ARCHITECTURE GUIDE

Hitachi Solution for Analytics Infrastructure using Hitachi Unified Compute Platform RS V225G for Deep Learning

Reference Architecture Guide

By David Pascuzzi

August 2019

Feedback

Hitachi Vantara welcomes your feedback. Please share your thoughts by sending an email message to SolutionLab@HitachiVantara.com. To assist the routing of this message, use the paper number in the subject and the title of this white paper in the text.

Revision History

Revision	Changes	Date
MK-SL-140-00	Initial release	February 22, 2019
MK-SL-140-01	Add clustered options, including storage and 100 Gb network to solution. Updated the paper's title.	August 13, 2019

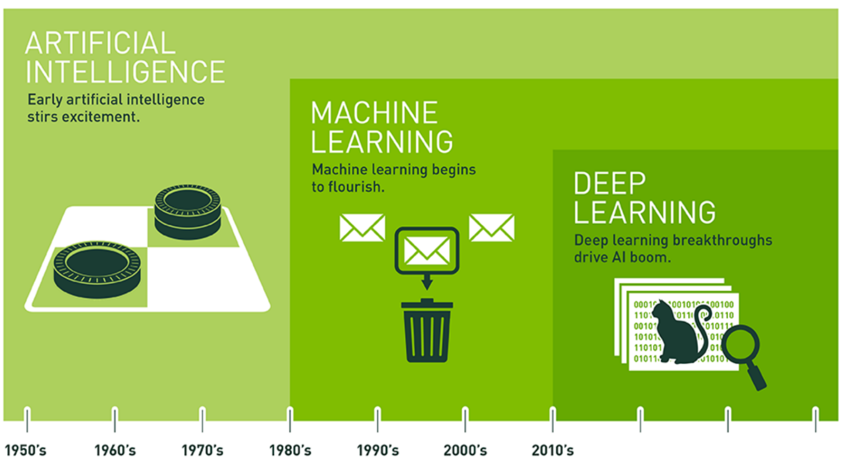
Table of Contents

Hitachi Solution for Analytics Infrastructure using Hitachi Unified Compute Platform RS V225G for Deep Learning

Reference Architecture Guide

Machine learning (ML) and Deep Learning (DL) are rapidly growing offerings relating to Artificial Intelligence (AI). They allow for processing of large sets of data using large amounts of processing power via building and running of models. DL will only be the referenced offering going forward in this document, due to its advantageous and most widely use of GPUs. While this reference architecture focuses on a standalone node for a data scientist to perform deep learning, the solution can also be deployed in a cluster.

Figure 1



Some common deep learning applications are:

- **Natural language processing**. This can allow you to create system to not only understand words, but also understand phrases. With deep learning the translation can be done in real time. This allows improved communications with co-workers and customers. The same sort of processing can be used to understand the sentiment of a customer.
- **Computer Visioning** – This allows a computer to see and process visual items similar to the way our eyes and brains process what we see.
- **Improve Market Segmentation**: Companies can use deep learning techniques to study the purchasing behaviors and consumer information, to identify similarities and create specific market segments of the consumers.
- **Personalize Service and Enhance Customer Experience**: By learning from large datasets of customer information, deep learning enables businesses to understand the needs of customers and create engaging content that addresses the demand of different market segments.
- **Risk Management**: Deep learning trains the system to learn rules of transaction from large datasets of historical transactions, user activities and network information. After the system learns the pattern of transactions, it is then able to detect fraud by recognizing abnormal patterns of transactions.

- **Predictive Analysis**: Supervised deep learning enables the system to form a model that predicts possible outcomes. It is often used to make business predictions:
  - In marketing: identify the types of products that are most likely to be purchased by a given type of person.
  - In the airline industry: predict the number of passengers for a future flight.
  - In the finance industry: predict the future fluctuations in stock prices, as well as fraud detection and prevention.
  - In retail: predict consumer preferences and align inventory with consumer demand.
  - In manufacturing: predict when a machine will break down and replace it ahead of time, reducing unplanned downtime.
- **Security**: Cyber-attack prediction and prevention, threat management, anomaly detection.
- **Healthcare**: Increase the speed of detection and diagnosis of medical conditions. Faster detection allows treatment to be provided earlier, reduce impacts of medical conditions, and provide savings to all parties.

Hitachi's solution is based upon the following key components:

- **Hitachi Advanced Server DS225** – A compact and optimized 2U accelerator server, which delivers the compute, memory and storage needed for advanced analytics, artificial intelligence, and deep learning applications.
- **Nvidia Tesla V100** - the most advanced data center GPU ever built to accelerate AI, High Performance Computing (HPC), and graphics.
- **Cisco Switches** – Industry leading switches.
- **Deep Learning Frameworks** – A library, tool, or environment that simplifies the task of developing deep learning related applications
- **Pentaho** - Analytics platform that offers both data integration and visualizing in a scalable offering via the use of GUI based tools.

Advantages of Hitachi's solution:

- Provides a solution that can be run in a datacenter. Unlike solutions running on desktop GPUs. the DS225 with Nvidia Tesla V100 GPUs can run in a data center
- Resource sharing – with a data center solution, the server(s) can be shared between multiple data scientists
- Clustering of Nodes – allows you to apply more nodes and GPUs to work on your deep learning tasks
- Performance – GPU based systems can reduce training times from weeks to hours
- Multi-Tasking – when training you can assign a subset of GPUs to a process, allowing other training tasks to use the other GPUs
- Availability – redundant power supplies, RAID controllers, and other components help to ensure the availability of the solution
- Integration with Pentaho
- Backed by Hitachi Vantara

Solution Overview

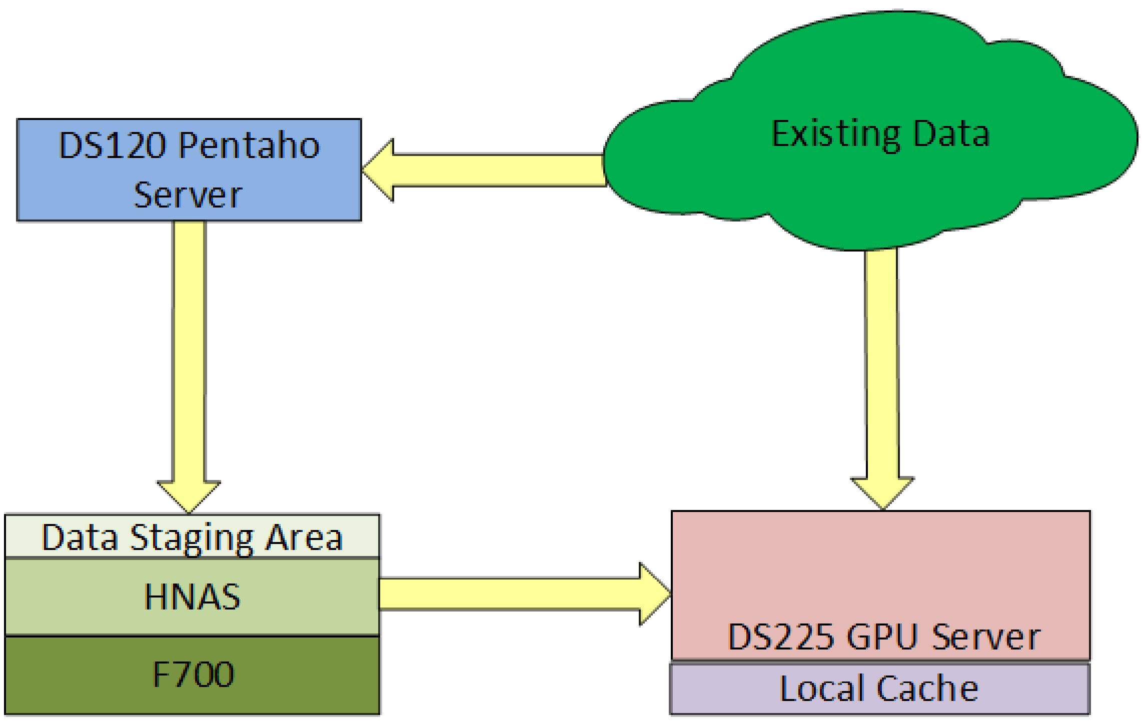
There are many different designs for a deep learning cluster. Our solution provides the basic capabilities without over-complicating the environment. Figure 2 shows an overview of our solution.

Your deep learning starts with your existing data. This data can be on your data lake or anywhere on your network in any format. This data may be able to be pulled directly into your DL environment, but it may need to be massaged. Pentaho Data Integrate can be used by the data engineer to pull data together, scrub it, help with the tagging, and perform preprocessing of the data.

Data can then be stored in the data staging area. This provides centralized storage on an NFS server. Multiple DS225 servers may need to access the same data. Learning algorithms run against the data then the algorithms can be adjusted and rerun. This staging area helps improve performance and consistency of the data.

The DS225 is the core of our deep learning solution. It can be used as a single node or as part of a cluster of nodes. In deep learning a local cache can be used to increase your performance. When a local cache is used, the network speed to access the data is irrelevant. After the first read of the data, the next hundred or more reads can access the local cache. The local cache can be implemented in multiple ways: standard OS file caching, NFS caching, or DL framework caching.

Figure 2



Note — Testing of this configuration was in a lab environment. Many things affect performance beyond prediction or duplication in a lab environment. These results were for the TensorFlow version of the ResNet-50 benchmark. This benchmark was designed by a team of data scientists and programmers to take full advantage of multiple GPUs, so it is possible that your machine learning code will not see the same level of performance improvements.

Key Solution Elements

Hitachi Advanced Server DS225

Hitachi Advanced Server DS225, shown in Figure 3, delivers unparalleled compute density and efficiency to meet the needs of your most demanding high-performance applications in the data center. DS225 takes full advantage of the ground-breaking Intel Xeon Scalable Processor family. By combining the Intel processors with up to four dual-width 300W graphic accelerator cards and up to 3 TB memory capacity in a 2U rack space package, this server stands ready to address the most challenging demands on today's IT infrastructure.

The DS225 server provides the reliability, availability and serviceability features demanded by your business-critical enterprise applications. The server's modular design simplifies cable routing and reduces service time. Redundant hot-swap drives and power supplies provide a resilient architecture for important applications.

Figure 3

The highly-scalable memory supports up to 3 TB RAM using 24 slots of 2666 MHz DDR4 RDIMM. DS225 is powered by the Intel Xeon scalable processor family for complex and demanding workloads. There are flexible OCP and PCIe I/O expansion card options available. This server supports up to 12 small form factor storage devices with up to 4 NVMe.

Hitachi Advanced Server DS120

Optimized for performance, high density, and power efficiency in a dual-processor server, Hitachi Advanced Server DS120 delivers a balance of compute and storage capacity. This rack mounted server has the flexibility to power a wide range of solutions and applications.

The highly-scalable memory supports up to 3 TB RAM using 24 slots of 2666 MHz DDR4 RDIMM. DS120 is powered by the Intel Xeon scalable processor family for complex and demanding workloads. There are flexible OCP and PCIe I/O expansion card options available. This server supports up to 12 small form factor storage devices with up to 4 NVMe.

In this solution the DS120 is used as a server for Pentaho Data Integrator and is also used as a management server.

Hitachi Virtual Storage Platform F Series Family

Use Hitachi Virtual Storage Platform F series family storage for a flash-powered cloud platform for your mission critical applications. This storage meets demanding performance and uptime business needs. Extremely scalable, its 4.8 million random read IOPS allows you to consolidate more applications for more cost savings.

Hitachi Storage Virtualization Operating System RF is at the heart of the Virtual Storage Platform F series family. It provides storage virtualization, high availability, flash optimized performance, quality of service controls, and advanced data protection. This proven, mature software provides common features, management, and interoperability across the Hitachi portfolio. This means you can reduce migration efforts, consolidate assets, reclaim space, and extend life.

Reduce risks and solve problems faster. Integrated power analytics and automation features bring artificial intelligence to your data center. Cloud-assessable monitoring tools give your product support experts access wherever they have an Internet connection for fast troubleshooting and remediation.

Figure 4

Hitachi NAS Platform

Hitachi NAS Platform is an advanced and integrated network attached storage (NAS) solution. It provides a powerful tool for file sharing, file server consolidation, data protection, and business-critical NAS workloads.

- Powerful hardware-accelerated file system with multi-protocol file services, dynamic provisioning, intelligent tiering, virtualization, and cloud infrastructure
- Seamless integration with Hitachi SAN storage, Hitachi Command Suite, and Hitachi Data Discovery Suite for advanced search and index
- Integration with Hitachi Content Platform for active archiving, regulatory compliance, and large object storage for cloud infrastructure

Figure 5

NVIDIA Tesla V100 Tensor Core Graphical Processing Unit

NVIDIA Tesla V100 Tensor Core, shown in Figure 6, is the most advanced data center GPU ever built to accelerate AI, High Performance Computing (HPC), and graphics. It is powered by NVIDIA Volta architecture, comes in 16 and 32 GB configurations, and offers the performance of up to 100 CPUs in a single GPU. Data scientists, researchers, and engineers can now spend less time optimizing memory usage and more time designing the next AI breakthrough. Nvidia Tesla GPUs are a key component in the next generation of computing, including:

- Deep learning training
- Deep learning inference
- Virtual desktop infrastructure
- High-performance computing

Figure 6

Network Switches

These solutions reduce complexity and cost, as well as enable virtualization and cloud computing to increase business agility.

Our solution includes the following Cisco switches to provide Ethernet connectivity:

- Cisco Nexus 3048 – This 48-port 1 GbE switch provides a management network
- Cisco Nexus 9336C - The Cisco Nexus 9336C-FX2 switch is a 1RU switch that supports 7.2 Tbps of bandwidth and over 2.8 billion packets per second (Bpps). The switch can be configured to work as 1/10/25/40/100-Gbps offering flexible options in a compact form factor. This switch is used to provide a 100 GbE network for communications between DS225 nodes and a 10/25 GbE network for accessing the OS.
- Cisco Nexus 93180YC-E/FX – When a 100 Gb data network isn't required, this 48-port switch provides 10/25 GbE connectivity for intra-rack networks.

Pentaho

Pentaho Data Integration (PDI) allows you to ingest, blend, cleanse, and prepare diverse data from any source. With visual tools to eliminate coding and complexity, Pentaho puts all data sources and the best quality data at the fingertips of businesses and IT users.

Using intuitive drag-and-drop data integration coupled with data agnostic connectivity, your use of Pentaho Data Integration can span from flat files and RDBMS to Hadoop to Spark and beyond. Go beyond a standard extract-transform-load (ETL) designer to scalable and flexible management for end-to-end data flows. PDI can be run on-premise and in the cloud, offers a native execution engine and the ability to run PDI out in a Spark cluster via AEL-Spark engine. Some ways PDI can help you in a ML/DL environment are:

- Locate your data — Data Engineers will assist in getting the data sources identified to meet the requirements from the Data Science team. PDI connectors can be implemented for on-premise, hybrid cloud and multi-cloud data source retrieval across many different data sources: Hadoop, RDBMS, files, ERP, DW, and more. Here is a link to what Pentaho is compatible with in regard to data sources, Big Data sources and SQL Dialect data sources: https://help.pentaho.com/Documentation/8.2/Setup/Components_Reference#Data_Sources
- Evaluate the quality of your data — Before proceeding too far, it's important to determine how much of your data is usable. Data sources are often incomplete and/or are duplicative with regard to content. PDI can assist in exploring data using a range of multiple visualizations with the Spoon client tool to review data distribution, extent of duplicates, nulls, out of range/outliers, and more. PDI will quickly and productively identify the fixes needed now, thus will save considerable time and expense down the road: https://help.pentaho.com/Documentation/8.2/Products/Data_Integration/Data_Integration_Perspective/Inspect_Your_Data
- Determine the data to use — Perform a feature review on the data to determine what will be used for the ML and DL models. PDI can be collaboratively used between a data engineer and a domain expert to determine what features would be of the best value to the ML and DL models upstream.
- Clean your data — Scrub data to implement fixes to get your data in final shape to be of value for downstream ML and DL processing. PDI can fill in duplicates, perform fuzzy lookups and searching, filter out invaluable data, implement custom Java and Python ETL code to complement the GUI development environment of PDI Spoon, and do much more. Perform re-inspections of your data as mentioned previously to have the reiterative process of inspecting and scrubbing until it the data is ready to be processed.
- Consolidate data location — Centralize data to maintain and protect your new, high-quality data by storing and managing it centrally, for example, in DS225 cluster storage. PDI can get the final data set(s) to this central location and then be picked up and used by data scientists as training, test and evaluation data sets. PDI will get the data onto the DS225 to have close data locality between the data and the high-end processing made available by the GPUs within the DS255 cluster.
- Provide access to the data — Implement data connections for production inference usage with ML/DL models. PDI can be used in production to bring the data required for production model usage that will perform inference processing on the data on the DS225 production cluster. This data can be data at rest or data in motion, as PDI has batch and streaming capabilities.

Deep Learning Components

The following are the main components of DL data set and model management processing:

- **Data analyses and cleaning** – Data is cleaned and analyzed and enhanced to identify the results before it is used in machine learning. Often you will have three sets of data: training data, validation data, and testing data.
- **Model Management**
  - **Model development** – The mathematical model is developed to approximate the data.
  - **Model training** – During model training, the training and validation data is processed against the model. This teaches the model how to behave with real data. This is an iterative process and the model may require modification via adjusting of hyper parameters, adding or removing layers, or changing the algorithm used. It will then be retrained, and this process repeated until it is accurate enough for your purposes.
  - **Inference** – Inference is the task of getting an answer with live data from the deep learning model.

A deep learning framework is used to simplify this process. With a framework, you no longer need an advanced degree in mathematics and years of programing experience to develop a DL application.

This solution is focused on TensorFlow but can be deployed with other deep learning frameworks. Some of the popular DL frameworks that are GPU accelerated are:

- **TensorFlow** is an open source software library for high performance numerical computation. Its flexible architecture allows easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices. Originally developed by researchers and engineers from the Google Brain team within Google's AI organization, it comes with strong support for machine learning, deep learning, and the flexible numerical computation core is used across many other scientific domains.
- **PyTorch** an open source machine learning library for Python, based on Torch, used for applications such as natural language processing. It is primarily developed by Facebook's artificial-intelligence research group.
- **Deeplearning4j** is the first commercial-grade, open-source, distributed deep-learning library written for Java and Scala. Integrated with Hadoop and Apache Spark, DL4J brings AI to business environments for use on distributed GPUs and CPUs.

There are different ways to configure your environment for deep learning. We support the following:

- **Bare metal**

  This is the simplest way of configuring an environment. The deep learning software runs directly on the Operating System. The disadvantage of this setup is you are more likely going to have to completely reinstall the machine

- **Python Virtual Environment**

  Virtualenv is a tool that creates an isolated environment separate from other projects. This allows you to install many different software packages without impacting other environments.

- **Anaconda**

  The Conda package manager helps you find and install packages. If you need a package that requires a different version of Python, you do not need to switch to a different environment manager, because Conda is also an environment manager. With just a few commands, you can set up a totally separate environment to run a different version of Python, while continuing to run your usual version of Python in your normal environment. Conda Environments is the most widely used environment for data scientists.

- **Docker Containers**

  A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.

- **Nvidia GPU Cloud**

  NVIDIA offers GPU-accelerated deep learning and HPC containers from Nvidia GPU Cloud (NGC) that are optimized to deliver maximum performance on NVIDIA GPUs. The NGC container registry includes NVIDIA tuned, tested, certified, and maintained containers for the top deep learning software like TensorFlow, PyTorch, MXNet, TensorRT, and more. NGC also has third-party managed HPC application containers, and NVIDIA HPC visualization containers. This eliminates the need for developers, data scientists and researchers to manage packages and dependencies or build deep learning frameworks from source.

- **Singularity**

  Singularity is a free, cross-platform and open-source computer program that performs operating-system-level virtualization also known as containerization. One of the main uses of Singularity is to bring containers and reproducibility to scientific computing and the high-performance computing world. For HPC and larger clusters it is considered to be superior to Docker.

  In most cases the environment will not be hundreds of servers and data scientists. For larger and complex environments, you will want tools to help with management and monitoring.

- **Bright Computing**

  Bright Cluster Manger for data scientists provides an environment to deploy and monitor your deep learning cluster. Its graphical interface allows you to easily get different environments up and running. It also provides the means to monitor them.

- **Mesosphere DC/OS**

  Mesosphere simplifies administration and maximizes resource utilization by abstracting the datacenter into a single giant computer. It allows for the running of Docker images.

  Mesosphere Jupyter Notebook server provisions pre-configured Jupyter Notebooks. This allows scientists to access the environments they are familiar with while optimizing resource utilization across the datacenter.

- **Kubernetes**

  Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.

- **Slurm**

  Slurm is an open-source workload manager designed specifically to satisfy the demanding needs of high-performance computing.

- **Data Center GPU Manger**

- NVIDIA Data Center GPU Manger is a suite of tools for managing and monitoring GPUs in cluster environments. It includes active health monitoring, comprehensive diagnostics, system alerts, and governance policies including power and clock management. It can be used standalone by system administrators and easily integrates into cluster management, resource scheduling, and monitoring products from NVIDIA partners.

- **RAPIDS**

  RAPIDS are a collection of libraries for running end-to-end data science pipelines completely on the GPU.

This solution supports running on the common operating systems associated with machine learning:

- **Ubuntu**

  Ubuntu is an open source software operating system that runs from the desktop, to the cloud, to all your Internet connected things.

- **Red Hat Enterprise Linux (RHEL)**

  Using the stability and flexibility of Red Hat Enterprise Linux, reallocate your resources towards meeting the next challenges instead of maintaining the status quo. Deliver meaningful business results by providing exceptional reliability on military-grade security. Use Enterprise Linux to tailor your infrastructure as markets shift and technologies evolve.

- **Microsoft® Windows Server® 2016**

  Microsoft Windows Server is a multi-purpose server that increases the reliability and flexibility of your server or private cloud infrastructure.

This solution supports GPUDirect which is also called GPU RDMA. GPUDirect allows communication between GPUs on different computers, bypassing the CPUs. This can allow for improved performance over standard TCP communications when using multiple nodes. Using GPUDirect requires special configuration of the operating system and the switches. GPUDirect can increase performance when multiple nodes are used.

Accessing Learning Data

There are many ways to store and access data. If you are using a cluster, then one of the shared methods is needed:

- Local Storage – all data is copied to the local storage devices outside of the ML\DL code.
- NFS – this allows you store data in a central location.
- Remote Data – ML\DL code can directly access remote data. This can be a remote database, a data stream, or HDFS.
- Shared Storage with a clustered file system – this option uses shared SAN or shared NVMe over fabric. A clustered file system, like GFS2 is used so all nodes can access the data.
- Distributed storage and file system – GlusterFS or BeeGFS. These file systems spread the data across local storage on all the nodes and it can be treated as one unit of storage.

Caching the data can improve performance. ML\DL will take one, usually small, set of data and process it tens, hundreds, or even thousands of time. Reducing the data access time for each loop by caching the data in memory or fast local disk can have a significant impact. If you are using a file system, the operating system can perform file level caching for the data that fits in memory.

You can have ML\DL cache the data. This would require coding to make use of these features. It has the following advantages:

- Works with all data sources
- Can cache prepared data, which will improve performance
- Can cache the data in whatever underlying method you prefer memory, local storage, SAN, or NVMe over Fabric

Whether using a single standalone, multiple standalone servers, or a cluster of servers, all of these options can meet your needs.

Solution Details Hitachi Advanced Server DS225

This solution supports multiple configurations to meet your needs. Table 1 shows the options available. When performing deep learning, the GPUs will run at near full power. Because of that demand, this solution is only supported on 3-Phase power.

TABLE 1. DS225 COMPONENTS

Component	Description
DS225	▪ 1 × DS225 Chassis
Power Supply	▪ 2 × 2200 watt 3-Phase
CPU	▪ 2 × Intel 6154 Gold (18C 3.0GHZ, 200W)
Memory	▪ 512 GB – 8 × 64 GB DDR4 R-DIMM 266MHz ▪ Option of sizes from 64 GB to 1.5 TB are available
RAID Controller	▪ When not using a Mellanox ConnectX-5 an option is to have an LSI SAS QS3516 with SuperCap
Storage devices	▪ 480 GB SATA 6 Gbps SSD ▪ 960 GB SATA 6 Gbps SSD, ▪ 1.92 TB SATA 6 Gbps SSD ▪ 3.84 TB SATA 6 Gbps SSD ▪ Device models are subject to change
Operating System Devices	▪ 1 or 2 × 960 GB SATA 6 Gbps SSD
Data Devices	▪ 0 - 7 × 1.92 TB SATA 6 Gbps 1DWDP SFF SSD, S4500 ▪ Option of zero to seven storage devices of any size

Network Card	▪ 1 × Mellanox ConnectX-4 LX EN Dual Port 10/25 OCP Mezzanine GB Ethernet adapter
	▪ One of
	  ▪ 1 × Mellanox ConnectX-4 LX EN Dual Port 10/25 PCIe Ethernet adapter
	  ▪ 1 × Mellanox ConnectX-5 Dual Port 100 GB PCIe Ethernet adapter

When sizing memory it is recommended to have memory to be at least as large as the total memory for all graphic cards. For larger datasets, choosing a larger system memory size can have a significant improvement on overall performance.

When performing ML/DL the same set of data is run through the learning code multiple times. If there is enough memory to allow the file system or DL framework to cache the data there is minor impact to performance no matter how the files are accessed: local, remote, 10 GbE connection, or 100 GbE connection.

If your data is stored in something like a Hadoop Distributed File System (HDFS) where the file system does not automatically cache it, TensorFlow allows you to specifically cache the data in either memory or on disk.

Storage is determined on individual needs. Often the operating system disks will provide enough storage to hold all the data used in the ML\DL system.

Advanced Server DS120

The DS120 server configuration is used for both the Pentaho server and hardware management server. Table 2 shows the configuration of these two nodes.

TABLE 2. DS120 COMPONENTS

Component	Description
DS120	▪ 1 × DS120 Chassis
Power Supply	▪ 2 × PSU - POWER SUPPLY 800W Titanium
CPU	▪ 2 × Intel 6154 Gold (18C 3.0GHZ, 200W)
RAID Card	▪ 1 × LSI SAS QS3516 for 2.5" (R5) Mezz
Network Interfaces	▪ 1 × PCIe - Intel XXV710 SFP28 10/25 GbE ▪ 1 × PCIe - Mellanox ConnectX-5 Dual Port 100 GB PCIe Ethernet Adapter

Storage

For an optional shared staging area this solution uses a VSP F700 and a pair of HNAS. The standard hardware configuration for the major components of the external storage is shown in Table 3.

TABLE 3. STORAGE SUBSYSTEM

Component	Description
Storage	▪ 1 × F700 Chassis ▪ 16 × 32 GB DIMMs -512GB cache ▪ 4 × 16 Gbps FC port ▪ 1 × 1U Server (SVP) ▪ 2 x Storage trays ▪ 48 × 1.92 TB SSD – 24 per tray
HNAS	▪ 1 × 4100 HNAS pair

The storage is configured as follows:

- 3 × (6+2) RAID groups per tray
- 1 × HDP Pool
- 4 × HDP Volumes exported to HNAS 4100 nodes
- 1 × HNAS Pool
- 1 × HNAS NFS export

Networking

This solution supports having three networks as shown in Figure 7. The networks are:

- GPU Network – a 100 Gb network used for communication between the nodes when doing multi node training. It allows you to perform GPUDirect RDMA communication. This network uses a Mellanox ConnectX-5 NIC.
- OS Network – a 10 Gb network used for standard OS access by the data scientists, access staging data area, access corporate data, and all non-GPU communication.
- Management Network – a 1 Gb network used for out of band management of the solution.

Figure 7

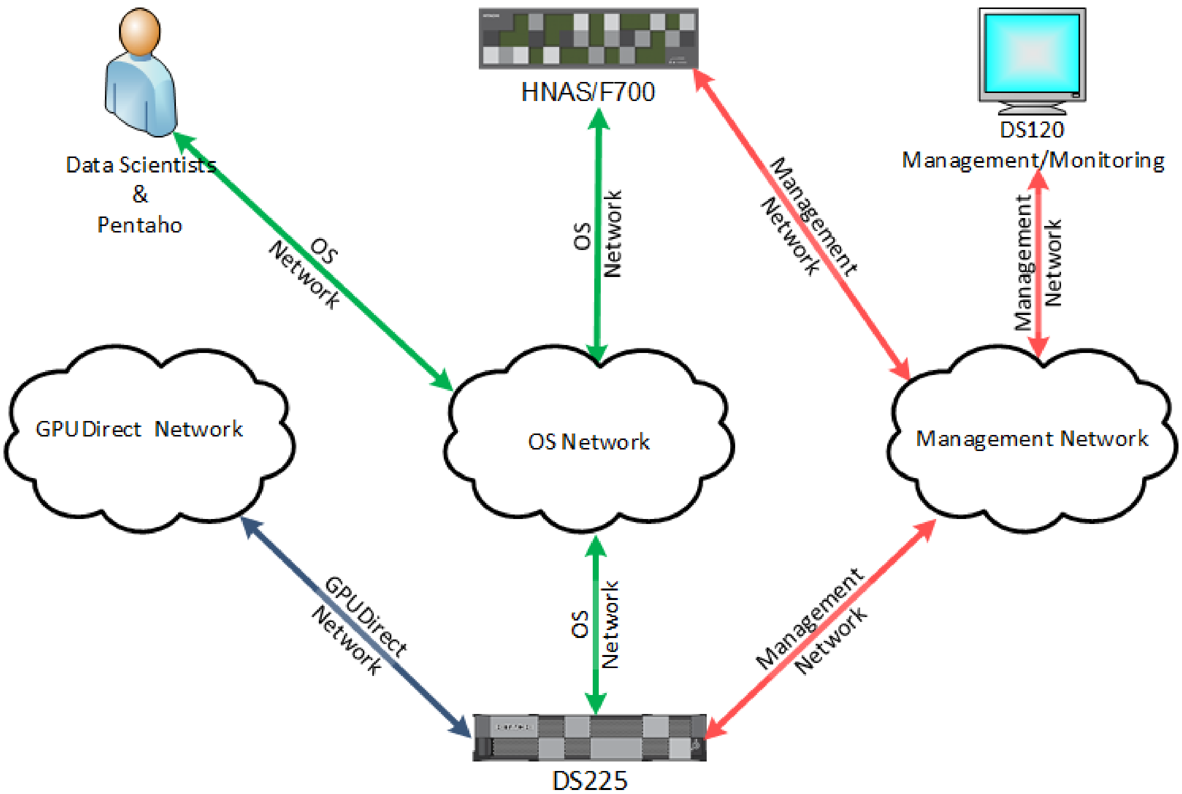


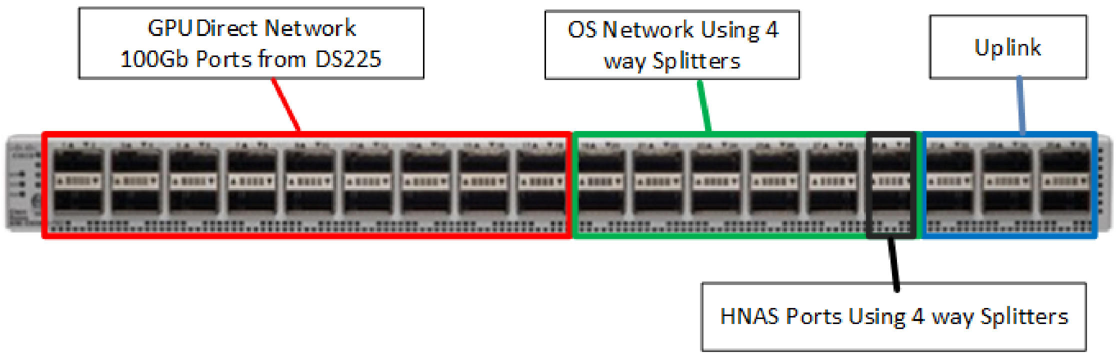
Table 4 lists the ToR switches. If a 100 GbE network isn't needed the 2 × Cisco Nexus 93180YC-EX can be used.

TABLE 4. NETWORK SWITCHES

Component	Description
Management Switch ToR	▪ 1 × Cisco Nexus 3048
ToR Switches	▪ 2 × Cisco Nexus 9336C used for both 100 Gb GPU and 10 Gb OS networks
10/25 GB ToR Data Switch	▪ 2 × Cisco Nexus 93180YC-E/FX can be used when a 100 Gb Network isn't required. Provides a 10 Gb network

The ports on the Cisco Nexus 9336C switches are shared between the networks as shown in Figure 8.

Figure 8

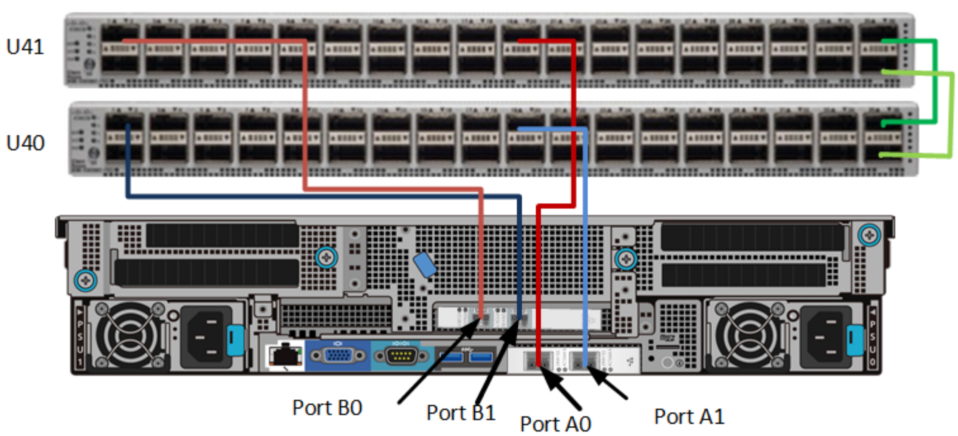


The connections between the nodes and the network switches are as follows:

- The on-board LoM port is connected to the out of band management network on the Cisco Nexus 3058 switch.
- For the 100 GbE network on the DS225, the ConnectX-5 NIC is connect to ports on both Cisco Nexus 9336C switches using ports reserved for the GPU Network.
- For the 25 Gb network on the DS225, the ConnectX-4 NIC uses splitters to connect to both Cisco Nexus 9336C switches using ports reserved for the OS Network.
- The four network ports on the HNAS units are connect to splitters on two ports on each Cisco Nexus 9336C accessing the OS Network.
- For the 25 Gb network on the DS120, the ConnectX-4 NIC uses splitters to connect to both Cisco Nexus 9336C switches using ports reserved for the OS Network.
- The ports reserved for Uplink are used to connect customer switches, spine switches, or other DS225 server racks.

The connections for the DS225 GPU and OS networks are showed in Figure 9.

Figure 9



Racking

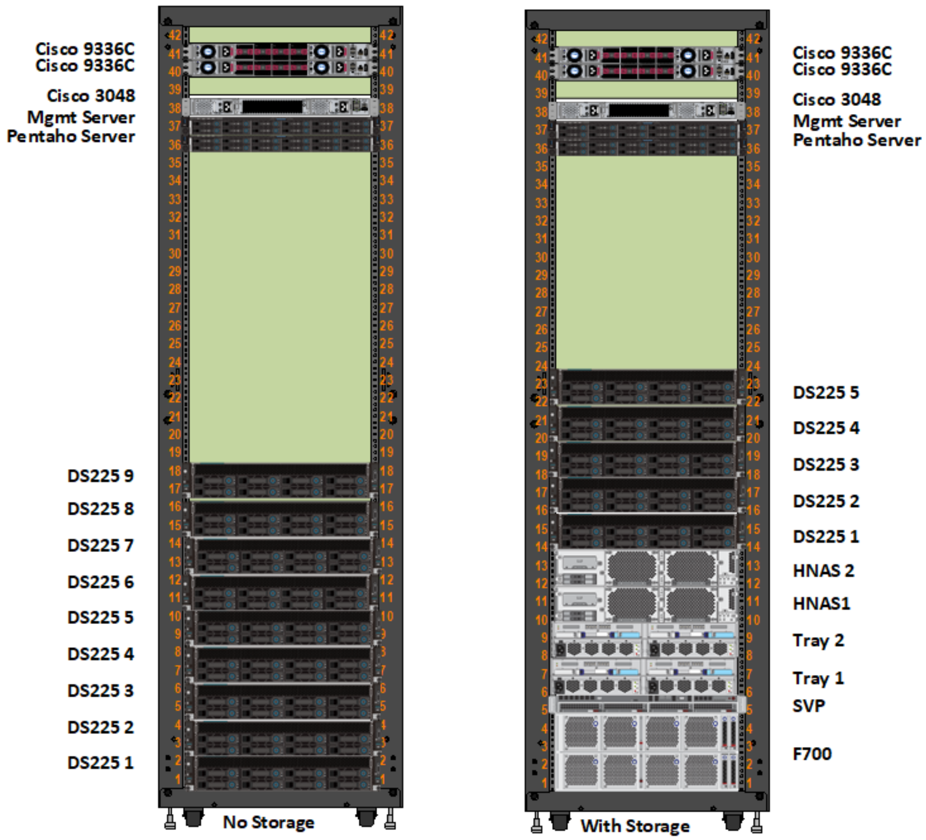
When racking, the storage unit is placed at the bottom of the rack followed by the DS225 nodes. The ToR switches, management server, and Pentaho server will be placed at the top of the rack. Figure 10 shows the layout of a rack with and without storage system.

NOTE: When doing deep learning the GPUs can run at near maximum power consumption for an extended length of time.

The power requirements of a DS225 node have the following impact on racking:

- There can only be one DS225 per circuit
- 3-Phase power is required
- To ensure proper cooling, it is recommended that filler panels are used in all empty slots

Figure 10



Validation Testing

Hardware Test Environment

Table 5 lists the hardware that was used in this validation testing.

TABLE 5. TESTED HARDWARE

Component	Description
Chassis	■ 2 × DS225
GPU	■ 4 × NVIDIA TESLA V100
Memory	■ 24 × 64 GB DIMMS- 1.5 TB total
CPU	■ 2 × Intel 6140 18 core; 135W; 2.3GH
Storage	■ 1 × 900 GB SSD for boot device ■ 3 × 3.4 TB SSDs for data
Network Cards	■ 1 × Mellanox ConnectX-4 LX EN Dual Port 25GbE OCP Mezzanine ■ 1 × Mellanox ConnectX-5 Dual Port 100GbE PCIe
Shared Storage	■ 1 × F700 ■ 2 × disk trays ■ 48 × 1.8 TB SSD ■ 2 × HNAS
Switches	■ 2 × Cisco Nexus 9336C ■ 1 × Cisco Nexus 3048

We require a BIOS setting that allows greater than 4 GB of memory-mapped I/O (MMIO). This setting is enabled by default on the BIOS setting of the servers. To verify or change this, use the following option in BIOS settings:

- `Advanced > PCI Subsystem Settings > Above 4G Decoding [Enabled]`

Software

This solution supports using multiple operating systems and frameworks. This reference architecture focuses on testing on Ubuntu with TensorFlow.

Table 6 provides a partial list of the software used in this solution. With a TensorFlow project, many other packages may be required for your specific code. For testing purposes, a bare metal deployment was used.

TABLE 6. CORE SOFTWARE

Component	Version
BMC	4.23.06
BIOS	3a10.h10
NIC Firmware	ConnectX-4: 14.23.1020 ConnectX-5: 16.23.1020
NIC Driver	5.0.0
Ubuntu	18.0.4 (LTS)
TensorFlow	1.13.1 GPU
Python	3.6.8
Nvidia Video Driver for Telsa V100	418.67
Nvidia CUDA	10.1
Nvidia cudNN	9.1.85
Miscellaneous other Python packages	As needed
Open SSH	1:7.6p1-4
Open MPI	2.1.1

ResNet-50 Benchmark

There are many options to install TensorFlow: pip3, docker image, build from source, or Anaconda. For these tests TensorFlow was installed using pip3 on bare metal.

Deep residual networks, or ResNets for short, provided the breakthrough idea of identity mappings in order to enable training of very deep convolutional neural networks. This test runs an implementation of ResNet for the ImageNet dataset written in TensorFlow. The code for this test can be found in TensorFlow's github: https://github.com/tensorflow/benchmarks

Note — When testing with these benchmarks on TensorFlow, make sure that you get the version of the test that matches the version of TensorFlow you are using.

To facilitate multiple node training password-less SSH was configured between the two nodes.

The tf_cnn_benchmark script ran using the ResNet50 model with the ImageNet dataset. We used a batch size of 100 and both local and distributed tests were launched using Open MPI. During the runs the test was monitored with nvidia-smi. Figure 11 and Figure 12 show a sample of the output during the 2 node test. The average GPU utilization for all runs was greater than 93%.

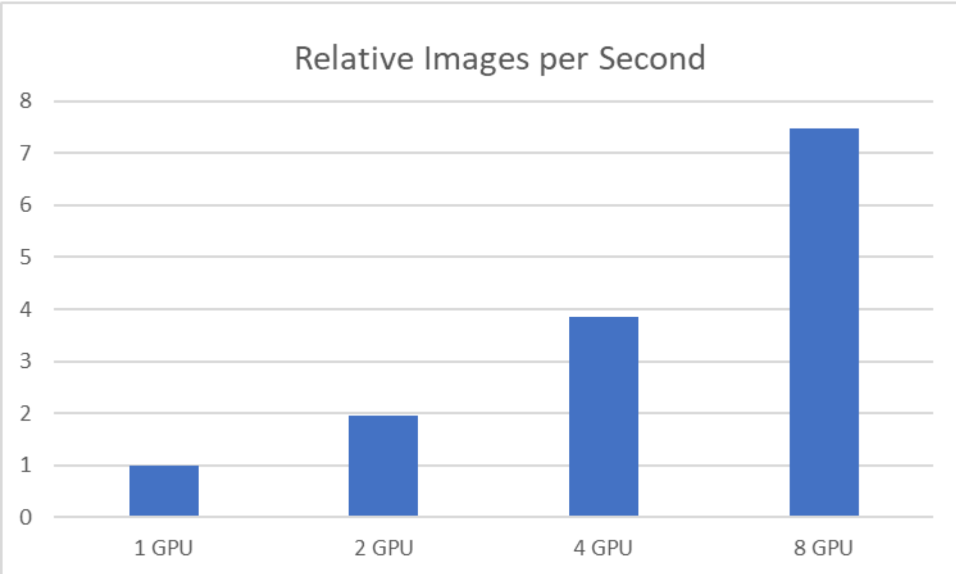
Figure 11

```
|===============================+======================+======================|
|   0  Tesla V100-PCIE...  On   | 00000000:18:00.0 Off |                    0 |
| N/A   53C    P0   197W / 250W |  15856MiB / 16130MiB |     99%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla V100-PCIE...  On   | 00000000:5E:00.0 Off |                    0 |
| N/A   56C    P0   213W / 250W |  15856MiB / 16130MiB |     95%      Default |
+-------------------------------+----------------------+----------------------+
|   2  Tesla V100-PCIE...  On   | 00000000:86:00.0 Off |                    0 |
| N/A   55C    P0   216W / 250W |  15856MiB / 16130MiB |     94%      Default |
+-------------------------------+----------------------+----------------------+
|   3  Tesla V100-PCIE...  On   | 00000000:AF:00.0 Off |                    0 |
| N/A   55C    P0   207W / 250W |  15856MiB / 16130MiB |     96%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|    0      8692      C   python                                     15841MiB |
|    1      8692      C   python                                     15841MiB |
|    2      8692      C   python                                     15841MiB |
|    3      8692      C   python                                     15841MiB |
+-----------------------------------------------------------------------------+
```

Figure 12

```
|===============================+======================+======================|
|   0  Tesla V100-PCIE...  On   | 00000000:18:00.0 Off |                    0 |
| N/A   60C    P0   211W / 250W |  15860MiB / 16130MiB |     94%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla V100-PCIE...  On   | 00000000:5E:00.0 Off |                    0 |
| N/A   62C    P0   221W / 250W |  15860MiB / 16130MiB |     95%      Default |
+-------------------------------+----------------------+----------------------+
|   2  Tesla V100-PCIE...  On   | 00000000:86:00.0 Off |                    0 |
| N/A   60C    P0   219W / 250W |  15860MiB / 16130MiB |     94%      Default |
+-------------------------------+----------------------+----------------------+
|   3  Tesla V100-PCIE...  On   | 00000000:AF:00.0 Off |                    0 |
| N/A   60C    P0   233W / 250W |  15860MiB / 16130MiB |     96%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|    0      6231      C   python                                     15841MiB |
|    1      6231      C   python                                     15841MiB |
|    2      6231      C   python                                     15841MiB |
|    3      6231      C   python                                     15841MiB |
+-----------------------------------------------------------------------------+
```

These tests were performed for 1, 2, and 4 GPUs on one server and 8 GPUs across two servers. The data is the relative number of images per second vs the performance of 1 GPU.



The tests with TensorFlow using the tf_cnn_benchmark with resnet50 and ImageNet data show a near linear improvement in the ability to process images. This test has been under development on for years and has many contributors and testers. Unless you are prepared to do the same level of effort, it is unlikely that you will see this same scalability with your code.

On one node, scaling from one to four GPUs requires your CPU code to be able to process the data fast enough so that the GPUs aren't spending a lot of time waiting for data. The test performed here preprocessed the data so that the CPUs only had to load the data. This preprocessing improves the performance of the test.

When distributing across multiple nodes the raw increase in processing images per second may not correlate to the same increase in performance when training to a specific level of accuracy.

Hitachi Vantara

Corporate Headquarters
5355 Augustine Drive
Santa Clara, CA 96054 USA
HitachiVantara.com | community.HitachiVantara.com

Contact Information
USA: 1-800-446-0744
Global: 1-858-547-4526
HitachiVantara.com/contact

MK-SL-140-01, August 2019