Alice in Wonderland, the Red Queen's race, and microprocessor design in a world of Deep Learning

Tim Mattson
(timothy.g.mattson@intel.com)
Intel, Parallel Computing Lab

An Update from my last talk at ADMS

In 2015 I spoke at ADMS about many core CPUs and Graph Analytics

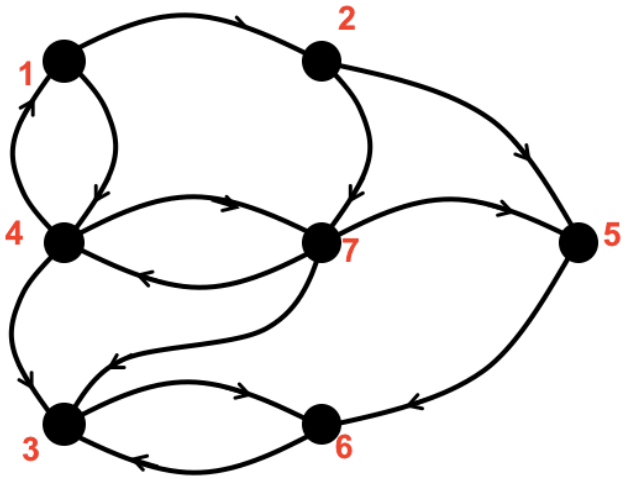
Graph Analytics with lots of CPU cores

Tim Mattson
(timothy.g.mattson@intel.com)
Intel, Parallel Computing Lab

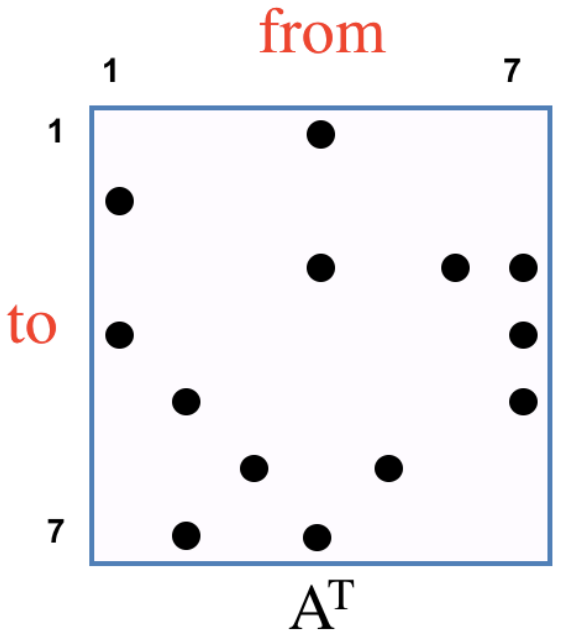
An Update from my last talk at ADMS

I spoke of Graphs in the language of Linear Algebra and the GraphBLAS project

Graphs in the Language of Linear Algebra

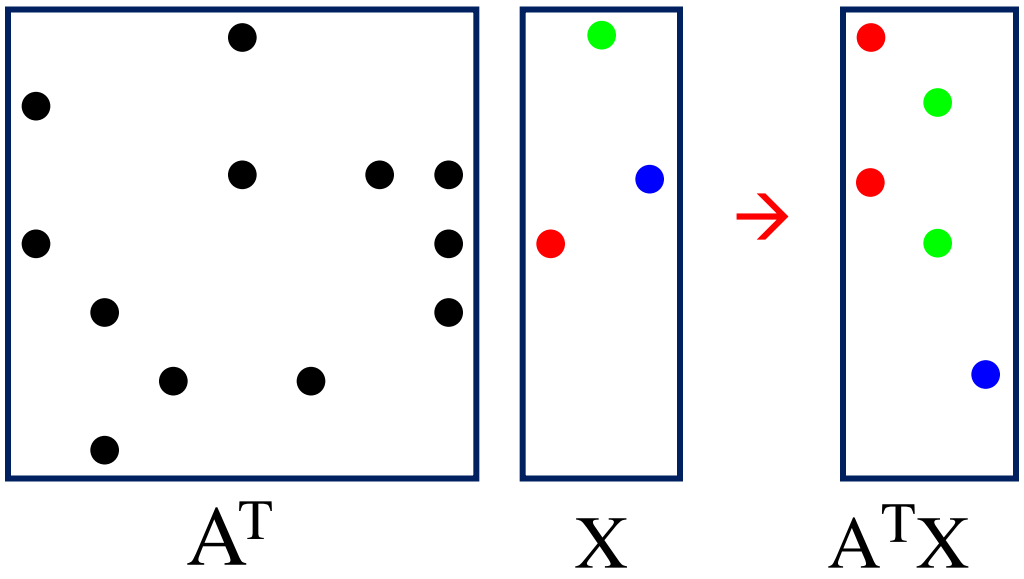


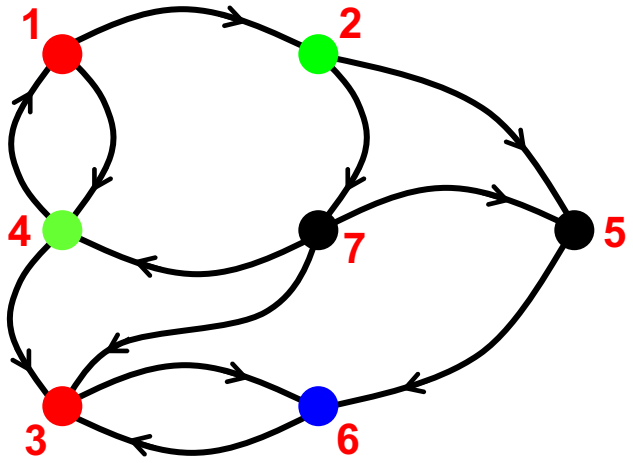
These two diagrams are equivalent representations of a graph.



A = the adjacency matrix … Elements nonzero when vertices are adjacent

The fundamental primitive of the GraphBLAS is SpGEMM: Example: Multiple-source breadth-first search





- Sparse array representation => space efficient
- Sparse matrix-matrix multiplication => work efficient
- Three possible levels of parallelism: searches, vertices, edges

Multiplication of sparse matrices captures Breadth first search and serves as the foundation of all algorithms based on BFS

An Update from my last talk at ADMS

We finished our specification … The GraphBLAS are alive!!!!!

Multiple Open Source implementations will be available soon (both GPU and CPU)

The GraphBLAS C Spec Lives! (released May 2017)

The GraphBLAS C API Specification†:

Provisional Release, Version 1.0.1

Aydın Buluç, Timothy Mattson, Scott McMillan, José Moreira, Carl Yang

†Based on *GraphBLAS Mathematics* by Jeremy Kepner

Spec available at: graphblas.org

Let's return to the planned talk

Alice in Wonderland, the Red Queen's race, and microprocessor design in a world of Deep Learning

Tim Mattson
(timothy.g.mattson@intel.com)
Intel, Parallel Computing Lab

Legal Disclaimer & Optimization Notice

- INFORMATION IN THIS DOCUMENT IS PROVIDED “AS IS”. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO THIS INFORMATION INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.
- 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.
- Copyright © , Intel Corporation. All rights reserved. Intel, the Intel logo, Xeon, Xeon Phi, Core, VTune, and Cilk are trademarks of Intel Corporation in the U.S. and other countries.

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

Disclaimer

- The views expressed in this talk are those of the speaker and not his employer.

- If I say something "smart" or worthwhile:
  - Credit goes to the many smart people I work with.

- If I say something stupid…
  - It's my own fault

I work in Intel's research labs. I don't build products. Instead, I get to poke into dark corners and think silly thoughts… just to make sure we don't miss any great ideas.

Hence, my views are by design far "off the roadmap".

Disclaimer

- The views expressed in this talk are those of the speaker and not his employer.

- If I say something "smart" or worthwhile:
  - Credit goes to the many smart people I work with.

- If I say something stupid…
  - It's my own fault

I work in Intel's research labs. I don't build products.
Instead, I get to poke into dark corners and think silly thoughts… just to make sure we don't mis

Hence, my views are by design far "off the roadmap".

Actually, for this talk, I'll be talking about Intel's roadmap … this is very scary for me

Acknowledgments

- I am presenting results from the following people:
  - Debbie Marr, Director Intel Accelerator lab
  - Eriko Nurvitadhi, Research Scientist, Intel Accelerator Lab
  - Nadathur Satish, Research Scientist, Intel Parallel Computing Lab
  - Andres Rodriguez, Sr. Principal Engineer, Intel AI products Group

- Others have contributed to my understanding of these issues:
  - David Sheffield, Research Scientist, Intel Accelerator Lab
  - Shekar Borkar, QUALCOMM
  - Michael Anderson, Research Scientist, Intel Parallel Computing Lab

Be careful believing anything a vendor says!

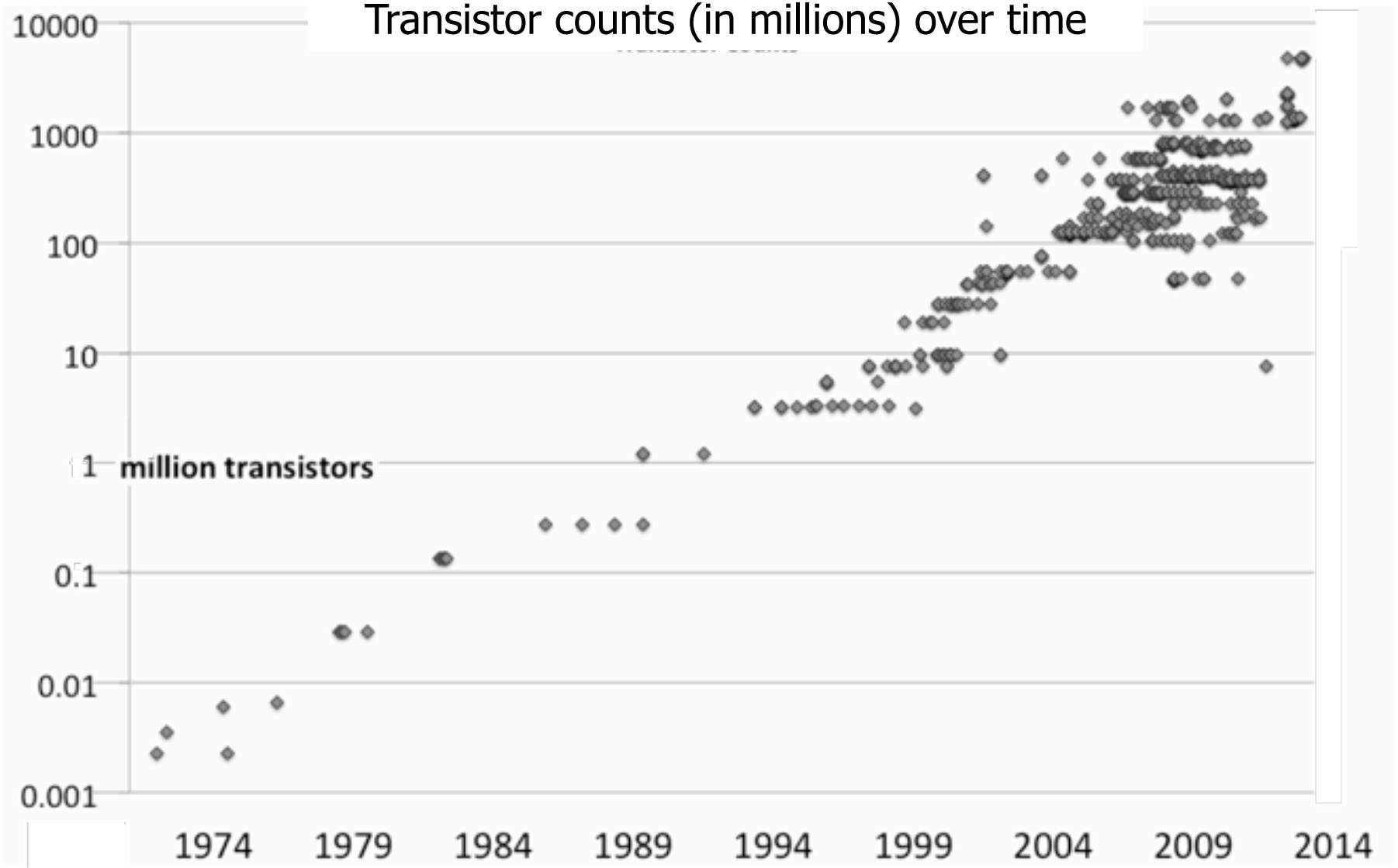
- Remember the immortal words of Upton Sinclair

It is difficult to get a man to understand something when his salary depends upon his not understanding it.

- I work for Intel … so be cautious with any hardware performance comparisons I might make.

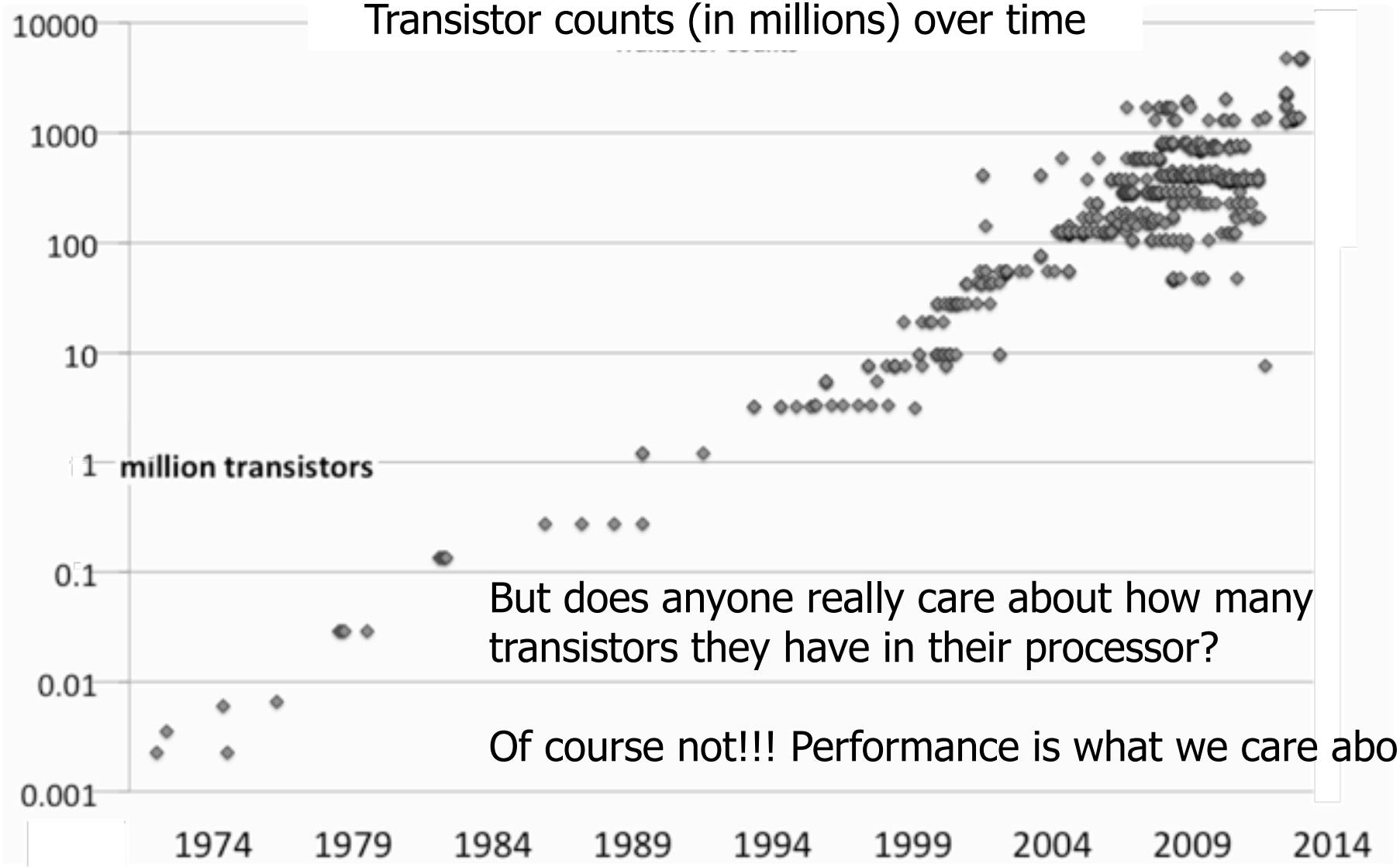
- And if I do make mistakes with these comparisons, they are never deliberate! I really try to be fair.

Moore's Law is alive and well



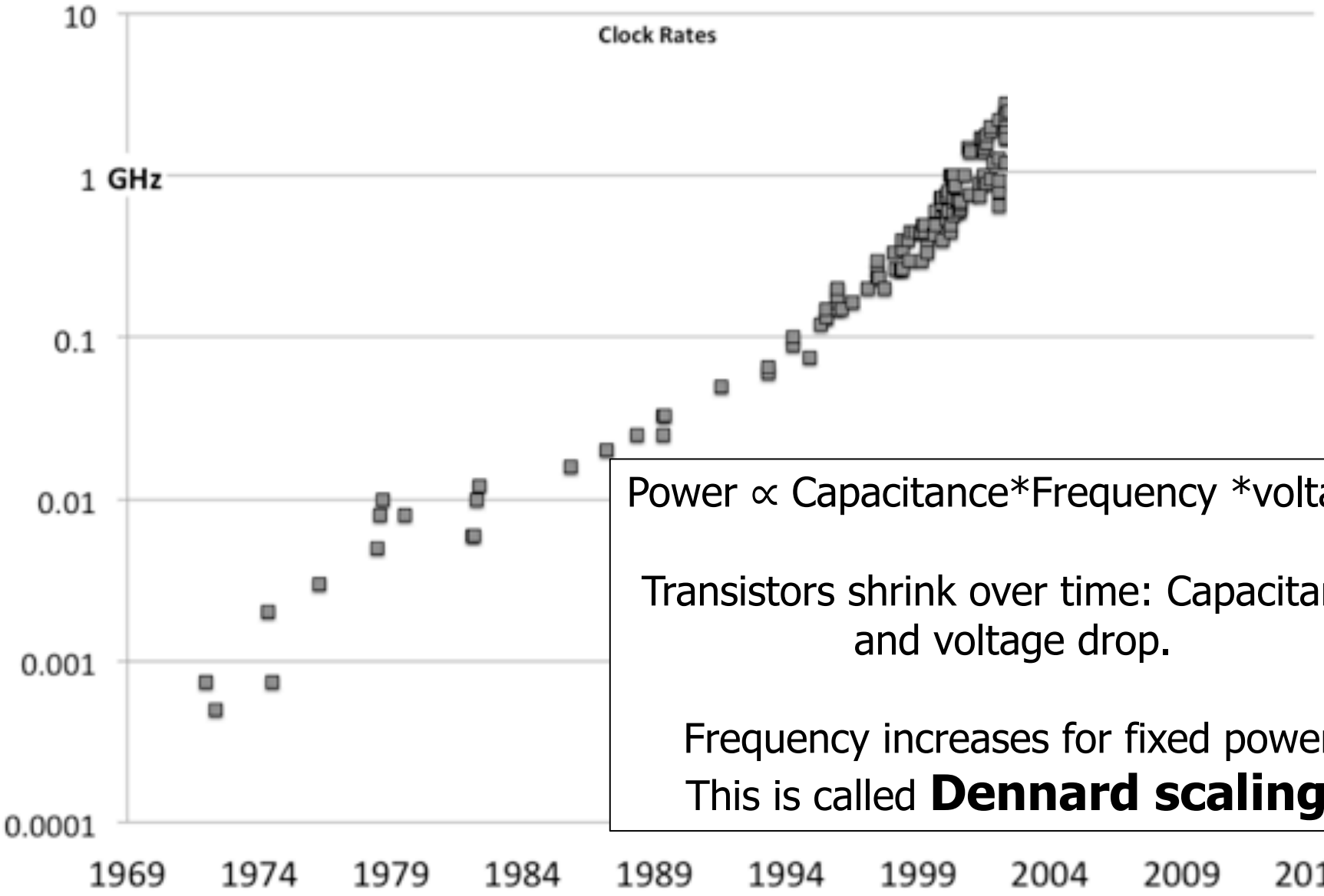
Source: James Reinders (from the book "structured parallel programming")

Moore's Law is alive and well



Source: James Reinders (from the book "structured parallel programming")

CPU Frequency (GHz) over time (years)



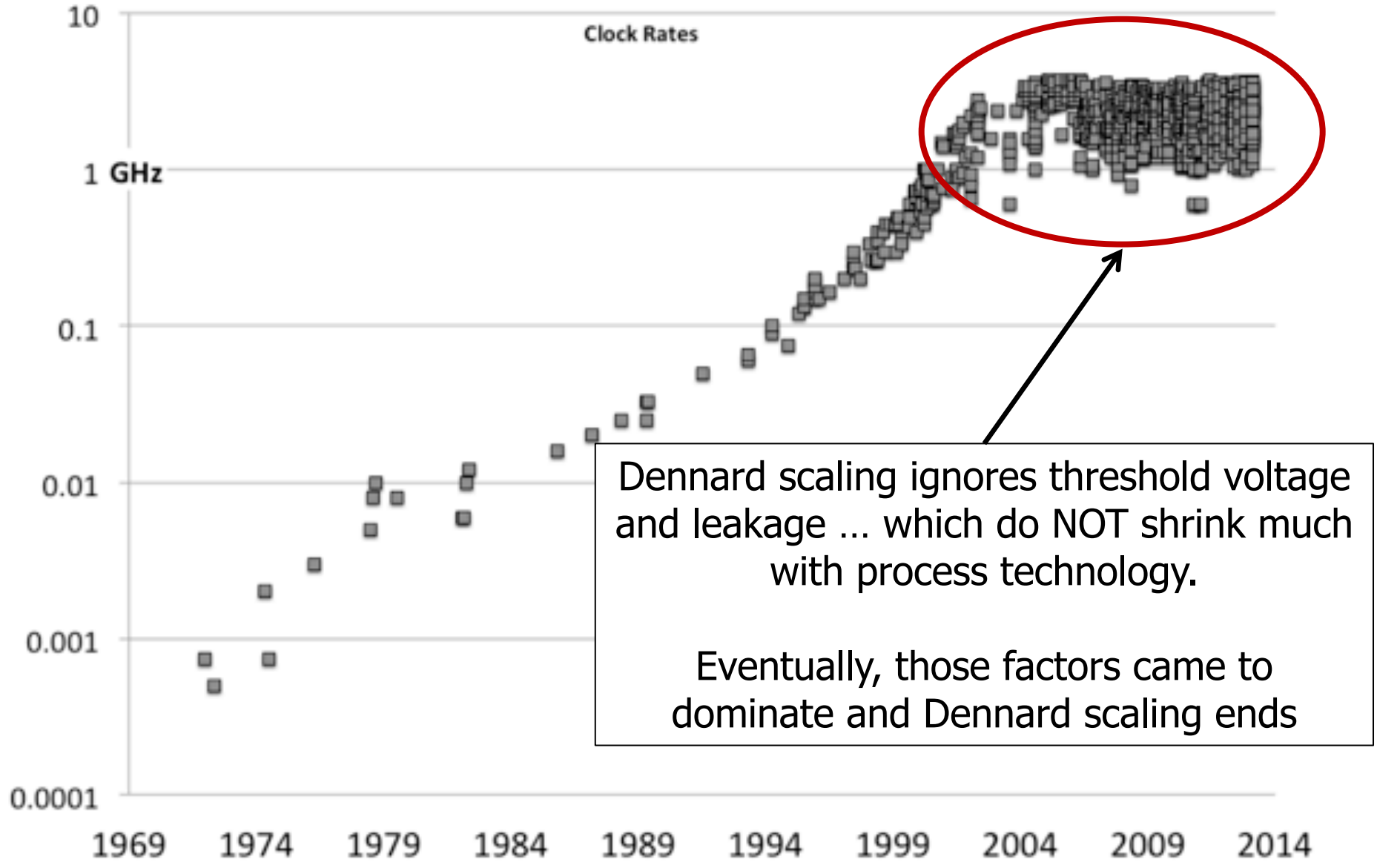
Power ∝ Capacitance*Frequency *voltage^2

Transistors shrink over time: Capacitance and voltage drop.

Frequency increases for fixed power.
This is called **Dennard scaling**.

Source: James Reinders (from the book "structured parallel programming")

CPU Frequency (GHz) over time (years)

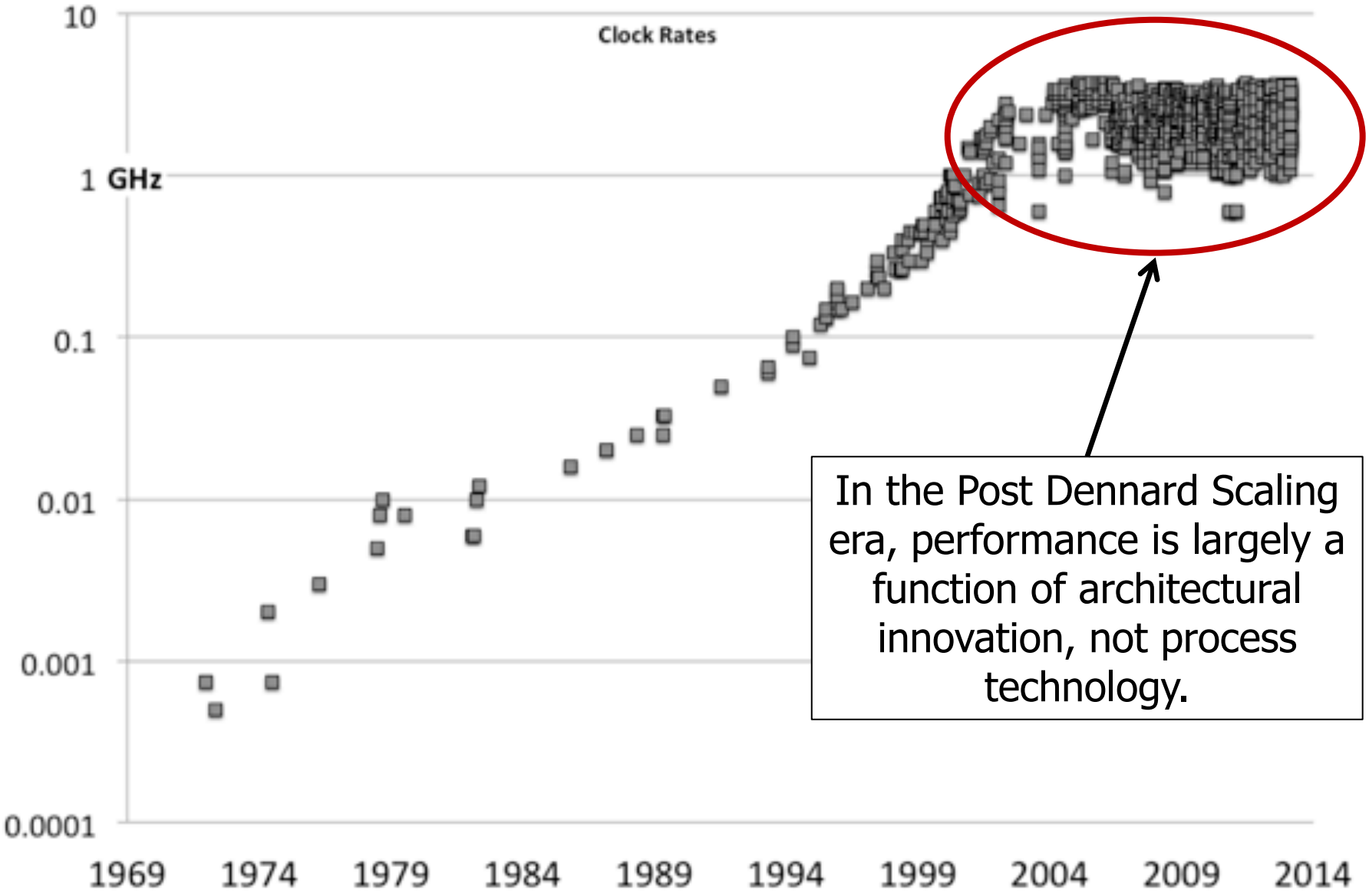


Dennard scaling ignores threshold voltage and leakage … which do NOT shrink much with process technology.

Eventually, those factors came to dominate and Dennard scaling ends

Source: James Reinders (from the book "structured parallel programming")

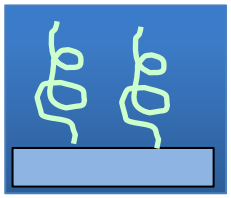
CPU Frequency (GHz) over time (years)



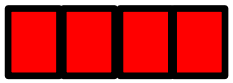
Source: James Reinders (from the book "structured parallel programming")

Hardware Diversity:
Basic Building Blocks

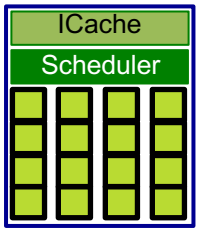
Commercial Off the Shelf (COTS) Response to Dennard Scaling's end



CPU Core: one or more hardware threads sharing an address space. Optimized for low latencies.



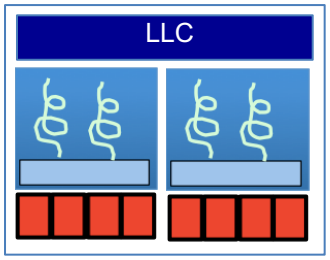
SIMD: Single Instruction Multiple Data.
Vector registers/instructions with 128 to 512 bits (or more) so a single stream of instructions drives multiple data elements.



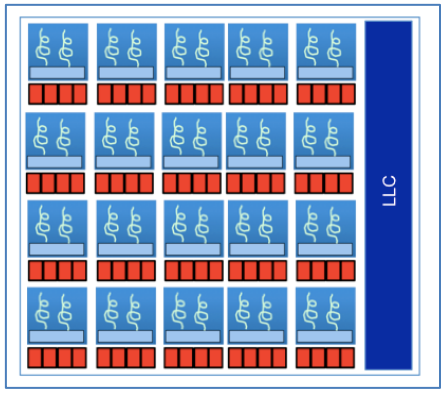
SIMT: Single Instruction Multiple Threads.
A single stream of instructions drives many threads. More threads than functional units. Over subscription to hide latencies. Optimized for throughput.

Hardware Diversity:
Combining building blocks to construct nodes

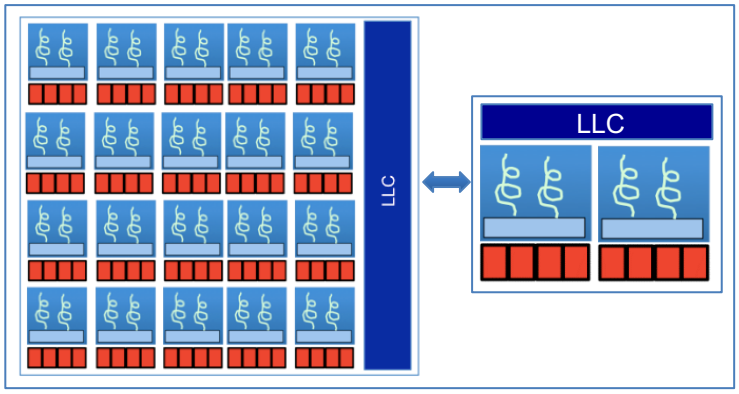
Commercial Off the Shelf (COTS) Response to Dennard Scaling's end



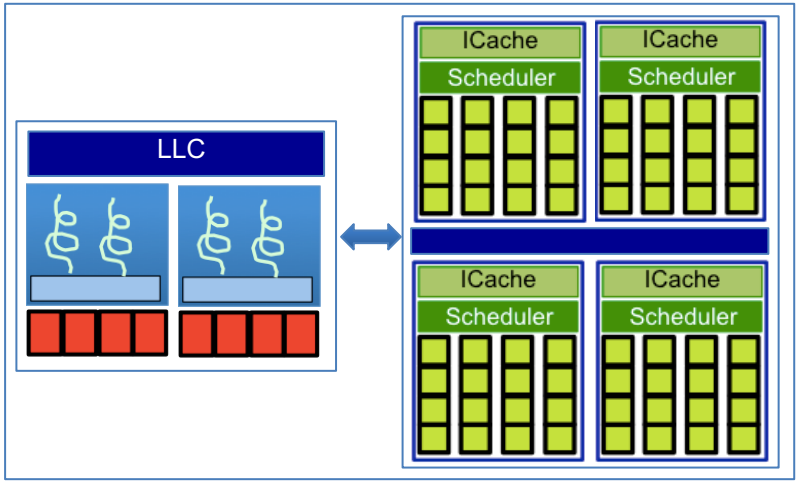
Multicore CPU



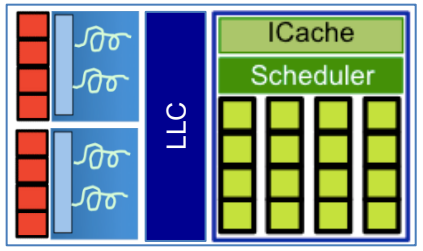
Manycore CPU



Heterogeneous:
CPU+manycore CPU

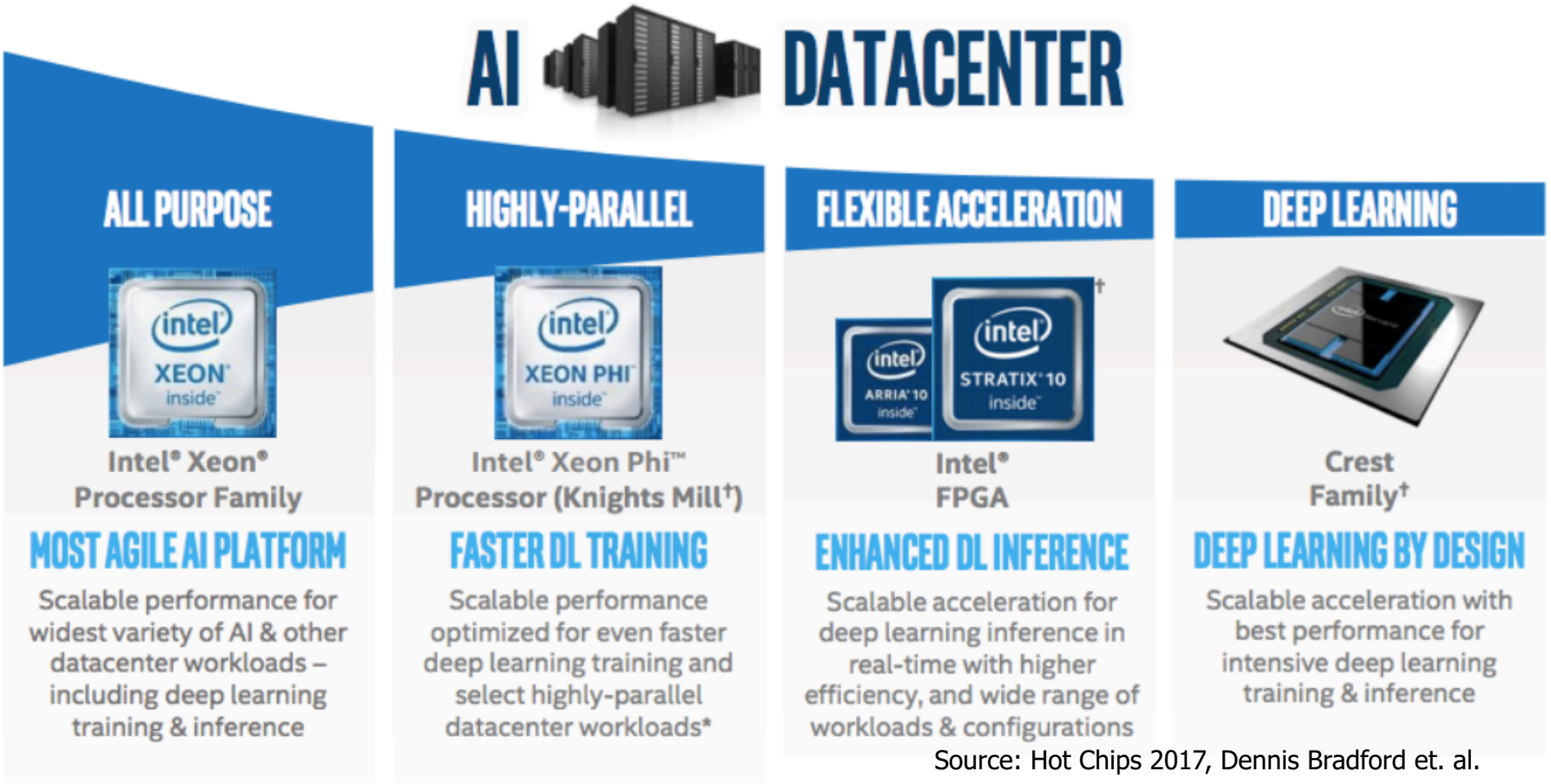


Heterogeneous: CPU+GPU



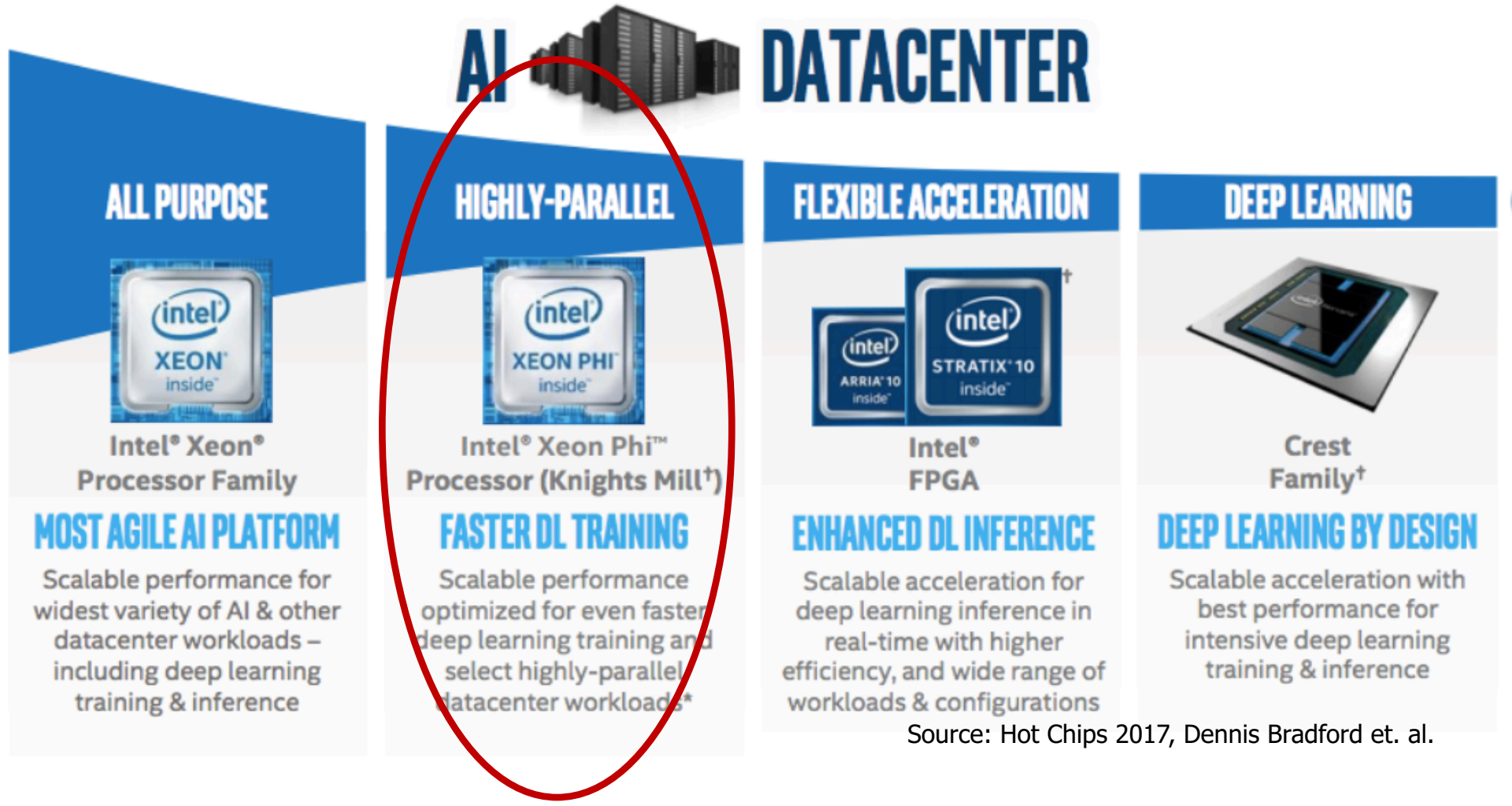
Heterogeneous:
Integrated CPU+GPU

Let's look at Intel Processors with an eye towards deep learning



Source: Hot Chips 2017, Dennis Bradford et. al.

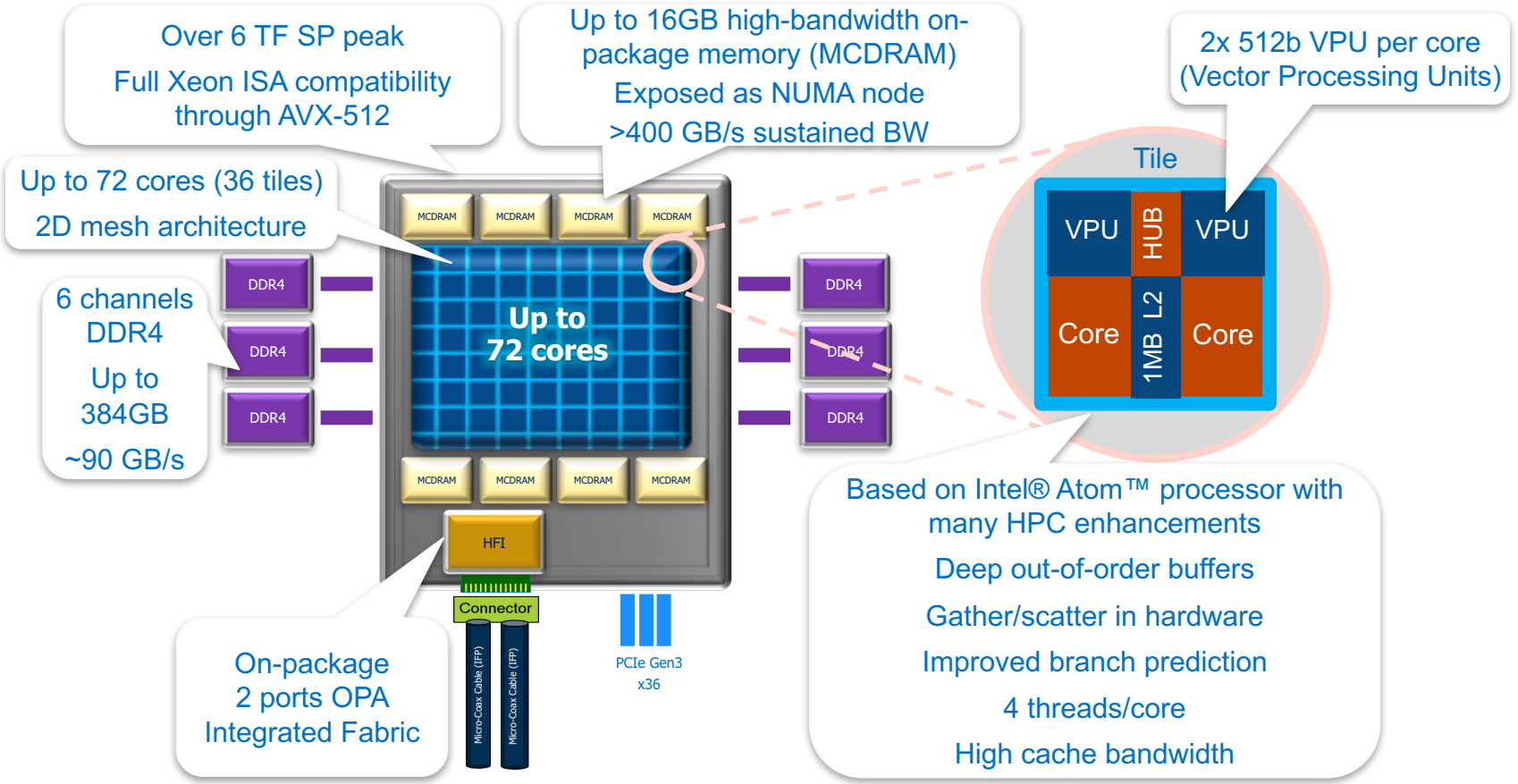
[+]Codename for product that is coming soon
All Performance positioning claims are relative to other processor technologies in Intel's AI datacenter Portfolio
*Knights Mill (KNM); select = single-precision highly-parallel workloads generally scale to >100 threads and benefit from more vectorization, and may benefit from greater memory bandwidth e.g. energy (reverse time migration), deep learning training, etc.
All products, computer systems, dates, and figures specified are preliminary based on current expectation, and are subject to change without notice.



Source: Hot Chips 2017, Dennis Bradford et. al.

†Codename for product that is coming soon
All Performance positioning claims are relative to other processor technologies in Intel's AI datacenter Portfolio
*Knights Mill (KNM); select = single-precision highly-parallel workloads generally scale to >100 threads and benefit from more vectorization, and may benefit from greater memory bandwidth e.g. energy (reverse time migration), deep learning training, etc.
All products, computer systems, dates, and figures specified are preliminary based on current expectation, and are subject to change without notice.

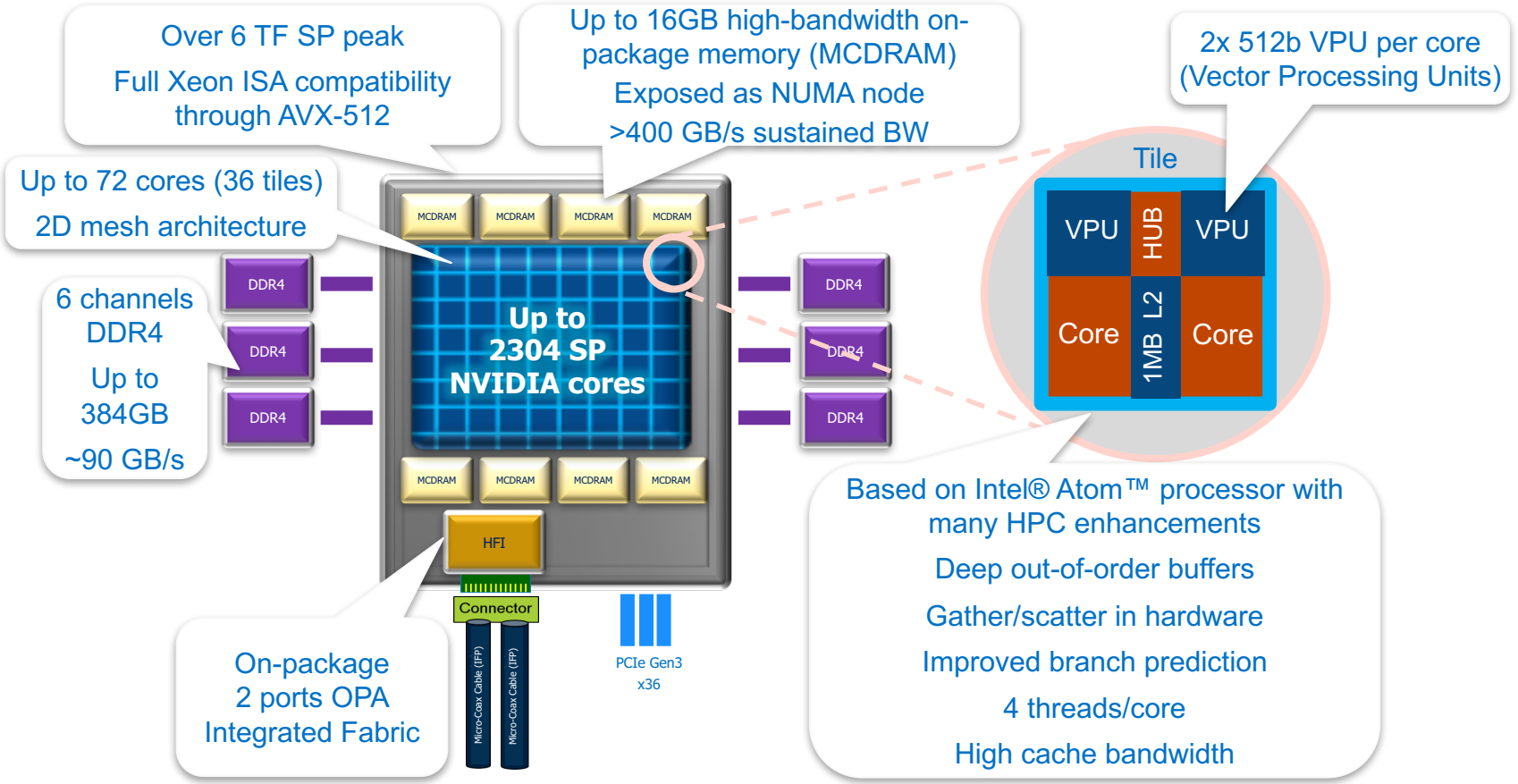
Key Intel® Xeon Phi™ Architecture Features



A "small simple core" solution … low-power cores used to keep their large vector units busy

Diagram is for conceptual purposes only and only illustrates a CPU and memory – it is not to scale and does not include all functional areas of the CPU, nor does it represent actual component layout.

Key Intel® Xeon Phi™ Architecture Features



A "small simple core" solution … low-power cores used to keep their large vector units busy

Diagram is for conceptual purposes only and only illustrates a CPU and memory – it is not to scale and does not include all functional areas of the CPU, nor does it represent actual component layout.

CNN Image Analysis Results for Intel® Xeon Phi™ 7250 CPU @ 1.2 GHz

After cache blocking, register blocking, restructuring data (--> SOA), vectorization and multithreading (details in backup slides) ….

Single Core, minibatch size: 256, inner convolutions

Topology	IFM	OFM	OFH	OFW	KH	KW	GFLOPS	%peak
OverFeat	96	256	24	24	5	5	60.0	78.13
OverFeat	256	512	12	12	3	3	60.8	79.19
OverFeat	512	1024	12	12	3	3	61.0	79.43
OverFeat	1024	1024	12	12	3	3	60.9	79.30

High single core efficiency

64 Cores, minibatch size: 256, inner convolutions

Topology	IFM	OFM	OFH	OFW	KH	KW	GFLOPS	%peak
OverFeat	96	256	24	24	5	5	3714.0	73.27
OverFeat	256	512	12	12	3	3	3685.0	72.70
OverFeat	512	1024	12	12	3	3	3625.0	71.71
OverFeat	1024	1024	12	12	3	3	3550.0	70.03

Maintained at multi-core => good scaling

IFM, OFM, OFH, OFW, KH, and KW define problem sizes. See backup slides for details

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 information go to http://www.intel.com/performance . *Other names and brands may be property of others
Configurations: Intel measured performance of convolutions in Overfeat* network
- Intel® Xeon Phi™ Processor 7250 (68 Cores, 1.4+ GHz, 16GB MCDRAM), 96 GB memory, Red Hat* Enterprise Linux 6.7, Intel® Optimized Caffe framework, Intel® C++ compiler version 16.0.2, Intel® MKL 11.3 Release 2

All numbers measured without taking data manipulation into account.

Training Deep Learning networks at multi petaflops

DOE NERSC Cray XC40 Supercomputer

9688 Intel® Xeon Phi™ 7250(KNL) processors with up to 50 SP FLOPS

A team of Intel and NERSC scientist trained DNN at sustained rate of 13 petaflops working with a 15 TB metrological data set and 11 PetaFLOPS with a high energy physics problem. They claim this is the first time such high levels of performance has been achieved for deep learning in the sciences

Source: https://www.top500.org/news/deep-learning-hits-15-petaflops-on-cori-supercomputer/

Intel® Xeon Phi™ Processor (Knights Mill)

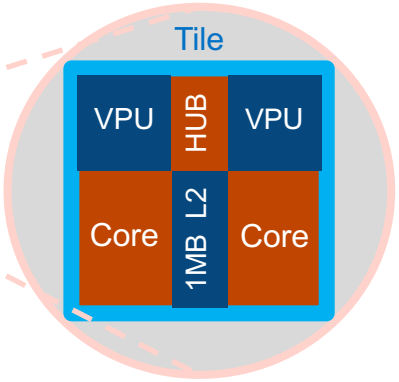
The latest addition to the Intel® Xeon Phi™ family … our first chip optimized specifically for the deep learning market

6 DDR4 2400 Channels

16 GP MCDRAM

36 Lanes PCIE Gen 3

36 tiles



1 MB L2/tile
16 DP FLOPS/VPU/cycle
128 SP flops/VPU/cycle
256 VP-ops/VPU/cycle

Key new instructions

QFMA	Quadruple packed single-precisions multiply-add
VNNI	Multiply16 bit ints and accumulate into 32 bit destination
QVNNI	QFMA + VNNI

DP = Double Precision, SP = Single Precisions, VP = variable precision (16/32)

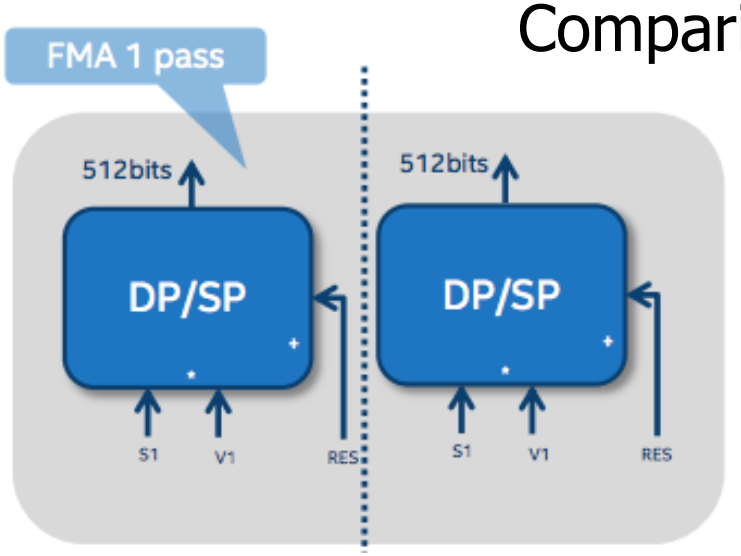
Source: Hot Chips 2017, Dennis Bradford et. al.

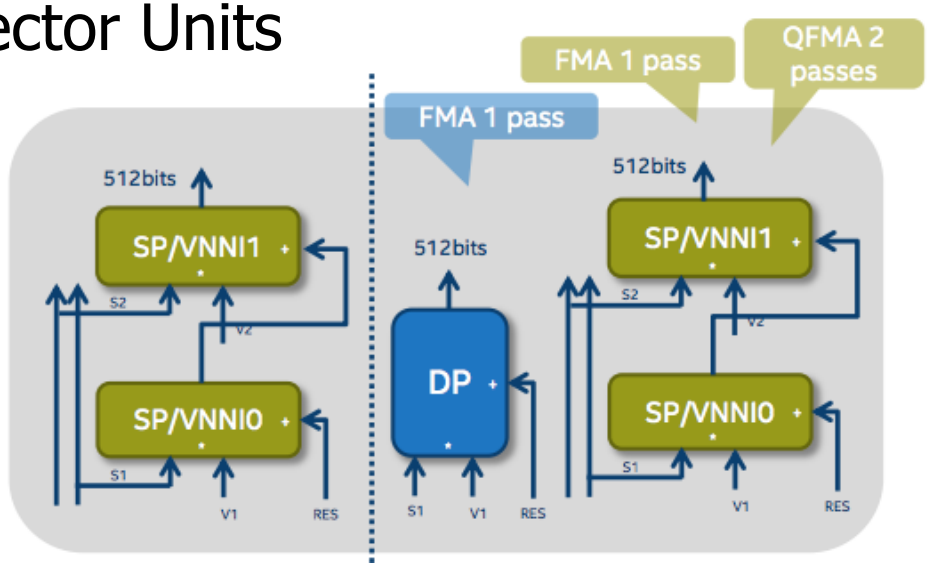
VNNI: Vector Neural Network Instructions

Intel® Xeon Phi™ processors

- Knights Mill vs. Knights Landing (Intel® Xeon Phi™ 7200 series):
  - They are the same generation Intel® Xeon Phi™ processor and share a great deal … but there are some key differences.

Comparing Vector Units





- Knights Landing
  - Optimized for HPC
    - 2 DP stacks

0.5x DP
2x SP
4x VNNI

- Knights Mill
  - Optimized for Deep Learning Training
    - 1 DP Stack
    - 2 SM/VNNI stack

Source: Hot Chips 2017, Dennis Bradford et. al.

AI DATACENTER

ALL PURPOSE

Intel® Xeon® Processor Family

MOST AGILE AI PLATFORM

Scalable performance for widest variety of AI & other datacenter workloads – including deep learning training & inference

HIGHLY-PARALLEL

Intel® Xeon Phi™ Processor (Knights Mill†)

FASTER DL TRAINING

Scalable performance optimized for even faster deep learning training and select highly-parallel datacenter workloads*

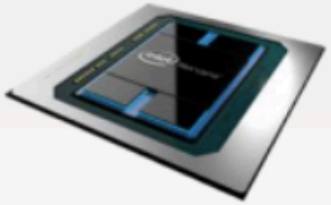
FLEXIBLE ACCELERATION

Intel® FPGA

ENHANCED DL INFERENCE

Scalable acceleration for deep learning inference in real-time with higher efficiency, and wide range of workloads & configurations

DEEP LEARNING



Crest Family†

DEEP LEARNING BY DESIGN

Scalable acceleration with best performance for intensive deep learning training & inference

Source: Hot Chips 2017, Dennis Bradford et. al.

†Codename for product that is coming soon
All Performance positioning claims are relative to other processor technologies in Intel's AI datacenter Portfolio
*Knights Mill (KNM); select = single-precision highly-parallel workloads generally scale to >100 threads and benefit from more vectorization, and may benefit from greater memory bandwidth e.g. energy (reverse time migration), deep learning training, etc.
All products, computer systems, dates, and figures specified are preliminary based on current expectation, and are subject to change without notice.

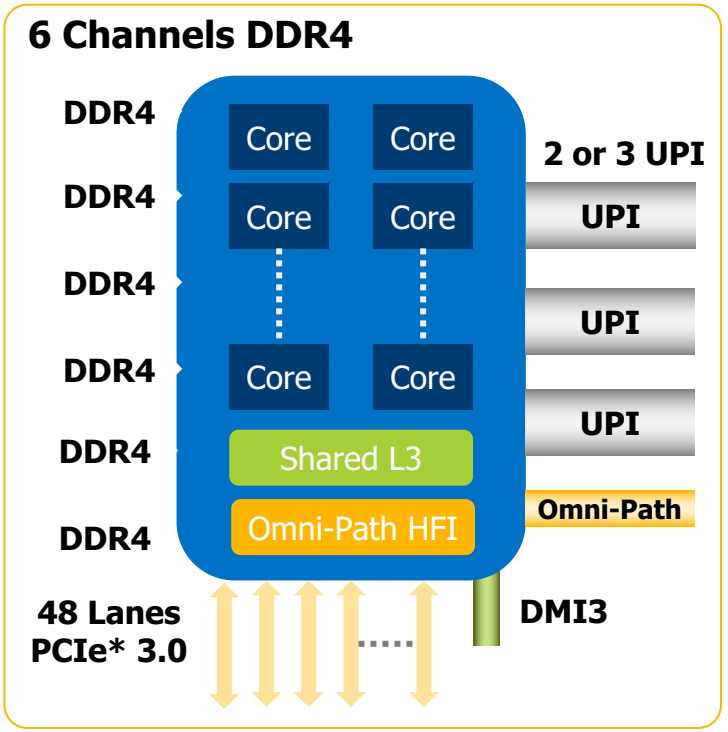
Intel® Xeon® Scalable Processors

Re-architected from the ground up

A ***Large*** core solution

- Intel® AVX-512 with 32 DP flops per core
- Data center optimized cache hierarchy – 1MB L2 per core, non-inclusive L3
- New Intel® Mesh interconnect architecture
- Enhanced memory subsystem
- Modular IO with integrated devices
- New Intel® Ultra Path Interconnect (Intel® UPI)
- Optional Integrated Intel® Omni-Path Fabric (Intel® OPA)

Features	Intel® Xeon® Scalable Processor
Cores Per Socket	Up to 28
Threads Per Socket	Up to 56 threads
Last-level Cache (LLC)	Up to 38.5 MB (non-inclusive)
QPI/UPI Speed (GT/s)	Up to 3x UPI @ 10.4 GT/s
PCIe* Lanes/ Controllers/Speed(GT/s)	48 / 12 / PCIe 3.0 (2.5, 5, 8 GT/s)
Memory Population	6 channels of up to 2 RDIMMs, LRDIMMs, or 3DS LRDIMMs
Max Memory Speed	Up to 2666 Hz
TDP (W)	70W-205W



Intel® Advanced Vector Extensions 512 (Intel® AVX-512)

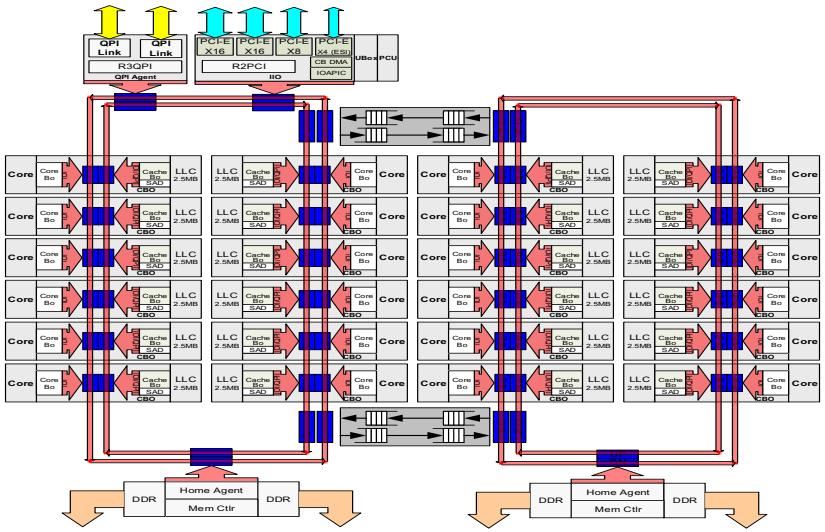
- 512-bit wide vectors
- 32 operand registers
- 8 64b mask registers
- Embedded broadcast
- Embedded rounding

Microarchitecture	Instruction Set	SP FLOPs / cycle	DP FLOPs / cycle
Skylake	Intel® AVX-512 & FMA	64	32
Haswell / Broadwell	Intel AVX2 (256b) & FMA	32	16
Sandybridge	Intel AVX (256b)	16	8
Nehalem	SSE (128b)	8	4

Intel AVX-512 Instruction Types	
AVX-512-F	AVX-512 Foundation Instructions
AVX-512-VL	Vector Length Orthogonality : ability to operate on sub-512 vector sizes
AVX-512-BW	512-bit Byte/Word support
AVX-512-DQ	Additional D/Q/SP/DP instructions (converts, transcendental support, etc.)
AVX-512-CD	Conflict Detect : used in vectorizing loops with potential address conflicts

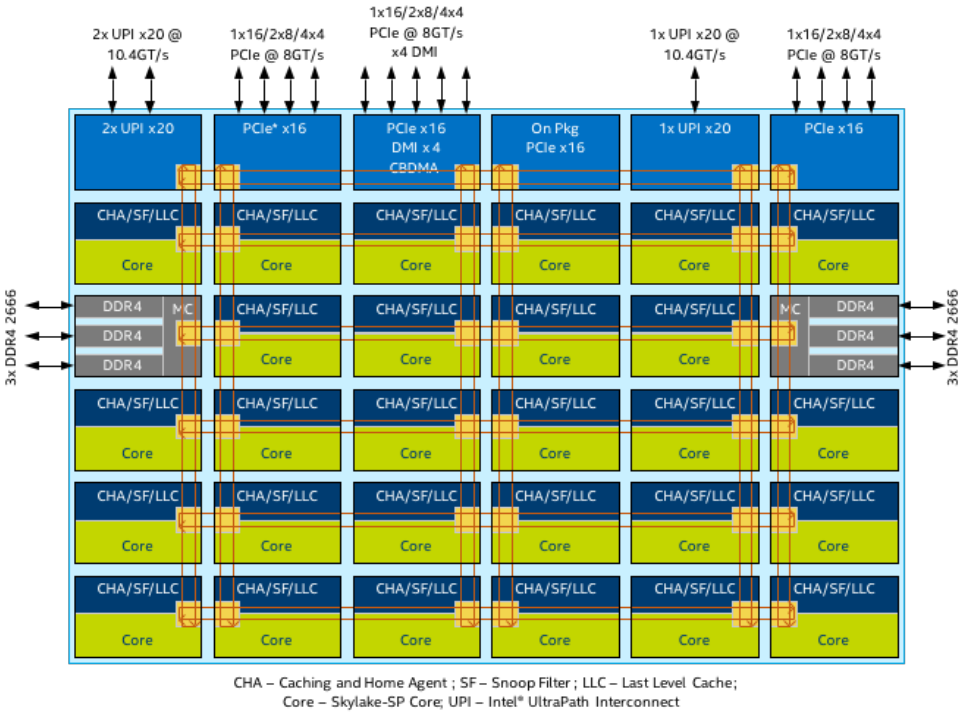
New Intel® Mesh Interconnect Architecture

2016: Intel® Xeon® processor E7 v4, 14nm (Broadwell EX 24-core die)



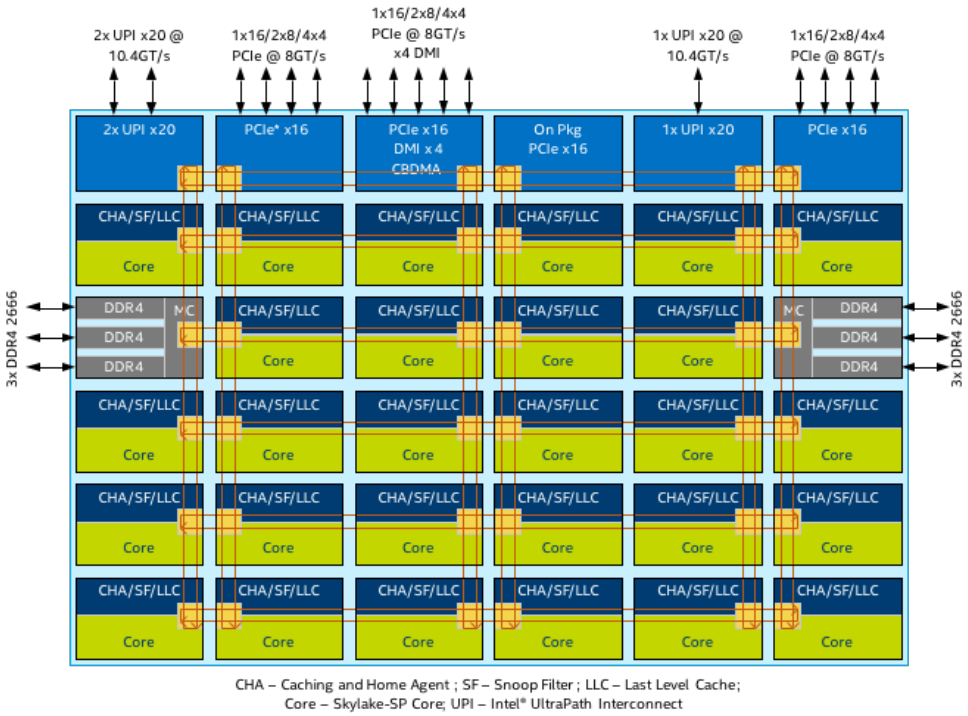
Pair of rings
Network diameter O(N/2)

2017: Intel® Xeon® Scalable Processor, 14nm (Skylake-SP 28-core die)

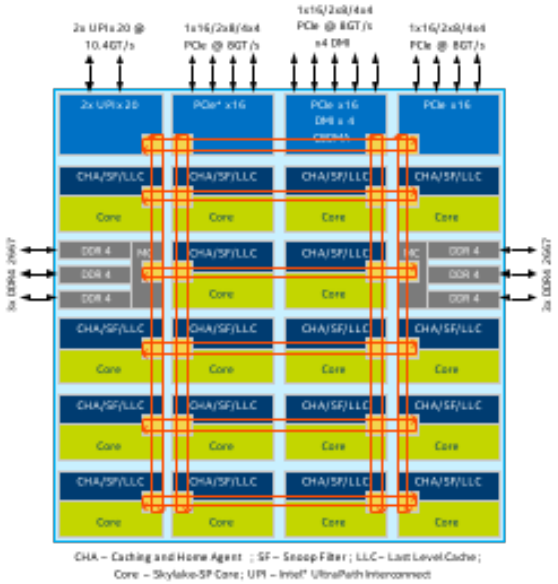


Grid
Network diameter O($N^{1/2}$)

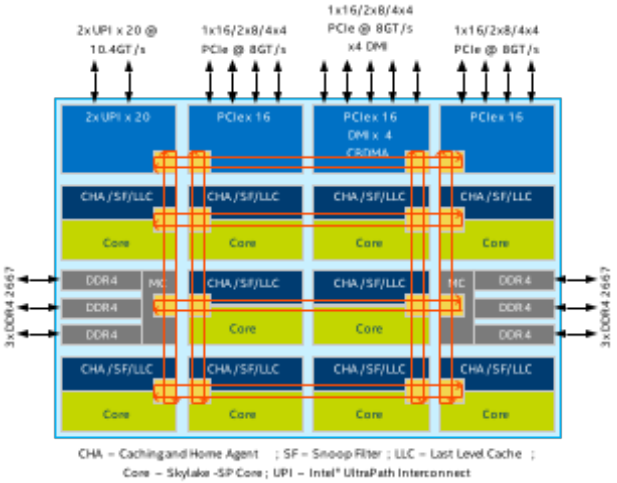
Skylake processor configurations



XCC die with 28 cores



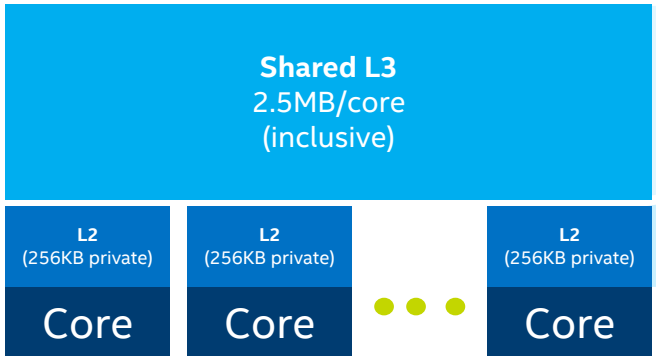
HCC die with up to 18 cores



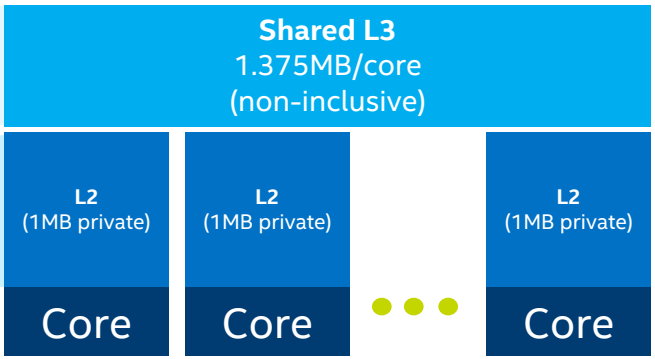
LCC die with up to 10 cores

Re-Architected L2 & L3 Cache Hierarchy

Previous Architectures



Skylake-SP Architecture



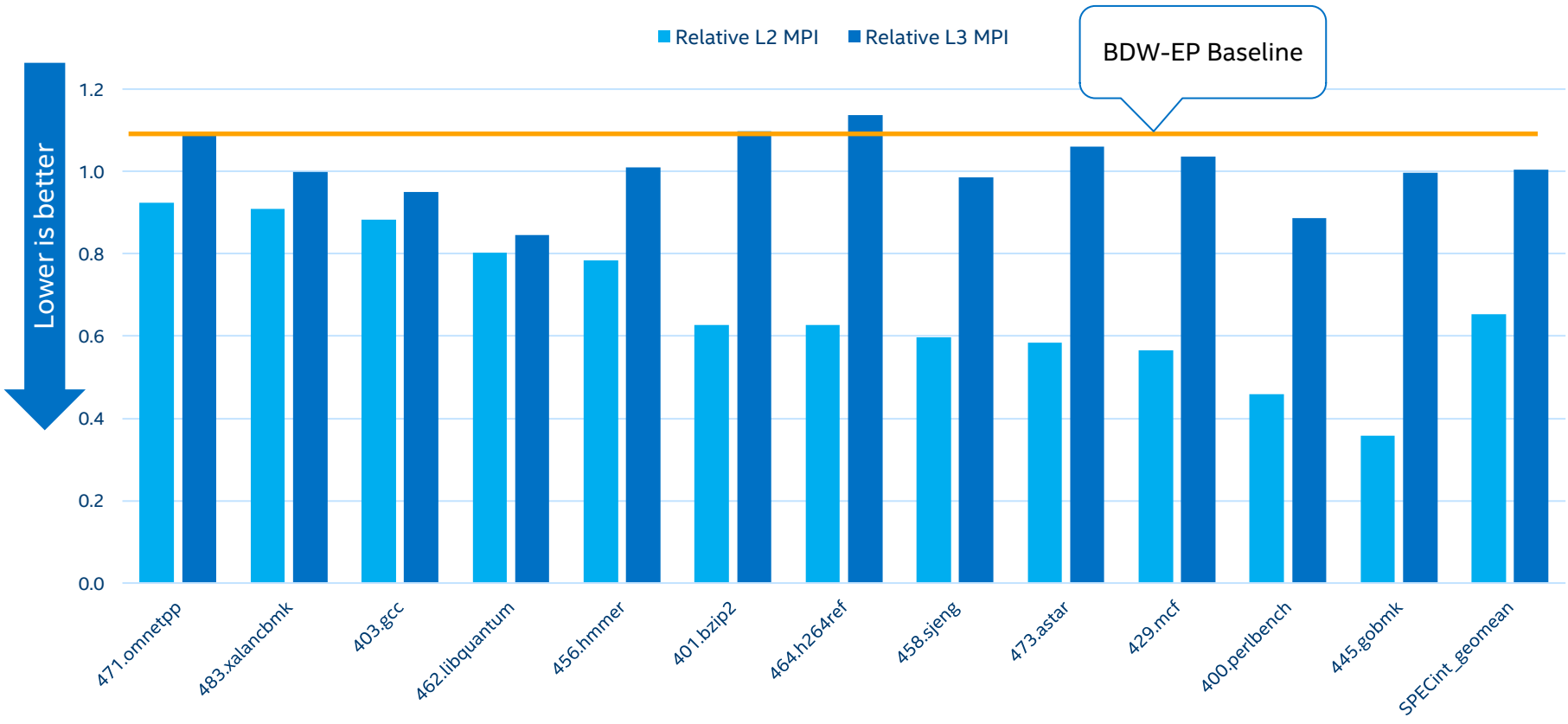
Shared-distributed
L3 is primary cache
Inclusive L3
- L3 has copies of all lines in L2

Private-local
private L2 is primary cache
shared L3 used as overflow cache
non-inclusive L3
- lines in L2 ***may not*** exist in L3

Cache Performance

Skylake-SP cache hierarchy significantly reduces L2 misses without increasing L3 misses compared to Broadwell-EP

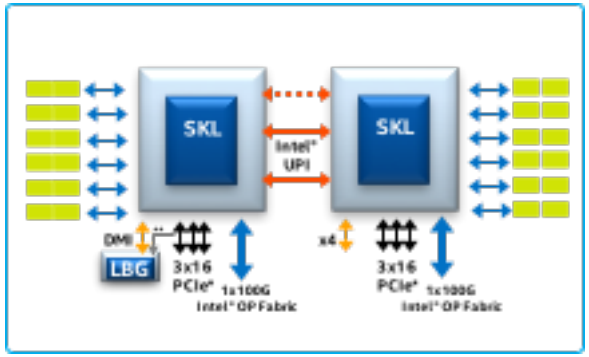
Relative Change in L2 and L3 Misses Per Instruction for SPECint*_rate 2006 from Broadwell-EP to Skylake-SP



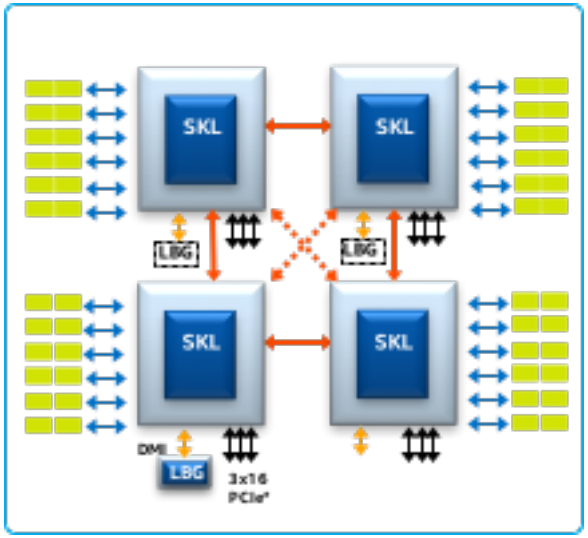
Source as of June 2017: Intel internal measurements on platform with Xeon Platinum 8180, Turbo enabled, UPI=10.4, SNC1, 6x32GB DDR4-2666 per CPU, 1 DPC, and platform with E5-2699 v4, Turbo enabled, 4x32GB DDR4-2400, RHEL 7.0. Software and workloads used in performance tests may have been optimized for 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. Copyright © 2017 Intel Corporation.

Skylake Platform Configurations

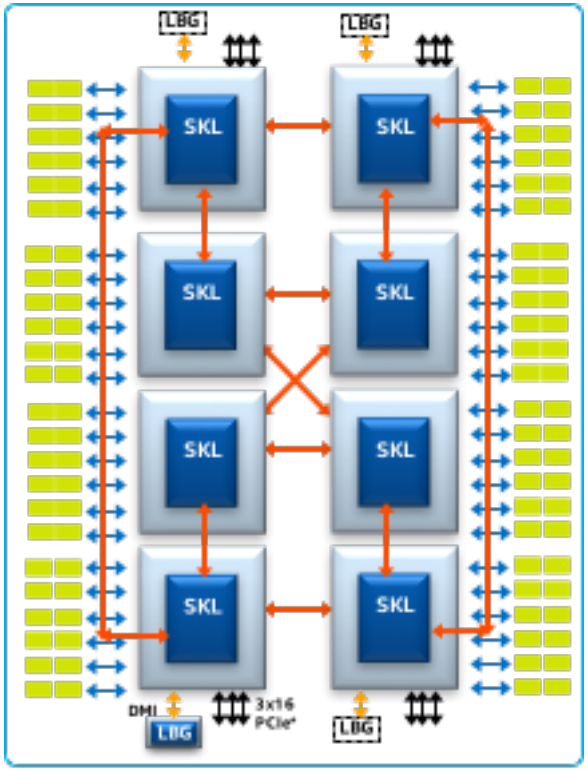
2S Configurations



4S Configurations

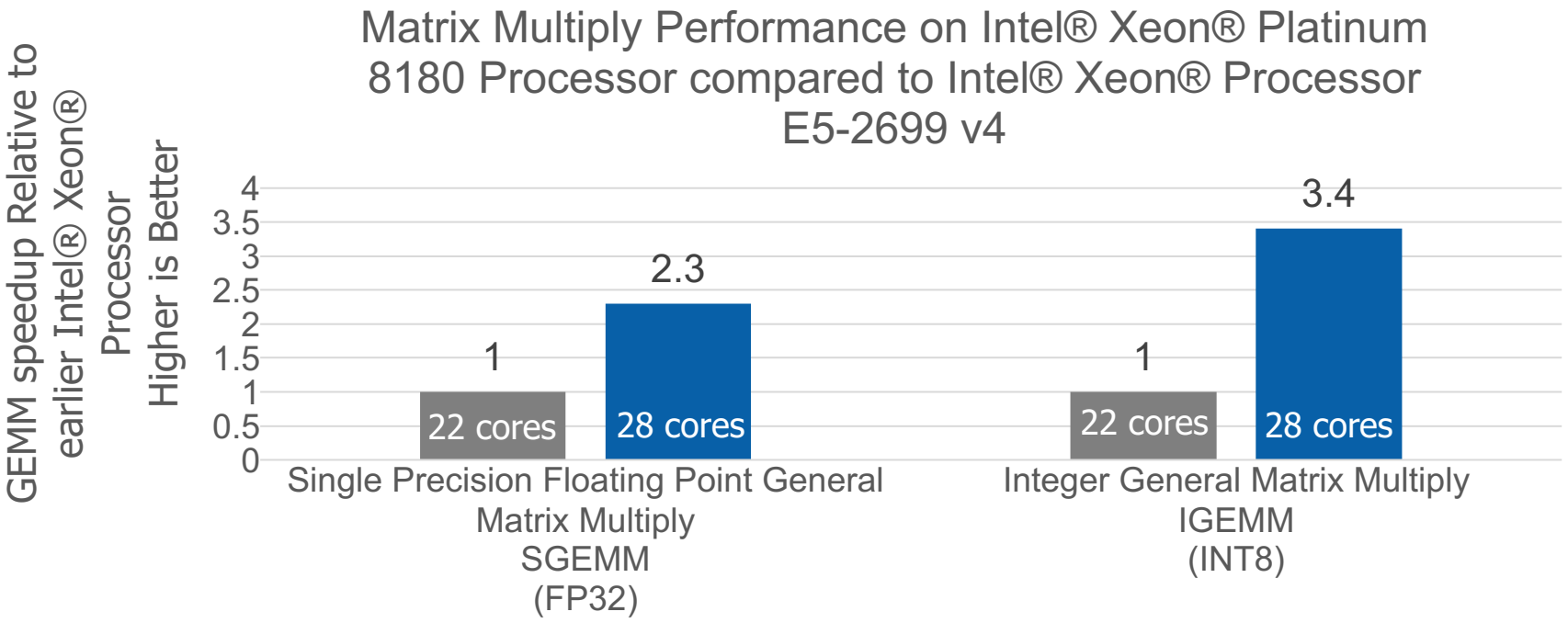


8S Configuration



Up to 3.4x Integer Matrix Multiply Performance

on Intel® Xeon® Platinum 8180 Processor



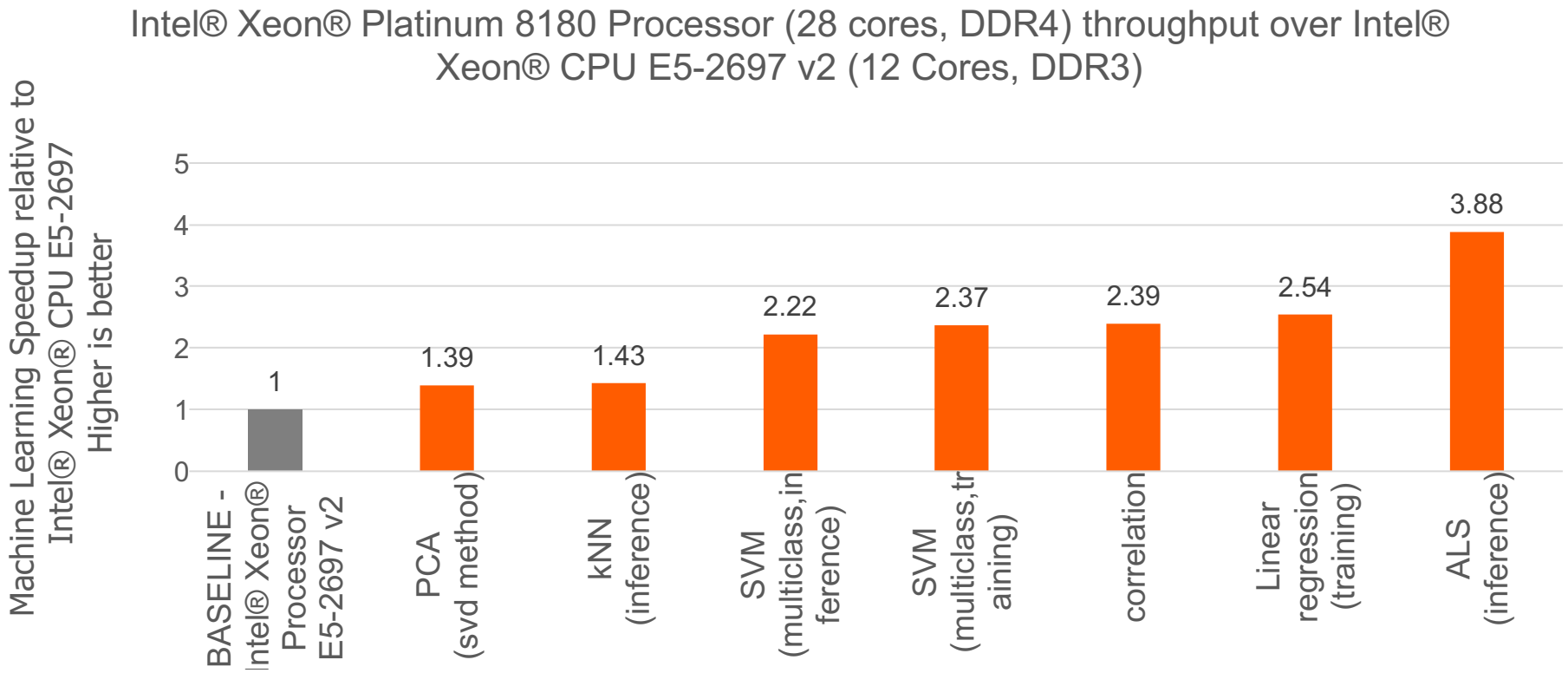
8bit IGEMM will be available in Intel® Math Kernel Library (Intel® MKL) 2018 Gold to be released by end of Q3 2017

Enhanced matrix multiply performance on Intel® Xeon® Scalable Processors

Configuration Details on Slide: 24

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.

Up to 3.8x Higher Throughput Performance
on Intel® Xeon® Platinum 8180 Processor



Intel® Xeon® Processor family delivers high throughput performance on wide range of Machine Learning algorithms

PCA: principle Component Analysis, kNN: k nearest neighbors, SVM: Support vector machine. ALS: Alternating least squares

Configuration Details on slide 31

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.

AI Performance – Gen over Gen

INFERENCE THROUGHPUT

Up to
2.4x

Intel® Xeon® Platinum 8180 Processor
higher Neon ResNet 18 inference throughput compared to
Intel® Xeon® Processor E5-2699 v4

TRAINING THROUGHPUT

Up to
2.2x

Intel® Xeon® Platinum 8180 Processor
higher Neon ResNet 18 training throughput compared to
Intel® Xeon® Processor E5-2699 v4

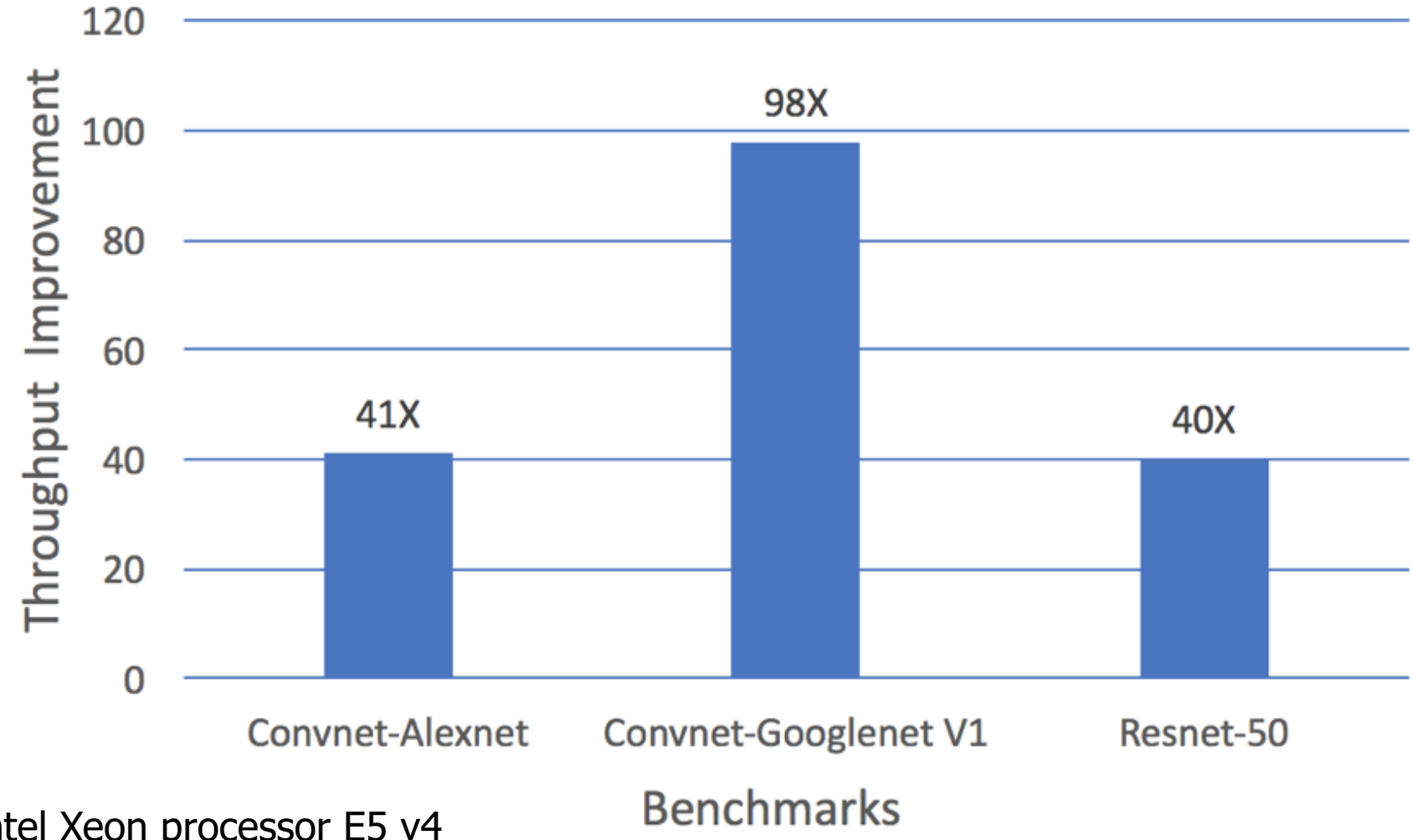
Inference and training throughput measured with FP32 instructions.
Inference with INT8 will be higher.

Advance previous generation AI workload performance with Intel® Xeon® Scalable Processors

Inference throughput batch size: 1 Training throughput batch size: 256 Configuration Details on Slide: 18, 20 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

Neon™ 2.0 improvements due to MKL

Don’t forget software … Using the right library can have a huge impact



Intel Xeon processor E5 v4

Source: https://www.intelnervana.com/neon-2-0-optimized-for-intel-architectures/

AI Performance – Software + Hardware

INFERENCE THROUGHPUT

Up to
138x

Intel® Xeon® Platinum 8180 Processor
higher Intel optimized Caffe GoogleNet v1 with Intel® MKL inference throughput compared to Intel® Xeon® Processor E5-2699 v3 with BVLC-Caffe

TRAINING THROUGHPUT

Up to
113x

Intel® Xeon® Platinum 8180 Processor
higher Intel Optimized Caffe AlexNet with Intel® MKL training throughput compared to Intel® Xeon® Processor E5-2699 v3 with BVLC-Caffe

Inference and training throughput measured with FP32 instructions. Inference with INT8 will be higher.

Deliver significant AI performance with hardware and software optimizations on Intel® Xeon® Scalable Processors

INFERENCE using FP32 Batch Size Caffe GoogleNet v1 256 AlexNet 256 Configuration Details on Slide: 18, 25
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.

AI DATACENTER

ALL PURPOSE

Intel® Xeon® Processor Family

MOST AGILE AI PLATFORM

Scalable performance for widest variety of AI & other datacenter workloads – including deep learning training & inference

HIGHLY-PARALLEL

Intel® Xeon Phi™ Processor (Knights Mill†)

FASTER DL TRAINING

Scalable performance optimized for even faster deep learning training and select highly-parallel datacenter workloads*

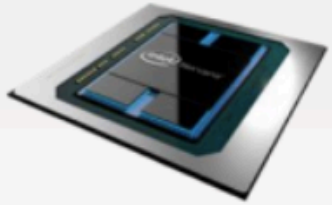
FLEXIBLE ACCELERATION

Intel® FPGA

ENHANCED DL INFERENCE

Scalable acceleration for deep learning inference in real-time with higher efficiency, and wide range of workloads & configurations

DEEP LEARNING



Crest Family†

DEEP LEARNING BY DESIGN

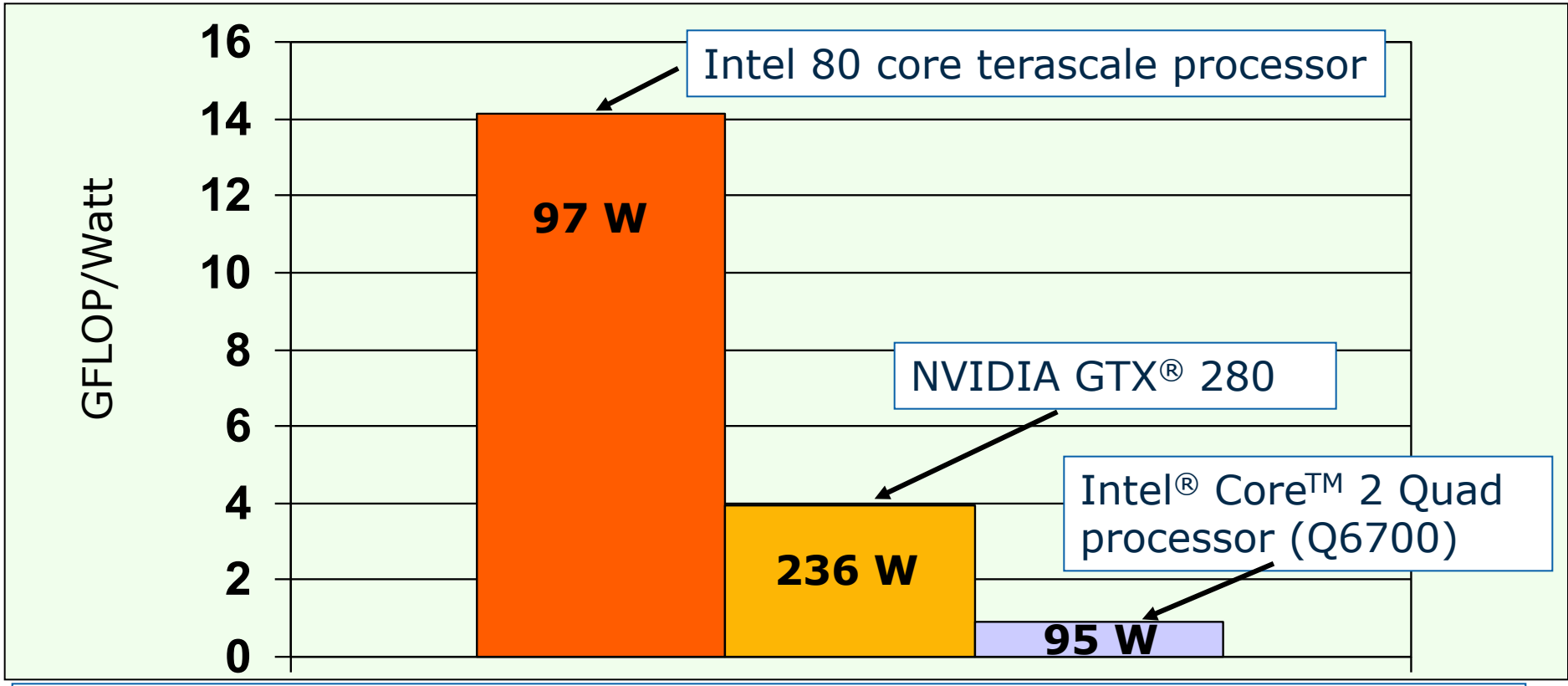
Scalable acceleration with best performance for intensive deep learning training & inference

Source: Hot Chips 2017, Dennis Bradford et. al.

†Codename for product that is coming soon
All Performance positioning claims are relative to other processor technologies in Intel's AI datacenter Portfolio
*Knights Mill (KNM); select = single-precision highly-parallel workloads generally scale to >100 threads and benefit from more vectorization, and may benefit from greater memory bandwidth e.g. energy (reverse time migration), deep learning training, etc.
All products, computer systems, dates, and figures specified are preliminary based on current expectation, and are subject to change without notice.

The benefits of Specialization

- Fair comparisons of performance per watt are very difficult.
  - 65 nm process technology
  - Peak single precision GFLOPS at thermal design point Watts



Specialized cores doing the work they are specialized to do deliver the best performance … specialization is a good thing

Sources: Intel and Technical Brief, NVIDIA GeForce® GTX 200 GPU Architectural Overview, TB-04044-001_v01, May 2008

Third party names are the property of their owners.

Deep Learning Processors from Intel® Nervana™

The Crest* family of products from Nervana … high performance for deep learning and bandwidth to support high throughput on real problems

> "It's a tensor processor, it deals with instructions that are matrix operations," Khosrowshahi explained. "So the instruction set is matrix 1 multiplied by matrix 2, go through a lookup table, and these big instructions that are high-level.[1]

"Knights Crest" processor: best-in-class Intel® Xeon™ processors integrated with with the Crest technology.

> "We expect the Intel Nervana platform to produce breakthrough performance and dramatic reductions in the time to train complex neural networks. Before the end of the decade, Intel will deliver a 100-fold increase in performance that will turbocharge the pace of innovation in the emerging deep learning space."[2]

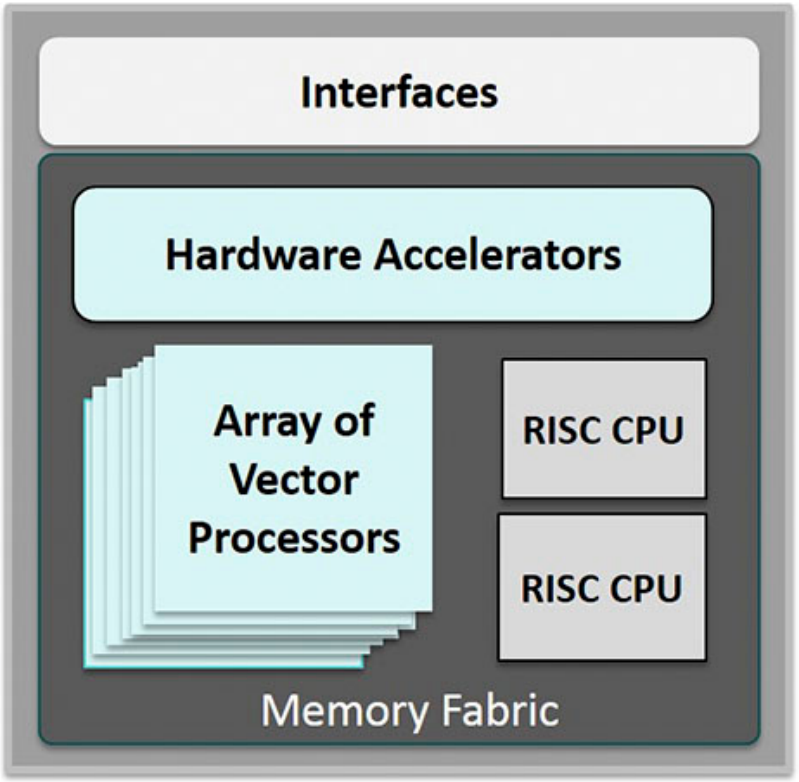
*Crest, knights crest and lake crest are code names. Actual product names may be completely different.

source:1 https://www.top500.org/news/intel-retools-product-roadmap-with-ai-silicon-1/
2 http://www.zdnet.com/article/ai-training-needs-a-new-chip-architecture-intel/#ftag=YHFb1d24ec?yptr=yahoo

Low Power Inferencing: Movidus™ VPU

Myriad™ X: 4+ TOPS of total performance in a tiny low power form factor for autonomous device solutions

- 16 Programmable 128-bit VLIW Vector Processors:
- 16 Configurable MIPI Lanes:
  - connect up to 8 HD cameras
  - Support 700M pixels/sec image processing throughput
- 20 HW Vision Accelerators (optical flow, stereo depth, etc)
- 2.5 MB of Homogenous On-Chip Memory for up to 450 GB per second of internal bandwidth.



Myriad 2 Vision Processor Unit (VPU)

Programmable through the Eclipse based Myriad SDK supporting Café

The Specialized processor shaking up the hardware world today

In-Datacenter Performance Analysis of a Tensor Processing Unit™

Norman P. Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, Rick Boyle, Pierre-luc Cantin, Clifford Chao, Chris Clark, Jeremy Coriell, Mike Daley, Matt Dau, Jeffrey Dean, Ben Gelb, Tara Vazir Ghaemmaghami, Rajendra Gottipati, William Gulland, Robert Hagmann, C. Richard Ho, Doug Hogberg, John Hu, Robert Hundt, Dan Hurt, Julian Ibarz, Aaron Jaffey, Alek Jaworski, Alexander Kaplan, Harshit Khaitan, Andy Koch, Naveen Kumar, Steve Lacy, James Laudon, James Law, Diemthu Le, Chris Leary, Zhuyuan Liu, Kyle Lucke, Alan Lundin, Gordon MacKean, Adriana Maggiore, Maire Mahony, Kieran Miller, Rahul Nagarajan, Ravi Narayanaswami, Ray Ni, Kathy Nix, Thomas Norrie, Mark Omernick, Narayana Penukonda, Andy Phelps, Jonathan Ross, Matt Ross, Amir Salek, Emad Samadiani, Chris Severn, Gregory Sizikov, Matthew Snelham, Jed Souter, Dan Steinberg, Andy Swing, Mercedes Tan, Gregory Thorson, Bo Tian, Horia Toma, Erick Tuttle, Vijay Vasudevan, Richard Walter, Walter Wang, Eric Wilcox, and Doe Hyun Yoon

Google, Inc., Mountain View, CA USA

Email: {jouppi, cliffy, nishantpatil, davidpatterson} @google.com

To appear at the 44th International Symposium on Computer Architecture (ISCA), Toronto, Canada, June 26, 2017.

Third Party Names are the property of their owners.

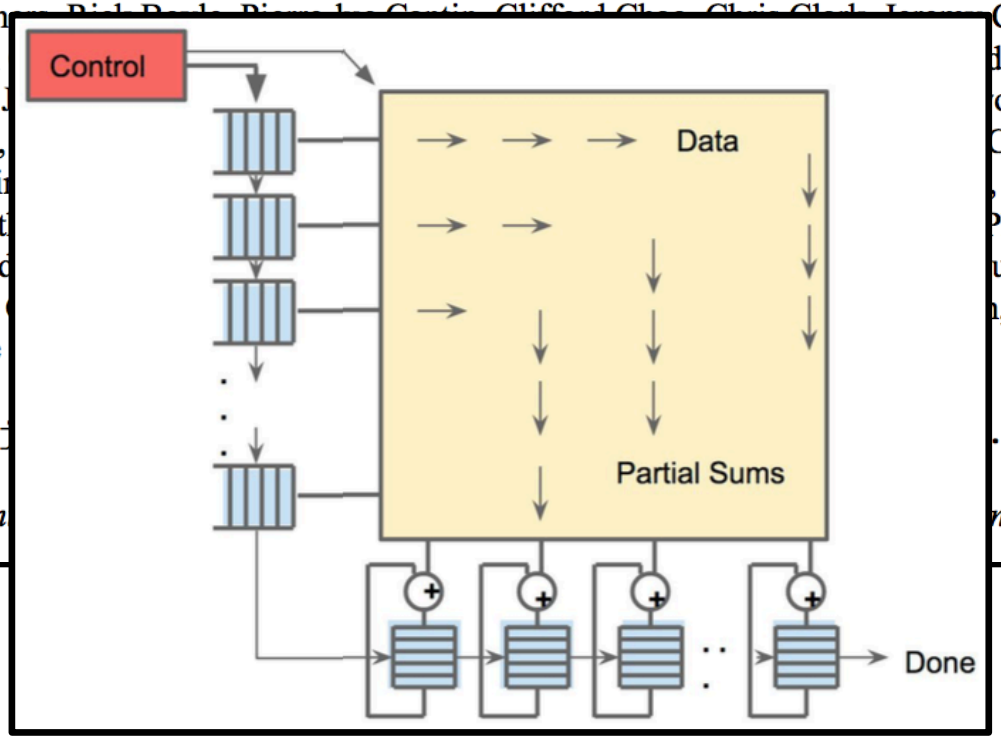
The Specialized processor shaking up the hardware world today

In-Datacenter Performance Analysis of a Tensor Processing Unit™

Norman P. Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borch… Coriell, Mike Daley, Matt Dau, Jeffrey Dean, Ben … d, Robert Hagmann, C. Richard Ho, Doug Hogberg, J… orski, Alexander Kaplan, Harshit Khaitan, Andy Koch, … Chris Leary, Zhuyuan Liu, Kyle Lucke, Alan Lundi… , Rahul Nagarajan, Ravi Narayanaswami, Ray Ni, Kat… Phelps, Jonathan Ross, Matt Ross, Amir Salek, Emad… uter, Dan Steinberg, Andy Swing, Mercedes Tan, … , Richard Walter, Walter Wang, Eric Wilcox, and Doe…

Email: {… .com

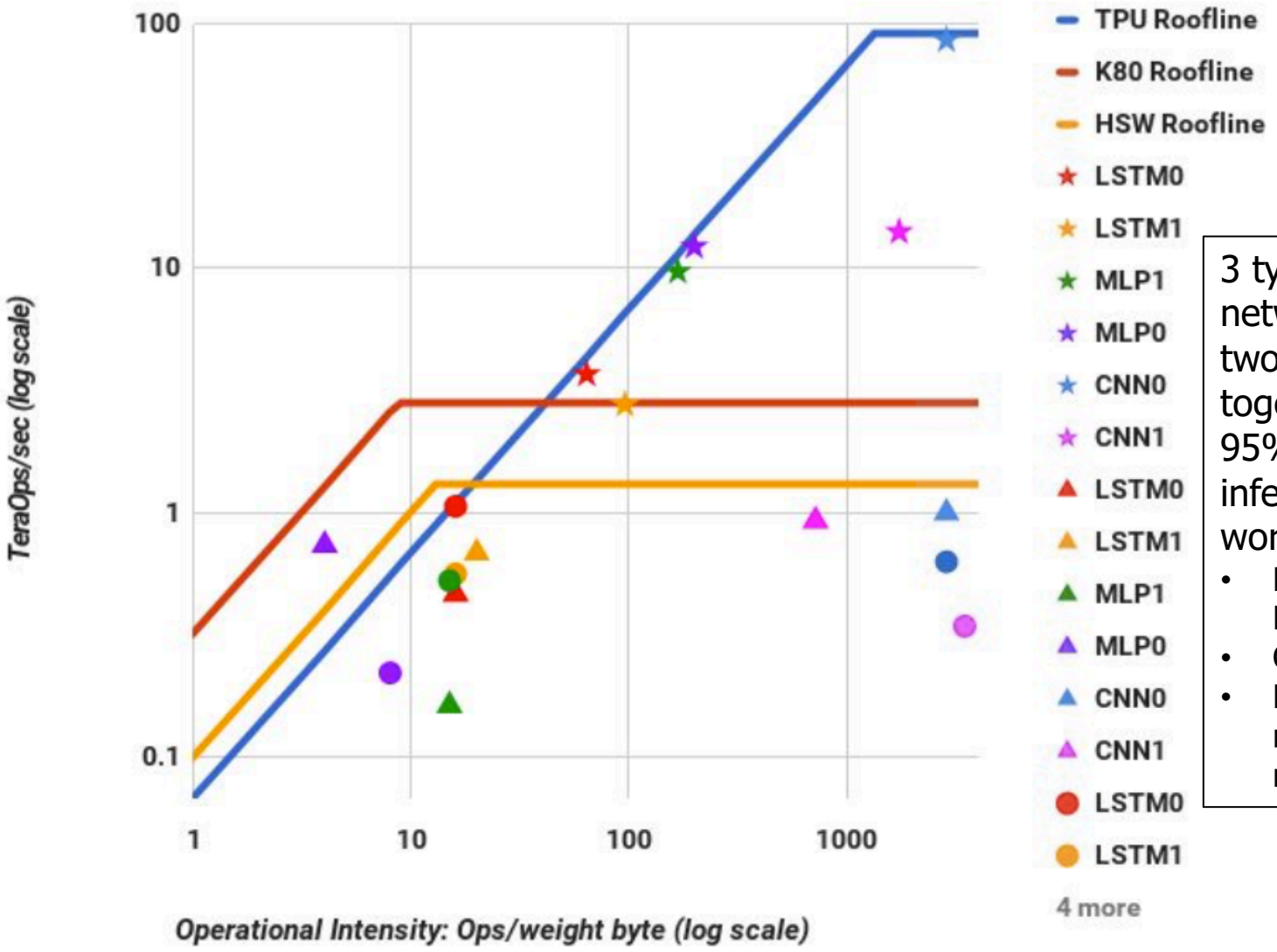
To appear at the 44th In… nada, June 26, 2017.



Systolic matrix multiply unit … output
256 results per cycle once pipeline filled

Third Party Names are the property of their owners.

TPU Performance results



3 types of neural networks (NN) in two flavors each … together comprise 95% of Google inferencing workloads

- Multi-Layer Perceptrons
- Convolutional NN
- Long Short-term memory (a recurrent NN)

Stars TPU, triangles for K80, circles for Haswell

Third party names are the property of their owners.

TPU

- TPU has disrupted the status quo … there is no turning back.
- Motivated by the needs of AI, Google took its DL-HW-fate in its own hands and created an amazing application specific ASIC … and it took them about 15 months to do this.

Order-of-magnitude differences between commercial products are rare in computer architecture, which may lead to the TPU becoming an archetype for domain-specific architectures. We expect that many will build successors that will raise the bar even higher.

Third party names are the property of their owners.

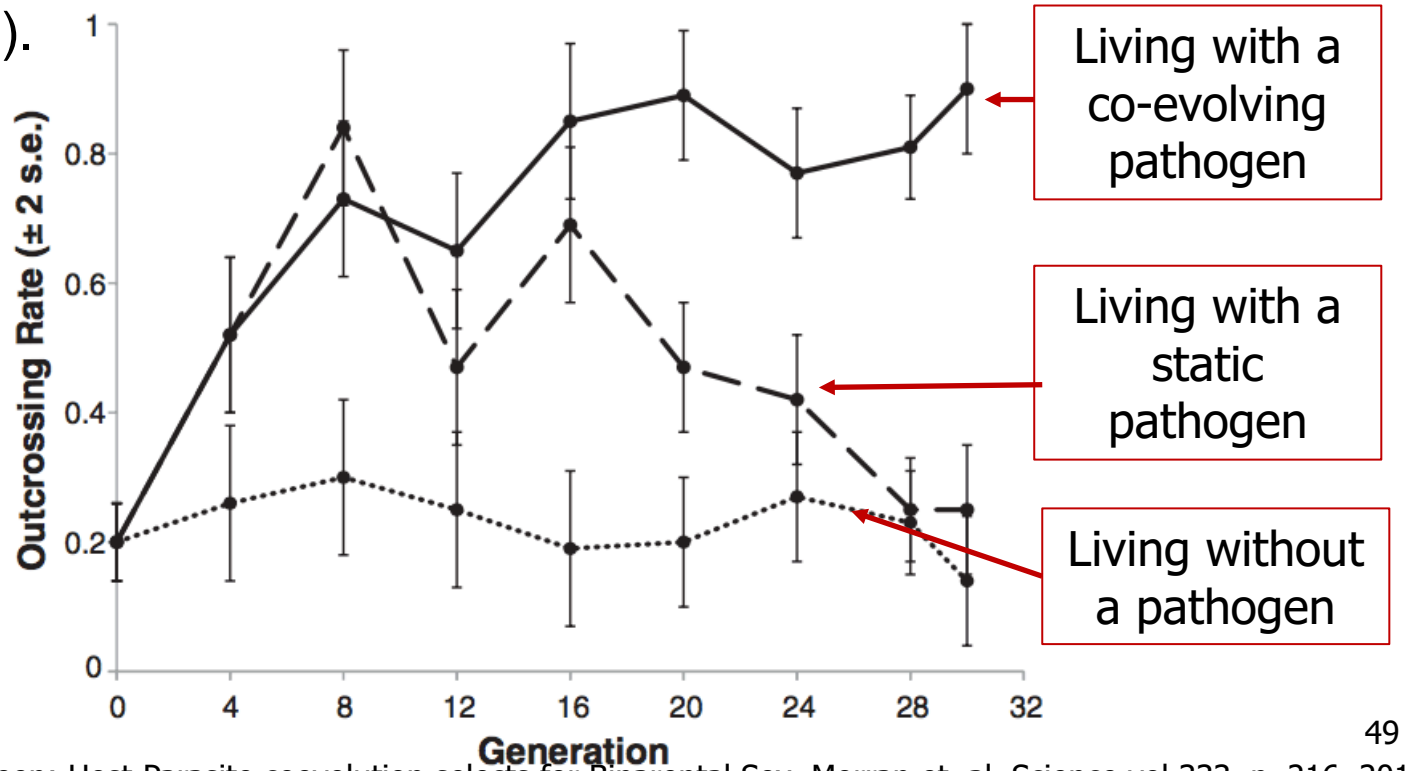
HW is moving so fast, it feels like we are living within the Red Queen's race

it takes all the running you can do, to keep in the same place. If you want to get somewhere else, you must run at least twice as fast as that!' *Lewis Carrol, Through the looking glass, chapter 2*

Red Queen's Race and Evolutionary Biology

- Why is sex almost universal in animals?
  - Sex is expensive … it takes two to make babies instead of one.
- Red Queen hypothesis:
  - Species locked in a co-evolution relationship put pressure on each other. They must evolve rapidly just to keep each other in check … i.e. they must run as fast as they can (i.e. mix-up their genomes) just so stay in place (i.e. to survive).

Fraction of a population of C. Elegans having sex.



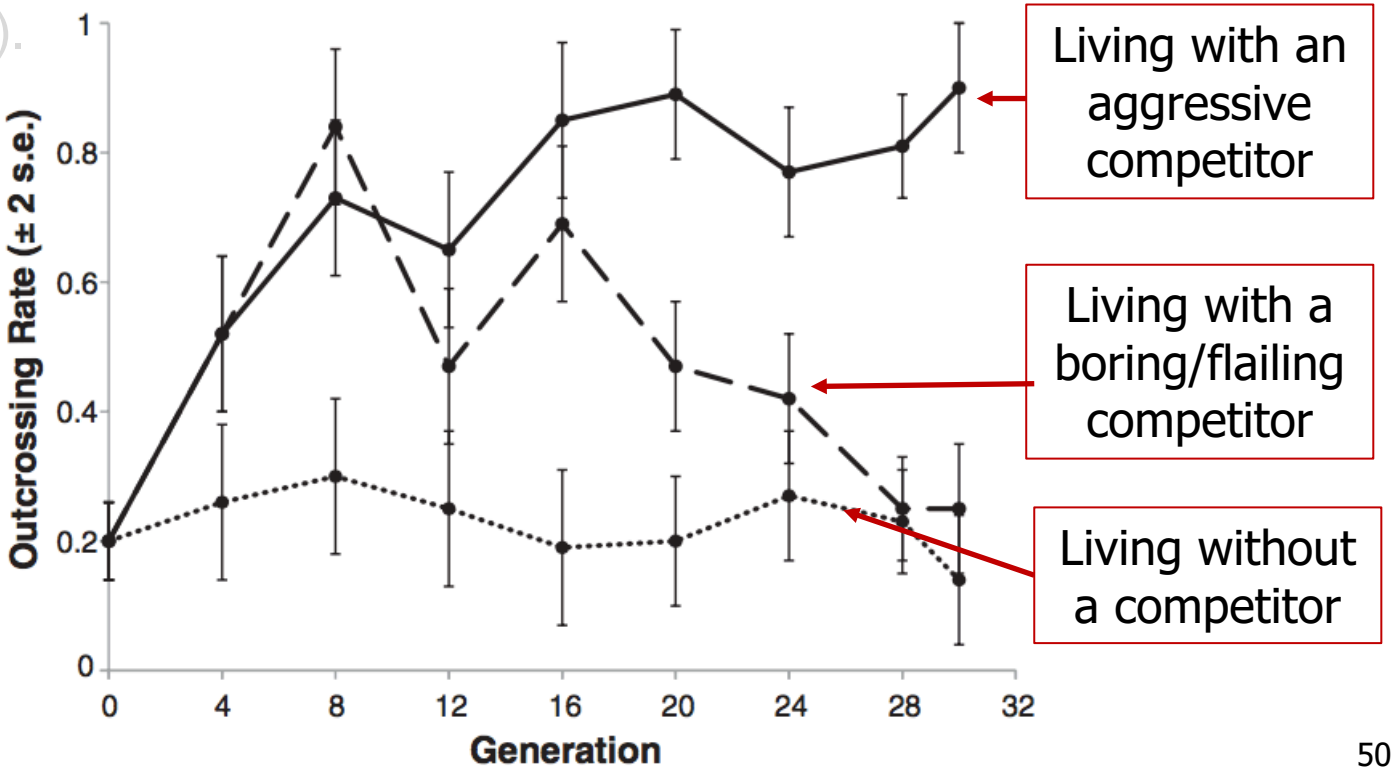
Source: Running with the Red Queen: Host-Parasite coevolution selects for Biparental Sex, Morran et. al. Science vol 333, p. 216, 2011

Red Queen's Race and Computers

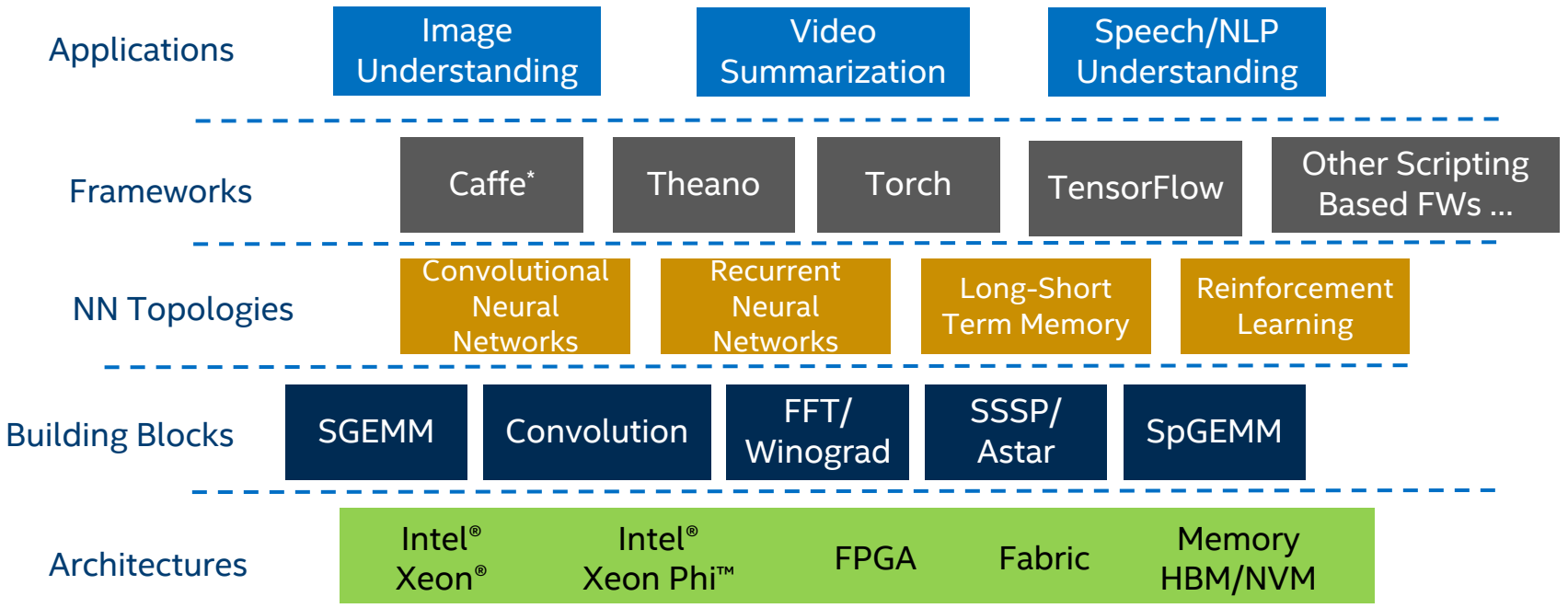
- Why is sex almost universal in animals?
  - Sex is expensive … it takes two to make babies instead of one.
- Red
  - Species locked in a co-evolution relationship put pressure on each other. They must evolve rapidly just to keep each other in check … i.e. they must run as fast as they can (i.e. mix-up their genomes) just so stay in place (i.e. to survive).

The analogy to the world of Hardware is striking!

Fraction of time spent with HW innovation



How do programmers survive?



Given common building blocks across different neural networks … we can hide complexity behind high-level interfaces

High level interfaces that hide HW Complexity

- **Libraries**: common operations exposed through a fixed API
- **Frameworks**: A partial solutions to a class of problems that can be specialized to solve a specific problem.
- **Domain Specific Languages**: programmable solutions specialized to a domain as Stand-alone languages or embedded inside a host language.

DAAL Pandas MKL

Spark MLlib

Torch

Caffe

theano

Caffe2

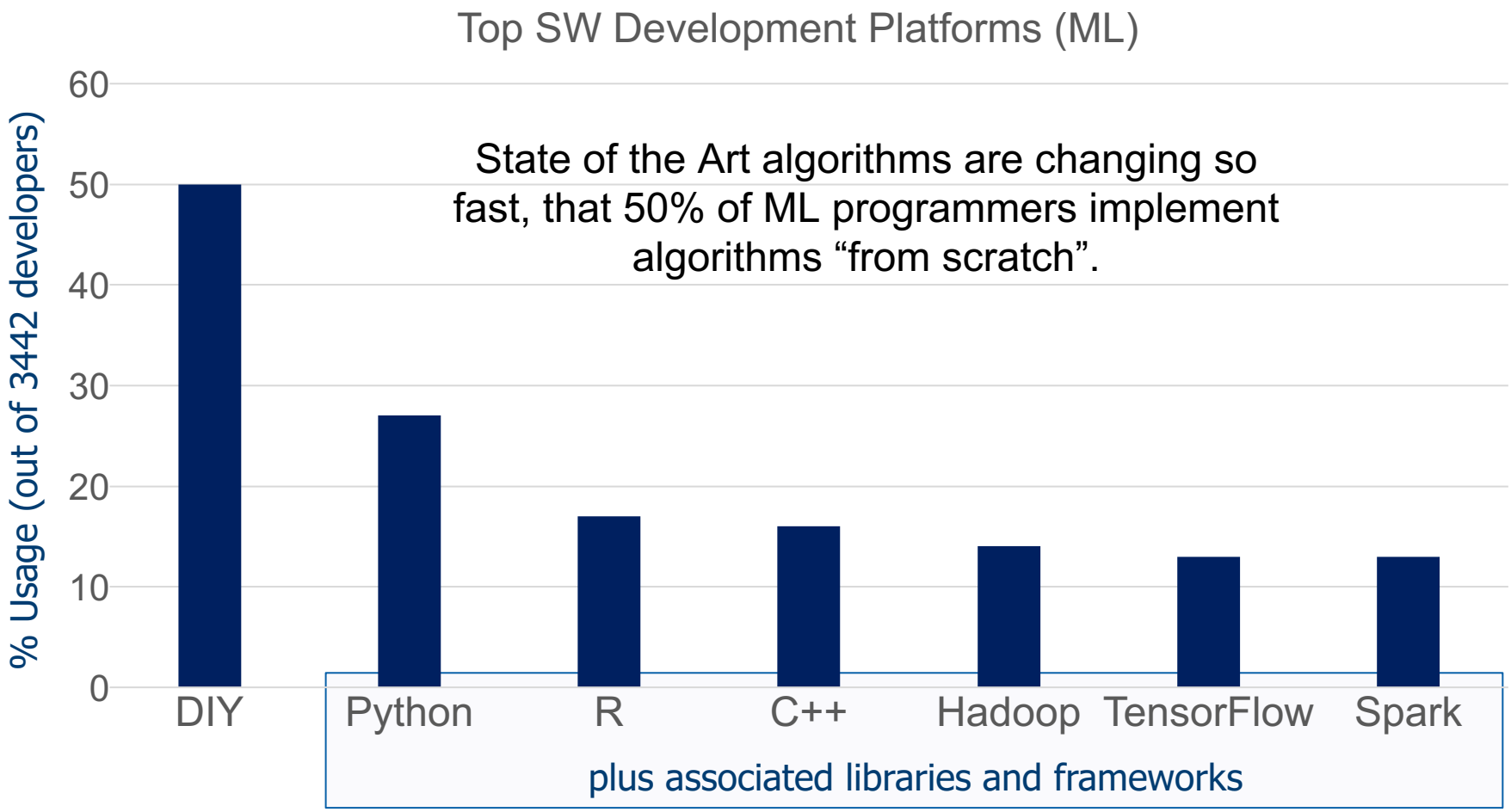
Nervana Graph

TensorFlow

Halide

Third party names are the property of their owners.

… But Algorithms are changing too fast for libraries/frameworks to keep up!!

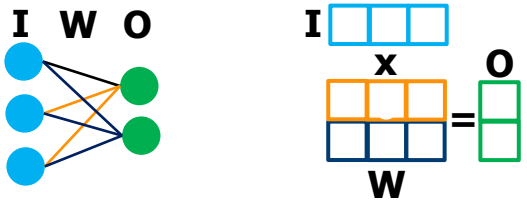


DIY refers to developers writing their own ML algorithms “from scratch”.

Third party names are the property of their owners.

Source: State of the Developer Nation Q!’2017, Vision Mobile, http://vmob.me/DE1Q17

DNNs Evolving Rapidly



Deeper
More
params?
Larger
model?

LeNet5	AlexNet	VGG	GoogLeNet	ResNet
5 layers	(~80% Top5)	(~89% top5)	(~89% top5)	(~94% top5)
Params: 1M	8 layers	19 layers	22 layers	152 layers
Model: 4MB	Params: 60M	Params: 140M	Params: 6M	Params: 60M
	Model: 240MB	Model: 500MB	Model: 24MB	Model: 240MB

Before 2000 — 2012 — 2013 — 2014 — 2015 — 2016

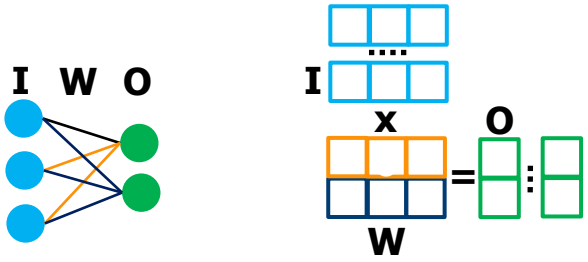
Source: Can FPGAs Beat GPUs in Accelerating Next-Generation Deep Neural Networks? E. Nurvitadhi, G. Venkatesh, J. Sim, D. Marr, R. Huang, J. G. H. Ong, Y. T. Liew, K. Srivatsan, D. Moss, S. Subhaschandra, G. Boudoukh, FPGA'17, Feb 2017

Third party names are the property of their owners.

DNNs Evolving Rapidly

Many efforts to improve efficiency

Batching



Deeper
More params?
Larger model?

LeNet5	AlexNet		VGG	GoogLeNet	ResNet	
5 layers	(~80% Top5)		(~89% top5)	(~89% top5)	(~94% top5)	
Params: 1M	8 layers		19 layers	22 layers	152 layers	
Model: 4MB	Params: 60M		Params: 140M	Params: 6M	Params: 60M	
	Model: 240MB		Model: 500MB	Model: 24MB	Model: 240MB	
Before 2000	2012	2013	2014		2015	2016

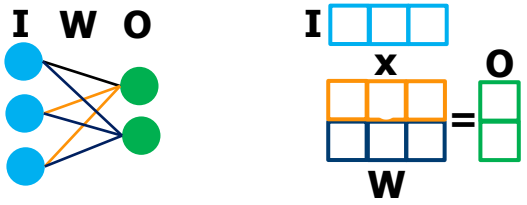
Source: Can FPGAs Beat GPUs in Accelerating Next-Generation Deep Neural Networks? E. Nurvitadhi, G. Venkatesh, J. Sim, D. Marr, R. Huang, J. G. H. Ong, Y. T. Liew, K. Srivatsan, D. Moss, S. Subhaschandra, G. Boudoukh, FPGA'17, Feb 2017

Third party names are the property of their owners.

DNNs Evolving Rapidly

Many efforts to improve efficiency

Batching
Reduce bitwidth



BinaryConnect [NIPS'15]

XNORNet

Deeper
More params?
Larger model?

LeNet5	AlexNet	VGG	GoogLeNet	ResNet
5 layers	(~80% Top5)	(~89% top5)	(~89% top5)	(~94% top5)
Params: 1M	8 layers	19 layers	22 layers	152 layers
Model: 4MB	Params: 60M	Params: 140M	Params: 6M	Params: 60M
	Model: 240MB	Model: 500MB	Model: 24MB	Model: 240MB

Before 2000 — 2012 — 2013 — 2014 — 2015 — 2016

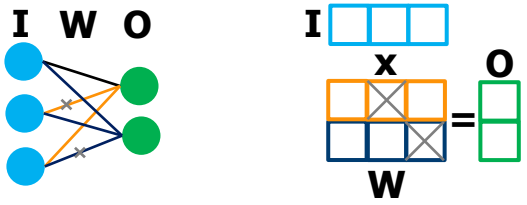
Source: Can FPGAs Beat GPUs in Accelerating Next-Generation Deep Neural Networks? E. Nurvitadhi, G. Venkatesh, J. Sim, D. Marr, R. Huang, J. G. H. Ong, Y. T. Liew, K. Srivatsan, D. Moss, S. Subhaschandra, G. Boudoukh, FPGA'17, Feb 2017

Third party names are the property of their owners.

DNNs Evolving Rapidly

Many efforts to improve efficiency

Batching
Reduce bitwidth
Sparse weights



BinaryConnect [NIPS'15]

XNORNet

Deeper More params? Larger model?

LeNet5	AlexNet	VGG	GoogLeNet	ResNet
5 layers	(~80% Top5)	(~89% top5)	(~89% top5)	(~94% top5)
Params: 1M	8 layers	19 layers	22 layers	152 layers
Model: 4MB	Params: 60M	Params: 140M	Params: 6M	Params: 60M
	Model: 240MB	Model: 500MB	Model: 24MB	Model: 240MB

Before 2000 — 2012 — 2013 — 2014 — 2015 — 2016

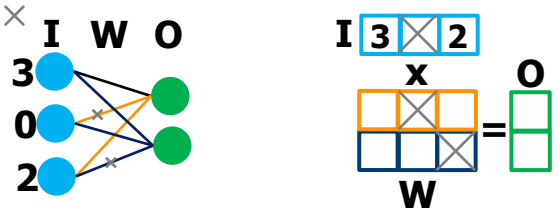
Source: Can FPGAs Beat GPUs in Accelerating Next-Generation Deep Neural Networks? E. Nurvitadhi, G. Venkatesh, J. Sim, D. Marr, R. Huang, J. G. H. Ong, Y. T. Liew, K. Srivatsan, D. Moss, S. Subhaschandra, G. Boudoukh, FPGA'17, Feb 2017

Third party names are the property of their owners.

DNNs Evolving Rapidly

Many efforts to improve efficiency

Batching
Reduce bitwidth
Sparse weights
Sparse activations



BinaryConnect [NIPS'15]
SparseCNN [CVPR'15]
Pruning [NIPS'15]

XNORNet
TernaryConnect [ICLR'16]

Spatially SparseCNN [CIFAR-10 winner '14]

Deeper
More params?
Larger model?

LeNet5
5 layers
Params: 1M
Model: 4MB

AlexNet
(~80% Top5)
8 layers
Params: 60M
Model: 240MB

VGG
(~89% top5)
19 layers
Params: 140M
Model: 500MB

GoogLeNet
(~89% top5)
22 layers
Params: 6M
Model: 24MB

ResNet
(~94% top5)
152 layers
Params: 60M
Model: 240MB

Before 2000 — 2012 — 2013 — 2014 — 2015 — 2016

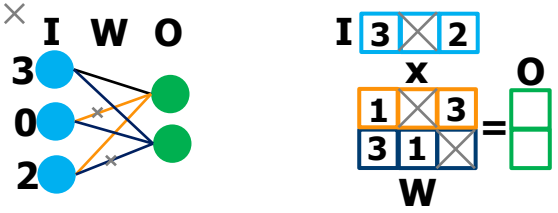
Source: Can FPGAs Beat GPUs in Accelerating Next-Generation Deep Neural Networks? E. Nurvitadhi, G. Venkatesh, J. Sim, D. Marr, R. Huang, J. G. H. Ong, Y. T. Liew, K. Srivatsan, D. Moss, S. Subhaschandra, G. Boudoukh, FPGA'17, Feb 2017

Third party names are the property of their owners.

DNNs Evolving Rapidly

Many efforts to improve efficiency

Batching
Reduce bitwidth
Sparse weights
Sparse activations
Compression



BinaryConnect [NIPS'15]
SparseCNN [CVPR'15]
Pruning [NIPS'15]

XNORNet
TernaryConnect [ICLR'16]

Spatially SparseCNN [CIFAR-10 winner '14]

Deeper
More params?
Larger model?

LeNet5
5 layers
Params: 1M
Model: 4MB

AlexNet
(~80% Top5)
8 layers
Params: 60M
Model: 240MB

VGG
(~89% top5)
19 layers
Params: 140M
Model: 500MB

GoogLeNet
(~89% top5)
22 layers
Params: 6M
Model: 24MB

ResNet
(~94% top5)
152 layers
Params: 60M
Model: 240MB

Before 2000 — 2012 — 2013 — 2014 — 2015 — 2016

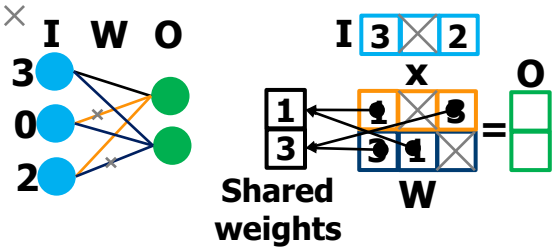
Source: Can FPGAs Beat GPUs in Accelerating Next-Generation Deep Neural Networks? E. Nurvitadhi, G. Venkatesh, J. Sim, D. Marr, R. Huang, J. G. H. Ong, Y. T. Liew, K. Srivatsan, D. Moss, S. Subhaschandra, G. Boudoukh, FPGA'17, Feb 2017

Third party names are the property of their owners.

DNNs Evolving Rapidly

Many efforts to improve efficiency

Batching
Reduce bitwidth
Sparse weights
Sparse activations
Compression



Deeper
More params?
Larger model?

Before 2000	2012	2013	2014	2015	2016
LeNet5 **5 layers** **Params: 1M** **Model: 4MB**	**AlexNet** **(~80% Top5)** **8 layers** **Params: 60M** **Model: 240MB**		**VGG** **(~89% top5)** **19 layers** **Params: 140M** **Model: 500MB** **GoogLeNet** **(~89% top5)** **22 layers** **Params: 6M** **Model: 24MB**	**ResNet** **(~94% top5)** **152 layers** **Params: 60M** **Model: 240MB**	

Spatially SparseCNN [CIFAR-10 winner '14]

BinaryConnect [NIPS'15]
SparseCNN [CVPR'15]
Pruning [NIPS'15]
HashedNets [ICML'15]

XNORNet
TernaryConnect [ICLR'16]
DeepComp [ICLR'16]

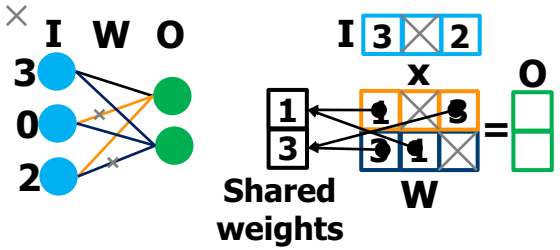
Source: Can FPGAs Beat GPUs in Accelerating Next-Generation Deep Neural Networks? E. Nurvitadhi, G. Venkatesh, J. Sim, D. Marr, R. Huang, J. G. H. Ong, Y. T. Liew, K. Srivatsan, D. Moss, S. Subhaschandra, G. Boudoukh, FPGA'17, Feb 2017

Third party names are the property of their owners.

DNNs Evolving Rapidly

Many efforts to improve efficiency

Batching
Reduce bitwidth
Sparse weights
Sparse activations
Compression
Compact network



Deeper
More params?
Larger model?

Before 2000	2012	2013	2014	2015	2016
LeNet5 **5 layers** **Params: 1M** **Model: 4MB**	**AlexNet** **(~80% Top5)** **8 layers** **Params: 60M** **Model: 240MB**		**VGG** **(~89% top5)** **19 layers** **Params: 140M** **Model: 500MB** **GoogLeNet** **(~89% top5)** **22 layers** **Params: 6M** **Model: 24MB**	**ResNet** **(~94% top5)** **152 layers** **Params: 60M** **Model: 240MB**	

Spatially SparseCNN [CIFAR-10 winner '14]

BinaryConnect [NIPS'15]
SparseCNN [CVPR'15]
Pruning [NIPS'15]
HashedNets [ICML'15]

XNORNet
TernaryConnect [ICLR'16]
DeepComp [ICLR'16]
SqueezeNet

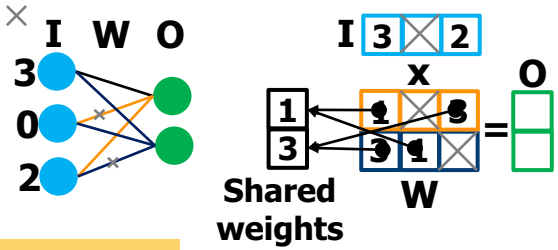
Source: Can FPGAs Beat GPUs in Accelerating Next-Generation Deep Neural Networks? E. Nurvitadhi, G. Venkatesh, J. Sim, D. Marr, R. Huang, J. G. H. Ong, Y. T. Liew, K. Srivatsan, D. Moss, S. Subhaschandra, G. Boudoukh, FPGA'17, Feb 2017

Third party names are the property of their owners.

DNNs Evolving Rapidly

Many efforts to improve efficiency

Batching
Reduce bitwidth
Sparse weights
Sparse activations
Compression
Compact network



All applicable for inference
Some for training

Deeper
More params?
Larger model?

LeNet5
5 layers
Params: 1M
Model: 4MB

AlexNet
(~80% Top5)
8 layers
Params: 60M
Model: 240MB

Spatially SparseCNN [CIFAR-10 winner '14]

VGG
(~89% top5)
19 layers
Params: 140M
Model: 500MB

GoogLeNet
(~89% top5)
22 layers
Params: 6M
Model: 24MB

BinaryConnect [NIPS'15]
SparseCNN [CVPR'15]
Pruning [NIPS'15]
HashedNets [ICML'15]

ResNet
(~94% top5)
152 layers
Params: 60M
Model: 240MB

XNORNet
TernaryConnect [ICLR'16]
DeepComp [ICLR'16]
SqueezeNet

Before 2000 | 2012 | 2013 | 2014 | 2015 | 2016

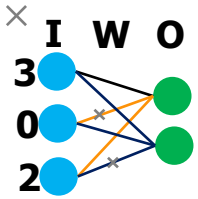
Source: Can FPGAs Beat GPUs in Accelerating Next-Generation Deep Neural Networks? E. Nurvitadhi, G. Venkatesh, J. Sim, D. Marr, R. Huang, J. G. H. Ong, Y. T. Liew, K. Srivatsan, D. Moss, S. Subhaschandra, G. Boudoukh, FPGA'17, Feb 2017

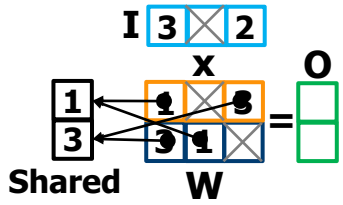
Third party names are the property of their owners.

DNNs Evolving Rapidly

Many efforts to improve efficiency

Batching
Reduce bitwidth
Sparse weights
Sparse activations
Compression
Compact network





Shared weights

All applicable for inference
Some for training

SqueezeNet + DeepCompression: 6-bit, 20-50% sparse AlexNet accuracy, ~500x smaller (0.5MB)

XNORnet (1-bit) → ~2% AlexNet

TernaryNet (2-bit, 50% sparse) → ~1% ResNet

Deeper
More params?
Larger model?

LeNet5
5 layers
Params: 1M
Model: 4MB

AlexNet
(~80% Top5)
8 layers
Params: 60M
Model: 240MB

Spatially SparseCNN [CIFAR-10 winner '14]

VGG
(~89% top5)
19 layers
Params: 140M
Model: 500MB

GoogLeNet
(~89% top5)
22 layers
Params: 6M
Model: 24MB

[NIPS'15]
SparseCNN [CVPR'15]
Pruning [NIPS'15]
HashedNets [ICML'15]

ResNet
(~94% top5)
152 layers
Params: 60M
Model: 240MB

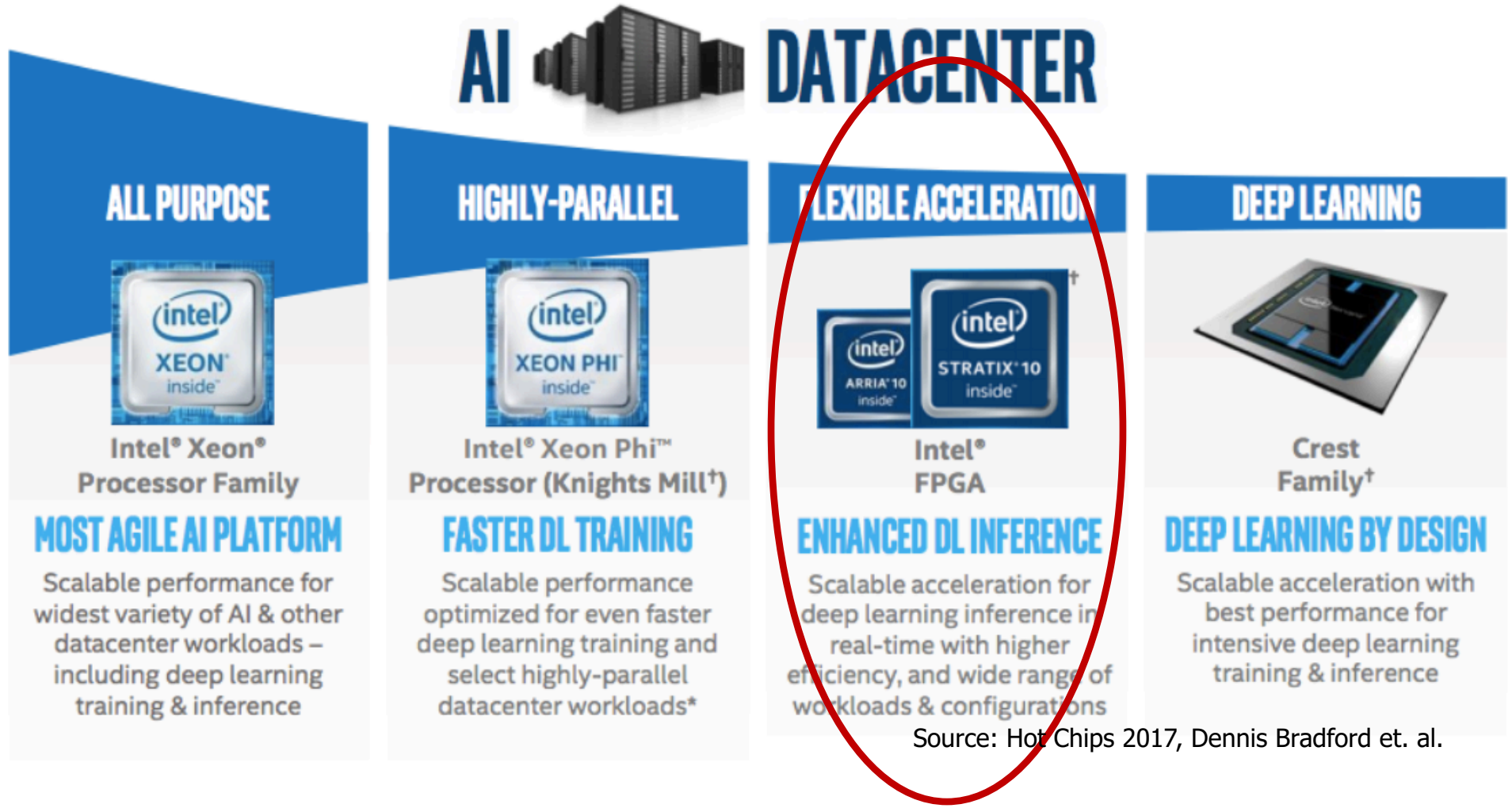
XNORNet
TernaryConnect [ICLR'16]
DeepComp [ICLR'16]
SqueezeNet

Before 2000 | 2012 | 2013 | 2014 | 2015 | 2016

Next-gen DNNs: more irregular with custom data types

Source: Can FPGAs Beat GPUs in Accelerating Next-Generation Deep Neural Networks? E. Nurvitadhi, G. Venkatesh, J. Sim, D. Marr, R. Huang, J. G. H. Ong, Y. T. Liew, K. Srivatsan, D. Moss, S. Subhaschandra, G. Boudoukh, FPGA'17, Feb 2017

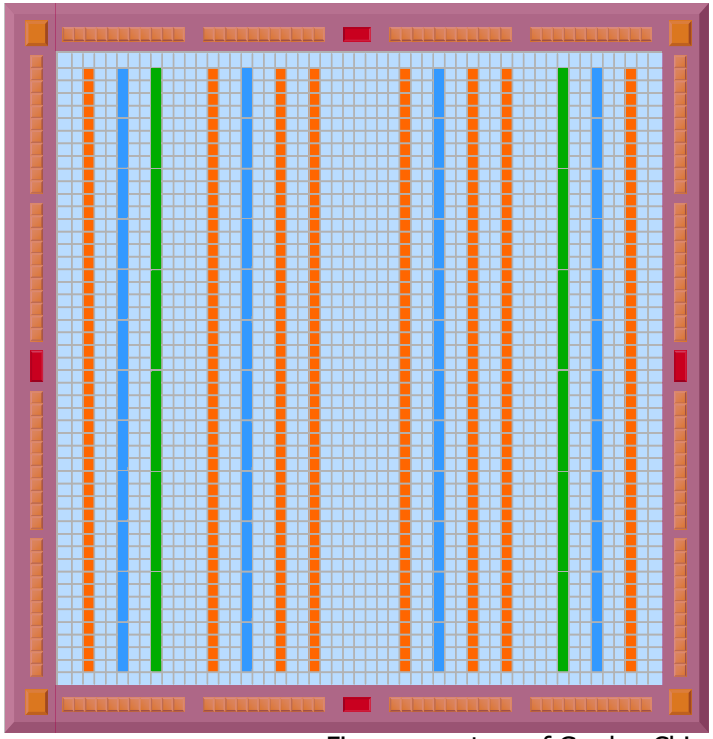
Third party names are the property of their owners.



Source: Hot Chips 2017, Dennis Bradford et. al.

†Codename for product that is coming soon
All Performance positioning claims are relative to other processor technologies in Intel's AI datacenter Portfolio
*Knights Mill (KNM); select = single-precision highly-parallel workloads generally scale to >100 threads and benefit from more vectorization, and may benefit from greater memory bandwidth e.g. energy (reverse time migration), deep learning training, etc.
All products, computer systems, dates, and figures specified are preliminary based on current expectation, and are subject to change without notice.

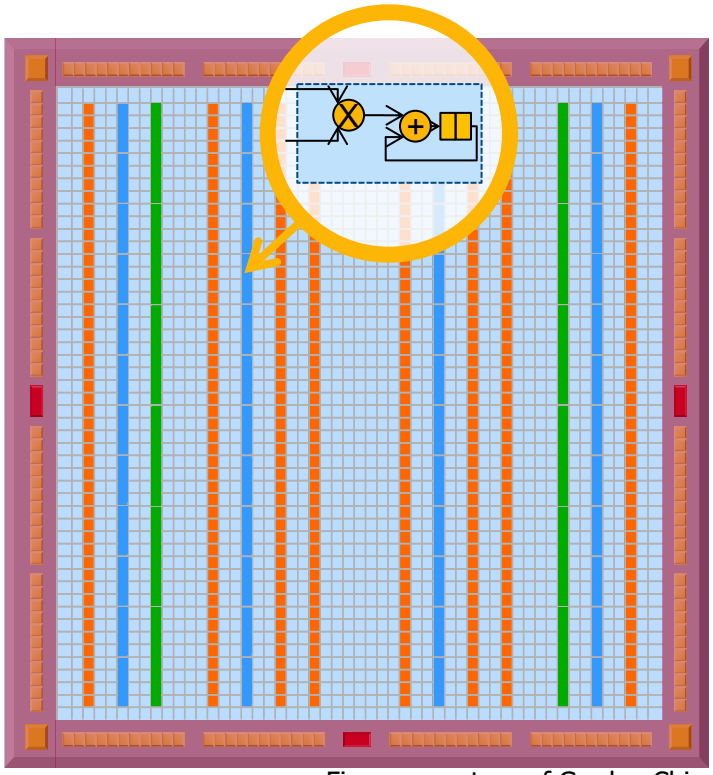
FPGA fabric is great for irregular (and regular) computation



Figures courtesy of Gordon Chiu

FPGA fabric is great for irregular (and regular) computation

1000s of hard DSPs (floating-point units)



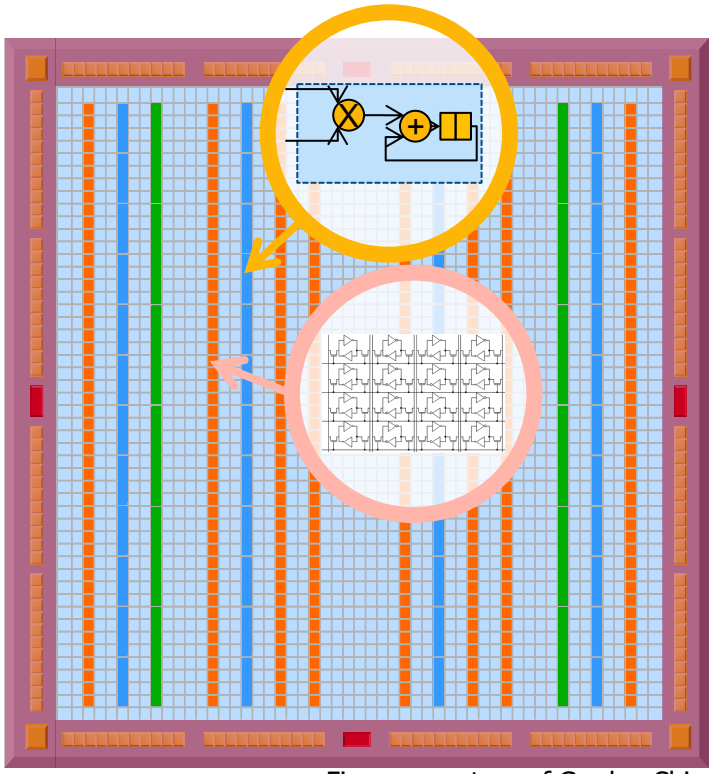
Figures courtesy of Gordon Chiu

M20K: A 2.5kB SRAM memory block.

spad: Scratch pad

FPGA fabric is great for irregular (and regular) computation

1000s of hard DSPs (floating-point units)
1000s of Hard "M20K" SRAMs (2.5KB/SRAM)



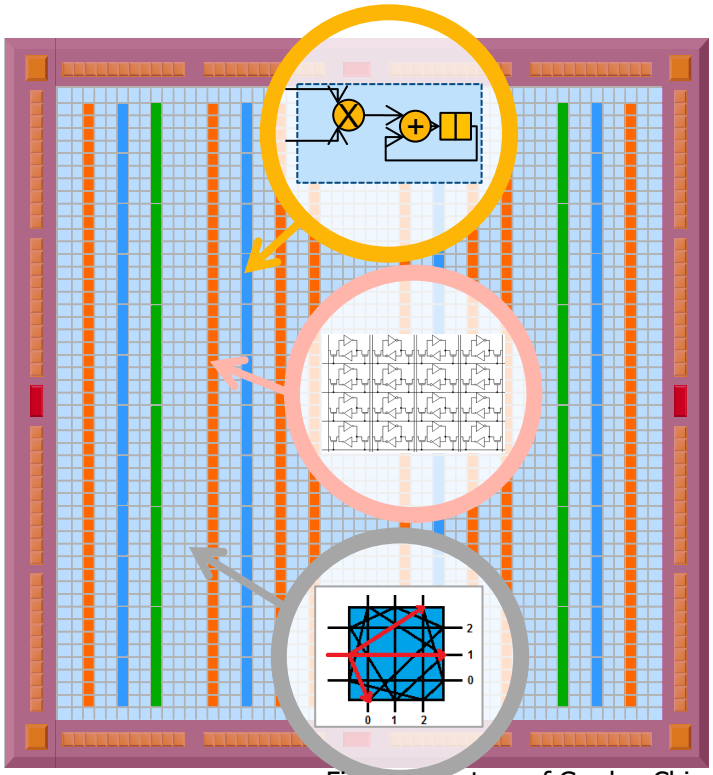
Figures courtesy of Gordon Chiu

M20K: A 2.5kB SRAM memory block.

spad: Scratch pad

FPGA fabric is great for irregular (and regular) computation

1000s of hard DSPs (floating-point units)
1000s of Hard "M20K" SRAMs (2.5KB/SRAM)
Sea of Programmable Logic and Routing



Figures courtesy of Gordon Chiu

M20K: A 2.5kB SRAM memory block.

spad: Scratch pad

FPGA fabric is great for irregular (and regular) computation

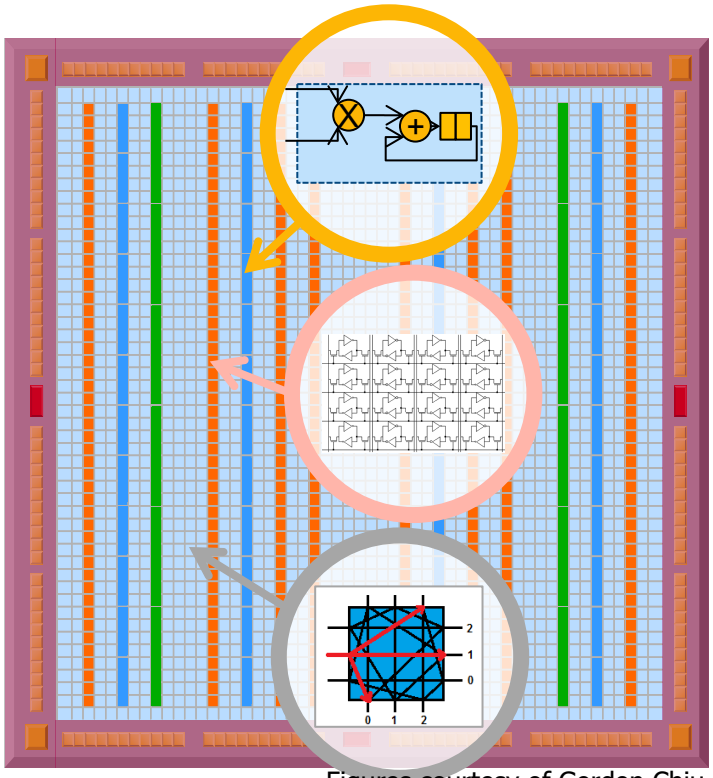
1000s of hard DSPs (floating-point units)
1000s of Hard "M20K" SRAMs (2.5KB/SRAM)

Sea of Programmable Logic and Routing

Extreme degree of customizations

Arbitrary bitwidth, mix bitwidths, etc

Arbitrary SRAMs compositions (spad, $, fifo, ..)



Figures courtesy of Gordon Chiu

M20K: A 2.5kB SRAM memory block.

spad: Scratch pad

FPGA fabric is great for irregular (and regular) computation

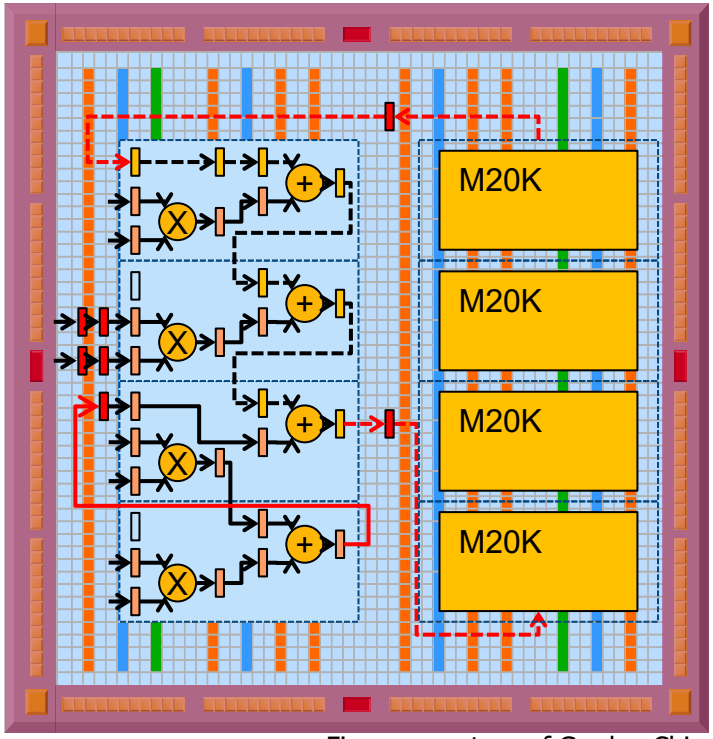
1000s of hard DSPs (floating-point units)
1000s of Hard "M20K" SRAMs (2.5KB/SRAM)
Sea of Programmable Logic and Routing

Extreme degree of customizations

Arbitrary bitwidth, mix bitwidths, etc

Arbitrary SRAMs compositions (spad, $, fifo, ..)

Arbitrary DNN architectures



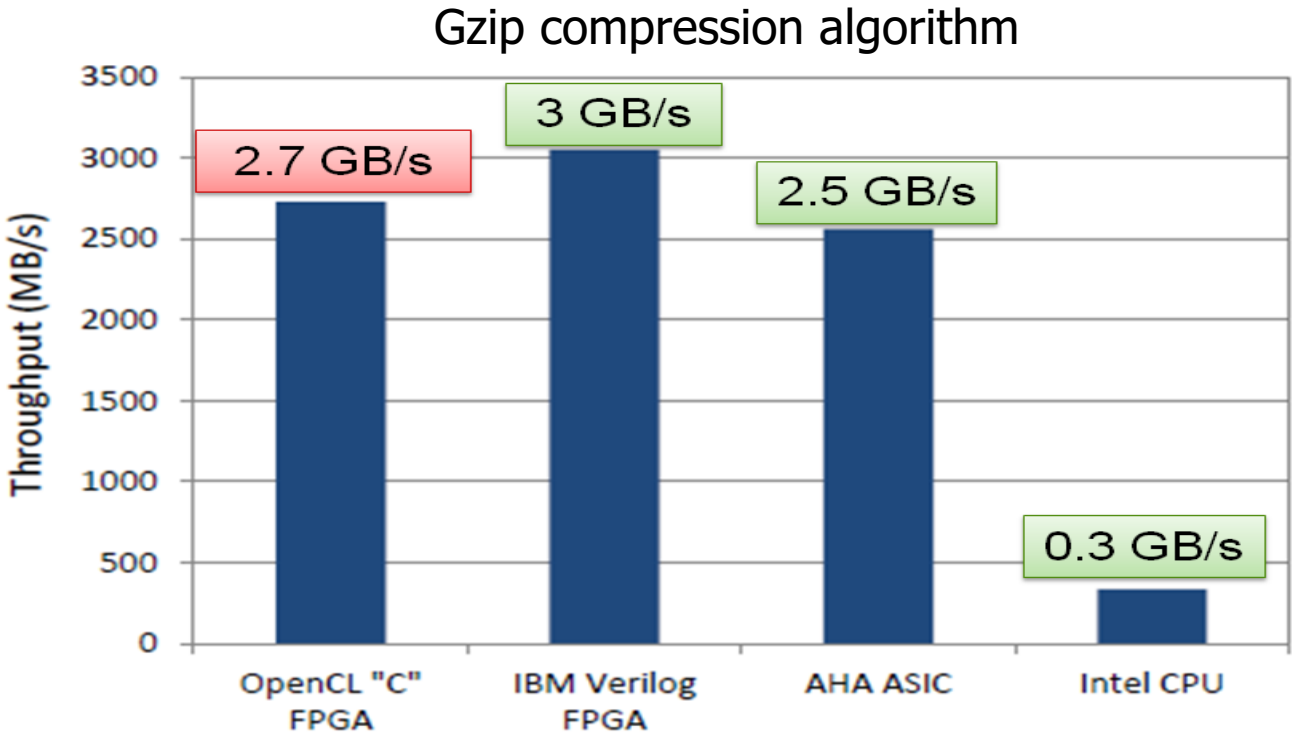
Figures courtesy of Gordon Chiu

FPGAs well positioned for deep learning

M20K: A 2.5kB SRAM memory block.

spad: Scratch pad

Do you need Verilog or VHDL to use an FPGA?
No … OpenCL will do.



OpenCL Was

10% Slower

12% more resources

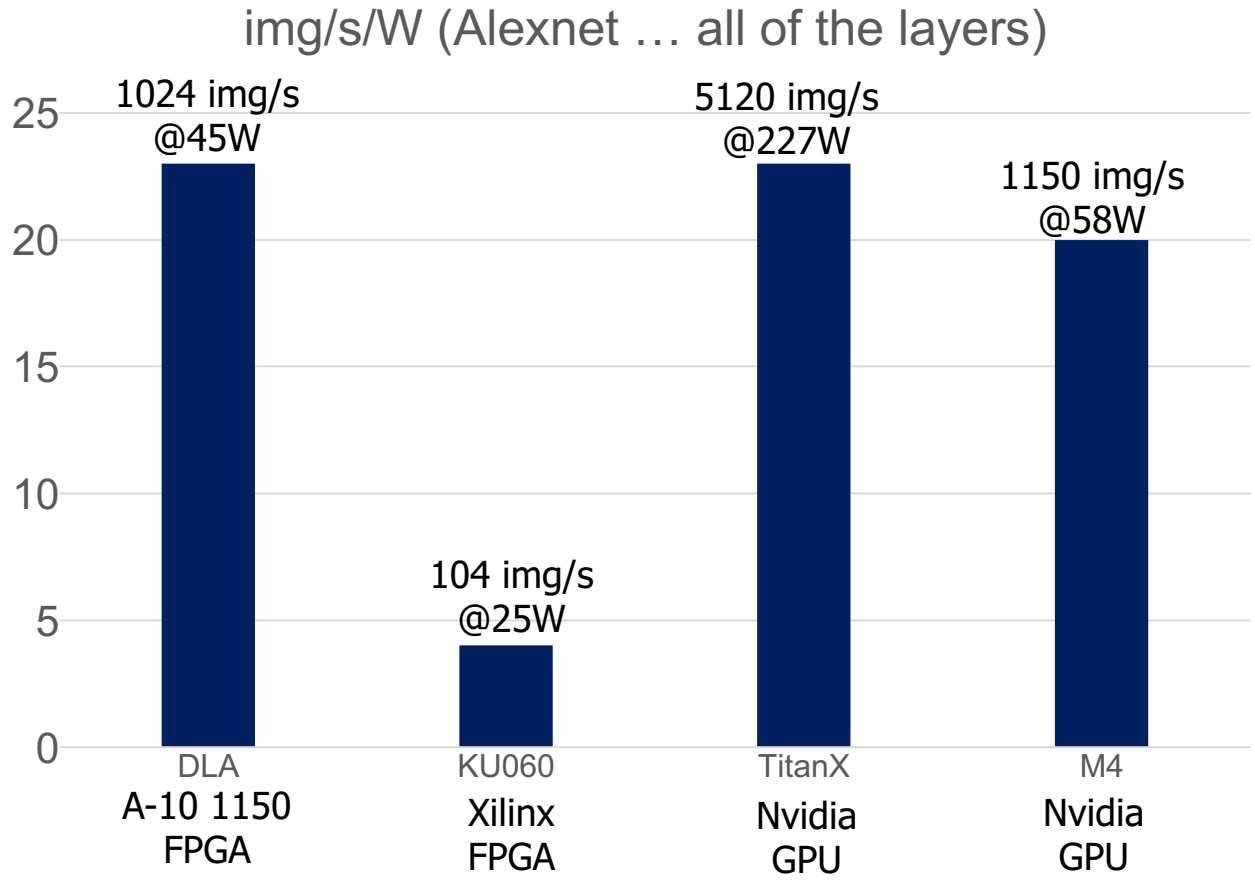
3x faster development time

- Altera summer intern ported and optimized GZIP algorithm in less than a month
- Industry leading companies FPGA engineer coded Verilog in 3 months

*Source: http://www.eecg.utoronto.ca/~mohamed/iwocl14.pdf

An OpenCL™ Deep Learning Accelerator on Arria 10

- **An OpenCL™ Deep Learning Accelerator on Arria 10**
- Utku Aydonat, Shane O'Connell, Davor Capalija, Andrew C. Ling, Gordon R. Chiu DOI: http://dx.doi.org/10.1145/3020078.3021738



This shows you can program an FPGA with OpenCL and get good results.
I do not like comparing to NVIDIA or Xilinx or anyone else …. I include those number to show that the FPGA/OCL results are reasonably good compared to competitors.

Third Party Names are the property of their owners.

Comparing FPGAs and GPUs

- FPGAs and GPU under study

	Arria 10 1150 FPGA	Stratix 10 2800 FPGA	TitanX Pascal GPU
Peak FP32 TFLOPs	1.36	9.2	11
On-chip RAMs	6.6 MB (M20Ks)	28.6 MB (M20Ks)	13.5 MB (RF, SM, L2)
Memory BW	Assume same as Titan X	Assume same as Titan X	480 GB/s

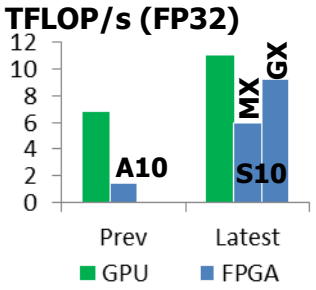
- Evaluation
  - GPU: used known library (cuBLAS) or framework (Torch with cuDNN)
  - FPGA: Statix 10 numbers are estimated using Quartus and PowerPlay

Source: Can FPGAs Beat GPUs in Accelerating Next-Generation Deep Neural Networks? E. Nurvitadhi, G. Venkatesh, J. Sim, D. Marr, R. Huang, J. G. H. Ong, Y. T. Liew, K. Srivatsan, D. Moss, S. Subhaschandra, G. Boudoukh, FPGA'17, Feb 2017

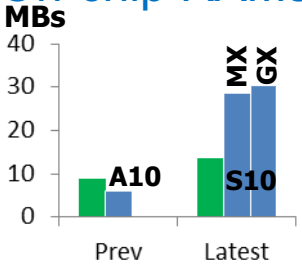
Third party names are the property of their owners.

FPGAs are becoming much more capable

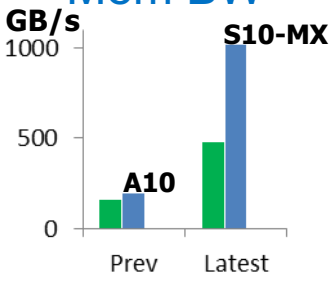
FLOPs



On-chip RAMs



Mem BW



Higher Freq

HyperFlex™ ARCHITECTURE

2x core frequency

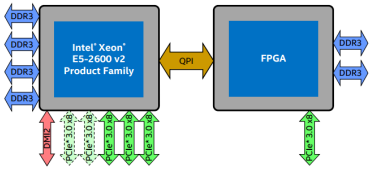
Power Efficient

10s-100s W

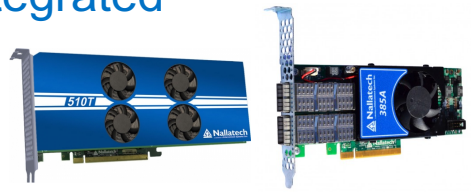
High-Level Programming

A++

More Integrated



Xeon+FPGA

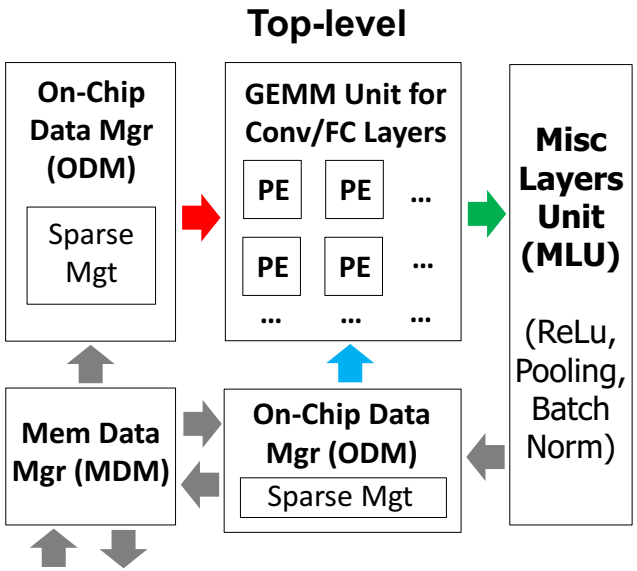


Discrete cards

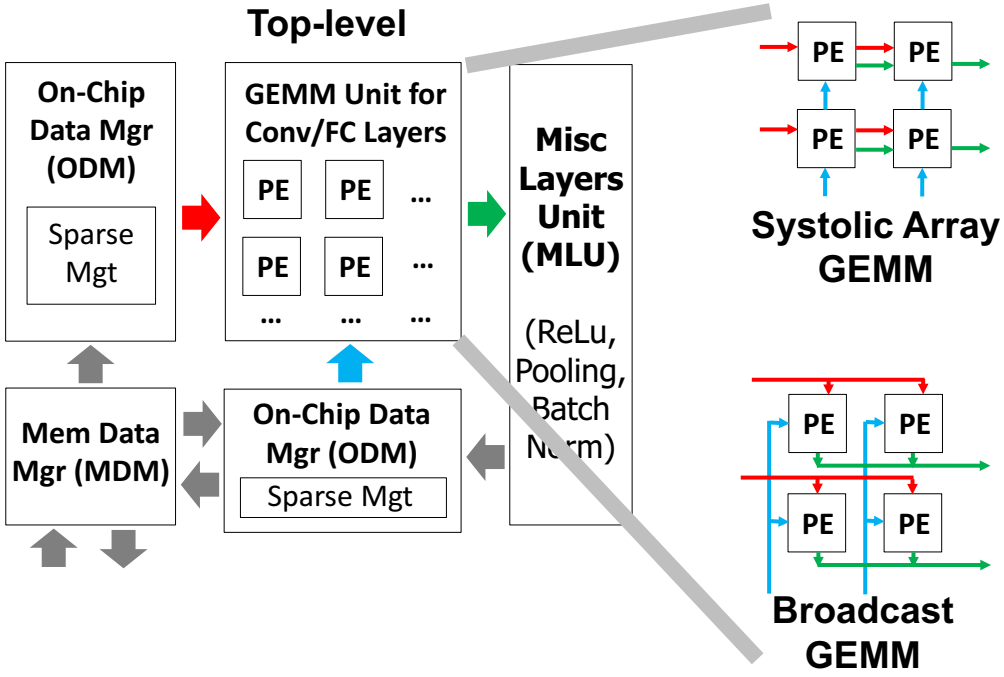
Upcoming Stratix 10 will be more competitive to GPUs

All Stratix 10 numbers are estimated. GPU and A10 numbers are from actual computations.

DNN accelerator template for FPGA used in our studies

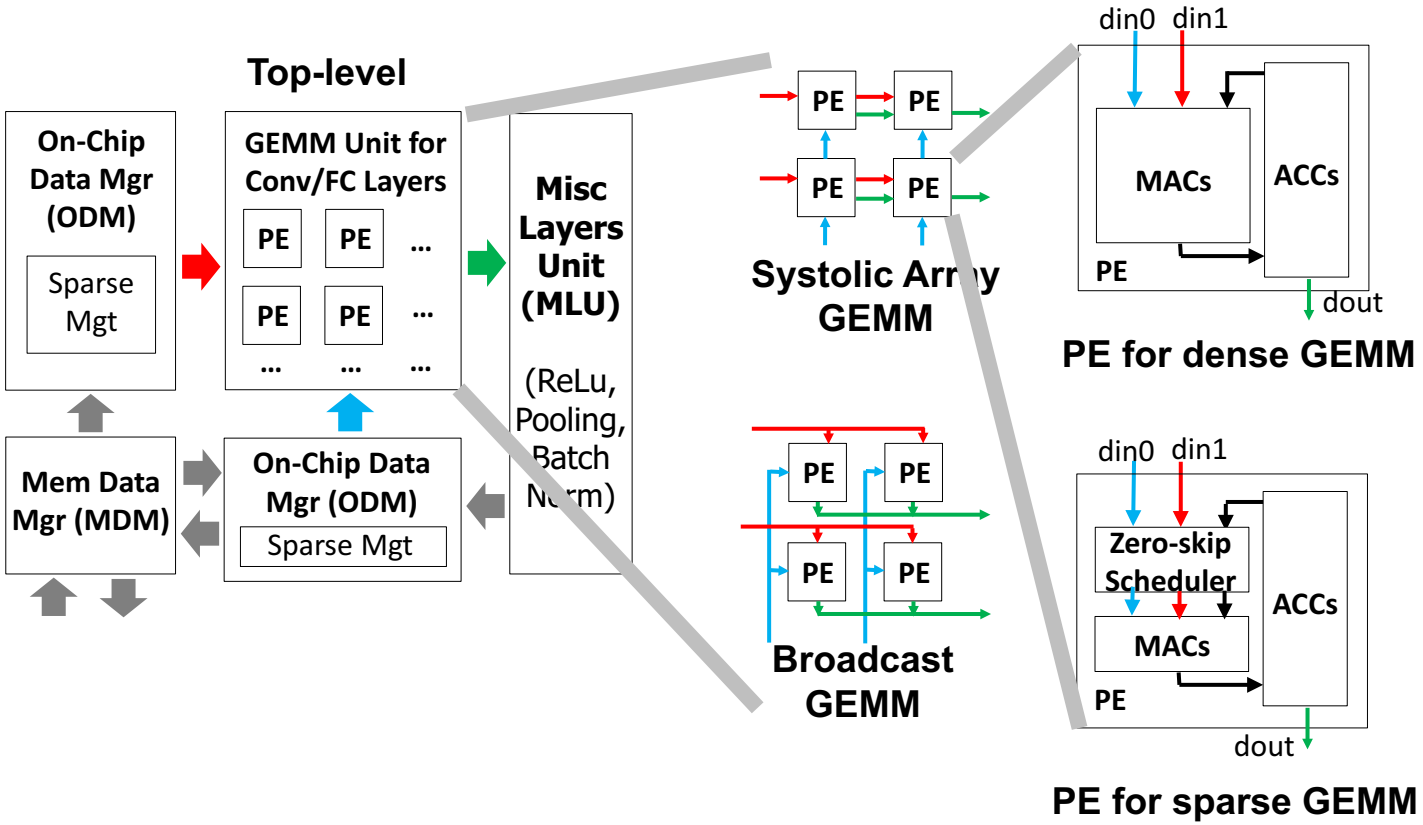


DNN accelerator template for FPGA used in our studies



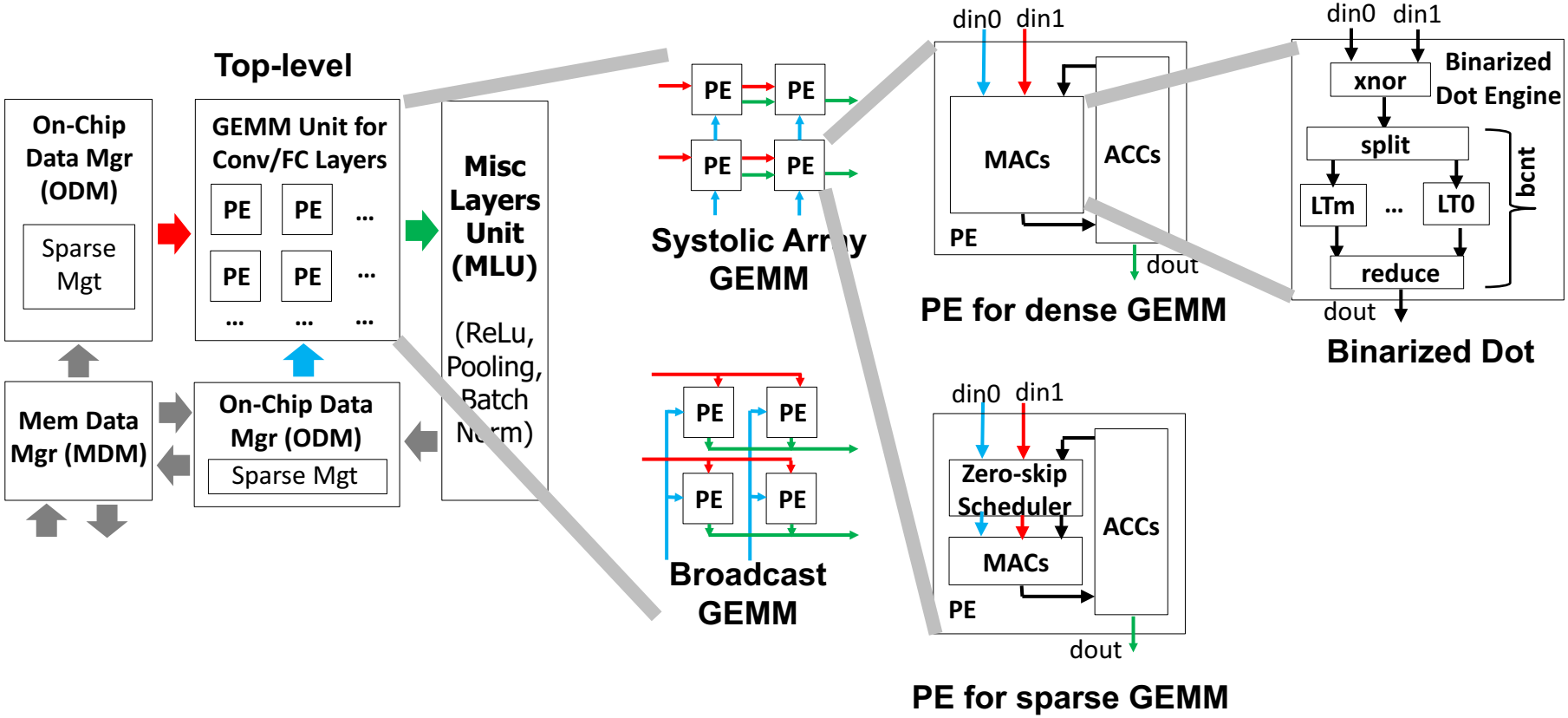
Allows studying various design instances

DNN accelerator template for FPGA used in our studies



Allows studying various design instances

DNN accelerator template for FPGA used in our studies



Allows studying various design instances

Only use the bits you need: Matrix Multiplies used in binarized DNN

Neural networks with parameters of +1 or -1

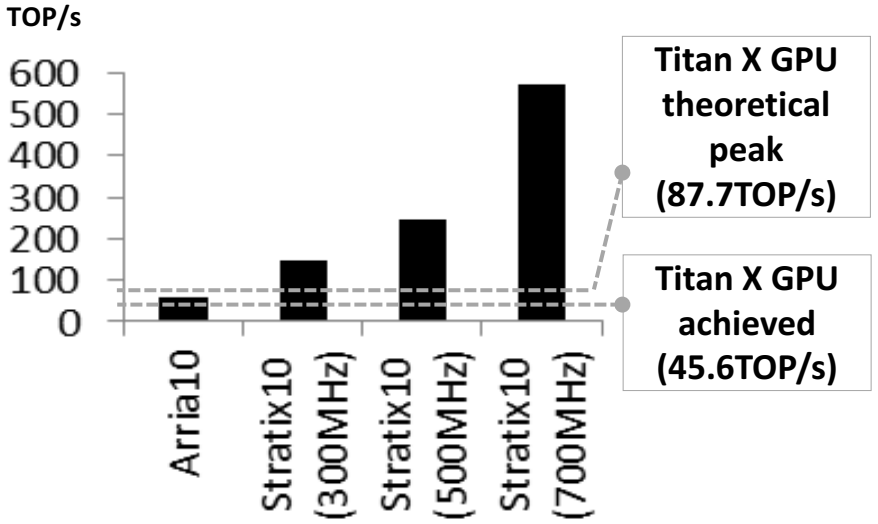
Matrix x Vector, with +1 or -1

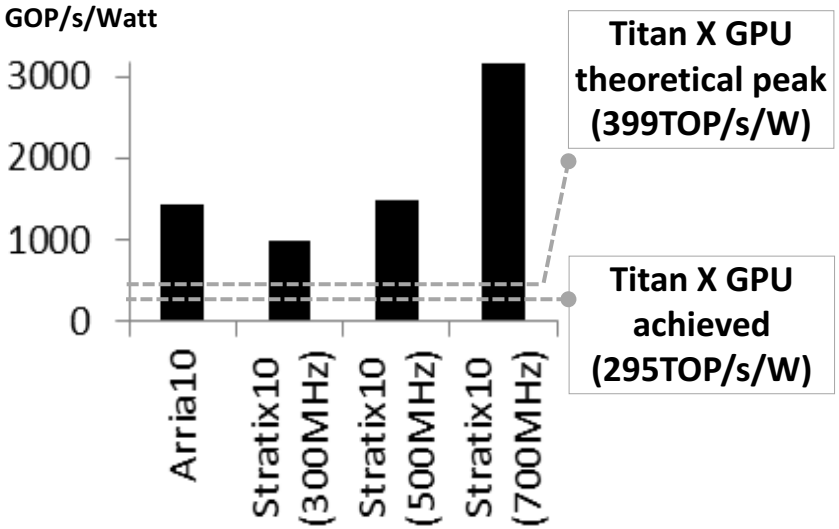
I		W				O
-1		-1	+1	+1	(-1.-1)+(1.1)+(1.1)	3
+1	x	-1	-1	-1	= (-1.-1)+(1.-1)+(1.-1)	= -1
+1		+1	+1	-1	(-1.1)+(1.1)+(1.-1)	-1

Binarized Matrix x Vector

I		W				O
0		0	1	1	bcnt(xnor(011,011))	3
1	x	0	0	0	= bcnt(xnor(011,000))	= -1
1		1	1	0	bcnt(xnor(011,110))	-1

Used optimized GPU code and FPGA design from [FPT'16]





S10 FPGA can offer significantly better performance than Titan X GPU

Third party names are the property of their owners.

All Stratix 10 numbers are estimated. GPU and A10 numbers are from actual computations.

Only use the bits you need: sometimes you need three

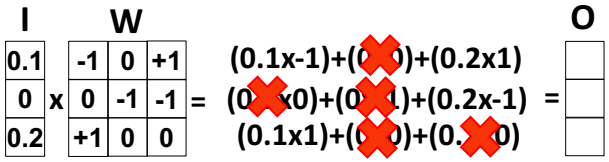
Ternary NN: neural net with weights of +1,-1,0

I		W					O
0.1		-1	0	+1		(0.1x-1)+(0x0)+(0.2x1)	
0	x	0	-1	-1	=	(0.1x0)+(0x-1)+(0.2x-1)	=
0.2		+1	0	0		(0.1x1)+(0x0)+(0.2x0)	

TNN: weights use 2 bits, neurons use FP32

Only use the bits you need: sometimes you need three

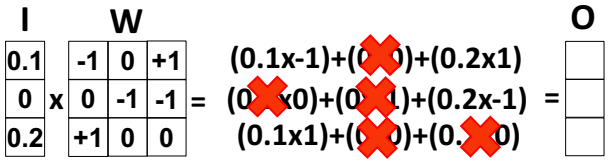
Ternary NN: neural net with weights of +1,-1,0

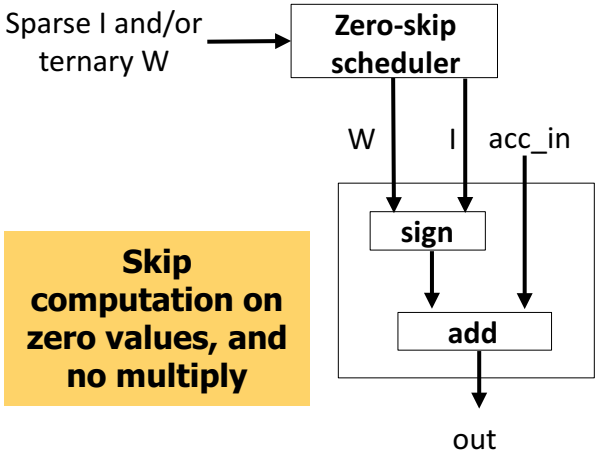


TNN: weights use 2 bits, neurons use FP32

Only use the bits you need: sometimes you need three

Ternary NN: neural net with weights of +1,-1,0



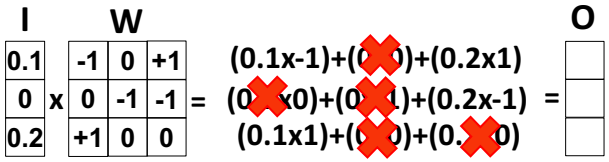


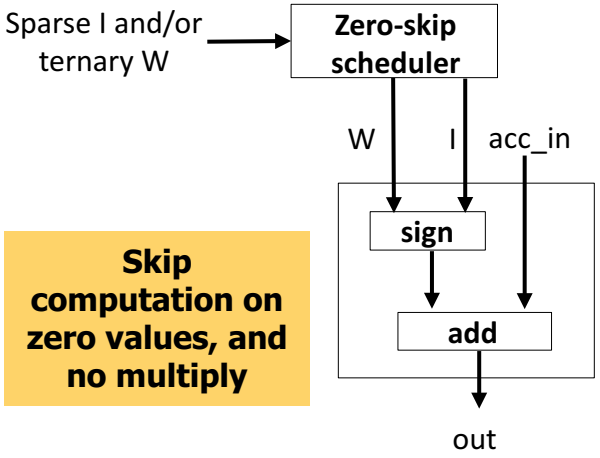
Skip computation on zero values, and no multiply

TNN: weights use 2 bits, neurons use FP32

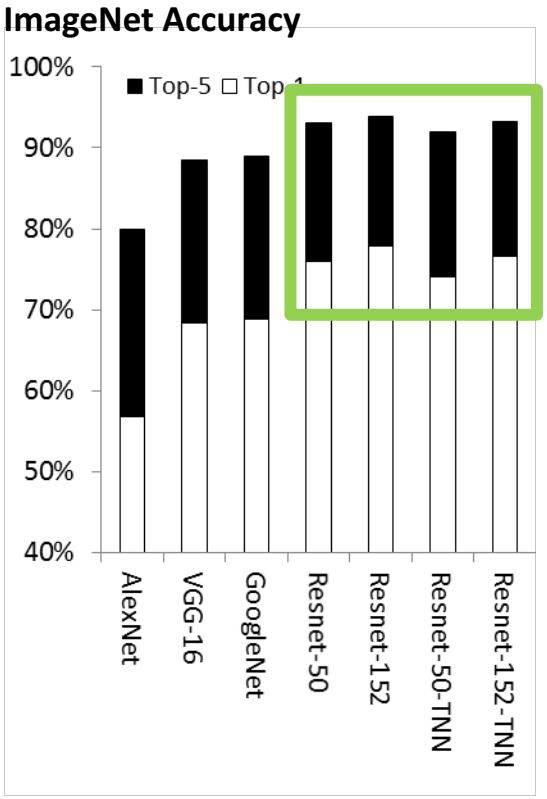
Only use the bits you need: sometimes you need three

Ternary NN: neural net with weights of +1,-1,0





Skip computation on zero values, and no multiply

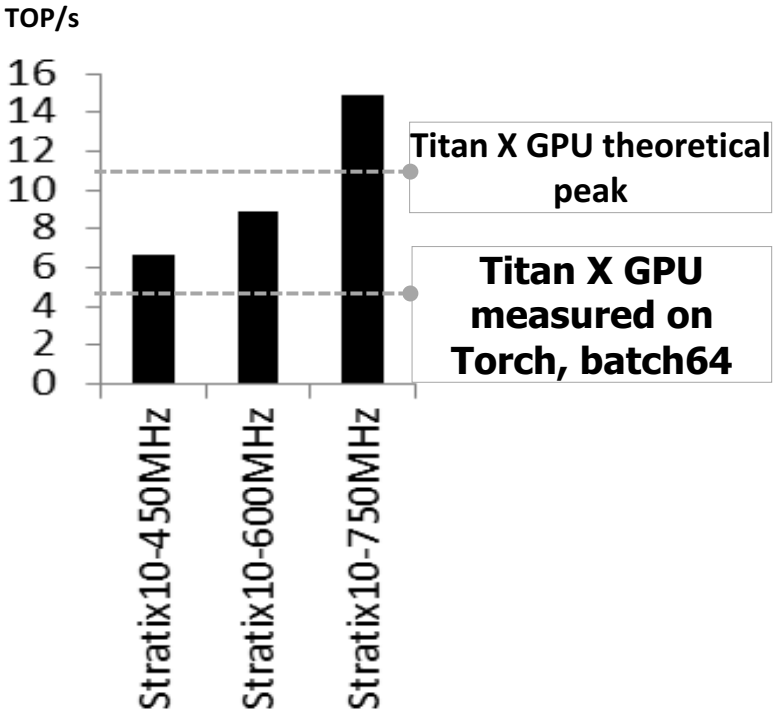


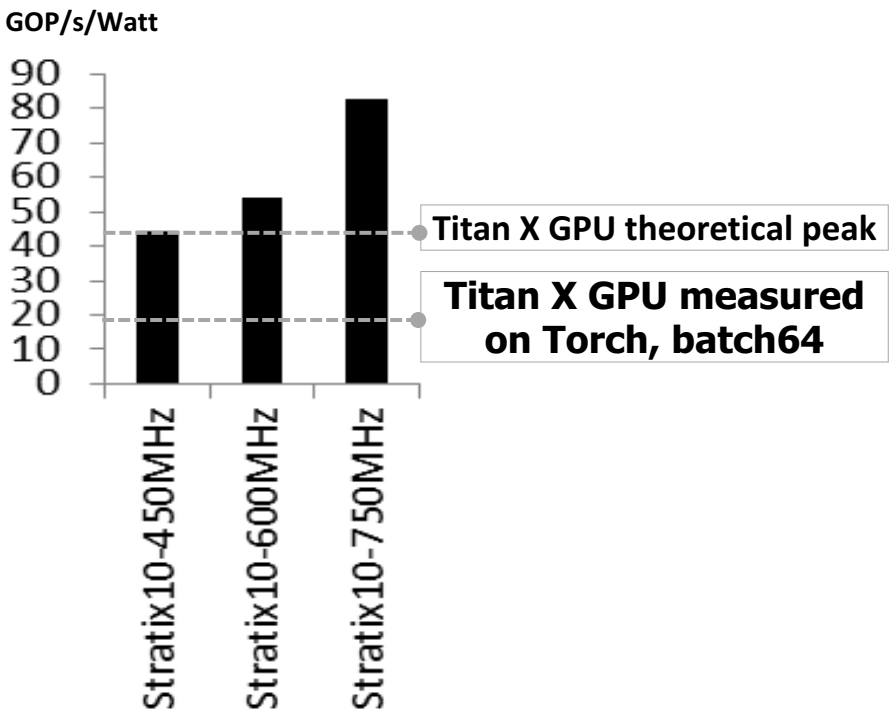
Ternary ResNet offers state-of-the-art accuracy [ICASSP'17]

We target Resnet-50-TNN in this study

TNN: weights use 2 bits, neurons use FP32

Results (ResNet-50 for ImageNet problem sizes)





Stratix 10 numbers are estimated. GPU results are for aggressively optimized code and represent the best numbers among all tried configurations (and exceed published numbers on this benchmark)

S10 FPGA performs better, across all frequency targets

Third party names are the property of their owners.

FPGA for AI: moving into production

Microsoft Brainwave Demo at Hot Chips 2017

- Brainwave … uses Microsoft's Cloud-based FPGA infrastructure:
- Common DNN techniques (batching, small dense mat-muls) don't work for latency sensitive, real time AI with complex networks (e.g. GRU or LSTM) used in NLP.

- Running on early Stratix 10 silicon:
  - Achieved record-setting performance with a GRU model, 5X larger than Resnet-50.
  - 8-bit floating point format with no accuracy losses (on average) across a range of models.
  - sustained 39.5 Teraflops on this large GRU. Each request ran in under one millisecond.

Intel's 14 nm Stratix 10 FPGA

"Running on Stratix 10, Project Brainwave thus achieves unprecedented levels of demonstrated real-time AI performance on extremely challenging models. As we tune the system over the next few quarters, we expect significant further performance improvements."

Source: Doug Burger of Microsoft: https://www.microsoft.com/en-us/research/blog/microsoft-unveils-project-brainwave/, 8/22/2017

Conclusion

- It is a confusing world out there … we are all running as fast as we can just to keep up.
  - Processor architecture is evolving rapidly … SW people struggle to adapt.
  - Algorithms for DL are changing quickly … Timescale of HW evolution is too slow to keep up.
- Accelerators (TPU, Myriad™ X, Crest family, etc) just make it worse
- The jury is still out, but maybe FPGAs are the answer?

Performance Drivers for AI Workloads

Compute

Higher number of operations per second

Intel® Xeon® Platinum 8180 Processor (1-socket)

- up to 3570 GFLOPS on SGEMM (FP32)
- up to 5185 GOPS on IGEMM (Int8)

Increased parallelism and vectorization

Intel® Xeon® Scalable Processor offers Intel® AVX-512 with up to 2 512bit FMA units computing in parallel per core [1]

Higher number of cores

Up to 28 core Intel® Xeon® Scalable Processors

Bandwidth

High Throughput, Low Latency

- Intel® Xeon® Scalable Processors offer up to 6 DDR4 channels per socket and new mesh architecture
- Intel® Xeon® Processor 8180 Up to 199GB/s of STREAM Triad performance on a 2 socket system

Efficient Large Sized Caches

Intel® Xeon® Scalable Processors offer increased private local Mid-Level Cache MLC up to 1MB per core

1 Available on Intel® Xeon® Platinum Processors, Intel® Xeon® Gold Processors and Intel® Xeon® 5122 Processor

Configuration Details on Slide: 23 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.

AI performance Gen over Gen Configuration Details

- 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“.
- 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.

AI performance Gen over Gen Configuration Details

- Platform: 2S Intel® Xeon® CPU E5-2699 v4 @ 2.20GHz (22 cores), HT enabled, turbo disabled, scaling governor set to “performance” via acpi-cpufreq driver, 256GB DDR4-2133 ECC RAM. CentOS Linux release 7.3.1611 (Core), Linux kernel 3.10.0-514.10.2.el7.x86_64. SSD: Intel® SSD DC S3500 Series (480GB, 2.5in SATA 6Gb/s, 20nm, MLC).
- **Performance measured with**: Environment variables: KMP_AFFINITY='granularity=fine, compact,1,0‘, OMP_NUM_THREADS=44, CPU Freq set with cpupower frequency-set -d 2.2G -u 2.2G -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). GCC 4.8.5, Intel MKL small libraries version 2017.0.2.20170110.
- 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 44, 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 e9f281a27584cdb78db8ce6b66e648b3dbc10d37. 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 2017.0.2.20170110.
- 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.