hls4ml

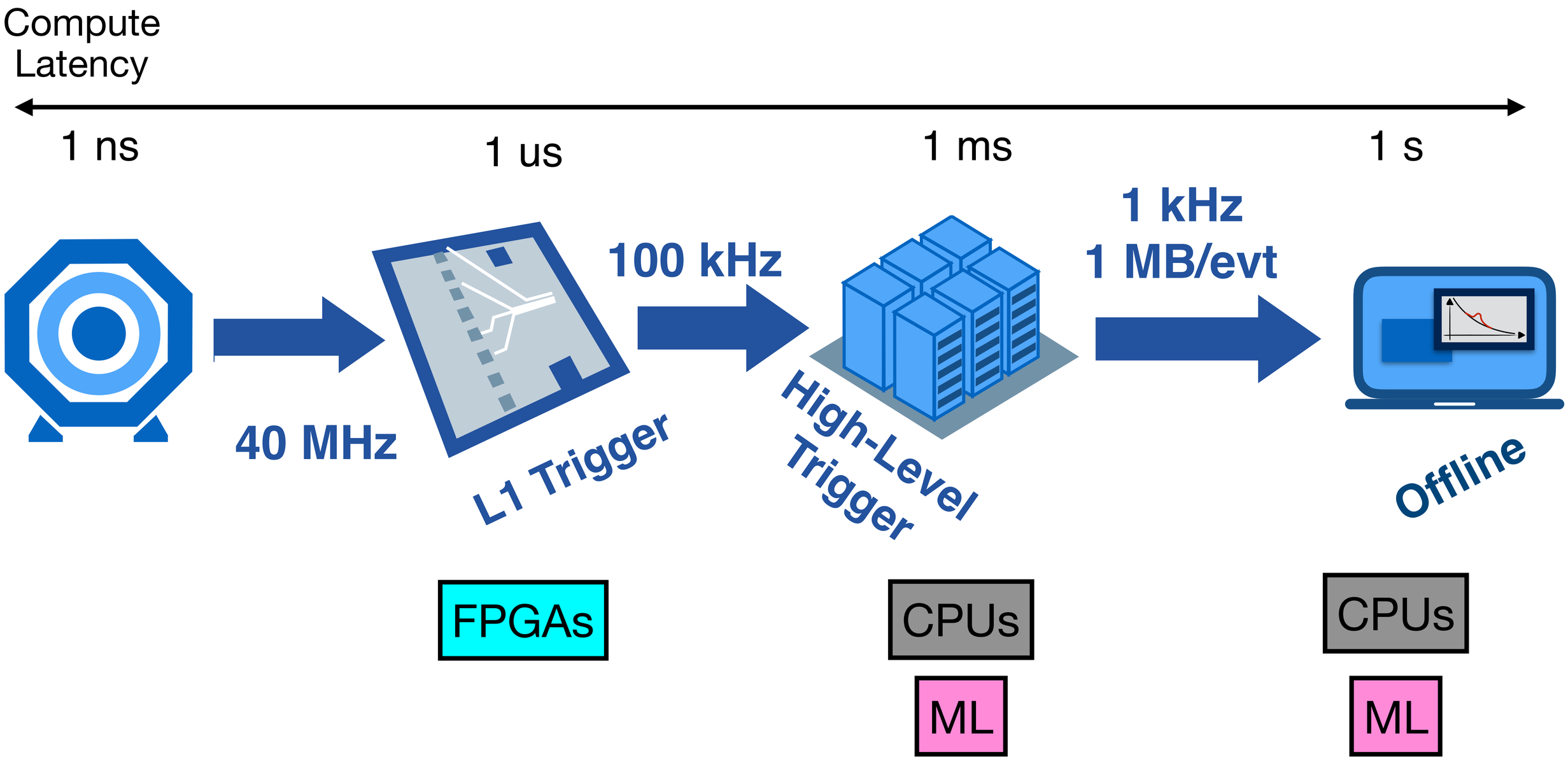
Real-time fast machine learning inference

Nhan Tran

January 14, 2019

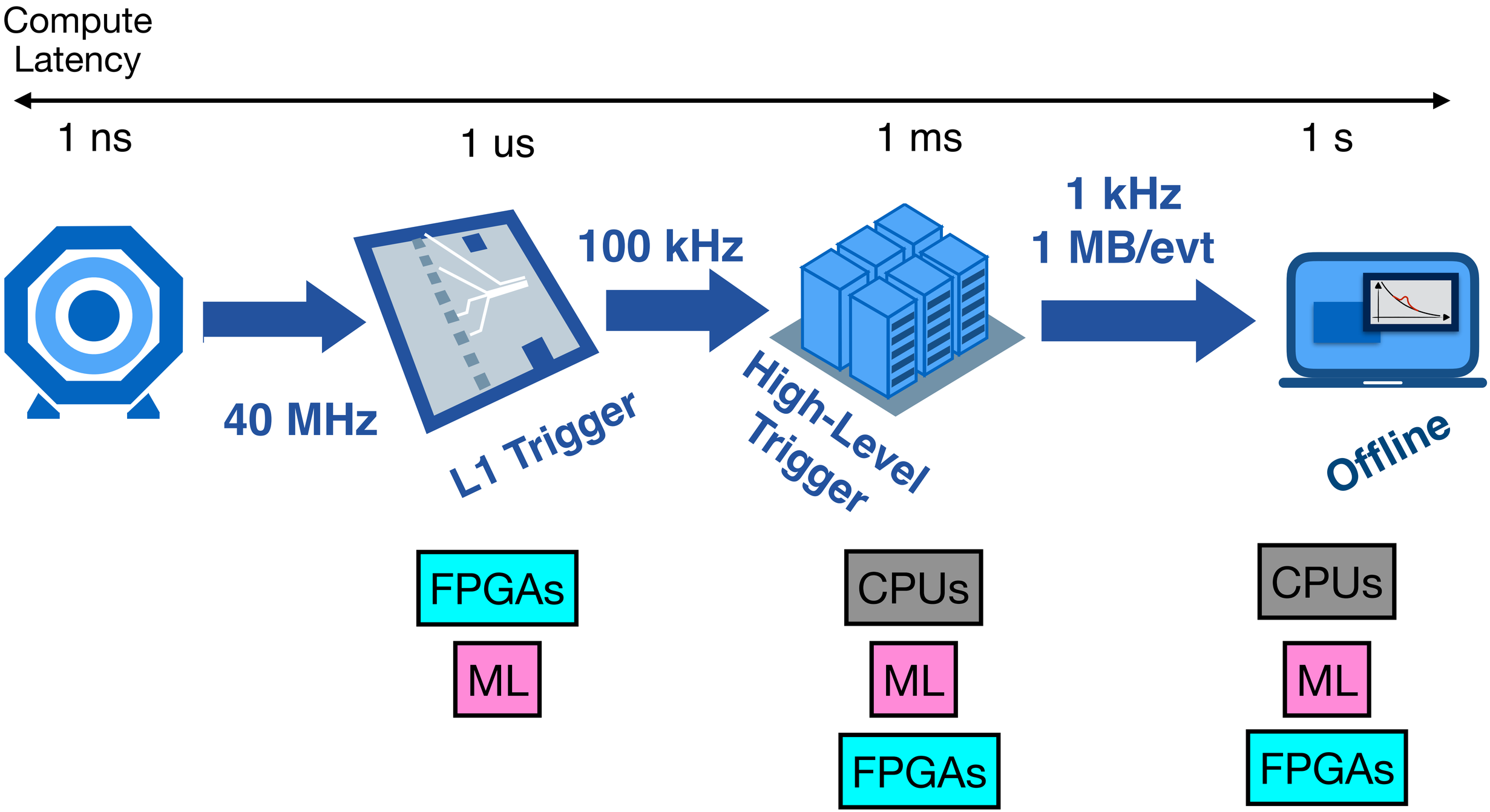
Chicagoland HEP mini-workshop

CMS EVENT PROCESSING



Apologies to my ATLAS colleagues, this is the only picture I had handy
Some latencies and stages are different, but the idea is the same

CMS EVENT PROCESSING



GPUs are popular here, particularly for training. For inference, FPGAs and ASICs are actively being developed in industry (more later)

OUTLINE

Part I: Machine learning in the L1 hardware trigger

Part II: FPGA (and other hardware) accelerators for compute acceleration

Contributors:

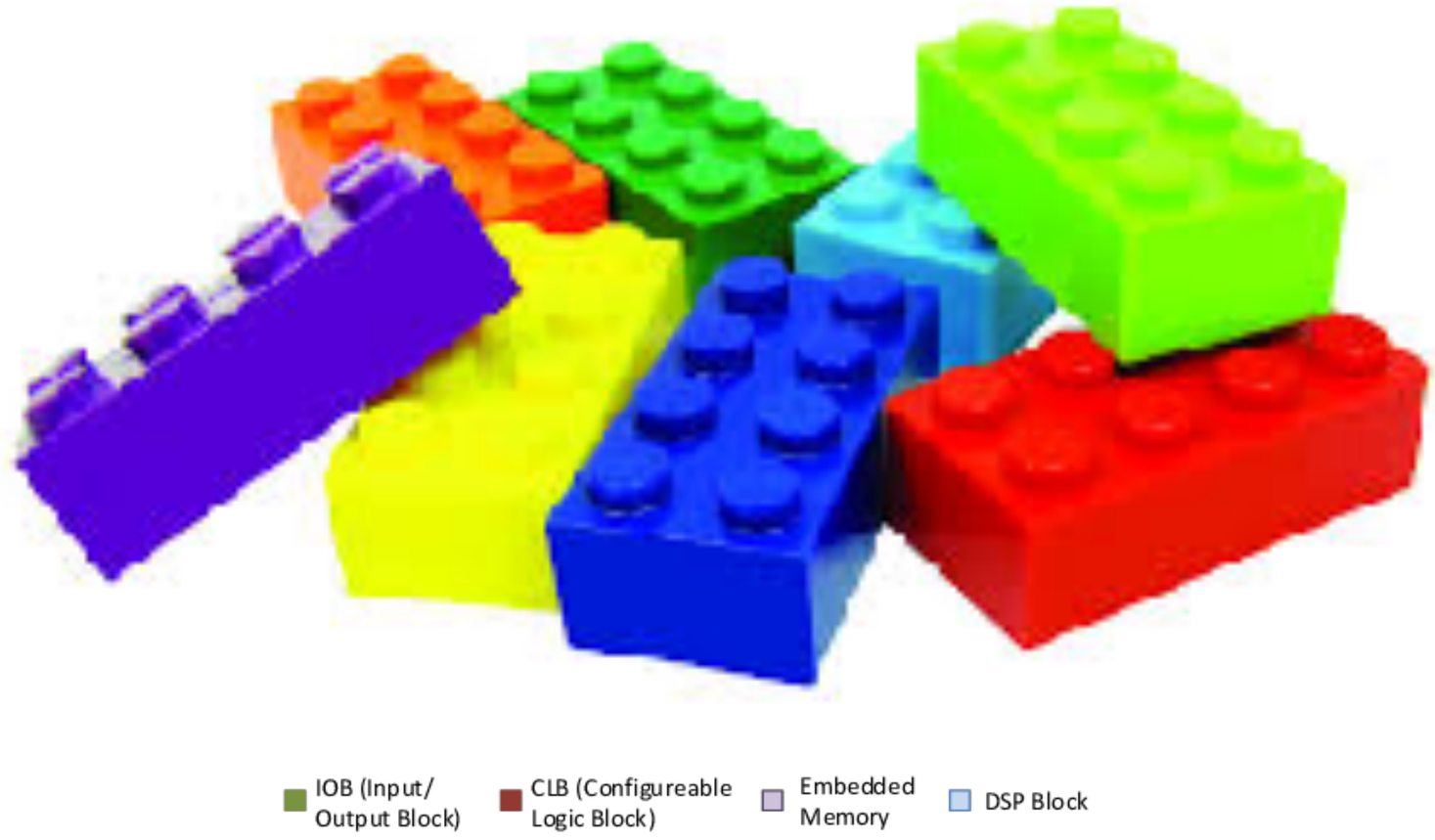
Vladimir Loncar, Jennifer Ngadiuba, Maurizio Pierini (CERN)
Javier Duarte, Sergo Jindariani, Ben Kreis, Ryan Rivera, N.T. (FNAL)
Sioni Summers (Imperial College)
Phil Harris, Dylan Rankin (MIT)
Zhenbin Wu (UIC)
+
Paolo D'alberto (Xilinx), Giuseppe di Guglielmo (Columbia, EE),
Song Han (MIT,EE), EJ Kreinar (Hawkeye 360),

ALL FPGA ARCHITECTURE

O(50-100) optical transceivers running at ~O(15) Gbs

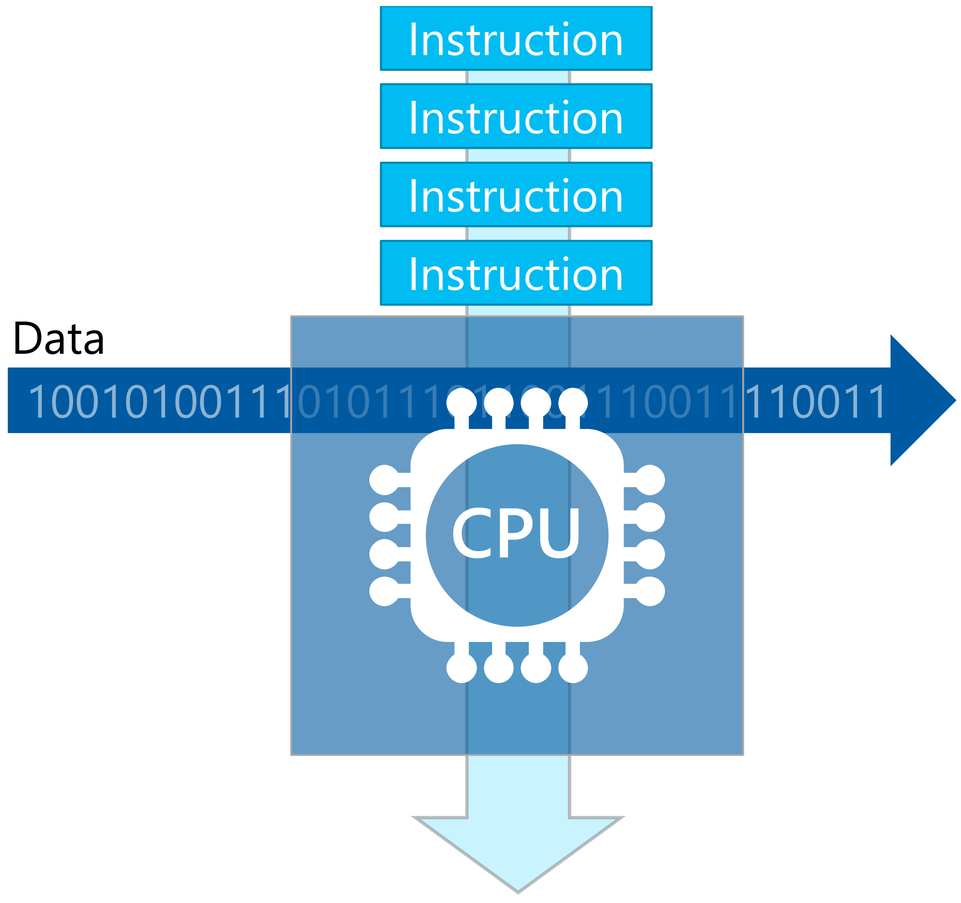
Typical modern FPGA:
(Kintex ultrascale+)
1.3M FFs
700k LUTs
5500 DSPs
2200 BRAMs

FPGA
"programmable hardware"



DSPs (multiply-accumulate, etc.)
Flip Flops (registers/distributed memory)
LUTs (logic)
Block RAMs (memories)

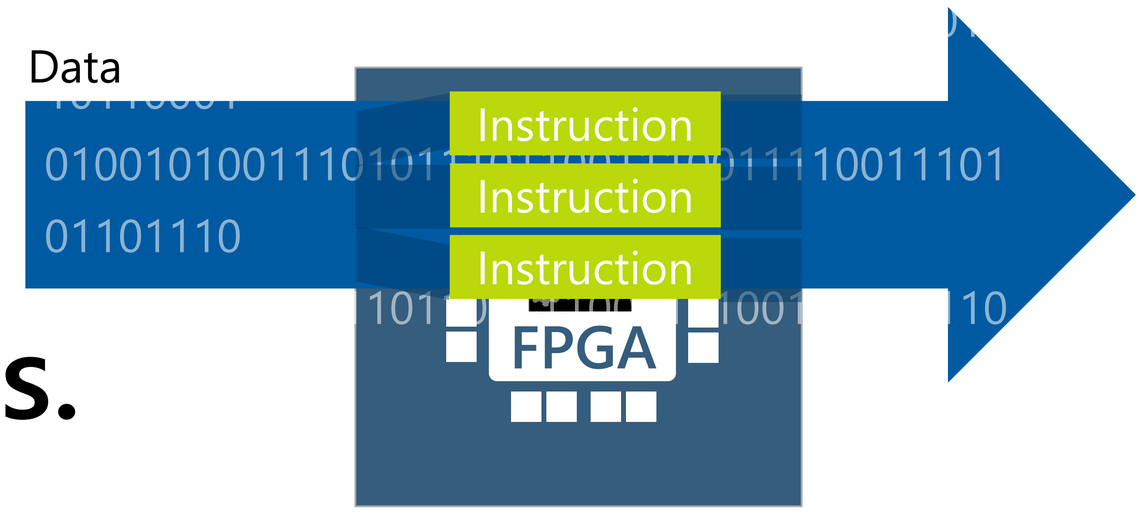
FPGA PROGRAMMING



CPU: temporal compute

VS.

Traditionally, FPGAs programmed with low-level languages like Verilog and VHDL



FPGA: spatial compute

High level synthesis (HLS)

New languages C-level programming with specialized preprocessor directives which synthesizes optimized firmware;

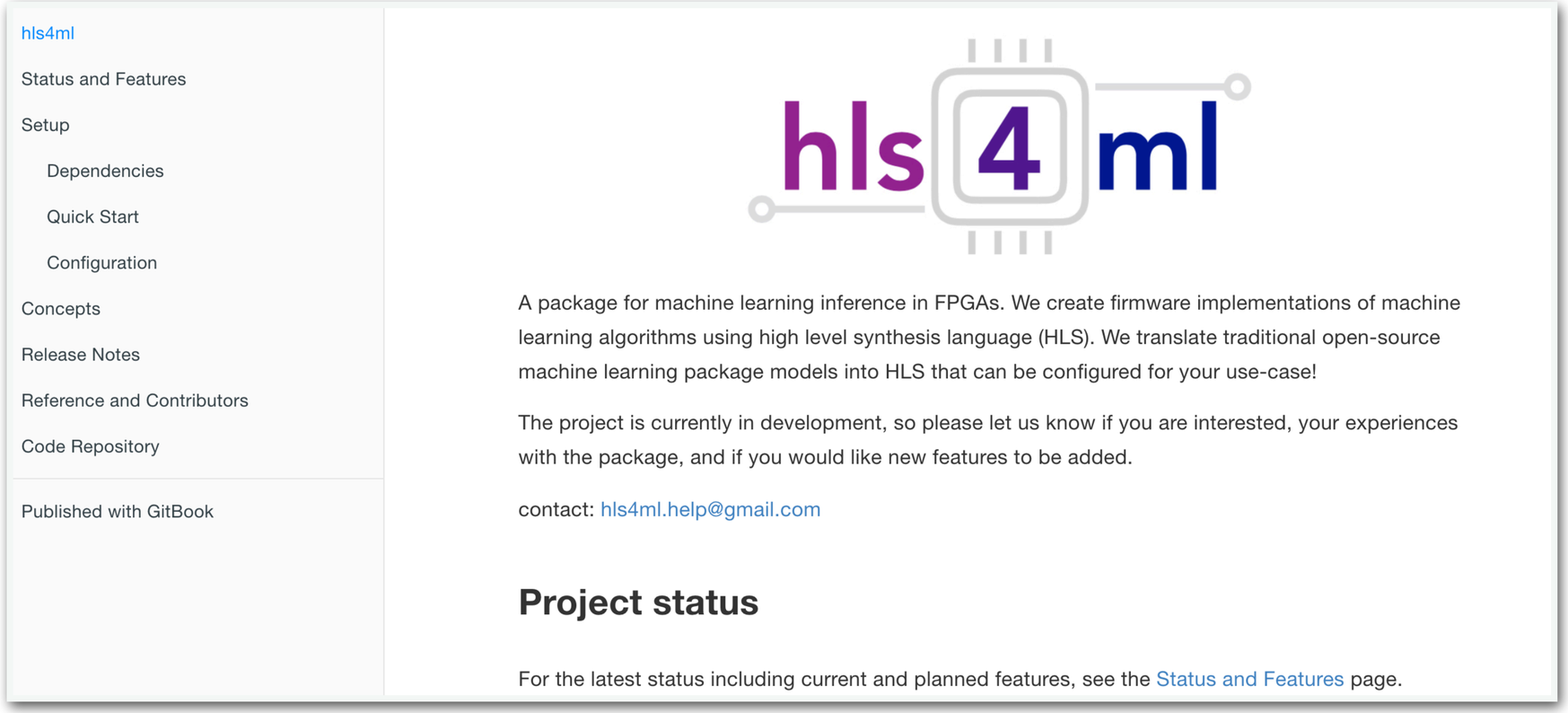
Drastically reduces development times for firmware

high level synthesis for machine learning

~~**hlsfml**~~

hls4ml

https://hls-fpga-machine-learning.github.io/hls4ml/

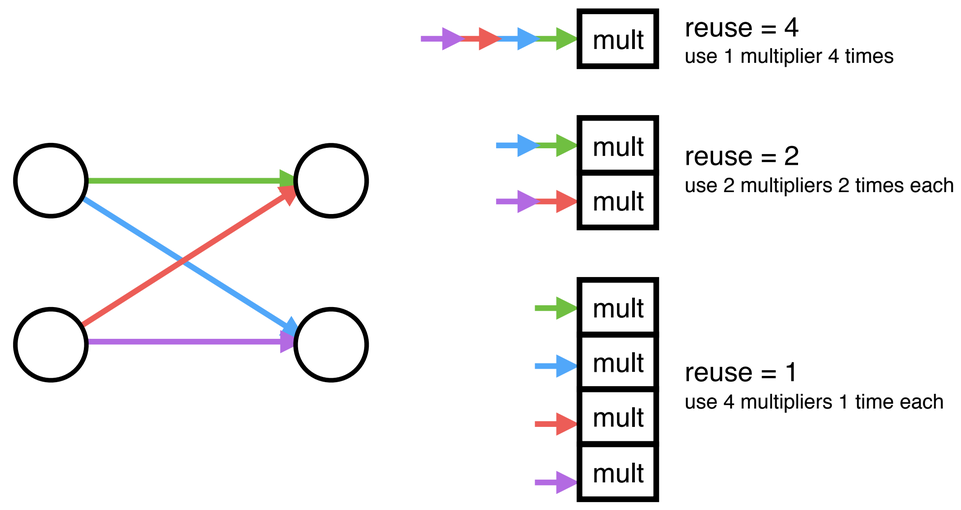


arXiv:1804.06193

(Energy) Efficient Neural Networks

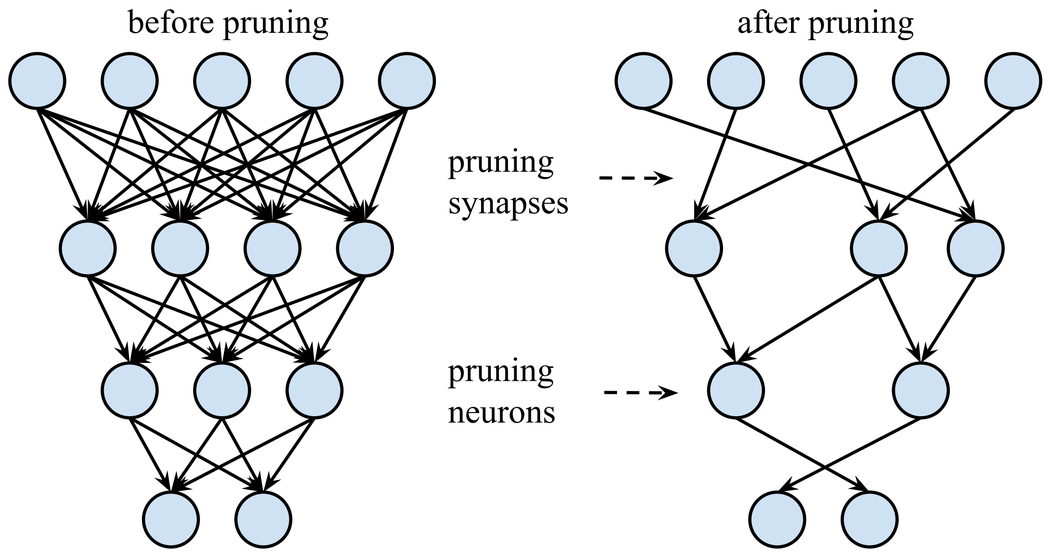
Emergent engineering field, efficient implementation of NN architecture

Parallelization: controlling operations to be performed simultaneously



Compression/Pruning:

maintain the same performance while removing low weight synapses and neurons (many schemes)



Quantization/Approximate math:

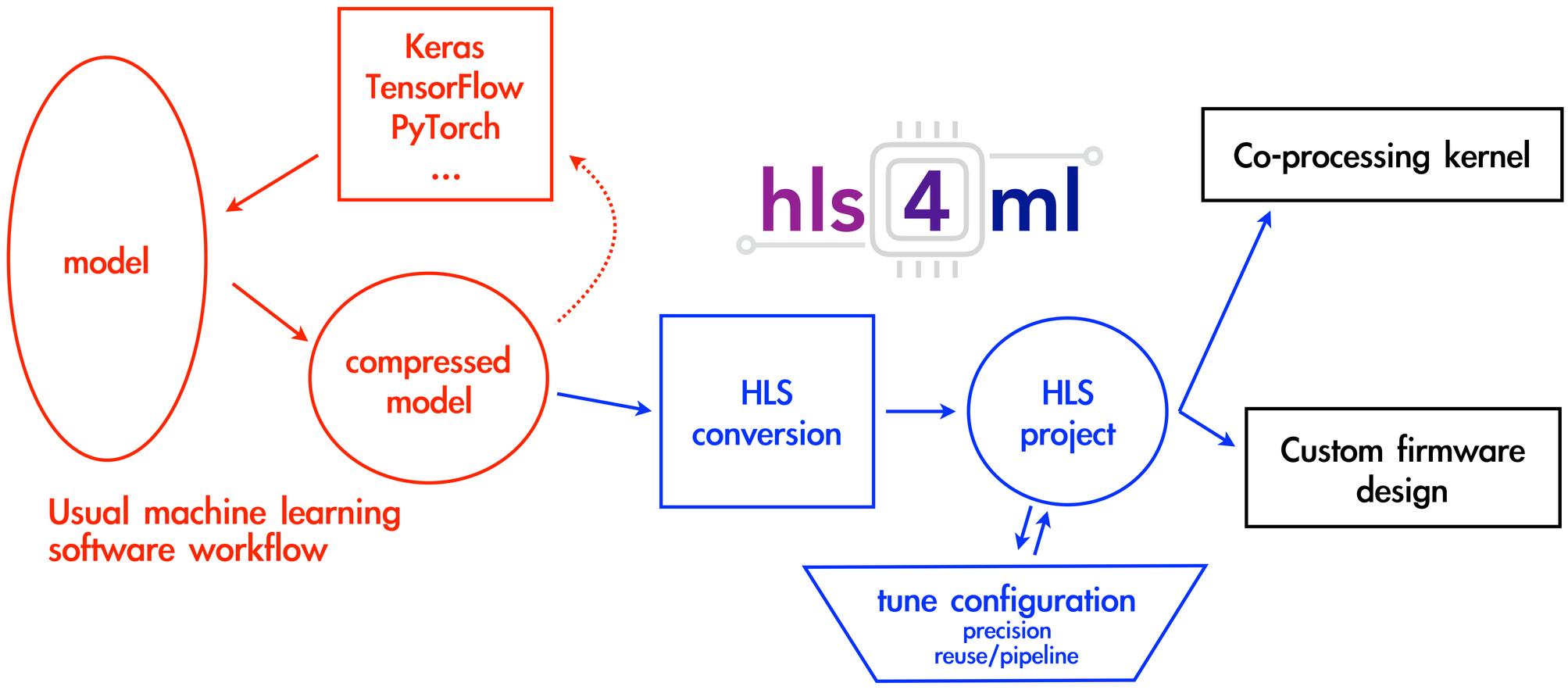
32-bit floating point math is overkill

20-bit, 18-bit, …? fixed point, integers? binarized NNs?

For further reading, start here: https://arxiv.org/pdf/1510.00149v5.pdf

PROJECT OVERVIEW

Quantization, Compression, Parallelization made easy with hls4ml!



Results and outlook:

4000 parameter network inferred in < 100 ns with 30% of FPGA resources!

Muon pT reconstruction with NN reduces rate by 80%

Larger networks and different architectures actively developed (CNN, RNN, Graph)

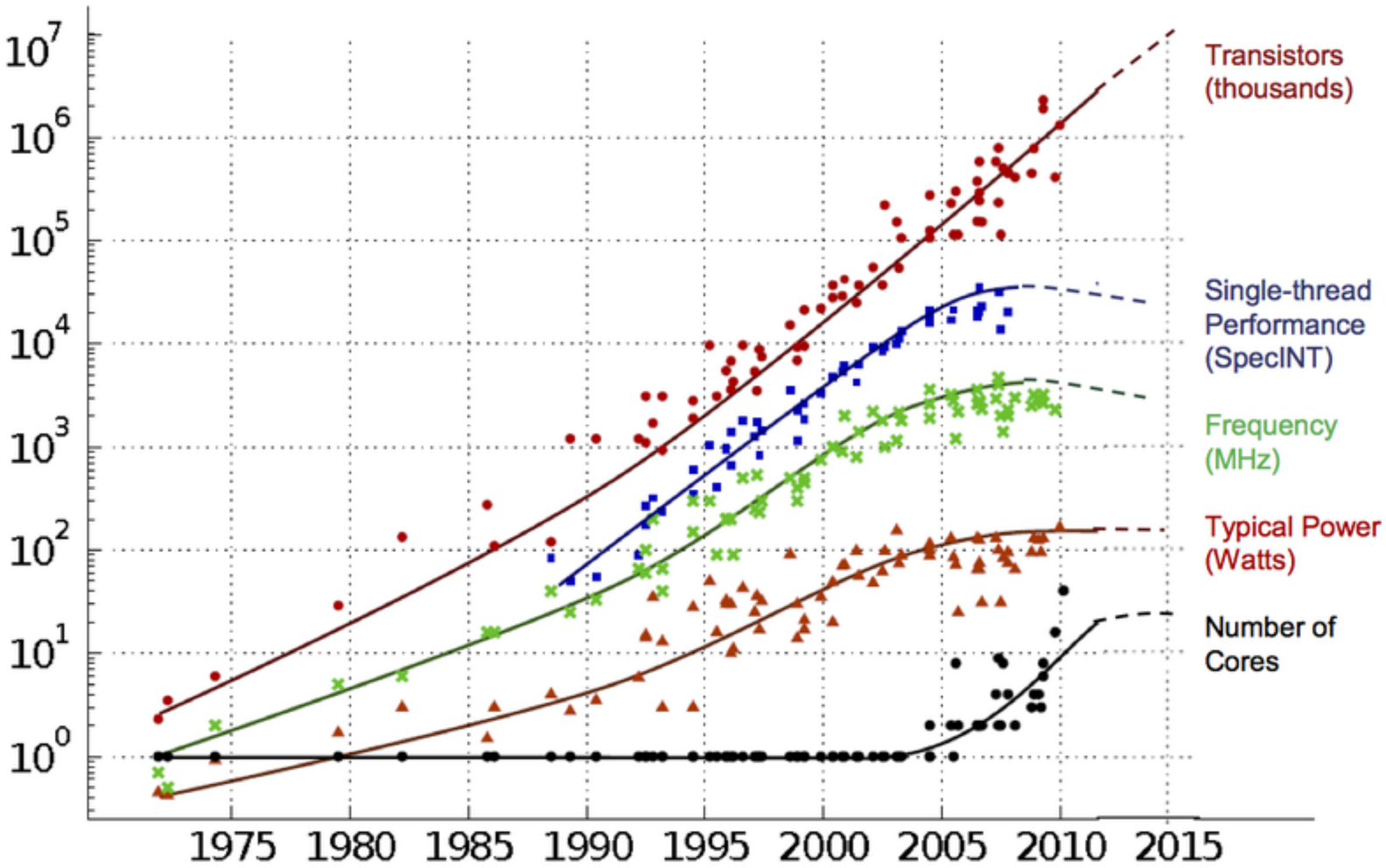
Outline

Part I: Machine learning in the L1 hardware trigger

Part II: FPGA (and other hardware) accelerators for compute acceleration

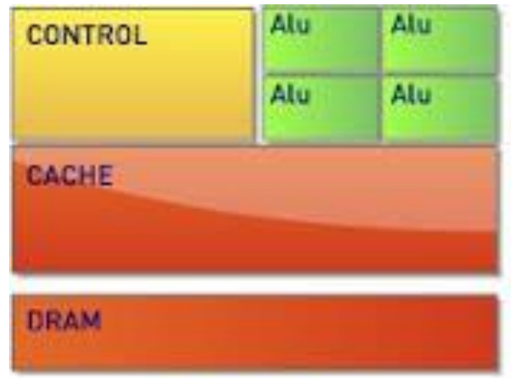
Contributors:
Jennifer Ngadiuba, Maurizio Pierini **(CERN)**
Javier Duarte, Burt Holzman, Sergo Jindariani, Ben Kreis, Kevin Pedro, Mia Liu, Nhan Tran, Aris Tsaris **(Fermilab)**
Phil Harris, Dylan Rankin **(MIT)**
Zhenbin Wu **(UIC)**
+
Ted Way, David Lee, Suffian Khan **(MS Azure ML)**
Scott Hauk, Shih-Chieh Hsu, Dustin Werran **(U Washington)**

Moore's Law and Dennard Scaling

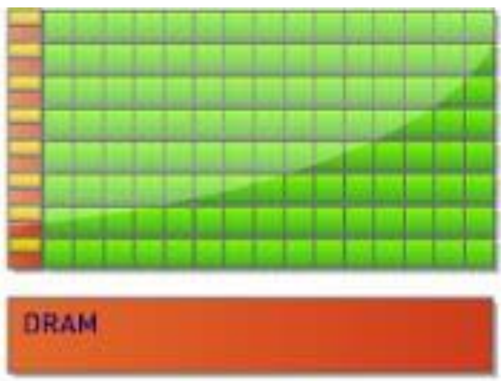


Original data collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond and C. Batten
Dotted line extrapolations by C. Moore

Moore's Law continues
…but Dennard Scaling fails



CPU



GPU

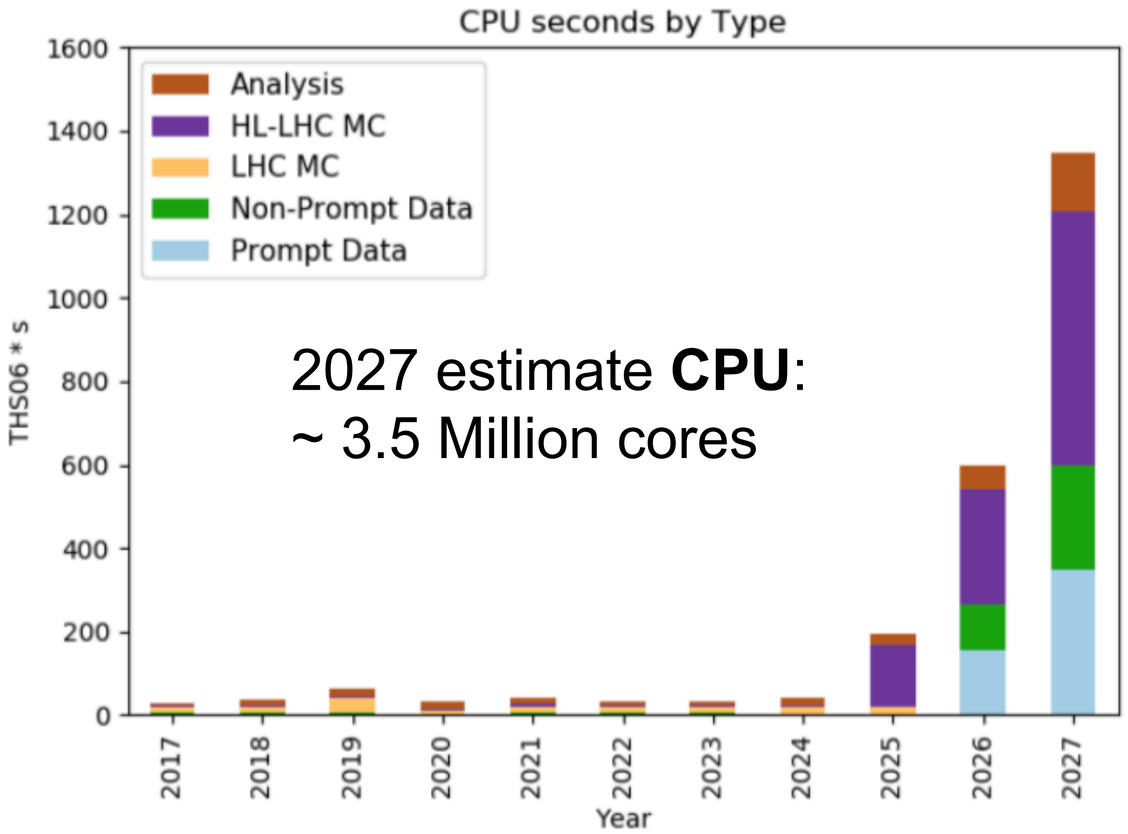
Single threaded performance not improving

Circa ~2005: "The Era of Multicore"

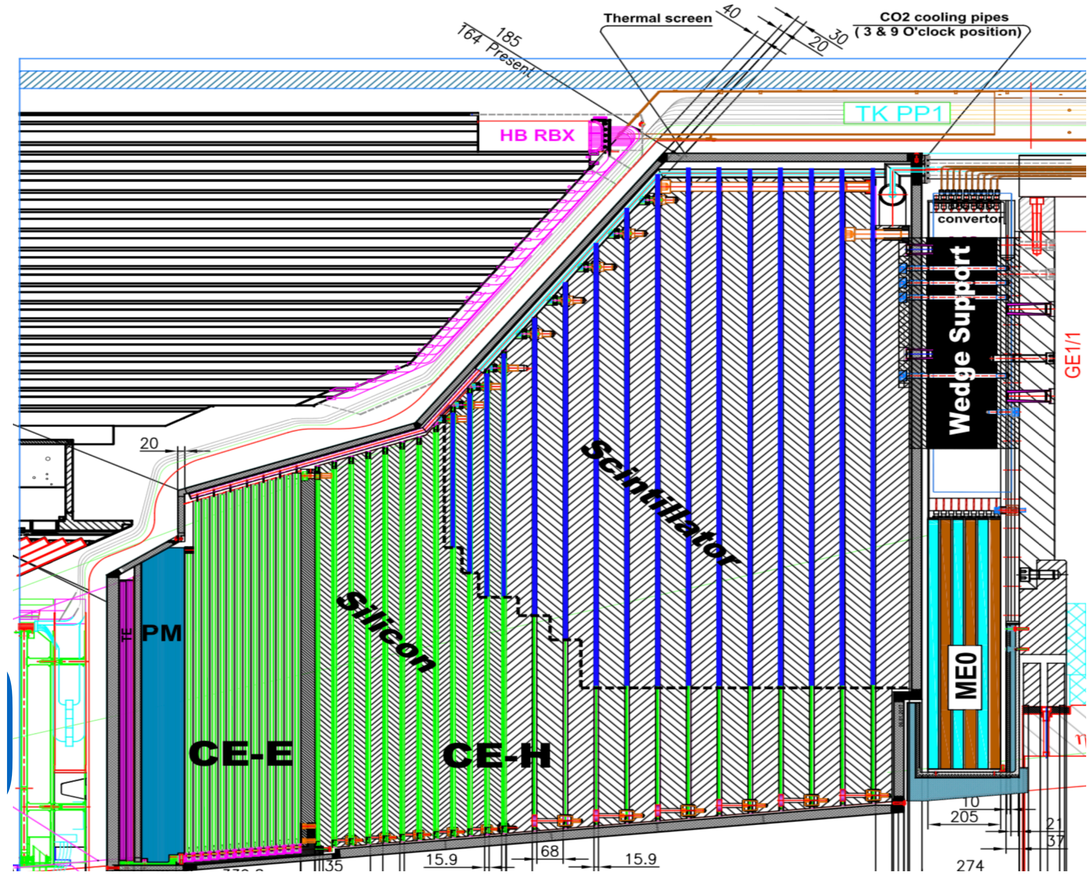
→ Today: Transition to the "Era of Specialization"? (c.f. Doug Burger)

HEP COMPUTING CHALLENGES

Computing is a major concern for HL-LHC



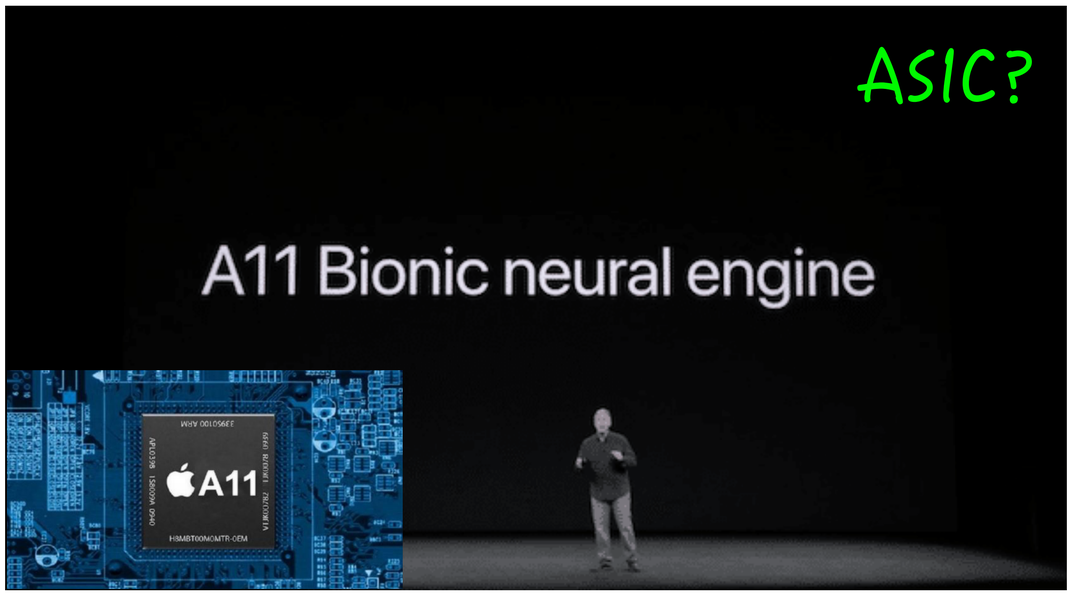
Data volume: 50x

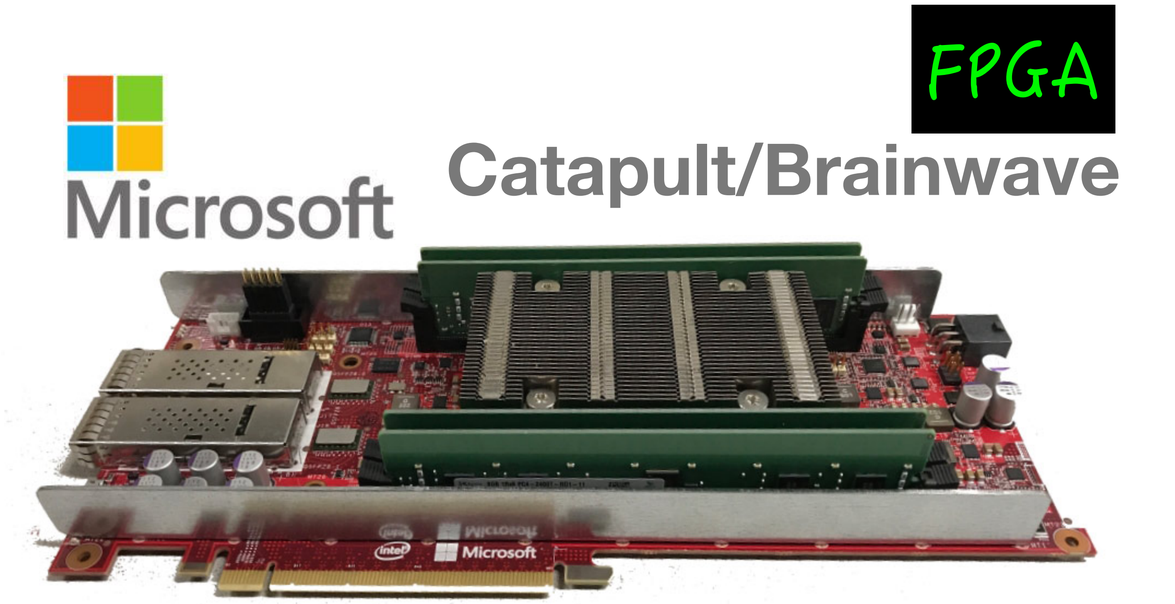


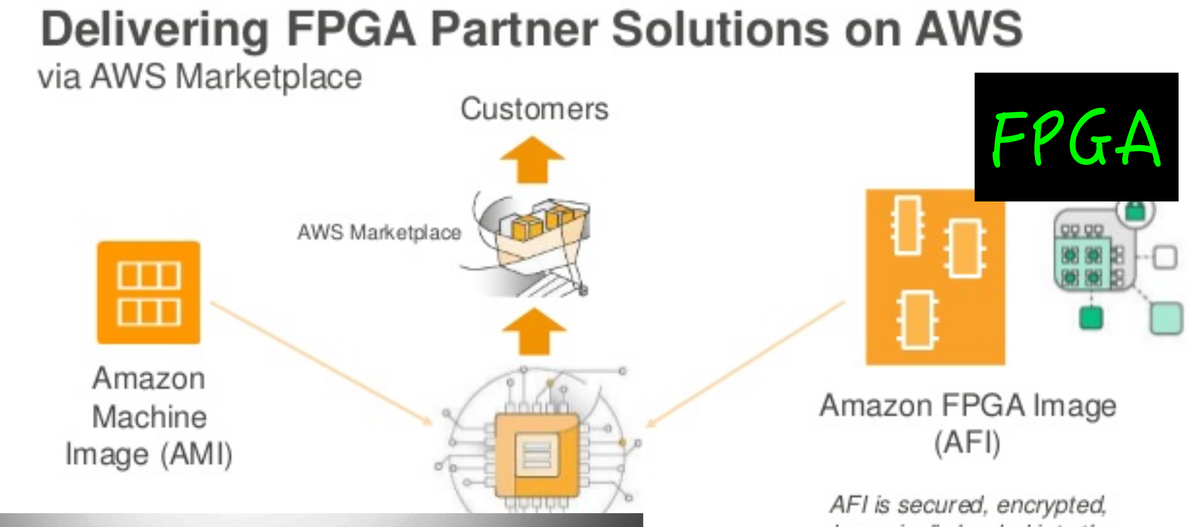
Environment complexity: 5x
Detector complexty: 10x

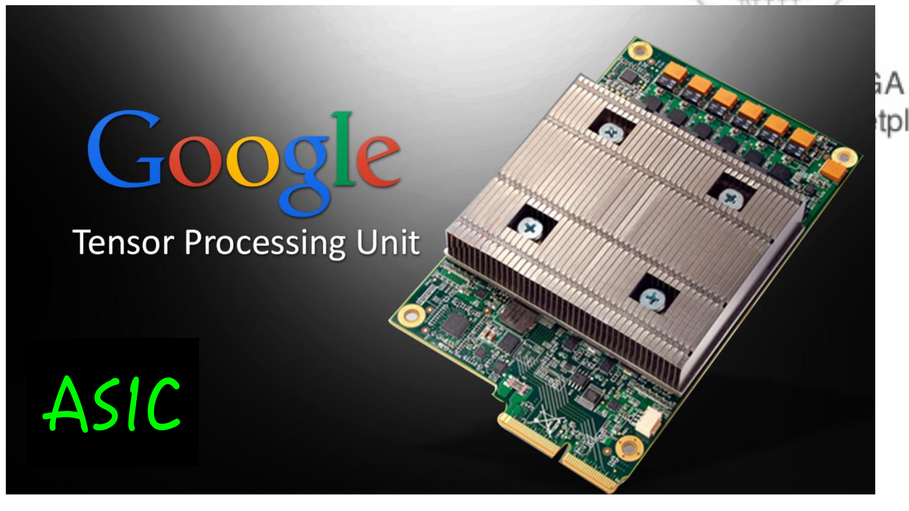
#TRENDING

Specialized co-processor hardware for machine learning inference









INTEL® FPGA ACCELERATION HUB

The Intel® Xeon® Acceleration Stack for FPGAs is a robust framework enabling data center applications to leverage an FPGA's potential to increase

ML Babel Fish

Large gains from hardware accelerating co-processors
Industry trending towards specialized computing paradigms

Option 1

re-write physics algorithms for new hardware

Language: OpenCL, OpenMP, HLS, …?

Hardware: FPGA, GPU

Option 2

re-cast physics problem as a machine learning problem

Language: C++, Python (TensorFlow, PyTorch,…)

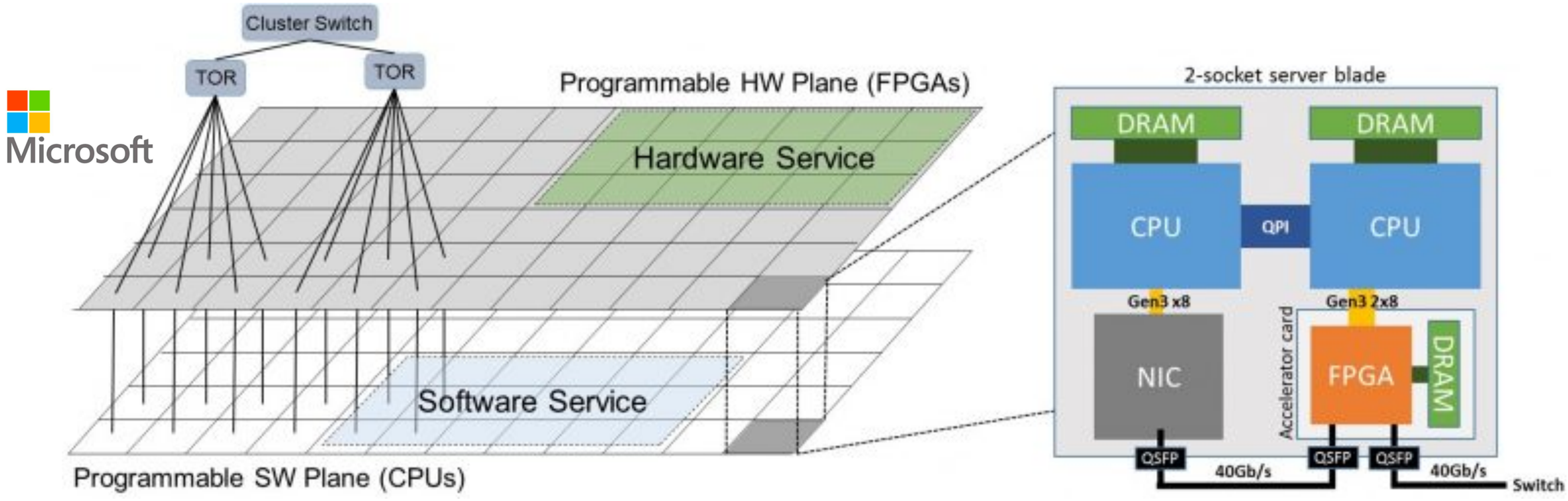
Hardware: FPGA, GPU, ASIC

hls4ml/ Industry

Why (Deep) Machine Learning?

a common *language* for solving problems
which can universally be expressed on optimized computing hardware
and follow industry trends
(simuation, reconstruction, & analysis!)

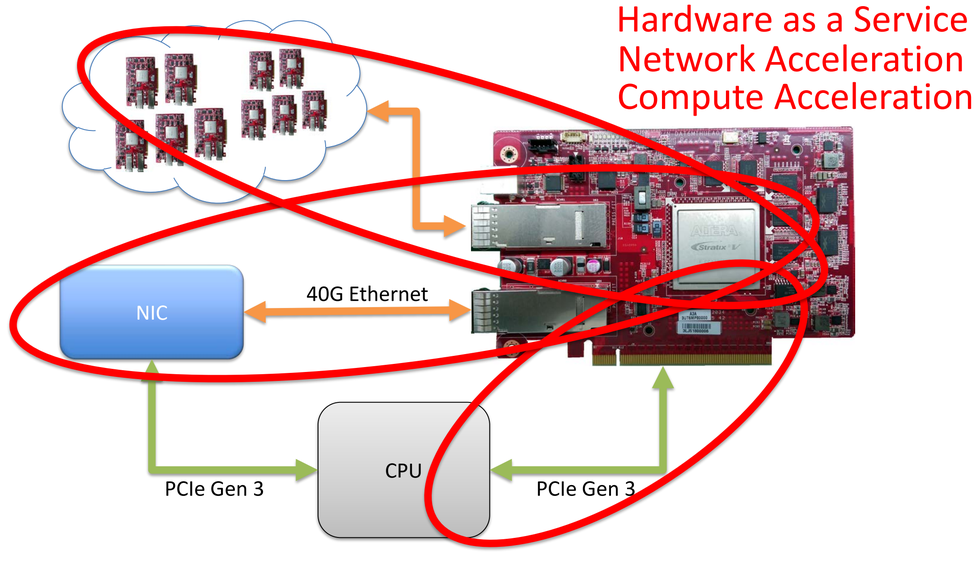
Services for **O**ptimized **N**etwork **I**nference on **C**oprocessors



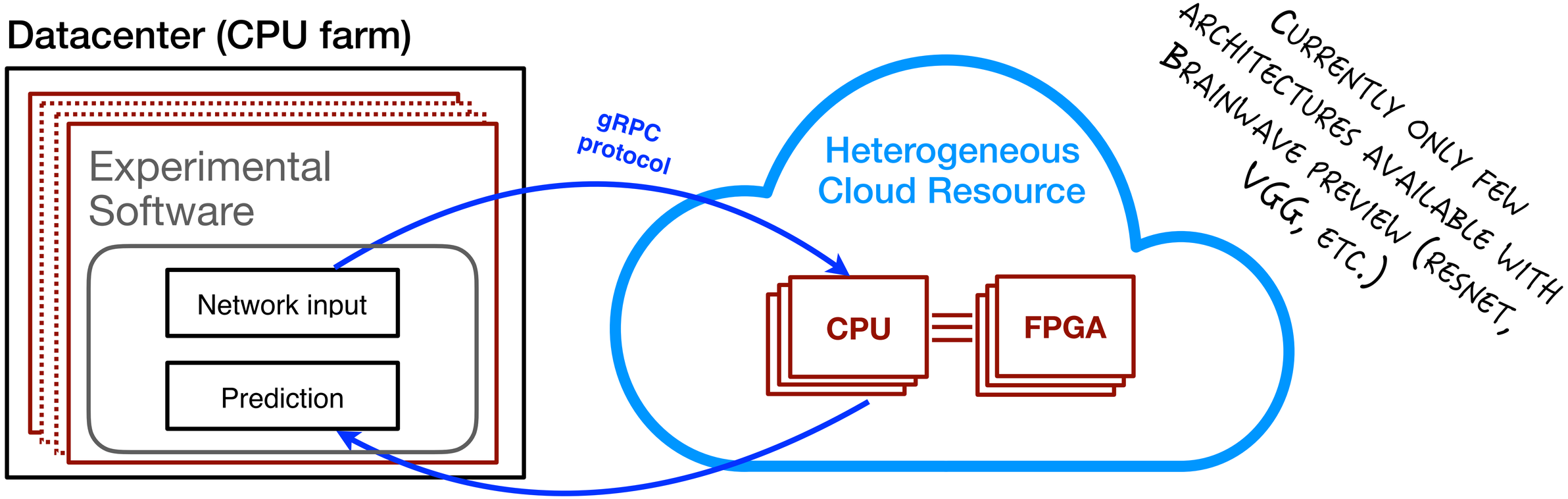
Most viable way to deploy accelerating coprocessor hardware ***as a service***

Not viable: attach GPU/FPGA/ASIC to CPU on every node on the computing grid

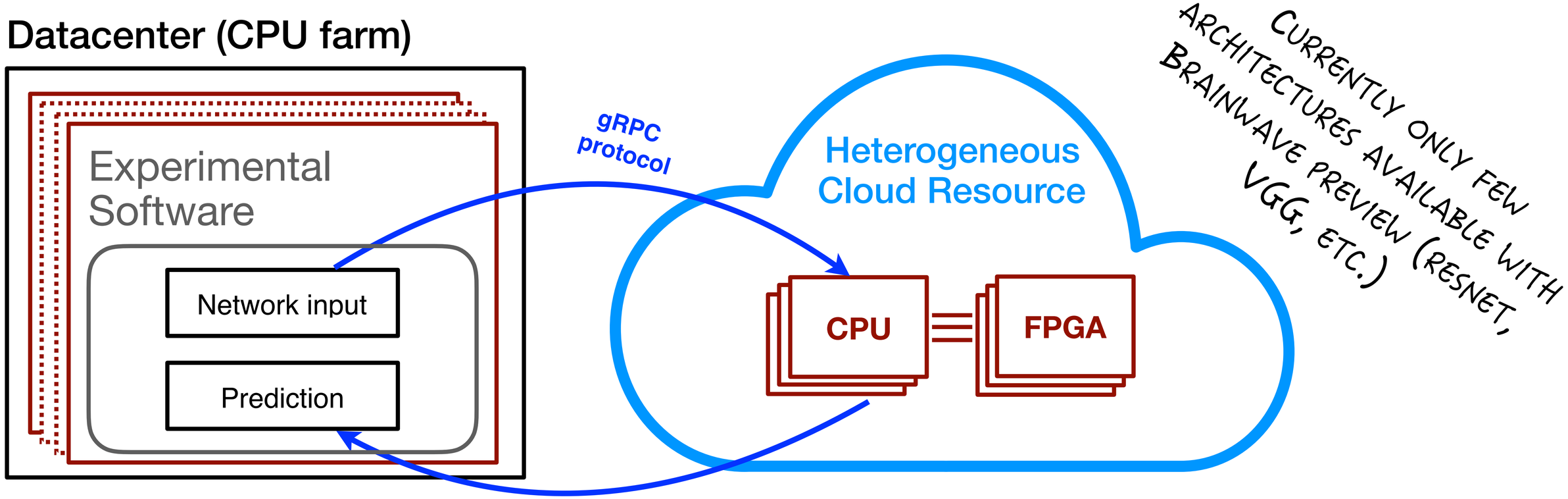
Abstracts away the hardware;
think of each inference as a “web query”



First tests with MS Brainwave



First tests with MS Brainwave



CPU comparison:

Intel i7 3.6 GHz (8 core, TF v1.10)
180 ms

Intel i7 3.6 GHz (1 core, TF v1.10)
500 ms

Intel i7 3.6 GHz (1 core, TF v1.06)
1.2 s

Intel Xeon 2.6 GHz (1 core, TF v1.06)
1.75 s

FPGA (Brainwave)

Local (Edge)
10 ms

Remote (FNAL to Virginia)
50 ms

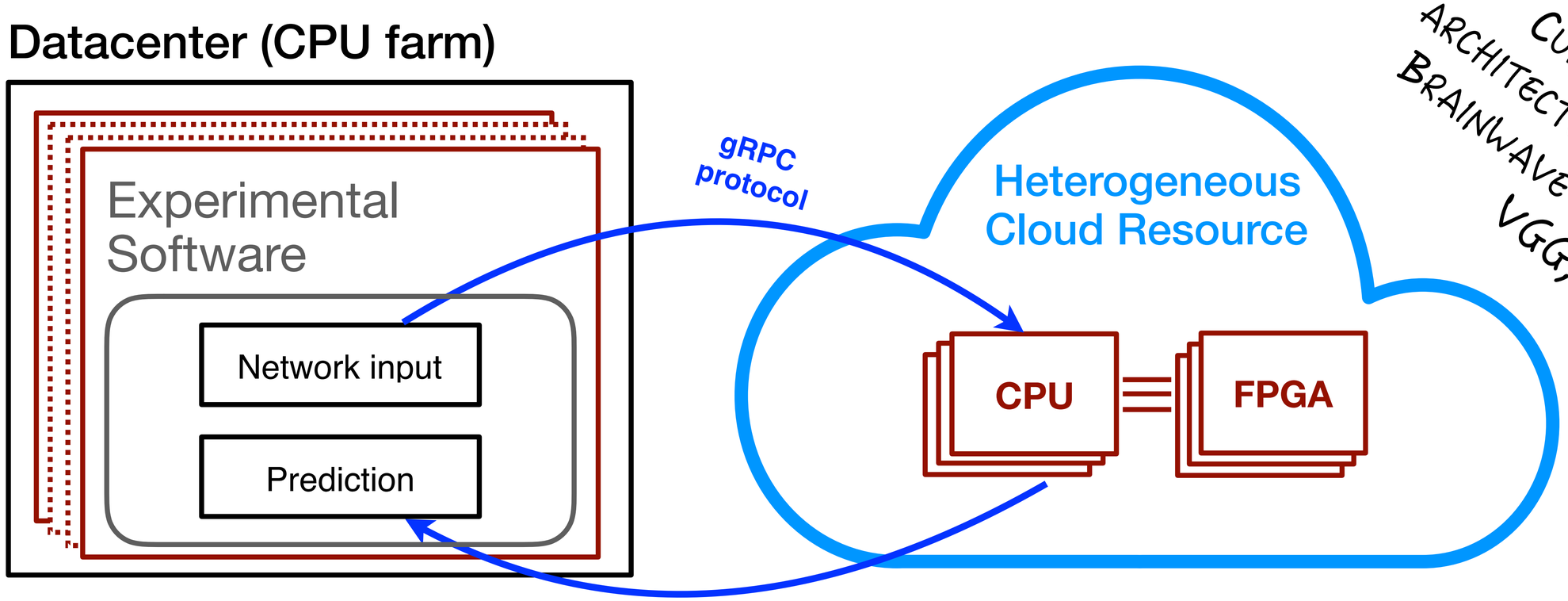
GPU (Local)

Nvidia GTX 1080 (batch 1)
10-20 ms

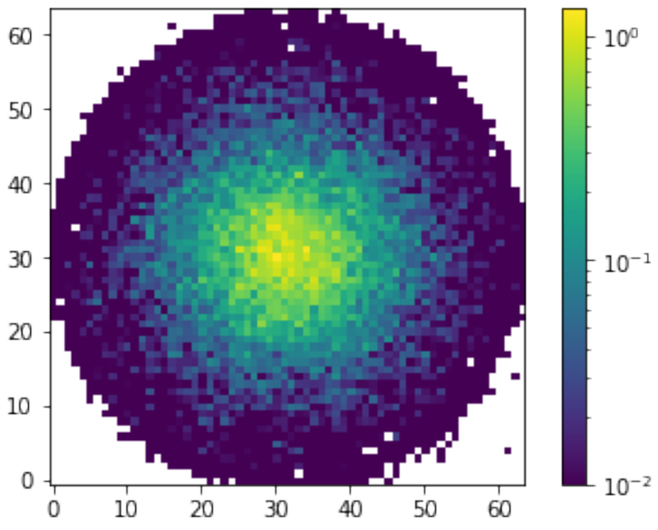
Nvidia GTX (batch 128)
few ms

10-100x speedup over CPU!

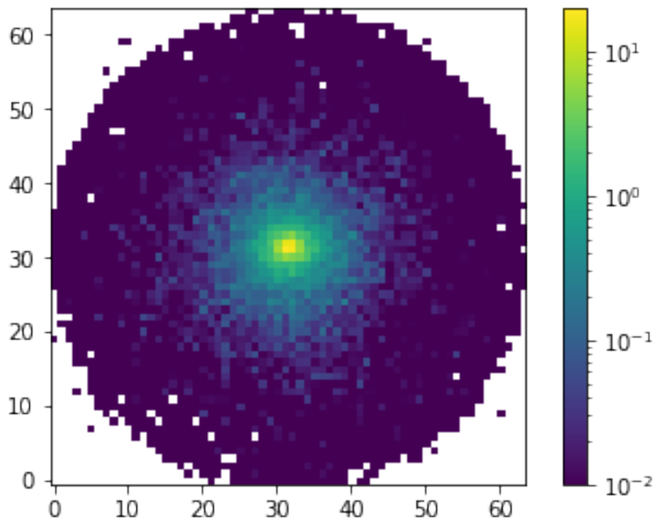
First tests with MS Brainwave



Top jets



QCD jets



Using public top tagging data, able to achieve state-of-the-art performance with ResNet50 architecture

Outlook

Co-processor architecture with machine learning an exciting solution to future computing challenges

SONIC:

Deploying coprocessors as a cloud (or edge) *service* is viable and non-disruptive solution for modern HEP computing paradigm

First demonstration with Microsoft Brainwave shows impressive latency and good performance

Exploring other industry platforms

hls4ml on AWS using Xilinx VU9P co-processors

Xilinx ML suite/DeePhi on AWS

Google TPUs, Intel Accelerator cards