Deep Learning using iAbra stack on Dell EMC PowerEdge Servers with Intel technology

Abstract

This whitepaper evaluates the performance and efficiency of running Deep Learning training and inference using iAbra framework on the Dell EMC PowerEdge C6420 server. The objective of this whitepaper is to demonstrate how iAbra on PowerEdge server infrastructure has a new approach to solving both training & inference using heterogeneous environment.

June 2019

Revisions

Date	Description
June 19	Initial release

Acknowledgements

This paper was produced by the following members of the Dell EMC SIS team, iAbra and Intel:

Authors: Bhavesh Patel [Dell EMC], Greg Compton [iAbra], Greg Nash [Intel PSG]

The information in this publication is provided “as is.” Dell Inc. makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

Dell believes the information in this document is accurate as of its publication date. The information is subject to change without notice.

Table of contents

Executive summary

Why

Enterprise needs solutions to business problems from deep learning not more technical challenges. Many challenges exist in providing an integrated platform for deep learning development especially in the IoT era. From semiconductor parts to interconnects and servers through to software libraries and domain-centric data science every layer in the system has important consequences to overall solution business effectiveness. Therefore Dell EMC, iAbra and Intel have combined and optimized best in breed components to offer an end to end off the self-solution, which can kick start enterprise AI programs with solutions not problems, getting you from domain specific data to deploy-able solution in the datacenter or edge

What

The solution based around Dell EMC PowerEdge C6420 high density server offers pre-configured scalability of human and server resources, portability between datacenter and edge, GUI workflow from sample creation through to neural network design evolution then deployment. iAbra's Deep Neural Network algorithm with auto ML built in from the core takes advantage of latest Intel CPUs with AI extensions, interconnects & FPGA's combined with Dell EMC PowerEdge C6420 high density to overcome the performance limitations to practical power efficient self-optimizing AI and model deployment portability

How

Dell EMC, iAbra and Intel solution provides a new paradigm for Deep Learning training and inference, built from the ground up for enterprise needs. Providing your enterprise a pre integrated clustered stack for training and inference kick starts programs by removing uncertainty and time spent choosing myriad of DIY components up and down the solution stack, clear and certain workflow to finding the optimal network to your business problem removes program risks and costs, horizontal cluster scalability provides project speed up with increased cluster size, and clear defined path to deploy-ability with model portability fidelity removes the risks to productization.

1 Overview of Deep Learning

Deep learning consists of two phases: Training and inference. As illustrated in Figure 1, training involves learning a neural network model from a given training dataset over a certain number of training iterations and loss function [1]. The output of this phase, the learned model, is then used in the inference phase to speculate on new data.

The major difference between training and inference is training employs *forward propagation* and *backward propagation* (two classes of the deep learning process) whereas inference mostly consists of forward propagation [2]. To generate models with good accuracy, the training phase involves several training iterations and substantial training data samples, thus requiring many-core CPUs or GPUs to accelerate performance.

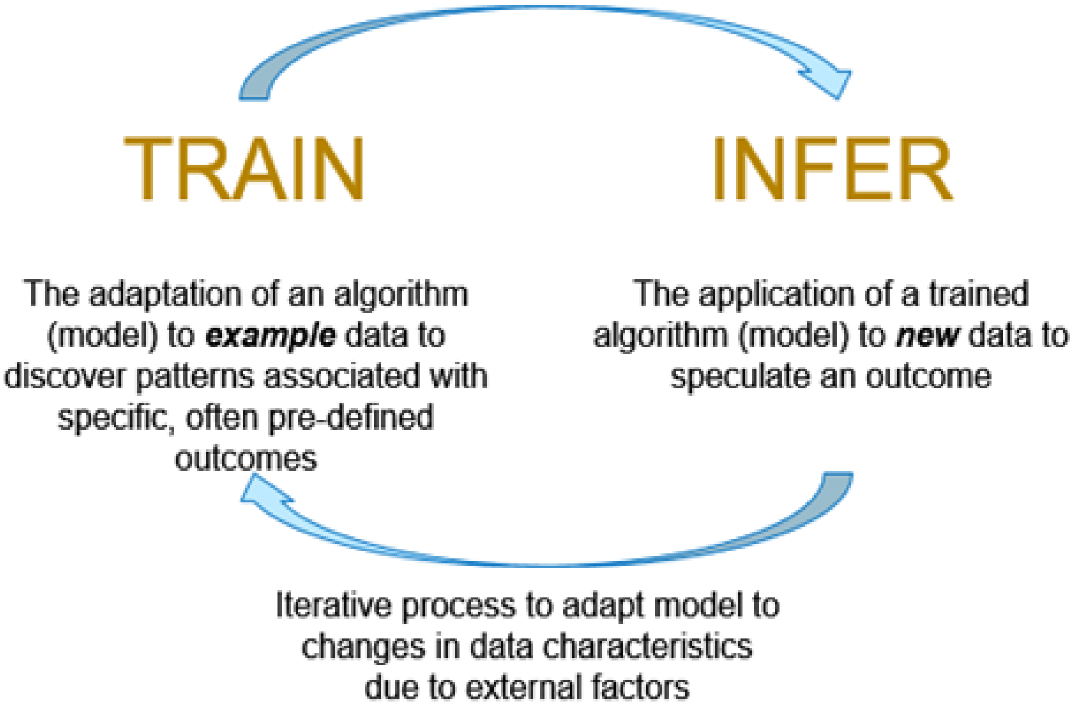

TRAIN
INFER
The adaptation of an algorithm (model) to ***example*** data to discover patterns associated with specific, often pre-defined outcomes
The application of a trained algorithm (model) to ***new*** data to speculate an outcome
Iterative process to adapt model to changes in data characteristics due to external factors

Figure 1 Deep Learning phases.

1.1 Deep Learning Inferencing

After a model is trained, the generated model may be *deployed* (forward propagation only) e.g., on FPGAs, CPUs or GPUs to perform a specific business-logic function or task such as identification, classification, recognition and segmentation. [Figure 2].

For this model deployment and inferencing stage, FPGAs are getting more and more attractive for CNN because of their increasing floating-point operation (FLOP) performance with hardened support in Intel's Generation 10 FPGA's, and secondly

their support for both sparse data and compact data types. These trends are favoring FPGA-based platforms since FPGAs are designed to handle irregular parallelism and fine-grained computations compared to GPUs and CPUs. The focus of this blog will be on FPGA as accelerated inference platform for CNNs.

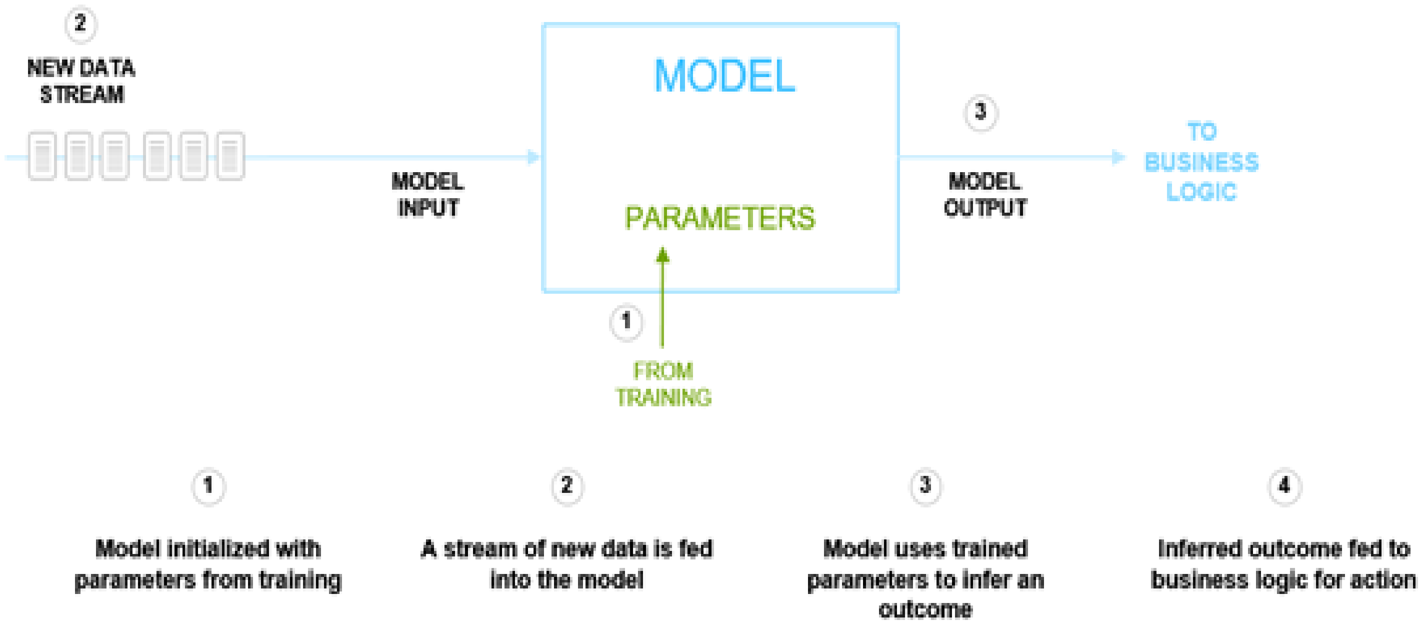

2
NEW DATA STREAM
MODEL INPUT
MODEL
PARAMETERS
1
FROM TRAINING
3
MODEL OUTPUT
TO BUSINESS LOGIC
1
Model initialized with parameters from training
2
A stream of new data is fed into the model
3
Model uses trained parameters to infer an outcome
4
Inferred outcome fed to business logic for action

Figure 2 Inference Flow.

2 Why iAbra?

2.1 Introduction to PathWorks

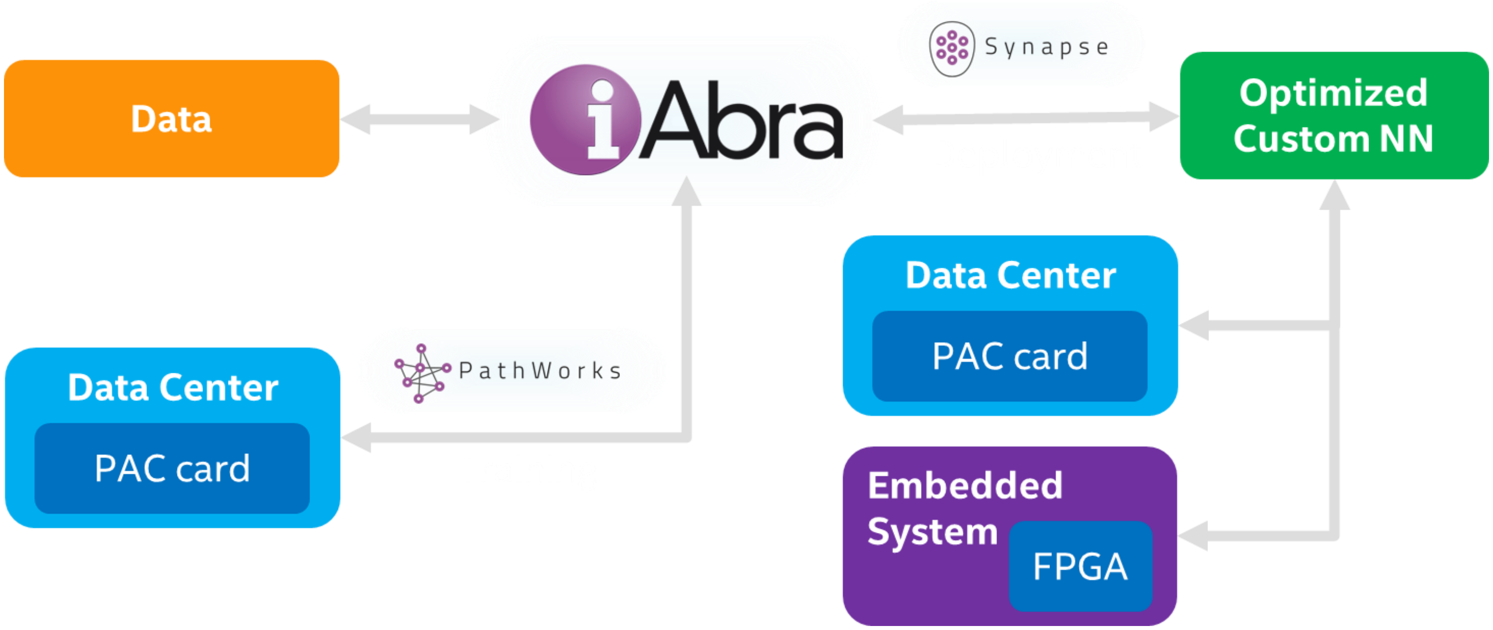


Figure 3 PathWorks Flow Diagram

The use of FPGA in both the data center and embedded / IoT applications for AI inference can deliver greater efficiency in terms of silicon size, weight and power (SWaP). However, the exploitation of the inherent benefits that the FPGA can deliver in terms of SWaP requires that the machine learnt models to be inferred can “fit” into the resources available on the target FPGA.

To enable this iAbra has developed a framework, PathWorks, that not only creates smaller yet equally capable neural networks that “fit” the FPGA, but also creates novel network architectures depending on the training data, further optimizing the model required.

This is possible due to the evolutionary architecture approach used but most importantly, since the reduced number of neurons have multiple (3D) connections to their neighbors, the search space for regression while larger, it remains possible to regress the model due to the scale out architecture inherent in PathWorks which is accelerated using FPGA for this model training.

The result is a framework that both obfuscates the complexity of machine learning for the data citizen (rather than relying on data scientists) while targeting the highly optimized FPGA silicon for inference.

2.2 iAbra AI Use Case Qualification

Embedded Use Case Characteristics

- Low Size, Weight and Power
- Type Approval or
- Plan of Record Compliant
- Low Failure Rates
- Environmental Survival
- security

Network Creation Use Case Characteristics

- Embedded End Use Case
- Abstracted Training Platform
- Non-Data Scientist Users
- Reduced time to Solution
- Smaller more targeted networks (e.g. sub 1000 neurons)
- Training to Inference Fidelity

2.2.1 How are the use cases addressed?

iAbra's tools provide Machine Learning (ML) optimized for FPGA Inference. While the resurgence in ML has been facilitated by the growth in computing power attributable to Moore's Law and specialized GPU hardware and associated software, the use of the subsequent models for processing data is another matter.

Artificial intelligence inference, or the processing of data with a machine learnt model might not be possible with real work constraints. In the military defense realm these can be numerous and include silicon size, weight and power (SWaP) along with security, environmental survival and integration with plan of record.

iAbra's focus on tools that create AI models for inference on FPGA seeks to ensure that real world constraints do not prevent the use of AI. Our perspective is that AI use cases are multiple not just outside of the data center, but at the edge, where bandwidth might be limited, or network access denied. When data connectivity constraints are coupled with low SWaP requirements, FPGAs are a rational choice.

Neural Network creation and training is often equal parts art and Science. iAbra's experience of the fragmented ecosystems relating to ML and AI while developing FPGA based models for inference, motivated the creation of our end to end tool chain which automates the network creation and training. This enables a user to create a neural network, compiled to an FPGA for AI inference, using an abstracted interface. This enables an analyst, without specific ML skills, to create a model that can not only amplify and augment the process of extracting utility from data in support of defined outcomes but be used on FPGA silicon for down range implementation not practicable with some other silicon.

2.2.2 Implementation

- iAbra's tool flow takes raw training data and creates a compact application specific NN

- The NN is then optimised for power and performance using the training data– this can be done in the datacentre on the Intel ARRIA10 PAC card or an embedded system
- As the next is compact training times are relatively short and the end result is a highly optimised NN that can be tested and run in the datacentre or an embedded system

Short development time is a key benefit but also the fact that the end result is easier to implement and more efficient in an embedded system than when using other development flows.

iAbra's Tools provide an Ecosystem to Target FPGA Silicon for embedded AI Inference.

Targeting FPGA for Neural Network Inference, or embedded AI, requires optimisation at the neural network training stage and an inference architecture that exploits the inherent advantages available to the FPGA silicon. These advantages are most valuable in embedded solutions where power, heat, silicon footprint and environmental harshness are constraint factors.

FPGAs provide the ideal AI inference platform where low size, weight and power are design criteria. The use of FPGAs in harsh environments where security and ultra-low failure rates are required is well proven, while the range of FPGA products allows the optimal logic capacity to be selected for an application.

Training a neural network using GPUs and inefficient architectures tends to result in AI models that are too big and complex for embedded systems where power and silicon resources are constrained. By optimising the neural network at the training stage, iAbra's Neural PathWorks ensures the network is of minimal size for the problem domain provided in the training set, requiring fewer computational operations for inference. This optimisation effort ensures the network can be stored in the FPGA's memory and logic elements.

2.2.3 How does iAbra use FPGA?

iAbra's Neural Synapse exploits the flexibility of the FPGA hardware allowing image data requiring analysis by the AI inference model to be pipelined from input to output without the need for external memory resources which increase on-chip and system power consumption. With the performance of the FPGA's dedicated computing architecture we can reach an optimal solution for various applications.

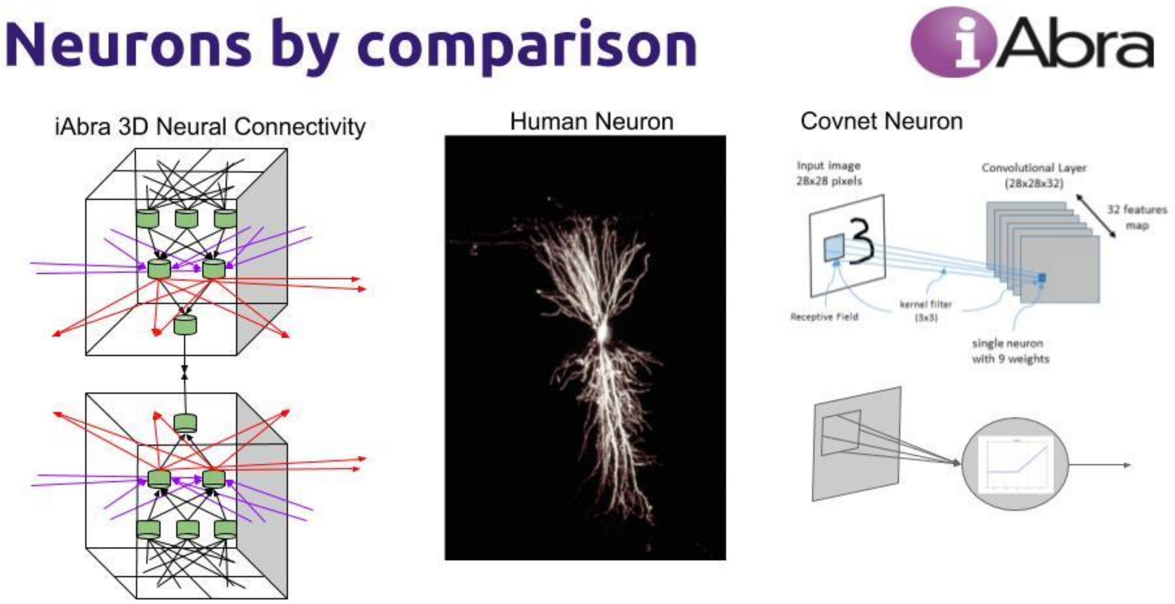


Figure 4 iAbra Neuron implementation

2.3 Application areas for PathWorks

PathWorks provides the ability to structure unstructured high dimensional data in a low SWaP (size, weight and power) footprint for custom problem domains. Generically this makes it ideal for solutions with the following demands:

- Recognition problem domains which don’t fit ImageNet mold
- Machine augmented annotation of data known as auto labeling
- Large volumes of archive or streaming imaging data
- Sensor fusion applications with multiple channels of imaging
- Edge IOT image analysis
- Low SWaP for dense data centers or edge systems
- Domain experts who wish to create their own classifiers or many classifiers need to be created during operational lifecycle

2.3.1 Market segments for PathWorks

- Defense / Intelligence analytics data center or edge
- Wide area Geographic Information System (GIS) analytics for optical and radar data
- Smart agriculture
- Autonomous driving and advanced driver awareness ADAS++

3 Why the PowerEdge C6420 server?

3.1 Overview C6420

The PowerEdge C6420 optimizes compute, processors, memory and large volume local storage for an IT services platform that can be efficiently and predictably scaled, while drastically reducing complexity. With up to 4 independent hot-swappable 2-socket servers in a very dense 2U chassis, servers can be easily repurposed as workloads change. Numerous options for compute, storage, connectivity and chassis offerings provide flexibility to configure servers for your specific workloads.

- High Performance Computing (HPC)
- High Performance Data Analytics (HPDA)
- Web Scale Applications / Software as a Service (SaaS) / Infrastructure as a Service (IaaS)
- Financial Modeling and High Frequency Trading (HFT)
- Render nodes for Visual Effects Rendering (VFX)
- Private Cloud Infrastructure
- Hyper-converged Infrastructure (HCI)

The PowerEdge C6420, with its flexible configurations, hyper-scale capabilities and overall efficiency, is ideal for modern hyper-converged infrastructures including validated, pre-bundled Dell EMC HPC solutions, VxRail and VxRack, the Dell EMC XC Series.

Features latest generation Intel® Xeon® SP family processors and up to 56 cores per node

Supports up to 512GB memory per node Offers flexible I/O options including, low-latency InfiniBand™ and next-generation Intel Omni-Path and provides new Direct Liquid Cooling options for improved power efficiency.

4 Why FPGA?

FPGA (Field Programmable Gate Arrays) allow a blank slate to build the solution that fits the problem best instead of fitting a solution into a predefined architecture. FPGAs provide flexibility for AI system architects searching for competitive deep learning accelerators that also support differentiating customization. The ability to tune the underlying hardware architecture, including variable data precision, and software-defined processing allows FPGA-based platforms to deploy state-of-the-art deep learning innovations as they emerge. Other customizations include co-processing of custom user functions adjacent to the software-defined deep neural network. Underlying applications are in-line image & data processing, front-end signal processing, network ingest, and IO aggregation. This flexibility allows systems deployed with FPGA to adapt to new advancements in Deep Learning or other value add tasks without redeployment.

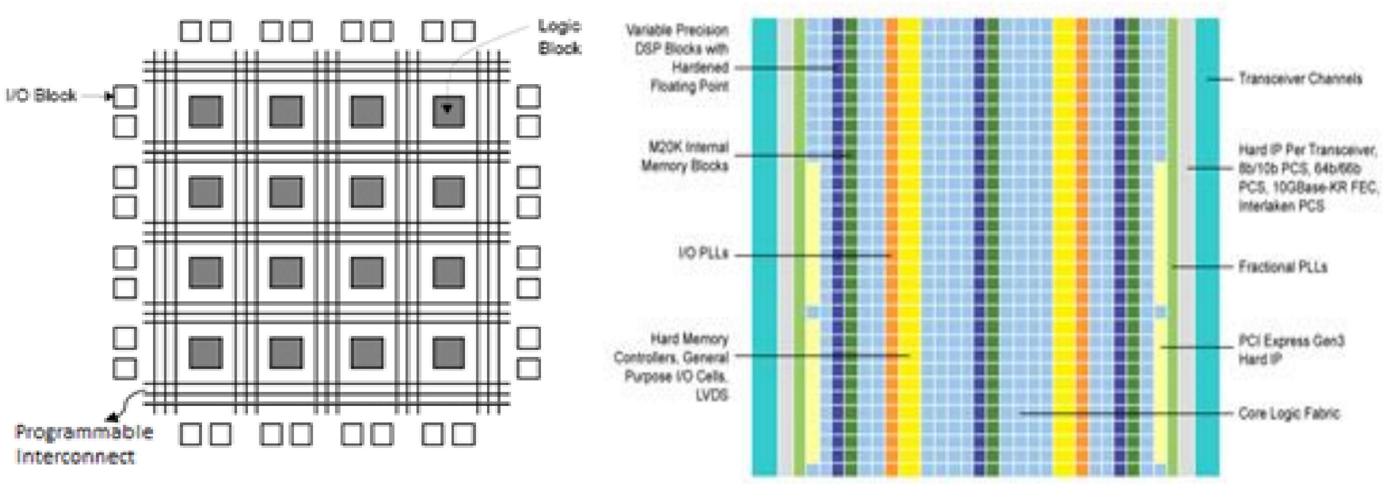

Logic Block
I/O Block
Programmable Interconnect
Variable Precision DSP Blocks with Hardened Floating Point
M20K Internal Memory Blocks
I/O PLLs
Hard Memory Controllers, General Purpose I/O Cells, LVDS
Transceiver Channels
Hard IP Per Transceiver, 8b/10b PCS, 64b/66b PCS, 10GBase-KR FEC, Interlaken PCS
Fractional PLLs
PCI Express Gen3 Hard IP
Core Logic Fabric

Figure 5a (Left) Arrayed building blocks are connected via interconnect wires; (Right) Fully featured FPGAs include a variety of advanced building blocks

Figure 5b (Right) illustrates the variety of building blocks available in an FPGA. The core fabric implements digital logic with Look-up tables (LUTs), Flip-Flops (FFs), Wires, and I/O pads. FPGAs today also include Multiply-accumulate (MAC) blocks for DSP functions, Off-chip memory controllers, High-speed serial transceivers, embedded, distributed memories, Phase-locked loops (PLLs), hardened PCIe interfaces, and range from 1,000 to over 2,000,0000 logic elements.

4.1 FPGAs in Mission-Critical Applications

Mission-critical applications (e.g., autonomous vehicle, defense and intelligence, manufacturing, smart agriculture, smart cities, etc.) require deterministic low-latency. The data flow pattern in such applications may be in streaming form, requiring pipelined-oriented processing. FPGAs are excellent for these kinds of use cases given their support for fine-grained, bit level operations in comparison to CPU and GPUs. FPGAs also provide customizable I/O, allowing their integration with these sorts of applications.

In autonomous driving or factory automation where response time can be critical, one benefit of FPGAs is that they allow tailored logic for dedicated functions. This means that the FPGA logic becomes custom circuitry but highly reconfigurable, yielding very low compute time and latency. Another key factor may be power – the cost per performance per watt may be of concern when determining long-term viability. Since the logic in FPGA has been tailored for a specific application/workload, the logic is very efficient at executing that application which leads to lower power or increased perf per watt. By comparison, CPUs may need to execute 1000’s of instructions to perform the same function that an FPGA maybe able to implement in just a few cycles.

Figure 4. Examples of mission-critical applications require deterministic, fast response.

4.2 Intel Programmable Acceleration Card

The Intel Programmable Accelerator Card (PAC) features an Intel Arria® 10 FPGA, an industry-leading programmable logic built on 20 nm process technology, integrating a rich feature set of embedded peripherals, embedded high-speed transceivers, hard memory controllers and IP protocol controllers. Variable-precision digital signal processing (DSP) blocks integrated with hardened floating point (IEEE 754-compliant) enable Intel Arria® 10 FPGAs to deliver floating point performance of up to 1.5 TFLOPS. Arria® 10 FPGAs have a comprehensive set of power-saving features. Combined, these features allow developers to build a versatile set of acceleration solutions.

Figure 5. Intel Programmable Acceleration card

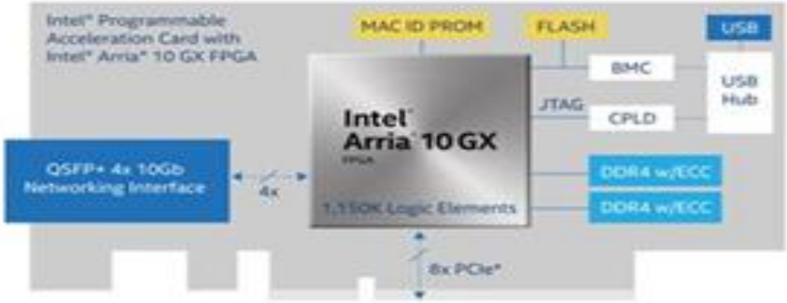
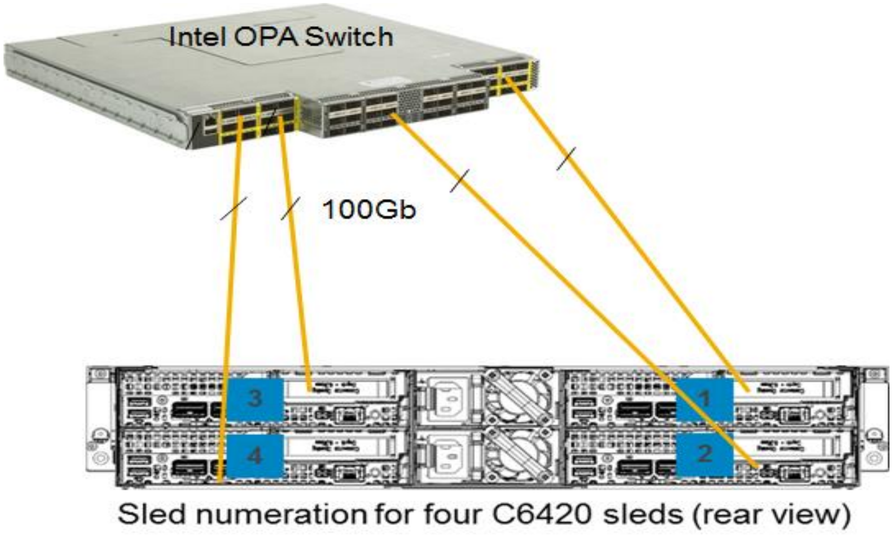


Figure 6. Intel PAC Block Diagram

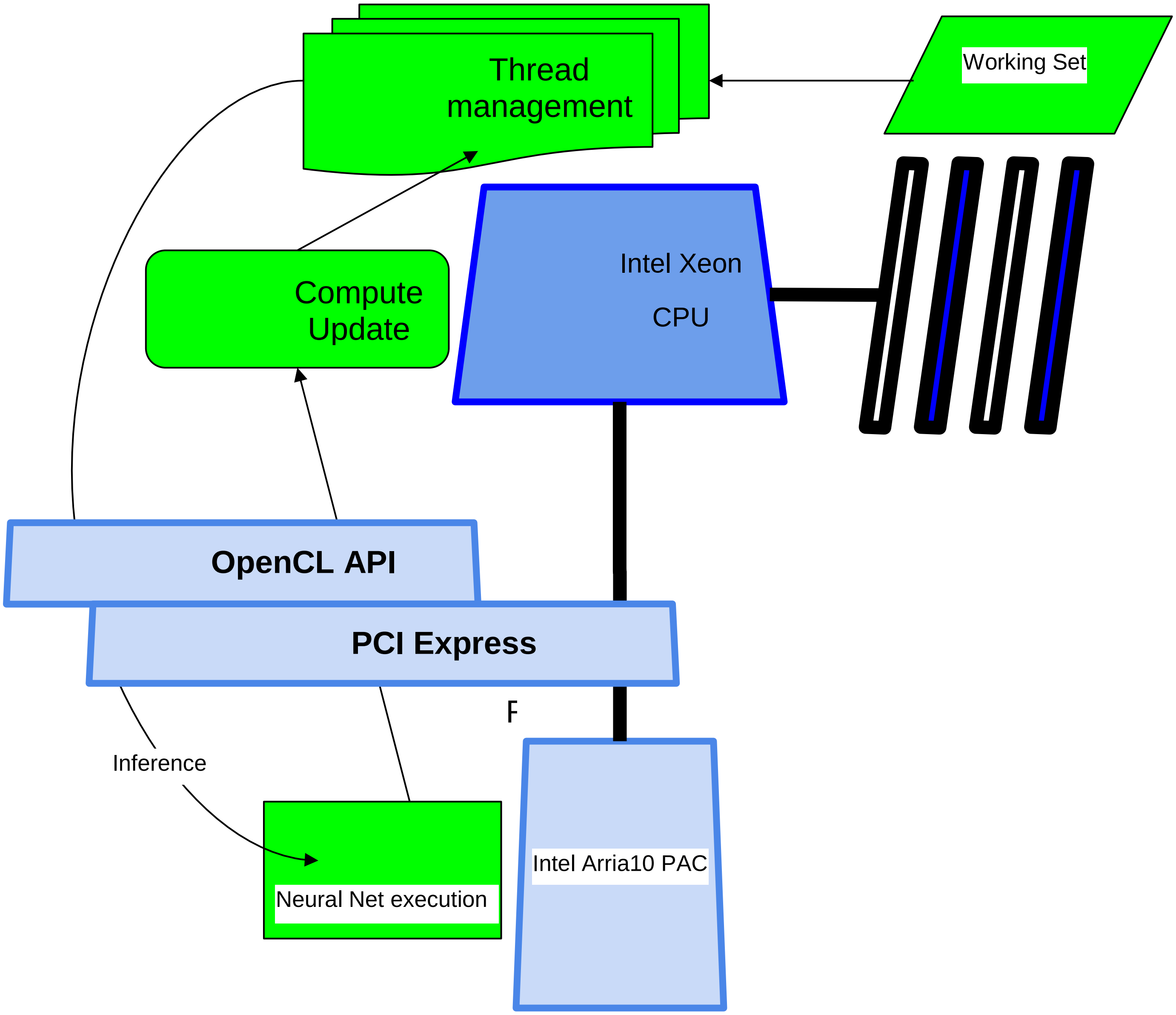
5 Reference Setup of PowerEdge C6420



Features	C6420 Chassis	Sled 1- 4
PSU	2x1600W PSU	
CPU		Skylake 6148 20C 2.4GHz
Memory		256GB Memory (16G DIMM)
Networking		OPA 100Gbps
Accelerator		Intel PAC Aria10 FPGA adapter
OS		Rhel 7.4
Storage		2x1.6TB SSD SAS Mix use, PM1635a,3 DWPD,8760 TBW - Mirror the disk for redundancy
TOR		Intel OPA

6 High level data flow for performance testing

6.1 Performance metrics

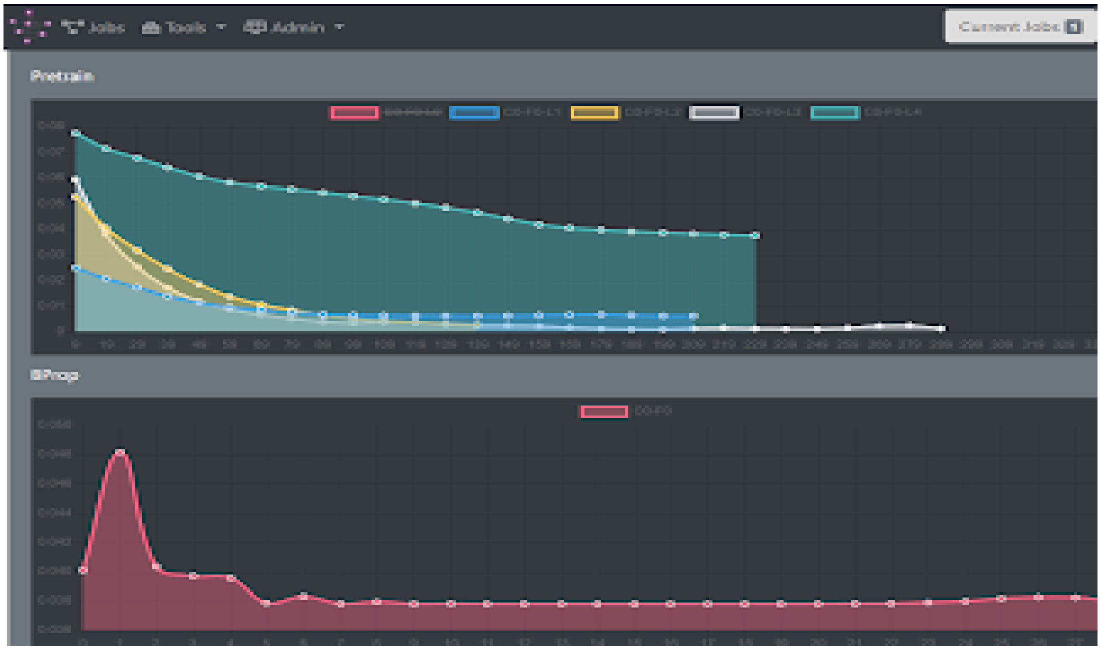


6.1.1 CPU & FPGA utilization (Average during training time)

The iAbra application has been tuned to balance the load across CPU and FPGA resulting in utilizations of 82% on the CPU(40 cores) and 94% on the FPGA.

6.1.2 Top 1% accuracy & Average individual run time

The iAbra application developed a network and weights resulting in 96.2% accuracy in 42mins.



The figure above shows the network convergence over time.

Y axis is Root squared mean error and Y axis number of iterations, as can be sheen iAbra PathWorks learning algorithm converges very rapidly over a small number of iterations enabling practical evolutionary training

6.1.3 Throughput images/sec Vs batch size

2280 batch size 2000

6.1.4 Inference Images within 7ms window

16 images (0.795 megapixels) accelerator using 60 mW

6.1.5 Megapixels per watt

This resulted in a power efficiency of 14.25 megapixels per watt which scales linearly with image size, another advantage of the iAbra neural network architecture.