ARTIFICIAL INTELLIGENCE 501

Lesson 7
Hardware

LEGAL NOTICES & DISCLAIMERS

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer. No computer system can be absolutely secure.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit http://www.intel.com/performance.

Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

Statements in this document that refer to Intel's plans and expectations for the quarter, the year, and the future, are forward-looking statements that involve a number of risks and uncertainties. A detailed discussion of the factors that could affect Intel's results and plans is included in Intel's SEC filings, including the annual report on Form 10-K.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Performance estimates were obtained prior to implementation of recent software patches and firmware updates intended to address exploits referred to as "Spectre" and "Meltdown." Implementation of these updates may make these results inapplicable to your device or system.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

Intel, the Intel logo, Pentium, Celeron, Atom, Core, Xeon and others are trademarks of Intel Corporation in the U.S. and/or other countries.
**Other names and brands may be claimed as the property of others.*

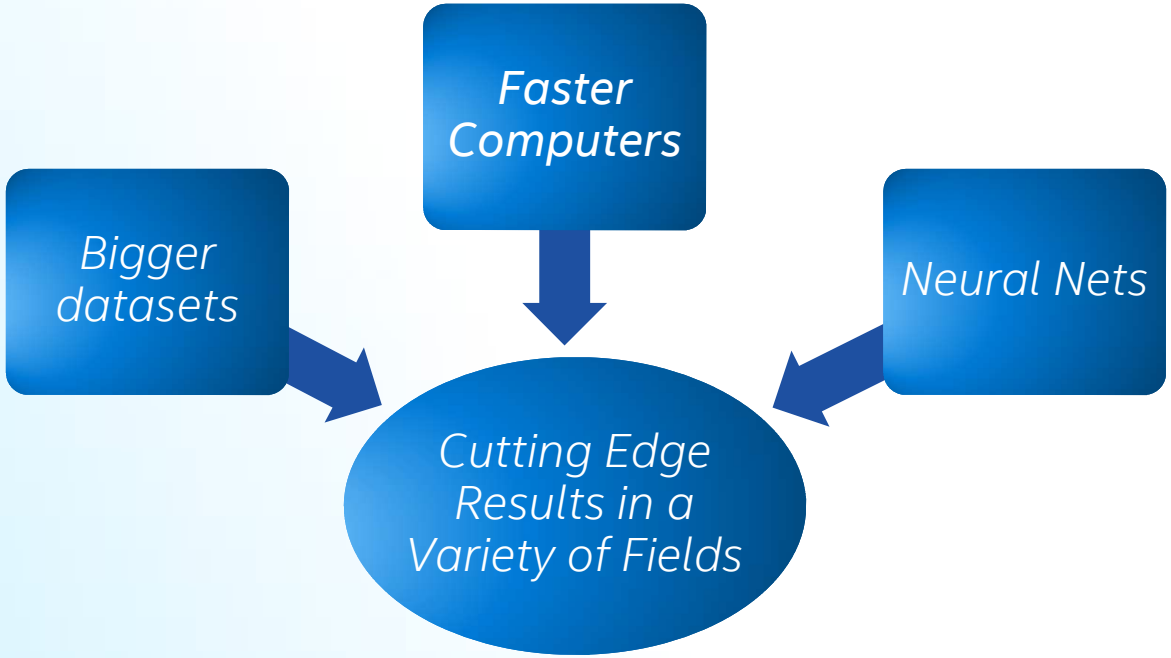
Learning Objectives

You will be able to:

- Define “datacenter”, “gateway”, and ”edge computing”
- Recall the key processor types for AI
- Explain how Intel® hardware applies to AI
- Compare Central (CPU) and Graphic (GPU) processing units
- Describe Field Programmable Gate Arrays (FPGA)

AI in the Modern Era

Faster hardware is one of the key areas driving the modern era of AI.



Intel® AI Portfolio

Experiences

Intel® AI Portfolio

Platforms

Intel® Deep Learning Studio

Intel® Saffron™

Intel® Computer Vision SDK

Intel® Movidius™ MDK

Intel® Deep Learning Cloud & Appliance

Intel® AI Portfolio

Frameworks

Intel® AI Portfolio

Libraries

Intel® Data Analytics Acceleration Library (Intel® DAAL)

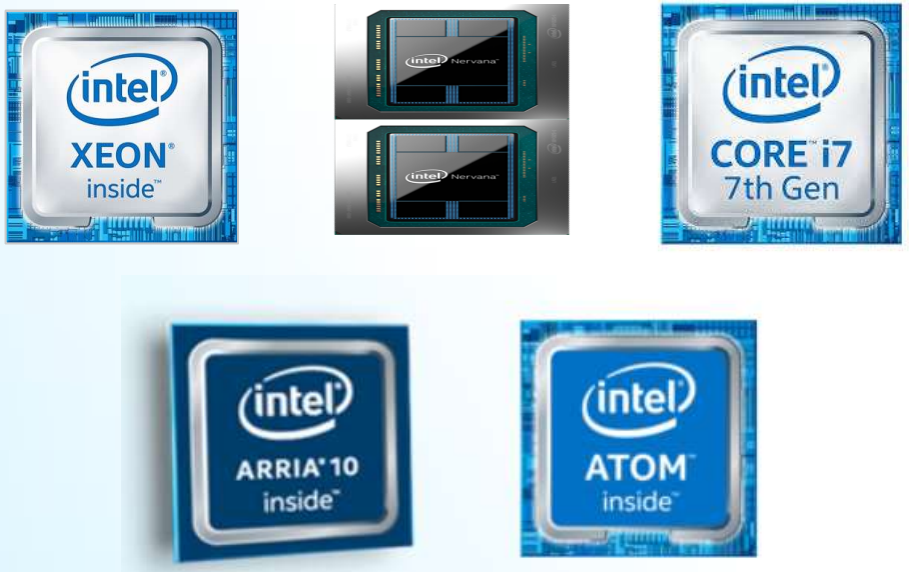
Intel® Distribution for Python (Performance Optimized)*

Intel® Math Kernel Library (Intel® MKL, MKL-DNN)

Intel® nGraph™ Library

Intel® AI Portfolio

Hardware




intel
END-TO-END AI COMPUTING

End-to-End Computing for AI

DATACENTER

Many-to-many hyperscale,
for stream and massive batch data processing

Ethernet
& Wireless

GATEWAY

1-to-many,
majority streaming data from devices

Wireless & non-IP wired,
secure, high throughput, real-time

EDGE

1-to-1 devices,
lower power and UX requirements

End-to-End Computing for AI

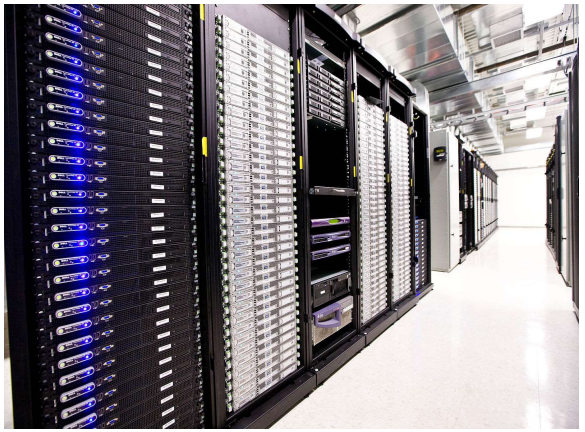
There are various hardware requirements for different AI tasks.

- AI tasks can include data collection and fusion, training, and inference.
- The number of operations for training can be on the order of exaFLOPS, making this task more suited for datacenters.
- Inference takes fewer operations than training and can be done on both edge devices and at datacenters.
- Certain applications can have constraints on the processor size and power. For example, wearable devices and drones.

Datacenter

Datacenters are used to store and process data for applications, from websites to Internet of Things (IoT) systems.

- Low latency, high bandwidth – needed to access high compute resources.
- Reliability, lack of downtime, high performance.
- AI is the fastest-growing datacenter workload.
- A ~12X growth in demand by 2020 is expected.
- Training is typically done in a datacenter, allowing high processor power and physical size.

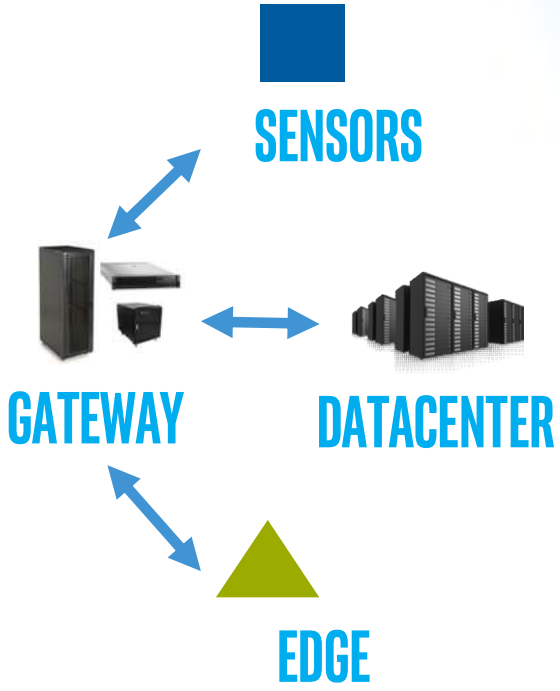


Datacenter

Gateway

Gateway computers route information from edge devices into and out of datacenters.

- High bandwidth – able to handle input from many devices and route correctly.
- Lightweight protocols – can't require extensive CPU usage, keeping gateway computer fast.
- Secure protocols – gateway computers are often a security point of failure.



Edge

Edge computing refers to computing happening as close as possible to where the computation is required.

- Sensors read information and compute a result directly on an Internet of Things (IoT) device, without sending information to data center.
- Reduces communication between sensors and datacenter.
- Uses resources that are not continuously connected to a network.
- Edge computing often used for inference rather than model training.

End-to-End Example: Automated Driving

Autonomous vehicles produce ~4 terabytes of data per day.

One car, driving for one hour, requires ~5 exaFLOPS of computational power in order to safely keep it on the road.

DARPA Self-Driving Car Challenge Won in 2005

End-to-End Example: Automated Driving

Vehicle requires:

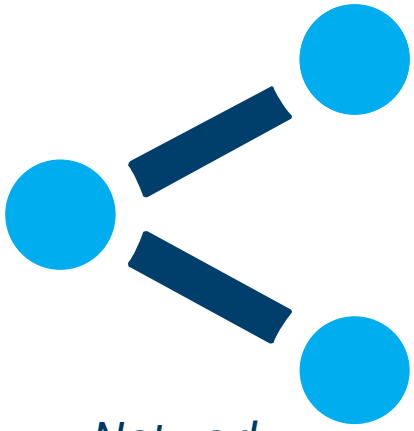
- Human-Machine Interface (HMI) – to build trust between driver and vehicle and provide advanced virtualization and graphics capabilities
- Scalable, powerful in-vehicle computing
- Sensor processing
- Environment modeling
- Driving functions

Car

End-to-End Example: Automated Driving

Network requires:

- Vehicle-to-vehicle (V2V) and Vehicle-to-everything (V2X) communication
- Next-generation network connectivity
- Over-the-air-updates
- High-definition maps

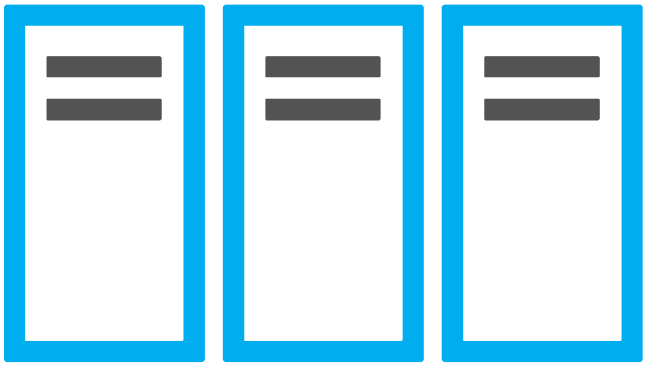


Network

End-to-End Example: Automated Driving

Datacenter requires:

- High-performance data center and cloud
- AI to fleet management to data mining
- Model training
- Greater than real time inference



Datacenter


intel
DATACENTER AI

Datacenter AI

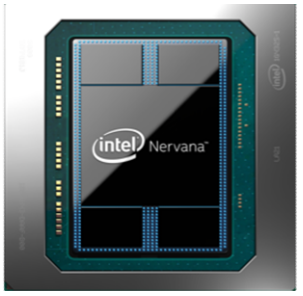
Intel® offers a range of processors to compute in the datacenter.

Intel® hardware at the datacenter:

- Field-programmable gate array (FPGA): spectrum of datacenter to edge.
- Intel® Xeon® processors: widest variety of AI and other data center workloads
- Intel® Nervana™ Neural Network Processor: custom built processor

Intel® Stratix® 10 FPGA

Intel® Xeon® Processors



Intel® Nervana™

Intel® Xeon® Scalable Processor Family

Intel® Xeon® Scalable Processor Family

Scalable performance for widest variety of AI, datacenter workloads, and DL.

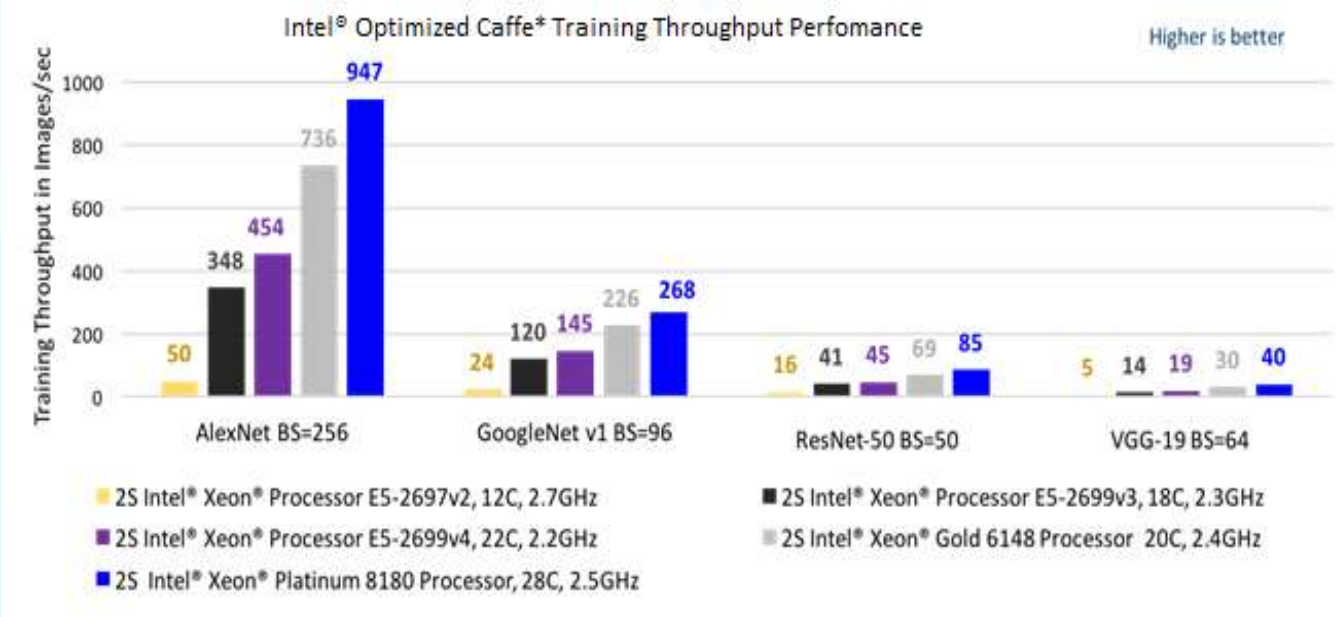
- Can use existing general-purpose Intel® Xeon® processor clusters for DL training and inference, classical ML, and big data workloads.
- Popular AI frameworks such as TensorFlow* and MXNet* have been optimized for Intel architecture
- BigDL for scale deep learning training on Spark Hadoop*.
- Production-ready robust support for full range of AI deployments.

Intel® Xeon® Scalable Processor Family

Intel® Xeon® processors power the vast majority of general purpose ML and inference workloads for businesses today.

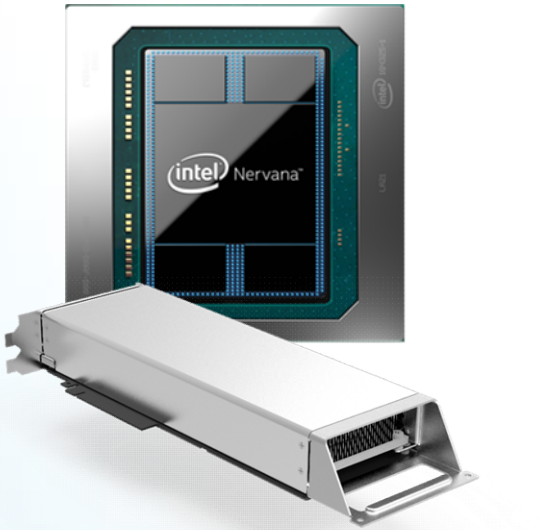
- Reduced time-to-train by using more server nodes scaling near linearly to hundreds of nodes.
- On November 7, 2017 researchers published record training time results for both ResNet-50 and AlexNet* using Intel Xeon scalable processors and achieved state-of-the-art accuracy.
- When combined with Intel® Math Kernel Library (Intel® MKL), 100x speedup in Deep Neural Net processing.

Intel® Xeon® Scalable Processor Family



Training throughput of Intel® Optimized Caffe across Intel® Xeon® processor v2, Intel Xeon processor v3, Intel Xeon processor v4, Intel Xeon Gold processor, and Intel Xeon Platinum processors with AlexNet* using batch size (BS) equal to 256, GoogleNet* v1 BS=96, ResNet-50 BS=50, and VGG-19 BS=64. Intel® MKL-DNN provides significant performance gains starting with the Intel Xeon processors v3 when AVX-2 instructions are introduction and another significant jump with the Intel Xeon Scalable processors when AVX-512 instructions are introduced.*

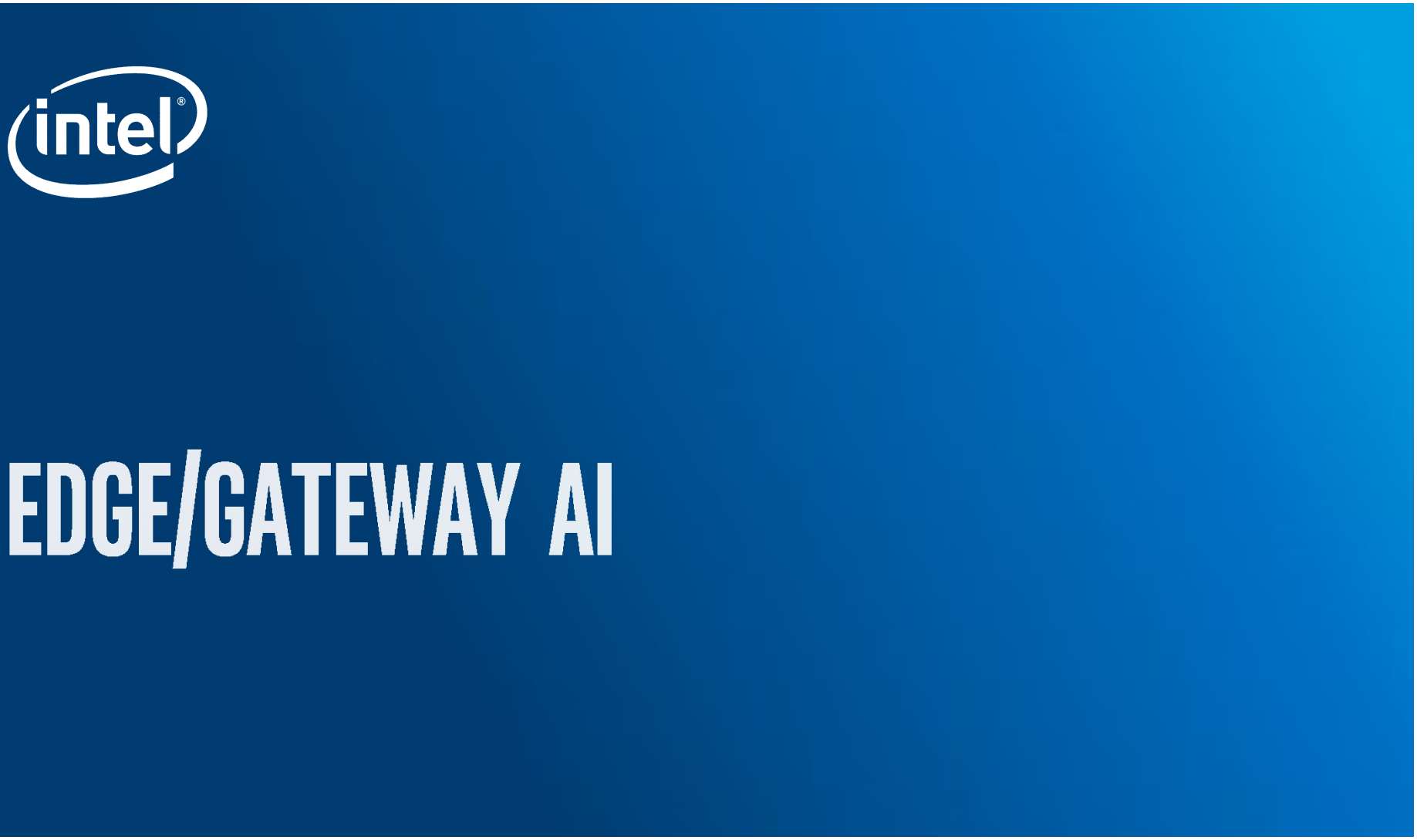
Intel® Nervana™ Neural Network Processor (NNP)



Intel® Nervana™ Neural Network Processor (NNP)

The Intel® Nervana™ neural network processor (NNP) is a custom-designed chip for modern AI.

- New memory management system and chip architecture optimized for DL utilization.
- New data format “Flexpoint” for DL.
- Hardware increases potential for parallel computation.
- Intel® in close collaboration with Facebook for development.
- Provides the needed flexibility to support DL primitives while making core hardware components as efficient as possible.


intel
EDGE/GATEWAY AI

Edge-Specific Processors

Intel® offers a variety of edge computing-focused processors.

GENERAL PURPOSE PROCESSORS

SPECIFIC USE PROCESSORS

Intel Atom® and Intel® Core™

Intel® processor graphics

Intel® Movidius™ Myriad™ VPU

Intel® Atom® and Core™ Processors

Intel® Atom® and Core™ Processors

Intel® Atom® and Core™ processors are low power general purpose processors for embedded applications.

- Lower power than Intel® Xeon® processors.
- Can handle a variety of workloads including basic DL inference and classical ML within a larger application.
- Widely used in Mobile Internet Devices (MIDs): feature-limited tablets.
- Power edge computing in automobiles and other Internet of Things (IoT) applications.

Intel® Atom® and Core™ Processors

Amazon's DeepLens* video camera uses the Intel® Atom® processor to run deep learning models locally for object detection and recognition. This can be used for face detection, activity recognition, and artistic style transfer.

Intel® Iris Pro™ Graphics

Intel® Iris Pro™ Graphics

The Intel® Iris Pro™ graphics chips are integrated graphics processors that sit on the same die as the CPU.

- Widely used in laptops that do not have dedicated GPUs (e.g. most Apple laptops).
- Architecture makes them well-suited for higher inference throughput, that is their most common use in AI.
- Provide performance for media or graphics intensive workloads, that often are required for AI.

Intel® Movidius™ Myriad™ X Vision Processing Unit (VPU)

Intel® Movidius™ Myriad™ X Vision Processing Unit (VPU)

Movidius™, acquired by Intel in 2016, offers the Myriad™ X VPU.

- Features the Neural Compute Engine – a dedicated hardware accelerator for deep neural network inference.
- Form factor makes it easy to deploy on small computing devices, such as Raspberry Pi* and use on Internet of Things (IoT) applications.
- Comes with Intel's Movidius™ Myriad™ development kit (MDK).

Intel® Movidius™ Vision Processing Unit (VPU)

The Myriad™ X VPU powers the Intel® Movidius™ Neural Compute Stick (NCS).

- Enables rapid prototyping, validation and deployment of Deep Neural Network inference at the edge.
- Allows for AI applications that aren't reliant on a connection to the cloud.
- The Intel® Movidius™ Neural Compute SDK allows developers to deploy real-time CNNs for inferencing on low-power applications.

HARDWARE FOR AI: FPGAS

Stratix® 10 FPGA

Intel® Stratix® 10 FPGA

FPGA: Overview

Field Programmable Gate Arrays (FPGAs) are hardware that can be reconfigured – programmed in the field.

- FPGAs can become any digital circuit as long as the unit has enough logic blocks to implement that circuit.
- Enables the creation of custom hardware for individual solutions in an optimal way that other devices cannot efficiently support.
- High-throughput, low-latency processing of complex algorithms, such as neural networks.
- Flexible fabric enables direct connection of various inputs, such as cameras, without needing an intermediary.

FPGA: Power Efficiency

FPGAs have exceptional power efficiency.

- Applications like hyper-scale datacenters, where total cost of ownership is a key factor.
- Intel® FPGAs have over 8 TB/s of on-die memory bandwidth.
- Solutions tend to keep the data on the device tightly coupled with the next compute.
- Minimizes the need to access external memory, that results in running at significantly lower frequencies.
- With a custom setup and when programmed for your application FPGAs show up to 80% power reduction when using AlexNet* (a convolutional neural network)

FPGA: Future-Proofing

FPGAs flexible fabric and re-configurability enables future-proofing hardware.

- Users can update system hardware capabilities without requiring a hardware refresh, resulting in longer lifespans of deployed products.
- Run current and future neural network topologies on the same hardware.
- FPGAs are flexible to any precision type – optimize accuracy, performance, and speed to the exact need.
- As precisions drop from 32-bit to 8-bit or even ternary/binary networks, an FPGA has the flexibility to support those changes instantly.
- Have been used in space, military, and extremely high reliability environments for decades.

Intel® FPGAs

Intel® has shortened the design time for developers by creating a set of API layers. Developers can interface with the API layers based on level of expertise.

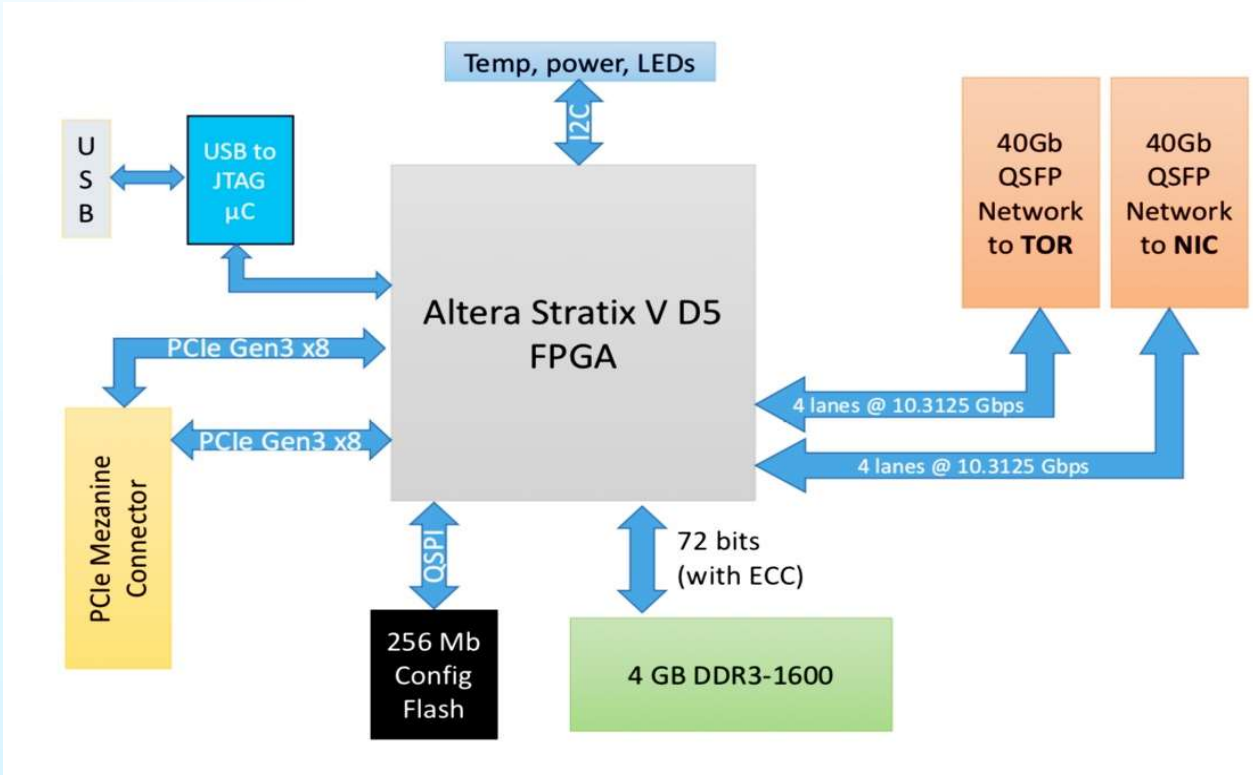
- Typical users can start at the SDK or framework level.
- Users who want to build their own software stack can enter at the Software API layer.
- Users who want to build their own software stack can enter at the C++ embedded API layer.
- Advanced platform developers who want to add more than machine learning to their FPGA can enter in at the OpenCL™ host runtime API level.

FPGAs for AI: Microsoft

Project Catapult uses clusters of FPGA chips attached to Intel® Xeon® processors

- Microsoft has been deploying FPGAs in every Azure* server over the last several years.
- This configurable cloud provides more efficient execution that CPUs for many scenarios without the inflexibility of fixed-function ASICs at scale.
- Microsoft is already using FPGAs for Bing* search ranking, deep neural networks (DNN) evaluation, and software defined networking (SDN) acceleration.

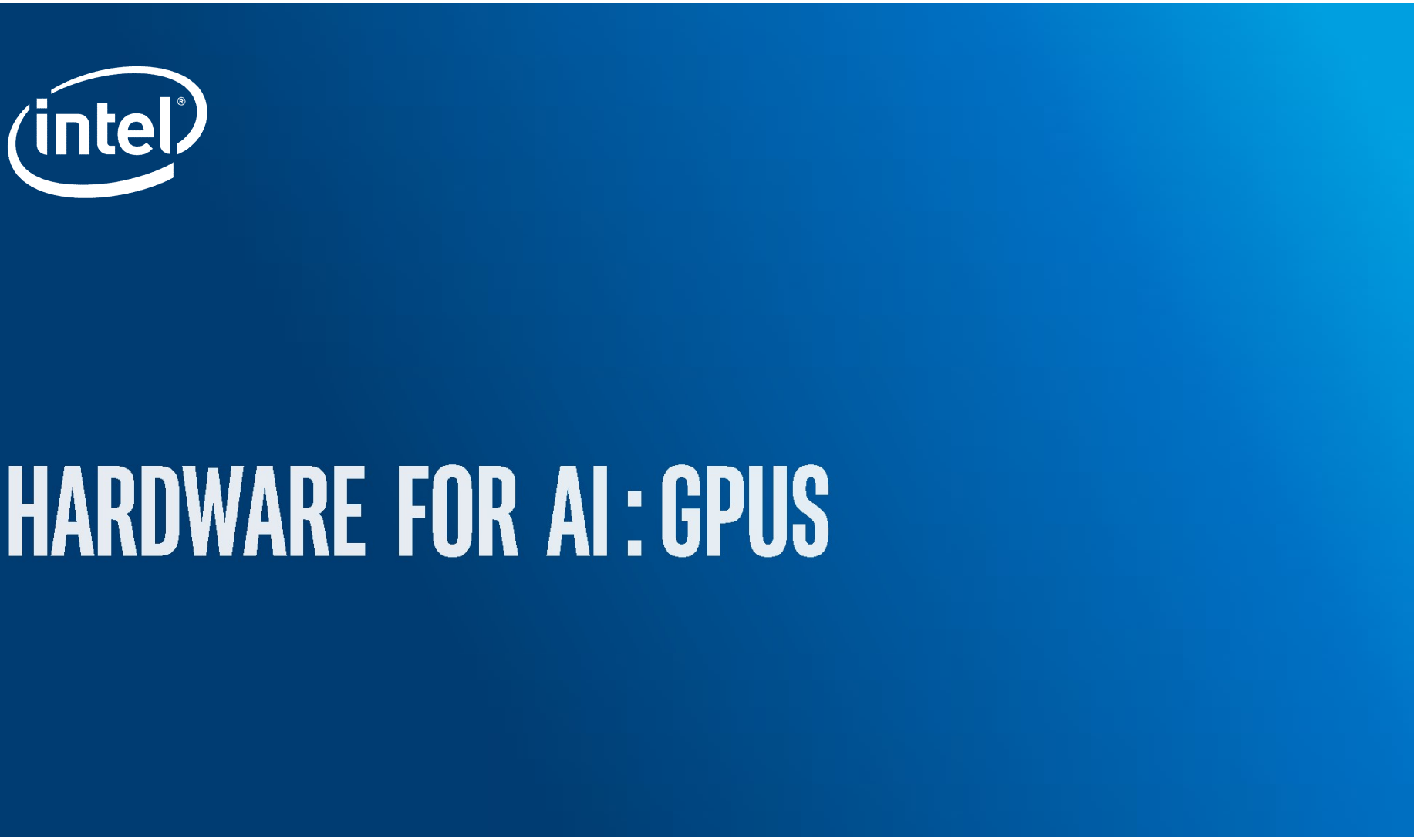
Microsoft Configurable Cloud



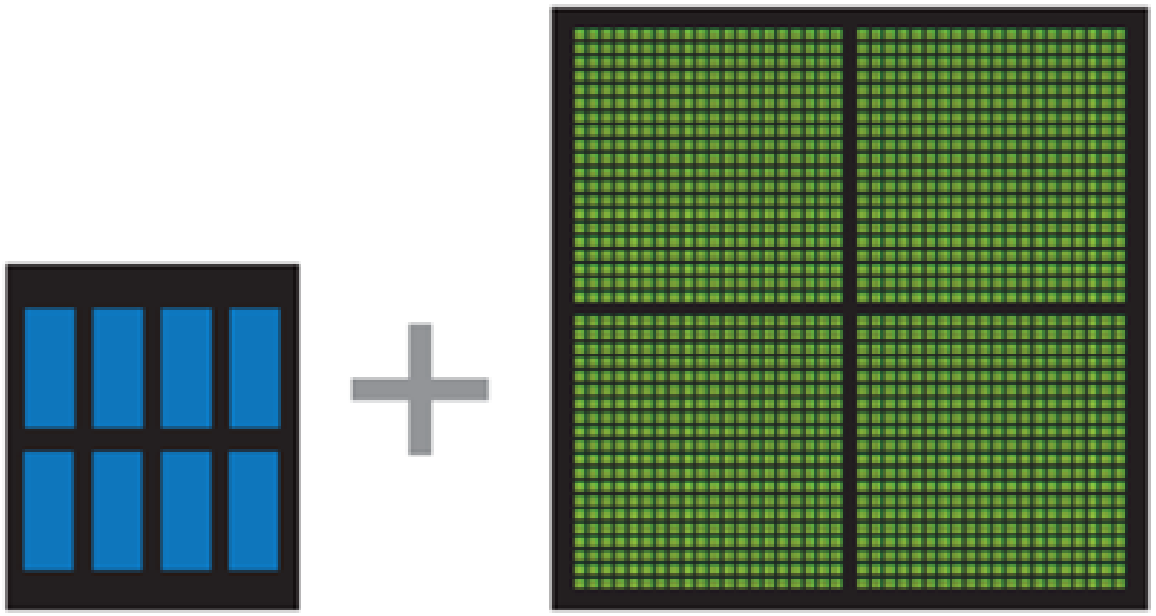
FPGAs for AI: Microsoft

Project Brainwave: a DL acceleration platform.

- Announced at the 2017 Hot Chips Conference.
- A DNN hardware engine synthesized onto FPGAs.
- High performance, distributed system architecture
- Compiler and runtime for low-friction deployment of trained models
- Leverages Microsoft's vast FPGA infrastructure
- By attaching FPGAs directly to the datacenter network DNNs can be served as hardware microservices.


intel
HARDWARE FOR AI: GPUS

Hardware for AI: GPUs



CPU

DOZENS OF CORES

GPU

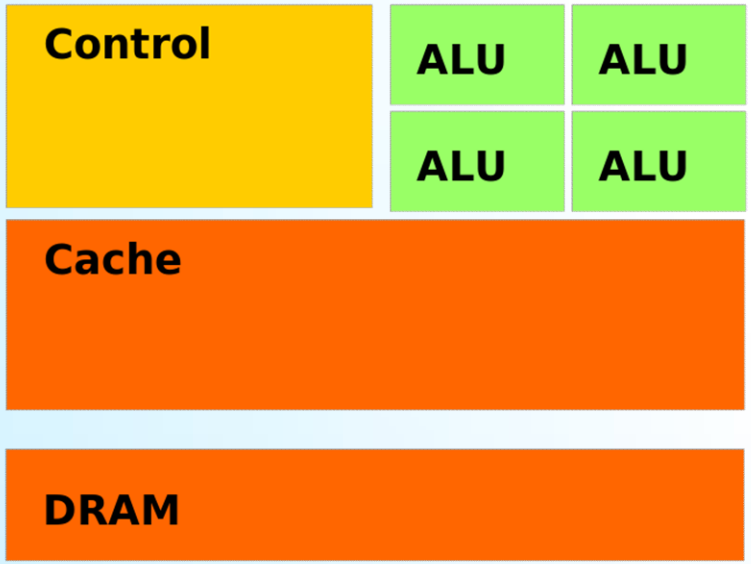
THOUSANDS OF CORES

Hardware for AI: GPUs

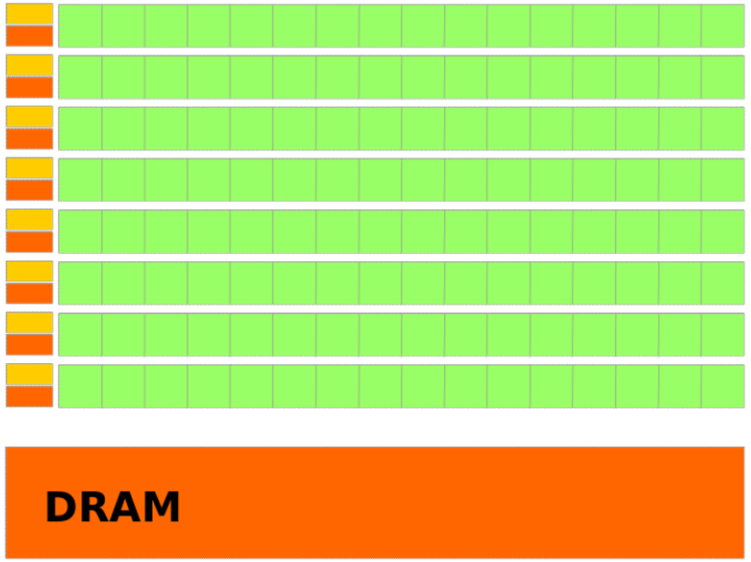
Originally designed to process graphics, GPUs have become a popular DL workhorse.

- Feature thousands of small, simple cores specialized for numeric, parallel computations.
- Many transistors dedicated to computation.
- GPUs excel at repeated similar instructions in parallel.
- Optimized for parallel data throughput computation.
- Major neural net breakthroughs since 2012 have been powered by GPU computations – performance has since increased >5x.
- Once the data is in the GPU memory the bottlenecks are small.
- NVIDIA is highly popular in discrete GPUs

CPUs vs GPUs



CPU



GPU

CPUs vs GPUs

While GPUs are specialized processors, CPUs are still the main computation engines on most computers.

- CPUs have dozens of cores compared to GPUs that have thousands of less powerful cores.
- CPUs have fewer arithmetic logic units (ALUs) and a lower compute density than GPUs
- CPUs are lower latency and have larger cache memory compared to GPUs

CPUs vs GPUs

While GPUs are specialized processors, CPUs are still the main computation engines on most computers.

- GPUs are designed for parallel tasks and perform well when a single instruction is to be performed over a large amount of data.
- GPUs have additional overhead when copying data from main memory is required so CPUs can be better when a large number of memory swaps are needed.
- GPUs are not good for tasks that cannot be parallelized or when heavy processing on fewer data streams is needed. CPUs excel at serial tasks.
- CPUs are easy to program – popular programming languages compile to machine code, to be run on CPU by default.

NVIDIA GPUs

NVIDIA‘s main business is in dedicated hardware for discrete graphics processors.

Several factors have made NVIDIA highly popular in this space.

- CUDA* framework which lets developers write code in C++, Python*, and other popular languages that run optimized operations on the GPU.
- Early incorporation of GPUs with cloud services, such as Amazon Web Services*.
- For several years, NVIDIA facilitated ecosystem adoption by optimizing their libraries and abstracting the complexities of the GPU through SDKs, tools, and libraries.


intel
CUSTOM HARDWARE FOR AI

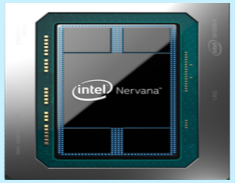
Hardware for AI: General Purpose vs. Specialized

AI runs on both general purpose processors and on specialized hardware.

SPECIALIZED HARDWARE

GOOGLE TPU*

APPLE NEURAL ENGINE*



INTEL® NERVANA™ NNP

INTEL® MOVIDIUS™

*Other names and brands may be claimed as the property of others.

Google Tensor Processing Unit* (TPU)

Google's TPU* is the first major example of a chip designed for AI.

- Google's first-gen TPU (2015) was designed specifically for fast inference.
- Unlike a GPU, no capabilities for graphics tasks, such as rasterization or texture mapping.
- Second-gen (2017) increased the precision available for computations, making this version usable for training as well.

Apple Neural Engine*

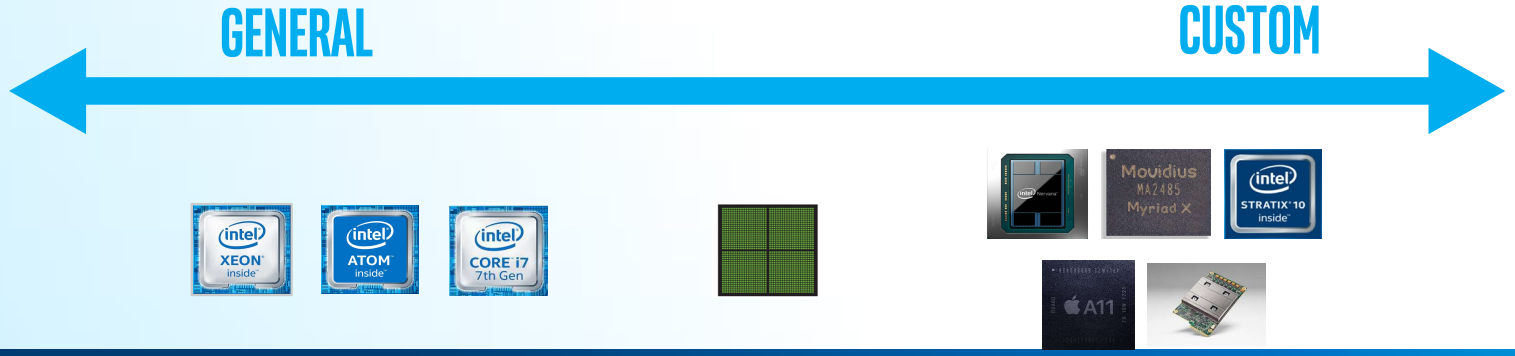
Apple Neural Engine* system is on a chip designed for fast inference.

- Apple Neural Engine shipped on late 2017 iPhones.
- Separate from main CPU and GPU on iPhone.
- Specifically designed for fast neural net inference, capable of 600 billion operations per second.
- Highlighted importance of fast neural net inference to Apple's ability to provide its desired user experience.

Processor Types

We've covered processors that span general purpose use to custom built solutions.

- CPUs handle all workloads, serial and parallel.
- GPUs for fast parallel processing for images, graphics, and highly parallel workloads.
- Highly flexible and custom built solutions like FPGAs and Intel® NNP™



Learning Objectives Recap

In this lesson, we worked to:

- Define “datacenter”, “gateway”, and ”edge” computing
- Recall the key processor types for AI
- Explain how Intel® hardware applies to AI
- Describe Field Programmable Gate Arrays (FPGA)
- Compare Central (CPU) and Graphic (GPU) processing units


intel
HOMEWORK EXERCISE

Get Access to Hardware

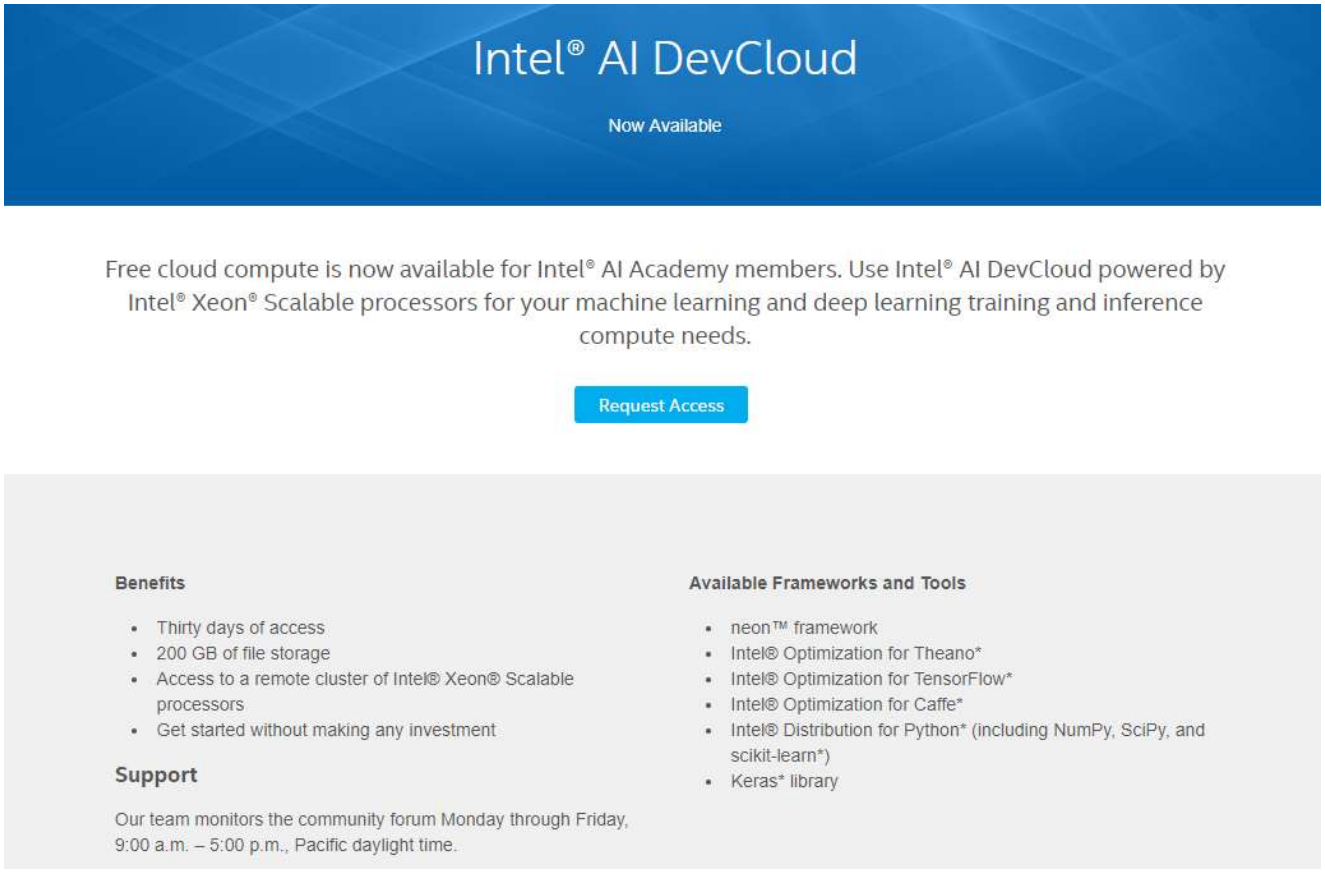
Sign up for Intel® AI DevCloud access

- Understand the tools & libraries that are available to use via remote access
- Sign up & receive access to start work on the Jupyter* notebooks from the Artificial Intelligence 501, Machine Learning 501, Deep Learning 501 or TensorFlow 501 student kits.
- Learn more at https://software.intel.com/en-us/ai-academy/students/kits

Intel® AI DevCloud Overview

Intel AI DevCloud™ is a server cluster consisting of Intel® Xeon® scalable processors.

- Storage, compute needed for DL.
- Pre-loaded with several Intel® optimized DL packages:
- neon™, Theano*, TensorFlow*, Caffe*, Keras*.
- Intel® Distribution for Python*.



https://software.intel.com/en-us/ai-academy/tools/devcloud


ADDITIONAL INTEL® COURSES

Other Resources: AI Student Kits

Intel® AI student kits includes courses on machine learning and deep learning.

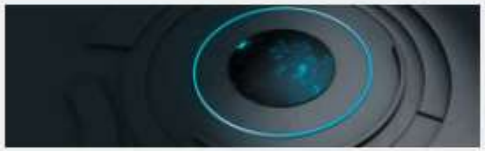
- All three contain concrete examples of implementing AI techniques.

Machine Learning 501

Get an overview of the fundamentals of machine learning on modern Intel® architecture. (12 weeks)

Get Started

Previously named Machine Learning 101



Deep Learning 501

Learn the basic techniques and foundations of deep learning on modern Intel® architecture. (12 weeks)

Get Started

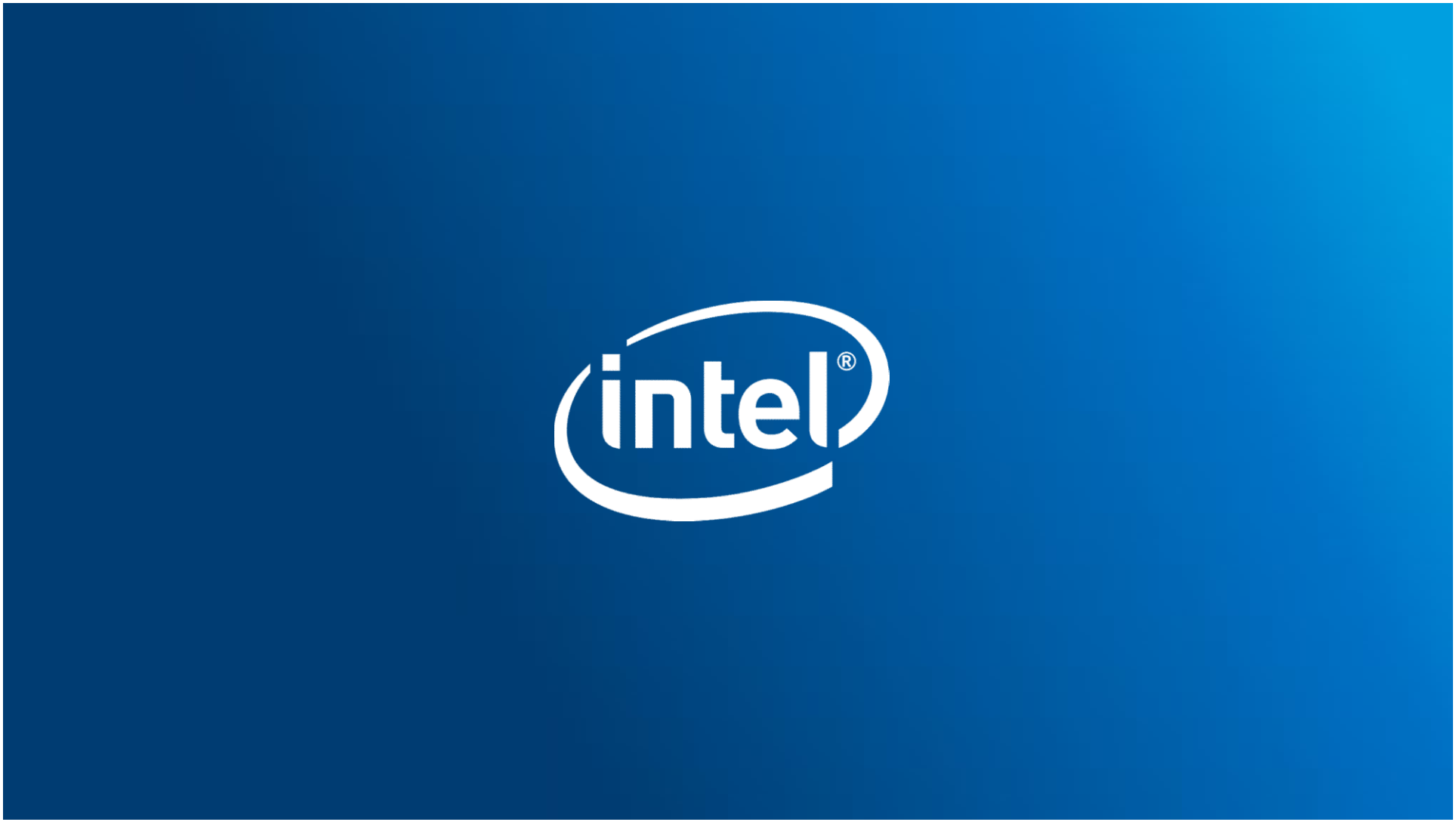
Previously named Deep Learning 101

TensorFlow* 501

Master the basics of using TensorFlow* with Intel® architecture. (8 weeks)

Get Started

Previously named TensorFlow 101


intel®

CONFIGURATION DETAILS (CONT'D)

2S Intel® Xeon® processor E5-2697A v4 on Apache Spark™ with MKL2017 up to 18x performance increase compared to 2S E5-2697 v2 + F2JBLAS machine learning training
BASELINE: Intel® Xeon® Processor E5-2697 v2 (12 Cores, 2.7 GHz), 256GB memory, CentOS 6.6*, F2JBLAS: https://github.com/fommil/netlib-java, Relative performance 1.0

Intel® Xeon® processor E5-2697 v2 Apache* Spark* Cluster: 1-Master + 8-Workers, 10Gbit/sec Ethernet fabric, Each system with 2 Processors, Intel® Xeon® processor E5-2697 v2 (12 Cores, 2.7 GHz), Hyper-Threading Enabled, 256GB RAM per System, 1-240GB SSD OS Drive, 12-3TB HDDs Data Drives Per System, CentOS* 6.6, Linux* 2.6.32-642.1.1.el6.x86_64, Intel® Intel® MKL 2017 build U1_20160808 , Cloudera Distribution for Hadoop (CDH) 5.7, Apache* Spark* 1.6.1 standalone, OMP_NUM_THREADS=1 set in CDH*, Total Java Heap Size of 200GB for Spark* Master and Workers, Relative performance up to 3.4x

Intel® Xeon® processor E5-2699 v3 Apache* Spark* Cluster: 1-Master + 8-Workers, 10Gbit/sec Ethernet fabric, Each system with 2 Processors, Intel® Xeon® processor E5-2699 v3 (18 Cores, 2.3 GHz), Hyper-Threading Enabled, 256GB RAM per System, 1-480GB SSD OS Drive, 12-4TB HDDs Data Drives Per System, CentOS* 7.0, Linux 3.10.0-229.el7.x86_64, Intel® Intel® MKL 2017 build U1_20160808 , Cloudera Distribution for Hadoop (CDH) 5.7, Apache* Spark* 1.6.1 standalone, OMP_NUM_THREADS=1 set in CDH*, Total Java Heap Size of 200GB for Spark* Master and Workers, Relative performance up to 8.8x

Intel® Xeon® processor E5-2697A v4 Apache* Spark* Cluster: 1-Master + 8-Workers, 10Gbit Ethernet/sec fabric, Each system with 2 Processors, Intel® Xeon® processor E5-2697A v4 (16 Cores, 2.6 GHz), Hyper-Threading Enabled, 256GB RAM per System, 1-800GB SSD OS Drive, 10-240GB SSDs Data Drives Per System, CentOS* 6.7, Linux 2.6.32-573.12.1.el6.x86_64, Intel® MKL 2017 build U1_20160808 , Cloudera Distribution for Hadoop (CDH) 5.7, Apache* Spark* 1.6.1 standalone, OMP_NUM_THREADS=1 set in CDH*, Total Java Heap Size of 200GB for Spark* Master and Workers, Relative performance up to 18x

Machine learning algorithm used for all configurations : Alternating Least Squares ALS Machine Learning Algorithm https://github.com/databricks/spark-perf

Intel® Xeon Phi™ Processor 7250 GoogleNet V1 Time-To-Train Scaling Efficiency up to 97% on 32 nodes
32 nodes of Intel® Xeon Phi™ processor 7250 (68 Cores, 1.4 GHz, 16GB MCDRAM: flat mode), 96GB DDR4 memory, Red Hat* Enterprise Linux 6.7, export OMP_NUM_THREADS=64 (the remaining 4 cores are used for driving communication) MKL 2017 Update 1, MPI: 2017.1.132, Endeavor KNL bin1 nodes, export I_MPI_FABRICS=tmi, export I_MPI_TMI_PROVIDER=psm2, Throughput is measured using "train" command. Data pre-partitioned across all nodes in the cluster before training. There is no data transferred over the fabric while training. Scaling efficiency computed as: (Single node performance / (N * Performance measured with N nodes))*100, where N = Number of nodes

Intel® optimized Caffe*: Intel internal version of Caffe

GoogLeNetV1: http://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43022.pdf, batch size 1536

Intel® Xeon Phi ™ processor 7250 up to 400x performance increase with Intel Optimized Frameworks compared to baseline out of box performance
BASELINE: Caffe Out Of the Box, Intel® Xeon Phi™ processor 7250 (68 Cores, 1.4 GHz, 16GB MCDRAM: cache mode), 96GB memory, Centos 7.2 based on Red Hat* Enterprise Linux 7.2, BVLC-Caffe: https://github.com/BVLC/caffe, with OpenBLAS, Relative performance 1.0

NEW: Caffe: Intel® Xeon Phi™ processor 7250 (68 Cores, 1.4 GHz, 16GB MCDRAM: cache mode), 96GB memory, Centos 7.2 based on Red Hat* Enterprise Linux 7.2, Intel® Caffe: : https://github.com/intel/caffe based on BVLC Caffe as of Jul 16, 2016, MKL GOLD UPDATE1, Relative performance up to 400x

AlexNet used for both configuration as per https://papers.nips.cc/paper/4824-Large image database-classification-with-deep-convolutional-neural-networks.pdf, Batch Size: 256

Intel® Xeon Phi™ Processor 7250, 32 node cluster with Intel® Omni Path Fabric up to 97% GoogleNetV1 Time-To-Train Scaling Efficiency
Caffe: Intel® Xeon Phi™ processor 7250 (68 Cores, 1.4 GHz, 16GB MCDRAM: flat mode), 96GB DDR4 memory, Red Hat* Enterprise Linux 6.7, Intel® Caffe: https://github.com/intel/caffe, not publically available yet

export OMP_NUM_THREADS=64 (the remaining 4 cores are used for driving communication)
MKL 2017 Update 1, MPI: 2017.1.132, Endeavor KNL bin1 nodes, export I_MPI_FABRICS=tmi, export I_MPI_TMI_PROVIDER=psm2, Throughput is measured using "train" command. Split the images across nodes and copied locally on each node at the beginning of training. No IO happens over fabric while training.

GoogLeNetV1: http://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43022.pdf, batch size 1536

Intel® Xeon Phi ™ processor Knights Mill up to 4x estimated performance improvement over Intel® Xeon Phi™ processor 7290
BASELINE: Intel® Xeon Phi™ Processor 7290 (16GB, 1.50 GHz, 72 core) with 192 GB Total Memory on Red Hat Enterprise Linux* 6.7 kernel 2.6.32-573 using MKL 11.3 Update 4, Relative performance 1.0

NEW: Intel® Xeon phi™ processor family – Knights Mill, Relative performance up to 4x

Intel® Arria 10 – 1150 FPGA energy efficiency on Caffe/AlexNet up to 25 img/s/w with FP16 at 297MHz
Vanilla AlexNet Classification Implementation as specified by http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf, Training Parameters taken from Caffe open-source Framework are 224x224x3 Input, 1000x1 Output, FP16 with Shared Block-Exponents, All compute layers (incl. Fully Connected) done on the FPGA except for Softmax, Arria 10-1150 FPGA, -1 Speed Grade on Altera PCIe DevKit with x72 DDR4 @ 1333 MHz, Power measured through on-board power monitor (FPGA POWER ONLY), ACDS 16.1 Internal Builds + OpenCL SDK 16.1 Internal Build, Compute machine is an HP Z620 Workstation, Xeon E5-1660 at 3.3 GHz with 32GB RAM. The Xeon is not used for compute.

Knights Mill performance : Results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling, and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance. Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice Revision #20110804
Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit: http://www.intel.com/performance Source: Intel measured everything except Knights Mill which is estimated as of November 2016

CONFIGURATION DETAILS (CONT'D)

Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice Revision #20110804.

Results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling, and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors.
Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit http://www.intel.com/performance/datacenter. Tested by Intel as of 14 June 2016. Configurations:

Faster and more scalable than GPU claim based on Intel analysis and testing

Up to 2.3x faster training per system claim based on AlexNet* topology workload (batch size = 1024) using a large image database running 4-nodes Intel Xeon Phi processor 7250 (16 GB MCDRAM, 1.4 GHz, 68 Cores) in Intel® Server System LADMP2312KXXX41, 96GB DDR4-2400 MHz, quad cluster mode, MCDRAM flat memory mode, Red Hat Enterprise Linux* 6.7 (Santiago), 1.0 TB SATA drive WD1003FZEX-00MK2A0 System Disk, running Intel® Optimized DNN Framework (internal development version) training 1.33 million images in 10.5 hours compared to 1-node host with four NVIDIA "Maxwell" GPUs training 1.33 million images in 25 hours (source: http://www.slideshare.net/NVIDIA/gtc-2016-opening-keynote slide 32).

Up to 38% better scaling efficiency at 32-nodes claim based on GoogLeNet deep learning image classification training topology using a large image database comparing one node Intel Xeon Phi processor 7250 (16 GB MCDRAM, 1.4 GHz, 68 Cores) in Intel® Server System LADMP2312KXXX41, DDR4 96GB DDR4-2400 MHz, quad cluster mode, MCDRAM flat memory mode, Red Hat* Enterprise Linux 6.7, Intel® Optimized DNN Framework with 87% efficiency to unknown hosts running 32 each NVIDIA Tesla* K20 GPUs with a 62% efficiency (Source: http://arxiv.org/pdf/1511.00175v2.pdf showing FireCaffe* with 32 NVIDIA Tesla* K20s (Titan Supercomputer*) running GoogLeNet* at 20x speedup over Caffe* with 1 K20).

Up to 6 SP TFLOPS based on the Intel Xeon Phi processor peak theoretical single-precision performance is preliminary and based on current expectations of cores, clock frequency and floating point operations per cycle. FLOPS = cores x clock frequency x floating-point operations per second per cycle

Up to 3x faster single-threaded performance claim based on Intel estimates of Intel Xeon Phi processor 7290 vs. coprocessor 7120 running XYZ workload.

Up to 2.3x faster training per system claim based on AlexNet* topology workload (batch size = 1024) using a large image database running 4-nodes Intel Xeon Phi processor 7250 (16 GB MCDRAM, 1.4 GHz, 68 Cores) in Intel® Server System LADMP2312KXXX41, 96GB DDR4-2400 MHz, quad cluster mode, MCDRAM flat memory mode, Red Hat Enterprise Linux* 6.7 (Santiago), 1.0 TB SATA drive WD1003FZEX-00MK2A0 System Disk, running Intel® Optimized DNN Framework, Intel® Optimized Caffe (internal development version) training 1.33 billion images in 10.5 hours compared to 1-node host with four NVIDIA "Maxwell" GPUs training 1.33 billion images in 25 hours (source: http://www.slideshare.net/NVIDIA/gtc-2016-opening-keynote slide 32).

Up to 38% better scaling efficiency at 32-nodes claim based on GoogLeNet deep learning image classification training topology using a large image database comparing one node Intel Xeon Phi processor 7250 (16 GB MCDRAM, 1.4 GHz, 68 Cores) in Intel® Server System LADMP2312KXXX41, DDR4 96GB DDR4-2400 MHz, quad cluster mode, MCDRAM flat memory mode, Red Hat* Enterprise Linux 6.7, Intel® Optimized DNN Framework with 87% efficiency to unknown hosts running 32 each NVIDIA Tesla* K20 GPUs with a 62% efficiency (Source: http://arxiv.org/pdf/1511.00175v2.pdf showing FireCaffe* with 32 NVIDIA Tesla* K20s (Titan Supercomputer*) running GoogLeNet* at 20x speedup over Caffe* with 1 K20).

Up to 50x faster training on 128-node as compared to single-node based on AlexNet* topology workload (batch size = 1024) training time using a large image database running one node Intel Xeon Phi processor 7250 (16 GB MCDRAM, 1.4 GHz, 68 Cores) in Intel® Server System LADMP2312KXXX41, 96GB DDR4-2400 MHz, quad cluster mode, MCDRAM flat memory mode, Red Hat Enterprise Linux* 6.7 (Santiago), 1.0 TB SATA drive WD1003FZEX-00MK2A0 System Disk, running Intel® Optimized DNN Framework, training in 39.17 hours compared to 128-node identically configured with Intel® Omni-Path Host Fabric Interface Adapter 100 Series 1 Port PCIe x16 connectors training in 0.75 hours. Contact your Intel representative for more information on how to obtain the binary. For information on workload, see https://papers.nips.cc/paper/4824-Large image database-classification-with-deep-convolutional-neural-networks.pdf.

Up to 30x software optimization improvement claim based on customer CNN training workload running 2S Intel® Xeon® processor E5-2680 v3 running Berkeley Vision and Learning Center* (BVLC) Caffe + OpenBlas* library and then run tuned on the Intel® Optimized Caffe (internal development version) + Intel® Math Kernel Library (Intel® MKL).

CONFIGURATION DETAILS (CONT'D)

Platform: 2S Intel® Xeon® Platinum 8180 processor @ 2.50GHz (28 cores), HT disabled, turbo disabled, scaling governor set to "performance" via intel_pstate driver, 384GB DDR4-2666 ECC RAM. CentOS Linux* release 7.3.1611 (Core), Linux kernel 3.10.0-514.10.2.el7.x86_64. SSD: Intel® SSD DC S3700 Series (800GB, 2.5in SATA 6Gb/s, 25nm, MLC).

Performance measured with: Environment variables: KMP_AFFINITY='granularity=fine, compact', OMP_NUM_THREADS=56, CPU Freq set with cpupower frequency-set -d 2.5G -u 3.8G -g performance

Deep Learning Frameworks:

- Caffe*: (http://github.com/intel/caffe/), revision f96b759f71b2281835f690af267158b82b150b5c. Inference measured with "caffe time --forward_only" command, training measured with "caffe time" command. For "ConvNet" topologies, dummy dataset was used. For other topologies, data was stored on local storage and cached in memory before training. Topology specs from https://github.com/intel/caffe/tree/master/models/intel_optimized_models (GoogLeNet, AlexNet, and ResNet-50), https://github.com/intel/caffe/tree/master/models/default_vgg_19 (VGG-19), and https://github.com/soumith/convnet-benchmarks/tree/master/caffe/imagenet_winners (ConvNet benchmarks; files were updated to use newer Caffe prototxt format but are functionally equivalent). Intel C++ compiler ver. 17.0.2 20170213, Intel MKL small libraries version 2018.0.20170425. Caffe run with "numactl -l".

- TensorFlow*: (https://github.com/tensorflow/tensorflow), commit id 207203253b6f8ea5e938a512798429f91d5b4e7e. Performance numbers were obtained for three convnet benchmarks: alexnet, googlenetv1, vgg(https://github.com/soumith/convnet-benchmarks/tree/master/tensorflow) using dummy data. GCC 4.8.5, Intel MKL small libraries version 2018.0.20170425, interop parallelism threads set to 1 for alexnet, vgg benchmarks, 2 for googlenet benchmarks, intra op parallelism threads set to 56, data format used is NCHW, KMP_BLOCKTIME set to 1 for googlenet and vgg benchmarks, 30 for the alexnet benchmark. Inference measured with --caffe time -forward_only -engine MKL2017option, training measured with --forward_backward_only option.

- MxNet: (https://github.com/dmlc/mxnet/), revision 5efd91a71f36fea483e882b0358c8d46b5a7aa20. Dummy data was used. Inference was measured with "benchmark_score.py", training was measured with a modified version of benchmark_score.py which also runs backward propagation. Topology specs from https://github.com/dmlc/mxnet/tree/master/example/image-classification/symbols. GCC 4.8.5, Intel MKL small libraries version 2018.0.20170425.

- Neon: ZP/MKL_CHWN branch commit id:52bd02acb947a2adabb8a227166a7da5d9123b6d. Dummy data was used. The main.py script was used for benchmarking , in mkl mode. ICC version used : 17.0.3 20170404, Intel MKL small libraries version 2018.0.20170425.

CONFIGURATION DETAILS (CONT'D)

Platform: 2S Intel® Xeon® processor E5-2697 v2 @ 2.70GHz (12 cores), HT enabled, turbo enabled, scaling governor set to "performance" via intel_pstate driver, 256GB DDR3-1600 ECC RAM. CentOS Linux* release 7.3.1611 (Core), Linux kernel 3.10.0-514.21.1.el7.x86_64. SSD: Intel® SSD 520 Series 240GB, 2.5in SATA 6Gb/s, 25nm, MLC.

Performance measured with: Environment variables: KMP_AFFINITY='granularity=fine, compact,1,0', OMP_NUM_THREADS=24, CPU Freq set with cpupower frequency-set -d 2.7G -u 3.5G -g performance

Deep Learning Frameworks:

- Caffe*: (http://github.com/intel/caffe/), revision b0ef3236528a2c7d2988f249d347d5fdae831236. Inference measured with "caffe time --forward_only" command, training measured with "caffe time" command. For "ConvNet" topologies, dummy dataset was used. For other topologies, data was stored on local storage and cached in memory before training. Topology specs from https://github.com/intel/caffe/tree/master/models/intel_optimized_models (GoogLeNet, AlexNet, and ResNet-50), https://github.com/intel/caffe/tree/master/models/default_vgg_19 (VGG-19), and https://github.com/soumith/convnet-benchmarks/tree/master/caffe/imagenet_winners (ConvNet benchmarks; files were updated to use newer Caffe prototxt format but are functionally equivalent). GCC 4.8.5, Intel MKL small libraries version 2017.0.2.20170110.

CONFIGURATION DETAILS (CONT'D)

Intel® Caffe*: Intel internal version of Caffe

Intel® Arria 10 – 1150 FPGA energy efficiency on Caffe/AlexNet up to 25 img/s/w with FP16 at 297MHz
Vanilla AlexNet Classification Implementation as specified by http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf, Training Parameters taken from Caffe open-source Framework are 224x224x3 Input, 1000x1 Output, FP16 with Shared Block-Exponents, All compute layers (incl. Fully Connected) done on the FPGA except for Softmax, Arria 10-1150 FPGA, -1 Speed Grade on Altera PCIe DevKit with x72 DDR4 @ 1333 MHz, Power measured through on-board power monitor
(FPGA POWER ONLY), ACDS 16.1 Internal Builds + OpenCL SDK 16.1 Internal Build, Compute machine is an HP Z620 Workstation, Intel® Xeon E5-1660 processor at 3.3 GHz with 32GB RAM. The Intel Xeon processor is not used for compute.

Projected data comes from verified deterministic model of DLA IP implementation on FPGA

CONFIGURATION DETAILS (CONT'D)

INFERENCE using FP32 Batch Size Caffe GoogleNet v1 256 AlexNet 256.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit http://www.intel.com/performance Source: Intel measured as of June 2017 Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Configurations 40, 41:

Platform: 2S Intel® Xeon® Platinum 8180 CPU @ 2.50GHz (28 cores), HT disabled, turbo disabled, scaling governor set to "performance" via intel_pstate driver, 384GB DDR4-2666 ECC RAM. CentOS Linux release 7.3.1611 (Core), Linux kernel 3.10.0-514.10.2.el7.x86_64. SSD: Intel® SSD DC S3700 Series (800GB, 2.5in SATA 6Gb/s, 25nm, MLC). Performance measured with: Environment variables: KMP_AFFINITY='granularity=fine, compact', OMP_NUM_THREADS=56, CPU Freq set with cpupower frequency-set -d 2.5G -u 3.8G -g performance. Deep Learning Frameworks: Caffe*: (http://github.com/intel/caffe/), revision f96b759f71b2281835f690af267158b82b150b5c. Inference measured with "caffe time --forward_only" command, training measured with "caffe time" command. For "ConvNet" topologies, dummy dataset was used. For other topologies, data was stored on local storage and cached in memory before training. Topology specs from https://github.com/intel/caffe/tree/master/models/intel_optimized_models (GoogLeNet, AlexNet, and ResNet-50), https://github.com/intel/caffe/tree/master/models/default_vgg_19 (VGG-19), and https://github.com/soumith/convnet-benchmarks/tree/master/caffe/imagenet_winners (ConvNet benchmarks; files were updated to use newer Caffe prototxt format but are functionally equivalent). Intel C++ compiler ver. 17.0.2 20170213, Intel MKL small libraries version 2018.0.20170425. Caffe run with "numactl -l".

Platform: 2S Intel® Xeon® CPU E5-2699 v3 @ 2.30GHz (18 cores), HT enabled, turbo disabled, scaling governor set to "performance" via intel_pstate driver, 256GB DDR4-2133 ECC RAM. CentOS Linux release 7.3.1611 (Core), Linux kernel 3.10.0-514.el7.x86_64. OS drive: Seagate* Enterprise ST2000NX0253 2 TB 2.5" Internal Hard Drive. Performance measured with: Environment variables: KMP_AFFINITY='granularity=fine, compact,1,0', OMP_NUM_THREADS=36, CPU Freq set with cpupower frequency-set -d 2.3G -u 2.3G -g performance. Deep Learning Frameworks: Intel Caffe: (http://github.com/intel/caffe/), revision b0ef3236528a2c7d2988f249d347d5fdae831236. Inference measured with "caffe time --forward_only" command, training measured with "caffe time" command. For "ConvNet" topologies, dummy dataset was used. For other topologies, data was stored on local storage and cached in memory before training. Topology specs from https://github.com/intel/caffe/tree/master/models/intel_optimized_models (GoogLeNet, AlexNet, and ResNet-50), https://github.com/intel/caffe/tree/master/models/default_vgg_19 (VGG-19), and https://github.com/soumith/convnet-benchmarks/tree/master/caffe/imagenet_winners (ConvNet benchmarks; files were updated to use newer Caffe prototxt format but are functionally equivalent). GCC 4.8.5, MKLML version 2017.0.2.20170110. BVLC-Caffe: https://github.com/BVLC/caffe, Inference & Training measured with "caffe time" command. For "ConvNet" topologies, dummy dataset was used. For other topologies, data was st ored on local storage and cached in memory before training BVLC Caffe (http://github.com/BVLC/caffe), revision 91b09280f5233cafc62954c98ce8bc4c204e7475 (commit date 5/14/2017). BLAS: atlas ver. 3.10.1.

CONFIGURATION DETAILS (CONT'D)

Intel® and SURFsara* Research Collaboration
MareNostrum4/BSC* Configuration Details

***MareNostrum4/Barcelona Supercomputing Center**: https://www.bsc.es/

Compute Nodes: 2 sockets Intel® Xeon® Platinum 8160 CPU with 24 cores each @ 2.10GHz for a total of 48 cores per node, 2 Threads per core, L1d 32K; L1i cache 32K; L2 cache 1024K; L3 cache 33792K, 96 GB of DDR4, Intel® Omni-Path Host Fabric Interface, dual-rail. Software: Intel® MPI Library 2017 Update 4Intel® MPI Library 2019 Technical Preview OFI 1.5.0PSM2 w/ Multi-EP, 10 Gbit Ethernet, 200 GB local SSD, Red Hat* Enterprise Linux* 6.7.

Intel optimized Caffe*: Intel® version of Caffe; http://github.com/intel/caffe/, revision 8012927bf2bf70231cbc7ff55de0b1bc11de4a69.
Intel® MKL version: mklml_lnx_2018.0.20170425; Intel® MLSL version: l_mlsl_2017.1.016

Model: Topology specs from https://github.com/intel/caffe/tree/master/models/intel_optimized_models (ResNet-50) and modified for wide-RedNet-50. Batch size as stated in the performance chart

Time-To-Train: measured using "train" command. Data copied to memory on all nodes in the cluster before training. No input image data transferred over the fabric while training;

Performance measured with:
export OMP_NUM_THREADS=44 (the remaining 4 cores are used for driving communication), export I_MPI_FABRICS=tmi, export I_MPI_TMI_PROVIDER=psm2

OMP_NUM_THREADS=44 KMP_AFFINITY="proclist=[0-87],granularity=thread,explicit" KMP_HW_SUBSET=1t MLSL_NUM_SERVERS=4 mpiexec.hydra -PSM2 -l -n $SLURM_JOB_NUM_NODES -ppn 1 -f hosts2 -genv OMP_NUM_THREADS 44 -env KMP_AFFINITY "proclist=[0-87],granularity=thread,explicit" -env KMP_HW_SUBSET 1t -genv I_MPI_FABRICS tmi -genv I_MPI_HYDRA_BRANCH_COUNT $SLURM_JOB_NUM_NODES -genv I_MPI_HYDRA_PMI_CONNECT alltoall sh -c 'cat /ilsvrc12_train_lmdb_striped_64/data.mdb > /dev/null ; cat /ilsvrc12_val_lmdb_striped_64/data.mdb > /dev/null ; ulimit -u 8192 ; ulimit -a ; numactl -H ; /caffe/build/tools/caffe train --solver=/caffe/models/intel_optimized_models/multinode/resnet_50_256_nodes_8k_batch/solver_poly_quick_large.prototxt -engine "MKL2017"

SURFsara blog: https://blog.surf.nl/en/imagenet-1k-training-on-intel-xeon-phi-in-less-than-40-minutes/ ; Researchers: Valeriu Codreanu, Ph.D. (PI).; Damian Podareanu, MSc; SURFsara* & Vikram Saletore, Ph.D. (co-PI): Intel Corp.

*SURFsara B.V. is the Dutch national high-performance computing and e-Science support center. Amsterdam Science Park, Amsterdam, The Netherlands.