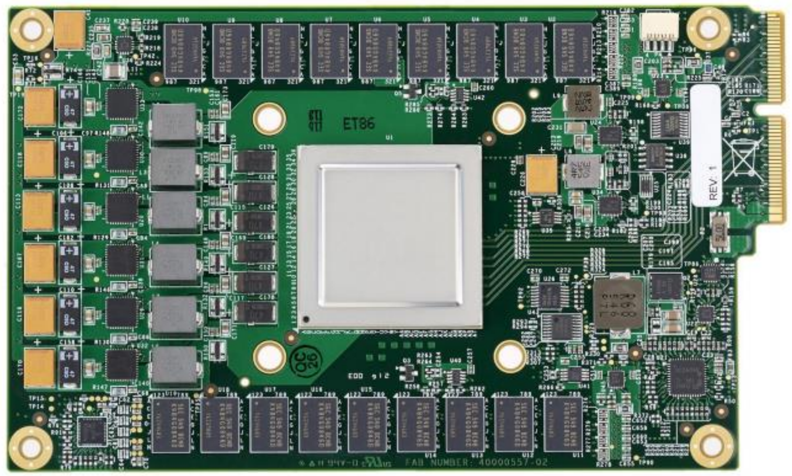
Application-Specific Hardware

…in the real world

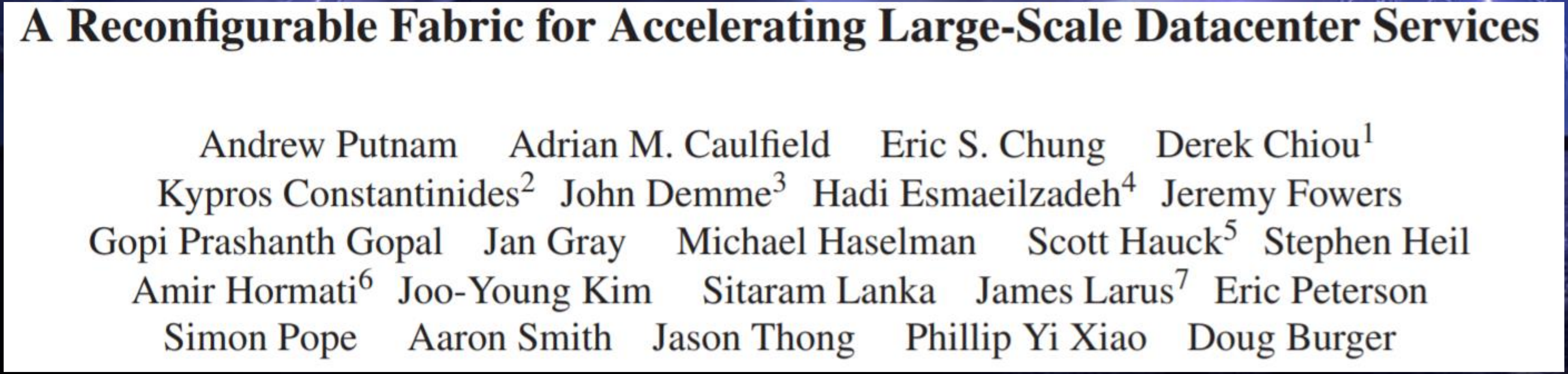
Microsoft

http://warfarehistorynetwork.com/wp-content/uploads/Military-Weapons-the-Catapult.jpg

Google



Large-Scale Reconfigurable Computing in a Microsoft Datacenter



A Reconfigurable Fabric for Accelerating Large-Scale Datacenter Services

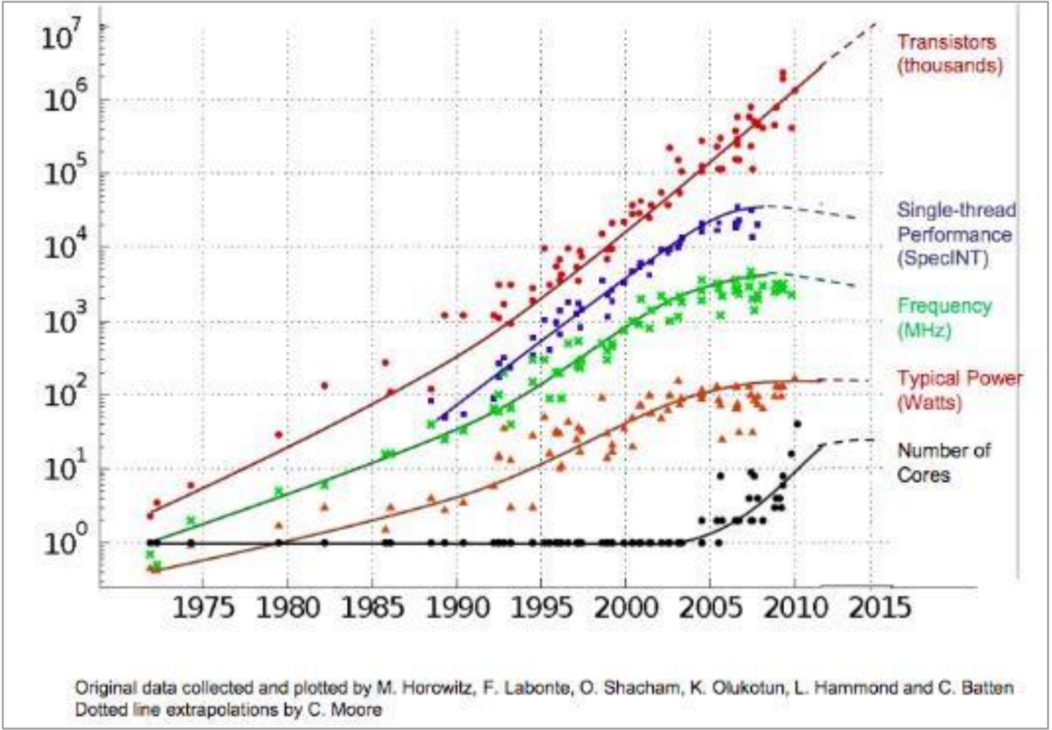
Andrew Putnam Adrian M. Caulfield Eric S. Chung Derek Chiou[1]
Kypros Constantinides[2] John Demme[3] Hadi Esmaeilzadeh[4] Jeremy Fowers
Gopi Prashanth Gopal Jan Gray Michael Haselman Scott Hauck[5] Stephen Heil
Amir Hormati[6] Joo-Young Kim Sitaram Lanka James Larus[7] Eric Peterson
Simon Pope Aaron Smith Jason Thong Phillip Yi Xiao Doug Burger

Andrew Putnam Microsoft

ISCA’14

Microsoft Cloud Services

Windows Azure
bing
Microsoft SQL Server
XBOX LIVE
COSMOS
skype
msn
OneDrive
Outlook.com
Microsoft Office 365



Original data collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond and C. Batten
Dotted line extrapolations by C. Moore

$$\text{Capabilities} \propto \frac{\boldsymbol{\mathit{Performance/Watt}}}{\$}$$

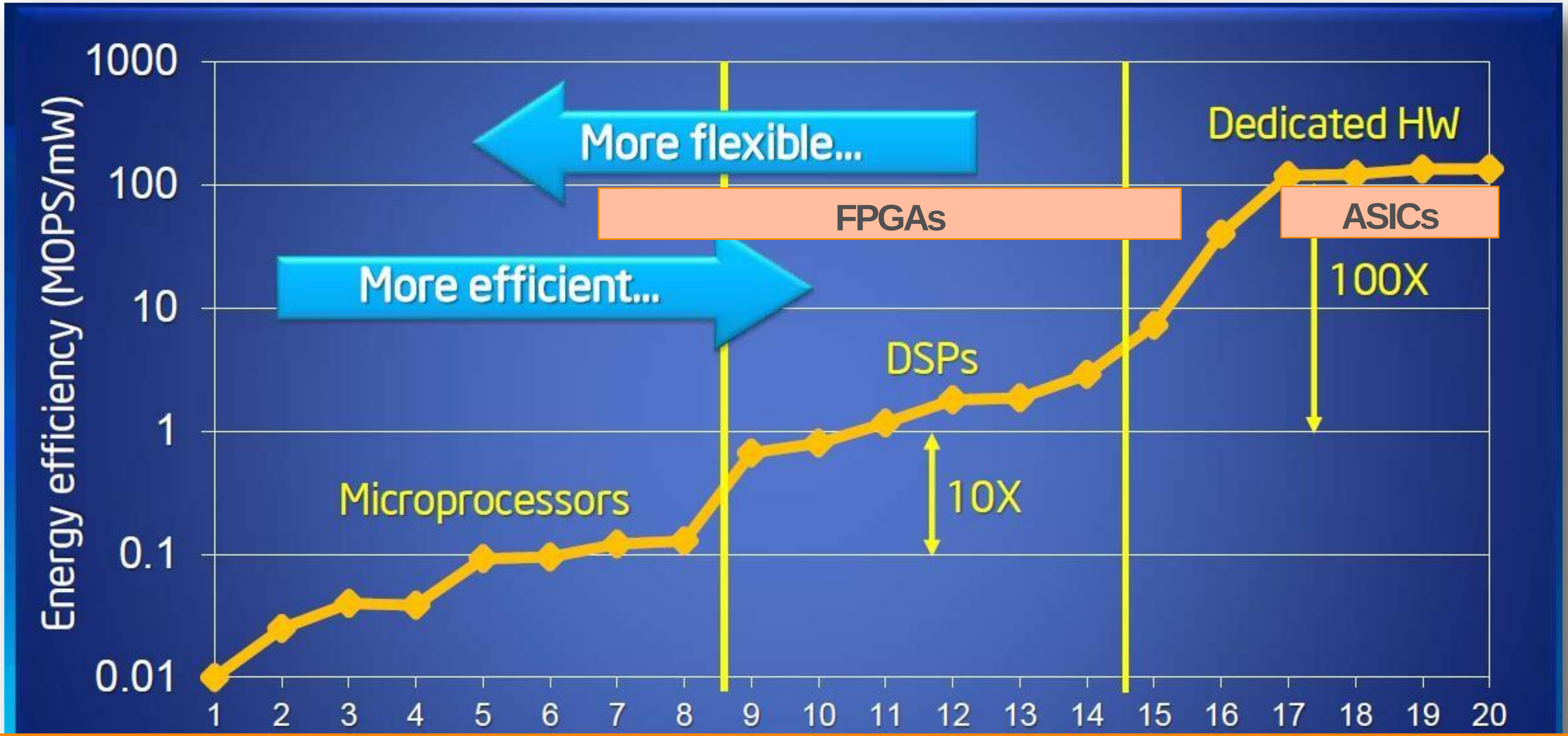
Datacenter Environment

- Software services change monthly
- Machines last 3 years, purchased on a rolling basis
- Machines repurposed ~½ way into lifecycle
- Little/no HW maintenance, no accessibility

- Homogeneity is highly desirable

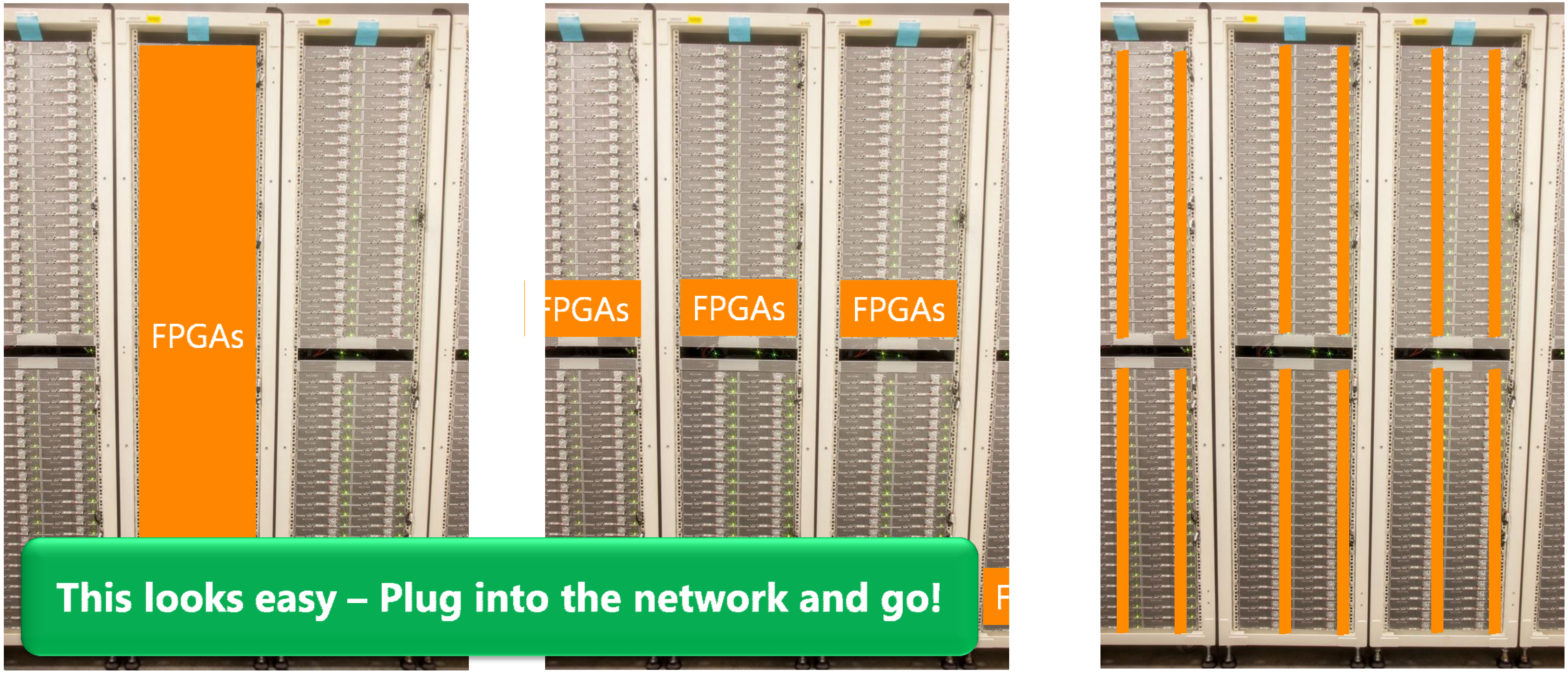
The paradox: Specialization *and* homogeneity

Efficiency via Specialization



Increase Efficiency with Hardware Specialization

Integrating FPGAs into the Datacenter

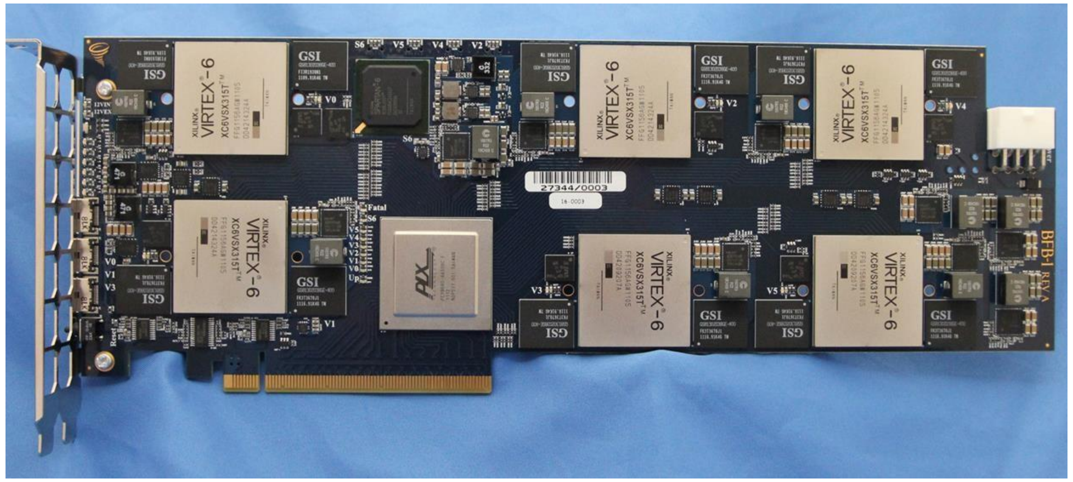


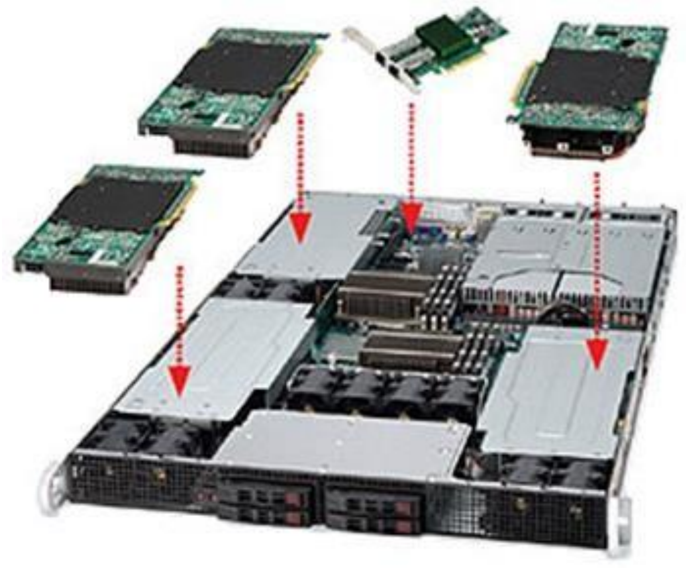
This looks easy – Plug into the network and go!

Centralized **Distributed**

Prototype #1: BFB board

- Prototyped a 6 FPGA board
- 3x2 GPIO mesh
- PCIe connecting all FPGAs, CPU
- Plugs into Supermicro GPU server
- Serves L2 scoring for 48 server pod

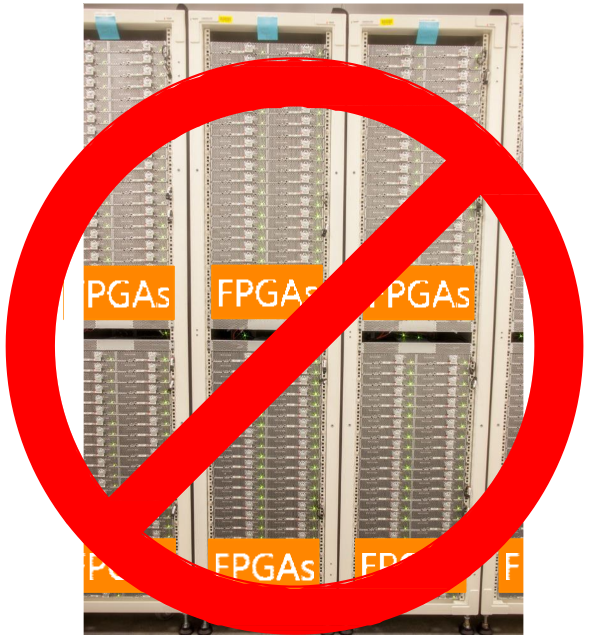




- 1U, 2U, or 4U rack-mounted
- 1/2/4 x 10Ge ports
- Up to 4 PCIe x16 slots
- 2 sockets, 6-core Intel Westmere

Centralized Model Unsuitable for Datacenter

- Single point of failure
- Complicates rack design, thermals, maintainability
- Network communication for any use of FPGA
  - Definition of the Network In cast problem
  - Precludes many latency sensitive workloads
- Limited elasticity
  - What if you need more than six FPGAs?



Our Design Requirements

Don’t Cost Too Much

<30% Cost of Current Servers

1. Specialize HW with an FPGA Fabric
2. Keep Servers Homogeneous

Don’t Burn Too Much Power

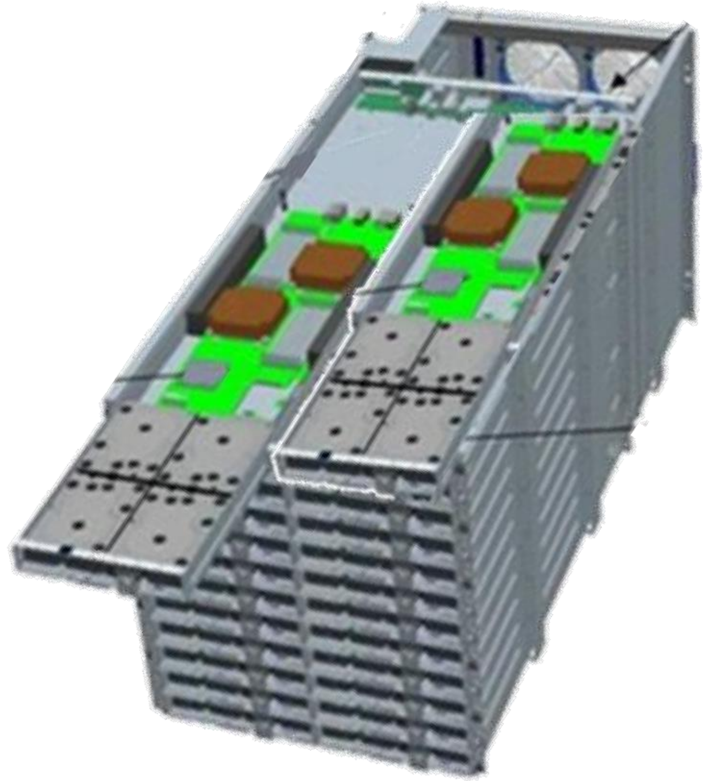
<10% Power Draw
(25W max, all from PCIe)

Don’t Break Anything

Work in existing servers
No Network Modifications
Do not increase hardware failure rate

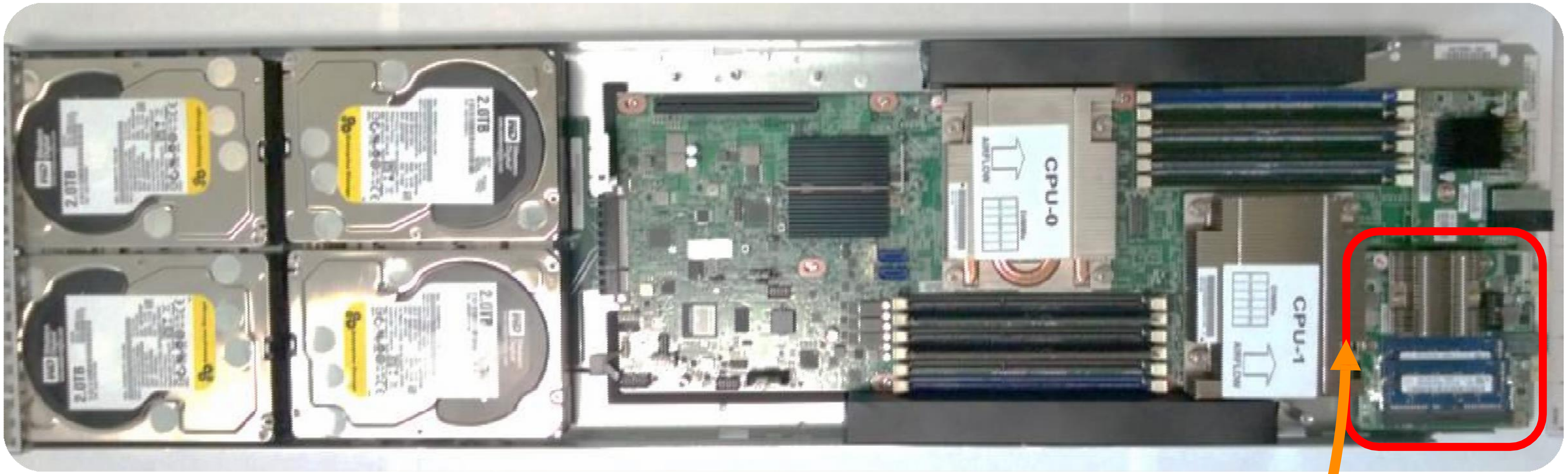
Datacenter Servers

- Microsoft Open Compute Server
- 1U, ½ wide servers
- Enough space & power for ½ height, ½ length PCIe card
- Squeeze in a single FPGA
- Won't fit (or power) GPU

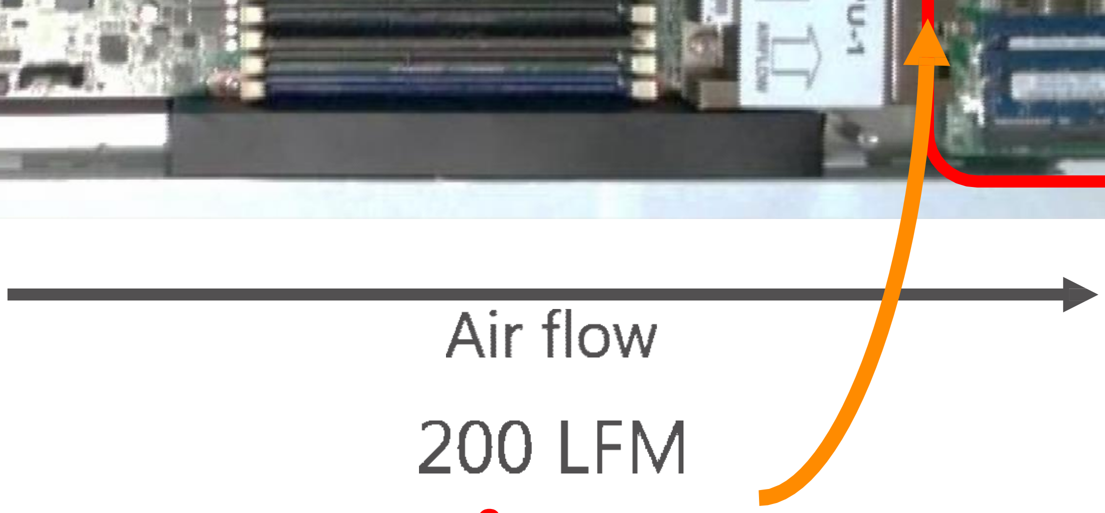


http://www.globalfoundationservices.com/posts/2014/january/27/microsoft-contributes-cloud-server-specification-to-open-compute-project.aspx

Microsoft Open Compute Server



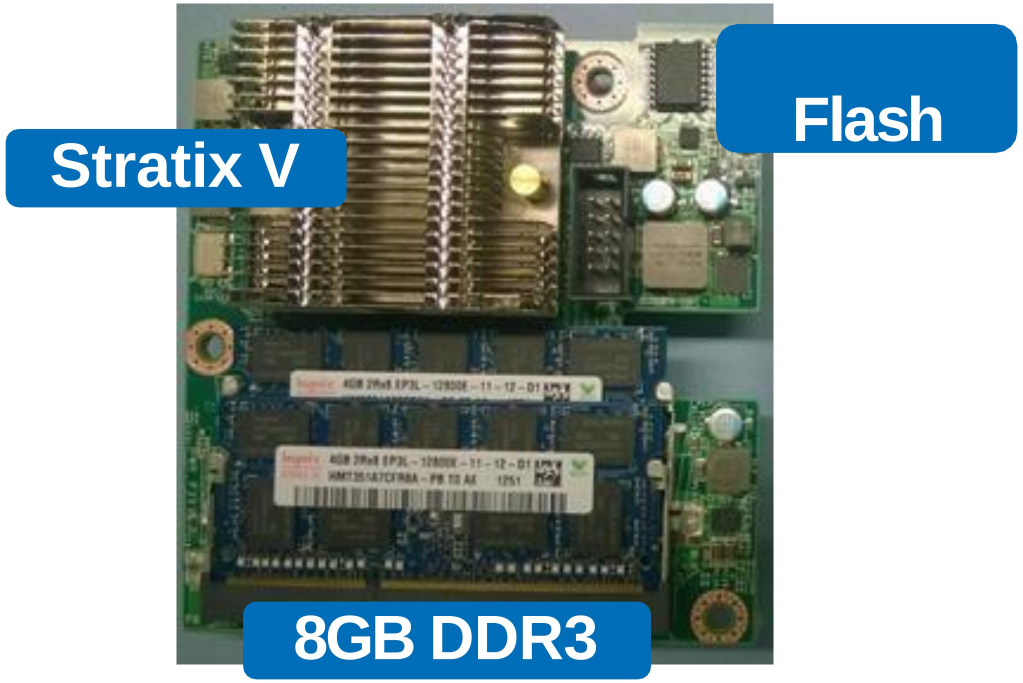
- Two 8-core Xeon 2.1 GHz CPUs
- 64 GB DRAM
- 4 HDDs @ 2 TB, 2 SSDs @ 512 GB
- 10 Gb Ethernet
- No cable attachments to server

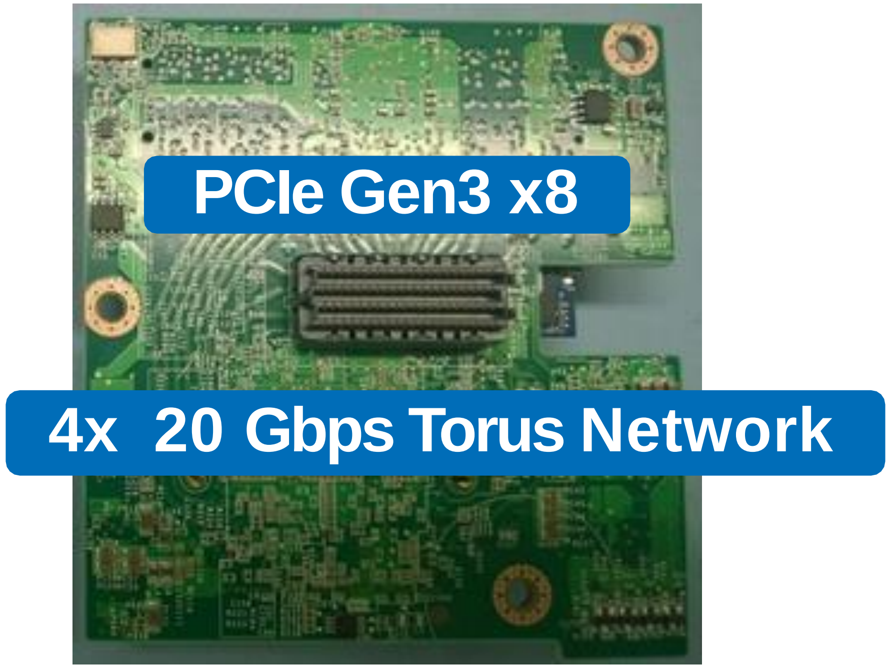


68 ^{o}C Inlet

Catapult FPGA Accelerator Card

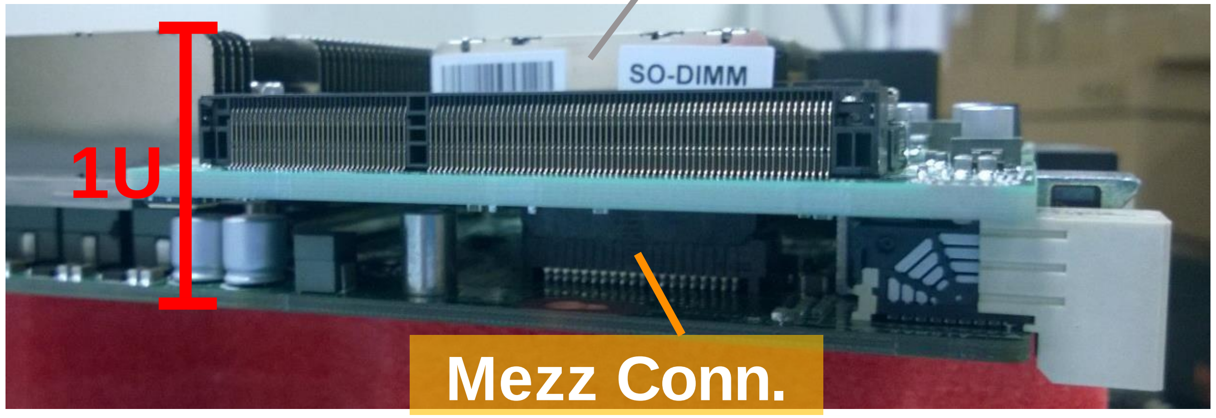
- **Altera Stratix V GS D5**
  - **172k ALMs, 2,014 M20Ks, 1,590 DSPs**
- **8GB DDR3-1333**
- **32 MB Configuration Flash**
- **PCIe Gen 3 x8**
- **8 lanes to Mini-SAS SFF-8088 connectors**
- **Powered by PCIe slot**

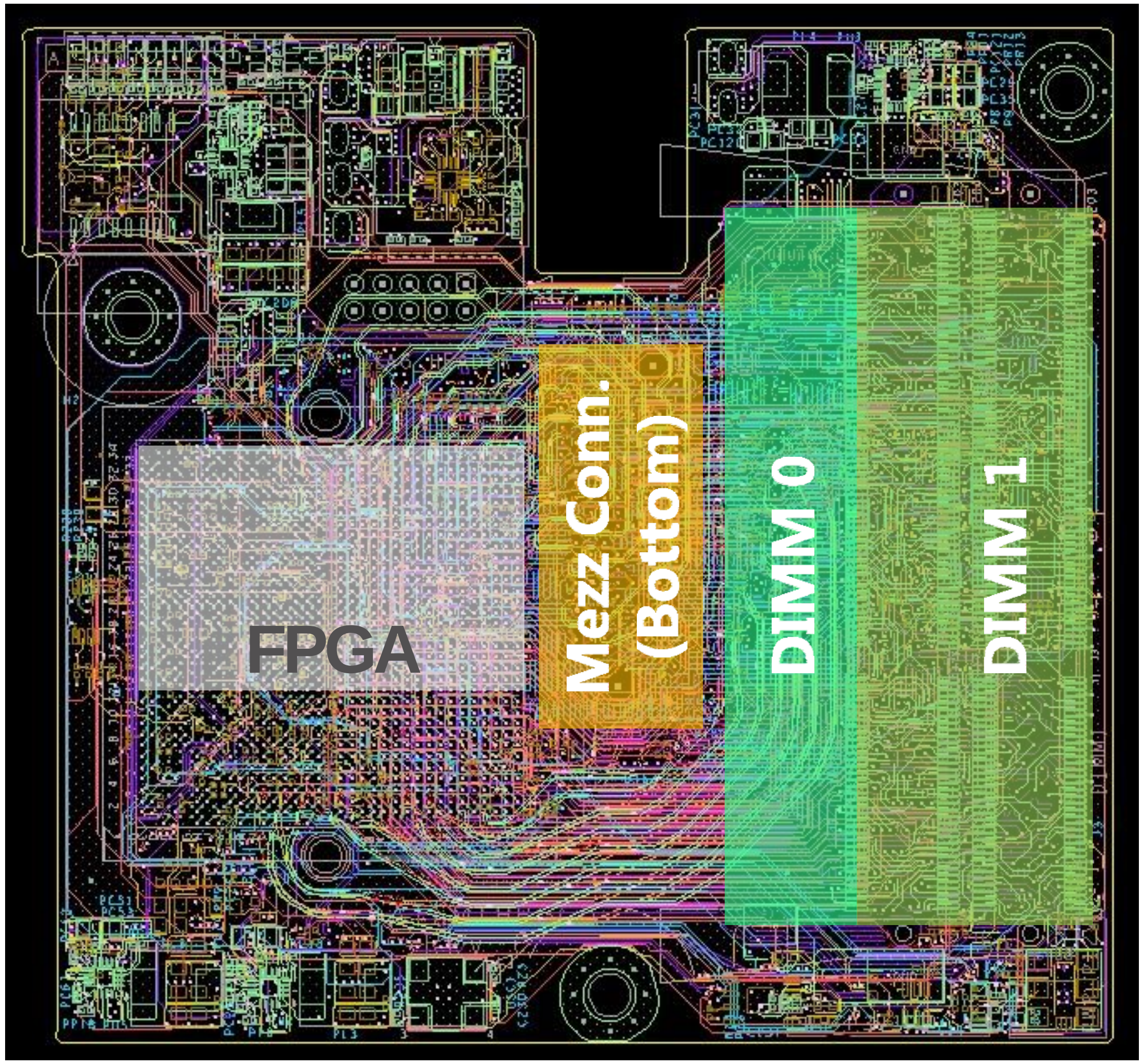




Board Details

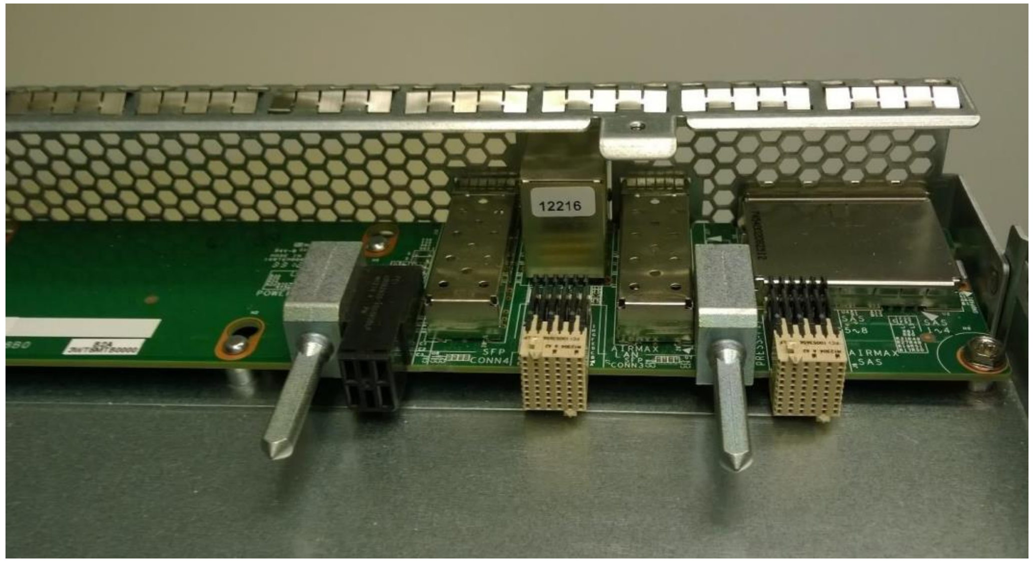
- 16 Layer, FR408
- 9.5cm x 8.8cm x 115.8 mil
- 35mm x 35mm FPGA
- 14.2mm high heatsink



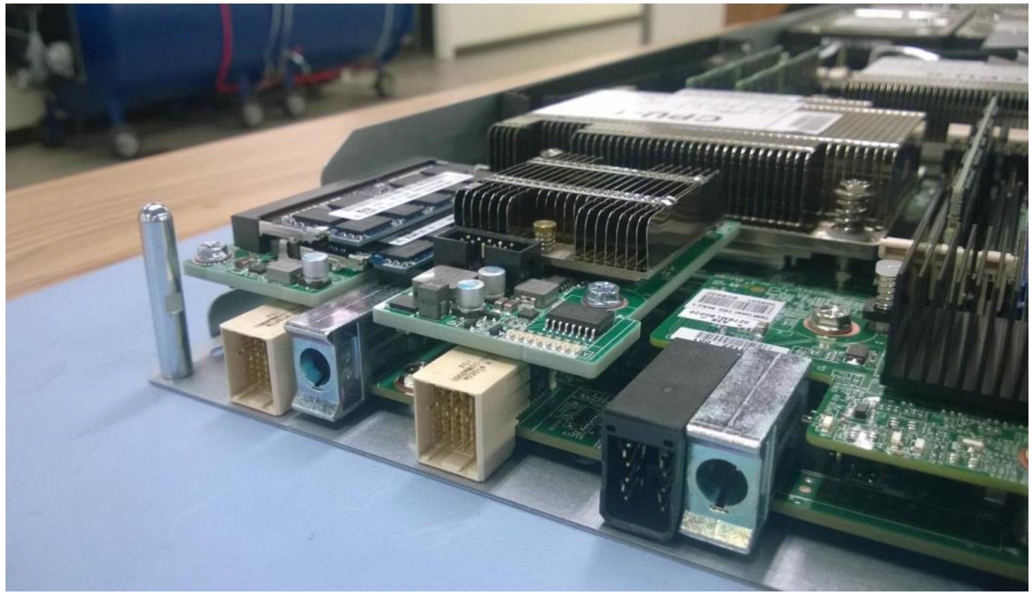


Board / Server Integration

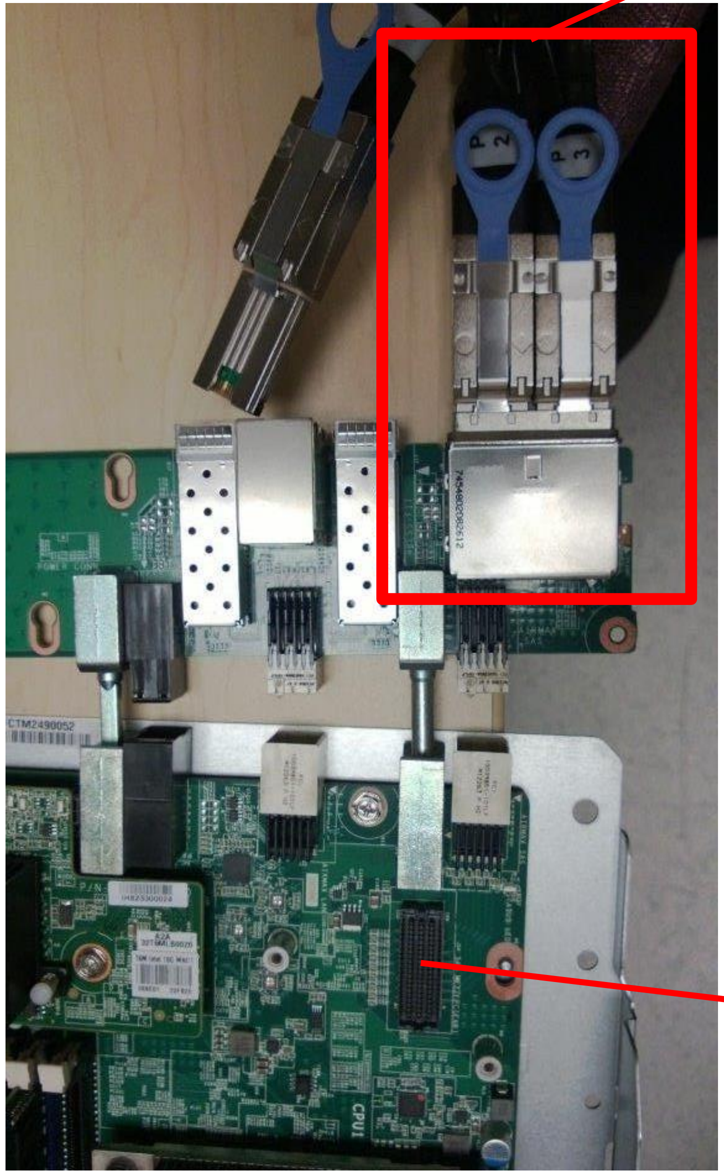
I/O Backplane



Server w/ Catapult Board



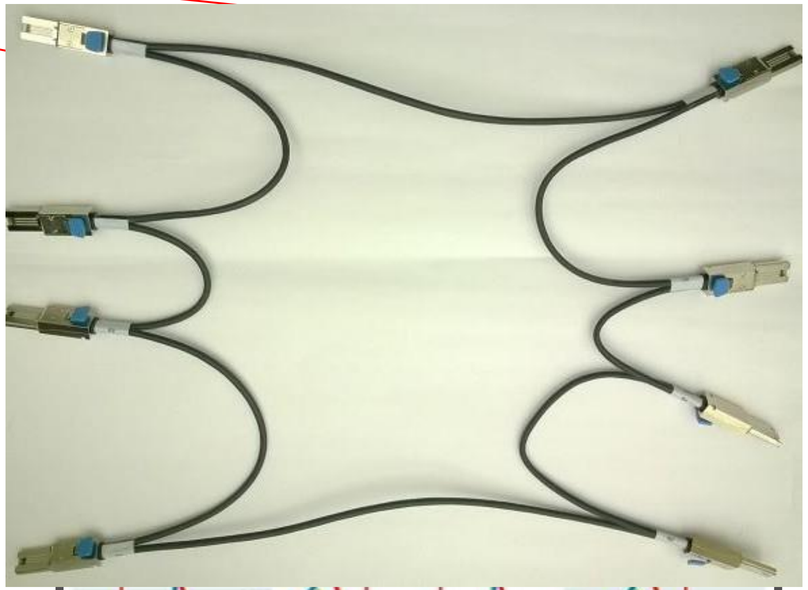
Catapult Network Cables
(Mini SAS / SFF 8088)

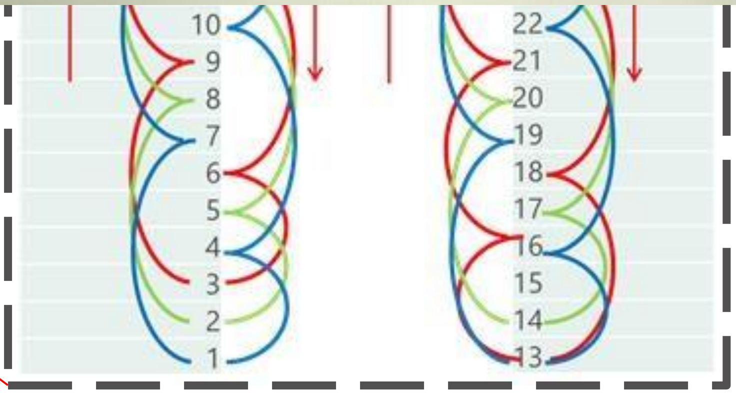


Boards Connected Together

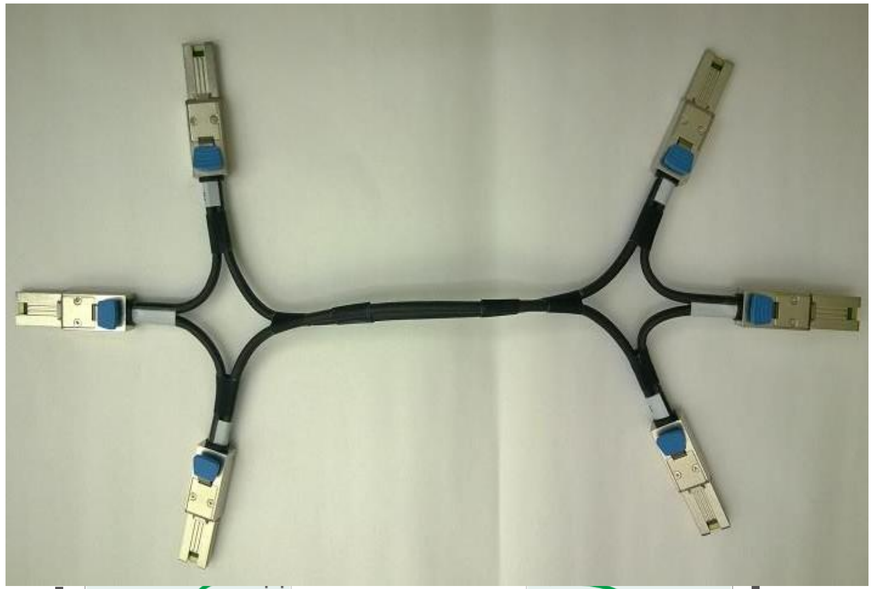
Catapult Board
Mezzanine Slot

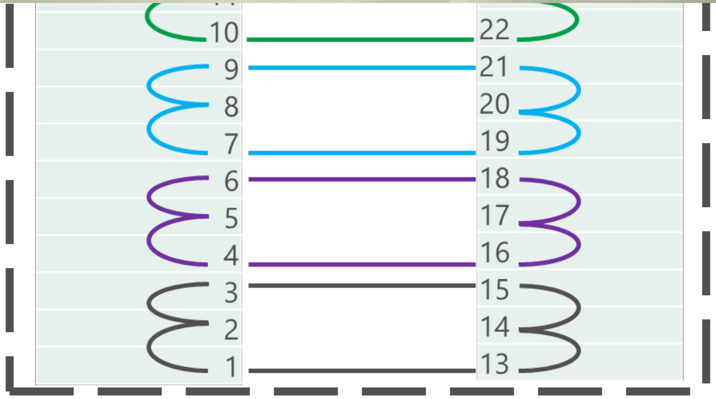
6x8 Torus in a 2x24 Server Layout





8 shell cables

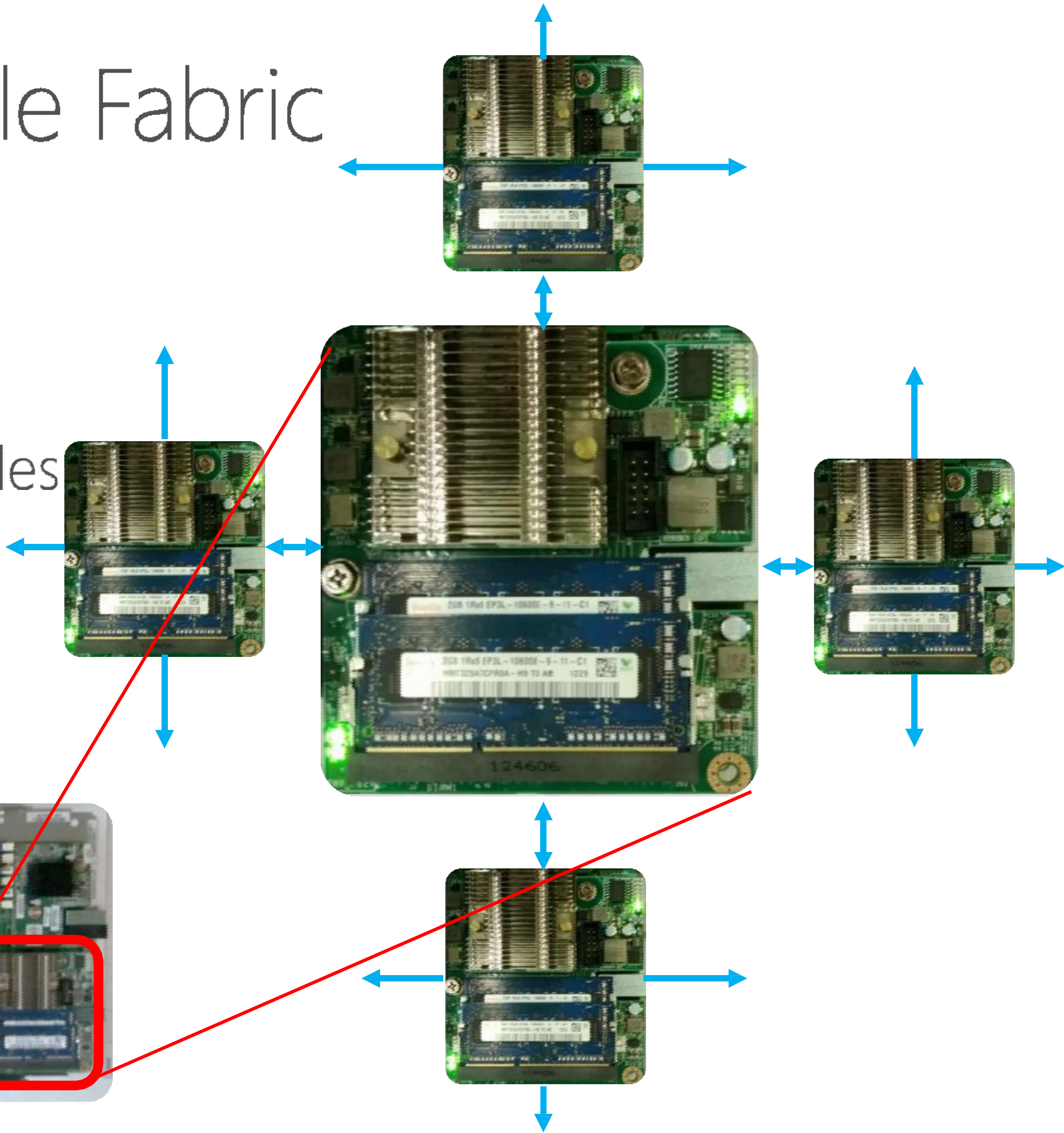




6 shell cables

Scalable Reconfigurable Fabric

- 1 FPGA board per Server
- 48 Servers per ½ Rack
- 6x8 Torus Network among FPGAs
  - 20 Gb over SAS SFF 8088 cables



Data Center Server (1U, ½ width)

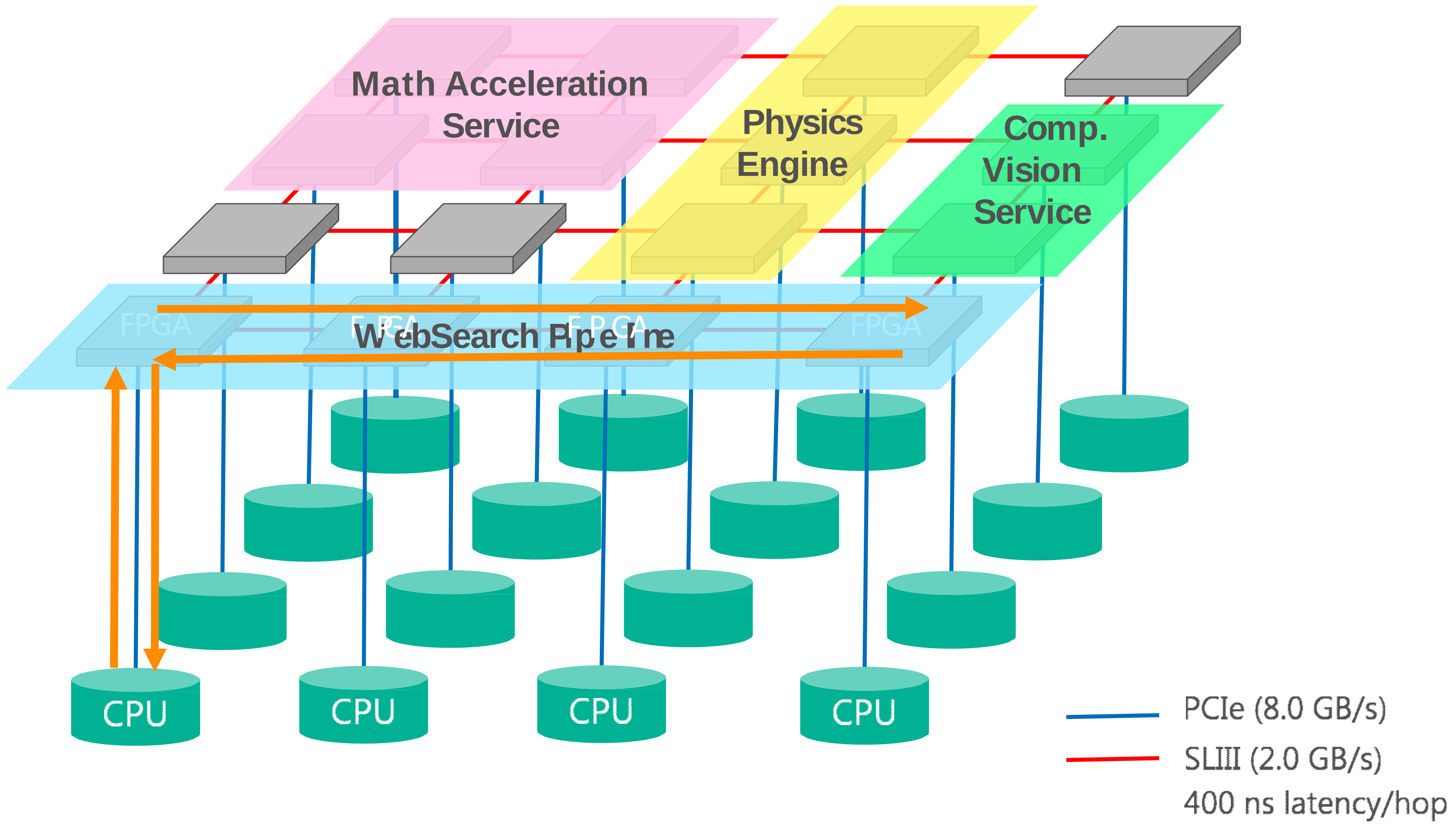
Infrastructure and Platform Architecture

- To enable productive use of the FPGA:
  - (1) APIs for interfacing software with the FPGA → SW Interface
  - (2) interfaces b/n FPGA and board-level functions → shell
  - (3) support for resilience and debugging → Error correction

Software Interface

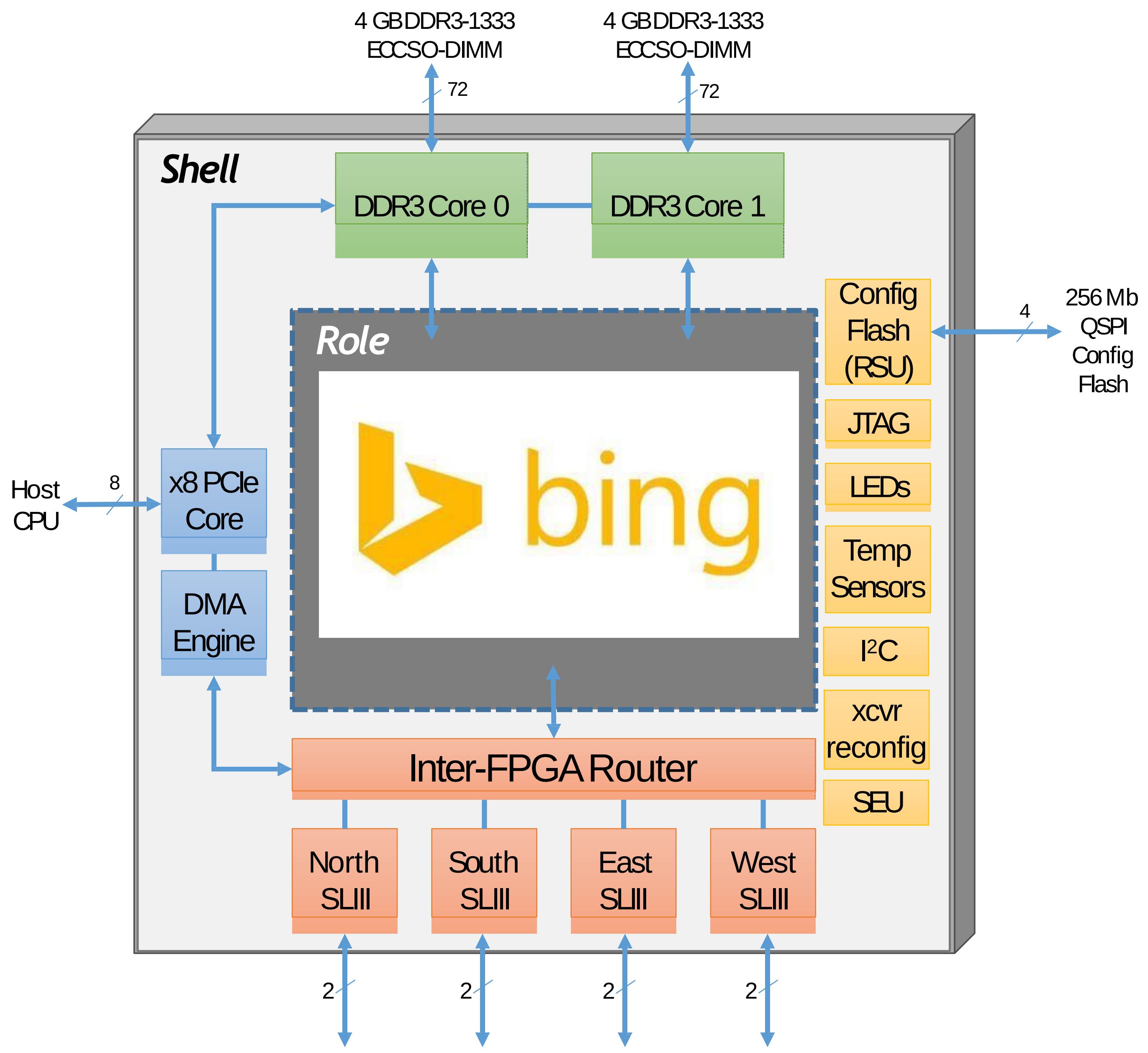
- Design goals:
  - Host-to-FPGA:
    - ➢Latency (10us/16KB)
    - ➢multithreading
- Custom PCIe interface with DMA support
  - No sys calls
  - 1 I/P, 1 O/P buffer in non-paged memory
  - Buffer has 64 slots of size 64KB
  - Status bits
- Services initiated through calls to low-level software library

An Elastic Reconfigurable Fabric

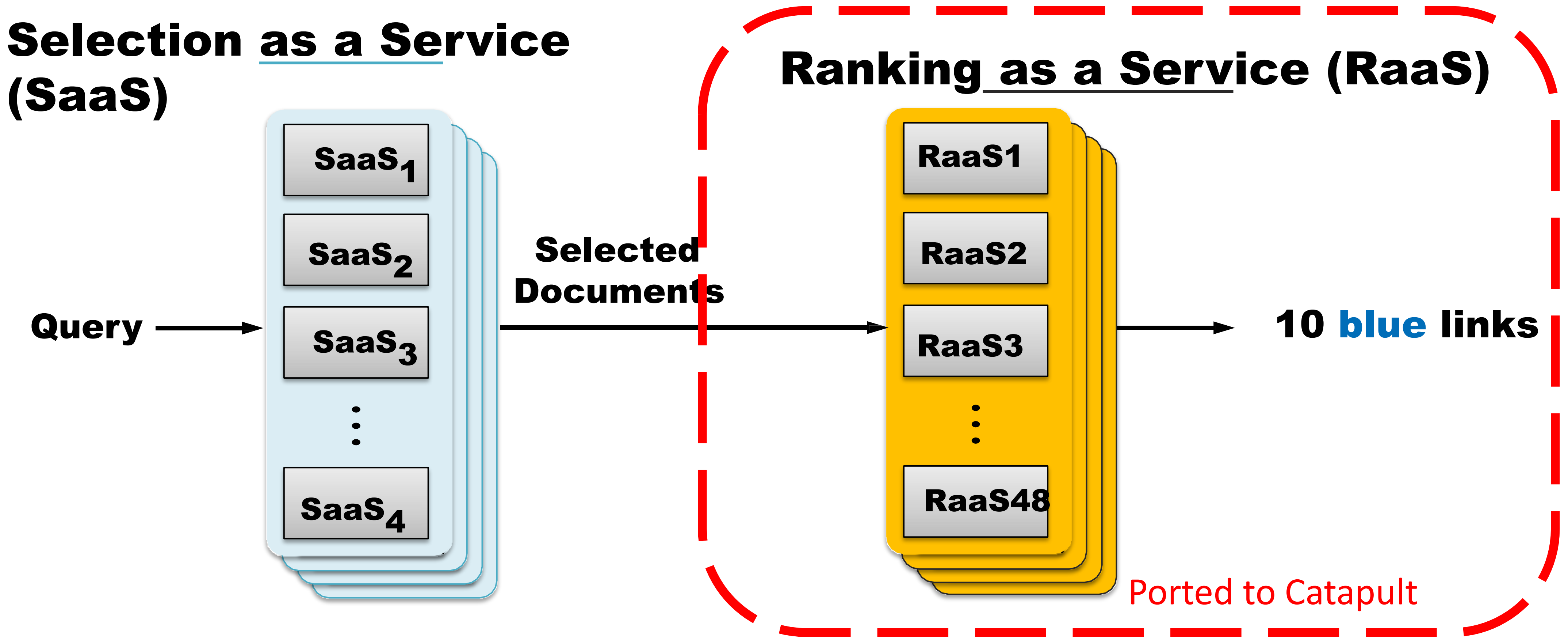


Shell & Role

- *Shell* handles all I/O & management tasks
- *Role* is only application logic
- Shell exposes simple FIFOs
- Flight data recorder for scale out debug
- Role is Partial Reconfig boundary



Bing Document Ranking Flow

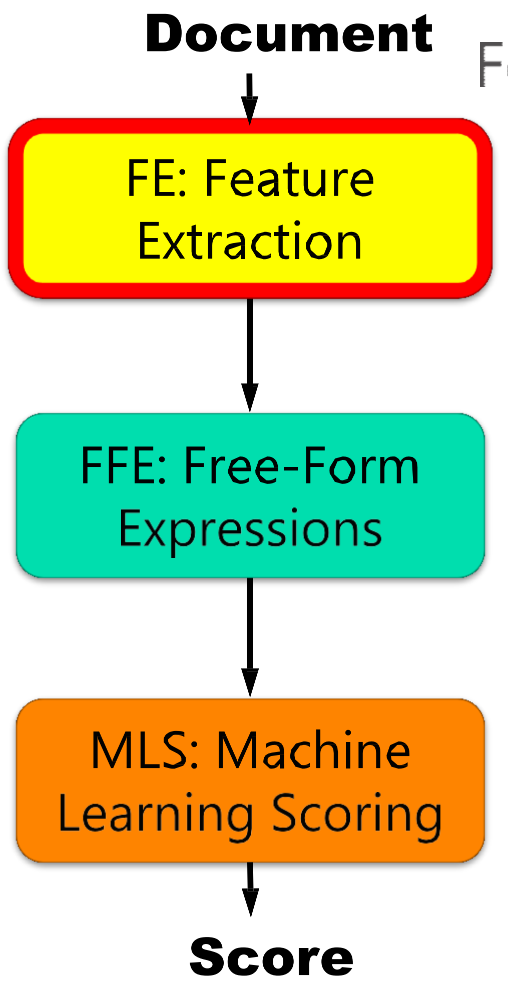


Selection-as-a-Service (SaaS)
-Find all docs that contain query terms,
-Filter and select candidate documents for ranking

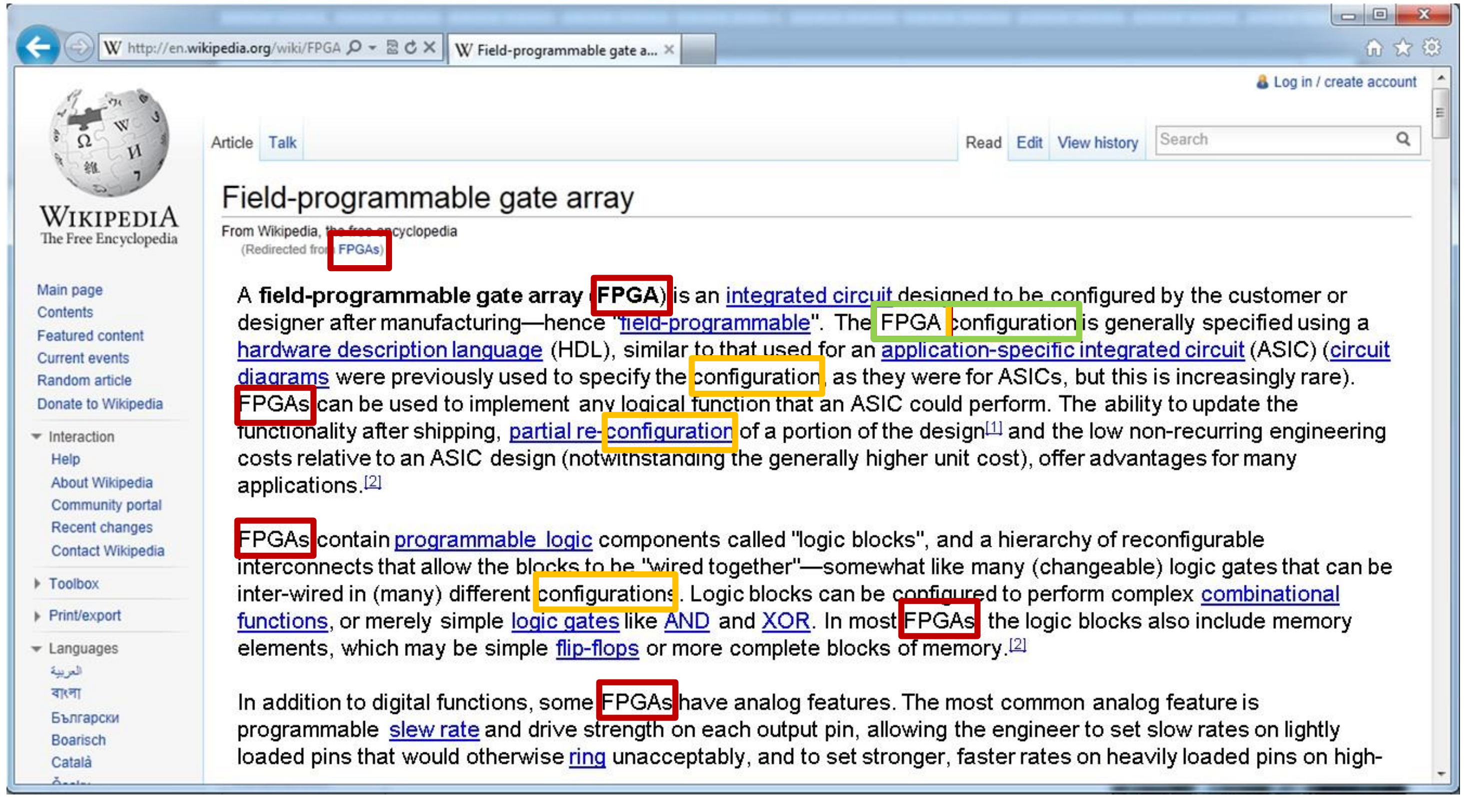
Ranking-as-a-Service (RaaS)
- **Compute scores for how relevant each selected document is for the search query**
- **Sort the scores and return the results**

FE: Feature Extraction

Query: “FPGA Configuration”



Features: NumberOfOccurrences_0 = 7 | NumberOfOccurrences_1 = 4 | NumberOfTuples_0_1 = 1



Field-programmable gate array

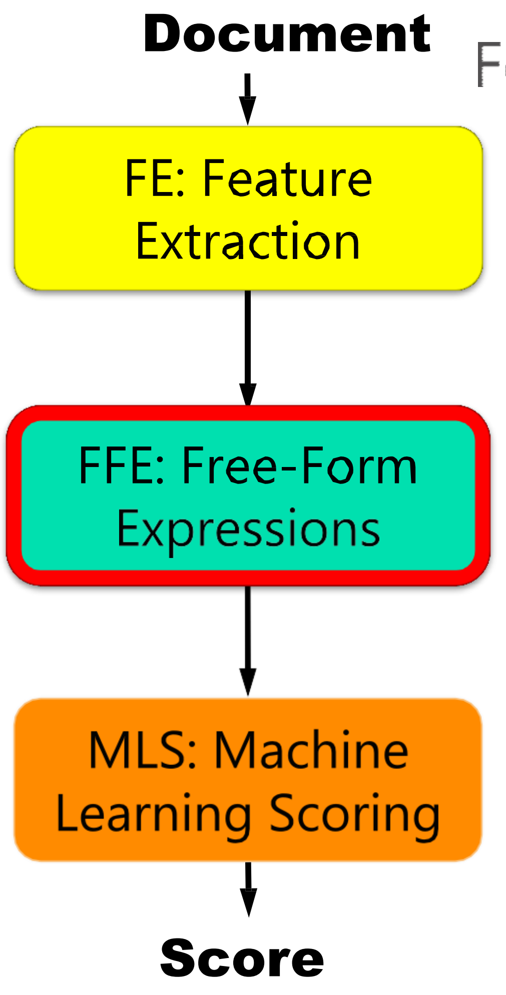
From Wikipedia, the free encyclopedia
(Redirected from FPGAs)

A **field-programmable gate array** (**FPGA**) is an integrated circuit designed to be configured by the customer or designer after manufacturing—hence "field-programmable". The FPGA configuration is generally specified using a hardware description language (HDL), similar to that used for an application-specific integrated circuit (ASIC) (circuit diagrams were previously used to specify the configuration, as they were for ASICs, but this is increasingly rare). FPGAs can be used to implement any logical function that an ASIC could perform. The ability to update the functionality after shipping, partial re-configuration of a portion of the design[1] and the low non-recurring engineering costs relative to an ASIC design (notwithstanding the generally higher unit cost), offer advantages for many applications.[2]

FPGAs contain programmable logic components called "logic blocks", and a hierarchy of reconfigurable interconnects that allow the blocks to be "wired together"—somewhat like many (changeable) logic gates that can be inter-wired in (many) different configurations. Logic blocks can be configured to perform complex combinational functions, or merely simple logic gates like AND and XOR. In most FPGAs the logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory.[2]

In addition to digital functions, some FPGAs have analog features. The most common analog feature is programmable slew rate and drive strength on each output pin, allowing the engineer to set slow rates on lightly loaded pins that would otherwise ring unacceptably, and to set stronger, faster rates on heavily loaded pins on high-

FFE: Free Form Expressions

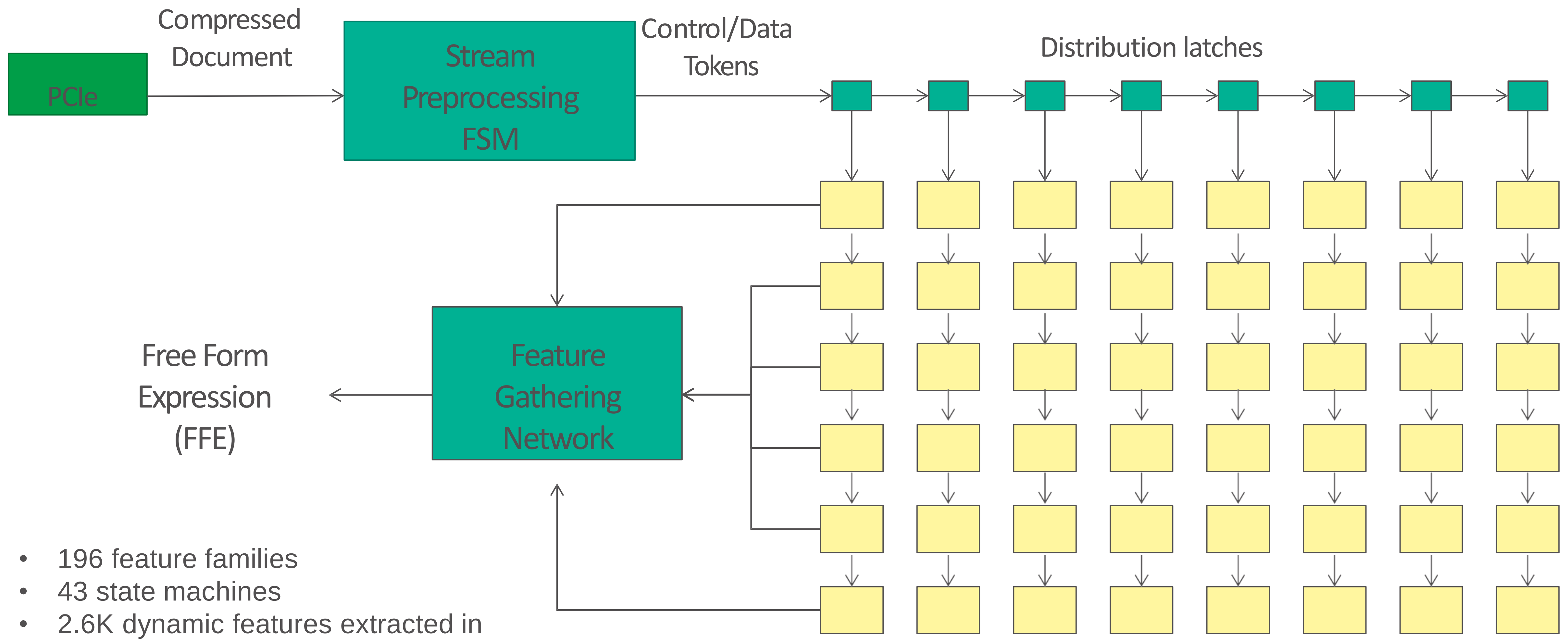


Features: NumberOfOccurrences_0 = 7 | NumberOfOccurrences_1 = 4 | NumberOfTuples_0_1 = 1

$$\text{FFE \#1} = \frac{(2*\text{NumberOfOccurrences\_0} + \text{NumberOfOccurrences\_1})}{(2 * \text{NumberOfTuples\_0\_1})}$$

Metafeature #1 = 9

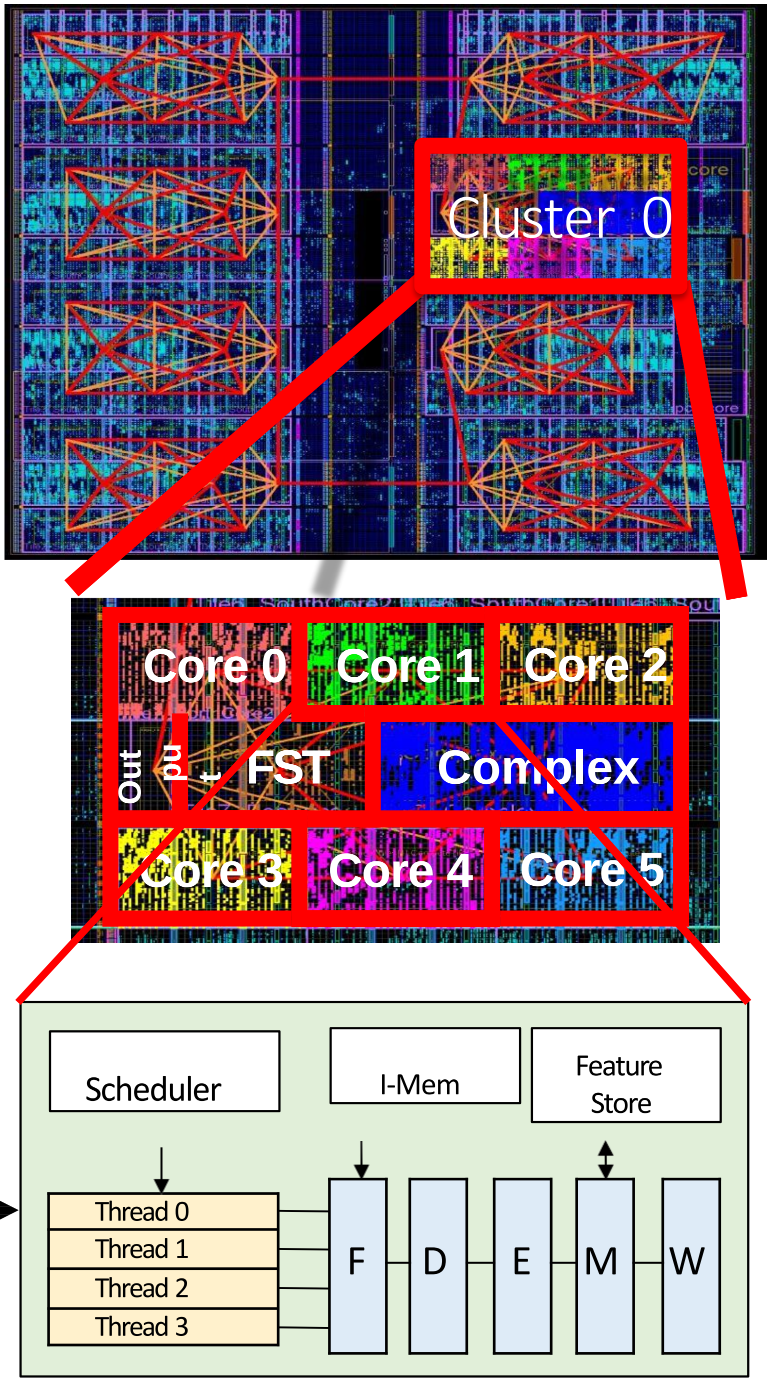
Feature Extraction Accelerator

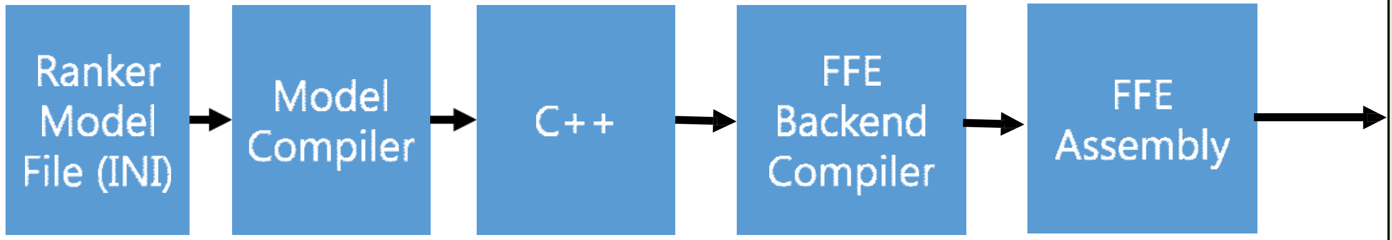


- 196 feature families
- 43 state machines
- 2.6K dynamic features extracted in less than 4us (~600us in SW)

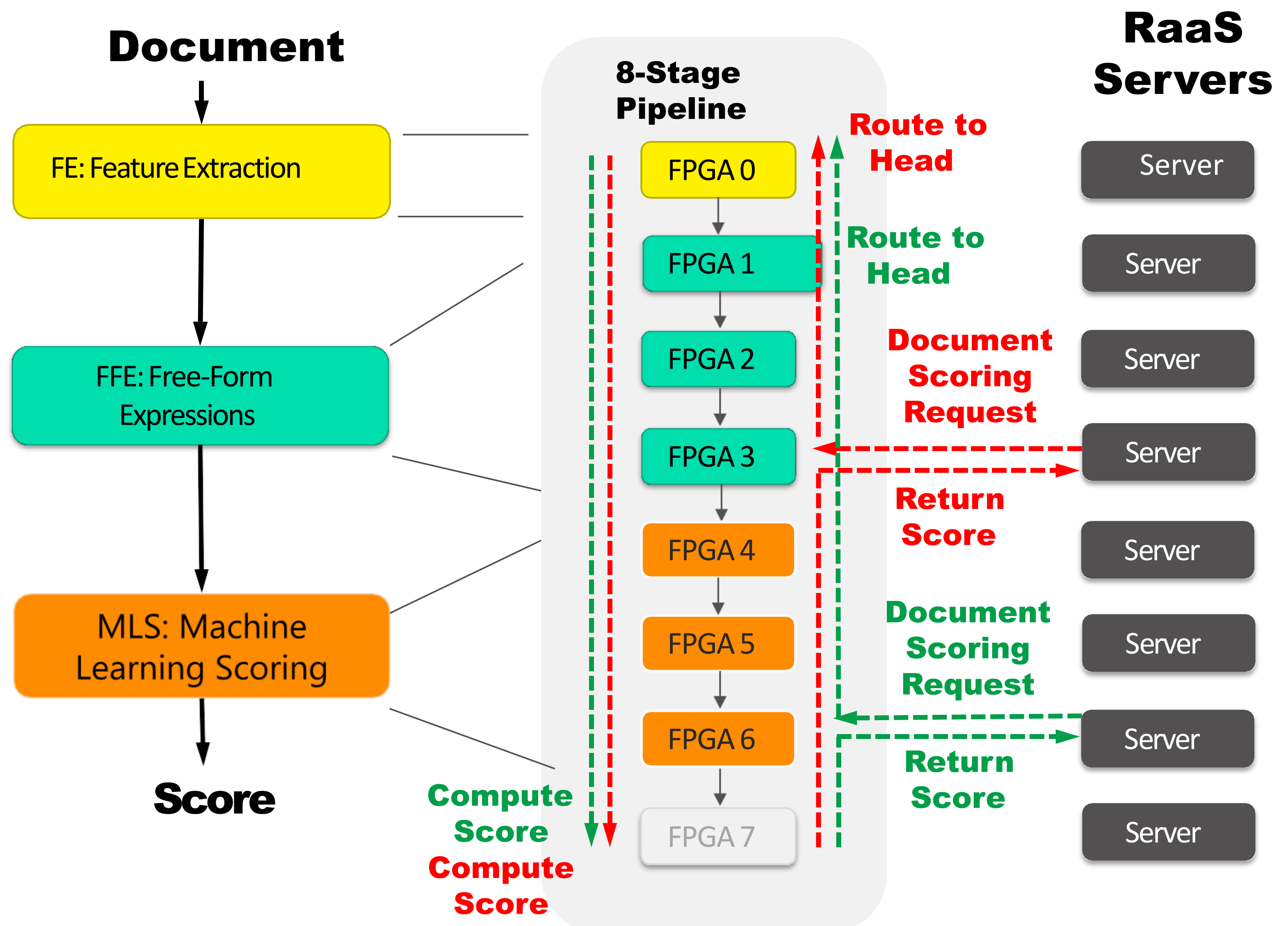
FFE Soft Cores

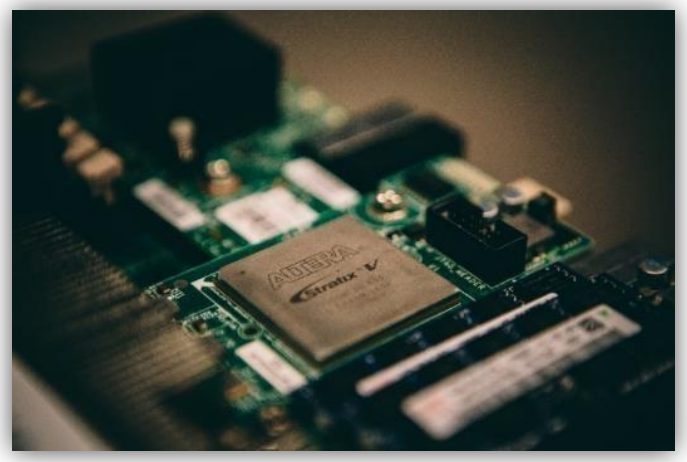
- Soft processor for multi threaded throughput
- 4 HW threads per core
- 6 cores share a complex ALU
- log, divide, exp, float/int conv.
- 10 clusters (240 HW threads) per FPGA

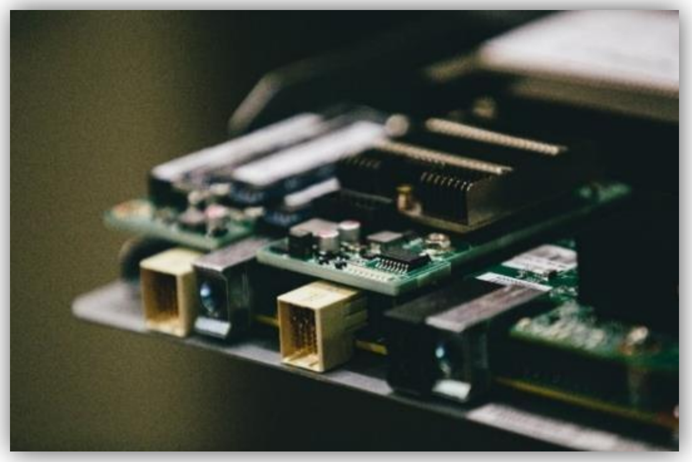




Putting it all together



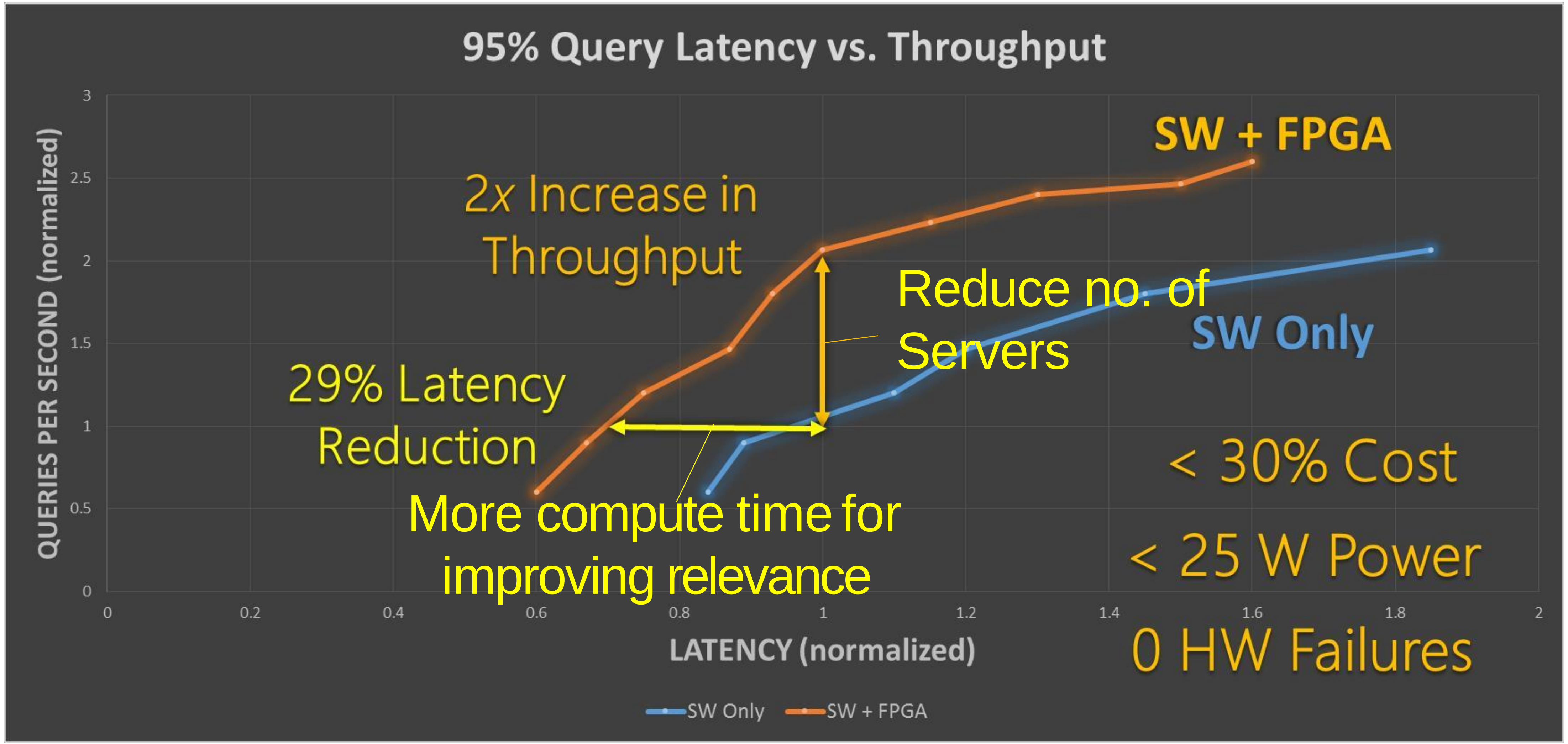




1,632 Server Pilot Deployed in a Production Datacenter

Accelerating Large-Scale Services – Bing Search

1,632 Servers with FPGAs Running Bing Page Ranking Service (~30,000 lines of C++)



Key Needs for FPGA Computing

Software & Language

- Huge need for high-productivity languages
  - C to gates tools did not do well on FE state machines
  - Domain specific languages, OpenCL, BlueSpec both show promise

Compiler & Tools

- Faster compilation times
- Fewer warnings… NO warnings on IP libraries
- Better debugging integration

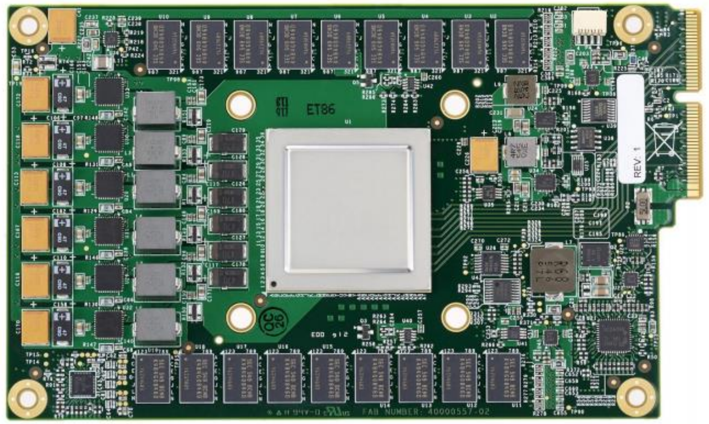
RTL & Hardware

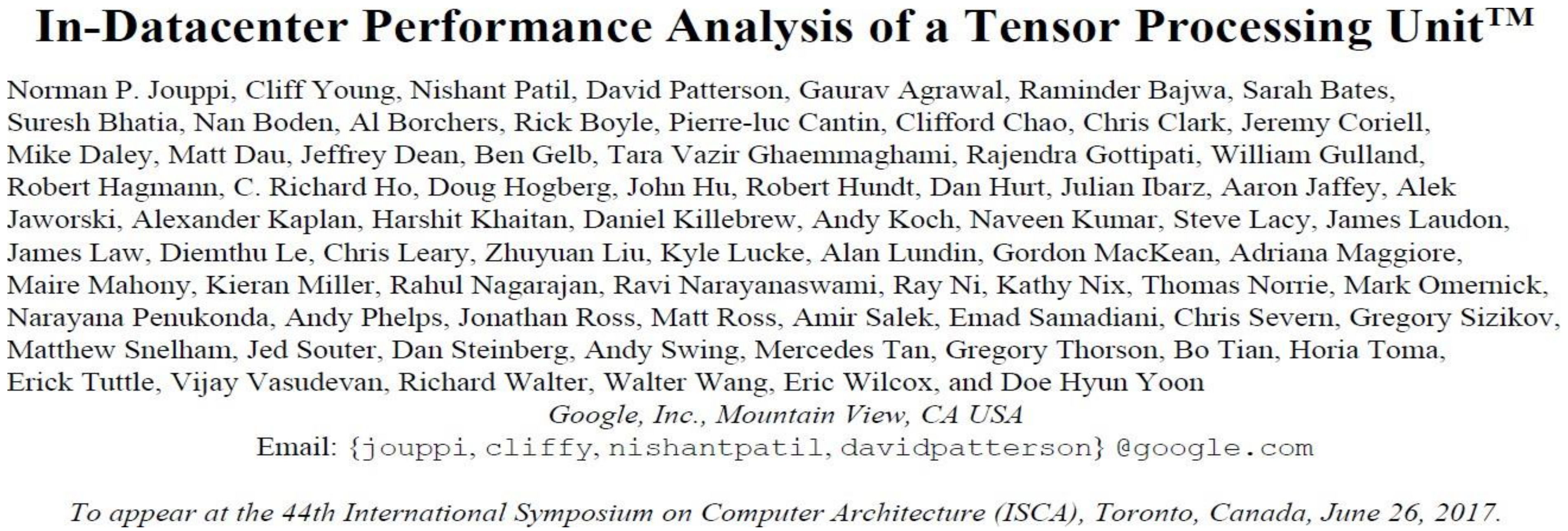
- Hardened PCIe, DDR, JTAG debugging
- Faster, more efficient DDR
- Improved floating point performance

Conclusions

- Hardware specialization is a (the?) way to gain efficiency and performance
- The Catapult reconfigurable fabric offers a flexible, elastic pool of resources to accelerate services
- Results for Bing: **½ the number of ranking servers**, lower latency, reduced variance, proven scalability, proven resilience
- Bing going to **production in early 2015**
- Biggest future problem is programmability

Google - Tensor Processing Unit





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, Daniel Killebrew, 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

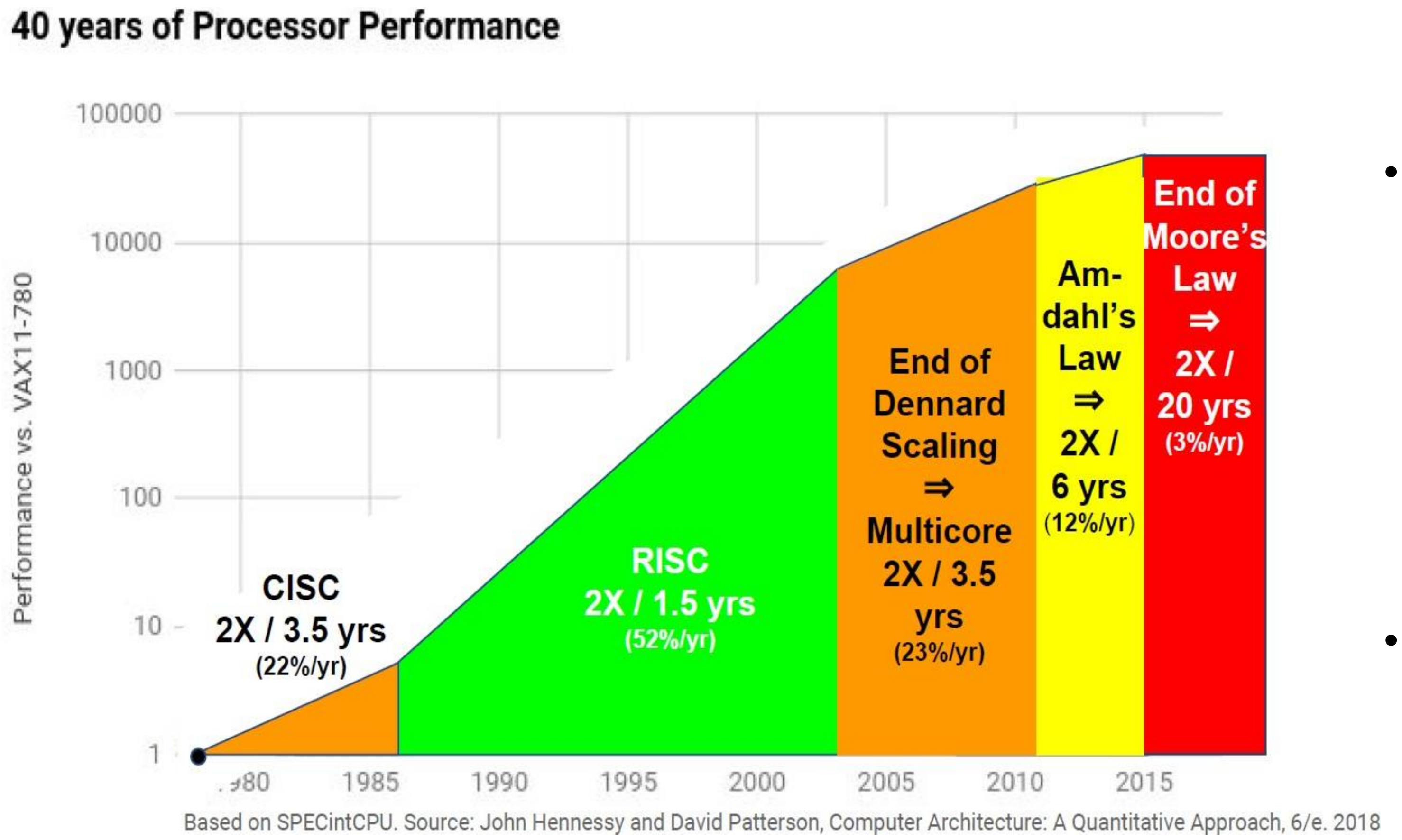
Email: {jouppi, cliffy, nishantpatil, davidpatterson} @google.com

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

A Golden Age in Microprocessor Design

- Stunning progress in microprocessor design 40 years ≈ 10^6x faster!
- Three architectural innovations (~1000x)
  - Width: 8→16→32→64 bit (~8x)
  - Instruction level parallelism:
    - 4-10 *clock cycles per instruction* to 4+ *instructions per clock cycle* (~10-20x)
  - Multicore: 1 processor to 16 cores (~16x)
- Clock rate: 3 to 4000 MHz (~1000x thru technology & architecture)
- Made possible by IC technology:
  - **Moore's Law**: growth in transistor count (2X every 1.5 years)
  - **Dennard Scaling**: power/transistor shrinks at same rate as transistors are added (constant per mm^2 of silicon)

End of Growth of Performance?

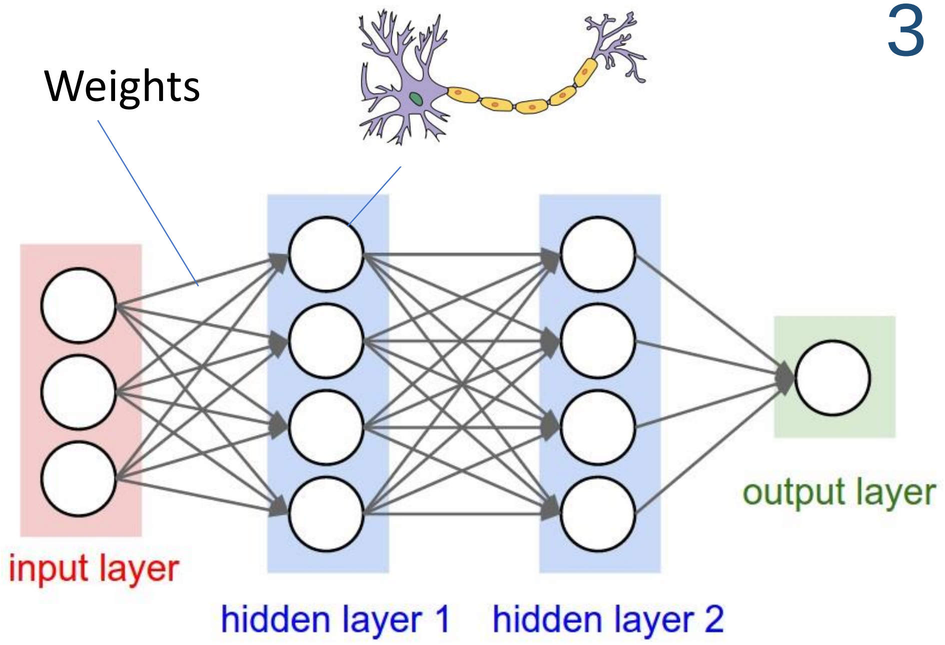


- Since
  - Transistors not getting much better
  - Power budget not getting much higher
  - Already switched from 1 inefficient processor/chip to N efficient processors/chip
- Only path left is Domain Specific Architetures
  - Just do a few tasks, but extremely well

TPU Origin

- Starting as far back as 2006, Google engineers had discussions about deploying GPUs, FPGAs, or custom ASICs in their data centers. They concluded that they can use the excess capacity of the large data centers.
- The conversation changed in 2013 when it was projected that if people used voice search for 3 minutes a day using speech recognition DNNs, it would have required Google's data centers to double in order to meet computation demands.
- Google then started a high-priority project to quickly produce a custom ASIC for inference.
- The goal was to improve cost-performance by 10x over GPUs.
- Given this mandate, the TPU was designed, verified, built, and deployed in data centers in just 15 months

Neural nets

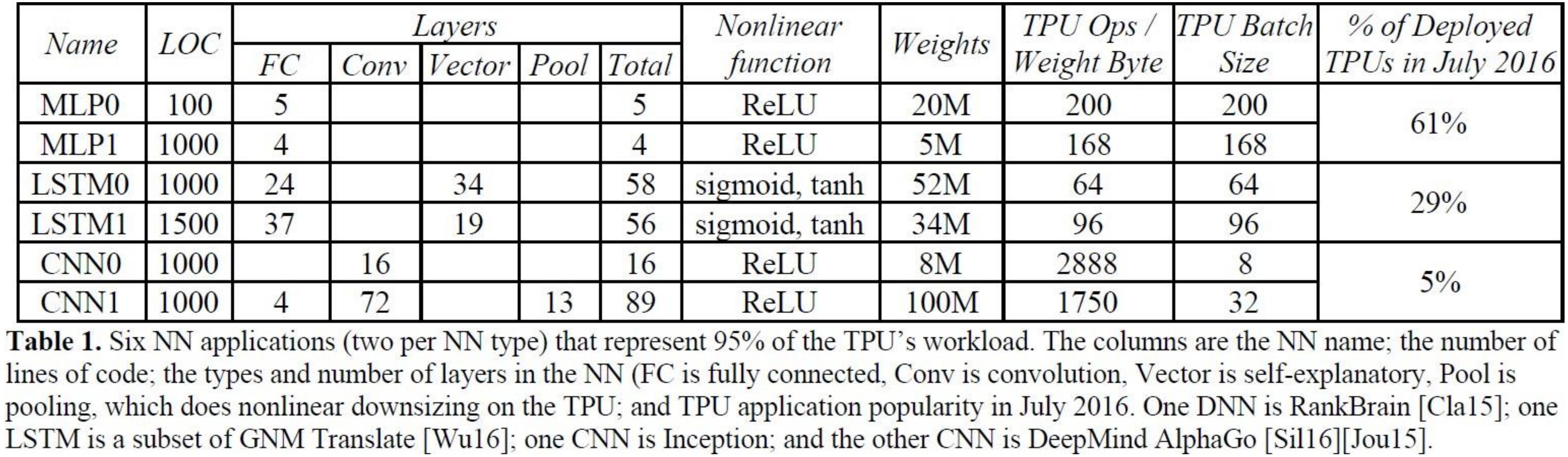


http://cs231n.github.io/neural-networks-1/

3 Kinds of Popular NNs

- Multi-Layer Perceptrons(MLP)
  - Each new layer is a set of nonlinear functions of weighted sum of all outputs (*fully connected*) from a prior one
- Convolutional Neural Networks(CNN)
  - Each ensuing layer is a set of nonlinear functions of weighted sums of spatially nearby subsets of outputs from the prior layer, which also reuses the weights.
- Recurrent Neural Networks(RNN)
  - Each subsequent layer is a collection of nonlinear functions of weighted sums of outputs and the previous state. The most popular RNN is *Long Short-Term Memory* (LSTM).

Inference Datacenter Workload(95%)

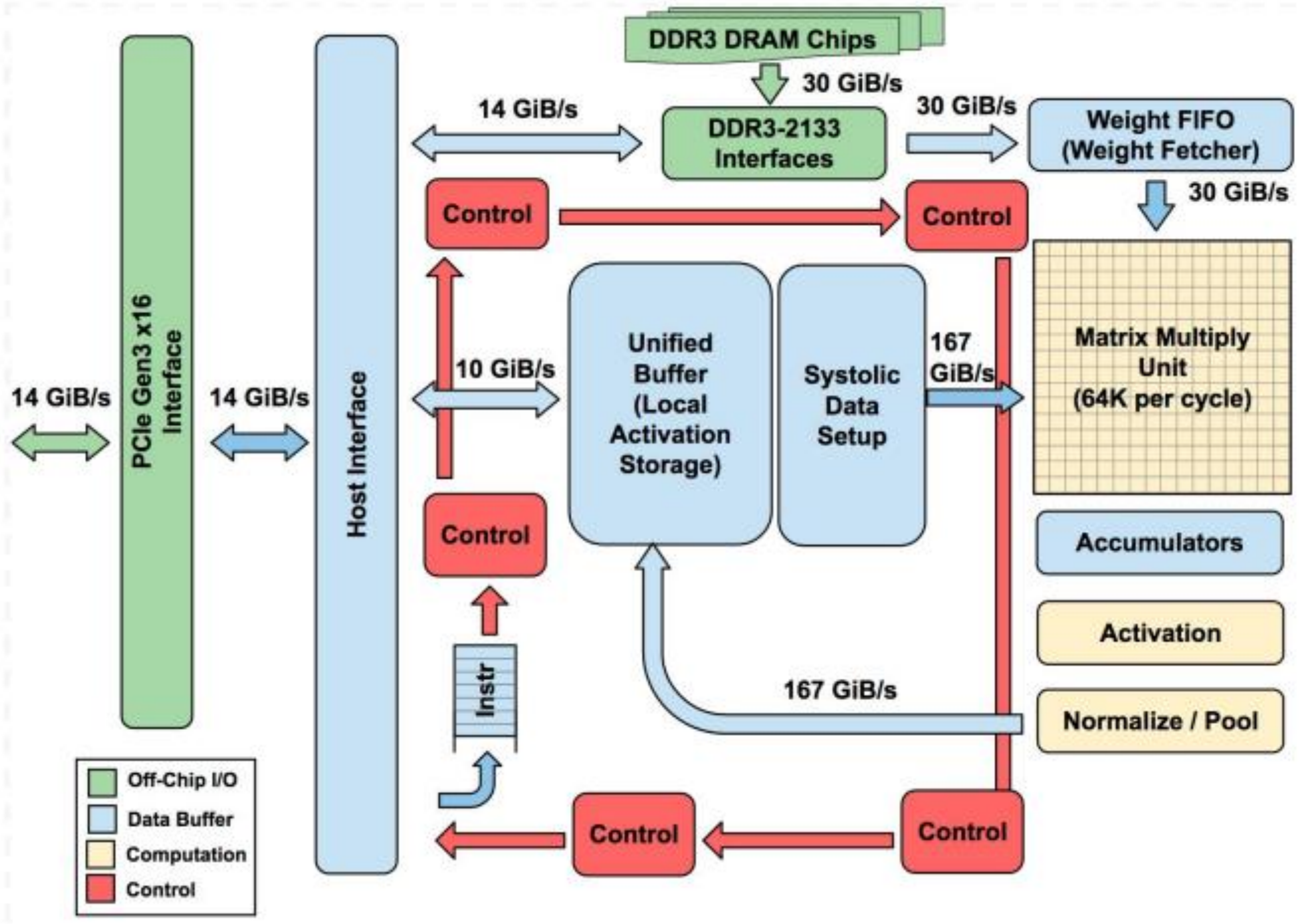


Name	*LOC*	*Layers*					*Nonlinear function*	*Weights*	*TPU Ops / Weight Byte*	*TPU Batch Size*	*% of Deployed TPUs in July 2016*
		FC	*Conv*	*Vector*	*Pool*	*Total*					
MLP0	100	5				5	ReLU	20M	200	200	61%
MLP1	1000	4				4	ReLU	5M	168	168	
LSTM0	1000	24		34		58	sigmoid, tanh	52M	64	64	29%
LSTM1	1500	37		19		56	sigmoid, tanh	34M	96	96	
CNN0	1000		16			16	ReLU	8M	2888	8	5%
CNN1	1000	4	72		13	89	ReLU	100M	1750	32	

Table 1. Six NN applications (two per NN type) that represent 95% of the TPU's workload. The columns are the NN name; the number of lines of code; the types and number of layers in the NN (FC is fully connected, Conv is convolution, Vector is self-explanatory, Pool is pooling, which does nonlinear downsizing on the TPU; and TPU application popularity in July 2016. One DNN is RankBrain [Cla15]; one LSTM is a subset of GNM Translate [Wu16]; one CNN is Inception; and the other CNN is DeepMind AlphaGo [Sil16][Jou15].

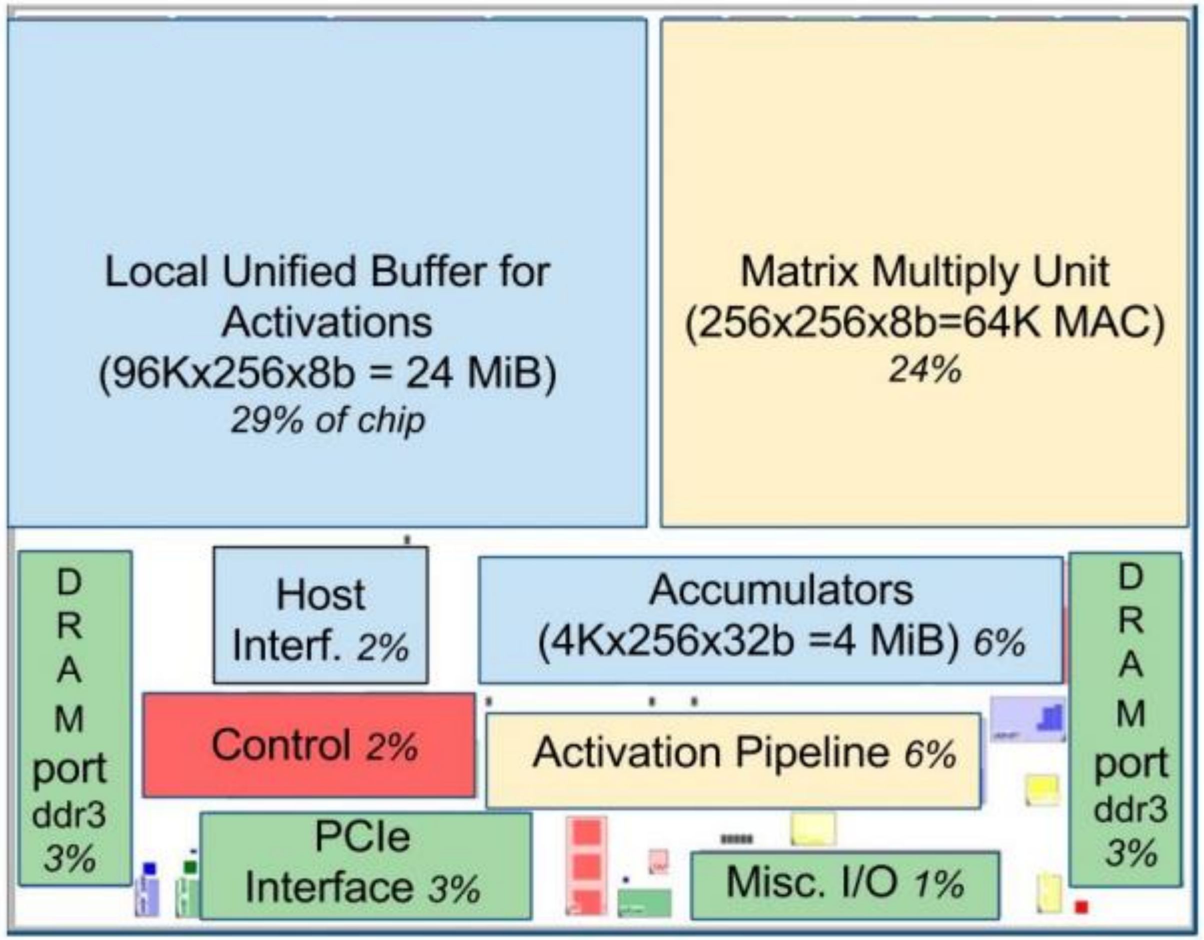
TPU architecture

- PCIe coprocessor
- No internal instruction fetch
    - CISC-like instructions from host:
        - Read_Host_Memory
        - Read_Weights
        - MatrixMultiply/Convolve
        - Activate
        - Write_Host_Memory
- Off-chip DDR3 weight memory

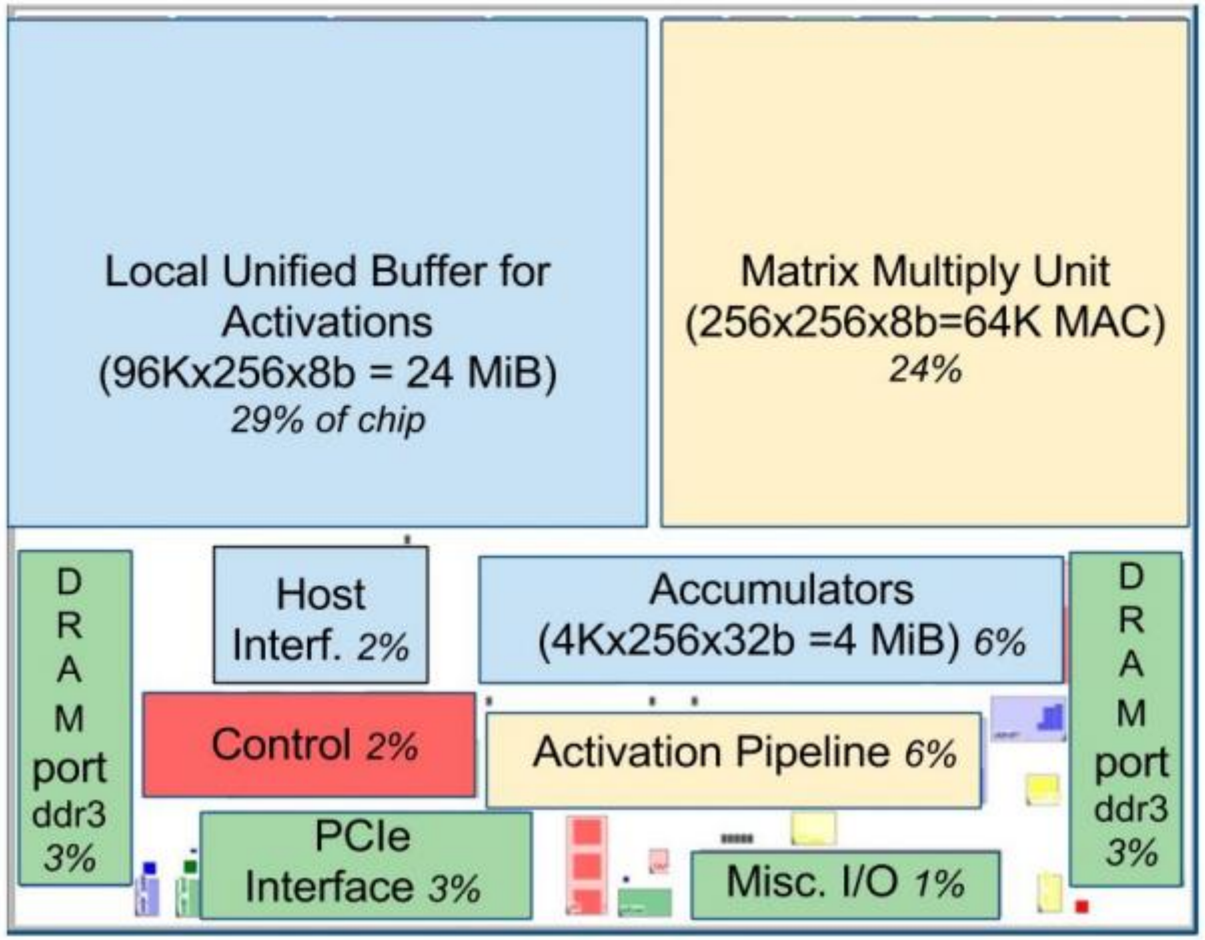


TPU architecture

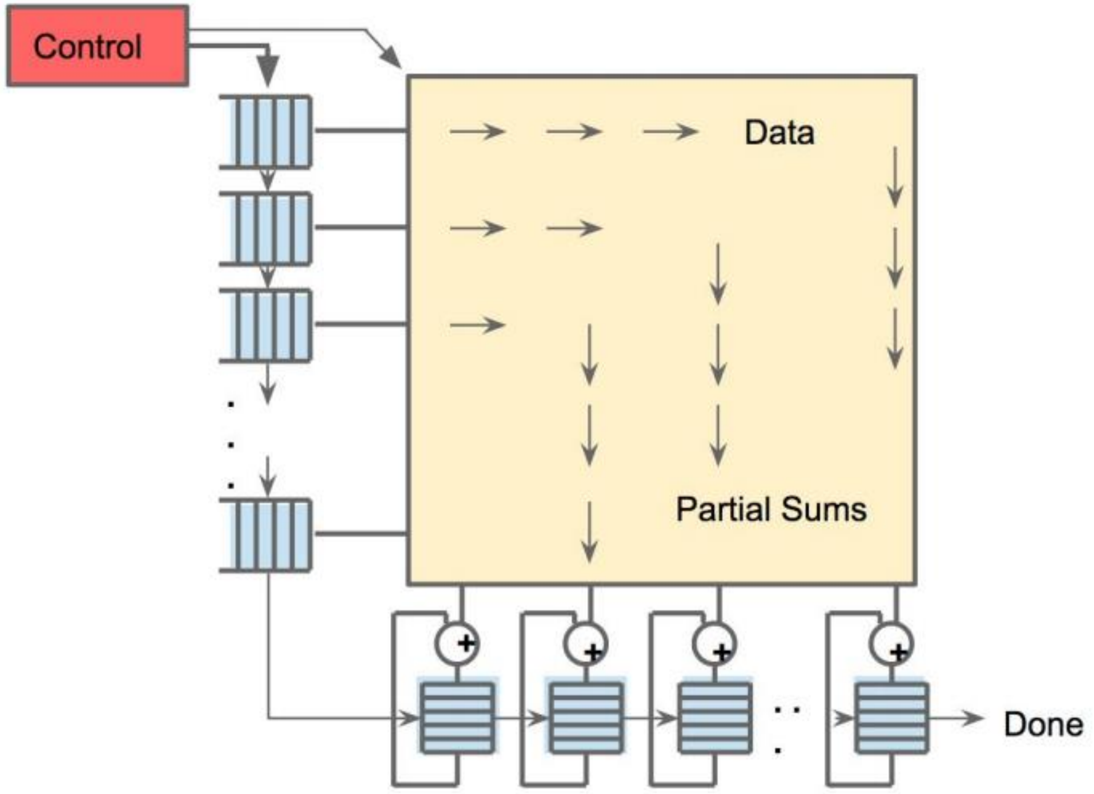
- MACs for core computation
- 24MB Unified Buffer
  - Store intermediate results
  - Sized to match pitch of matmult unit, simplify compilation w/ specific apps
- Tiny control logic


Local Unified Buffer for Activations (96Kx256x8b = 24 MiB) 29% of chip
Matrix Multiply Unit (256x256x8b=64K MAC) 24%
DRAM port ddr3 3%
Host Interf. 2%
Accumulators (4Kx256x32b =4 MiB) 6%
DRAM port ddr3 3%
Control 2%
Activation Pipeline 6%
PCIe Interface 3%
Misc. I/O 1%

TPU architecture – systolic structure

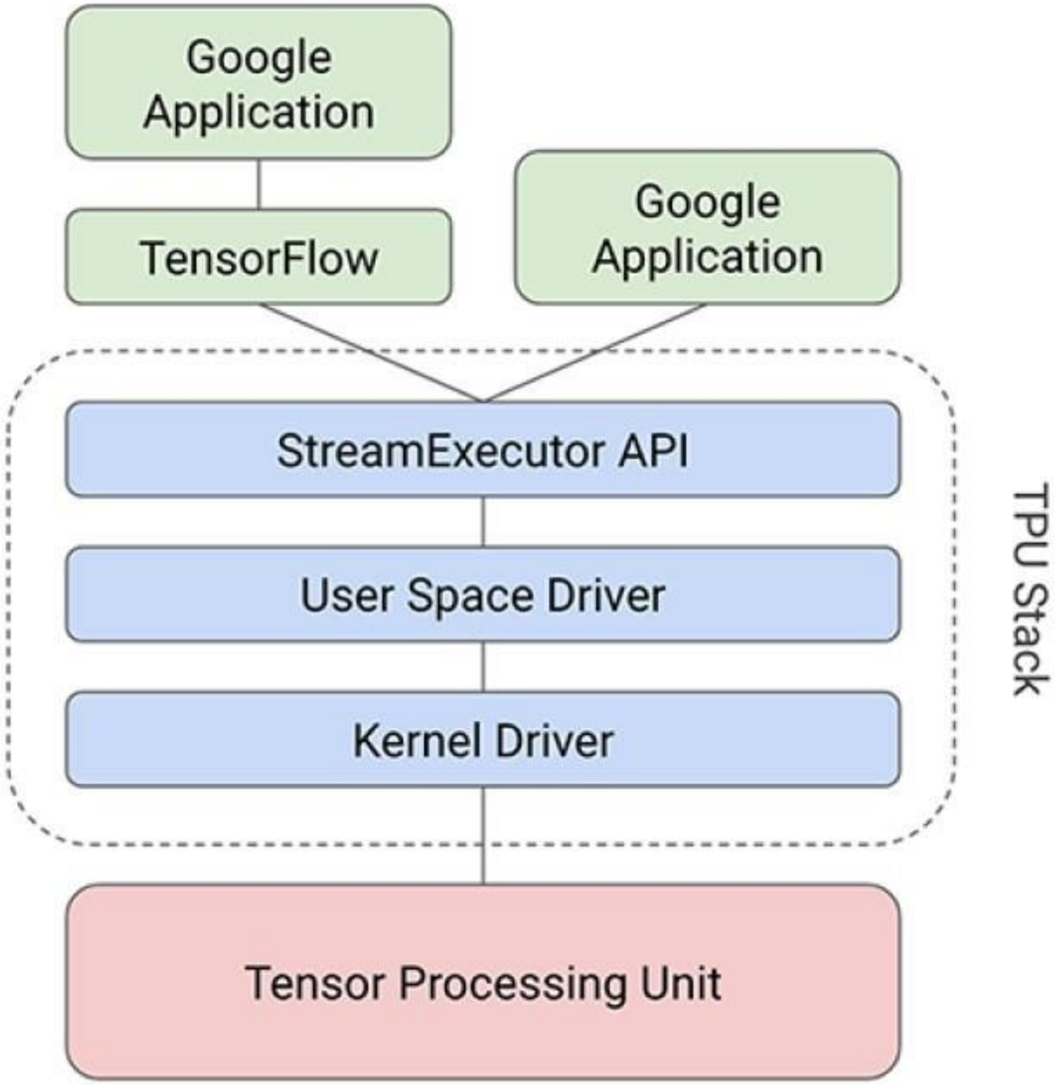


Local Unified Buffer for Activations (96Kx256x8b = 24 MiB) 29% of chip
Matrix Multiply Unit (256x256x8b=64K MAC) 24%
DRAM port ddr3 3%
Host Interf. 2%
Accumulators (4Kx256x32b =4 MiB) 6%
DRAM port ddr3 3%
Control 2%
Activation Pipeline 6%
PCIe Interface 3%
Misc. I/O 1%

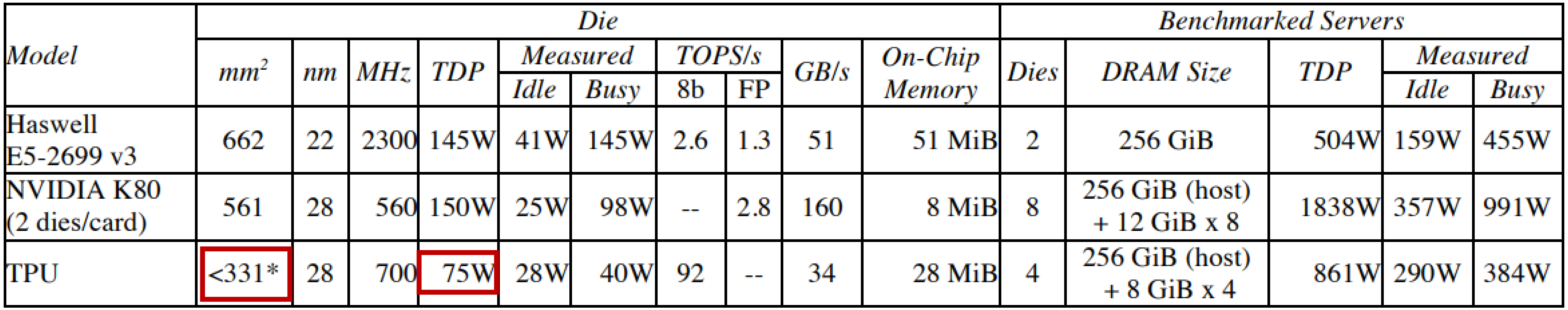


Software Stack

- Software stack is split into a User Space Driver and a Kernel Driver.
- The Kernel Driver is lightweight
  - Handles only memory management and interrupts.
- The User Space driver changes frequently.
  - It sets up and controls TPU execution
  - Reformats data into TPU order
  - Translates API calls into TPU instructions, and turns them into an application binary.



System configurations



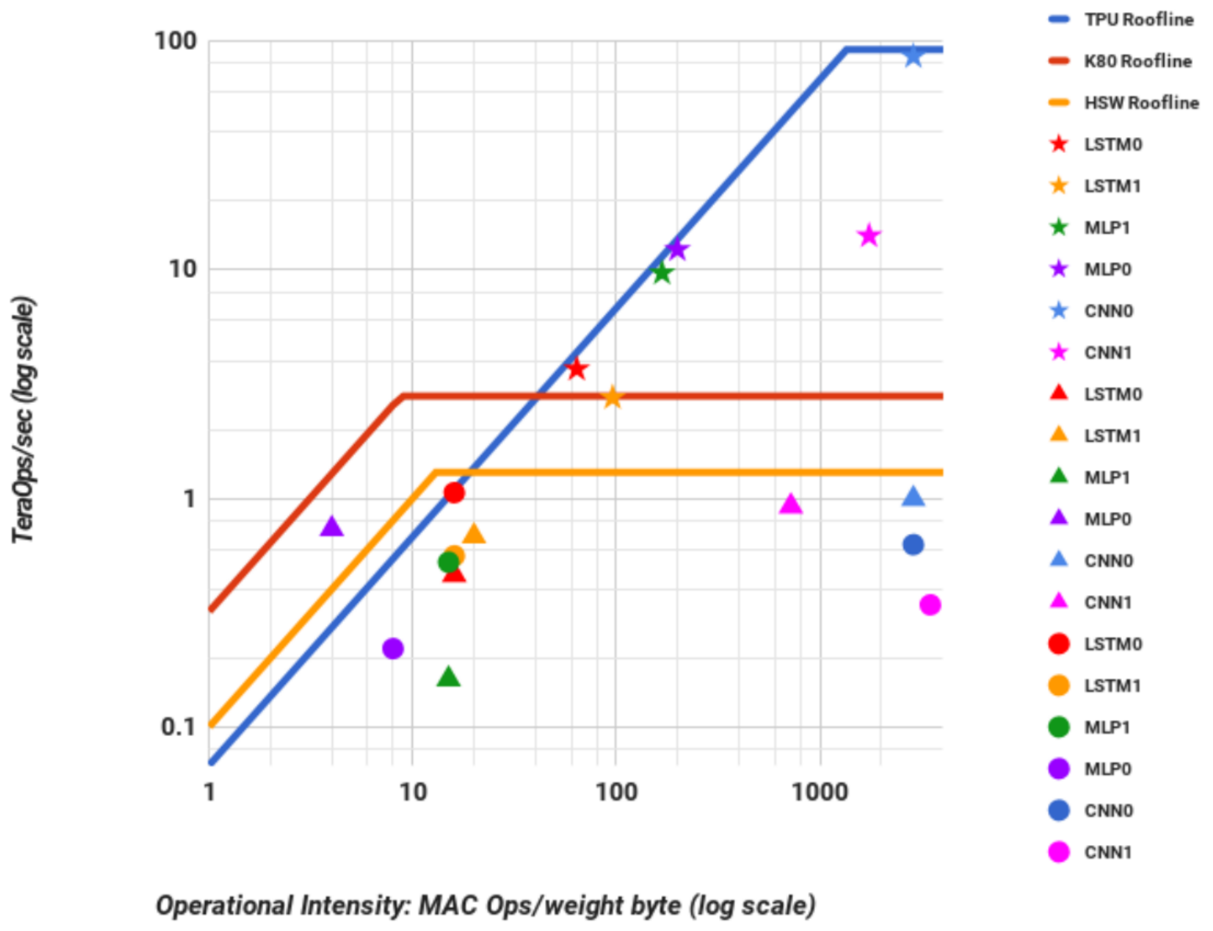
Model	Die										Benchmarked Servers				
	mm²	nm	MHz	TDP	Measured		TOPS/s		GB/s	On-Chip Memory	Dies	DRAM Size	TDP	Measured	
					Idle	Busy	8b	FP						Idle	Busy
Haswell E5-2699 v3	662	22	2300	145W	41W	145W	2.6	1.3	51	51 MiB	2	256 GiB	504W	159W	455W
NVIDIA K80 (2 dies/card)	561	28	560	150W	25W	98W	--	2.8	160	8 MiB	8	256 GiB (host) + 12 GiB x 8	1838W	357W	991W
TPU	<331*	28	700	75W	28W	40W	92	--	34	28 MiB	4	256 GiB (host) + 8 GiB x 4	861W	290W	384W

* TPU is less than half die size of the Intel Haswell processor

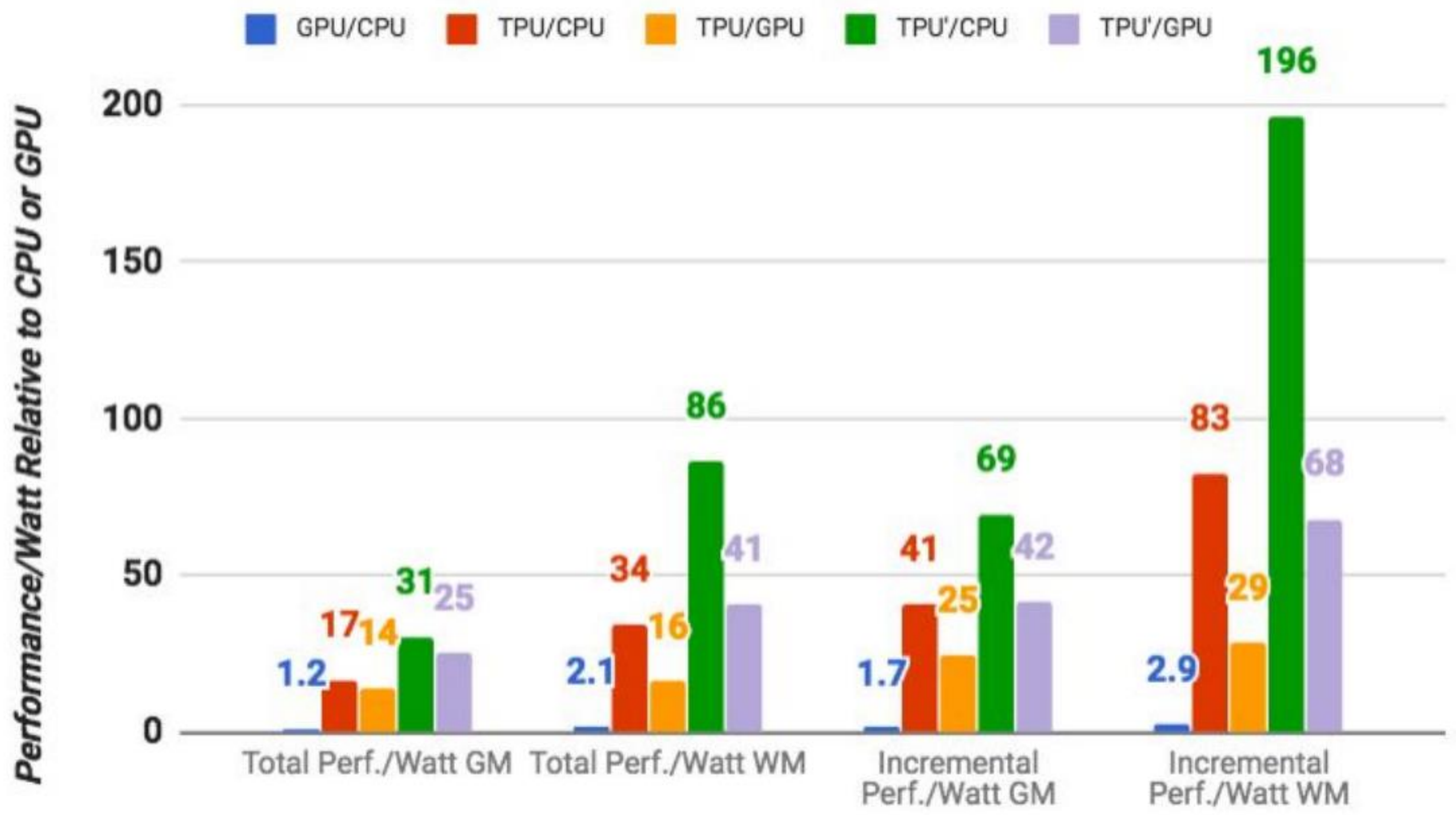
- K80 and TPU in 28nm process, Haswell fabbed in intel 22nm process
- Widely deployed in Google data centers

Performance

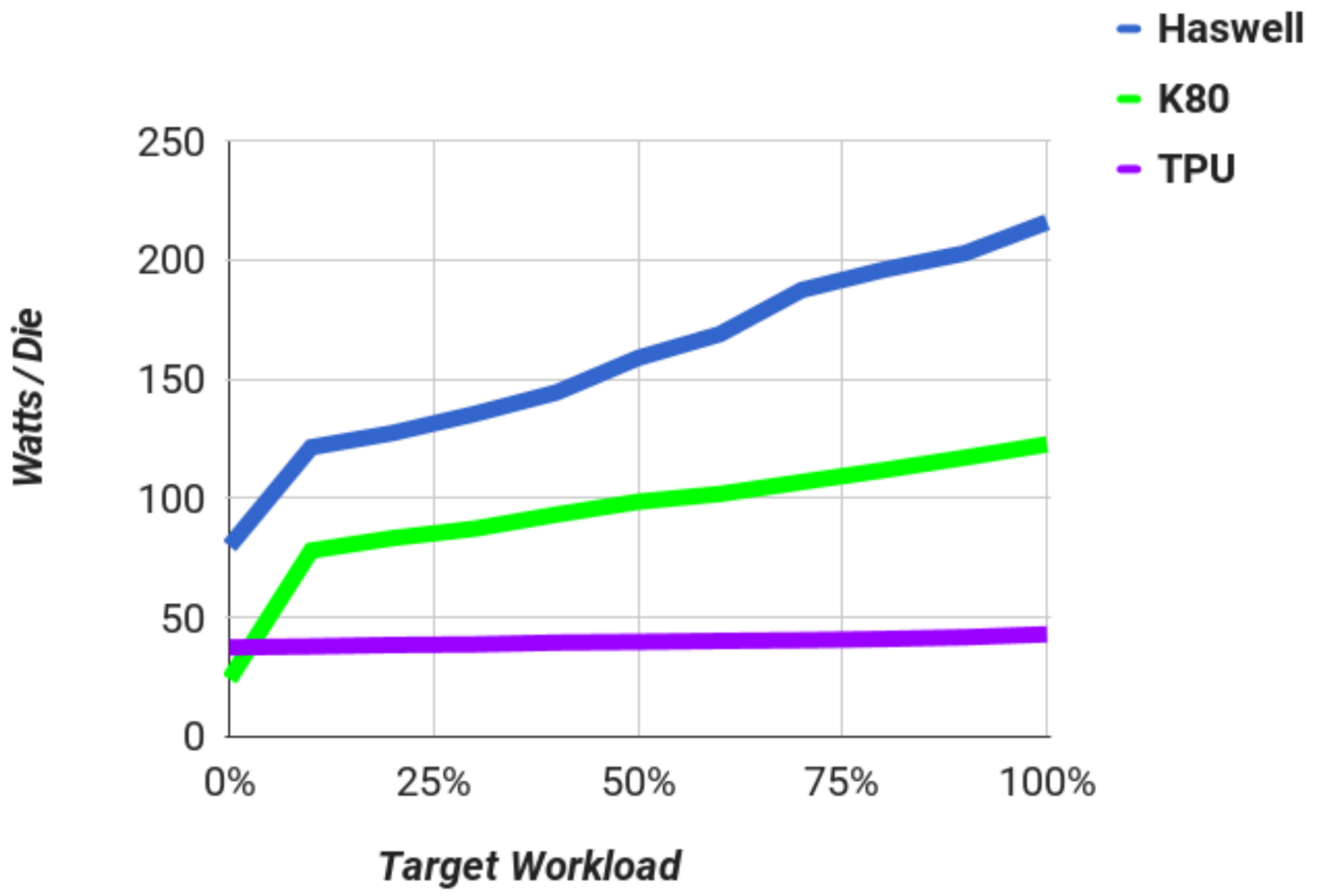
- "Roofline curves" – computation vs memory-intensity
  - "Ridge point" at intensity where app becomes compute-bound
  - Before ridge = memory-bound
  - After ridge = compute-bound
- Below curve = response time-constrained



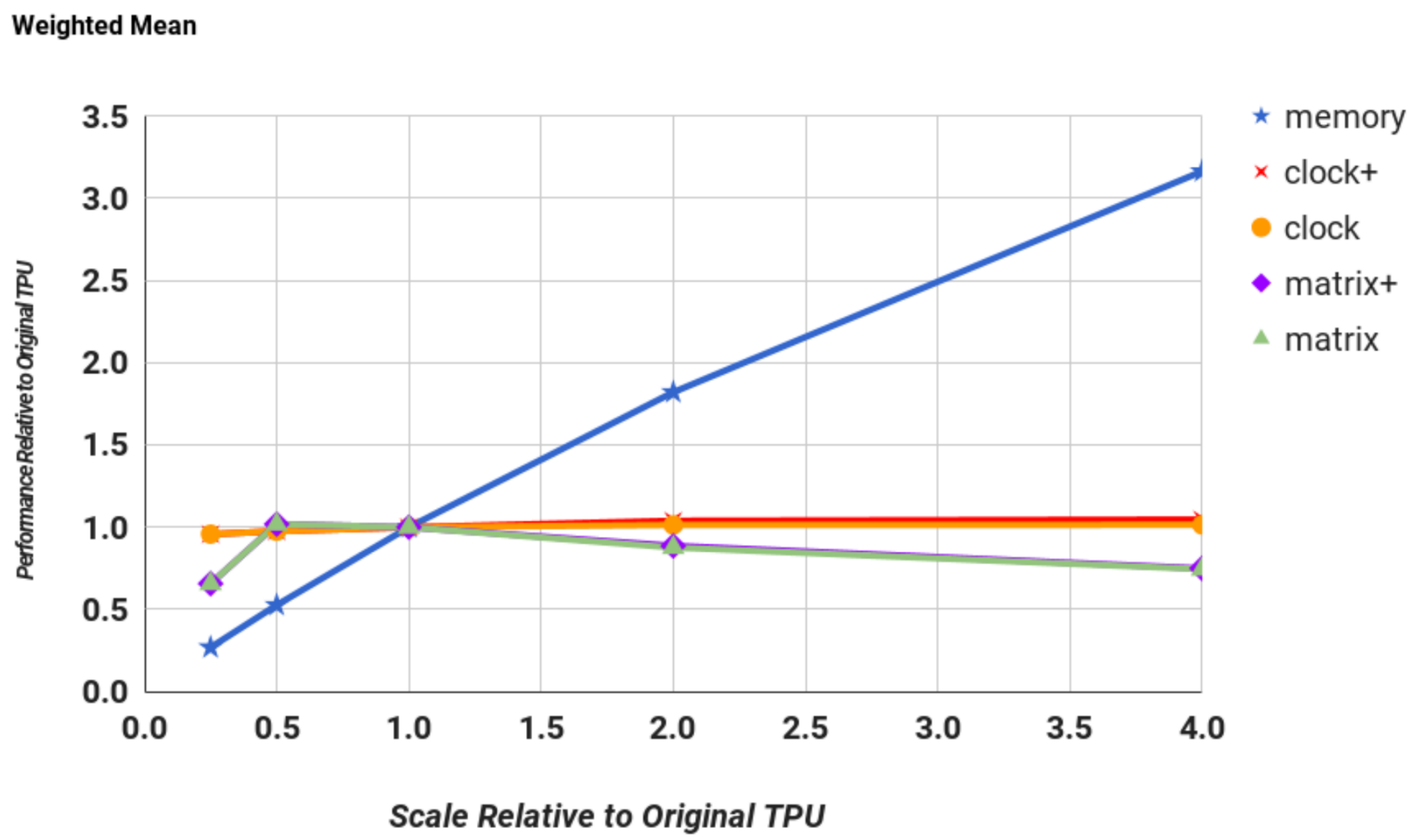
Performance – energy efficiency



Performance – energy proportionality

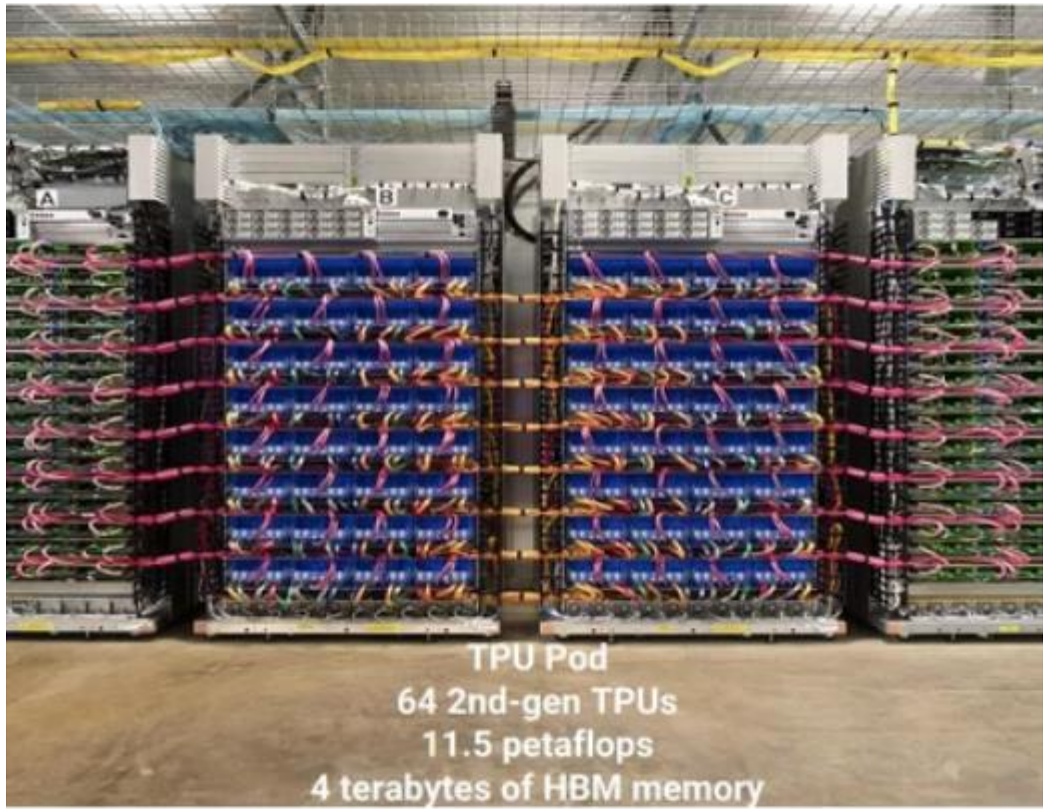


Design space exploration



TPU v2

- At HotChips 2017:
  - 2x 128x128x32b “mixed multiply units” (MXUs)
  - 64GB HBM
  - 64x TPU modules per “pod” → 4TB HBM
  - Some available in TensorFlow cloud svc



Google vs. Microsoft

- Why Google ASIC? Why Microsoft FPGA?
- Flexibility? Programmability?
- Cost and usefulness over time?

References

Figures and slides are from

- *Norman P. Jouppi*, et al., "In-Datacenter Performance Analysis of a Tensor Processing Unit", *44th IEEE/ACM International Symposium on Computer Architecture (ISCA-44)*, Toronto, Canada, June 2017. https://arxiv.org/abs/1704.04760

- *David Patterson*, "Evaluation of the Tensor Processing Unit: A Deep Neural Network Accelerator for the Datacenter", *NAE Regional Meeting*, April 2017. https://sites.google.com/view/naeregionalsymposium

- *Kaz Sato*, "An in-depth look at Google's first Tensor Processing Unit (TPU)", https://cloud.google.com/blog/big-data/2017/05/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu