FUJITSU

White paper

A PRIMEFLEX for HPC Reference Model for scaling out AI applications using an HPC infrastructure

Leveraging the scalability and parallelism capabilities inherent in an HPC infrastructure to benefit the training of Deep Neural Networks

Content

What is Artificial Intelligence? 2
- AI is not new 2
- Transforming Industry 2

Deep Learning process 3
- Training the system 4
- The ultimate goal is “inferencing” 4

Where does HPC fit in 4
- HPC scaling for Deep Learning networks 5
  - Parallel Processes 5
  - Data 5
  - Interconnect 5
- HPC technologies for AI 5

Fujitsu’s Approach 6

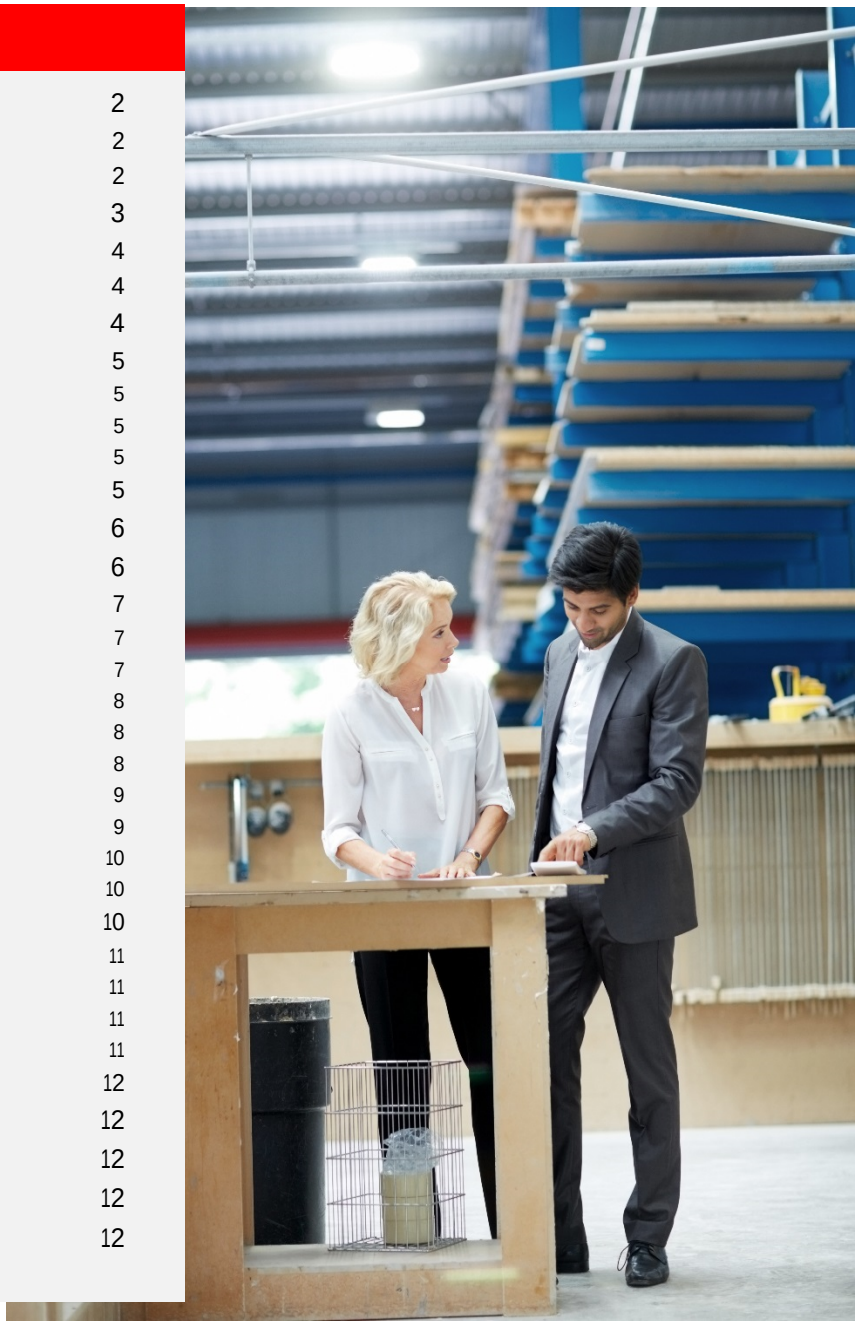
PRIMEFLEX for HPC Reference Models 6
- Core components 7
  - Server systems 7
  - Interconnect 7
  - Storage subsystem 8
    - Metadata server subsystem 8
    - Data storage building blocks 8
    - Storage usage scenarios 9
  - GPGPUs 9
  - Software stack 10
  - AI frameworks 10
- Sample configurations 10
  - Small projects, research projects or POCs 11
  - Moderate project or diverse projects 11
  - Large projects and system 11
  - Storage configurations 11
- Sizing the system 12

Upcoming technologies 12

Advantages of the HPC-AI Reference Model 12

Conclusion 12

References 12



What is Artificial Intelligence?

Artificial Intelligence (AI) has transformed from a specialized IT technology into a mainstream computing paradigm that builds "intelligent" computing frameworks modelled after the human cognitive thinking process. As a generalization, AI can be defined as those applications or algorithms which mimic, in some form, the cognitive functions of the human brain to identify and solve problems. AI includes the processing domains of machine learning and deep learning, in which deep learning is itself a more specific domain of machine learning.

AI is not new

Even if AI has only recently become a mainstream technology it is not exactly new. Its early roots date back to the 1950's and 1960's. However, it is only with the recent advent of Deep Learning, an avenue of AI utilizing Deep Neural Networks (DNN's) at its core to solve problems, that a new era of AI processing has been ushered in which advances the way we apply AI to solving everyday problems.

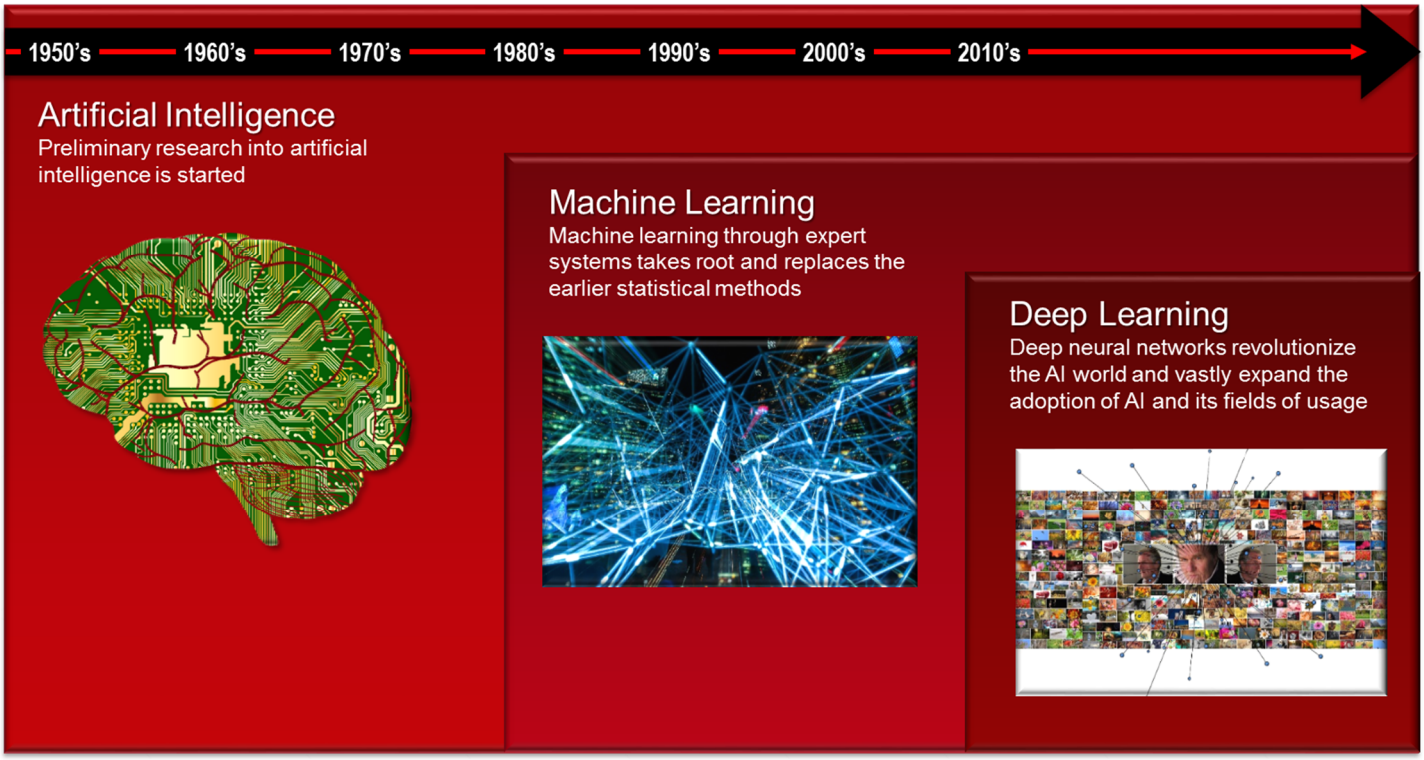


Figure 1 - Stages of AI processing development

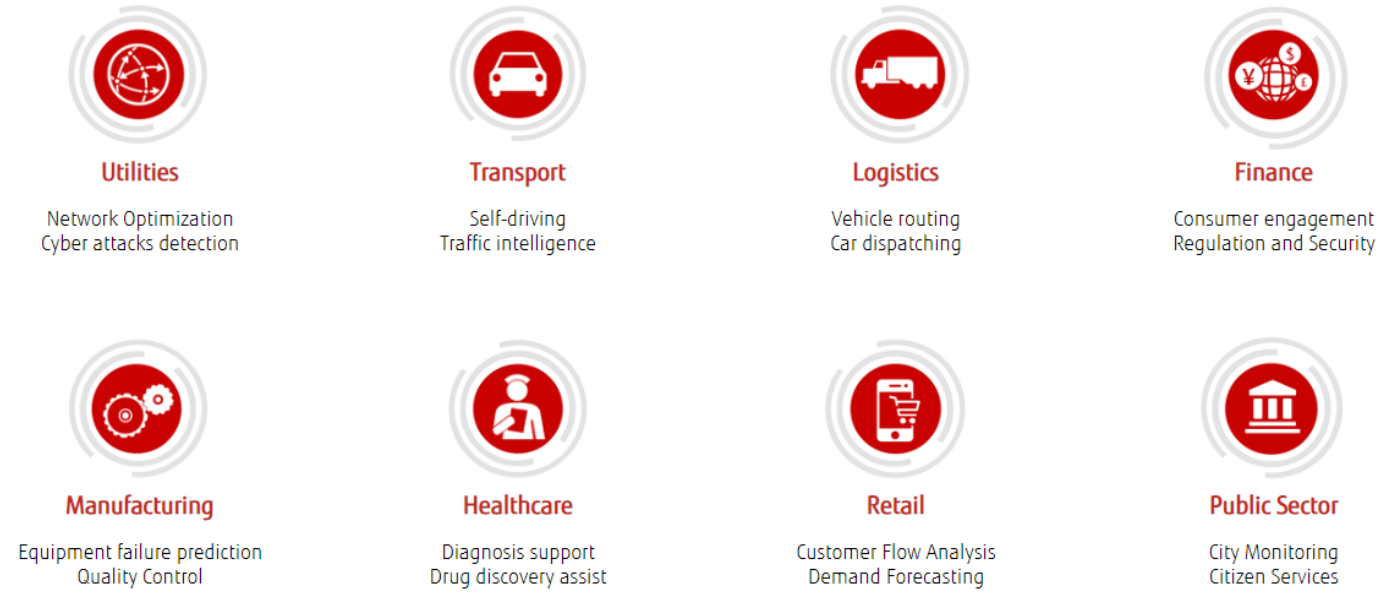
In particular, the most significant advances in this field have taken place in the last few years, when AI really started to take off through the newly derived Deep Learning domain. With the increased availability of parallel computing through powerful GPGPU's and multi-core processors and with the expressiveness of the newly developed DNN's, solution developers have been able to solve problems previously thought too difficult or too complex. Trained properly, a DNN can solve these problems merely by observing huge amounts of data and "learning" from it.

A new era of AI processing has been ushered in which uses Deep Neural Networks (DNNs) to solve everyday problems

Transforming Industry

In the business world, AI is transforming many industries. With its ability to identify patterns and detect anomalies in enormous quantities of digital information it's adding a new dimension to the way we can process information. And once trained, an AI framework constitutes a tireless workhorse for processing many standard tasks. For example – by adding AI to service desks and call centers, staff are freed from performing low-level, monotonous tasks, enabling them to focus on addressing more complex problems/requests, or for delivering an enhanced customer experience leading to improved customer care.

Many market segments have already been influenced by AI in their everyday operations. The following figure highlights some of the major areas where AI is already used and some of the fields where it is applied.



In this short introduction we have briefly explained how Deep Learning has enriched the scope of AI processing and some of the areas in which it is being used today.
However, in this paper we wish to highlight how the scalability features associated with an HPC architecture can be an effective means to scale-out Deep Learning frameworks, enabling them to address more complex and much larger problems.

Before we dive into that we'll briefly explain the processes involved in getting a Deep learning framework ready for use and then see how HPC can contribute to scaling and growing the capabilities of these frameworks. Finally, we will propose some reference architectures based on key HPC technologies that bring benefits such as reduced time to market and higher return on investment when used for training AI frameworks.

Deep Learning process

Loosely speaking the Deep Learning process consists of 2 major steps, training the Deep Neural Network and then using the trained model to infer the desired result. The "Training" process for a DNN is simply an educational or learning process and similarly to a human's training, a neural network gets an education to learn how to do a job. This step relies on large quantities of data which are feed into the model. The critical factor in training a DNN is to achieve a reliable level of accuracy. This ensures the result will be dependable and we can have a high degree of confidence with it.

The critical factor in training a DNN is to achieve a reliable level of accuracy so that you can have confidence in the result

Once trained, the neural network is then allowed to go to work, well at least in the digital world, using what it has learned.

The "Inference" process is where the capabilities learned during the DNN training are put to work. Inference can't happen without first training the framework, because, just like humans, that's where the learned information comes from, that's where the DNN is taught what it is to do.

Another very interesting analogy with human training is the fact that once trained, a neural network does not need all the infrastructure of the training environment to be able to run the inference process, it just needs the trained model. Similarly, you and I don't carry around a set of library books or a group of our teachers, because once we have been educated we have our own trained inference capabilities.

Training the system

So, as we have just discussed before letting an AI system loose in the real world, it has to "learn" (or be taught) how to recognize the things it has to do, whether that be to recognize images, the spoken word, identify diseases in cells, or suggest the type of restaurant you should go to next. This training process helps the neural network categorize and classify information from the patterns or objects it observes just like humans do when we see things and experience things. By feeding the neural network with a huge set of information that is relevant to its "education" (or what we are trying to teach it), it builds up a vast array of attributes it can use to categorize and identify individual objects. And given enough information, this "learning" process allows the DNN to identify new sets of objects it has not encountered before by referring to its "trained" network of information.

It should therefore, not come as a surprise to know that Deep learning networks, in general, need to be feed with large quantities of data in order to be accurately trained. An early example of this is the "Google Brain" which learnt to recognize cats after being shown over ten million images.

Given enough information, the "learning" process allows a DNN to identify new objects it has not encountered before

The ultimate goal is "inferencing"

Training is a vital and necessary step that lays the foundation for the deep learning system to do what it was really intended to do, "infer" from its' knowledge. Without being conscious of it, our brain constantly tries to make sense of the things we come into contact with by referring to existing information it has seen in the past (i.e. all the images of objects we have seen during our life). This is an "inference" process which we do naturally. Thus, when we encounter something new, our brain compares this to what it already knowns to help determine or understand what this new object could be.

In an analogous way, Convolutional Neural networks determine newly encountered objects from trained data since they are modelled on the human visual cortex process. From their trained data they "infer" what an unencountered object may be. If we provide enough information in the form of images, voice samples, driving scenarios or whatever happens to be the relevant data for the neural network we are building, then the more accurate the result of the "inference" is likely to be coming out of the trained network.

As DNN datasets increase in size and the DNN grows in complexity, a cluster of machines with high-performance computing architecture becomes essential

Where does HPC fit in

Deep Neural Networks are already growing at a startling rate and the complexity of the models are increasing. In addition, the size of the datasets they require as part of their training can be huge. Increasing the data size and coding more complex DNN models has an effect of placing increased demands on computational and memory resources, making it virtually essential to use a cluster of machines with high-performance computing architecture to complete the training process efficiently.

The HPC market has already faced many of the challenges regarding the scalability of processes and data. In particular, HPC frameworks have been using parallelism for many years as a means to increase the scalability and throughput of the underlying physical resources.

HPC systems have 10's, 100's, 1000's or more servers, all similarly configured and connected to a high-speed network enabling both low latency high-bandwidth inter-process communication as well as high speed access to a common single namespace storage layer.

Figure 2 shows a typical HPC system in which compute nodes share a common storage area and a high-speed low-latency interconnect.

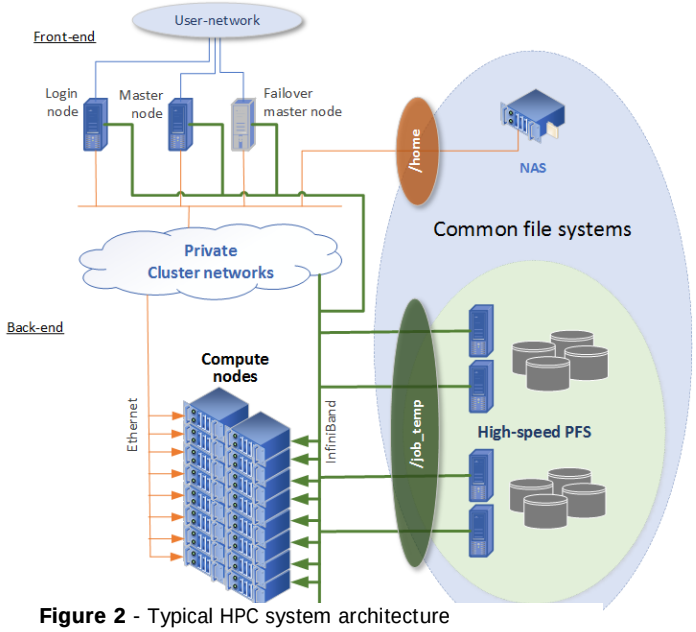


Figure 2 - Typical HPC system architecture

HPC scaling for Deep Learning networks

The specific features of HPC that are of interest to AI frameworks include:

- Multi-node process parallelism
- Data access via a high-speed parallel file system with a single uniform namespace
- High-speed interconnect to improve inter-node process communication performance

Parallel Processes

A single HPC application typically consists of multiple parallel processes communicating with each other across hundreds or thousands of servers. The **Message Passing Interface** (or MPI) programming paradigm is used to enable multiple processes to communicate and synchronize their activities in a parallel fashion. This is a fundamental technology to enable parallelism of applications which in turn drives application scalability.

Data

In addition, HPC applications often read and write Terabytes of data to achieve their results. A standard single server-based filesystem could not cope with the level of I/O throughput required by the HPC subsystem, therefore considerable effort has been spent developing "parallel" file systems, such as Lustre, BeeGFS and IBM's Spectrum Scale™, where every compute node is a client and multiple I/O server nodes are configured to serve the file system data. These client/server based parallel file systems allow hundreds or thousands of clients simultaneous access to a single file system. Distributing or striping file data blocks across the storage of the multiple I/O servers ensures maximum I/O throughput. In addition, the I/O performance of the file system scales linearly as more I/O servers are added to the subsystem.

Typical HPC file systems have between a few GB/s to 100's of GB/s throughput. Access to the filesystem is via the low latency high-speed interconnect with each node having up to 7GB/s data throughput to the storage layer.

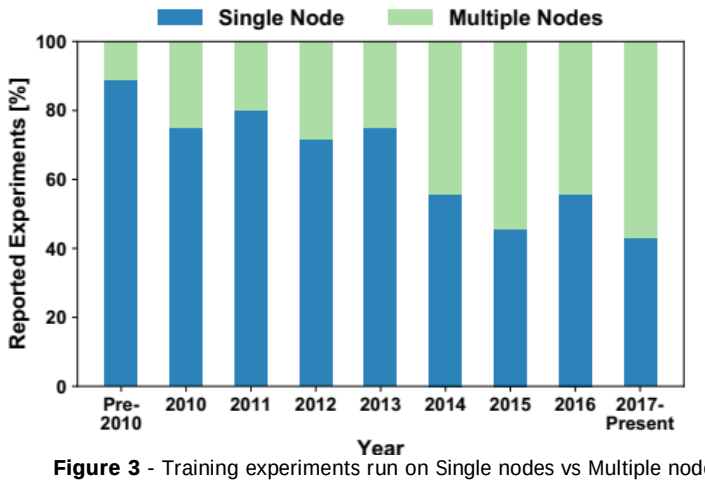


Figure 3 - Training experiments run on Single nodes vs Multiple nodes

Interconnect

And finally, to ensure the inter-process communication (MPI) and file system activity can occur at speed and with the lowest possible latencies, a high-speed interconnect is used based on InfiniBand or Omni-Path. These networks can process messages at 100Gbps or 200Gbps with hardware latencies in the order of 60-100 ns and with MPI latency of about 1 µs.

HPC technologies for AI

The key to unlocking the potential of deep and machine learning systems is the ability to feed them the massive amount of data they need to drive the training process and enable training processes to run in parallel across multiple servers.

This is important, because the training phase is actually the most challenging aspect for neural networks, particularly from the perspective of the processing power they require.

Whereas a single server with GPU's provides a certain level of parallelism, there is obviously a physical limitation to the level of scalability you can achieve from a single server.

In a study called "Demystifying Parallel and Distributed Deep Learning: An In-Depth Concurrency Analysis" (Ben-Nun & Hoefler, 2018) into research projects using AI,

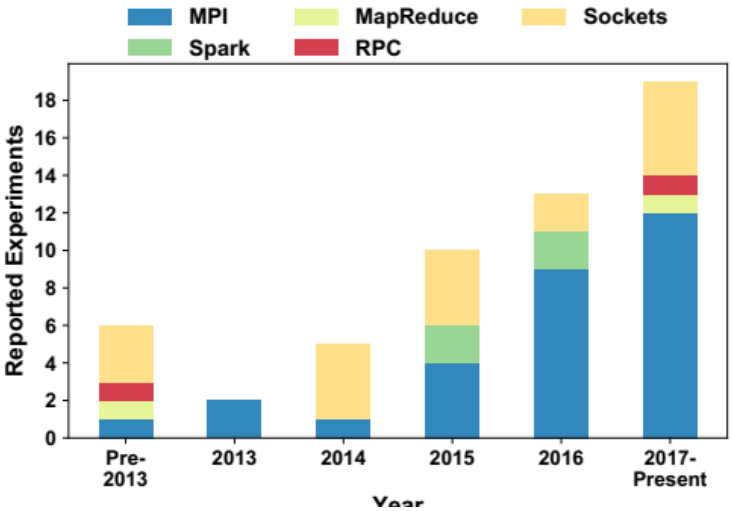


Figure 4 - Increased use of MPI to parallelize the training process

there was an increasing trend for using multiple nodes as part of the training process (see Figure 3) as well as observing a corresponding increase in the adoption of MPI based frameworks as a means to run the training stage (see Figure 4).

Given the fact that all the major deep learning frameworks such as Caffe, Theano, Tensorflow and others offer support for parallel processing via MPI this should not be a surprising fact. And if nothing else, this fact gives further evidence that Deep Learning frameworks are becoming complex and large enough to warrant the capability to run their processes in parallel over multiple nodes.

In most cases we are solely applying these HPC architectures to the DNN training process, which requires significant processing power, however in some cases the inference engine may also be large enough and complex enough to warrant the use of a parallel processing environment.

Therefore, in order to accelerate the processing speed for the training process, computational processes can be distributed across multiple machines. In this scenario the most important metrics revolve are interconnect latency, bandwidth and message-rate. In this respect, InfiniBand provides higher bandwidth, significantly lower latencies and higher message rates compared to Ethernet networks and therefore becomes the solution of choice when the training model is large.

Fujitsu's Approach

We primarily focus the reference architecture for the purposes of the training step, though for larger models a smaller scale HPC infrastructure may also provide benefits for the inference.

The approach includes:

- Identify technologies to:
  - Build a platform architecture by utilizing existing infrastructure resources, focusing on simplicity and the best price / performance ratio
  - Support for data distribution across intra and inter-node memory hierarchies
  - Support the combination of HPC workloads, including serial to massively parallel applications, simultaneously with the AI frameworks
  - Identify the processes needed to make existing applications run effectively on systems with large scale many-core processors
- Maintain portability of the solution
  - It is mandatory to ensure that the benefits of these efforts are not limited to the current platforms but can be re-used and expanded for future generations
  - Capitalize on architecture design and system optimization techniques and internal "know-how" of Fujitsu engineers with high end computing techniques
  - Build a software stack enforcing language/runtime interoperability as well as the most efficient production environment
- Capitalize on Fujitsu's years of business experience with both Integrated systems on HPC as well as AI
  - Strong link and cooperation with major HPC players globally.
  - Market recognition of Fujitsu as a major player in AI solutions.
  - With the Fujitsu Dynamic Infrastructures approach, Fujitsu offers a full portfolio of IT products, solutions and services, ranging from clients to data center solutions, Managed Infrastructure and Infrastructure as a Service.

PRIMEFLEX for HPC Reference Models

The reference components and sample configurations presented address ways to fulfil the processing needs for training of an AI model. They are particularly pertinent when large quantities of images, voice or video recordings are required to train the network. Thus, the HPC-AI Reference models use the best architectural components of an HPC infrastructure in combination with the required AI frameworks.

The components described here come from either the Fujitsu product catalog or from 3rd party vendors using products commonly used to construct HPC infrastructures. They are combined in ways to build an efficient and scalable environment.

It is important to note that the HPC architecture makes it very easy to scale-out an existing environment, so even when starting out with a small initial installation it can easily be expanded as your demands for higher processing levels increases.

Core components

The core components used to construct the overall solution include the following:

- Server systems
- Interconnect
- Storage subsystem
- GPGPU's
- Software Stack
- AI frameworks

The main configuration options and usage notes for these components are highlighted in the following paragraphs.

Server systems

Server options are based on either the 2U PRIMERGY CX400/CX2570 chassis/servers combo or the 2U RX2540 server.

Option	For air-cooled environments	For water cooled environments	
1	-	**PRIMERGY CX400 chassis + 2x CX2570** 2 servers per 2U, each with dual-proc and 4 GPU's per node using NVLINK, 8 GPU's per 2U. Limited options and moderate memory configurations	
2	**PRIMERGY RX2540** 2U dual-proc server, 2 GPGPU capable, 2 GPUs PCIe Many configuration options are large memory configurations	**PRIMERGY RX2540** 2U server, 2 GPGPU capable, provides 2 GPUs per 2U, with many configuration options are large memory configurations	

Notes:

1) The water cooled CX400/CX2570 with 4GPU's per node offers the highest density and performance for a 2U configuration.
2) For maximum configuration options and extremely large disk and/or memory configurations then the RX2540 is the best choice.

Interconnect

The high-speed interconnect is constructed using either Mellanox InfiniBand (IB) or Intel OmniPath (OPA). Both implementations support speeds of 100Gbps and Mellanox recently added a 200Gbps IB product range for extreme performance requirements. In our opinion we are yet to see the need for the 200Gbps data rates in the AI space just yet.

Option	Description
1 or 10Gbps	Ethernet
100 Gbps	Mellanox EDR InfiniBand Intel OmniPath
200Gbps	Mellanox HDR InfiniBand

Notes:

1) Ethernet is viable for small cluster sizes of up to 8 nodes
2) For larger clusters 100Gbps InfiniBand or OPA should be used
3) Mellanox HDR would only be a consideration in extreme processing requirements

Storage subsystem

The storage subsystem is centered around an HPC parallel file system (PFS), built using either Lustre or BeeGFS or IBM's Spectrum Scale™. The main reasons for using this are:

- High-speed data access
- Single namespace ensures uniform data access from any client (compute) node
- Easily redundancy without the need for data erasure encoding
- Central management of all data
- External access via NFS if needed

Due to the single namespace capability, data locality is never an issue and data does NOT have to be moved to the compute nodes where the AI process runs. This significantly simplifies the execution environment and creates an infrastructure that enables concurrent execution of multiple AI frameworks or the concurrent execution of AI and HPC workloads together.

The storage subsystem can be configured and scaled-out as needed. As previously mentioned the I/O subsystem performance scales linearly when adding complete building blocks to the subsystem.

The storage building blocks consist of:

a) Metadata server subsystem
b) Data storage subsystem

Metadata server subsystem

Item	MetaData server	MetaData storage
Model	Fujitsu PRIMERGY RX2540, dual proc server, 64GB memory, 2x Intel Xeon Gold processors	ETERNUS DX200 CE, 2U controller shelf with up to 24x 2.5" SSD or disk drives, up to 8 host connections
Storage configuration	1x 250GB M.2 SSD	4x 960GB SSD RAID 1+0
Interfaces	2x FC 32Gbps to DX200 storage 1x IB 100Gbps to compute node network	SCSI interface to controller shelf
Storage configuration	1x 250GB M.2 boot disk	4x 960GB SSD RAID 1+0 for meta data
Storage capacity	250GBytes	1.8TBytes
		

Data storage building blocks

The storage subsystem is very specific to each project and installation therefore we will highlight the building blocks that can be used for constructing the storage

Item	DX200 controller shelf	DX200 disk enclosure	DX200 HD-DE (high density disk enclosure)
Height	2U	2U	4U
Host interfaces	Up to 8 @ 32Gbps FC	-	-
Interface to storage	2x SAS 12Gbps		
Shelf type	Controller shelf (with disks)	Standard extension shelf12x 3.5" disks	High density disk enclosure extension
Disk type	Up to 12x 3.5" (up to 24x 2.5" as option)	Up to 12x 3.5" (up to 24x 2.5" as option)	Up to 60x 3.5" (up to 60x 2.5" as option)
Disk configuration	2x 12TB NLSAS as host spares		60x 12TB 6x RAID6 (8D+2P)
Capacity	Used as hot spares only		500TB
Interconnect	InfiniBand 100Gbps		
			

Storage usage scenarios

Direct HPC storage

When there is no need for a Big Data option then the file system is used as a standard block device with data access governed by the very traditional POSIX file system interface.

Note, when using Lustre it is necessary to implement multiple MDS servers and spread the training inputs across different directories so that Metadata I/O operations can be parallelized across the multiple metadata servers. As an alternative, you can consider using the new Lustre DoM (Data on MDS) feature for speeding up access for small files.

HDFS based on HPC storage

For solutions requiring a more specialized interface to storage, then the Lustre, BeeGFS or IBM Spectrum Scale PFS layer can include a HDFS compatible interface enabling it to be used as a Hadoop file storage area. Benefits of this include improved data access speeds over standard HDFS implementations, no need to write data multiple times for redundancy, uniform access from any compute node removing the need to copy data to the node for processing. Using IB/OPA for data access ensures speeds faster than local disks and even faster than the new NVMe based SSD PCIe disks.

Object storage interface

Under some circumstances you may wish to use a specific object interface to your data such as Swift or S3. If so, then we would propose the use of the Fujitsu's ETERNUS CD10000 Reference Architecture solution which is a scale-out data storage solution based on the Ceph distributed object storage platform. Integrated support for Swift and S3 and fundamental features for this storage layer.

It should be noted that unlike Lustre, BeeGFS and IBM's Spectrum Scale™ which are built specifically to optimize high-speed data access, Ceph is designed to be flexible, elastic and provide multiple access methods. Even though it still performs I/O at a fast rate it is not as fast as either Lustre, BeeGFS or Spectrum Scale.

GPGPUs

The GPU's to be used should be capable of supporting the latest small floating point (FP32 and FP16) instructions as well as the new small integer arithmetic (INT16 and INT8) instructions. It has been shown that many DNNs are very well suited to these lower precision operations.

NVIDIA GPU's based on their Tensor cores (NVIDIA TESLA V100) or the new Turing Tensor cores (NVIDIA TESLA T4) are optimized for accelerating matrix multiplication with single or mixed precision and are therefore ideal for AI workloads. As the data processing width decreases from 32 to 16 and then to 8 bits, the processing operations per second doubles each time.

Software stack

The software stack serves 2 basic purposes:

i) To allow admins to install and manage the computing and storage software infrastructure

ii) providing the environment for users run their desired AI frameworks (and HPC workloads if that exists)

The HPC software stack provides a means to deploy and provision any number of compute nodes from a single master node. The images used to provision nodes is built on the master node and then sent over the network to each node. Changes, such as adding new AI frameworks to the software environment, can easily be managed from the master node and then propagated to each compute node.

Classification	Item	Software component
Required components	OS layer	RedHat Enterprise Linux or CentOS
	Cluster software	OpenHPC or Bright Cluster Manager
	Batch Job Manager	SLURM or PBS Professional or PBS Professional OSS or PBS Professional
	Local storage layer	BeeGFS or Lustre or IBM Spectrum Scale™
	Parallelized AI frameworks	Any MPI enabled AI framework such as Caffe, Theano, Tensorflow … etc
Optional components	PFS support for HDFS	BeeGFS: use the “BeeGFS Hadoop Connector” Lustre: use the “Hadoop Adapter for Lustre (HAL)” IBM Spectrum Scale™ Hadoop connector
	Big Data engines	Hadoop, Spark, YARN
	HPC apps	Any HPC applications

AI frameworks

In general, any AI framework can be used on the reference architecture systems. However, ultimately the MPI enabled versions of the AI frameworks should be chosen such as TensorFlow_MPI, Theano MPI and Caffe-MPI. The MPI enabled frameworks ensure the training process can make best use of the multi-node configurations.

Sample configurations

The sample configurations that follow show how to construct solutions based on the general building blocks or components that we have already discussed. You should note that these samples are not sized for any particular problem size so the names should be considered rather generic and purely as an indication of the relative capacity differences between the various sample configurations. A sizing exercise needs to run for each customer situation to determine the precise configurations needed.

In addition, when existing HPC infrastructures are in place, enabling a cluster for AI usually surmounts to adding GPGPU cards into some of the existing compute nodes (if that option is supported) and installing the related AI frameworks. Otherwise, additional GPU enabled nodes may need to be purchased and added to the cluster.

Small projects, research projects or POCs

Item	Air cooling	Water cooling *1
Cluster size	4 nodes	2 nodes
Node type: recommended alternative	PRIMERGY RX2540	PRIMERGY CX400/CX2570
GPUs	8x NVIDIA V100 PCIe 32GB / 8x NVIDIA T4	8x NVIDIA V100 SXM2 32GB
Tensor FLOPS / Turing TOPS (8bit)	896 / 1040	1,000
Memory	256GB / 64GB per node	128GB / 64GB per node
Direct Attached Storage	4TB per node (2x2TB disk)	256GB boot disk
Interconnect	Ethernet 10Gbps	

*1: only valid when being added to an existing water-cooled cluster

Moderate project or diverse projects

Item	Air cooling	Water cooling *1
Cluster size	16 nodes	8 nodes
Node type: recommended alternative	PRIMERGY RX2540	PRIMERGY CX400/CX2570
GPUs	32x NVIDIA V100 PCIe 32GB / 32x NVIDIA T4	32x NVIDIA V100 SXM2 32GB
Tensor FLOPS / Turing TOPS (8bit)	3,584 / 4,160	4,000
Memory	1024GB / 64GB per node	512GB / 64GB per node
Direct Attached Storage	4TB per node (2x2TB disk)	256GB boot disk
Interconnect	InfiniBand 100Gbps	

*1: only valid when being added to an existing water-cooled cluster

Large projects and system

Item	Air cooling	Water cooling
Cluster size	64 nodes	32 nodes
Node type: recommended alternative	PRIMERGY RX2540	PRIMERGY CX400/CX2570
GPUs	128x NVIDIA V100 PCIe 32GB / 128x NVIDIA T4	128x NVIDIA V100 SXM2 32GB
Tensor FLOPS / Turing TOPS (8bit)	14,336 / 16,640	16,000
Memory	4096GB / 64GB per node	2048GB / 64GB per node
Direct Attached Storage	4TB per node (2x2TB disk)	256GB boot disk
Interconnect	InfiniBand 100Gbps	

Storage configurations

The following table highlights some sample combinations of the storage building blocks. These give you an idea of the capacity and rack size required. The precise configuration for a specific project will be determined based on the actual data requirements.

Item	Small	Medium	Large	Very Large
MDS servers (2U)	2	2	2	4
Metadata disk controller shelves (2U)	1	1	1	1
Disk Controller shelves (2U)	1	1	4	8
Disk enclosure shelves (2U)	1	-	-	-
High Density disk enclosure shelves (4U)	-	1	4	8
Storage capacity (raw)	171 TB	513 TB	2.0 PB	4.1 PB
Total rack height units	10U	12U	30U	58U

Sizing the system

The sizing of an AI Training solution relies on a number of parameters that come from the project and are not simple metrics that can be determined by a formula. The time taken to train a deep-learning model can vary enormously from a few hours to weeks or more. And for example, simply knowing the data sizes involved does not allow a system architect to create an optimal solution.

In order to size a solution properly a number of key parameters must be used, such as:

- Which neural network model is to be used
- The number of layers in the neural network
- Expected size of the datasets to be used for the training
- An estimate of the time-to-model (i.e. how much data is likely needed for convergence)
- Project requirement for time-to-model (i.e. when a trained model is expected to be ready for production use)

A sizing exercise must be factored into the early phases of any deployment project incorporating AI technologies. Fujitsu provides a team of AI professionals who can work with customers to determine the optimal solution for their AI needs. This team will rely heavily on the building blocks outlined in this paper to construct an environment that ensures processing and project timelines are met.

Upcoming technologies

Fujitsu will be announcing the release of a system based on its Deep Learning Unit (DLU), which is a domain specific processor built for running AI applications. We have not covered the use of the DLU platform since this paper covers the use of HPC based infrastructures as a multi-purpose environment where HPC and AI workloads cohabitant together.

Advantages of the HPC-AI Reference Model

1) Ensure the AI training process can scale-out to meet demand
   - Based on proven technologies used in HPC to speed up processing, inter process communication and I/O operations
   - Time-to-model for complex or large models can be achieved in a timely manner and project milestones can be met
2) Ability to deploy on dedicated hardware for stand-alone departmental or smaller systems
   - Provides a performance optimized platform for high-speed training
   - Used when high performance AI processing is required and no HPC workload exists
3) Non-intrusive deployment on existing HPC platforms
   - Easily extend the capabilities of an existing HPC installation to incorporate AI based training. The additional packages are compatible with the standard HPC software stack (OpenHPC or Bright Cluster manager)
   - Normal HPC operation is kept un-interrupted
   - Agile infrastructure which can be easily scaled based on the end-user requirements

Conclusion

The use of accelerated computing or high-performance computing on advanced commercialized AI has been increasing and is expected to grow exponentially. In simple terms, AI workloads continue to become more complex and time critical and hence benefit from using HPC resources to obtain parallel execution, fast communication as well as high-speed storage access.

Such approach enables construction of agile systems which deliver effective utilization of both AI and if desired, HPC workloads, leading to optimized TCO, improved ROI and quicker time to market or result.

References

Ben-Nun, T., & Hoefler, T. (2018). *Demystifying Parallel and Distributed Deep Learning: An In-Depth Concurrency Analysis.* arXiv 1802.09941.