Fast machine learning for physics, detectors, and computing

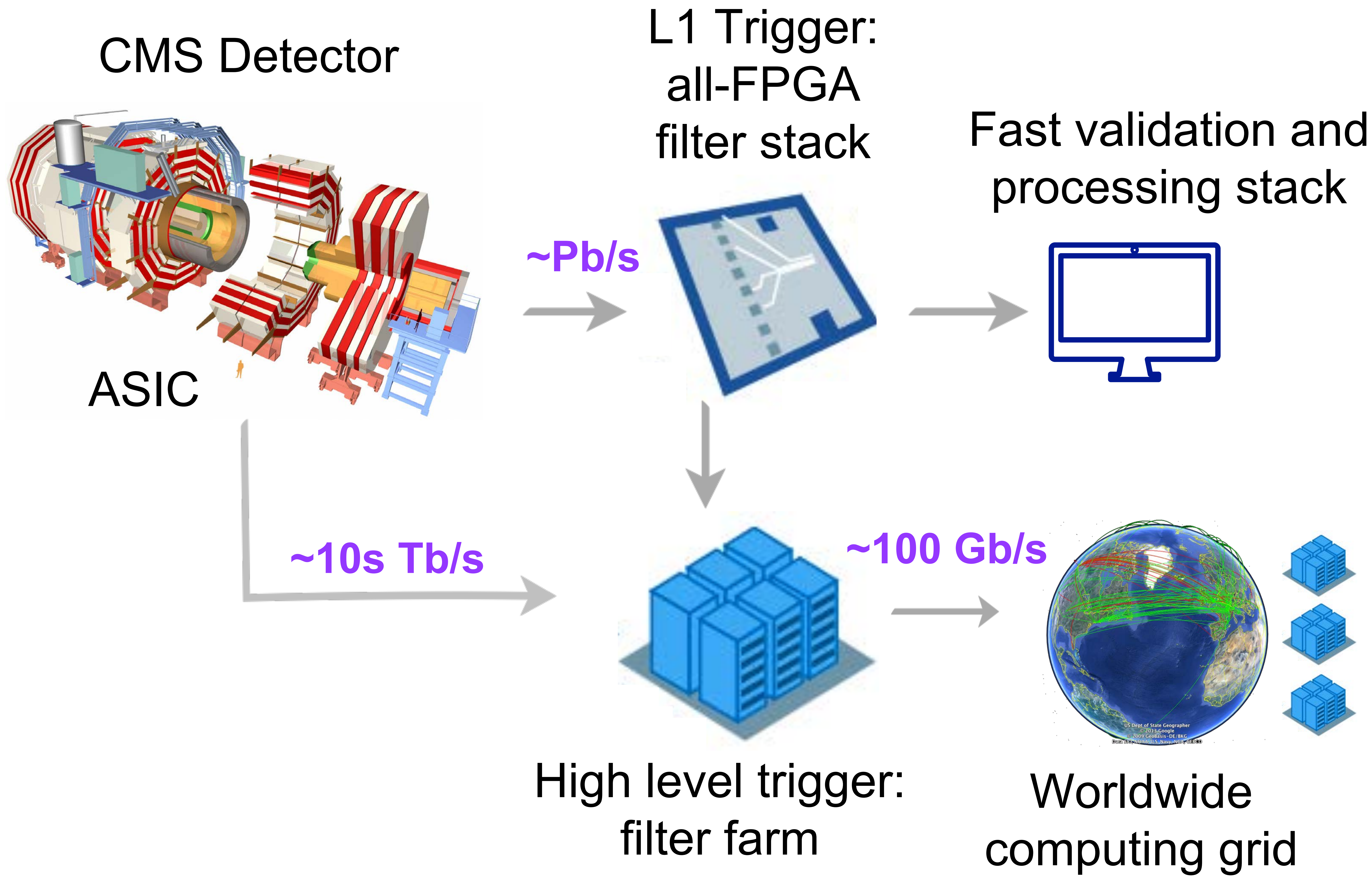
Nhan Tran
Fermilab/Northwestern
September 24, 2020

Outline

- Opportunities and challenges
  - Real-time and big data challenges in particle physics
  - Machine learning in physics in a nutshell
- Near sensor and on-detector ML
  - hls4ml and the LHC trigger
- Accelerated ML for HEP computing
  - SONIC for ProtoDUNE

Opportunities and Challenges

Physics and big data



DUNE upstream DAQ

~Pb/s

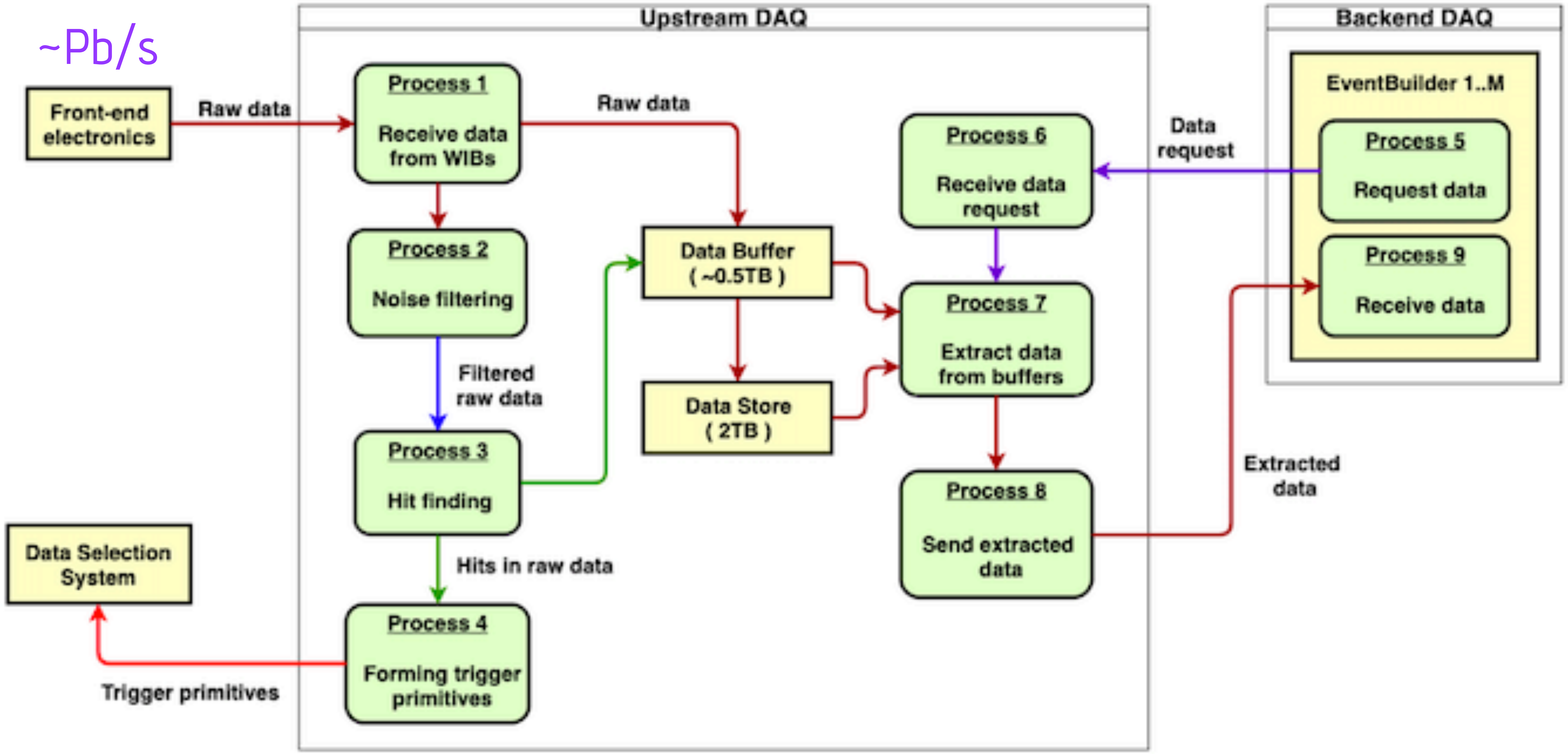
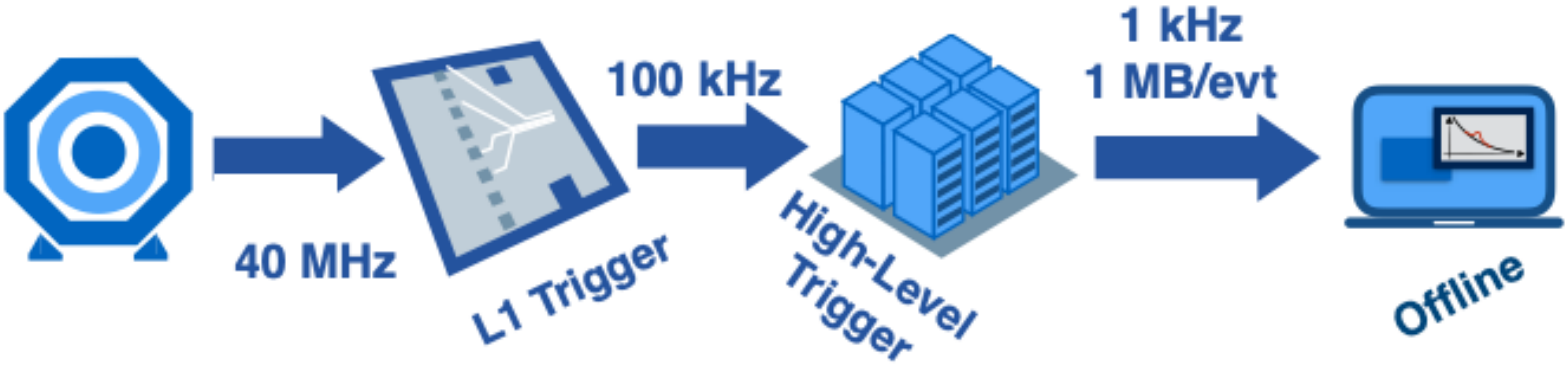


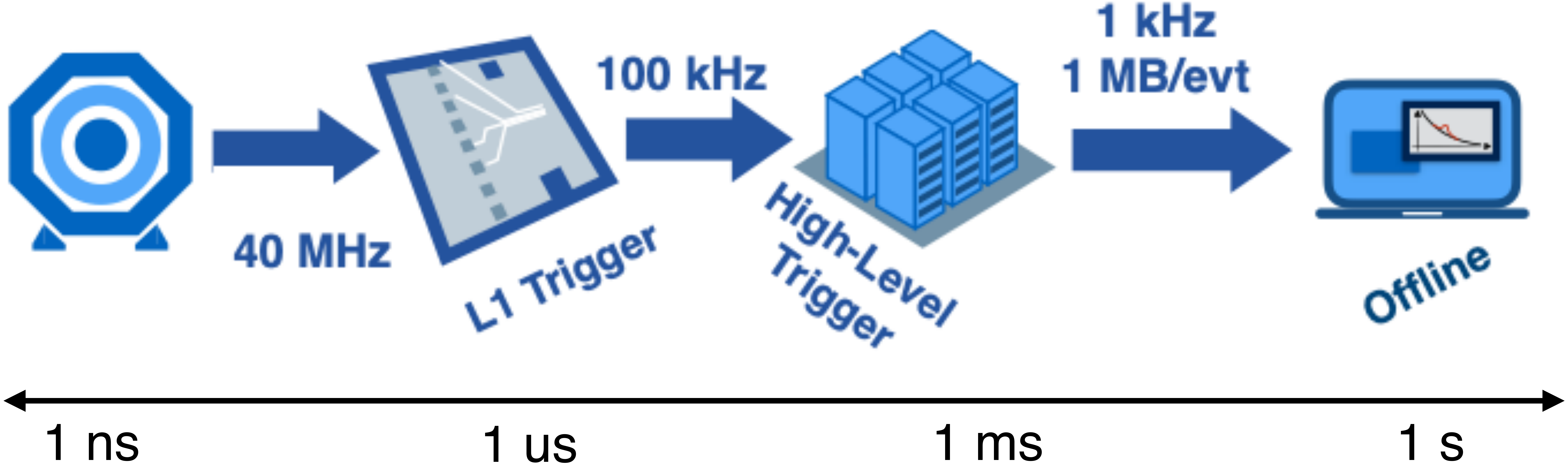
Figure 7.5: DUNE upstream DAQ subsystem functional blocks.

CMS real-time processing



> 99% of events are not saved for prompt offline analysis

CMS real-time processing



Custom electronics
Latency ~ 25ns - 1 μs

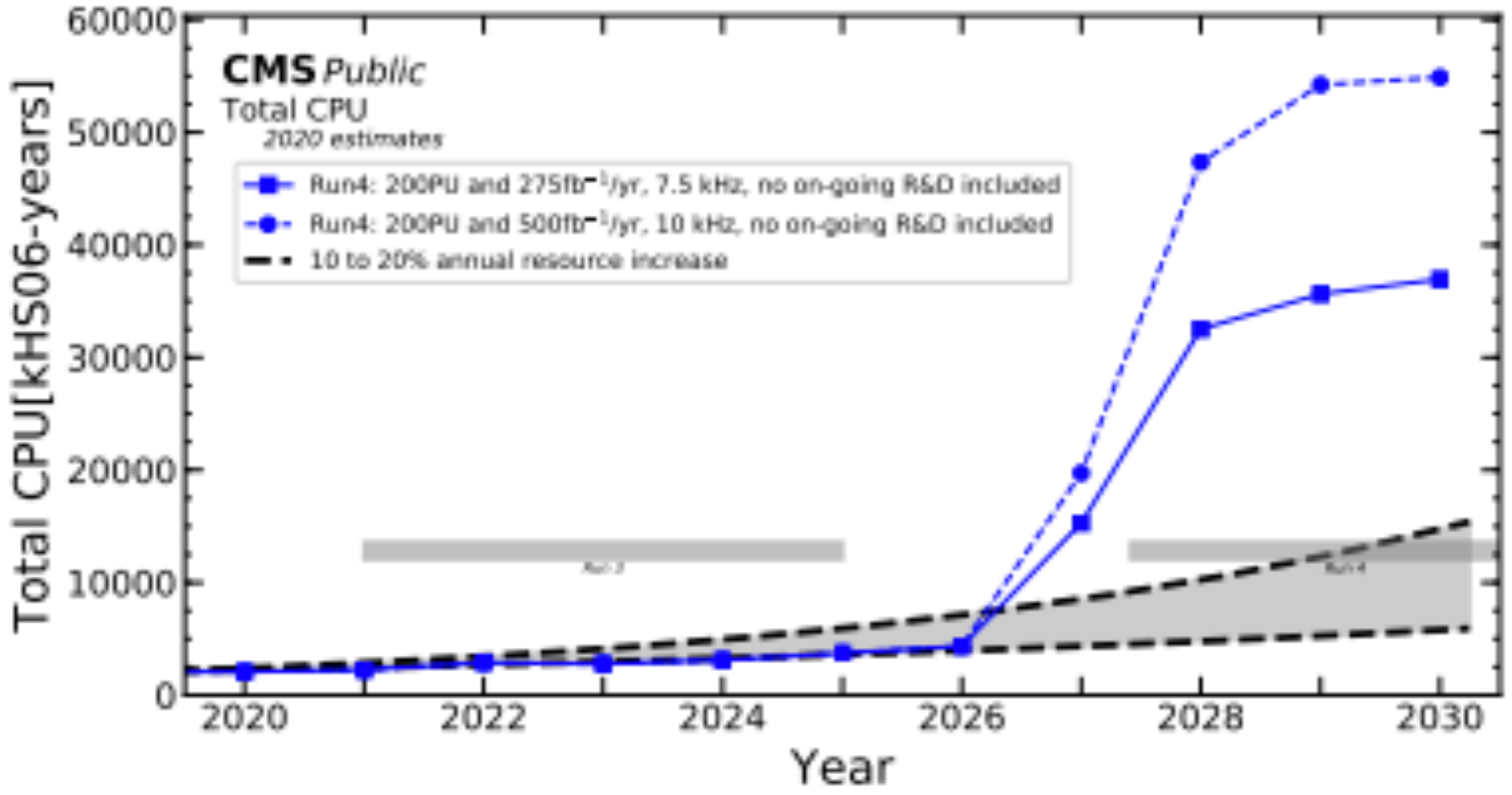
FPGAs/ASICs - high bandwidth low latency specialized compute hardware

Off-the-shelf computing
Latency ~ O(1+ ms)

"standard" CPU computing, coprocessors

The computing conundrum

CMS offline computing profile projection

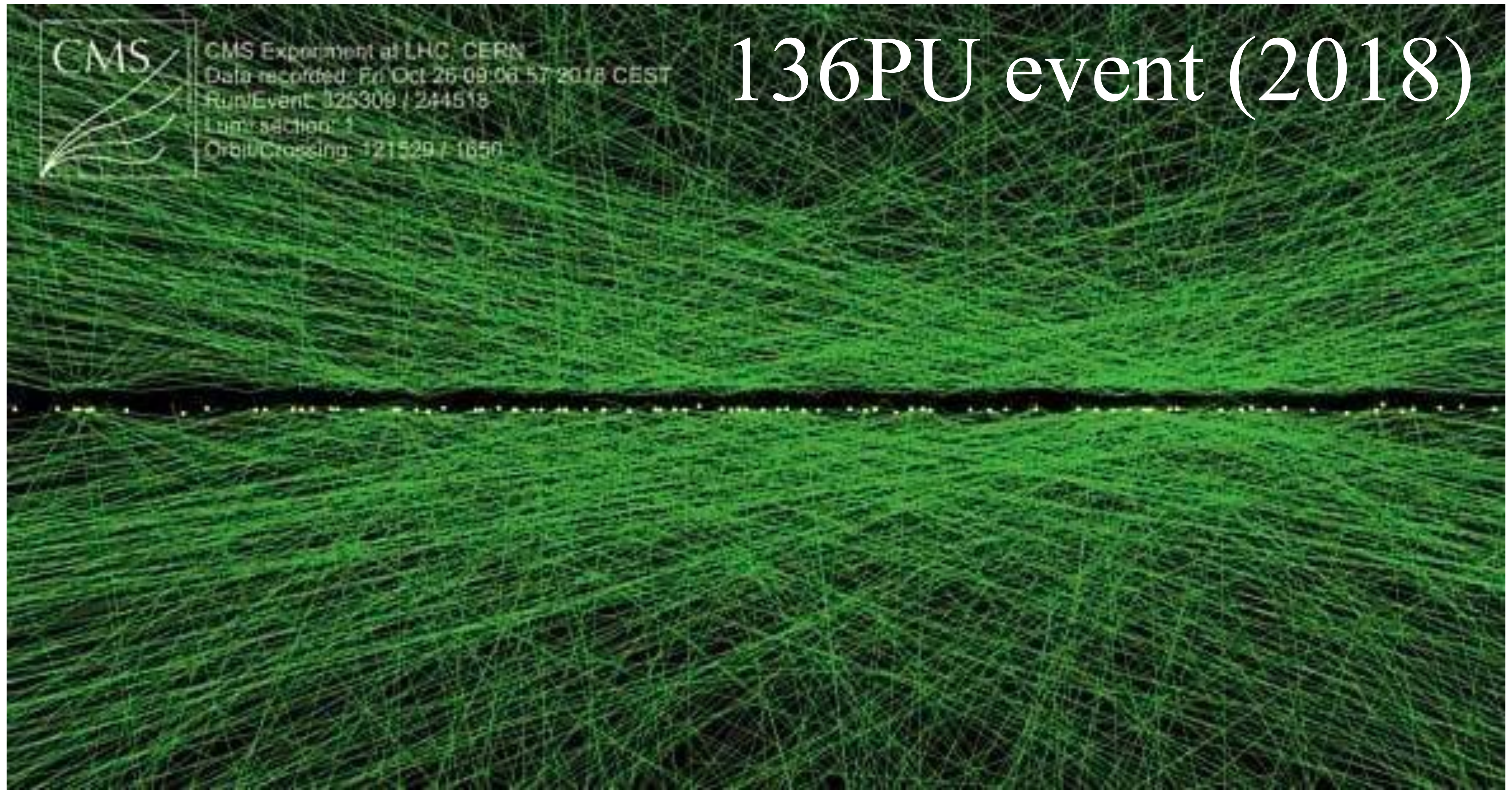


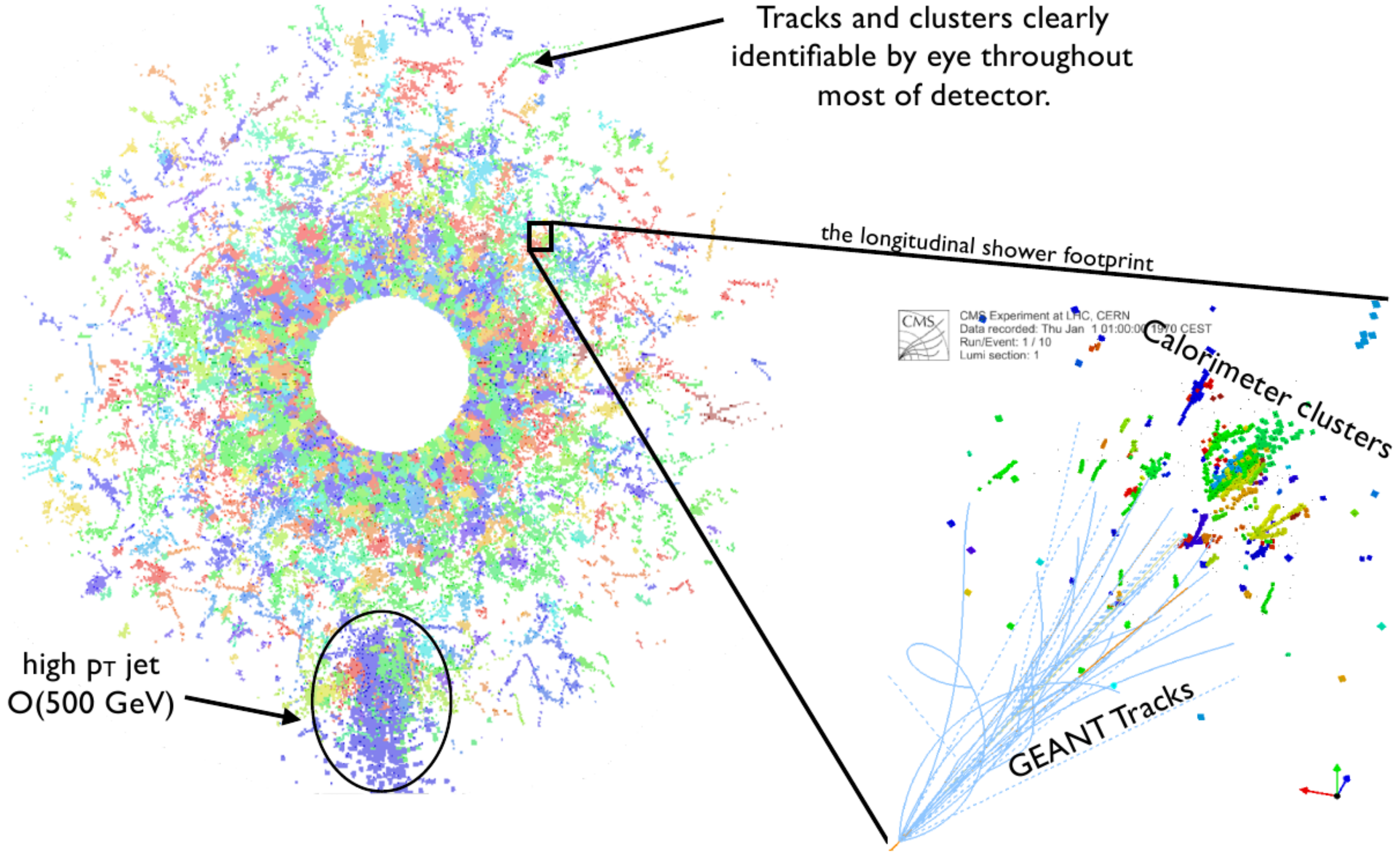
CMS online filter farm project

CMS detector	LHC (current)	HL-LHC (upgraded)
Simultaneous interactions	60	200
L1 accept rate	100 kHz	750 kHz
HLT accept rate	1 kHz	7.5 kHz
Event size	2.0 MB	7.4 MB
HLT computing power	0.5 MHS06	9.2 MHS06
Storage throughput	2.5 GB/s	61 GB/s
Event network throughput	1.6 Tb/s	44 Tb/s

Compute needs growing by up to 10x
Environments getting more complex
Need more sophisticated analysis techniques

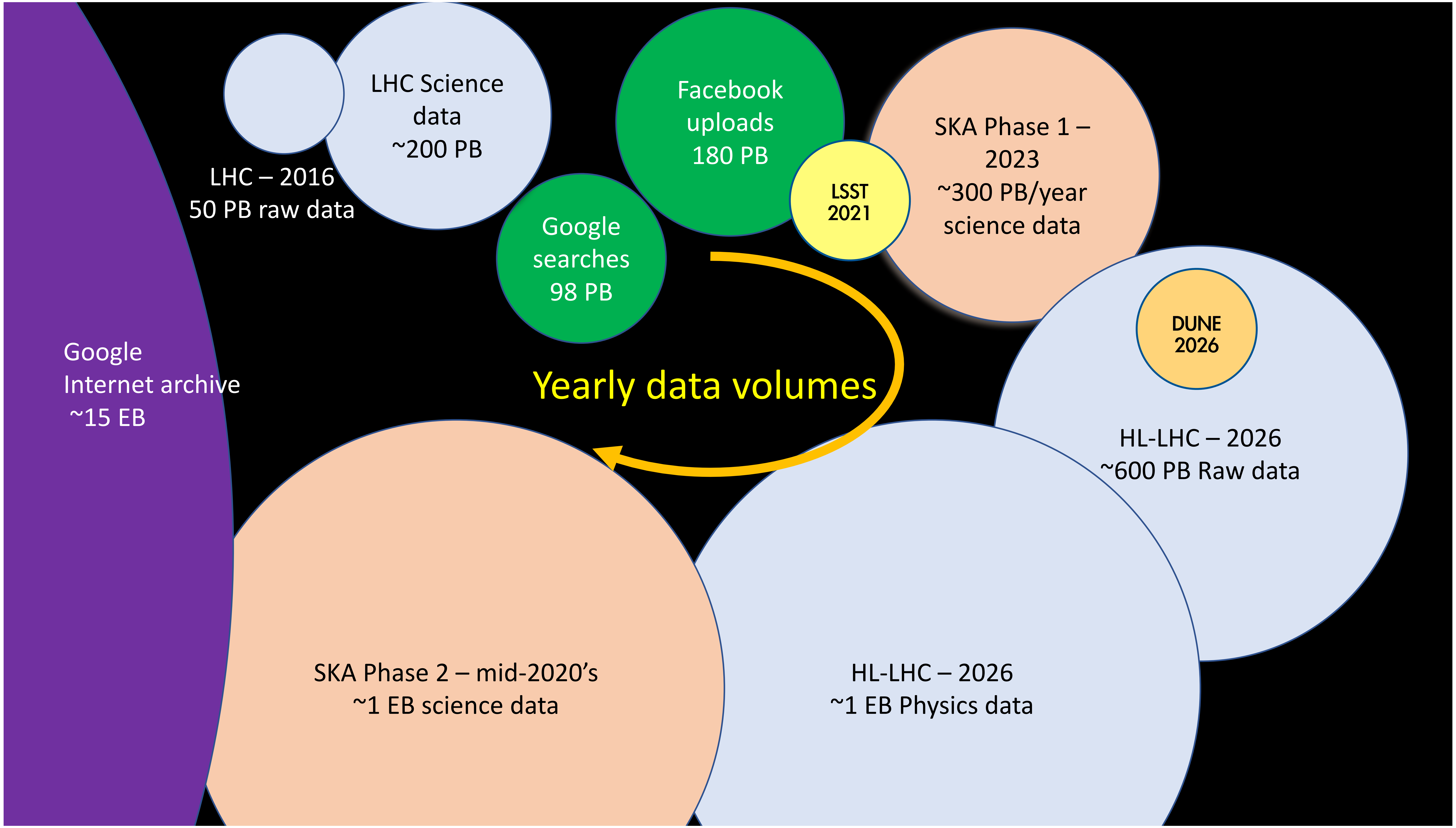
The computing conundrum



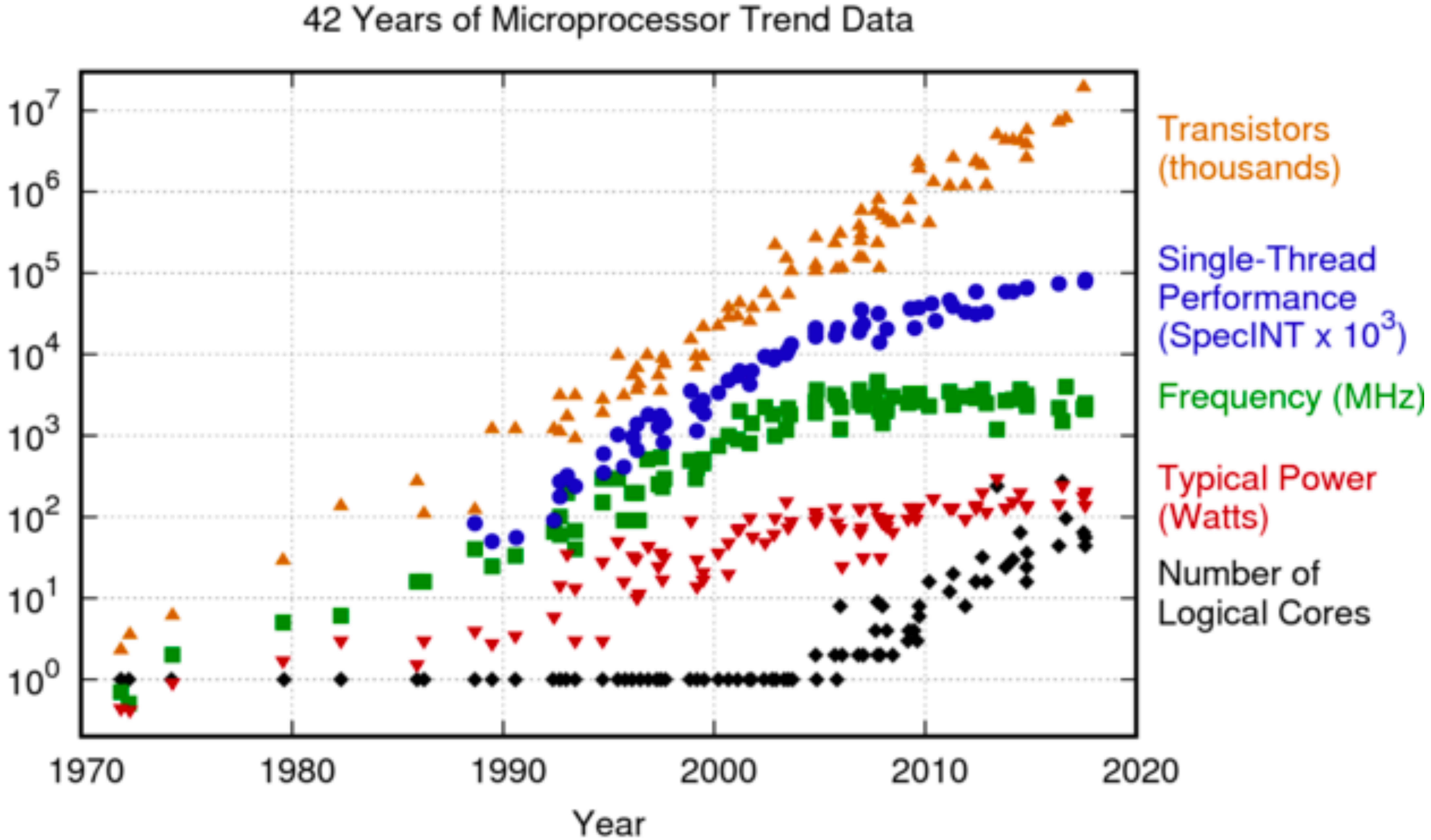


Compute needs growing by more than 10x
Environments getting more complex
Need more sophisticated analysis techniques

The computing conundrum

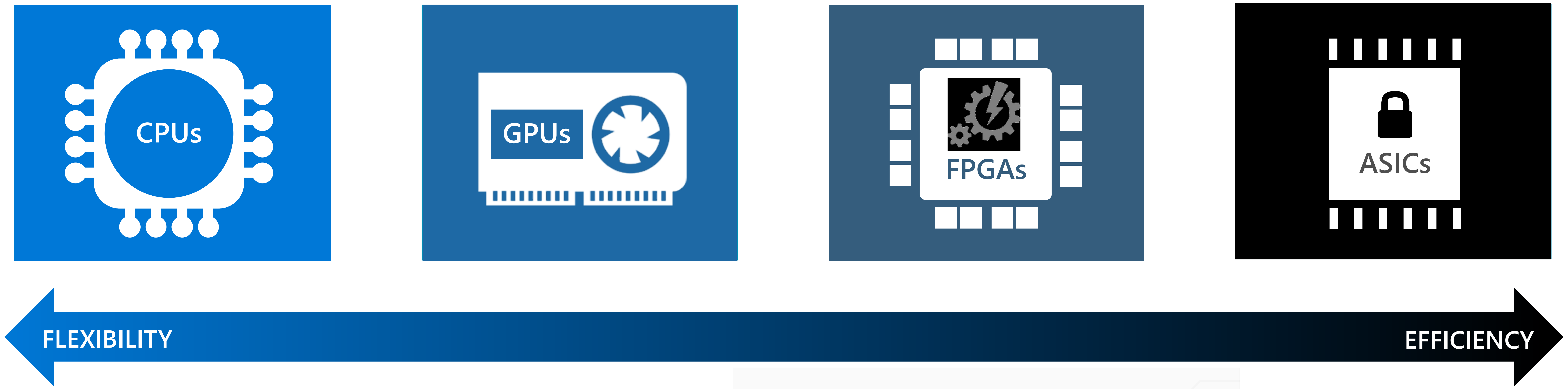


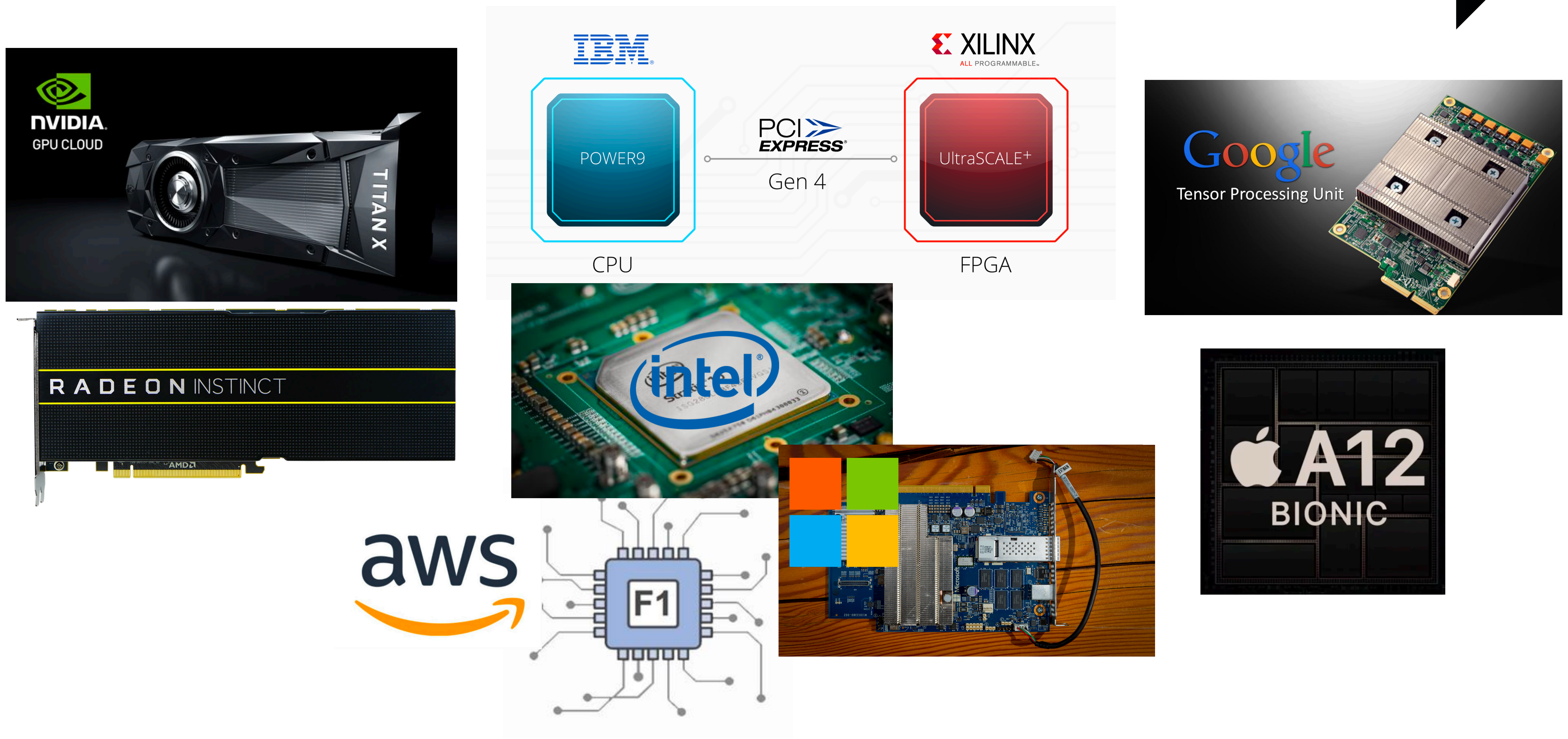
The computing conundrum



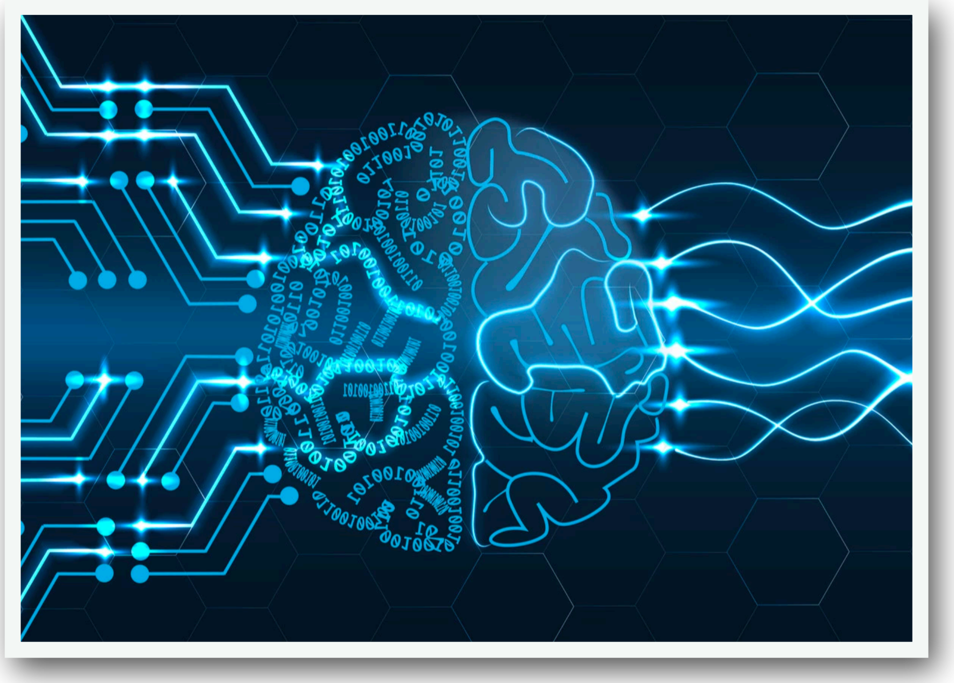
Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

Heterogeneous compute

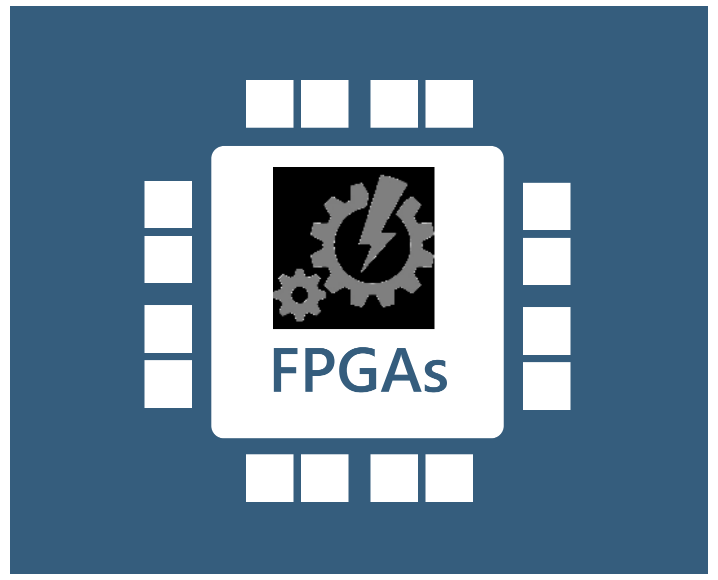


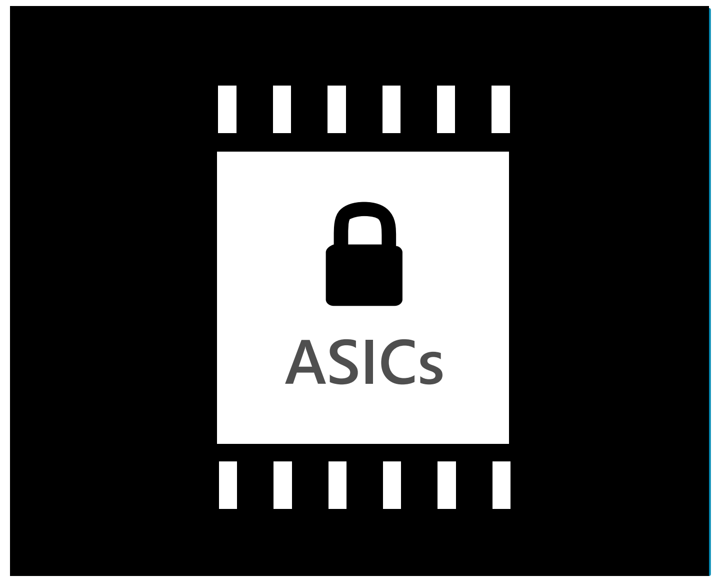


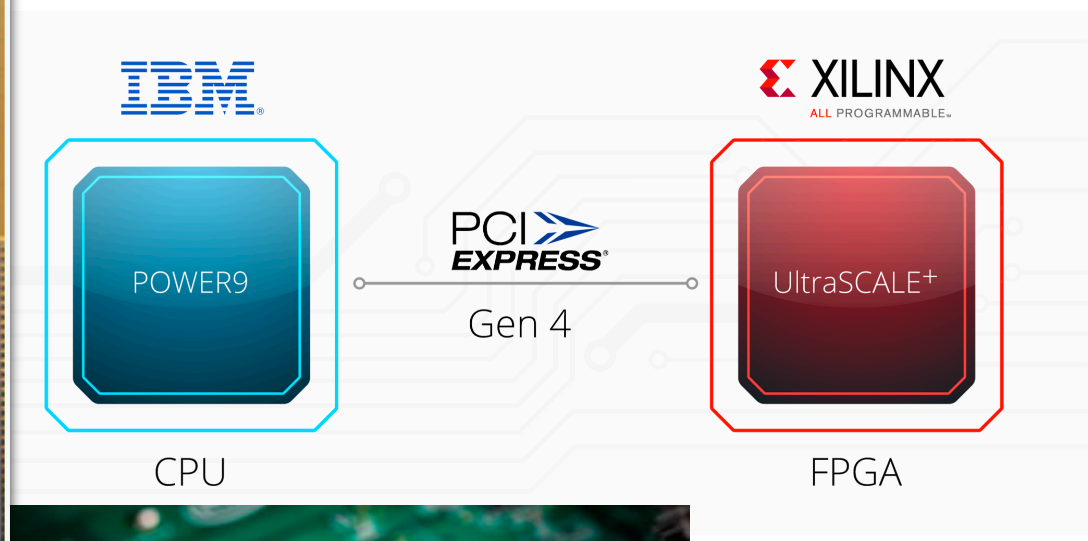
Advances in heterogeneous computing driven by machine learning

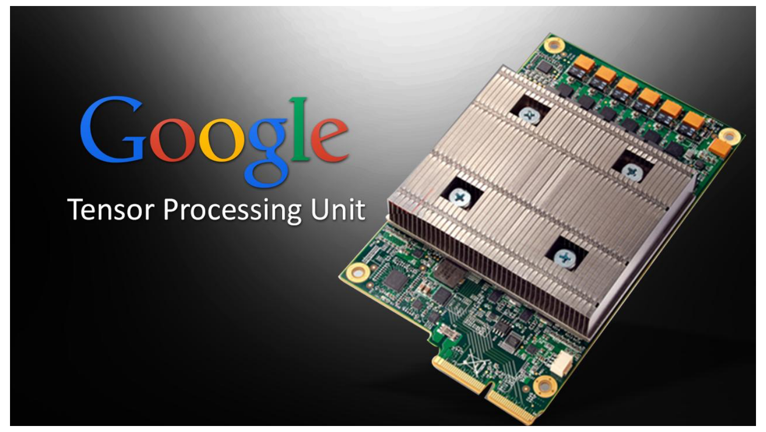


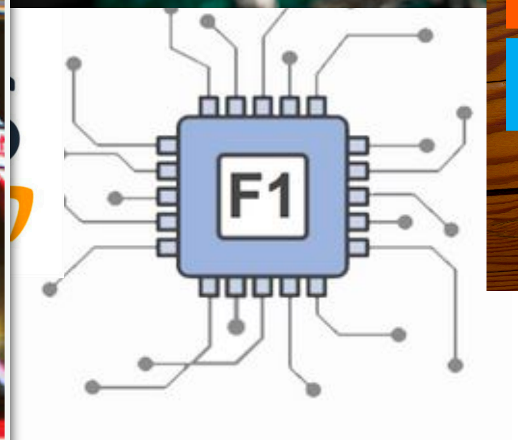
Heterogeneous compute

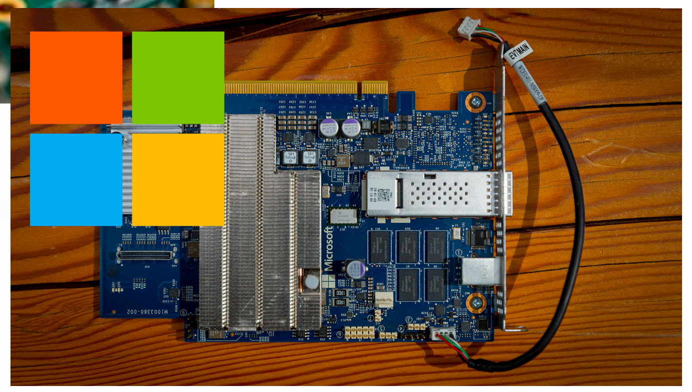




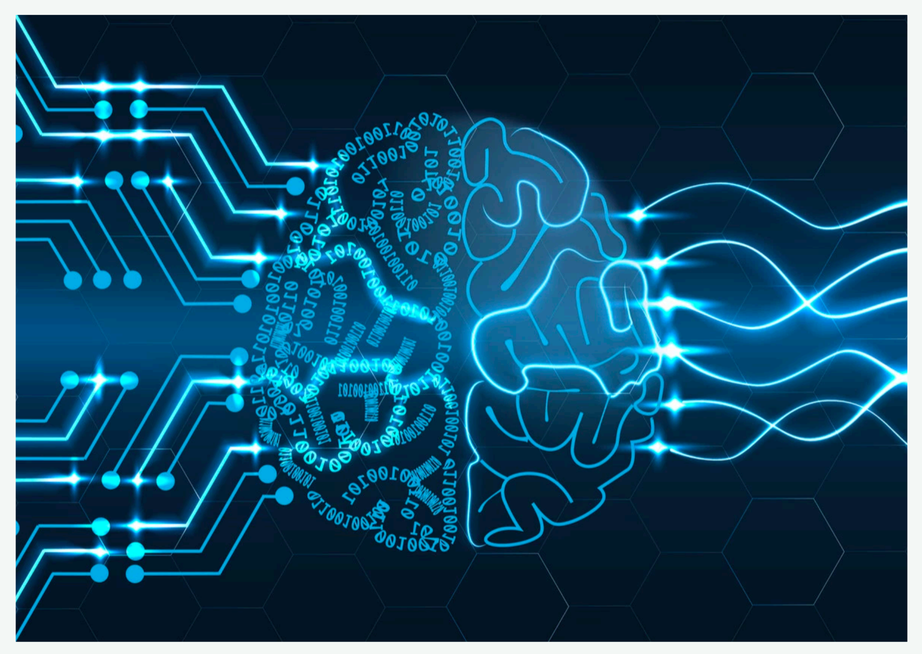








Advances in heterogeneous computing driven by machine learning

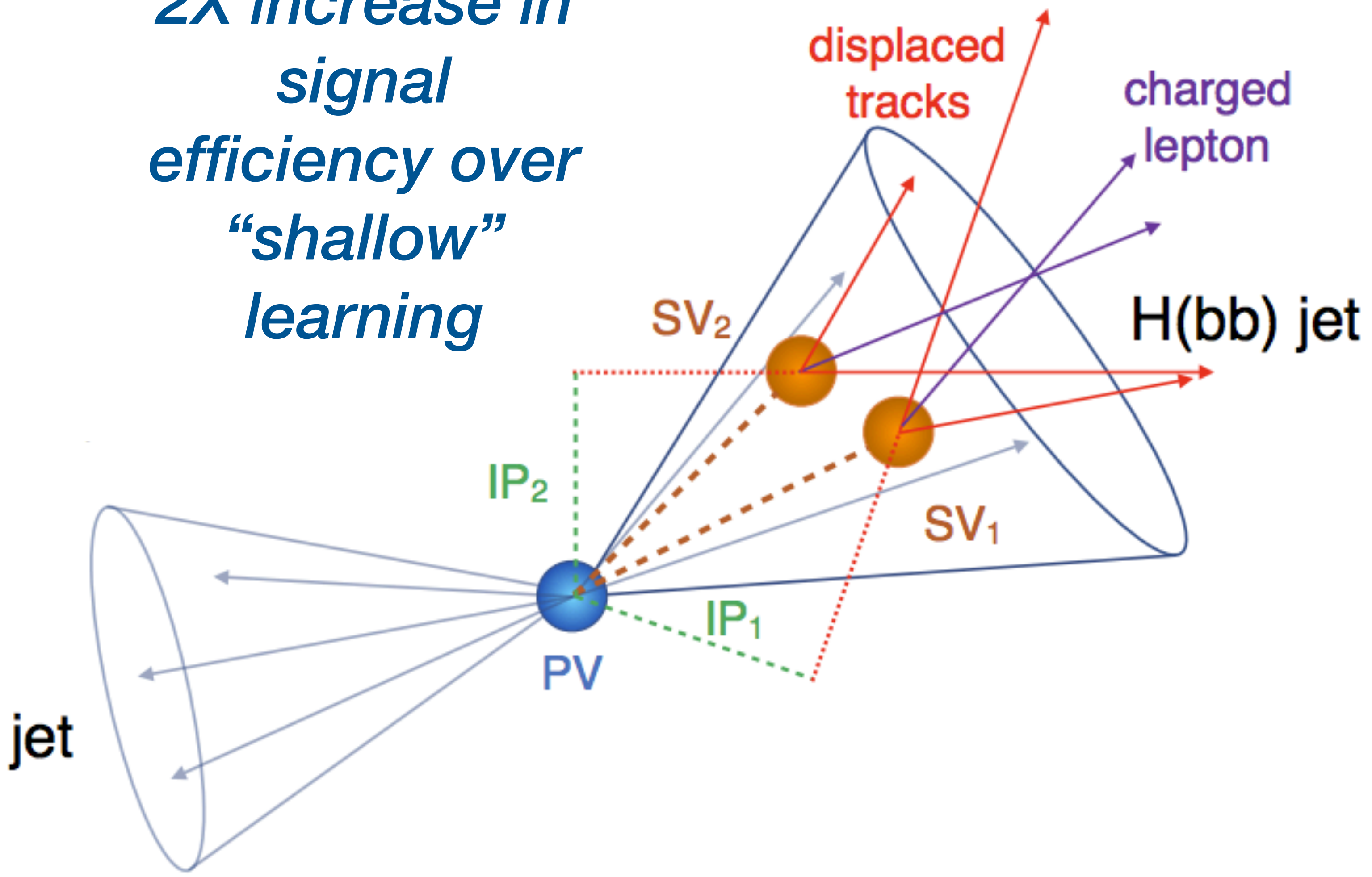


Machine learning

Energy

Identification of boosted Higgs jet decay to two bottom quarks

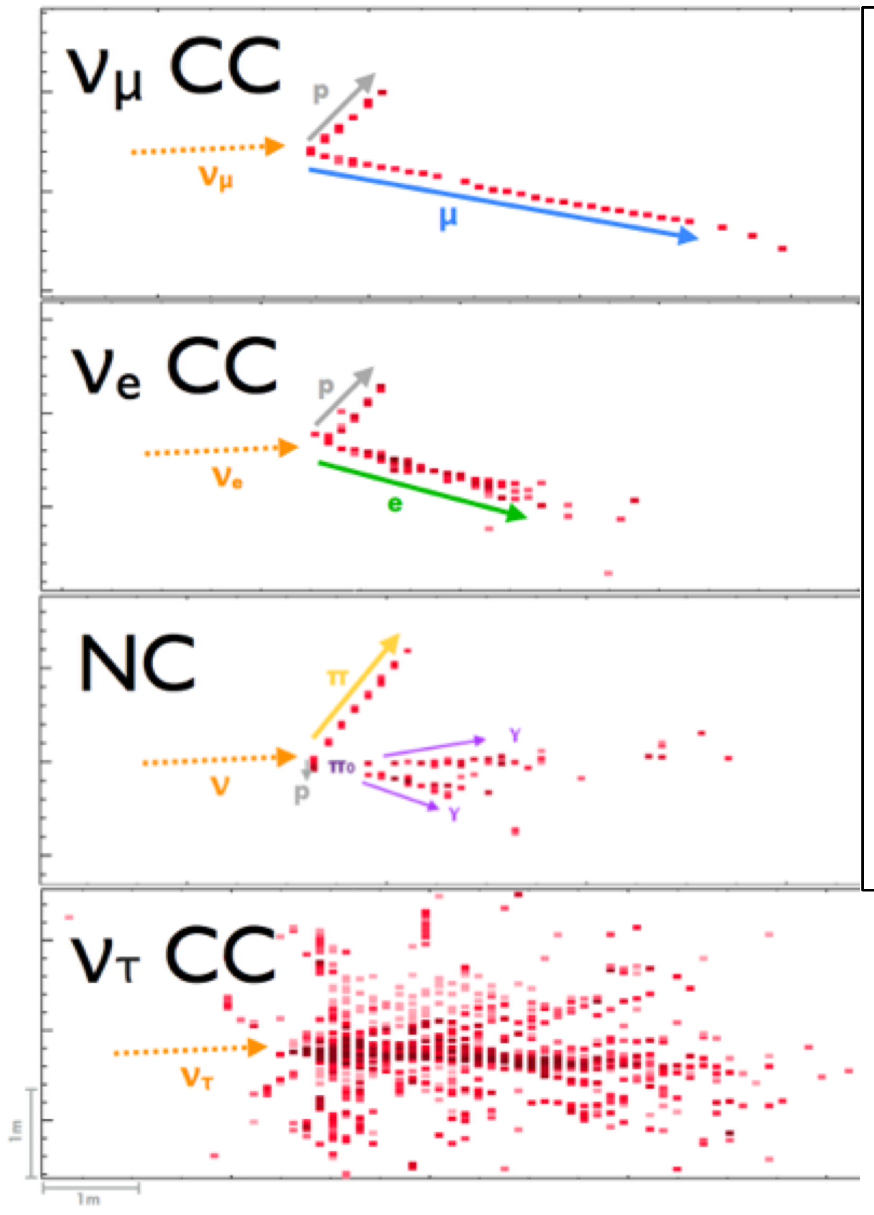
2X increase in signal efficiency over "shallow" learning



J. Duarte et al., CMS DP-2018/046

Intensity

NOvA event classification



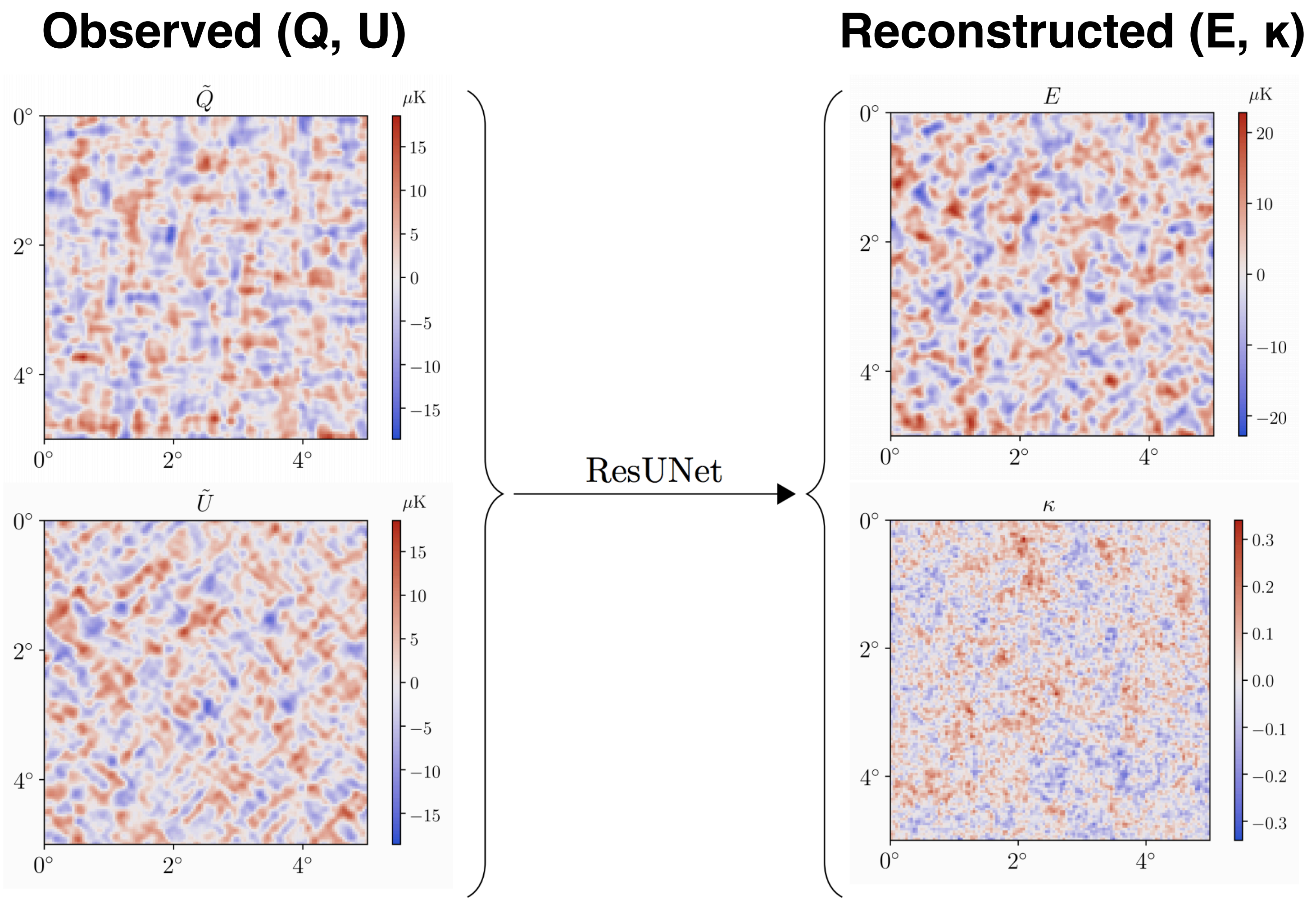
Deep learning improvement:

Effective 40% increase in NOvA active volume

A. Himmel, E. Niner, F. Pshihas et al.
https://arxiv.org/abs/1604.01444
1st deployed in oscillation analysis
https://arxiv.org/abs/1703.03328

Cosmic

Reconstruction of CMB polarization map from Stokes parameters



50% less noise vs. traditional methods across large range of scales

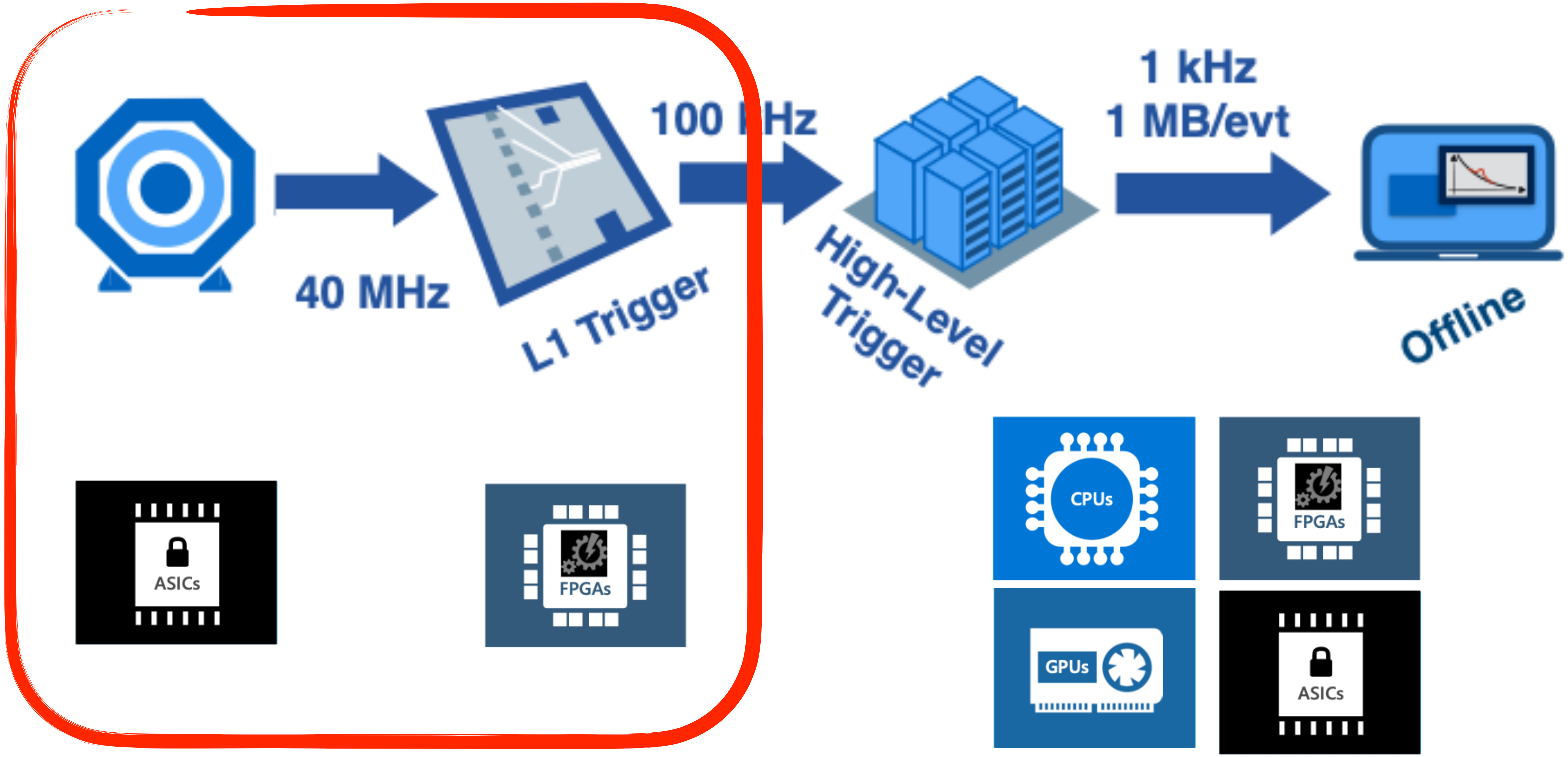
J. Caldeira, B. Nord, et al.,
https://arxiv.org/abs/1810.01483

Machine learning

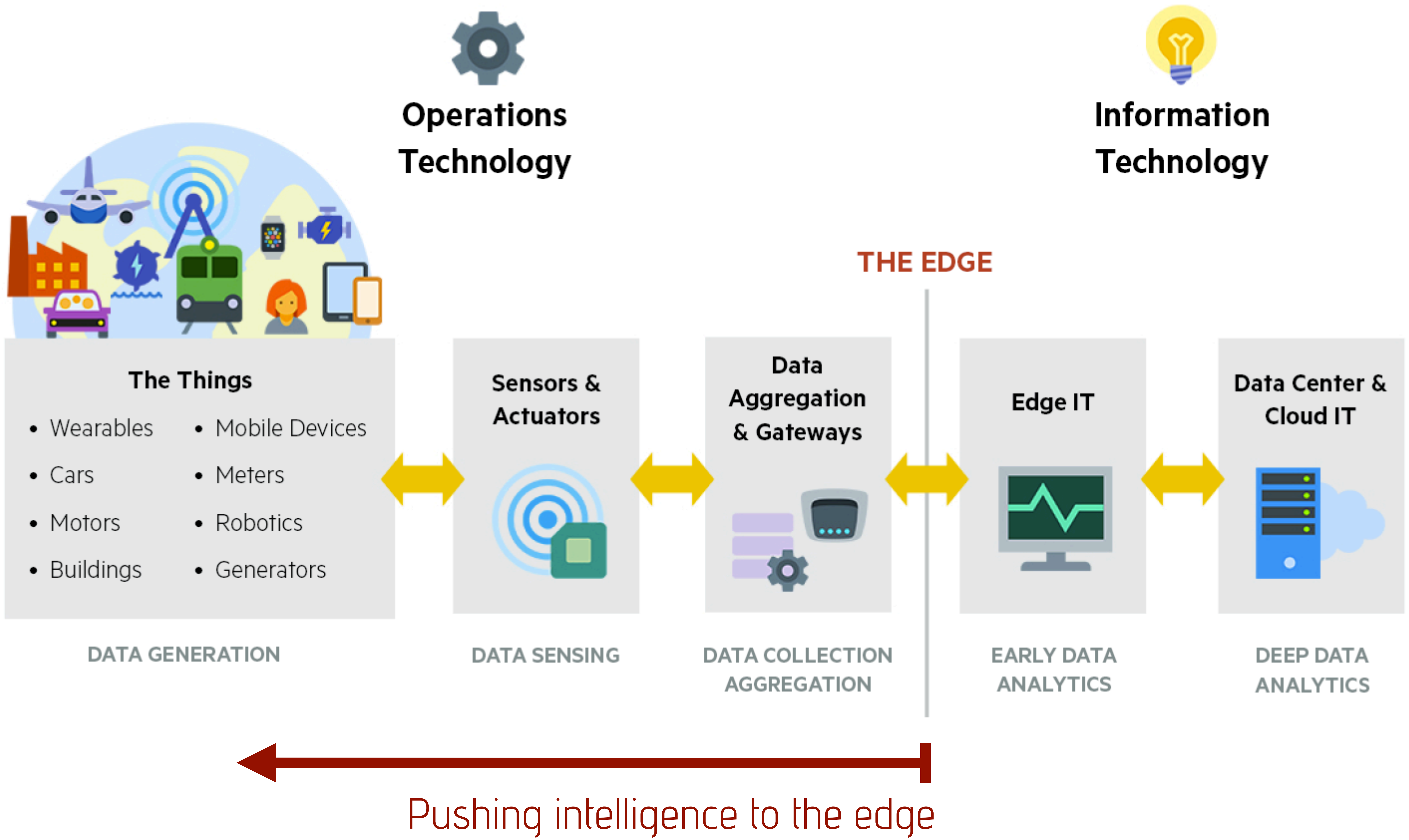
- We are just scratching the surface of AI applications in physics
  - Thus far most “standard” neural network architectures and **supervised learning** are in operation (taggers, reconstruction, regression,...)

- Particle physics has interesting and rich data based on the principles of physics and very challenging big data applications
  - **Physics for AI:** Learning on point clouds, physics-inspired neural networks, unsupervised techniques (clustering, anomaly detection) in real data, real-time efficient algorithms, ...
  - **AI for physics:** Across the entire scientific process from operations to algorithms to detectors to computing

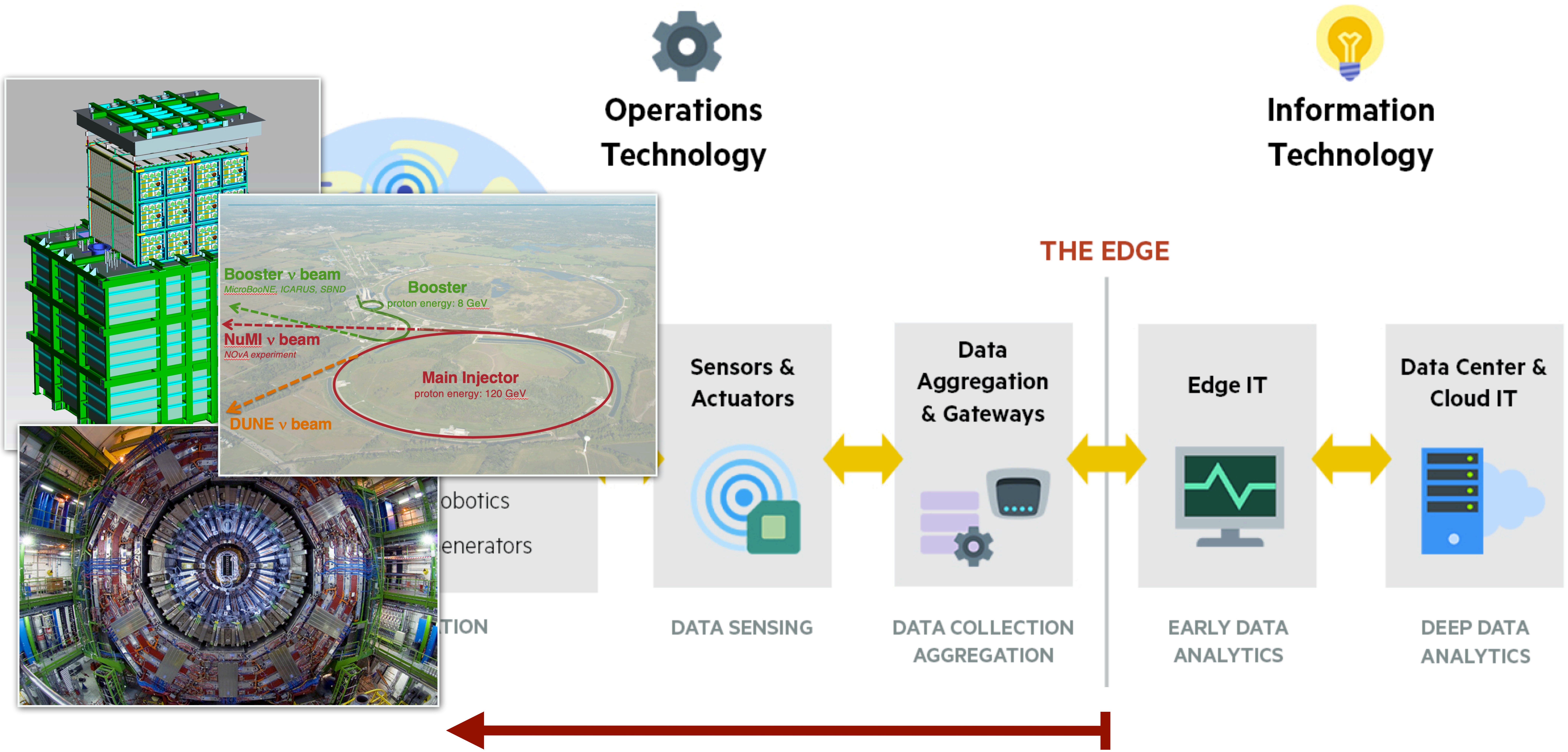
Near sensor and on-detector ML



Internet of things…particle physics



Internet of things...particle physics



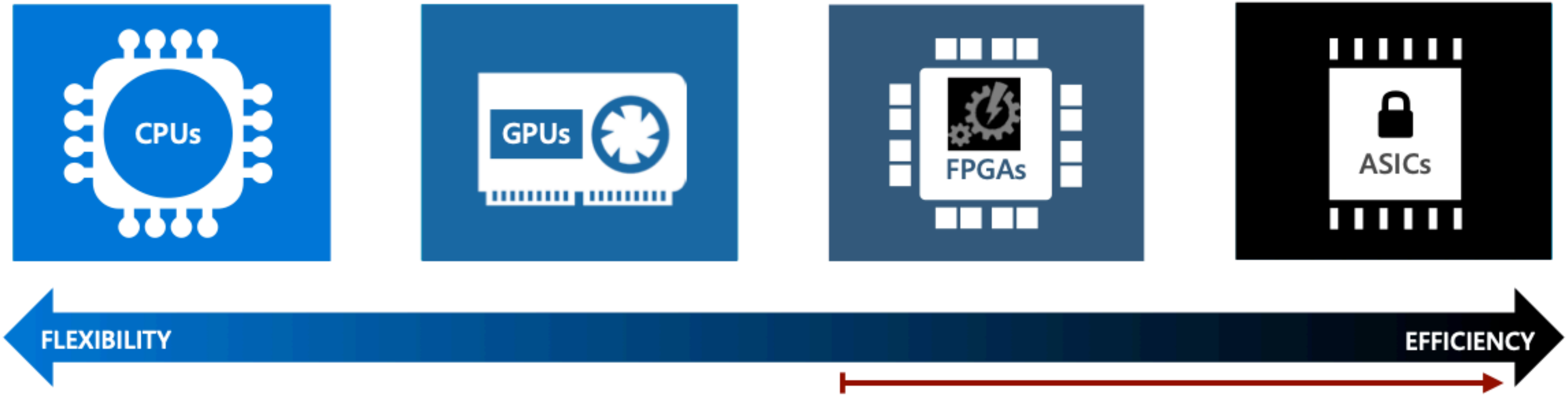
Pushing intelligence to the edge

Processing hardware

- Power hungry
- Batching for optimal performance
- Mature software ecosystem

- Middle solution, flexible and less power hungry than GPU
- Does not require batching

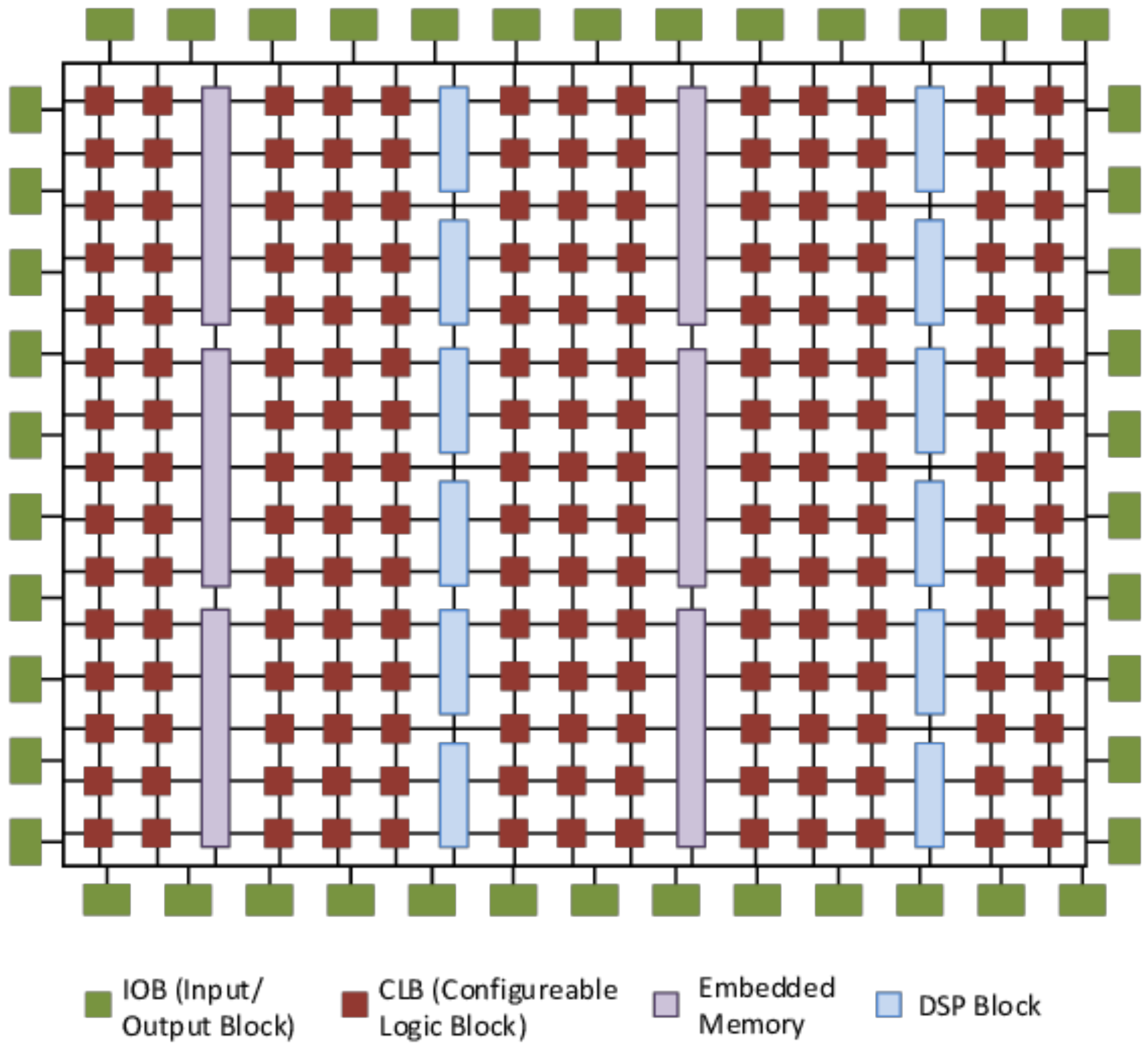
- Most efficient Op/W
- Less flexible



Rough guidelines:
> 100 Gbps throughput
< 1ms computational latency
< 10W power budget

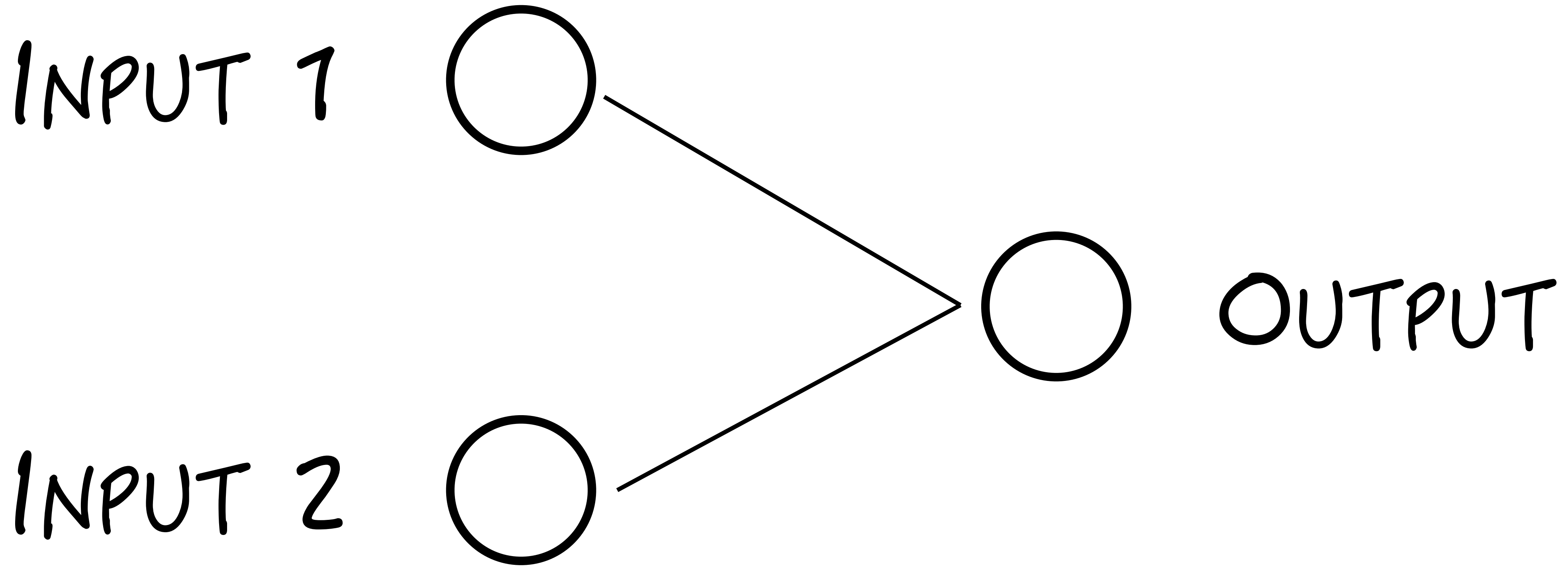
Digital circuit design

FPGA
"programmable hardware"



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

NN inference in a nutshell

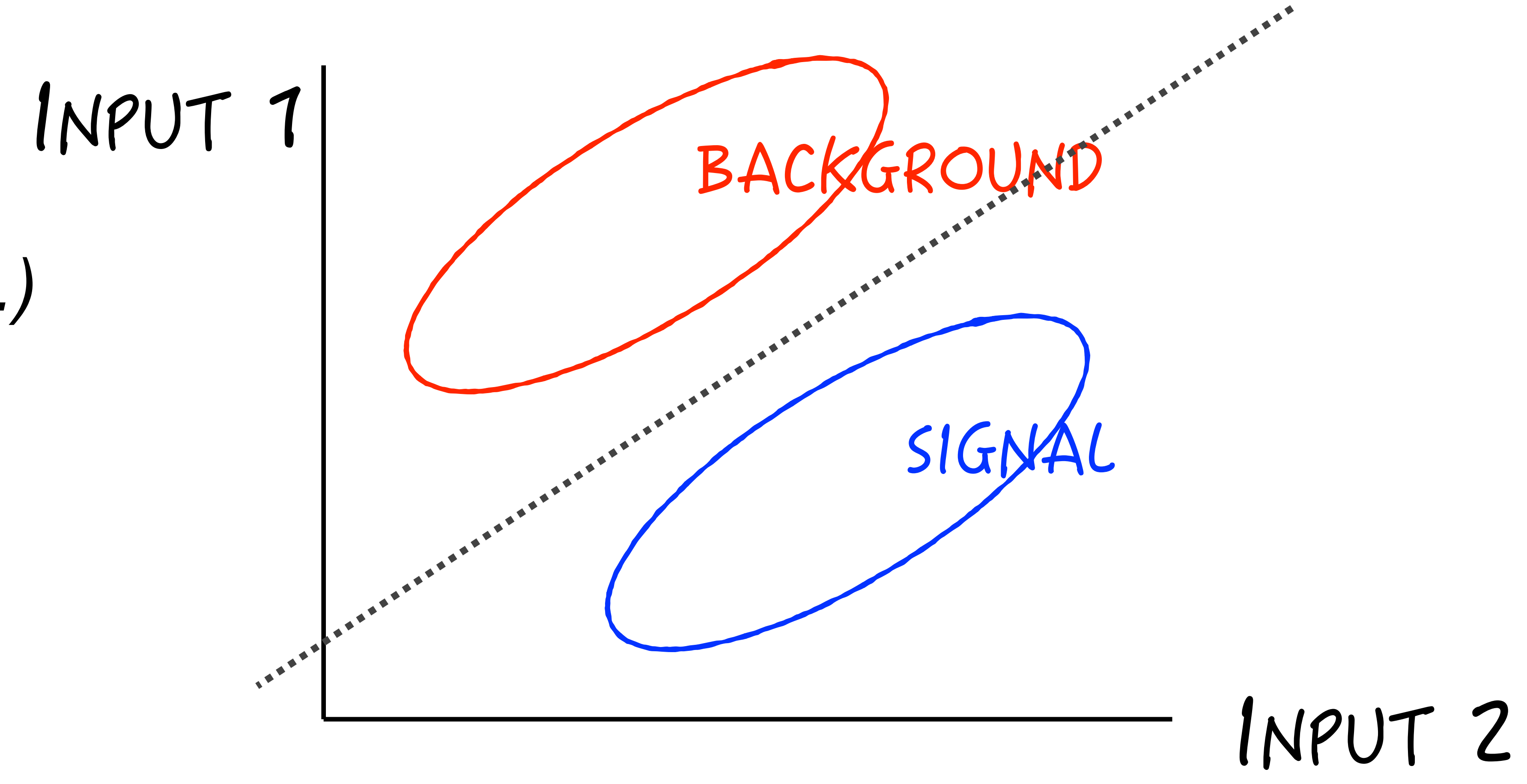


$$\vec{O}_j = \vec{I}_i \times \overleftrightarrow{W}_{ij} + \vec{b}_j$$

Simple 2 input example

(Fisher linear discriminant, linear support vector machine,…)

$$O_1 = I_1 \times W_{11} + I_2 \times W_{21} + b_1$$

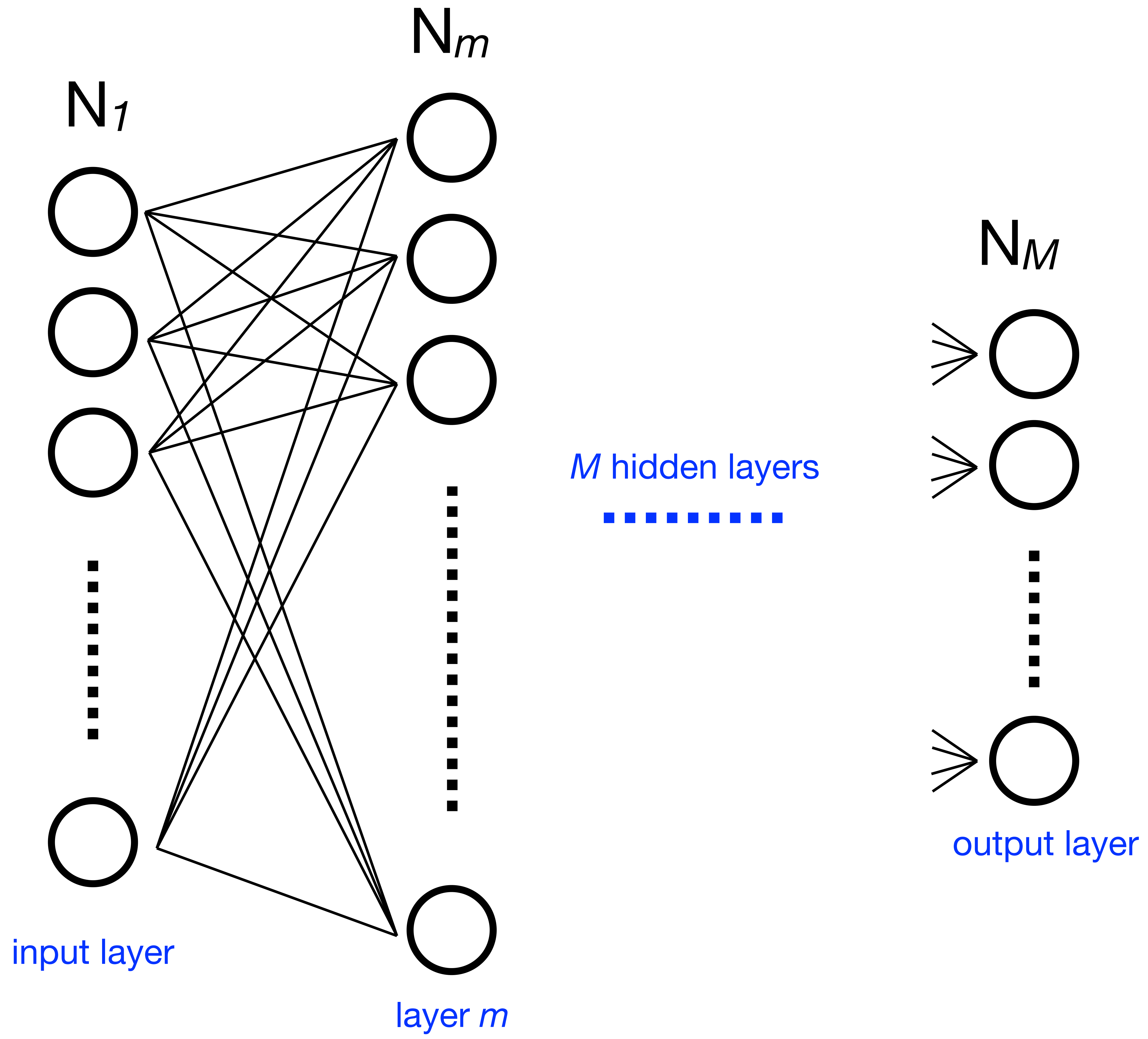


NN inference in a nutshell

$$\vec{O}_j = \Phi(\vec{I}_i \times \overleftrightarrow{W}_{ij} + \vec{b}_j)$$

Φ = ACTIVATION FUNCTION
(NON-LINEARITY)

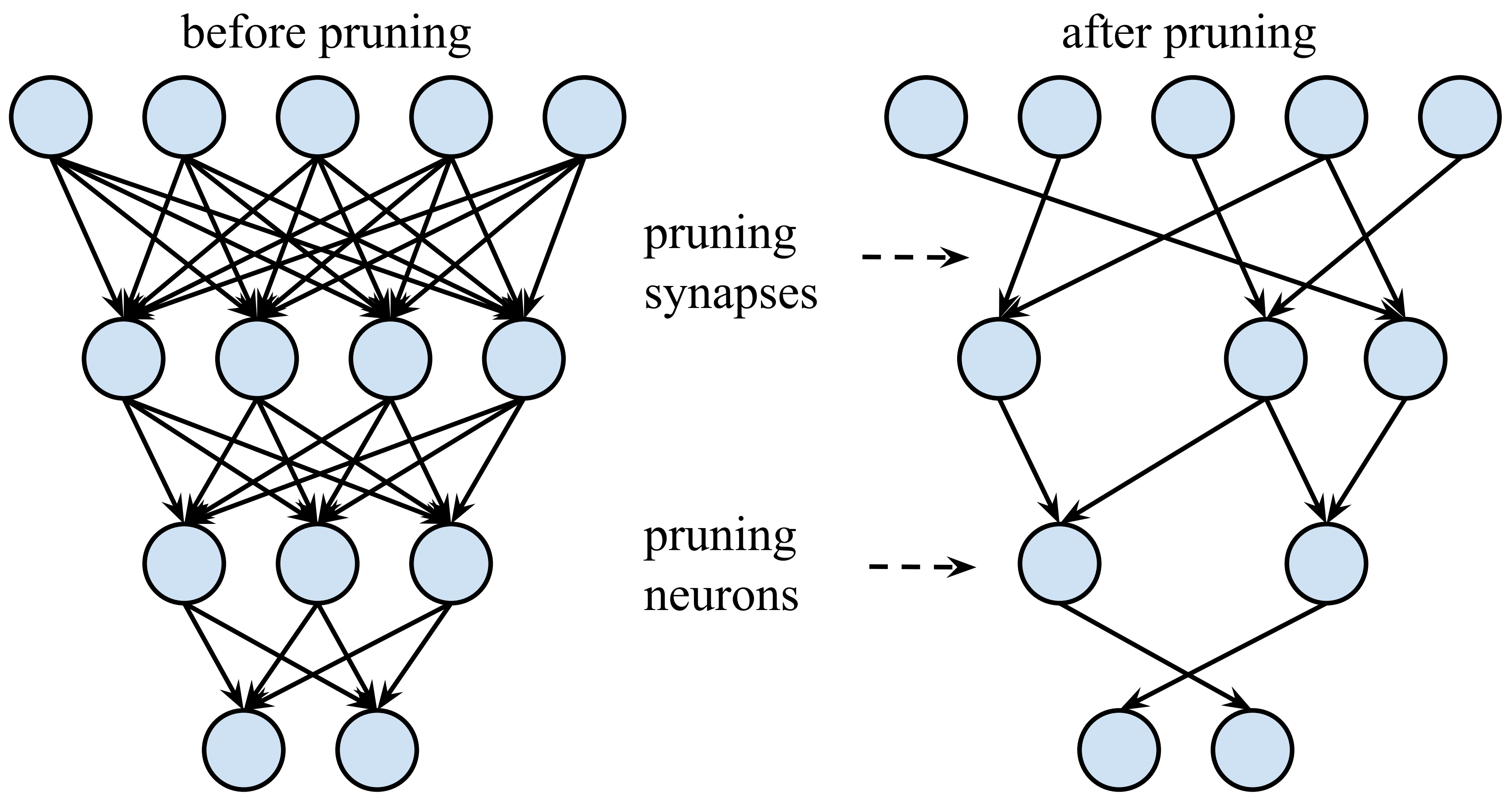
NN inference =
a bunch of multiplications /additions and LUTs (look up tables) for activation functions



FULLY CONNECTED HIDDEN LAYER

(Energy) Efficient Neural Networks

- Emergent engineering field, efficient implementation of NN architecture
- **Parallelization**: performing operations simultaneously (see next page)
- **Compression/Pruning**:
  - maintain the same performance while removing low weight synapses and neurons (many schemes)

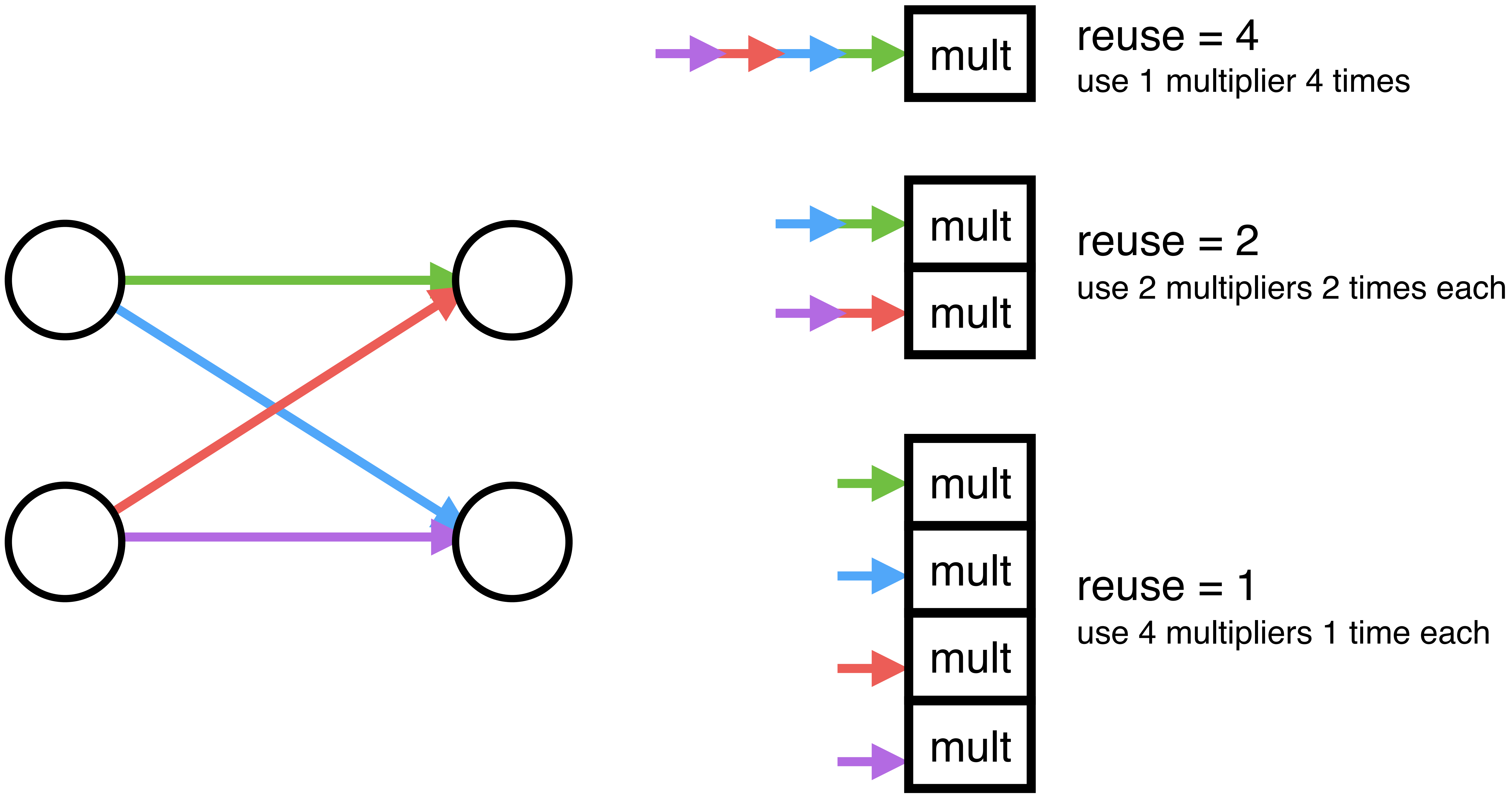


- **Quantization/Approximate math**:
  - 32-bit floating point math is overkill
  - 20-bit, 18-bit, ...? fixed point, integers? binarized NNs?

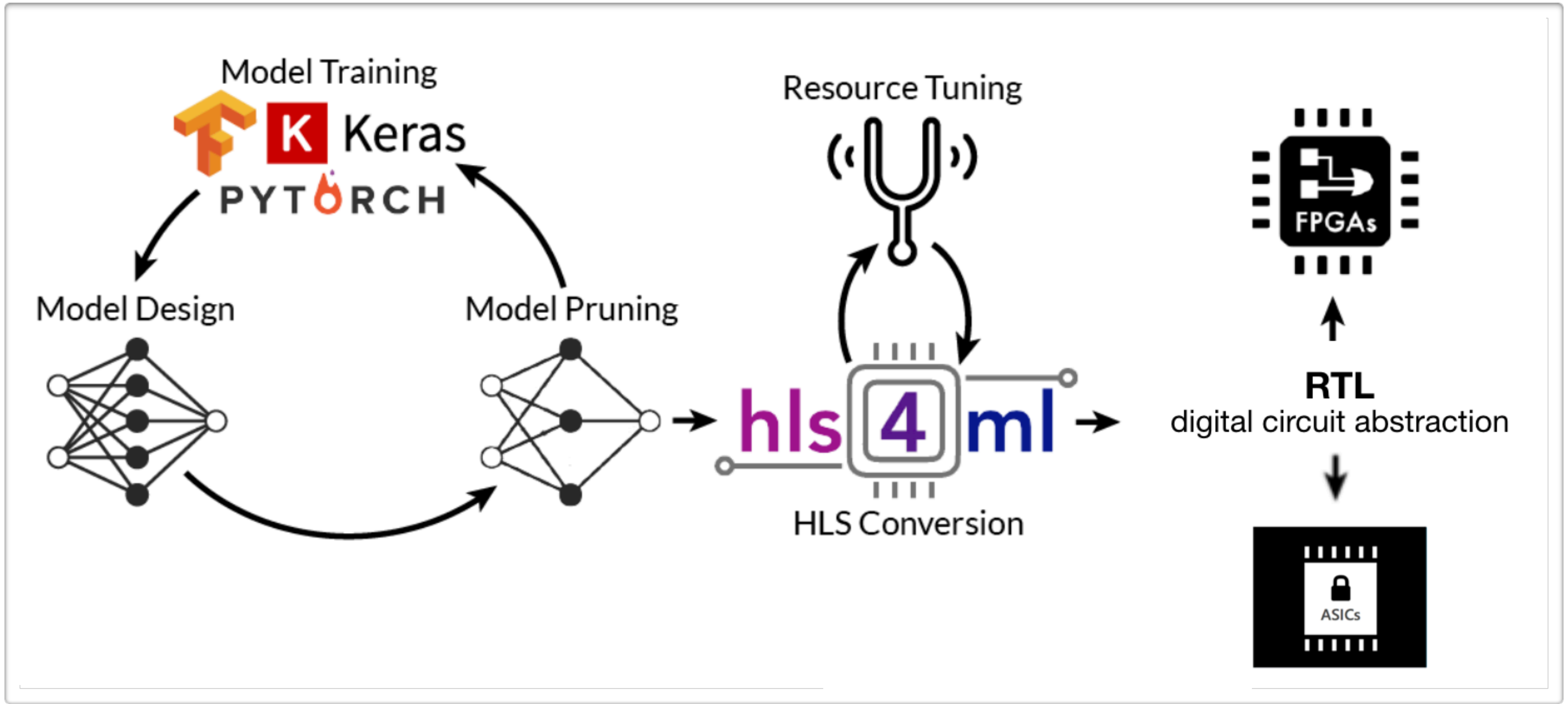
For further reading, start here: https://arxiv.org/pdf/1510.00149v5.pdf

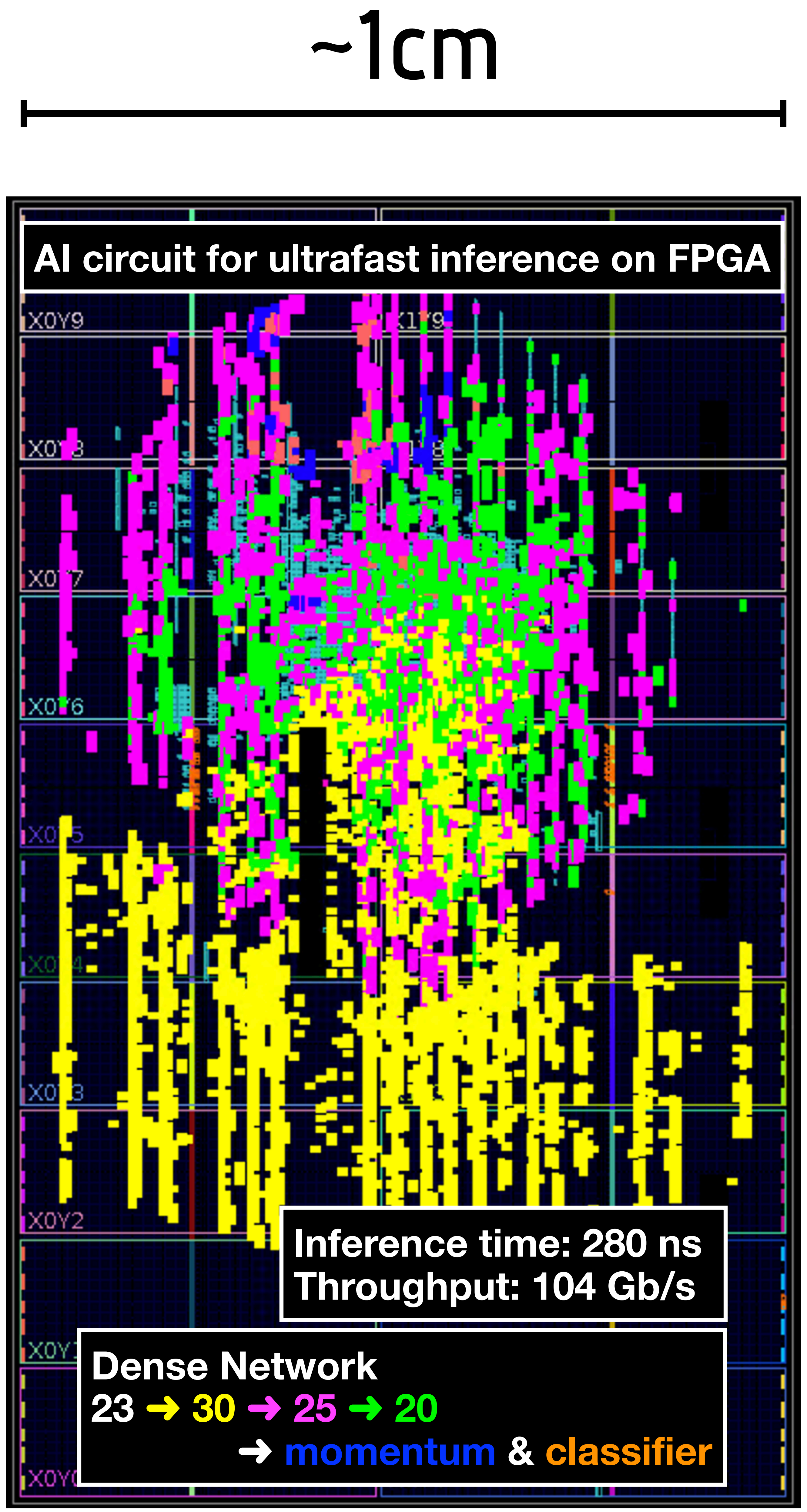
Example: Parallelization

ReuseFactor: how much to parallelize operations a hidden layer



hls4ml



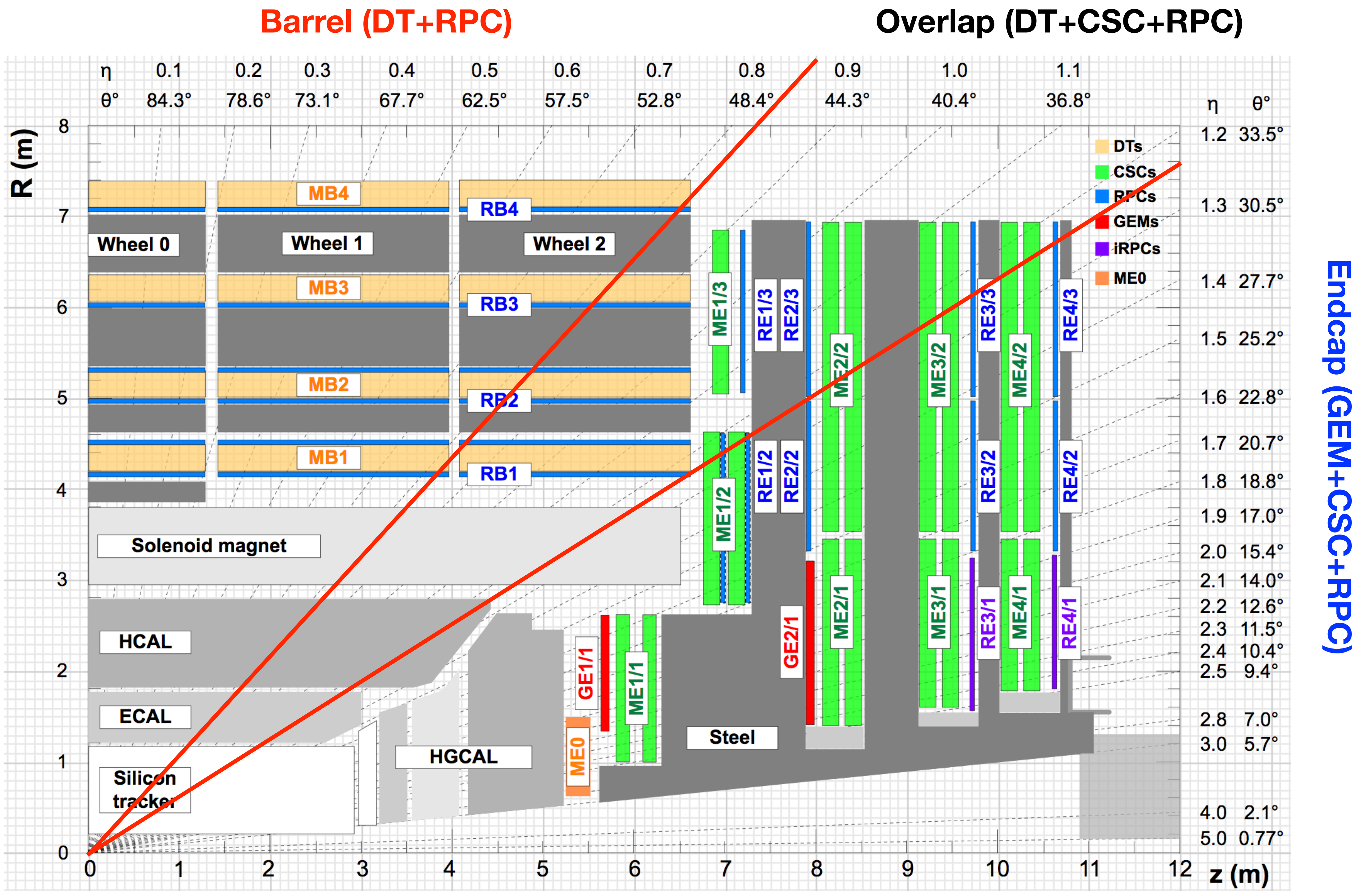


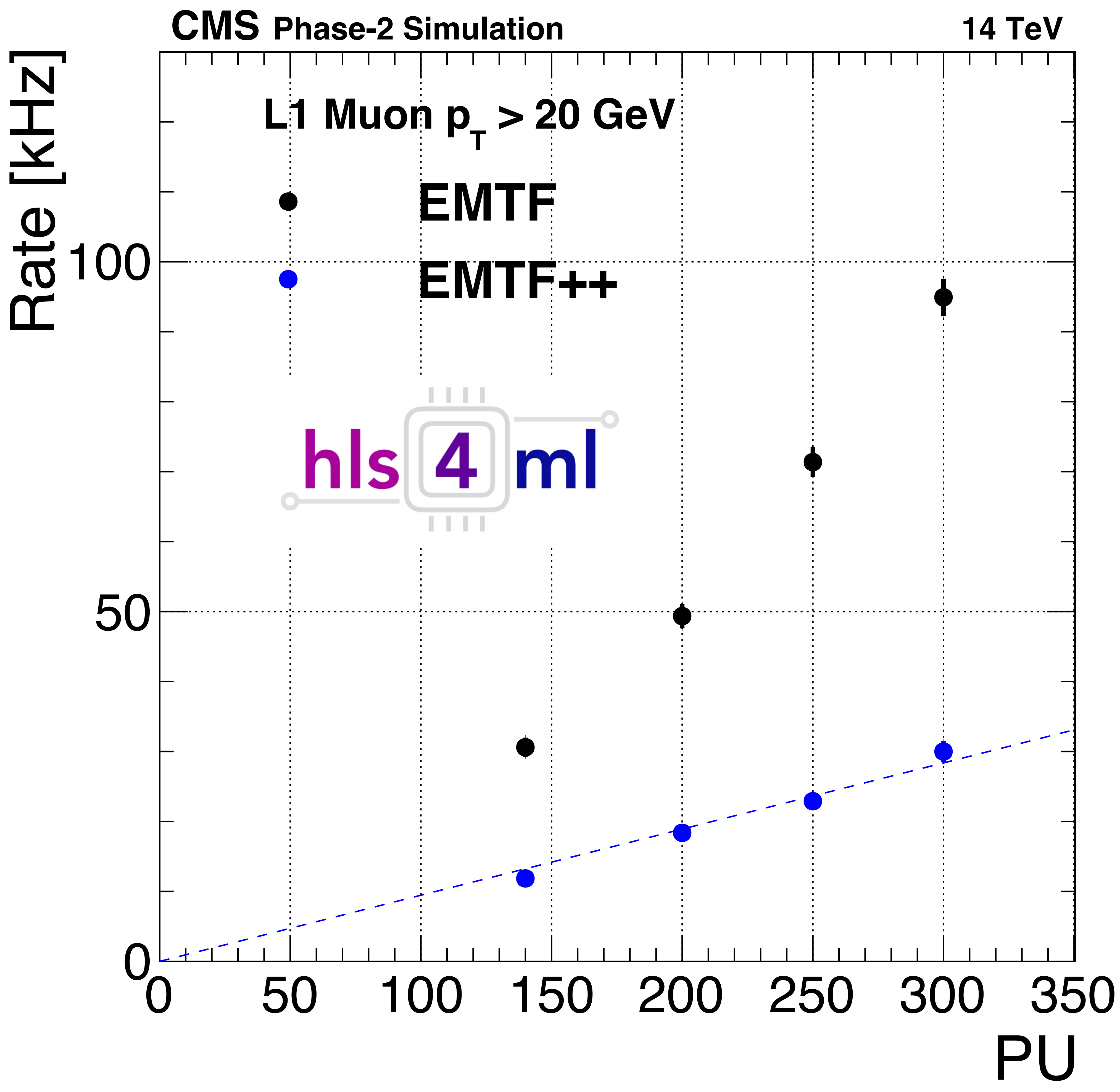
Deployed for LHC trigger systems
Active developments in new neural architectures, different hardware, more systems from ASICs to coprocessors, many domains, inter-FPGA networking

hls4ml - complete results

- Distance-Weighted **Graph Neural Networks** on FPGAs for Real-Time Particle Reconstruction in High Energy Physics, arXiv:2008.03601 [physics.comp-ph].
- Ultra Low-latency, Low-area Inference Accelerators using Heterogeneous **Deep Quantization with QKeras and hls4ml**, arXiv:2006.10159 [physics.ins-det].
- Compressing deep neural networks on FPGAs to **binary and ternary** precision with hls4ml, MLST (2020).
- Fast inference of **Boosted Decision Trees** in FPGAs for particle physics, JINST 15, P05026 (2020).
- **ESP4ML**: Platform-Based Design of Systems-on-Chip for Embedded Machine Learning, DATE Conference 2020 .
- **Fast inference of deep neural networks** in FPGAs for particle physics, JINST 13, P07027 (2018)

Case study: muon trigger upgrade





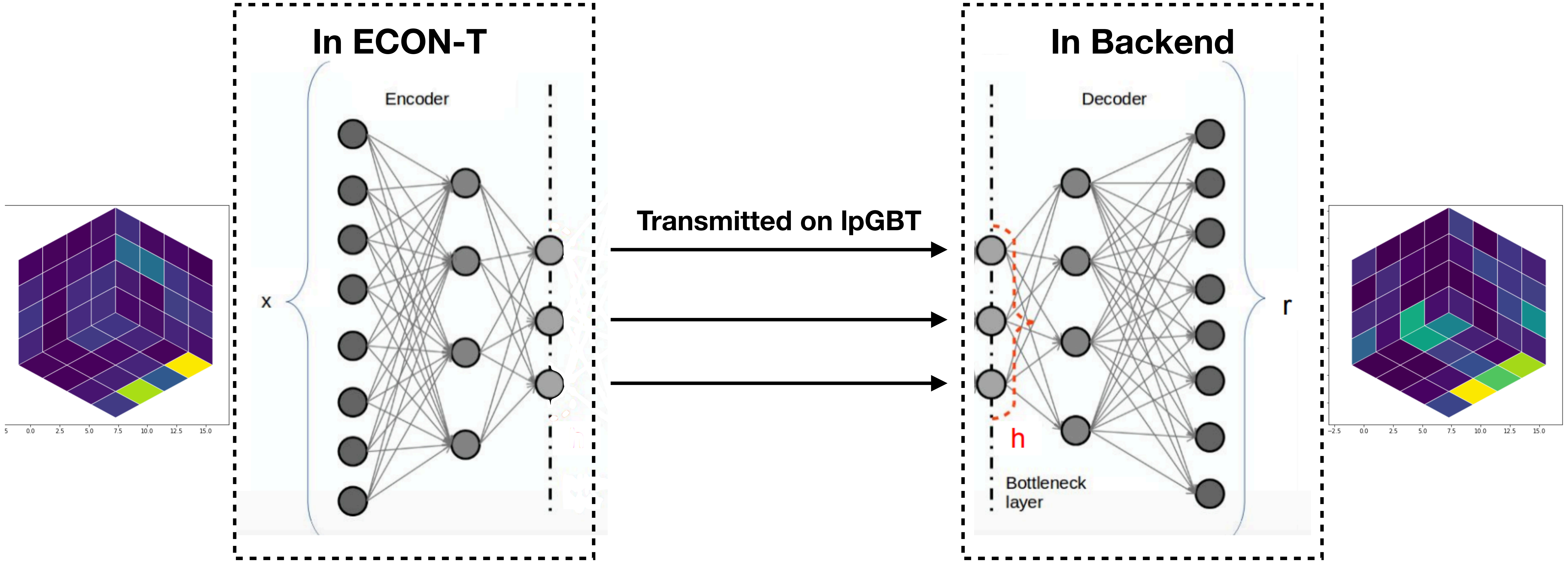
EMTF = BDT (external memory)

EMTF++ = NN

~3x reduction in the trigger rate for neural network!

What about ASICs?

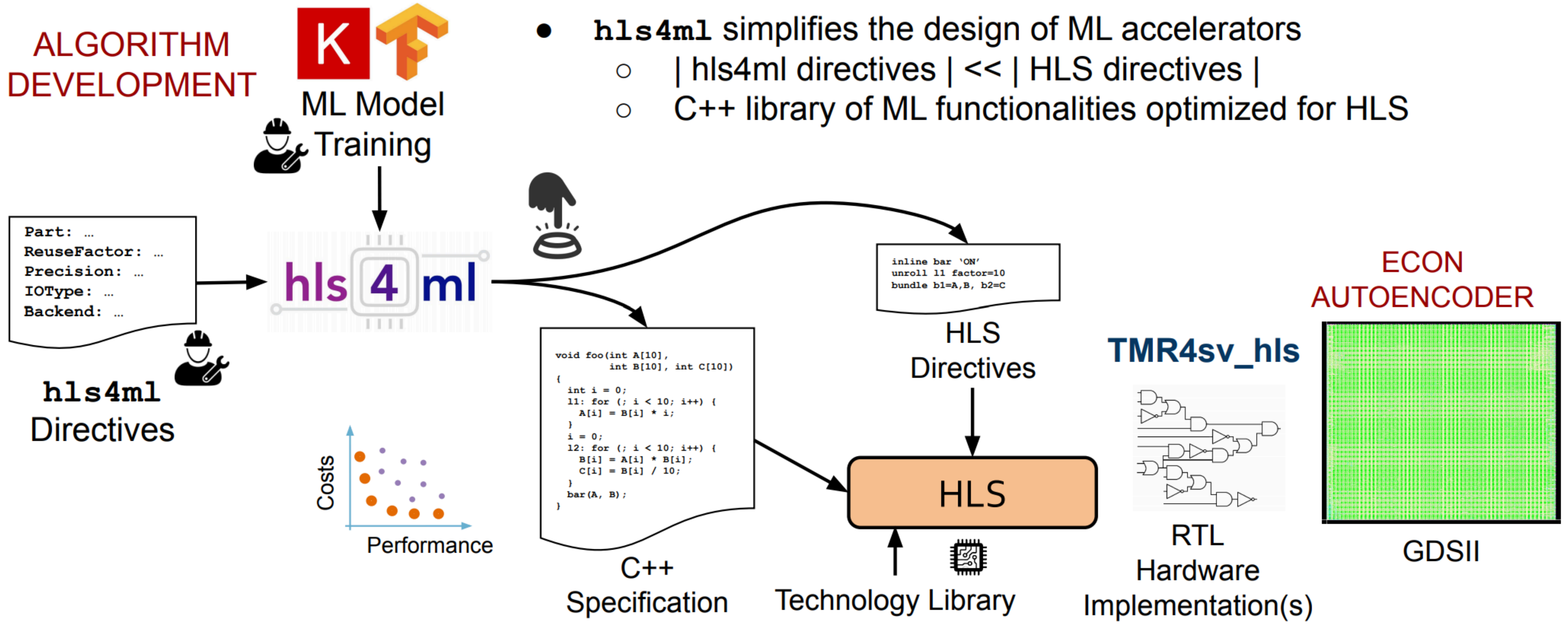
- Putting a neural network on the detector front-ends for data compression
  - ASIC required due to radiation tolerance and energy budget
- **Fully reconfigurable** to address future 'unknown unknowns' including evolving LHC conditions (pileup, beam bkgs), detector performance (noise, dead channels), performance metrics (resolution, substructure, new physics signatures)



ASIC workflow

Quantization aware training very important!

- **`hls4ml`** simplifies the design of ML accelerators
    - | hls4ml directives | << | HLS directives |
    - C++ library of ML functionalities optimized for HLS



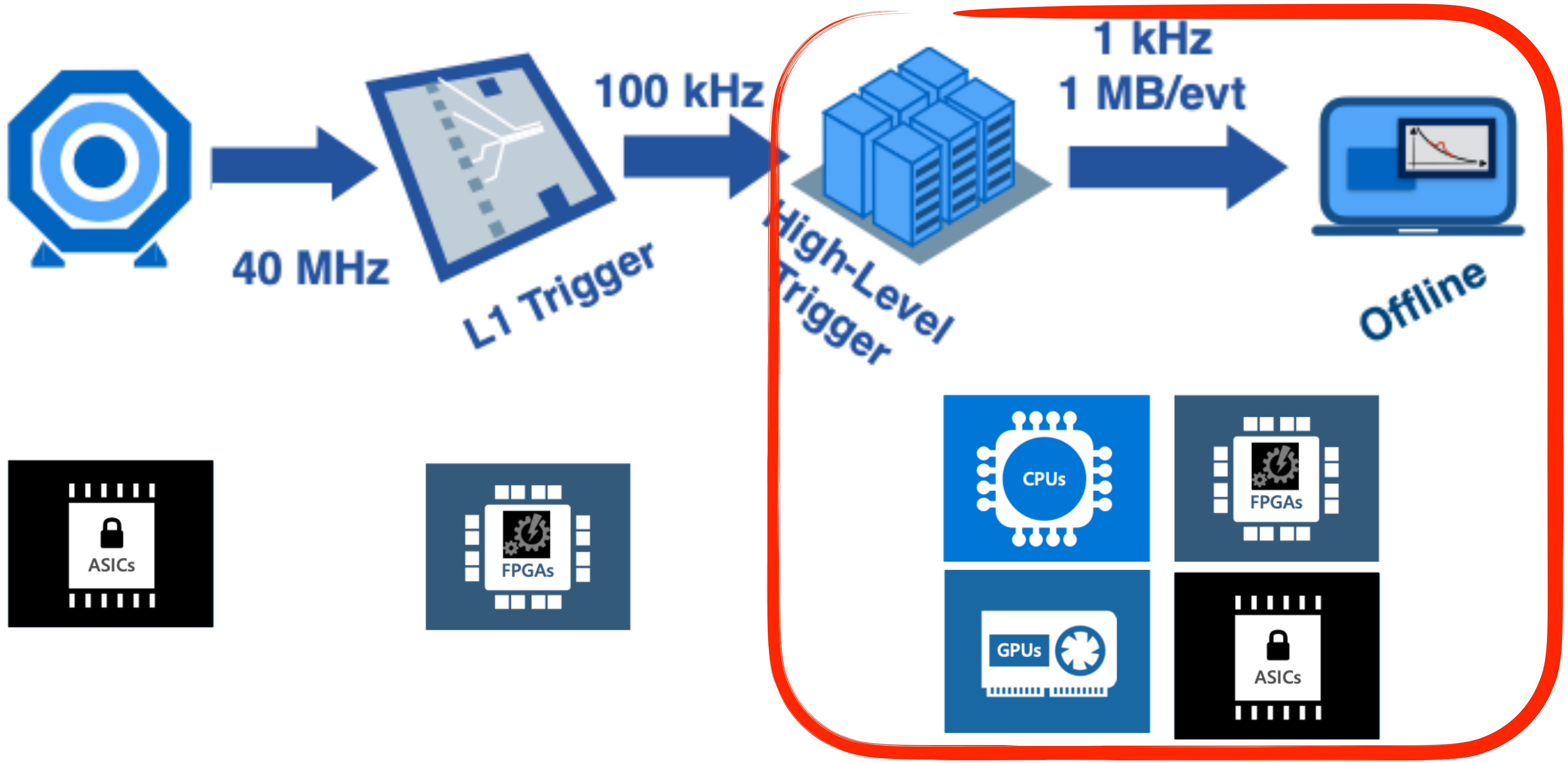
Look forward to public results at IEEE NSS and IEEE real-time 2020

Mini-summary

- Particle physics have been doing IoT for decades!
- On-sensor or near detector AI is powerful in reducing data rates while maintaining good physics performance

- **hls4ml** allows machine learning to be accessible in front-end electronics by physicists
- Broad range of applications
  - At LHC, from front-end ASICs to sub-detector electronics to back-end trigger algorithms
  - Many other applications in physics and beyond!
    - DUNE supernovae trigger
    - Accelerator real-time controls and operations
    - Other domains: nuclear physics, microscopy, signal processing,…

Accelerated ML for HEP computing



Why fast inference?

- Training has its own computing challenges
  - But happens ~once/year and outside of compute infrastructure
- Inference happens on billions of events many times a year
  - Unique challenge across HEP
  - Massive datasets of statistically independent events

Opportunities for Accelerated Machine Learning Inference in Fundamental Physics

Javier Duarte[1], Philip Harris[2], Alex Himmel[3], Burt Holzman[3], Wesley Ketchum[3], Jim Kowalkowski[3], Miaoyuan Liu[3], Brian Nord[3], Gabriel Perdue[3], Kevin Pedro[3], Nhan Tran[3], and Mike Williams[2]

[1]University of California San Diego, La Jolla, CA 92093, USA
[2]Massachusetts Institute of Technology, Cambridge, MA 02139, USA
[3]Fermi National Accelerator Laboratory, Batavia, IL 60510, USA

ABSTRACT

In this brief white paper, we discuss the future computing challenges for fundamental physics experiments. The use cases for deploying machine learning across physics for simulation, reconstruction, and analysis is rapidly growing. This will lead us to many applications where exploring accelerated machine learning algorithm inference could bring valuable and necessary gains in performance. Finally, we conclude by discussing the future challenges in deploying new heterogeneous computing hardware.

This community report is inspired by discussions at the Fast Machine Learning Workshop[1] held September 10-13, 2019.

Contents

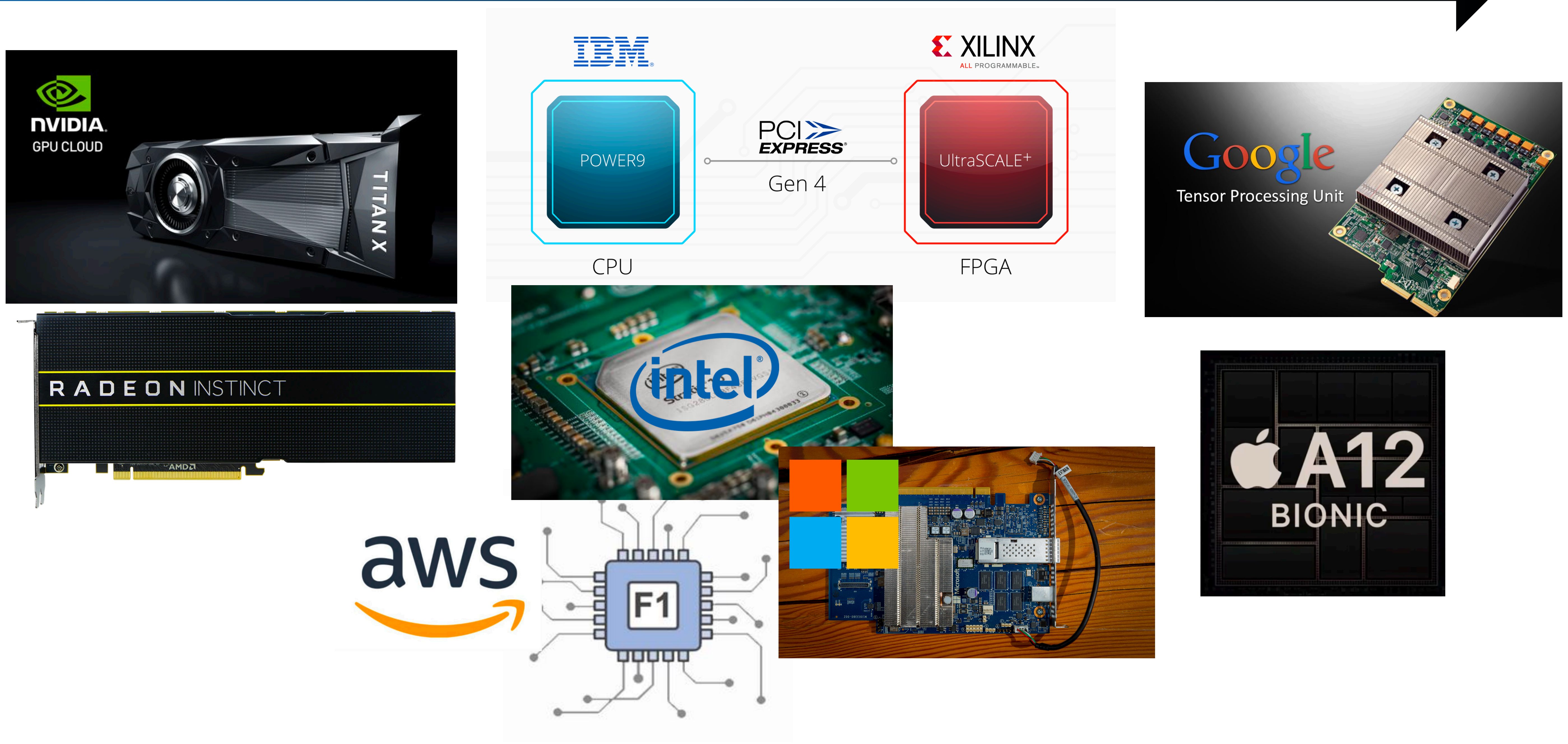
1 Introduction 1
1.1 Computing model in particle physics . . . 1
1.2 Machine Learning . . . 2
2 Challenges and Applications for Accelerated Machine Learning Inference 2
2.1 CMS and ATLAS . . . 2
2.2 LHCb . . . 3
2.3 LSST . . . 4
2.4 LIGO . . . 4
2.5 DUNE . . . 5
3 Outlook and Opportunities 6

Link

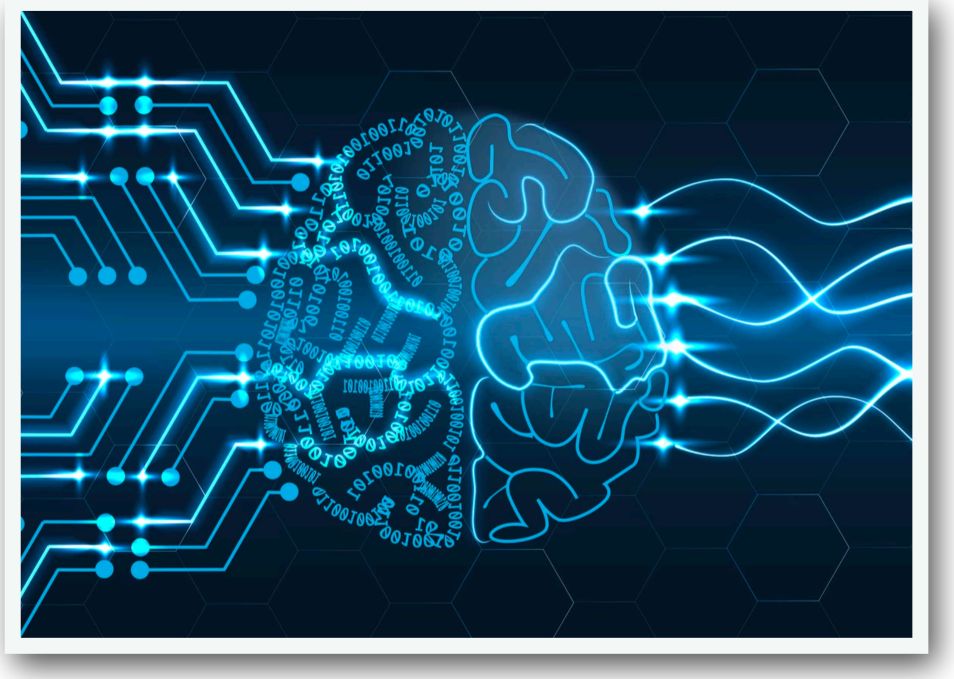
Heterogeneous compute

CPUs | GPUs | FPGAs | ASICs

FLEXIBILITY ⟷ EFFICIENCY



Advances in heterogeneous computing driven by machine learning

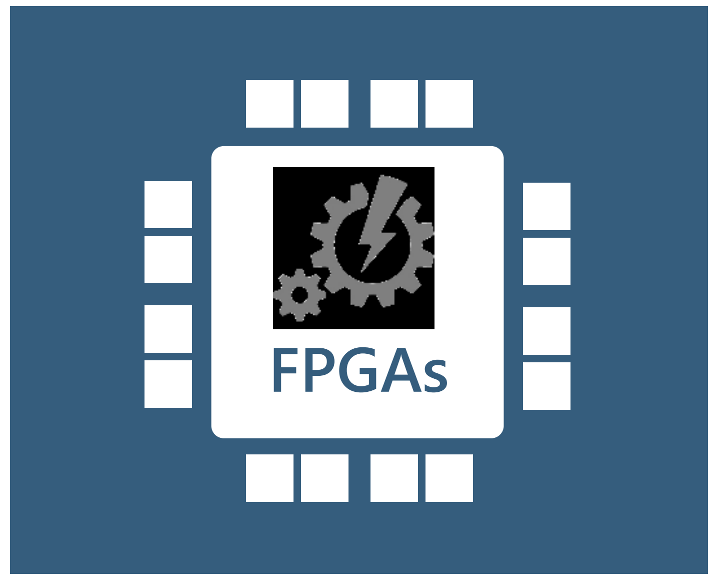


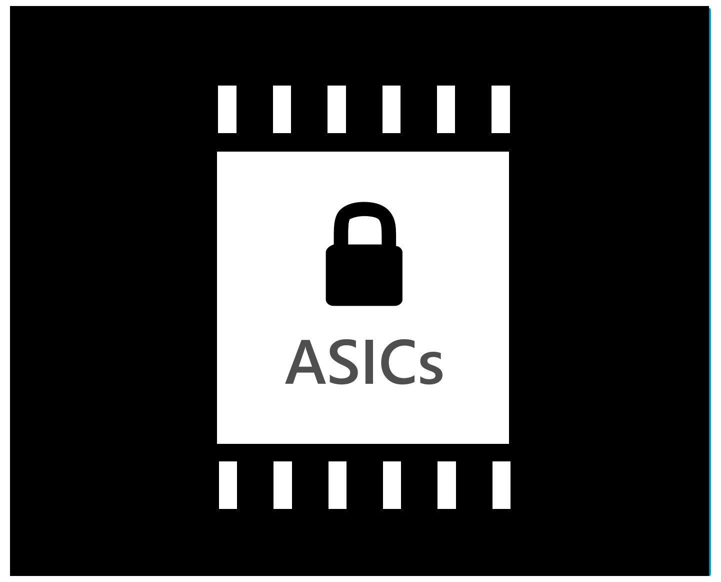
Heterogeneous compute

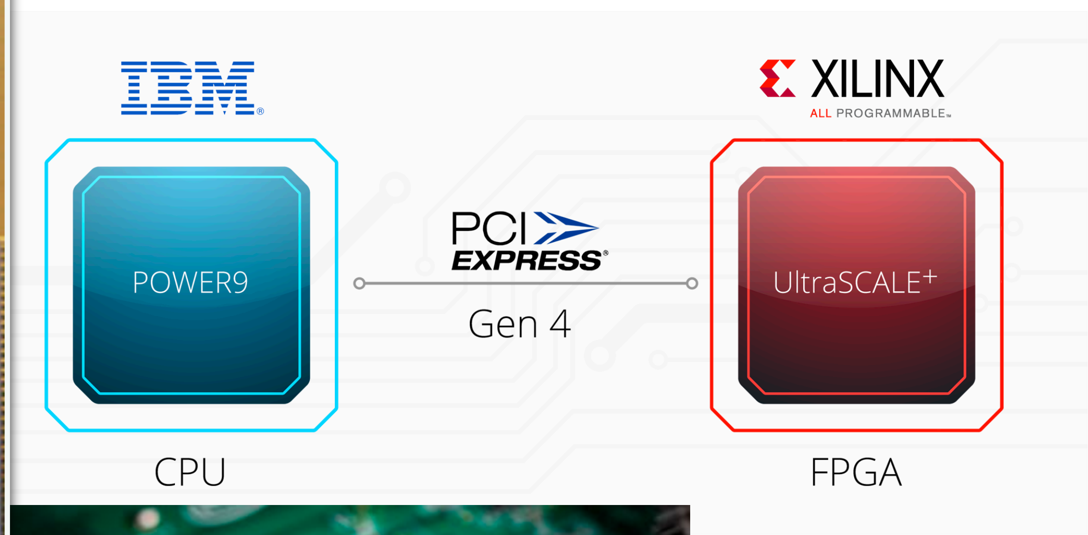

COMMUNICATIONS OF THE ACM
CACM.ACM.ORG
02/2019 VOL.62 NO.02

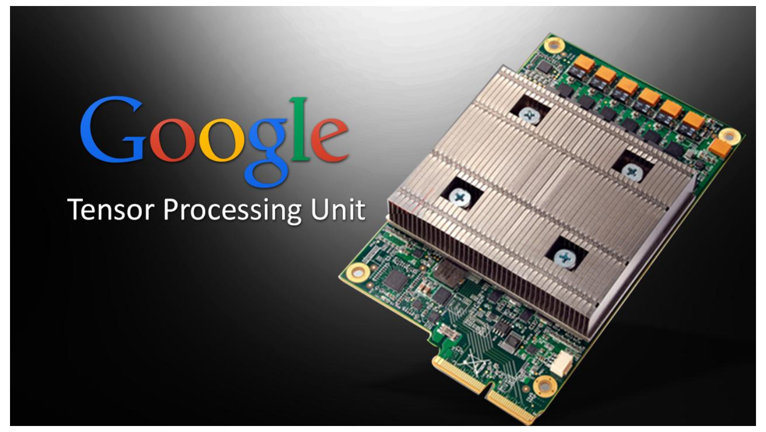
A New Golden Age for Computer Architecture

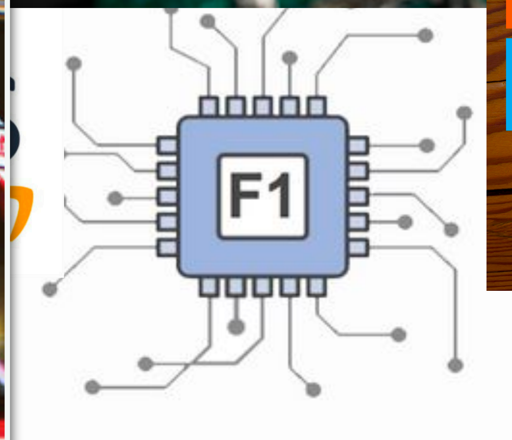
Agriculture Technology
Monitoring Noise Pollution
The Computational Sprinting Game
Blockchain from a Distributed Computing Perspective

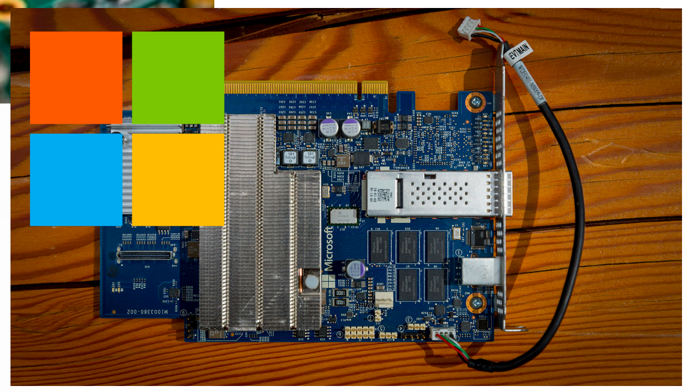




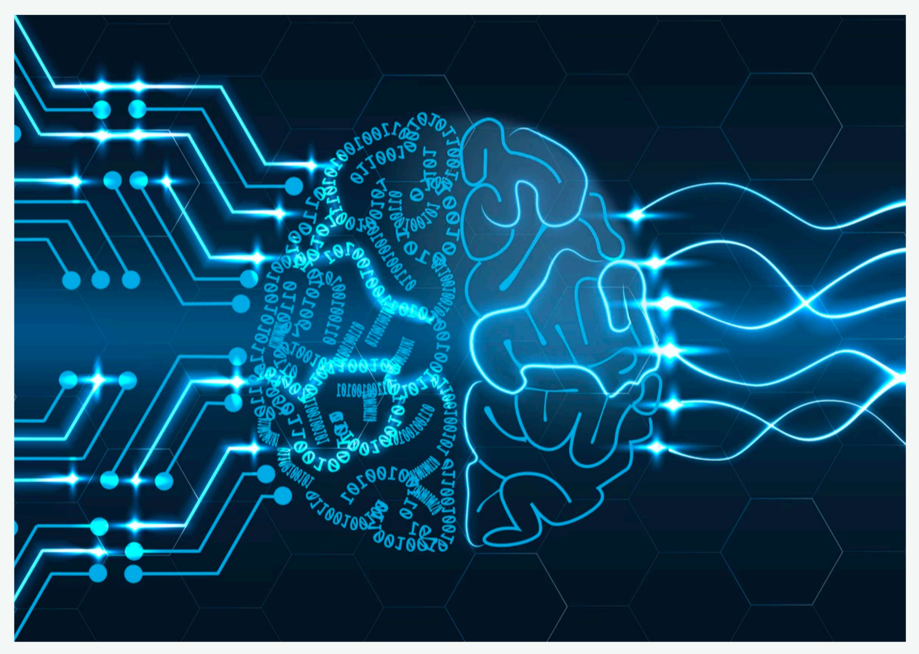




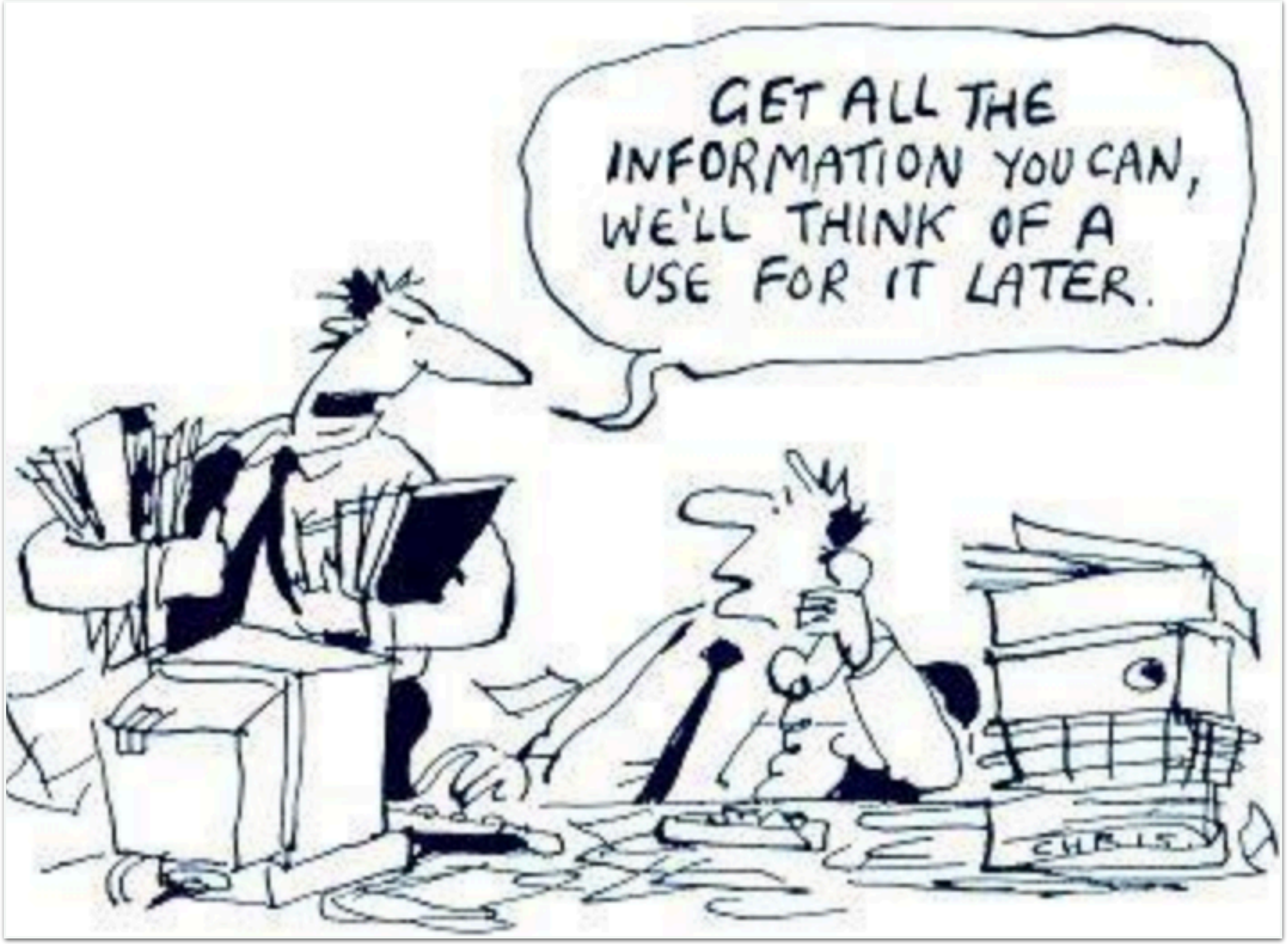




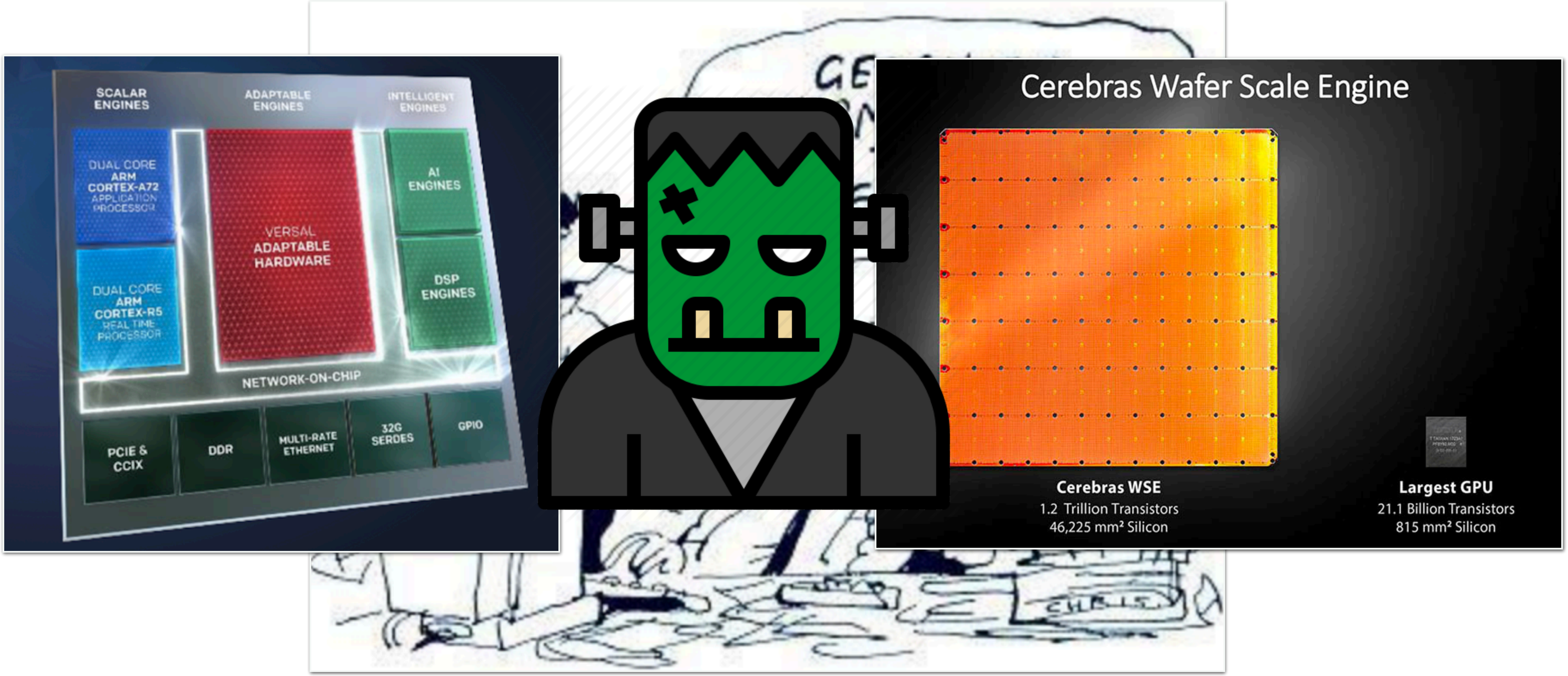
Advances in heterogeneous computing driven by machine learning



hardware choices

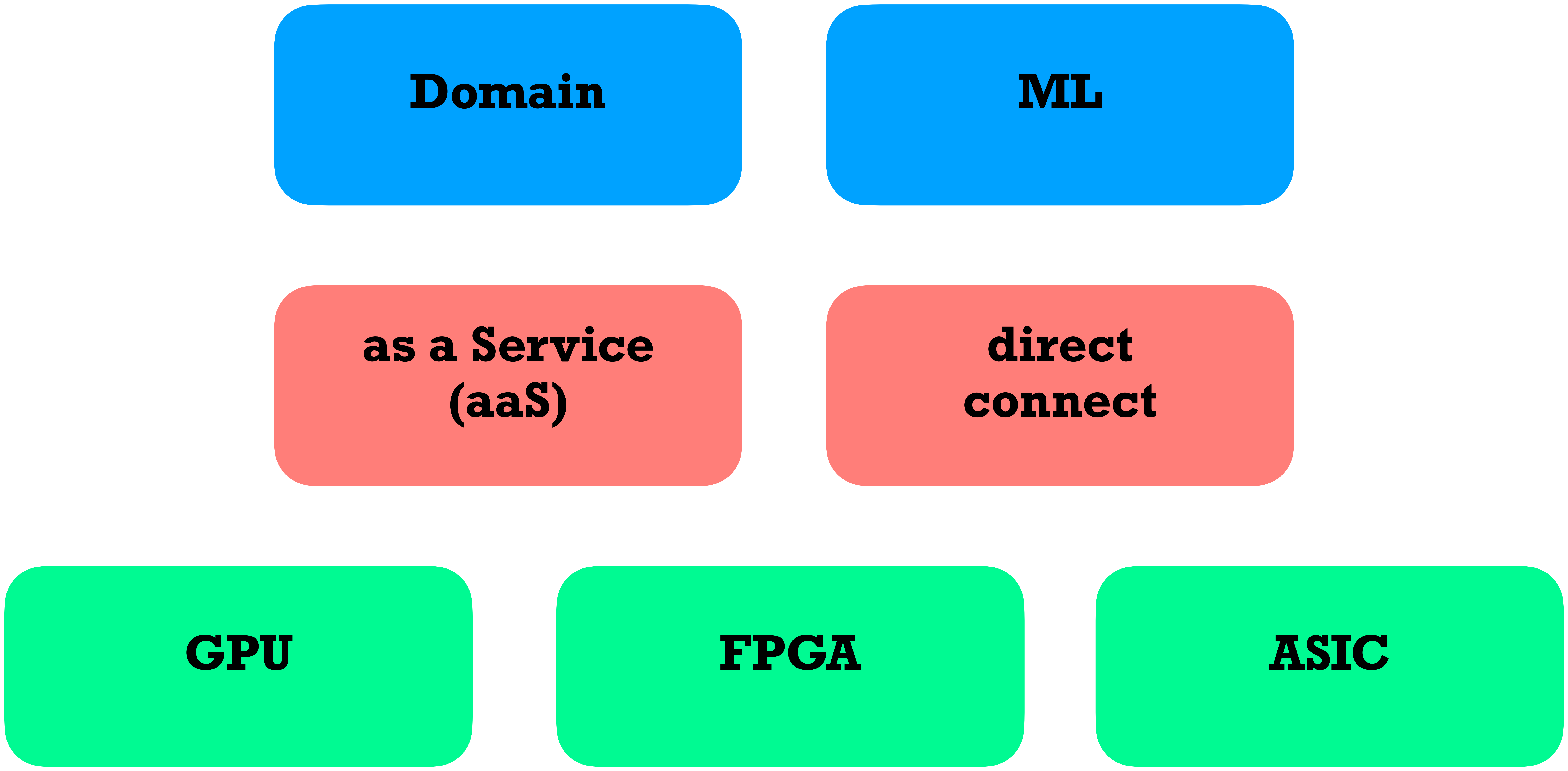


hardware choices



Pros & Cons

On how to integrate heterogeneous compute into our computing model



To ML or not to ML

Re-engineer physics algorithms for new hardware

Language: OpenCL, OpenMP, HLS, Kokkos,…?

Hardware: CPU, FPGA, GPU

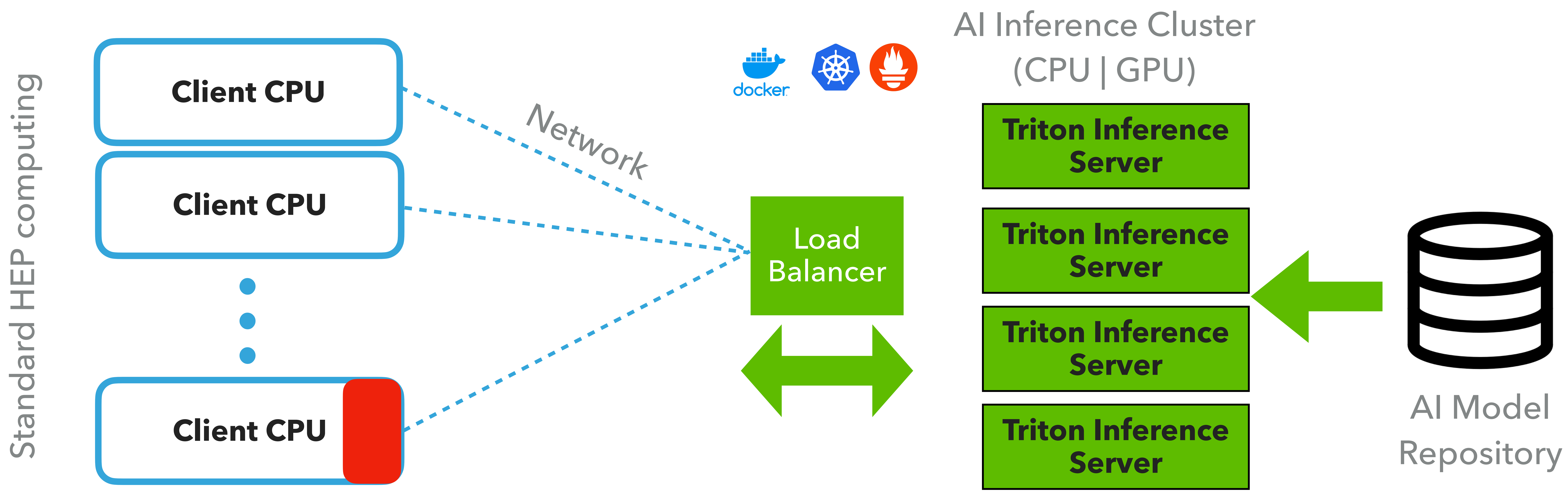
Re-cast physics problem as a machine learning problem

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

Hardware: CPU, FPGA, GPU, ASIC

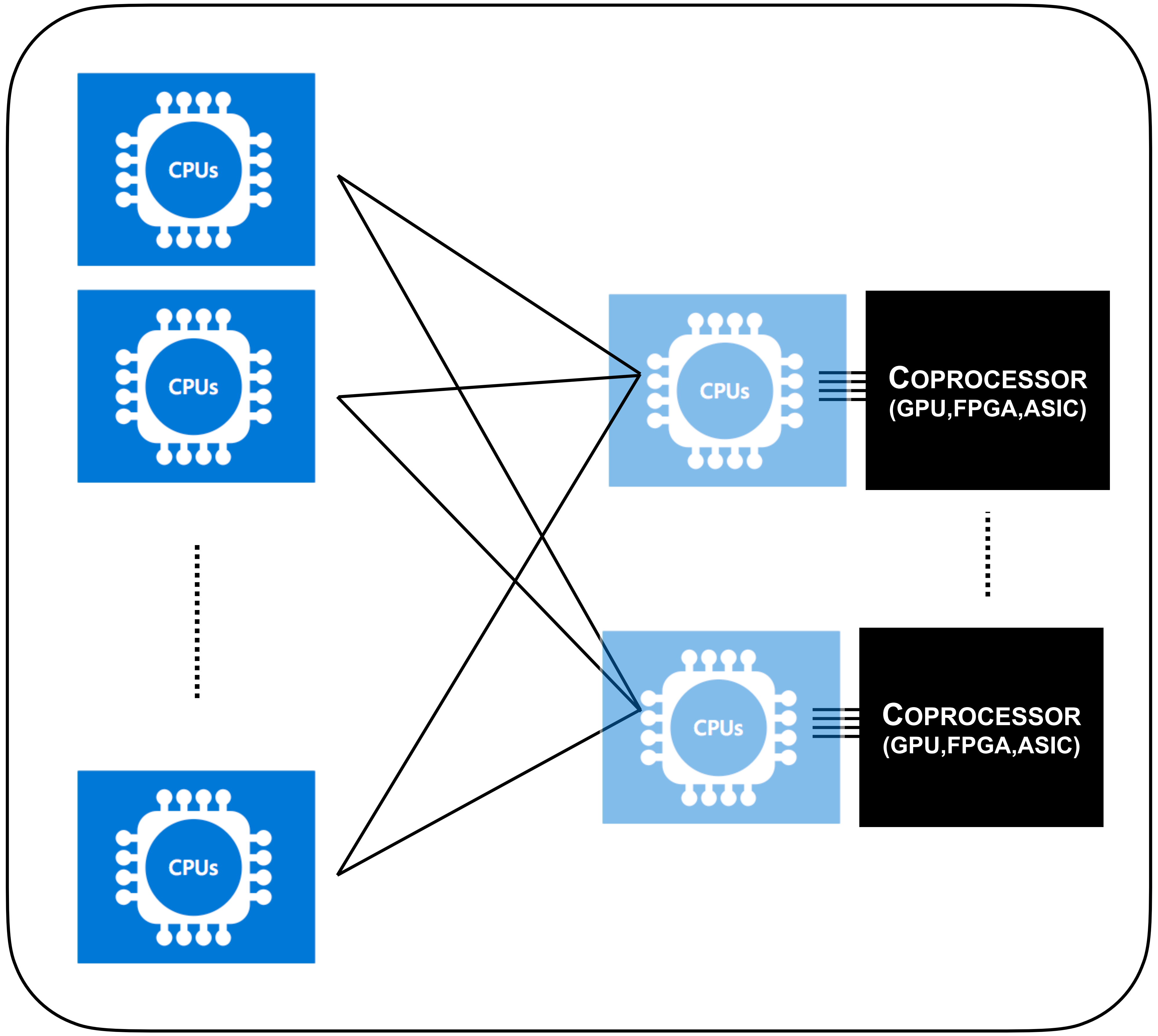
Is there a way to have the best of both worlds with physics aware ML?

GPUaaS + SONIC

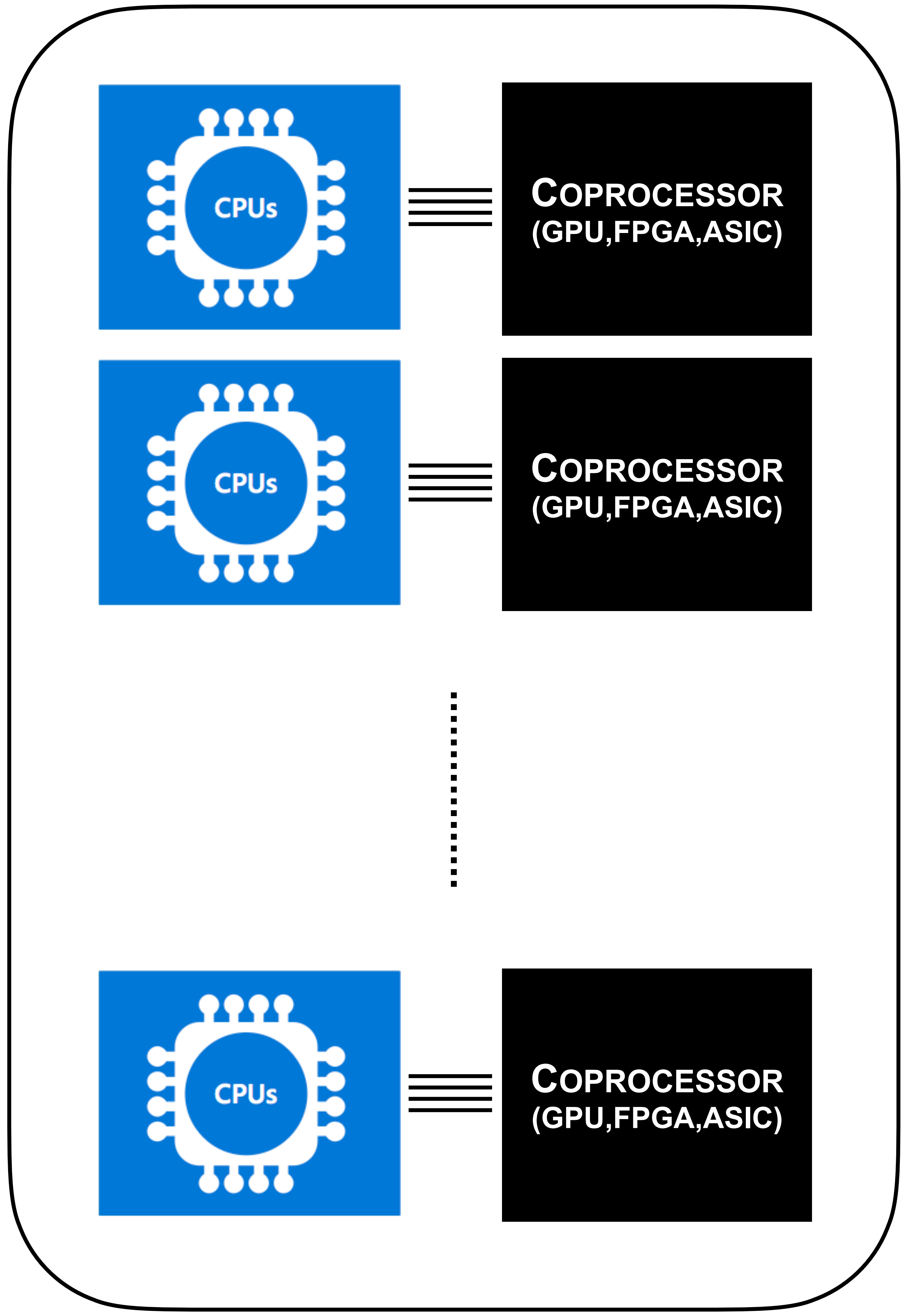


SONIC: Services Optimized for Network Inference on Coprocessors

aaS or direct connect

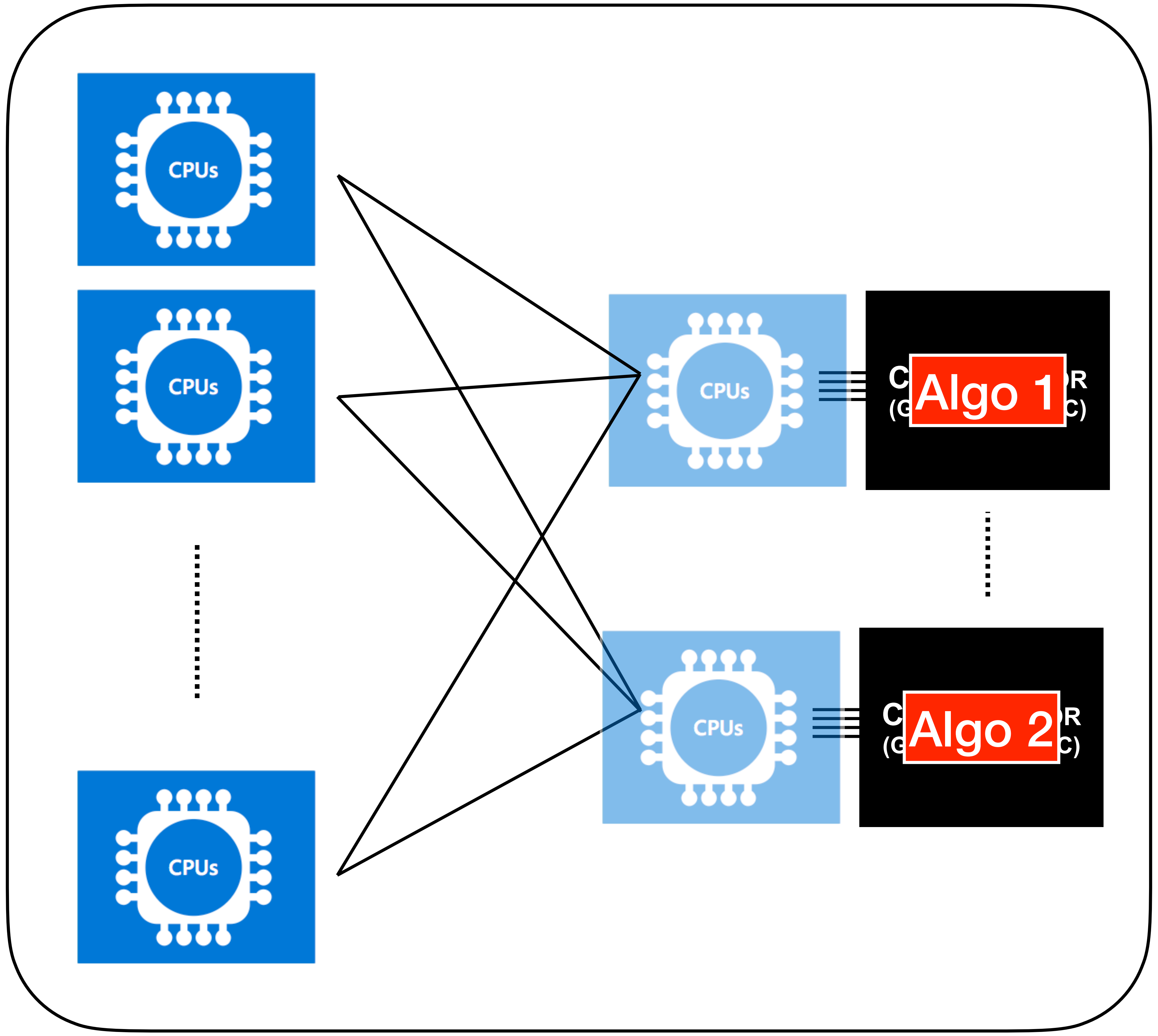


Pros:
scalable algorithms
scalable to the grid/cloud
Heterogeneous heterogeneity (mixed hardwares)

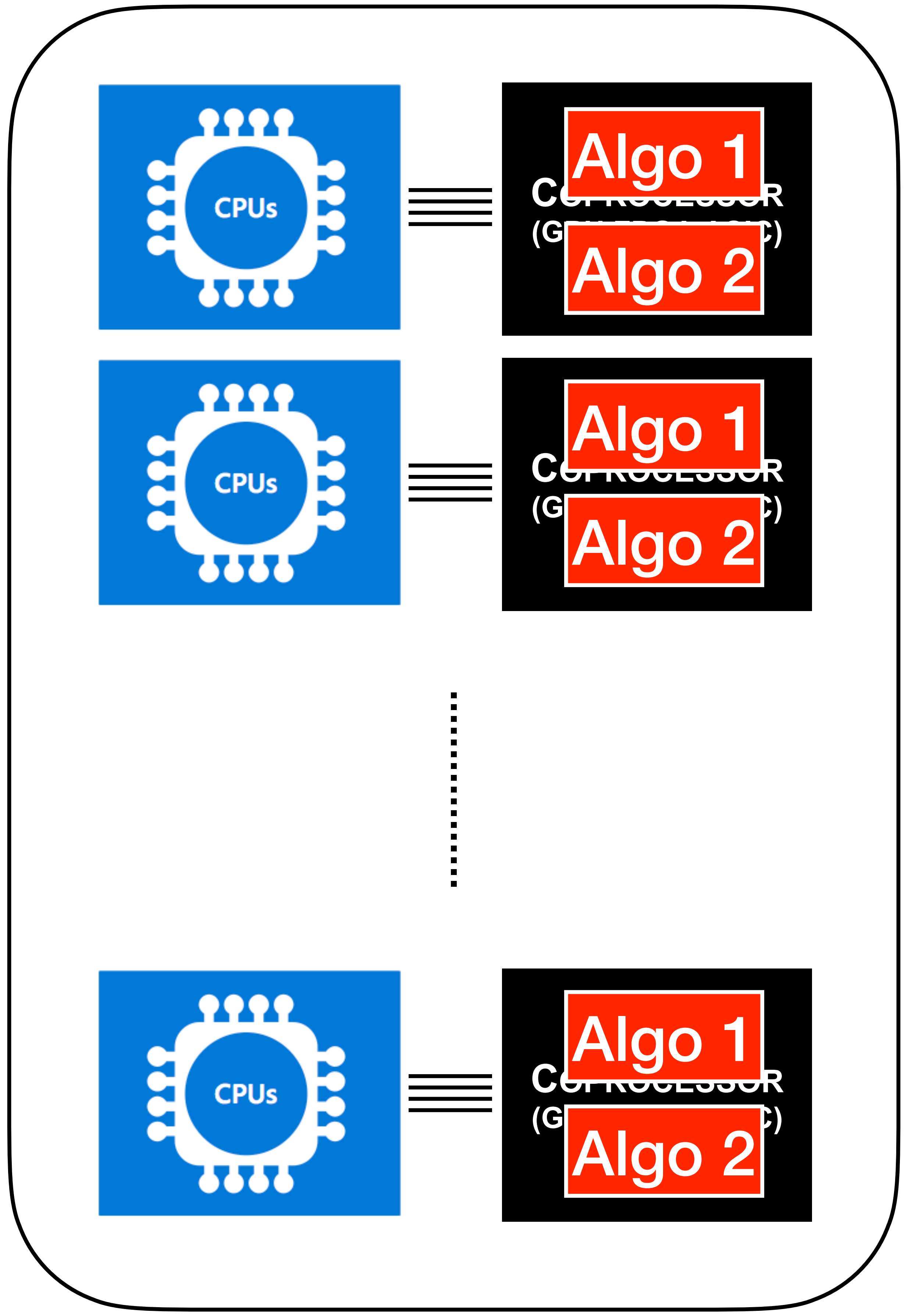


Pros:
less system complexity
no network latency

aaS or direct connect



Pros:
scalable algorithms
scalable to the grid/cloud
Heterogeneous heterogeneity (mixed hardwares)



Pros:
less system complexity
no network latency

Towards abstraction:
on-premises, in the cloud, oh my!

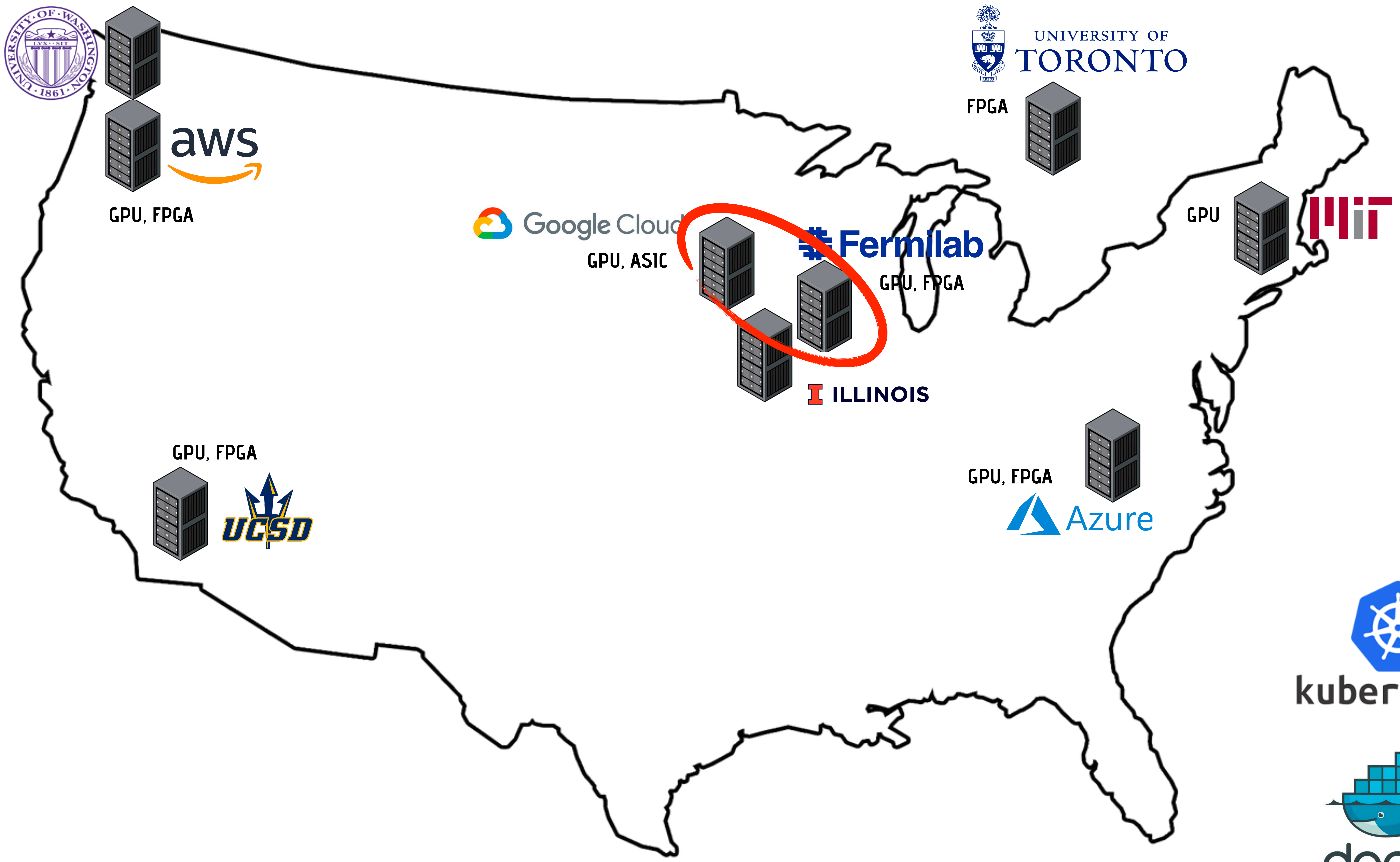
kubernetes

docker

Building a network of heterogeneous resources in the cloud and on-premises

Work-in-progress: how to coordinate and orchestrate distributed heterogeneous resources

Towards abstraction:
on-premises, in the cloud, oh my!



kubernetes

docker

Building a network of heterogeneous resources in the cloud and on-premises

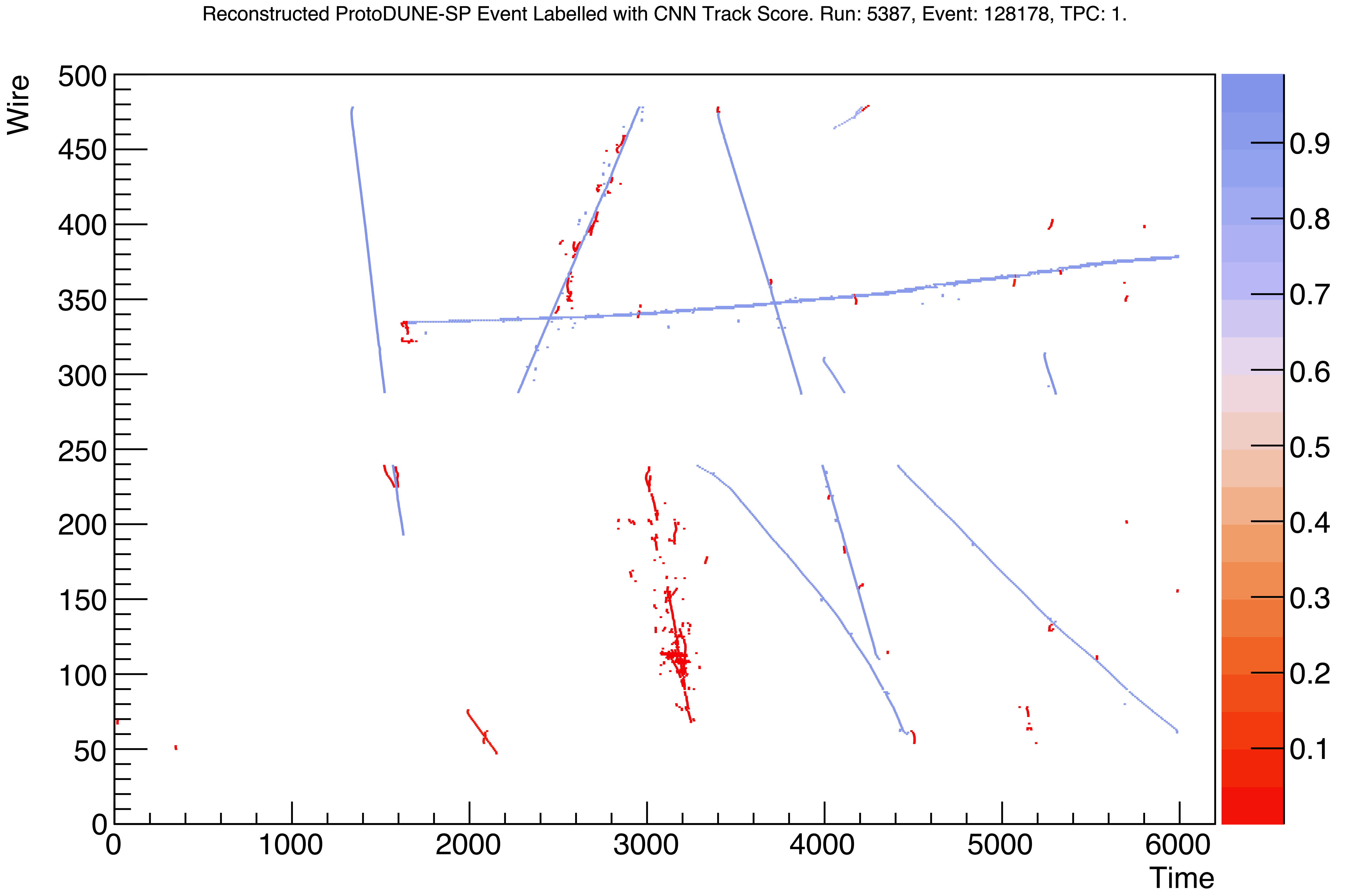
Work-in-progress: how to coordinate and orchestrate distributed heterogeneous resources https://arxiv.org/abs/2009.04509

Neutrino case study

GPU-accelerated machine learning inference as a service for computing in neutrino experiments

Michael Wang [1,*], **Tingjun Yang** [1], **Maria Acosta Flechas** [1], **Philip Harris** [2], **Benjamin Hawks** [1], **Burt Holzman** [1], **Kyle Knoepfel** [1], **Jeffrey Krupa** [2], **Kevin Pedro** [1], **Nhan Tran** [1,3]

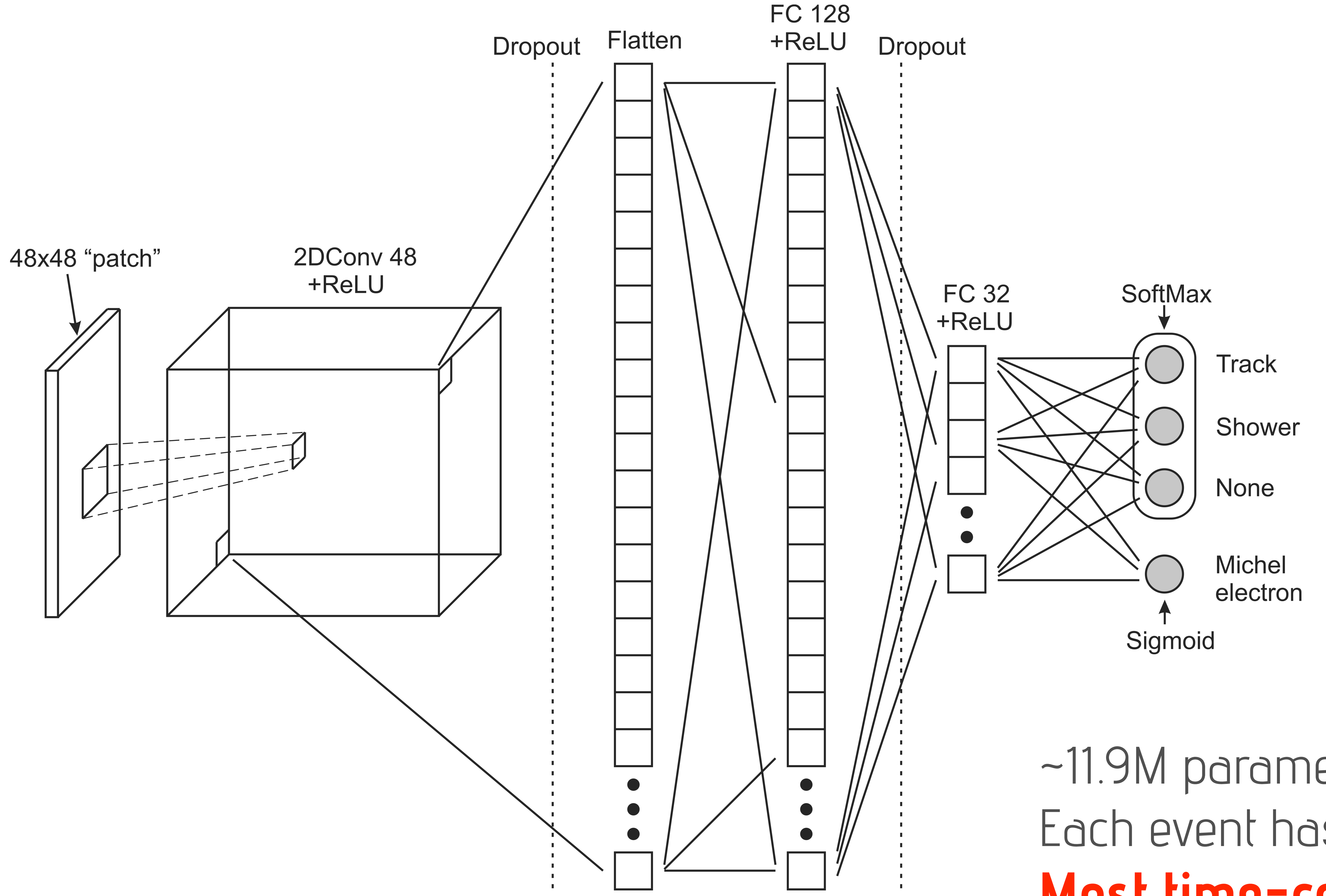
[1] *Fermi National Accelerator Laboratory, Batavia, IL 60510, USA*
[2] *Massachusetts Institute of Technology, Cambridge, MA 02139, USA*
[3] *Northwestern University, Evanston, IL 60208, USA*



ProtoDUNE reconstruction

Reconstruction chain
- Noise mitigation and deconvolution
- Hit finder
- Pandora pattern recognition
- **CNN EmTrkMichelId**

- Largest LArTPC ever built
  - 7.2 x 6.0 x 6.9 m^3
  - 15,360 channels
  - Wire spacing 5 mm
  - Readout window 3 ms
- Lots of activities in the TPC
  - Cosmic ray muons
  - Beam particles



~11.9M parameters
Each event has ~55k patches
Most time-consuming module in the reco chain.

ProtoDUNE reconstruction

Reconstruction chain

- Noise mitigation and deconvolution
- Hit finder
- Pandora pattern recognition
- **CNN EmTrkMichelId**

- Largest LArTPC ever built
  - 7.2 x 6.0 x 6.9 m^3
  - 15,360 channels
  - Wire spacing 5 mm
  - Readout window 3 ms
- Lots of activities in the TPC
  - Cosmic ray muons
  - Beam particles

Wall time (s)		
ML module	non-ML modules	Total
220	110	330

CPU type	fraction (%)
AMD EPYC 7502 @ 2.5 GHz	11.7
AMD Opteron 6134 @ 2.3 GHz	0.6
AMD Opteron 6376 @ 2.3 GHz	4.6
Intel Xeon E5-2650 v2 @ 2.6 GHz	30.8
Intel Xeon E5-2650 v3 @ 2.3 GHz	5.2
Intel Xeon E5-2670 v3 @ 2.3 GHz	7.3
Intel Xeon E5-2680 v4 @ 2.4 GHz	17.3
Intel Xeon Gold 6140 @ 2.3 GHz	22.6

~11.9M parameters

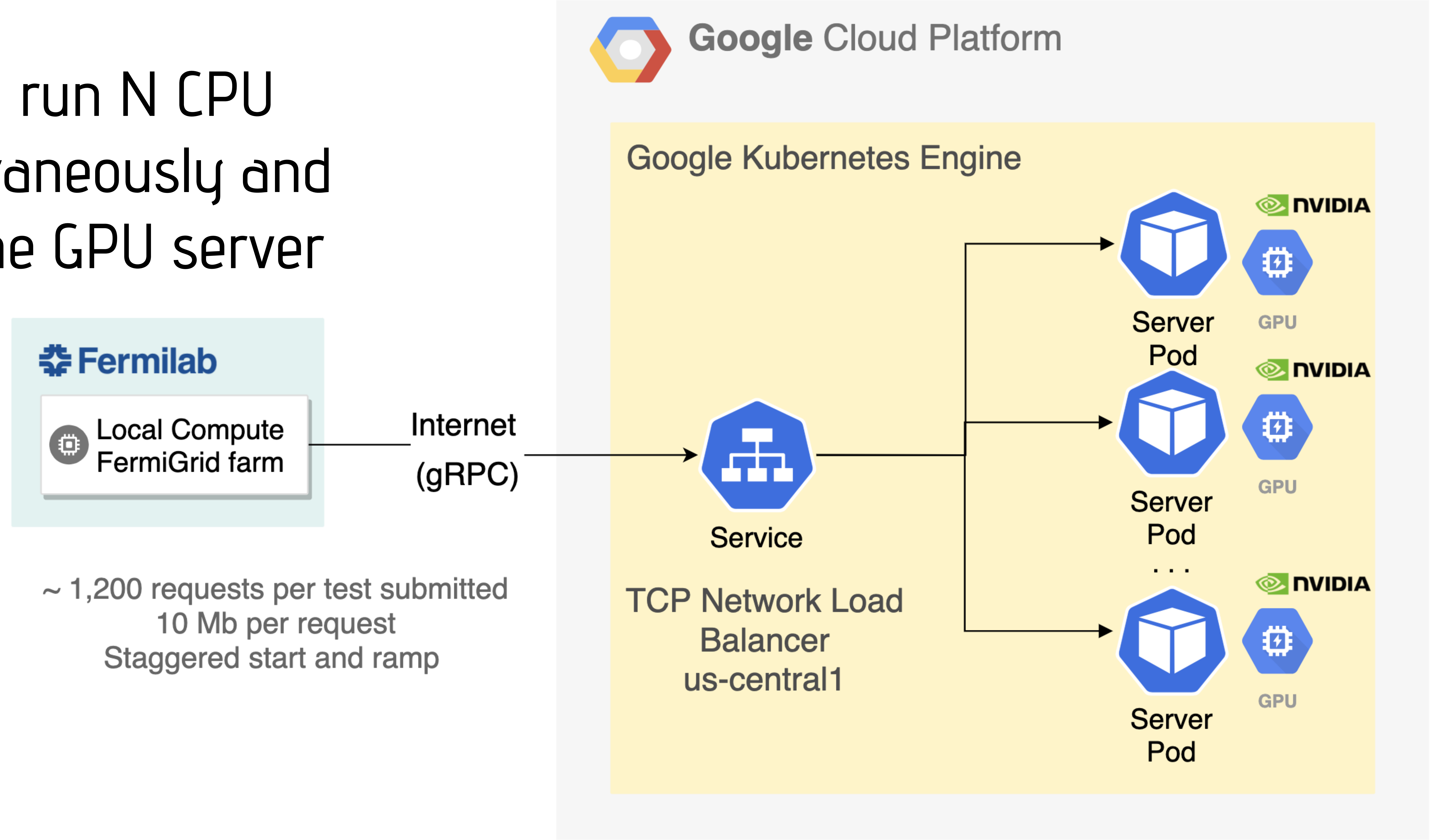
Each event has ~55k patches

Most time-consuming module in the reco chain.

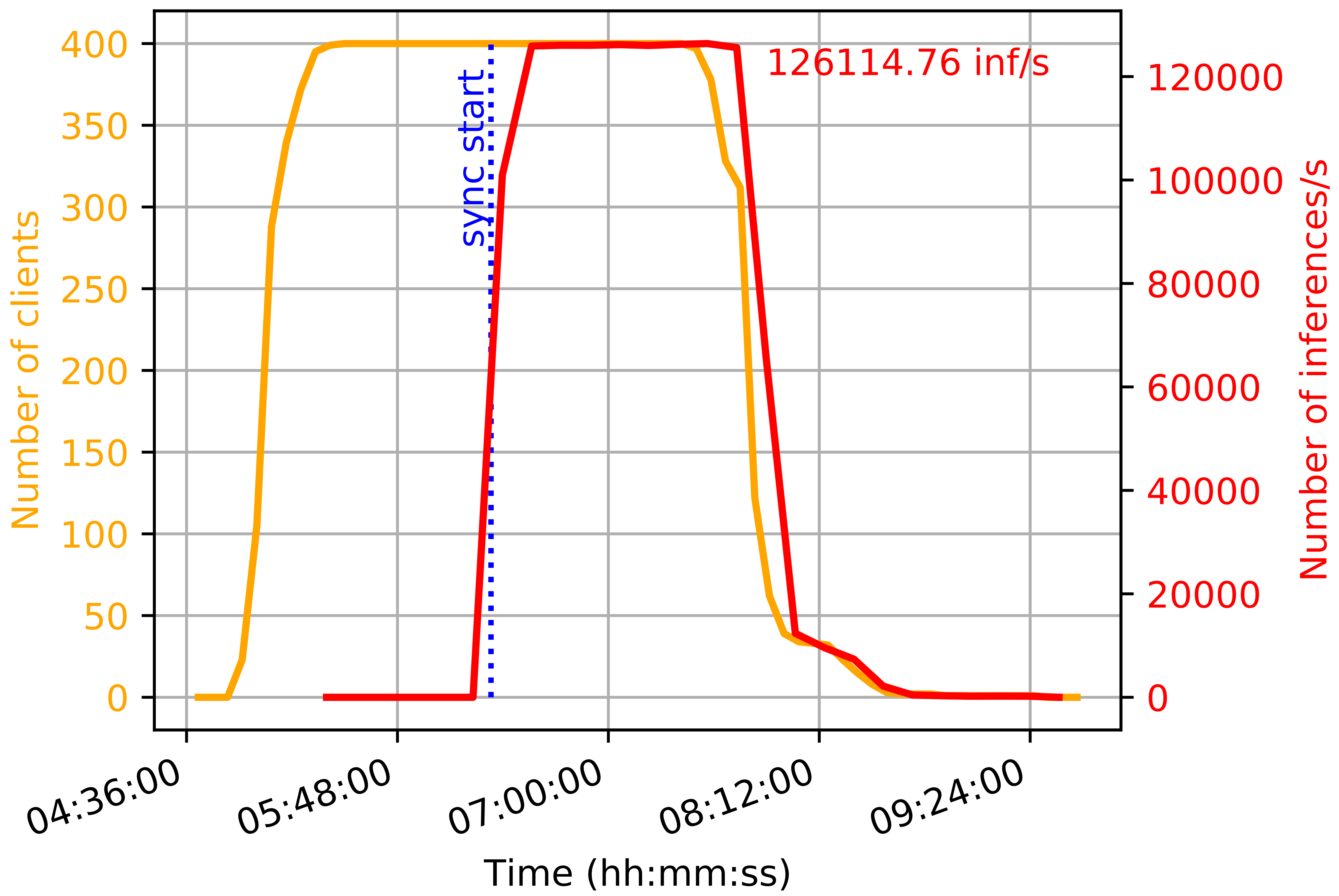
Client-server configuration

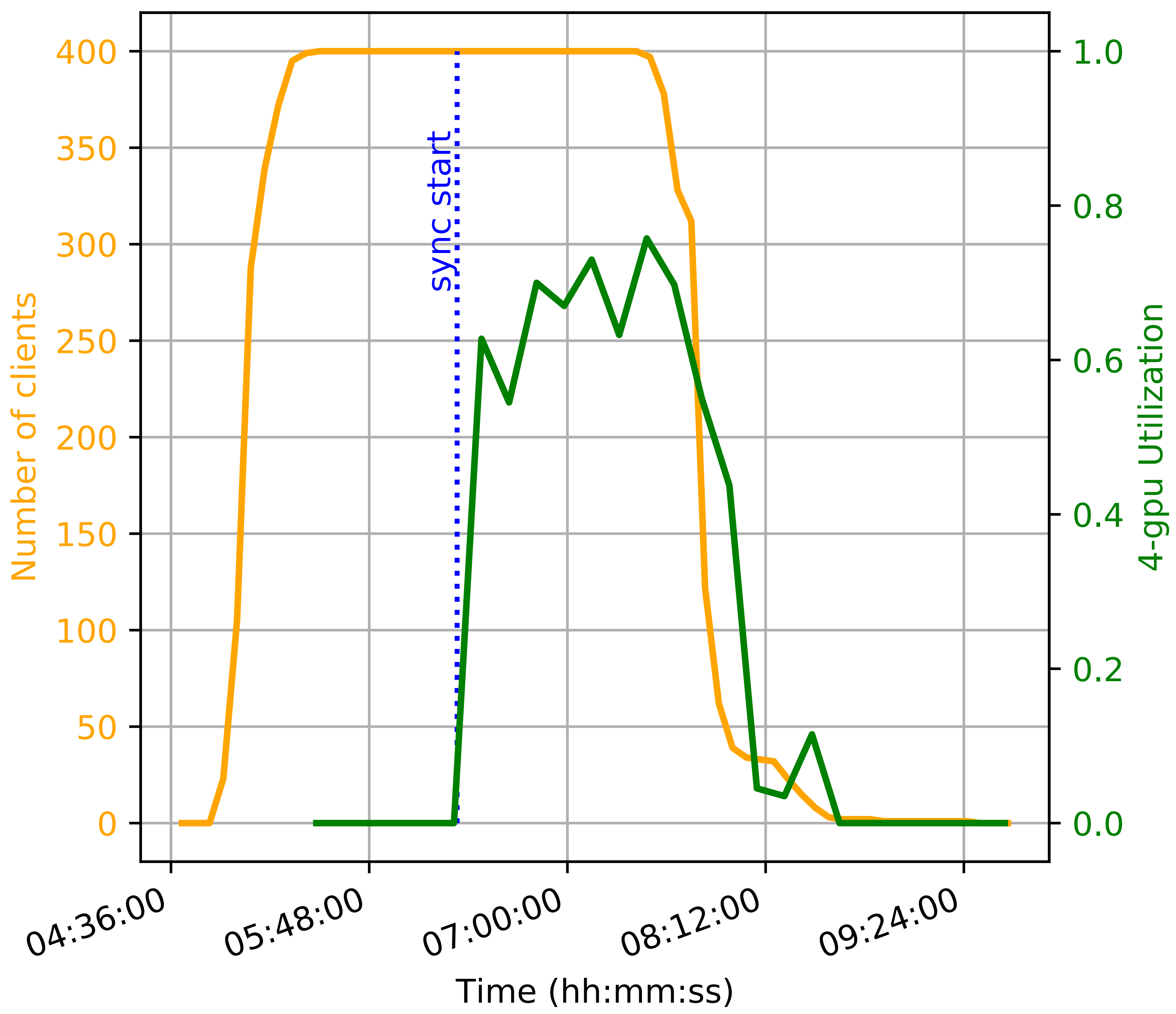
Server side:
4 NVidia T4 GPUs

Client side: run N CPU jobs simultaneously and hammer the GPU server



Server metrics





126k inferences/s for 4 GPUs with 60% GPU usage

Breakdown

Wall time (s)		
ML module	non-ML modules	Total
~~220~~ ~11s	110	330

$11s \sim t_{preprocess} + t_{transmit} + t_{travel} + t_{GPU}$

$t_{preprocess}$	$t_{transmit}$	t_{travel}	t_{GPU}
7s	2s	0.4s	1.8s
On CPU, preparing NN inputs	Based on 2Gbps ethernet bandwidth	Ping latency between Iowa and FNAL	Time on the GPU

** subtleties in the numbers: affected by dynamic batching, ethernet bandwidth, and batch sizes, can change total time by ~6s more

Modeling

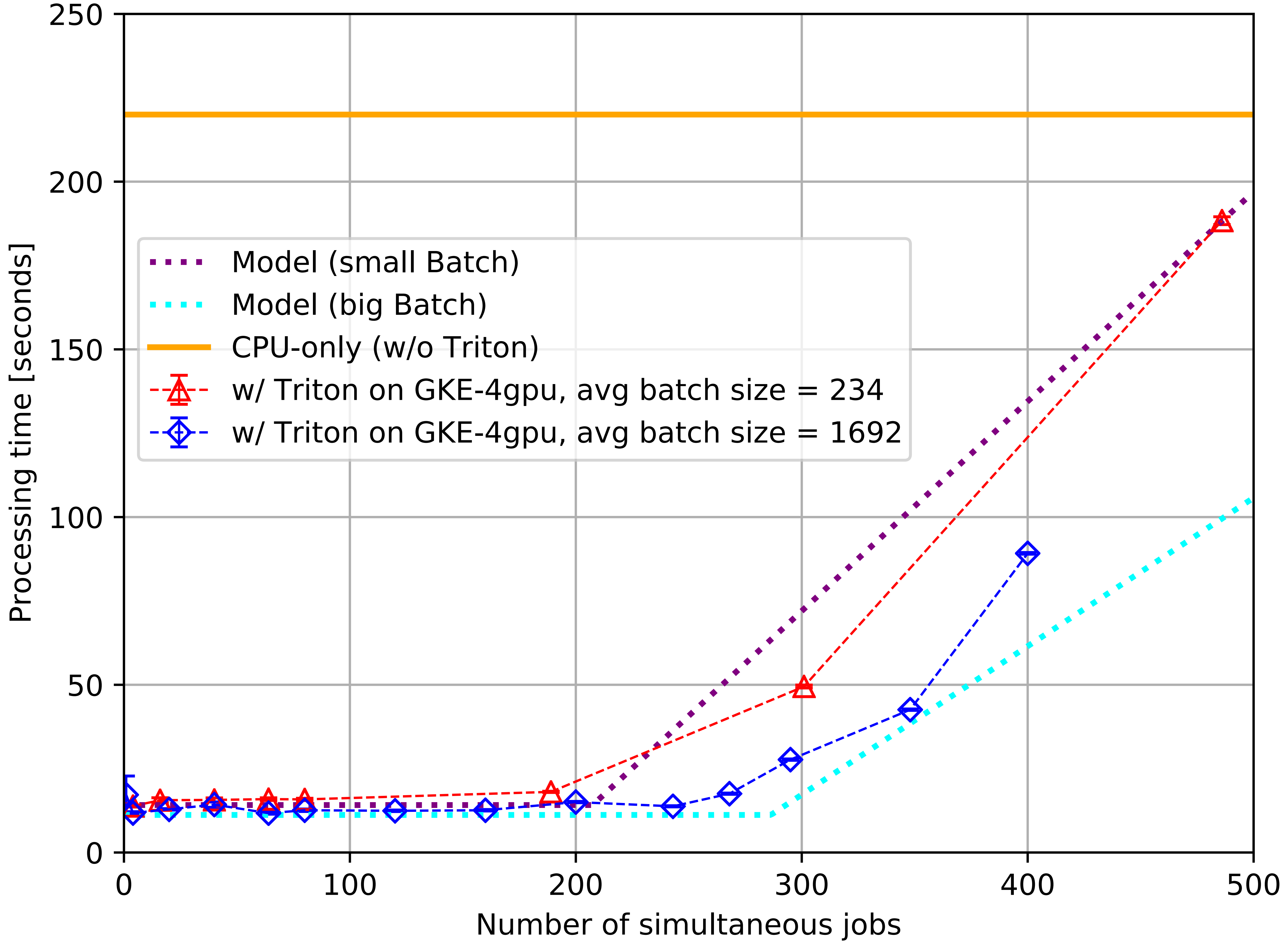
Wall time (s)		
ML module	non-ML modules	Total
~~220~~ ~11s	110	330

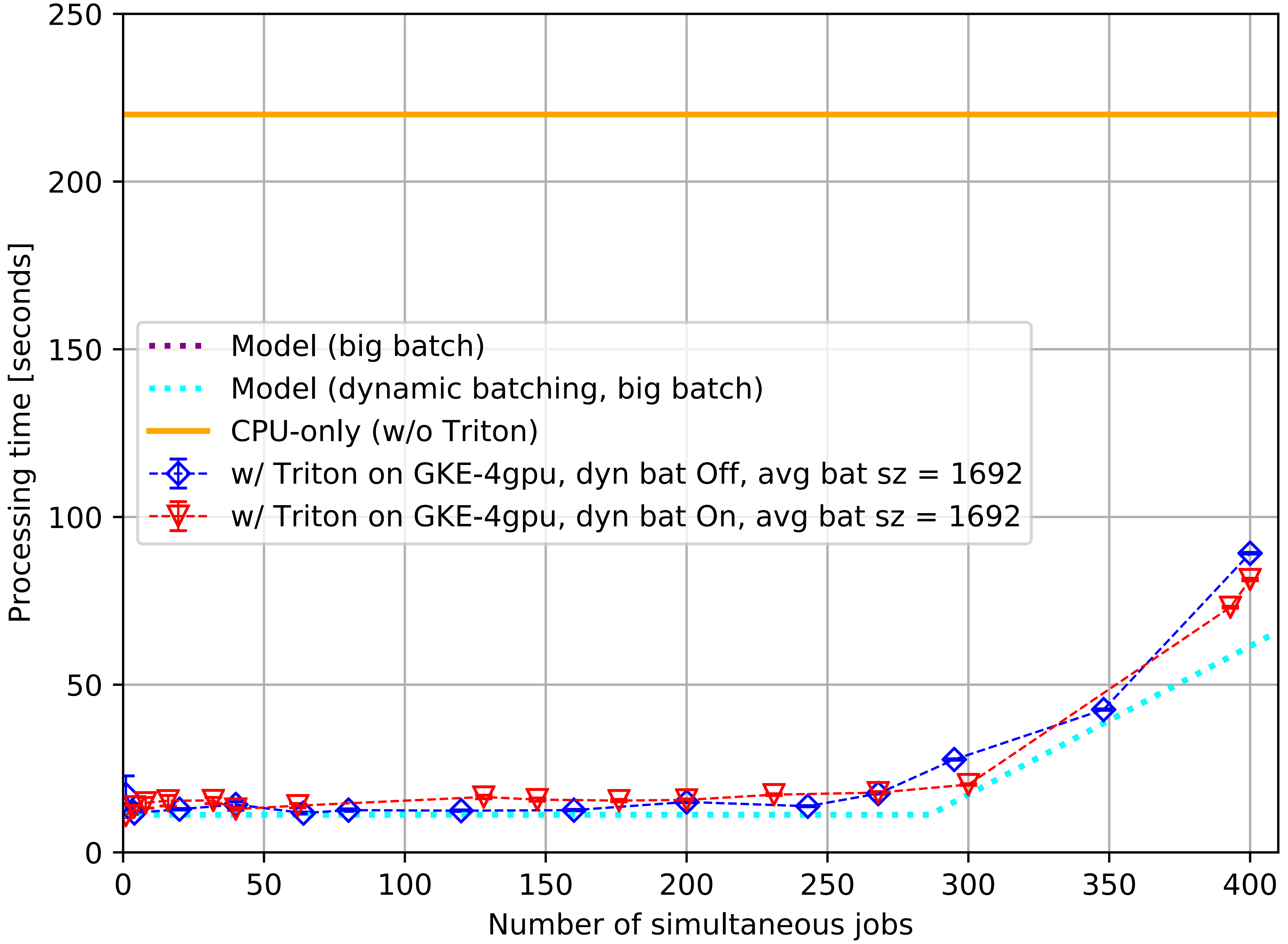
$$t_{\text{SONIC}} = (1-p) \times t_{\text{CPU}} + t_{\text{GPU}} \left[1 + \max\left(0, \underbrace{\frac{N_{\text{CPU}}}{N_{\text{GPU}}} - \frac{t_{\text{ideal}}}{t_{\text{GPU}}}}\right)\right] + t_{\text{latency}}.$$

Saturation effect:

What if N_{CPU} saturates the GPUs and they can't keep up?

Results





~20x speedup of EMMichelTrackID module

2.7x speed up of the full ProtoDUNE-SP processing chain

1 GPU can handle 68 CPU processes simulateneously

Other results

https://arxiv.org/pdf/1904.08986.pdf
https://arxiv.org/pdf/2007.10359.pdf

FPGA-accelerated machine learning inference as a service for particle physics computing

Javier Duarte · Philip Harris · Scott Hauck · Burt Holzman · Shih-Chieh Hsu · Sergo Jindariani · Suffian Khan · Benjamin Kreis · Brian Lee · Mia Liu · Vladimir Lončar · Jennifer Ngadiuba · Kevin Pedro · Brandon Perez · Maurizio Pierini · Dylan Rankin · Nhan Tran · Matthew Trahms · Aristeidis Tsaris · Colin Versteeg · Ted W. Way · Dustin Werran · Zhenbin Wu

GPU coprocessors as a service for deep learning inference in high energy physics

Jeffrey Krupa[1], Kelvin Lin[2], Maria Acosta Flechas[3], Jack Dinsmore[1], Javier Duarte[4], Philip Harris[1], Scott Hauck[2], Burt Holzman[3], Shih-Chieh Hsu[2], Thomas Klijnsma[3], Mia Liu[3], Kevin Pedro[3], Natchanon Suaysom[2], Matt Trahms[2], Nhan Tran[3,5]

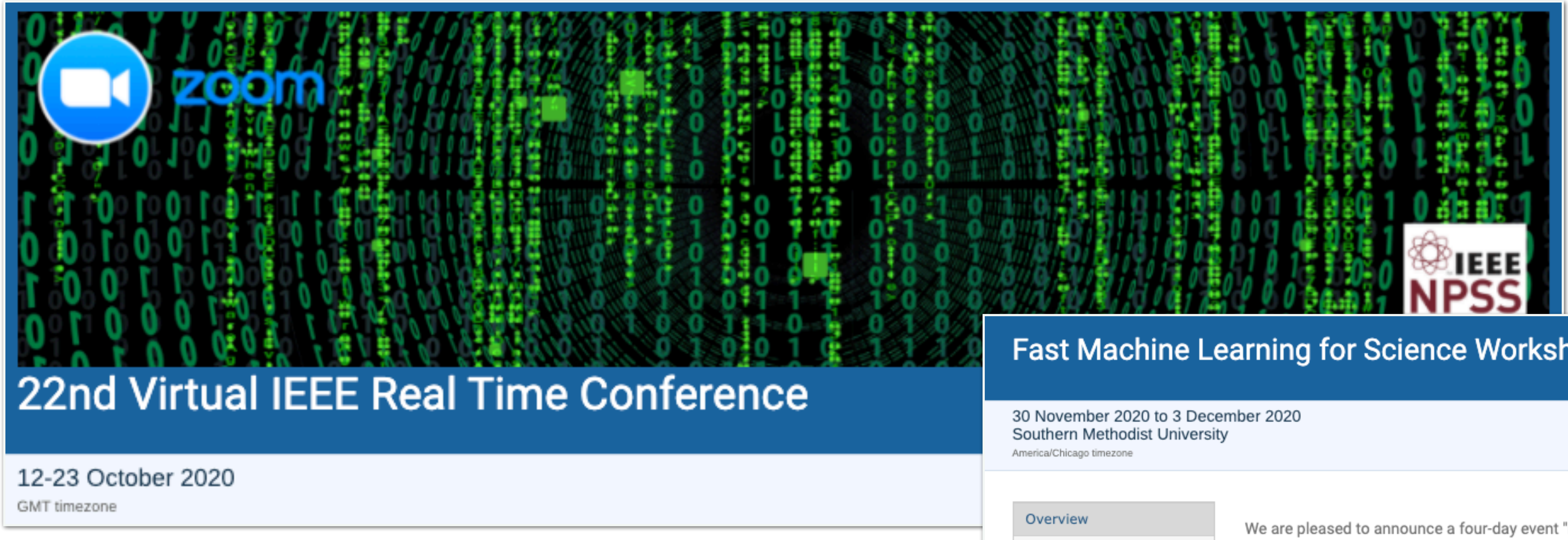
[1] Massachusetts Institute of Technology, Cambridge, MA 02139
[2] University of Washington, Seattle, WA, 98195
[3] Fermi National Accelerator Laboratory, Batavia, IL 60510
[4] University of California San Diego, La Jolla, CA 92093
[5] Northwestern University, Evanston, IL 60208

Visit Microsoft story

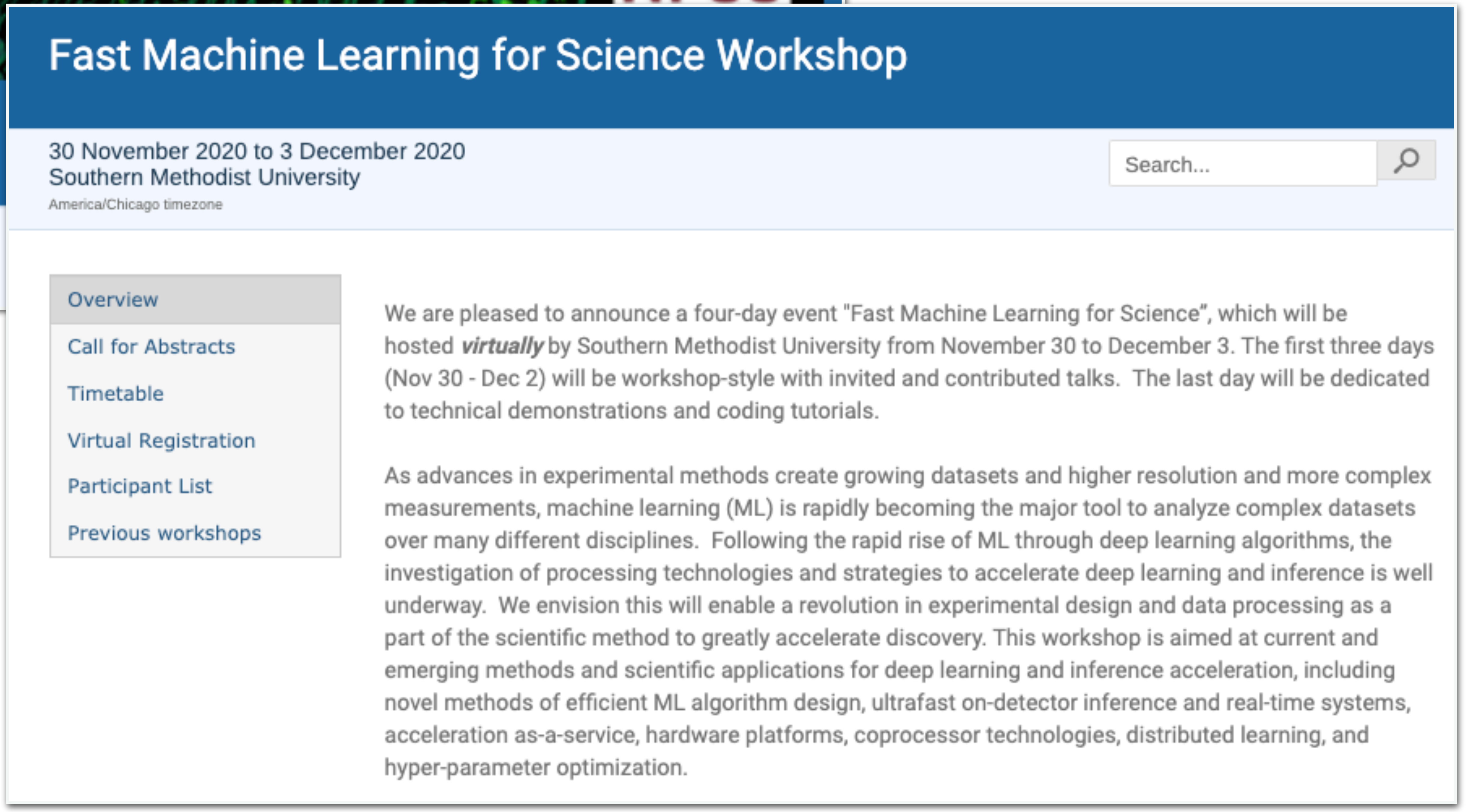
Azure Data Box Edge with Intel FPGAs installed at Fermilab

Summary and outlook

Upcoming events



https://indico.cern.ch/event/737461/



Fast Machine Learning for Science Workshop

30 November 2020 to 3 December 2020
Southern Methodist University
America/Chicago timezone

Search...

- Overview
- Call for Abstracts
- Timetable
- Virtual Registration
- Participant List
- Previous workshops

We are pleased to announce a four-day event "Fast Machine Learning for Science", which will be hosted ***virtually*** by Southern Methodist University from November 30 to December 3. The first three days (Nov 30 - Dec 2) will be workshop-style with invited and contributed talks. The last day will be dedicated to technical demonstrations and coding tutorials.

As advances in experimental methods create growing datasets and higher resolution and more complex measurements, machine learning (ML) is rapidly becoming the major tool to analyze complex datasets over many different disciplines. Following the rapid rise of ML through deep learning algorithms, the investigation of processing technologies and strategies to accelerate deep learning and inference is well underway. We envision this will enable a revolution in experimental design and data processing as a part of the scientific method to greatly accelerate discovery. This workshop is aimed at current and emerging methods and scientific applications for deep learning and inference acceleration, including novel methods of efficient ML algorithm design, ultrafast on-detector inference and real-time systems, acceleration as-a-service, hardware platforms, coprocessor technologies, distributed learning, and hyper-parameter optimization.

https://indico.cern.ch/event/924283/

Both events will have hls4ml tutorials!

Getting involved

fastmachinelearning.org

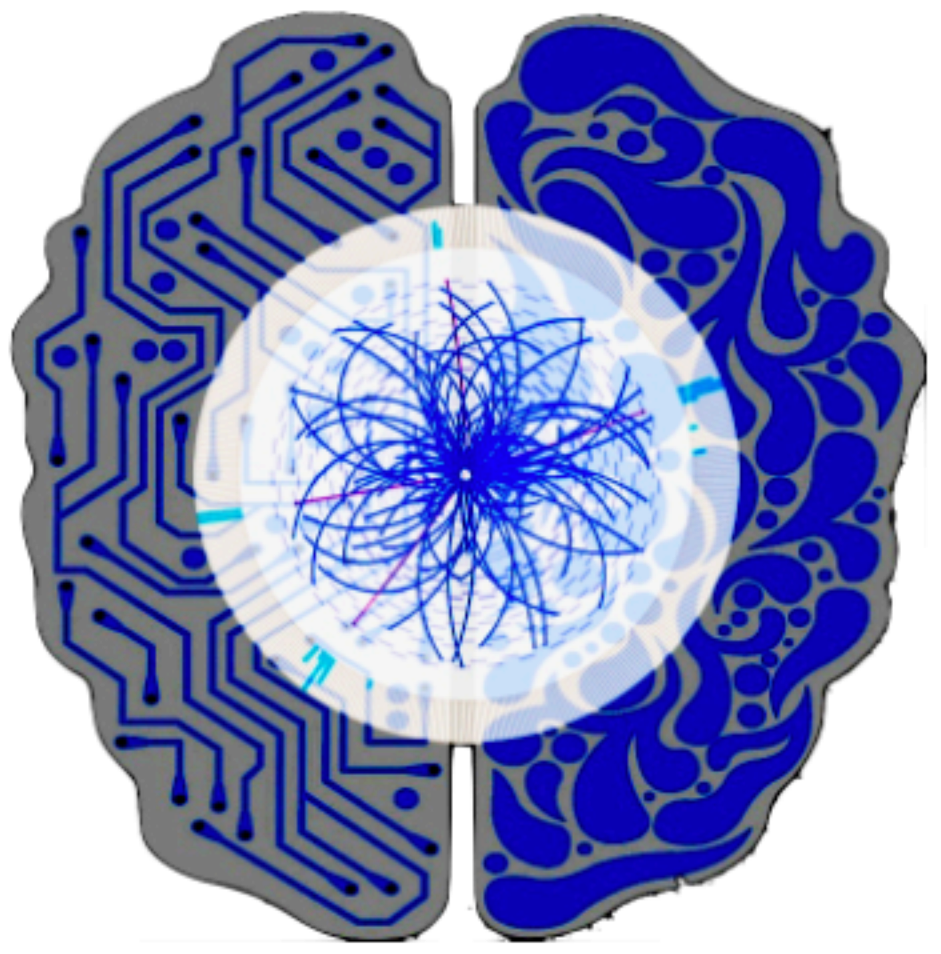
HOMEPAGE PROJECTS PEOPLE COLLABORATION CONTACT US

FAST MACHINE LEARNING LAB

ABOUT THE FAST ML LAB

Real-time and accelerated ML for fundamental sciences

Fast ML Lab is a research collective of physicists, engineers, and computer scientists interested in deploying machine learning algorithms for unique and challenging scientific applications. Our projects range from real-time, on-detector and low latency machine learning applications to high-throughput heterogeneous computing big data challenges. We are interested in deploying sophisticated machine learning algorithms to advance the exploration of fundamental physics from the world's biggest colliders to the most intense particle beams to the cosmos.



Contact me if interested!

Many other applications

As advances in experimental methods create growing datasets and higher resolution and more complex measurements, machine learning (ML) is rapidly becoming the major tool to analyze complex datasets over many different disciplines. Following the rapid rise of ML through deep learning algorithms, the investigation of processing technologies and strategies to accelerate deep learning and inference is well underway. We envision this will enable a revolution in experimental design and data processing as a part of the scientific method to greatly accelerate discovery.

Scientific interest and collaborations:

Microscopy/Spectroscopy

Accelerator controls, superconducting magnet diagnostics

RF signal processing

Cosmic surveys and gravitational wave astronomy

...

Summary

- Particle physics presents unique big data and real-time processing challenges to deliver fundamental science
- Technology is advanced by solving the impossible!

- Machine learning brings significant promise to accelerate physics discoveries
  - From operations and control to experimental design and the scientific process to improving our data simulation and reconstruction to our understanding of underlying physics principles
- The confluence of physics, detectors, and computing will play an important role in moving physics experimentation forward

“bonus”

–Johnny Appleseed

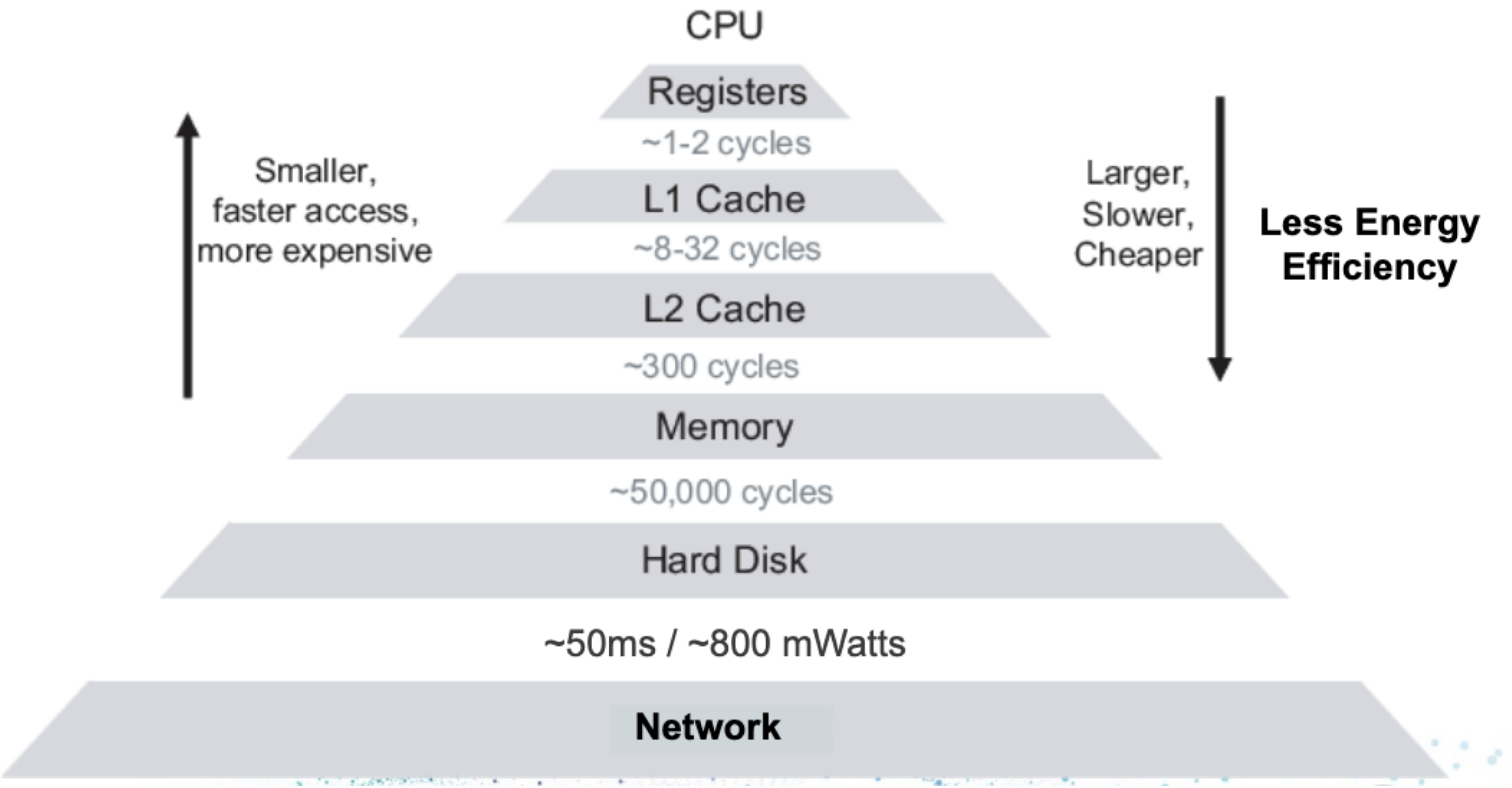
7

Power Dominated by Data Movement

Operation:	Energy (pJ)	Area (μm^2)
8b Add	0.03	36
16b Add	0.05	67
32b Add	0.1	137
16b FP Add	0.4	1360
32b FP Add	0.9	4184
8b Mult	0.2	282
32b Mult	3.1	3495
16b FP Mult	1.1	1640
32b FP Mult	3.7	7700
32b SRAM Read (8KB)	5	N/A
32b DRAM Read	640	N/A

Memory access is **orders of magnitude** higher energy than compute

Vivienne Sze (@eems_mit) *[Horowitz, ISSCC 2014]* MIT


CPU
Registers
~1-2 cycles
L1 Cache
~8-32 cycles
L2 Cache
~300 cycles
Memory
~50,000 cycles
Hard Disk
~50ms / ~800 mWatts
Network
Smaller,
faster access,
more expensive
Larger,
Slower,
Cheaper
Less Energy
Efficiency