ZIB Workshop "Hot Topics in Machine Learning”

Machine Learning on Accelerated Platforms

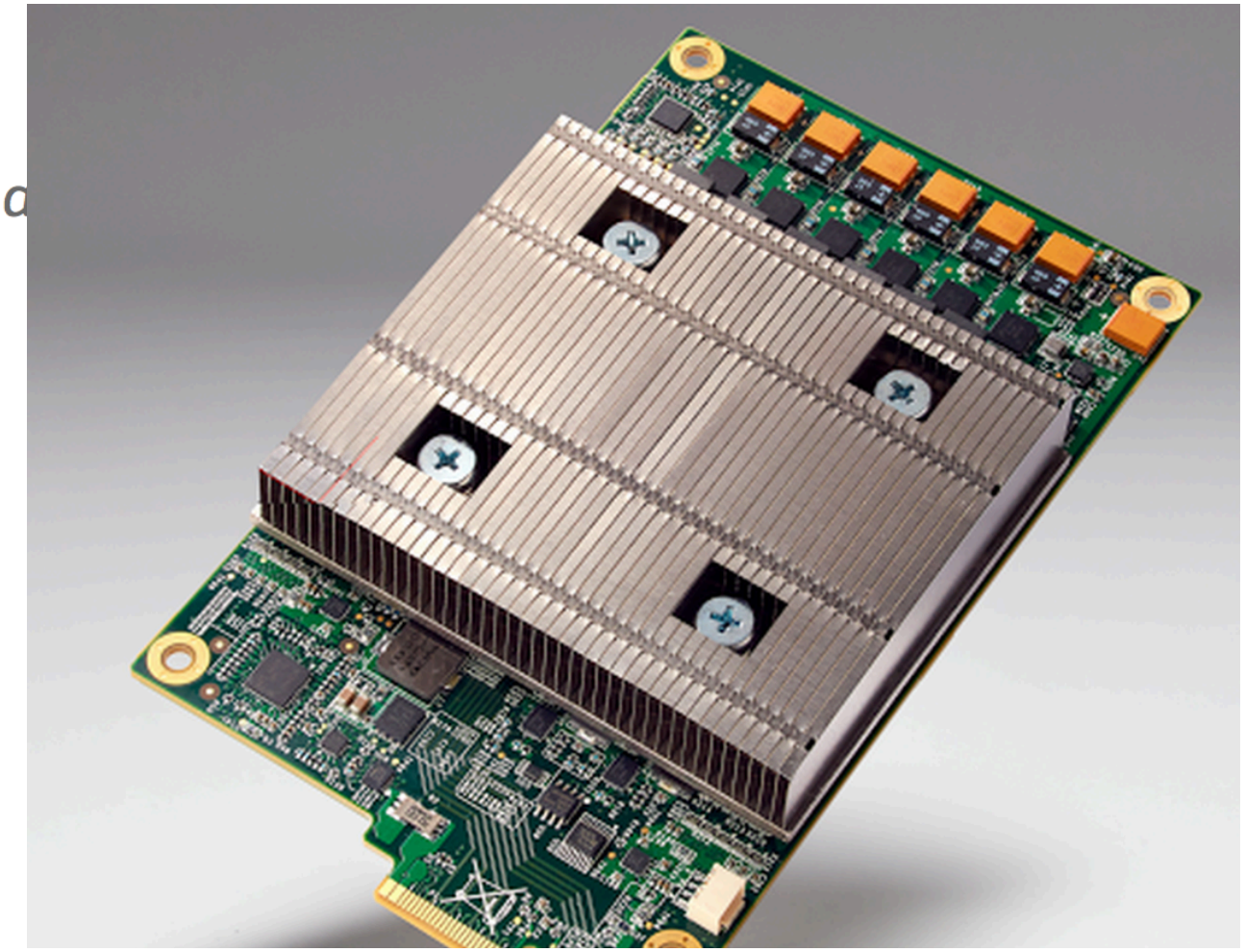
Thomas Steinke

February 27, 2017

Google reveals the mysterious custom hardware that powers AlphaGo

Meet TPU, a custom ASIC processor to power ma

TensorFlow



CPUs & CUDA GPUs and Google's TPU

ABOUT AN ORDER OF MAGNITUDE BETTER

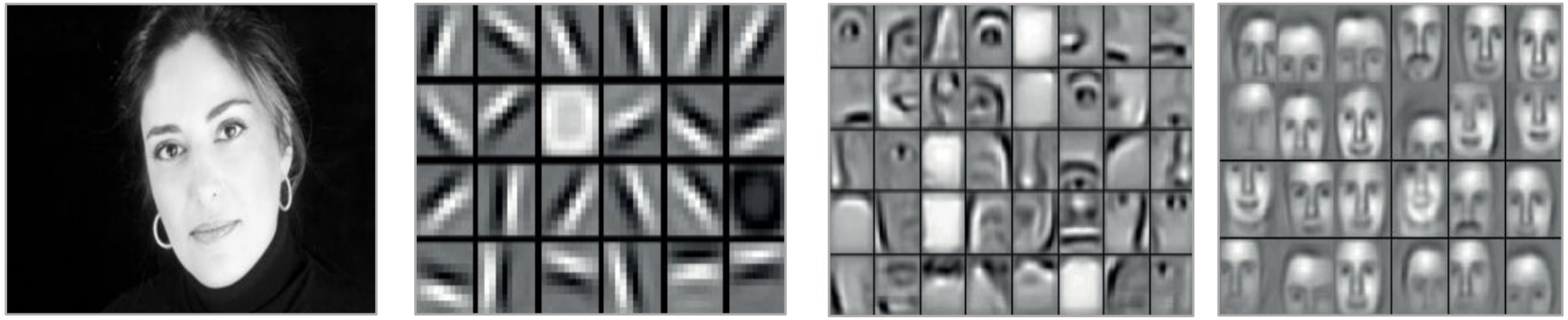
Source: http://www.theverge.com/circuitbreaker/2016/5/19/11716818/google-alphago-hardware-asic-chip-tensor-processor-unit-machine-learning

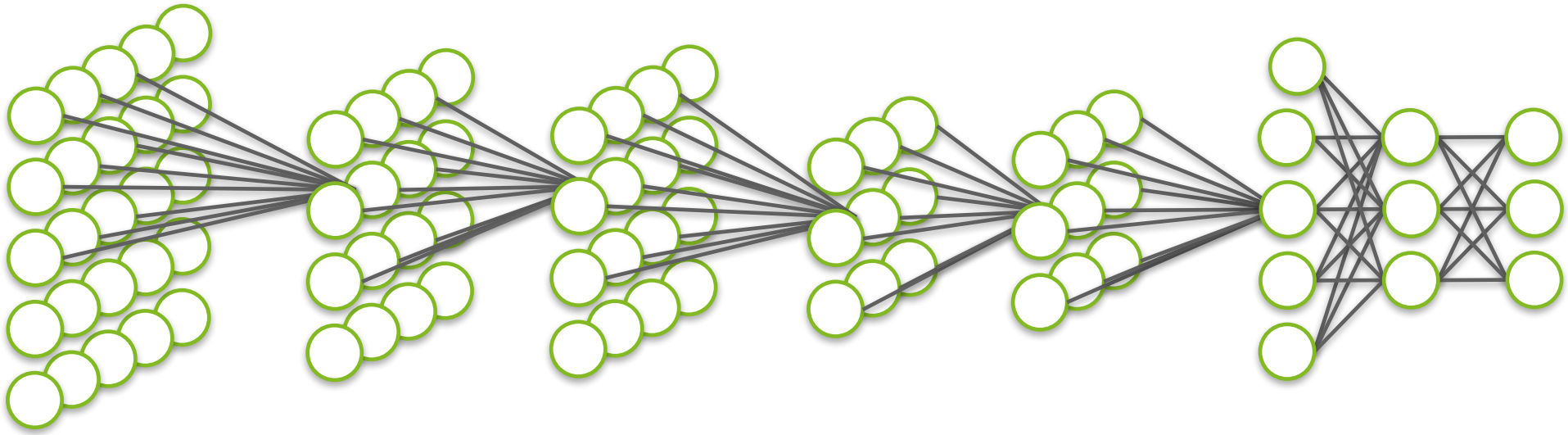
Outline

- Machine Learning on Accelerators in Research Environments
  - Hardware
  - Software
  - Relative performance data
- Outlook
  - Programming Heterogeneous Platforms
  - Future Devices

Machine Learning on Accelerators

Deep Learning: Face Recognition





Training data:

- 10-100**M** images

Network architecture:

- 10+ layers
- 1**B** parameters

Learning algorithm:

- **~30 exaflops**
- ~30 GPU days

Source: Nvidia

Key Requirements in Research Environments

- Easy of use
- Low barriers for programmability
- Broad spectrum of optimized libraries and tools
- Affordable hardware and software
- Sufficient speed
- ???

What Needs to be Accelerated?

Building blocks in learning algorithms:

- Compute:
  - Convolution
  - Matrix-matrix multiply in Deep Neural Networks (DNN)
    - Do not require IEEE floating-point ops
    - Weights may need only 3-8 bits

- Access to data:
  - High memory bandwidth (low computational intensity?)

- Scalability:
  - Inter-node communication

S. Gupta et al., Deep Learning with Limited Numerical Precision, arXiv, 2015
J. Pak et al., FPGA Based Implementation of Deep Neural Networks Using On-chip Memory only, arXiv, 2016

Basic Compute Devices

- Multi- and Many-core CPU
- Graphics Processing Unit (GPU)
- Field Programmable Gate Array (FGPA)

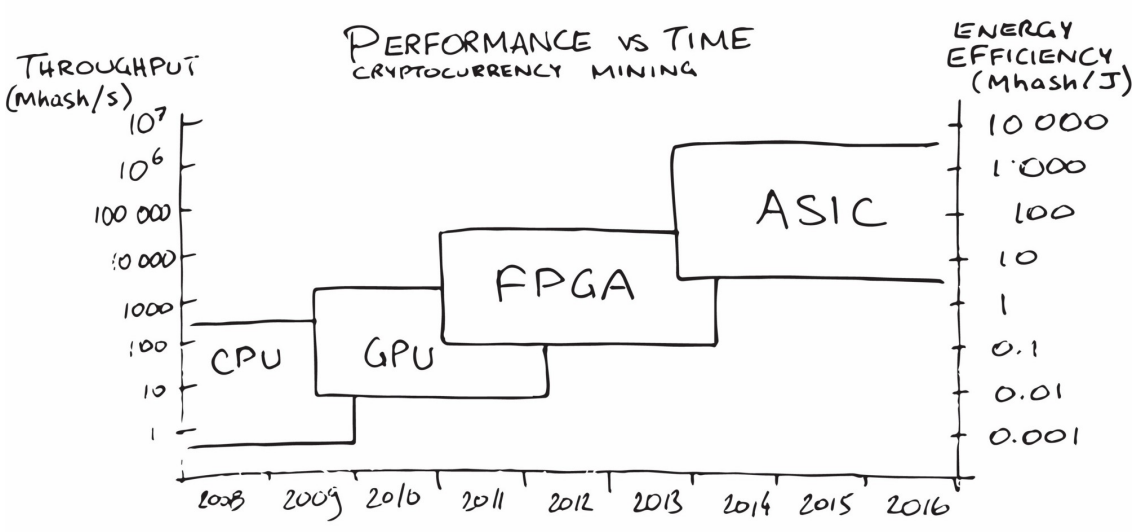
- Application-Specific Integrated Circuit (ASIC)

Matching Requirements of DNN by Hardware

Deep Neural Network	CPU	ManyC CPU	GPU	FPGA	ASIC
Parallelism	o	+	+	+	+
Matrix operations	o	+	+	o/+	++
FLOp/s	o	+	+	o	++
Memory bandwidth	-	+	+	+	++
Low precision ops	-	- / +	+	++	++
Energy efficiency	o	+	+	++	+++
Easy of programming	+	+	+	-- (+)	?
Price	$...$$	$$	$...$$	$$	$$$$

A Look at Compute Performance

- Computational capacity (throughput)
- Energy-efficiency (computations per Joule)
- Cost-efficiency (throughput per €)



Data for Mining Cryptocurrencies

- Single-core CPU: **1**
- Multi-core CPU: **10**
- Many-core CPU: **100**
- GPU: **100**
- FPGA: **1’000**
- ASIC: **10’000…1’000’000**

Source: Philip Jama, The Future of Machine Learning Hardware https://hackernoon.com/the-future-of-machine-learning-hardware-c872a0448be8#.x8t8vwae5

Processors for NN Training and Inference (I)

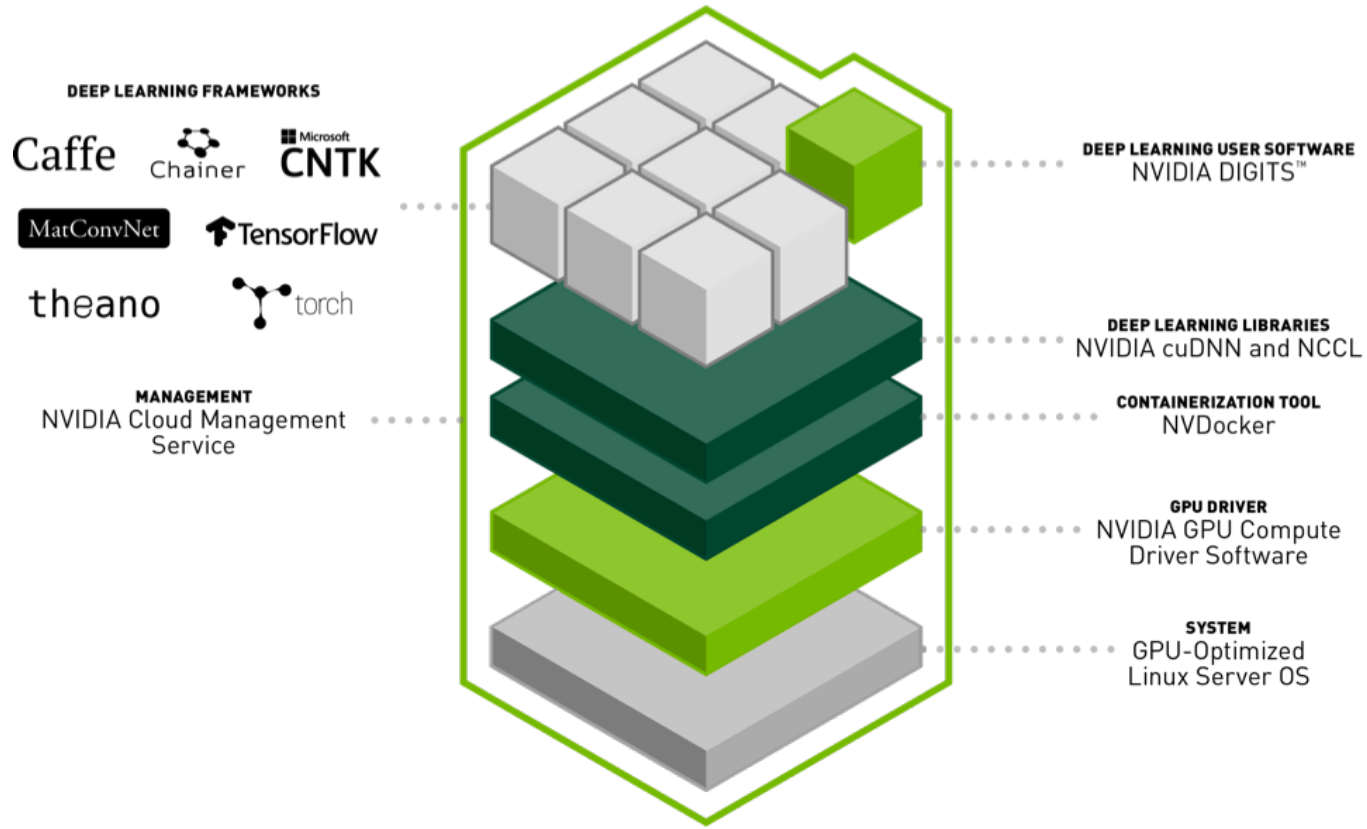
- Nvidia P40 (Pascal, 2017):
  - Performance: 47 INT8 TOp/s, 12 TFlop/s 32bit
  - Configuration: 24 GB global mem, 346 GB/s bandwidth, 250 W

- Nvidia M4 (Maxwell, 04/2016):
  - Performance: 2.2 Tflop/s 32bit
  - Configuration: 4 GB global mem, 88 GB/s bandwidth, 75W

NVIDIA DGX-1

AI supercomputer-appliance-in-a-box

- **8x Tesla P100** (170 TFlop/s 16bit perf)
- **NVLink** Hybrid Cube Mesh,
  160 GB/s bi-direct per peer
- Dual Xeon, 512 GB Memory
- 7 TB SSD Deep Learning Cache
- Dual 10GbE, Quad IB 100Gb
- 3RU – 3200W
- Optimized Deep Learning Software across the entire stack

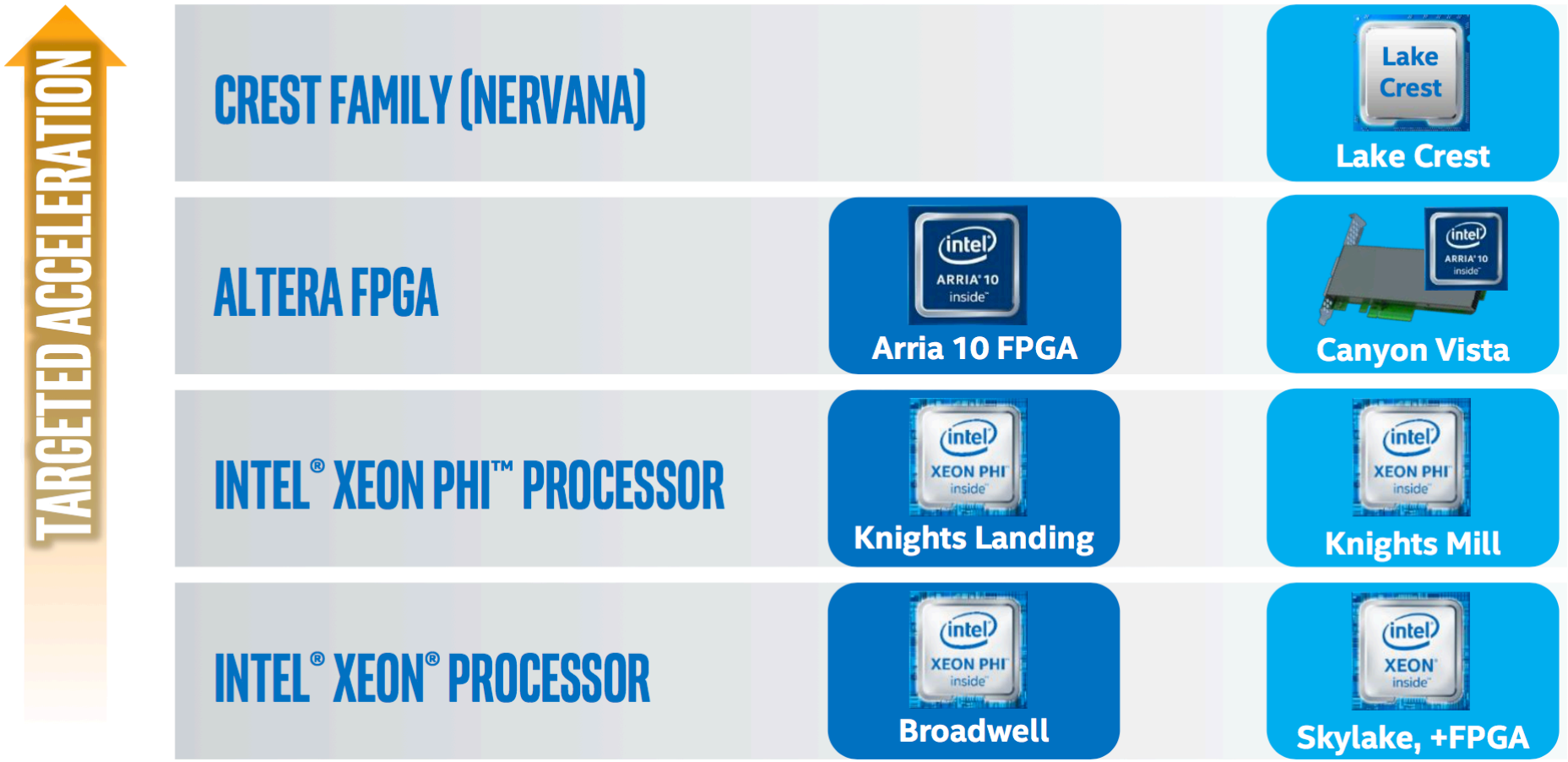


Source: Nvidia

Processors for NN Training and Inference (II)

- Intel Knights Mill: 3rd gen Intel Xeon Phi, 2017
  - many cores
  - FP16 arithmetic, "enhanced variable precision"
- Intel Xeon Skylake (AVX512)
- Intel Xeon Phi (Knights Landing)
  - Performance: 3.5 TFlop/s 64bit, 6+ TFlop/s 32bit
  - Configuration: 64-72 cores, 16 GB MCDRAM, 400+ GB/s mem bandwidth

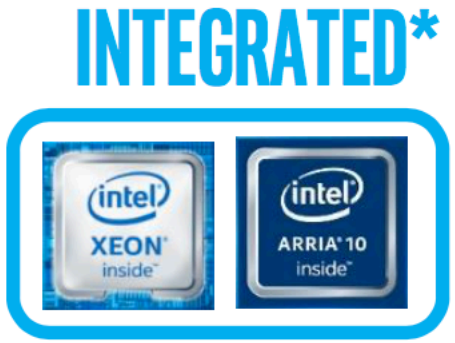
Intel Nervana Platform



Source: Intel, 2017

FPGA in Image Recognition

- FPGA vs. GPU: ~ 5x slower, 10x better energy efficiency
- any precision possible: 32 float … 8 bit fixed



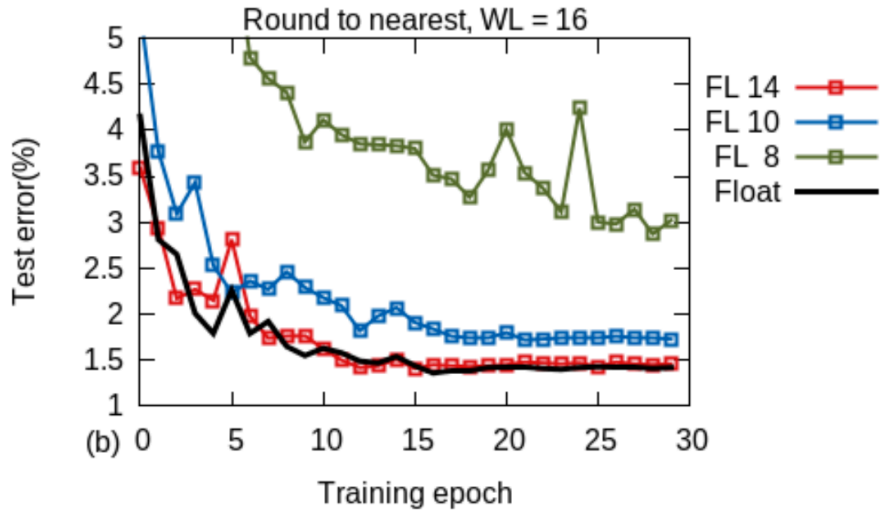
Topology	Datatype	Throughput*
AlexNet*	FloatP32	~360 Image/s
	FixedP16	~600 Image/s
	FixedP8	~750 Images/s
	FixedP8 Winograd	~1200 Images/s

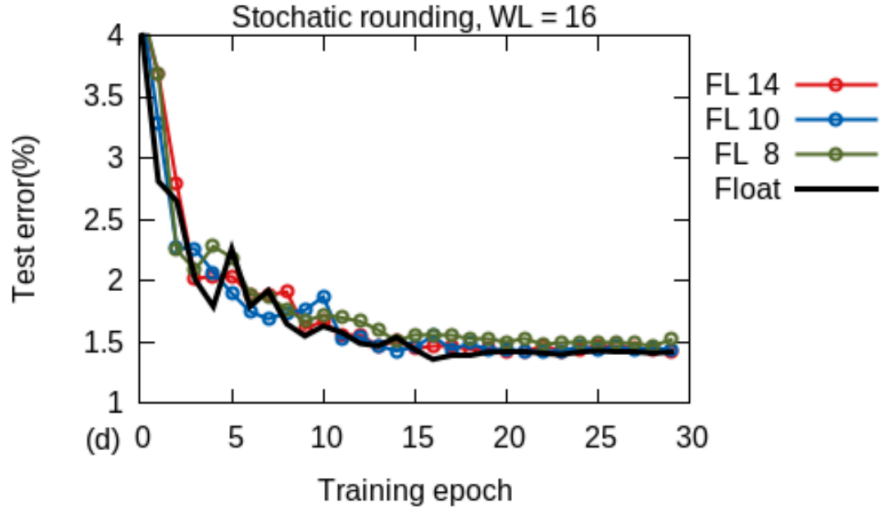
Winograd minimal filtering algorithm, reduce compute complexity by up to 4x w.r.t convolution

Source: Intel, 2017

Using FPGAs for NN Learning

- Fixed point: 16 bit,
  fractional part from 8 to 14 bits
- Multiply and accumulate
  (MACC) for inner products
- Stochastic rounding





S. Gupta et al., Deep Learning with Limited Numerical Precision, arXiv, 2015

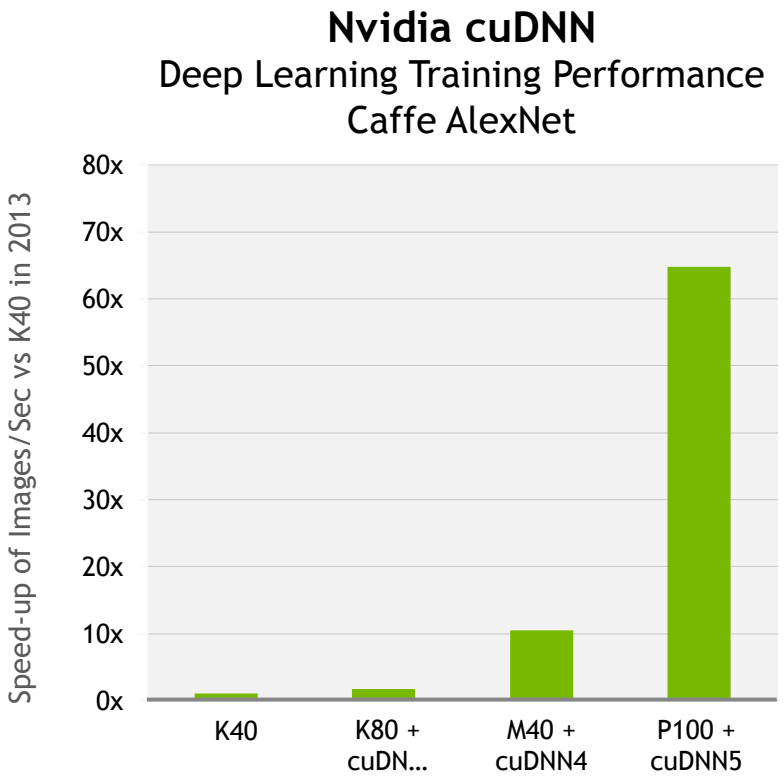
Software Stacks

Vendor Support of ML Software Frameworks

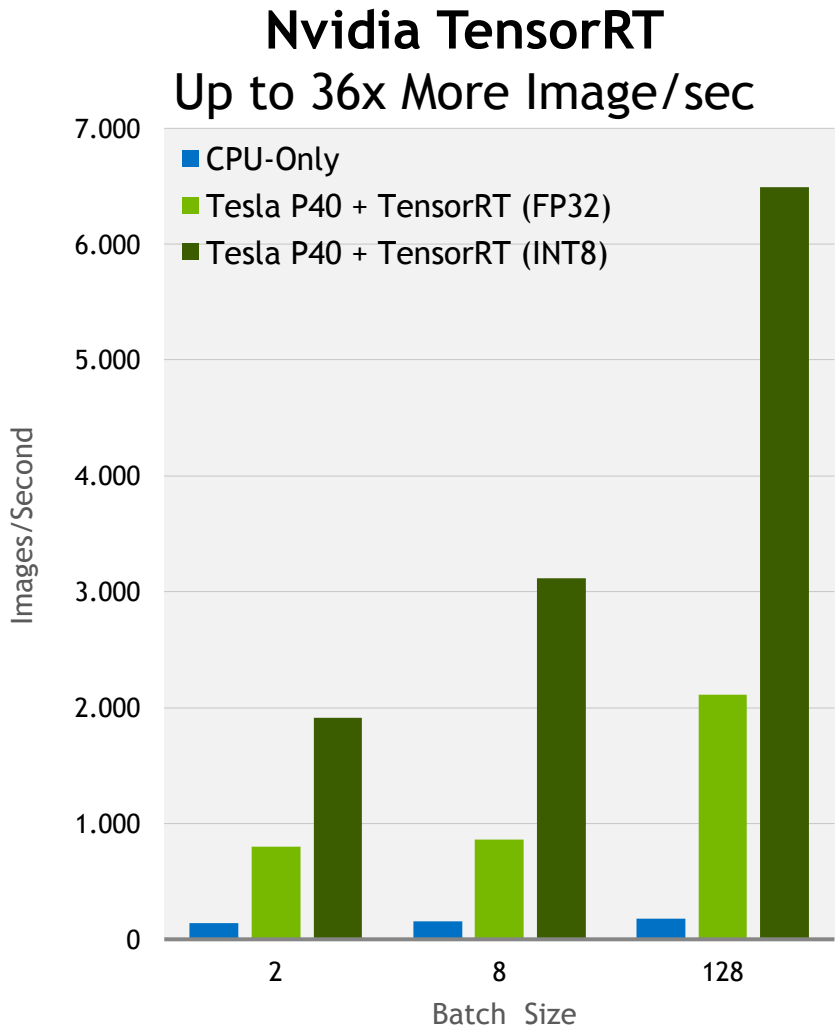
Caffe
Chainer
DL4J
Deeplearning4j
Mocha.jl
julia
KERAS
MatConvNet
Microsoft CNTK
mxnet
MINERVA
Purine
OpenDeep
Pylearn2
TensorFlow
torch
theano

#IntelAI

Nvidia Solutions



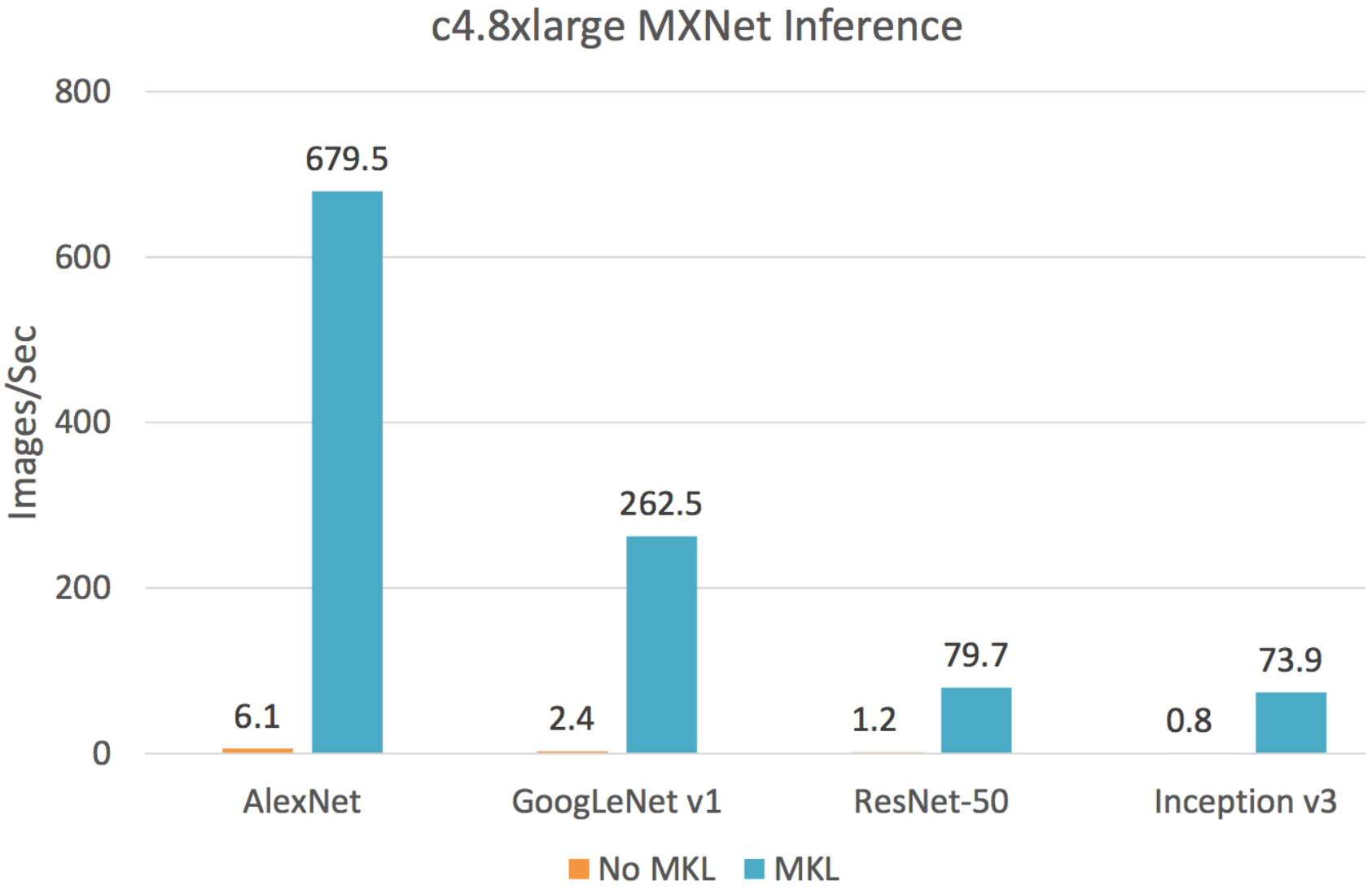
AlexNet training throughput on CPU: 1x E5-2680v3 12 Core 2.5GHz.
128GB System Memory, Ubuntu 14.04
M40 bar: **8x** M40 GPUs in a node, P100: **8x** P100 NVLink-enabled



GoogLenet, CPU-only vs Tesla P40 + TensorRT
CPU: 1 socket E4 2690 v4 @2.6 GHz, HT-on
GPU: 2 socket E5-2698 v3 @2.3 GHz, HT off, 1 P40 card in the box

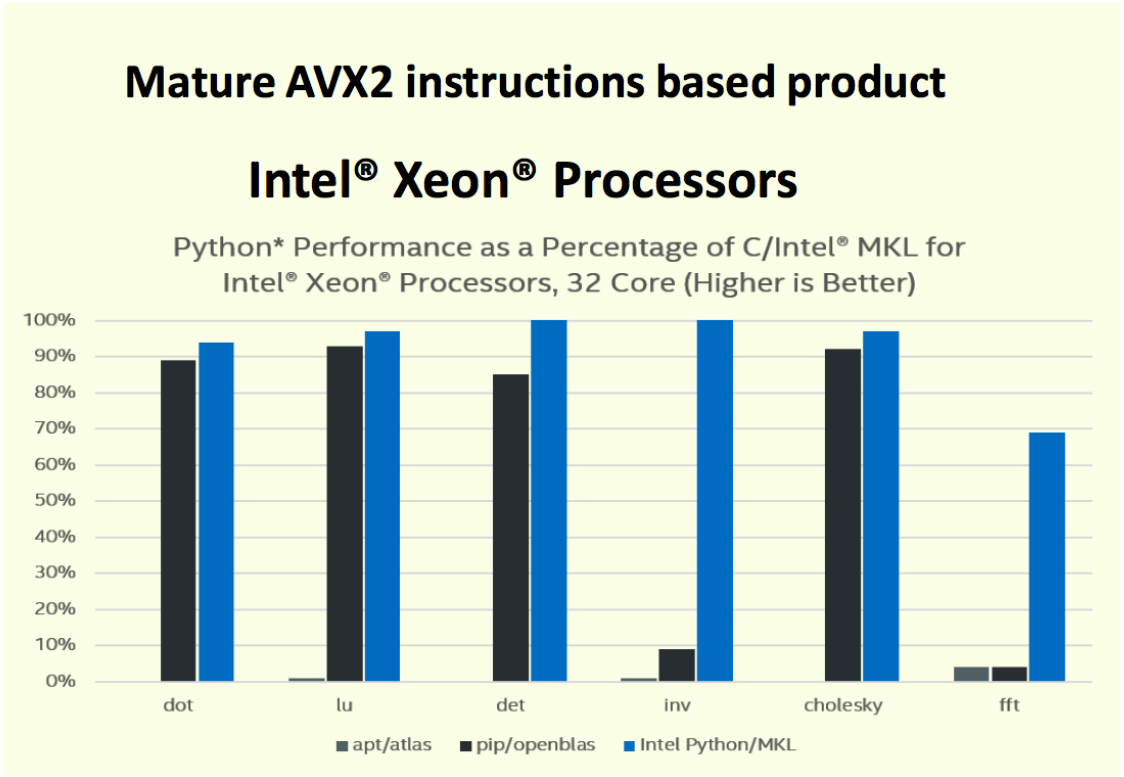
Intel MKL & Python

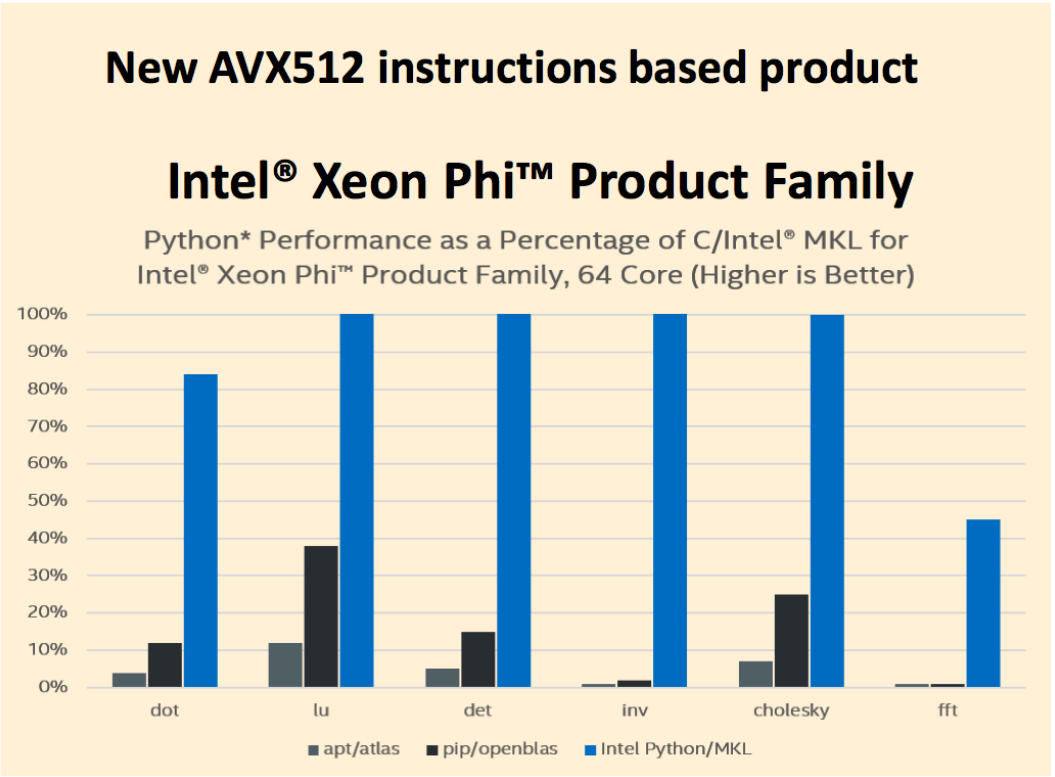
using Intel MKL



Intel MKL & Python

using Intel Python distro





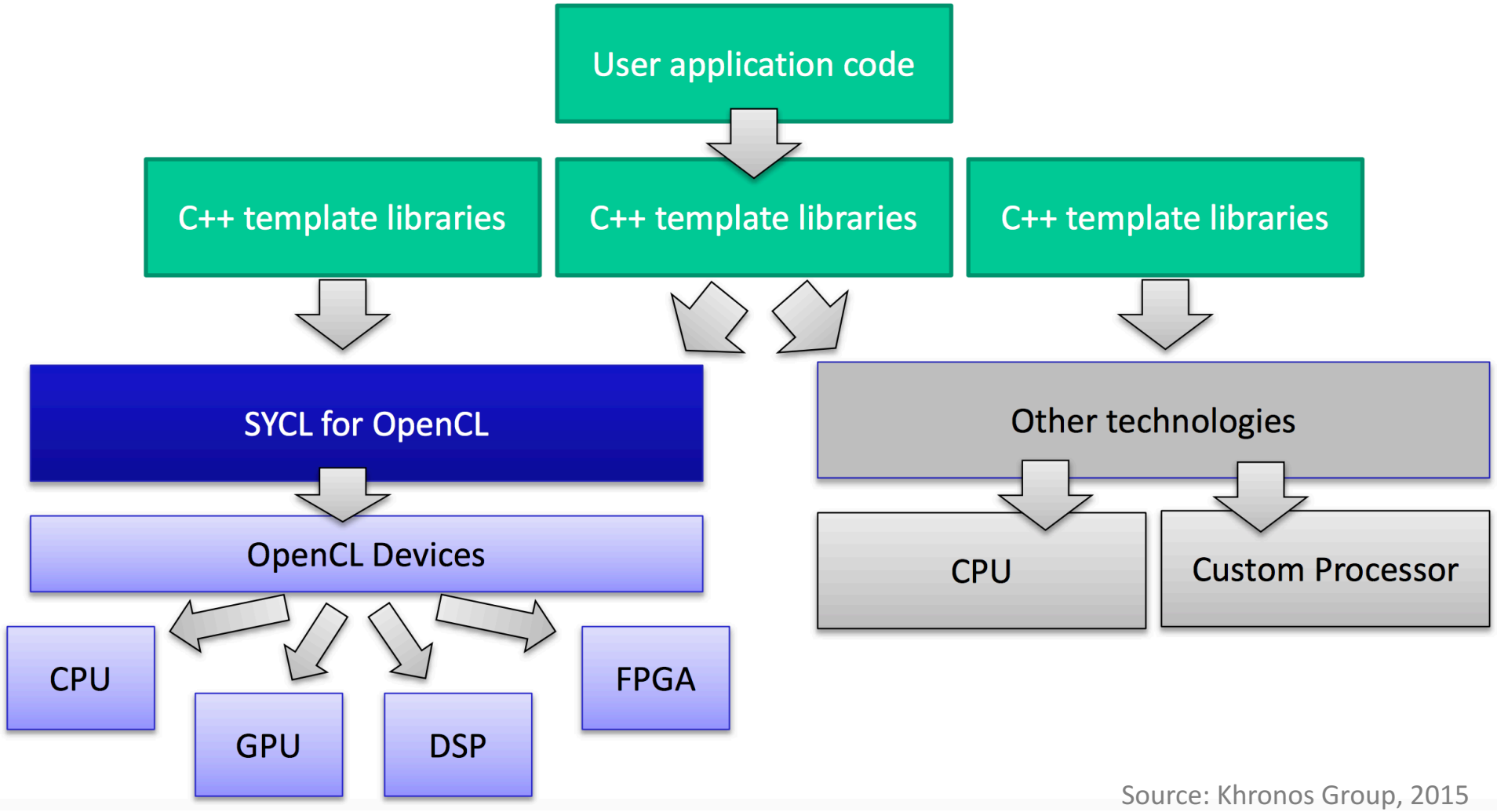
Programming for Heterogenous Platforms

SYCL - C++ Single-Source Heterogeneous Programming for OpenCL

- Cross-platform abstraction layer build on concept + portability of OpenCL
- Enables “single-source” style standard C++ code for heterogeneous procs
- C++ template functions with both host and device code
- Open-source C++ 17 Parallel STL for SYCL and OpenCL 2.2 features
- Target devices: CPU, GPU, FPGA (any OpenCL supported device)

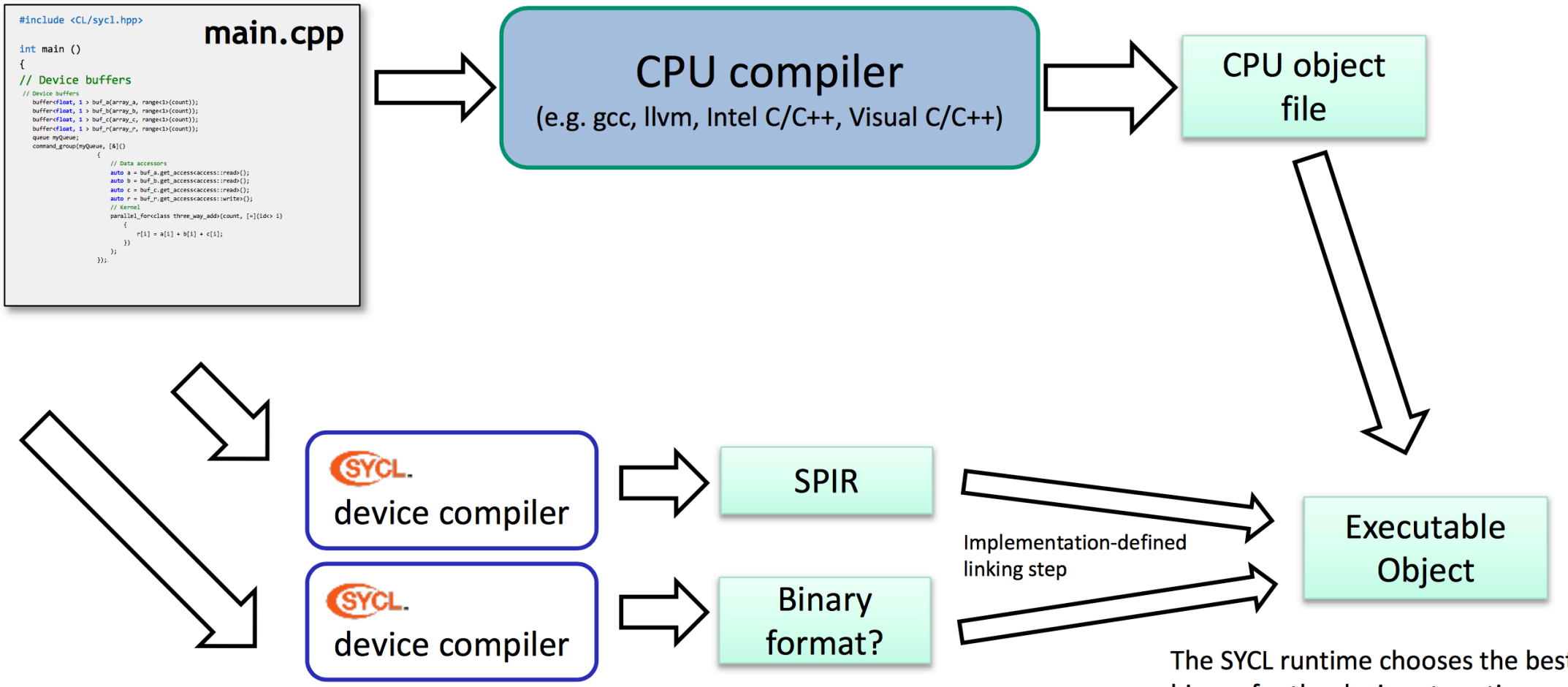
- Compiler:
  - ComputeCpp (Codeplay), free community + commercial version
  - triSYCL (Xilinx), open-source (github.com/Xilinx/triSYCL)

SYCL / OpenCL Stack



Source: Khronos Group, 2015

Code Generation Process with SYCL

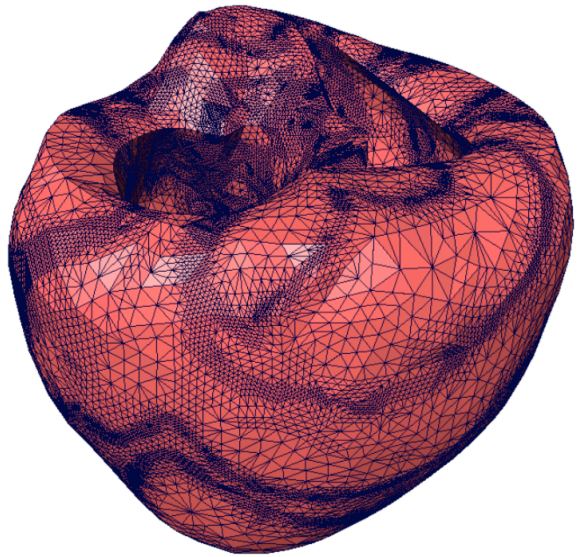


Source: Khronos Group, 2015

SYCL in Practice?

- TensorFlow for OpenCL using SYCL (Codeplay)
- Parallel STL (C++ extensions for parallelism)
  → Part of future C++ standard

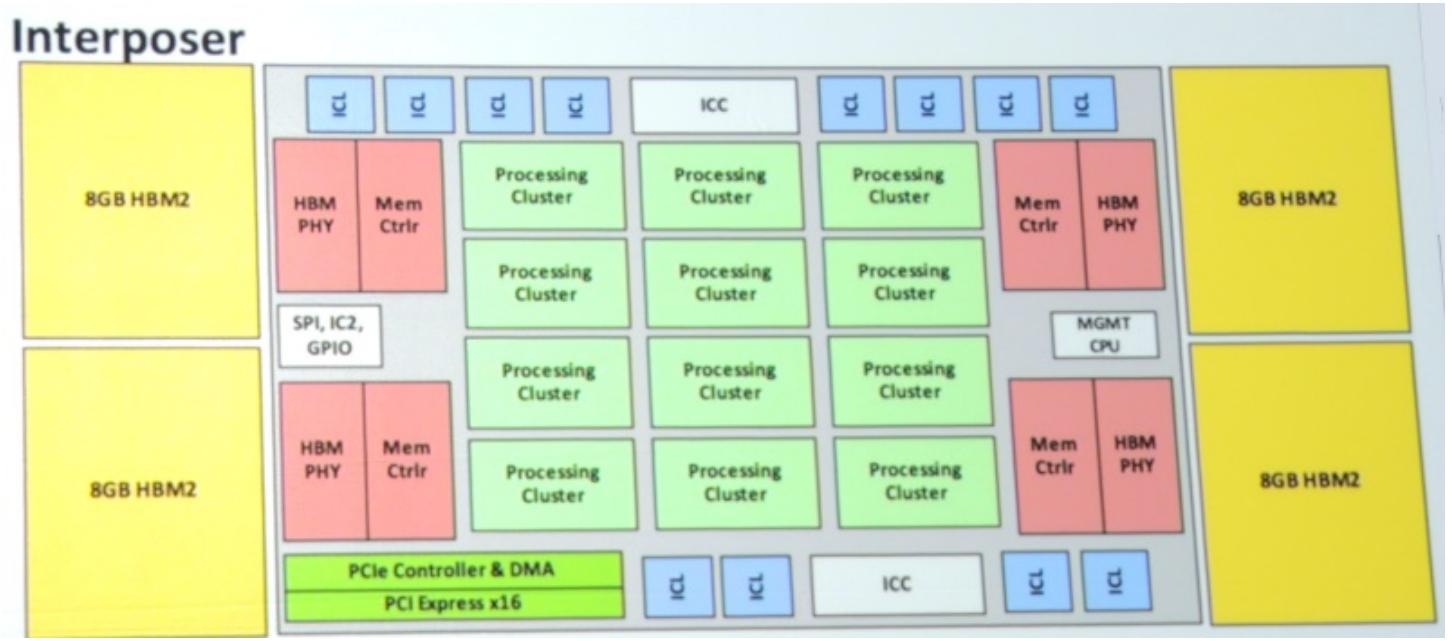
- BMBF Project HighPerMeshes (04/2017-03/2020)
  - ZIB + Paderborn + Erlangen + Fraunhofer ITWM
    (M. Weiser, Th. Steinke)



(Near) Future Devices for Deep Learning

Intel Lake Crest

- 1st gen Nervana Engine (2017)
- Hardware for DL training
- 32 GB HBM2
- 1 TByte/s mem access speed
- No caches
- Flexpoint arithmetic
  (between fixed and floating point)
- 12 bi-direct links for inter-chip communication (20x of PCIe)



Source: http://www.eetimes.com/document.asp?doc_id=1330854

Intel Knights Crest (ca. 2020)

- 2nd gen of Nervana Engine with Xeon cores
- Bootable = stand-alone CPU

IBM Resistive Cross-Point Devices (RPU)

Table 1. Summary of comparison of various RPU system designs and state-of-the-art CPU and GPU

System	Throughput (TeraOps/s)	Power (W)	Power Efficiency (G-Ops/s/W)	Network Size (Number of weights)	Acceleration vs CPU
CPU Power8 12 Cores	0.676	250	2.7	-	1
GPU NVidia Tesla K40	4.3	242	17.8	-	6.4
Design 1	5,000	250	20,100	200 M	7,400
Design 2	21,000	250	83,800	840 M	31,000
Design 3	420	22	19,000	1,680 M	620

Source: Tayfun Gokmen, Yurii Vlasov (IBM), Acceleration of Deep Neural Network Training with Resistive Cross-Point Devices

More Information

- Intel:
  - www.intel.com/ai
    - Intel AI Day 2017: http://www.inteldevconference.com/ai-day-munich-slides/
    - Intel AI Technical Workshop 2017: http://www.inteldevconference.com/intel-ai-workshop-slides

- Nvidia:
  - https://developer.nvidia.com/deep-learning-software

Acknowledgements

Axel Köhler, Nvidia

Thorsten Schmidt, Intel