FPGAs as a Service to Accelerate Machine Learning Inference

LPC Topic of the Week
April 15, 2019

Fermilab
Javier Duarte
Burt Holzman
Sergo Jindariani
Benjamin Kreis
Mia Liu
Kevin Pedro
Nhan Tran
Aristeidis Tsaris

Massachusetts Institute of Technology
Philip Harris
Dylan Rankin

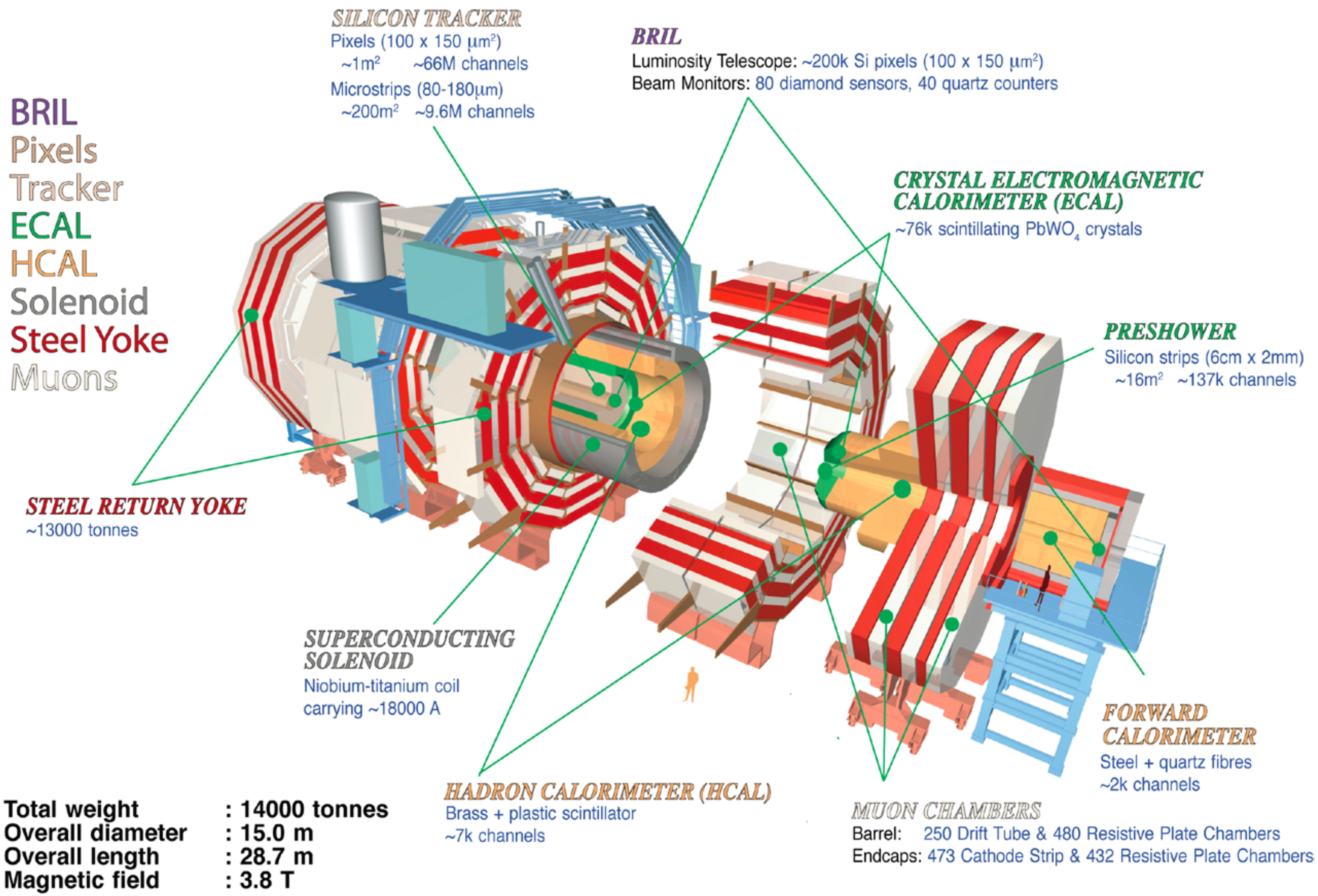
University of Washington
Scott Hauck
Shih-Chieh Hsu
Matthew Trahms
Dustin Werran

UIC University of Illinois at Chicago
Zhenbin Wu

Microsoft
Suffian Khan
Brandon Perez
Colin Versteeg
Ted W. Way

CERN
Vladimir Loncar
Jennifer Ngadiuba
Maurizio Pierini

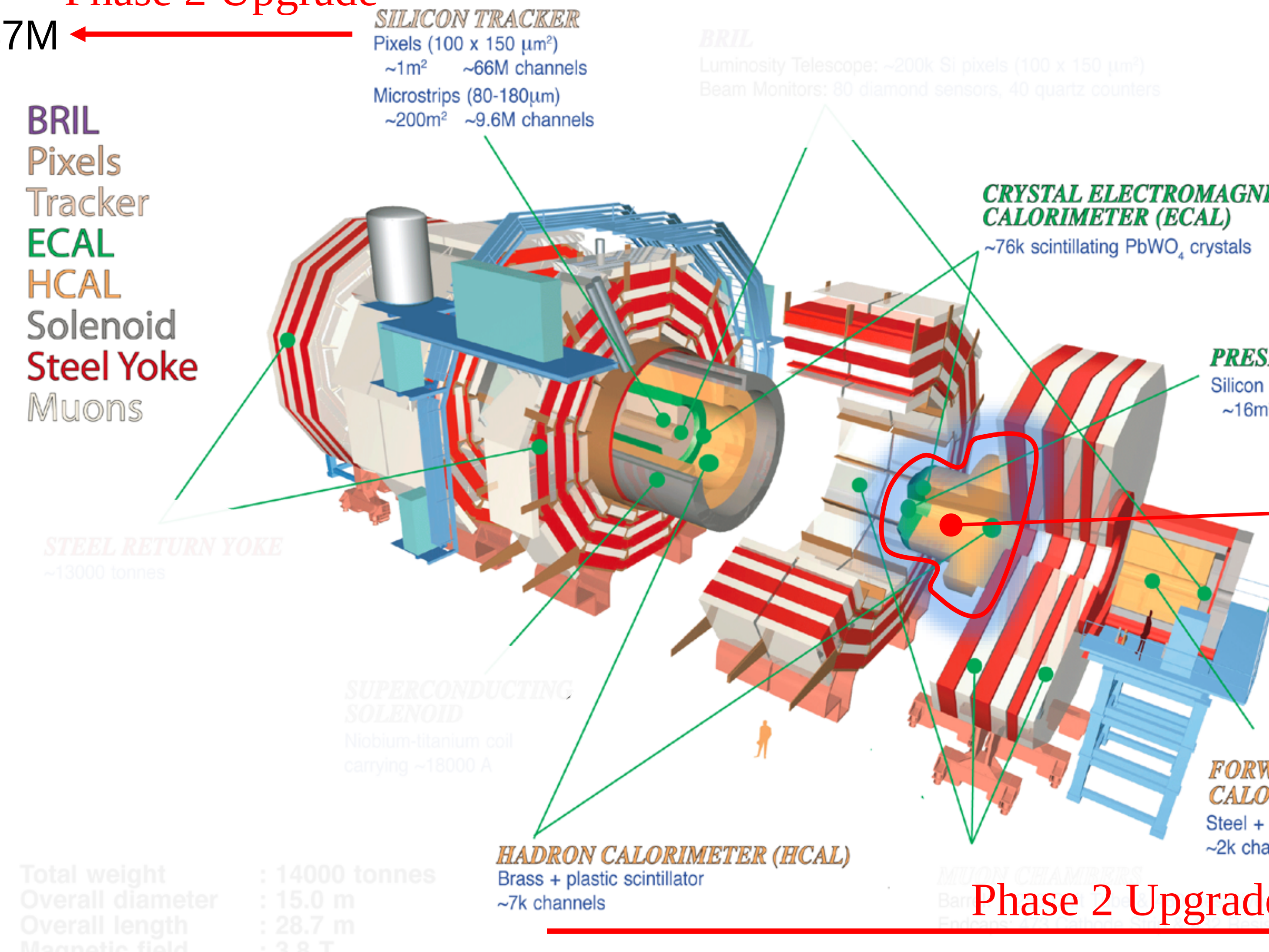
The CMS Detector: Phase 0



The CMS Detector: Phase 2

Phase 2 Upgrade

1947M



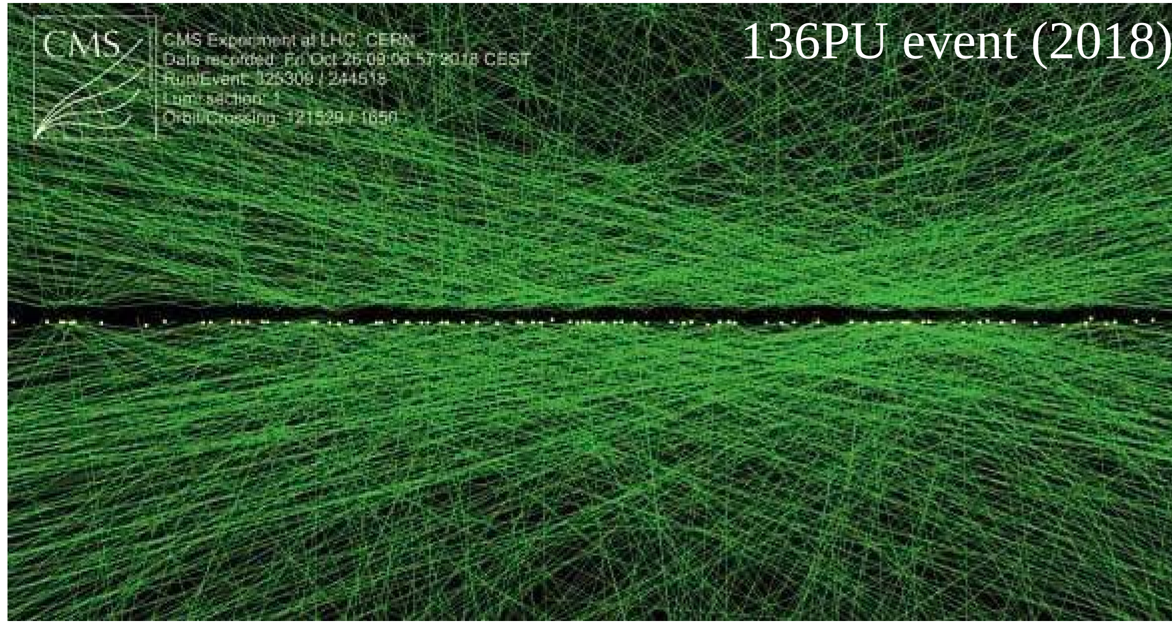
High Granularity Calorimeter (HGCal)
Silicon, scintillator
~6M channels

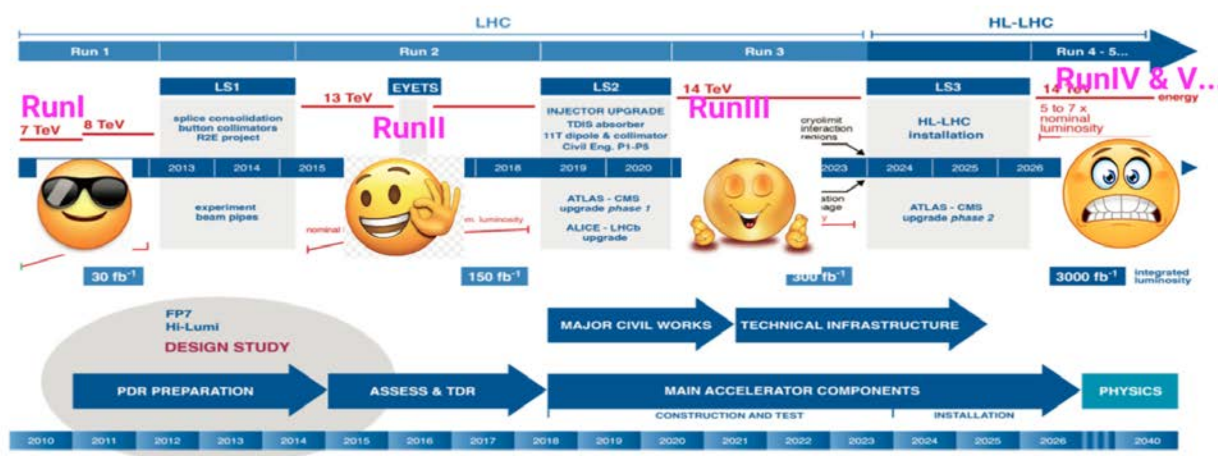
Phase 2 Upgrade

CMS Computing Challenges

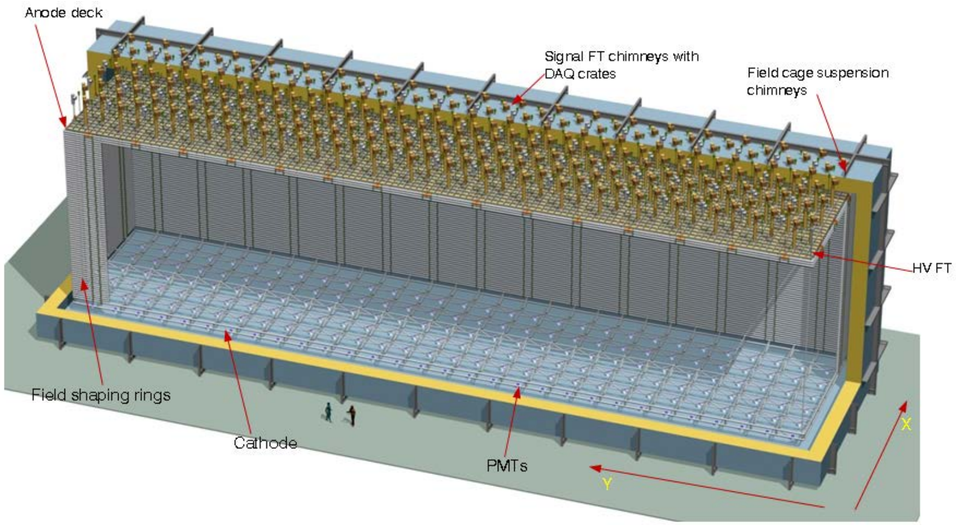
Energy frontier: HL-LHC

- 10× data vs. Run 2/3
  → exabytes
- 200PU
  (vs. ~30PU in Run 2)
- CMS:
  - 15× increase in pixel channels
  - 65× increase in calorimeter channels
  - (similar for ATLAS)



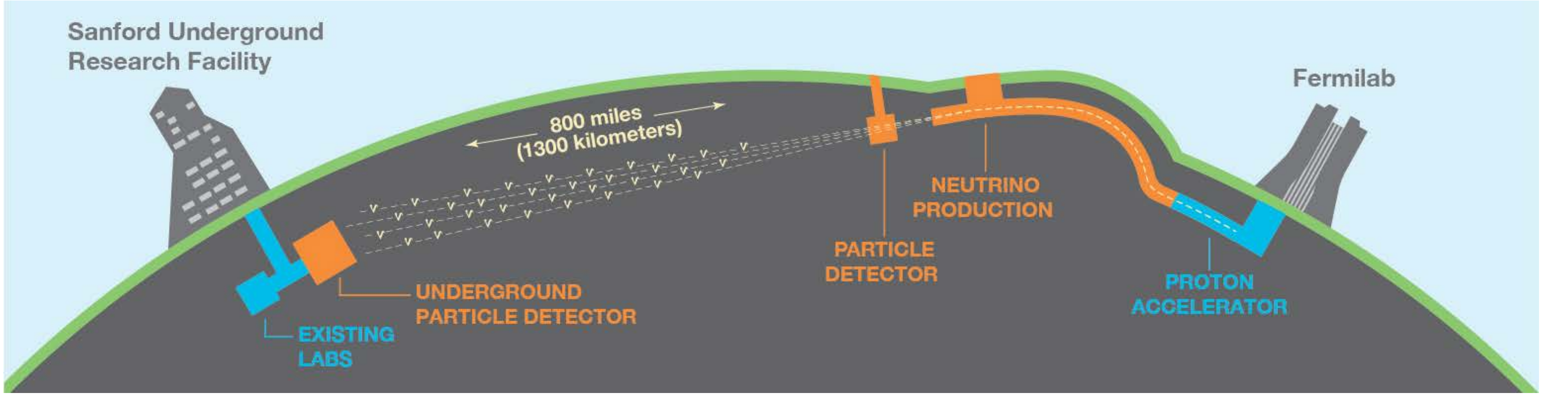


Neutrino Computing Challenges



Intensity frontier: DUNE

- Largest liquid argon detector ever designed
- ~1M channels, 1 ms integration time w/ MHz sampling
  → 30+ petabytes/year



➢ CPU needs for particle physics will increase by *more than an order of magnitude* in the next decade

How Bad Is It?

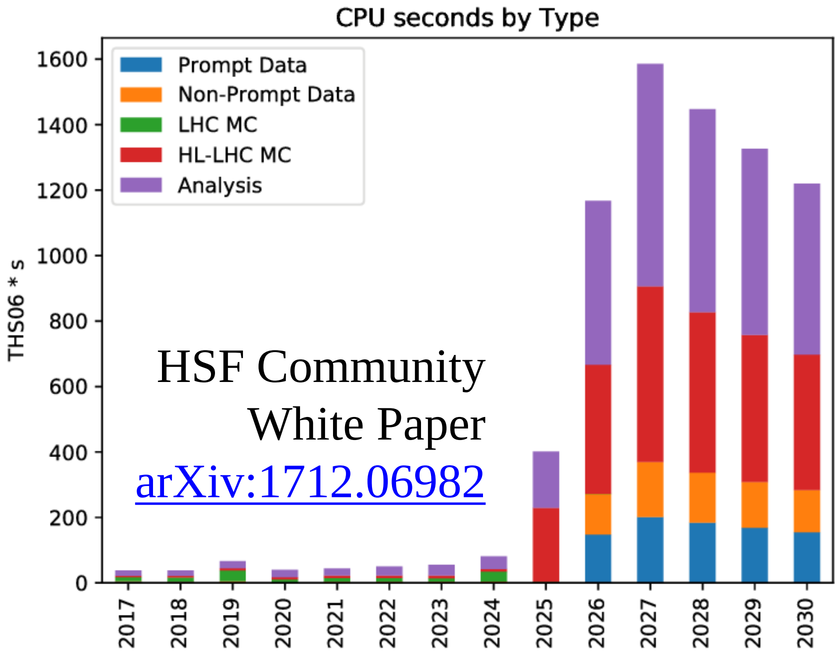
- CMS does extensive resource projections for HL-LHC
- Expected needs for 2027 (HL-LHC startup):
  - 1,400,000 CPU cores
  - 2.2 exabytes disk, 3 exabytes tape
- Usually project 20%/year “technology improvement” from Moore’s law

➢ Still a shortfall of 2–5×

- More realistic: 10%/year

➢ Shortfall of **4–10×**

- Try to *thrive*, not just survive

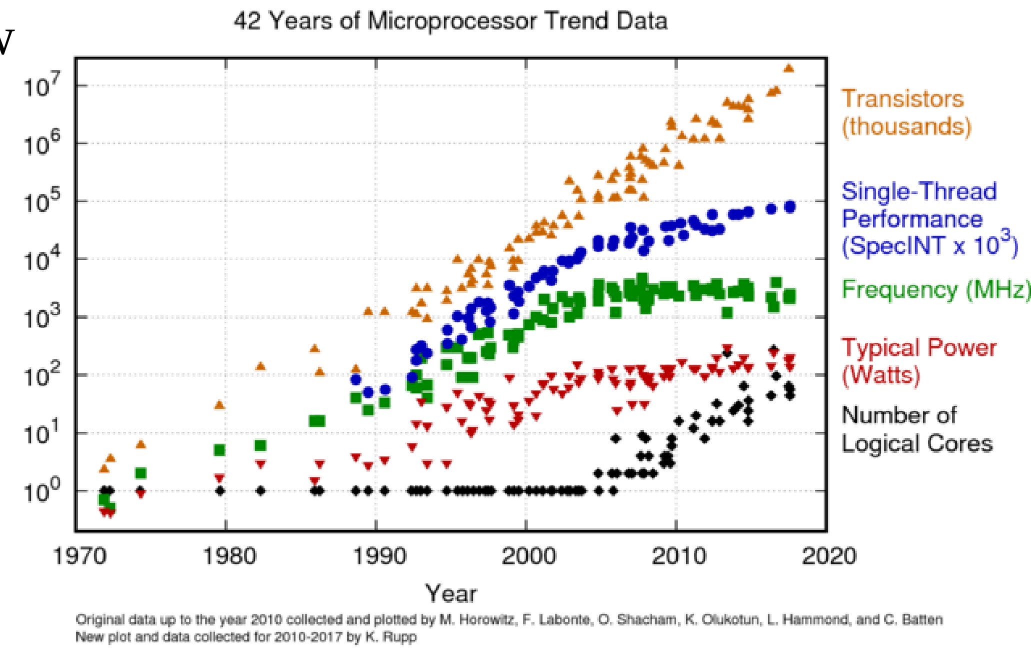


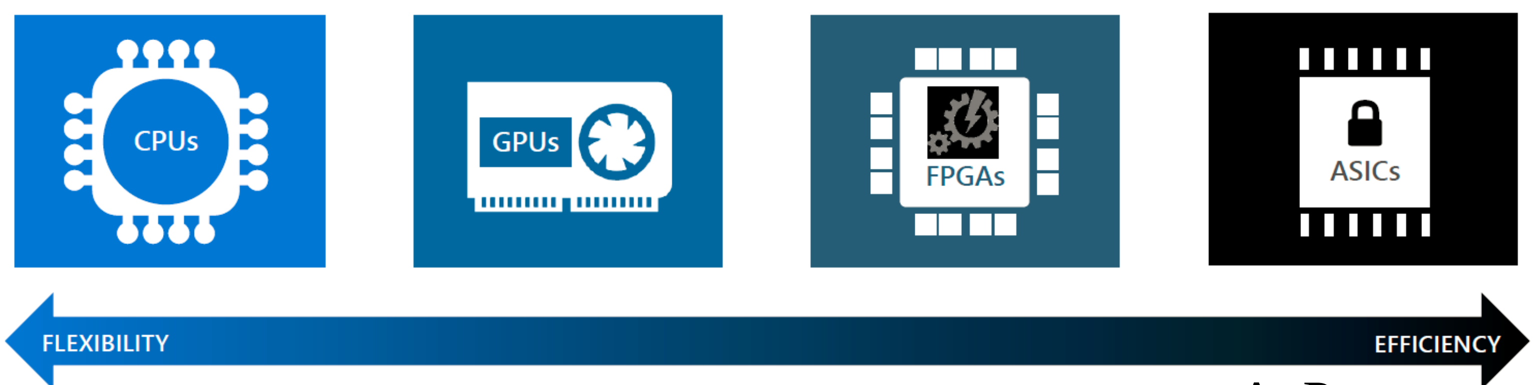
HSF Community White Paper
arXiv:1712.06982

More at CMSOfflineComputingResults

The Computing Landscape

- Transistor counts continue to grow
- Clock speed and single-thread performance have stagnated
  - ➢ *No longer expect* **traditional CPUs** to keep up with demands from particle physics
  - ➢ **Coprocessors**: *specialized hardware* attached to CPU, dedicated to *specific tasks*





A. Putnam

Development for Coprocessors

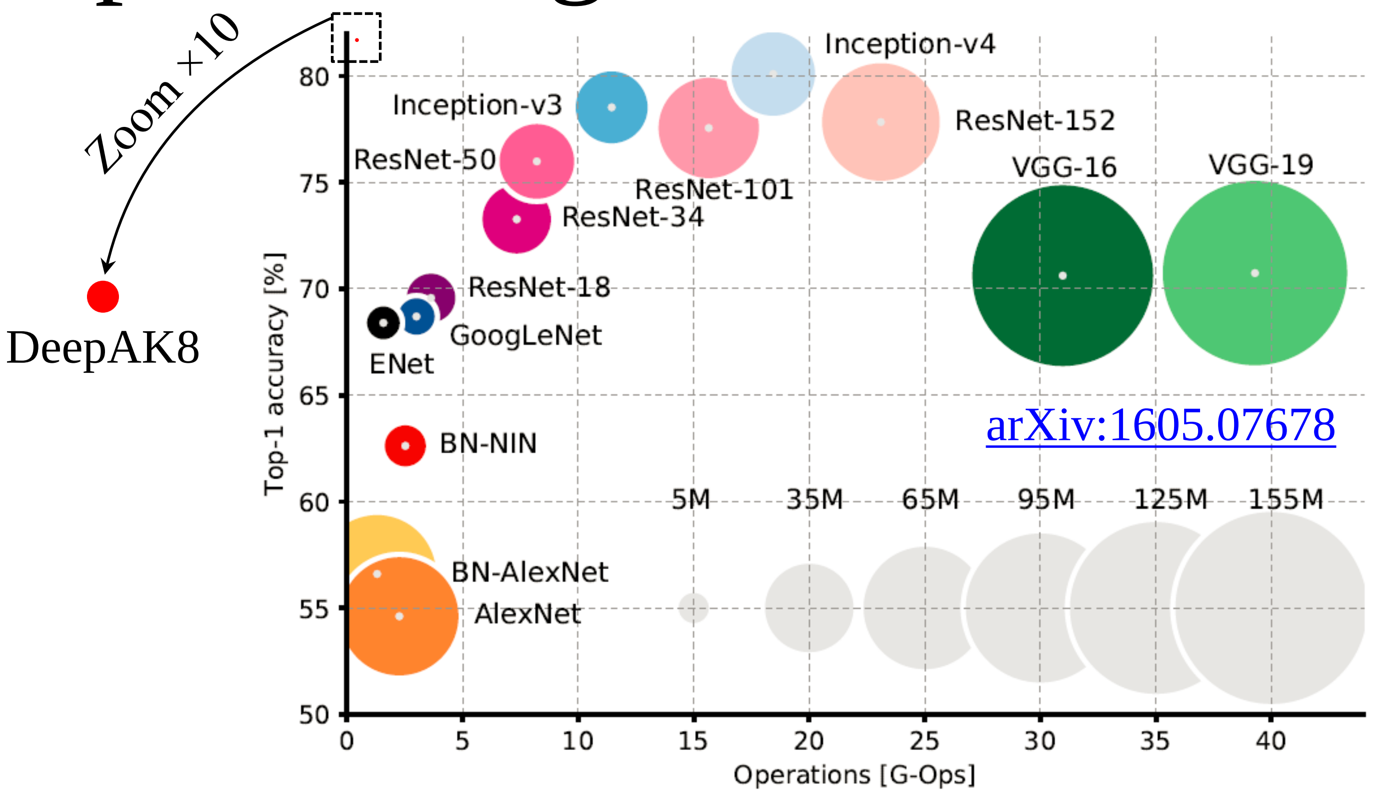
- Large speed improvement from hardware accelerated coprocessors
  - Architectures and tools are geared toward **machine learning**

Option 1	**Option 2**
re-write physics algorithms for new hardware	**re-cast physics problem as machine learning problem**
Language: OpenCL, OpenMP, HLS, CUDA, …?	Language: C++, Python (TensorFlow, PyTorch,…)
Hardware: FPGA, GPU	Hardware: FPGA, GPU, ASIC

Why (Deep) Machine Learning?

- Common *language* for solving problems: simulation, reconstruction, analysis!
- Can be universally expressed on optimized computing hardware (follow industry trends)

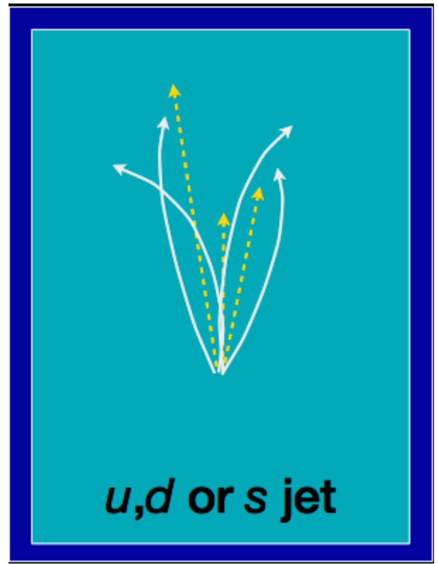
Deep Learning in Science and Industry

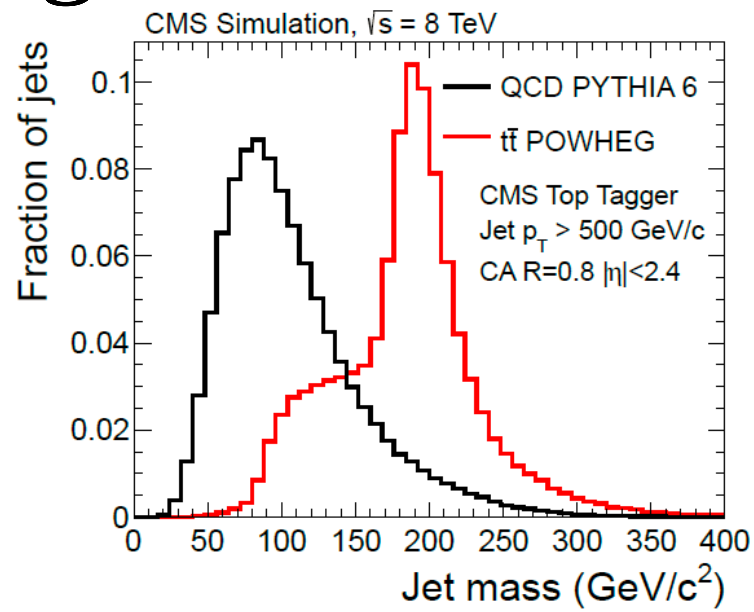


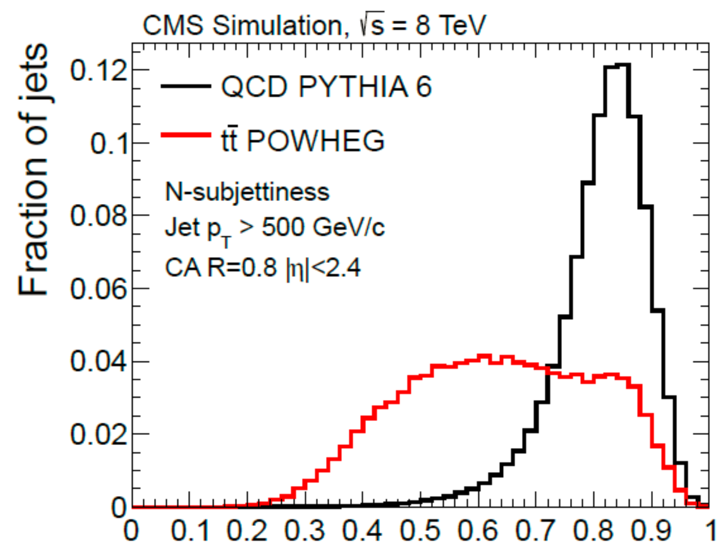
[arXiv:1605.07678](https://arxiv.org/abs/1605.07678)

- ResNet-50: 25M parameters, 7B operations
- Largest network currently used by CMS:
  - DeepAK8, 500K parameters, 15M operations
- Newer approaches w/ larger networks in development…

Top Tagging



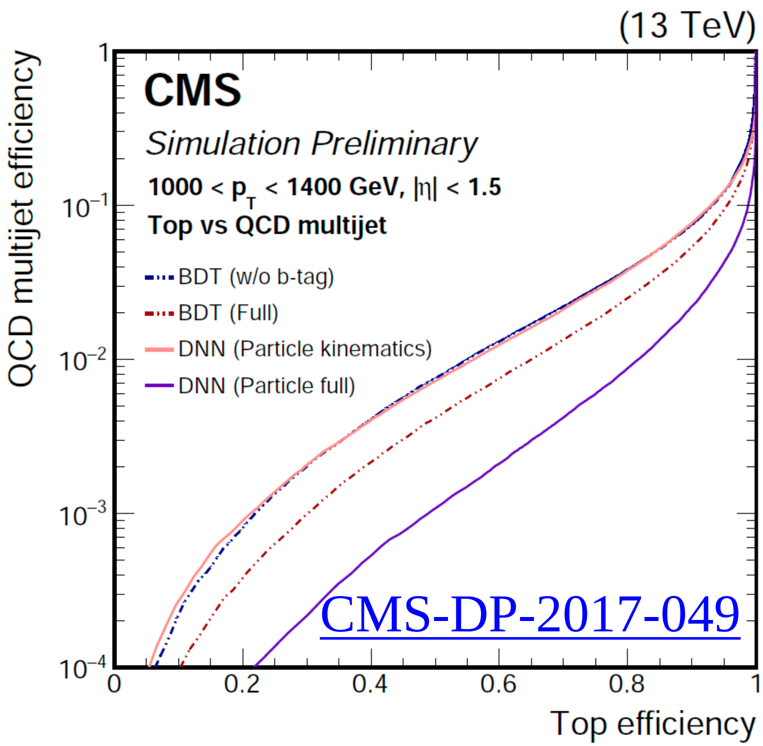




- High p_T top quarks are boosted: form a single large-radius jet with substructure
- Top tagging started as simple cuts on high-level variables (right)
- Now advanced to particle-level deep neural networks (next slide)
- (Can also do Higgs tagging, W/Z tagging, etc.)

CMS-PAS-JME-17-003

Top Tagging with Deep Learning

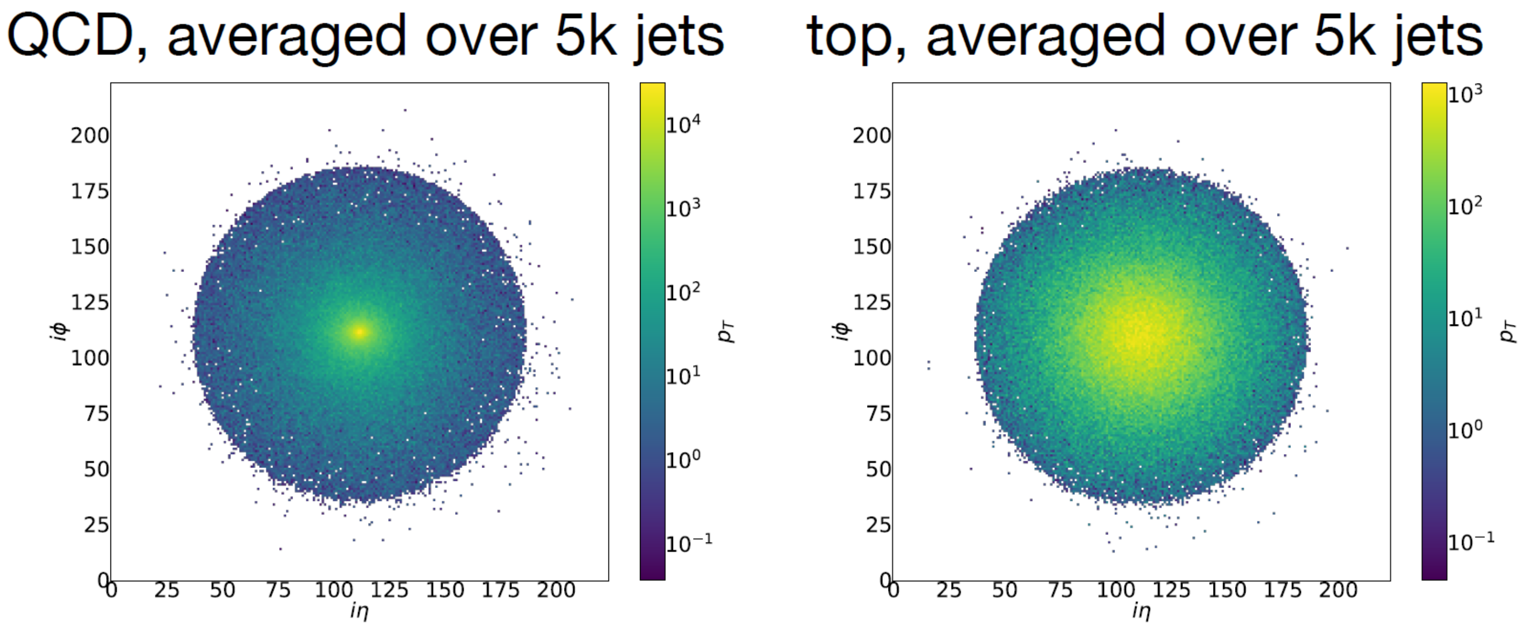


- DNNs outperform simpler ML algorithms (such as BDTs)
- Many approaches now developed
  - Some use much larger networks than DeepAK8: Particle cloud, ResNet-like
- Metrics: AUC, accuracy, $1/\varepsilon_B$ @ ε_S=30%

Approach	AUC	Acc.	1/eB (@ eS=0.3)	Contact	Comments
LoLa	0.980	0.928	680	GK / Simon Leiss	Preliminary number, based on LoLa
LBN	0.981	0.931	863	Marcel Rieger	Preliminary number
CNN	0.981	0.93	780	David Shih	Model from *Pulling Out All the Tops with Computer Vision and Deep Learning (1803.00107)*
P-CNN (1D CNN)	0.980	0.930	782	Huilin Qu, Loukas Gouskos	Preliminary, use kinematic info only (https://indico.physics.lbl.gov/indico/event/546/contributions/1270/)
6-body N-subjettiness (+mass and pT) NN	0.979	0.922	856	Karl Nordstrom	Based on 1807.04769 (*Reports of My Demise Are Greatly Exaggerated: N-subjettiness Taggers Take On Jet Images*)
8-body N-subjettiness (+mass and pT) NN	0.980	0.928	795	Karl Nordstrom	Based on 1807.04769 (*Reports of My Demise Are Greatly Exaggerated: N-subjettiness Taggers Take On Jet Images*)
Linear EFPs	0.980	0.932	380	Patrick Komiske, Eric Metodiev	d<= 7, chi <= 3 EFPs with FLD. Based on 1712.07124: *Energy Flow Polynomials: A complete linear basis for jet substructure.*
Particle Flow Network (PFN)	0.982	0.932	888	Patrick Komiske, Eric Metodiev	Median over ten trainings. Based on Table 5 in 1810.05165: *Energy Flow Networks: Deep Sets for Particle Jets.*
Energy Flow Network (EFN)	0.979	0.927	619	Patrick Komiske, Eric Metodiev	Median over ten trainings. Based on Table 5 in 1810.05165: *Energy Flow Networks: Deep Sets for Particle Jets.*
2D CNN [ResNeXt50]	0.984	0.936	1086	Huilin Qu, Loukas Gouskos	Preliminary from indico.cern.ch/event/745718/contributions/3202526
DGCNN	0.984	0.937	1160	Huilin Qu, Loukas Gouskos	Preliminary from indico.cern.ch/event/745718/contributions/3202526

G. Kasieczka

Top Tagging with Images



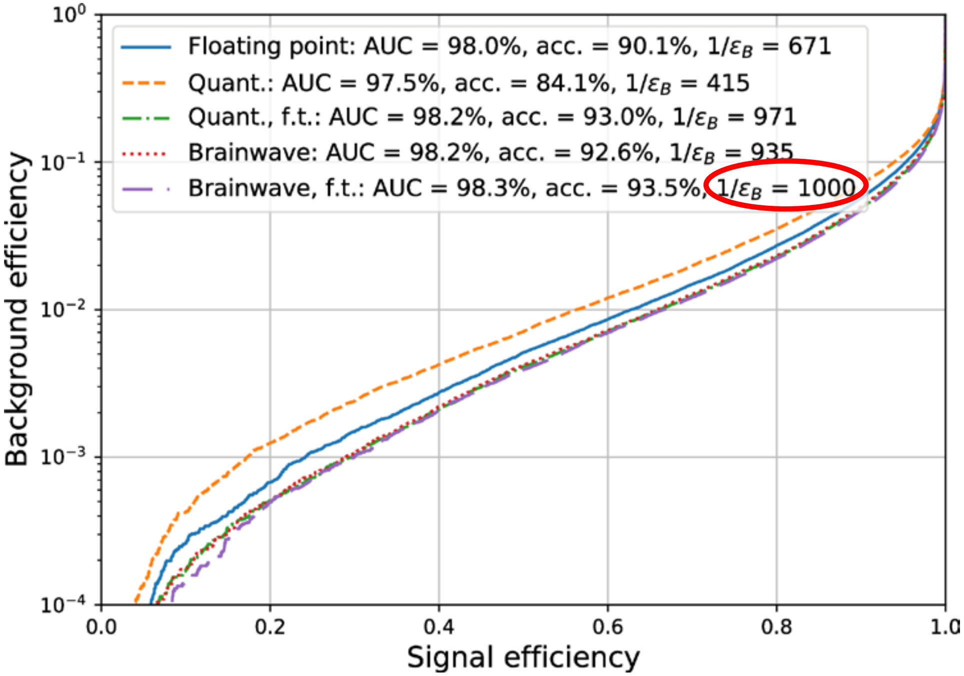
- Deep learning in industry focuses on image recognition
- Jets are not images, but can pretend in order to test industry networks
- Convert jets into images using constituent p_T, η, φ: 224x224 pixels
- Standardized top quark tagging dataset is publicly available: https://goo.gl/XGYju3

Top Tagging with ResNet-50

- Retrain ResNet-50 on publicly available top quark tagging dataset

  → New set of weights, optimized for physics

  - Add custom classifier layers to interpret features from ResNet-50

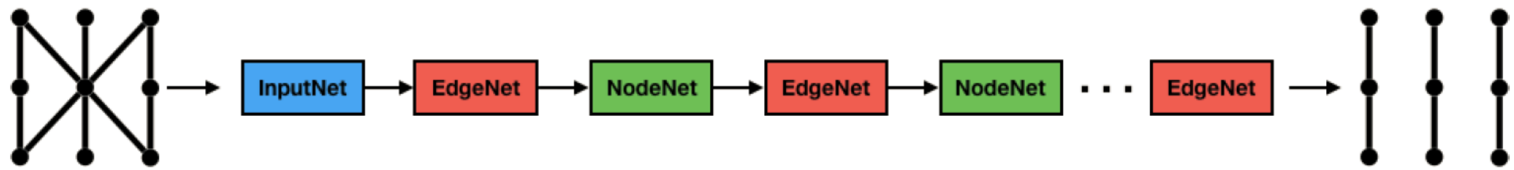


- ResNet-50 model that runs on FPGAs is "quantized"

  - Tune weights to achieve similar performance

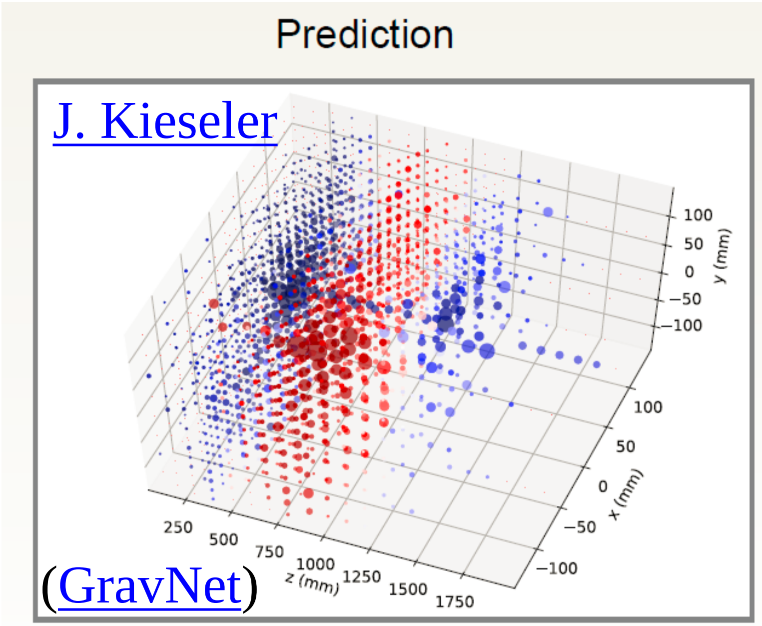
➤ State-of-the-art results vs. other leading algorithms

Beyond Tagging

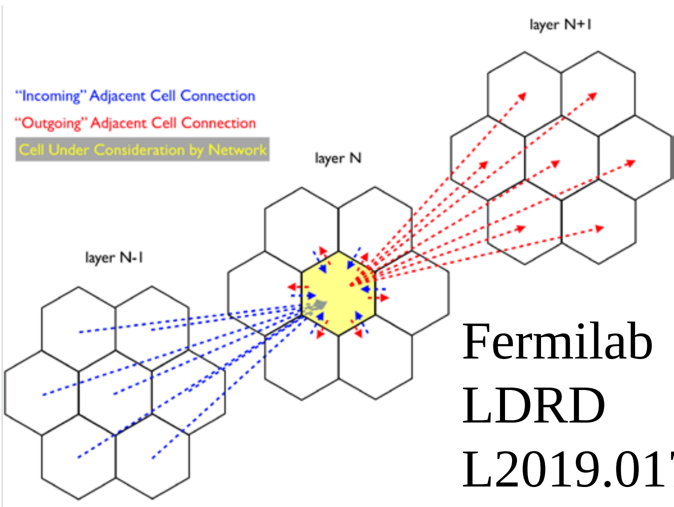


- R&D to use novel network architectures for tracking and clustering
- Optimal use of complex new detectors
- May be crucial for Phase 2

[HEP.TrkX](#)
Message Passing Neural Network
Graph-based ML for tracking

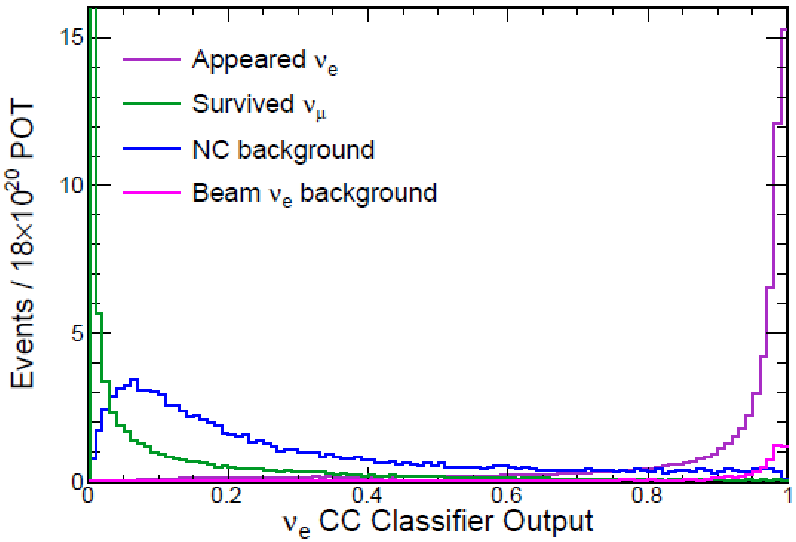


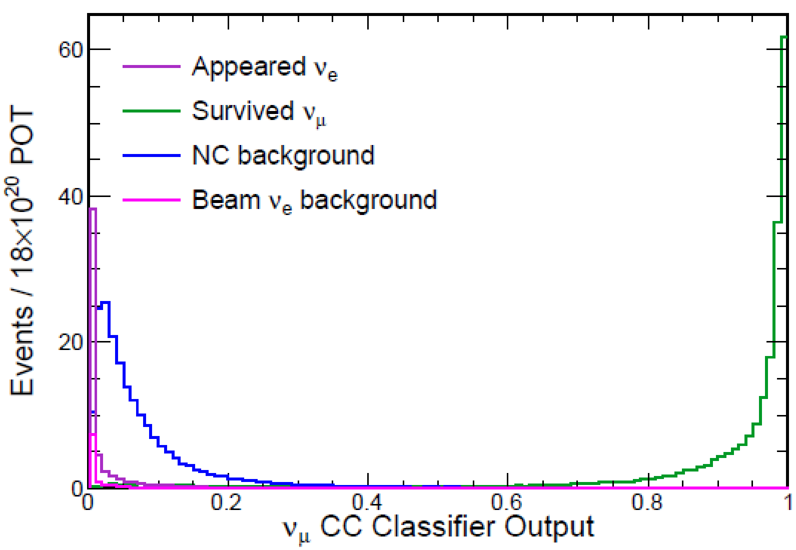
MPNN for HGCal
Exploit hexagonal cells & 3D structure



Fermilab LDRD L2019.017

NoVA: First Particle Physics CNN

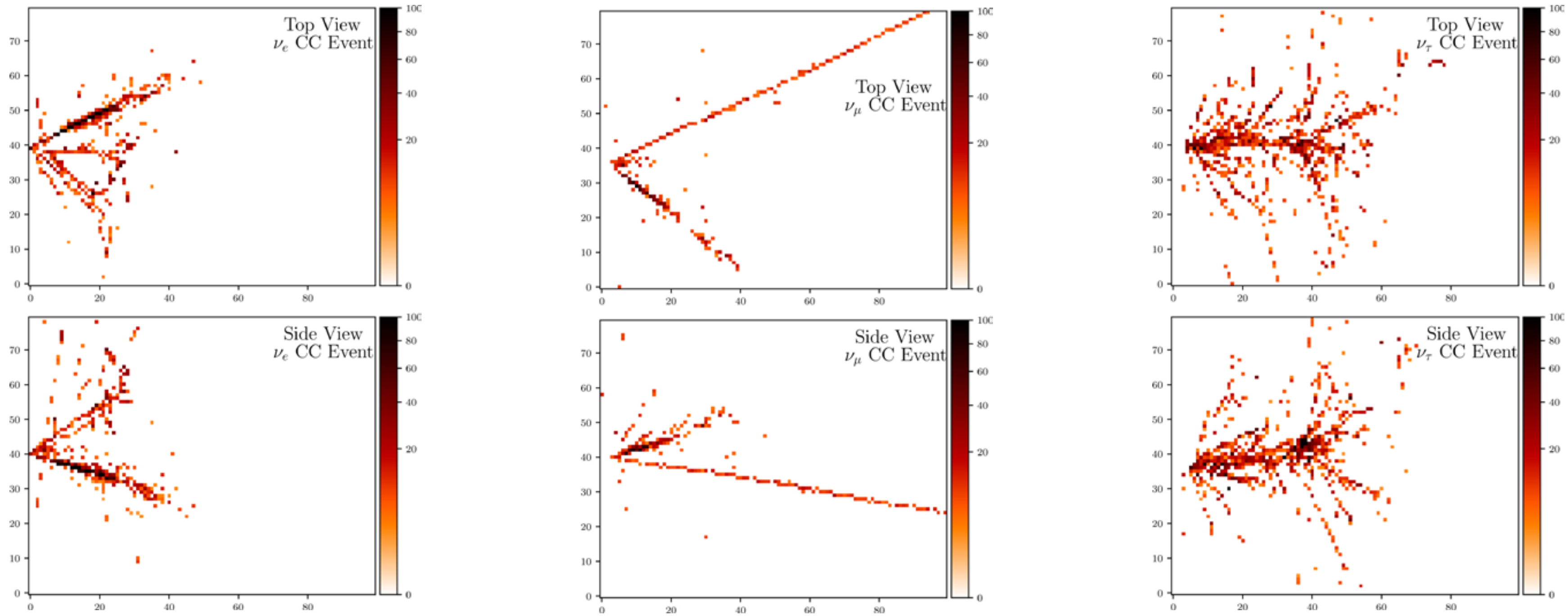




- NOvA was the first particle physics experiment to publish a result obtained using a CNN
- Achieved similar efficiency for ν_μ (58% vs. 57%), improvement for ν_e (49% vs. 35%) vs. existing algorithms (arXiv:1604.01444)
- Used to constrain oscillation parameters: (arXiv:1703.03328)

➢ Inverted mass hierarchy w/ θ_{23} in lower octant excluded at 93% CL for all δ_{CP} values

Neutrino Recognition with ResNet-50



- ResNet-50 can also classify neutrino events to reject cosmic ray backgrounds
- Use *transfer learning*: keep default featurizer weights, retrain classifier layers
- Events above selected w/ probability > 0.9 in different categories
- CNN inference already a large fraction of neutrino reconstruction time

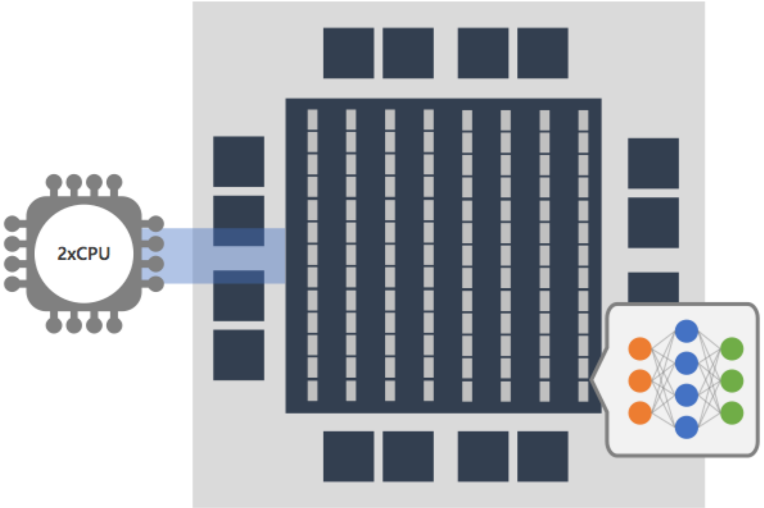
➢ Prime candidate for acceleration with coprocessors

Why Accelerate Inference?

- Training is viewed as "hard" part of machine learning
- But…
- DNN training happens ~once/year/algorithm
    - Cloud GPUs or new HPCs are good options
- Once DNN is in common use, inference will happens *billions* of times
    - MC production, analysis, prompt reconstruction, high level trigger…

➢ Training is done by developers, inference is done by everyone

Inference as a Service

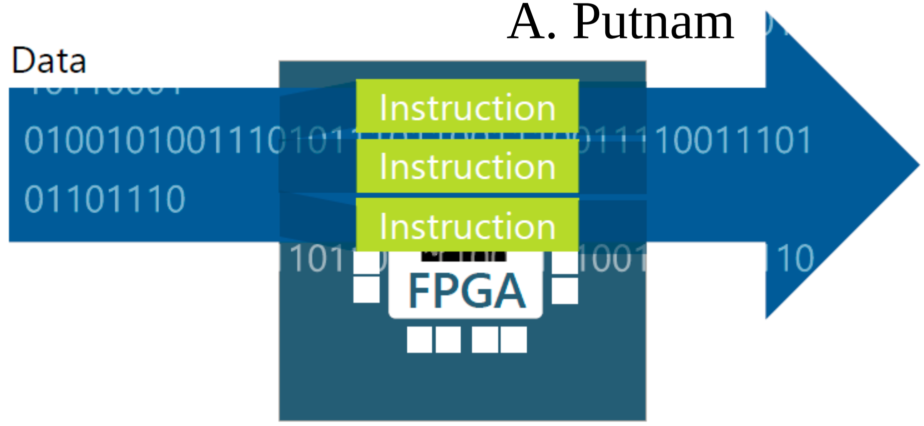


- Inference as a service:
  - Minimize disruption to existing computing model
  - Minimize dependence on specific hardware
  - Avoid need for large input batches
    - Large batches are necessary to use GPUs efficiently
    - Not a good fit for most particle physics applications: prefer to load one (complex) event into memory, then run all algorithms on it
- Performance metrics:
  - Latency (time for a single request to complete)
  - Throughput (number of requests per unit time)

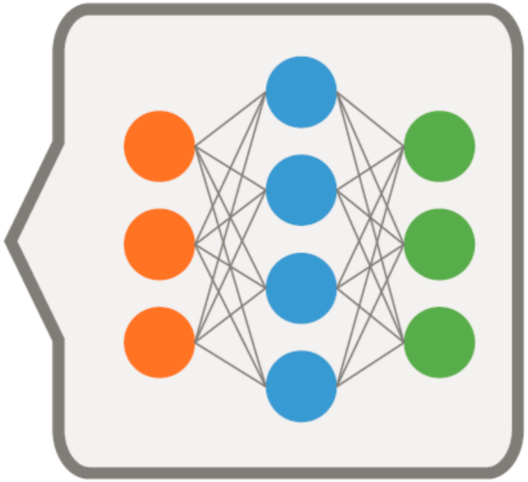
What are FPGAs?

- Field Programmable Gate Arrays
- "Programmable hardware":
  arrange gates to perform desired task
- Execute many instructions *in parallel* on input data: **spatial computing**
  - vs. CPUs, which execute instructions *in series*: **temporal computing**
- Can be *reconfigured* in 100ms–1s
- Ideal for deep networks: many operations, fixed model parameters
- ASICs (Application Specific Integrated Circuits) not configurable – require stable problem to solve

A. Putnam

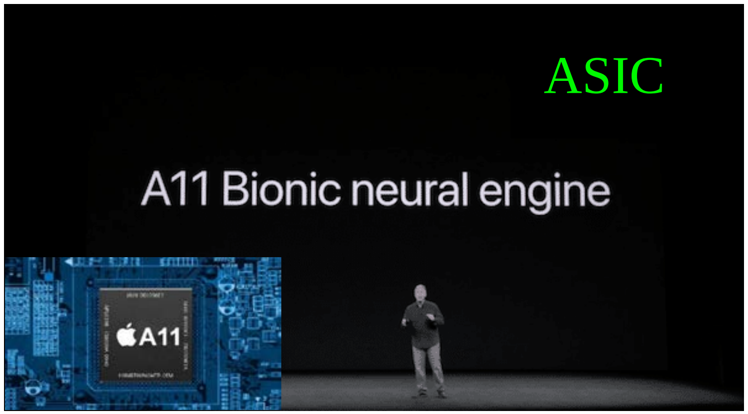


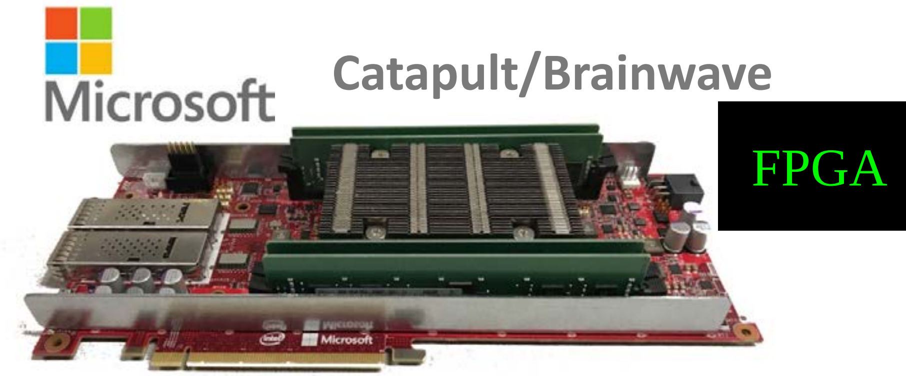
FPGA: spatial compute

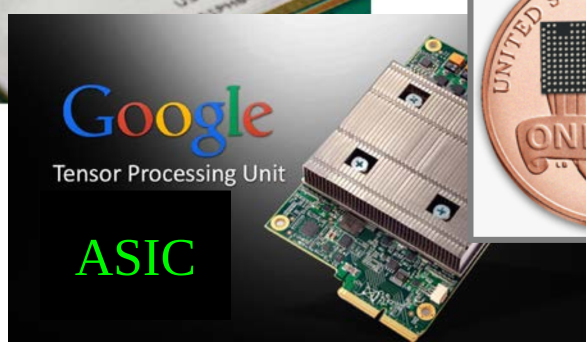


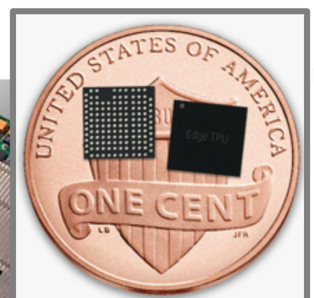
Coprocessors: An Industry Trend

Specialized coprocessor hardware for machine learning inference



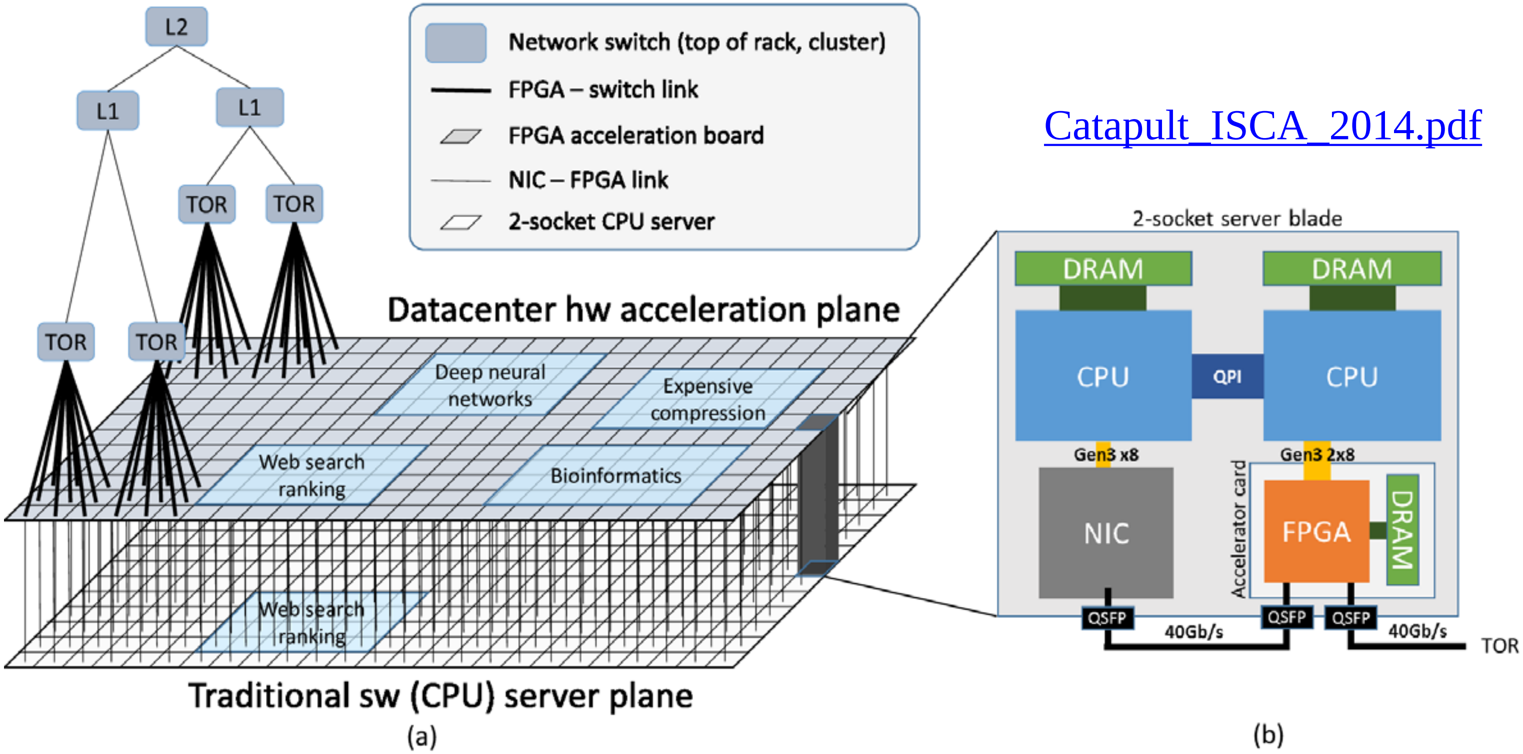






Microsoft Brainwave

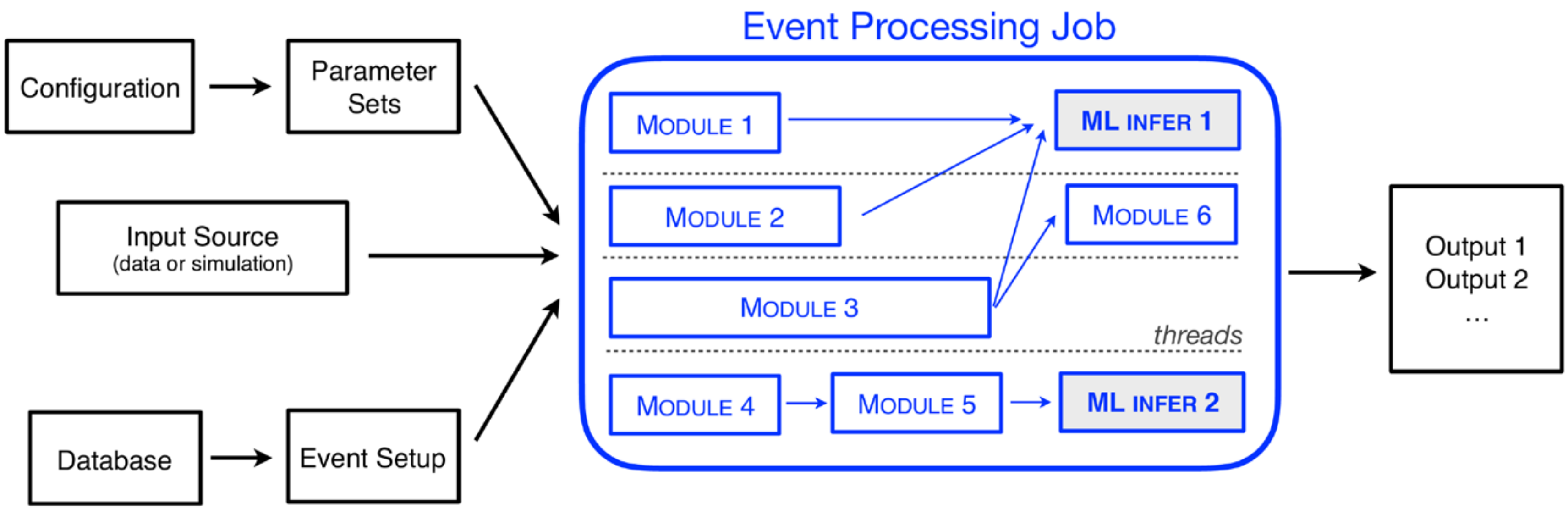
[Catapult_ISCA_2014.pdf]



- Provides a full service at scale (more than just a single co-processor)
- Multi-FPGA/CPU fabric accelerates *both* **computing** and **network**
- Weight retuning available: retrain supported networks to optimize for a different problem

Brainwave supports:
- ResNet50
- ResNet152
- DenseNet121
- VGGNet16

Particle Physics Computing Model

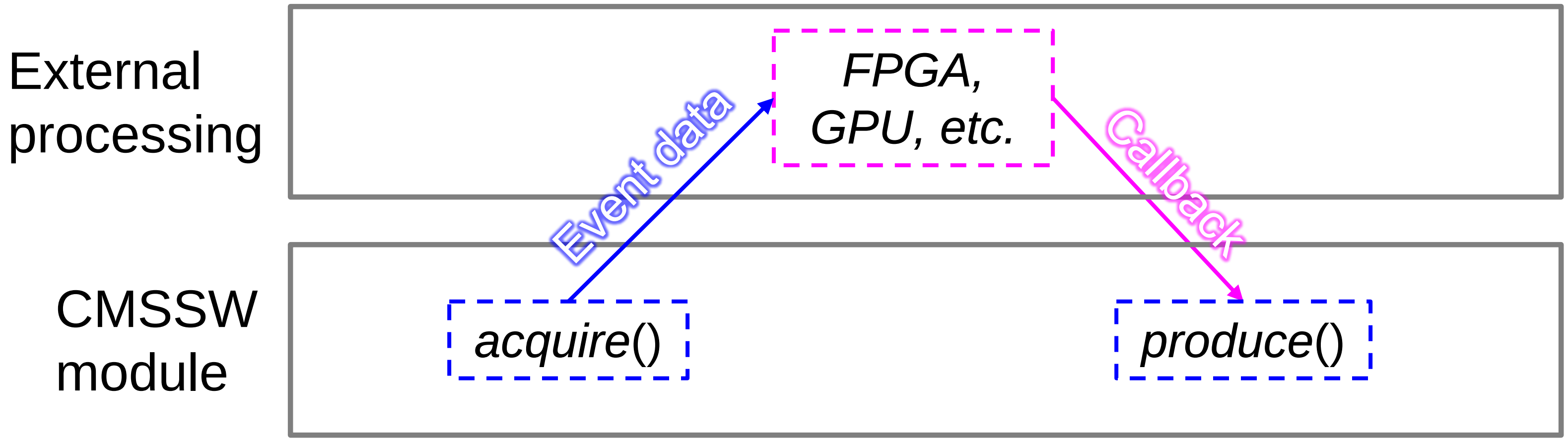


- Event-based processing
  - o Events are very complex with hundreds of products
  - o Load one event into memory, then execute all algorithms on it

➢ Most applications not a good fit for large batches, which are required for best GPU performance

Accessing Heterogeneous Resources

- New CMSSW feature called ExternalWork:
  - Asynchronous task-based processing



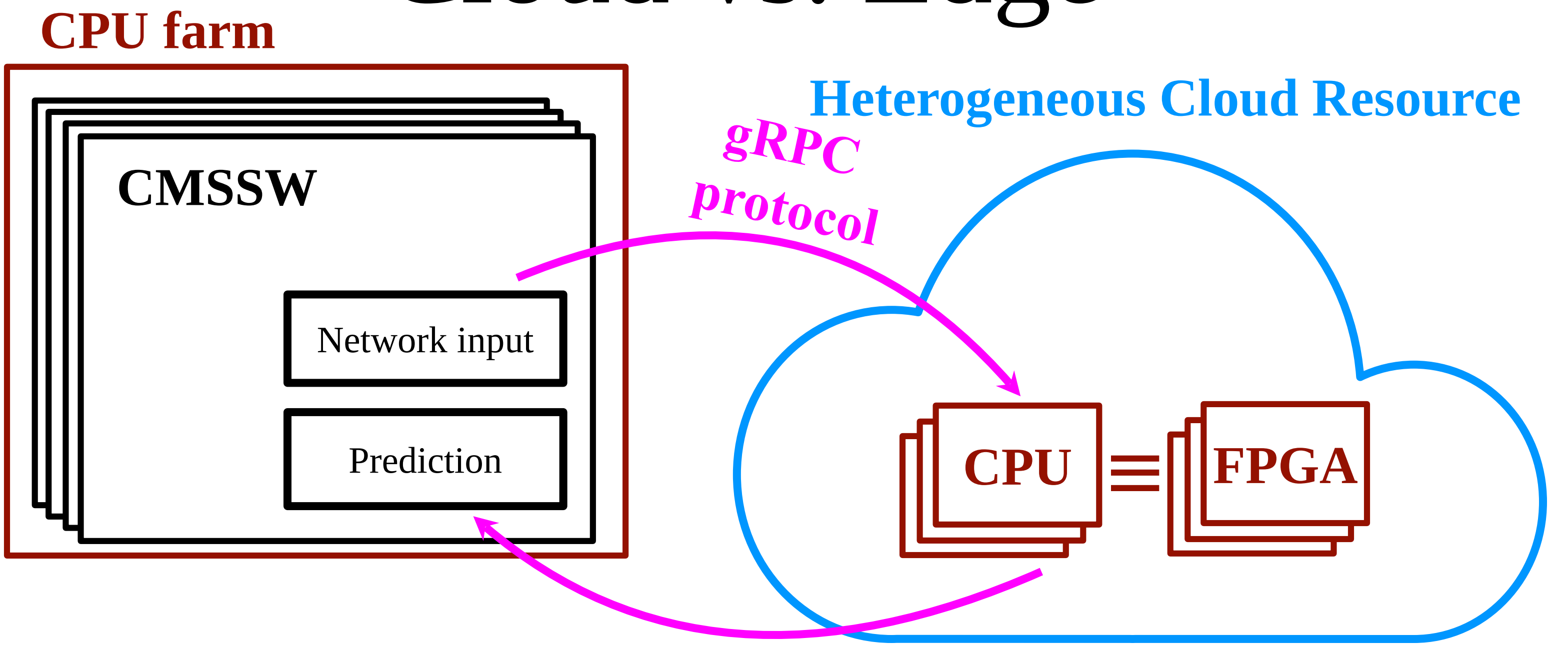
  - Non-blocking: schedule other tasks while waiting for external processing
- Can be used with GPUs, FPGAs, cloud, …
  - Even other software running on CPU that wants to schedule its own tasks

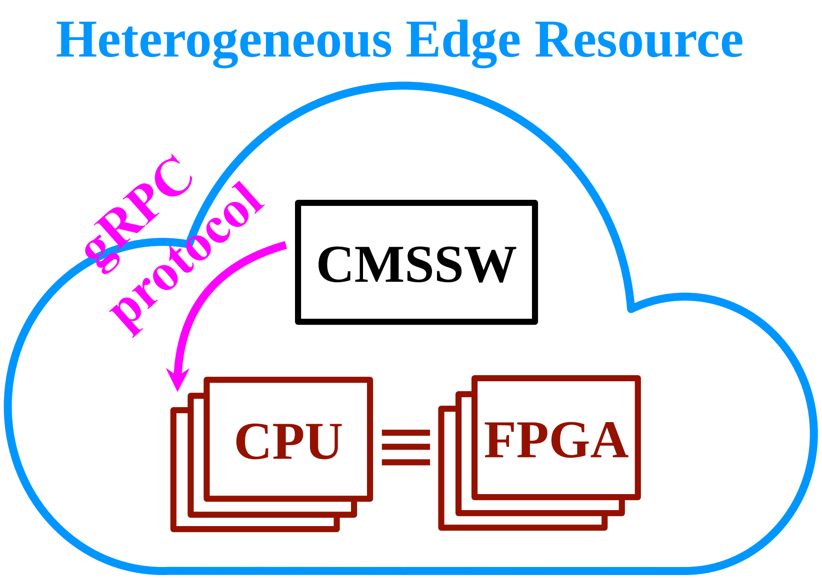
➢ Now demonstrated to work with Microsoft Brainwave!

SONIC in CMSSW

- **S**ervices for **O**ptimized **N**etwork **I**nference on **C**oprocessors
  - Convert experimental data into neural network input
  - Send neural network input to coprocessor using communication protocol
  - Use ExternalWork mechanism for asynchronous requests
- Currently supports:
  - gRPC communication protocol
    - Callback interface for C++ API in development
      → wait for return in lightweight std::thread
  - TensorFlow w/ inputs sent as TensorProto (protobuf)
- Tested w/ Microsoft Brainwave service (cloud FPGAs)
- gRPC [SonicCMS]() repository on GitHub

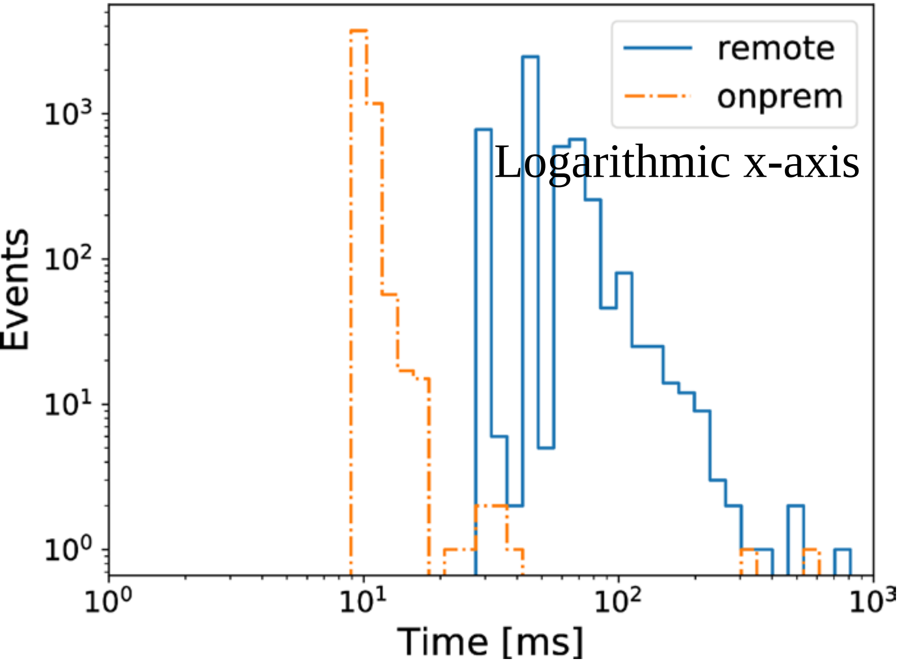
Cloud vs. Edge

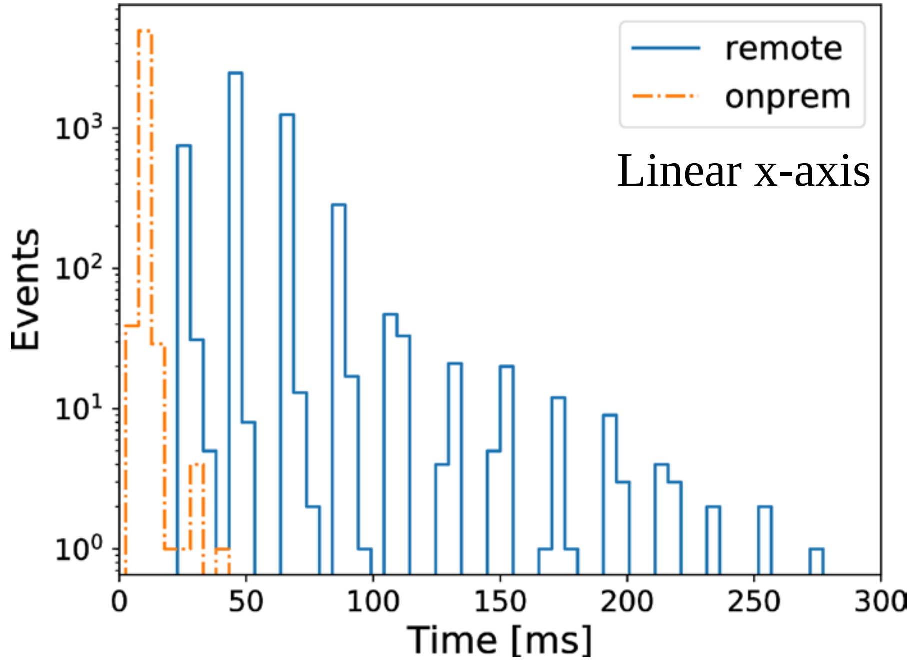




- Cloud service has latency
- Run CMSSW on Azure cloud machine → simulate local installation of FPGAs (“on-prem” or “edge”)
- Provides test of ultimate performance
- Use gRPC protocol either way

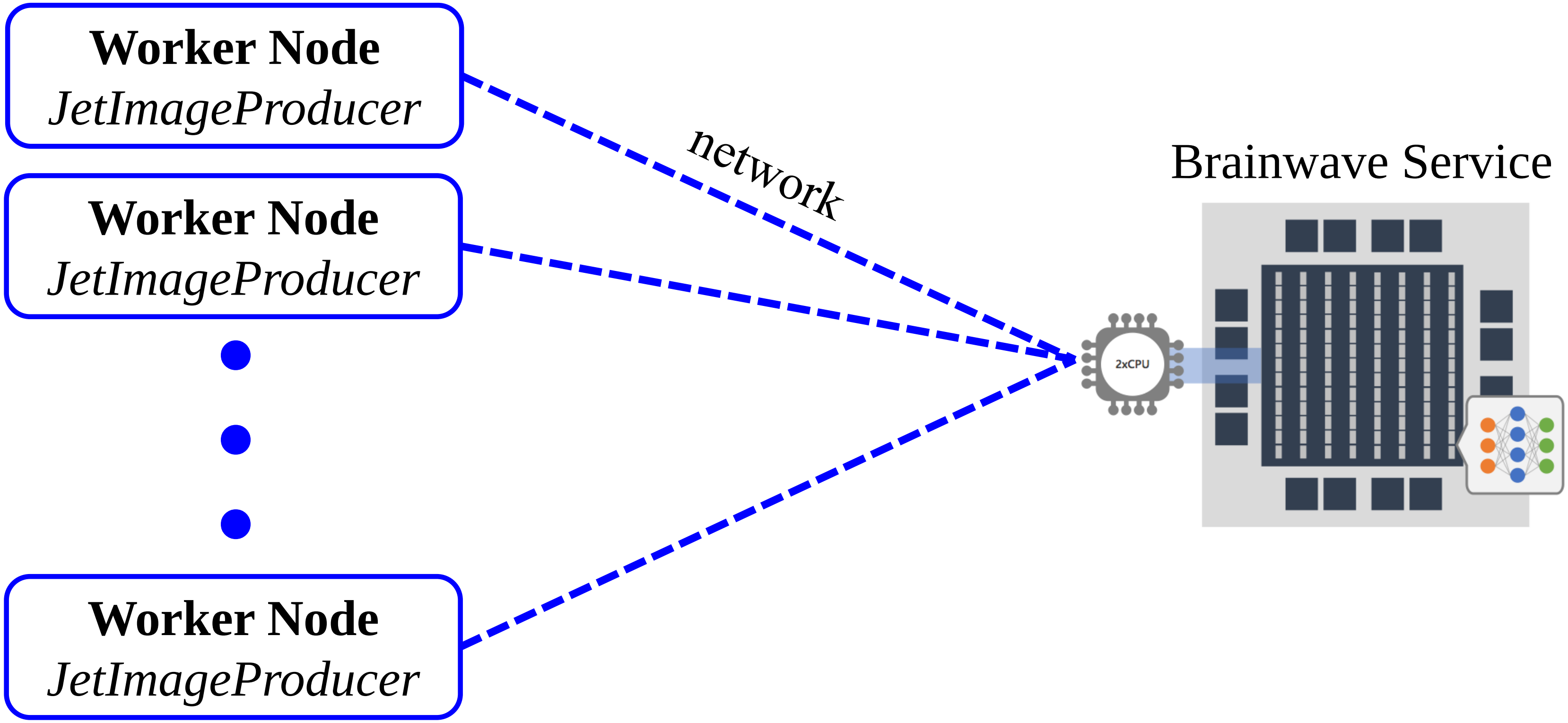
SONIC Latency





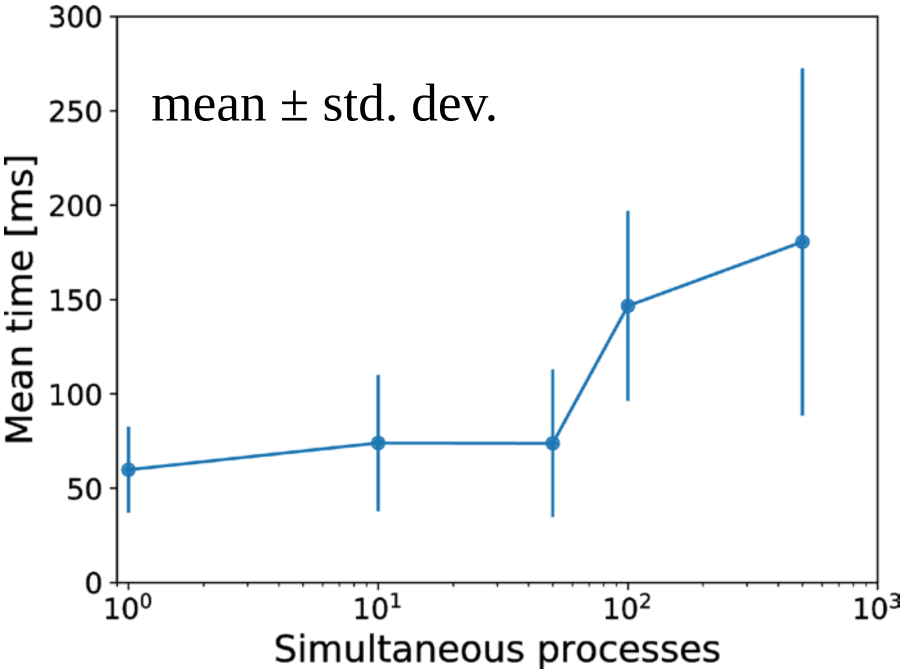
- Remote: cmslpc @ FNAL to Azure (VA), ‹time› = 60 ms
    - Highly dependent on network conditions
- On-prem: run CMSSW on Azure VM, ‹time› = 10 ms
    - FPGA: 1.8 ms for inference
    - Remaining time used for classifying and I/O

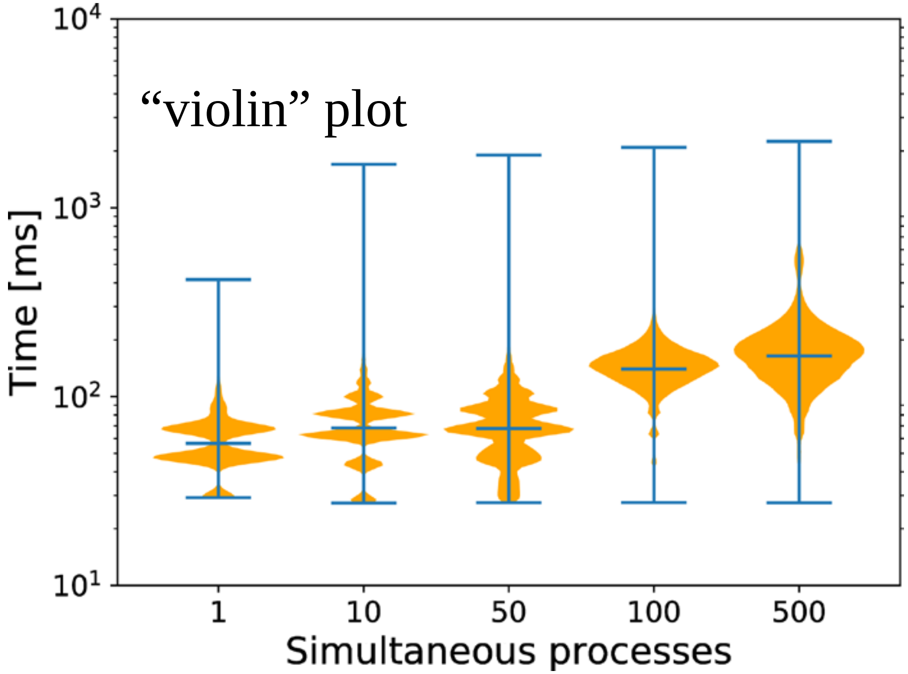
Scaling Tests



- Run N simultaneous processes, all sending requests to 1 BrainWave service
- Processes only run JetImageProducer from SONIC → "worst case" scenario
  - Standard reconstruction process would have many non-SONIC modules
- FPGA performs inference serially (1 image at a time)

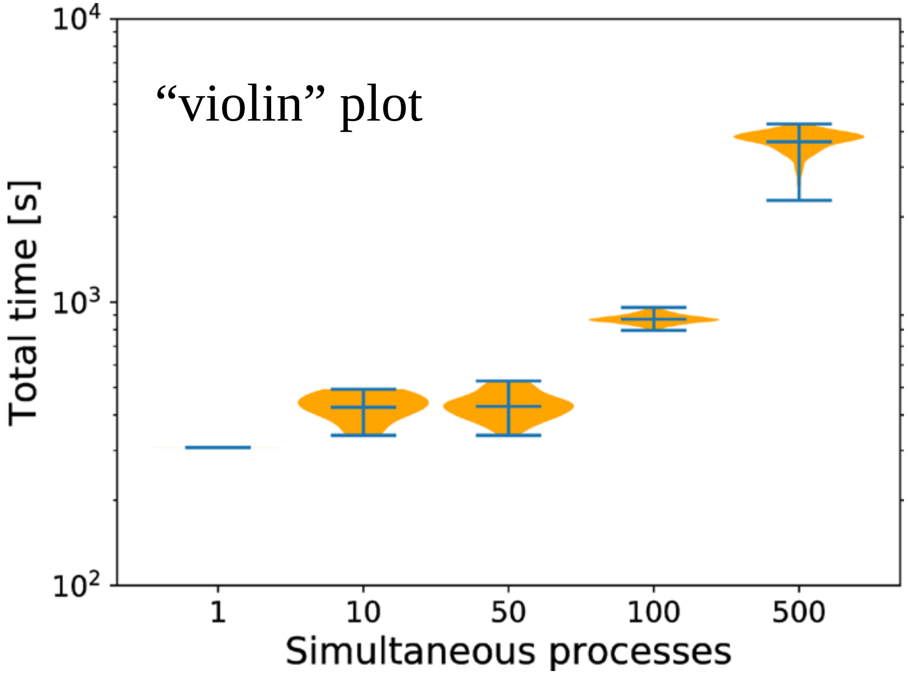
SONIC Latency: Scaling

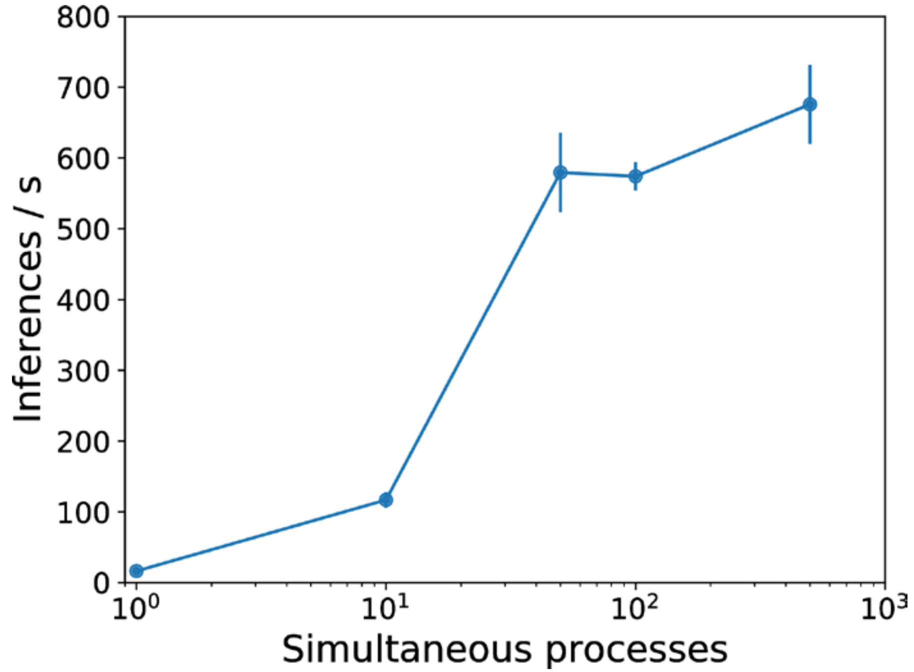




- Tests: N = 1, 10, 50, 100, 500
- Only moderate increases in mean, standard deviation, and long tail for latency
  - Fairly stable up to N = 50

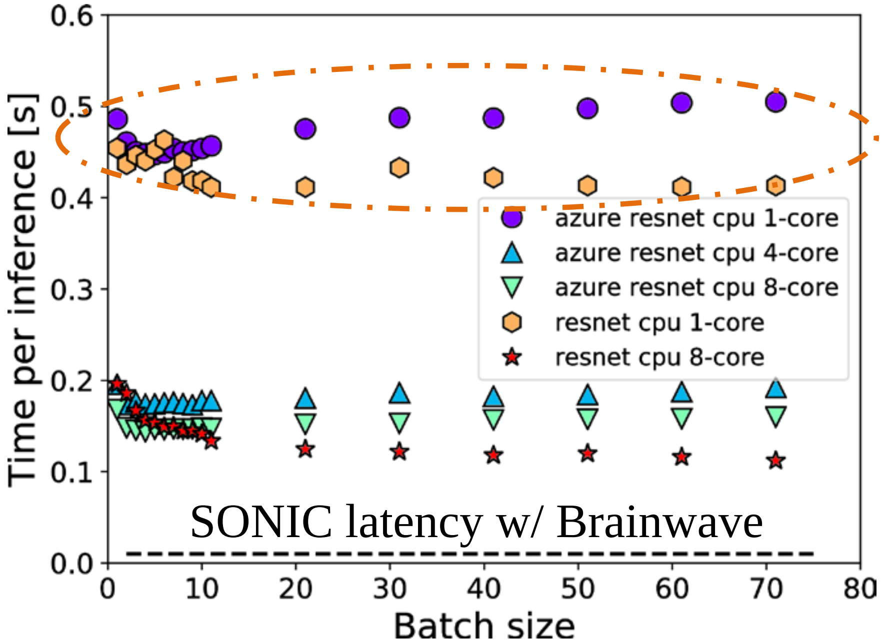
SONIC Throughput

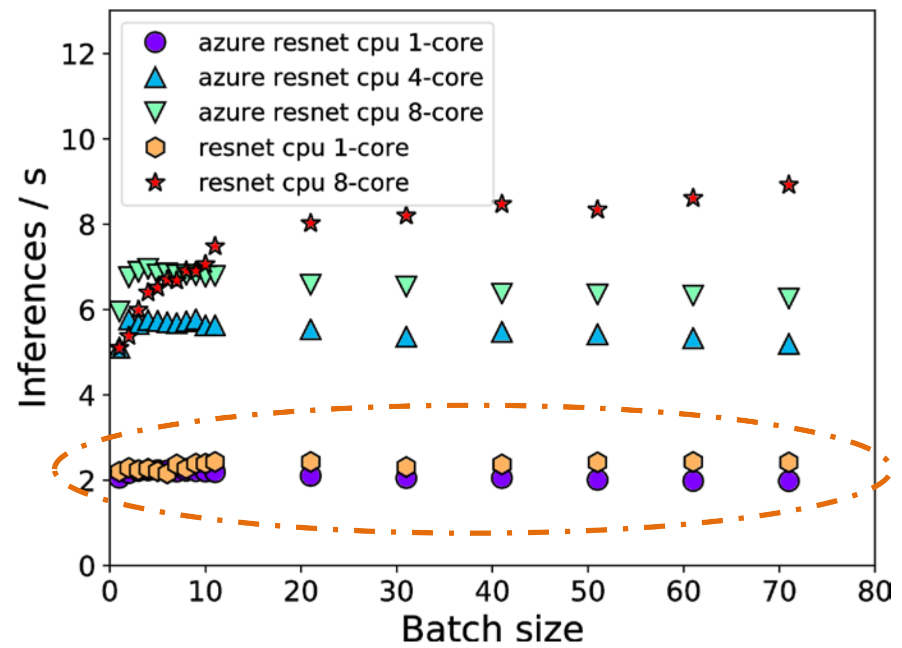




- Each process evaluates 5000 jet images in series
- Remarkably consistent total time for each process to complete
  - o Brainwave load balancer works well
- Compute inferences per second as $(5000 \cdot N)/(\text{total time})$
- N = 50 ~fully occupies FPGA:
  - o Throughput up to 600 inferences per second (max ~650)

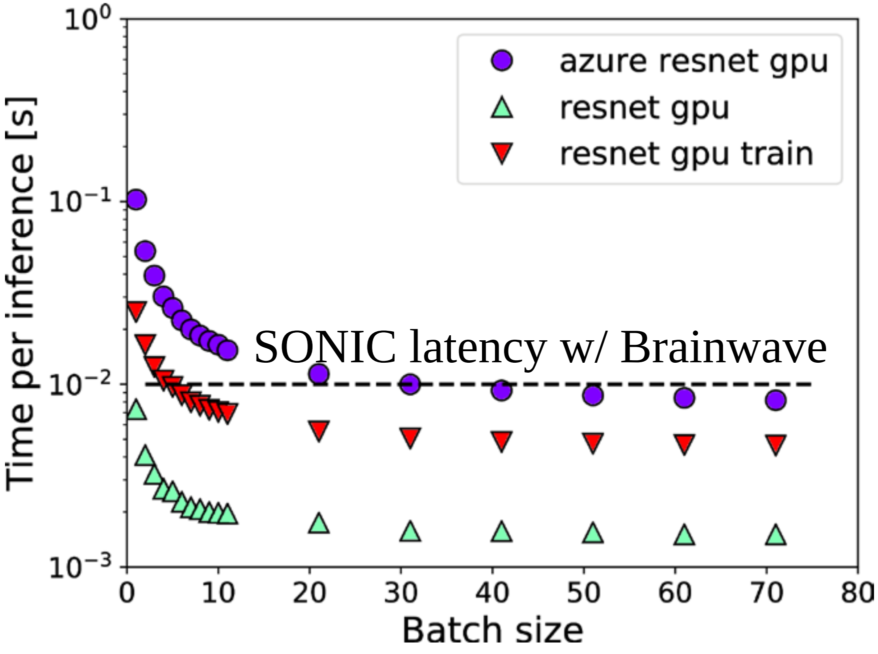
CPU Performance

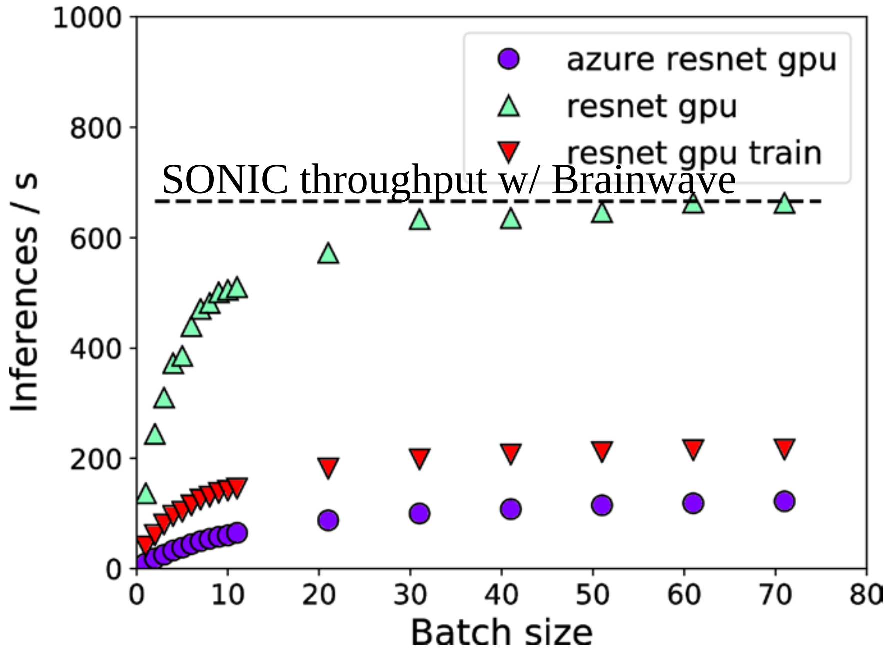




- Above plots use i7 3.6 GHz, TensorFlow v1.10
- Local test with CMSSW on cluster @ FNAL:
  - Xeon 2.6 GHz, TensorFlow v1.06
  - 5 min to import Brainwave version of ResNet-50
  - 1.75 sec/inference subsequently

GPU Performance





- Above plots use NVidia GTX 1080, TensorFlow v1.10
- GPU directly connected to CPU via PCIe
- TF built-in version of ResNet-50 performs better on GPU than quantized version used in Brainwave

Performance Comparisons

Type	Note	Latency [ms]	Throughput [img/s]
CPU*	Xeon 2.6 GHz	1750	0.6
	i7 3.6 GHz	500	2
GPU**	batch = 1	7	143
	batch = 32	1.5	667
Brainwave	remote	60	660
	on-prem	10 (1.8 on FPGA)	660

- *CPU performance depends on:
  - clock speed, TensorFlow version, # threads (=1 here)
- **GPU caveats:
  - Directly connected to CPU via PCIe – not a service
  - Performance depends on batch size & optimization of ResNet-50 network
- SONIC achieves:
  - ➢ 175× (30×) on-prem (remote) improvement in latency vs. CMSSW CPU!
  - ➢ Competitive throughput vs. GPU, w/ single-image batch as a service!

Summary

- Particle physics experiments face extreme computing challenges
  - More data, more complex detectors, more pileup
- Growing interest in machine learning for reconstruction and analysis
  - As networks get larger, inference takes longer
- FPGAs are a promising option to accelerate neural network inference
  - Can achieve order of magnitude improvement in latency over CPU
  - Comparable throughput to GPU, without batching
    - ➢ Better fit for event-based computing model
- SONIC infrastructure developed and tested
  - Compatible with any service that uses gRPC and TensorFlow

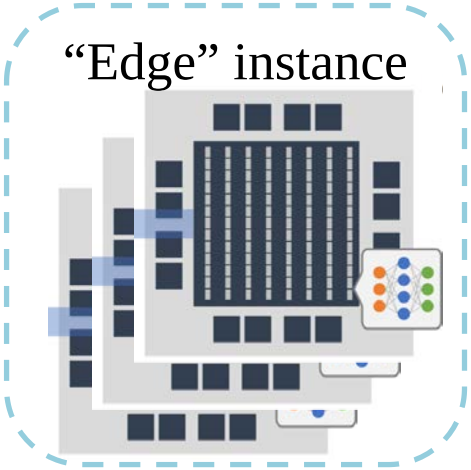
➢ **Paper with these results in preparation**

- Thanks to Microsoft for lots of help and advice!
  - Azure Machine Learning, Bing, Project Brainwave teams
  - Doug Burger, Eric Chung, Jeremy Fowers, Kalin Ovtcharov, Andrew Putnam

Continuing Work

- **Continue to translate particle physics algorithms into machine learning**
  - Easier to accelerate inference w/ commercial coprocessors
- **Develop tools for generic model translation**
  - E.g. graph NNs used for HEP.TrkX and other projects
- **Explore broad offering of potential hardware**
  - Google TPUs, Xilinx ML suite on AWS, Intel OpenVINO, …
- **Continue to build infrastructure and study scalability/cost**
  - Adapt SONIC to handle other protocols, other network architectures and ML libraries, other experiments (e.g. neutrinos)

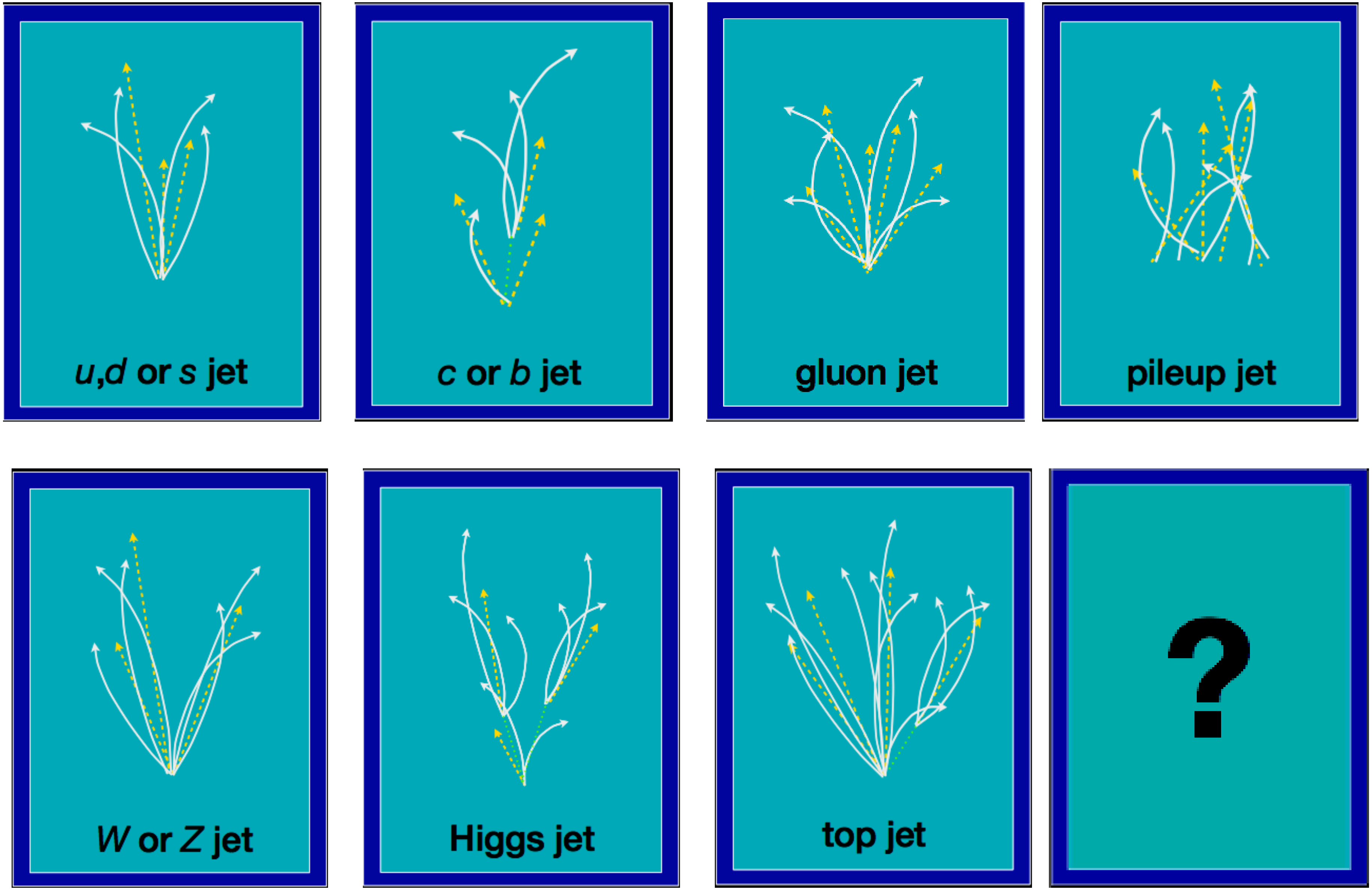
A Vision of the Future



- A single FPGA can support many CPUs → cost-effective
  - SONIC throughput results indicate 1 FPGA for 100–1000 CPUs running realistic processes (many algorithms, only some ML inferences)
- Install small “edge” instances at T1s and T2s
  - Can also install a dedicated instance for CMS HLT farm at CERN

Backup

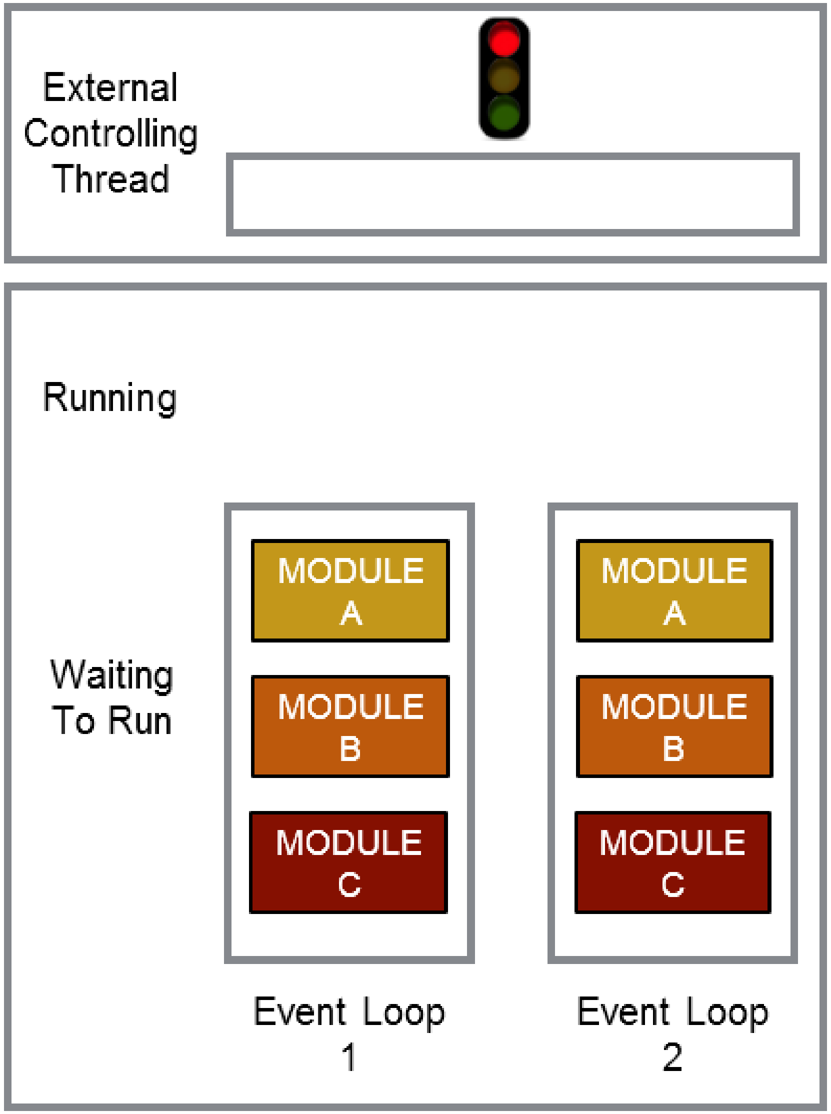
Jet Substructure



External Work in CMSSW (1)

Setup:

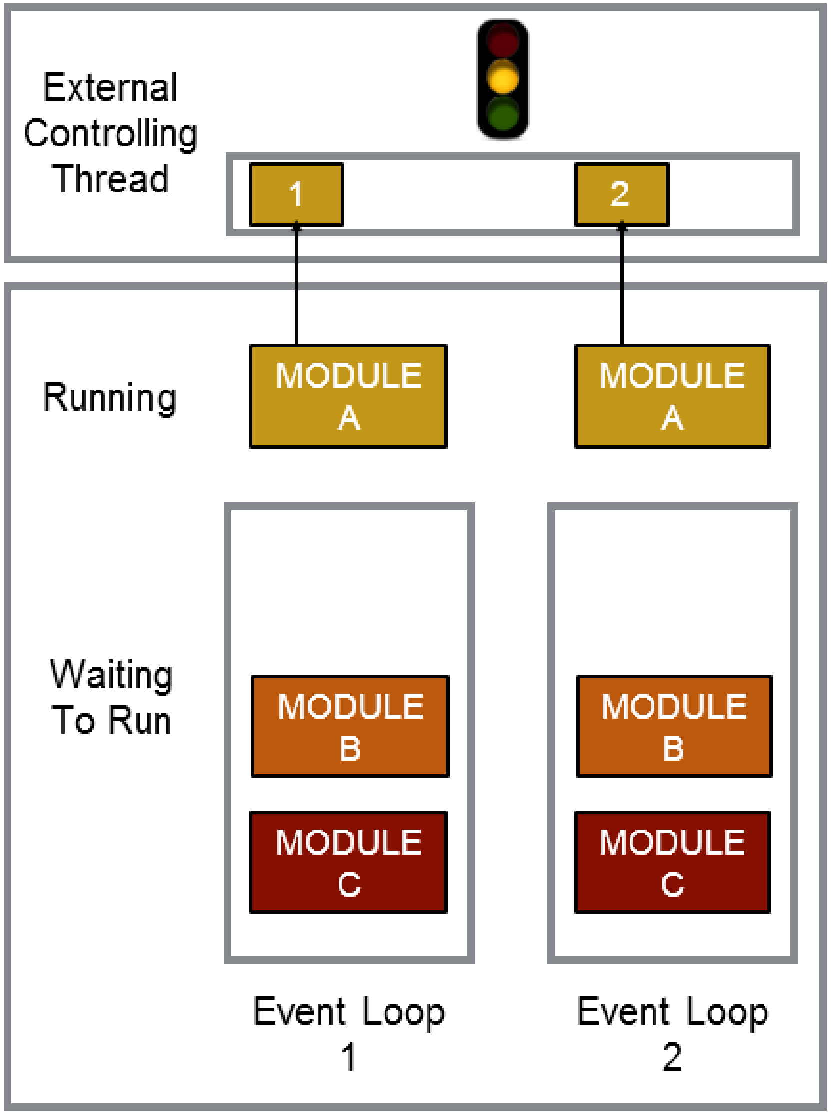
- TBB controls running modules
- Concurrent processing of multiple events
- Separate helper thread to control external
- Can wait until enough work is buffered before running external process



External Work in CMSSW (2)

Acquire:

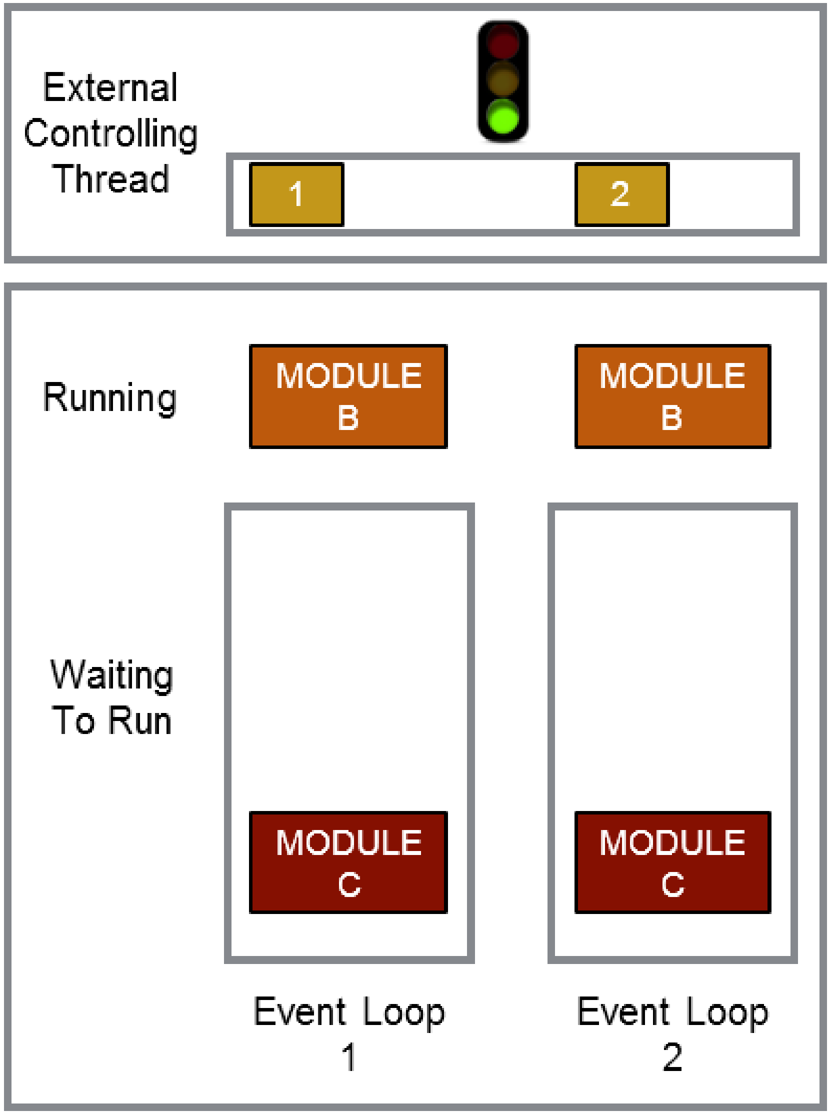
- Module *acquire*() method called
- Pulls data from event
- Copies data to buffer
- Buffer includes callback to start next phase of module running



External Work in CMSSW (3)

Work starts:

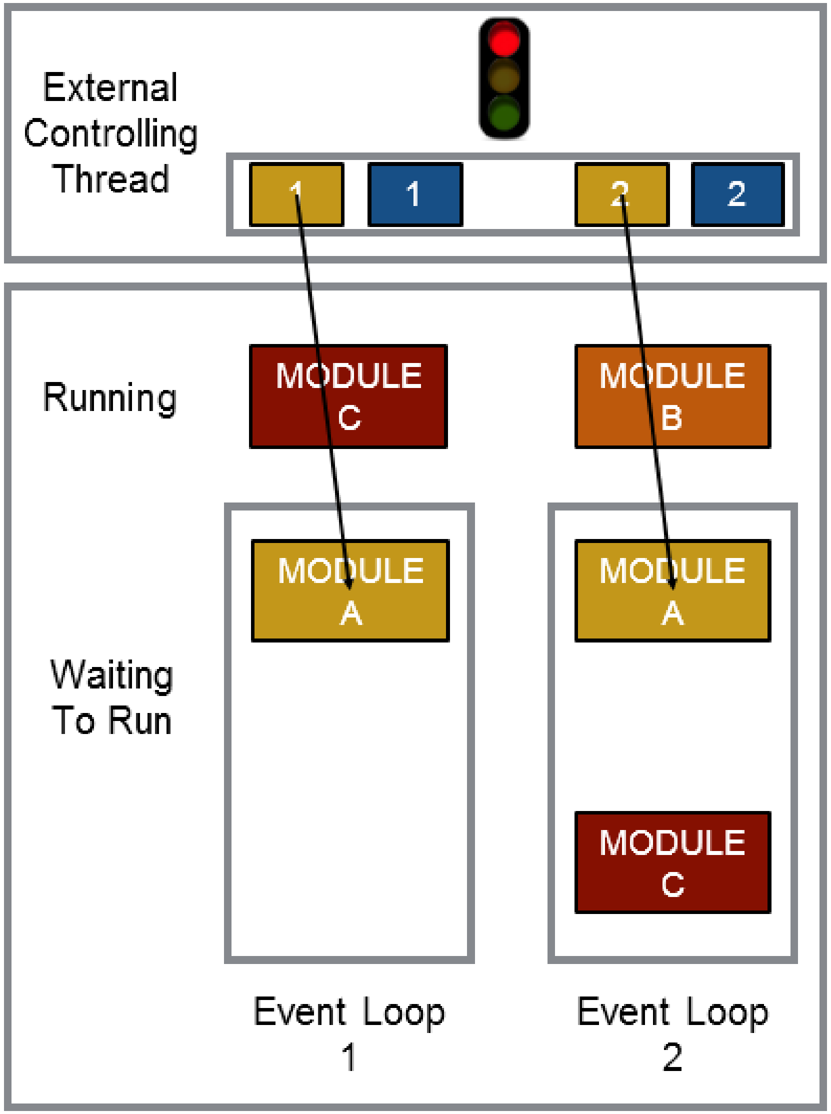
- External process runs
- Data pulled from buffer
- Next waiting modules can run (concurrently)



External Work in CMSSW (4)

Work finishes:

- Results copied to buffer
- Callback puts module back into queue



External Work in CMSSW (5)

Produce:

- Module *produce*() method is called
- Pulls results from buffer
- Data used to create objects to put into event