- BGQ - --export=NONE option for srun to make it so only the SLURM_JOB_ID
   and SLURM_STEP_ID env vars are set.
-- Munge plugins - Add sleep between retries if can't connect to socket.
-- Added DebugFlags value of "License".
-- Added --enable-developer which will give you -Werror when compiling.
-- Fix for job request with GRES count of zero.
-- Fix a potential memory leak in hostlist.
-- Job array dependency logic: Cache results for major performance improvement.
-- Modify squeue to support filter on job states Special_Exit and Resizing.
-- Defer purging job record until after EpilogSlurmctld completes.
-- Add -j option for jobid to sbcast.
-- Fix handling RPCs from a 14.03 slurmctld to a 2.6 slurmd

* Changes in Slurm 14.03.0pre6
=====
-- Modify slurmstepd to log messages according to the LogTimeFormat
   parameter in slurm.conf.
-- Insure that overlapping reservations do not oversubscribe available
   licenses.
-- Added core specialization logic to select/cons_res plugin.
-- Added whole_node field to job_resources structure and enable gang scheduling
   for jobs with core specialization.
```

```

-- When using FastSchedule = 1 the nodes with less than configured resources
   are not longer set DOWN, they are set to DRAIN instead.
-- Modified 'sacctmgr show associations' command to show GrpCPURunMins
   by default.
-- Replace the hostlist_push() function with a more efficient
   hostlist_push_host().
-- Modify the reading of lustre file system statistics to print more
   information when debug and when io error occur.
-- Add specialized core count field to job credential data.
   NOTE: This changes the communications protocol from other pre-releases of
   version 14.03. All programs must be cancelled and daemons upgraded from
   previous pre-releases of version 14.03. Upgrades from version 2.6 or earlier
   can take place without loss of jobs
-- Add version number to node and front-end configuration information visible
   using the scontrol tool.
-- Add idea of a RESERVED flag for node state so idle resources are marked
   not "idle" when in a reservation.
-- Added core specialization plugin infrastructure.
-- Added new job_submit/trottle plugin to control the rate at which a user
   can submit jobs.
-- CRAY - added network performance counters option.
-- Allow scontrol suspend/resume to accept jobid in the format jobid_taskid
   to suspend/resume array elements.
-- In the slurmd job record, split "shared" variable into "share_res" (share
   resource) and "whole_node" fields.
-- Fix the format of SLURM_STEP_RESV_PORTS. It was generated incorrectly
   when using the hostlist_push_host function and input surrounded by [].
-- Modify the srun --slurmd-debug option to accept debug string tags
   (quiet, fatal, error, info verbose) beside the numerical values.
-- Fix the bug where --cpu_bind=map_cpu is interpreted as mask_cpu.
-- Update the documentation regarding the state of cpu frequencies after
   a step using --cpu-freq completes.
-- CRAY - Fix issue when a job is requeued and nhc is still running as it is
   being scheduled to run again. This would erase the previous job info
   that was still needed to clean up the nodes from the previous job run.
   (Bug 526).
-- Set SLURM_JOB_PARTITION environment variable set for all job allocations.
-- Set SLURM_JOB_PARTITION environment variable for Prolog program.
-- Added SchedulerParameters option of partition_job_depth to limit scheduling
   logic depth by partition.
-- Handle the case in which errno is not reset to 0 after calling
   getgrent_r(), which causes the controller to core dump.

* Changes in Slurm 14.03.0pre5
=====
-- Added squeue format option of "%X" (core specialization count).
-- Added core specialization web page (just a start for now).
-- Added the SLURM_ARRAY_JOB_ID and SLURM_ARRAY_TASK_ID
   in epilog slurmd environment.
-- Fix bug in job step allocation failing due to memory limit.
-- Modify the pbsnodes script to reflect its output on a TORQUE system.
-- Add ability to clear a node's DRAIN flag using scontrol or sview by setting
   it's state to "UNDRAIN". The node's base state (e.g. "DOWN" or "IDLE") will
   not be changed.
-- Modify the output of 'scontrol show partition' by displaying

```

Appendix G. SLURM Release Information

```
DefMemPerCPU=UNLIMITED and MaxMemPerCPU=UNLIMITED when these limits are
configured as 0.
-- mpirun-mic - Major re-write of the command wrapper for Xeon Phi use.
-- Add new configuration parameter of AuthInfo to specify port used by
authentication plugin.
-- Fixed conditional RPM compiling.
-- Corrected slurmstepd ident name when logging to syslog.
-- Fixed sh5util loop when there are no node-step files.
-- Add SLURM_CLUSTER_NAME to environment variables passed to PrologSlurmctld,
Prolog, EpilogSlurmctld, and Epilog
-- Add the idea of running a prolog right when an allocation happens
instead of when running on the node for the first time.
-- If user runs 'scontrol reconfig' but hostnames or the host count changes
the slurmctld throws a fatal error.
-- gres.conf - Add "NodeName" specification so that a single gres.conf file
can be used for a heterogeneous cluster.
-- Add flag to accounting RPC to indicate if job data is packed or not.
-- After all srun tasks have terminated on a node close the stdout/stderr
channel with the slurmstepd on that node.
-- In case of i/o error with slurmstepd log an error message and abort the
job.
-- Add --test-only option to sbatch command to validate the script and options.
The response includes expected start time and resources to be allocated.
```

* Changes in Slurm 14.03.0pre4

```
=====
-- Remove the ThreadID documentation from slurm.conf. This functionality has
been obsoleted by the LogTimeFormat.
-- Sched plugins - rename global and plugin functions names for consistency
with other plugin types.
-- BGQ - Added RebootQOSList option to bluegene.conf to allow an implicate
reboot of a block if only jobs in the list are running on it when cnodes
go into a failure state.
-- Correct task count of pending job steps.
-- Improve limit enforcement for jobs, set RLIMIT_RSS, RLIMIT_AS and/or
RLIMIT_DATA to enforce memory limit.
-- Pending job steps will have step_id of INFINITE rather than NO_VAL and
will be reported as "TBD" by scontrol and squeue commands.
-- Add logic so PMI_Abort or PMI2_Abort can propagate an exit code.
-- Added SlurmdPlugstack configuration parameter.
-- Added PriorityFlag DEPTH_OBLIVIOUS to have the depth of an association
not effect it's priority.
-- Multi-thread the sinfo command (one thread per partition).
-- Added sgather tool to gather files from a job's compute nodes into a
central location.
-- Added configuration parameter FairShareDampeningFactor to offer a greater
priority range based upon utilization.
-- Change MaxArraySize and job's array_task_id from 16-bit to 32-bit field.
Additional Slurm enhancements are be required to support larger job arrays.
-- Added -S/--core-spec option to salloc, sbatch and srun commands to reserve
specialized cores for system use. Modify scontrol and svview to get/set
the new field. No enforcement exists yet for these new options.
struct job_info / slurm_job_info_t: Added core_spec
struct job_descriptor/job_desc_msg_t: Added core_spec
```

* Changes in Slurm 14.03.0pre3

=====

- Do not set SLURM_NODEID environment variable on front-end systems.
- Convert bitmap functions to use int32_t instead of int in data structures and function arguments. This is to reliably enable use of bitmaps containing up to 4 billion elements. Several data structures containing index values were also changed from data type int to int32_t:
 - Struct job_info / slurm_job_info_t: Changed exc_node_inx, node_inx, and req_node_inx from type int to type int32_t
 - job_step_info_t: Changed node_inx from type int to type int32_t
 - Struct partition_info / partition_info_t: Changed node_inx from type int to type int32_t
 - block_job_info_t: Changed cnode_inx from type int to type int32_t
 - block_info_t: Changed ionode_inx and mp_inx from type int to type int32_t
 - Struct reserve_info / reserve_info_t: Changed node_inx from type int to type int32_t
- Modify qsub wrapper output to match torque command output, just print the job ID rather than "Submitted batch job #"
- Change Slurm error string for ESLURM_MISSING_TIME_LIMIT from "Missing time limit" to "Time limit specification required, but not provided"
- Change salloc job_allocate error message header from "Failed to allocate resources" to "Job submit/allocate failed"
- Modify slurmctld message retry logic to support Cray cold-standby SDB.

* Changes in Slurm 14.03.0pre2

=====

- Added "JobAcctGatherParams" configuration parameter. Value of "NoShare" disables accounting for shared memory.
- Added fields to "scontrol show job" output: boards_per_node, sockets_per_board, ntasks_per_node, ntasks_per_board, ntasks_per_socket, ntasks_per_core, and nice.
- Add queue output format options for job command and working directory (%o and %Z respectively).
- Add stdin/out/err to sview job output.
- Add new job_state of JOB_BOOT_FAIL for job terminations due to failure to boot it's allocated nodes or BlueGene block.
- CRAY - Add SelectTypeParameters NHC_NO_STEPS and NHC_NO which will disable the node health check script for steps and allocations respectfully.
- Reservation with CoreCnt: Avoid possible invalid memory reference.
- Add new error code for attempt to create a reservation with duplicate name.
- Validate that a hostlist file contains text (i.e. not a binary).
- switch/generic - propagate switch information from srun down to slurmd and slurmstepd.
- CRAY - Do not package Slurm's libpmi or libpmi2 libraries. The Cray version of those libraries must be used.
- Added a new option to the scontrol command to view licenses that are configured in use and available. 'scontrol show licenses'.
- MySQL - Made Slurm compatible with 5.6

* Changes in Slurm 14.03.0pre1

=====

- sview - improve scalability
- Add task pointer to the task_post_term() function in task plugins. The

Appendix G. SLURM Release Information

```
terminating task's PID is available in task->pid.
-- Move select/cray to select/alps
-- Defer sending SIGKILL signal to processes while core dump in progress.
-- Added JobContainerPlugin configuration parameter and plugin infrastructure.
-- Added partition configuration parameters AllowAccounts, AllowQOS,
  DenyAccounts and DenyQOS.
-- The rpmbuild option for a cray system with ALPS has changed from
  %_with_cray to %_with_cray_alps.
-- The log file timestamp format can now be selected at runtime via the
  LogTimeFormat configuration option. See the slurm.conf and slurmdbd.conf
  man pages for details.
-- Added switch/generic plugin to a job's convey network topology.
-- BLUEGENE - If block is in 'D' state or has more cnodes in error than
  MaxBlockInError set the job wait reason appropriately.
-- API use: Generate an error return rather than fatal error and exit if the
  configuraiton file is absent or invalid. This will permit Slurm APIs to be
  more reliably used by other programs.
-- Add support for load-based scheduling, allocate jobs to nodes with the
  largest number of available CPUs. Added SchedulingParameters paramter of
  "CR_LLN" and partition parameter of "LLN=yes|no".
-- Added job_info() and step_info() functions to the gres plugins to extract
  plugin specific fields from the job's or step's GRES data structure.
-- Added sbatch --signal option of "B:" to signal the batch shell rather than
  only the spawned job steps.
-- Added sinfo and squeue format option of "%all" to print all fields available
  for the data type with a vertical bar separating each field.
-- Add mechanism for job_submit plugin to generate error message for srun,
  salloc or sbatch to stderr. New argument added to job_submit function in
  the plugin.
-- Add StdIn, StdOut, and StdErr paths to job information dumped with
  "scontrol show job".
-- Permit Slurm administrator to submit a batch job as any user.
-- Set a job's RLIMIT_AS limit based upon it's memory limit and VsizeFactor
  configuration value.
-- Remove Postgres plugins
-- Make jobacct_gather/cgroup work correctly and also make all jobacct_gather
  plugins more maintainable.
-- Proctrack/pgid - Add support for proctrack_p_plugin_get_pids() function.
-- Sched/backfill - Change default max_job_bf parameter from 50 to 100.
-- Added -I|--item-extract option to sh5util to extract data item from series.

* Changes in Slurm 2.6.10
=====
-- Switch/nrt - On switch resource allocation failure, free partial allocation.
-- Switch/nrt - Properly track usage of CAU and RDMA resources with multiple
  tasks per compute node.
-- Fix issue where user is requesting --acctg-freq=0 and no memory limits.
-- BGQ - Temp fix issue where job could be left on job_list after it finished.
-- BGQ - Fix issue where limits were checked on midplane counts instead of
  cnode counts.
-- BGQ - Move code to only start job on a block after limits are checked.
-- Handle node ranges better when dealing with accounting max node limits.
-- Fix perlapi to compile correctly with perl 5.18
-- BGQ - Fix issue with uninitialized variable.
-- Correct sinfo --sort fields to match documentation: E => Reason,
```


H -> Reason Time (new), R -> Partition Name, u/U -> Reason user (new)
 -- If an invalid assoc_ptr comes in don't use the id to verify it.
 -- Sched/backfill modified to avoid using nodes in completing state.
 -- Correct support for job --profile=none option and related documentation.
 -- Properly enforce job --requeue and --norequeue options.
 -- If a job --mem-per-cpu limit exceeds the partition or system limit, then
 scale the job's memory limit and CPUs per task to satisfy the limit.
 -- Correct logic to support Power7 processor with 1 or 2 threads per core
 (CPU IDs are not consecutive).

* Changes in Slurm 2.6.9

=====

-- Fix sinfo to work correctly with draining/mixed nodes as well as filtering
 on Mixed state.
 -- Fix sacctmgr update user with no "where" condition.
 -- Fix logic bugs for SchedulerParameters option of max_rpc_cnt.

* Changes in Slurm 2.6.8

=====

-- Add support for Torque/PBS job array options and environment variables.
 -- CRAY/ALPS - Add support for CLE52
 -- Fix issue where jobs still pending after a reservation would remain
 in waiting reason ReqNodeNotAvail.
 -- Update last_job_update when a job's state_reason was modified.
 -- Free job_ptr->state_desc where ever state_reason is set.
 -- Fixed sacct.1 and srun.1 manual pages which contains a hyphen where
 a minus sign for options was intended.
 -- sinfo - Make sure if partition name is long and the default the last char
 doesn't get chopped off.
 -- task/affinity - Protect against zero divide when simulating more hardware
 than you really have.
 -- NRT - Fix issue with 1 node jobs. It turns out the network does need to
 be setup for 1 node jobs.
 -- Fix recovery of job dependency on task of job array when slurmctld restarts.
 -- mysql - Fix invalid memory reference.
 -- Lock the /cgroup/freezer subsystem when creating files for tracking processes.
 -- Fix preempt/partition_prio to avoid preempting jobs in partitions with
 PreemptMode=OFF
 -- launch/poe - Implicitly set --network in job step create request as needed.
 -- Permit multiple batch job submissions to be made for each run of the
 scheduler logic if the job submissions occur at the nearly same time.
 -- Fix issue where associations weren't correct if backup takes control and
 new associations were added since it was started.
 -- Fix race condition is corner case with backup slurmctld.
 -- With the backup slurmctld make sure we reinit beginning values in the
 slurmdbd plugin.
 -- Fix sinfo to work correctly with draining/mixed nodes.
 -- MySQL - Fix it so a lock isn't held unnecessarily.
 -- Added new SchedulerParameters option of max_rpc_cnt when too many RPCs
 are active.
 -- BGQ - Fix deny_pass to work correctly.
 -- BGQ - Fix sub block steps using a block when the block has passthrough's
 in it.

* Changes in Slurm 2.6.7

Appendix G. SLURM Release Information

```
=====
-- Properly enforce a job's cpus-per-task option when a job's allocation is
   constrained on some nodes by the mem-per-cpu option.
-- Correct the slurm.conf man pages and checkpoint_blcr.html page
   describing that jobs must be drained from cluster before deploying
   any checkpoint plugin. Corrected in version 14.03.
-- Fix issue where if using munge and munge wasn't running and a slurmd
   needed to forward a message, the slurmd would core dump.
-- Update srun.1 man page documenting the PMI2 support.
-- Fix slurmctld core dump when a jobs gets its QOS updated but there
   is not a corresponding association.
-- If a job requires specific nodes and can not run due to those nodes being
   busy, the main scheduling loop will block those specific nodes rather than
   the entire queue/partition.
-- Fix minor memory leak when updating a job's name.
-- Fix minor memory leak when updating a reservation on a partition using "ALL"
   nodes.
-- Fix minor memory leak when adding a reservation with a nodelist and core
   count.
-- Update sacct man page description of job states.
-- BGQ - Fix minor memory leak when selecting blocks that can't immediately be
   placed.
-- Fixed minor memory leak in backfill scheduler.
-- MYSQL - Fixed memory leak when querying clusters.
-- MYSQL - Fix when updating QOS on an association.
-- NRT - Fix to supply correct error messages to poe/pmd when a launch fails.
-- Add SLURM_STEP_ID to Prolog environment.
-- Add support for SchedulerParameters value of bf_max_job_start that limits
   the total number of jobs that can be started in a single iteration of the
   backfill scheduler.
-- Don't print negative number when dealing with large memory sizes with
   sacct.
-- Fix sinfo output so that host in state allocated and mixed will not be
   merged together.
-- GRES: Avoid crash if GRES configurations is inconstent.
-- Make S_SLURM_RESTART_COUNT item available to SPANK.
-- Munge plugins - Add sleep between retries if can't connect to socket.
-- Fix the database query to return all pending jobs in a given time interval.
-- switch/nrt - Correct logic to get dynamic window count.
-- Remove need to use job->ctx_params in the launch plugin, just to simplify
   code.
-- NRT - Fix possible memory leak if using multiple adapters.
-- NRT - Fix issue where there are more than NRT_MAXADAPTERS on a system.
-- NRT - Increase Max number of adapters from 8 -> 9
-- NRT - Initialize missing variables when the PMD is starting a job.
-- NRT - Fix issue where we are launching hosts out of numerical order,
   this would cause pmd's to hang.
-- NRT - Change xmalloc's to malloc just to be safe.
-- NRT - Sanity check to make sure a jobinfo is there before packing.
-- Add missing options to the print of TaskPluginParam.
-- Fix a couple of issues with scontrol reconfig and adding nodes to
   slurm.conf. Rebooting daemons after adding nodes to the slurm.conf
   is highly recommended.
```

* Changes in Slurm 2.6.6

```

=====
-- sched/backfill - Fix bug that could result in failing to reserve resources
  for high priority jobs.
-- Correct job RunTime if requeued from suspended state.
-- Reset job priority from zero (held) on manual resume from suspend state.
-- If FastSchedule=0 then do not DOWN a node with low memory or disk size.
-- Remove vestigial note.
-- Update sshare.1 man page making it consistent with sacctmgr.1.
-- Do not reset a job's priority when the slurmctld restarts if previously
  set to some specific value.
-- svview - Fix regression where the Node tab wasn't able to add/remove columns.
-- Fix slurmstepd lock when job terminates inside the infiniband
  network traffic accounting plugin.
-- Correct the documentation to read filesystem instead of Lustre. Update
  the srun help.
-- Fix the acct_gather_filesystem_lustre.c to compute the Lustre accounting
  data correctly accumulating differences between sampling intervals.
  Fix the data structure mismatch between acct_gather_filesystem_lustre.c
  and slurm_jobacct_gather.h which caused the hdf5 plugin to log incorrect
  data.
-- Don't allow PMI_TIME to be zero which will cause floating exception.
-- Fix purging of old reservation errors in database.
-- MYSQL - If starting the plugin and the database isn't up attempt to
  connect in a loop instead of producing a fatal.
-- BLUEGENE - If IONodesPerMP changes in bluegene.conf recalculate bitmaps
  based on ionode count correctly on slurmctld restart.
-- Fix step allocation when some CPUs are not available due to memory limits.
  This happens when one step is active and using memory that blocks the
  scheduling of another step on a portion of the CPUs needed. The new step
  is now delayed rather than aborting with "Requested node configuration is
  not available".
-- Make sure node limits get assessed if no node count was given in request.
-- Removed obsolete slurm_terminate_job() API.
-- Update documentation about QOS limits
-- Retry task exit message from slurmstepd to srun on message timeout.
-- Correction to logic reserving all nodes in a specified partition.
-- Added support for selecting AMD GPU by setting GPU_DEVICE_ORDINAL env var.
-- Properly enforce GrpSubmit limit for job arrays.
-- CRAY - fix issue with using CR_ONE_TASK_PER_CORE
-- CRAY - fix memory leak when using accelerators

```

* Changes in Slurm 2.6.5

```

=====
-- Correction to hostlist parsing bug introduced in v2.6.4 for hostlists with
  more than one numeric range in brackets (e.g. rack[0-3]_blade[0-63]).
-- Add notification if using proctrack/cgroup and task/cgroup when oom hits.
-- Corrections to advanced reservation logic with overlapping jobs.
-- job_submit/lua - add cpus_per_task field to those available.
-- Add cpu_load to the node information available using the Perl API.
-- Correct a job's GRES allocation data in accounting records for non-Cray
  systems.
-- Substantial performance improvement for systems with Shared=YES or FORCE
  and large numbers of running jobs (replace bubble sort with quick sort).
-- proctrack/cgroup - Add locking to prevent race condition where one job step
  is ending for a user or job at the same time another job stepsis starting

```

Appendix G. SLURM Release Information

and the user or job container is deleted from under the starting job step.

- Fixed sh5util loop when there are no node-step files.
- Fix race condition on batch job termination that could result in a job exit code of 0xffffffffe if the slurmd on node zero registers its active jobs at the same time that slurmstepd is recording the job's exit code.
- Correct logic returning remaining job dependencies in job information reported by scontrol and squeue. Eliminates vestigial descriptors with no job ID values (e.g. "afterany").
- Improve performance of REQUEST_JOB_INFO_SINGLE RPC by removing unnecessary locks and use hash function to find the desired job.
- jobcomp/filetxt - Reopen the file when slurmd daemon is reconfigured or gets SIGHUP.
- Remove notice of CVE with very old/deprecated versions of Slurm in news.html.
- Fix if hwloc_get_nbobjs_by_type() returns zero core count (set to 1).
- Added ApbasilTimeout parameter to the cray.conf configuration file.
- Handle in the API if parts of the node structure are NULL.
- Fix srun hang when IO fails to start at launch.
- Fix for GRES bitmap not matching the GRES count resulting in abort (requires manual resetting of GRES count, changes to gres.conf file, and slurmd restarts).
- Modify svview to better support job arrays.
- Modify squeue to support longer job ID values (for many job array tasks).
- Fix race condition in authentication credential creation that could corrupt memory. (NOTE: This race condition has existed since 2003 and would be exceedingly rare.)
- HDF5 - Fix minor memory leak.
- Slurmstepd variable initialization - Without this patch, free() is called on a random memory location (i.e. whatever is on the stack), which can result in slurmstepd dying and a completed job not being purged in a timely fashion.
- Fix slurmstepd race condition when separate threads are reading and modifying the job's environment, which can result in the slurmstepd failing with an invalid memory reference.
- Fix erroneous error messages when running gang scheduling.
- Fix minor memory leak.
- scontrol modified to suspend, resume, hold, uhold, or release multiple jobs in a space separated list.
- Minor debug error when a connection goes away at the end of a job.
- Validate return code from calls to slurm_get_peer_addr
- BGQ - Fix issues with making sure all cnodes are accounted for when multiple steps cause multiple cnodes in one allocation to go into error at the same time.
- scontrol show job - Correct NumNodes value calculated based upon job specifications.
- BGQ - Fix issue if user runs multiple sub-block jobs inside a multiple midplane block that starts on a higher coordinate than it ends (i.e if a block has midplanes [0010,0013] 0013 is the start even though it is listed second in the hostlist).
- BGQ - Add midplane to the total_cnodes used in the runjob_mux plugin for better debug.
- Update AllocNodes paragraph in slurm.conf.5.

* Changes in Slurm 2.6.4

=====

```

-- Fixed sh5util to print its usage.
-- Corrected commit f9a3c7e4e8ec.
-- Honor ntasks-per-node option with exclusive node allocations.
-- sched/backfill - Prevent invalid memory reference if bf_continue option is
  configured and slurm is reconfigured during one of the sleep cycles or if
  there are any changes to the partition configuration or if the normal
  scheduler runs and starts a job that the backfill scheduler is actively
  working on.
-- Update man pages information about acct-freq and JobAcctGatherFrequency
  to reflect only the latest supported format.
-- Minor document update to include note about PrivateData=Usage for the
  slurm.conf when using the DBD.
-- Expand information reported with DebugFlags=backfill.
-- Initiate jobs pending to run in a reservation as soon as the reservation
  becomes active.
-- Purged expired reservation even if it has pending jobs.
-- Corrections to calculation of a pending job's expected start time.
-- Remove some vestigial logic treating job priority of 1 as a special case.
-- Memory freeing up to avoid minor memory leaks at close of daemons
-- Updated documentation to give correct units being displayed.
-- Report AccountingStorageBackupHost with "scontrol show config".
-- init scripts ignore quotes around Pid file name specifications.
-- Fixed typo about command case in quickstart.html.
-- task/cgroup - handle new cpuset files, similar to commit c4223940.
-- Replace the tempname() function call with mkstemp().
-- Fix for --cpu_bind=map_cpu/mask_cpu/map_ldom/mask_ldom plus
  --mem_bind=map_mem/mask_mem options, broken in 2.6.2.
-- Restore default behavior of allocating cores to jobs on a cyclic basis
  across the sockets unless SelectTypeParameters=CR_CORE_DEFAULT_DIST_BLOCK
  or user specifies other distribution options.
-- Enforce JobRequeue configuration parameter on node failure. Previously
  always requeued the job.
-- acct_gather_energy/ipmi - Add delay before retry on read error.
-- select/cons_res with GRES and multiple threads per core, fix possible
  infinite loop.
-- proctrack/cgroup - Add cgroup create retry logic in case one step is
  starting at the same time as another step is ending and the logic to create
  and delete cgroups overlaps.
-- Improve setting of job wait "Reason" field.
-- Correct sbatch documentation and job_submit/pbs plugin "%j" is job ID,
  not "%J" (which is job_id.step_id).
-- Improvements to sinfo performance, especially for large numbers of
  partitions.
-- SlurmdDebug - Permit changes to slurmd debug level with "scontrol reconfig"
-- smap - Avoid invalid memory reference with hidden nodes.
-- Fix sacctmgr modify qos set preempt+/-=.
-- BLUEGENE - fix issue where node count wasn't set up correctly when srun
  performs the allocation, regression in 2.6.3.
-- Add support for dependencies of job array elements (e.g.
  "sbatch --depend=afterok:123_4 ...") or all elements of a job array (e.g.
  "sbatch --depend=afterok:123 ...").
-- Add support for new options in sbatch qsub wrapper:
  -W block=true (wait for job completion)
  Clear PBS_NODEFILE environment variable
-- Fixed the MaxSubmitJobsPerUser limit in QOS which limited submissions

```

Appendix G. SLURM Release Information

```
a job too early.
-- sched/wiki, sched/wiki2 - Fix to work with change logic introduced in
   version 2.6.3 preventing Maui/Moab from starting jobs.
-- Updated the QOS limits documentation and man page.

* Changes in Slurm 2.6.3
=====
-- Add support for some new #PBS options in sbatch scripts and qsub wrapper:
   -l accelerator=true|false (GPU use)
   -l mpiprocs=# (processors per node)
   -l naccelerators=# (GPU count)
   -l select=# (node count)
   -l ncpus=# (task count)
   -v key=value (environment variable)
   -W depend=opts (job dependencies, including "on" and "before" options)
   -W umask=# (set job's umask)
-- Added qalter and qrerun commands to torque package.
-- Corrections to qstat logic: job CPU count and partition time format.
-- Add job_submit/pbs plugin to translate PBS job dependency options to the
   extend possible (no support for PBS "before" options) and set some PBS
   environment variables.
-- Add spank/pbs plugin to set a bunch of PBS environment variables.
-- Backported sh5util from master to 2.6 as there are some important
   bugfixes and the new item extraction feature.
-- select/cons_res - Correct MacCPUsPerNode partition constraint for CR_Socket.
-- scontrol - for setdebugflags command, avoid parsing "-flagname" as an
   scontrol command line option.
-- Fix issue with step accounting if a job is requeued.
-- Close file descriptors on exec of prolog, epilog, etc.
-- Fix issue when a user has held a job and then sets the begin time
   into the future.
-- Scontrol - Enable changing a job's stdout file.
-- Fix issues where memory or node count of a srun job is altered while the
   srun is pending. The step creation would use the old values and possibly
   hang srun since the step wouldn't be able to be created in the modified
   allocation.
-- Add support for new SchedulerParameters value of "bf_max_job_part", the
   maximum depth the backfill scheduler should go in any single partition.
-- acct_gather/infiniband plugin - Correct packets_in/out values.
-- BLUEGENE - Don't ignore a conn-type request from the user.
-- BGQ - Force a request on a Q for a MESH to be a TORUS in a dimension that
   can only be a TORUS (1).
-- Change max message length from 100MB to 1GB before generating "Insane
   message length" error.
-- sched/backfill - Prevent possible memory corruption due to use of
   bf_continue option and long running scheduling cycle (pending jobs could
   have been cancelled and purged).
-- CRAY - fix AcceleratorAllocation depth correctly for basil 1.3
-- Created the environment variable SLURM_JOB_NUM_NODES for srun jobs and
   updated the srun man page.
-- BLUEGENE/CRAY - Don't set env variables that pertain to a node when Slurm
   isn't doing the launching.
-- gres/gpu and gres/mic - Do not treat the existence of an empty gres.conf
   file as a fatal error.
-- Fixed for if hours are specified as 0 the time days-0:min specification
```

```

is not parsed correctly.
-- switch/nrt - Fix for memory leak.
-- Subtract the PMII_COMMANDLEN_SIZE in contribs/pmi2/pmi2_api.c to prevent
  certain implementation of snprintf() to segfault.

* Changes in Slurm 2.6.2
=====
-- Fix issue with reconfig and GrpCPURunMins
-- Fix of wrong node/job state problem after reconfig
-- Allow users who are coordinators update their own limits in the accounts
  they are coordinators over.
-- BackupController - Make sure we have a connection to the DBD first thing
  to avoid it thinking we don't have a cluster name.
-- Correct value of min_nodes returned by loading job information to consider
  the job's task count and maximum CPUs per node.
-- If running jobacct_gather/none fix issue on unpacking step completion.
-- Reservation with CoreCnt: Avoid possible invalid memory reference.
-- sjstat - Add man page when generating rpms.
-- Make sure GrpCPURunMins is added when creating a user, account or QOS with
  sacctmgr.
-- Fix for invalid memory reference due to multiple free calls caused by
  job arrays submitted to multiple partitions.
-- Enforce --ntasks-per-socket=1 job option when allocating by socket.
-- Validate permissions of key directories at slurmctld startup. Report
  anything that is world writable.
-- Improve GRES support for CPU topology. Previous logic would pick CPUs then
  reject jobs that can not match GRES to the allocated CPUs. New logic first
  filters out CPUs that can not use the GRES, next picks CPUs for the job,
  and finally picks the GRES that best match those CPUs.
-- Switch/nrt - Prevent invalid memory reference when allocating single adapter
  per node of specific adapter type
-- CRAY - Make Slurm work with CLE 5.1.1
-- Fix segfault if submitting to multiple partitions and holding the job.
-- Use MAXPATHLEN instead of the hardcoded value 1024 for maximum file path
  lengths.
-- If OverTimeLimit is defined do not declare failed those jobs that ended
  in the OverTimeLimit interval.

* Changes in Slurm 2.6.1
=====
-- slurmdbd - Allow job derived ec and comments to be modified by non-root
  users.
-- Fix issue with job name being truncated to 24 chars when sending a mail
  message.
-- Fix minor issues with spec file, missing files and including files
  erroneously on a bluegene system.
-- sacct - fix --name and --partition options when using
  accounting_storage/filetxt.
-- squeue - Remove extra whitespace of default printout.
-- BGQ - added head ppcfloor as an include dir when building.
-- BGQ - Better debug messages in runjob_mux plugin.
-- PMI2 Updated the Makefile.am to build a versioned library.
-- CRAY - Fix srun --mem_bind=local option with launch/aprun.
-- PMI2 Corrected buffer size computation in the pmi2_api.c module.
-- GRES accounting data wrong in database: gres_alloc, gres_req, and gres_used

```

Appendix G. SLURM Release Information

```
fields were empty if the job was not started immediately.
-- Fix sbatch and srun task count logic when --ntasks-per-node specified,
but no explicit task count.
-- Corrected the hdf5 profile user guide and the acct_gather.conf
documentation.
-- IPMI - Fix Math bug getting new wattage.
-- Corrected the AcctGatherProfileType documentation in slurm.conf
-- Corrected the sh5util program to print the header in the csv file
only once, set the debug messages at debug() level, make the argument
check case insensitive and avoid printing duplicate \n.
-- If cannot collect energy values send message to the controller
to drain the node and log error slurmd log file.
-- Handle complete removal of CPURunMins time at the end of the job instead
of at multifactor poll.
-- svview - Add missing debug_flag options.
-- PGSQL - Notes about Postgres functionality being removed in the next
version of Slurm.
-- MYSQL - fix issue when rolling up usage and events happened when a cluster
was down (slurmctld not running) during that time period.
-- sched/wiki2 - Insure that Moab gets current CPU load information.
-- Prevent infinite loop in parsing configuration if including file containing
one blank line.
-- Fix pack and unpack between 2.6 and 2.5.
-- Fix job state recovery logic in which a job's accounting frequency was
not set. This would result in a value of 65534 seconds being used (the
equivalent of NO_VAL in uint16_t), which could result in the job being
requeued or aborted.
-- Validate a job's accounting frequency at submission time rather than
waiting for it's initiation to possibly fail.
-- Fix CPURunMins if a job is requeued from a failed launch.
-- Fix in accounting_storage/filetxt to correct start times which sometimes
could end up before the job started.
-- Fix issue with potentially referencing past an array in parse_time()
-- CRAY - fix issue with accelerators on a cray when parsing BASIL 1.3 XML.
-- Fix issue with a 2.5 slurmd step locking up when talking to a 2.6 slurmd.
-- Add argument to priority plugin's priority_p_reconfig function to note
when the association and QOS used_cpu_run_secs field has been reset.

* Changes in Slurm 2.6.0
=====
-- Fix it so bluegene and serial systems don't get warnings over new NODEDATA
enum.
-- When a job is aborted send a message for any tasks that have completed.
-- Correction to memory per CPU calculation on system with threads and
allocating cores or sockets.
-- Requeue batch job if it's node reboots (used to abort the job).
-- Enlarge maximum size of srun's hostlist file.
-- IPMI - Fix first poll to get correct consumed_energy for a step.
-- Correction to job state recovery logic that could result in assert failure.
-- Record partial step accounting record if allocated nodes fail abnormally.
-- Accounting - fix issue where PrivateData=jobs or users could potentially
show information to users that had no associations on the system.
-- Make PrivateData in slurmdbd.conf case insensitive.
-- sacct/sstat - Add format option ConsumedEnergyRaw to print full energy
values.
```


* Changes in Slurm 2.6.0rc2

=====

- HDF5 - Fix issue with Ubuntu where HDF5 development headers are overwritten by the parallel versions thus making it so we need handle both cases.
- ACCT_GATHER - handle suspending correctly for polling threads.
- Make SLURM_DISTRIBUTION env var hold both types of distribution if specified.
- Remove hardcoded /usr/local from slurm.spec.
- Modify slurmctld locking to improve performance under heavy load with very large numbers of batch job submissions or job cancellations.
- sstat - Fix issue where if -j wasn't given allow last argument to be checked for as the job/step id.
- IPMI - fix adjustment on poll when using EnergyIPMICalcAdjustment.

* Changes in Slurm 2.6.0rc1

=====

- Added helper script for launching symmetric and MIC-only MPI tasks within SLURM (in contribs/mic/mpirun-mic).
- Change maximum delay for state save from 2 secs to 5 secs. Make timeout configurable at build time by defining SAVE_MAX_WAIT.
- Modify slurmctld data structure locking to interleave read and write locks rather than always favor write locks over read locks.
- Added sacct format option of "ALL" to print all fields.
- Deprecate the SchedulerParameters value of "interval" use "bf_interval" instead as documented.
- Add acct_gather_profile/hdf5 to profile jobs with hdf5
- Added MaxCPUsPerNode partition configuration parameter. This can be especially useful to schedule systems with GPUs.
- Permit "scontrol reboot_node" for nodes in MAINT reservation.
- Added "PriorityFlags" value of "SMALL_RELATIVE_TO_TIME". If set, the job's size component will be based upon not the job size alone, but the job's size divided by it's time limit.
- Added sbatch option "--ignore-pbs" to ignore "#PBS" options in the batch script.
- Rename slurm_step_ctx_params_t field from "mem_per_cpu" to "pn_min_memory". Job step now accepts memory specification in either per-cpu or per-node basis.
- Add ability to specify host repetition count in the srun hostfile (e.g. "host1*2" is equivalent to "host1,host1").

* Changes in Slurm 2.6.0pre3

=====

- Add milliseconds to default log message header (both RFC 5424 and ISO 8601 time formats). Disable milliseconds logging using the configure parameter "--disable-log-time-msec". Default time format changes to ISO 8601 (without time zone information). Specify "--enable-rfc5424time" to restore the time zone information.
- Add username (%u) to the filename pattern in the batch script.
- Added options for front end nodes of AllowGroups, AllowUsers, DenyGroups, and DenyUsers.
- Fix sched/backfill logic to initiate jobs with maximum time limit over the partition limit, but the minimum time limit permits it to start.
- gres/gpu - Fix for gres.conf file with multiple files on a single line

Appendix G. SLURM Release Information

```
using a slurm expression (e.g. "File=/dev/nvidia[0-1]").
-- Replaced ipmi.conf with generic acct_gather.conf file for all acct_gather
plugins. For those doing development to use this follow the model set
forth in the acct_gather_energy_ipmi plugin.
-- Added more options to update a step's information
-- Add DebugFlags=ThreadID which will print the thread id of the calling
thread.
-- CRAY - Allocate whole node (CPUs) in reservation despite what the
user requests. We have found any srun/aprun afterwards will work on a
subset of resources.

* Changes in Slurm 2.6.0pre2
=====
-- Do not purge inactive interactive jobs that lack a port to ping (added
for MR+ operation).
-- Advanced reservations with hostname and core counts now supports asymmetric
reservations (e.g. specific different core count for each node).
-- Added slurmctld/dynalloc plugin for MapReduce+ support.
-- Added "DynAllocPort" configuration parameter.
-- Added partition parameter of SelectTypeParameters to override system-wide
value.
-- Added cr_type to partition_info data structure.
-- Added allocated memory to node information available (within the existing
select_nodeinfo field of the node_info_t data structure). Added Allocated
Memory to node information displayed by sview and scontrol commands.
-- Make sched/backfill the default scheduling plugin rather than sched/builtin
(FIFO).
-- Added support for a job having different priorities in different partitions.
-- Added new SchedulerParameters configuration parameter of "bf_continue"
which permits the backfill scheduler to continue considering jobs for
backfill scheduling after yielding locks even if new jobs have been
submitted. This can result in lower priority jobs from being backfill
scheduled instead of newly arrived higher priority jobs, but will permit
more queued jobs to be considered for backfill scheduling.
-- Added support to purge reservation records from accounting.
-- Cray - Add support for Basil 1.3

* Changes in SLURM 2.6.0pre1
=====
-- Add "state" field to job step information reported by scontrol.
-- Notify srun to retry step creation upon completion of other job steps
rather than polling. This results in much faster throughput for job step
execution with --exclusive option.
-- Added "ResvEpilog" and "ResvProlog" configuration parameters to execute a
program at the beginning and end of each reservation.
-- Added "slurm_load_job_user" function. This is a variation of
"slurm_load_jobs", but accepts a user ID argument, potentially resulting
in substantial performance improvement for "squeue --user=ID"
-- Added "slurm_load_node_single" function. This is a variation of
"slurm_load_nodes", but accepts a node name argument, potentially resulting
in substantial performance improvement for "sinfo --nodes=NAME".
-- Added "HealthCheckNodeState" configuration parameter identify node states
on which HealthCheckProgram should be executed.
-- Remove sacct --dump --formatted-dump options which were deprecated in
2.5.
```

```
-- Added support for job arrays (phase 1 of effort). See "man sbatch" option
--a/--array for details.
-- Add new AccountStorageEnforce options of 'nojobs' and 'nosteps' which will
allow the use of accounting features like associations, qos and limits but
not keep track of jobs or steps in accounting.
-- Cray - Add new cray.conf parameter of "AlpsEngine" to specify the
communication protocol to be used for ALPS/BASIL.
-- select/cons_res plugin: Correction to CPU allocation count logic in for
cores without hyperthreading.
-- Added new SelectTypeParameter value of "CR_ALLOCATE_FULL_SOCKET".
-- Added PriorityFlags value of "TICKET_BASED" and merged priority/multifactor2
plugin into priority/multifactor plugin.
-- Add "KeepAliveTime" configuration parameter controlling how long sockets
used for srun/slurmstepd communications are kept alive after disconnect.
-- Added SLURM_SUBMIT_HOST to salloc, sbatch and srun job environment.
-- Added SLURM_ARRAY_TASK_ID to environment of job array.
-- Added squeue --array/-r option to optimize output for job arrays.
-- Added "SlurmctldPlugstack" configuration parameter for generic stack of
slurmctld daemon plugins.
-- Removed contribs/arrayrun tool. Use native support for job arrays.
-- Modify default installation locations for RPMs to match "make install":
_prefix /usr/local
_slurm_sysconffdir %{_prefix}/etc/slurm
_mandir %{_prefix}/share/man
_infodir %{_prefix}/share/info
-- Add acct_gather_energy/ipmi which works off freeipmi for energy gathering
```

* Changes in Slurm 2.5.8

=====

```
-- Fix for slurmctld segfault on NULL front-end reason field.
-- Avoid gres step allocation errors when a job shrinks in size due to either
down nodes or explicit resizing. Generated slurmctld errors of this type:
"step_test ... gres_bit_alloc is NULL"
-- Fix bug that would leak memory and over-write the AllowGroups field if on
"scontrol reconfig" when AllowNodes is manually changed using scontrol.
-- Get html/man files to install in correct places with rpms.
-- Remove --program-prefix from spec file since it appears to be added by
default and appeared to break other things.
-- Updated the automake min version in autogen.sh to be correct.
-- Select/cons_res - Correct total CPU count allocated to a job with
--exclusive and --cpus-per-task options
-- switch/nrt - Don't allocate network resources unless job step has 2+ nodes.
-- select/cons_res - Avoid extraneous "oversubscribe" error messages.
-- Reorder get config logic to avoid deadlock.
-- Enforce QOS MaxCPUsMin limit when job submission contains no user-specified
time limit.
-- EpilogSlurmctld pthread is passed required arguments rather than a pointer
to the job record, which under some conditions could be purged and result
in an invalid memory reference.
```

* Changes in Slurm 2.5.7

=====

```
-- Fix for linking to the select/cray plugin to not give warning about
undefined variable.
-- Add missing symbols to the xlator.h
```

Appendix G. SLURM Release Information

- Avoid placing pending jobs in AdminHold state due to backfill scheduler interactions with advanced reservation.
- Accounting - make average by task not cpu.
- CRAY - Change logging of transient ALPS errors from error() to debug().
- POE - Correct logic to support poe option "-euidevice sn_all" and "-euidevice sn_single".
- Accounting - Fix minor initialization error.
- POE - Correct logic to support srun network instances count with POE.
- POE - With the srun --launch-cmd option, report proper task count when the --cpus-per-task option is used without the --ntasks option.
- POE - Fix logic binding tasks to CPUs.
- sview - Fix race condition where new information could of slipped past the node tab and we didn't notice.
- Accounting - Fix an invalid memory read when slurmctld sends data about start job to slurmdbd.
- If a prolog or epilog failure occurs, drain the node rather than setting it down and killing all of its jobs.
- Priority/multifactor - Avoid underflow in half-life calculation.
- POE - pack missing variable to allow fanout (more than 32 nodes)
- Prevent clearing reason field for pending jobs. This bug was introduced in v2.5.5 (see "Reject job at submit time ...").
- BGQ - Fix issue with preemption on sub-block jobs where a job would kill all preemptable jobs on the midplane instead of just the ones it needed to.
- switch/nrt - Validate dynamic window allocation size.
- BGQ - When --geo is requested do not impose the default conn_types.
- CRAY - Support CLE 4.2.0
- RebootNode logic - Defers (rather than forgets) reboot request with job running on the node within a reservation.
- switch/nrt - Correct network_id use logic. Correct support for user sn_all and sn_single options.
- sched/backfill - Modify logic to reduce overhead under heavy load.
- Fix job step allocation with --exclusive and --hostlist option.
- Select/cons_res - Fix bug resulting in error of "cons_res: sync loop not progressing, holding job #"
- checkpoint/blcr - Reset max_nodes from zero to NO_VAL on job restart.
- launch/poe - Fix for hostlist file support with repeated host names.
- priority/multifactor2 - Prevent possible divide by zero.
- srun - Don't check for executable if --test-only flag is used.
- energy - On a single node only use the last task for gathering energy. Since we don't currently track energy usage per task (only per step). Otherwise we get double the energy.

* Changes in Slurm 2.5.6

=====

- Gres fix for queued jobs.
- Gres accounting - Fix regression in 2.5.5 for keeping track of gres requested and allocated.

* Changes in Slurm 2.5.5

=====

- Fix for sacctmgr add qos to handle the 'flags' option.
- Export SLURM_ environment variables from sbatch, even if "--export" option does not explicitly list them.
- If node is in more than one partition, correct counting of allocated CPUs.
- If step requests more CPUs than possible in specified node count of job

```

allocation then return ESLURM_TOO_MANY_REQUESTED_CPUS rather than
ESLURM_NODES_BUSY and retrying.
-- CRAY - Fix SLURM_TASKS_PER_NODE to be set correctly.
-- Accounting - more checks for strings with a possible `` in it.
-- sreport - Fix by adding planned down time to utilization reports.
-- Do not report an error when sstat identifies job steps terminated during
  its execution, but log using debug type message.
-- Select/cons_res - Permit node removed from job by going down to be returned
  to service and re-used by another job.
-- Select/cons_res - Tighter packing of job allocations on sockets.
-- SlurmDBD - fix to allow user root along with the slurm user to register a
  cluster.
-- Select/cons_res - Fix for support of consecutive node option.
-- Select/cray - Modify build to enable direct use of libslurm library.
-- Bug fixes related to job step allocation logic.
-- Cray - Disable enforcement of MaxTasksPerNode, which is not applicable
  with launch/aprun.
-- Accounting - When rolling up data from past usage ignore "idle" time from
  a reservation when it has the "Ignore_Jobs" flag set. Since jobs could run
  outside of the reservation in it's nodes without this you could have
  double time.
-- Accounting - Minor fix to avoid reuse of variable erroneously.
-- Reject job at submit time if the node count is invalid. Previously such a
  job submitted to a DOWN partition would be queued.
-- Purge vestigial job scripts when the slurmd cold starts or slurmstepd
  terminates abnormally.
-- Add support for FreeBSD.
-- Add sanity check for NULL cluster names trying to register.
-- BGQ - Push action 'D' info to scontrol for admins.
-- Reset a job's reason from PartitionDown when the partition is set up.
-- BGQ - Handle issue where blocks would have a pending job on them and
  while it was free cnodes would go into software error and kill the job.
-- BGQ - Fix issue where if for some reason we are freeing a block with
  a pending job on it we don't kill the job.
-- BGQ - Fix race condition were a job could of been removed from a block
  without it still existing there. This is extremely rare.
-- BGQ - Fix for when a step completes in Slurm before the runjob_mux notifies
  the slurmd there were software errors on some nodes.
-- BGQ - Fix issue on state recover if block states are not around
  and when reading in state from DB2 we find a block that can't be created.
  You can now do a clean start to rid the bad block.
-- Modify slurmdbd to retransmit to slurmd daemon if it is not responding.
-- BLUEGENE - Fix issue where when doing backfill preemptable jobs were
  never looked at to determine eligibility of backfillable job.
-- Cray/BlueGene - Disable srun --pty option unless LaunchType=launch/slurm.
-- CRAY - Fix sanity check for systems with more than 32 cores per node.
-- CRAY - Remove other objects from MySQL query that are available from
  the XML.
-- BLUEGENE - Set the geometry of a job when a block is picked and the job
  isn't a sub-block job.
-- Cray - avoid check of macro versions of CLE for version 5.0.
-- CRAY - Fix memory issue with reading in the cray.conf file.
-- CRAY - If hostlist is given with srun make sure the node count is the same
  as the hosts given.
-- CRAY - If task count specified, but no tasks-per-node, then set the tasks

```

Appendix G. SLURM Release Information

```
per node in the BASIL reservation request.
-- CRAY - fix issue with --mem option not giving correct amount of memory
per cpu.
-- CRAY - Fix if srun --mem is given outside an allocation to set the
APRUN_DEFAULT_MEMORY env var for aprun. This scenario will not display
the option when used with --launch-cmd.
-- Change svview to use GMutex instead of GStaticMutex
-- CRAY - set APRUN_DEFAULT_MEMROY instead of CRAY_AUTO_APRUN_OPTIONS
-- svview - fix issue where if a partition was completely in one state the
cpu count would be reflected correctly.
-- BGQ - fix for handling half rack system in STATIC of OVERLAP mode to
implicitly create full system block.
-- CRAY - Dynamically create BASIL XML buffer to resize as needed.
-- Fix checking if QOS limit MaxCPUMinsPJ is set along with DenyOnLimit to
deny the job instead of holding it.
-- Make sure on systems that use a different launcher than launch/slurm not
to attempt to signal tasks on the frontend node.
-- Cray - when a step is requested count other steps running on nodes in the
allocation as taking up the entire node instead of just part of the node
allocated. And always enforce exclusive on a step request.
-- Cray - display correct nodelist, node/cpu count on steps.

* Changes in Slurm 2.5.4
=====
-- Fix bug in PrologSlurmctld use that would block job steps until node
responds.
-- CRAY - If a partition has MinNodes=0 and a batch job doesn't request nodes
put the allocation to 1 instead of 0 which prevents the allocation to
happen.
-- Better debug when the database is down and using the --cluster option in
the user commands.
-- When asking for job states with sacct, default to 'now' instead of midnight
of the current day.
-- Fix for handling a test-only job or immediate job that fails while being
built.
-- Comment out all of the logic in the job_submit/defaults plugin. The logic
is only an example and not meant for actual use.
-- Eliminate configuration file 4096 character line limitation.
-- More robust logic for tree message forward
-- BGQ - When cnodes fail in a timeout fashion correctly look up parent
midplane.
-- Correct sinfo "%c" (node's CPU count) output value for Bluegene systems.
-- Backfill - Responsive improvements for systems with large numbers of jobs
(>5000) and using the SchedulerParameters option bf_max_job_user.
-- slurmstepd: ensure that IO redirection openings from/to files correctly
handle interruption
-- BGQ - Able to handle when midplanes go into Hardware::SoftwareFailure
-- GRES - Correct tracking of specific resources used after slurmctld restart.
Counts would previously go negative as jobs terminate and decrement from
a base value of zero.
-- Fix for priority/multifactor2 plugin to not assert when configured with
--enable-debug.
-- Select/cons_res - If the job request specified --ntasks-per-socket and the
allocation using is cores, then pack the tasks onto the sockets up to the
specified value.
```

- BGQ - If a cnode goes into an 'error' state and the block containing the cnode does not have a job running on it do not resume the block.
- BGQ - Handle blocks that don't free themselves in a reasonable time better.
- BGQ - Fix for signaling steps when allocation ends before step.
- Fix for backfill scheduling logic with job preemption; starts more jobs.
- xcgrouop - remove bugs with EINTR management in write calls
- jobacct_gather - fix total values to not always == the max values.
- Fix for handling node registration messages from older versions without energy data.
- BGQ - Allow user to request full dimensional mesh.
- sdiag command - Correction to jobs started value reported.
- Prevent slurmctld assert when invalid change to reservation with running jobs is made.
- BGQ - If signal is NODE_FAIL allow forward even if job is completing and timeout in the runjob_mux trying to send in this situation.
- BGQ - More robust checking for correct node, task, and ntasks-per-node options in srun, and push that logic to salloc and sbatch.
- GRES topology bug in core selection logic fixed.
- Fix to handle init.d script for querying status and not return 1 on success.

* Changes in SLURM 2.5.3

=====

- Gres/gpu plugin - If no GPUs requested, set CUDA_VISIBLE_DEVICES=NoDevFiles. This bug was introduced in 2.5.2 for the case where a GPU count was configured, but without device files.
- task/affinity plugin - Fix bug in CPU masks for some processors.
- Modify sacct command to get format from SACCT_FORMAT environment variable.
- BGQ - Changed order of library inclusions and fixed incorrect declaration to compile correctly on newer compilers
- Fix for not building svview if glib exists on a system but not the gtk libs.
- BGQ - Fix for handling a job cleanup on a small block if the job has long since left the system.
- Fix race condition in job dependency logic which can result in invalid memory reference.

* Changes in SLURM 2.5.2

=====

- Fix advanced reservation recovery logic when upgrading from version 2.4.
- BLUEGENE - fix for QOS/Association node limits.
- Add missing "safe" flag from print of AccountStorageEnforce option.
- Fix logic to optimize GRES topology with respect to allocated CPUs.
- Add job_submit/all_partitions plugin to set a job's default partition to ALL available partitions in the cluster.
- Modify switch/nrt logic to permit build without libnrt.so library.
- Handle srun task launch failure without duplicate error messages or abort.
- Fix bug in QoS limits enforcement when slurmctld restarts and user not yet added to the QOS list.
- Fix issue where sjstat and sjobexitmod was installed in 2 different RPMs.
- Fix for job request of multiple partitions in which some partitions lack nodes with required features.
- Permit a job to use a QOS they do not have access to if an administrator manually set the job's QOS (previously the job would be rejected).
- Make more variables available to job_submit/luva plugin: slurm.MEM_PER_CPU,

Appendix G. SLURM Release Information

```
    slurm.NO_VAL, etc.
-- Fix topology/tree logic when nodes defined in slurm.conf get re-ordered.
-- In select/cons_res, correct logic to allocate whole sockets to jobs. Work
  by Magnus Jonsson, Umea University.
-- In select/cons_res, correct logic when job removed from only some nodes.
-- Avoid apparent kernel bug in 2.6.32 which apparently is solved in
  at least 3.5.0. This avoids a stack overflow when running jobs on
  more than 120k nodes.
-- BLUEGENE - If we made a block that isn't runnable because of a overlapping
  block, destroy it correctly.
-- Switch/nrt - Dynamically load libnrt.so from within the plugin as needed.
  This eliminates the need for libnrt.so on the head node.
-- BLUEGENE - Fix in reservation logic that could cause abort.

* Changes in SLURM 2.5.1
=====
-- Correction to hostlist sorting for hostnames that contain two numeric
  components and the first numeric component has various sizes (e.g.
  "rack9blade1" should come before "rack10blade1")
-- BGQ - Only poll on initialized blocks instead of calling getBlocks on
  each block independently.
-- Fix of task/affinity plugin logic for Power7 processors having hyper-
  threading disabled (cpu mask has gaps).
-- Fix of job priority ordering with sched/builtin and priority/multifactor.
  Patch from Chris Read.
-- CRAY - Fix for setting up the aprun for a large job (+2000 nodes).
-- Fix for race condition related to compute node boot resulting in node being
  set down with reason of "Node <name> unexpectedly rebooted"
-- RAPL - Fix for handling errors when opening msr files.
-- BGQ - Fix for salloc/sbatch to do the correct allocation when asking for
  -N1 -n#.
-- BGQ - in emulation make it so we can pretend to run large jobs (>64k nodes)
-- BLUEGENE - Correct method to update conn_type of a job.
-- BLUEGENE - Fix issue with preemption when needing to preempt multiple jobs
  to make one job run.
-- Fixed issue where if an srun dies inside of an allocation abnormally it
  would of also killed the allocation.
-- FRONTEND - fixed issue where if a systems nodes weren't defined in the
  slurm.conf with NodeAddr's signals going to a step could be handled
  incorrectly.
-- If sched/backfill starts a job with a QOS having NO_RESERVE and not job
  time limit, start it with the partition time limit (or one year if the
  partition has no time limit) rather than NO_VAL (140 year time limit);
-- Alter hostlist logic to allocate large grid dynamically instead of on
  stack.
-- Change RPC version checks to support version 2.5 slurmctld with version 2.4
  slurmd daemons.
-- Correct core reservation logic for use with select/serial plugin.
-- Exit scontrol command on stdin EOF.
-- Disable job --exclusive option with select/serial plugin.

* Changes in SLURM 2.5.0
=====
-- Add DenyOnLimit flag for QOS to deny jobs at submission time if they
  request resources that reach a 'Max' limit.
```


- Permit SlurmUser or operator to change QOS of non-pending jobs (e.g. running jobs).
- BGQ - move initial poll to beginning of realtime interaction, which will also cause it to run if the realtime server ever goes away.

* Changes in SLURM 2.5.0-rc2

=====

- Modify sbcast logic to survive slurmd daemon restart while file a transmission is in progress.
- Add retry logic to munge encode/decode calls. This is needed if the munge daemon is under very heavy load (e.g. with 1000 slurmd daemons per compute node).
- Add launch and acct_gather_energy plugins to RPMs.
- Restore support for srun "--mpi=list" option.
- CRAY - Introduce step accounting for a Cray.
- Modify srun to abandon I/O 60 seconds after the last task ends. Otherwise an aborted slurmd can cause the srun process to hang indefinitely.
- ENERGY - RAPL - alter code to close open files (and only open them once where needed)
- If the PrologSlurmctld fails, then requeue the job an indefinite number of times instead of only one time.

* Changes in SLURM 2.5.0-rc1

=====

- Added Prolog and Epilog Guide (web page). Based upon work by Jason Sollom, Cray Inc. and used by permission.
- Restore gang scheduling functionality. Preemptor was not being scheduled. Fix for bugzilla #3.
- Add "cpu_load" to node information. Populate CPULOAD in node information reported to Moab cluster manager.
- Preempt jobs only when insufficient idle resources exist to start job, regardless of the node weight.
- Added priority/multifactor2 plugin based upon ticket distribution system. Work by Janne Blomqvist, Aalto University.
- Add SLURM_NODELIST to environment variables available to Prolog and Epilog.
- Permit reservations to allow or deny access by account and/or user.
- Add ReconfigFlags value of KeepPartState. See "man slurm.conf" for details.
- Modify the task/cgroup plugin adding a task_pre_launch_priv function and move slurmd outside of the step's cgroup. Work by Matthieu Hautreux.
- Intel MIC processor support added using gres/mic plugin. BIG thanks to Olli-Pekka Lehto, CSC-IT Center for Science Ltd.
- Accounting - Change empty jobacctinfo structs to not actually be used instead of putting 0's into the database we put NO_VALS and have sacct figure out jobacct_gather wasn't used.
- Cray - Prevent calling basil_confirm more than once per job using a flag.
- Fix bug with topology/tree and job with min-max node count. Now try to get max node count rather than minimizing leaf switches used.
- Add AccountingStorageEnforce=safe option to provide method to avoid jobs launching that wouldn't be able to run to completion because of a GrpCPUMins limit.
- Add support for RFC 5424 timestamps in logfiles. Disable with configuration option of "--disable-rfc5424time". By Janne Blomqvist, Aalto University.
- CRAY - Replace srun.pl with launch/aprun plugin to use srun to wrap the aprun process instead of a perl script.
- srun - Rename --runjob-opts to --launcher-opts to be used on systems other

Appendix G. SLURM Release Information

```
    than BGQ.
-- Added new DebugFlags - Energy for AcctGatherEnergy plugins.
-- start deprecation of sacct --dump --fdump
-- BGQ - added --verbose=OFF when srun --quiet is used
-- Added acct_gather_energy/rapl plugin to record power consumption by job.
    Work by Yiannis Georgiou, Martin Perry, et. al., Bull

* Changes in SLURM 2.5.0.pre3
=====
-- Add Google search to all web pages.
-- Add sinfo -T option to print reservation information. Work by Bill Brophy,
    Bull.
-- Force slurmd exit after 2 minute wait, even if threads are hung.
-- Change node_req field in struct job_resources from 8 to 32 bits so we can
    run more than 256 jobs per node.
-- sched/backfill: Improve accuracy of expected job start with respect to
    reservations.
-- sinfo partition field size will be set the the length of the longest
    partition name by default.
-- Make it so the parse_time will return a valid 0 if given epoch time and
    set errno == ESLURM_INVALID_TIME_VALUE on error instead.
-- Correct srun --no-alloc logic when node count exceeds node list or task
    task count is not a multiple of the node count. Work by Hongjia Cao, NUDT.
-- Completed integration with IBM Parallel Environment including POE and IBM's
    NRT switch library.

* Changes in SLURM 2.5.0.pre2
=====
-- When running with multiple slurmd daemons per node, enable specifying a
    range of ports on a single line of the node configuration in slurm.conf.
-- Add reservation flag of Part_Nodes to allocate all nodes in a partition to
    a reservation and automatically change the reservation when nodes are
    added to or removed from the reservation. Based upon work by
    Bill Brophy, Bull.
-- Add support for advanced reservation for specific cores rather than whole
    nodes. Current limiations: homogeneous cluster, nodes idle when reservation
    created, and no more than one reservation per node. Code is still under
    development. Work by Alejandro Lucero Palau, et. al, BSC.
-- Add DebugFlag of Switch to log switch plugin details.
-- Correct job node_cnt value in job completion plugin when job fails due to
    down node. Previously was too low by one.
-- Add new srun option --cpu-freq to enable user control over the job's CPU
    frequency and thus it's power consumption. NOTE: cpu frequency is not
    currently preserved for jobs being suspended and later resumed. Work by
    Don Albert, Bull.
-- Add node configuration information about "boards" and optimize task
    placement on minimum number of boards. Work by Rod Schultz, Bull.

* Changes in SLURM 2.5.0.pre1
=====
-- Add new output to "scontrol show configuration" of LicensesUsed. Output is
    "name:used/total"
-- Changed jobacct_gather plugin infrastructure to be cleaner and easier to
    maintain.
-- Change license option count separator from "*" to ":" for consistency with
```

```

the gres option (e.g. "--licenses=foo:2 --gres=gpu:2"). The "*" will still
be accepted, but is no longer documented.
-- Permit more than 100 jobs to be scheduled per node (new limit is 250
jobs).
-- Restructure of srun code to allow outside programs to utilize existing
logic.

* Changes in SLURM 2.4.6
=====
-- Correct WillRun authentication logic when issued for non-job owner.
-- BGQ - fix memory leak
-- BGQ - Fix to check block for action 'D' if it also has nodes in error.

* Changes in SLURM 2.4.5
=====
-- Cray - On job kill request, send SIGCONT, SIGTERM, wait KillWait and send
SIGKILL. Previously just sent SIGKILL to tasks.
-- BGQ - Fix issue when running srun outside of an allocation and only
specifying the number of tasks and not the number of nodes.
-- BGQ - validate correct ntasks_per_node
-- BGQ - when srun -Q is given make runjob be quiet
-- Modify use of OOM (out of memory protection) for Linux 2.6.36 kernel
or later. NOTE: If you were setting the environment variable
SLURMSTEPD_OOM_ADJ=-17, it should be set to -1000 for Linux 2.6.36 kernel
or later.
-- BGQ - Fix job step timeout actually happen when done from within an
allocation.
-- Reset node MAINT state flag when a reservation's nodes or flags change.
-- Accounting - Fix issue where QOS usage was being zeroed out on a
slurmctld restart.
-- BGQ - Add 64 tasks per node as a valid option for srun when used with
overcommit.
-- BLUEGENE - With Dynamic layout mode - Fix issue where if a larger block
was already in error and isn't deallocating and underlying hardware goes
bad one could get overlapping blocks in error making the code assert when
a new job request comes in.
-- BGQ - handle pending actions on a block better when trying to deallocate it.
-- Accounting - Fixed issue where if nodenames have changed on a system and
you query against that with -N and -E you will get all jobs during that
time instead of only the ones running on -N.
-- BGP - Fix for HTC mode
-- Accounting - If a job start message fails to the SlurmDBD reset the db_inx
so it gets sent again. This isn't a major problem since the start will
happen when the job ends, but this does make things cleaner.
-- If an salloc is waiting for an allocation to happen and is canceled by the
user mark the state canceled instead of completed.
-- Fix issue in accounting if a user puts a '\' in their job name.
-- Accounting - Fix for if asking for users or accounts that were deleted
with associations get the deleted associations as well.
-- BGQ - Handle shared blocks that need to be removed and have jobs running
on them. This should only happen in extreme conditions.
-- Fix inconsistency for hostlists that have more than 1 range.
-- BGQ - Add mutex around recovery for the Real Time server to avoid hitting
DB2 so hard.
-- BGQ - If an allocation exists on a block that has a 'D' action on it fail

```

Appendix G. SLURM Release Information

job on future step creation attempts.

* Changes in SLURM 2.4.4

=====

- BGQ - minor fix to make build work in emulated mode.
- BGQ - Fix if large block goes into error and the next highest priority jobs are planning on using the block. Previously it would fail those jobs erroneously.
- BGQ - Fix issue when a cnode going to an error (not SoftwareError) state with a job running or trying to run on it.
- Execute slurm_spank_job_epilog when there is no system Epilog configured.
- Fix for srun --test-only to work correctly with timelimits
- BGQ - If a job goes away while still trying to free it up in the database, and the job is running on a small block make sure we free up the correct node count.
- BGQ - Logic added to make sure a job has finished on a block before it is purged from the system if its front-end node goes down.
- Modify strigger so that a filter option of "--user=0" is supported.
- Correct --mem-per-cpu logic for core or socket allocations with multiple threads per core.
- Fix for older < glibc 2.4 systems to use euidaccess() instead of eaccess().
- BLUEGENE - Do not alter a pending job's node count when changing it's partition.
- BGQ - Add functionality to make it so we track the actions on a block. This is needed for when a free request is added to a block but there are jobs finishing up so we don't start new jobs on the block since they will fail on start.
- BGQ - Fixed InactiveLimit to work correctly to avoid scenarios where a user's pending allocation was started with srun and then for some reason the slurmctld was brought down and while it was down the srun was removed.
- Fixed InactiveLimit math to work correctly
- BGQ - Add logic to make it so blocks can't use a midplane with a nodeboard in error for passthrough.
- BGQ - Make it so if a nodeboard goes in error any block using that midplane for passthrough gets removed on a dynamic system.
- BGQ - Fix for printing realtime server debug correctly.
- BGQ - Cleaner handling of cnode failures when reported through the runjob interface instead of through the normal method.
- smap - spread node information across multiple lines for larger systems.
- Cray - Defer salloc until after PrologSlurmctld completes.
- Correction to slurmdbd communications failure handling logic, incorrect error codes returned in some cases.

* Changes in SLURM 2.4.3

=====

- Accounting - Fix so complete 32 bit numbers can be put in for a priority.
- cgroups - fix if initial directory is non-existent SLURM creates it correctly. Before the errno wasn't being checked correctly
- BGQ - fixed srun when only requesting a task count and not a node count to operate the same way salloc or sbatch did and assign a task per cpu by default instead of task per node.
- Fix salloc --gid to work correctly. Reported by Brian Gilmer
- BGQ - fix smap to set the correct default MloaderImage
- BLUEGENE - updated documentation.
- Close the batch job's environment file when it contains no data to avoid

leaking file descriptors.

- Fix sbcast's credential to last till the end of a job instead of the previous 20 minute time limit. The previous behavior would fail for large files 20 minutes into the transfer.
- Return ESLURM_NODES_BUSY rather than ESLURM_NODE_NOT_AVAIL error on job submit when required nodes are up, but completing a job or in exclusive job allocation.
- Add HWLOC_FLAGS so linking to libslurm works correctly
- BGQ - If using backfill and a shared block is running at least one job and a job comes through backfill and can fit on the block without ending jobs don't set an end_time for the running jobs since they don't need to end to start the job.
- Initialize bind_verbose when using task/cgroup.
- BGQ - Fix for handling backfill much better when sharing blocks.
- BGQ - Fix for making small blocks on first pass if not sharing blocks.
- BLUEGENE - Remove force of default conn_type instead of leaving NAV when none are requested. The Block allocator sets it up temporarily so this isn't needed.
- BLUEGENE - Fix deadlock issue when dealing with bad hardware if using static blocks.
- Fix to mysql plugin during rollup to only query suspended table when jobs reported some suspended time.
- Fix compile with glibc 2.16 (Kacper Kowalik)
- BGQ - fix for deadlock where a block has error on it and all jobs running on it are preemptable by scheduling job.
- proctrack/cgroup: Exclude internal threads from "scontrol list pids". Patch from Matthieu Hautreux, CEA.
- Memory leak fixed for select/linear when preempting jobs.
- Fix if updating begin time of a job to update the eligible time in accounting as well.
- BGQ - make it so you can signal steps when signaling the job allocation.
- BGQ - Remove extra overhead if a large block has many cnode failures.
- Priority/Multifactor - Fix issue with age factor when a job is estimated to start in the future but is able to run now.
- CRAY - update to work with ALPS 5.1
- BGQ - Handle issue of speed and mutexes when polling instead of using the realtime server.
- BGQ - Fix minor sorting issue with sview when sorting by midplanes.
- Accounting - Fix for handling per user max node/cpus limits on a QOS correctly for current job.
- Update documentation for -/+ when updating a reservation's users/accounts/flags
- Update pam module to work if using aliases on nodes instead of actual host names.
- Correction to task layout logic in select/cons_res for job with minimum and maximum node count.
- BGQ - Put final poll after realtime comes back into service to avoid having the realtime server go down over and over again while waiting for the poll to finish.
- task/cgroup/memory - ensure that ConstrainSwapSpace=no is correctly handled. Work by Matthieu Hautreux, CEA.
- CRAY - Fix for sacct -N option to work correctly
- CRAY - Update documentation to describe installation from rpm instead or previous piecemeal method.
- Fix sacct to work with QOS' that have previously been deleted.

Appendix G. SLURM Release Information

```
-- Added all available limits to the output of sacctmgr list qos

* Changes in SLURM 2.4.2
=====
-- BLUEGENE - Correct potential deadlock issue when hardware goes bad and
  there are jobs running on that hardware.
-- If job is submitted to more than one partition, it's partition pointer can
  be set to an invalid value. This can result in the count of CPUs allocated
  on a node being bad, resulting in over- or under-allocation of its CPUs.
  Patch by Carles Fenoy, BSC.
-- Fix bug in task layout with select/cons_res plugin and --ntasks-per-node
  option. Patch by Martin Perry, Bull.
-- BLUEGENE - remove race condition where if a block is removed while waiting
  for a job to finish on it the number of unused cpus wasn't updated
  correctly.
-- BGQ - make sure we have a valid block when creating or finishing a step
  allocation.
-- BLUEGENE - If a large block (> 1 midplane) is in error and underlying
  hardware is marked bad remove the larger block and create a block over
  just the bad hardware making the other hardware available to run on.
-- BLUEGENE - Handle job completion correctly if an admin removes a block
  where other blocks on an overlapping midplane are running jobs.
-- BLUEGENE - correctly remove running jobs when freeing a block.
-- BGQ - correct logic to place multiple (< 1 midplane) steps inside a
  multi midplane block allocation.
-- BGQ - Make it possible for a multi midplane allocation to run on more
  than 1 midplane but not the entire allocation.
-- BGL - Fix for syncing users on block from Tim Wickberg
-- Fix initialization of protocol_version for some messages to make sure it
  is always set when sending or receiving a message.
-- Reset backfilled job counter only when explicitly cleared using scontrol.
  Patch from Alejandro Lucero Palau, BSC.
-- BLUEGENE - Fix for handling blocks when a larger block will not free and
  while it is attempting to free underlying hardware is marked in error
  making small blocks overlapping with the freeing block. This only
  applies to dynamic layout mode.
-- Cray and BlueGene - Do not treat lack of usable front-end nodes when
  slurmctld daemon starts as a fatal error. Also preserve correct front-end
  node for jobs when there is more than one front-end node and the slurmctld
  daemon restarts.
-- Correct parsing of srun/sbatch input/output/error file names so that only
  the name "none" is mapped to /dev/null and not any file name starting
  with "none" (e.g. "none.o").
-- BGQ - added version string to the load of the runjob_mux plugin to verify
  the current plugin has been loaded when using runjob_mux_refresh_config
-- CGROUPS - Use system mount/umount function calls instead of doing fork
  exec of mount/umount from Janne Blomqvist.
-- BLUEGENE - correct start time setup when no jobs are blocking the way
  from Mark Nelson
-- Fixed sacct --state=S query to return information about suspended jobs
  current or in the past.
-- FRONTEND - Made error warning more apparent if a frontend node isn't
  configured correctly.
-- BGQ - update documentation about runjob_mux_refresh_config which works
  correctly as of IBM driver V1R1M1 efix 008.
```

* Changes in SLURM 2.4.1

=====

- Fix bug for job state change from 2.3 -> 2.4 job state can now be preserved correctly when transitioning. This also applies for 2.4.0 -> 2.4.1, no state will be lost. (Thanks to Carles Fenoy)

* Changes in SLURM 2.4.0

=====

- Cray - Improve support for zero compute node resource allocations. Partition used can now be configured with no nodes.
- BGQ - make it so srun -i<taskid> works correctly.
- Fix parse_uint32/16 to complain if a non-digit is given.
- Add SUBMITHOST to job state passed to Moab vial sched/wiki2. Patch by Jon Bringhurst (LANL).
- BGQ - Fix issue when running with AllowSubBlockAllocations=Yes without compiling with --enable-debug
- Modify scontrol to require "--dd" option to report batch job's script. Patch from Don Albert, Bull.
- Modify SchedulerParameters option to match documentation: "bf_res=" changed to "bf_resolution=". Patch from Rod Schultz, Bull.
- Fix bug that clears job pending reason field. Patch from Don Lipari, LLNL.
- In etc/init.d/slurm move check for scontrol after sourcing /etc/sysconfig/slurm. Patch from Andy Wettstein, University of Chicago.
- Fix in scheduling logic that can delay jobs with min/max node counts.
- BGQ - fix issue where if a step uses the entire allocation and then the next step in the allocation only uses part of the allocation it gets the correct cnodes.
- BGQ - Fix checking for IO on a block with new IBM driver V1R1M1 previous function didn't always work correctly.
- BGQ - Fix issue when a nodeboard goes down and you want to combine blocks to make a larger small block and are running with sub-blocks.
- BLUEGENE - Better logic for making small blocks around bad nodeboard/card.
- BGQ - When using an old IBM driver cnodes that go into error because of a job kill timeout aren't always reported to the system. This is now handled by the runjob_mux plugin.
- BGQ - Added information on how to setup the runjob_mux to run as SlurmUser.
- Improve memory consumption on step layouts with high task count.
- BGQ - quieter debug when the real time server comes back but there are still messages we find when we poll but haven't given it back to the real time yet.
- BGQ - fix for if a request comes in smaller than the smallest block and we must use a small block instead of a shared midplane block.
- Fix issues on large jobs (>64k tasks) to have the correct counter type when packing the step layout structure.
- BGQ - fix issue where if a user was asking for tasks and ntasks-per-node but not node count the node count is correctly figured out.
- Move logic to always use the 1st alphanumeric node as the batch host for batch jobs.
- BLUEGENE - fix race condition where if a nodeboard/card goes down at the same time a block is destroyed and that block just happens to be the smallest overlapping block over the bad hardware.
- Fix bug when querying accounting looking for a job node size.
- BLUEGENE - fix possible race condition if cleaning up a block and the removal of the job on the block failed.

Appendix G. SLURM Release Information

- BLUEGENE - fix issue if a cable was in an error state make it so we can check if a block is still makable if the cable wasn't in error.
- Put nodes names in alphabetic order in node table.
- If preempted job should have a grace time and preempt mode is not cancel but job is going to be canceled because it is interactive or other reason it now receives the grace time.
- BGQ - Modified documents to explain new plugin_flags needed in bg.properties in order for the runjob_mux to run correctly.
- BGQ - change linking from libslurm.o to libslurmhelper.la to avoid warning.

* Changes in SLURM 2.4.0.rc1

=====

- Improve task binding logic by making fuller use of HWLOC library, especially with respect to Opteron 6000 series processors. Work contributed by Komoto Masahiro.
- Add new configuration parameter PriorityFlags, based upon work by Carles Fenoy (Barcelona Supercomputer Center).
- Modify the step completion RPC between slurmd and slurmstepd in order to eliminate a possible deadlock. Based on work by Matthieu Hautreux, CEA.
- Change the owner of slurmd and slurmdbd log files to the appropriate user. Without this change the files will be created by and owned by the user starting the daemons (likely user root).
- Reorganize the slurmstepd logic in order to better support NFS and Kerberos credentials via the AUKS plugin. Work by Matthieu Hautreux, CEA.
- Fix bug in allocating GRES that are associated with specific CPUs. In some cases the code allocated first available GRES to job instead of allocating GRES accessible to the specific CPUs allocated to the job.
- spank: Add callbacks in slurmd: slurm_spank_slurmd_{init,exit} and job epilog/prolog: slurm_spank_job_{prolog,epilog}
- spank: Add slurm_option_getopt() function to api
- Change resolution of switch wait time from minutes to seconds.
- Added CrpCPUMins to the output of sshare -l for those using hard limit accounting. Work contributed by Mark Nelson.
- Added mpi/pmi2 plugin for complete support of pmi2 including acquiring additional resources for newly launched tasks. Contributed by Hongjia Cao, NUDT.
- BGQ - fixed issue where if a user asked for a specific node count and more tasks than possible without overcommit the request would be allowed on more nodes than requested.
- Add support for new SchedulerParameters of bf_max_job_user, maximum number of jobs to attempt backfilling per user. Work by Bjørn-Helge Mevik, University of Oslo.
- BLUEGENE - fixed issue where MaxNodes limit on a partition only limited larger than midplane jobs.
- Added cpu_run_min to the output of sshare --long. Work contributed by Mark Nelson.
- BGQ - allow regular users to resolve Rack-Midplane to XYZ coords.
- Add sinfo output format option of "%R" for partition name without "*" appended for default partition.
- Cray - Add support for zero compute node resource allocation to run batch script on front-end node with no ALPS reservation. Useful for pre- or post-processing.
- Support for cyclic distribution of cpus in task/cgroup plugin from Martin Perry, Bull.
- GrpMEM limit for QOSes and associations added Patch from Bjørn-Helge Mevik,


```

University of Oslo.
-- Various performance improvements for up to 500% higher throughput depending
upon configuration. Work supported by the Oak Ridge National Laboratory
Extreme Scale Systems Center.
-- Added jobacct_gather/cgroup plugin. It is not advised to use this in
production as it isn't currently complete and doesn't provide an equivalent
substitution for jobacct_gather/linux yet. Work by Martin Perry, Bull.

* Changes in SLURM 2.4.0.pre4
=====
-- Add logic to cache GPU file information (bitmap index mapping to device
file number) in the slurmd daemon and transfer that information to the
slurmstepd whenever a job step is initiated. This is needed to set the
appropriate CUDA_VISIBLE_DEVICES environment variable value when the
devices are not in strict numeric order (e.g. some GPUs are skipped).
Based upon work by Nicolas Bigaouette.
-- BGQ - Remove ability to make a sub-block with a geometry with one or more
of it's dimensions of length 3. There is a limitation in the IBM I/O
subsystem that is problematic with multiple sub-blocks with a dimension
of length 3, so we will disallow them to be able to be created. This
mean you if you ask the system for an allocation of 12 c-nodes you will
be given 16. If this is ever fix in BGQ you can remove this patch.
-- BLUEGENE - Better handling blocks that go into error state or deallocate
while jobs are running on them.
-- BGQ - fix for handling mix of steps running at same time some of which
are full allocation jobs, and others that are smaller.
-- BGQ - fix for core dump after running multiple sub-block jobs on static
blocks.
-- BGQ - fixed sync issue where if a job finishes in SLURM but not in mmcs
for a long time after the SLURM job has been flushed from the system
we don't have to worry about rebooting the block to sync the system.
-- BGQ - In scontrol/sview node counts are now displayed with
CnodeCount/CnodeErrCount so to point out there are cnodes in an error state
on the block. Draining the block and having it reboot when all jobs are
gone will clear up the cnodes in Software Failure.
-- Change default SchedulerParameters max_switch_wait field value from 60 to
300 seconds.
-- BGQ - catch errors from the kill option of the runjob client.
-- BLUEGENE - make it so the epilog runs until slurmd tells it the job is
gone. Previously it had a timelimit which has proven to not be the right
thing.
-- FRONTEND - fix issue where if a compute node was in a down state and
an admin updates the node to idle/resume the compute nodes will go
instantly to idle instead of idle* which means no response.
-- Fix regression in 2.4.0.pre3 where number of submitted jobs limit wasn't
being honored for QOS.
-- Cray - Enable logging of BASIL communications with environment variables.
Set XML_LOG to enable logging. Set XML_LOG_LOC to specify path to log file
or "SLURM" to write to SlurmctldLogFile or unset for "slurm_basil_xml.log".
Patch from Steve Tronfinoff, CSCS.
-- FRONTEND - if a front end unexpectedly reboots kill all jobs but don't
mark front end node down.
-- FRONTEND - don't down a front end node if you have an epilog error
-- BLUEGENE - if a job has an epilog error don't down the midplane it was
running on.

```

Appendix G. SLURM Release Information

- BGQ - added new DebugFlag (NoRealTime) for only printing debug from state change while the realtime server is running.
- Fix multi-cluster mode with sview starting on a non-bluegene cluster going to a bluegene cluster.
- BLUEGENE - ability to show Rack Midplane name of midplanes in sview and scontrol.

* Changes in SLURM 2.4.0.pre3

=====

- Let a job be submitted even if it exceeds a QOS limit. Job will be left in a pending state until the QOS limit or job parameters change. Patch by Phil Eckert, LLNL.
- Add sacct support for the option "--name". Work by Yuri D'Elia, Center for Biomedicine, EURAC Research, Italy.
- BGQ - handle preemption.
- Add an srun shepard process to cancel a job and/or step of the srun process is killed abnormally (e.g. SIGKILL).
- BGQ - handle deadlock issue when a nodeboard goes into an error state.
- BGQ - more thorough handling of blocks with multiple jobs running on them.
- Fix man2html process to compile in the build directory instead of the source dir.
- Behavior of srun --multi-prog modified so that any program arguments specified on the command line will be appended to the program arguments specified in the program configuration file.
- Add new command, sdiag, which reports a variety of job scheduling statistics. Based upon work by Alejandro Lucero Palau, BSC.
- BLUEGENE - Added DefaultConnType to the bluegene.conf file. This makes it so you can specify any connection type you would like (TORUS or MESH) as the default in dynamic mode. Previously it always defaulted to TORUS.
- Made squeue -n and -w options more consistent with salloc, sbatch, srun, and scancel. Patch by Don Lipari, LLNL.
- Have sacctmgr remove user records when no associations exist for that user.
- Several header file changes for clean build with NetBSD. Patches from Aleksej Saushev.
- Fix for possible deadlock in accounting logic: Avoid calling jobacct_gather_g_getinfo() until there is data to read from the socket.
- Fix race condition that could generate "job_cnt_comp underflow" errors on front-end architectures.
- BGQ - Fix issue where a system with missing cables could cause core dump.

* Changes in SLURM 2.4.0.pre2

=====

- CRAY - Add support for GPU memory allocation using SLURM GRES (Generic Resource) support. Work by Steve Trofinoff, CSCS.
- Add support for job allocations with multiple job constraint counts. For example: salloc -C "[rack1*2&rack2*4]" ... will allocate the job 2 nodes from rack1 and 4 nodes from rack2. Support for only a single constraint name been added to job step support.
- BGQ - Remove old method for marking cnodes down.
- BGQ - Remove BGP images from view in sview.
- BGQ - print out failed cnodes in scontrol show nodes.
- BGQ - Add srun option of "--runjob-opts" to pass options to the runjob command.
- FRONTEND - handle step launch failure better.
- BGQ - Added a mutex to protect the now changing ba_system pointers.

```
-- BGQ - added new functionality for sub-block allocations - no preemption
for this yet though.
-- Add --name option to squeue to filter output by job name. Patch from Yuri
D'Elia.
-- BGQ - Added linking to runjob client library which gives support to totalview
to use srun instead of runjob.
-- Add numeric range checks to scontrol update options. Patch from Phil
Eckert, LLNL.
-- Add ReconfigFlags configuration option to control actions of "scontrol
reconfig". Patch from Don Albert, Bull.
-- BGQ - handle reboots with multiple jobs running on a block.
-- BGQ - Add message handler thread to forward signals to runjob process.
```

* Changes in SLURM 2.4.0.pre1

=====

```
-- BGQ - use the ba_geo_tables to figure out the blocks instead of the old
algorithm. The improves timing in the worst cases and simplifies the code
greatly.
-- BLUEGENE - Change to output tools labels from BP to Midplane
(i.e. BP List -> MidplaneList).
-- BLUEGENE - read MPs and BPs from the bluegene.conf
-- Modify srun's SIGINT handling logic timer (two SIGINTs within one second) to
be based microsecond rather than second timer.
-- Modify advance reservation to accept multiple specific block sizes rather
than a single node count.
-- Permit administrator to change a job's QOS to any value without validating
the job's owner has permission to use that QOS. Based upon patch by Phil
Eckert (LLNL).
-- Add trigger flag for a permanent trigger. The trigger will NOT be purged
after an event occurs, but only when explicitly deleted.
-- Interpret a reservation with Nodes=ALL and a Partition specification as
reserving all nodes within the specified partition rather than all nodes
on the system. Based upon patch by Phil Eckert (LLNL).
-- Add the ability to reboot all compute nodes after they become idle. The
RebootProgram configuration parameter must be set and an authorized user
must execute the command "scontrol reboot_nodes". Patch from Andriy
Grytsenko (Massive Solutions Limited).
-- Modify slurmdbd.conf parsing to accept DebugLevel strings (quiet, fatal,
info, etc.) in addition to numeric values. The parsing of slurm.conf was
modified in the same fashion for SlurmctldDebug and SlurmdDebug values.
The output of sview and "scontrol show config" was also modified to report
those values as strings rather than numeric values.
-- Changed default value of StateSaveLocation configuration parameter from
/tmp to /var/spool.
-- Prevent associations from being deleted if it has any jobs in running,
pending or suspended state. Previous code prevented this only for running
jobs.
-- If a job can not run due to QOS or association limits, then do not cancel
the job, but leave it pending in a system held state (priority = 1). The
job will run when its limits or the QOS/association limits change. Based
upon a patch by Phil Ekcerc (LLNL).
-- BGQ - Added logic to keep track of cnodes in an error state inside of a
booted block.
-- Added the ability to update a node's NodeAddr and NodeHostName with
scontrol. Also enable setting a node's state to "future" using scontrol.
```

Appendix G. SLURM Release Information

- Add a node state flag of CLOUD and save/restore NodeAddr and NodeHostName information for nodes with a flag of CLOUD.
- Cray: Add support for job reservations with node IDs that are not in numeric order. Fix for Bugzilla #5.
- BGQ - Fix issue with smap -R
- Fix association limit support for jobs queued for multiple partitions.
- BLUEGENE - fix issue for sub-midplane systems to create a full system block correctly.
- BLUEGENE - Added option to the bluegene.conf to tell you are running on a sub midplane system.
- Added the UserID of the user issuing the RPC to the job_submit/lua functions.
- Fixed issue where if a job ended with ESLURMD_UID_NOT_FOUND and ESLURMD_GID_NOT_FOUND where slurm would be a little over zealous in treating missing a GID or UID as a fatal error.
- If job time limit exceeds partition maximum, but job's minimum time limit does not, set job's time limit to partition maximum at allocation time.

* Changes in SLURM 2.3.6

=====

- Fix DefMemPerCPU for partition definitions.
- Fix to create a reservation with licenses and no nodes.
- Fix issue with assoc_mgr if a bad state file is given and the database isn't up at the time the slurmctld starts, not running the priority/multifactor plugin, and then the database is started up later.
- Gres: If a gres has a count of one and an associated file then when doing a reconfiguration, the node's bitmap was not cleared resulting in an underflow upon job termination or removal from scheduling matrix by the backfill scheduler.
- Fix race condition in job dependency logic which can result in invalid memory reference.

* Changes in SLURM 2.3.5

=====

- Improve support for overlapping advanced reservations. Patch from Bill Brophy, Bull.
- Modify Makefiles for support of Debian hardening flags. Patch from Simon Ruderich.
- CRAY: Fix support for configuration with SlurmdTimeout=0 (never mark node that is DOWN in ALPS as DOWN in SLURM).
- Fixed the setting of SLURM_SUBMIT_DIR for jobs submitted by Moab (BZ#1467). Patch by Don Lipari, LLNL.
- Correction to init.d/slurmdbd exit code for status option. Patch by Bill Brophy, Bull.
- When the optional max_time is not specified for --switches=count, the site max (SchedulerParameters=max_switch_wait=seconds) is used for the job. Based on patch from Rod Schultz.
- Fix bug in select/cons_res plugin when used with topology/tree and a node range count in job allocation request.
- Fixed moab_2_slurmdb.pl script to correctly work for end records.
- Add support for new SchedulerParameters of max_depend_depth defining the maximum number of jobs to test for circular dependencies (i.e. job A waits for job B to start and job B waits for job A to start). Default value is 10 jobs.
- Fix potential race condition if MinJobAge is very low (i.e. 1) and using

```

slurmdbd accounting and running large amounts of jobs (>50 sec). Job
information could be corrupted before it had a chance to reach the DBD.
-- Fix state restore of job limit set from admin value for min_cpus.
-- Fix clearing of limit values if an admin removes the limit for max cpus
and time limit where it was previously set by an admin.
-- Fix issue where log message is more than 256 chars and then has a format.
-- Fix sched/wiki2 to support job account name, gres, partition name, wckey,
or working directory that contains "#" (a job record separator). Also fix
for wckey or working directory that contains a double quote '\ "'.
-- CRAY - fix for handling memory requests from user for an allocation.
-- Add support for switches parameter to the job_submit/lua plugin. Work by
Par Andersson, NSC.
-- Fix to job preemption logic to preempt multiple jobs at the same time.
-- Fix minor issue where uid and gid were switched in sview for submitting
batch jobs.
-- Fix possible illegal memory reference in slurmctld for job step with
relative option. Work by Matthieu Hautreux (CEA).
-- Reset priority of system held jobs when dependency is satisfied. Work by
Don Lipari, LLNL.

```

* Changes in SLURM 2.3.4

=====

```

-- Set DEFAULT flag in partition structure when slurmctld reads the
configuration file. Patch from Rémi Palancher.
-- Fix for possible deadlock in accounting logic: Avoid calling
jobacct_gather_g_getinfo() until there is data to read from the socket.
-- Fix typo in accounting when using reservations. Patch from Alejandro
Lucero Palau.
-- Fix to the multifactor priority plugin to calculate effective usage earlier
to give a correct priority on the first decay cycle after a restart of the
slurmctld. Patch from Martin Perry, Bull.
-- Permit user root to run a job step for any job as any user. Patch from
Didier Gazen, Laboratoire d'Aerologie.
-- BLUEGENE - fix for not allowing jobs if all midplanes are drained and all
blocks are in an error state.
-- Avoid slurmctld abort due to bad pointer when setting an advanced
reservation MAINT flag if it contains no nodes (only licenses).
-- Fix bug when requeued batch job is scheduled to run on a different node
zero, but attempts job launch on old node zero.
-- Fix bug in step task distribution when nodes are not configured in numeric
order. Patch from Hongjia Cao, NUDT.
-- Fix for srun allocating running within existing allocation with --exclude
option and --nnodes count small enough to remove more nodes. Patch from
Phil Eckert, LLNL.
-- Work around to handle certain combinations of glibc/kernel
(i.e. glibc-2.14/Linux-3.1) to correctly open the pty of the slurmstepd
as the job user. Patch from Mark Grondona, LLNL.
-- Modify linking to include "-ldl" only when needed. Patch from Aleksey
Saushev.
-- Fix smap regression to display nodes that are drained or down correctly.
-- Several bug fixes and performance improvements with related to batch
scripts containing very large numbers of arguments. Patches from Par
Andersson, NSC.
-- Fixed extremely hard to reproduce threading issue in assoc_mgr.
-- Correct "scontrol show daemons" output if there is more than one

```

Appendix G. SLURM Release Information

```
ControlMachine configured.
-- Add node read lock where needed in slurmctld/agent code.
-- Added test for LUA library named "liblua5.1.so.0" in addition to
  "liblua5.1.so" as needed by Debian. Patch by Remi Palancher.
-- Added partition default_time field to job_submit LUA plugin. Patch by
  Remi Palancher.
-- Fix bug in cray/srun wrapper stdin/out/err file handling.
-- In cray/srun wrapper, only include aprun "-q" option when srun "--quiet"
  option is used.
-- BLUEGENE - fix issue where if a small block was in error it could hold up
  the queue when trying to place a larger than midplane job.
-- CRAY - ignore all interactive nodes and jobs on interactive nodes.
-- Add new job state reason of "FrontEndDown" which applies only to Cray and
  IBM BlueGene systems.
-- Cray - Enable configure option of "--enable-salloc-background" to permit
  the srun and salloc commands to be executed in the background. This does
  NOT remove the ALPS limitation that only one job reservation can be created
  for each Linux session ID.
-- Cray - For srun wrapper when creating a job allocation, set the default job
  name to the executable file's name.
-- Add support for Cray ALPS 5.0.0
-- FRONTEND - if a front end unexpectedly reboots kill all jobs but don't
  mark front end node down.
-- FRONTEND - don't down a front end node if you have an epilog error.
-- Cray - fix for if a frontend slurmd was started after the slurmctld had
  already pinged it on startup the unresponding flag would be removed from
  the frontend node.
-- Cray - Fix issue on smap not displaying grid correctly.
-- Fixed minor memory leak in svview.

* Changes in SLURM 2.3.3
=====
-- Fix task/cgroup plugin error when used with GRES. Patch by Alexander
  Bersenev (Institute of Mathematics and Mechanics, Russia).
-- Permit pending job exceeding a partition limit to run if its QOS flag is
  modified to permit the partition limit to be exceeded. Patch from Bill
  Brophy, Bull.
-- BLUEGENE - Fixed preemption issue.
-- sacct search for jobs using filtering was ignoring wckey filter.
-- Fixed issue with QOS preemption when adding new QOS.
-- Fixed issue with comment field being used in a job finishing before it
  starts in accounting.
-- Add slashes in front of derived exit code when modifying a job.
-- Handle numeric suffix of "T" for terabyte units. Patch from John Thiltges,
  University of Nebraska-Lincoln.
-- Prevent resetting a held job's priority when updating other job parameters.
  Patch from Alejandro Lucero Palau, BSC.
-- Improve logic to import a user's environment. Needed with --get-user-env
  option used with Moab. Patch from Mark Grondona, LLNL.
-- Fix bug in svview layout if node count less than configured grid_x_width.
-- Modify PAM module to prefer to use SLURM library with same major release
  number that it was built with.
-- Permit gres count configuration of zero.
-- Fix race condition where sbcast command can result in deadlock of slurmd
  daemon. Patch by Don Albert, Bull.
```

```
-- Fix bug in srun --multi-prog configuration file to avoid printing duplicate
record error when "*" is used at the end of the file for the task ID.
-- Let operators see reservation data even if "PrivateData=reservations" flag
is set in slurm.conf. Patch from Don Albert, Bull.
-- Added new sbatch option "--export-file" as needed for latest version of
Moab. Patch from Phil Eckert, LLNL.
-- Fix for sacct printing CPUTime(RAW) where the the is greater than a 32 bit
number.
-- Fix bug in --switch option with topology resulting in bad switch count use.
Patch from Alejandro Lucero Palau (Barcelona Supercomputer Center).
-- Fix PrivateFlags bug when using Priority Multifactor plugin. If using sprio
all jobs would be returned even if the flag was set.
Patch from Bill Brophy, Bull.
-- Fix for possible invalid memory reference in slurmctld in job dependency
logic. Patch from Carles Fenoy (Barcelona Supercomputer Center).
```

* Changes in SLURM 2.3.2

=====

```
-- Add configure option of "--without-rpath" which builds SLURM tools without
the rpath option, which will work if Munge and BlueGene libraries are in
the default library search path and make system updates easier.
-- Fixed issue where if a job ended with ESLURMD_UID_NOT_FOUND and
ESLURMD_GID_NOT_FOUND where slurm would be a little over zealous
in treating missing a GID or UID as a fatal error.
-- Backfill scheduling - Add SchedulerParameters configuration parameter of
"bf_res" to control the resolution in the backfill scheduler's data about
when jobs begin and end. Default value is 60 seconds (used to be 1 second).
-- Cray - Remove the "family" specification from the GPU reservation request.
-- Updated set_oomadj.c, replacing deprecated oom_adj reference with
oom_score_adj
-- Fix resource allocation bug, generic resources allocation was ignoring the
job's ntasks_per_node and cpus_per_task parameters. Patch from Carles
Fenoy, BSC.
-- Avoid orphan job step if slurmctld is down when a job step completes.
-- Fix Lua link order, patch from Pär Andersson, NSC.
-- Set SLURM_CPUS_PER_TASK=1 when user specifies --cpus-per-task=1.
-- Fix for fatal error managing GRES. Patch by Carles Fenoy, BSC.
-- Fixed race condition when using the DBD in accounting where if a job
wasn't started at the time the eligible message was sent but started
before the db_index was returned information like start time would be lost.
-- Fix issue in accounting where normalized shares could be updated
incorrectly when getting fairshare from the parent.
-- Fixed if not enforcing associations but want QOS support for a default
qos on the cluster to fill that in correctly.
-- Fix in select/cons_res for "fatal: cons_res: sync loop not progressing"
with some configurations and job option combinations.
-- BLUEGNE - Fixed issue with handling HTC modes and rebooting.
```

* Changes in SLURM 2.3.1

=====

```
-- Do not remove the backup slurmctld's pid file when it assumes control, only
when it actually shuts down. Patch from Andriy Grytsenko (Massive Solutions
Limited).
-- Avoid clearing a job's reason from JobHeldAdmin or JobHeldUser when it is
otherwise updated using scontrol or sview commands. Patch based upon work
```

Appendix G. SLURM Release Information

by Phil Eckert (LLNL).

- BLUEGENE - Fix for if changing the defined blocks in the bluegene.conf and jobs happen to be running on blocks not in the new config.
- Many cosmetic modifications to eliminate warning message from GCC version 4.6 compiler.
- Fix for sview reservation tab when finding correct reservation.
- Fix for handling QOS limits per user on a reconfig of the slurmd.
- Do not treat the absence of a gres.conf file as a fatal error on systems configured with GRES, but set GRES counts to zero.
- BLUEGENE - Update correctly the state in the reason of a block if an admin sets the state to error.
- BLUEGENE - handle reason of blocks in error more correctly between restarts of the slurmd.
- BLUEGENE - Fix minor potential memory leak when setting block error reason.
- BLUEGENE - Fix if running in Static/Overlap mode and full system block is in an error state, won't deny jobs.
- Fix for accounting where your cluster isn't numbered in counting order (i.e. 1-9,0 instead of 0-9). The bug would cause 'sacct -N nodename' to not give correct results on these systems.
- Fix to GRES allocation logic when resources are associated with specific CPUs on a node. Patch from Steve Trofinoff, CSCS.
- Fix bugs in sched/backfill with respect to QOS reservation support and job time limits. Patch from Alejandro Lucero Palau (Barcelona Supercomputer Center).
- BGQ - fix to set up corner correctly for sub block jobs.
- Major re-write of the CPU Management User and Administrator Guide (web page) by Martin Perry, Bull.
- BLUEGENE - If removing blocks from system that once existed cleanup of old block happens correctly now.
- Prevent slurmd crashing with configuration of MaxMemPerCPU=0.
- Prevent job hold by operator or account coordinator of his own job from being an Administrator Hold rather than User Hold by default.
- Cray - Fix for srun.pl parsing to avoid adding spaces between option and argument (e.g. "-N2" parsed properly without changing to "-N 2").
- Major updates to cgroup support by Mark Grondona (LLNL) and Matthieu Hautreux (CEA) and Sam Lang. Fixes timing problems with respect to the task_epilog. Allows cgroup mount point to be configurable. Added new configuration parameters MaxRAMPercent and MaxSwapPercent. Allow cgroup configuration parameters that are percentages to be floating point.
- Fixed issue where sview wasn't displaying correct nice value for jobs.
- Fixed issue where sview wasn't displaying correct min memory per node/cpu value for jobs.
- Disable some SelectTypeParameters for select/linear that aren't compatible.
- Move slurmd_select_init to proper place to avoid loading multiple select plugins in the slurmd.
- BGQ - Include runjob_plugin.so in the bluegene rpm.
- Report correct job "Reason" if needed nodes are DOWN, DRAINED, or NOT_RESPONDING, "Resources" rather than "PartitionNodeLimit".
- BLUEGENE - Fixed issues with running on a sub-midplane system.
- Added some missing calls to allow older versions of SLURM to talk to newer.
- BGQ - allow steps to be ran.
- Do not attempt to run HeathCheckProgram on powered down nodes. Patch from Ramiro Alba, Centre Tecnològic de Transferència de Calor, Spain.

* Changes in SLURM 2.3.0-2


```

=====
-- Fix for memory issue inside svview.
-- Fix issue where if a job was pending and the slurmctld was restarted a
   variable wasn't initialized in the job structure making it so that job
   wouldn't run.

* Changes in SLURM 2.3.0
=====
-- BLUEGENE - make sure we only set the jobinfo_select start_loc on a job
   when we are on a small block, not a regular one.
-- BGQ - fix issue where not copying the correct amount of memory.
-- BLUEGENE - fix clean start if jobs were running when the slurmctld was
   shutdown and then the system size changed. This would probably only happen
   if you were emulating a system.
-- Fix svview for calling a cray system from a non-cray system to get the
   correct geometry of the system.
-- BLUEGENE - fix to correctly import pervious version of block state file.
-- BLUEGENE - handle loading better when doing a clean start with static
   blocks.
-- Add sinfo format and sort option "%n" for NodeHostName and "%o" for
   NodeAddr.
-- If a job is deferred due to partition limits, then re-test those limits
   after a partition is modified. Patch from Don Lipari.
-- Fix bug which would crash slurmctld if job's owner (not root) tries to clear
   a job's licenses by setting value to "".
-- Cosmetic fix for printing out debug info in the priority plugin.
-- In svview when switching from a bluegene machine to a regular linux cluster
   and vice versa the node->base partition lists will be displayed if setup
   in your .slurm/svviewrc file.
-- BLUEGENE - Fix for creating full system static block on a BGQ system.
-- BLUEGENE - Fix deadlock issue if toggling between Dynamic and Static block
   allocation with jobs running on blocks that don't exist in the static
   setup.
-- BLUEGENE - Modify code to only give HTC states to BGP systems and not
   allow them on Q systems.
-- BLUEGENE - Make it possible for an admin to define multiple dimension
   conn_types in a block definition.
-- BGQ - Alter tools to output multiple dimensional conn_type.

* Changes in SLURM 2.3.0.rc2
=====
-- With sched/wiki or sched/wiki2 (Maui or Moab scheduler), insure that a
   requeued job's priority is reset to zero.
-- BLUEGENE - fix to run steps correctly in a BGL/P emulated system.
-- Fixed issue where if there was a network issue between the slurmctld and
   the DBD where both remained up but were disconnected the slurmctld would
   get registered again with the DBD.
-- Fixed issue where if the DBD connection from the ctld goes away because of
   a POLLERR the dbd_fail callback is called.
-- BLUEGENE - Fix to smap command-line mode display.
-- Change in GRES behavior for job steps: A job step's default generic
   resource allocation will be set to that of the job. If a job step's --gres
   value is set to "none" then none of the generic resources which have been
   allocated to the job will be allocated to the job step.
-- Add srun environment value of SLURM_STEP_GRES to set default --gres value

```

Appendix G. SLURM Release Information

```
for a job step.
-- Require SchedulerTimeSlice configuration parameter to be at least 5 seconds
to avoid thrashing slurmd daemon.
-- Cray - Fix to make nodes state in accounting consistent with state set by
ALPS.
-- Cray - A node DOWN to ALPS will be marked DOWN to SLURM only after reaching
SlurmdTimeout. In the interim, the node state will be NO_RESPOND. This
change makes behavior makes SLURM handling of the node DOWN state more
consistent with ALPS. This change effects only Cray systems.
-- Cray - Fix to work with 4.0.* instead of just 4.0.0
-- Cray - Modify srun/aprun wrapper to map --exclusive to -F exclusive and
--share to -F share. Note this does not consider the partition's Shared
configuration, so it is an imperfect mapping of options.
-- BLUEGENE - Added notice in the print config to tell if you are emulated
or not.
-- BLUEGENE - Fix job step scalability issue with large task count.
-- BGQ - Improved c-node selection when asked for a sub-block job that
cannot fit into the available shape.
-- BLUEGENE - Modify "scontrol show step" to show I/O nodes (BGL and BGP) or
c-nodes (BGQ) allocated to each step. Change field name from "Nodes=" to
"BP_List=".
-- Code cleanup on step request to get the correct select_jobinfo.
-- Memory leak fixed for rolling up accounting with down clusters.
-- BGQ - fix issue where if first job step is the entire block and then the
next parallel step is ran on a sub block, SLURM won't over subscribe cnodes.
-- Treat duplicate switch name in topology.conf as fatal error. Patch from Rod
Schultz, Bull
-- Minor update to documentation describing the AllowGroups option for a
partition in the slurm.conf.
-- Fix problem with _job_create() when not using qos's. It makes
_job_create() consistent with similar logic in select_nodes().
-- GrpCPURunMins in a QOS flushed out.
-- Fix for squeue -t "CONFIGURING" to actually work.
-- CRAY - Add cray.conf parameter of SyncTimeout, maximum time to defer job
scheduling if SLURM node or job state are out of synchronization with ALPS.
-- If salloc was run as interactive, with job control, reset the foreground
process group of the terminal to the process group of the parent pid before
exiting. Patch from Don Albert, Bull.
-- BGQ - set up the corner of a sub block correctly based on a relative
position in the block instead of absolute.
-- BGQ - make sure the recently added select_jobinfo of a step launch request
isn't sent to the slurmd where environment variables would be overwritten
incorrectly.

* Changes in SLURM 2.3.0.rc1
=====
-- NOTE THERE HAVE BEEN NEW FIELDS ADDED TO THE JOB AND PARTITION STATE SAVE
FILES AND RPCS. PENDING AND RUNNING JOBS WILL BE LOST WHEN UPGRADING FROM
EARLIER VERSION 2.3 PRE-RELEASES AND RPCS WILL NOT WORK WITH EARLIER
VERSIONS.
-- select/cray: Add support for Accelerator information including model and
memory options.
-- Cray systems: Add support to suspend/resume salloc command to insure that
aprun does not get initiated when the job is suspended. Processes suspended
and resumed are determined by using process group ID and parent process ID,
```

- so some processes may be missed. Since salloc runs as a normal user, it's ability to identify processes associated with a job is limited.
- Cray systems: Modify smap and svview to display all nodes even if multiple nodes exist at each coordinate.
 - Improve efficiency of select/linear plugin with topology/tree plugin configured, Patch by Andriy Grytsenko (Massive Solutions Limited).
 - For front-end architectures on which job steps are run (emulated Cray and BlueGene systems only), fix bug that would free memory still in use.
 - Add squeue support to display a job's license information. Patch by Andy Roosen (University of Delaware).
 - Add flag to the select APIs for job suspend/resume indicating if the action is for gang scheduling or an explicit job suspend/resume by the user. Only an explicit job suspend/resume will reset the job's priority and make resources exclusively held by the job available to other jobs.
 - Fix possible invalid memory reference in sched/backfill. Patch by Andriy Grytsenko (Massive Solutions Limited).
 - Add select_jobinfo to the task launch RPC. Based upon patch by Andriy Grytsenko (Massive Solutions Limited).
 - Add DefMemPerCPU/Node and MaxMemPerCPU/Node to partition configuration. This improves flexibility when gang scheduling only specific partitions.
 - Added new enums to print out when a job is held by a QOS instead of an association limit.
 - Enhancements to sched/backfill performance with select/cons_res plugin. Patch from Bjørn-Helge Mevik, University of Oslo.
 - Correct job run time reported by smap for suspended jobs.
 - Improve job preemption logic to avoid preempting more jobs than needed.
 - Add contribs/arrayrun tool providing support for job arrays. Contributed by Bjørn-Helge Mevik, University of Oslo. NOTE: Not currently packaged as RPM and manual file editing is required.
 - When suspending a job, wait 2 seconds instead of 1 second between sending SIGTSTP and SIGSTOP. Some MPI implementation were not stopping within the 1 second delay.
 - Add support for managing devices based upon Linux cgroup container. Based upon patch by Yiannis Georgiou, Bull.
 - Fix memory buffering bug if a AllowGroups parameter of a partition has 100 or more users. Patch by Andriy Grytsenko (Massive Solutions Limited).
 - Fix bug in generic resource tracking of gres associated with specific CPUs. Resources were being over-allocated.
 - On systems with front-end nodes (IBM BlueGene and Cray) limit batch jobs to only one CPU of these shared resources.
 - Set SLURM_MEM_PER_CPU or SLURM_MEM_PER_NODE environment variables for both interactive (salloc) and batch jobs if the job has a memory limit. For Cray systems also set CRAY_AUTO_APRUN_OPTIONS environment variable with the memory limit.
 - Fix bug in select/cons_res task distribution logic when tasks-per-node=0. Patch from Rod Schultz, Bull.
 - Restore node configuration information (CPUs, memory, etc.) for powered down when slurmctld daemon restarts rather than waiting for the node to be restored to service and getting the information from the node (NOTE: Only relevant if FastSchedule=0).
 - For Cray systems with the srun2aprun wrapper, rebuild the srun man page identifying the srun options which are valid on that system.
 - BlueGene: Permit users to specify a separate connection type for each dimension (e.g. "--conn-type=torus,mesh,torus").
 - Add the ability for a user to limit the number of leaf switches in a job's

Appendix G. SLURM Release Information

allocation using the `--switch` option of `salloc`, `sbatch` and `srun`. There is also a new `SchedulerParameters` value of `max_switch_wait`, which a SLURM administrator can use to set a maximum job delay and prevent a user job from blocking lower priority jobs for too long. Based on work by Rod Schultz, Bull.

* Changes in SLURM 2.3.0.pre6

=====

- NOTE: THERE HAS BEEN A NEW FIELD ADDED TO THE CONFIGURATION RESPONSE RPC AS SHOWN BY "SCONTROL SHOW CONFIG". THIS FUNCTION WILL ONLY WORK WHEN THE SERVER AND CLIENT ARE BOTH RUNNING SLURM VERSION 2.3.0.pre6
- Modify job expansion logic to support licenses, generic resources, and currently running job steps.
- Added an `rpath` if using the `--with-munge` option of `configure`.
- Add support for multiple sets of `DEFAULT` node, partition, and frontend specifications in `slurm.conf` so that default values can be changed multiple times as the configuration file is read.
- BLUEGENE - Improved logic to place small blocks in free space before freeing larger blocks.
- Add optional argument to `srun`'s `--kill-on-bad-exit` so that user can set its value to zero and override a SLURM configuration parameter of `KillOnBadExit`.
- Fix bug in `GraceTime` support for preempted jobs that prevented proper operation when more than one job was being preempted. Based on patch from Bill Brophy, Bull.
- Fix for running `svview` from a non-bluegene cluster to a bluegene cluster. Regression from pre5.
- If job's `TMPDIR` environment is not set or is not usable, reset to `/tmp`. Patch from Andriy Grytsenko (Massive Solutions Limited).
- Remove logic for defunct RPC: `DBD_GET_JOBS`.
- Propagate `DebugFlag` changes by `scontrol` to the plugins.
- Improve accuracy of `REQUEST_JOB_WILL_RUN` start time with respect to higher priority pending jobs.
- Add `-R/--reservation` option to `squeue` command as a job filter.
- Add `scancel` support for `--clusters` option.
- Note that `scontrol` and `sprio` can only support a single cluster at one time.
- Add support to `salloc` for a new environment variable `SALLOC_KILL_CMD`.
- Add `scontrol` ability to increment or decrement a job or step time limit.
- Add support for `SLURM_TIME_FORMAT` environment variable to control time stamp output format. Work by Gerrit Renker, CSCS.
- Fix error handling in `mvapich` plugin that could cause `srun` to enter an infinite loop under rare circumstances.
- Add support for multiple task plugins. Patch from Andriy Grytsenko (Massive Solutions Limited).
- Addition of per-user node/cpu limits for QOS's. Patch from Aaron Knister, UMBC.
- Fix logic for multiple job resize operations.
- BLUEGENE - many fixes to make things work correctly on a L/P system.
- Fix bug in layout of job step with `--nodelist` option plus node count. Old code could allocate too few nodes.

* Changes in SLURM 2.3.0.pre5

=====

- NOTE: THERE HAS BEEN A NEW FIELD ADDED TO THE JOB STATE FILE. UPGRADES FROM VERSION 2.3.0-PRE4 WILL RESULT IN LOST JOBS UNLESS THE "orig_dependency"

FIELD IS REMOVED FROM JOB STATE SAVE/RESTORE LOGIC. ON CRAY SYSTEMS A NEW "confirm_cookie" FIELD WAS ADDED AND HAS THE SAME EFFECT OF DISABLING JOB STATE RESTORE.

- BLUEGENE - Improve speed of start up when removing blocks at the beginning.
- Correct init.d/slurm status to have non-zero exit code if ANY Slurm daemon that should be running on the node is not running. Patch from Rod Schulz, Bull.
- Improve accuracy of response to "srun --test-only jobid=#".
- Fix bug in front-end configurations which reports job_cnt_comp underflow errors after slurmctld restarts.
- Eliminate "error from _trigger_slurmctld_event in backup.c" due to lack of event triggers.
- Fix logic in BackupController to properly recover front-end node state and avoid purging active jobs.
- Added man pages to html pages and the new cpu_management.html page. Submitted by Martin Perry / Rod Schultz, Bull.
- Job dependency information will only show the currently active dependencies rather than the original dependencies. From Dan Rusak, Bull.
- Add RPCs to get the SPANK environment variables from the slurmctld daemon. Patch from Andrej N. Gritsenko.
- Updated plugins/task/cgroup/task_cgroup_cpuset.c to support newer HWLOC_API_VERSION.
- Do not build select/bluegene plugin if C++ compiler is not installed.
- Add new configure option --with-srun2aprun to build an srun command which is a wrapper over Cray's aprun command and supports many srun options. Without this option, the srun command will advise the user to use the aprun command.
- Change container ID supported by proctrack plugin from 32-bit to 64-bit.
- Added contribs/cray/libalps_test_programs.tar.gz with tools to validate SLURM's logic used to support Cray systems.
- Create RPM for srun command that is a wrapper for the Cray/ALPS aprun command. Dependent upon .rpmmacros parameter of "%_with_srun2aprun".
- Add configuration parameter MaxStepCount to limit effect of bad batch scripts.
- Moving to github
- Fix for handling a 2.3 system talking to a 2.2 slurmctld.
- Add contribs/luajob_submit/license.lua script. Update job_submit and Lua related documentation.
- Test if _make_batch_script() is called with a NULL script.
- Increase hostlist support from 24k to 64k nodes.
- Renamed the Accounting Storage database's "DerivedExitString" job field to "Comment". Provided backward compatible support for "DerivedExitString" in the sacctmgr tool.
- Added the ability to save the job's comment field to the Accounting Storage db (to the formerly named, "DerivedExitString" job field). This behavior is enabled by a new slurm.conf parameter: AccountingStoreJobComment.
- Test if _make_batch_script() is called with a NULL script.
- Increase hostlist support from 24k to 64k nodes.
- Fix srun to handle signals correctly when waiting for a step creation.
- Preserve the last job ID across slurmctld daemon restarts even if the job state file can not be fully recovered.
- Made the hostlist functions be able to arbitrarily handle any size dimension no matter what the size of the cluster is in dimensions.

Appendix G. SLURM Release Information

* Changes in SLURM 2.3.0.pre4

=====

- Add GraceTime to Partition and QOS data structures. Preempted jobs will be given this time interval before termination. Work by Bill Brophy, Bull.
- Add the ability for scontrol and svview to modify slurmctld DebugFlags values.
- Various Cray-specific patches:
 - Fix a bug in distinguishing XT from XE.
 - Avoids problems with empty nodenames on Cray.
 - Check whether ALPS is hanging on to nodes, which happens if ALPS has not yet cleaned up the node partition.
 - Stops select/cray from clobbering node_ptr->reason.
 - Perform 'safe' release of ALPS reservations using inventory and apkill.
 - Compile-time sanity check for the apbasil and apkill files.
 - Changes error handling in do_basil_release() (called by select_g_job_fini()).
 - Warn that salloc --no-shell option is not supported on Cray systems.
- Add a reservation flag of "License_Only". If set, then jobs using the reservation may use the licenses associated with it plus any compute nodes. Otherwise the job is limited to the compute nodes associated with the reservation.
- Change slurm.conf node configuration parameter from "Procs" to "CPUs". Both parameters will be supported for now.
- BLUEGENE - fix for when user requests only midplane names with no count at job submission time to process the node count correctly.
- Fix job step resource allocation problem when both node and tasks counts are specified. New logic selects nodes with larger CPU counts as needed.
- BGQ - make it so srun wraps runjob (still under construction, but works for most cases)
- Permit a job's QOS and Comment field to both change in a single RPC. This was previously disabled since Moab stored the QOS within the Comment field.
- Add support for jobs to expand in size. Submit additional batch job with the option "--dependency=expand:<jobid>". See web page "faq.html#job_size" for details. Restrictions to be removed in the future.
- Added --with-alps-emulation to configure, and also an optional cray.conf to setup alps location and database information.
- Modify PMI data types from 16-bits to 32-bits in order to support MPICH2 jobs with more than 65,536 tasks. Patch from Hongjia Cao, NUDT.
- Set slurmd's soft process CPU limit equal to it's hard limit and notify the user if the limit is not infinite.
- Added proctrack/cgroup and task/cgroup plugins from Matthieu Hautreux, CEA.
- Fix slurmctld restart logic that could leave nodes in UNKNOWN state for a longer time than necessary after restart.

* Changes in SLURM 2.3.0.pre3

=====

- BGQ - Appears to work correctly in emulation mode, no sub blocks just yet.
- Minor typos fixed
- Various bug fixes for Cray systems.
- Fix bug that when setting a compute node to idle state, it was failing to set the systems up_node_bitmap.
- BLUEGENE - code reorder
- BLUEGENE - Now only one select plugin for all Bluegene systems.
- Modify srun to set the SLURM_JOB_NAME environment variable when srun is used to create a new job allocation. Not set when srun is used to create a

```

    job step within an existing job allocation.
-- Modify init.d/slurm script to start multiple slurmd daemons per compute
   node if so configured. Patch from Matthieu Hautreux, CEA.
-- Change license data structure counters from uint16_t to uint32_t to support
   larger license counts.

* Changes in SLURM 2.3.0.pre2
=====
-- Log a job's requeue or cancellation due to preemption to that job's stderr:
   "*** JOB 65547 CANCELLED AT 2011-01-21T12:59:33 DUE TO PREEMPTION ***".
-- Added new job termination state of JOB_PREEMPTED, "PR" or "PREEMPTED" to
   indicate job termination was due to preemption.
-- Optimize advanced reservations resource selection for computer topology.
   The logic has been added to select/linear and select/cons_res, but will
   not be enabled until the other select plugins are modified.
-- Remove checkpoint/xlch plugin.
-- Disable deletion of partitions that have unfinished jobs (pending,
   running or suspended states). Patch from Martin Perry, BULL.
-- In svview, disable the sorting of node records by name at startup for
   clusters over 1000 nodes. Users can enable this by selecting the "Name"
   tab. This change dramatically improves scalability of svview.
-- Report error when trying to change a node's state from scontrol for Cray
   systems.
-- Do not attempt to read the batch script for non-batch jobs. This patch
   eliminates some inappropriate error messages.
-- Preserve NodeHostName when reordering nodes due to system topology.
-- On Cray/ALPS systems do node inventory before scheduling jobs.
-- Disable some salloc options on Cray systems.
-- Disable scontrol's wait_job command on Cray systems.
-- Disable sruncmd command on native Cray/ALPS systems.
-- Updated configure option "--enable-cray-emulation" (still under
   development) to emulate a cray XT/XE system, and auto-detect a real
   Cray XT/XE systems (removed no longer needed --enable-cray configure
   option). Building on native Cray systems requires the
   cray-MySQL-devel-enterprise rpm and expat XML parser library/headers.

* Changes in SLURM 2.3.0.pre1
=====
-- Added that when a slurmd closes the connection to the database it's
   registered host and port are removed.
-- Added flag to slurmd.conf TrackSlurmdDown where if set will mark idle
   resources as down on a cluster when a slurmd disconnects or is no
   longer reachable.
-- Added support for more than one front-end node to run slurmd on
   architectures where the slurmd does not execute on the compute nodes
   (e.g. BlueGene). New configuration parameters FrontendNode and FrontendAddr
   added. See "man slurm.conf" for more information.
-- With the scontrol show job command when using the --details option, show
   a batch job's script.
-- Add ability to create reservations or partitions and submit batch jobs
   using svview. Also add the ability to delete reservations and partitions.
-- Added new configuration parameter MaxJobId. Once reached, restart job ID
   values at FirstJobId.
-- When restarting slurmd with priority/basic, increment all job priorities
   so the highest job priority becomes TOP_PRIORITY.

```

Appendix G. SLURM Release Information

* Changes in SLURM 2.2.8

=====

- Prevent background salloc disconnecting terminal at termination. Patch by Don Albert, Bull.
- Fixed issue where preempt mode is skipped when creating a QOS. Patch by Bill Brophy, Bull.
- Fixed documentation (html) for PriorityUsageResetPeriod to match that in the man pages. Patch by Nancy Kritkauskay, Bull.

* Changes in SLURM 2.2.7

=====

- Eliminate zombie process created if salloc exits with stopped child process. Patch from Gerrit Renker, CSCS.
- With default configuration on non-Cray systems, enable salloc to be spawned as a background process. Based upon work by Don Albert (Bull) and Gerrit Renker (CSCS).
- Fixed Regression from 2.2.4 in accounting where an inherited limit would not be set correctly in the added child association.
- Fixed issue with accounting when asking for jobs with a hostlist.
- Avoid clearing a node's Arch, OS, BootTime and SlurmdStartTime when "scontrol reconfig" is run. Patch from Martin Perry, Bull.

* Changes in SLURM 2.2.6

=====

- Fix displaying of account coordinators with sacctmgr. Possibility to show deleted accounts. Only a cosmetic issue, since the accounts are already deleted, and have no associations.
- Prevent opaque ncurses WINDOW struct on OS X 10.6.
- Fix issue with accounting when using PrivateData=jobs... users would not be able to view their own jobs unless they were admin or coordinators which is obviously wrong.
- Fix bug in node stat if slurmd is restarted while nodes are in the process of being powered up. Patch from Andriy Grytsenko.
- Change maximum batch script size from 128k to 4M.
- Get slurmd -f option working. Patch from Andriy Grytsenko.
- Fix for linking problem on OSX. Patches from Jon Bringham (LANL) and Tyler Strickland.
- Reset a job's priority to zero (suspended) when Moab requeues the job. Patch from Par Andersson, NSC.
- When enforcing accounting, fix polling for unknown uids for users after the slurmd started. Previously one would have to issue a reconfigure to the slurmd to have it look for new uids.
- BLUEGENE - if a block goes into an error state. Fix issue where accounting wasn't updated correctly when the block was resumed.
- Synchronize power-save module better with scheduler. Patch from Andriy Grytsenko (Massive Solutions Limited).
- Avoid SEGV in association logic with user=NULL. Patch from Andriy Grytsenko (Massive Solutions Limited).
- Fixed issue in accounting where it was possible for a new association/wckey to be set incorrectly as a default the new object was added after an original default object already existed. Before the slurmd would need to be restarted to fix the issue.
- Updated the Normalized Usage section in priority_multifactor.shtml.
- Disable use of SQUEUE_FORMAT env var if squeue -l, -o, or -s option is

used. Patch from Aaron Knister (UMBC).

* Changes in SLURM 2.2.5

=====

- Correct init.d/slurm status to have non-zero exit code if ANY Slurm daemon that should be running on the node is not running. Patch from Rod Schulz, Bull.
- Improve accuracy of response to "srun --test-only jobid=#".
- Correct logic to properly support --ntasks-per-node option in the select/cons_res plugin. Patch from Rod Schulz, Bull.
- Fix bug in select/cons_res with respect to generic resource (gres) scheduling which prevented some jobs from starting as soon as possible.
- Fix memory leak in select/cons_res when backfill scheduling generic resources (gres).
- Fix for when configuring a node with more resources than in real life and using task/affinity.
- Fix so slurmctld will pack correctly 2.1 step information. (Only needed if a 2.1 client is talking to a 2.2 slurmctld.)
- Set powered down node's state to IDLE+POWER after slurmctld restart instead of leaving in UNKNOWN+POWER. Patch from Andrej Gritsenko.
- Fix bug where srun's executable is not on it's current search path, but can be found in the user's default search path. Modify slurmstepd to find the executable. Patch from Andrej Gritsenko.
- Make sview display correct cpu count for steps.
- BLUEGENE - when running in overlap mode make sure to check the connection type so you can create overlapping blocks on the exact same nodes with different connection types (i.e. one torus, one mesh).
- Fix memory leak if MPI ports are reserved (for OpenMPI) and srun's --resv-ports option is used.
- Fix some anomalies in select/cons_res task layout when using the --cpus-per-task option. Patch from Martin Perry, Bull.
- Improve backfill scheduling logic when job specifies --ntasks-per-node and --mem-per-cpu options on a heterogeneous cluster. Patch from Bjorn-Helge Mevik, University of Oslo.
- Print warning message if srun specifies --cpus-per-task larger than used to create job allocation.
- Fix issue when changing a users name in accounting, if using wckey would execute correctly, but bad memcopy would core the DBD. No information would be lost or corrupted, but you would need to restart the DBD.

* Changes in SLURM 2.2.4

=====

- For batch jobs for which the Prolog fails, substitute the job ID for any "%j" in the job's output or error file specification.
- Add licenses field to the sview reservation information.
- BLUEGENE - Fix for handling extremely overloaded system on Dynamic system dealing with starting jobs on overlapping blocks. Previous fallout was job would be requeued. (happens very rarely)
- In accounting_storage/filetxt plugin, substitute spaces within job names, step names, and account names with an underscore to insure proper parsing.
- When building contribs/perlapi ignore both INSTALL_BASE and PERL_MM_OPT. Use PREFIX instead to avoid build errors from multiple installation specifications.
- Add job_submit/cnode plugin to support resource reservations of less than a full midplane on BlueGene computers. Treat cnodes as licenses which can

Appendix G. SLURM Release Information

be reserved and are consumed by jobs. This reservation mechanism for less than an entire midplane is still under development.

- Clear a job's "reason" field when a held job is released.
- When releasing a held job, calculate a new priority for it rather than just setting the priority to 1.
- Fix for sview started on a non-bluegene system to pick colors correctly when talking to a real bluegene system.
- Improve sched/backfill's expected start time calculation.
- Prevent abort of sacctmgr for dump command with invalid (or no) filename.
- Improve handling of job updates when using limits in accounting, and updating jobs as a non-admin user.
- Fix for "squeue --states=all" option. Bug would show no jobs.
- Schedule jobs with reservations before those without reservations.
- Fix squeue/scancel to query correctly against accounts of different case.
- Abort an srun command when it's associated job gets aborted due to a dependency that can not be satisfied.
- In jobcomp plugins, report start time of zero if pending job is cancelled. Previously may report expected start time.
- Fixed sacctmgr man to state correct variables.
- Select nodes based upon their Weight when job allocation requests include a constraint field with a count (e.g. "srun --constraint=gpu*2 -N4 a.out").
- Add support for user names that are entirely numeric and do not treat them as UID values. Patch from Dennis Leepow.
- Patch to un/pack double values properly if negative value. Patch from Dennis Leepow
- Do not reset a job's priority when requeued or suspended.
- Fix problem that could let new jobs start on a node in DRAINED state.
- Fix cosmetic sacctmgr issue where if the user you are trying to add doesn't exist in the /etc/passwd file and the account you are trying to add them to doesn't exist it would print (null) instead of the bad account name.
- Fix associations/qos for when adding back a previously deleted object the object will be cleared of all old limits.
- BLUEGENE - Added back a lock when creating dynamic blocks to be more thread safe on larger systems with heavy load.

* Changes in SLURM 2.2.3

=====

- Update srun, salloc, and sbatch man page description of --distribution option. Patches from Rod Schulz, Bull.
- Applied patch from Martin Perry to fix "Incorrect results for task/affinity block second distribution and cpus-per-task > 1" bug.
- Avoid setting a job's eligible time while held (priority == 0).
- Substantial performance improvement to backfill scheduling. Patch from Bjorn-Helge Mevik, University of Oslo.
- Make timeout for communications to the slurmctld be based upon the MessageTimeout configuration parameter rather than always 3 seconds. Patch from Matthieu Hautreux, CEA.
- Add new scontrol option of "show aliases" to report every NodeName that is associated with a given NodeHostName when running multiple slurmd daemons per compute node (typically used for testing purposes). Patch from Matthieu Hautreux, CEA.
- Fix for handling job names with a "'" in the name within MySQL accounting. Patch from Gerrit Renker, CSCS.
- Modify condition under which salloc execution delayed until moved to the

```

foreground. Patch from Gerrit Renker, CSCS.
Job control for interactive salloc sessions: only if ...
a) input is from a terminal (stdin has valid termios attributes),
b) controlling terminal exists (non-negative tpgid),
c) salloc is not run in allocation-only (--no-shell) mode,
d) salloc runs in its own process group (true in interactive
shells that support job control),
e) salloc has been configured at compile-time to support background
execution and is not currently in the background process group.
-- Abort salloc if no controlling terminal and --no-shell option is not used
("setsid salloc ..." is disabled). Patch from Gerrit Renker, CSCS.
-- Fix to gang scheduling logic which could cause jobs to not be suspended
or resumed when appropriate.
-- Applied patch from Martin Perry to fix "Slurmd abort when using task
affinity with plane distribution" bug.
-- Applied patch from Yiannis Georgiou to fix "Problem with cpu binding to
sockets option" behaviour. This change causes "--cpu_bind=sockets" to bind
tasks only to the CPUs on each socket allocated to the job rather than all
CPUs on each socket.
-- Advance daily or weekly reservations immediately after termination to avoid
having a job start that runs into the reservation when later advanced.
-- Fix for enabling users to change there own default account, wckey, or QOS.
-- BLUEGENE - If using OVERLAP mode fixed issue with multiple overlapping
blocks in error mode.
-- Fix for sacctmgr to display correctly default accounts.
-- scancel -s SIGKILL will always sent the RPC to the slurmd rather than
the slurmd daemon(s). This insures that tasks in the process of getting
spawned are killed.
-- BLUEGENE - If using OVERLAP mode fixed issue with jobs getting denied
at submit if the only option for their job was overlapping a block in
error state.

```

* Changes in SLURM 2.2.2

```

=====

```

```

-- Correct logic to set correct job hold state (admin or user) when setting
the job's priority using scontrol's "update jobid=..." rather than its
"hold" or "holdu" commands.
-- Modify queue to report unset --mincores, --minthreads or --extra-node-info
values as "*" rather than 65534. Patch from Rod Schulz, BULL.
-- Report the StartTime of a job as "Unknown" rather than the year 2106 if its
expected start time was too far in the future for the backfill scheduler
to compute.
-- Prevent a pending job reason field from inappropriately being set to
"Priority".
-- In sched/backfill with jobs having QOS_FLAG_NO_RESERVE set, then don't
consider the job's time limit when attempting to backfill schedule. The job
will just be preempted as needed at any time.
-- Eliminated a bug in sbatch when no valid target clusters are specified.
-- When explicitly sending a signal to a job with the scancel command and that
job is in a pending state, then send the request directly to the slurmd
daemon and do not attempt to send the request to slurmd daemons, which are
not running the job anyway.
-- In slurmd, properly set the up_node_bitmap when setting it's state to
IDLE (in case the previous node state was DOWN).
-- Fix smap to process block midplane names correctly when on a bluegene

```

Appendix G. SLURM Release Information

```
system.
-- Fix smap to once again print out the Letter 'ID' for each line of a block/
partition view.
-- Corrected the NOTES section of the scancel man page
-- Fix for accounting_storage/mysql plugin to correctly query cluster based
transactions.
-- Fix issue when updating database for clusters that were previously deleted
before upgrade to 2.2 database.
-- BLUEGENE - Handle mesh torus check better in dynamic mode.
-- BLUEGENE - Fixed race condition when freeing block, most likely only would
happen in emulation.
-- Fix for calculating used QOS limits correctly on a slurmctld reconfig.
-- BLUEGENE - Fix for bad conn-type set when running small blocks in HTC mode.
-- If salloc's --no-shell option is used, then do not attempt to preserve the
terminal's state.
-- Add new SLURM configure time parameter of --disable-salloc-background. If
set, then salloc can only execute in the foreground. If started in the
background, then a message will be printed and the job allocation halted
until brought into the foreground.
NOTE: THIS IS A CHANGE IN DEFAULT SALLOC BEHAVIOR FROM V2.2.1, BUT IS
CONSISTENT WITH V2.1 AND EARLIER.
-- Added the Multi-Cluster Operation web page.
-- Removed remnant code for enforcing max sockets/cores/threads in the
cons_res plugin (see last item in 2.1.0-pre5). This was responsible
for a bug reported by Rod Schultz.
-- BLUEGENE - Set correct env vars for HTC mode on a P system to get correct
block.
-- Correct RunTime reported by "scontrol show job" for pending jobs.

* Changes in SLURM 2.2.1
=====
-- Fix setting derived exit code correctly for jobs that happen to have the
same jobid.
-- Better checking for time overflow when rolling up in accounting.
-- Add scancel --reservation option to cancel all jobs associated with a
specific reservation.
-- Treat reservation with no nodes like one that starts later (let jobs of any
size get queued and do not block any pending jobs).
-- Fix bug in gang scheduling logic that would temporarily resume to many jobs
after a job completed.
-- Change srun message about job step being deferred due to SlurmctldProlog
running to be more clear and only print when --verbose option is used.
-- Made it so you could remove the hold on jobs with svview by setting the
priority to infinite.
-- BLUEGENE - better checking small blocks in dynamic mode whether a full
midplane job could run or not.
-- Decrease the maximum sleep time between srun job step creation retry
attempts from 60 seconds to 29 seconds. This should eliminate a possible
synchronization problem with gang scheduling that could result in job
step creation requests only occurring when a job is suspended.
-- Fix to prevent changing a held job's state from HELD to DEPENDENCY
until the job is released. Patch from Rod Schultz, Bull.
-- Fixed sprio -M to reflect PriorityWeight values from remote cluster.
-- Fix bug in svview when trying to update arbitrary field on more than one
job. Formerly would display information about one job, but update next
```

```

selected job.
-- Made it so QOS with UsageFactor set to 0 would make it so jobs running
under that QOS wouldn't add time to fairshare or association/qos
limits.
-- Fixed issue where QOS priority wasn't re-normalized until a slurmctld
restart when a QOS priority was changed.
-- Fix sprrio to use calculated numbers from slurmctld instead of
calculating it own numbers.
-- BLUEGENE - fixed race condition with preemption where if the wind blows the
right way the slurmctld could lock up when preempting jobs to run others.
-- BLUEGENE - fixed epilog to wait until MMCS job is totally complete before
finishing.
-- BLUEGENE - more robust checking for states when freeing blocks.
-- Added correct files to the slurm.spec file for correct perl api rpm
creation.
-- Added flag "NoReserve" to a QOS to make it so all jobs are created equal
within a QOS. So if larger, higher priority jobs are unable to run they
don't prevent smaller jobs from running even if running the smaller
jobs delay the start of the larger, higher priority jobs.
-- BLUEGENE - Check preemptees one by one to preempt lower priority jobs first
instead of first fit.
-- In select/cons_res, correct handling of the option
SelectTypeParameters=CR_ONE_TASK_PER_CORE.
-- Fix for checking QOS to override partition limits, previously if not using
QOS some limits would be overlooked.
-- Fix bug which would terminate a job step if any of the nodes allocated to
it were removed from the job's allocation. Now only the tasks on those
nodes are terminated.
-- Fixed issue when using a storage_accounting plugin directly without the
slurmDBD updates weren't always sent correctly to the slurmctld, appears to
OS dependent, reported by Fredrik Tegenfeldt.

```

* Changes in SLURM 2.2.0

=====

```

-- Change format of Duration field in "scontrol show reservation" output from
an integer number of minutes to "[days-]hours:minutes:seconds".
-- Add support for changing the reservation of pending or running jobs.
-- On Cray systems only, salloc sends SIGKILL to spawned process group when
job allocation is revoked. Patch from Gerrit Renker, CSCS.
-- Fix for sacctmgr to work correctly when modifying user associations where
all the associations contain a partition.
-- Minor mods to salloc signal handling logic: forwards more signals and
releases allocation on real-time signals. Patch from Gerrit Renker, CSCS.
-- Add salloc logic to preserve tty attributes after abnormal exit. Patch
from Mark Grondona, LLNL.
-- BLUEGENE - Fix for issue in dynamic mode when trying to create a block
overlapping a block with no job running on it but in configuring state.
-- BLUEGENE - Speedup by skipping blocks that are deallocating for other jobs
when starting overlapping jobs in dynamic mode.
-- Fix for sacct --state to work correctly when not specifying a start time.
-- Fix upgrade process in accounting from 2.1 for clusters named "cluster".
-- Export more jobacct_common symbols needed for the slurm api on some systems.

```

* Changes in SLURM 2.2.0.rc4

=====

Appendix G. SLURM Release Information

- Correction in logic to spread out over time highly parallel messages to minimize lost messages. Effects slurmd epilog complete messages and PMI key-pair transmissions. Patch from Gerrit Renker, CSCS.
- Fixed issue where if a system has unset messages to the dbd in 2.1 and upgrades to 2.2. Messages are now processed correctly now.
- Fixed issue where assoc_mgr cache wasn't always loaded correctly if the slurmdbd wasn't running when the slurmctld was started.
- Make sure on a pthread create in step launch that the error code is looked at. Improves fault-tolerance of slurmd.
- Fix setting up default acct/wckey when upgrading from 2.1 to 2.2.
- Fix issue with associations attached to a specific partition with no other association, and requesting a different partition.
- Added perlapi to the slurmdb to the slurm.spec.
- In sched/backfill, correct handling of CompleteWait parameter to avoid backfill scheduling while a job is completing. Patch from Gerrit Renker, CSCS.
- Send message back to user when trying to launch job on computing lacking that user ID. Patch from Hongjia Cao, NUDT.
- BLUEGENE - Fix it so 1 midplane clusters will run small block jobs.
- Add Command and WorkDir to the output of "scontrol show job" for job allocations created using srun (not just sbatch).
- Fixed sacctmgr to not add blank defaultqos' when doing a cluster dump.
- Correct processing of memory and disk space specifications in the salloc, sbatch, and srun commands to work properly with a suffix of "MB", "GB", etc. and not only with a single letter (e.g. "M", "G", etc.).
- Prevent nodes with suspended jobs from being powered down by SLURM.
- Normalized the way pidfile are created by the slurm daemons.
- Fixed modifying the root association to no read in it's last value when clearing a limit being set.
- Revert some resent signal handling logic from salloc so that SIGHUP sent after the job allocation will properly release the allocation and cause salloc to exit.
- BLUEGENE - Fix for recreating a block in a ready state.
- Fix debug flags for incorrect logic when dealing with DEBUG_FLAG_WIKI.
- Report reservation's Nodes as a hostlist expression of all nodes rather than using "ALL".
- Fix reporting of nodes in BlueGene reservation (was reporting CPU count rather than cnode count in scontrol output for NodeCnt field).

* Changes in SLURM 2.2.0.rc3

=====

- Modify sacctmgr command to accept plural versions of options (e.g. "Users" in addition to "User"). Patch from Don Albert, BULL.
- BLUEGENE - make it so reset of boot counter happens only on state change and not when a new job comes along.
- Modify srun and salloc signal handling so they can be interrupted while waiting for an allocation. This was broken in version 2.2.0.rc2.
- Fix NULL pointer reference in sview. Patch from Gerrit Renker, CSCS.
- Fix file descriptor leak in slurmstepd on spank_task_post_fork() failure. Patch from Gerrit Renker, CSCS.
- Fix bug in preserving job state information when upgrading from SLURM version 2.1. Bug introduced in version 2.2.0-pre10. Patch from Par Andersson, NSC.
- Fix bug where if using the slurmdbd if a job wasn't able to start right away some accounting information may be lost.

- BLUEGENE - when a prolog failure happens the offending block is put in an error state.
- Changed the last column heading of the sshare output from "FS Usage" to "FairShare" and added more detail to the sshare man page.
- Fix bug in enforcement of reservation by account name. Used wrong index into an array. Patch from Gerrit Renker, CSCS.
- Modify job_submit/lua plugin to treat any non-zero return code from the job_submit and job_modify functions as an error and the user request should be aborted.
- Fix bug which would permit pending job to be started on completing node when job preemption is configured.

* Changes in SLURM 2.2.0.rc2

=====

- Fix memory leak in job step allocation logic. Patch from Hongjia Cao, NUDT.
- If a preempted job was submitted with the --no-requeue option then cancel rather than requeue it.
- Fix for problems when adding a user for the first time to a new cluster with a 2.1 sacctmgr without specifying a default account.
- Resend TERMINATE_JOB message only to nodes that the job still has not terminated on. Patch from Hongjia Cao, NUDT.
- Treat time limit specification of "0:300" as a request for 300 seconds (5 minutes) instead of one minute.
- Modify sched/backfill plugin logic to continue working its way down the queue of jobs rather than restarting at the top if there are no changes in job, node, or partition state between runs. Patch from Hongjia Cao, NUDT.
- Improve scalability of select/cons_res logic. Patch from Matthieu Hautreux, CEA.
- Fix for possible deadlock in the slurmstepd when cancelling a job that is also writing a large amount of data to stderr.
- Fix in select/cons_res to eliminate "mem underflow" error when the slurmctld is reconfigured while a job is in completing state.
- Send a message to the a user's job when it's real or virtual memory limit is exceeded. :
- Apply rlimits right before execing the users task so to lower the risk of the task exiting because the slurmstepd ran over a limit (log file size, etc.)
- Add scontrol command of "uhold <job_id>" so that an administrator can hold a job and let the job's owner release it. The scontrol command of "hold <job_id>" when executed by a SLURM administrator can only be released by a SLURM administrator and not the job owner.
- Change atoi to slurm_atoul in mysql plugin, needed for running on 32-bit systems in some cases.
- If a batch job is found to be missing from a node, make its termination state be NODE_FAIL rather than CANCELLED.
- Fatal error put back if running a bluegene or cray plugin from a controller not of that type.
- Make sure jobacct_gather plugin is not shutdown before messing with the process list.
- Modify signal handling in srun and salloc commands to avoid deadlock if the malloc function is interrupted and called again. The malloc function is thread safe, but not reentrant, which is a problem when signal handling if the malloc function itself has a lock. Problem fixed by moving signal handling in those commands to a new pthread.
- In srun set job abort flag on completion to handle the case when a user

Appendix G. SLURM Release Information

```
    cancels a job while the node is not responding but slurmctld has not yet
    the node down. Patch from Hongjia Cao, NUDT.
-- Streamline the PMI logic if no duplicate keys are included in the key-pairs
    managed. Substantially improves performance for large numbers of tasks.
    Adds support for SLURM_PMI_KVS_NO_DUP_KEYS environment variable. Patch
    from Hongjia Cao, NUDT.
-- Fix issues with svview dealing with older versions of svview and saving
    defaults.
-- Remove references to --mincores, --minsockets, and --minthreads from the
    salloc, sbatch and srun man pages. These options are defunct, Patch from
    Rod Schultz, Bull.
-- Made openssl not be required to build RPMs, it is not required anymore
    since munge is the default crypto plugin.
-- sacctmgr now has smarts to figure out if a qos is a default qos when
    modifying a user/acct or removing a qos.
-- For reservations on BlueGene systems, set and report c-node counts rather
    than midplane counts.

* Changes in SLURM 2.2.0.rc1
=====
-- Add show_flags parameter to the slurm_load_block_info() function.
-- perlapi has been brought up to speed courtesy of Hongjia Coa. (make sure to
    run 'make clean' if building in a different dir than source)
-- Fixed regression in prel2 in crypto/munge when running with
    --enable-multiple-slurmd which would cause the slurmd's to core.
-- Fixed regression where cpu count wasn't figured out correctly for steps.
-- Fixed issue when using old mysql that can't handle a '.' in the table
    name.
-- Mysql plugin works correctly without the SlurmDBD
-- Added ability to query batch step with sstat. Currently no accounting data
    is stored for the batch step, but the internals are inplace if we decide to
    do that in the future.
-- Fixed some backwards compatibility issues with 2.2 talking to 2.1.
-- Fixed regression where modifying associations didn't get sent to the
    slurmctld.
-- Made sshare sort things the same way sacctmgr list assoc does
    (alphabetically)
-- Fixed issue with default accounts being set up correctly.
-- Changed sorting in the slurmctld so sshare output is similar to that of
    sacctmgr list assoc.
-- Modify reservation logic so that daily and weekly reservations maintain
    the same time when daylight savings time starts or ends in the interim.
-- Edit to make reservations handle updates to associations.
-- Added the derived exit code to the slurmctld job record and the derived
    exit code and string to the job record in the SLURM db.
-- Added slurm-sjobexit RPM for SLURM job exit code management tools.
-- Added ability to use sstat/sacct against the batch step.
-- Added OnlyDefaults option to sacctmgr list associations.
-- Modified the fairshare priority formula to  $F = 2^{**}(-Ue/S)$ 
-- Modify the PMI functions key-pair exchange function to support a 32-bit
    counter for larger job sizes. Patch from Hongjia Cao, NUDT.
-- In sched/builtin - Make the estimated job start time logic faster (borrowed
    new logic from sched/backfill and added pthread) and more accurate.
-- In select/cons_res fix bug that could result in a job being allocated zero
    CPUs on some nodes. Patch from Hongjia Cao, NUDT.
```



```
-- Fix bug in sched/backfill that could set expected start time of a job too
  far in the future.
-- Added ability to enforce new limits given to associations/qos on
  pending jobs.
-- Increase max message size for the slurmdbd from 1000000 to 16*1024*1024
-- Increase number of active threads in the slurmdbd from 50 to 100
-- Fixed small bug in src/common/slurmdb_defs.c reported by Bjorn-Helge Mevik
-- Fixed sacctmgr's ability to query associations against qos again.
-- Fixed svview show config on non-bluegene systems.
-- Fixed bug in selecting jobs based on sacct -N option
-- Fix bug that prevented job Epilog from running more than once on a node if
  a job was requeued and started no job steps.
-- Fixed issue where node index wasn't stored correcting when using DBD.
-- Enable srun's use of the --nodes option with --exclusive (previously the
  --nodes option was ignored).
-- Added UsageThreshold and Flags to the QOS object.
-- Patch to improve threadsafeness in the mysql plugins.
-- Add support for fair-share scheduling to be based upon resource use at
  the level of bank accounts and ignore use of individual users. Patch by
  Par Andersson, National Supercomputer Centre, Sweden.
```

* Changes in SLURM 2.2.0.pre12

```
=====
-- Log if Prolog or Epilog run for longer than MessageTimeout / 2.
-- Log the RPC number associated with messages from slurmctld that timeout.
-- Fix bug in select/cons_res logic when job allocation includes --overcommit
  and --ntasks-per-node options and the node has fewer CPUs than the count
  specified by --ntasks-per-node.
-- Fix bug in gang scheduling and job preemption logic so that preempted jobs
  get resumed properly after a slurmctld hot-start.
-- Fix bug in select/linear handling of gang scheduled jobs that could result
  in run_job_cnt underflow error message.
-- Fix bug in gang scheduling logic to properly support partitions added
  using the scontrol command.
-- Fix a segmentation fault in svview where the 'excluded_partitions' field
  was set to NULL, caused by the absence of ~/.slurm/svviewrc.
-- Rewrote some calls to is_user_any_coord() in src/plugins/accounting_storage
  modules to make use of is_user_any_coord()'s return value.
-- Add configure option of --with=dimensions=#.
-- Modify srun ping logic so that srun would only be considered not responsive
  if three ping messages were not responded to. Patch from Hongjia Cao (NUDT).
-- Preserve a node's ReasonTime field after scontrol reconfig command. Patch
  from Hongjia Cao (NUDT).
-- Added the authority for users with AdminLevel's defined in the SLURM db
  (Operators and Admins) and account coordinators to invoke commands that
  affect jobs, reservations, nodes, etc.
-- Fix for slurmd restart on completing node with no tasks to get the correct
  state, completing. Patch from Hongjia Cao (NUDT).
-- Prevent scontrol setting a node's Reason="". Patch from Hongjia Cao (NUDT).
-- Add new functions hostlist_ranged_string_malloc,
  hostlist_ranged_string_xmalloc, hostlist_deranged_string_malloc, and
  hostlist_deranged_string_xmalloc which will allocate memory as needed.
-- Make the slurm commands support both the --cluster and --clusters option.
  Previously, some commands support one of those options, but not the other.
-- Fix bug when resizing a job that has steps running on some of those nodes.
```

Appendix G. SLURM Release Information

Avoid killing the job step on remaining nodes. Patch from Rod Schultz (BULL). Also fix bug related to tracking the CPUs allocated to job steps on each node after releasing some nodes from the job's allocation.

- Applied patch from Rod Schultz / Matthieu Hautreux to keep the Node-to-Host cache from becoming corrupted when a hostname cannot be resolved.
- Export more symbols in libslurm for job and node state information translation (numbers to strings). Patch from Hongia Cao, NUDT.
- Add logic to retry sending RESPONSE_LAUNCH_TASKS messages from slurmd to srun. Patch from Hongia Cao, NUDT.
- Modify bit_unfmt_hexmask() and bit_unfmt_binmask() functions to clear the bitmap input before setting the bits indicated in the input string.
- Add SchedulerParameters option of bf_window to control how far into the future that the backfill scheduler will look when considering jobs to start. The default value is one day. See "man slurm.conf" for details.
- Fix bug that can result in duplicate job termination records in accounting for job termination when slurmd restarts or reconfigures.
- Modify plugin and library logic as needed to support use of the function slurm_job_step_stat() from user commands.
- Fix race condition in which PrologSlurmctld failure could cause slurmd to abort.
- Fix bug preventing users in secondary user groups from being granted access to partitions configured with AllowGroups.
- Added support for a default account and wckey per cluster within accounting.
- Modified select/cons_res plugin so that if MaxMemPerCPU is configured and a job specifies it's memory requirement, then more CPUs than requested will automatically be allocated to a job to honor the MaxMemPerCPU parameter.
- Added the derived_ec (exit_code) member to job_info_t. exit_code captures the exit code of the job script (or salloc) while derived_ec contains the highest exit code of all the job steps.
- Added SLURM_JOB_EXIT_CODE and SLURM_JOB_DERIVED_EC variables to the EpilogSlurmctld environment
- More work done on the accounting_storage/pgsql plugin, still beta. Patch from Hongjia Cao (NUDT).
- Major updates to svview from Dan Rusak (Bull), including:
 - Persistent option selections for each tab page
 - Clean up topology in grids
 - Leverage AllowGroups and Hidden options
 - Cascade full-info popups for ease of selection
- Add locks around the MySQL calls for proper operation if the non-thread safe version of the MySQL library is used.
- Remove libslurm.a, libpmi.a and libslurmdb.a from SLURM RPM. These static libraries are not generally usable.
- Fixed bug in sacctmgr when zeroing raw usage reported by Gerrit Renker.

* Changes in SLURM 2.2.0.pre11
=====

- Permit a regular user to change the partition of a pending job.
- Major re-write of the job_submit/lua plugin to pass pointers to available partitions and use lua metatables to reference the job and partition fields.
- Add support for several new trigger types: SlurmDBD failure/restart, Database failure/restart, Slurmctld failure/restart.
- Add support for SLURM_CLUSTERS environment variable in the sbatch, sinfo, squeue commands.
- Modify the sinfo and squeue commands to report state of multiple clusters if the --clusters option is used.

```
-- Added printf __attribute__ qualifiers to info, debug, ... to help prevent
bad/incorrect parameters being sent to them. Original patch from
Eygene Ryabinkin (Russian Research Centre).
-- Fix bug in slurmctld job completion logic when nodes allocated to a
completing job are re-booted. Patch from Hongjia Cao (NUDT).
-- In slurmctld's node record data structure, rename "hilbert_integer" to
"node_rank".
-- Add topology/node_rank plugin to sort nodes based upon rank loaded from
BASIL on Cray computers.
-- Fix memory leak in the auth/munge and crypto/munge plugins in the case of
some failure modes.

* Changes in SLURM 2.2.0.pre10
=====
-- Fix issue when EnforcePartLimits=yes in slurm.conf all jobs where no nodecnt
was specified the job would be seen to have maxnodes=0 which would not
allow jobs to run.
-- Fix issue where if not suspending a job the gang scheduler does the correct
kill procedure.
-- Fixed some issues when dealing with jobs from a 2.1 system so they live
after an upgrade.
-- In srun, log if --cpu_bind options are specified, but not supported by the
current system configuration.
-- Various Patches from Hongjia Cao dealing with bugs found in sacctmgr and
the slurmdbd.
-- Fix bug in changing the nodes allocated to a running job and some node
names specified are invalid, avoid invalid memory reference.
-- Fixed filename substitution of %h and %n based on patch from Ralph Bean
-- Added better job sorting logic when preempting jobs with qos.
-- Log the IP address and port number for some communication errors.
-- Fix bug in select/cons_res when --cpus_per_task option is used, could
oversubscribe resources.
-- In srun, do not implicitly set the job's maximum node count based upon a
required hostlist.
-- Avoid running the HealthCheckProgram on non-responding nodes rather than
DOWN nodes.
-- Fix bug in handling of poll() functions on OS X (SLURM was ignoring POLLIN
if POLLHUP flag was set at the same time).
-- Pulled Cray logic out of common/node_select.c into it's own
select/cray plugin cons_res is the default. To use linear add 'Linear' to
SelectTypeParameters.
-- Fixed bug where resizing jobs didn't correctly set used limits correctly.
-- Change sched/backfill default time interval to 30 seconds and defer attempt
to backfill schedule if slurmctld has more than 5 active RPCs. General
improvements in logic scalability.
-- Add SchedulerParameters option of default_sched_depth=# to control how
many jobs on queue should be tested for attempted scheduling when a job
completes or other routine events. Default value is 100 jobs. The full job
queue is tested on a less frequent basis. This option can dramatically
improve performance on systems with thousands of queued jobs.
-- Gres/gpu now sets the CUDA_VISIBLE_DEVICES environment to control which
GPU devices should be used for each job or job step and CUDA version 3.1+
is used. NOTE: SLURM's generic resource support is still under development.
-- Modify select/cons_res to pack jobs onto allocated nodes differently and
minimize system fragmentation. For example on nodes with 8 CPUs each, a
```

Appendix G. SLURM Release Information

job needing 10 CPUs will now ideally be allocated 8 CPUs on one node and 2 CPUs on another node. Previously the job would have ideally been allocated 5 CPUs on each node, fragmenting the unused resources more.

- Modified the behavior of `update_job()` in `job_mgr.c` to return when the first error is encountered instead of continuing with more job updates.
- Removed all references to the following `slurm.conf` parameters, all of which have been removed or replaced since version 2.0 or earlier: `HashBase`, `HeartbeatInterval`, `JobAcctFrequency`, `JobAcctLogFile` (instead use `AccountingStorageLoc`), `JobAcctType`, `KillTree`, `MaxMemPerTask`, and `MpichGmDirectSupport`.
- Fix bug in `slurmctld` restart logic that improperly reported jobs had invalid features: "Job 65537 has invalid feature list: fat".
- BLUEGENE - Removed thread pool for destroying blocks. It turns out the memory leak we were concerned about for creating and destroying threads in a plugin doesn't exist anymore. This increases throughput dramatically, allowing multiple jobs to start at the same time.
- BLUEGENE - Removed thread pool for starting and stopping jobs. For similar reasons as noted above.
- BLUEGENE - Handle blocks that never deallocate.

* Changes in SLURM 2.2.0.pre9
=====

- `sbatch` can now submit jobs to multiple clusters and run on the earliest available.
- Fix bug introduced in pre8 that prevented job dependencies and job triggers from working without the `--enable-debug` configure option.
- Replaced `slurm_addr` with `slurm_addr_t`
- Replaced `slurm_fd` with `slurm_fd_t`
- Skeleton code added for BlueGeneQ.
- Jobs can now be submitted to multiple partitions (job queues) and use the one permitting earliest start time.
- Change `slurmdb_coord_table` back to `acct_coord_table` to keep consistent with < 2.1.
- Introduced locking system similar to that in the `slurmctld` for the `assoc_mgr`.
- Added ability to change a users name in accounting.
- Restore `squeue` support for "%G" format (group id) accidentally removed in 2.2.0.pre7.
- Added `preempt_mode` option to QOS.
- Added a `grouping=individual` for `sreport` size reports.
- Added `remove_qos` logic to jobs running under a QOS that was removed.
- `scancel` now exits with a 1 if any job is non-existent when canceling.
- Better handling of select plugins that don't exist on various systems for cross cluster communication. `Slurmctld`, `slurmd`, and `slurmstepd` now only load the default select plugin as well.
- Better error handling when loading plugins.
- Prevent `scontrol` from aborting if `getlogin()` returns NULL.
- Prevent `scontrol` segfault when there are hidden nodes.
- Prevent `srund` segfault after task launch failure.
- Added `job_submit/lua` plugin.
- Fixed `sinfo` on a bluegene system to print correctly the output for:
`sinfo -e -o "%9P %6m %.4c %.22F %f"`
- Add `scontrol` commands "hold" and "release" to simplify setting a job's priority to 0 or 1. Also tests that the job is in pending state.
- Increase maximum node list size (for incoming RPC) from 1024 bytes to 64k.

```
-- In the backup slurmctld, purge triggers before recovering trigger state to
avoid duplicate entries.
-- Fix bug in sacct processing of --fields= option.
-- Fix bug in checkpoint/blcr for jobs spanning multiple nodes introduced when
changing some variable names in version 2.2.0.pre5.
-- Removed the vestigial set_max_cluster_usage() function from the Priority
Plugin API.
-- Modify the output of "scontrol show job" for the field ReqS:C:T=. Fields
not specified by the user will be reported as "*" instead of 65534.
-- Added DefaultQOS option for an association.
-- BLUEGENE - Added -B option to the slurmctld to clear created blocks from
the system on start.
-- BLUEGENE - Added option to scontrol & svview to recreate existing blocks.
-- Fixed flags for returning messages to use the correct munge key when going
cross-cluster.
-- BLUEGENE - Added option to scontrol & svview to resume blocks in an error
state instead of just freeing them.
-- svview patched to allow multiple row selection of jobs, patch from Dan Rusak
-- Lower default slurmctld server thread count from 1024 to 256. Some systems
process threads on a last-in first-out basis and the high thread count was
causing unexpectedly high delays for some RPCs.
-- Added to sacctmgr the ability for admins to reset the raw usage of a user
or account
-- Improved the efficiency of a few lines in sacctmgr
```

* Changes in SLURM 2.2.0.pre8

=====

```
-- Add DebugFlags parameter of "Backfill" for sched/backfill detailed logging.
-- Add DebugFlags parameter of "Gang" for detailed logging of gang scheduling
activities.
-- Add DebugFlags parameter of "Priority" for detailed logging of priority
multifactor activities.
-- Add DebugFlags parameter of "Reservation" for detailed logging of advanced
reservations.
-- Add run time to mail message upon job termination and queue time for mail
message upon job begin.
-- Add email notification option for job requeue.
-- Generate a fatal error if the srun --relative option is used when not
within an existing job allocation.
-- Modify the meaning of InactiveLimit slightly. It will now cancel the job
allocation created using the salloc or srun command if those commands
cease responding for the InactiveLimit regardless of any running job steps.
This parameter will no longer effect jobs spawned using sbatch.
-- Remove AccountingStoragePass and JobCompPass from configuration RPC and
scontrol show config command output. The use of SlurmDBD is still strongly
recommended as SLURM will have limited database functionality or protection
otherwise.
-- Add sbatch options of --export and SBATCH_EXPORT to control which
environment variables (if any) get propogated to the spawned job. This is
particularly important for jobs that are submitted on one cluster and run
on a different cluster.
-- Fix bug in select/linear when used with gang scheduling and there are
preempted jobs at the time slurmctld restarts that can result in over-
subscribing resources.
-- Added keeping track of the qos a job is running with in accounting.
```

Appendix G. SLURM Release Information

- Fix for handling correctly jobs that resize, and also reporting correct stats on a job after it finishes.
- Modify gang scheduler so with SelectTypeParameter=CR_CPUS and task affinity is enabled, keep track of the individual CPUs allocated to jobs rather than just the count of CPUs allocated (which could overcommit specific CPUs for running jobs).
- Modify select/linear plugin data structures to eliminate underflow errors for the exclusive_cnt and tot_job_cnt variables (previously happened when slurmctld reconfigured while the job was in completing state).
- Change slurmd's working directory (and location of core files) to match that of the slurmctld daemon: the same directory used for log files, SlurmdLogFile (if specified with an absolute pathname) otherwise the directory used to save state, SlurmdSpoolDir.
- Add sattach support for the --pty option.
- Modify slurmctld communications logic to accept incoming messages on more than one port for improved scalability.
- Add SchedulerParameters option of "defer" to avoid trying to schedule a job at submission time, but to attempt scheduling many jobs at once for improved performance under heavy load.
- Correct logic controlling slurmctld thread limit eliminating check of RLIMIT_STACK.
- Make slurmctld's trigger logic more robust in the event that job records get purged before their trigger can be processed (e.g. MinJobAge=1).
- Add support for users to hold/release their own jobs (submit the job with srun/sbatch --hold/-H option or use "scontrol update jobid=# priority=0" to hold and "scontrol update jobid=# priority=1" to release).
- Added ability for sacct to query jobs by qos and a range of timelimits.
- Added ability for sstat to query pids of steps running.
- Support time specification in UTS format with a prefix of "uts" (e.g. "sbatch --begin=uts458389988 my.script").

* Changes in SLURM 2.2.0.pre7

=====

- Fixed issue with sacctmgr if querying against non-existent cluster it works the same way as 2.1.
- Added infrastructure to support allocation of generic node resources (gres).
 - Modified select/linear and select/cons_res plugins to allocate resources at the level of a job without oversubscription.
 - Get sched/backfill operating with gres allocations.
 - Get gres configuration changes (reconfiguration) working.
 - Have job steps allocate resources.
 - Modified job step credential to include the job's and step's gres allocation details.
 - Integrate with HWLOC library to identify GPUs and NICs configured on each node.
- SLURM commands (squeue, sinfo, etc...) can now go cross-cluster on like linux systems. Cross-cluster for bluegene to linux and such should work fine, even svview.
- Added the ability to configure PreemptMode on a per-partition basis.
- Change slurmctld's default thread limit count to 1024, but adjust that down as needed based upon the process's resource limits.
- Removed the non-functional "SystemCPU" and "TotalCPU" reporting fields from sstat and updated man page
- Correct location of apbasil command on Cray XT systems.
- Fixed bug in MinCPU and AveCPU calculations in sstat command

```
-- Send message to srun when the Prolog takes too long (MessageTimeout) to
complete.
-- Change timeout for socket connect() to be half of configured MessageTimeout.
-- Added high-throughput computing web page with configuration guidance.
-- Use more srun sockets to process incoming PMI (MPICH2) connections for
better scalability.
-- Added DebugFlags for the select/bluegene plugin: DEBUG_FLAG_BG_PICK,
DEBUG_FLAG_BG_WIRES, DEBUG_FLAG_BG_ALGO, and DEBUG_FLAG_BG_ALGO_DEEP.
-- Remove vestigial job record field "kill_on_step_done" (internal to the
slurmctld daemon only).
-- For MPICH2 jobs: Clear PMI state between job steps.
```

* Changes in SLURM 2.2.0.pre6

=====

```
-- svview - added ability to see database configuration.
-- svview - added ability to add/remove visible tabs.
-- svview - change way grid highlighting takes place on selected objects.
-- Added infrastructure to support allocation of generic node resources.
-Added node configuration parameter of Gres=.
-Added ability to view/modify a node's gres using scontrol, sinfo and svview.
-Added salloc, sbatch and srun --gres option.
-Added ability to view a job or job step's gres using scontrol, squeue and
svview.
-Added new configuration parameter GresPlugins to define plugins used to
manage generic resources.
-Added framework for gres plugins.
-Added DebugFlags option of "gres" for detailed debugging of gres actions.
-- Slurmd modified to log slow slurmd startup and note possible file system
problem.
-- svview - There is now a .slurm/svviewrc created when running svview.
Defaults are put in there as to how svview looks when first launched.
You can set these by Ctrl-S or Options->Set Default Settings.
-- Add scontrol "wait_job <job_id>" option to wait for nodes to boot as needed.
Useful for batch jobs (in Prolog, PrologSlurmctld or the script) if powering
down idle nodes.
-- Added salloc and sbatch option --wait-all-nodes. If set non-zero, job
initiation will be delayed until all allocated nodes have booted. Salloc
will log the delay with the messages "Waiting for nodes to boot" and "Nodes
are ready for job".
-- The Priority/multifactor plugin now takes into consideration size of job
in cpus as well as size in nodes when looking at the job size factor.
Previously only nodes were considered.
-- When using the SlurmDBD messages waiting to be sent will be combined
and sent in one message.
-- Remove srun's --core option. Move the logic to an optional SPANK plugin
(currently in the contribs directory, but plan to distribute through
http://code.google.com/p/slurm-spank-plugins/).
-- Patch for adding CR_CORE_DEFAULT_DIST_BLOCK as a select option to layout
jobs using block layout across cores within each node instead of cyclic
which was previously the default.
-- Accounting - When removing associations if jobs are running, those jobs
must be killed before proceeding. Before the jobs were killed
automatically thus causing user confusion on what is most likely an
admin's mistake.
-- svview - color column keeps reference color when highlighting.
```

Appendix G. SLURM Release Information

- Configuration parameter MaxJobCount changed from 16-bit to 32-bit field. The default MaxJobCount was changed from 5,000 to 10,000.
- SLURM commands (squeue, sinfo, etc...) can now go cross-cluster on like linux systems. Cross-cluster for bluegene to linux and such does not currently work. You can submit jobs with sbatch. Salloc and srun are not cross-cluster compatible, and given their nature to talk to actual compute nodes these will likely never be.
- salloc modified to forward SIGTERM to the spawned program.
- In sched/wiki2 (for Moab support) - Add GRES and WCKEY fields to MODIFYJOBS and GETJOBS commands. Add GRES field to GETNODES command.
- In struct job_descriptor and struct job_info: rename min_sockets to sockets_per_node, min_cores to cores_per_socket, and min_threads to threads_per_core (the values are not minimum, but represent the target values).
- Fixed bug in clearing a partition's DisableRootJobs value reported by Hongjia Cao.
- Purge (or ignore) terminated jobs in a more timely fashion based upon the MinJobAge configuration parameter. Small values for MinJobAge should improve responsiveness for high job throughput.

* Changes in SLURM 2.2.0.pre5

=====

- Modify commands to accept time format with one or two digit hour value (e.g. 8:00 or 08:00 or 8:00:00 or 08:00:00).
- Modify time parsing logic to accept "minute", "hour", "day", and "week" in addition to the currently accepted "minutes", "hours", etc.
- Add slurmd option of "-C" to print actual hardware configuration and exit.
- Pass EnforcePartLimits configuration parameter from slurmd to user commands to see the correct value instead of always "NO".
- Modify partition data structures to replace the default_part, disable_root_jobs, hidden and root_only fields with a single field called "flags" populated with the flags PART_FLAG_DEFAULT, PART_FLAG_NO_ROOT PART_FLAG_HIDDEN and/or PART_FLAG_ROOT_ONLY. This is a more flexible solution besides making for smaller data structures.
- Add node state flag of JOB_RESIZING. This will only exist when a job's accounting record is being written immediately before or after it changes size. This permits job accounting records to be written for a job at each size.
- Make calls to jobcomp and accounting_storage plugins before and after a job changes size (with the job state being JOB_RESIZING). All plugins write a record for the job at each size with intermediate job states being JOB_RESIZING.
- When changing a job size using scontrol, generate a script that can be executed by the user to reset SLURM environment variables.
- Modify select/linear and select/cons_res to use resources released by job resizing.
- Added to contribs foundation for Perl extension for slurmdb library.
- Add new configuration parameter JobSubmitPlugins which provides a mechanism to set default job parameters or perform other site-configurable actions at job submit time.
- Better postgres support for accounting, still beta.
- Speed up job start when using the slurmdbd.
- Forward step failure reason back to slurmd before in some cases it would just be SLURM_FAILURE returned.
- Changed squeue to fail when passed invalid -o <output_format> or

-S <sort_list> specifications.

* Changes in SLURM 2.2.0.pre4

=====

- Add support for a PropagatePrioProcess configuration parameter value of 2 to restrict spawned task nice values to that of the slurmd daemon plus 1. This insures that the slurmd daemon always have a higher scheduling priority than spawned tasks.
- Add support in slurmctld, slurmd and slurmdbd for option of "-n <value>" to reset the daemon's nice value.
- Fixed slurm_load_slurmd_status and slurm_pid2jobid to work correctly when multiple slurmds are in use.
- Altered srun to set max_nodes to min_nodes if not set when doing an allocation to mimic that which salloc and sbatch do. If running a step if the max isn't set it remains unset.
- Applied patch from David Egolf (David.Egolf@Bull.com). Added the ability to purge/archive accounting data on a day or hour basis, previously it was only available on a monthly basis.
- Add support for maximum node count in job step request.
- Fix bug in CPU count logic for job step allocation (used count of CPUS per node rather than CPUs allocated to the job).
- Add new configuration parameters GroupUpdateForce and GroupUpdateTime. See "man slurm.conf" for details about how these control when slurmctld updates its information of which users are in the groups allowed to use partitions.
- Added sacctmgr list events which will list events that have happened on clusters in accounting.
- Permit a running job to shrink in size using a command of "scontrol update JobId=# NumNodes=#" or "scontrol update JobId=# NodeList=<names>". Subsequent job steps must explicitly specify an appropriate name count to work properly.
- Added resize_time field to job record noting the time of the latest job size change (to be used for accounting purposes).
- sview/smap now hides hidden partitions and their jobs by default, with an option to display them.

* Changes in SLURM 2.2.0.pre3

=====

- Refine support for TotalView partial attach. Add parameter to configure program of "--enable-partial-attach".
- In select/cons_res, the count of CPUs on required nodes was formerly ignored in enforcing the maximum CPU limit. Also enforce maximum CPU limit when the topology/tree plugin is configured (previously ignored).
- In select/cons_res, allocate cores for a job using a best-fit approach.
- In select/cons_res, for jobs that can run on a single node, use a best-fit packing approach.
- Add support for new partition states of DRAIN and INACTIVE and new partition option of "Alternate" (alternate partition to use for jobs submitted to partitions that are currently in a state of DRAIN or INACTIVE).
- Add group membership cache. This can substantially speed up slurmctld startup or reconfiguration if many partitions have AllowGroups configured.
- Added slurmdb api for accessing slurm DB information.
- In select/linear: Modify data structures for better performance and to avoid underflow error messages when slurmctld restarts while jobs are in completing state.

Appendix G. SLURM Release Information

- Added hash for slurm.conf so when nodes check in to the controller it can verify the slurm.conf is the same as the one it is running. If not an error message is displayed. To silence this message add NO_CONF_HASH to DebugFlags in your slurm.conf.
 - Added error code ESLURM_CIRCULAR_DEPENDENCY and prevent circular job dependencies (e.g. job 12 dependent upon job 11 AND job 11 is dependent upon job 12).
 - Add BootTime and SlurmdStartTime to available node information.
 - Fixed moab_2_slurmdb to work correctly under new database schema.
 - Slurmd will drain a compute node when the SlurmdSpoolDir is full.
- * Changes in SLURM 2.2.0.pre2
=====
- Add support for spank_get_item() to get S_STEP_ALLOC_CORES and S_STEP_ALLOC_MEM. Support will remain for S_JOB_ALLOC_CORES and S_JOB_ALLOC_MEM.
 - Kill individual job steps that exceed their memory limit rather than killing an entire job if one step exceeds its memory limit.
 - Added configuration parameter VSizeFactor to enforce virtual memory limits for jobs and job steps as a percentage of their real memory allocation.
 - Add scontrol ability to update job step's time limits.
 - Add scontrol ability to update job's NumCPUs count.
 - Add --time-min options to salloc, sbatch and srun. The scontrol command has been modified to display and modify the new field. sched/backfill plugin has been changed to alter time limits of jobs with the --time-min option if doing so permits earlier job initiation.
 - Add support for TotalView symbol MPIR_partial_attach_ok with srun support to release processes which TotalView does not attach to.
 - Add new option for SelectTypeParameters of CR_ONE_TASK_PER_CORE. This option will allocate one task per core by default. Without this option, by default one task will be allocated per thread on nodes with more than one ThreadsPerCore configured.
 - Avoid accounting separately for a current pid corresponds to a Light Weight Process (Thread POSIX) appearing in the /proc directory. Only account for the original process (pid==tgid) to avoid accounting for memory use more than once.
 - Add proctrack/cgroup plugin which uses Linux control groups (aka cgroup) to track processes on Linux systems having this feature enabled (kernel >= 2.6.24).
 - Add logging of license transactions including job_id.
 - Add configuration parameters SlurmSchedLogFile and SlurmSchedLogLevel to support writing scheduling events to a separate log file.
 - Added contribs/web_apps/chart_stats.cgi, a web app that invokes sreport to retrieve from the accounting storage db a user's request for job usage or machine utilization statistics and charts the results to a browser.
 - Massive change to the schema in the storage_accounting/mysql plugin. When starting the slurmdbd the process of conversion may take a few minutes. You might also see some errors such as 'error: mysql_query failed: 1206 The total number of locks exceeds the lock table size'. If you get this, do not worry, it is because your setting of innodb_buffer_pool_size in your my.cnf file is not set or set too low. A decent value there should be 64M or higher depending on the system you are running on. See RELEASE_NOTES for more information. But setting this and then restarting the mysqld and slurmdbd will put things right. After this change we have noticed 50-75% increase in performance with sreport and

```

sacct.
-- Fix for MaxCPUs to honor partitions of 1 node that have more than the
maxcpus for a job.
-- Add support for "scontrol notify <message>" to work for batch jobs.

* Changes in SLURM 2.2.0.prel
=====
-- Added RunTime field to scontrol show job report
-- Added SLURM_VERSION_NUMBER and removed SLURM_API_VERSION from
slurm/slurm.h.
-- Added support to handle communication with SLURM 2.1 clusters. Job's
should not be lost in the future when upgrading to higher versions of
SLURM.
-- Added withdeleted options for listing clusters, users, and accounts
-- Remove PLPA task affinity functions due to that package being deprecated.
-- Preserve current partition state information and node Feature and Weight
information rather than use contents of slurm.conf file after slurmctld
restart with -R option or SIGHUP. Replace information with contents of
slurm.conf after slurmctld restart without -R or "scontrol reconfigure".
See RELEASE_NOTES file fore more details.
-- Modify SLURM's PMI library (for MPICH2) to properly execute an executable
program stand-alone (single MPI task launched without srun).
-- Made GrpCPUs and MaxCPUs limits work for select/cons_res.
-- Moved all SQL dependant plugins into a seperate rpm slurm-sql. This
should be needed only where a connection to a database is needed (i.e.
where the slurmdbd is running)
-- Add command line option "no_sys_info" to PAM module to supress system
logging of "access granted for user ...", access denied and other errors
will still be logged.
-- sinfo -R now has the user and timestamp in separate fields from the reason.
-- Much functionality has been added to account_storage/pgsql. The plugin
is still in a very beta state. It is still highly advised to use the
mysql plugin, but if you feel like living on the edge or just really
like postgres over mysql for some reason here you go. (Work done
primarily by Hongjia Cao, NUDT.)

* Changes in SLURM 2.1.17
=====
-- Correct format of --begin reported in salloc, sbatch and srun --help
message.
-- Correct logic for regular users to increase nice value of their own jobs.

* Changes in SLURM 2.1.16
=====
-- Fixed minor warnings from gcc-4.5
-- Fixed initialization of accounting_stroage_enforce in the slurmctld.
-- Fixed bug where if GrpNodes was lowered while pending jobs existed and where
above the limit the slurmctld would seg fault.
-- Fixed minor memory leak when unpack error happens on an
association_shares_object_t.
-- Set Lft and Rgt correctly when adding association. Fix for regression
caused in 2.1.15, cosmetic fix only.
-- Replaced optarg which was undefined in some spots to make sure ENV vars are
set up correctly.
-- When removing an account from a cluster with sacctmgr you no longer get

```

Appendix G. SLURM Release Information

```
a list of previously deleted associations.
-- Fix to make jobcomp/(pg/my)sql correctly work when the database name is
different than the default.

* Changes in SLURM 2.1.15
=====
-- Fix bug in which backup slurmctld can purge job scripts (and kill batch
jobs) when it assumes primary control, particularly when this happens
multiple times in a short time interval.
-- In sched/wiki and sched/wiki2 add IWD (Initial Working Directory) to the
information reported about jobs.
-- Fix bug in calculating a daily or weekly reservation start time when the
reservation is updated. Patch from Per Lundqvist (National Supercomputer
Centre, Linköping University, Sweden).
-- Fix bug in how job step memory limits are calculated when the --relative
option is used.
-- Restore operation of srun -X option to forward SIGINT to spawned tasks
without killing them.
-- Fixed a bug in calculating the root account's raw usage reported by Par
Andersson
-- Fixed a bug in sshare displaying account hierarchy reported by Per
Lundqvist.
-- In select/linear plugin, when a node allocated to a running job is removed
from a partition, only log the event once. Fixes problem reported by Per
Lundqvist.

* Changes in SLURM 2.1.14
=====
-- Fixed coding mistakes in _slurm_rpc_resv_show() and job_alloc_info() found
while reviewing the code.
-- Fix select/cons_res logic to prevent allocating resources while jobs
previously allocated resources on the node are still completing.
-- Fixed typo in job_mgr.c dealing with qos instead of associations.
-- Make sure associations and qos' are initiated when added.
-- Fixed wrong initialization for wckeys in the association manager.
-- Added wiki.conf configuration parameter of HidePartitionNodes. See
"man wiki.conf" for more information.
-- Add "JobAggregationTime=#" field SchedulerParameter configuration parameter
output.
-- Modify init.d/slurm and slurmdbd scripts to prevent the possible
inadvertent inclusion of "." in LD_LIBRARY_PATH environment variable.
To fail, the script would need to be executed by user root or SlurmUser
without the LD_LIBRARY_PATH environment variable set and there would
have to be a maliciously altered library in the working directory.
Thanks to Raphael Geissert for identifying the problem. This addresses
security vulnerability CVE-2010-3380.

* Changes in SLURM 2.1.13
=====
-- Fix race condition which can set a node state to IDLE on slurmctld startup
even if it has running jobs.

* Changes in SLURM 2.1.12
=====
-- Fixes for building on OS X 10.5.
```

```
-- Fixed a few '-' without a '\' in front of them in the man pages.
-- Fixed issues in client tools where a requeued job did get displayed
  correctly.
-- Update typos in doc/html/accounting.shtml doc/html/resource_limits.shtml
  doc/man/man5/slurmdbd.conf.5 and doc/man/man5/slurm.conf.5
-- Fixed a bug in exitcode:signal display in sacct
-- Fix bug when request comes in for consumable resources and the -c option
  is used in conjunction with -O
-- Fixed squeue -o "%h" output formatting
-- Change select/linear message "error: job xxx: best_fit topology failure"
  to debug type.
-- BLUEGENE - Fix for sinfo -R to group all midplanes together in a single
  line for midplanes in an error state instead of 1 line for each midplane.
-- Fix srun to work correctly with --uid when getting an allocation
  and creating a step, also fix salloc to assume identity at the correct
  time as well.
-- BLUEGENE - Fixed issue with jobs being refused when running dynamic mode
  and every job on the system happens to be the same size.
-- Removed bad #define _SLURMD_H from slurmd/get_mach_stat.h. Didn't appear
  to cause any problems being there, just incorrect syntax.
-- Validate the job ID when salloc or srun receive an SRUN_JOB_COMPLETE RPC to
  avoid killing the wrong job if the original command exits and the port gets
  re-used by another command right away.
-- Fix to node in correct state in accounting when updating it to drain from
  scontrol/sview.
-- BLUEGENE - Removed incorrect unlocking on error cases when starting jobs.
-- Improve logging of invalid sinfo and squeue print options.
-- BLUEGENE - Added check to libsched_if to allow root to run even outside of
  SLURM. This is needed when running certain blocks outside of SLURM in HTC
  mode.
```

* Changes in SLURM 2.1.11-2

=====

```
-- BLUEGENE - make it so libsched_if.so is named correctly on 'L' it is
  libsched_if64.so and on 'P' it is libsched_if.so
```

* Changes in SLURM 2.1.11

=====

```
-- BLUEGENE - fix sinfo to not get duplicate entries when running command
  sinfo -e -o "%9P %6m %.4c %.22F %f"
-- Fix bug that caused segv when deleting a partition with pending jobs.
-- Better error message for when trying to modify an account's name with
  sacctmgr.
-- Added back removal of #include "src/common/slurm_xlator.h" from
  select/cons_res.
-- Fix incorrect logic in global_accounting in regression tests for
  setting QOS.
-- BLUEGENE - Fixed issue where removing a small block in dynamic mode,
  and other blocks also in that midplane needed to be removed and were in
  and error state. They all weren't removed correctly in accounting.
-- Prevent scontrol segv with "scontrol show node <name>" command with nodes
  in a hidden partition.
-- Fixed sizing of popup grids in sview.
-- Fixed sacct when querying against a jobid the start time is not set.
-- Fix configure to get correct version of pkg-config if both 32bit and 64bit
```

Appendix G. SLURM Release Information

```
    libs are installed.
-- Fix issue with sshare not sorting correctly the tree of associations.
-- Update documentation for sreport.
-- BLUEGENE - fix regression in 2.1.10 on assigning multiple jobs to one block.
-- Minor memory leak fixed when killing job error happens.
-- Fix sacctmgr list assoc when talking to a 2.2 slurmdbd.

* Changes in SLURM 2.1.10
=====
-- Fix memory leak in sched/builtin plugin.
-- Fixed sbatch to work correctly when no nodes are specified, but
   --ntasks-per-node is.
-- Make sure account and wckey for a job are lower case before inserting into
   accounting.
-- Added note to squeue documentation about --jobs option displaying jobs
   even if they are on hidden partitions.
-- Fix srun to work correctly with --uid when getting an allocation
   and creating a step.
-- Fix for when removing a limit from a users association inside the
   fairshare tree the parents limit is now inherited automatically in
   the slurmctld. Previously the slurmctld would have to be restarted.
   This problem only exists when setting a users association limit to -1.
-- Patch from Matthieu Hautreux (CEA) dealing with possible overflows that
   could come up with the select/cons_res plugin with uint32_t's being treated
   as uint16_t's.
-- Correct logic for creating a reservation with a Duration=Infinite (used to
   set reservation end time in the past).
-- Correct logic for creating a reservation that properly handles the OVERLAP
   and IGNORE_JOBS flags (flags were ignored under some conditions).
-- Fixed a fair-share calculation bug in the priority/multifactor plugin.
-- Make sure a user entry in the database that was previously deleted is
   restored clean when added back, i.e. remove admin privileges previously
   given.
-- BLUEGENE - Future start time is set correctly when eligible time for a job
   is in the future, but the job can physically run earlier.
-- Updated Documentation for sacctmgr for Wall and CPUMin options stating when
   the limit is reached running jobs will be killed.
-- Fix deadlock issue in the slurmctld when lowering limits in accounting to
   lower than that of pending jobs.
-- Fix bug in salloc, sbatch and srun that could under some conditions process
   the --threads-per-core, --cores-per-socket and --sockets-per-node options
   improperly.
-- Fix bug in select/cons_res with memory management plus job preemption with
   job removal (e.g. requeue) which under some conditions failed to preempt
   jobs.
-- Fix deadlock potential when using qos and associations in the slurmctld.
-- Update documentation to state --ntasks-per-* is for a maximum value
   instead of an absolute.
-- Get ReturnToService=2 working for front-end configurations (e.g. Cray or
   BlueGene).
-- Do not make a non-responding node available for use after running
   "scontrol update nodename=<name> state=resume". Wait for node to respond
   before use.
-- Added slurm_xlator.h to jobacct_gather plugins so they resolve symbols
   correctly when linking to the slurm api.
```

- You can now update a jobs QOS from scontrol. Previously you could only do this from svview.
- BLUEGENE - Fixed bug where if running in non-dynamic mode sometimes the start time returned for a job when using test-only would not be correct.

* Changes in SLURM 2.1.9

=====

- In select/linear - Fix logic to prevent over-subscribing memory with shared nodes (Shared=YES or Shared=FORCE).
- Fix for handling -N and --ntasks-per-node without specifying -n with salloc and sbatch.
- Fix jobacct_gather/linux if not polling on tasks to give tasks time to start before doing initial gather.
- When changing priority with the multifactor plugin we make sure we update the last_job_update variable.
- Fixed svview for gtk < 2.10 to display correct debug level at first.
- Fixed svview to not select too fast when using a mouse right click.
- Fixed sacct to display correct timelimits for jobs from accounting.
- Fixed sacct when running as root by default query all users as documented.
- In proctrack/linuxproc, skip over files in /proc that are not really user processes (e.g. "/proc/bus").
- Fix documentation bug for slurmdbd.conf
- Fix slurmctld to update qos preempt list without restart.
- Fix bug in select/cons_res that in some cases would prevent a preempting job from using of resources already allocated to a preemptable running job.
- Fix for sreport in interactive mode to honor parsable/2 options.
- Fixed minor bugs in sacct and sstat commands
- BLUEGENE - Fixed issue if the slurmd becomes unresponsive and you have blocks in an error state accounting is correct when the slurmd comes back up.
- Corrected documentation for -n option in srun/salloc/sbatch
- BLUEGENE - when running a willrun test along with preemption the bluegene plugin now does the correct thing.
- Fix possible memory corruption issue which can cause slurmctld to abort.
- BLUEGENE - fixed small memory leak when setting up env.
- Fixed deadlock if using accounting and cluster changes size in the database. This can happen if you mistakenly have multiple primary slurmctld's running for a single cluster, which should rarely if ever happen.
- Fixed sacct -c option.
- Critical bug fix in sched/backfill plugin that caused memory corruption.

* Changes in SLURM 2.1.8

=====

- Update BUILD_NOTES for AIX and bgp systems on how to get svview to build correctly.
- Update man page for scontrol when nodes are in the "MIXED" state.
- Better error messages for sacctmgr.
- Fix bug in allocation of CPUs with select/cons_res and --cpus-per-task option.
- Fix bug in dependency support for afterok and afternotok options to insure that the job's exit status gets checked for dependent jobs prior to purging completed job records.
- Fix bug in sched/backfill that could set an incorrect expected start time for a job.

Appendix G. SLURM Release Information

- BLUEGENE - Fix for systems that have midplanes defined in the database that don't exist.
- Accounting, fixed bug where if removing an object a rollback wasn't possible.
- Fix possible scontrol stack corruption when listing jobs with very a long job or working directory name (over 511 characters).
- Insure that SPANK environment variables set by salloc or sbatch get propagated to the Prolog on all nodes by setting SLURM_SPANK_* environment variables for srun's use.
- In sched/wiki2 - Add support for the MODIFYJOB command to alter a job's comment field
- When a cluster first registers with the SlurmDBD only send nodes in a non-usable state. Before all nodes were sent.
- Alter sacct to be able to query jobs by association id.
- Edit documentation for scontrol stating ExitCode as something not alterable.
- Update documentation about ReturnToService and silently rebooting nodes.
- When combining --ntasks-per-node and --exclusive in an allocation request the correct thing, giving the allocation the entire node but only ntasks-per-node, happens.
- Fix accounting transaction logs when deleting associations to put the ids instead of the lfts which could change over time.
- Fix support for salloc, sbatch and srun's --hint option to avoid allocating a job more sockets per node or more cores per socket than desired. Also when --hint=compute_bound or --hint=memory_bound then avoid allocating more than one task per hyperthread (a change in behavior, but almost certainly a preferable mode of operation).

* Changes in SLURM 2.1.7

=====

- Modify srun, salloc and sbatch parsing for the --signal option to accept either a signal name in addition to the previously supported signal numbers (e.g. "--signal=USR2@200").
- BLUEGENE - Fixed sinfo --long --Node output for cpus on a single cnode.
- In sched/wiki2 - Fix another logic bug in support of Moab being able to identify preemptable jobs.
- In sched/wiki2 - For BlueGene systems only: Fix bug preventing Moab from being able to correctly change the node count of pending jobs.
- In select/cons_res - Fix bug preventing job preemption with a configuration of Shared=FORCE:1 and PreemptMode=GANG,SUSPEND.
- In the TaskProlog, add support for an "unset" option to clear environment variables for the user application. Also add support for embedded white-space in the environment variables exported to the user application (everything after the equal sign to the end of the line is included without alteration).
- Do not install /etc/init.d/slurm or /etc/init.d/slurmdbd on AIX systems.
- BLUEGENE - fixed check for small blocks if a node card of a midplane is in an error state other jobs can still run on the midplane on other nodecards.
- BLUEGENE - Check to make sure job killing is in the active job table in DB2 when killing the job.
- Correct logic to support ResvOverRun configuration parameter.
- Get --acctg-freq option working for srun and salloc commands.
- Fix sinfo display of drained nodes correctly with the summarize flag.
- Fix minor memory leaks in slurmd and slurmdstepd.
- Better error messages for failed step launch.


```
-- Modify srun to insure compatability of the --relative option with the node
count requested.
```

* Changes in SLURM 2.1.6-2

```
=====
```

```
-- In sched/wiki2 - Fix logic in support of Moab being able to identify
preemptable jobs.
-- Applied fixes to a debug4 message in priority_multifactor.c sent in by
Per Lundqvist
-- BLUEGENE - Fixed issue where incorrect nodecards could be picked when
looking at combining small blocks to make a larger small block.
```

* Changes in SLURM 2.1.6

```
=====
```

```
-- For newly submitted jobs, report expected start time in squeue --start as
"N/A" rather than current time.
-- Correct sched/backfill logic so that it runs in a more timely fashion.
-- Fixed issue if running on accounting cache and priority/multifactor to
initialize the root association when the database comes back up.
-- Emulated BLUEGENE - fixed issue where blocks weren't always created
correctly when loading from state. This does not apply to a real
bluegene system, only emulated.
-- Fixed bug when job is completing and its cpu_cnt would be calculated
incorrectly, possibly resulting in an underflow being logged.
-- Fixed bug where if there are pending jobs in a partition which was
updated to have no nodes in it the slurmctld would dump core.
-- Fixed smap and svview to display partitions with no nodes in them.
-- Improve configure script's logic to detect LUA libraries.
-- Fix bug that could cause slurmctld to abort if select/cons_res is used AND a
job is submitted using the --no-kill option AND one of the job's nodes goes
DOWN AND slurmctld restarts while that job is still running.
-- In jobcomp plugins, job time limit was sometimes recorded improperly if not
set by user (recorded huge number rather than partition's time limit).
```

* Changes in SLURM 2.1.5

```
=====
```

```
-- BLUEGENE - Fixed display of draining nodes for sinfo -R.
-- Fixes to scontrol and svview when setting a job to an impossible start time.
-- Added -h to srun for help.
-- Fix for sacctmgr man page to remove erroneous 'with' statements.
-- Fix for unpacking jobs with accounting statistics, previously it appears
only steps were unpacked correctly, for the most case sacct would only
display this information making this fix a very minor one.
-- Changed scontrol and svview output for jobs with unknown end times from
'NONE' to 'Unknown'.
-- Fixed mysql plugin to reset classification when adding a
previously deleted cluster.
-- Permit a batch script to reset umask and have that propagate to tasks
spawed by subsequent srun. Previously the umask in effect when sbatch was
executed was propagated to tasks spawed by srun.
-- Modify slurm_job_cpus_allocated_on_node_id() and
slurm_job_cpus_allocated_on_node() functions to not write explanation of
failures to stderr. Only return -1 and set errno.
-- Correction in configurator.html script. Prolog and Epilog were reversed.
-- BLUEGENE - Fixed race condition where if a nodecard has an error on an
```

Appendix G. SLURM Release Information

un-booted block when a job comes to use it before the state checking thread notices it which could cause the slurmd to lock up on a non-dynamic system.

- In select/cons_res with FastSchedule=0 and Procs=# defined for the node, but no specific socket/core/thread count configured, avoid fatal error if the number of cores on a node is less than the number of Procs configured.
- Added ability for the perlapi to utilize opaque data types returned from the C api.
- BLUEGENE - made the perlapi get correct values for cpus per node, Previously it would give the number of cpus per cnode instead of midplane.
- BLEUGENE - Fixed issue where if a block being selected for a job to use and during the process a hardware failure happens, previously the block would still be allowed to be used which would fail or requeue the job depending on the configuration.
- For SPANK job environment, avoid duplicate "SPANK_" prefix for environment set by sbatch jobs.
- Make squeue select jobs on hidden partitions when requesting more than one.
- Avoid automatically cancelling job steps when all of the tasks on some node have gracefully terminated.

* Changes in SLURM 2.1.4

=====

- Fix for purge script in accounting to use correct options.
- If SelectType=select/linear and SelectTypeParameters=CR_Memory fix bug that would fail to release memory reserved for a job if "scontrol reconfigure" is executed while the job is in completing state.
- Fix bug in handling event trigger for job time limit while job is still in pending state.
- Fixed display of Ave/MaxCPU in sacct for jobs. Steps were printed correctly.
- When node current features differs from slurm.conf, log the node names using a hostlist expression rather than listing individual node names.
- Improve ability of srun to abort job step for some task launch failures.
- Fix mvapich plugin logic to release the created job allocation on initialization failure (previously the failures would cancel job step, but retain job allocation).
- Fix bug in srun for task count so large that it overflows int data type.
- Fix important bug in select/cons_res handling of ntasks-per-core parameter that was uncovered by a bug fixed in v2.1.3. Bug produced fatal error for slurmd: "cons_res: cpus computation error".
- Fix bug in select/cons_res handling of partitions configured with Shared=YES. Prior logic failed to support running multiple jobs per node.

* Changes in SLURM 2.1.3-2

=====

- Modified spec file to obsolete pam_slurm when installing the slurm-pam_slurm rpm.

* Changes in SLURM 2.1.3-1

=====

- BLUEGENE - Fix issues on static/overlap systems where if a midplane was drained you would not be able to create new blocks on it.
- In sched/wiki2 (for Moab): Add excluded host list to job information using new keyword "EXCLUDE_HOSTLIST".
- Correct slurmd reporting of incorrect socket/core/thread counts.

- For sched/wiki2 (Moab): Do not extend a job's end time for suspend/resume or startup delay due to node boot time. A job's end time will always be its start time plus time limit.
- Added build-time option (to configure program) of --with-pam_dir to specify the directory into which PAM modules get installed, although it should pick the proper directory by default. "make install" and "rpmbuild" should now put the pam_slurm.so file in the proper directory.
- Modify PAM module to link against SLURM API shared library and use exported slurm_hostlist functions.
- Do not block new jobs with --immediate option while another job is in the process of being requeued (which can take a long time for some node failure modes).
- For topology/tree, log invalid hostnames in a single hostlist expression rather than one per line.
- A job step's default time limit will be UNLIMITED rather than partition's default time limit. The step will automatically be cancelled as part of the job termination logic when the job's time limit is reached.
- sacct - fixed bug when checking jobs against a reservation
- In select/cons_res, fix support for job allocation with --ntasks_per_node option. Previously could allocate too few CPUs on some nodes.
- Adjustment made to init message to the slurmdbd to allow backwards compatibility with future 2.2 release. YOU NEED TO UPGRADE SLURMDBD BEFORE ANYTHING ELSE.
- Fix accounting when comment of down/drained node has double quotes in it.

* Changes in SLURM 2.1.2

=====

- Added nodelist to sview for jobs on non-bluegene systems
- Correction in value of batch job environment variable SLURM_TASKS_PER_NODE under some conditions.
- When a node silently fails which is already drained/down the reason for draining for the node is not changed.
- Srun will ignore SLURM_NNODES environment variable and use the count of currently allocated nodes if that count changes during the job's lifetime (e.g. job allocation uses the --no-kill option and a node goes DOWN, job step would previously always fail).
- Made it so sacctmgr can't add blank user or account. The MySQL plugin will also reject such requests.
- Revert libpmi.so version for compatibility with SLURM version 2.0 and earlier to avoid forcing applications using a specific libpmi.so version to rebuild unnecessarily (revert from libpmi.so.21.0.0 to libpmi.so.0.0.0).
- Restore support for a pending job's constraints (required node features) when slurmctld is restarted (internal structure needed to be rebuilt).
- Removed checkpoint_blcr.so from the plugin rpm in the slurm.spec since it is also in the blcr rpm.
- Fixed issue in sview where you were unable to edit the count of jobs to share resources.
- BLUEGENE - Fixed issue where tasks on steps weren't being displayed correctly with scontrol and sview.
- BLUEGENE - fixed wiki2 plugin to report correct task count for pending jobs.
- BLUEGENE - Added /etc/ld.so.conf.d/slurm.conf to point to the directory holding libsched_if64.so when building rpms.
- Adjust get_wckey call in slurmdbd to allow operators to list wckey.

Appendix G. SLURM Release Information

* Changes in SLURM 2.1.1

=====

- Fix for case sensitive databases when a slurmctld has a mixed case clustername to lower case the string to easy compares.
- Fix squeue if job is completing and failed to print remaining nodes instead of failed message.
- Fix svview core when searching for partitions by state.
- Fixed setting the start time when querying in sacct to the beginning of the day if not set previously.
- Defined slurm_free_reservation_info_msg and slurm_free_topo_info_msg in common/slurm_protocol_defs.h
- Avoid generating error when a job step includes a memory specification and memory is not configured as a consumable resource.
- Patch for small memory leak in src/common/pluginstack.c
- Fix svview search on node state.
- Fix bug in which improperly formed job dependency specification can cause slurmctld to abort.
- Fixed issue where slurmctld wouldn't always get a message to send cluster information when registering for the first time with the slurmdbd.
- Add slurm*_trigger.3 man pages for event trigger APIs.
- Fix bug in job preemption logic that would free allocated memory twice.
- Fix spelling issues (from Gennaro Oliva)
- Fix issue when changing parents of an account in accounting all children weren't always sent to their respected slurmctlds until a restart.
- Restore support for srun/salloc/sbatch option --hint=nomultithread to bind tasks to cores rather than threads (broken in slurm v2.1.0-pre5).
- Fix issue where a 2.0 sacct could not talk correctly to a 2.1 slurmdbd.
- BLUEGENE - Fix issue where no partitions have any nodes assigned them to alert user no blocks can be created.
- BLUEGENE - Fix smap to put BGP images when using -Dc on a Blue Gene/P system.
- Set SLURM_SUBMIT_DIR environment variable for srun and salloc commands to match behavior of sbatch command.
- Report WorkDir from "scontrol show job" command for jobs launched using salloc and srun.
- Update correctly the wckey when changing it on a pending job.
- Set wckeyid correctly in accounting when cancelling a pending job.
- BLUEGENE - critical fix where jobs would be killed incorrectly.
- BLUEGENE - fix for svview putting multiple ionodes on to nodelists when viewing the jobs tab.

* Changes in SLURM 2.1.0

=====

- Improve svview layout of blocks in use.
- A user can now change the dimensions of the grid in svview.
- BLUEGENE - improved startup speed further for large numbers of defined blocks
- Fix to _get_job_min_nodes() in wiki2/get_jobs.c suggested by Michal Novotny
- BLUEGENE - fixed issues when updating a pending job when a node count was incorrect for the asked for connection type.
- BLUEGENE - fixed issue when combining blocks that are in ready states to make a larger block from those or make multiple smaller blocks by splitting the larger block. Previously this would only work with block in a free state.
- Fix bug in wiki(2) plugins where if HostFormat=2 and the task list is

```
greater than 64 we don't truncate. Previously this would mess up Moab
by sending a truncated task list when doing a get jobs.
-- Added update slurmctld debug level to sview when in admin mode.
-- Added logic to make sure if enforcing a memory limit when using the
jobacct_gather plugin a user can no longer turn off the logic to enforce
the limit.
-- Replaced many calls to getpwuid() with reentrant uid_to_string()
-- The slurmstepd will now refresh it's log file handle on a reconfig,
previously if a log was rolled any output from the stepd was lost.
```

Notes

1. <http://slurm.schedmd.com/>

