Penguin Computing
White Paper

Critical Decisions:

A Guide to Building the Complete Artificial Intelligence Solution Without Regrets

From Penguin Computing

2018

Introduction:

Artificial intelligence (AI), and specifically machine learning (ML) and deep Learning (DL), is more than the next stage of data analytics. Combined with levels of computational power normally associated with high-performance computing (HPC), the adoption of ML/DL techniques is revolutionizing a wide range of application areas (i.e., web searches, medical diagnostics, etc.). By using these radically new approaches to solve difficult, existing problems, AI is leading to fundamental breakthroughs in real-world fields as diverse as agriculture, finance, manufacturing, and other industries.

However, organizations are facing critical challenges in constructing an AI infrastructure that can actually drive the business outcomes they seek. Partly because of the speed at which research is progressing, but also because of the general lack of experience researchers have with building systems at scale.

Many AI systems are designed without sufficient thought in key areas, including the following themes:
• Managing software ecosystem, focusing on orchestration and workload portability
• Accurately sizing AI infrastructure to optimize utilization
• Balancing overall data workflows and performance
• Deploying and scaling AI infrastructure efficiently

The leading-edge nature of the work and the complexities involved in getting systems up and running mean that AI researchers often struggle to integrate the most robust technology currently available and find administrators with the technical expertise to use these technologies to their maximum potential. But there is no turn-key AI solution that fits all situations.

This guide is meant to help organizations understand what factors to consider when building AI systems that suit their specific circumstances. The suggestions included in this document are derived from the design, implementation, and operation of multiple HPC-oriented Top 500 cluster designs as well as other, complex clusters purpose-built for specialty compute scenarios.

From a technical perspective, we will focus on tips from a HPC approach (versus a data analytics approach) and look at a bare-metal (versus a container-based) design, addressing issues related to graphics processing units (GPUs) because of their common usage in HPC-based AI systems.

Consideration 1: Build with Balance

The overarching consideration even before starting to design an AI system is that you should build the system with balance. What does that mean? Consider how your system is going to meet the "functional" needs of the technology as well as the needs of the research project.

This may sound obvious but, too often, AI systems are designed around specific aspects of how the team envisions achieving its research goals without understanding the requirements and limitations of the hardware and software that would support the research. The result is a less than optimal — even dysfunctional — system that fails to achieve the desired goals.

For example, deploying AI infrastructure efficiently requires balancing power and cooling with performance. To achieve high performance, AI systems often require hardware accelerators, such as GPUs. But, a single eight-GPU system requires 3,000 Watts of power all by itself. This can make filling a rack a challenge, since a ten kilowatt-per-rack limit would allow only three GPU servers per rack.

At the same time, you need to ensure sufficient bandwidth between storage, GPU, and networking in a single compute node. Similarly, you have to balance how the overall budget is spent to achieve research with the need to protect against power failure and other scenarios through redundancies. You may also need to build in flexibility to allow repurposing of hardware as user requirements change.

Perhaps your organization has long-term IT strategic initiatives that require certain components be included in your design. You must address these issues, but only in a way that is in accordance with performance goals and the proper functioning of your AI infrastructure.

A design partner with deep understanding of the latest technology as well as with practical experience in AI system design and management can help balance these and other concerns, ensuring that your AI system is technically sound and performs optimally while accounting for the practical issues your organization faces.

Consideration 2: Design Based on Software, Not Budgets

Artificial intelligence and deep learning projects are defined by the industry, function-specific software that is required to run the research. Each software framework has its own advantages and drawbacks, and each will have an impact on hardware requirements.

Several basic examples of hardware dependencies on software choice include:
• If you're developing or rewriting your app from scratch, would an AI oriented software framework, such as TensorFlow or PyTorch, be a better choice?
• Was your code written for CUDA? If so, you can only use NVIDIA GPU hardware for acceleration.
• Was your code written for OpenCL? OpenCL is an open-standard alternative to CUDA, so you're free to use additional accelerator providers such as Intel and AMD as well other emerging ASIC providers.
• What if your code is dependent on x86 hardware? If so, you will need to look into Intel® Xeon® Scalable or Intel Xeon Phi-based systems.

Beyond simple compatibility, however, you will also need to define ahead of time what level of performance is acceptable to reach your research goals and build the hardware (and any supporting software) specifications to support that into your design from the start.

The best way to determine if your AI partner has the expertise to understand the true infrastructure needs of your research is to find one that has not only designed AI systems for other clients but who has managed AI clusters that run TensorFlow, PyTorch, CNTK, and other common frameworks.

This means the design team has access to real life feedback on how systems perform under specific circumstances, data that a client is often too busy to provide after the system has gone online. Incorporation of that performance information means your design team will be able to troubleshoot issues that your in-house IT or scientific team may not be able to do.

However, it is important not to reject any options strictly because of cost. The question at the heart of AI system design should always be, does the system have the correct tools and computational power to accomplish what my organization is trying to achieve?

Engineers with a deep knowledge of the latest technologies can design surprisingly cost-effective yet powerful AI systems, once they understand the research goals and infrastructure requirements. With sufficient experience and expertise, they can help you make tradeoffs that will balance your budget with your research goals.

For example, choosing commodity hardware, including those based on Open Compute Project (OCP) specifications, can often help save budget in one area so that it can be allocated towards other areas, including more costly hardware required by the software needs of your research project.

This design approach certainly requires deep engineering expertise and out-of-the-box thinking as well as very strong partnerships with hardware and software vendors, but it has already been done to scale and is worth considering as a means to achieve strict performance goals that require more costly components in other areas.

Consideration 3: Include Storage As Part of the Plan from Day One

Improving algorithms is important to reaching research results. But without huge volumes of data to help build more accurate models, AI systems cannot improve enough to achieve their computing objectives. That's why inclusion of fast, optimized storage should be considered at the start of AI system design.

Deploying AI infrastructure efficiently requires understanding data center environment and capabilities. Storage for AI should be optimized for data ingest, workflow, and modeling. Some considerations come naturally. For example, including fast, solid state storage is known to be critical to keeping GPUs fully utilized.

However, other, less commonly considered factors you should think about — and adjust your storage to support — include:

• Sizing data needs, including getting a realistic assessment of how much data is necessary to process versus how much data needs to be archived

• Balancing power for efficient data center load and deploying density to match data center capabilities

• Taking into account what workloads various stakeholders need to address. Would all stakeholders need access to the production environment and only some need access to the research and development environment? What would be the percentage of data used for training versus inference?

• Understanding the workflow and data types required between projects and data scientists who will be working on your infrastructure. You might even want to think about whether it makes sense for your organization to deploy a platform or internal "data science as a service" to streamline access and infrastructure needs.

Adding storage after a compute cluster has been finalized WILL result in a redesign or a failure to achieve the research goals your organization planned to tackle with the AI system. For example, if you will be collecting data from customers, did you consider how you will anonymize that data to lessen the impact of privacy breaches? And there are literally dozens of other, non-trivial storage-related issues that need to be addressed in overall system design.

Another aspect to consider with storage is flexibility. Future needs are difficult to predict, but you can build a great deal of flexibility into your data center with software-defined storage (SDS) and policy-based provisioning and management. Software-defined storage dynamically optimizes infrastructure capabilities to application service-level requirements, providing both storage and staff efficiencies, a huge benefit when dealing with high-investment AI projects.

A storage vendor with experience in SDS can help match the most appropriate SDS file system to your research and IT needs. Taking the time to review your options can have a huge, positive impact to how the system runs once its online.

Get datacenter experts who have previously not just supported AI deployments but who have worked with the compute component designers to build a truly integrated system. This will reduce costs and increase system efficiency. It is even better if the two teams have worked together on previous projects, as it is likely they can speed the time-to-delivery and may even have developed optimizations specific to their storage and compute products.

In rare cases, frequent storage and compute collaborators will also be able to provide you with professional services (including implementation services) that are optimized for the storage and compute technologies you are using.

Consideration 4: Avoid Problems By Thinking Through Compute Issues

After you have clarified your storage needs, focus on selecting your nodes and their accelerators. As you do so, consider the more practical aspects of how the system will be used and even how it can physically be built.

This is the area where the concept of "balance" becomes both most important and most challenging. Part of the reason for this is the natural desire to focus on research goals. However, your design cannot be skewed so far towards those goals that you set yourself up for issues with your compute infrastructure later.

Avoid problems later in the process by thinking through non-research issues in advance. Some examples of these elements include:

- Your training sets or batch sizes — Batching is a way to improve utilization of GPU by making more data available for training at each iteration. However, batching can make it difficult to get a neural network to converge to a working configuration during training.
- Whether you need to pre-process the input data before sending it to GPUs for training — Uncompressing JPEG or other compressed data formats may require the power of the host CPU. However, these CPUs can become bottlenecks for GPU performance if too much work depends on the CPU.
- Training — Are you going to want or need to train across multiple chassis or can you fit training tasks into a single node or accelerator?
- What you can put into your rack — GPUs can consume a lot of power. Fewer GPUs per server could give more options for rack layout, easing power distribution and cooling issues.

An experienced AI design partner will be able to help you map out an even more comprehensive list of considerations. Addressing these issues will likely require several design iterations, but this will be time well-spent.

Consideration 5: Network Like Your Research Depends on It — Because It Does

Constructing a large data center is pointless unless the massive amounts of data required for AI projects can be accessed quickly and used appropriately. That's why a rigorous network design effort is critical to achieving research goals. In fact, a well-designed network can share responsibility for handling and accelerating data workloads.

Given you are building an AI system, your system will likely need to provide high bandwidth and low latency to each GPU. As you design your system interconnects, there are some fundamental considerations to take into account related to networking.

Start by working backwards, from the way that the system will run when it is live. For example, consider:

- What are the storage performance requirements for different data types, sizes, workflow stages, etc.?
- How should the data sets be shared for the specific software being used and the research goals?
- What sort of connectivity is required to deliver data to your processors fast enough to maximize their utilization? Why is this important? If you aren't using your computing capacity to the fullest, did you really need to spend all that money on it to begin with?
- Alternatively, if your research demands a certain level of computing power and your design does not allow you to reach it, again, what was the point of buying all that computing power?

Next, define and get organizational agreement on what performance levels are acceptable for your organization's research goals. This benchmark will give you and the team building your system a great deal of guidance as you build out your system's network infrastructure. A strong engineering team can also use this guidance to help your system be more efficient, so that you can reduce operating costs and missed deadlines resulting from slow networks.

Help them by thinking through some practical issues, such as:

- Is internode communication important to you?
- Will you have many jobs on single GPU?
- Will you need to scale to jobs that span a large number of GPUs?
- How many jobs will span groups of GPUs?

In addition, as we noted earlier, consider how open technologies can help extend the budget while still reaching performance goals. Software-defined networking based on open technologies, for example, enables greater flexibility and customizations and is ideal for high-bandwidth, dynamic applications, including those running AI research projects. Ensure that you work with a networking partner that uses rigorously tested and qualified software stacks (such as Cumulus Linux) or you may find that your flexible networking is failing to connect your data with users.

Consideration 6: Plan for Life After Go-Live

Making the jump from development to production forces a system designer to account for a whole new set of critical, but previously ignored system design constraints. Production issues, such as quality of service and request response time, are critical issues that determine the value of an application as well as the quality of a user's experience. These in turn become major drivers of issues regarding scalability and reliability.

Some elements of AI system design are architected as countermeasures against failures. However, it is equally important to plan for success and the implications that has on the infrastructure.

There are obvious physical issues, such as how much power, cooling, and floor-space will be required. Assuming all goes well, will you need to scale up or scale out? Are there elements that are necessary for expansion that need to be taken into consideration for the original design? Can performance degradation be avoided when you expand?

There are also research-related issues to consider. Before "go live," ensure that you have an optimization plan mapped out for your tools, frameworks, and applications. Develop a plan that accommodates regular and unexpected maintenance. Also, remember to create a user training plan.

Conclusion

While production AI systems have existed for several decades, the recent leveraging of vast amounts of compute power has transformed the field to the point where the field is experiencing a "Cambrian explosion" in the number of AI applications. And, this has brought with it the problem that there are few individuals with practical experience in building production AI systems, particularly at scale.

The hardest part of AI system design is knowing that, despite having an excellent in-house team, the fate of your research lies — ultimately — in the hands of strangers. Still worse, if these complex systems are designed or configured incorrectly, you could see huge performance hits.

Correcting any design mistakes can be extremely costly, both in terms of time and money. However, you can avoid these pitfalls by partnering with experts and spending the time up front to plan and develop your AI system that will be successful now and into the future.

Learn More

See related solutions at www.penguincomputing.com/ai

For pricing on your specific design needs, contact a representative by email at sales@penguincomputing.com or call 1-888-PENGUIN (736-4846).

Purchase with Financing

Finance products, services, even soft costs with Penguin Computing Capital. Choose from options such as no money down, flexible billing choices, extended repayment timelines, and a variety of end-of-term alternatives.

About Penguin Computing

Penguin Computing, Inc. is a 20-year-old, U.S.-based global provider of high-performance computing (HPC), artificial intelligence (AI), and data center solutions with more than 2,500 customers in 40 countries, across eight major vertical markets. Penguin Computing offers a comprehensive portfolio of hardware, software, and services, including solutions based on the Open Compute Project (OCP), as well as financing and top-rated customer support. Penguin Computing products include Linux-based servers, software, integrated, turn-key clusters, enterprise-grade storage, and bare metal HPC on cloud via Penguin Computing® On-Demand™ (POD).