Hewlett Packard Enterprise

Technical white paper

Accelerate performance for production AI

Meeting storage requirements for distributed AI environments with HPE, NVIDIA, WekaIO, and Mellanox

Contents

Introduction

While artificial intelligence (AI) has been driving innovation in research and development (R&D) for many years, the technology is now ready to go into production for a number of industries. In fact, according to Gartner, "Four years ago, AI implementation was rare, only 10 percent of survey respondents reported that their enterprises had deployed AI or would do so shortly. For 2019, that number has leapt to 37 percent—a 270 percent increase in four years."[1] Entire industries are now pivoting around AI, and especially the AI techniques classified as deep learning (DL). For example:

- Enterprises in many verticals use natural language processing to enable real-time translation for customer service bots.
- Retailers use AI to personalize promotions and boost incremental sales.
- Consumer product companies use AI to drive product innovation through AI-generated permutations of potential products
- Manufacturers use AI to improve forecasting, predictive maintenance, and quality assurance, which lowers costs and increases production yields and revenues.
- Financial services organizations use AI to prevent fraud and better market their services to consumers.
- Healthcare groups use AI to scan medical images to detect disease faster and with greater accuracy than humans.
- Automobile manufacturers use AI for driver assist features, as part of the journey to autonomous vehicles.

As AI moves from R&D into production training and inferencing environments, data sets can grow to tens or hundreds of petabytes, and AI compute resources are expected to scale up and out. As AI models grow larger and more complex, AI servers will not only use more GPUs within servers, but as with High Performance Computing (HPC), distributed processing will occur with clustered scale-out server environments.

Using local storage will no longer be an option, because the amount of data will exceed local storage capabilities. It may be possible to copy and update data around a cluster, but it would be extremely difficult to manage and very complex to maintain. As with HPC, a shared file system must be used to avoid data contention; parallel file systems allow multiple servers to access the same file simultaneously, so a server doesn't have to wait for another server to release a file to access it. The DL training process involved in production AI will require a high-performance, scalable, shared storage solution to handle production-sized data sets and maximize compute resources.

AI development is complex and requires the right technology and methodology to be successful. HPE provides a single source for both. Working closely with world-class partners, HPE delivers technology along with guidance from resources such as the HPE Deep Learning Cookbook and consultative services. Our partner, NVIDIA, has led the industry with GPU advancements that allow data scientists to build increasingly complex models with training data sets in the petascale range, putting greater focus on the underlying infrastructure required to create a balanced solution. Similarly, Mellanox provides efficient, high-performance networks, connecting not only the servers within the compute cluster, but also the cluster to the storage. WekaIO Matrix provides the high-performance storage throughput for DL training and inferencing as well as the shared file system necessary for cluster computing.

This paper explores the impact of storage I/O on training and inferencing within a DL workflow. The results shared in this paper show how WekaIO allows the current popular neural network models to fully utilize GPU resources without saturating storage resources. In general, increased storage performance is required to avoid I/O bottlenecks during model validation and inferencing, which reduces overall time for model development.

This paper was created in partnership with engineers from HPE, NVIDIA, WekaIO, and Mellanox. These engineers worked together to design a DL architecture that will provide high performance for production DL workflows. The results shared in this paper demonstrate that the resulting solution—based on HPE Apollo 6500 Gen10 Systems, NVIDIA® Tesla® V100 GPUs, WekaIO Matrix flash-optimized parallel file system, and Mellanox networking—delivers a high-performance solution for production DL workloads.

Together with our partners, HPE provides the tools, hardware, support, and guidance to enable modern AI solutions, along with the confidence of understanding how to optimize solutions to provide the desired outcomes.

[1] Gartner, "Gartner Survey Shows 37 Percent of Organizations Have Implemented AI in Some Form," January 2019.

Intended audience

This paper is intended for data scientists, solution builders, and IT personnel who are interested in optimizing compute and storage performance for production AI systems with distributed training and inferencing environments.

Deep learning dataflow

An earlier joint white paper, "Accelerate time to value and AI insights," covered the complexities of a DL dataflow:

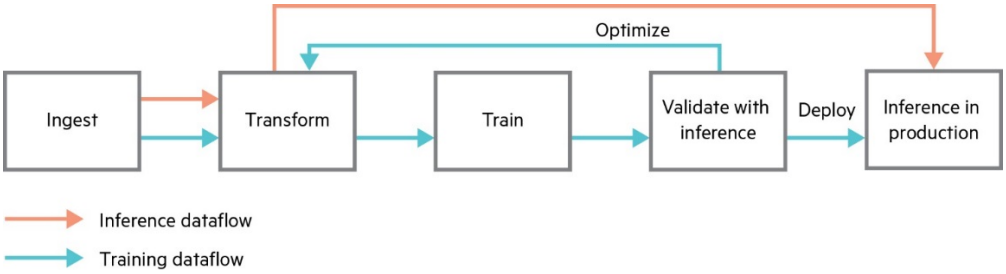


Figure 1a. Deep learning dataflow, single server perspective

Data must be ingested, cleaned, and pre-processed to be usable as part of a data set to train a DL model. And to ensure the trained model meets performance, accuracy, and quality standards, it must be validated with inferencing in a model validation stage. It may take many iterations to meet production requirements.

This dataflow model essentially provides a single server perspective on the dataflow, simplified to provide a basic understanding of the process, as shown in Figure 1b below.

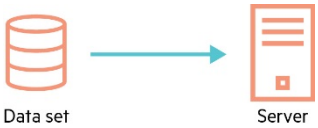


Figure 1b. Simple dataflow

With a distributed, clustered compute environment, the dataflow model is still accurate, but the actual flow of data becomes much more complex. Different servers may need to access the same data set at the same time, and the actual flow of data could be mixed between multiple servers. Our benchmark topology takes the next step, to show a single data source to four servers, as illustrated in Figure 1c.

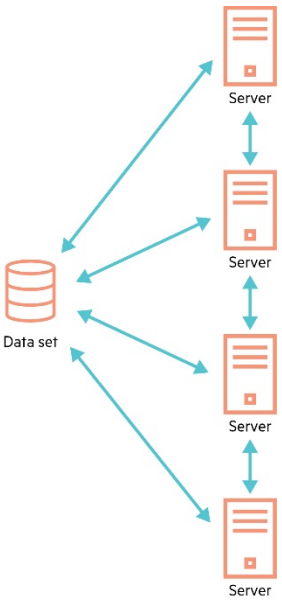


Figure 1c. Deep learning dataflow, single data set, multiple server perspective

With this benchmark, we use a parallel data approach to distribute the training for a model across the servers. Each server completes its share of the training, and the results are shared between the servers to calculate an overall update to the model.

Deep learning infrastructure

Infrastructure choices have a significant impact on the performance and scalability of a DL workflow. Model complexity, catalog data size, and input type (such as images and text) will impact key elements of a solution, including the number of GPUs, servers, network interconnects, and storage type (local disk or shared). The more complex the environment, the greater the need to balance components. HPE has integrated best-in-class components into its server infrastructure for DL, including GPUs from NVIDIA, 100 Gbps InfiniBand networking from Mellanox, and the Matrix high-performance shared file system software from WekaIO.

Compute: HPE Apollo 6500 Gen10 System

The HPE Apollo 6500 Gen10 System is an ideal DL platform that provides performance and flexibility with industry-leading GPUs, fast GPU interconnects, high-bandwidth fabric, and a configurable GPU topology to match varied workloads. The HPE Apollo 6500 System provides rock-solid reliability, availability, and serviceability (RAS) features and includes up to eight GPUs per server, next generation NVIDIA NVLink™ for fast GPU-to-GPU communication, support for Intel® Xeon® Scalable processors, and a choice of high-speed/low-latency fabric. It is also workload-enhanced using flexible configuration capabilities.

GPU: NVIDIA Tesla V100 GPU accelerator

The HPE Apollo 6500 Gen10 System supports up to eight NVIDIA Tesla V100 SXM2 32 GB GPU modules.[2] Powered by NVIDIA Volta architecture, the Tesla V100 is the world's most advanced data center GPU, designed to accelerate AI, HPC, and graphics workloads. Each Tesla V100 GPU processor offers the performance of up to 100 CPUs in a single GPU and can deliver 15.7 TFLOPS of single-precision performance and 125 TFLOPS of DL performance, for a total of one PFLOPS when fully populated with eight Tesla V100 GPUs.[3] The tested architecture leverages NVIDIA NVLink technology to provide higher bandwidth and scalability for multi-GPU configurations. A single V100 GPU supports up to six NVIDIA NVLink connections for GPU-to-GPU communication, for a total of 300 GB/sec.[4]

Networking: Mellanox 100 Gb EDR InfiniBand

When GPU workloads and data sets scale beyond a single HPE Apollo 6500 System, a high-performance network fabric is critical for maintaining high-performance, inter-node communication and enabling the external storage system to deliver full bandwidth to the GPU servers. For networking, Mellanox switches, cables, and network adapters provide industry-leading performance and flexibility for an HPE Apollo 6500 System in a DL solution. Mellanox is an industry-leading supplier of high-performance Ethernet and InfiniBand interconnects for high-performance GPU clusters used for DL workloads and for storage interconnect.

With technologies such as remote direct memory access (RDMA) and GPUDirect, Mellanox enables excellent deep learning scalability and efficiency at network speeds from 10 to 100 Gbps. The InfiniBand network provides a high-performance interconnect between multiple GPU servers as well as providing network connectivity to the shared storage solution.

Storage: WekaIO Matrix on HPE ProLiant DL360 Gen10 Servers

To maximize GPU utilization on the compute nodes, HPE partners with WekaIO for its high-performance shared storage. WekaIO Matrix includes the MatrixFS flash-optimized parallel file system, qualified on industry-leading HPE Apollo 2000 Gen10 systems and HPE ProLiant DL360 Gen10 Servers, and utilizes advanced Mellanox interconnect features.[5] Matrix transforms NVMe-based flash storage, compute nodes, and interconnect fabrics into a high-performance, scale-out parallel storage system that meets or exceeds the requirements of AI architectures. Matrix delivers single client performance sufficient to fully saturate a GPU with data, keeping DL workloads compute bound. This performance scales linearly to support large multi-node GPU clusters, even those with very high ingest rates.

The Matrix filesystem was purpose-built with distributed data and metadata support to avoid hotspots or bottlenecks encountered by traditional scale-out storage solutions, exceeding the performance capabilities of even local NVMe storage. It supports distributed data protection (MatrixDDP) for data resilience with minimal overhead and reliability that increases as the storage cluster scales.

[2] Note that 16 GB GPUs were used in the benchmarks described in this paper.
[3] NVIDIA data sheet, "NVIDIA Tesla V100 GPU Accelerator," March 2018.
[4] NVIDIA NVLink, "NVLink Fabric, A Faster, More Scalable Interconnect," December 2017.
[5] HPE architecture guide, "Architecture guide for HPE servers and WekaIO Matrix," June 2018.

Eight HPE ProLiant DL360 Servers, interconnected with Mellanox 100 Gbps EDR networking and running WekaIO Matrix, are capable of delivering 30 GB/sec for sequential 1 MB reads and over 2.5 million IOPS for small 4K random reads.[6] The infrastructure is capable of scaling to hundreds of storage nodes in a single namespace, more than enough to handle even the largest DL training data sets.

Guidance: HPE deep learning resources

HPE provides multiple resources for designing and benchmarking AI architectures:

- The HPE Deep Learning Cookbook delivers benchmarking standardization and insights from DL workloads.
- The HPE Deep Learning Benchmarking Suite is an automated benchmarking tool used to collect performance measurements on various solution configurations in a unified, consistent way.
- The HPE Deep Learning Performance Guide is a knowledgebase of benchmarking results. It enables querying and analysis of measured results as well as performance prediction based on analytical performance models. Reference solution configurations are also available for selected workloads.

Benchmark architecture

Hardware

Four HPE Apollo 6500 Gen10 Systems, each with eight NVIDIA Tesla V100 SXM2 16 GB GPUs were used as the testbed for running training and inference workloads. TFRecords of ImageNet were used for the data set hosted on the WekaIO MatrixFS cluster.

A cluster of eight HPE ProLiant DL360 Gen10 Servers was run WekaIO Matrix, containing a total of 32 NVMe SSDs, using the Matrix POSIX client. The four HPE Apollo 6500 Systems are connected to this cluster via Mellanox 100 Gbps EDR InfiniBand.

More details of the hardware under test are covered in Appendix A.

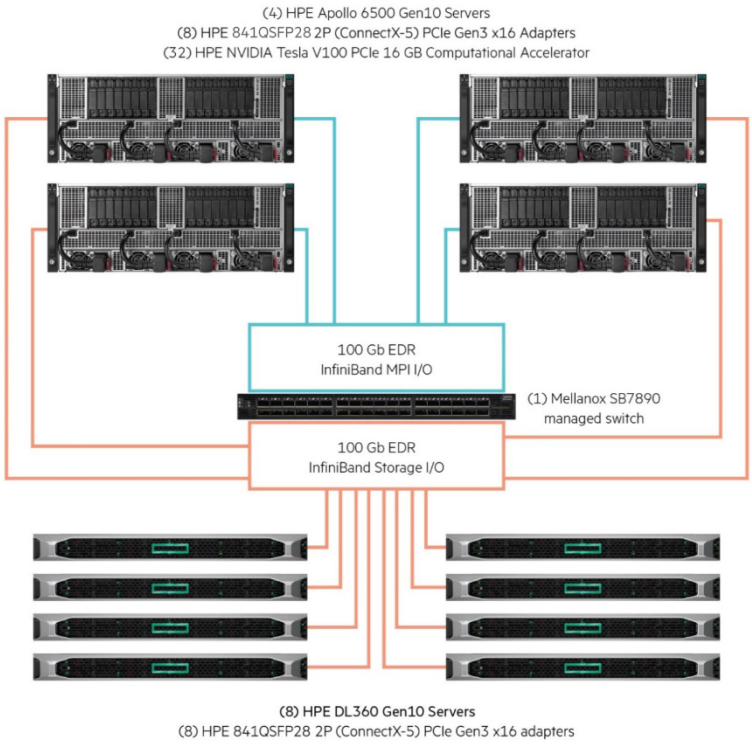


Figure 2a. Benchmark architectural diagram

[6] Testing was performed with WekaIO Matrix v3.1.8.2.

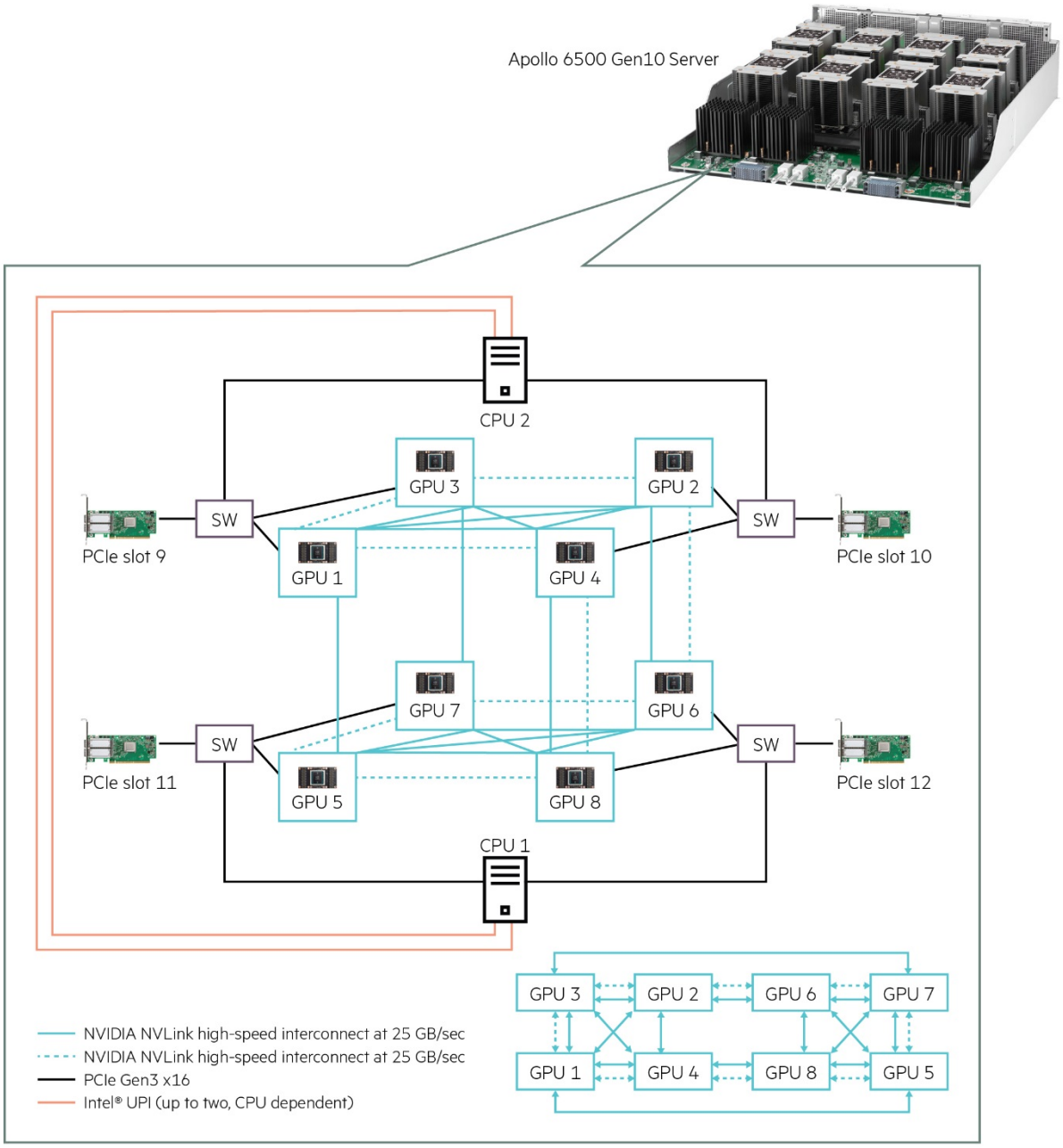


Figure 2b. NVIDIA Tesla V100 GPU topology with NVLink for each HPE Apollo 6500 Gen10 System

Software

To reproduce the training tests, a container from the NVIDIA GPU Cloud Deep Learning Platform must be built. A Singularity container was built for these tests. See the documentation provided by NVIDIA on their platform for detailed instructions on how to use this tool.

Coordination across compute servers for the training tests is accomplished with Horovod, which was created by Uber. This package uses a combination of MPI and the NVIDIA NCCL library to provide efficient collective communication functions for combining arrays that reside on the processing units, in our case GPUs.

A modified ImageNet data set was used for inference tests, where the images were decomposed to tensors of rank 3 using 24-bit RGB format, with each tensor file containing 20,000 images to maximize throughput.

TensorRT 5.0.2 was used as the inference runtime within the HPE Deep Learning Benchmarking Suite, with some custom code that enables the use of pinned memory. Pinning memory enables better performance by minimizing extra data copying to enable the data to be sent to the GPUs, and enables testing to be NUMA-aware, further optimizing performance.

To reproduce the inference tests, the HPE Deep Learning Benchmarking Suite would need to be cloned from GitHub, and both Docker and NVIDIA Docker must be installed. See the Deep Learning Benchmarking Suite GitHub page for a more detailed explanation of how to run the tool, and see Appendix B for more detailed versioning documentation on all software mentioning in this section.

Performance testing

A suite of benchmarks was completed utilizing an external storage system on four HPE Apollo 6500 Systems. The testing was designed to examine the scalability of the Matrix filesystem for various configurations of GPUs and Apollo 6500 Systems in training scenarios, as well as updating previous work to assess inference performance for individual Apollo 6500 Systems. Tests were conducted for both the training portion of the deep learning workload as well as inference validation. The benchmark tests were conducted on one, two, four, and eight NVIDIA V100 GPUs to understand how storage performance was impacted as the workload scaled.

Training

Significant work has been done at HPE to document expected training performance for common DL models, such as GoogleNet, ResNet, VGG, and Inception-v4.[7] The HPE Deep Learning Performance Guide is a central resource for HPE DL performance results. Since the models are convolutional neural networks, which are optimized for image recognition, the data set commonly tested against these models is a database of images called ImageNet.

The "Training results" and "Training analysis" sections of this paper provide a more detailed explanation of data that is already available on the HPE Deep Learning Performance Guide webpage.

Training results

To determine if storage can be a bottleneck for training four HPE Apollo 6500 Systems with NVIDIA GPUs, a Singularity container using the TensorFlow Docker image from the NVIDIA GPU Cloud (NGC) was used as the base image for benchmark training with the WekaIO Matrix shared storage solution. Refer to Appendix B for more software details.

However, synthetic benchmarks were also run to gauge the best possible performance of the system. With synthetic data being randomly generated to eliminate non-GPU bottlenecks, it is often used in the DL community as an upper bound of performance since it doesn't rely on data preprocessing or fetching. The results below show a nice linear scaling in performance as the number of GPUs are increased. Linear scaling indicates that predictable performance and maximum efficiency are gained from GPU investments.

[7] See a full list of supported models at the Deep Learning Benchmarking Suite GitHub page.

Table 1. Training benchmark batch sizes

	ResNet152	ResNet50	GoogleNet	VGG16	Inception-v4
Batch size	128	256	128	128	128

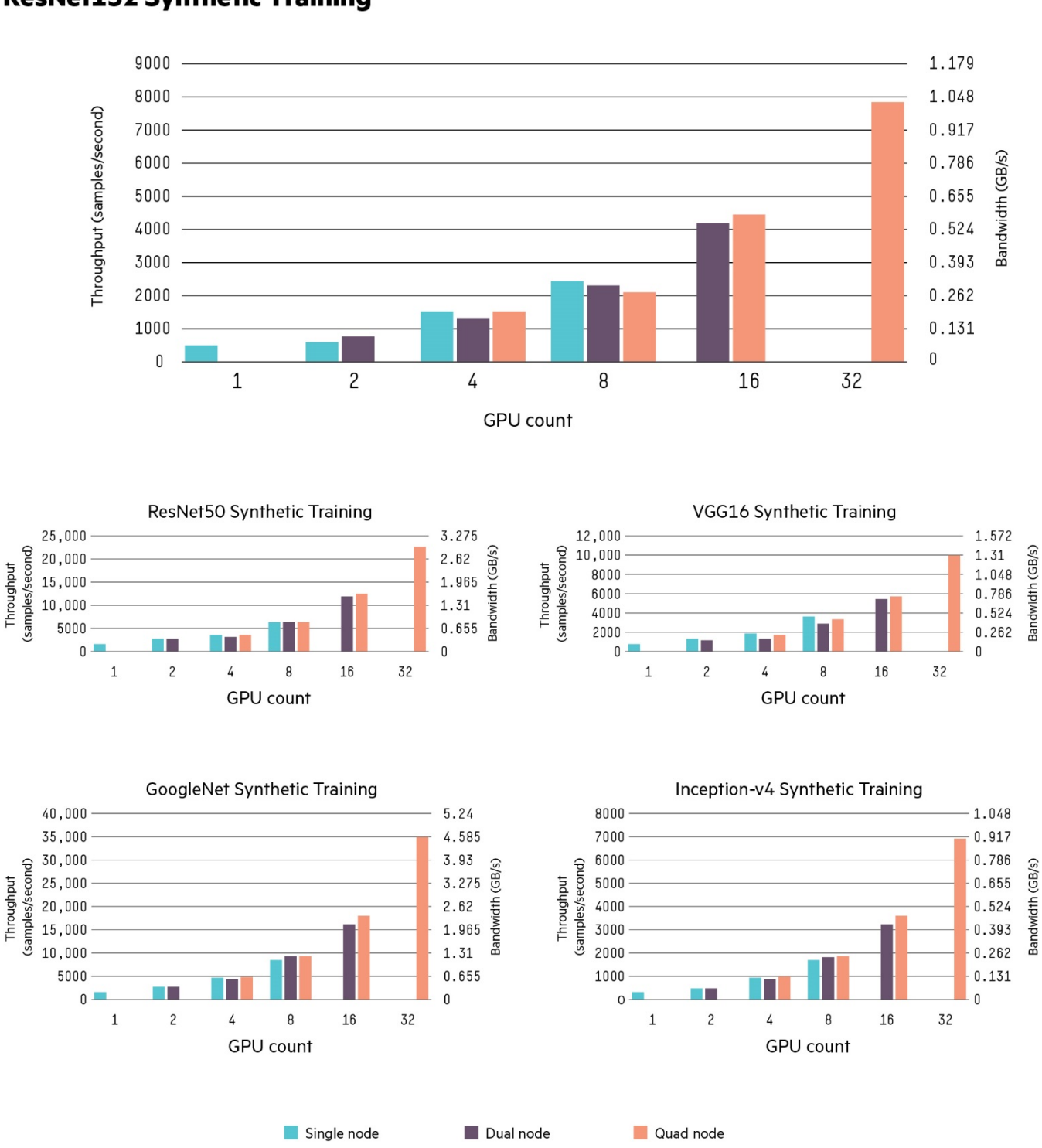


Figure 3. Synthetic training benchmark results indicate linear performance scaling

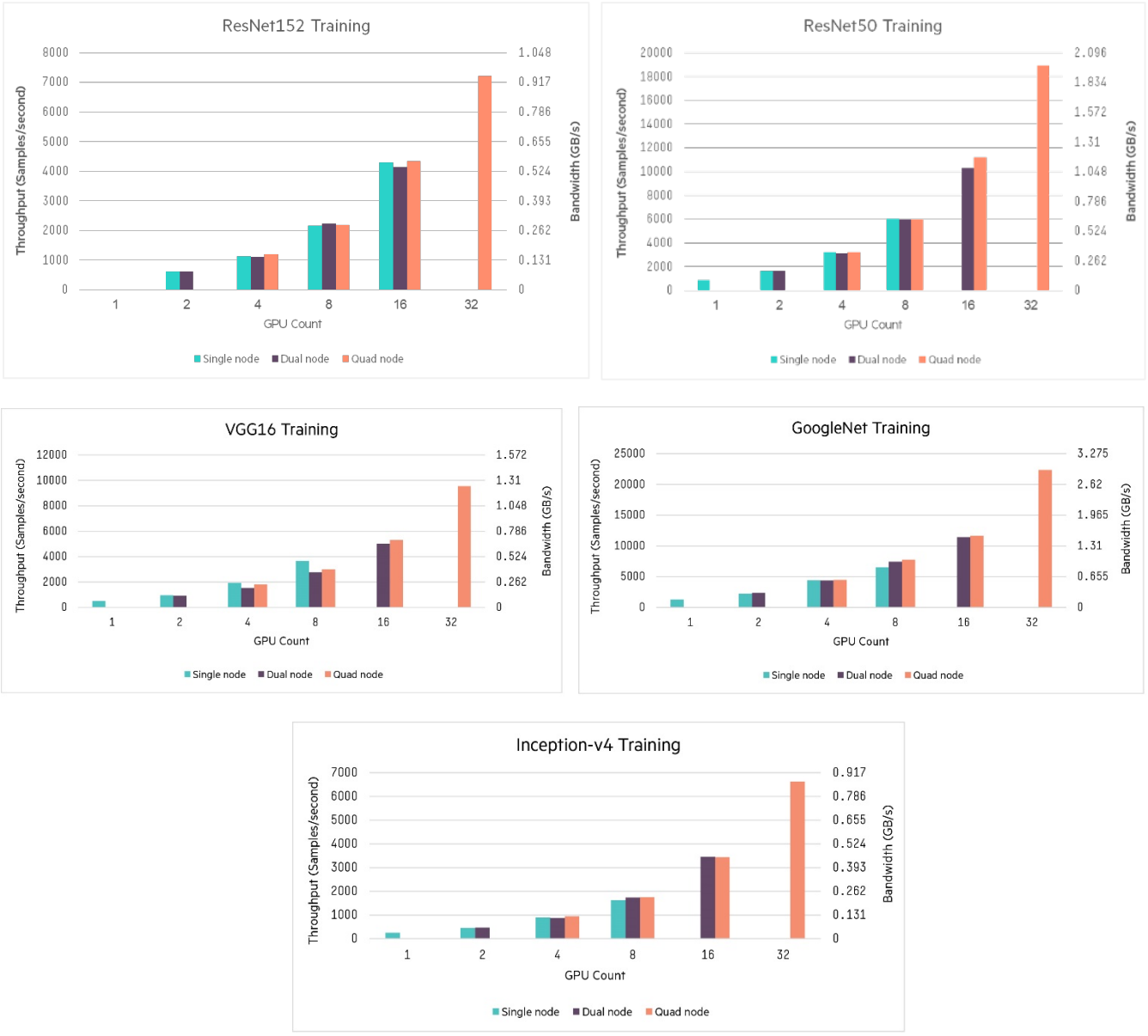


Figure 4. Real data training results demonstrate linear scaling similar to synthetic data training results

In the real world, performance generally doesn't scale linearly due to a variety of factors, including data movement between CPU and GPU, the limitations of various models and frameworks, and data preprocessing. Data preprocessing, in particular, can be very heavy, and often becomes a bottleneck for training across multiple GPU-enabled servers. WekaIO Matrix presents a shared POSIX file system to the GPU servers to reduce the performance overhead of copying data between multiple nodes, delivering the performance to utilize all GPU resources without straining I/O capacity.

HPE chose the popular ImageNet data set stored in a standard TFRecord format to enable reproducible results. Numerous tests were performed with various batch sizes, but only the batch sizes yielding the highest performance numbers for each of the five tested models are included in the results presented in Table 1. The following results were achieved using mixed precision.

The method used for scale-out testing leverages Horovod and data parallelism to distribute the training across a cluster of GPUs. Each GPU maintains a complete copy of the code and parameters for the neural network model, with data being split into partitions and parceled out to each GPU for each step. Each GPU then calculates local loss and gradient for that step. That information is exchanged between all GPUs in the cluster, generating total loss and gradient, which is then used to calculate an update to the model parameters.

Training analysis

The training data clearly show that HPE Apollo 6500 Systems with varying quantities of NVIDIA SXM2 16 GB Tesla V100 GPUs scale well both by adding additional systems and by utilizing more GPUs in a single system. The testing results show that using one server with eight GPUs delivers comparable performance to using two servers with four GPUs each, or four servers with two GPUs each.

Though storage bandwidth requirements continue to be relatively low on a per-client basis, there is evidence that, especially for less compute-intensive models, bandwidth requirements are scaling linearly. This means that even larger deployments than our four-node configuration could become performance-constrained without a high-performance file system such as WekaIO Matrix.

Worth noting is the ease of use of a configuration leveraging WekaIO Matrix. Due to the removed dependence on data locality, it is very easy to add new Apollo 6500 Systems to a test environment using this share filesystem architecture. No additional data copying is needed to prepare additional clients, which helps to create an agile and flexible architecture for such a rapidly growing use case. As compute requirements increase, WekaIO Matrix and Mellanox networking provide excellent performance and ease of scaling.

Inference

Inference is a process that typically occurs at the edge after a trained model has been deployed into production. As such, it does not require communication with the storage infrastructure used to train the neural network. However, inference is also used to validate the model during training, enabling more informed tuning of the neural network to improve performance or accuracy. In the latter use case, storage and computational devices have a bigger impact on the overall performance of a DL application. For that reason, the validation use case was tested using the HPE Deep Learning Cookbook to understand the impact of I/O on the overall model training time. A modified ImageNet data set was used for inference tests, where the images were stored in uncompressed 300x300 24-bit RGB format in files each containing 20,000 images, and large batch sizes were used to maximize throughput.

The inference tests produced here are a follow-on to tests executed previously by HPE in the interest of verifying performance from a new release of WekaIO Matrix.

Inferencing results

HPE tested five different DL models against updated Matrix software. Ten warmup batches were used, followed by 400 batches, the results of which were averaged into the data presented in Figure 5. The testing framework enables specifying the number of threads that prefetch data from storage, as well as the inference queue depth that the test framework works to maintain. These tests specify 13 prefetch threads and an inference queue depth of 32 with mixed precision.

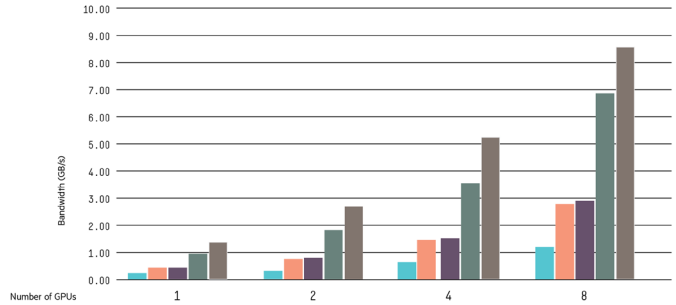
HPE is leveraging WekaIO Matrix updates that support specifying more resource usage for Matrix client software, which allows a given client to drive even more I/O performance to the Matrix filesystem. This enables even more resource utilization on the overall infrastructure within the same hardware footprint.

Since inference is driven by data processing and movement, synthetic benchmarks were not used in this round of testing, as real data benchmarks were expected to be more indicative of real world results. It is worth noting that these inference tests are throughput-oriented for large-scale model validation, and so large batch sizes were utilized to maximize throughput. With ongoing testing and tuning, the benchmarks shown here should continue to show improvements in the future.

Table 2. Inference benchmark batch sizes

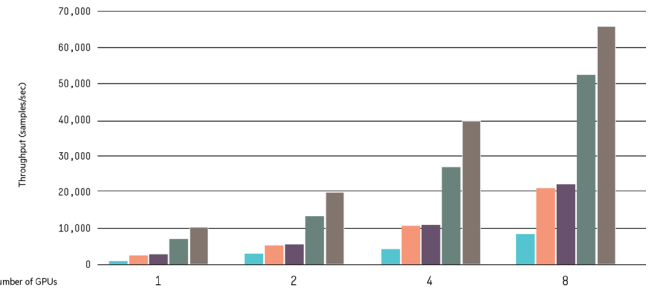
	ResNet152	ResNet50	GoogleNet	VGG16	Inception-v4
Batch size	128	256	128	128	128

WekaIO Inference Bandwidth by GPU



Number of GPUs	1	2	4	8
Inceptionv4	0.14	0.27	0.55	1.10
ResNet152	0.34	0.68	1.37	2.73
VGG16	0.36	0.72	1.43	2.87
ResNet50	0.88	1.74	3.49	6.86
GoogleNet	1.32	2.63	5.23	8.58

WekaIO Inference Throughput by GPU



Number of GPUs	1	2	4	8
Inceptionv4	1048	2094	4185	8375
ResNet152	2605	5210	10,421	20,846
VGG16	2738	5486	10,933	21,900
ResNet50	6682	13,312	26,675	52,368
GoogleNet	10,063	20,101	39,919	65,534

Figure 5. Single node inference throughput and bandwidth results by number of GPUs

Inference analysis

The inference benchmarks show that WekaIO Matrix continues to improve its ability to maximize the performance available in the underlying infrastructure. This directly translates to a user's ability to gain insights from their data, by accelerating model validation, allowing more data to be validated in the same time period as previously used, or improving time-to-response when using an Apollo 6500 as the deployment target for a production model.

Our WekaIO Matrix with HPE ProLiant DL360 Gen10 Servers configuration can now approach maximizing a single InfiniBand network connection by exceeding 8 GB/s of bandwidth from a single client. For reference, an HPE 100 Gb EDR InfiniBand network adapter can deliver over 12 GB/s of bandwidth.

Conclusion: real-world implications for production AI

Our benchmarks show close to linear performance scaling of training and inference in both scale-up scenarios—adding more GPUs per server—and scale-out scenarios—adding servers to provide additional GPUs. The ability to scale in either direction provides flexibility in choosing how to allocate servers to address workloads. Servers with fewer GPUs can be clustered as needed to tackle a larger job in a desired amount of time or for a desired performance level. Server capacity can be tailored for anticipated workloads, which helps increase ROI, optimize resource utilization, and reduce time to insight.

The ability of this test architecture to scale out demonstrates this requirement of a parallel file system. A parallel file system allows data to be shared by multiple servers at the same time, so that one server never has to wait for another server to release data, thus eliminating any contention for data between servers. In addition, WekaIO Matrix is also optimized for flash and the mixed file types and sizes in AI to provide the throughput performance for GPU clusters.

Considerations

The distributed training results shared in this paper show a variety of configurations where a cluster of HPE Apollo 6500 Gen10 Systems can achieve expected performance scaling across various GPU quantities and distributions within the cluster. In many cases, for a given GPU quantity (the difference between GPU distributions within a cluster), the performance was fairly consistent. This demonstrates that scaling is very consistent for GPU quantities regardless of the overall method of distribution.

For inference, this paper shows further enhancement from the WekaIO Matrix filesystem to accelerate workloads and improve time to insight. The combination of WekaIO Matrix, Mellanox 100 Gbps InfiniBand interconnect, and NVIDIA GPUs within the HPE Apollo 6500 Gen10 System provides a strong platform for production AI workloads.

WekaIO Matrix offers a level of performance that meets or exceeds a local file system as the number of GPUs scale. This provides an advantage when training data sets increase in capacity and clusters scale out beyond a single GPU client. Given that every training cycle needs to be followed by a validation cycle, the advantages of WekaIO are multiplied many times over, and can significantly reduce the overall time to create a production-ready DL model.

An aspect that shouldn't be overlooked is the requirement for a high-speed network fabric. Our results show that a 10 Gb Ethernet network would likely be saturated by 16 GPUs. A low-speed network would not allow efficient usage of a large cluster of servers. In light of our results, a high-speed network is an investment towards the future, when there could be hundreds of GPU servers.

Industry use cases

It's useful to consider how the benchmarking results discussed in this paper translate to possible production use cases. All models used in testing are convolutional neural networks (CNNs), which are primarily used for feature detection within images. Scalable storage from WekaIO can handle petabyte-scale storage quantities while maintaining high-performance characteristics as presented in this paper, without replicating data locally on any client servers. The HPE Apollo 6500 Gen10 Systems enabled scaling both when additional clients were added for use in model training and when GPUs were added to existing clients.

Autonomous vehicles

CNNs such as ResNet50 can be used in the automotive industry to perform semantic segmentation or object detection to enable self-driving vehicles to safely route themselves. During the CNN training phase, large quantities of images may be used to fine-tune an overall semantic segmentation or object detection model, which may be composed of multiple stages of neural networks. To improve the training of these neural networks, larger volumes of data—either from higher resolutions or higher quantities—and more permutations on base models will be necessary. Our performance results show that HPE Apollo 6500 Gen10 Systems can be scaled linearly to perform more parallel runs of different neural network models, or run the same quantity of models faster, reducing time to insight.

Medical imaging

Another workload that would benefit from the ability to flexibly cluster servers is medical imaging workflows. Typically, smaller data sets are used to achieve predefined metrics and then larger data sets are used for production-ready model training. Given the results from this paper, with small data sets, a cluster can be configured to quickly test many different models during the development phase, and then configured to aggregate compute resources to minimize throughput time for larger production data sets.

An AI partner you can count on

HPE provides the tools and expertise to guide the creation of DL solutions, including GPU-enabled servers, shared storage and networking, and DL application expertise. The HPE Deep Learning Cookbook enables clear, reproducible benchmarking for the AI solution space, and guidance with neural network models, data format, and solution architectures to quickly create effective DL applications.

Appendix A: Hardware configuration

This section contains a detailed description of the hardware components, SKUs, and quantities used for the benchmarks. It does not cover all components in a full solution order—such as service and support, or factory configuration options—and is intended only as an accurate representation of the testbed hardware.

Figure 6. HPE Apollo 6500 Gen10 System

Table 3. HPE Apollo 6500 Gen10 System configuration

Component name	Quantity	SKU
HPE XL270d Gen10 Node CTO Server	1	P00392-B21
HPE XL270d Gen10 Xeon-Gold 6150 FIO Processor Kit	1	P01278-L21
HPE XL270d Gen10 Xeon-Gold 6150 Processor Kit	1	P01278-B21
HPE 16 GB 2Rx8 PC4-2666V-R Smart Memory Kit	12	835955-B21
HPE DL38X Gen10 Premium 8 SFF/SATA Bay Kit	1	826690-B21
HPE XL270d Gen10 NVMe FIO Enable Kit	1	P01056-B22
HPE 6+2 NVMe Instr Spec FIO	1	878192-B21
HPE Apollo PCIe/SATA M.2 FIO Riser Kit	1	863661-B21
HPE InfiniBand EDR/Ethernet 100 Gbps 2-port 841QSFP28 Adapter	2	872726-B21
HPE 2200 W Platinum Hot Plug Power Supply Kit	4	P01062-B21
HPE 2.0 m 250 V 16 A C19-C20 WW Single IPD Enabled Jumper Cord	4	TK738A
HPE XL270d Gen10 8 SXM2 GPU FIO Module	1	P01786-B22
HPE XL270d Gen10 SXM2 Heat Sink FIO Kit	2	P02939-B22
HPE NVIDIA Tesla V100 SXM2 16 GB Computational Accelerator	8	Q2N66A

Table 4. HPE EDR InfiniBand fabric configuration

Component name	Quantity	SKU
HPE Mellanox InfiniBand EDR 100 Gb/sec v2 36-port Power-side-inlet Airflow Unmanaged Switch (SB7890)	1	834976-B22
HPE 3m InfiniBand EDR QSFP Copper Cable	16	834973-B25

Table 5. HPE ProLiant DL360 Gen10 Server configuration

Component name	Quantity	SKU
HPE DL360 Gen10 Premium 10 NVMe CTO Server	8	867960-B21
HPE DL360 Gen10 Intel Xeon-Gold 6134 (3.2 GHz/8-core/130 W) FIO Processor Kit	8	860683-L21
HPE DL360 Gen10 Intel Xeon-Gold 6134 (3.2 GHz/8-core/130 W) Processor Kit	8	860683-B21
HPE 8 GB (1 x 8 GB) Single Rank x8 DDR4-2666 CAS-19-19-19 Registered Smart Memory Kit	96	815097-B21
HPE 800 W Flex Slot Titanium Hot Plug Low Halogen Power Supply Kit	16	865438-B21
HPE InfiniBand EDR 100 Gbps 2-port 841QSFP28 Adapter	8	872726-B21
HPE DL360 Gen10 SATA M.2 2280 Riser Kit	8	867978-B21
HPE 240 GB SATA 6G Mixed Use M.2 2280, 3-year warranty, digitally signed firmware SSD	16	875488-B21
HPE 1.6 TB NVMe x4 Lanes Mixed Use SFF (2.5") SCN, 3-year warranty, digitally signed firmware SSD	32	877994-B21

Appendix B: Benchmarking software documentation

Table 6. Software documentation for training tests. See ngc.nvidia.com for access to this and later containers.

Operating system	Ubuntu 16.04.3 LTS
WekaIO MatrixFS version	3.1.8.2
Framework	TensorFlow 1.12.0
Distributed training framework	Horovod 0.15.2
Container	nvcr.io/nvidia/tensorflow:19.01-py3

Table 7. Software documentation for inference tests

Operating system	Ubuntu 16.04.3 LTS
WekaIO MatrixFS version	3.1.8.2
Framework	TensorRT 5.0.2 GA
Container	dlbs/tensorrt:18.10

Resources

HPE White Paper: Accelerate time to value and AI insights

HPE WekaIO Matrix product page

HPE DL solutions

HPE Deep Learning Cookbook

WekaIO Matrix product page

HPE/NVIDIA Alliance page

NVIDIA Volta architecture

NVIDIA Tesla

NVIDIA GPU Cloud

Mellanox Technologies

Learn more at

hpe.com/storage/wekaio

Our solution partners

a00059772ENW, May 2019, Rev. 2