Release Notes

Scyld ClusterWare Release 7.3.6-736g0000

About This Release

Scyld ClusterWare Release 7.3.6-736g0000 (released August 7, 2017) is the latest update to Scyld ClusterWare 7.

Scyld ClusterWare 7.3.6 expects to execute in a Red Hat RHEL7 Update 3 or CentOS 7.3 base distribution environment, each having been updated to the latest RHEL7/CentOS7 errata (https://rhn.redhat.com/errata/rhel-server-7-errata.html) as of the Scyld ClusterWare 7.3.6 release date. Any compatibility issues between Scyld ClusterWare 7.3.6 and RHEL7 are documented on the Penguin Computing Support Portal at http://www.penguincomputing.com/support.

Visit https://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux to view the Red Hat Enterprise Linux 7 *7.3 Release Notes* and other useful documents, including the *Migration Planning Guide* and *System Administrator's Guide*.

For the most up-to-date product documentation and other helpful information, visit the Penguin Computing Support Portal.

Important

Before continuing, make sure you are reading the most recent Scyld ClusterWare *Release Notes*, which can be found on the Penguin Computing Support Portal at http://www.penguincomputing.com/support/documentation. The most recent version will accurately reflect the current state of the Scyld ClusterWare yum repository of rpms that you are about to install. You may consult the *Installation Guide* for its more generic and expansive details about the installation process. The *Release Notes* document more specifically describes how to upgrade an earlier version of Scyld ClusterWare to Scyld ClusterWare 7.3.6 (see the Section called *Upgrading An Earlier Release of Scyld ClusterWare 7 to 7.3*), or how to install Scyld ClusterWare 7.3.6 as a fresh install (see the Section called *First Installation of Scyld ClusterWare 7 On A Server*).

Important for clusters using 3rd-party drivers or applications

Before installing or updating Scyld ClusterWare, if your cluster uses any 3rd-party drivers (e.g., Ethernet, InfiniBand, GPU, parallel storage) and if an install or update includes a new kernel, then verify that those 3rd-party drivers can be rebuilt or relinked to the new kernel. If an install or update involves upgrading to a new RHEL7 or CentOS7 base distribution, then verify that your cluster's 3rd-party applications are all supported by that new base distribution.

First Installation of Scyld ClusterWare 7 On A Server

When installing Scyld ClusterWare 7 on a system that does not yet contain Scyld ClusterWare, you should perform the following steps:

1. The directory `/etc/yum.repos.d/` must contain active repo config files bearing a suffix of `.repo`. If there is no ClusterWare repo file, then you should download `clusterware.repo` that gives your cluster access to the customer-facing Scyld ClusterWare yum repos.

   To download a yum repo file that is customized to your cluster:

   a. Login to the Penguin Computing Support Portal at http://www.penguincomputing.com/support.

   b. Click on the tab labeled *Assets*, and then select a specific *Asset Name* in the list.

   c. In the *Asset Detail* section, click on *YUM Repo File*, which downloads an asset-specific `clusterware.repo` file, and move that file to the `/etc/yum.repos.d/` directory.

   d. Set the permissions: `chmod 644 /etc/yum.repos.d/clusterware.repo`

e. The new `clusterware.repo` contains a *baseurl* entry that uses `https` by default. If your local site is configured to not support such encrypted accesses, then you must edit the repo file to instead use `http`.

The file contains three sections, labeled *cw-core*, *cw-updates*, and *cw-next*. Generally, the *cw-next* repo should not be enabled unless so directed by Penguin Computing Support.

2. Examine `/etc/yum.repos.d/clusterware.repo` to ensure that it specifies the desired yum repository release version. Employ *$releasever* or *7* to use rpms from the latest Scyld ClusterWare release, which currently is 7.3. Alternatively, a more specific major-minor pair, e.g., *7.2*, limits the rpms to just that version, even as ClusterWare releases march forward to newer versions.
3. If updating using a Red Hat yum repo, then your Red Hat yum configuration file should also look in the Red Hat Server Optional repo to find rpms such as `compat-dapl-devel` and `sharutils`. The regular CentOS7 yum repo contains these rpms.
4. Install a useful Scyld ClusterWare script that simplifies installing (and later updating) software, then execute that script:

```
yum install install-scyld
install-scyld
```

5. Configure the network for Scyld ClusterWare: edit `/etc/beowulf/config` to specify the cluster interface, the maximum number of compute nodes, and the beginning IP address of the first compute node. See the *Installation Guide* for more details.
6. If the private cluster network switch uses Spanning Tree Protocol (STP), then either reconfigure the switch to disable STP, or if that is not feasible because of network topology, then enable *Rapid STP* or *portfast* on the compute node and edge ports. See the Section called *Issues with Spanning Tree Protocol and portfast* for details.
7. Reboot the master node.
8. After rebooting the new kernel, and after installing any new kernel modules, you should rebuild the master node's list of modules and dependencies using **depmod**. See the Section called *Issues with kernel modules* for details.

Upgrading An Earlier Release of Scyld ClusterWare 7 to 7.3

If you wish to upgrade a RHEL6 (or CentOS6) or earlier base distribution to RHEL7/CentOS7, then we recommend you accomplish this with a full install of Release 7, rather than attempt to *update* from an earlier major release to Release 7. Visit https://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux for the Red Hat Enterprise Linux 7 *Installation Guide* for details. If you already have installed Scyld ClusterWare 6 (or earlier) on the physical hardware that you intend to convert to RHEL7/CentOS7, then we recommend that you backup your master node prior to the new installation of RHEL7/CentOS7, as some of the Scyld ClusterWare configuration files may be a useful reference for Release 7, especially files in `/etc/beowulf/`.

When upgrading from an earlier Scyld ClusterWare 7 version to a newer Scyld ClusterWare 7, you should perform the following steps:

1. Examine `/etc/yum.repos.d/clusterware.repo` to ensure that it specifies the desired yum repository release version. Employ *$releasever* or *7* to use rpms from the latest Scyld ClusterWare release, which currently is 7.3. Alternatively, a more specific major-minor pair, e.g., *7.2*, limits the rpms to just that version, even as ClusterWare releases march forward to newer versions.
2. Consider whether or not to stop the cluster prior to updating software. Most updates can be made to a running cluster, although some updates (e.g., those affecting daemons that execute on the master node) require a subsequent restart of the ClusterWare service. Other updates require rebooting the master node, in particular when updating to a new kernel,

and this obviously restarts the cluster nodes, too. The safest approach is to stop the cluster before updating the master node, and restart the cluster after the update completes.

```
systemctl stop clusterware
```

3. Update the software on the master node using the `install-scyld` script that guides you through the process, step by step.

   ```
   install-scyld -u
   ```

   The script first determines if it needs to update itself. If that self-update occurs, then the script exits and you should re-execute it.

4. Compare `/etc/beowulf/config`, which remains untouched by the Scyld ClusterWare update, with the new `config.rpmnew` (if that file exists), examine the differences:

   ```
   cd /etc/beowulf
   diff config config.rpmnew
   ```

   and carefully merge the `config.rpmnew` differences into `/etc/beowulf/config`. See the Section called *Resolve *.rpmnew and *.rpmsave configuration file differences* for details.

   Similarly, the preexisting `/etc/beowulf/fstab` may have been saved as `fstab.rpmsave` if it was locally modified. If so, merge those local changes back into `/etc/beowulf/fstab`.

5. If a new kernel has been installed, then reboot the master node. Otherwise, simply reboot the ClusterWare service:

   ```
   systemctl restart clusterware
   ```

6. After rebooting a new kernel, and after installing any new kernel modules, you should rebuild the master node's list of modules and dependencies using **depmod**. See the Section called *Issues with kernel modules* for details.

Post-Installation Configuration Issues

Following a successful update or install of Scyld ClusterWare, you may need to make one or more configuration changes, depending upon the local requirements of your cluster. Larger cluster configurations have additional issues to consider; see the Section called *Post-Installation Configuration Issues For Large Clusters*.

Resolve **.rpmnew* and **.rpmsave* configuration file differences

As with every Scyld ClusterWare upgrade, after the upgrade you should locate any Scyld ClusterWare `*.rpmsave` and `*.rpmnew` files and perform merges, as appropriate, to carry forward the local changes. Sometimes an upgrade will save the locally modified version as `*.rpmsave` and overwrite the basic file with a new version. Other times the upgrade will keep the locally modified version untouched, installing the new version as `*.rpmnew`.

For example,

```
cd /etc/beowulf
find . -name \*rpmnew
find . -name \*rpmsave
```

and examine each such file to understand how it differs from the configuration file that existed prior to the update. You may need to merge new lines from the newer `*.rpmnew` file into the existing file, or perhaps replace existing lines with new modifications. For instance, this is commonly done with `/etc/beowulf/config` and `config.rpmnew`. Or you may need to merge older local modifications in `*.rpmsave` into the newly installed pristine version of the file. For instance, this is occasionally done with `/etc/beowulf/fstab.rpmsave`.

Generally speaking, be careful when making changes to `/etc/beowulf/config`, as mistakes may leave your cluster in a non-working state. In particular, take care when modifying the keyword entries for *interface*, *nodes*, *iprange*, and *nodeassign*. The *kernelimage* and *node* entries are automatically managed by ClusterWare services and should not be merged.

The remaining differences are candidates for careful merging. Pay special attention to merge additions to the *bootmodule*, *modarg*, *server*, *libraries*, and *prestage* keyword entries. New *nodename* entries for *infiniband* or *ipmi* are offsets to each node's IP address on the private cluster network, and these offsets may need to be altered to be compatible with your local network subnet. Also, be sure to merge differences in `config.rpmnew` comments, as those are important documentation information for future reference.

Contact Scyld Customer Support if you are unsure about how to resolve particular differences, especially with `/etc/beowulf/config`.

Disable SELinux and NetworkManager

Scyld ClusterWare execution currently requires that SELinux and NetworkManager services be disabled. The **install-scyld** script performs this disabling.

Optionally reduce size of /usr/lib/locale/locale-archive

Glibc applications silently open the file `/usr/lib/locale/locale-archive`, which means it gets downloaded by each compute node early in a node's startup sequence. The default RHEL7 `locale-archive` is about 100 MBytes in size, thus consuming significant network bandwidth and potentially causing serialization delays if numerous compute nodes attempt to concurrently boot, and consuming significant RAM filesystem space on each node. It is likely that a cluster's users and applications do not require all the international locale data that is present in the default file. With care, the cluster administrator may choose to rebuild `locale-archive` with a greatly reduced set of locales and thus create a significantly smaller file. See the *Administrator's Guide* for details.

Optionally configure and enable compute node CPU speed/power management

Modern motherboards and processors support a degree of administrator management of CPU frequency within a range defined by the motherboard's BIOS. Scyld ClusterWare provides the `/etc/beowulf/init.d/30cpuspeed` script and its associated `/etc/beowulf/conf.d/cpuspeed.conf` configuration file to implement this management for compute nodes. The local cluster administrator is encouraged to review the *Administrator's Guide*'s *Configuring CPU speed/power for Compute Nodes* for details.

Optionally install a different TORQUE package

TORQUE is available in several versions: `torque-4-scyld` (which is the current default) and `torque-4-nocpuset-scyld` provide version 4, `torque-5-scyld` and `torque-5-nocpuset-scyld` provide version 5, and `torque-6-scyld` and `torque-6-nocpuset-scyld` provide version 6.

The `nocpuset` packages specifically disable the default *cpuset* functionality that optionally allows an application to constrain the movement of software threads between CPUs within a node in order to achieve optimal performance. See http://docs.adaptivecomputing.com/torque/4-1-4/help.htm#topics/3-nodes/linuxCpusetSupport.htm for details.

One, and only one, TORQUE must be installed at any one time. Since each TORQUE package specifies a list of package dependencies that should not be removed when uninstalling the existing TORQUE package, care must be taken to retain those dependencies when switching from one version of TORQUE to another. For example, to switch from `torque-4-scyld` to `torque-4-nocpuset-scyld`:

```
rpm -e --nodeps torque-4-scyld
yum install torque-4-nocpuset-scyld
```

Optionally enable job manager

The default Scyld ClusterWare installation includes two job managers: TORQUE and Slurm. TORQUE is available in several versions. See the Section called *Optionally install a different TORQUE package* for important details. Both Slurm and one, and only one, of these TORQUE versions must be installed on the master node, although only Slurm *or* one of the TORQUE versions may be enabled and executing at any one time.

To enable TORQUE, then after all compute nodes are up and running, you must first disable SLURM, then enable and configure TORQUE, then reboot all the compute nodes:

```
slurm-scyld.setup cluster-stop
beochkconfig 98slurm off
slurm-scyld.setup disable
beochkconfig 98torque on
torque-scyld.setup reconfigure      # when needed
torque-scyld.setup enable
torque-scyld.setup cluster-start
torque-scyld.setup status
bpctl -S all -R
```

To enable Slurm, then after all compute nodes are up and running, you must first disable TORQUE, then enable and configure Slurm, then reboot all the compute nodes:

```
torque-scyld.setup cluster-stop
beochkconfig 98torque off
torque-scyld.setup disable
beochkconfig 98slurm on
slurm-scyld.setup reconfigure      # when needed
slurm-scyld.setup enable
slurm-scyld.setup cluster-start
slurm-scyld.setup status
bpctl -S all -R
```

See the *Administrator's Guide* for more details about TORQUE configuration, and the *User's Guide* for details about how to use TORQUE.

Each Slurm user 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`.

See the *Administrator's Guide* for more details about TORQUE and Slurm configuration.

Optionally enable TORQUE scheduler

Scyld ClusterWare installs by default both the TORQUE resource manager and the associated Maui job scheduler. The Maui installation can coexist with an optionally licensed Moab job scheduler installation, although after the initial installation of either of these job schedulers, the cluster administrator needs to make a one-time choice of which job scheduler to employ.

If Moab is not installed, and if TORQUE is enabled as the operative job manager (see the Section called *Optionally enable job manager*), then simply activate Maui by moving into place two global profile files that execute **module load maui** and then start the *maui* service:

```
cp /opt/scyld/maui/scyld.maui.{csh,sh} /etc/profile.d
chkconfig maui on
service maui start
```

If Moab was previously installed, is currently active, and is the preferred job scheduler, then the cluster administrator can ignore the Maui installation (and any subsequent Maui updates) because Maui installs in a deactivated state and will not affect Moab.

If Maui is active and the cluster administrator subsequently installs Moab, or chooses to use an already installed Moab as the default scheduler, then deactivate Maui so as to not affect Moab:

```
rm /etc/profile.d/scyld.maui.*
chkconfig maui off
service maui stop
```

and then activate Moab as appropriate for the cluster.

Optionally enable Ganglia monitoring tool

To enable the Ganglia cluster monitoring tool,

```
chkconfig beostat on
systemctl enable xinetd
systemctl enable httpd
systemctl enable gmetad
```

then either reboot the master node, which automatically restarts these system services; or without rebooting, manually restart *xinetd* then start the remaining services that are not already running:

```
systemctl restart xinetd
systemctl start httpd
systemctl start gmetad
```

See the *Administrator's Guide* for more details.

Optionally enable NFS locking

To enable cluster-wide NFS locking for compute node clients, edit `/etc/beowulf/fstab` (or the appropriate node-specific `/etc/beowulf/fstab.`*N* file(s)) to remove the default option *nolock* for that mountpoint. See the *Administrator's Guide* for more details.

Optionally adjust the size limit for locked memory

OpenIB, MVAPICH, and MVAPICH2 require an override to the limit of how much memory can be locked.

Scyld ClusterWare adds a *memlock* override entry to `/etc/security/limits.conf` during a Scyld ClusterWare upgrade (if the override entry does not already exist in that file), regardless of whether or not Infiniband is present in the cluster. The new override line,

```
*    -    memlock    unlimited
```

raises the limit to *unlimited*. If Infiniband is not present, then this new override line is unnecessary and may be deleted. If Infiniband is present, we recommend leaving the new *unlimited* line in place. If you choose to experiment with a smaller discrete value, then understand that Scyld ClusterWare MVAPICH requires a minimum of 16,384 KBytes, which means changing *unlimited* to *16384*. If your new discrete value is too small, then MVAPICH reports a "CQ Creation" or "QP Creation" error.

Optionally increase the max number of processes per user

RHEL7 defaults to a maximum of 4096 processes per user, as specified in `/etc/security/limits.d/20-nproc.conf`, which contrasts with the RHEL5 default of 16,384 and the RHEL6 default of 1024. If this RHEL7 value is too low, then override the *nproc* entry in that file, as appropriate for your cluster workload needs. Use a discrete value, not *unlimited*.

Optionally enable SSHD on compute nodes

If you wish to allow users to execute MVAPICH2 applications, or to use **/usr/bin/ssh** or **/usr/bin/scp** from the master to a compute node, or from one compute node to another compute node, then you must enable **sshd** on compute nodes by enabling the script:

```
beochkconfig 81sshd on
```

The cluster is preconfigured to allow user *root* ssh access to compute nodes. The cluster administrator may wish to configure the cluster to allow ssh access for non-root users. See the *Administrator's Guide* for details.

Optionally allow IP Forwarding

By default, the master node does not allow IP Forwarding from compute nodes on the private cluster network to external IP addresses on the public network. If IP Forwarding is desired, then edit `/etc/beowulf/config` to enable the directive *ipforward yes*, and ensure that the file `/etc/sysconfig/iptables` eliminates or comments-out the default entry:

```
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
```

Optionally increase the nf_conntrack table size

Certain workloads may trigger a syslog message *nf_conntrack: table full, dropping packet*. At cluster startup, Scyld ClusterWare insures a NAT table max size of at least 524,288. However, this max value may still be inadequate for local workloads, and the *table full, dropping packet* syslog messages may still occur. Use:

```
sysctl -n net.nf_conntrack_max
```

to view the current max size, then keep manually increasing the max until the syslog messages stop occurring, e.g., use:

```
sysctl -w net.nf_conntrack_max=Nmax
```

to try new *Nmax* values. Make this value persist across master node reboots by adding:

```
net.nf_conntrack_max=Nmax
```

to `/etc/sysctl.conf`.

Optionally configure vm.zone_reclaim_mode on compute nodes

Because Scyld ClusterWare compute nodes are predominantly used for High Performance Computing, versus (for example) used as file servers, we suggest that the `/etc/beowulf/conf.d/sysctl.conf` file contain the line:

```
vm.zone_reclaim_mode=1
```

for optimal NUMA performance. Scyld ClusterWare's `node_up` script adds this line if it doesn't already exist, but will not alter an existing *vm.zone_reclaim_mode* declaration in that file. If the file `/etc/beowulf/conf.d/sysctl.conf` does not exist, then `node_up` creates it by replicating the master node's `/etc/sysctl.conf`, which may contain a *vm.zone_reclaim_mode=N* declaration that is perhaps not *=1* and thus not optimal for compute nodes, even if the value is optimal for the master node. In this case, the cluster administrator should consider manually editing `/etc/beowulf/conf.d/sysctl.conf` to change the line to *vm.zone_reclaim_mode=1*.

Optionally configure automount on compute nodes

If you wish to run automount from compute nodes, you must first set up all the necessary configuration files in `/etc/beowulf/conf.d/autofs/` before enabling the `/etc/beowulf/init.d/50autofs` script. These config files are similar to those normally found on a server in `/etc/`, such as `/etc/auto.master`, as the `50autofs` script copies the files in `/etc/beowulf/conf.d/autofs/` to each compute node's `/etc/`.

A default `/etc/beowulf/conf.d/autofs/auto.master` must exist. All automount config files that are listed in that `master.conf`, such as `/etc/auto.misc`, `/etc/auto.net`, etc., should also reside in `/etc/beowulf/conf.d/autofs/`.

Node-specific config files (`auto.master` and related `auto.*`) may reside in `/etc/beowulf/conf.d/autofs/$NODE/`. Those files override the default top level `/etc/beowulf/conf.d/auto.master`, etc., for the specific $NODE.

The `50autofs` script parses the config files as mentioned above. It creates mount point directories, installs the autofs4 kernel module, and starts **automount** on each booting compute node. The script exits with a warning if there are missing config files.

NOTE: This script does *not* validate the correctness of potential future automount mount requests (i.e., those described in the various `auto.*` config files). The cluster administrator should set up the config files, then enable `50autofs` and reboot one or a limited number of nodes and ensure that each potential automount will function properly prior to rebooting all compute nodes. Common failures include naming an unknown server or attempting to mount a directory that has not been properly exported by the server. Mount failures will be syslogged in `/var/log/messages`.

Optionally reconfigure node names

You may declare site-specific alternative node names for cluster nodes by adding entries to `/etc/beowulf/config`. The syntax for a node name entry is:

```
nodename format-string [IPv4offset] [netgroup]
```

For example,

```
nodename node%N
```

allows the user to refer to node 4 using the traditional *.4* name, or alternatively using names like *node4* or *node004*. See **man beowulf-config** and the *Administrator's Guide* for details.

Post-Installation Configuration Issues For Large Clusters

Larger clusters have additional issues that may require post-installation adjustments.

Optionally increase the number of nfsd threads

The default count of 8 **nfsd** NFS daemons may be insufficient for large clusters. One symptom of an insufficiency is a syslog message, most commonly seen when you currently boot all the cluster nodes:

```
nfsd: too many open TCP sockets, consider increasing the number of nfsd threads
```

Scyld ClusterWare automatically increases the nfsd thread count to at least one thread per compute node, with a lowerbound of eight (for <=8 nodes) and an upperbound of 64 (for >=64 nodes). If this increase is insufficient, then increase the thread count (e.g., to 16) by executing:

```
echo 16 > /proc/fs/nfsd/threads
```

Ideally, the chosen thread count should be sufficient to eliminate the syslog complaints, but not significantly higher, as that would unnecessarily consume system resources. One approach is to repeatedly double the thread count until the syslog error messages stop occurring, then make the satisfactory value *N* persistent across master node reboots by creating the file `/etc/sysconfig/nfs`, if it does not already exist, and adding to it an entry of the form:

```
RPCNFSDCOUNT=N
```

A value *N* of 1.5x to 2x the number of nodes is probably adequate, although perhaps excessive. See the *Administrator's Guide* for a more detailed discussion of NFS configuration.

Optionally increase the max number of processID values

The kernel defaults to using a maximum of 32,768 processID values. Scyld ClusterWare automatically increases this default to 98,304 [= 3*32768], which likely is adequate for small- to medium-size clusters and which keeps pid values at a familiar 5-column width maximum. Because BProc manages a common process space across the cluster, even the increase to 98,304 may be insufficient for very large clusters and/or workloads that create large numbers of concurrent processes. The cluster administrator can increase the value further by using the **sysctl** command, e.g.,

```
sysctl -w kernel.pid_max=N
```

directs the kernel to use pid values up to *N*. The kernel (and BProc) supports an upperbound of 4,194,304 [= (4*1024*1024)]. To set a value *N* that persists across master node reboots, add an entry

```
kernel.pid_max=N
```

to `/etc/sysctl.conf`.

Optionally increase the max number of open files

RHEL7 defaults to a maximum of 1024 concurrently open files. This value may be too low for large clusters. The cluster administrator can add a *nofile* override entry to `/etc/security/limits.conf` to specify a larger value. Caution: for *nofile*, use only a numeric upperbound value, never *unlimited*, as that will result in being unable to login.

Issues with Ganglia

The Ganglia cluster monitoring tool may fail for large clusters. If the `/var/log/httpd/error_log` shows a fatal error of the form *PHP Fatal error: Allowed memory size of 8388608 bytes exhausted*, then edit the file `/etc/php.ini` to increase the *memory_limit* parameter. The default is *memory_limit = 8M* can be safely doubled and re-doubled until the error goes away.

Post-Installation Release of Updated Packages

From time to time, Penguin Computing releases updated Scyld ClusterWare 7 rpms to track Red Hat kernel security or bug fix errata, or to fix Scyld ClusterWare problems or to introduce enhancements. Download the latest version of the Scyld ClusterWare 7 *Release Notes* from the Penguin Computing Support Portal (http://www.penguincomputing.com/services-support/documentation/) to ensure you have the latest guidance before updating your cluster.

First check for the availability of updated rpms:

```
yum check-update
```

and ascertain if the base distribution and/or Scyld ClusterWare would update to a newer kernel, or even more significantly to a new major-minor release. Upgrading the kernel will require updating, perhaps even rebuilding, any 3rd-party drivers that are installed and linked against the current kernel, and you should be prepared to do that if you proceed with the updates. Updating to a newer major-minor release may also affect 3rd-party applications that are validated only for the current base distribution release.

In general, if you choose to update software, then you should use:

```
install-scyld -u
```

and update all available packages.

Notable Feature Enhancements And Bug Fixes

New in Scyld ClusterWare 7.3.6 - Scyld Release 736g0000 - August 7, 2017

1. The base kernel is 3.10.0-514.26.2.el7.736g0000. See https://access.redhat.com/errata/RHSA-2017:1615.html and https://access.redhat.com/errata/RHBA-2017:1674.html for details.

New in Scyld ClusterWare 7.3.5 - Scyld Release 735g0000 - June 28, 2017

1. The base kernel is 3.10.0-514.21.2.el7.735g0000. See https://access.redhat.com/errata/RHSA-2017:1484.html for details.
2. Fix a memory leak that occurs when executing **bpsh**, **bprsh**, and **bpcp** commands. This memory leak could eventually lead to an out-of-memory (OOM) kernel panic.
3. Update `/lib64/scyld/libc-2.17*` to match the latest `/lib64/libc-2.17.so`. See https://access.redhat.com/security/cve/CVE-2017-1000366 for details.
4. Singularity updates to version 2.3.1. See http://singularity.lbl.gov/ and the Scyld ClusterWare *User's Guide* for details.

5. The Slurm job manager updates to version 17.02.5, derived from http://slurm.schedmd.com. See the *User's Guide* Appendix G, *SLURM Release Information* for details.

New in Scyld ClusterWare 7.3.4 - Scyld Release 734g0000 - June 16, 2017

1. The base kernel is 3.10.0-514.21.1.el7.734g0000. See https://access.redhat.com/errata/RHSA-2017:1308.html for details.
2. TORQUE 6 updates to version 6.1.1.1, from www.adaptivecomputing.com/products/open-source/torque/. See www.adaptivecomputing.com/support/documentation-index/torque-resource-manager-documentation/ for details.
3. The Slurm job manager updates to version 17.02.3, derived from http://slurm.schedmd.com. See the *User's Guide* Appendix G, *SLURM Release Information* for details.
4. The openmpi-2.1-scyld packages update to version 2.1.1, which by default update and replace only earlier version 2.1 packages and do not affect any installed OpenMPI version 2.0 and earlier packages. The openmpi-2.0-scyld packages update to version 2.0.3, which by default update and replace only earlier version 2.0 packages and do not affect any installed OpenMPI version 1.10 and earlier packages. The openmpi-1.10-scyld packages update to version 1.10.7, which by default update and replace only earlier version 1.10 packages and do not affect any installed OpenMPI version 1.8 and earlier packages. See the Section called *Installing and managing concurrent versions of packages* for general issues about supporting multiple concurrent versions. The libraries were built with Gnu version 4.8.5-11, Intel version 2013_sp1.3.174, and PGI version 14.6 compiler families. Scyld releases of OpenMPI derive from http://www.open-mpi.org/. See the *User's Guide* Appendix C, *OpenMPI Release Information* for details.
5. Singularity updates to version 2.3. See http://singularity.lbl.gov/ and the Scyld ClusterWare *User's Guide* for details.

New in Scyld ClusterWare 7.3.3 - Scyld Release 733g0000 - April 27, 2017

1. The base kernel is 3.10.0-514.16.1.el7.733g0000. See https://access.redhat.com/errata/RHSA-2017:0933.html for details. The Scyld ClusterWare kernel now includes built-in firmware to properly boot some compute node server models that employ bnx2, bnx2x, or cxgb3 Ethernet controllers.
2. The bproc *filecache* functionality now properly downloads files from the master node that were previously rejected because the files have restricted read access permissions. Now all files are downloaded to compute nodes - and, as always, downloaded files are given access permissions that are replicated from the master node.
3. The `/etc/yum.repos.d/clusterware.repo.template` file is updated to remove the # prefix characters in the *cw-next* section, thereby exposing that section, albeit with the default *enabled=0* line. The cluster administrator should make a similar removal of those # characters in the local `/etc/yum.repos.d/clusterware*repo` file(s) currently in active use.

New in Scyld ClusterWare 7.3.2 - Scyld Release 732g0000 - April 3, 2017

1. The base kernel is 3.10.0-514.10.2.el7.732g0000. See https://rhn.redhat.com/errata/RHSA-2017-0386.html for details.
2. Fix a problem of compute nodes too slowly reporting process CPU time stats to the master node, e.g., reported by the **top** command.

3. TORQUE 6 updates to version 6.1.1, from www.adaptivecomputing.com/products/open-source/torque/. See www.adaptivecomputing.com/support/documentation-index/torque-resource-manager-documentation/ for details.
4. Scyld ClusterWare now distributes openmpi-2.1-scyld packages, which are initially version 2.1.0. Installation of openmpi-2.1 does not affect any earlier OpenMPI version. The libraries were built with Gnu version 4.8.5-11, Intel version 2013_sp1.3.174, and PGI version 14.6 compiler families. Scyld releases of OpenMPI derive from http://www.open-mpi.org/. See the *User's Guide* Appendix C, *OpenMPI Release Information* for details.
5. Scyld ClusterWare now distributes Singularity, which is initially version 2.2.1. See the *User's Guide* for details.

New in Scyld ClusterWare 7.3.1 - Scyld Release 731g0000 - March 8, 2017

1. The base kernel is 3.10.0-514.6.1.el7.731g0000. See https://rhn.redhat.com/errata/RHSA-2017-0086.html for details.
2. The Slurm job manager updates to version 17.02.0, derived from http://slurm.schedmd.com. See the *User's Guide* Appendix G, *SLURM Release Information* for details.
3. The openmpi-2.0-scyld packages update to version 2.0.2, which by default update and replace only earlier version 2.0 packages. The openmpi-1.10-scyld packages update to version 1.10.6, which by default update and replace only earlier version 1.10 packages. The libraries were built with Gnu version 4.8.5-11, Intel version 2013_sp1.3.174, and PGI version 14.6 compiler families. Scyld releases of OpenMPI derive from http://www.open-mpi.org/. See the *User's Guide* Appendix C, *OpenMPI Release Information* for details, and the Section called *Installing and managing concurrent versions of packages* for general issues about supporting multiple concurrent versions of OpenMPI.
4. Fix a timing bug in beonss that exhibits itself as a syslog warning (in `/var/log/messages`) involving the parsing of `/etc/beowulf/config` *nodes* and *iprange* directives.

New in Scyld ClusterWare 7.3.0 - Scyld Release 730g0000 - January 20, 2017

1. The base kernel is 3.10.0-514.2.2.el7.730g0000. See https://rhn.redhat.com/errata/RHSA-2016-2574.html and https://rhn.redhat.com/errata/RHBA-2016-2862.html for details.
2. The Slurm job manager updates to version 16.05.8, derived from http://slurm.schedmd.com. See the *User's Guide* Appendix G, *SLURM Release Information* for details.
3. Various scripts in `/etc/beowulf/init.d/` have been renamed with different numeric prefixes in order to adjust the execution ordering: `95sudo`, `98slurm`, and `98torque`. If any of these scripts has been copied and modified locally (see the Section called *Caution when modifying Scyld ClusterWare scripts* for details), then you should rename the local copy to match the new numeric prefix.
4. TORQUE 6 now supports *cgroups* instead of *cpusets*.
5. Optional `openmpi-psm2-1.10-scyld` and `openmpi-psm2-2.0-scyld` packages are now available to download and install. These `psm2` packages work in conjunction with Intel(r) Omni-Path Architecture and its `libpsm2.so`.

New in Scyld ClusterWare 7.2.0 - Scyld Release 720g0000 - November 14, 2016

1. This is the first release of Scyld ClusterWare 7.
2. The base kernel is 3.10.0-327.36.3.el7.720g0000. See https://rhn.redhat.com/errata/RHSA-2016-2098.html for details.

Known Issues And Workarounds

The following are known issues of significance with the latest version of Scyld ClusterWare 7.3.6 and suggested workarounds.

Issues with bootmodule firmware

A Scyld compute node needs a functional Ethernet connection to the server(s) (typically the master node) that provides the kernel image, the `initrd` file, and various other files that establish the compute nodes execution environment. The kernel image and `initrd` are downloaded by the node's BIOS software, using the Ethernet controller's simple functionality. The subsequent downloads are typically done using the tcp protocol and the Ethernet controller's full functionality.

Some controllers require additional firmware, which is commonly found in `/lib/firmware/` files that are supplied in RHEL7 by the separate `kernel-firmware` package. These files are unavailable in the early node boot sequence and must be built-in to the Scyld ClusterWare kernel. The kernel builds-in firmware for various commonly seen non-Penguin Computing servers, but not all. Contact Penguin Computing Support if you desire kernel support for an non-booting compute node server.

Managing environment modules .version files

Several Scyld ClusterWare packages involve the use of environment modules. This functionality allows for users to dynamically set up a shell's user environment for subsequent compilations and executions of applications, and for viewing the manpages for commands that are associated with those compilations and executions.

The Scyld packages are found in the various `/opt/scyld/`*package/* subdirectories, and for each package there are subdirectories organized by package version number, compiler suite type, and per-version per-compiler subdirectories containing the associated scripts, libraries, executable binaries, and manpages for building and executing applications for that package. The `/opt/scyld/modulefiles/`*package/* subdirectories contain per-package per-version per-compiler files that contain various pathname strings that are prepended to the shell's $PATH, $LD_LIBRARY_PATH, and $MANPATH variables that properly find those `/opt/scyld/`*package/* scripts, libraries, executable files, and manpages.

For example, **module load mpich2/intel/1.5** sets up the environment so that the **mpicc** and **mpirun** commands build and execute MPI applications using using the Intel compiler suite and the `mpich2` libraries specifically crafted for mpich2 version 1.5. The **module load** command also understands defaults. For example, **module load mpich2/gnu** defaults to use the *gnu* compiler and the mpich2 version specified by the contents of the file `/opt/scyld/modulefiles/mpich2/gnu/.version` (if that file exists). Similarly, **module load mpich2** first looks at the contents of `/opt/scyld/modulefiles/mpich2/.version` to determine the default compiler suite, then (supposing *gnu* is that default) looks at the contents of `/opt/scyld/modulefiles/mpich2/gnu/.version` to determine which mpich2 software version to use.

As a general rule, after updating one of these Scyld packages that employs environment modules, the associated `/opt/scyld/modulefiles/`*package*'s subdirectories' `.version` files remain untouched. The responsibility for updating any `.version` file remains with the cluster administrator, presumably after consulting with users. If the contents of a `.version` points to a compiler suite or to a package version number that no longer exists, then a subsequent **module load** for that package which expects to use a default selection will fail with a message of the form:

```
ERROR:105: Unable to locate a modulefile
```

The user must then perform **module load** commands that avoid any reference to the offending `.version`, e.g., use the explicit **module load mpich2/intel/1.5**, until the cluster administrator resets the `.version` contents to the desired default. Each module-employing Scyld package installs sample files with the name `.version.`*versionNumber*.

The openmpi packages manage defaults differently. Suppose `openmpi-2.0-scyld` is currently version 2.0.1 and is updating to 2.0.2. Just as the default update behavior is to replace all 2.0.1 packages with the newer 2.0.2 packages, this openmpi-2.0 update also silently changes the `gnu`, `intel`, and `pgi` `.version` files which happen to specify the same major-minor version, e.g., those that specify version 2.0.1 are silently updated to the newer 2.0.2. If, however, the current `.version` files specify an older major-minor release, e.g., 1.10.4, then updating `openmpi-2.0-scyld` does not change any of these older major-minor `.version` specifiers.

Additionally, each set of openmpi-x.y-scyld packages maintain a major-minor symlink that points to the newest major-minor-release module file. For example, when `openmpi-2.0-scyld` version 2.0.1 is currently installed, then the `/opt/scyld/modulefiles/openmpi/gnu/2.0` symlink changes to the `2.0.1` module file. When `openmpi-2.0-scyld` updates to 2.0.2, then `/opt/scyld/modulefiles/openmpi/gnu/2.0` changes that symlink to point to the `2.0.2` module file. This convenient symlink allows for users to maintain job manager scripts that simply specify a major-minor number, e.g., **module load openmpi/intel/2.0**, that survives updates from openmpi-2.0-scyld 2.0.1 to 2.0.2 to 2.0.3, et al, versus using scripts that contain the more specific **module load openmpi/intel/2.0.1** that break when 2.0.1 packages update to 2.0.2.

Note that each compiler suite can declare a different default package version, although most commonly the cluster administrator edits the `/opt/scyld/modulefiles/`*package/compiler/* `.version` files so that for a given *package*, all compiler suites reference the same default version number.

One method to check the current package defaults is to execute:

```
cd /opt/scyld/modulefiles
module purge
module avail
for m in $(ls); do module load $m; done
module list
module purge
```

and then verify each loaded default against the **module avail** available alternatives.

Installing and managing concurrent versions of packages

Scyld ClusterWare distributes various repackaged Open Source software suites, including several variations of "MPI", e.g., `openmpi`, `mpich-scyld`, `mpich2-scyld`, `mvapich2-scyld`. Users manage the selection of which software stack to use via the **module load** command. See the Section called *Managing environment modules .version files* for details.

By default, **install-scyld -u** updates each existing package with the newest version of that package by installing the newest version and removing all earlier (i.e., lower-numbered) versions, thereby retaining only a single version of each software suite. For example, the `openmpi-2.0-scyld` packages update to the latest 2.0.x version (major 2, minor 0, version x), and the `openmpi-1.10-scyld` packages update to the latest latest 1.10.y (major 1, minor 10, version y). Thus, a default update of package `openmpi-2.0` installs the newest version 2.0.x and removes earlier versions of 2.0, leaving versions 1.10.x, 1.8.x, 1.7.x, etc. untouched.

Because Scyld ClusterWare installs a package's files into unique `/opt/scyld/`*package/version* version-specific directories, this permits multiple versions of each major-minor package to potentially co-exist on the master node, e.g., openmpi versions 2.0.2 and 2.0.1. Each such *package/version* subdirectory contains one or more *compiler* suite subdirectories, e.g., `gnu`, `intel`, and `pgi`, and each of those contain scripts, libraries, executable binaries, and manpages associated with that particular package, version, and compiler suite.

Some customers (albeit rarely) may wish to install multiple concurrent x.y.z versions for a given x.y major-minor because specific applications might only work properly when linked to a specific version, or applications might perform differently for different versions. For example, to retain openmpi version 2.0.1 prior to using **install-scyld -u** or **yum update**, which might replace those 2.0.1 packages with a newer 2.0.z version, first edit `/etc/yum.conf` to add the line:

```
exclude=openmpi-2.0-scyld*
```

which blocks **yum** from updating any and all currently installed `openmpi-2.0-scyld` packages. If the cluster administrator wishes to install (for example) the 2.0.2 packages and not disturb the 2.0.1 installation, then temporarily comment-out that `exclude=openmpi-2.0-scyld*` line and execute:

```
yumdownloader openmpi-2.0-scyld-*2.0.2*
```

and then re-enable the `exclude=` line to again protect against any inadvertant `openmpi-2.0-scyld` updates. Manually install these additional downloaded rpms using **rpm -iv** -- and *not* use **rpm -Uv** or even **yum install**, as both of those commands will remove older `openmpi-2.0-scyld` packages.

Issues with OpenMPI

Scyld ClusterWare distributes repackaged releases of the Open Source OpenMPI, derived from http://www.open-mpi.org/. The Scyld ClusterWare distributions consist of a `openmpi-`*x.y*`-scyld` base package for the latest OpenMPI version *x.y.z*, plus several compiler-environment-specific packages for `gnu`, `intel`, and `pgi`. For example, the distribution of OpenMPI non-psm2 version 2.0.1 consists of the base rpm `openmpi-2.0-scyld-2.0.1` and the various compiler-specific rpms: `openmpi-2.0-scyld-gnu-2.0.1`, `openmpi-2.0-scyld-intel-2.0.1`, and `openmpi-2.0-scyld-pgi-2.0.1`.

Scyld ClusterWare distributes versions `openmpi-2.0-scyld`, `openmpi-1.10-scyld`, and `openmpi-1.8-scyld`, as well as `openmpi-psm2-2.0-scyld` and `openmpi-psm2-1.10-scyld` for clusters using the Intel Omni-Path Architecture (OPA) networking (which also requires `hfi1-psm` rpms from the Intel OPA software bundle).

A set of `openmpi-`*x.y*`-scyld` packages installs *x.y.z* version-specific libraries, executable binaries, and manpages for each particular compiler into `/opt/scyld/openmpi/`*version/compiler* subdirectories, and installs modulefiles into `/opt/scyld/modulefiles/openmpi/`*compiler/version* files. The directory `/opt/scyld/openmpi/`*version*`/examples/` contains source code examples. The `openmpi-psm2` packages similarly install into `/opt/scyld/openmpi-psm2/` and `/opt/scyld/modulefiles/openmpi-psm2/`.

The modulefiles appends the current shell's $PATH, $LD_LIBRARY_PATH, and $MANPATH with pathnames that point to the associated compiler-specific version-specific `/opt/scyld/openmpi/`*version/compiler/* (or `/opt/scyld/openmpi-psm2/`*version/compiler/*) subdirectories. This permits multiple versions to co-exist on the master node, with each variation being user-selectable at runtime using the **module load** command.

Many customers support multiple OpenMPI versions because some applications might only work properly when linked to specific OpenMPI versions. Sometimes an application needs only to be recompiled and relinked against a newer version of the libraries. Other applications may have a dependency upon a particular OpenMPI version that a simple recompilation won't fix. The cluster administrator can specify which compiler and version is the default by manipulating the contents of the various `.version` files in the `/opt/scyld/modulefiles/openmpi/` (or `openmpi-psm2`) subdirectories. For example, a **module load openmpi** might default to specify version 1.10.4 of the gnu libraries, while **module load openmpi-psm2** might default to specify version 2.0.1 of the intel libraries, while at the same time a version-specific **module load openmpi-psm2/gnu/1.10.4** or **module load openmpi/pgi/1.8.8** allows the use of different compilers and libraries for different OpenMPI versions.

The latest Open Source release of `openmpi-2.0-scyld` is a "mandatory" install, and `openmpi-1.10-scyld`, `openmpi-1.8-scyld`, `openmpi-psm2-2.0-scyld`, and `openmpi-psm2-1.10-scyld` are "optional" and can be manually installed by the cluster administrator using (for example) **yum install openmpi-psm2-1.10-scyld-***. A subsequent **yum update** will update each and every installed `openmpi-`*x.y*`-scyld` and installed `openmpi-psm2-`*x.y*`-scyld` to the latest available version *x.y.z*. If the cluster administrator wishes to retain additional *x.y.z* releases within an *x.y* family, then instead of doing **yum update**, the administrator should **yum update --exclude=openmpi*scyld-***, then download specific rpms from the yum repo as desired using **yumdownloader**, and then manually install (not update) the rpms using **rpm -i**. Note that the use of **yumdownloader** and **rpm -i** is necessary because doing a simple (for example) **yum install**

openmpi-1.10-scyld-1.10.4 will not, in fact, execute a simple *install* and retain older 1.10.z packages. Rather, it actually executes an *update* and removes any and all older installed versions of `openmpi-1.10-scyld-1.10.z` rpms.

Issues with Scyld process migration in heterogeneous clusters

In a homogeneous cluster, all nodes (master and compute) are identical server models, including having identical amounts of RAM. In a heterogeneous cluster, the nodes are not all identical. The advantage of a homogeneous cluster is simplicity in scheduling work on the nodes, since every node is identical and interchangeable. However, in the real world, many if not most clusters are heterogeneous. Some nodes may have an attached GPU or different amounts of available RAM, or may even be different server models with different x86_64 processor technologies.

Scyld ClusterWare users have always needed to be aware of potential problems running applications on heterogeneous clusters. For example, applications expecting to employ a GPU have needed to take care to execute only on nodes with an attached GPU, and an application that is specifically compiled or linked to libraries that employ newer x86_64 instructions that are not universally understood by every x86_64 processor must ensure that the application only execute on the nodes with processors that understand those newer instructions.

However, RHEL7 heterogeneous clusters present a new set of challenges to users. The essence of the issue is this: when a software thread begins execution, some libraries (e.g., libc) make a one-time determination of which processor model is being used, and the library self-configures certain routines (e.g., strcmp) to use implementations that exploit processor model-specific instructions for optimal performance. However, if the software thread subsequently migrates to a different node in the cluster, then the thread's one-time determination state migrates to the destination node. If the destination node does not support the same x86_64 instructions that are supported by the original node, then the software thread will likely suffer a fatal "invalid opcode" trap if it attempts to execute one of these optimized library routines. Scyld ClusterWare performs such a thread migration through the use of the bproc_move() or bproc_rfork() library routines found in `libbproc`. These bproc routines are employed by the **bpcp** command.

The **bpcp** command, in Scyld ClusterWare 6.3.0 and beyond, links with a special Scyld libc that uses only generic, universally acceptable x86_64 instructions. Users may similarly link applications to this special library by adding:

```
Xlinker -rpath=/lib64/scyld
```

as a linker option.

Issues with MVAPICH2 and mpirun_rsh or mpispawn

Scyld ClusterWare has applied a workaround to **mpiexec** to fix a problem with MPICH2 and MVAPICH2 exec'ing the application executable binary across NFS. The problem is *not* fixed for launching the application using **mpirun_rsh** or **mpispawn**, which likely will result in the application hanging as it attempts to execve() the application. We strongly encourage using only **mpiexec** to launch MPICH2 and MVAPICH2 applications.

Issues with ptrace

Cluster-wide **ptrace** functionality is not yet supported in Scyld ClusterWare 7. For example, you cannot use a debugger running on the master node to observe or manipulate a process that is executing on a compute node, e.g., using **gdb -p procID**, where *procID* is a processID of a compute node process. **strace** does function in its basic form, although you cannot use the **-f** or **-F** options to trace forked children if those children move away from the parent's node.

Issues with IP Forwarding

If the *clusterware* service has started, then a subsequent **systemctl stop iptables** (or **restart**) will hang because it attempts to unload the ipt_MASQUERADE kernel module while the *clusterware* service is using (and not releasing) that module. For a workaround, edit `/etc/sysconfig/iptables-config` to change:

```
    IPTABLES_MODULES_UNLOAD="yes"
to: IPTABLES_MODULES_UNLOAD="no"
```

Issues with kernel modules

The **modprobe** command uses `/usr/lib/`\`uname -r\`/`modules.dep.bin` to determine the pathnames of the specified kernel module and that module's dependencies. The **depmod** command builds the human-readable `modules.dep` and the binary `module.dep.bin` files, and it should be executed *on the master node* after installing any new kernel module.

Executing **modprobe** on a compute node requires additional caution. The first use of **modprobe** retrieves the current `modules.dep.bin` from the master node using bproc's *filecache* functionality. Since any subsequent **depmod** on the master node rebuilds `modules.dep.bin`, then a subsequent **modprobe** on a compute node will only see the new `modules.dep.bin` if that file is copied to the node using **bpcp**, or if the node is rebooted and thereby silently retrieves the new file.

In general, you should not execute **depmod** on a compute node, since that command will only see those few kernel modules that have previously been retrieved from the master node, which means the node's newly built `modules.dep.bin` will only be a sparse subset of the master node's full `module.dep.bin`. Bproc's *filecache* functionality will always properly retrieve a kernel module from the master node, as long as the node's `module.dep.bin` properly specifies the pathname of that module, so the key is to have the node's `module.dep.bin` be a current copy of the master's file.

Issues with port numbers

Scyld ClusterWare employs several daemons that execute in cooperating pairs: a server daemon that executes on the master node, and a client daemon that executes on compute nodes. Each daemon pair communicates using tcp or udp through a presumably unique port number. By default, Scyld ClusterWare uses ports 932 (*beofs2*), 933 (*bproc*), 3045 (*beonss*), and 5545 (*beostats*). In the event that one or more of these port numbers collides with a non-Scyld ClusterWare daemon using the same port number, the cluster administrator can override Scyld ClusterWare default port numbers to use different, non-colliding unused ports using the `/etc/beowulf/config` file's *server* directive. See **man beowulf-config** and `/etc/beowulf/config` for a discussion of the *server* directive.

The official list of assigned ports and their associated services is http://www.iana.org/assignments/port-numbers, and `/etc/services` is a list shipped with your base distribution. However, the absence in either list of a specific port number is no guarantee that the port will not be used by some software on your cluster. Use **lsof -i :***portNumber* to determine if a particular port number is in active use.

A common collision is with *beofs2* port 932 or *bproc* port 933, since the **rpc.statd** or **rpc.mountd** daemons may randomly grab either of those ports before ClusterWare can grab them. However, ClusterWare automatically recognizes the conflict and tries alternative ports until it finds an unused port. If this flexible search causes problems with other daemons, you can edit `/etc/beowulf/config` to specify a tentative override value using the *server beofs2* or *server bproc* directive, as appropriate.

Less common are collisions with *beonss* port 3045 or *beostats* port 5545. The *server beonss* and *server beostats* override values are used as-specified and not adjusted by ClusterWare at runtime.

Issues with Spanning Tree Protocol and portfast

Network switches with Spanning Tree Protocol (STP) enabled will block packets received on a port for the first 30 seconds after the port comes online, giving the switch and the Spanning Tree algorithm time to determine if the device on the new link is a switch, and to determine if Spanning Tree will block or forward packets from this port. This is done to prevent "loops" which can cause packets to be endlessly repeated at a high rate and consume all network bandwidth. Each time the link goes down and comes back up, another 30-second blocking delay occurs. This delay can prevent PXE/DHCP from obtaining an IP address, or can prevent the node's initial kernel from downloading its initial root filesystem, which results in the node endlessly iterating in the early boot sequence, or can delay the node's ongoing *filecache* provisioning of libraries to the node.

We recommend disabling STP if feasible. If not feasible, then we recommend reconfiguring the switch to use *Rapid STP* or *portfast*, which avoids the 30-second delay, or employing some other port mode that will forward packets as a port comes up. There is no generic procedure for enabling these options. For Cisco switches, see http://www.cisco.com/en/US/products/hw/switches/ps700/products_tech_note09186a00800b1500.shtml. For other switch models, see the model-specific documentation.

If that reconfiguration is also not possible, you may need to increase the default Scyld ClusterWare timeout used by the node to a value safely greater than the STP delay: e.g., add *rootfs_timeout=120 getfile_timeout=120* to the `/etc/beowulf/config` *kernelcommandline* entry to increase the timeouts to 120 seconds.

Issues with Gdk

If you access a cluster master node using **ssh -X** from a workstation, some graphical commands or program may fail with:

```
Gdk-ERROR **: BadMatch (invalid parameter attributes)
  serial 798 error_code 8 request_code 72 minor_code 0
Gdk-ERROR **: BadMatch (invalid parameter attributes)
  serial 802 error_code 8 request_code 72 minor_code 0
```

Remedy this by doing:

```
export XLIB_SKIP_ARGB_VISUALS=1
```

prior to running the failing program. If this workaround is successful, then consider adding this line to `/etc/bashrc` or to `~/.bashrc`. See https://bugs.launchpad.net/ubuntu/+source/xmms/+bug/58192 for details.

Caution when modifying Scyld ClusterWare scripts

Scyld ClusterWare installs various scripts in `/etc/beowulf/init.d/` that **node_up** executes when booting each node in the cluster. Any site-local modification to one of these scripts will be lost when a subsequent Scyld ClusterWare update overwrites the file with a newer version. If a cluster administrator believes a local modification is necessary, we suggest:

1. Copy the to-be-edited original script to a file with a unique name, e.g.:

   ```
   cd /etc/beowulf/init.d
   cp 20ipmi 20ipmi_local
   ```

2. Remove the executable state of the original:

   ```
   beochkconfig 20ipmi off
   ```

3. Edit `20ipmi_local` as desired.

4. Thereafter, subsequent Scyld ClusterWare updates may install a new `20ipmi`, but that update will not re-enable the non-executable state of that script. The locally modified `20ipmi_local` remains untouched. However, keep in mind that the newer Scyld ClusterWare version of `20ipmi` may contain fixes or other changes that need to be reflected in `20ipmi_local` because that edited file was based upon an older Scyld ClusterWare version.

Caution using tools that modify config files touched by Scyld ClusterWare

Software tools exist that might make modifications to various system configuration files that Scyld ClusterWare also modifies. These tools do not have knowledge of the Scyld ClusterWare specific changes and therefore may undo or cause damage to the changes or configuration. Care must be taken when using such tools. One such example is **/usr/sbin/authconfig**, which manipulates `/etc/nsswitch.conf`.

Scyld ClusterWare modifies these system configuration files at install time:

```
/etc/exports
/etc/nsswitch.conf
/etc/security/limits.conf
/etc/sysconfig/syslog
```

Additionally, Scyld ClusterWare uses **chkconfig** to enable *nfs*.

Running nscd service on master node may cause kickbackdaemon to misbehave

The **nscd** (Name Service Cache Daemon) service executes by default on the master node, and **/usr/sbin/nscd** executes by default on each compute node via `/etc/beowulf/init.d/09nscd`. However, if this service is also enabled and executes on the master node, then it may cause the Scyld ClusterWare name service **kickbackdaemon** to misbehave.

Accordingly, when the ClusterWare service starts, it detects that the **nscd** service is running on the master node, then ClusterWare automatically stops that service. ClusterWare does not permanently disable that service on the master node. To do that:

```
systemctl disable nscd
```

Beofdisk does not support local disks without partition tables

Currently, **beofdisk** only supports disks that already have partition tables, even if those tables are empty. Compute nodes with preconfigured hardware RAID, where partition tables have been created on the LUNs, should be configurable. Contact Customer Service for assistance with a disk without partition tables.

Issues with bproc and the getpid() syscall

BProc interaction with *getpid()* may return incorrect processID values.

Details: The Red Hat's glibc implements the *getpid()* syscall by asking the kernel once for the current processID value, then caching that value for subsequent calls to *getpid()*. If a program calls *getpid()* before calling *bproc_rfork()* or *bproc_vrfork()*, then bproc silently changes the child's processID, but a subsequent *getpid()* continues to return the former cached processID value.

Workaround: do not call *getpid()* prior to calling *bproc_[v]rfork*.