
WEKA
insideHPC
Scaling Your GPU Deployment
Things to Know When Assessing, Piloting, and Deploying GPUs
DECEMBER 2020

Contents

Introduction

Traditionally, Graphical Processing Units (GPUs) were used in processing 3D content/data in gaming. Today, GPUs have become a key technology for applications in manufacturing, life sciences, and financial modeling. GPUs speed up simulations due to their fast parallel processing capabilities and are now being used extensively in Artificial Intelligence (AI) and Machine Learning (ML) applications. Creating an effective large-scale environment that utilizes GPUs takes planning, piloting, implementing at scale, and, finally, evaluation.

For an effective production environment using GPUs, organizations need a defined process that contains the following activities:

- Assessment
- Pilot Program
- Scaling implementation for anticipated workloads

Within the assessment and pilot activities, project leaders should expect to do significant upfront planning and then when in the pilot phase, complete multiple iterations to achieve a clear understanding of tuning of the components. Figure 1 represents the process and shows that iterating on these steps is key to a successful implementation at scale.

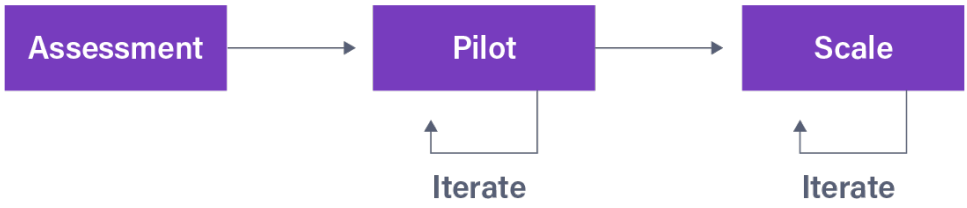

Figure 1 - Key Stages

Assessment

A key to creating a large-scale system that responds to user demands starts with understanding the challenges that a company or organization is trying to solve. Depending on the industry and algorithms used, a number of areas need to be understood when planning and deciding if a GPU-based installation is right for the workloads.

- What is the expected result from a new system design? Customers need to understand what the business needs and wants. Better decisions? More accurate voice recognition? Assistance with guiding software simulations? Before designing a working system, the goals must be understood and be attainable before starting a new project.
- What are the fundamental algorithms that will be used and the nature of the software? If the software chosen is provided by a third party, and the software has been optimized to use GPUs, then the project and implementation can proceed more quickly. However, suppose the implementation will use new algorithms that require software development. In that case, the implementation team needs a deep understanding of the software requirements to determine whether or not GPUs will shorten the application's time.
- How much data will be used? Even if using GPUs might reduce an application's running time, GPUs may not be the correct solution if the quantity of data is low. When determining the amount of data that will be used when setting up a new software system to analyze data, looking back at the amount of data collected in the past day, week, month, or year is extremely useful. Just as important is looking at what types of data are helpful for the task at hand. Just because 1 TB of data per day has been collected does not mean that a new system must analyze all of this data. Specific tools may be needed to separate the data being used compared to the data being captured. Extract, Transform, Load (ETL) functions may tag and categorize data early and reduce the working dataset. More data at this stage does not always equate to better data.
- A key to any system is understanding the type of data, where it is coming from, and what insights might be contained in the data. The metadata for any system needs to be examined to understand various properties of the data. Implementing software that can inspect the metadata is not just a one-time effort—comparing the metadata over a period of time,

whether hours or weeks, will give an insight into not only data growth but trends in data types, complexity, and relevance to the study at hand.

- Understanding the type of the data to be used is critical to developing an optimized system. For example, when analyzing images or videos, one would conclude that GPUs would be the best choice, while trying to make sense of IoT information may be better served with CPUs. A lot of this will depend on a mix of the amount of data that is anticipated to be of value and the type of data or combination of types of data in the pipeline.
- Besides the CPU and GPU choices, other infrastructure components need to be understood at the time of planning. Network bandwidth and storage types are integral parts of a smooth-running system that must be mapped out early to identify which of these pieces must be upgraded or purchased new.
- The eventual sizing of the infrastructure, CPUs, accelerators, networking, and storage capacity should support the data volumes, the SLAs' latencies, and the budget needed. A good sizing exercise starts at the result and then works backward to the required hardware and framework. All aspects should be understood, including network latencies and bandwidths and the storage system's ability to deliver data quickly to where it is needed. An example of incorrect sizing would be connecting a GPU with 12GB/s performance when inferencing but using remote/local storage that can do 2-6GB/s only. This would allow half of the GPU to be idle, reducing the value of that GPU.
- When assessing performance, evaluate pipeline bottlenecks and not just performance bottlenecks. For example, how do the GPU servers share the data? Do they copy it to local drives? This procedure takes time and is considered a pipeline bottleneck, not a strict performance bottleneck.
- When planning for performance, keep in mind that the required performance on Day 1 is not the same requirement for Day 2, Day 30, Day 100, and so on. Set several performance requirements (e.g., more GPUs or more data scientists) because the requirements at each stage will be different. Soliciting expert input helps the implementation team to understand the current situation and the future state in no uncertain terms. When planning for a new or upgraded system to analyze or learn from existing processes, getting the help of specialists who have experience in the domain field should be standard practice. An objective look at the amount of data in the system, the possible algorithms, and various options will ultimately lead to a better implementation at scale.

Pilot Program

Designing and implementing a full-scale system that uses GPUs can be complex, expensive, and prone to mistakes. A pilot program using a small number of systems with a reduced amount of data can lead to better outcomes after full workloads are implemented. Testing algorithms and accelerators based on the CPU and GPU combinations and location of the data are better understood and tuned in a pilot program rather than in a full-size system. Keep in mind that excellent performance may come out of the Pilot program, but an eye towards whether that performance will scale is key. A well-known pitfall is to design for the pilot, and then fail at scale.

- Should the pilot program be implemented in an on-premises system or on a system provided by a cloud provider? With GPUs available on many types of instances across many cloud providers, utilizing these resources for a pilot program usually makes sense. Purchasing a system that contains the necessary hardware and software for an on-site data center will be a cost-effective addition to an infrastructure. Flexible infrastructure is critical to making the cloud/on-prem choice. Algorithms and data pipeline from step-to-step may be the same and will allow you to refine how your data pipeline works, and then you can transfer the lessons learned from on-prem to cloud and back, and scale in the environment of your choice.
- A pilot program gives flexibility to experiment with CPU and GPU combinations. While assessing the ratios of CPUs to GPUs may explain how a running system will optimally deliver results to the end-user, experimentation with combinations of these components will give more confidence to the organization to know how to optimize the ratio.
- What specific GPUs and CPUs are best for the software system? There are many CPUs and GPUs available today, each with varying core counts, clock rates, instruction implementation, and I/O bandwidths. The higher-performing products come with additional costs but might not result in better performance. For example, a system that sends a lot of data to the GPUs for analysis and keeps the GPUs busy might not require the latest CPU, as CPU performance might not matter for this specific application. The pilot program can identify not only bottlenecks but also component best choices as well.
- Do you use greenfield applications (those that are entirely new to an organization) or brownfield applications (those that are part of an existing infrastructure)? Many organizations might already have applications that use GPUs but are looking

to either scale applications, improve performance, or implement new features. A pilot program would be ideal for this scenario, but the developer would need to "peel" off the code for the new feature or the area to investigate performance. Additional data might also need to be collected for the pilot program. Moving the entire dataset to a public cloud provider is not necessary, and it could be expensive. Only a small portion needs to be moved, which is the portion used to validate the new model or the software algorithms. Greenfield applications pose a different set of issues: not just the required algorithms, but from where do you get the pilot data? Do you purchase it? Make it up? Borrow it? These choices will lead to decisions in a pilot program that might have future implications. Consider them early.

- The length of time for a pilot program is also a required part of planning. Just getting a system running will not lead to conclusive results. An open-ended timeframe leads to stalling of the business outcome. However, when planning the pilot, weeks to months may be the optimal amount of time needed to understand the algorithms' bottlenecks, correct hardware size, and anticipated storage needs. As with any new technology, the possibilities are endless. After customers see the opportunities that an accelerated system shows, specifically with AI applications, they can add capabilities to the pilot so that they can gain experience with the new implementation and even discover new possibilities that had not been considered previously.
- Learning from a pilot program is an essential part of the full-scale implementation when working with a system that utilizes GPUs. In fact, expect a continuous learning process throughout the pilot because bottlenecks can occur even in a pilot environment. The GPUs might not be kept busy, or the system might not scale expected at first. Implementation teams might change their minds about which components to use, or during their discovery in the pilot phase, they may decide to re-assess what the measure of success will look like at the end of the pilot program. That is why a flexible pilot is essential for meeting long-term goals.

Scaling and Implementation

After the assessment has been performed and the pilot program has shown acceptable results, it is time to move on to the full-scale implementation phase. There are many considerations to consider in the move to a production system.

- Acquire the necessary new hardware and gather the existing and compatible equipment. If an organization already owns hardware systems close to the pilot program's systems, reuse them for a production environment. The delivery model can now include using systems, storage, and networking at a public cloud provider. If the scaling plan is in the public cloud, start automating the spinup/spindown processes for infrastructure in the cloud.
- Ensure that the network infrastructure can handle the higher workloads that a successful system will require as the data grows, the user base grows, and the applications grow.
- Plan for future growth and scale. Suppose that your successful pilot program is able to model the full-scale implementation. In that case, a serious implementation plan should include projected data workflows and infrastructure requirements that extend 5-10 years into the future. Fundamentally, the system's architecture should remain the same if you've successfully assessed and piloted in advance. Adding more resources should ensure that the project can scale transparently and proceed successfully.
- Understand and monitor where the bottlenecks are today and scope the possibility that the bottlenecks might move around as the implementation scale grows. For example, the storage system may respond quickly to deliver data in a small-scale implementation, but it might not keep up with additional scaling. Bottlenecks can affect the CPUs (not enough horsepower or clock rate), GPUs (waiting for data), storage systems (inability to deliver the data nor sufficient capacity), or networking (insufficient bandwidth for GPU-based computing).

- Perform an in-depth investigation to determine whether the end installation should be housed within a corporate data center or within a cloud provider. For various reasons, this decision affects many levels of the organization from the top down. Not only would the costs be different, but the implementation team needs a clear understanding of how and if a cloud provider can house the required hardware, storage, and networking infrastructure, and any on-premises support needs planning. While every functional level of an organization will have to decide for themselves, careful consideration of the following can help:
    - **On-Premises** – If the most recent releases of CPUs and GPUs are needed, and an organization wants to use features of these new components, even pre-release, then on-premises housing might be the correct choice. Be sure to discuss fast networking when exploring the viability of the on-prem vs. cloud options, however, as public cloud providers might not be able to supply the desired instances and the required networking concurrently. In contrast, an on-premises installation can provide this combination. Also, if the costs of the storage requirements and running the servers full-time are high, hosting on-site in a corporate data center might be the correct choice. In other words, account for both CAPEX and OPEX when considering this model. While data security might have been a reason to remain on-premises previously, this could be less of an issue moving forward for some organizations.
    - **Cloud Provider** – Many organizations do not have the in-house expertise to install and maintain high-end servers that contain acceleration technology. This expertise is paramount when working with a storage system that relies on a wide range of technologies. For such companies using a public cloud provider might be the most optimum choice. Smaller organizations might not have the ability to access the latest technology and might have to rely on public cloud providers. In most cases, using a cloud provider will increase OPEX, and CAPEX will be relatively minor.

Related Considerations—Storage

The performance of a large HPC or AI system that is based on GPUs or other accelerators usually depends on the utilization rates of the GPUs themselves. Another critical component to the efficient running of these systems is the storage choices. If CPUs or GPUs are starved for data, then expensive resources are not being used efficiently. Feeding the hardware that processes the data should be an upfront design decision, not an afterthought.

In the past, storage hardware relied on spinning disks that relied on mechanical parts to retrieve data. These hard disk drives (HDDs) have been available for decades. While the capacity has increased over time (although less than Moore's law) and is expected to continue, the latency and bandwidth to and from the HDD to the main memory has not increased as fast. Solid-state drives (SDDs) based entirely on electronics and not physically rotating components change the storage landscape quite quickly. Many organizations' storage systems have been based on various applications sending data sequentially to a disk or a set of disks. In high-performance environments writing to a single disk drive will create a significant bottleneck and slow down the entire system. For the majority of the working data set in GPU environments, having it on SSD devices will be the correct choice, with HDD based systems relegated to a datalake or archive tier.

Parallel file systems have been developed and used for quite some time in HPC environments. While a parallel file system reduces bottlenecks, traditionally these file systems have been difficult to install, requiring storage experts to install and monitor them within complex environments. Also, legacy parallel file systems could not tier the storage for new and innovative applications. Tiering refers to putting the more used data closer to the processing units and the less used data on slower, less expensive storage devices.

Different implementations may have varying ratios of CPUs to GPUs. Depending on this ratio and the workloads the requirements of the file system may vary. An implementation with just a few hundred CPU cores assigned to process older data may be able to wait for data to arrive from less performant storage devices. In contrast, other implementations that contain many thousands of GPU cores need data from higher performance devices.

Getting data to the GPUs or other accelerators directly from the storage system needs to take advantage of the latest technology. After all, the GPU controls the input and the output. Directly "talking" to the storage sub-system understandably speeds up performance, as I/O does not have to move through the main CPU. The advantage of this direct setup with a parallel file system is two-fold:

1. Keeping the GPUs busy
2. Allowing the CPUs to perform other tasks and not be slowed down with I/O traffic management

With applications that rely heavily on the GPUs, a speedy parallel file system must deliver data to the GPUs directly, often without involving the CPU. Read more about this at:
https://www.weka.io/blog/accelerated-dataops-with-weka-aiedge-to-core-to-cloud-pipelines-part-1/ and
https:/www.weka.io/blog/microsoft-performance-gpudirect/

For an understanding of why GPU utilization can suffer from "data stalling" and data copy pipeline starvation read more from Microsoft, Google, The University of Texas and more at:
A Machine Learning Data Processing Framework
Analyzing and Mitigating Data Stalls in DNN Training
Characterization and Prediction of Deep Learning Workloads in Large-Scale GPU Datacenters
Beyond the Hype: Is There a Typical AI/ML Storage Workload

910 E Hamilton Avenue, Suite 430, Campbell, CA 95008 **T:** 408.335.0085 **E:** info@weka.io www.weka.io