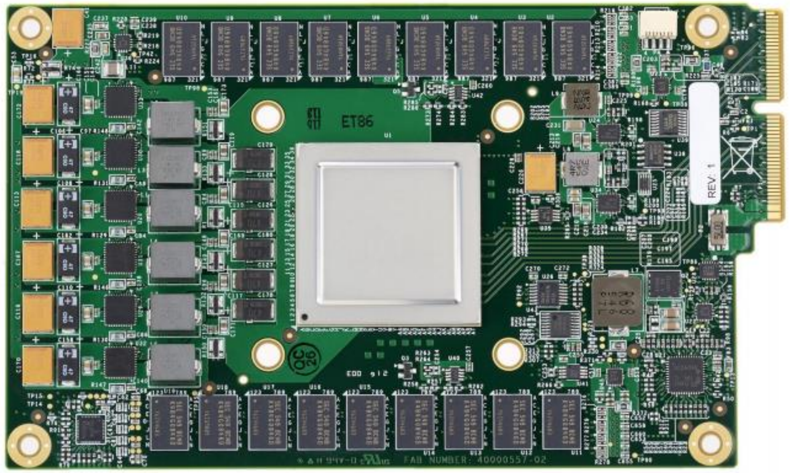
Application-Specific Hardware

…in the real world

Google

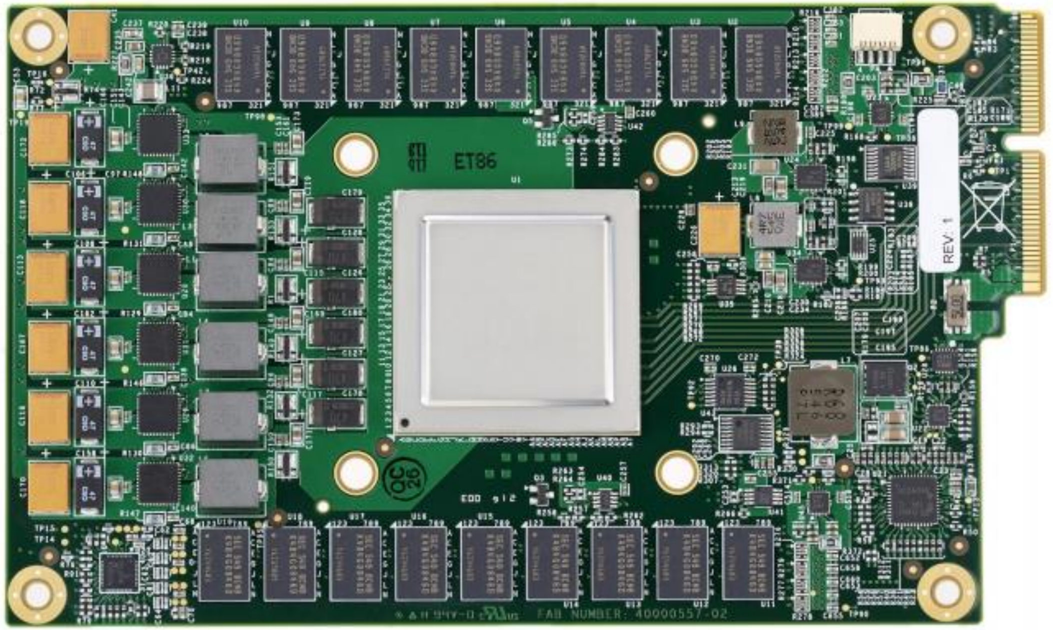


Microsoft

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

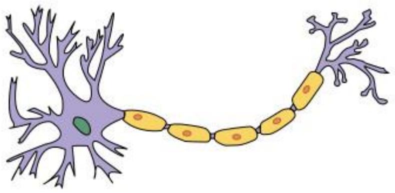
Google - Tensor Processing Unit

- Why?
  - 2006:
    - First considered datacenter ASIC/FPGA/GPU, decided excess capacity would suffice
  - 2013 projection:
    - Search by voice for 3min/day using DNNs → **double** datacenter computation needs
- Goals:
  - 10x better cost-performance vs GPUs
  - Deployment ASAP



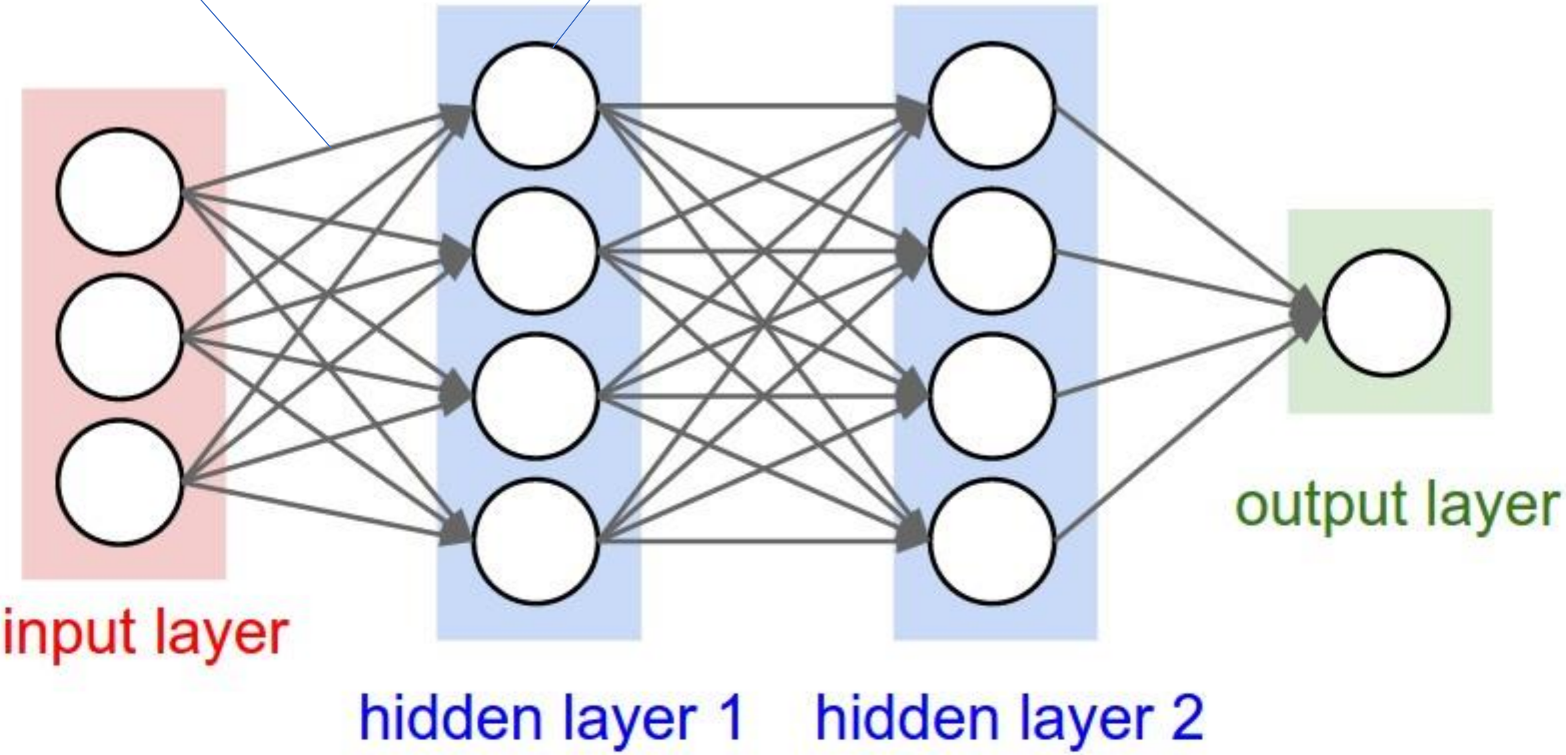
All images in this section are from TPU paper at ISCA 2017: https://www.cs.virginia.edu/~smk9u/CS6354F17/TPU.pdf

Neural nets



https://upload.wikimedia.org/wikipedia/commons/b/be/Derived_Neuron_schema_with_no_labels.svg

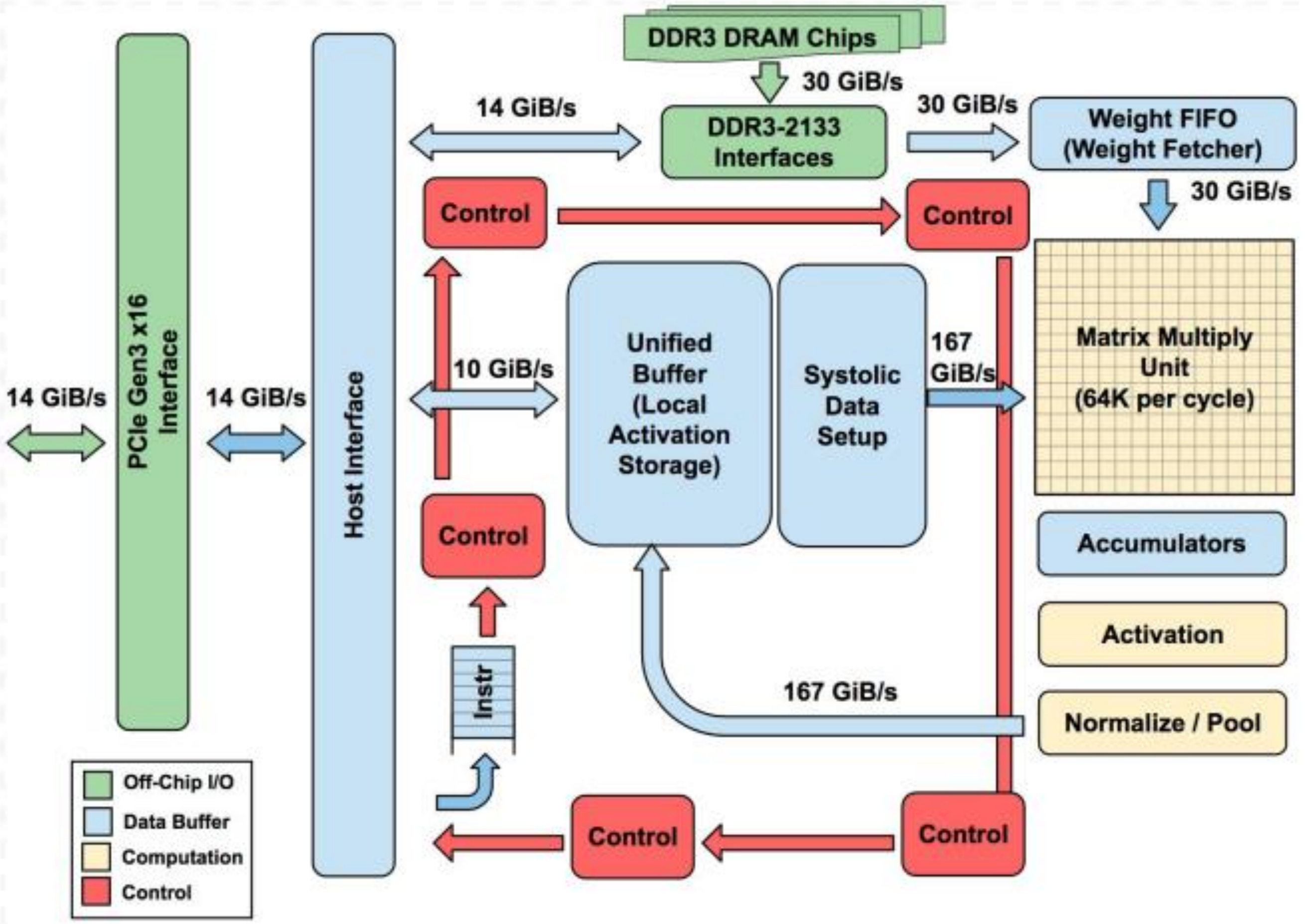
Weights



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

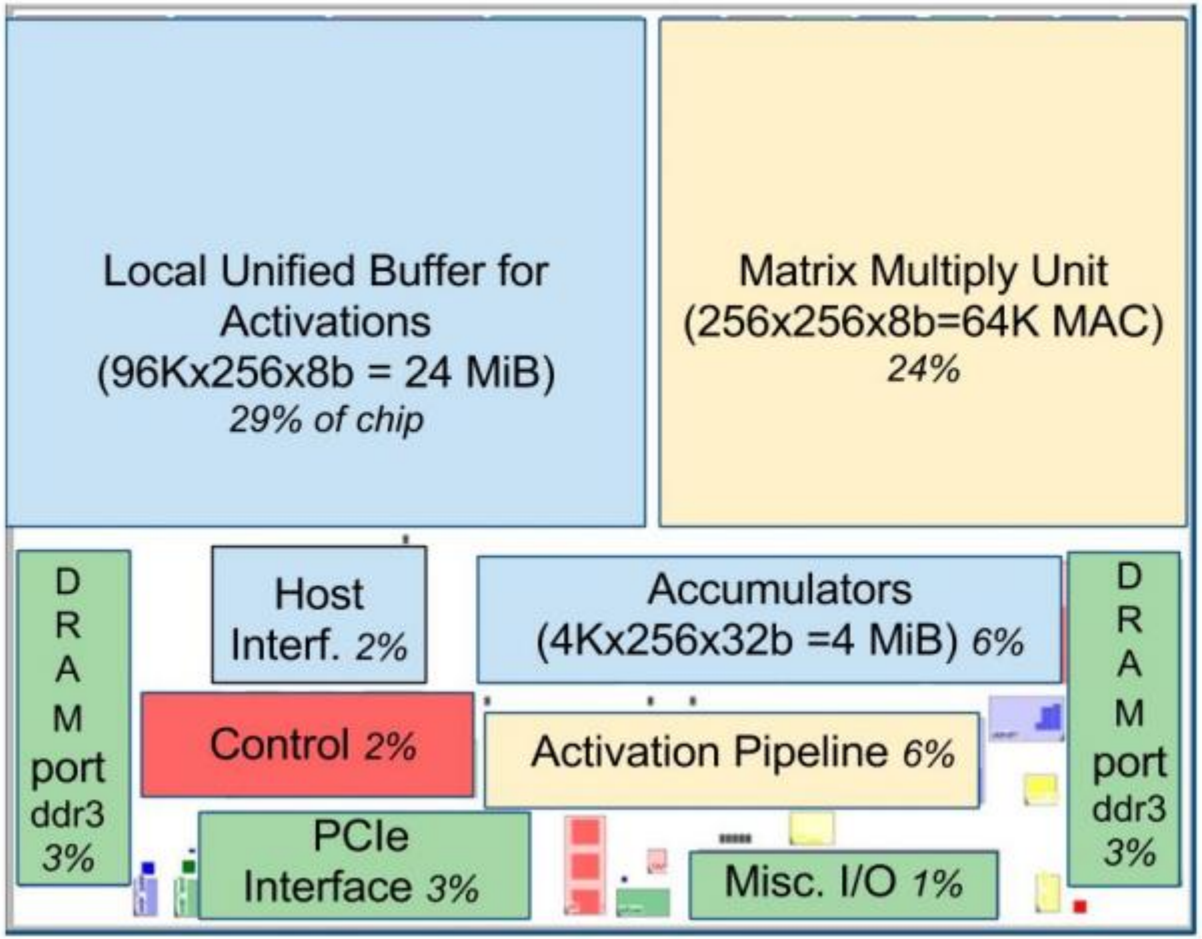
TPU architecture

- 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

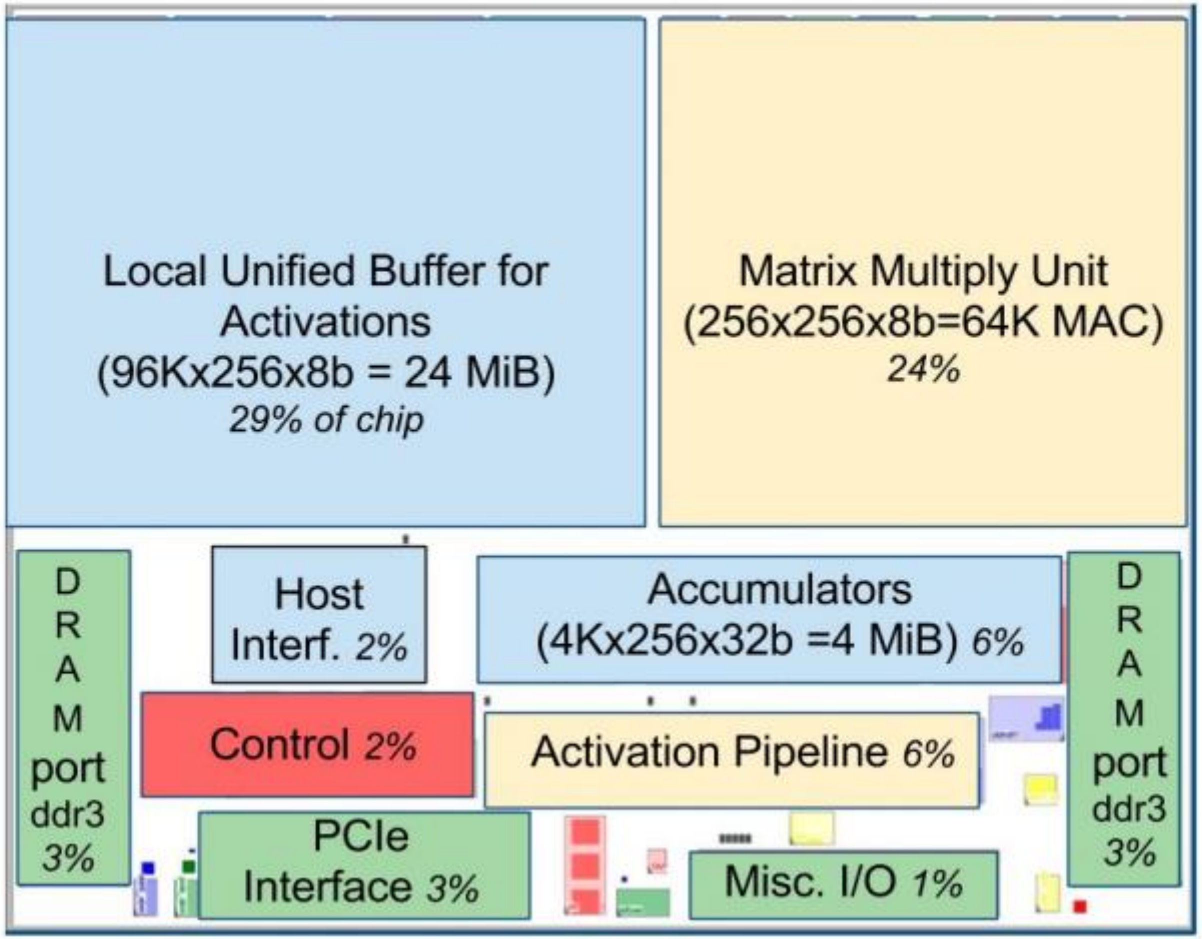


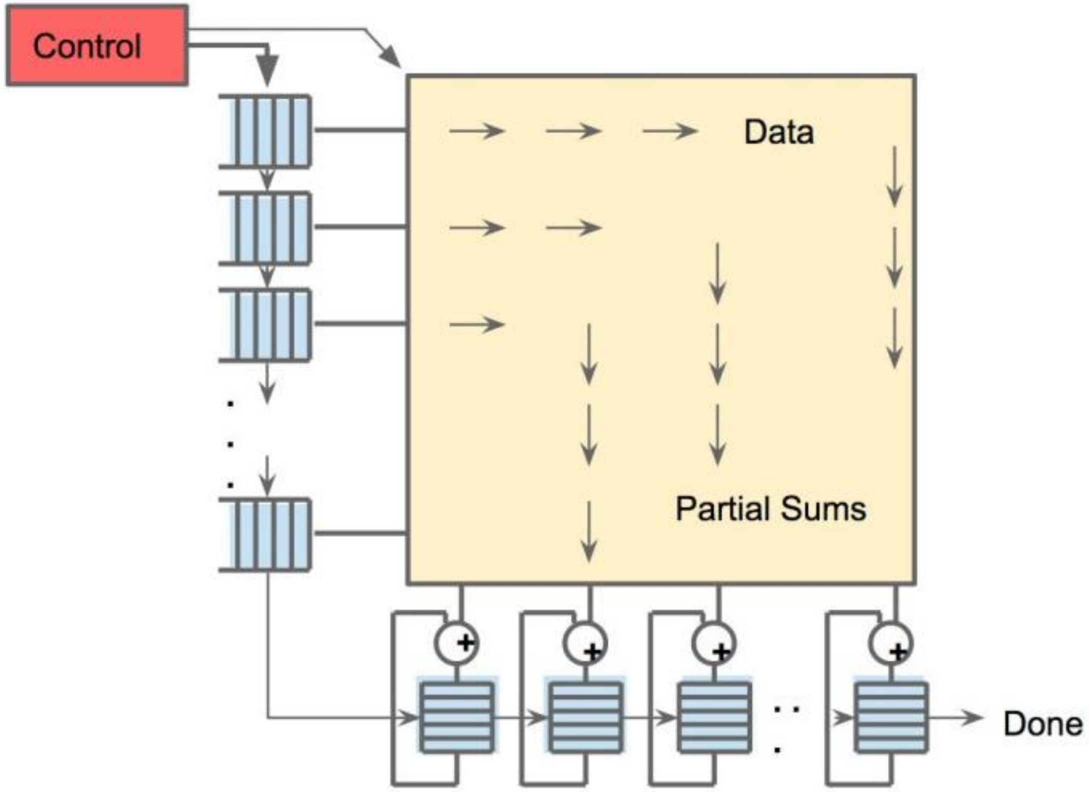
TPU architecture

- 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

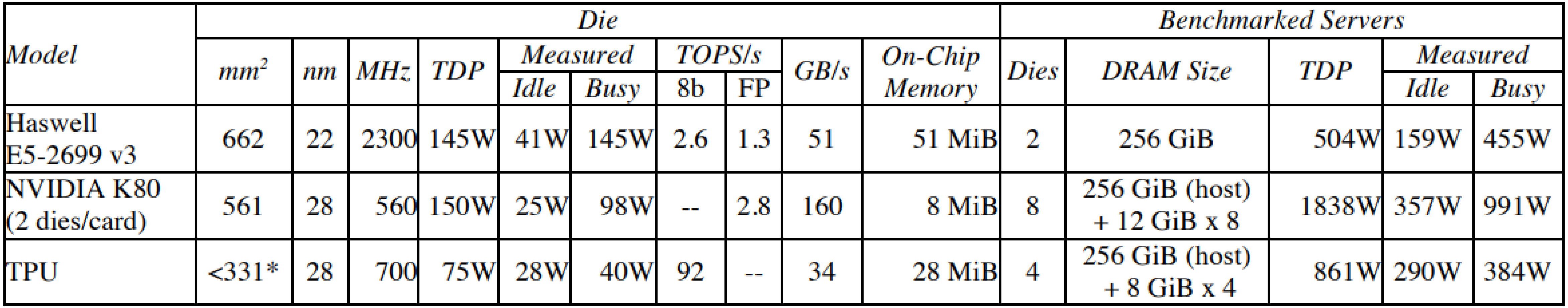

Local Unified Buffer for Activations (96Kx256x8b = 24 MiB) 29% of chip
Matrix Multiply Unit (256x256x8b=64K MAC) 24%
DRAM port ddr3 3%
Host Interf. 2%
Accumulators (4Kx256x32b =4 MiB) 6%
DRAM port ddr3 3%
Control 2%
Activation Pipeline 6%
PCIe Interface 3%
Misc. I/O 1%

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%



System configurations

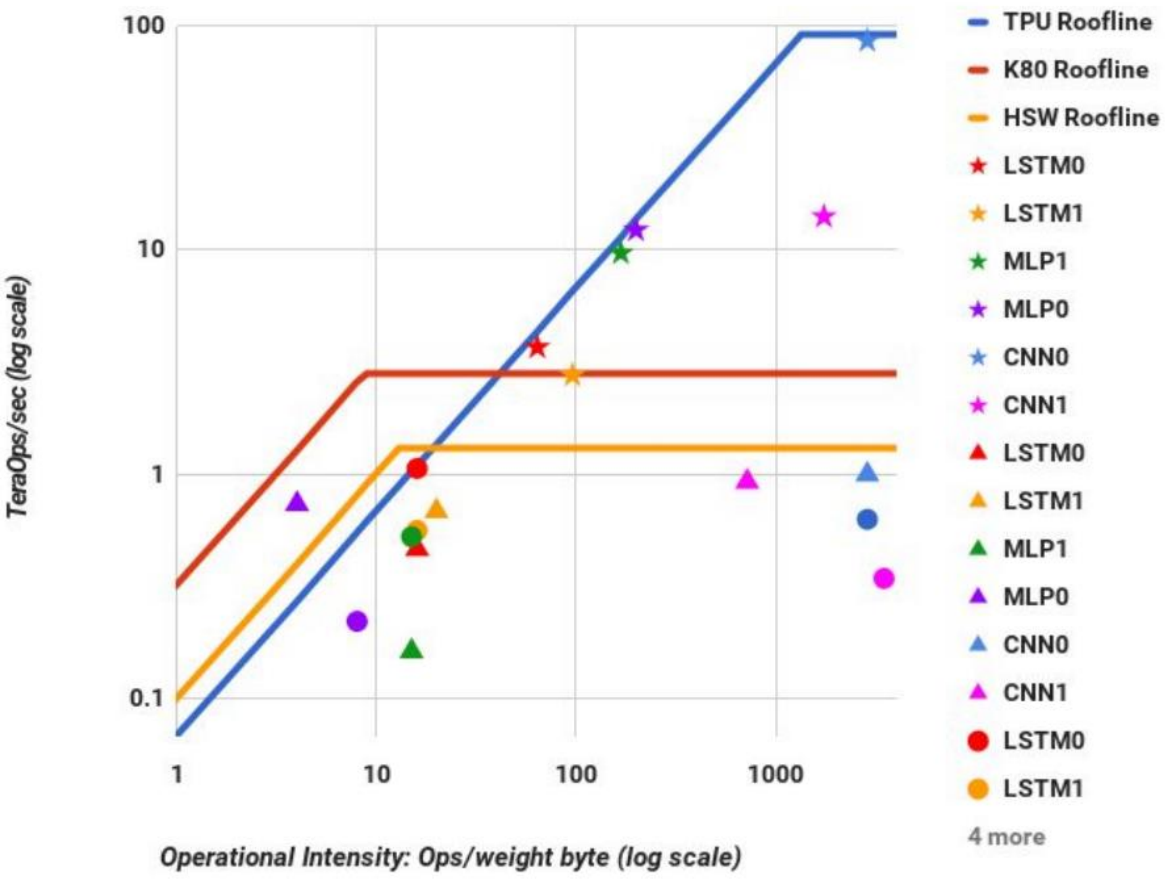


Model	*Die*										*Benchmarked Servers*				
	mm^2	*nm*	*MHz*	*TDP*	*Measured*		*TOPS/s*		*GB/s*	*On-Chip Memory*	*Dies*	*DRAM Size*	*TDP*	*Measured*	
					Idle	*Busy*	8b	FP						*Idle*	*Busy*
Haswell E5-2699 v3	662	22	2300	145W	41W	145W	2.6	1.3	51	51 MiB	2	256 GiB	504W	159W	455W
NVIDIA K80 (2 dies/card)	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

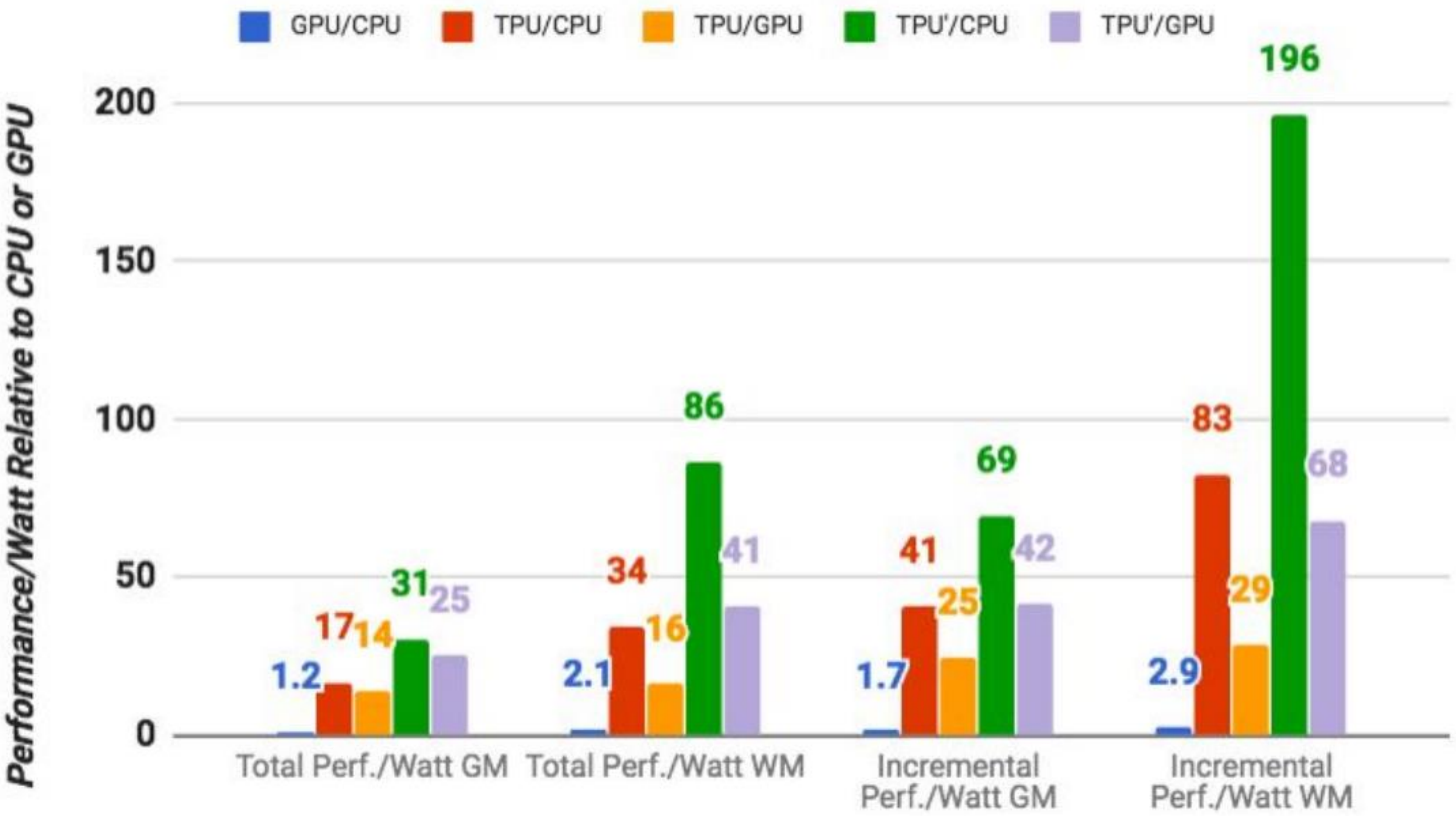
Performance

- “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

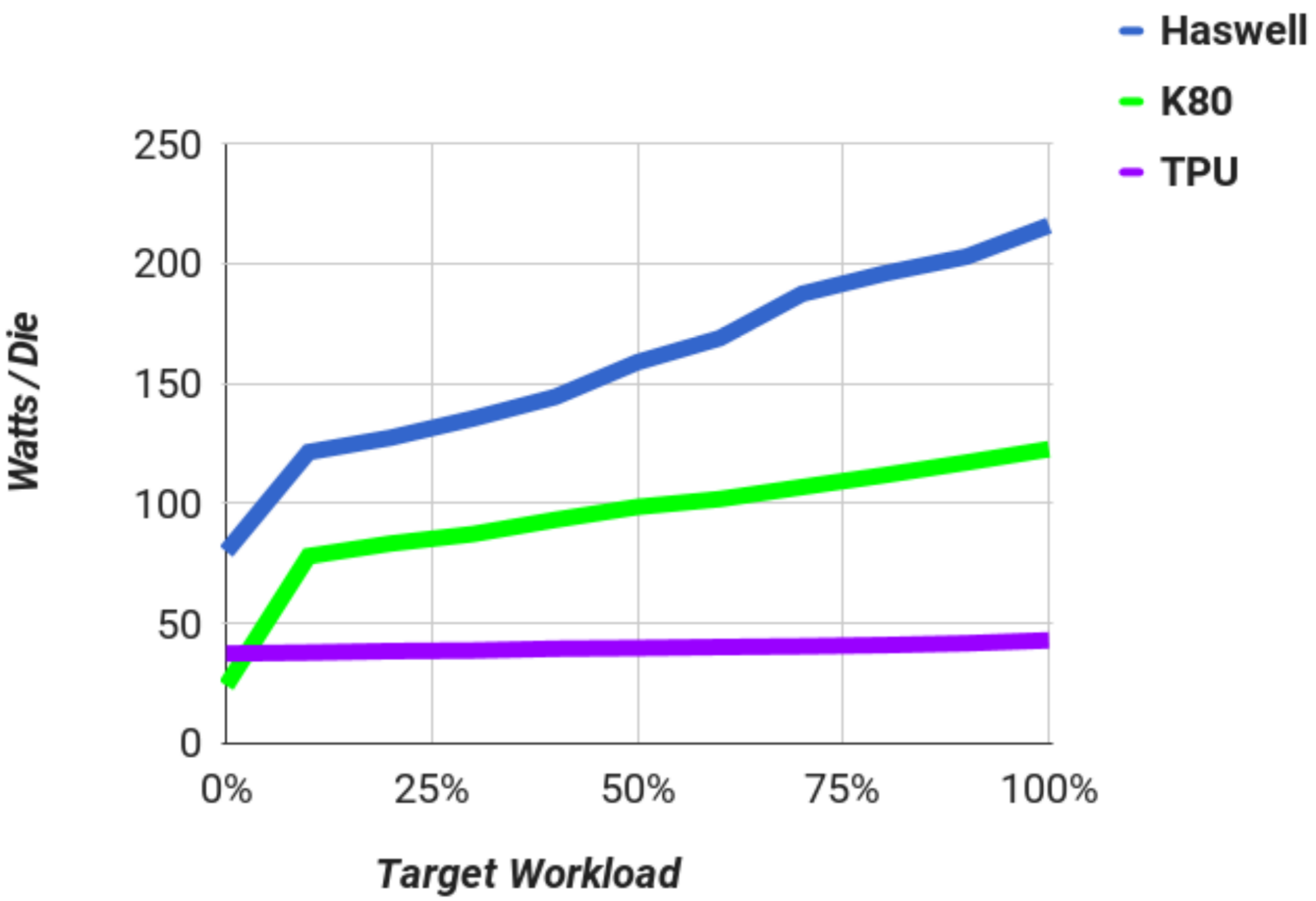
Log-Log Scale



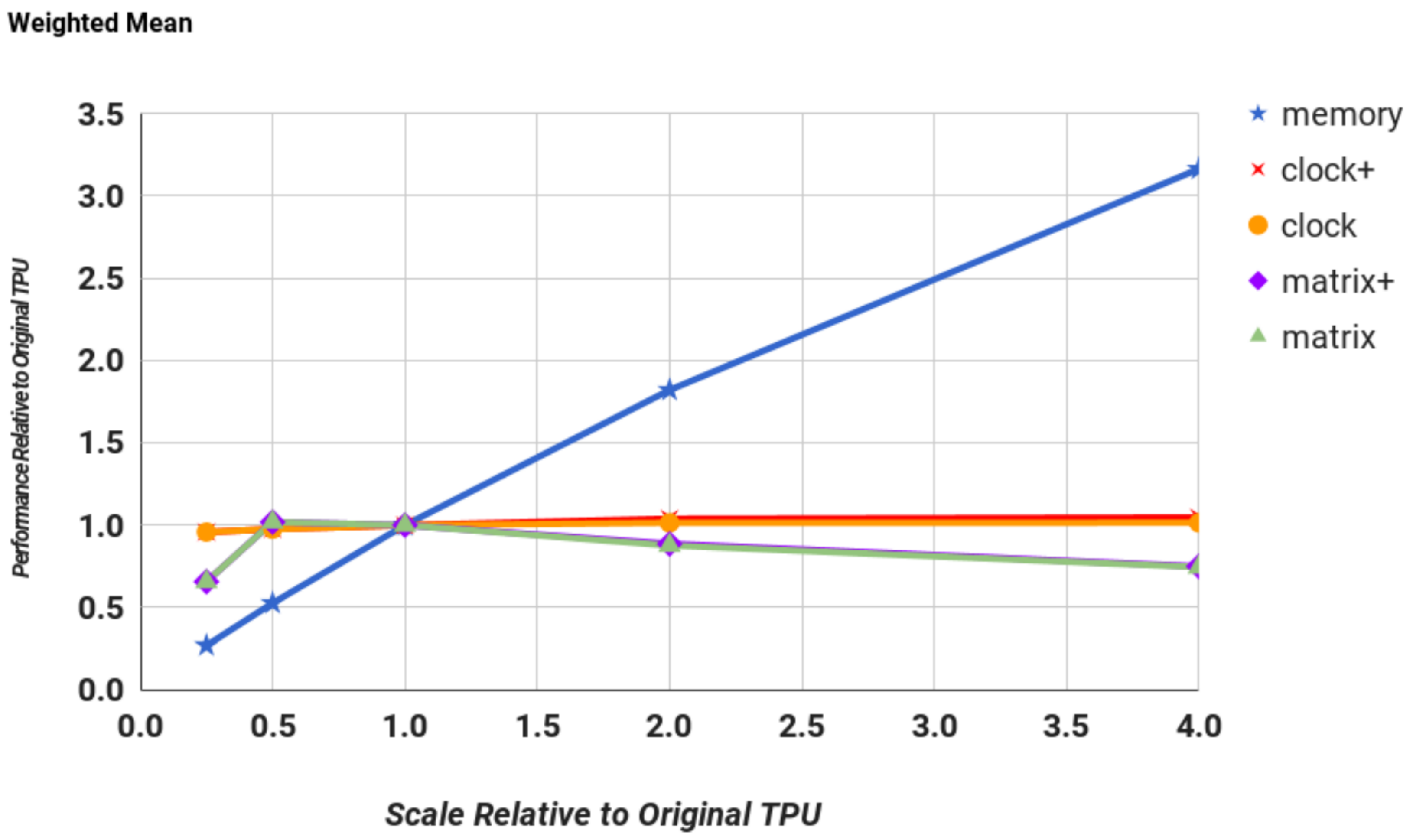
Performance – energy efficiency



Performance – energy proportionality

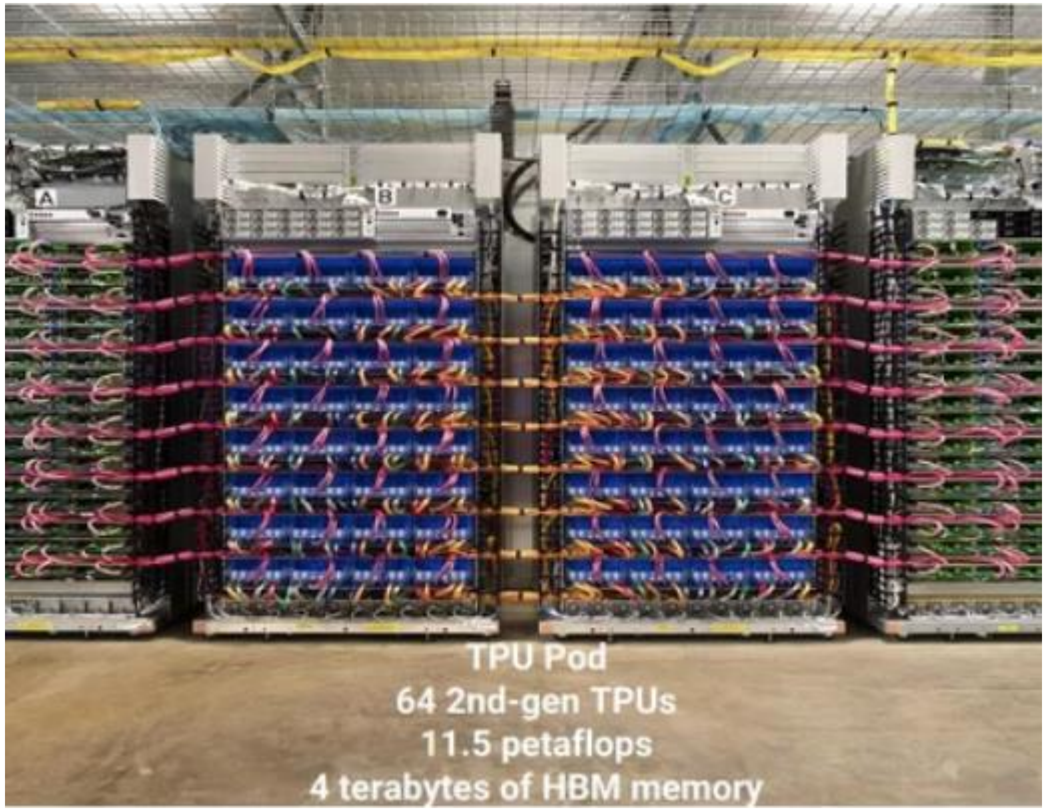


Design space exploration



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



http://www.tomshardware.com/news/tpu-v2-google-machine-learning,35370.html

Microsoft - Catapult

Google v. Microsoft

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

Takeaways

- Industry and academia have very different constraints

Takeaways

- Industry and academia have very different constraints

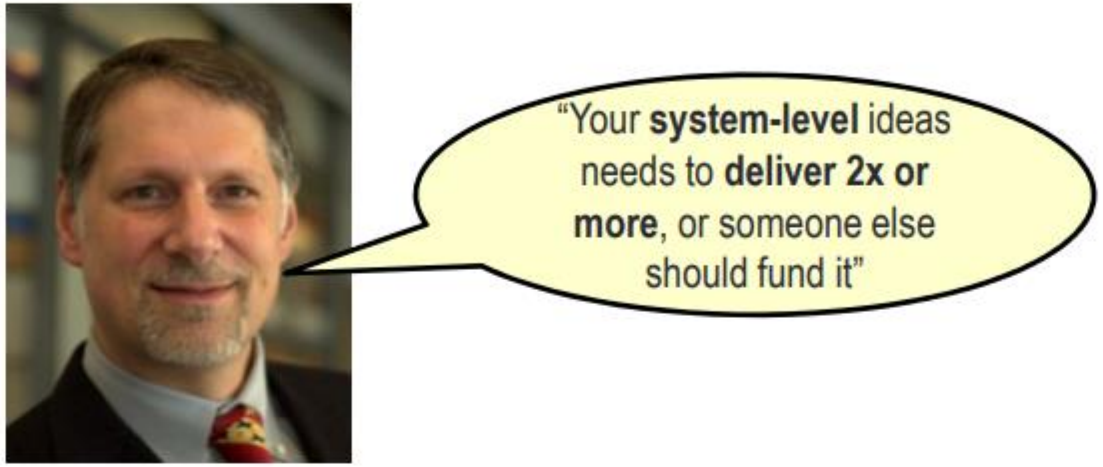
“Your **system-level** ideas needs to **deliver 2x or more**, or someone else should fund it”

Google

“Get me 10X in 15 months”

Takeaways

- Industry and academia have very different constraints
- Different goals may require fundamentally different tech


"Your **system-level** ideas needs to **deliver 2x or more**, or someone else should fund it"

Google

"Get me 10X in 15 months"

Takeaways

- Industry and academia have very different constraints
- Different goals may require fundamentally different tech
- Time and money dominate
  - (In academia, too!)

“Your **system-level** ideas needs to **deliver 2x or more**, or someone else should fund it”

Google

“Get me 10x in 15 months”

Large-Scale Reconfigurable Computing in a Microsoft Datacenter

Andrew Putnam – Microsoft

Hot Chips 26 – Aug 12, 2014

Microsoft Cloud Services

Windows Azure

bing

Microsoft SQL Server

XBOX LIVE

COSMOS

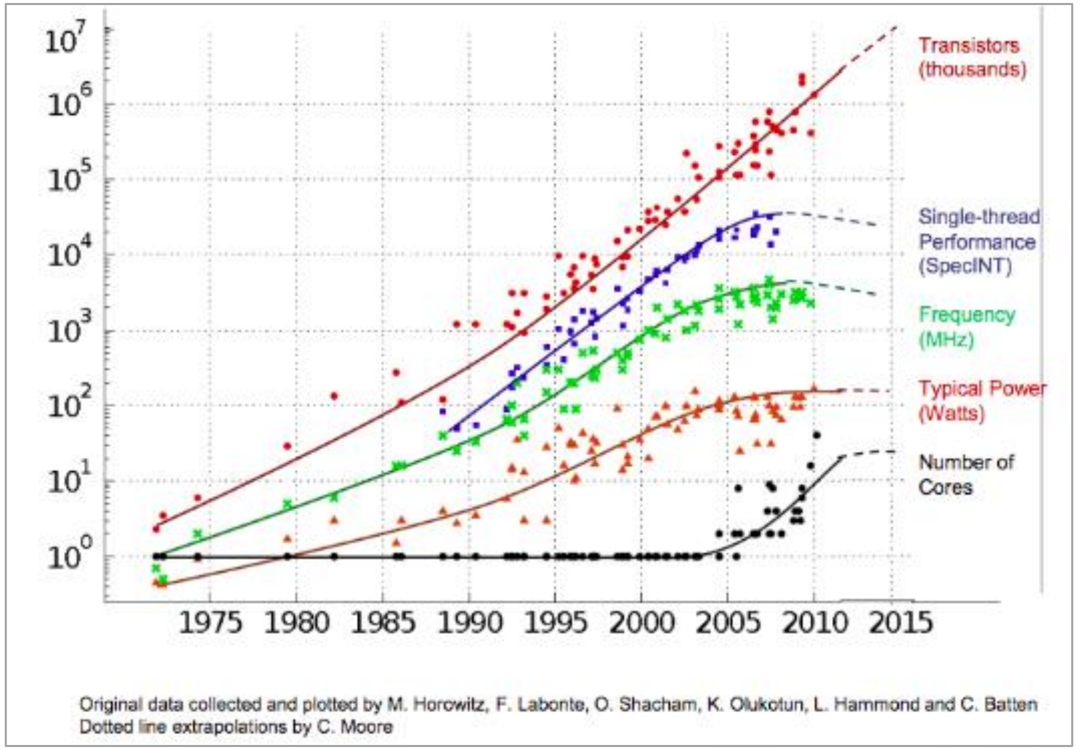
skype

msn

OneDrive

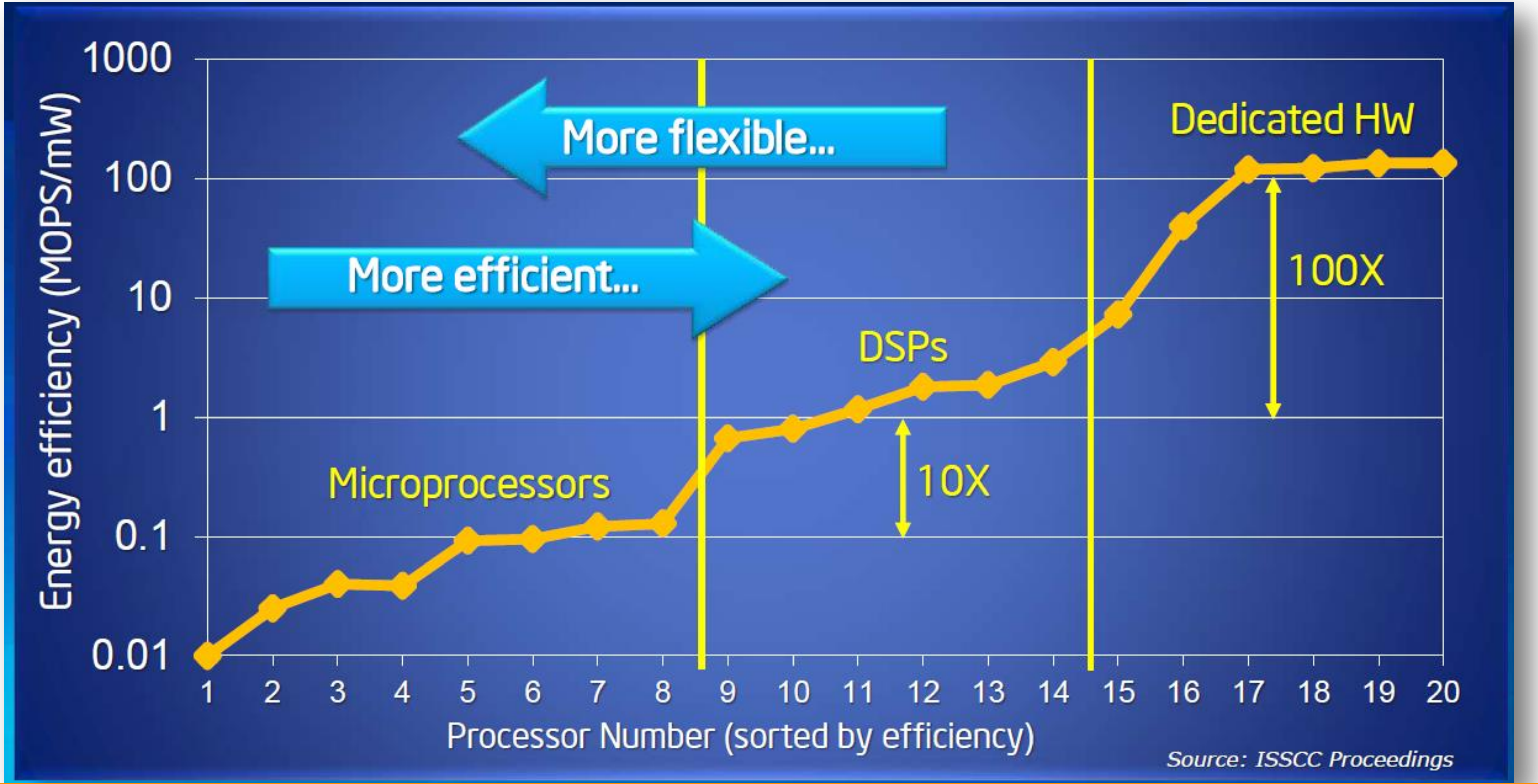
Outlook.com

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

$$\text{Capabilities, Costs} \propto \frac{Performance/Watt}{\$}$$



Source: ISSCC Proceedings

Increase Efficiency with Hardware Specialization

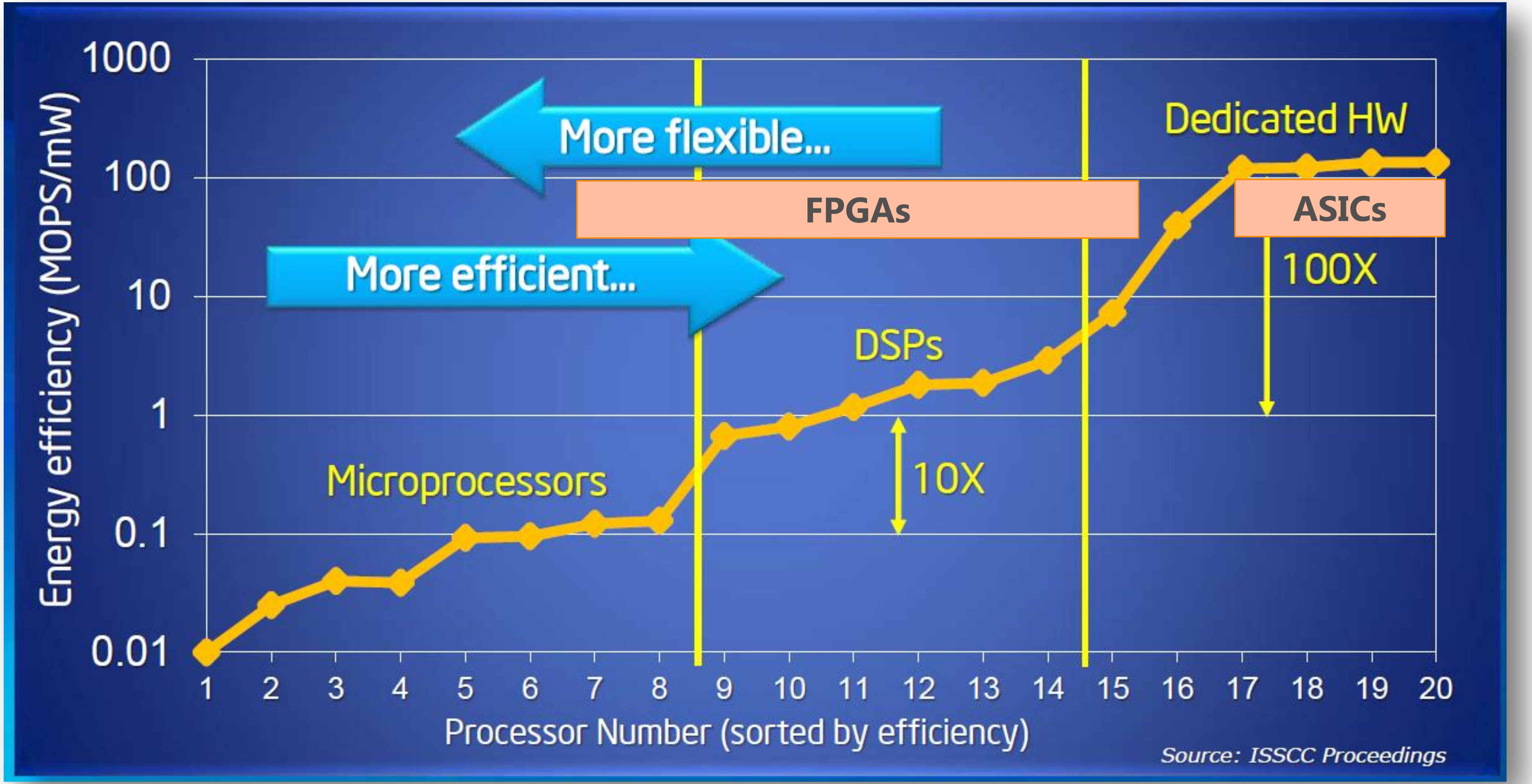
Datacenter Environment

- 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

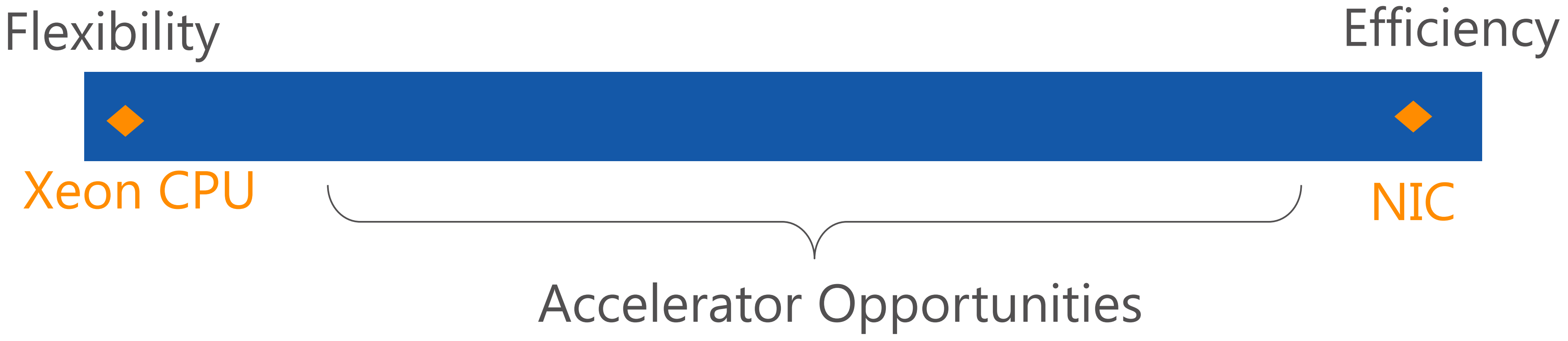
The paradox: Specialization *and* homogeneity

Efficiency via Specialization



Source: Bob Broderson, Berkeley Wireless group

One Application's Accelerator



Flexibility

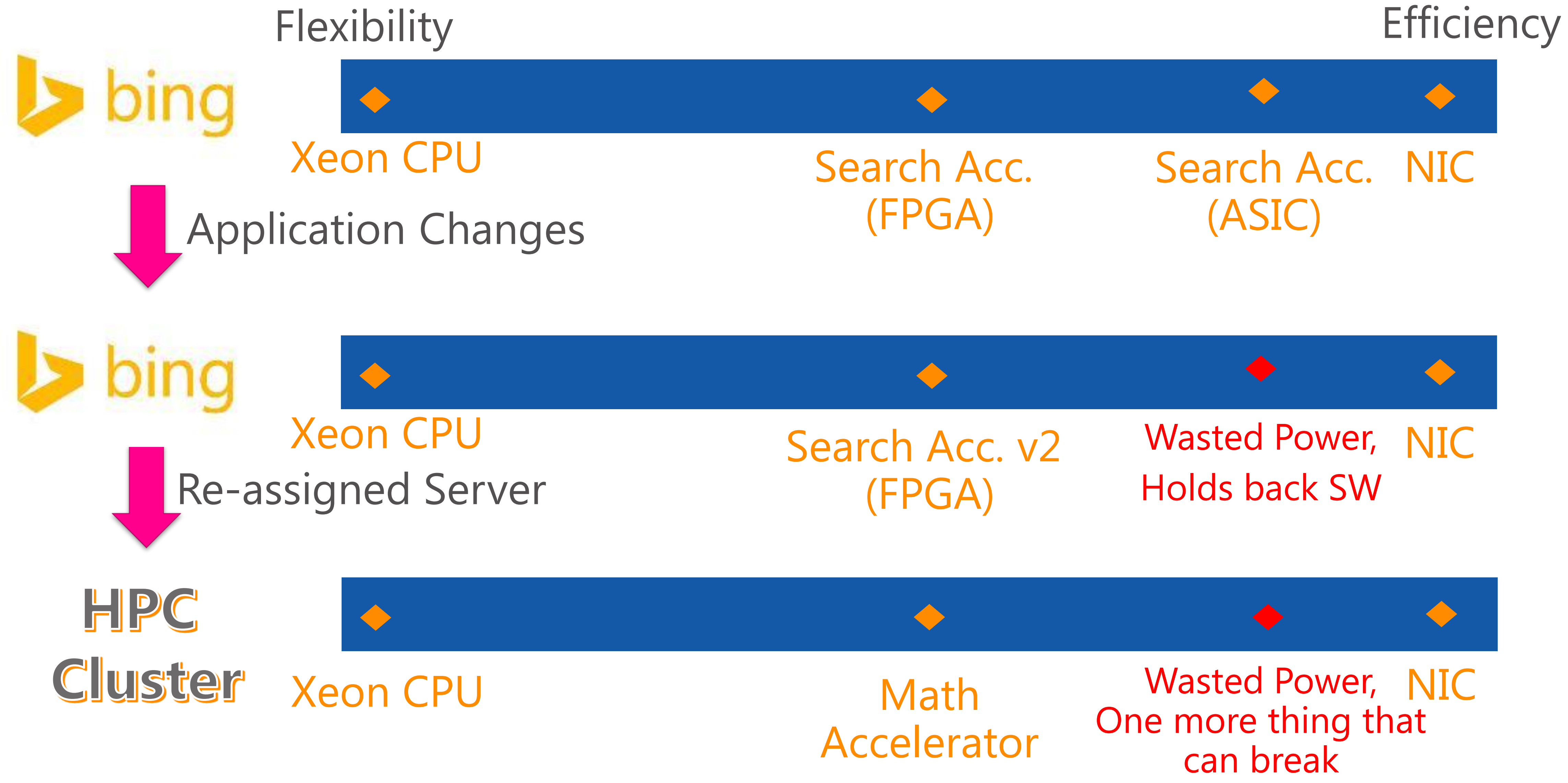
Efficiency

Xeon CPU

NIC

Accelerator Opportunities

One Application’s Accelerator



Flexibility

Efficiency

bing

Xeon CPU

Search Acc. (FPGA)

Search Acc. (ASIC)

NIC

Application Changes

bing

Xeon CPU

Search Acc. v2 (FPGA)

Wasted Power, Holds back SW

NIC

Re-assigned Server

HPC Cluster

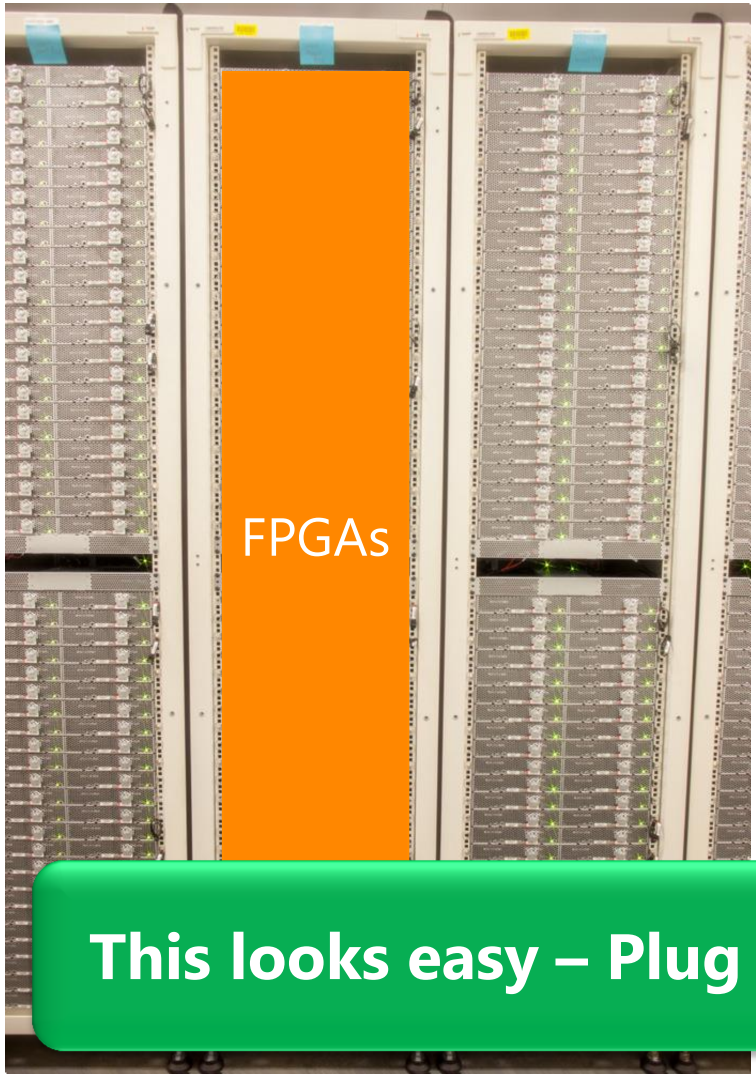
Xeon CPU

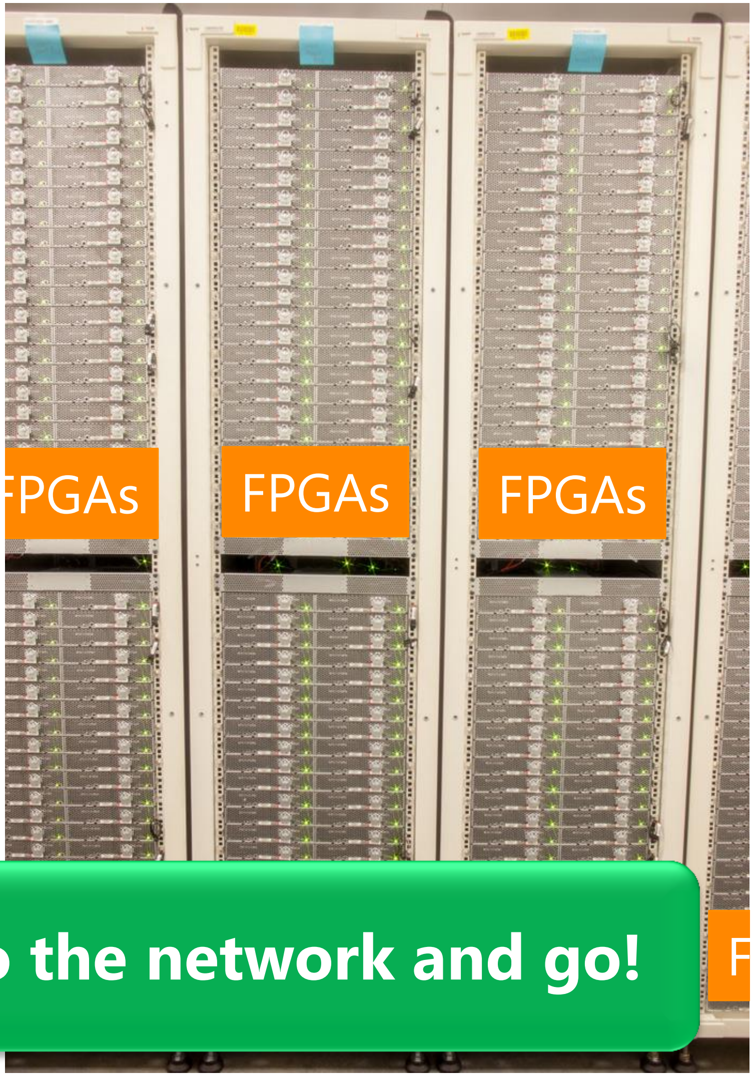
Math Accelerator

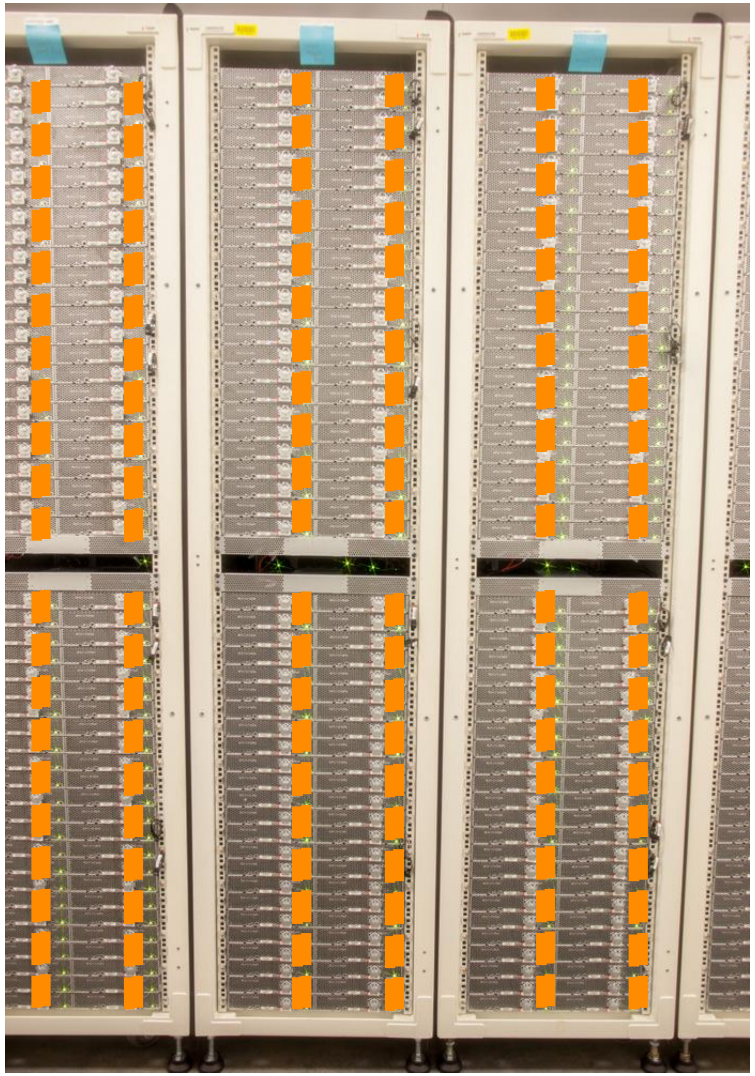
Wasted Power, One more thing that can break

NIC

Integrating FPGAs into the Datacenter



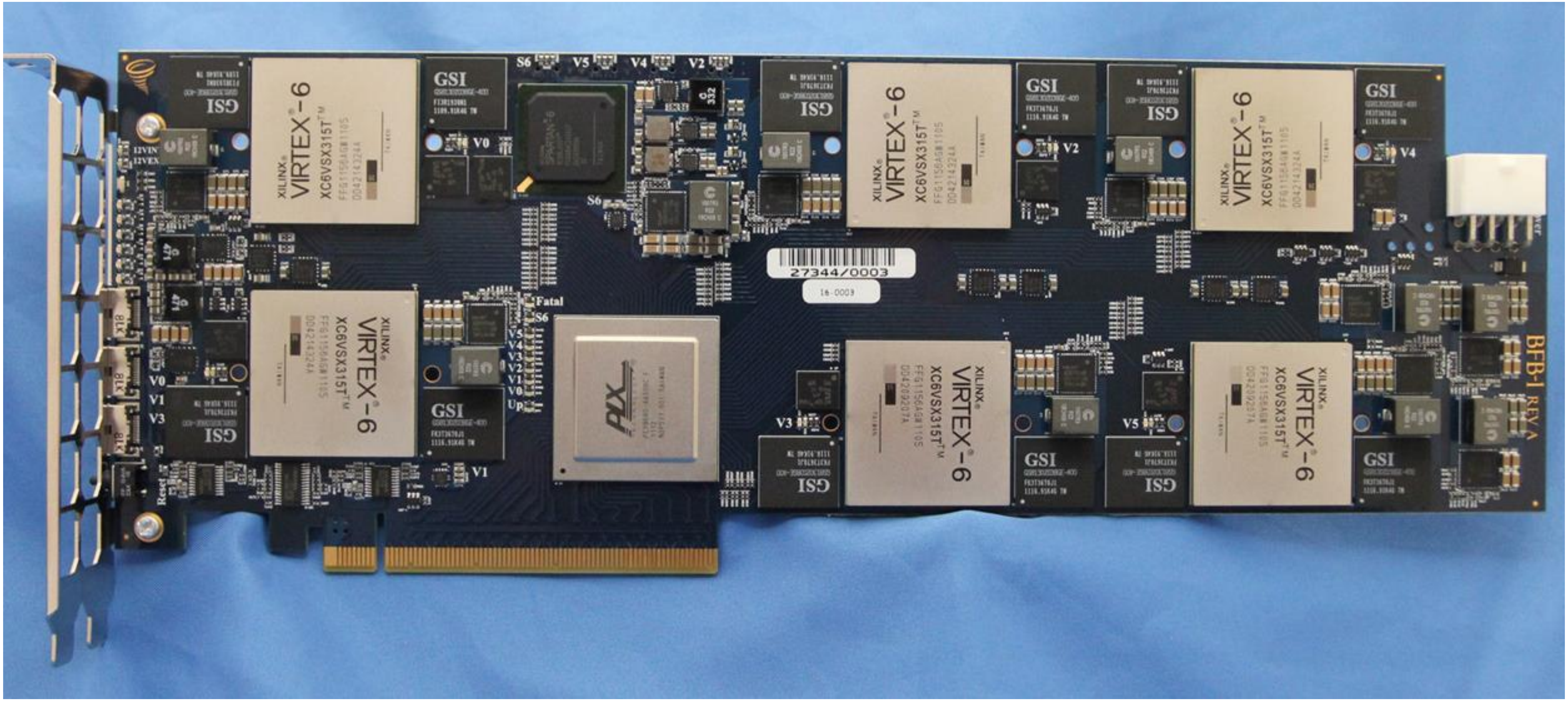




This looks easy – Plug into the network and go!

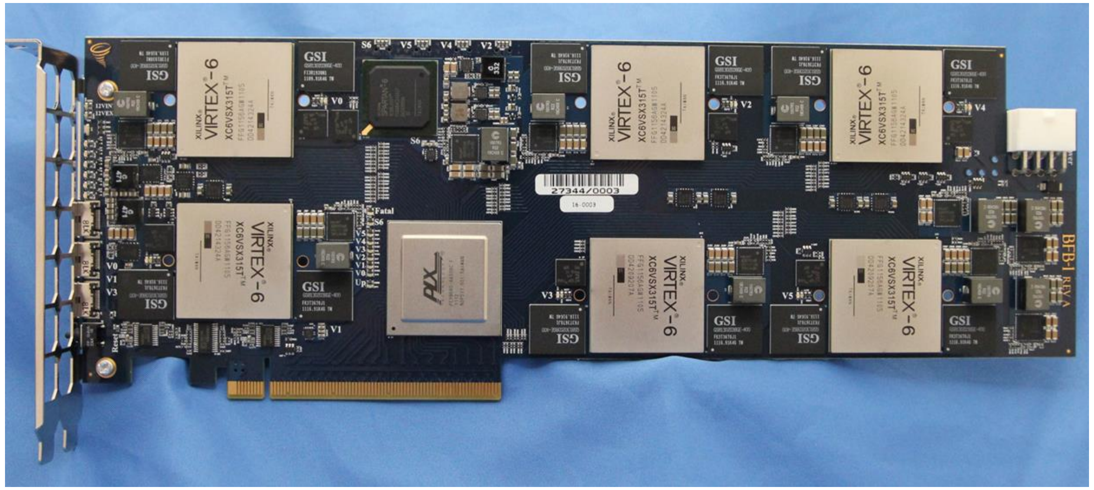
Centralized **Distributed**

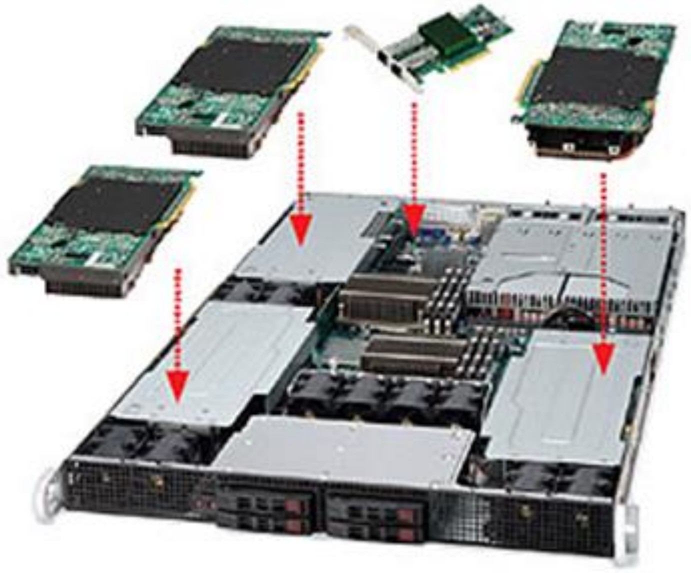
Prototype #1: BFB Board



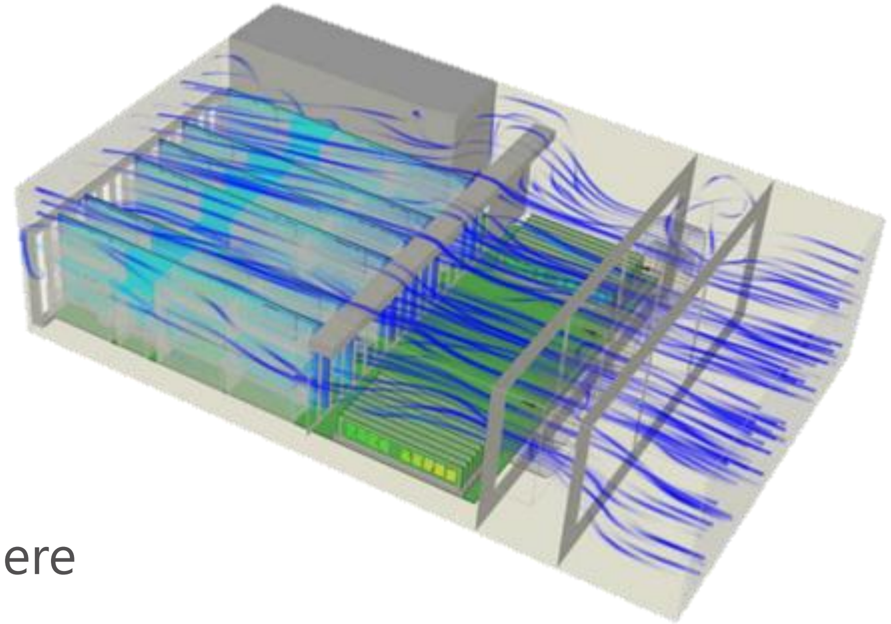
Prototype #1: BFB board

- 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



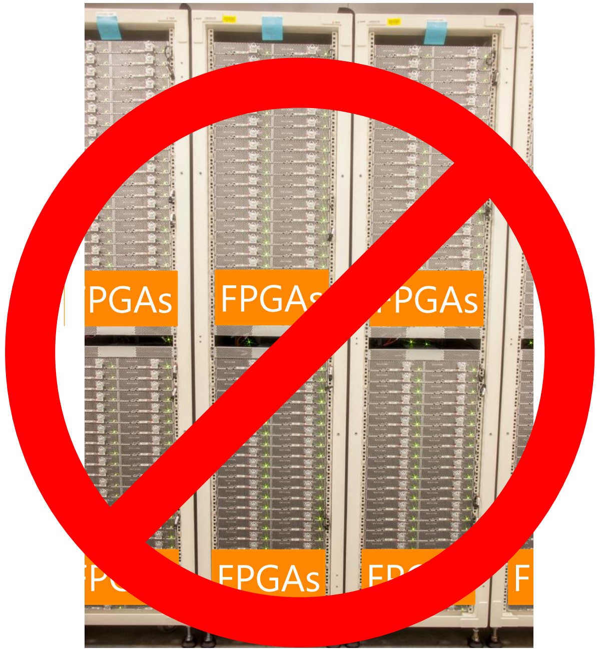


- 1U, 2U, or 4U rack-mounted
- 1/2/4 x 10Ge ports
- Up to 4 PCIe x16 slots
- 2 sockets, 6-core Intel Westmere



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

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

Don’t Cost Too Much

<30% Cost of Current Servers

Don’t Burn Too Much Power

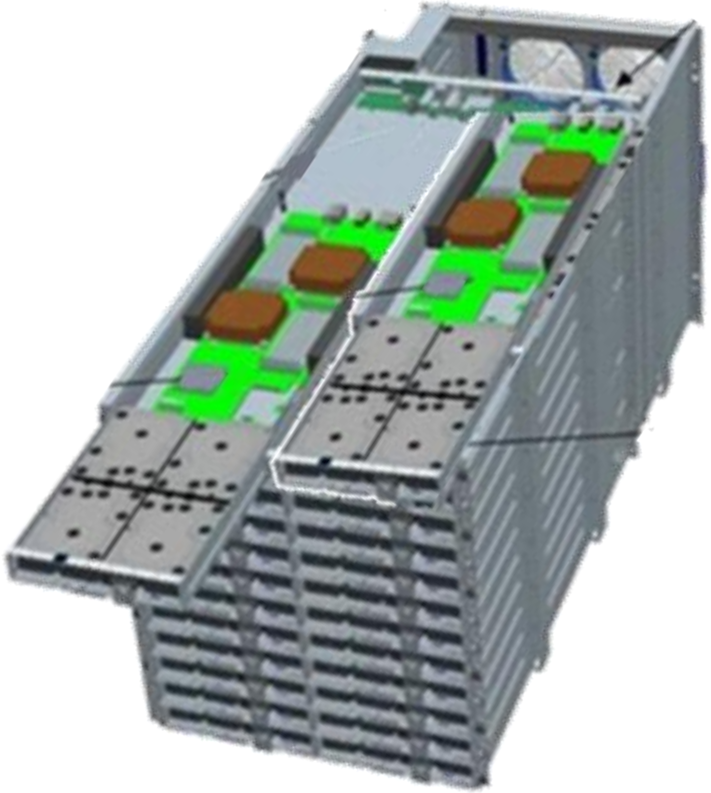
<10% Power Draw
(25W max, all from PCIe)

Don’t Break Anything

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

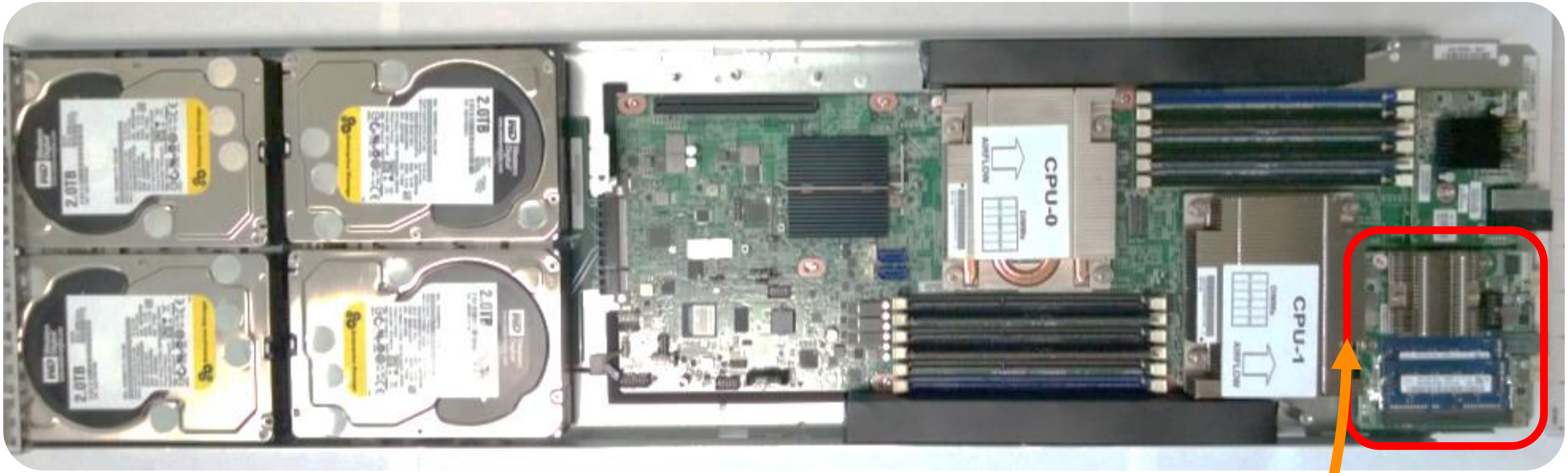
Datacenter Servers

- 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

Air flow

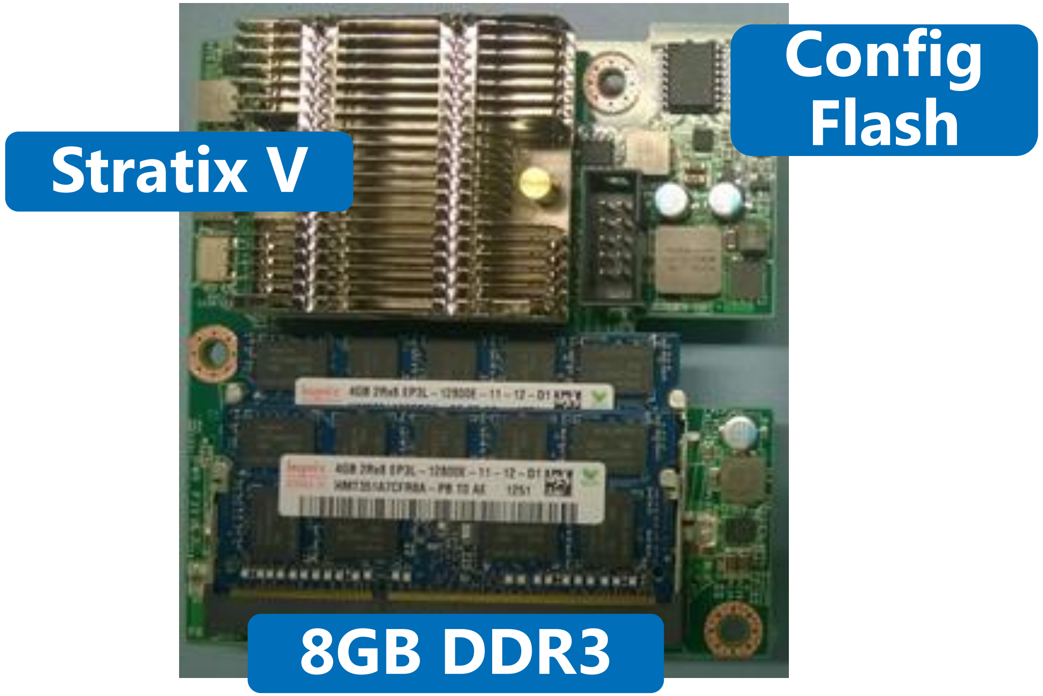
200 LFM

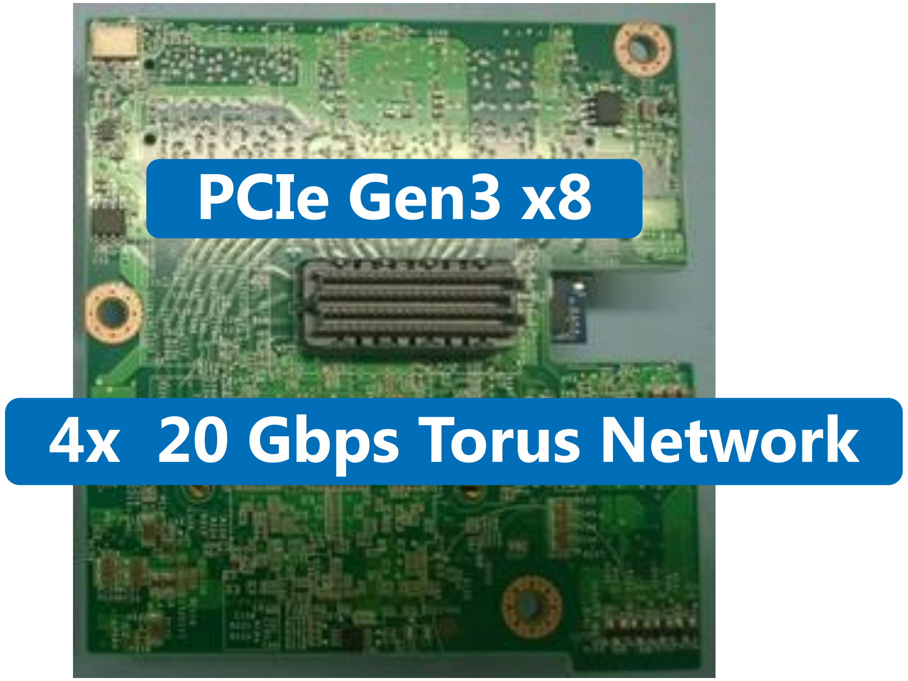
68 ^{0}C Inlet

Catapult FPGA Accelerator Card

- Altera Stratix V GS D5
  - 172k ALMs, 2,014 M20Ks, 1,590 DSPs
- 8GB DDR3-1333
- 32 MB Configuration Flash

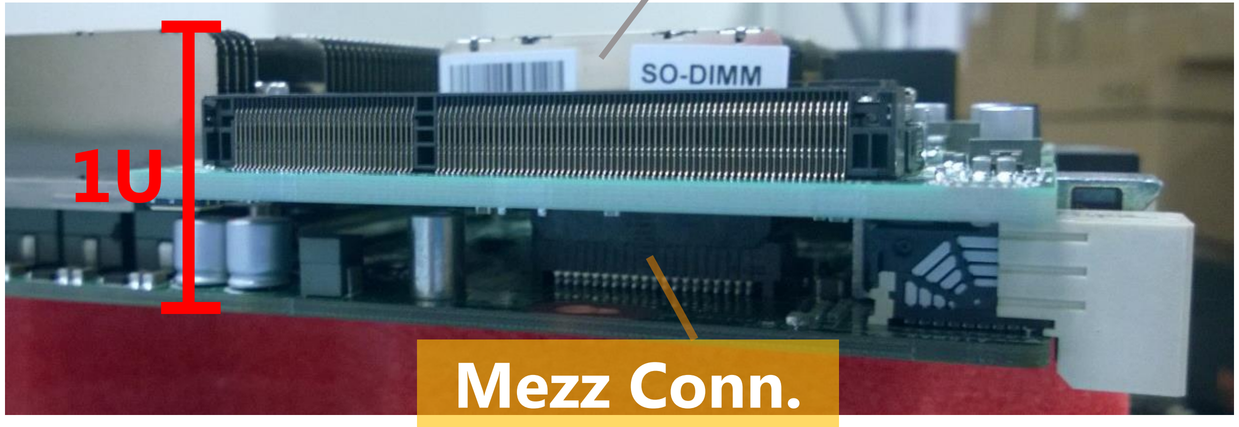
- PCIe Gen 3 x8
- 8 lanes to Mini-SAS SFF-8088 connectors
- Powered by PCIe slot

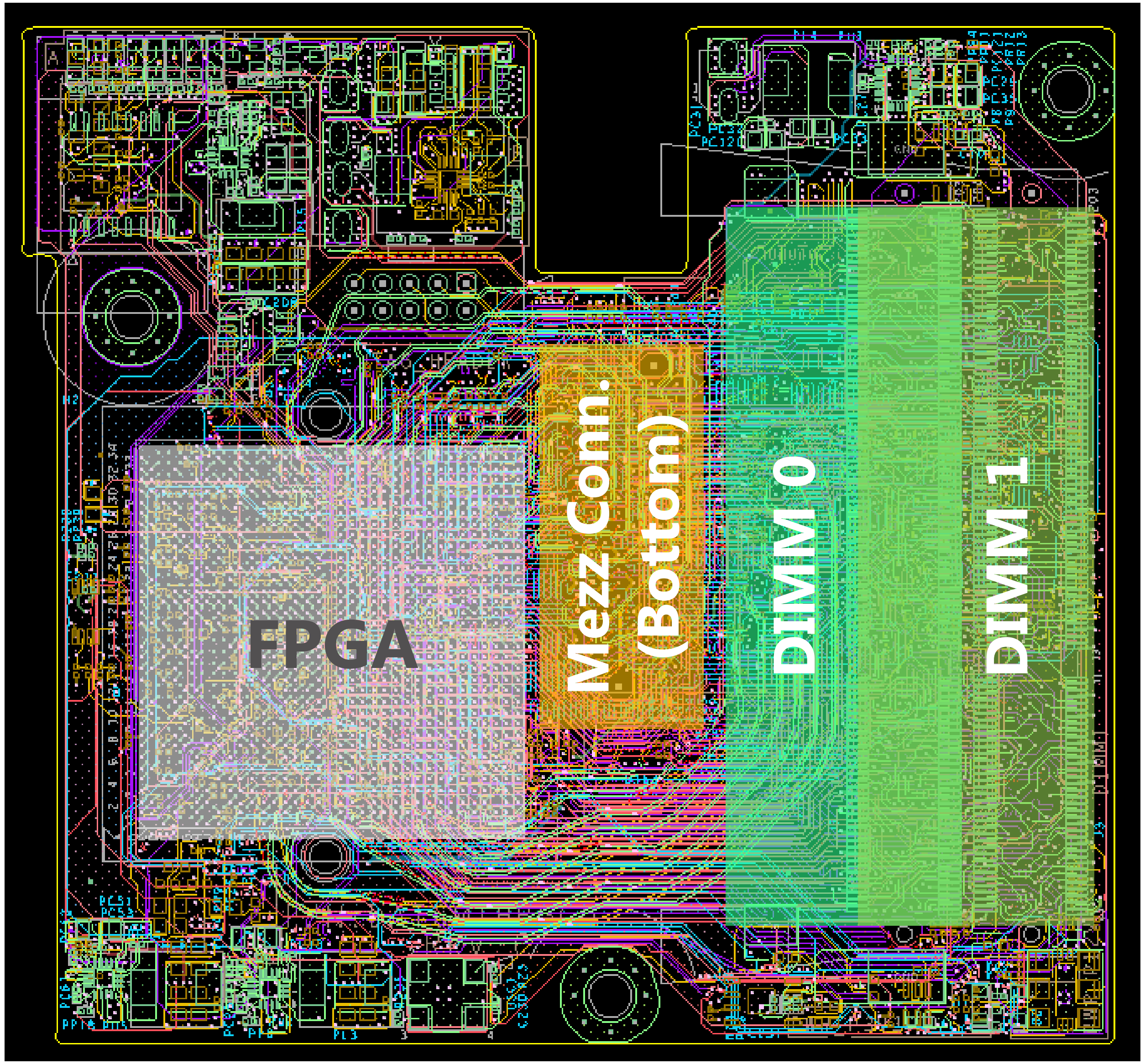




Board Details

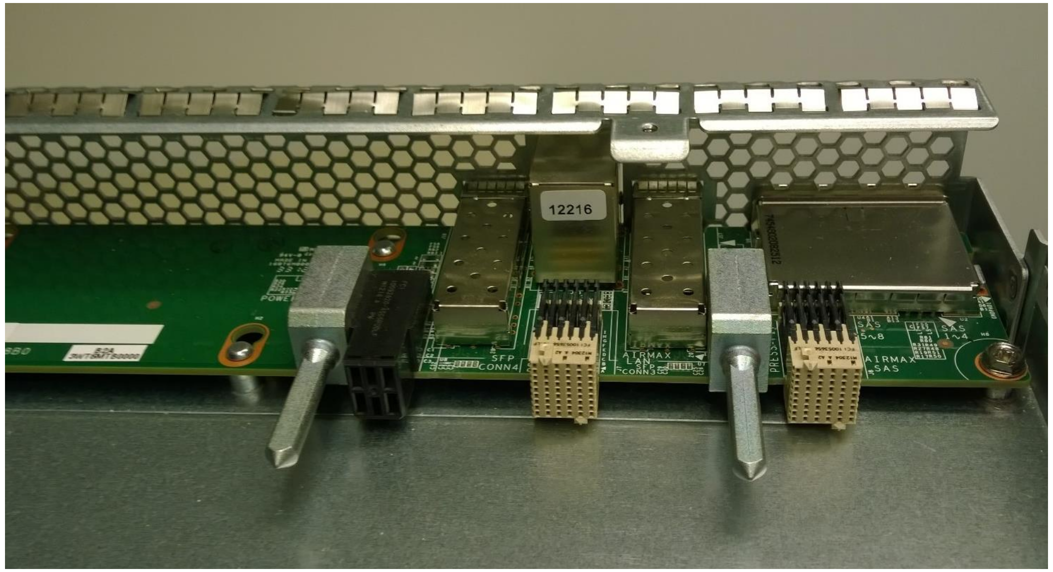
- 16 Layer, FR408
- 9.5cm x 8.8cm x 115.8 mil
- 35mm x 35mm FPGA
- 14.2mm high heatsink



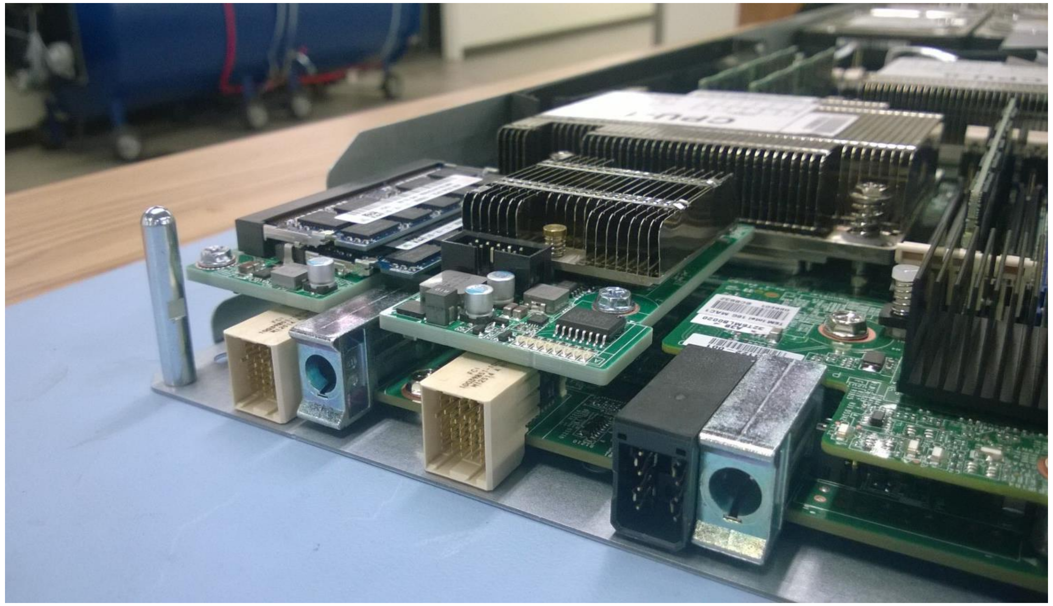


Board / Server Integration

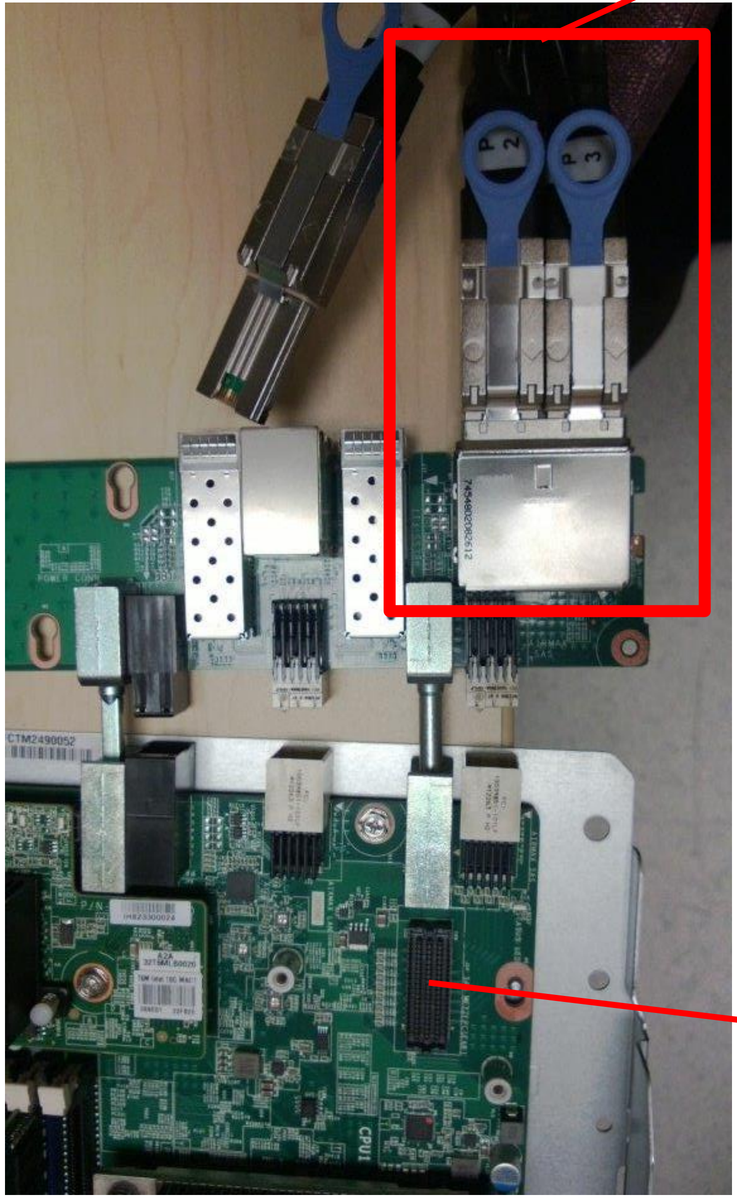
I/O Backplane



Server w/ Catapult Board



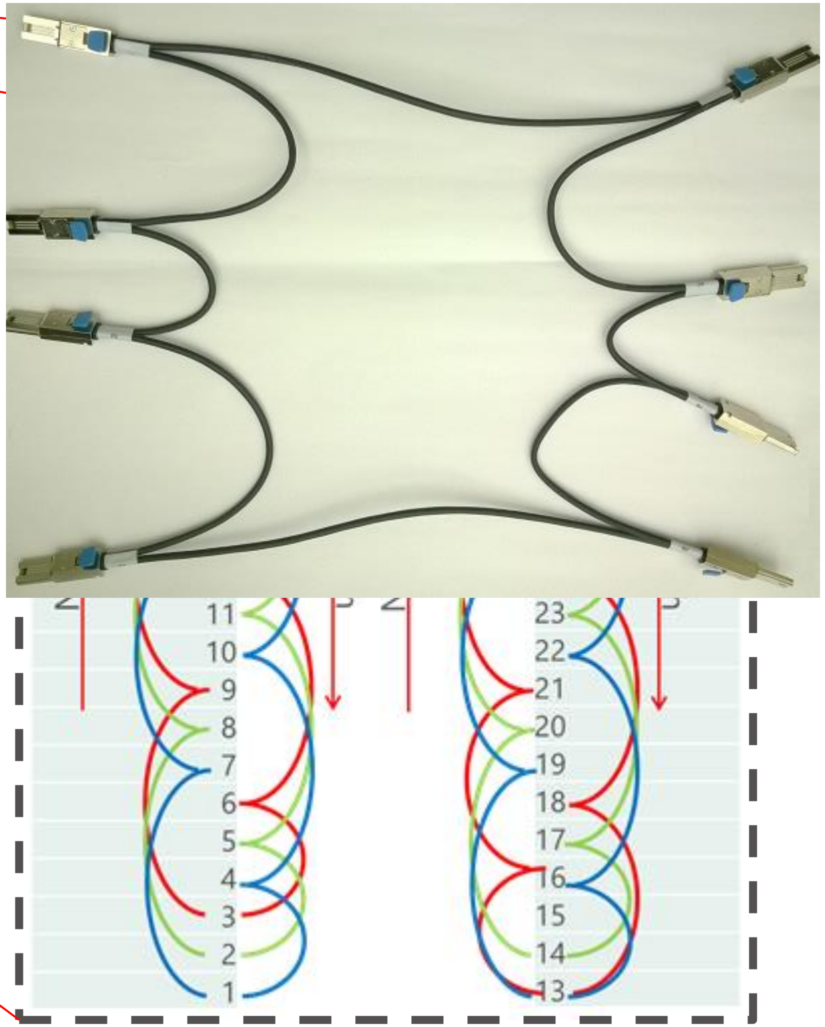
Catapult Network Cables
(Mini-SAS / SFF-8088)



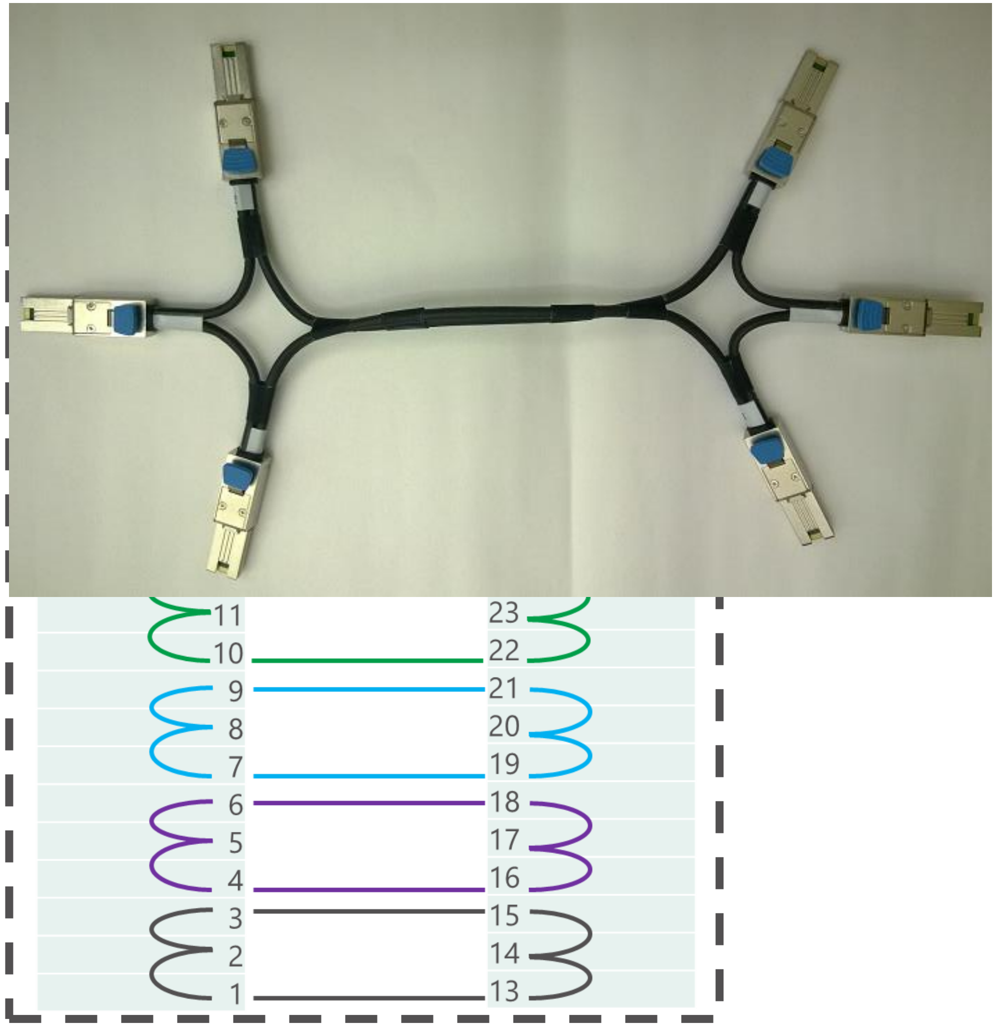
Boards Connected Together

Catapult Board
Mezzanine Slot

6x8 Torus in a 2x24 Server Layout



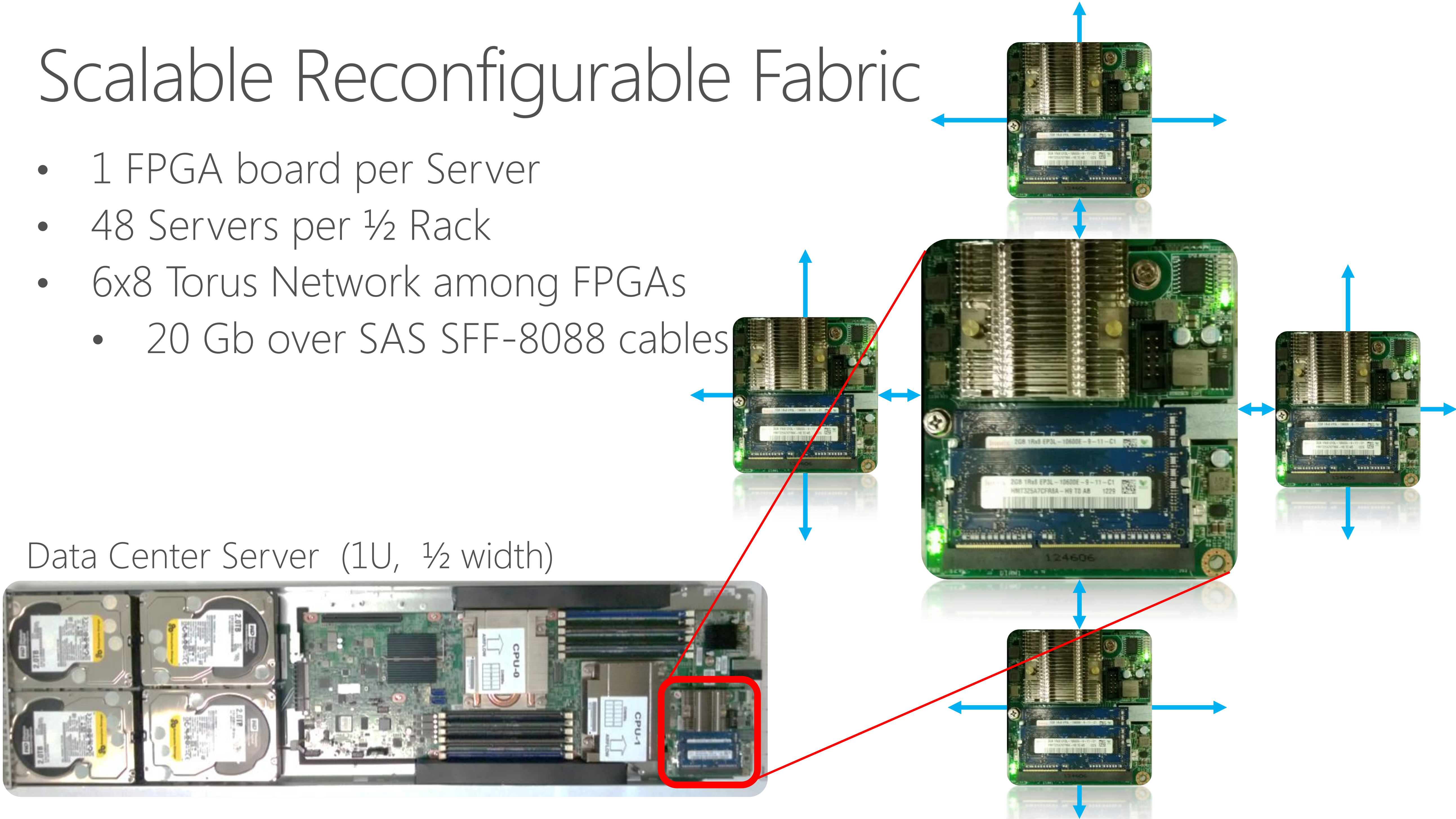
8-Shell Cables



6-Shell Cables

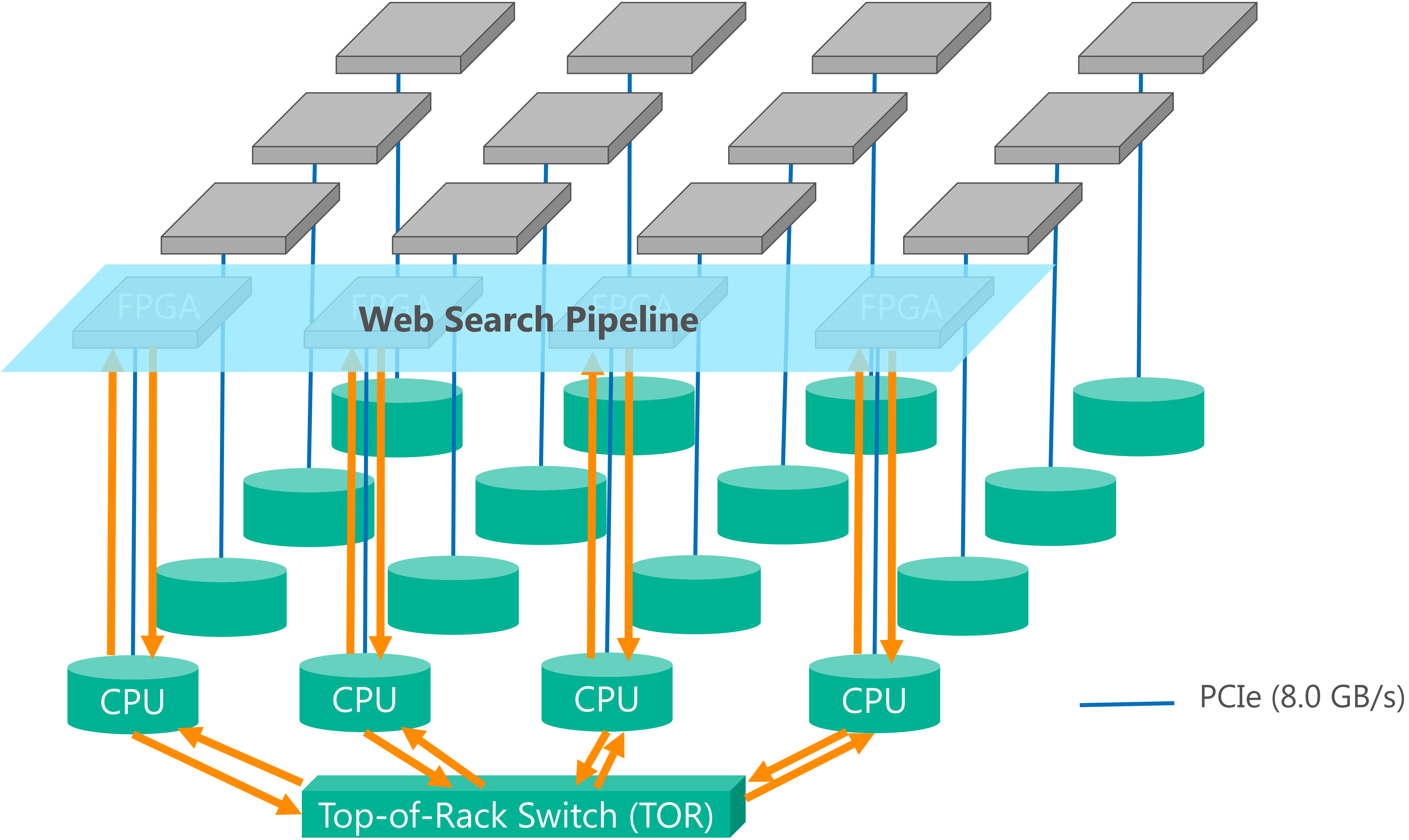
Scalable Reconfigurable Fabric

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

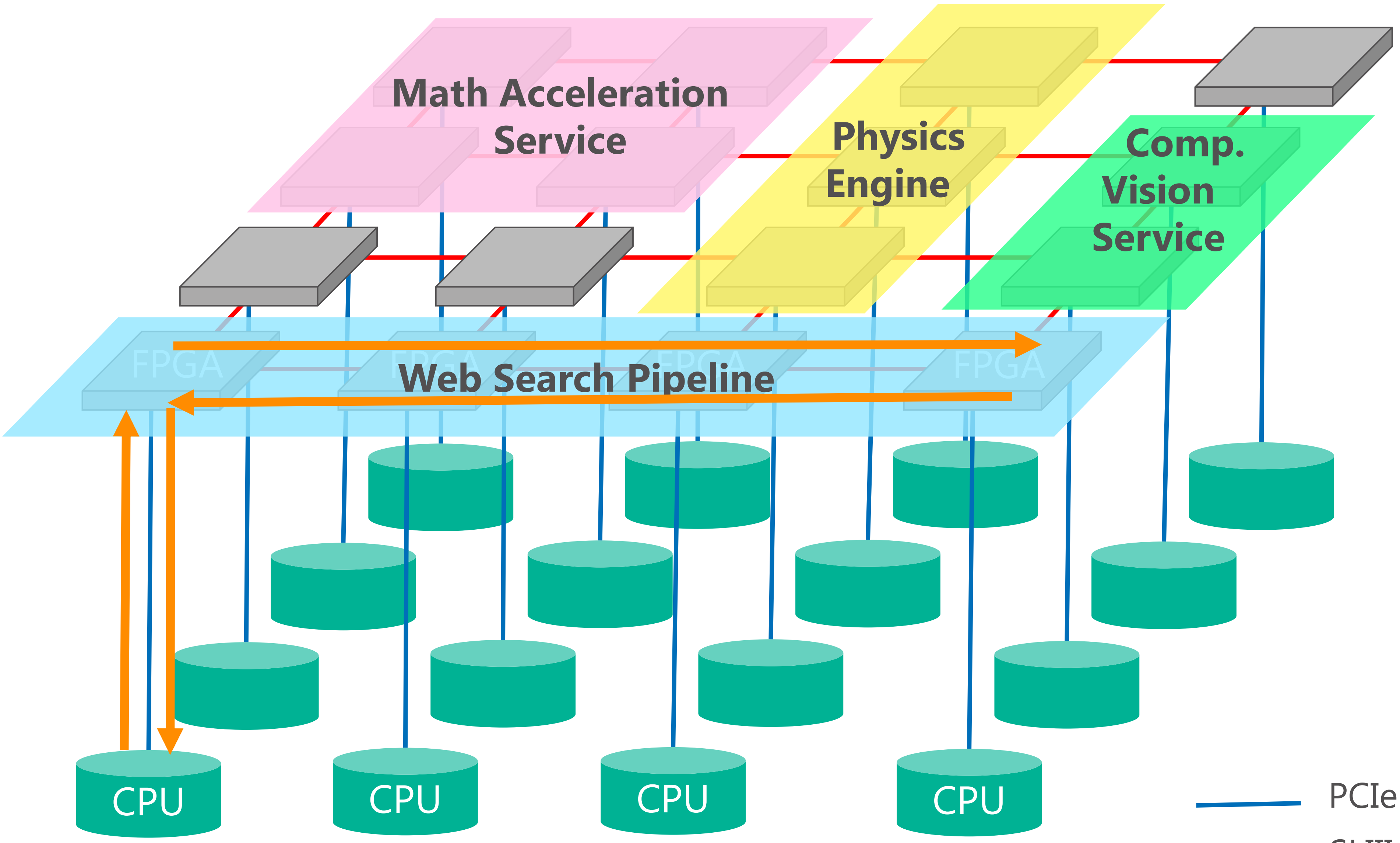


Data Center Server (1U, ½ width)

An Elastic Reconfigurable Fabric



An Elastic Reconfigurable Fabric



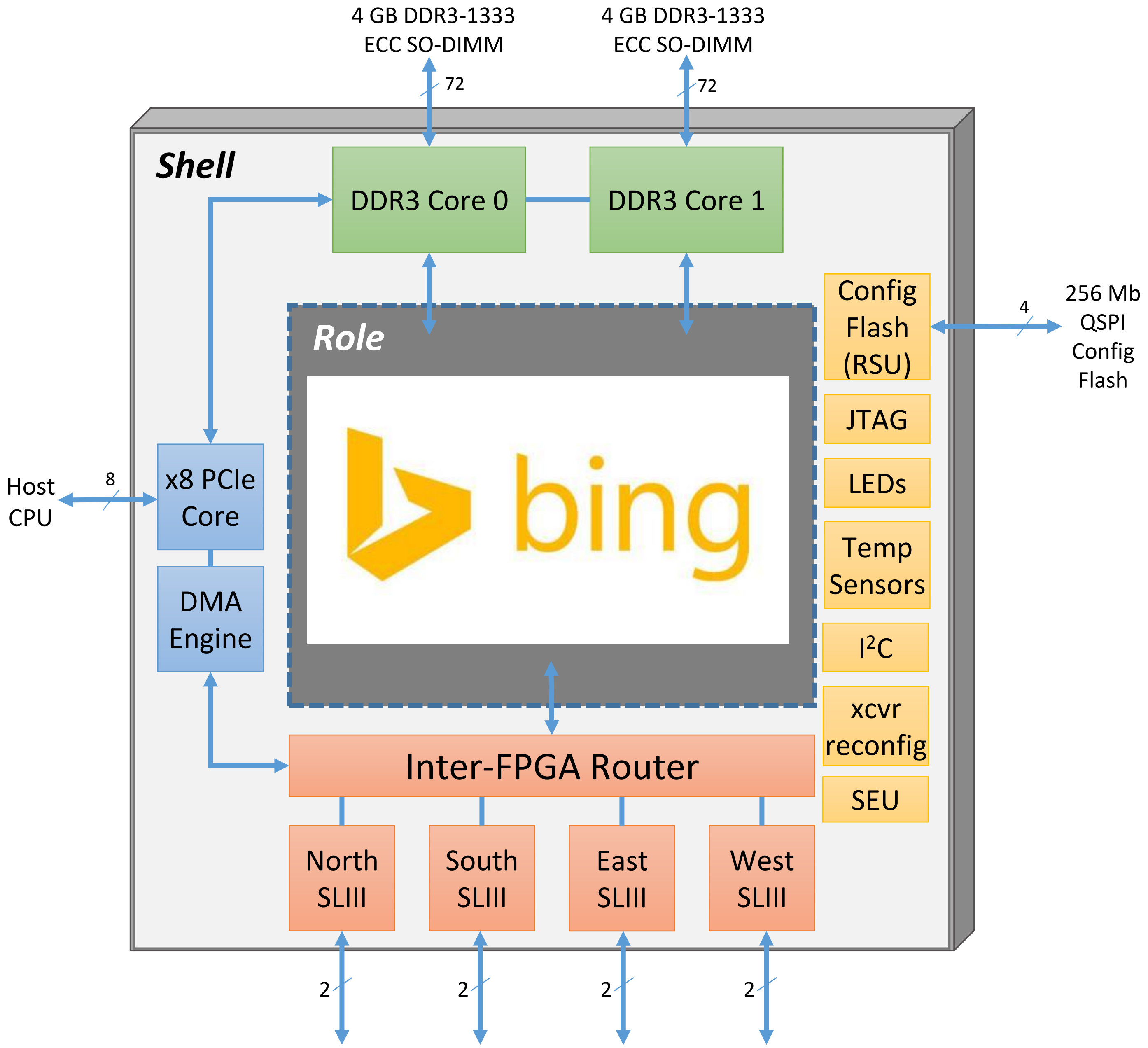
PCIe (8.0 GB/s)

SLIII (2.0 GB/s)

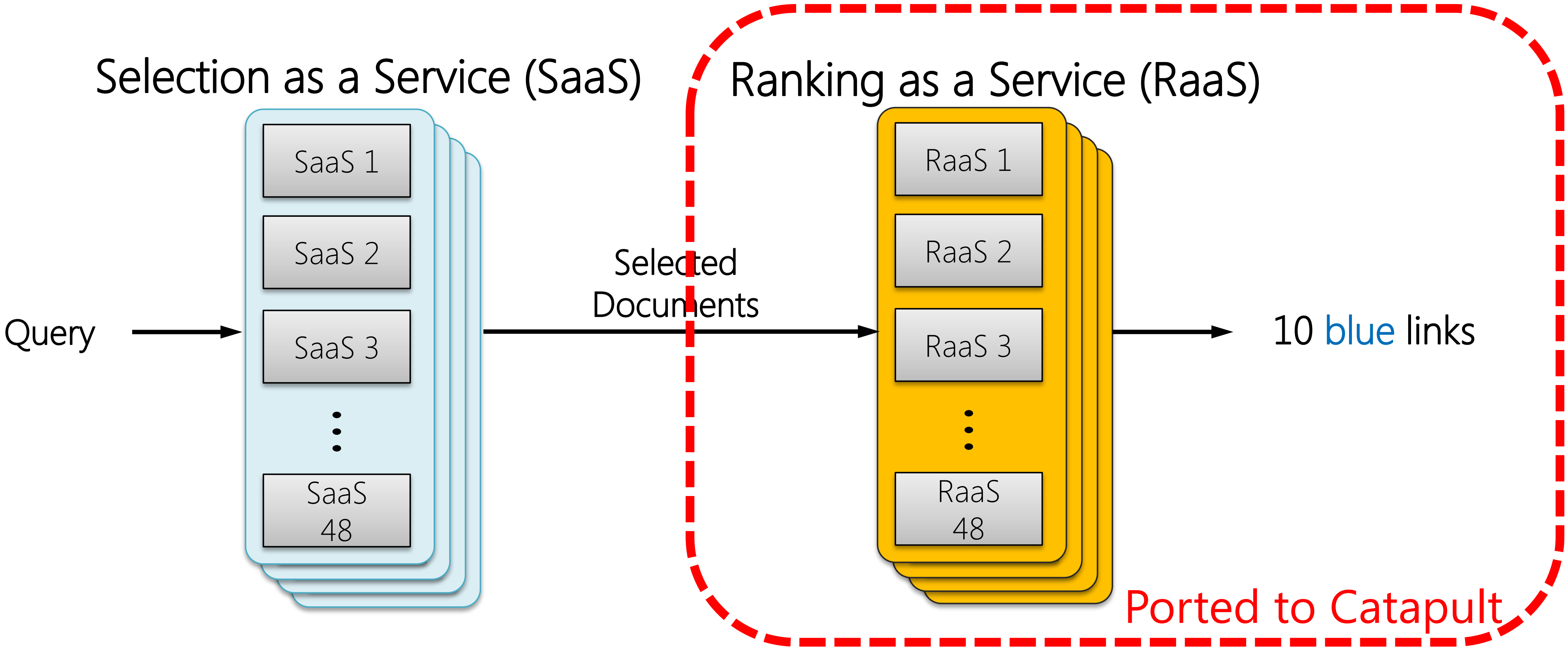
400 ns latency/hop

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



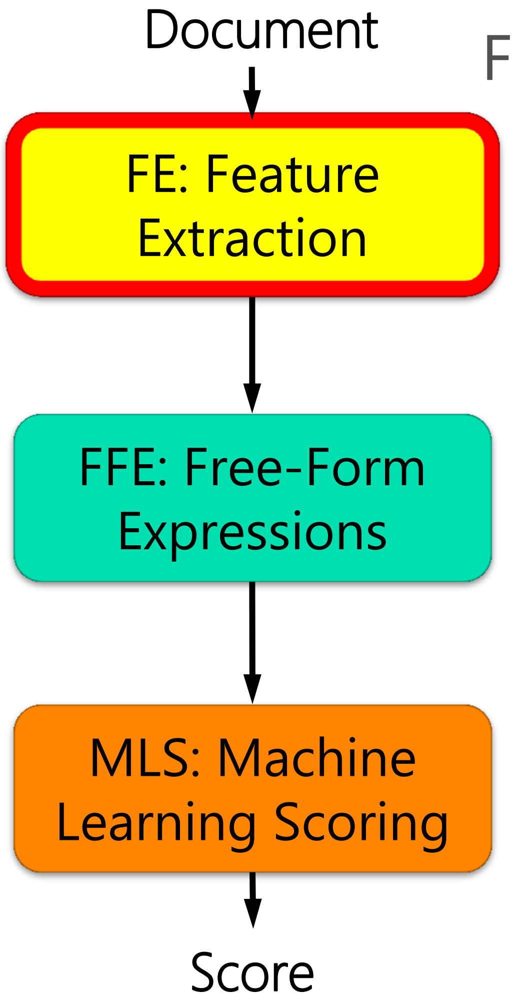
Selection-as-a-Service (SaaS)

- Find all docs that contain query terms,
- Filter and select candidate documents for ranking

Ranking-as-a-Service (RaaS)

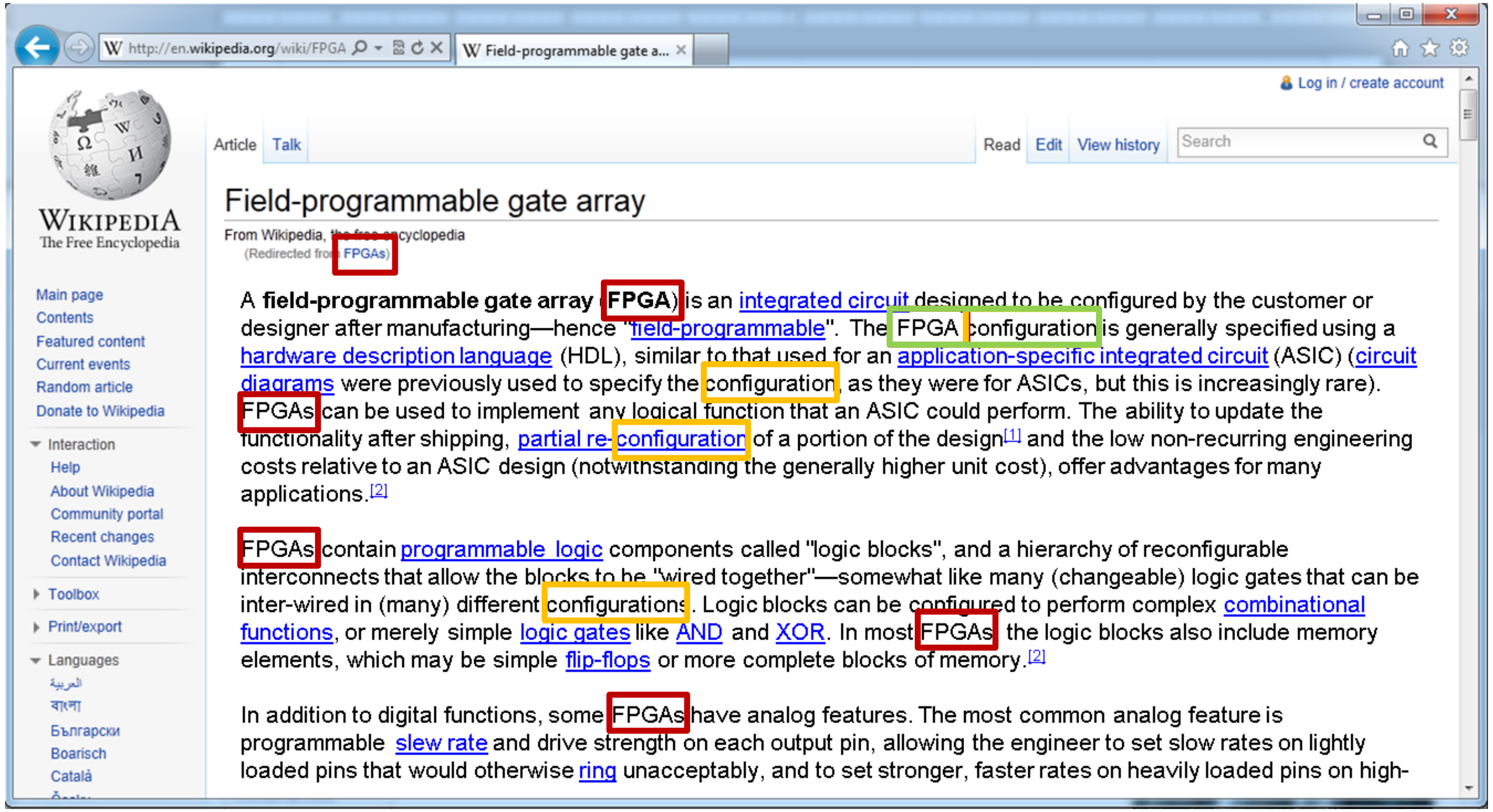
- Compute scores for how relevant each selected document is for the search query
- Sort the scores and return the results

FE: Feature Extraction



Query: “FPGA Configuration”

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



Field-programmable gate array

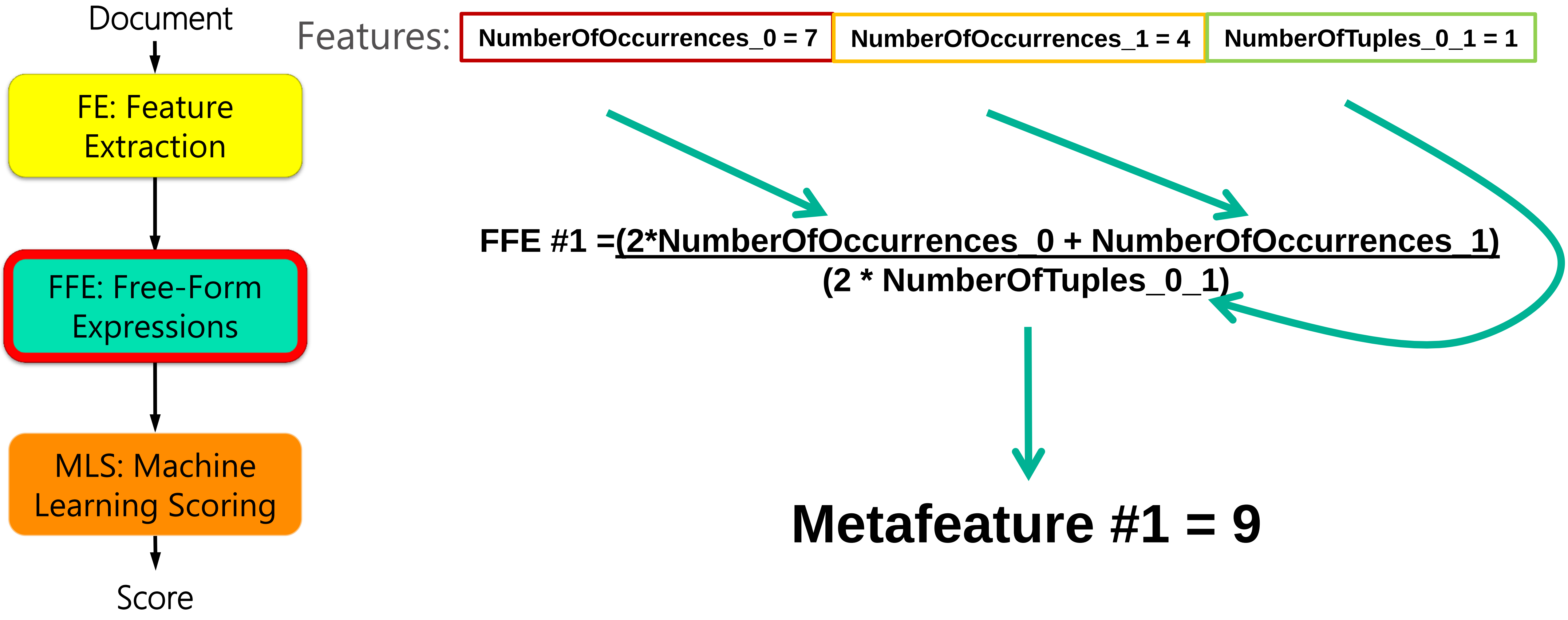
From Wikipedia, the free encyclopedia
(Redirected from FPGAs)

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

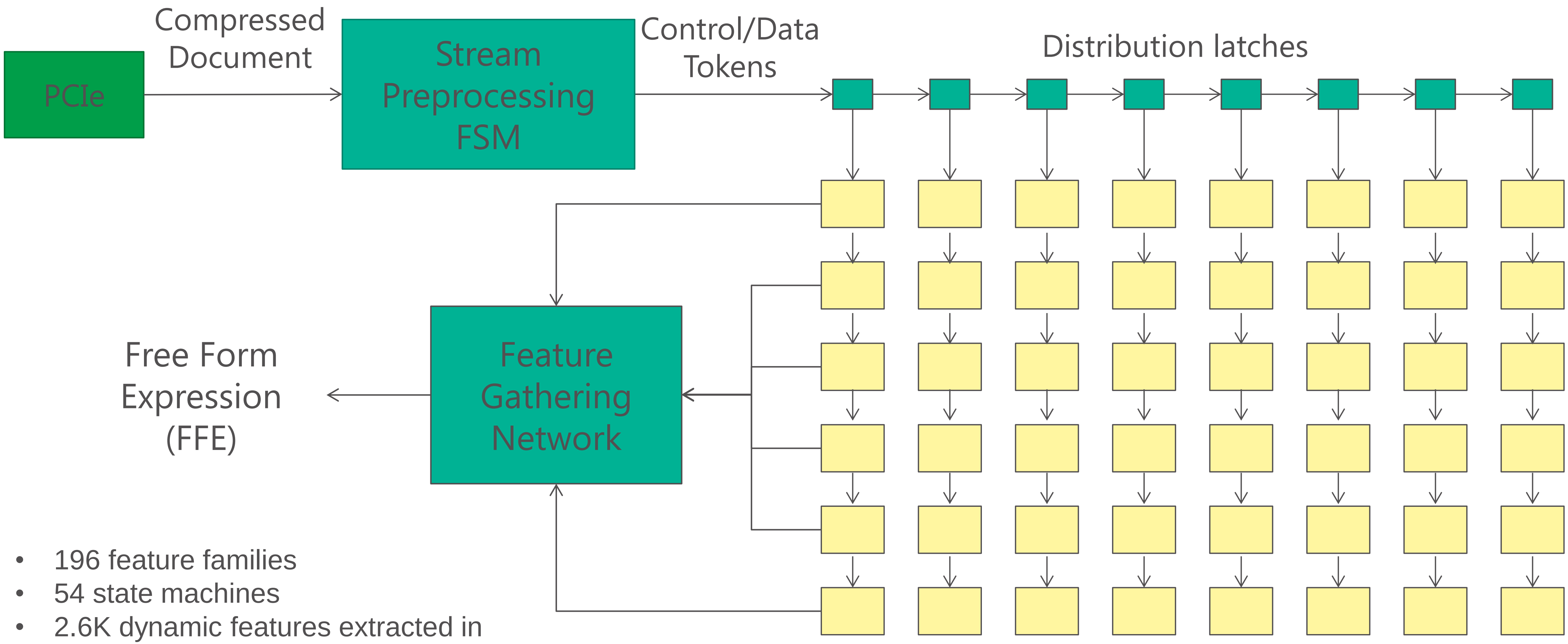


Features: NumberOfOccurrences_0 = 7 | NumberOfOccurrences_1 = 4 | NumberOfTuples_0_1 = 1

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

Metafeature #1 = 9

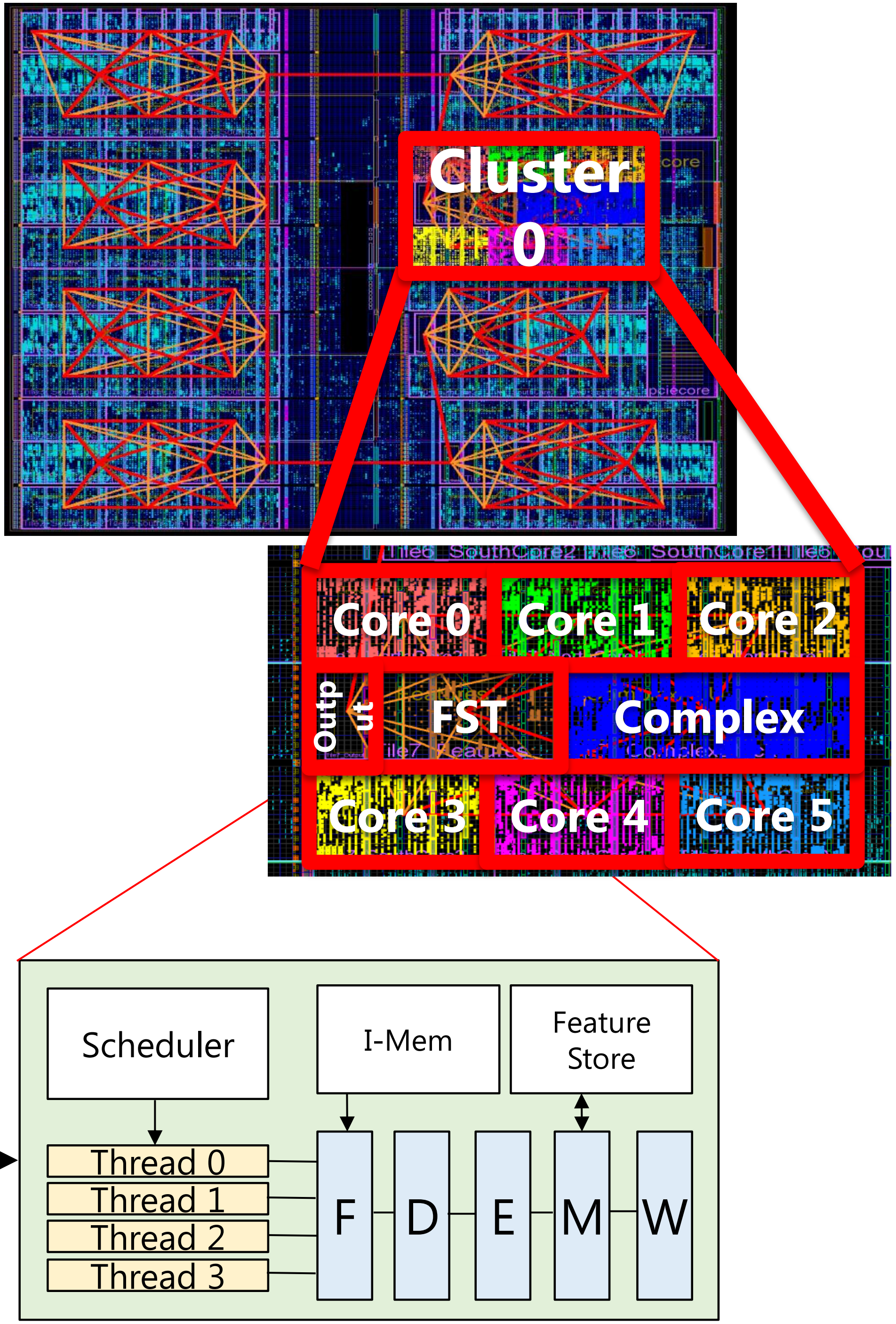
Feature Extraction Accelerator

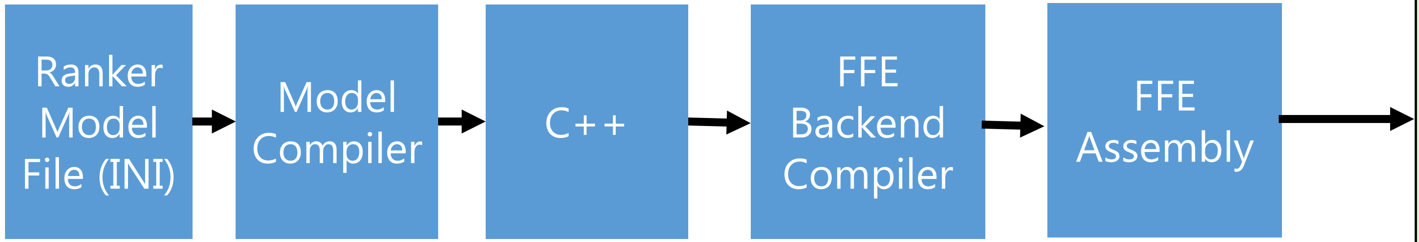


- 196 feature families
- 54 state machines
- 2.6K dynamic features extracted in less than 4us (~600us in SW)

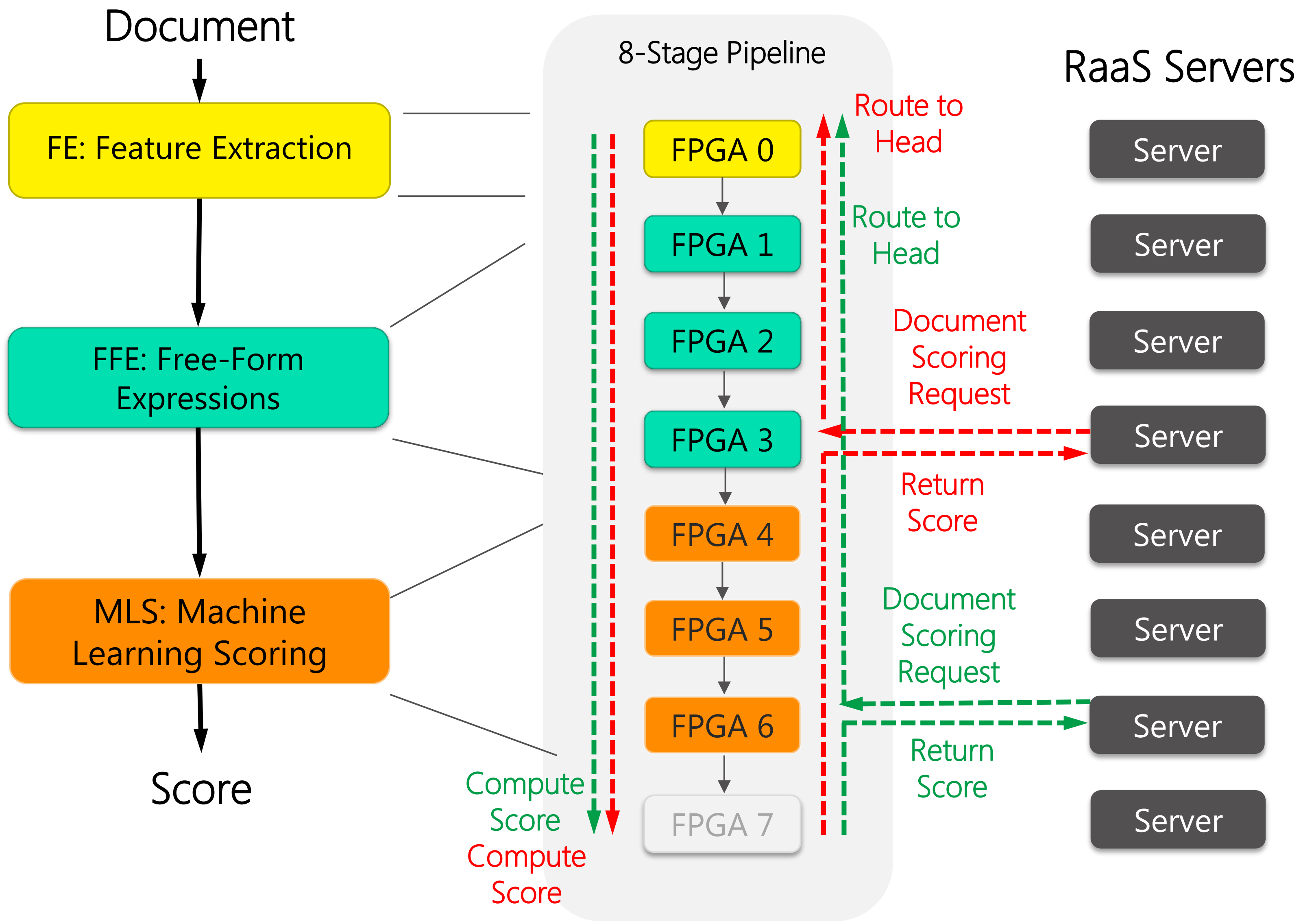
FFE Soft Cores

- 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




Ranker Model File (INI) → Model Compiler → C++ → FFE Backend Compiler → FFE Assembly →

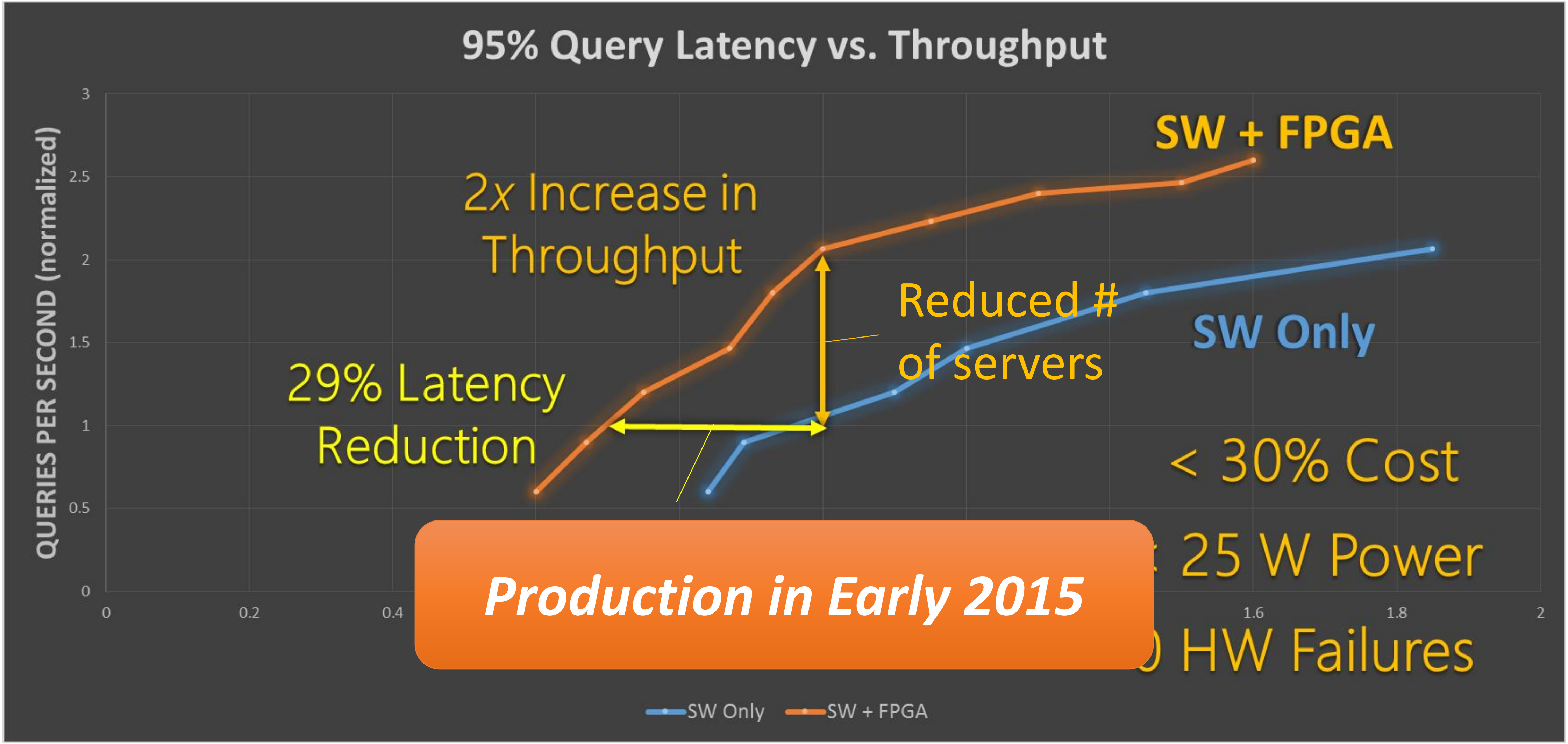
Putting it all together



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++)



2x Increase in Throughput

Reduced # of servers

29% Latency Reduction

< 30% Cost

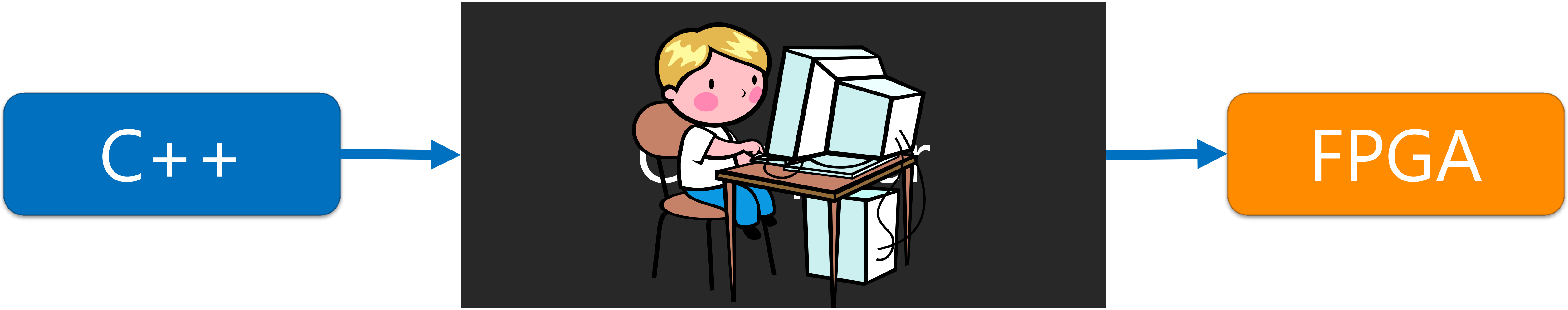
< 25 W Power

0 HW Failures

Production in Early 2015

FPGAs for Application Acceleration

- Hardware is the “easy” part
- Software changes fast
- Services last across hardware generations



C++

FPGA

Software &
Language

Compiler &
Tools

RTL &
Hardware

Ensuring Maintainability & Sustainability

Software & Language

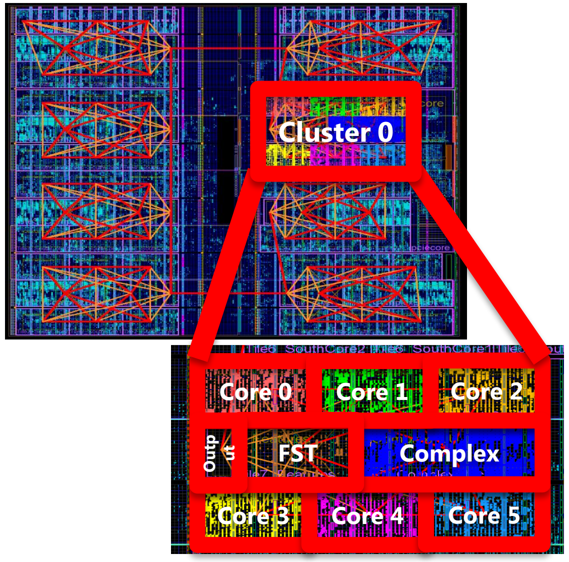
- Structure software for explicit data communication
  - Pass-by-value vs. Pass-by-reference

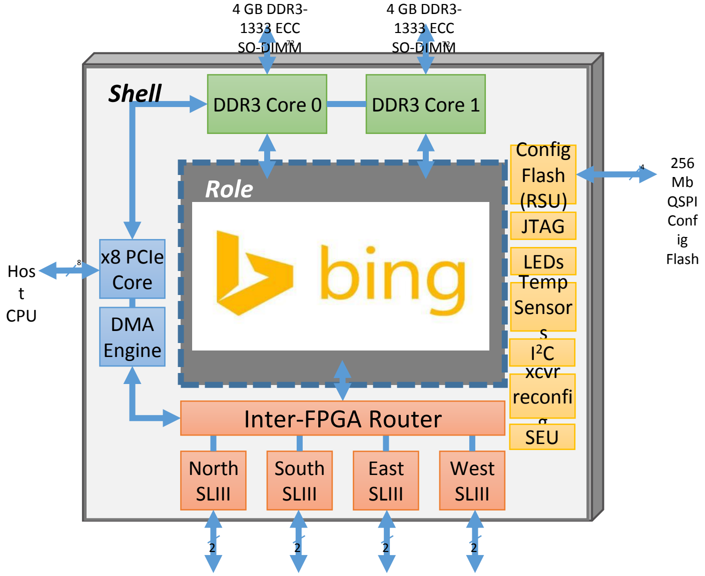
Compiler & Tools

- Create programmable substrates
  - “Programming” FFE is just writing C++ code
  - Wide spectrum between CPU and full custom RTL

RTL & Hardware

- Shell & Role, Common APIs
  - Abstract away board interfaces & FPGA details from the application





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

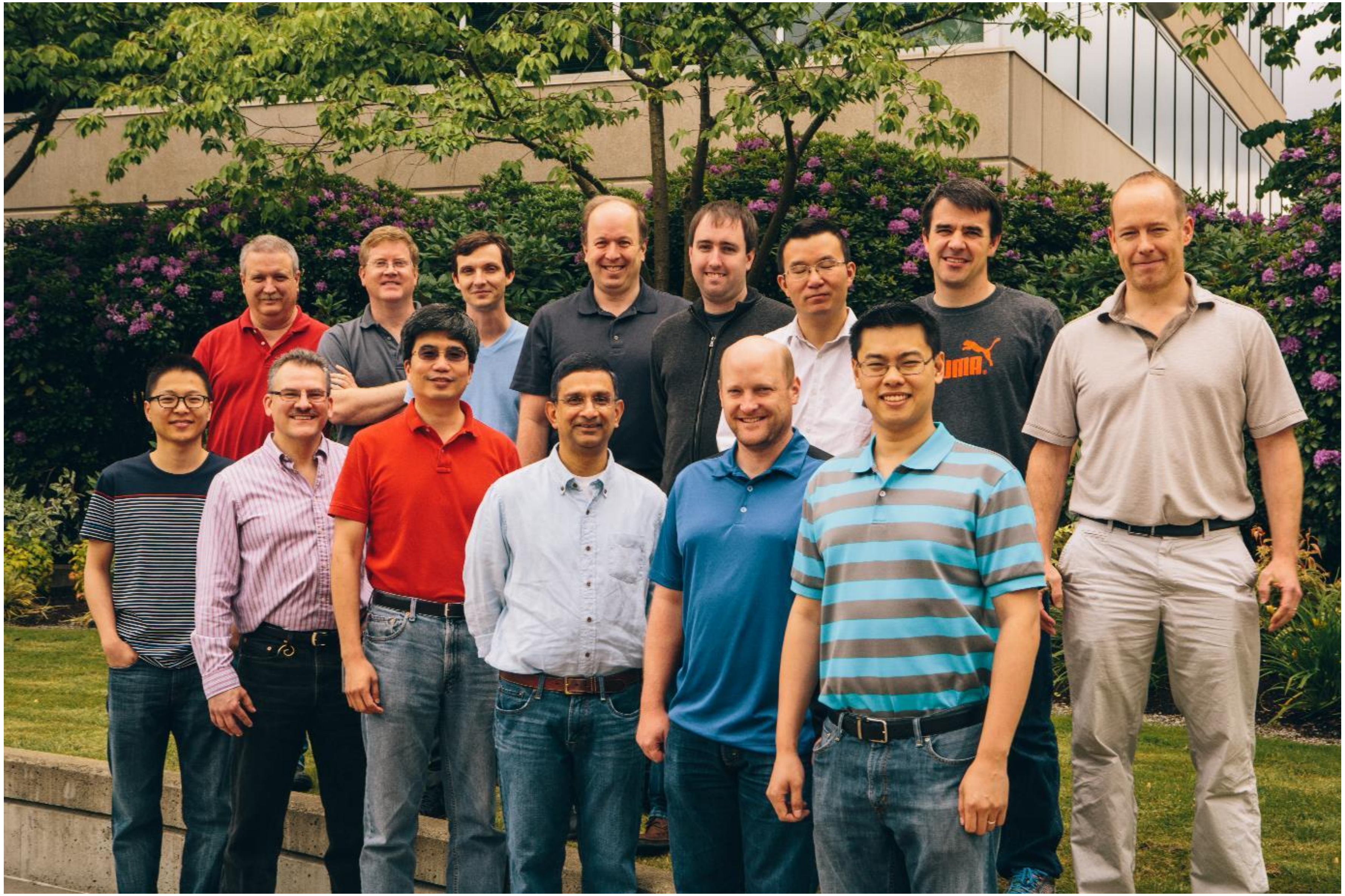
- Hardware specialization is a (the?) way to gain efficiency and performance
- The Catapult reconfigurable fabric offers a flexible, elastic pool of resources to accelerate services
- Results for Bing: **½ the number of ranking servers**, lower latency, reduced variance, proven scalability, proven resilience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

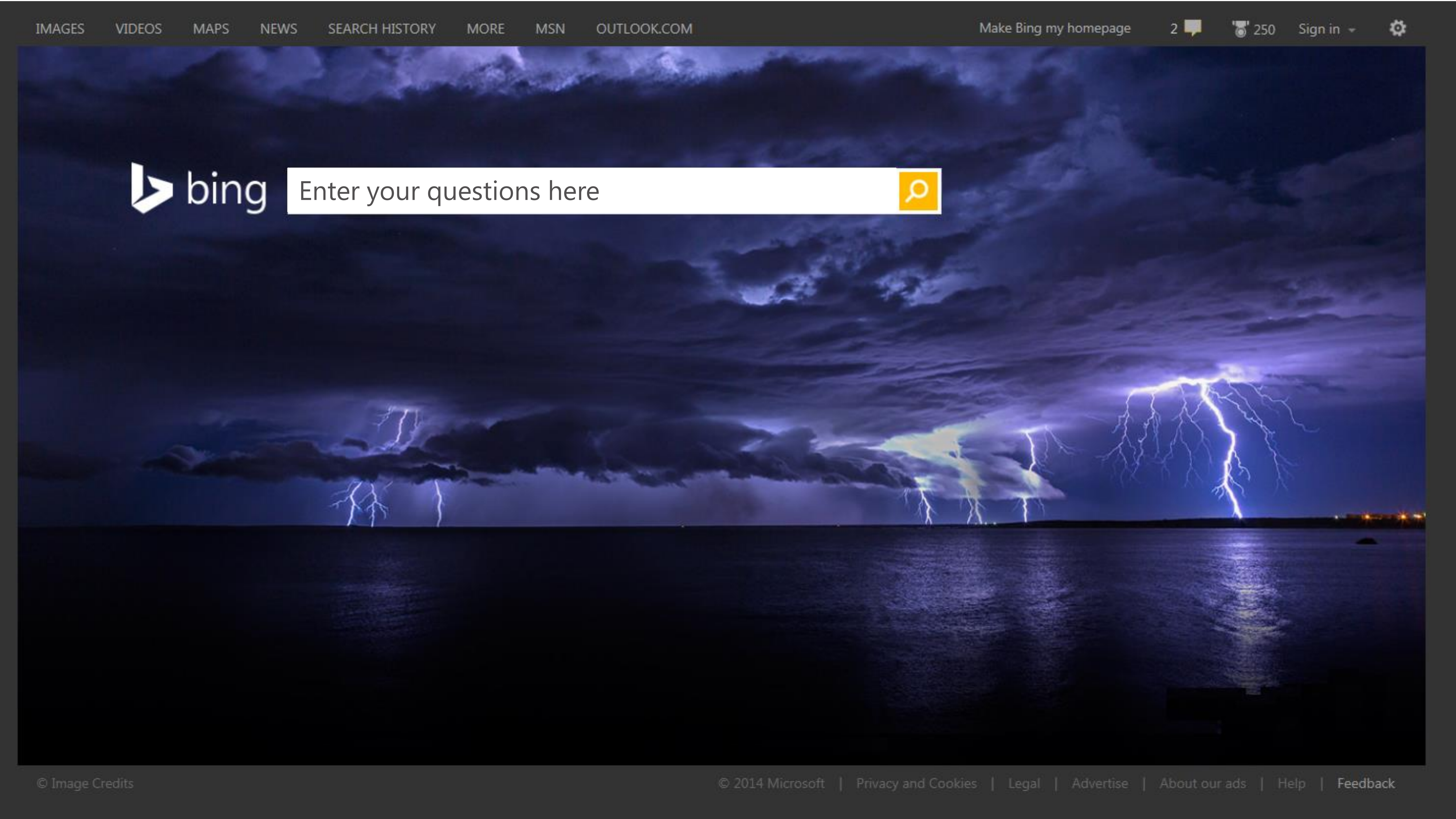
Top Row: Eric Peterson, Scott Hauck, Aaron Smith, Jan Gray, Adrian M. Caulfield, Phillip Yi Xiao, Michael Haselman, Doug Burger

Bottom Row: Joo-Young Kim, Stephen Heil, Derek Chiou, Sitaram Lanka, Andrew Putnam, Eric S. Chung,

Not Pictured: Kypros Constantinides, John Demme, Hadi Esmaeilzadeh, Jeremy Fowers, Gopi Prashanth Gopal, Amir Hormati, James Larus, Simon Pope, Jason Thong

Huge thanks to our partners at




IMAGES
VIDEOS
MAPS
NEWS
SEARCH HISTORY
MORE
MSN
OUTLOOK.COM
Make Bing my homepage
2
250
Sign in
bing
Enter your questions here
© Image Credits
© 2014 Microsoft
Privacy and Cookies
Legal
Advertise
About our ads
Help
Feedback

Microsoft Research

Microsoft

Microsoft

The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.

Microsoft Research

Microsoft

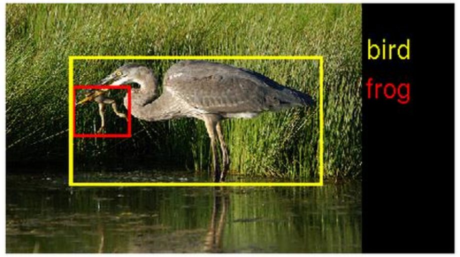
Toward Accelerating Deep Learning at Scale Using Specialized Hardware in the Datacenter

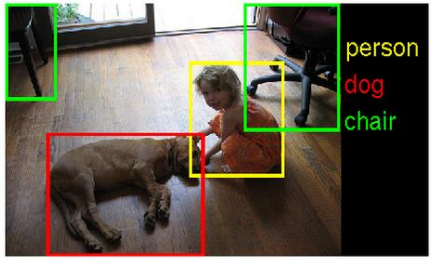
Kalin Ovtcharov, Olatunji Ruwase, Joo-Young Kim,
Jeremy Fowers, Karin Strauss, Eric S. Chung

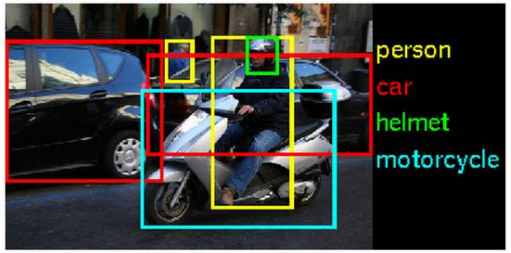
Acknowledgments: Doug Burger and the Catapult Team, Trishul Chilimbi and the Digital Cortex Team, Altera Corporation

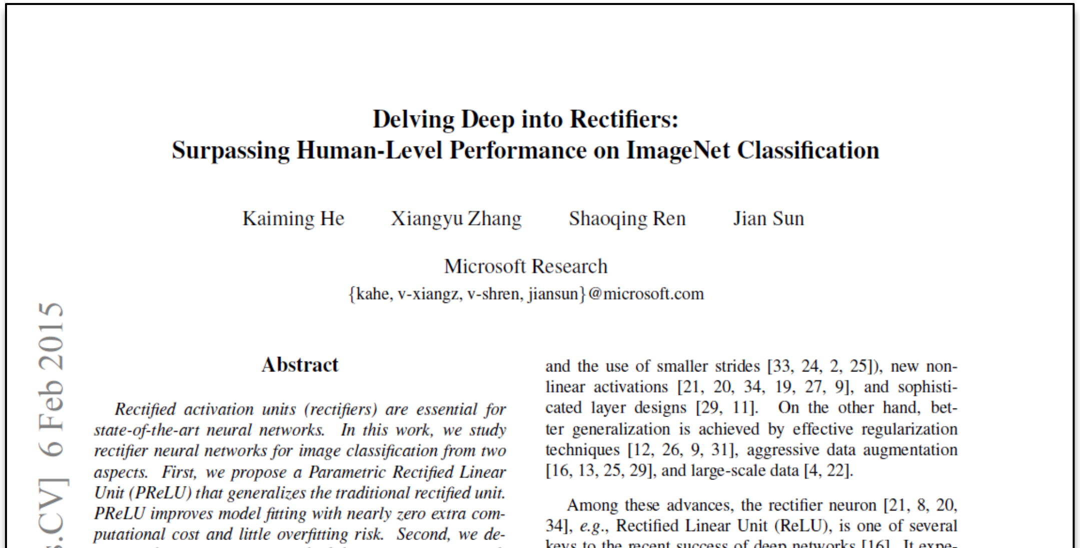
The Rise of Deep Learning

- Significant advances in
  - Computer vision
  - Speech recognition
  - Natural language processing
  - Recommendation systems
  - Intelligent agents
  - Etc.
- Examples
  - Convolutional Neural Networks (CNNs)
  - Deep Belief Networks (DBNs)
  - Recurrent Neural Networks (RNNs)
  - … ?









Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification

Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun

Microsoft Research

{kahe, v-xiangz, v-shren, jiansun}@microsoft.com

Abstract

Rectified activation units (rectifiers) are essential for state-of-the-art neural networks. In this work, we study rectifier neural networks for image classification from two aspects. First, we propose a Parametric Rectified Linear Unit (PReLU) that generalizes the traditional rectified unit. PReLU improves model fitting with nearly zero extra computational cost and little overfitting risk. Second, we de-

and the use of smaller strides [33, 24, 2, 25]), new nonlinear activations [21, 20, 34, 19, 27, 9], and sophisticated layer designs [29, 11]. On the other hand, better generalization is achieved by effective regularization techniques [12, 26, 9, 31], aggressive data augmentation [16, 13, 25, 29], and large-scale data [4, 22].

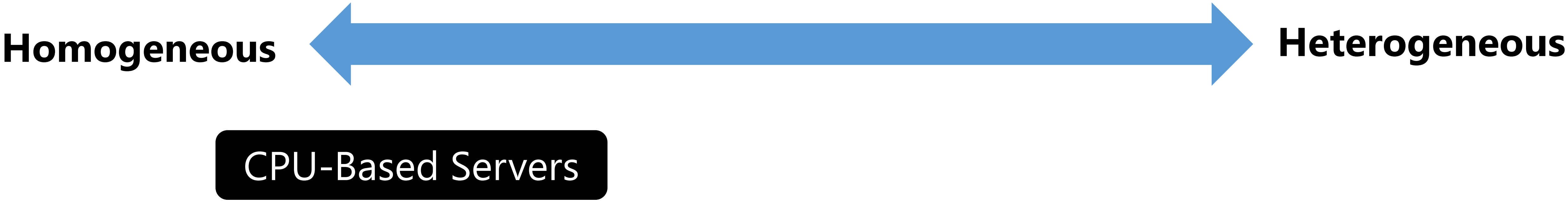
Among these advances, the rectifier neuron [21, 8, 20, 34], *e.g.*, Rectified Linear Unit (ReLU), is one of several keys to the recent success of deep networks [16]. It expe-

This Talk:

Are FPGAs a Promising Target in the Datacenter for Deep Learning[1]?

[1] *Case study: CNN-based Image Classification (inference)*

Cloud Specialization Tradeoffs



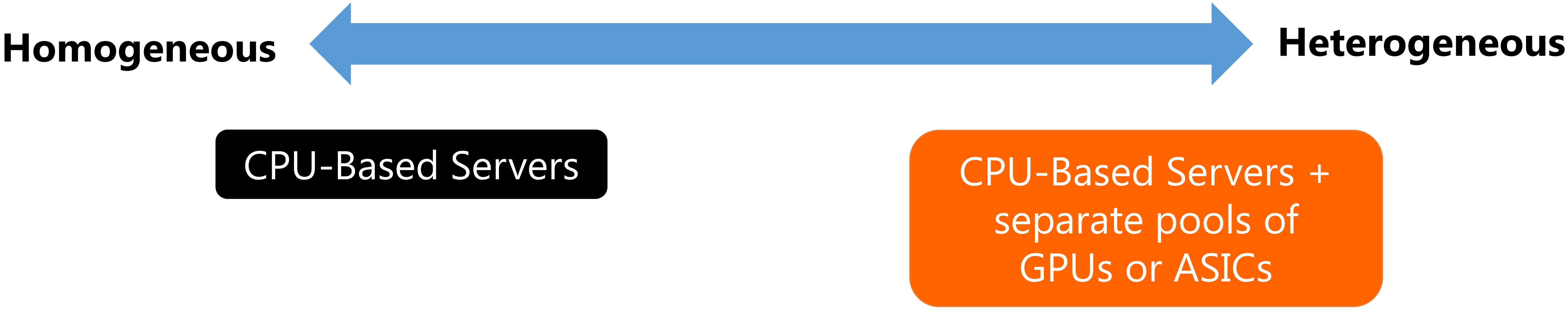
Homogeneous

Heterogeneous

CPU-Based Servers

+ Excellent maintainability in datacenter
+ Maximum flexibility for all workloads
- Performance of CNNs/DNNs vastly slower than specialized HW

Cloud Specialization Tradeoffs



Homogeneous

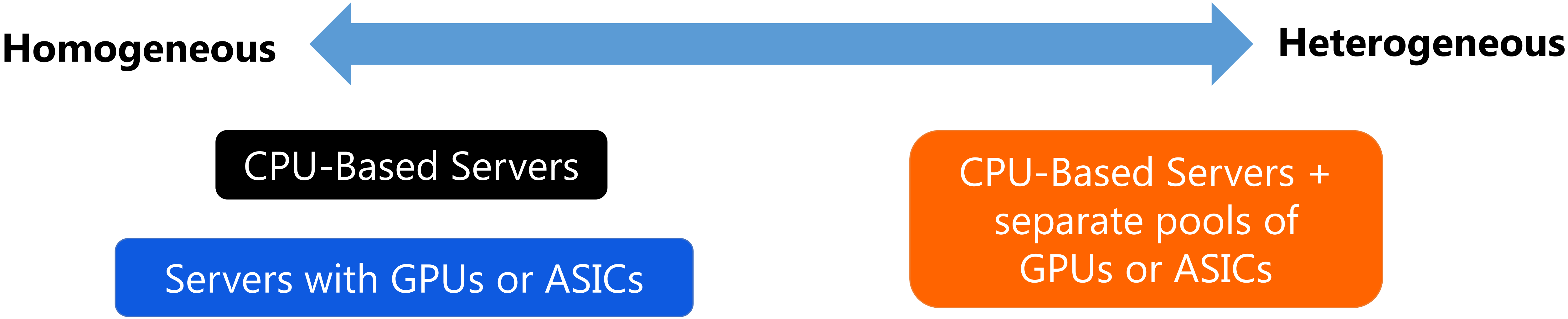
Heterogeneous

CPU-Based Servers

CPU-Based Servers + separate pools of GPUs or ASICs

+ CNNs/DNNs that utilize GPUs or ASICs benefit significantly
- CNNs/DNNs cannot scale beyond limited pools
- Heterogeneity challenging for maintainability

Cloud Specialization Tradeoffs



Homogeneous

Heterogeneous

CPU-Based Servers

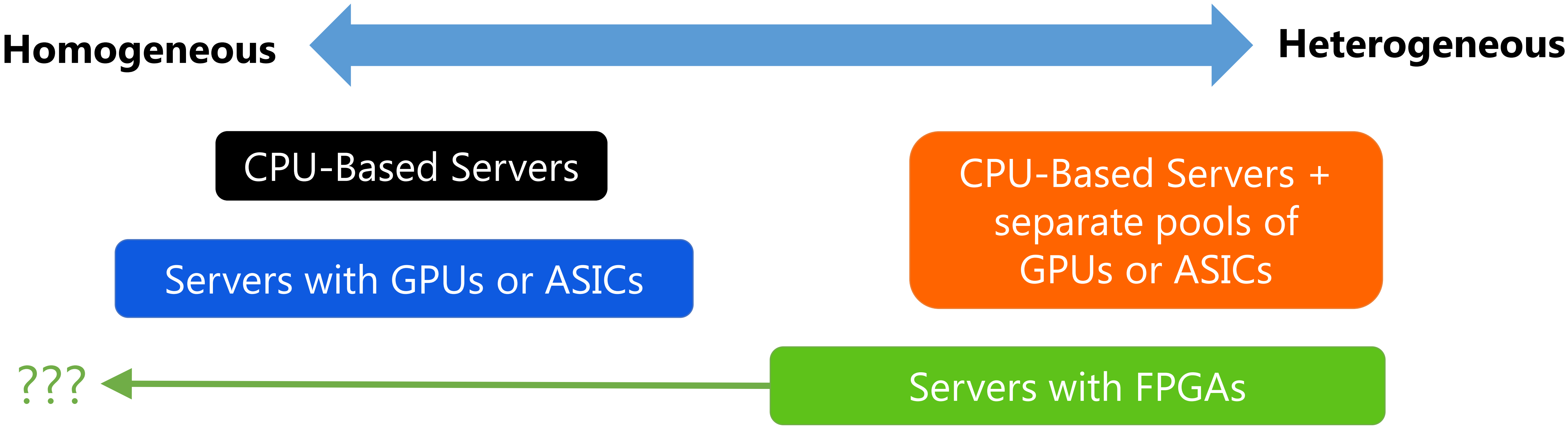
Servers with GPUs or ASICs

CPU-Based Servers + separate pools of GPUs or ASICs

+ Homogeneous

- Increased cost and power per server (particularly GPUs)
- Not economical for all applications in the datacenter (GPUs and ASICs)

Cloud Specialization Tradeoffs



Homogeneous

Heterogeneous

CPU-Based Servers

Servers with GPUs or ASICs

CPU-Based Servers + separate pools of GPUs or ASICs

???

Servers with FPGAs

+ Homogeneous

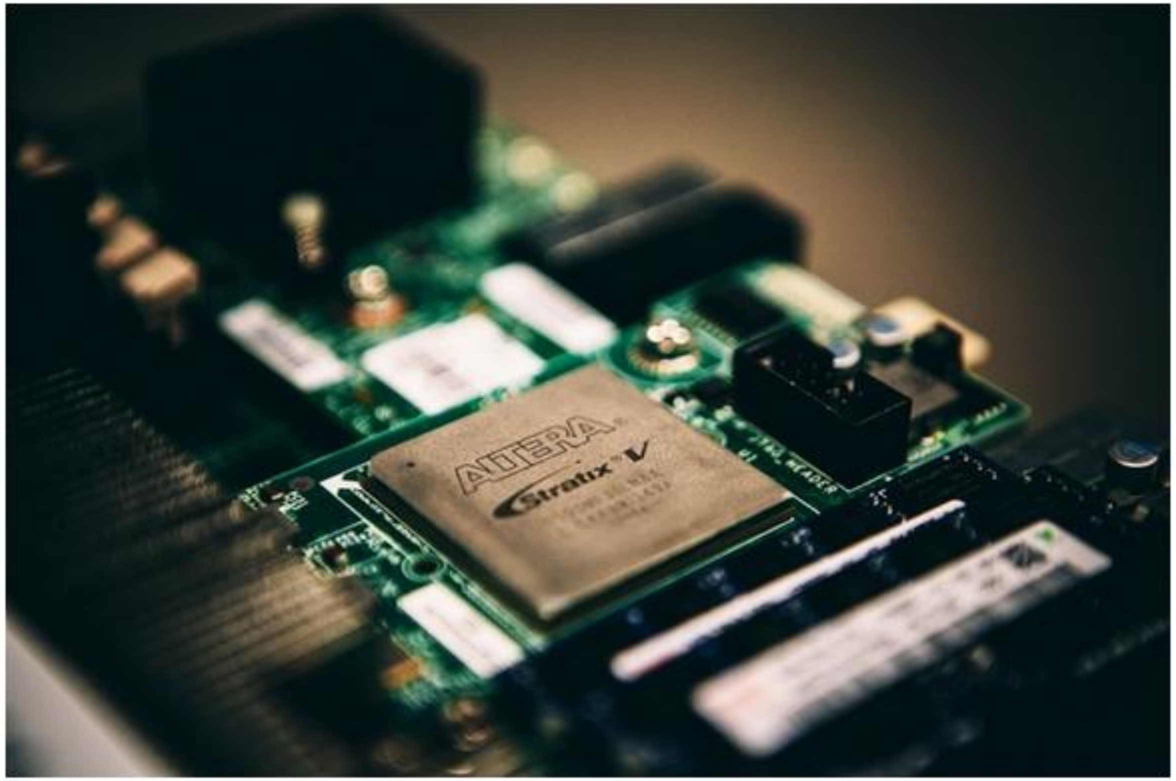
+ Low overhead in power and cost per server

+ Flexibility benefits many workloads?

- Lower peak performance than GPUs or ASICs on some workloads

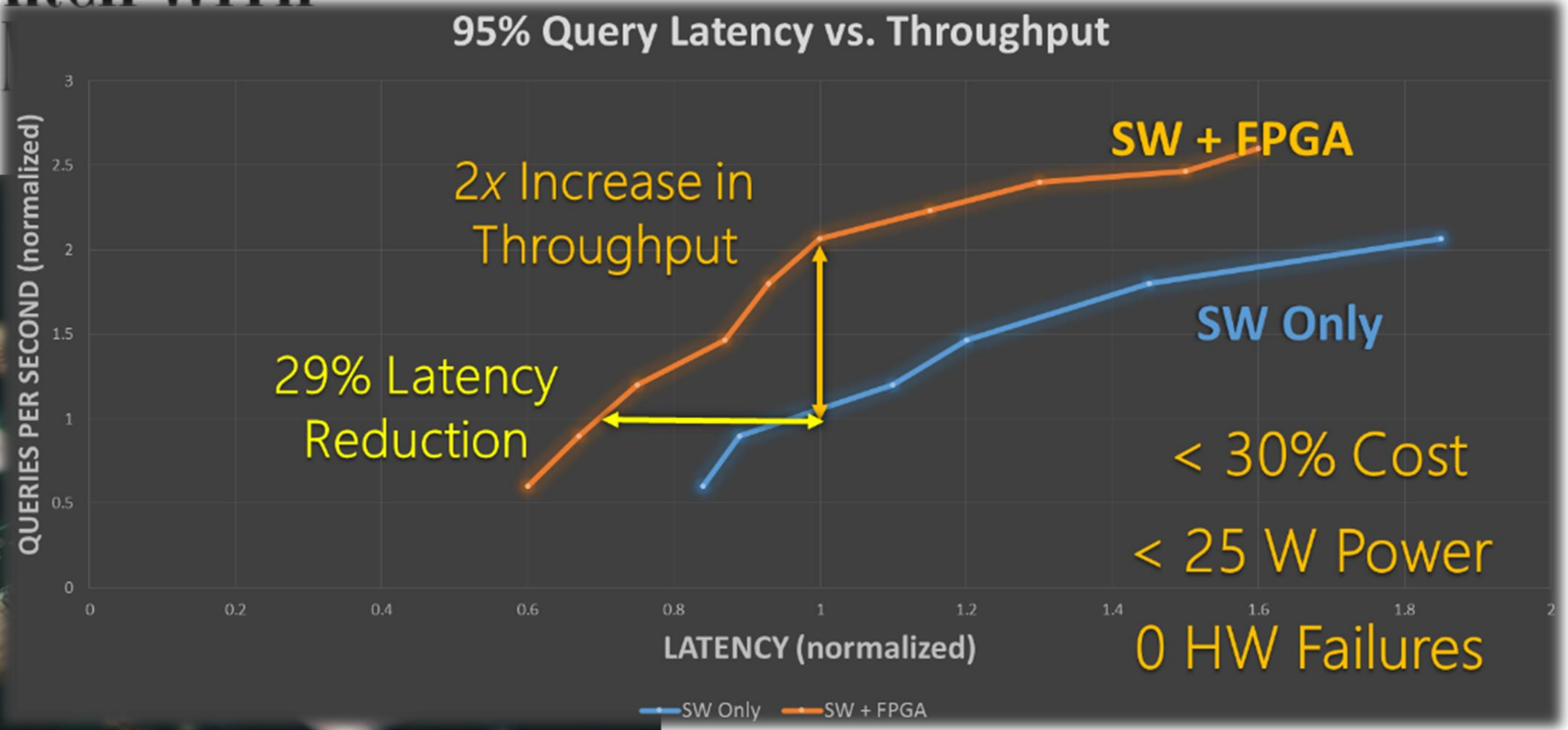
➔ ***Overtake through scale?***

MICROSOFT SUPERCHARGES BING SEARCH WITH PROGRAMMABLE CHIPS



http://www.wired.com/2014/06/microsoft-fpga/

MICROSOFT SUPERCHARGES BING SEARCH WITH PROGRA



95% Query Latency vs. Throughput

2*x* Increase in Throughput

29% Latency Reduction

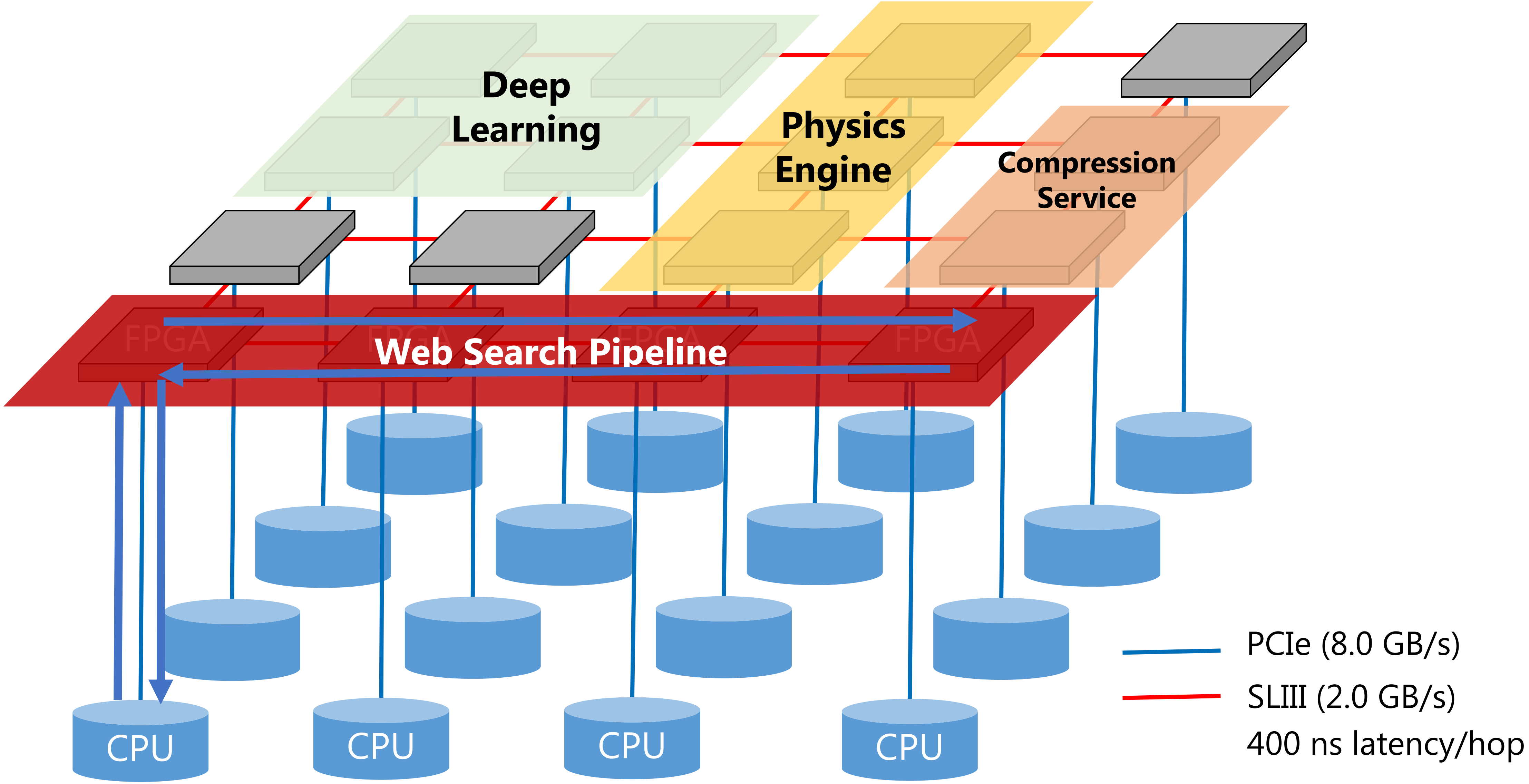
< 30% Cost

< 25 W Power

0 HW Failures

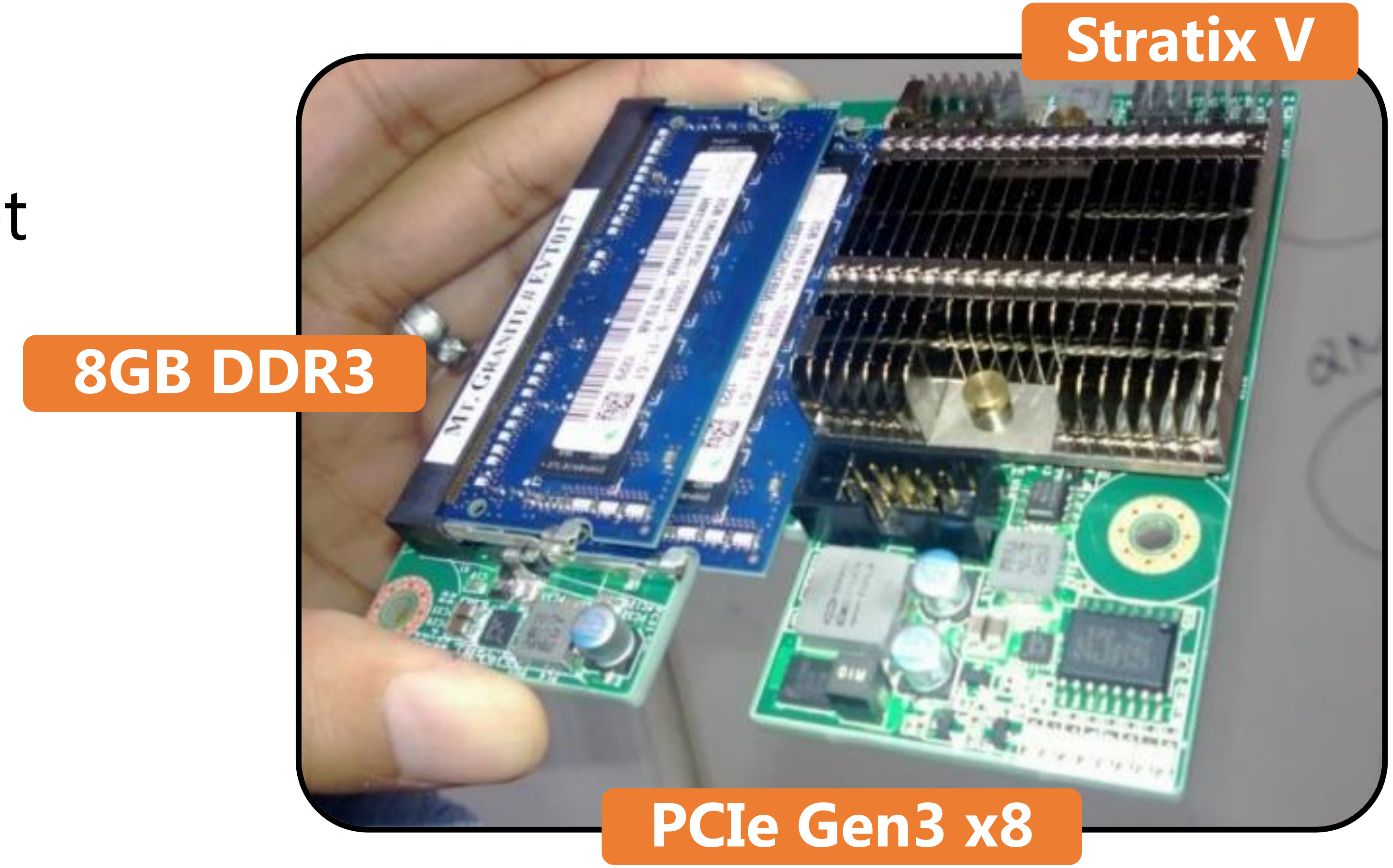
http://www.wired.com/2014/06/microsoft-fpga/

Catapult: An Elastic Reconfigurable Fabric for Datacenters

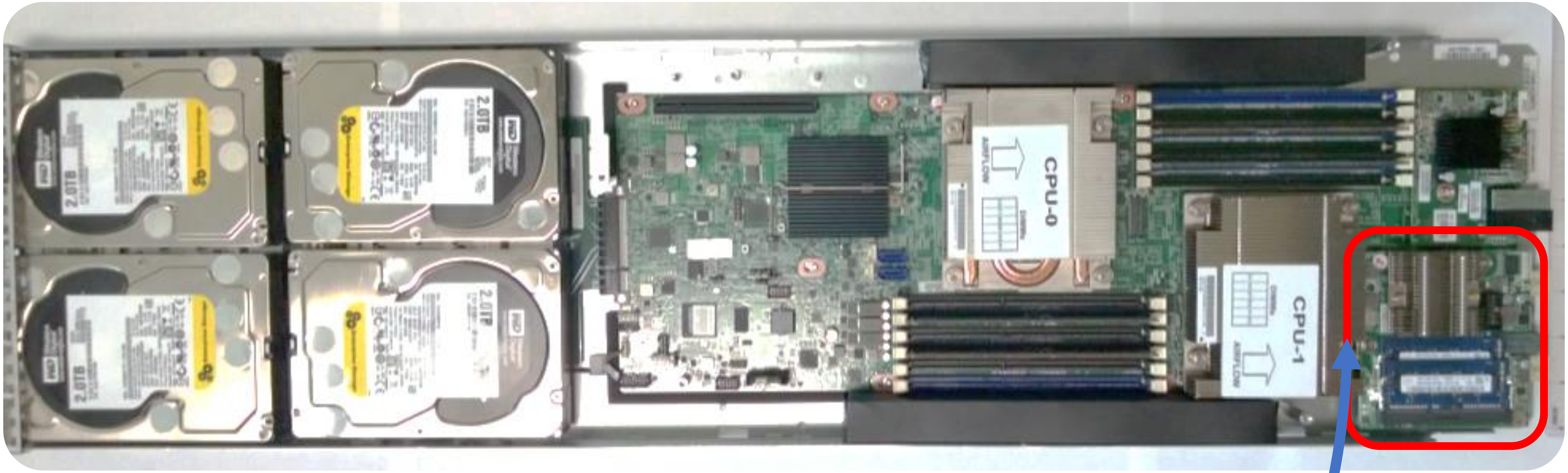


Catapult FPGA Accelerator Card

- Altera Stratix V D5
- 172,600 ALMs, 2,014 M20Ks, 1,590 DSPs
- PCIe Gen 3 x8
- 8GB DDR3-1333
- Powered by PCIe slot
- Torus Network



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

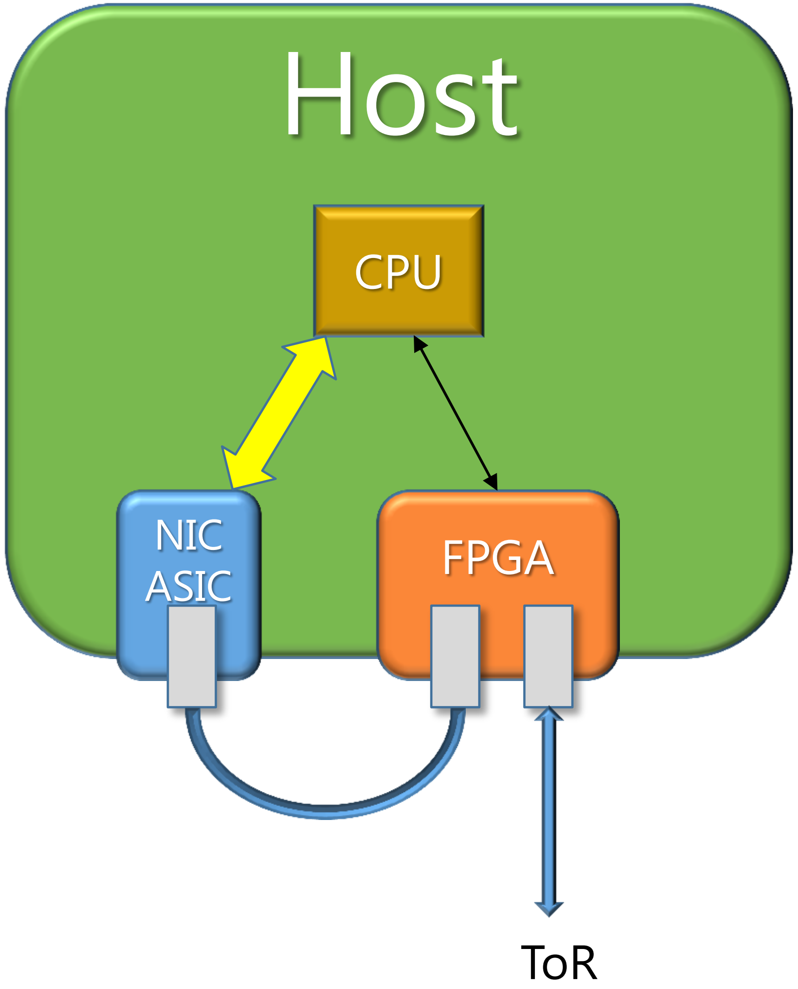
Air flow

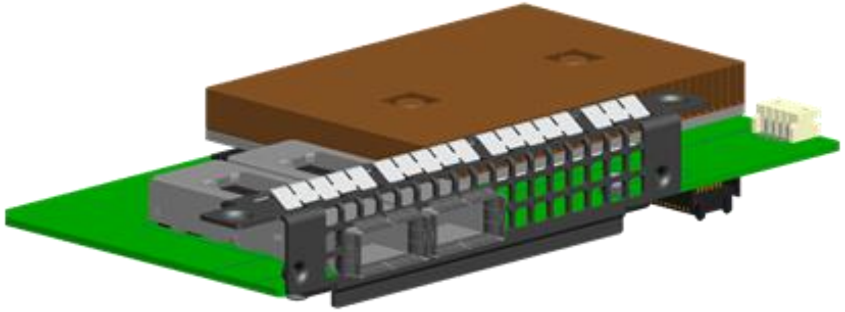
200 LFM

68 ^{0}C Inlet

Azure SmartNIC

- Use Catapult FPGAs for reconfigurable functions
  - Already used in Bing
  - Roll out Hardware as we do software
- Programmed using Generic Flow Tables (GFT)
  - Language for programming SDN to hardware
  - Uses connections and structured actions as primitives
- SmartNIC also does Crypto, QoS, storage acceleration, and more…



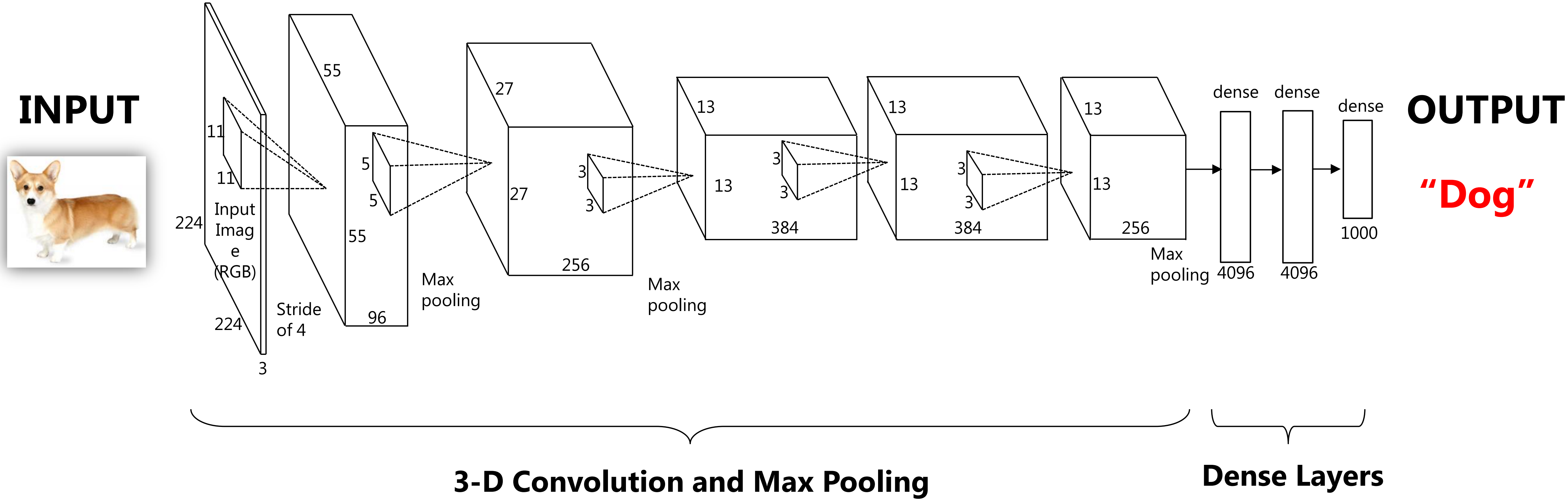


Harnessing Catapult for Deep CNNs

- Leverage abundant FPGA resources in the datacenter for scaling up evaluation and training[1] of deep CNNs
- Achieve order-of-magnitude performance gain relative to CPUs with low cost (<30%) and power (<10%) overheads
- Expose to practitioners as composable SW libraries

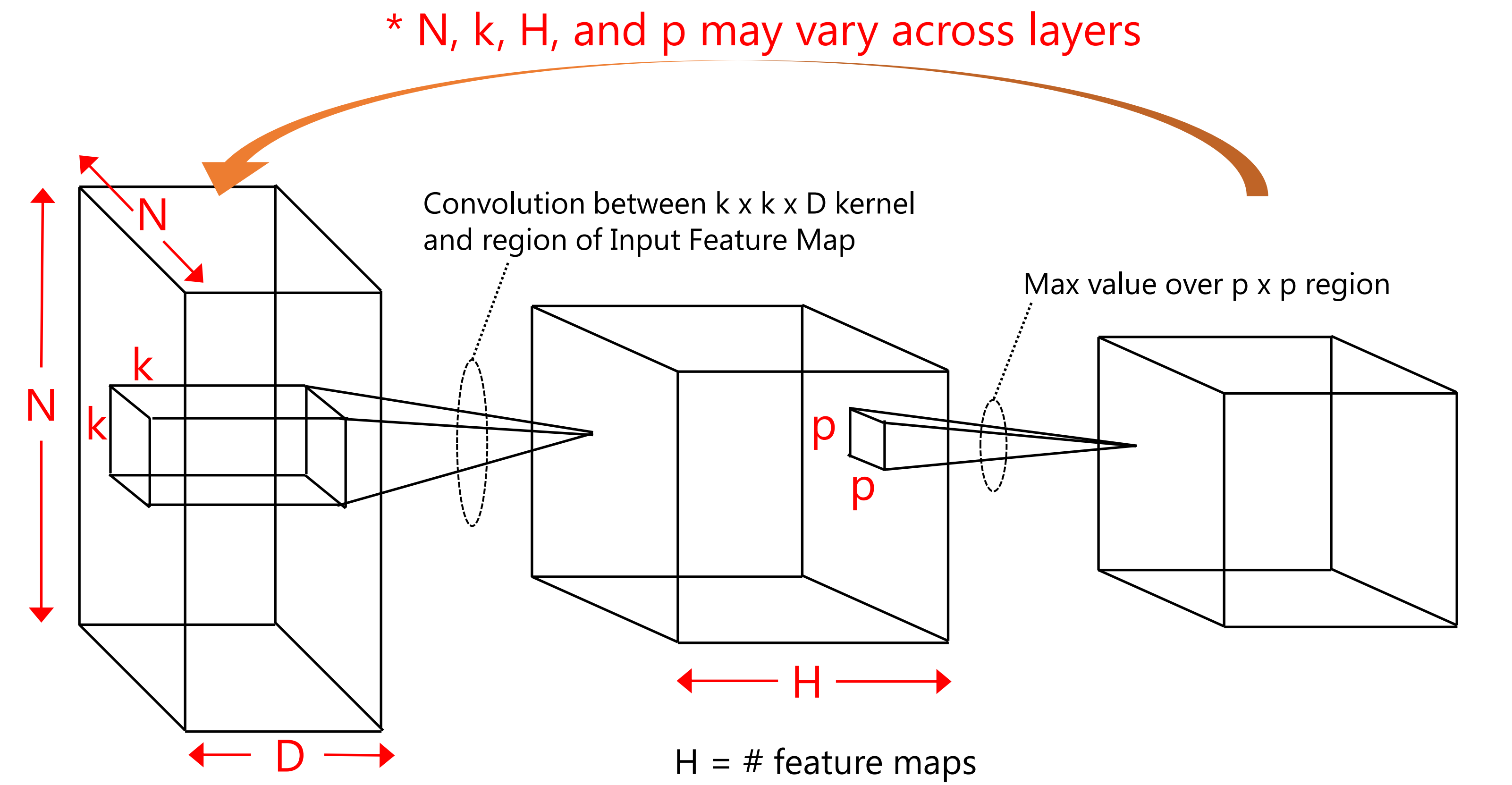
[1] *Under development*

Deep Convolutional Neural Networks



* Krizhevsky et al, NIPS'12

3-D Convolution and Max Pooling


* N, k, H, and p may vary across layers
N
N
k
k
Convolution between k x k x D kernel and region of Input Feature Map
Max value over p x p region
p
p
D
H

N = input height and width
k = kernel height and width
D = input depth

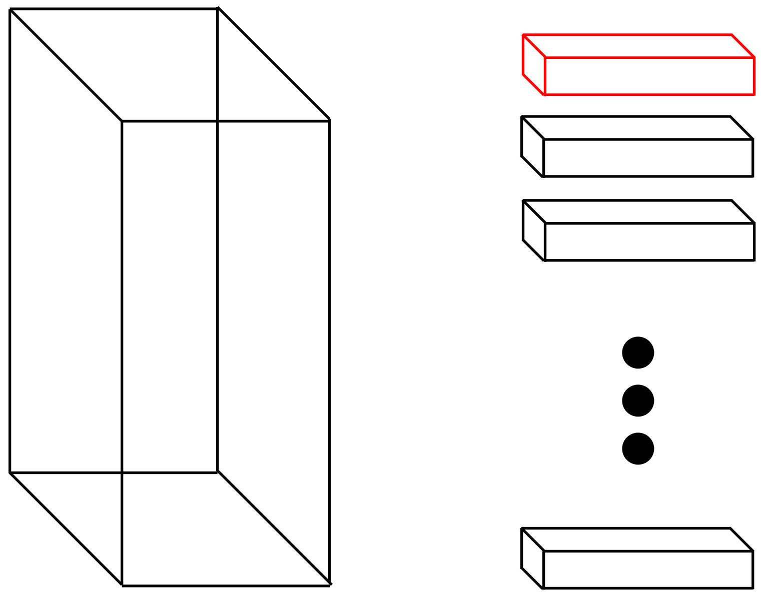
H = # feature maps
S = kernel stride

Input Feature Map

Convolution Output

Max Pooled Output (Optional)

3-D Convolution

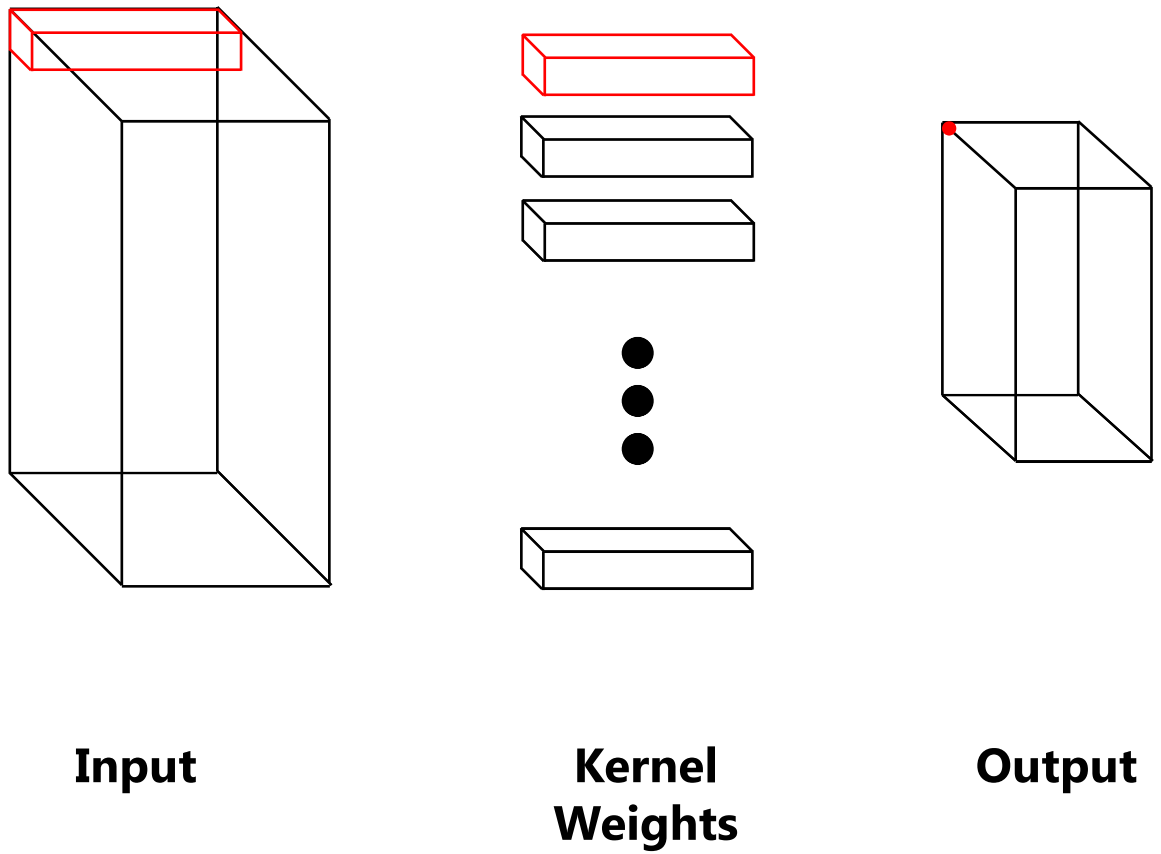


Input

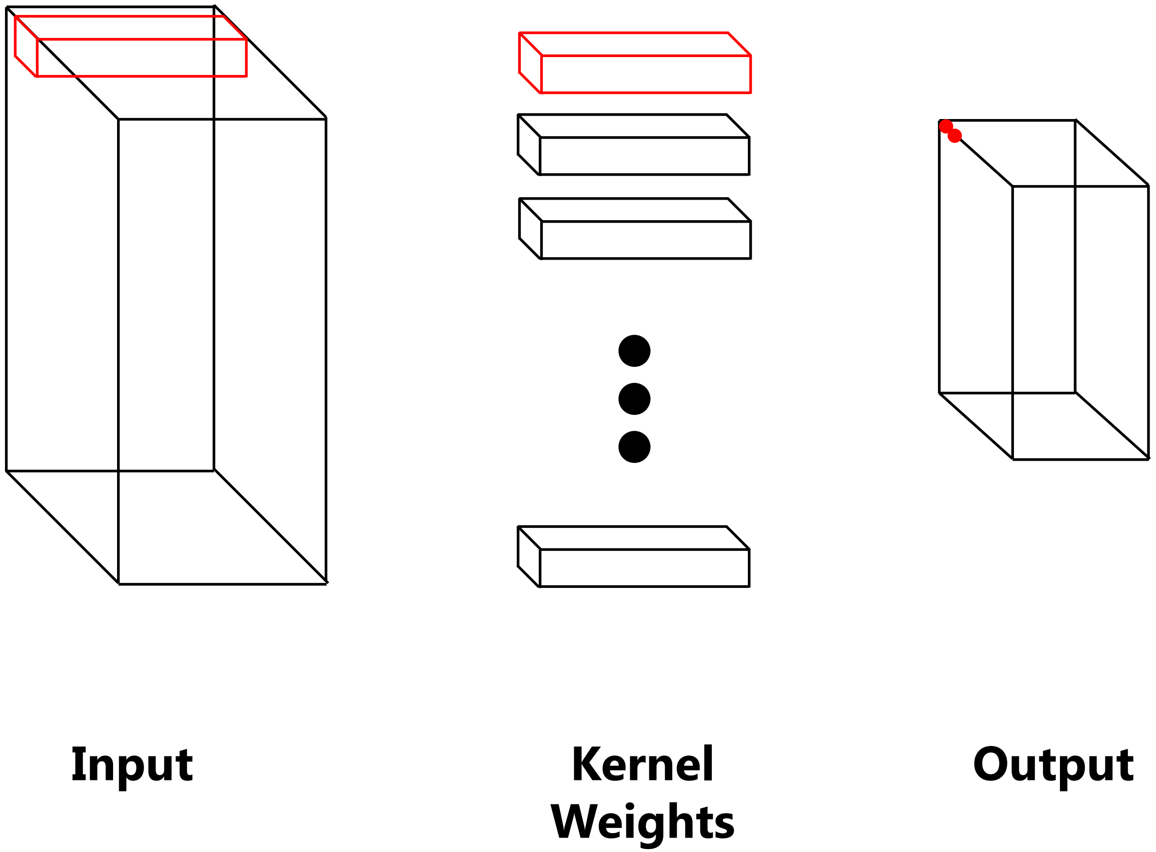
Kernel Weights

Output

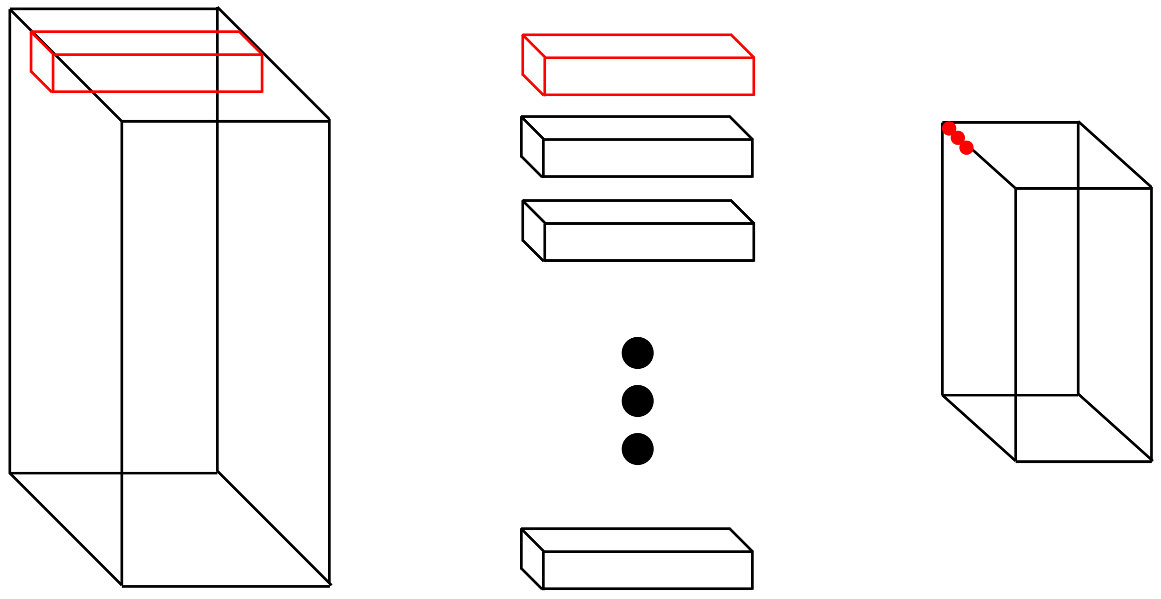
3-D Convolution



3-D Convolution



3-D Convolution

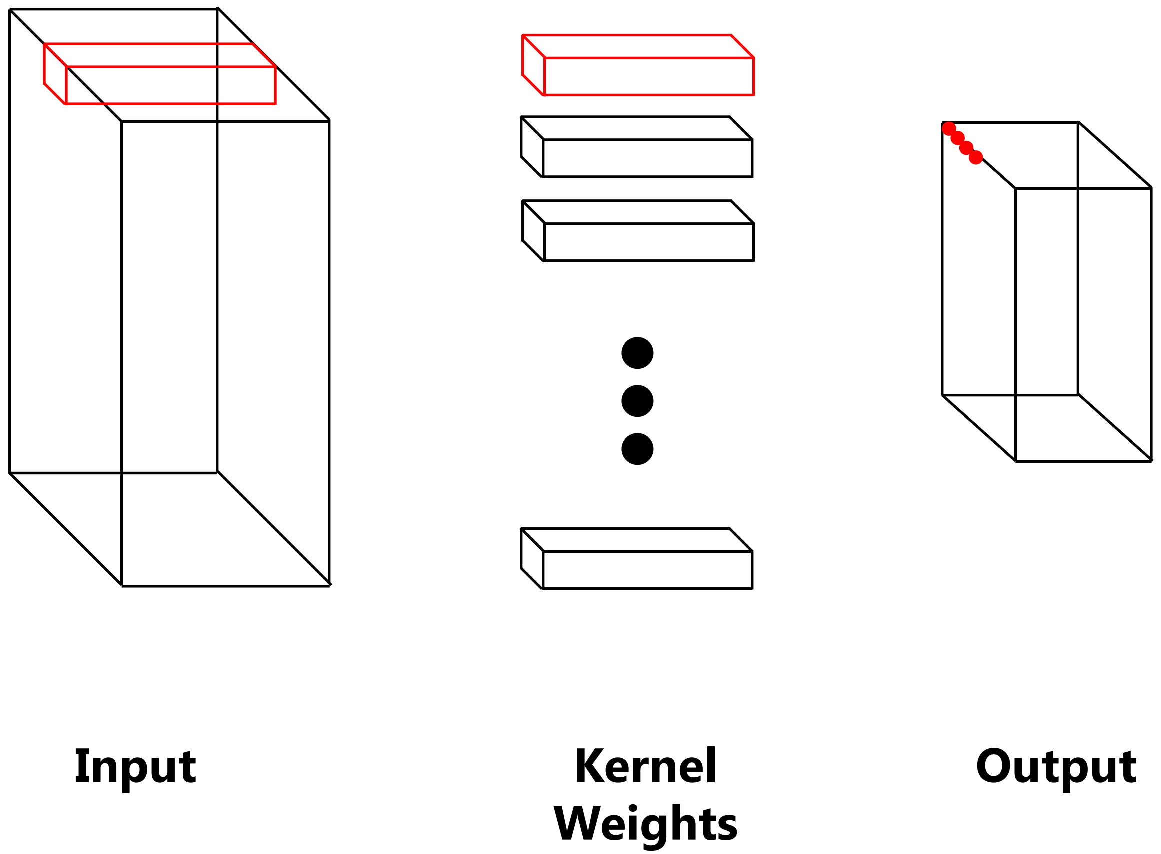


Input

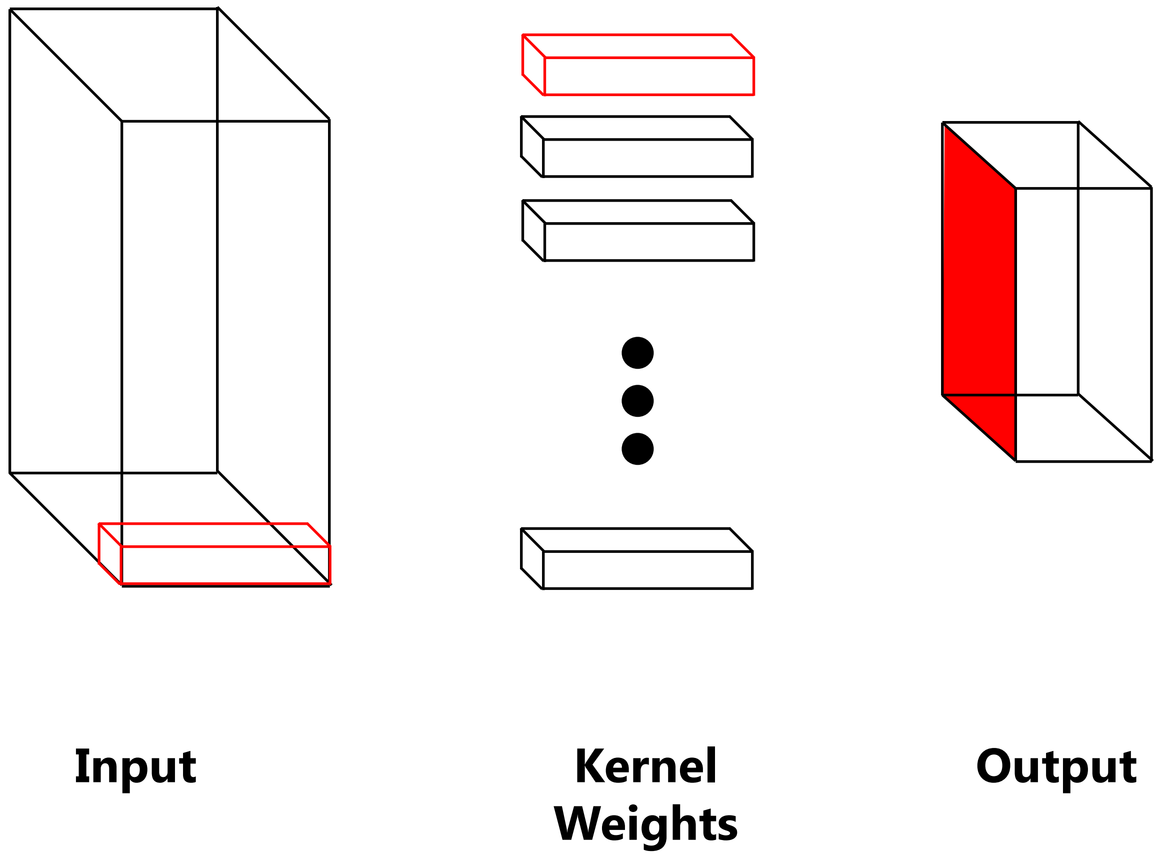
Kernel Weights

Output

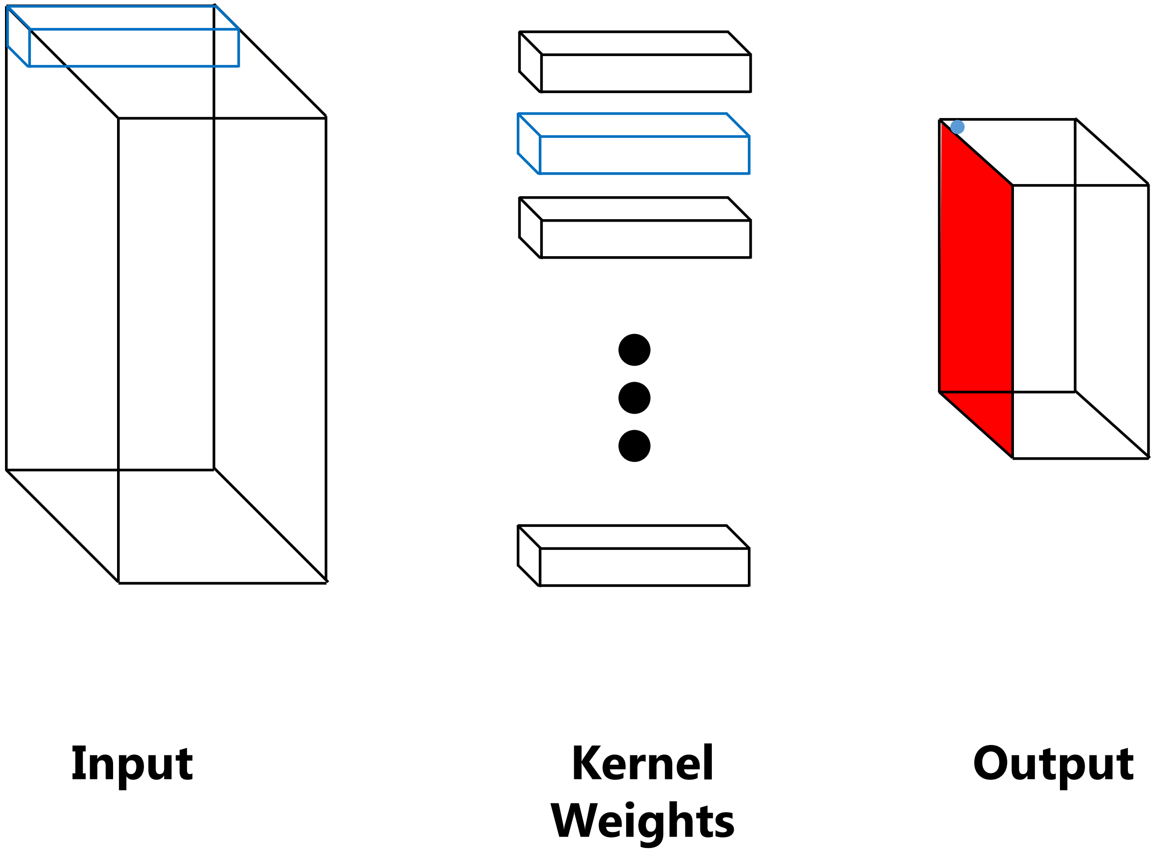
3-D Convolution



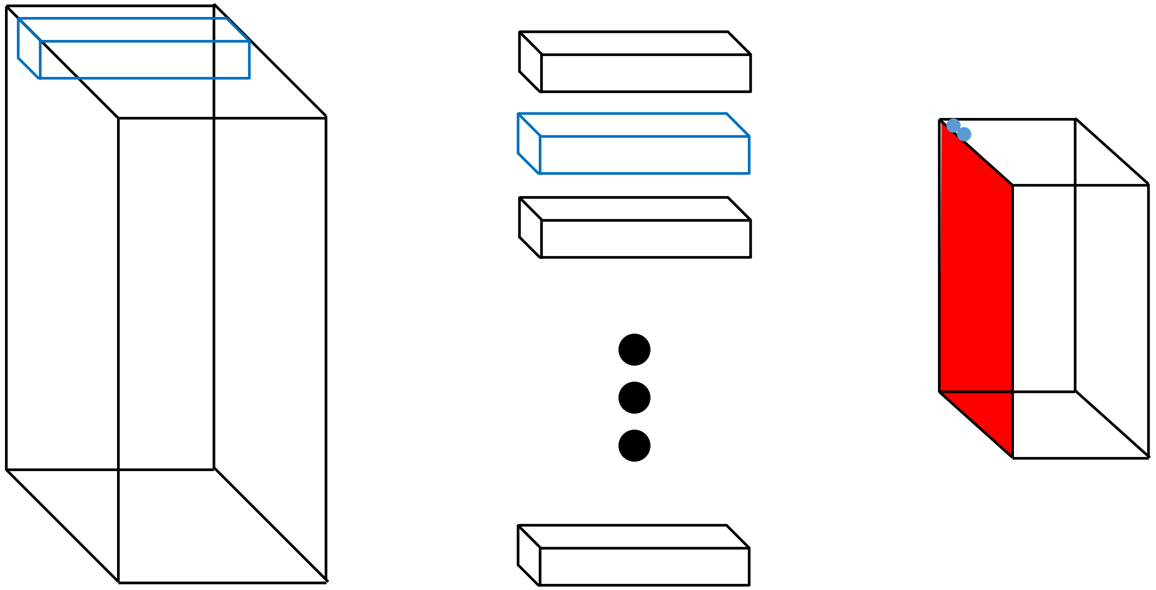
3-D Convolution



3-D Convolution



3-D Convolution

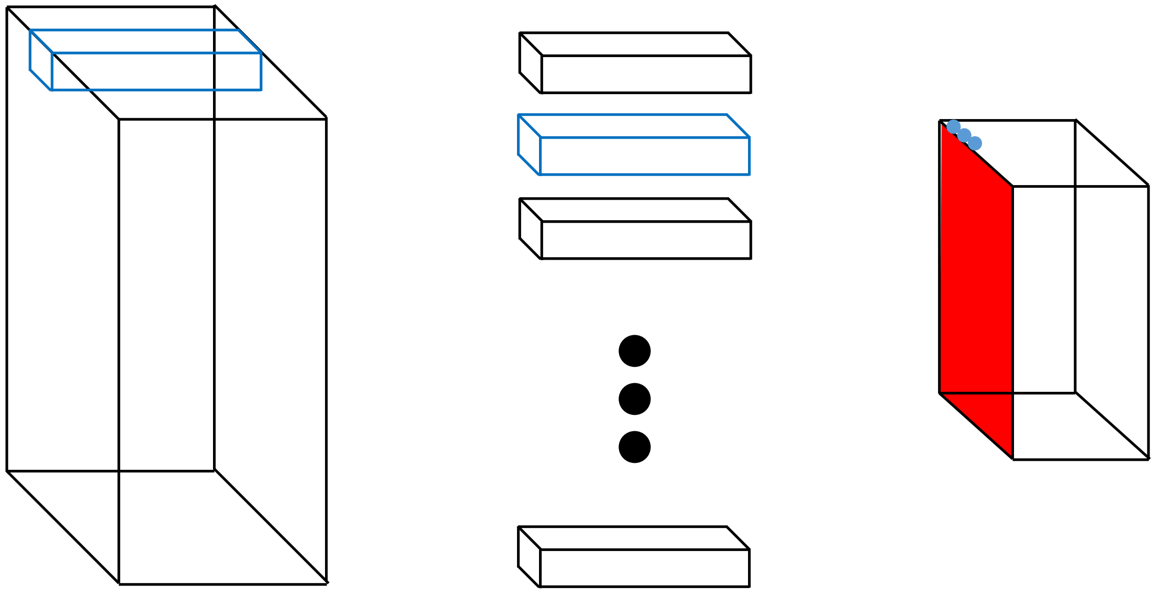


Input

Kernel Weights

Output

3-D Convolution

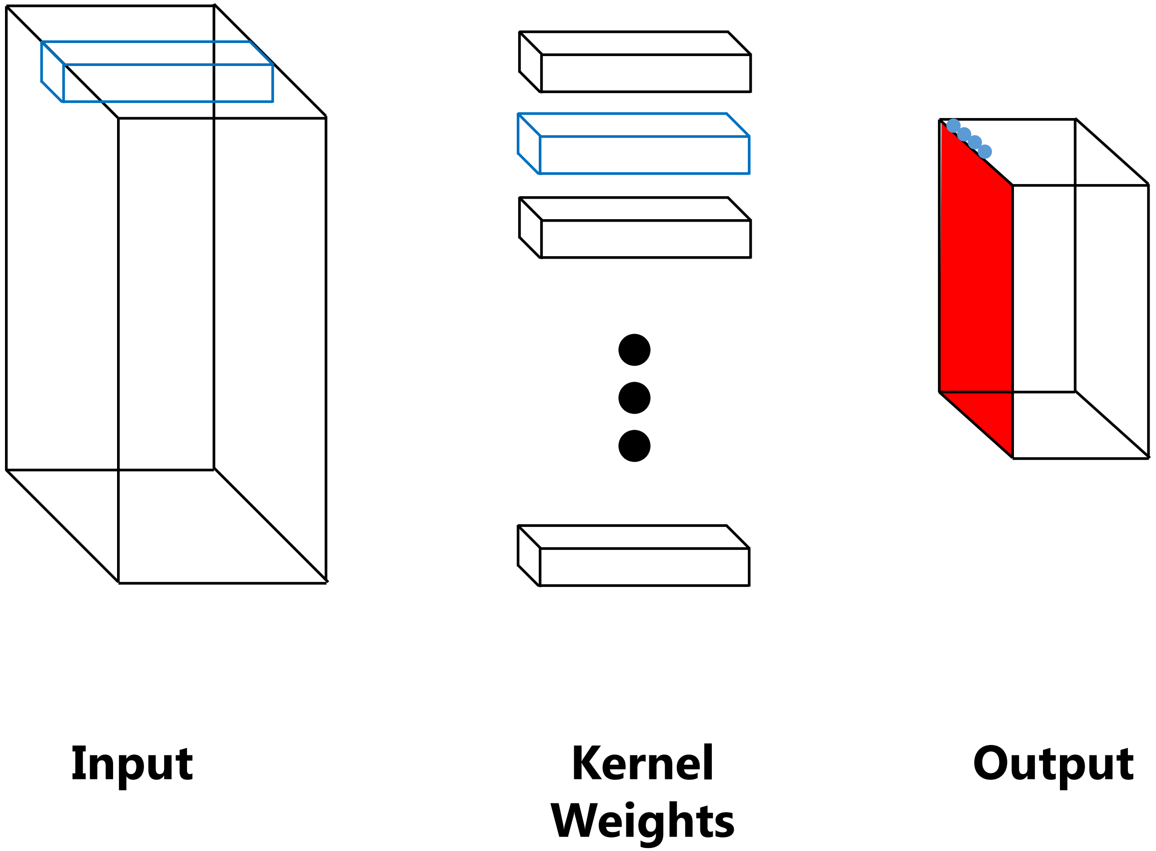


Input

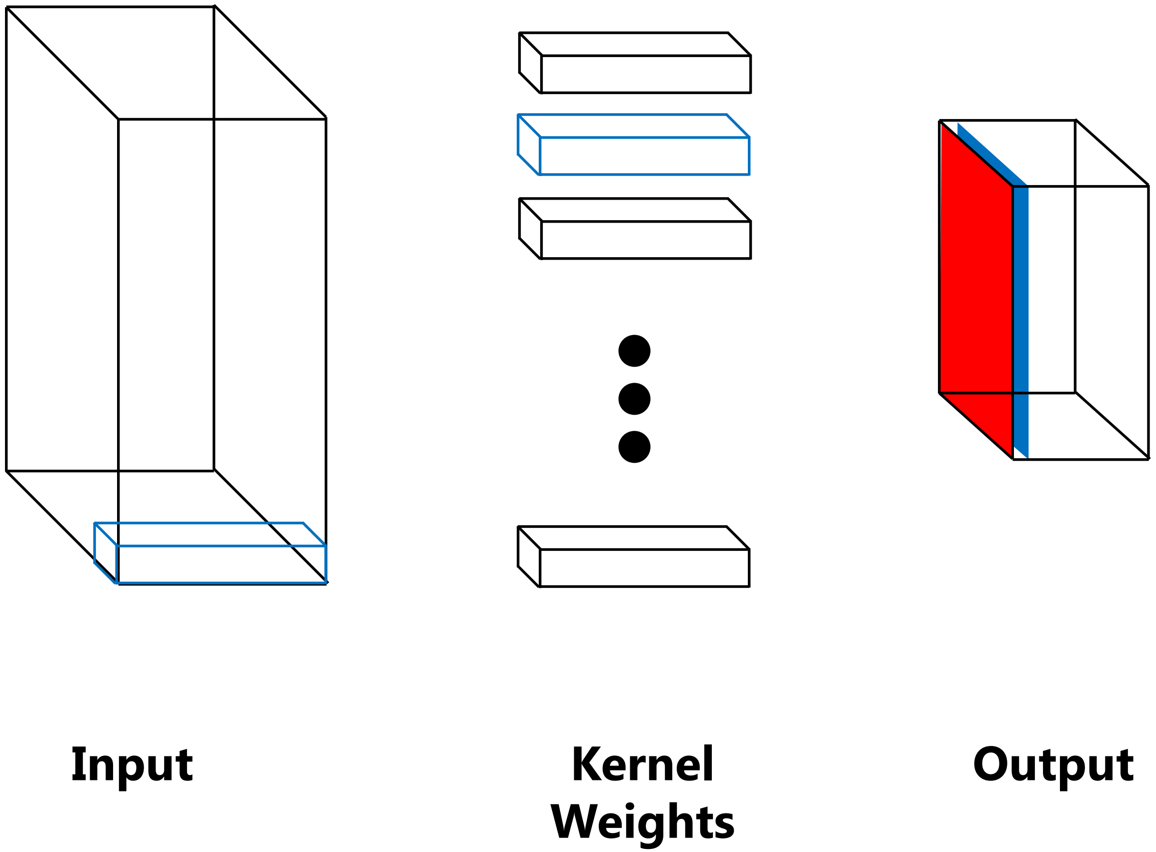
Kernel Weights

Output

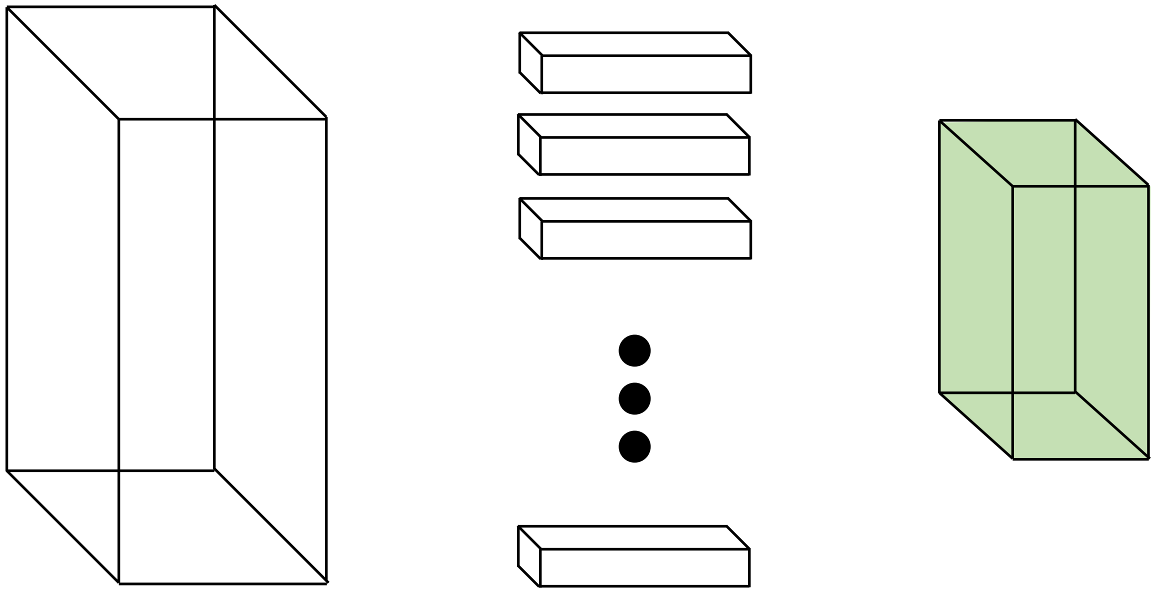
3-D Convolution



3-D Convolution



3-D Convolution



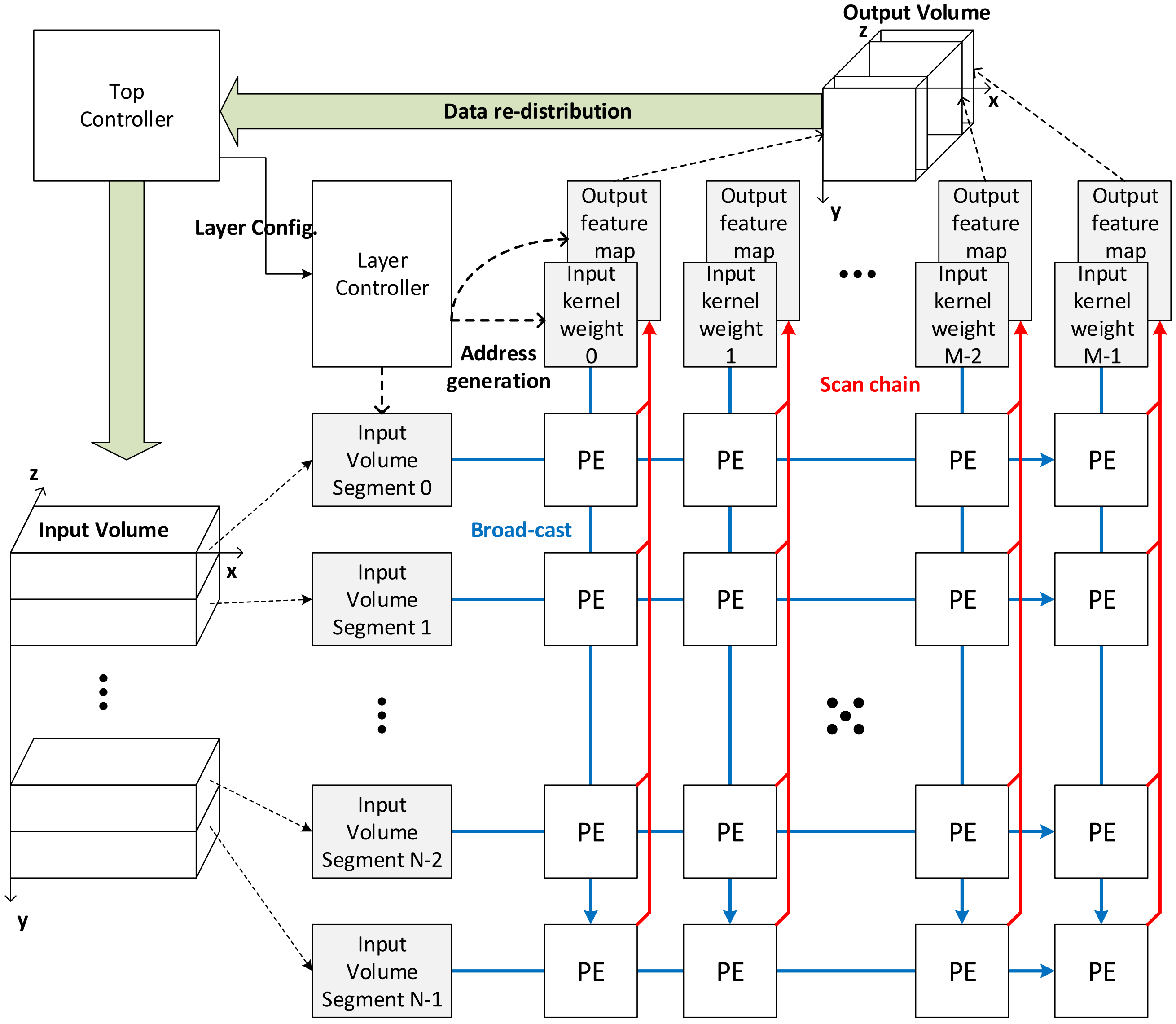
Input

Kernel Weights

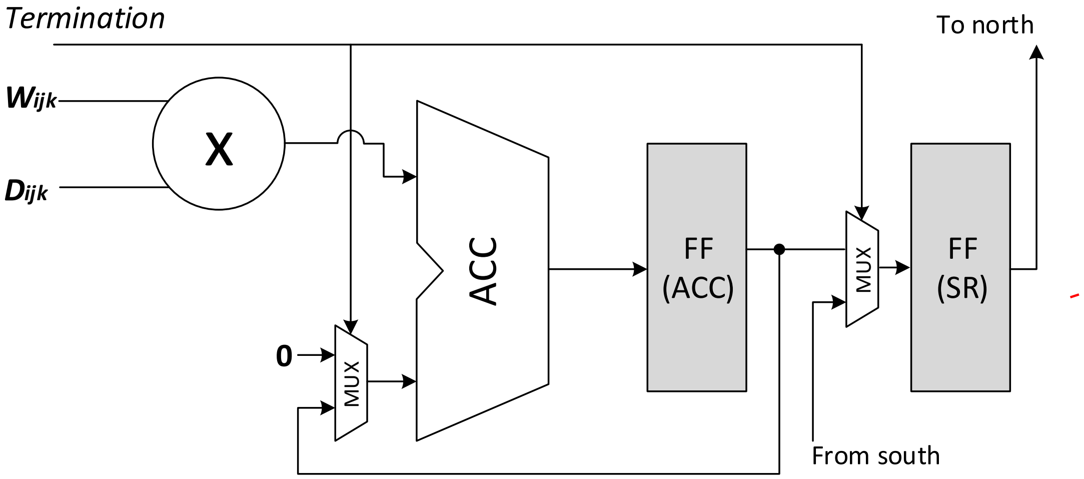
Output

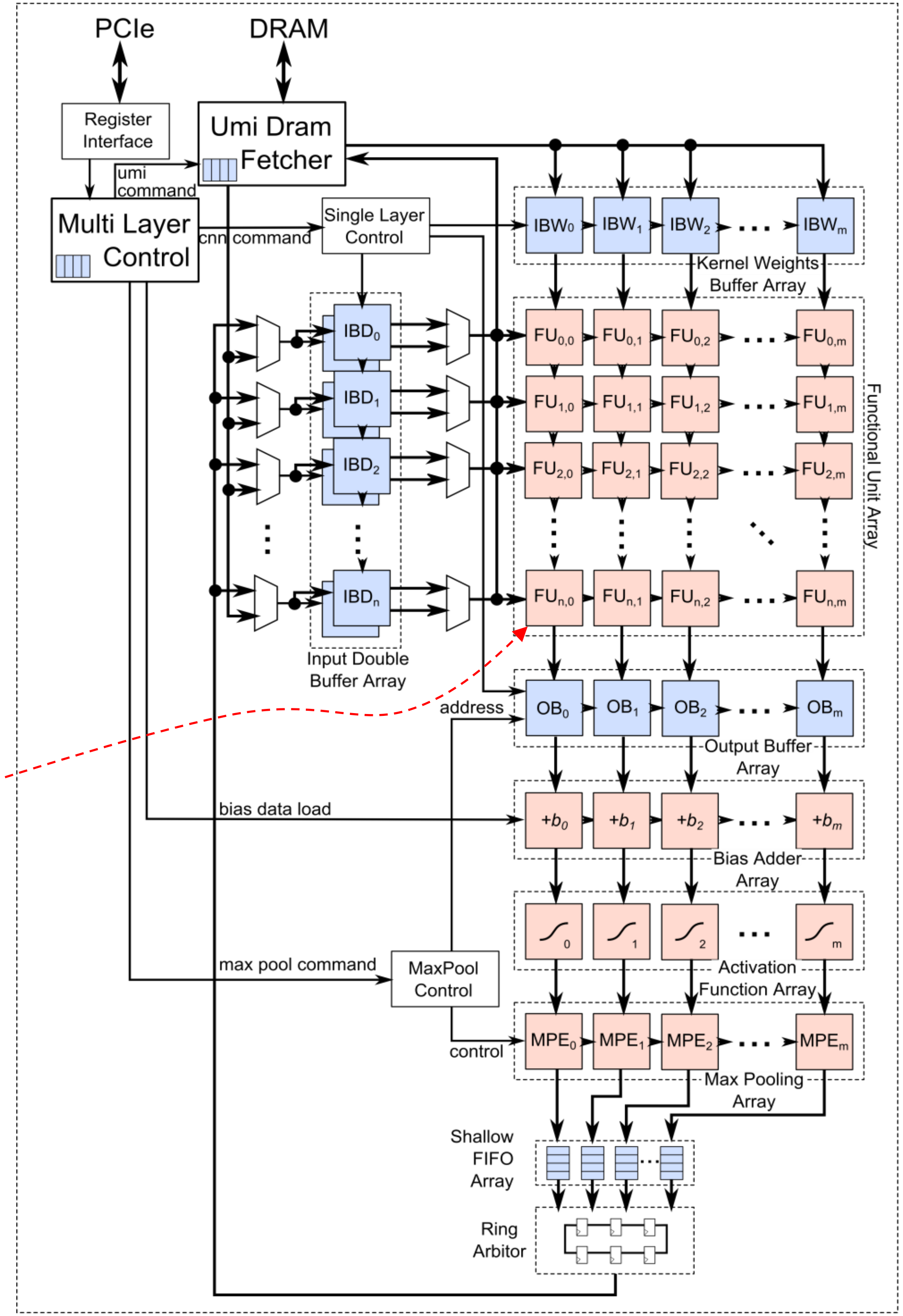
CNN Accelerator Building Block

- Configurable
  - Numerical precision (static)
  - Number of layers
  - Layer dimensions
  - Stride and pooling
- Scalable
  - Can compose multiple engines together over Catapult network
- Efficient
  - Minimize memory bandwidth via data re-distribution NoC
  - On-chip per-row broadcast



Systolic Array Microarchitecture





DEMO

ImageNet-1K Classification Performance

Platform	Library/OS	ImageNet 1K Inference Throughput	Peak TFLOPs	Effective TFLOPs	Estimated Peak Power with Server	Estimated GOPs/J (assuming peak power)
16-core, 2-socket Xeon E5-2450, 2.1GHz	Caffe + Intel MKL Ubuntu 14.04.1*	53 images/s	0.27T	0.074T (27%)	~225W	~0.3
Arria 10 GX1150	Windows Server 2012	369 images/s[1]	1.366T	0.51T (38%)	~265W	~1.9

[1] Dense layer time estimated

[2] https://github.com/soumith/convnet-benchmarks

ImageNet-1K Classification Performance

Platform	Library/OS	ImageNet 1K Inference Throughput	Peak TFLOPs	Effective TFLOPs	Estimated Peak Power with Server	Estimated GOPs/J (assuming peak power)
16-core, 2-socket Xeon E5-2450, 2.1GHz	Caffe + Intel MKL Ubuntu 14.04.1*	53 images/s	0.27T	0.074T (27%)	~225W	~0.3
Arria 10 GX1150	Windows Server 2012	369 images/s[1]	1.366T	0.51T (38%)	~265W	~1.9
NervanaSys-32 on NVIDIA Titan X	NervanaSys-32 on Ubuntu 14.0.4	4129 images/s[2]	6.1T	5.75T (94%)	~475W	~12.1

Includes server power; however, CPUs available to other jobs in the datacenter

[1] Dense layer time estimated

[2] https://github.com/soumith/convnet-benchmarks

ImageNet-1K Classification Performance

Platform	Library/OS	ImageNet 1K Inference Throughput	Peak TFLOPs	Effective TFLOPs	Estimated Peak Power for **CNN Computation**	Estimated GOPs/J (assuming peak power)
16-core, 2-socket Xeon E5-2450, 2.1GHz	Caffe + Intel MKL Ubuntu 14.04.1*	53 images/s	0.27T	0.074T (27%)	~225W	~0.3
Arria 10 GX1150	Windows Server 2012	369 images/s[1]	1.366T	0.51T (38%)	**~40W**	**~12.8**
NervanaSys-32 on NVIDIA Titan X	NervanaSys-32 on Ubuntu 14.0.4	4129 images/s[2]	6.1T	5.75T (94%)	**~250W**	**~23.0**

Under-utilized FPGA vs.
highly tuned GPU-friendly workload

[1] Dense layer time estimated

[2] https://github.com/soumith/convnet-benchmarks

Projected Improvements with Tuning

Platform	Library/OS	ImageNet 1K Inference Throughput	Peak TFLOPs	Effective TFLOPs	Estimated Peak Power for **CNN Computation**	Estimated GOPs/J (assuming peak power)
16-core, 2-socket Xeon E5-2450, 2.1GHz	Caffe + Intel MKL Ubuntu 14.04.1*	53 images/s	0.27T	0.074T (27%)	~225W	~0.3
Arria 10 GX1150	Windows Server 2012	~~369 images/s~~[1] **~880 images/s**	1.366T	~~0.51T (38%)~~ **~1.2T (89%)**	~40W	~~20.6~~ **~30.6**
NervanaSys-32 on NVIDIA Titan X	NervanaSys-32 on Ubuntu 14.0.4	4129 images/s[2]	6.1T	5.75T (94%)	~250W	~23.0

Projected Results Assuming Floorplanning and Scaling Up PEs

[1] Dense layer time estimated

[2] https://github.com/soumith/convnet-benchmarks

Are FPGAs a Promising Target in the Datacenter for Deep Learning? **Yes.**

- Best-case FPGA design is ~1/5th GPU throughput but can overtake at scale
- Although CNNs are ideal on GPUs, FPGAs with hardened FPUs can achieve GPU-like energy efficiency
- FPGA is 7X faster (~16X within reach) than multicore CPUs while flexible enough for diverse cloud scenarios (Bing Ranking, Azure SmartNIC)

Related Work

- ASICs
  - [Holler’90], [Chen’14], [Cavigelli’15], etc.
- FPGAs
  - [LeCun’09], [Farabet’10], [Aysegul’13], [Baidu’14], [Gokhale’15], [Zhang’15], etc.
- GPUs, Appliances
  - Nervana, Nvidia DIGITS, Ersatz, etc.

Thank you!

erchung@microsoft.com