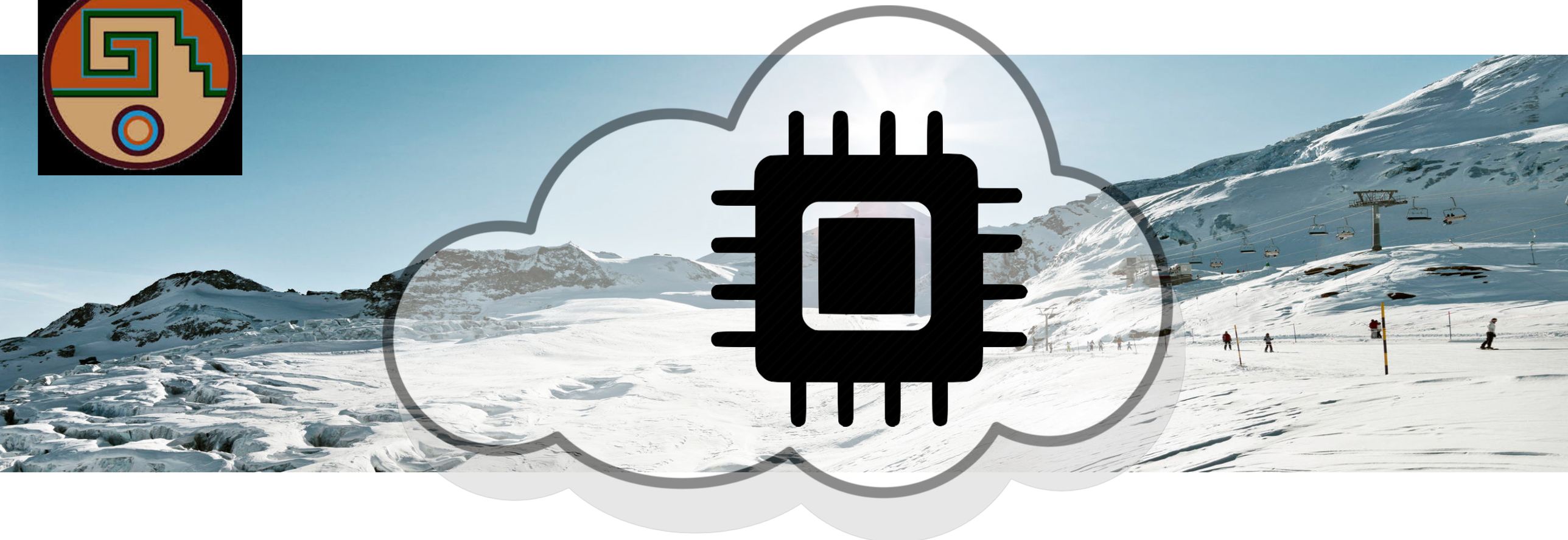
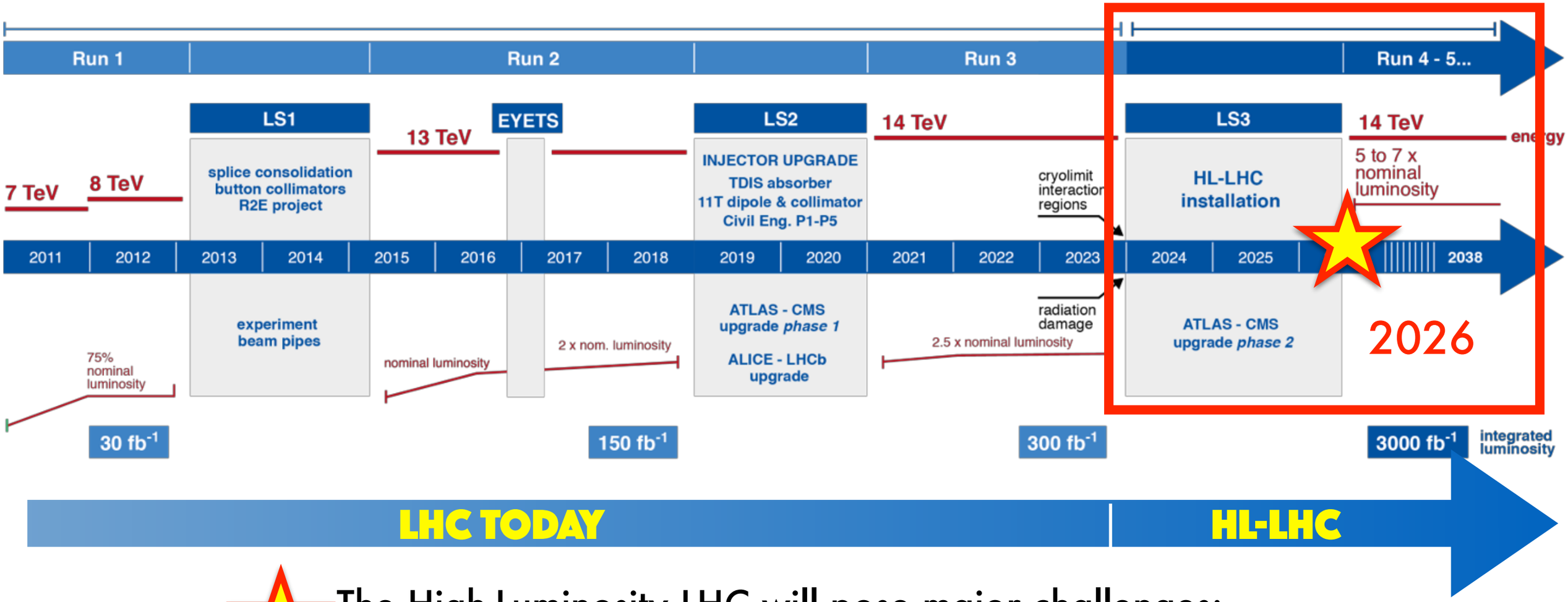


FPGA-accelerated machine learning inference as a service for particle physics computing

Jennifer Ngadiuba, Maurizio Pierini (CERN)
Javier Duarte, Burt Holzman, Ben Kreis, Kevin Pedro, Mia Liu, Nhan Tran, Aris Tsaris (FNAL)
Phil Harris, Dylan Rankin (MIT)
Zhenbin Wu (UIC)

ACAT, 11-15 March 2019, Saas Fee, Swizterland

The LHC big data problem

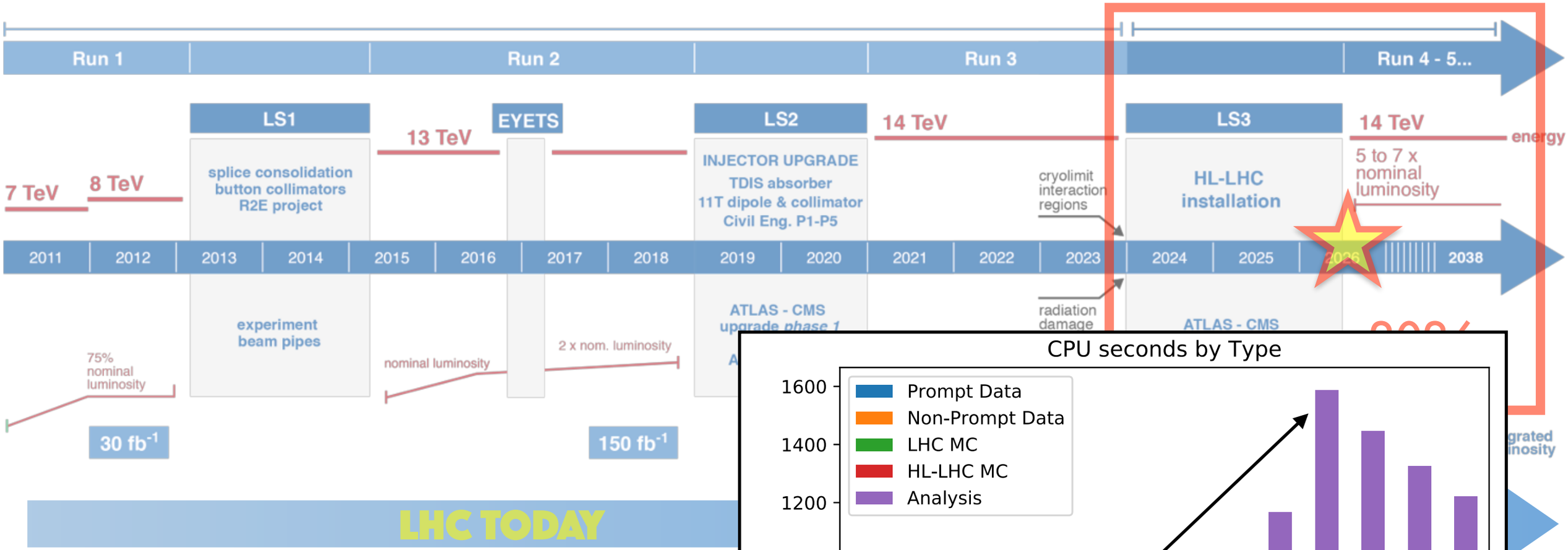


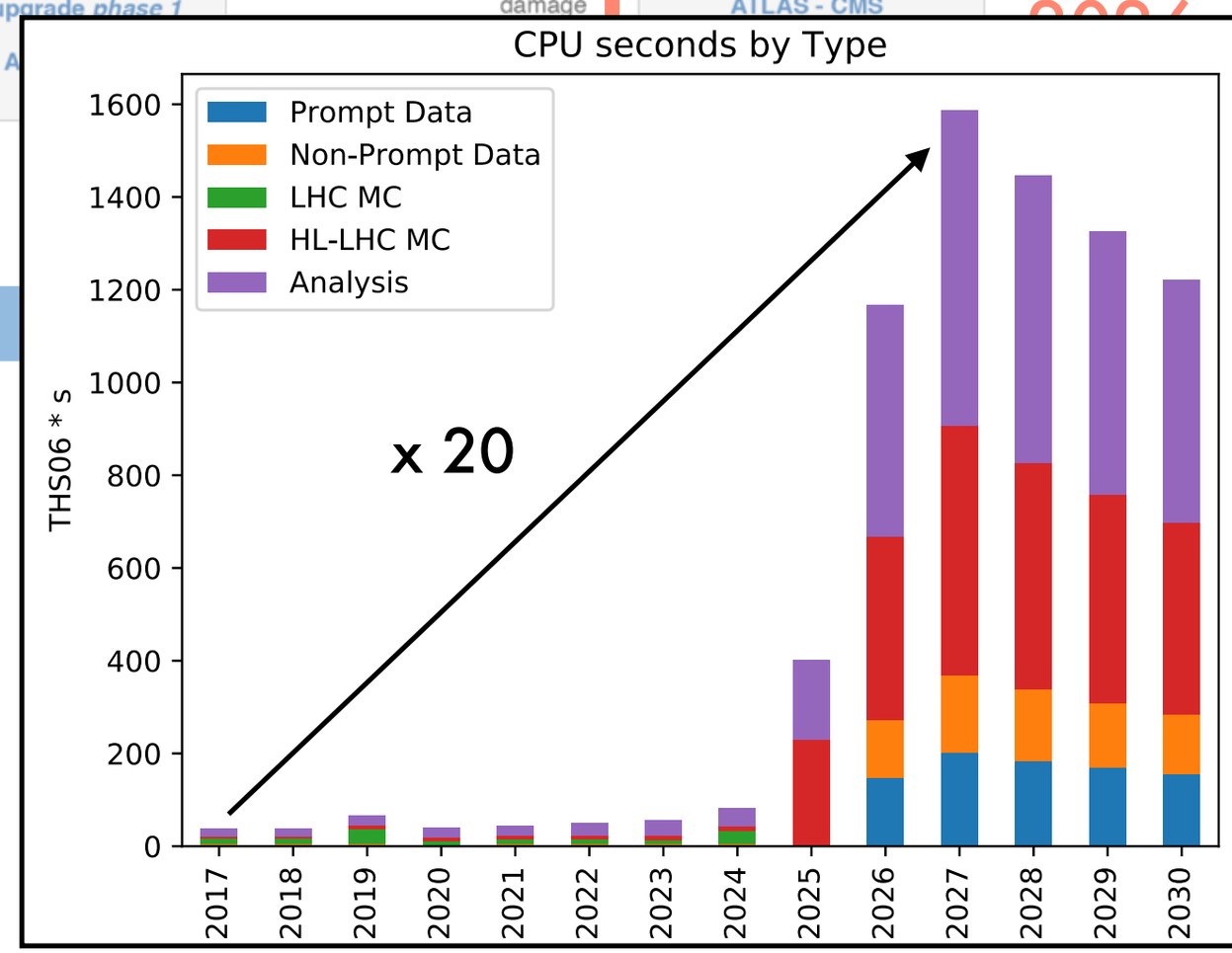
The High-Luminosity LHC will pose major challenges:

instantaneous luminosity x 5–7
particles per collision x 5
more data x 15
more granular detectors with x 10 readout channels

→ event rates & datasets will increase to unprecedented levels!

The LHC big data problem



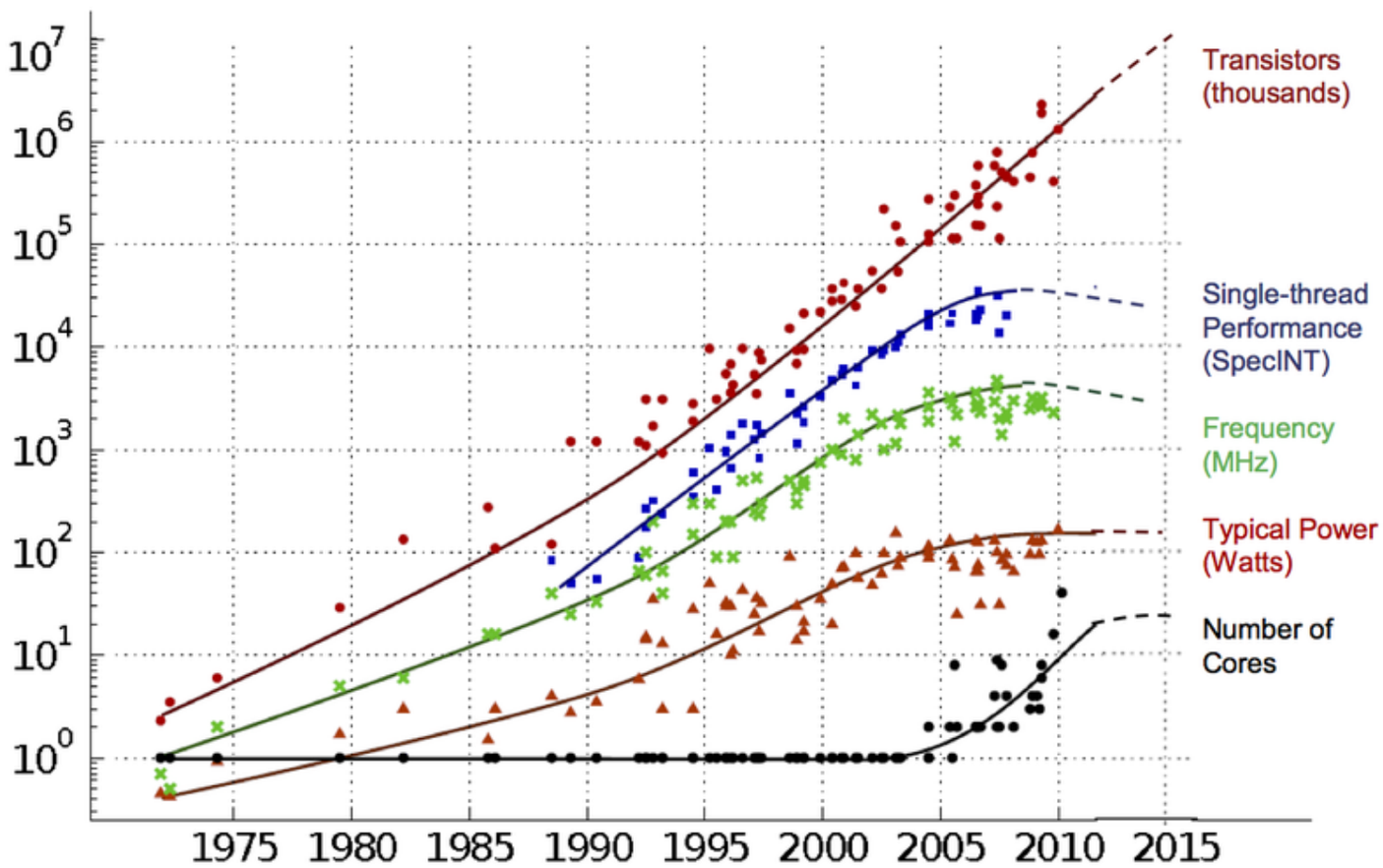


x 5 — Event complexity

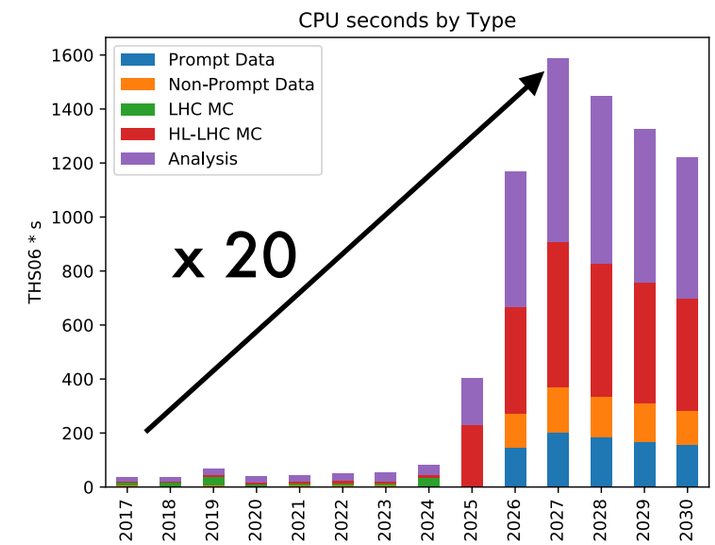
x 50 — Processing time

x 20 — Computing resources

The LHC big data problem



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



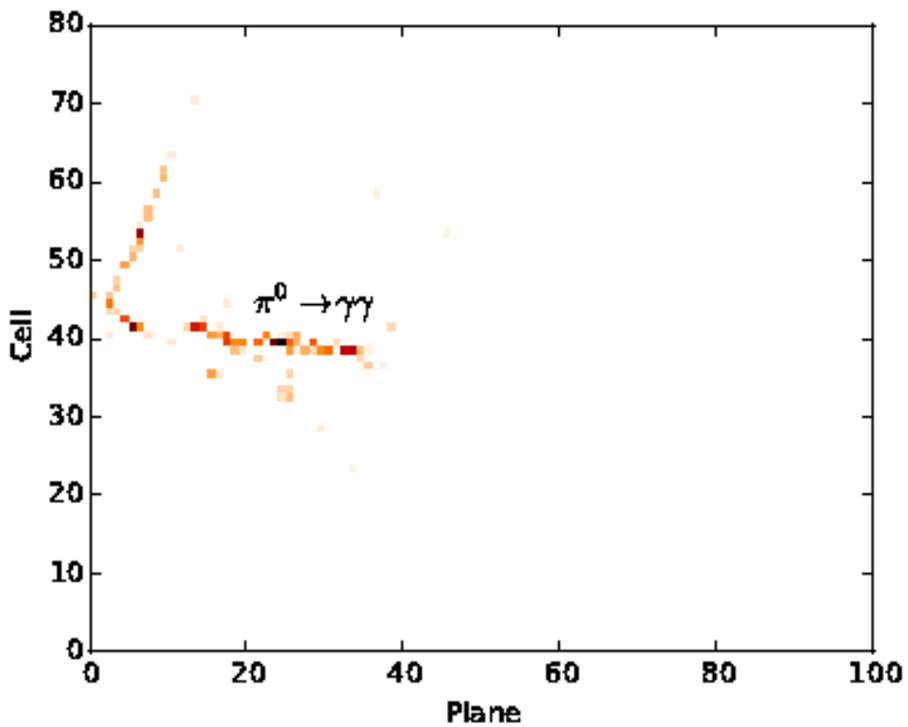
Moore's law still valid…
but Dennard's scaling no longer maintained!

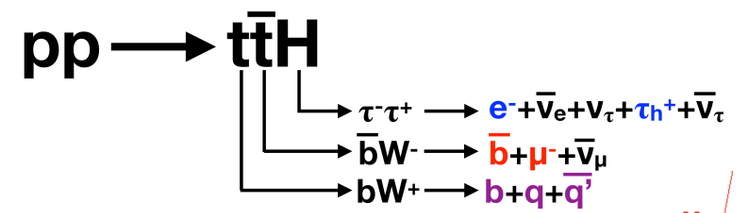
Current data processing paradigms will not be sustainable with flat budget!

New technologies needed:
machine learning & heterogeneous computing

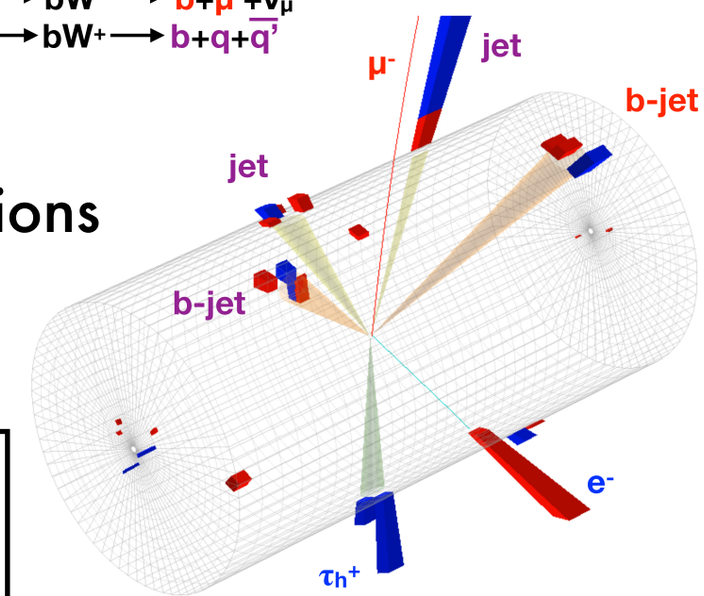
The success of ML in HEP

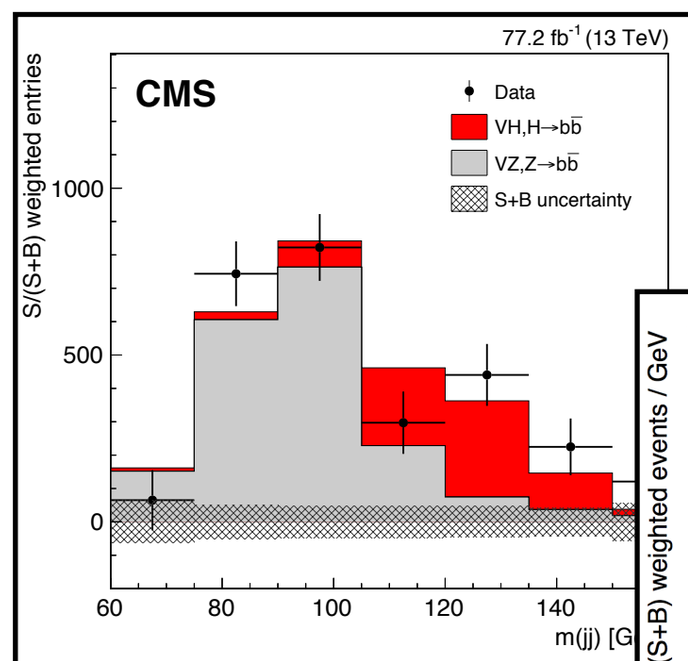
ex, neutrino event reconstruction with GoogleNet @ Nova

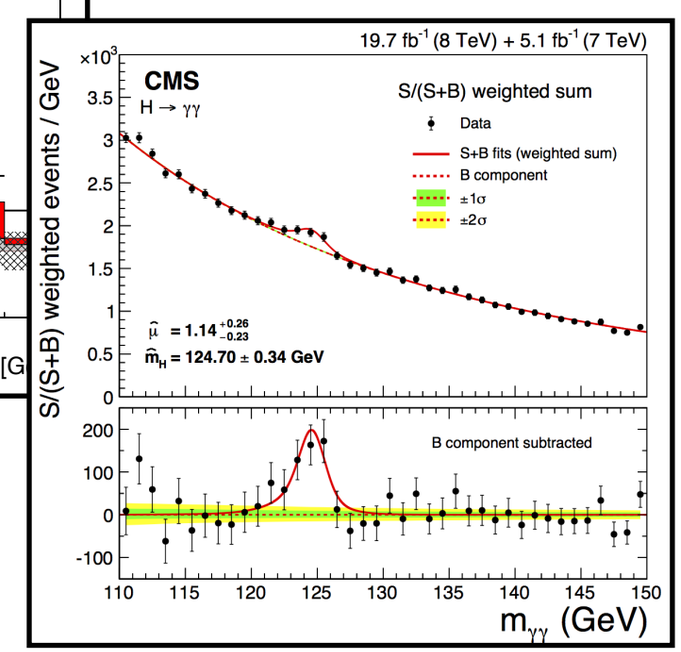




ex, Higgs boson observations @ ATLAS,CMS

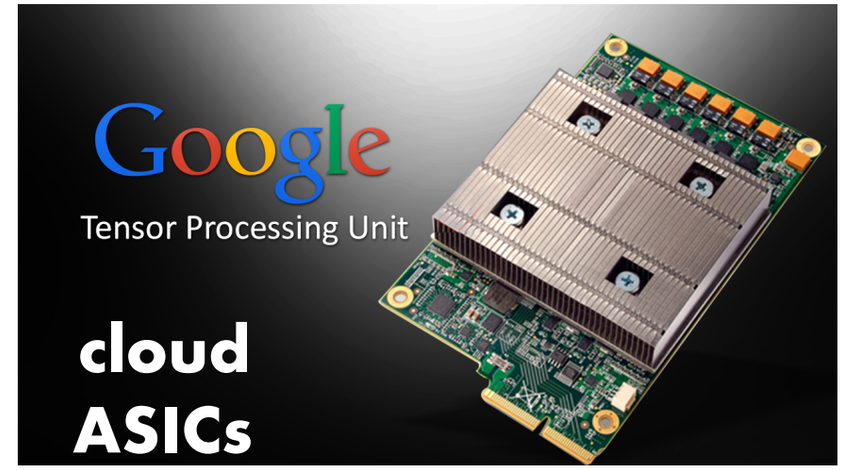




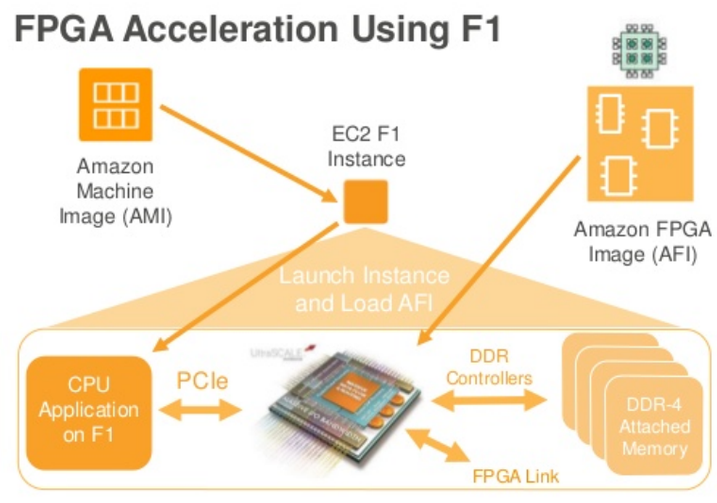


Heterougeneous computing

- Offload a CPU from the computational heavy parts to an "accelerator" → **co-processor system**
  - *CPU+FPGA / CPU+GPU / CPU+ASICs / ...*
  - *high parallelization and data throughput*
  - *optimal for ML algorithms*



- Increasing popularity of co-processor systems in industry
  - *exploit trends in developing new devices optimized for ML and speedup the inference*



cloud FPGAs

Microsoft Brainwave: cloud FPGAs

Solving computing challenges

Computing intensive physics problems can benefit from co-processor systems
ex, particle track reconstruction

Option 1

rewrite physics algorithms for new hardware

Languages: OpenCL, OpenMB, TBB, VHDL, ...

Hardware: FPGA, GPU

Challenge: difficult to adapt to new and changing hardware

Option 2

recast physics problem as a machine learning problem

Languages:C++, Python, ...

Hardware:
FPGA, GPU, ASIC

Challenge:
how to map physics ↔ ML

Solving computing challenges

Computing intensive physics problems can benefit from co-processor systems
ex, particle track reconstruction

THIS TALK!

Proof-of-concept: particle physics computing with Brainwave

Option 2

recast physics problem as a machine learning problem

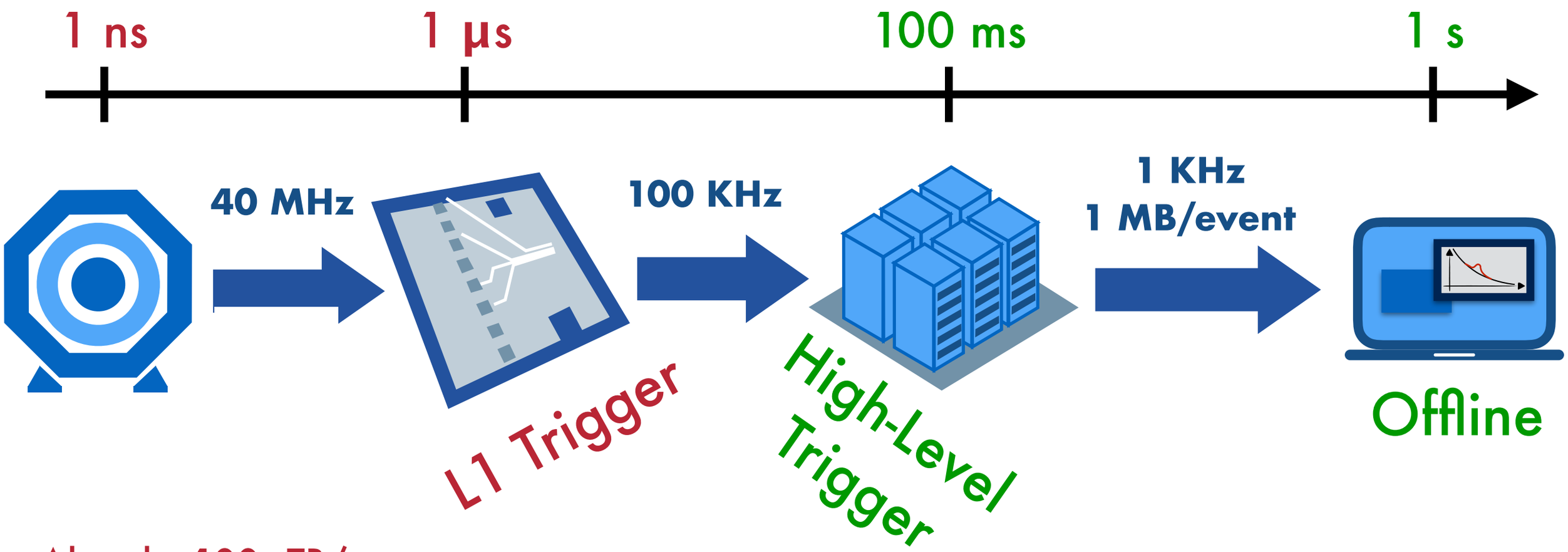
Languages:C++, Python, ...

Hardware:
FPGA, GPU, ASIC

Challenge:
how to map physics ↔ ML

Event processing @ LHC

Reduce data rates to manageable levels for offline processing
by filtering events through multiple stages:



Absorbs 100s TB/s

Trigger decision to be made in O(µs)

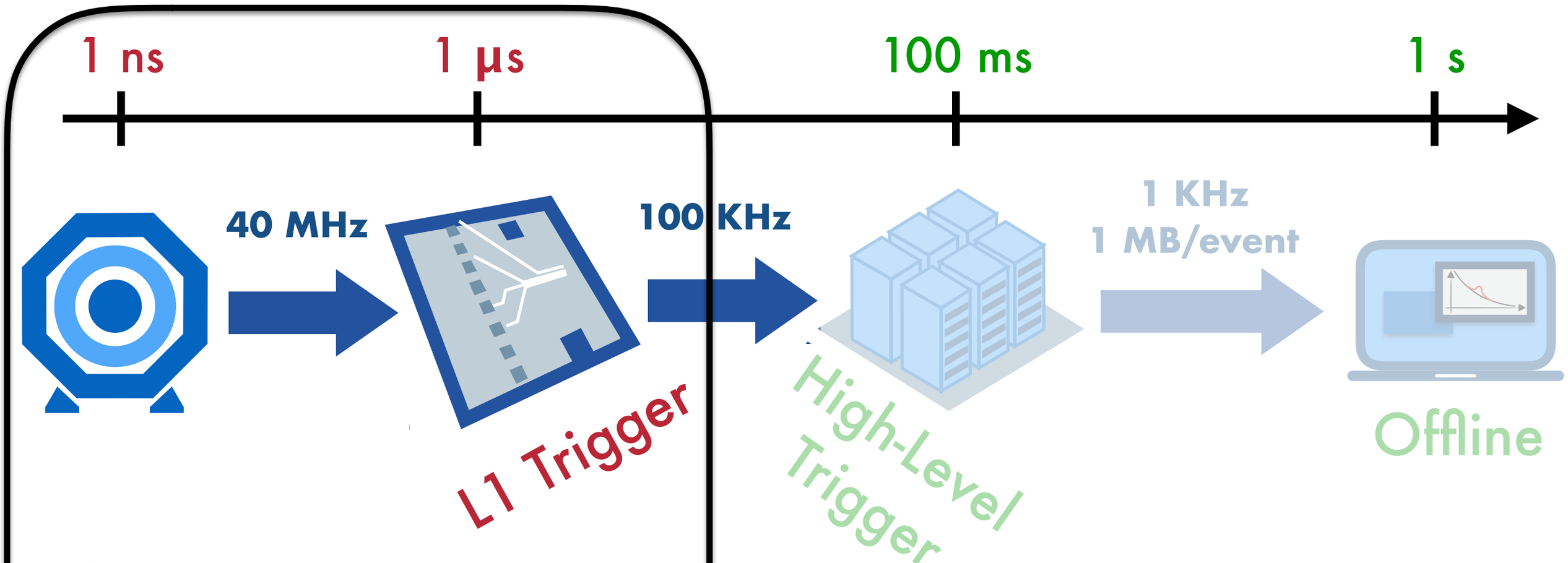
Latencies require all-FPGA design

Analysis of the full event runs on commercial computers (30k CPU cores)

Latency O(100 ms)

Event processing @ LHC

Reduce data rates to manageable levels for offline processing
by filtering events through multiple stages:



Absorbs 100s TB/s

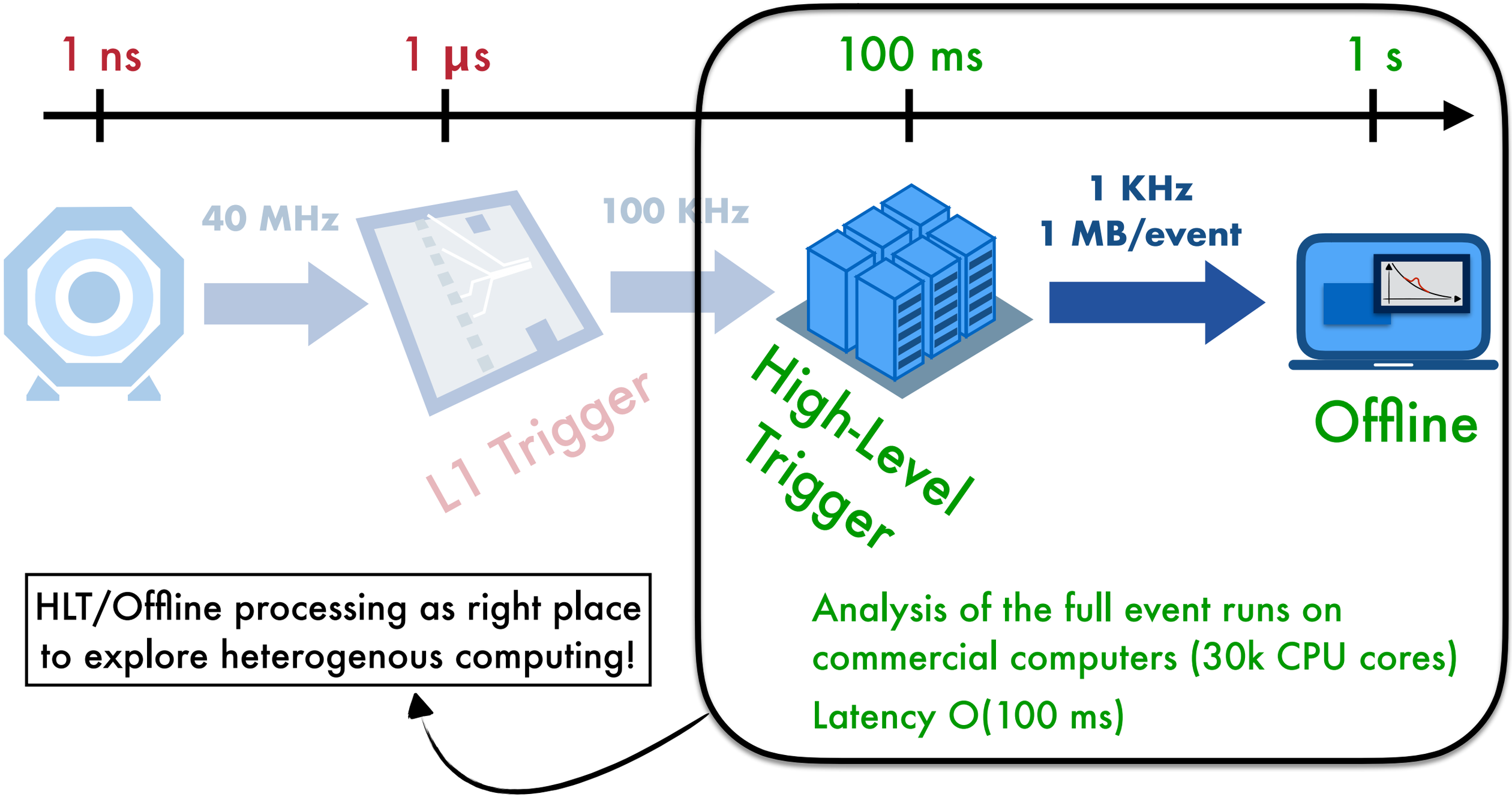
Trigger decision to be made in O(µs)

Latencies require all-FPGA design

Improvements at this stage a little tricky
→ see dedicated talk

Event processing @ LHC

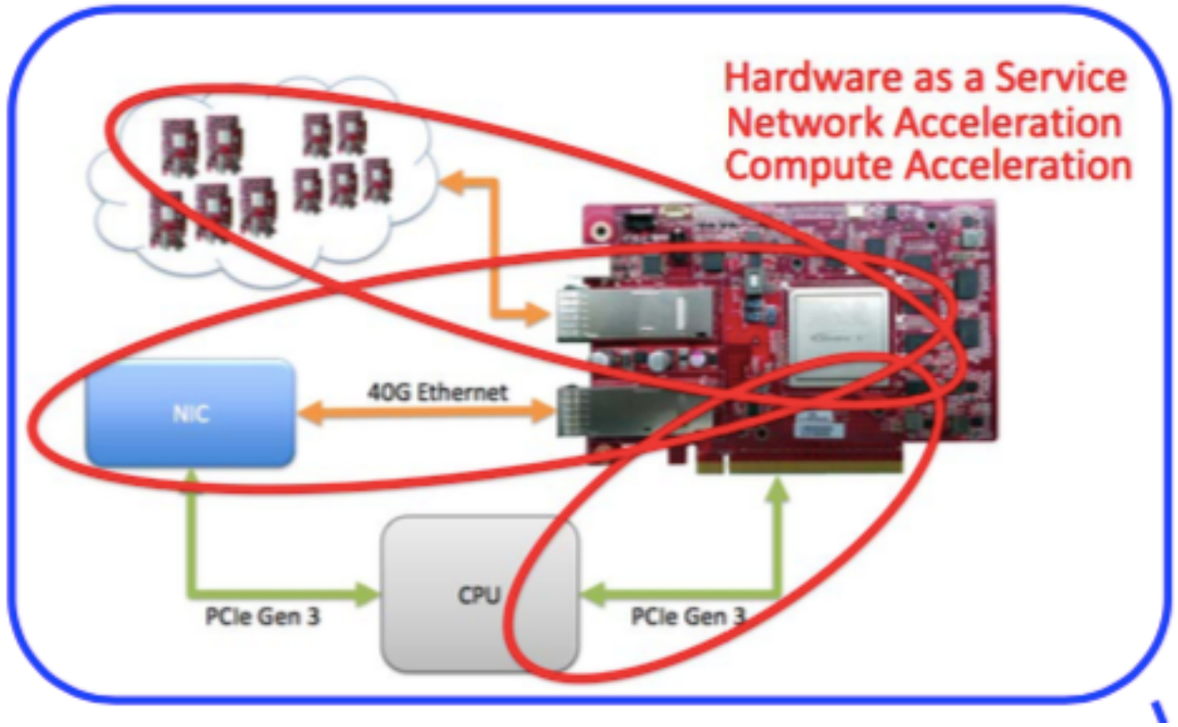
Reduce data rates to manageable levels for offline processing
by filtering events through multiple stages:

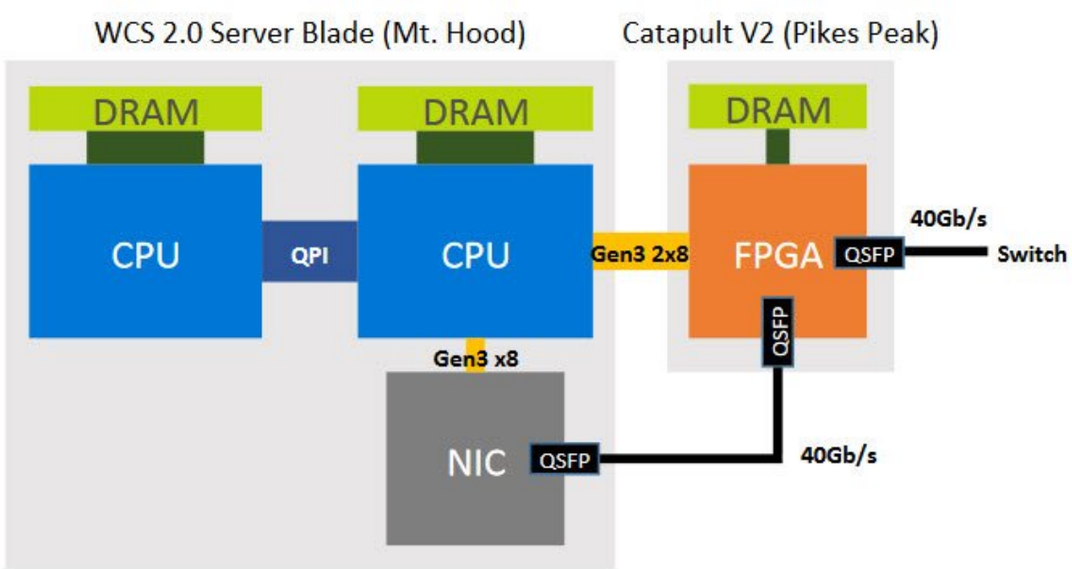


HLT/Offline processing as right place to explore heterogenous computing!

Co-processors as a service with Brainwave

- On-site co-processors interesting solution for HLT computing farm where latency is the bottleneck
- For offline, better solution is using co-processors as a service on the cloud
  - *not feasible to buy specialized hardware for each T1, T2, T3 computing center*
- **Project Brainwave provides a full scalable real-time AI service** on Azure cloud (more than just a single co-processor)
  - *Multi-FPGA+CPU fabric accelerating both computing and network*
  - *Caveat: currently supports only selected computing vision off-the-shelf networks*



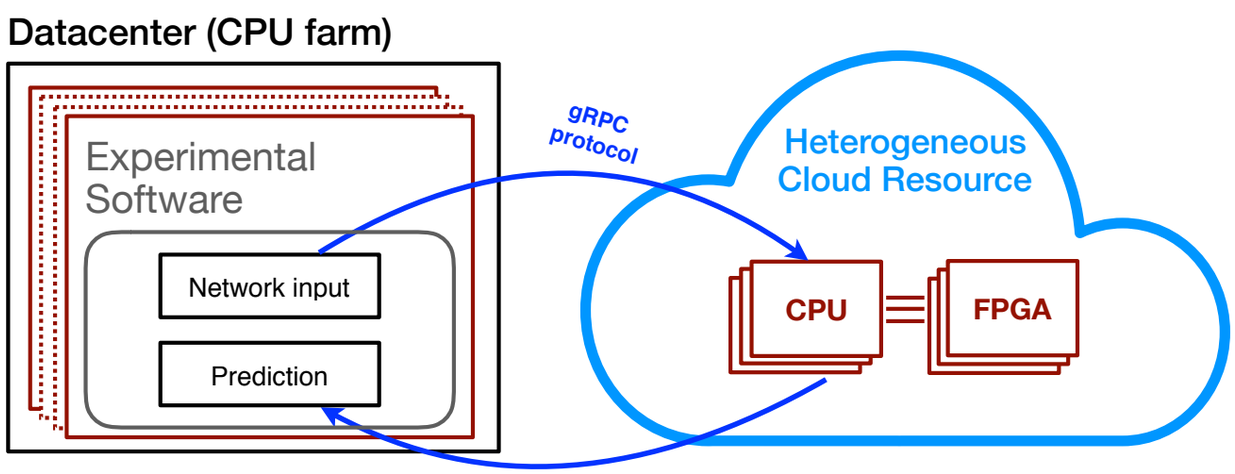


Proof-of-concept: SONIC

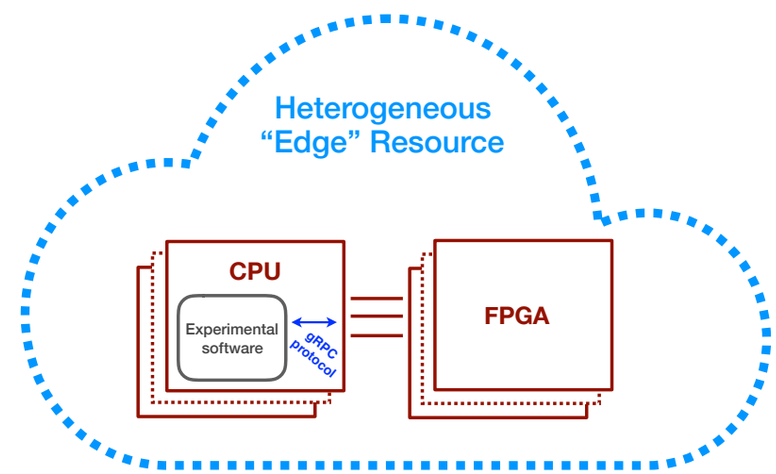
Service for Optimized Network Inference on Co-processors:
a framework to exploit heterogeneous resources for on-demand ML inference

<u>How to integrate FPGA co-processor into current multithreaded paradigms?</u>

Option 1: cloud service



Option 2: edge service



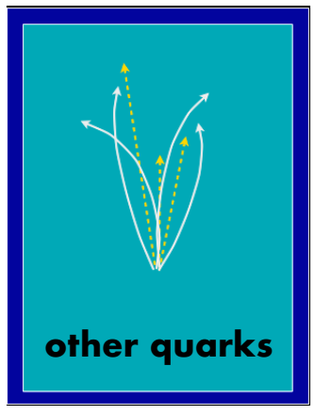
Cloud service has a latency due to data transfer

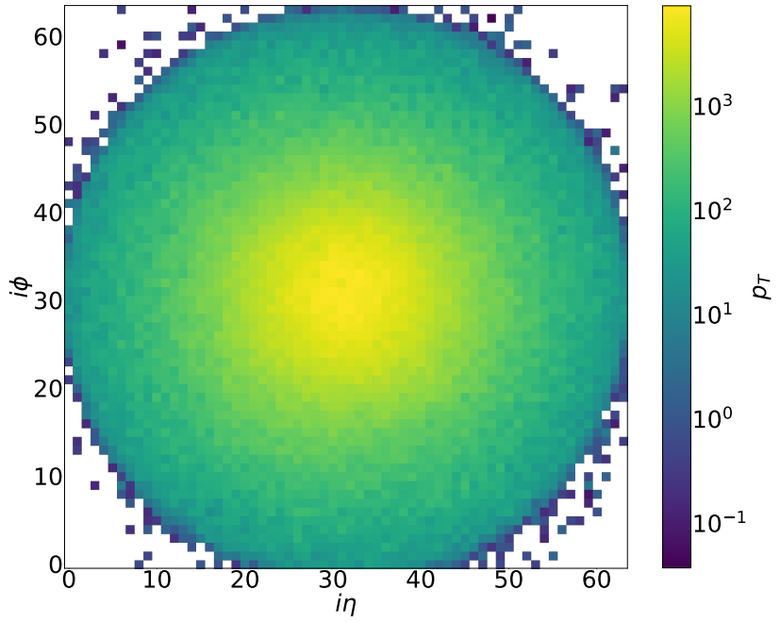
→ *option 2: explore also "edge" or "on-prem" case: run CMS software on Azure cloud machine to simulate on-site installation of FPGAs. Provide test of "HLT-like" performance.*

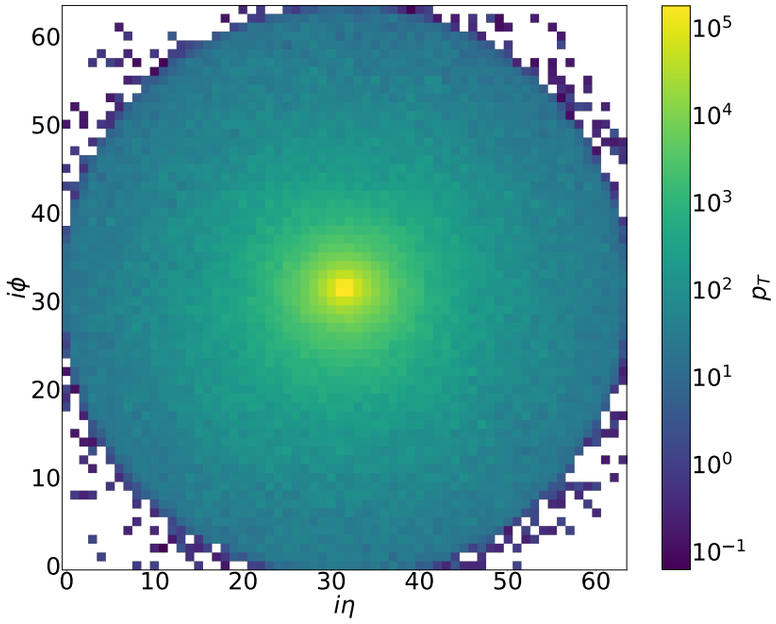
Physics case: top tagging with ResNet50

- Brainwave allows the use of custom weights for fixed architectures
- Train **ResNet-50** on 2D jet images to distinguish two types of jets

Top-quark jets **versus** **QCD jets**



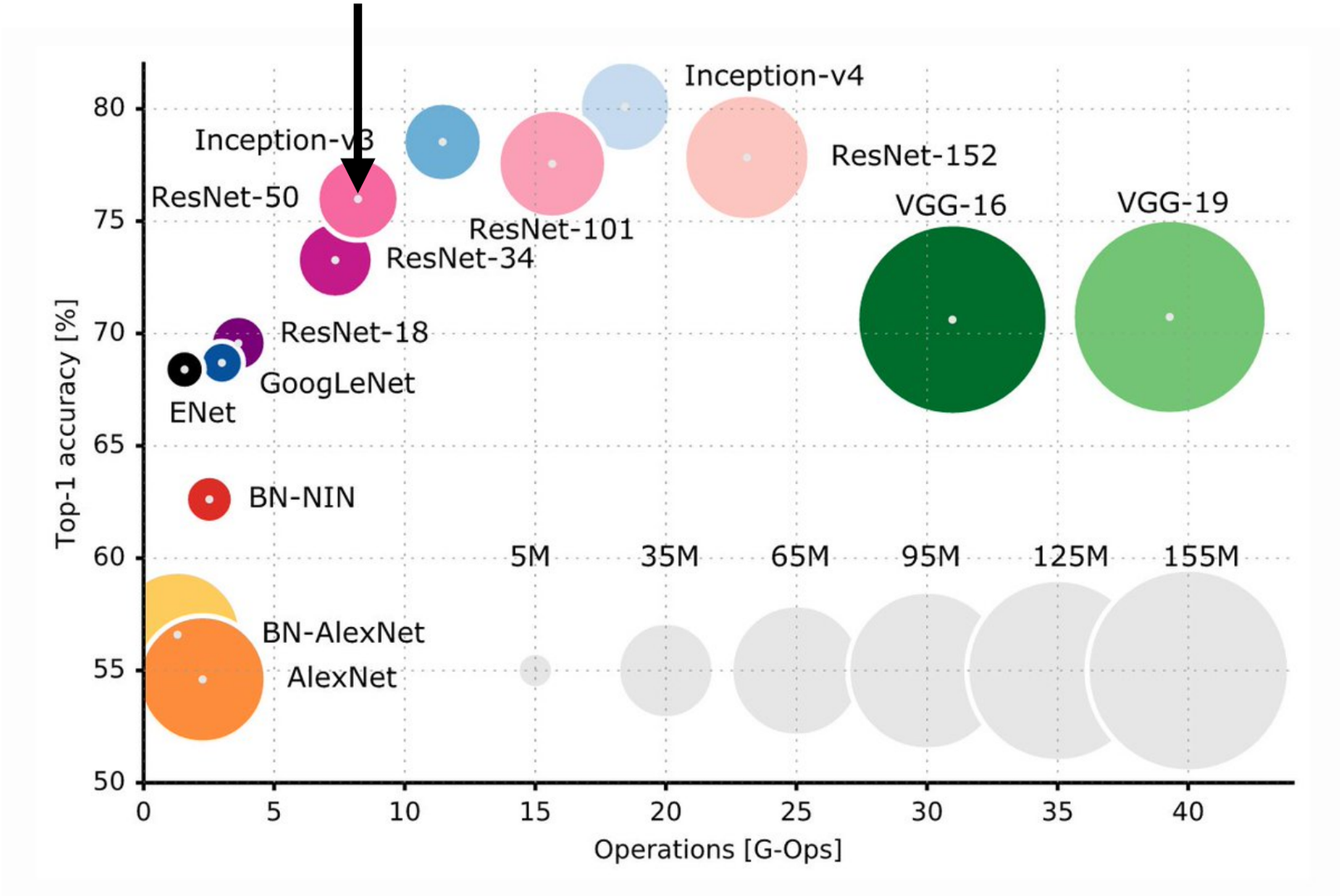




Physics case: top tagging with ResNet50

ResNet-50 made of two components:

- **featurizer**: several convolutional layers to extract image features
  → computationally intensive and accelerated on the FPGA
- **classifier:** few fully connected layers → inference performed on CPU

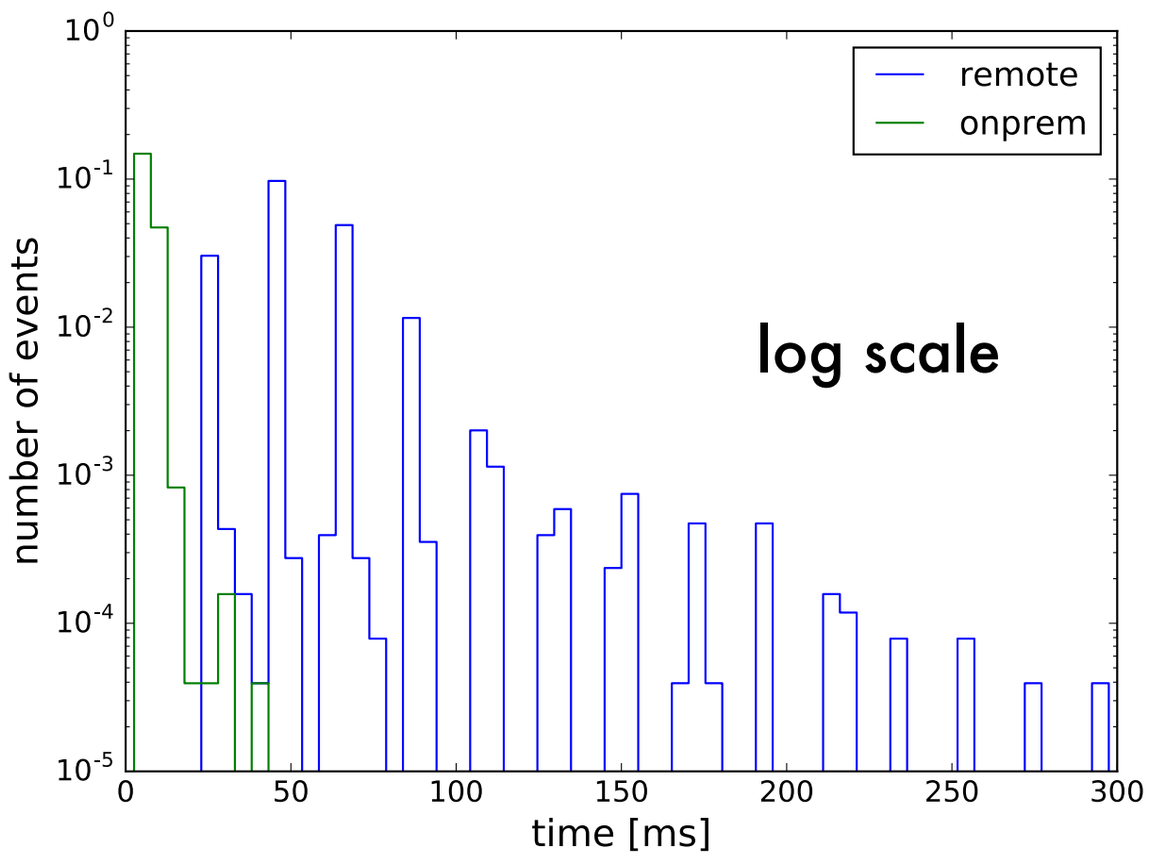


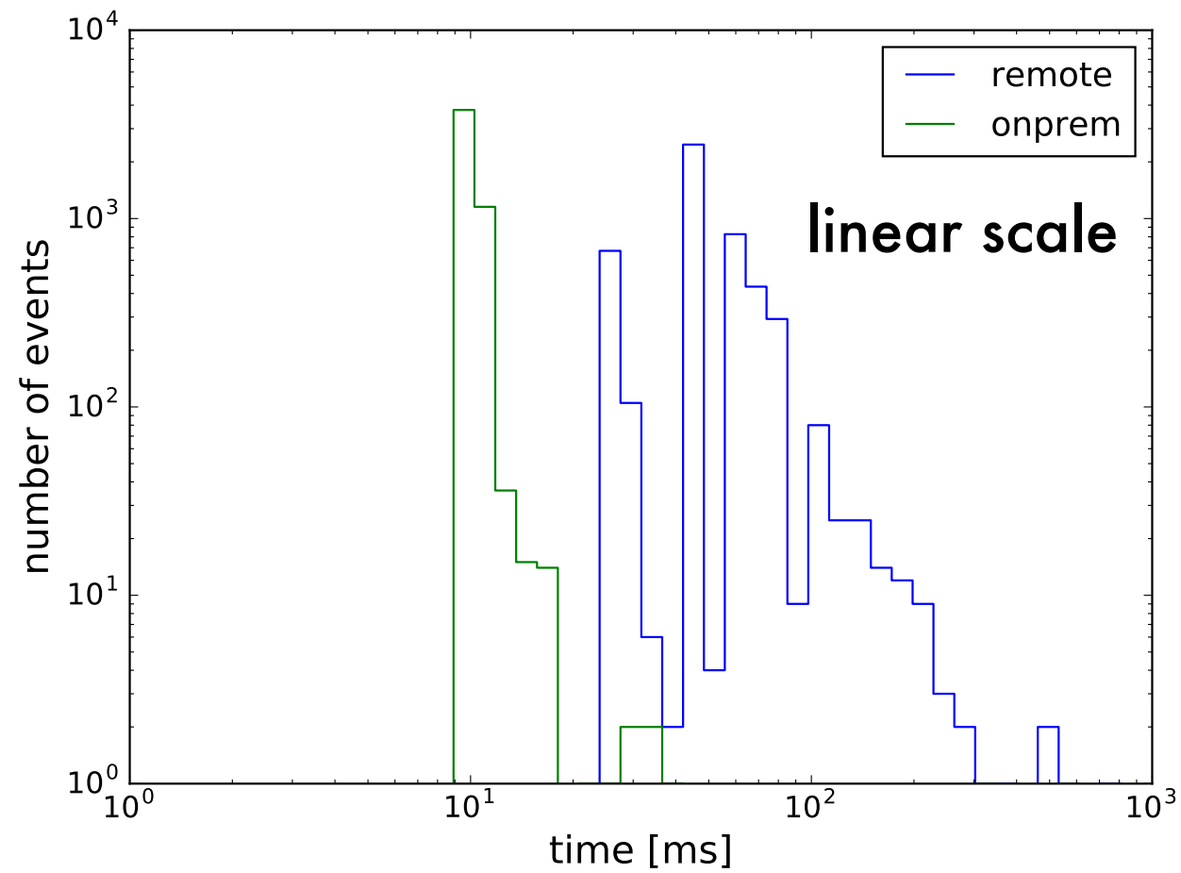
ResNet50: 25M parameters, 7B operations
Examples of large networks used in CMS:

- DeepAK8: 500K parameters, 15M operations (CMS-DP-2017-049)
- DeepDoubleB: 40K parameters, 700K operations (CMS-DP-2018-046)

Testing SONIC

Measure the performance of the SONIC package measuring the total end-to-end latency of an inference request to Brainwave within CMSSW





remote test:

FROM CPU @ Fermilab, Illinois

TO Azure @ Virginia

→ <time> = 60 ms

(limited by distance and speed-of-light)

on-prem test:

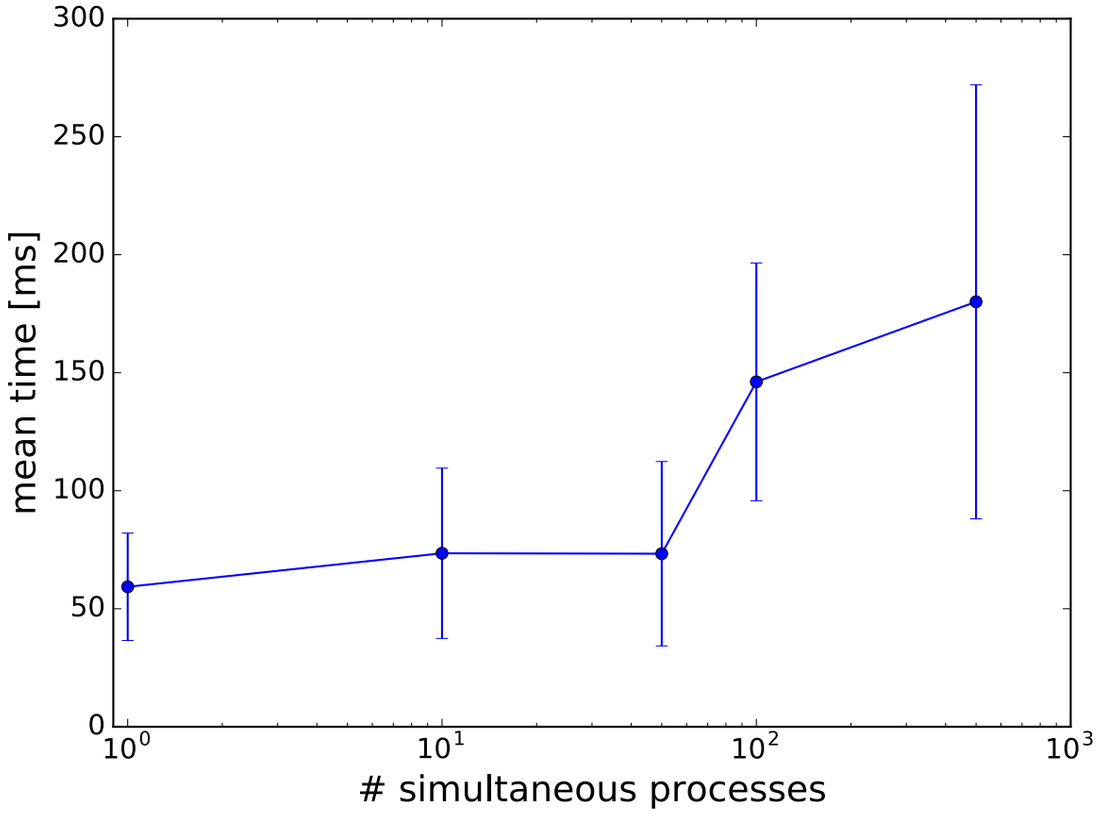
run CMSSW on Azure VM

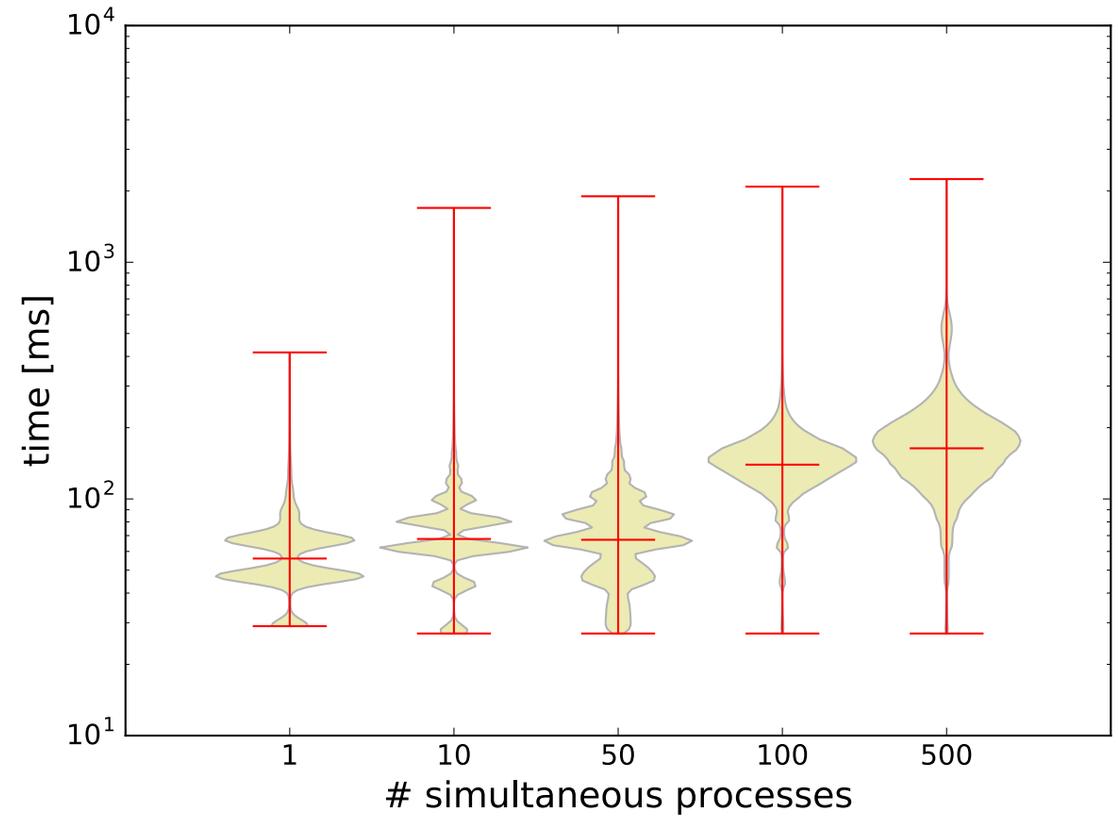
→ <time> = 10 ms

(~ 2ms on FPGA, rest is classifier and I/O)

Testing SONIC at scale

Test a large-scale deployment of cloud co-processors in a production environment





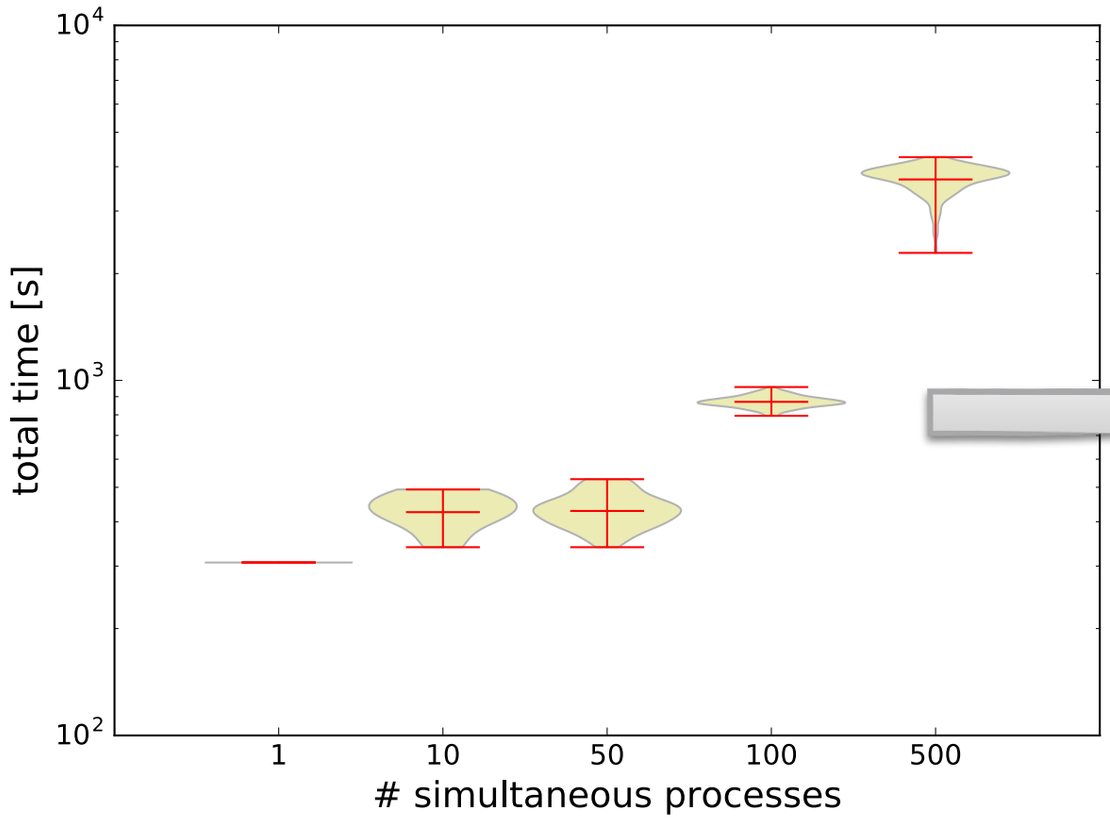
Here "worst-case" scenario: each process only executes the inference on the cloud

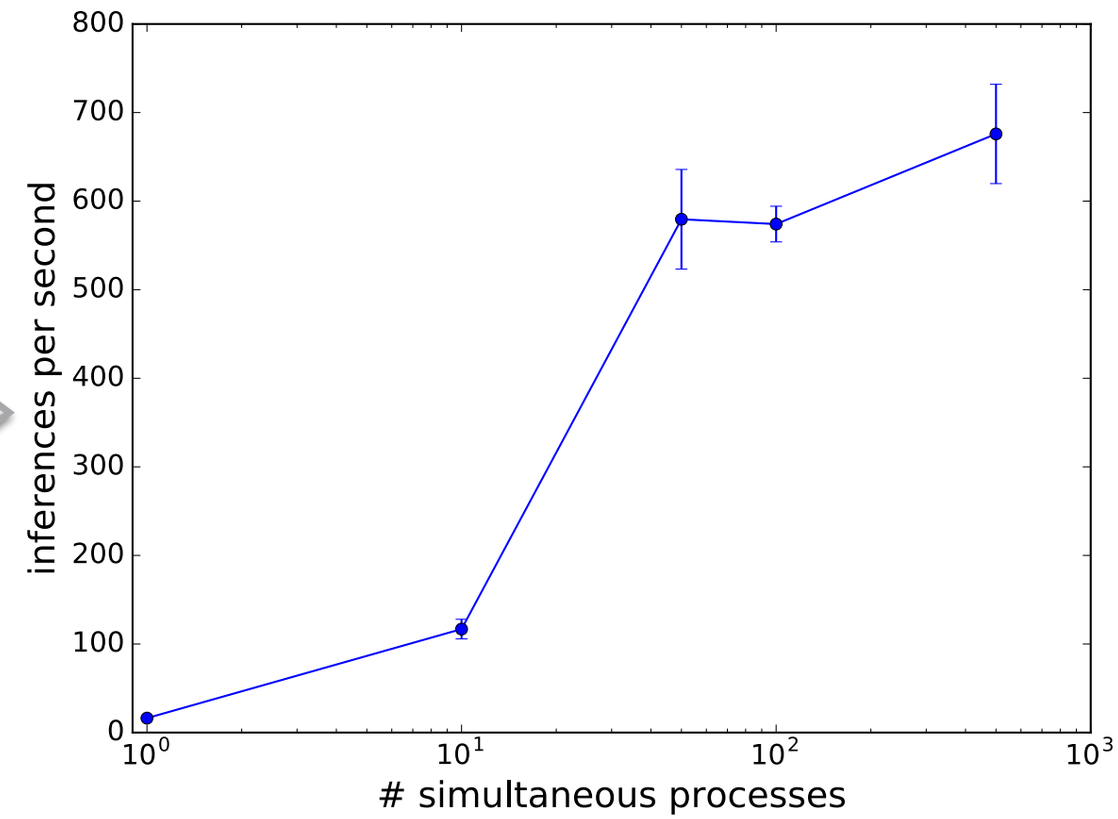
- more realistic case: inference runing alongside many other modules
  → reduced probability of simultaneous requests

1.8% failure rate only for largest number of simultaneous requests

Testing SONIC at scale

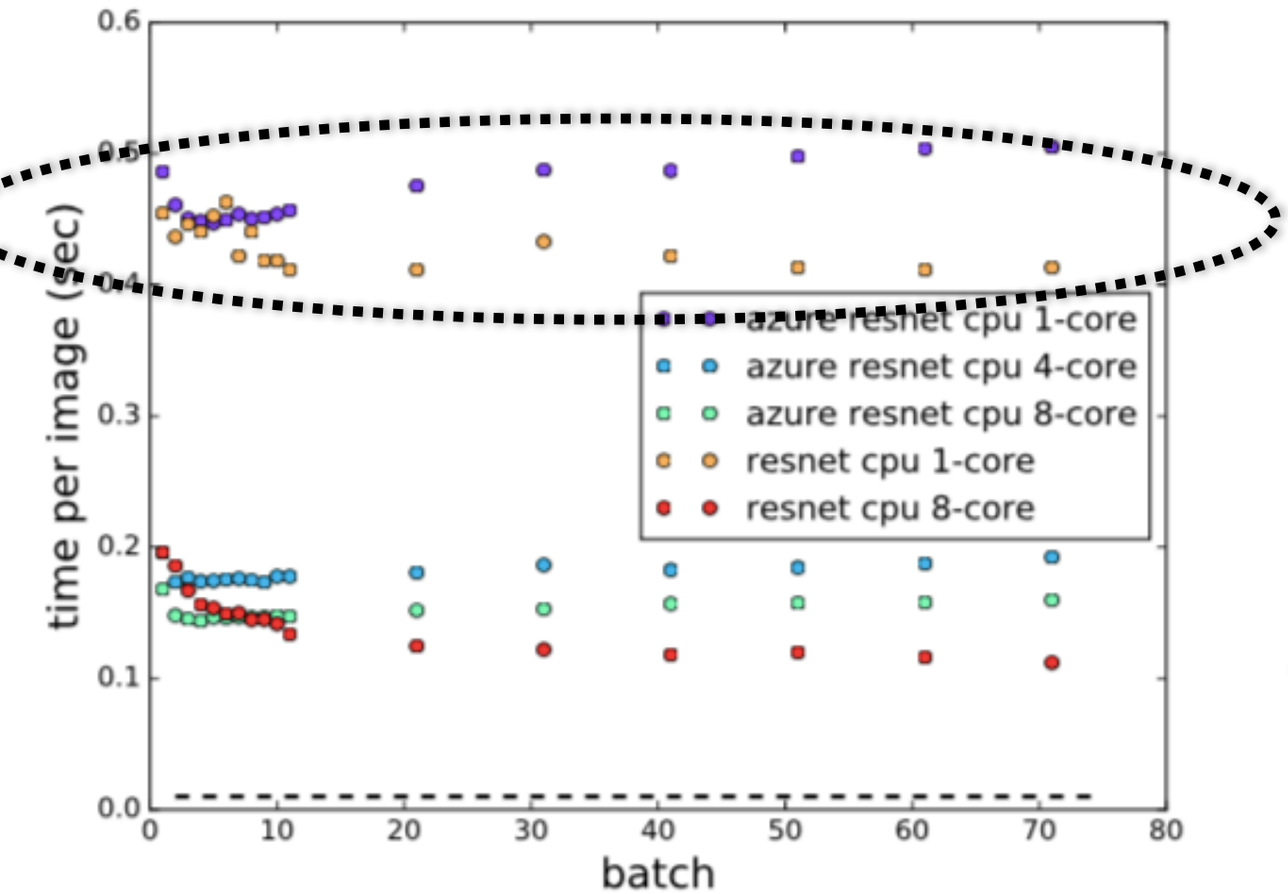
Test a large-scale deployment of cloud co-processors in a production environment

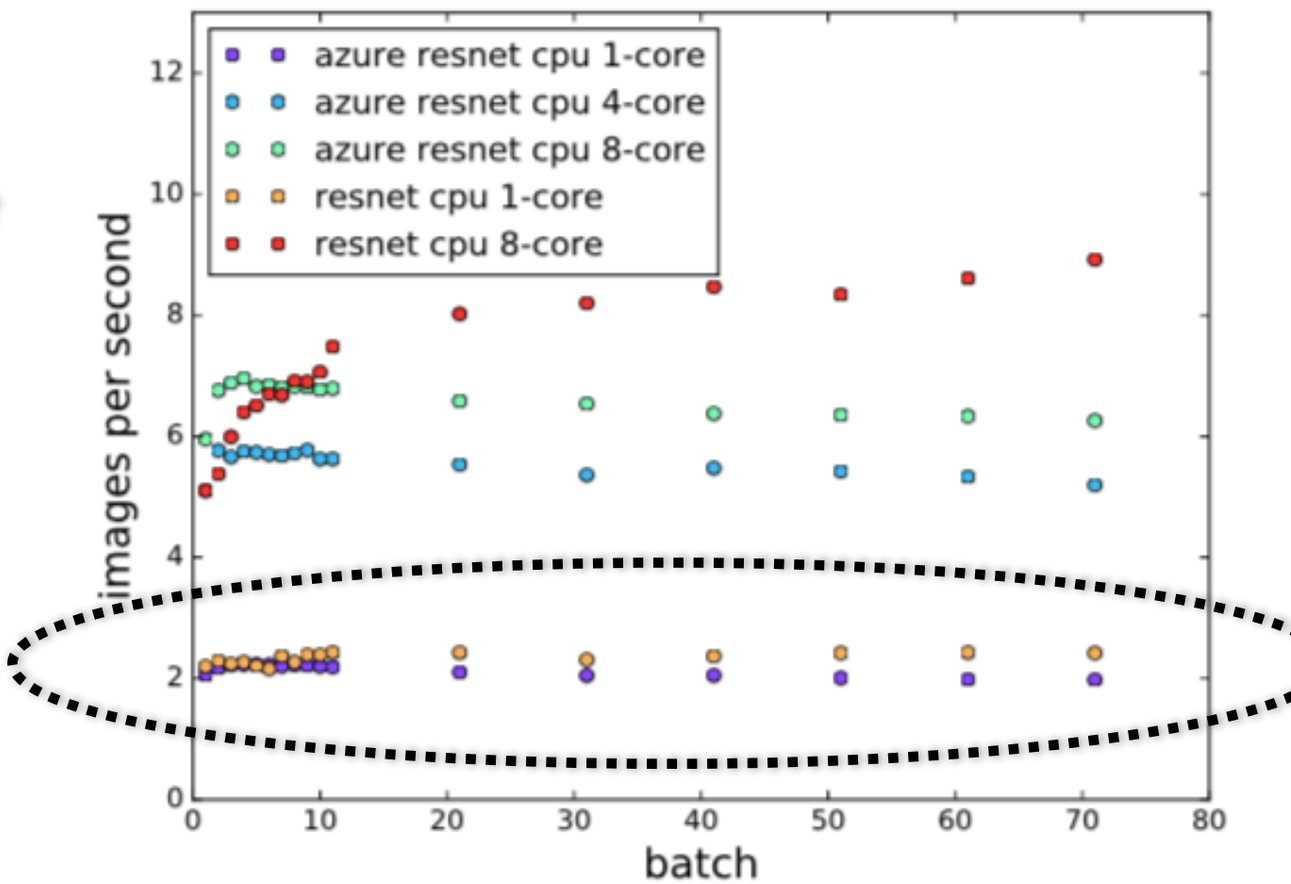




- Test with each simultaneous process completing serial processing of 5000 jet images
- Populate the pipeline of data streaming into the service
  → number of inferences per second (throughput) increases with number of simultaneous requests
- Plateau at ~ 650 inferences/s limited by FPGA inference time (~ 2ms)

Comparison with CPU

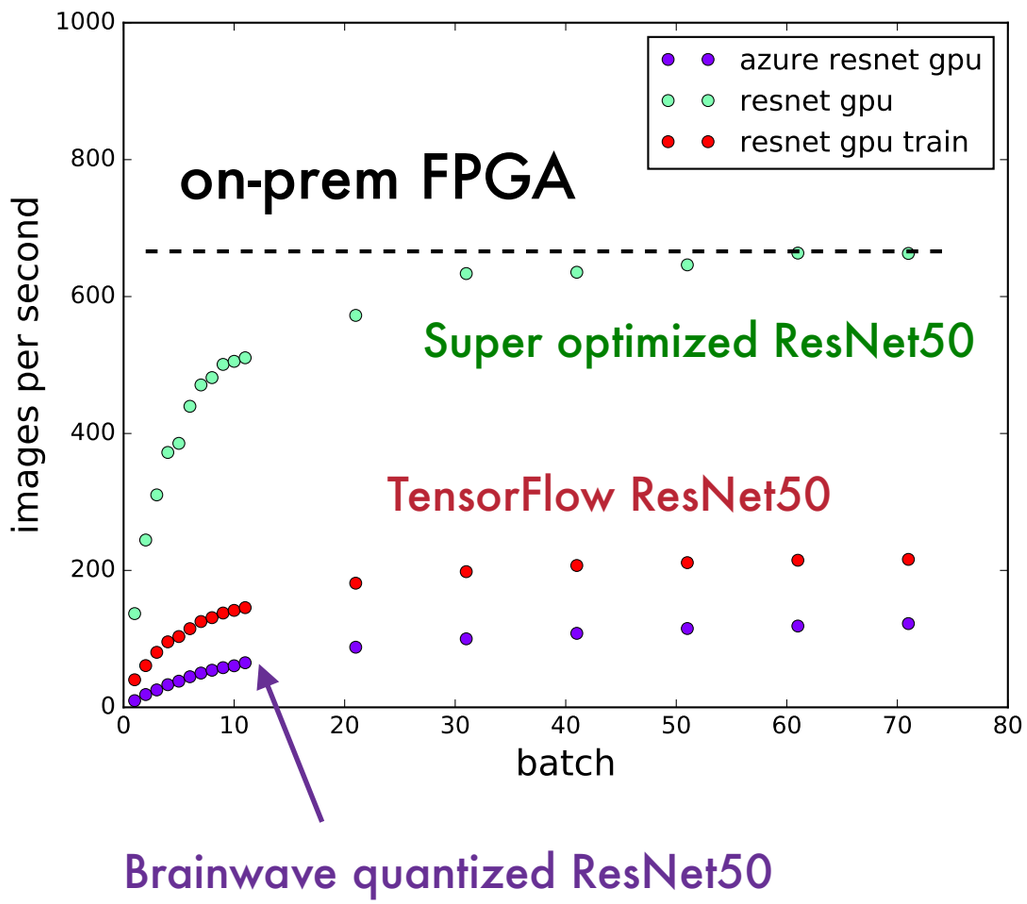


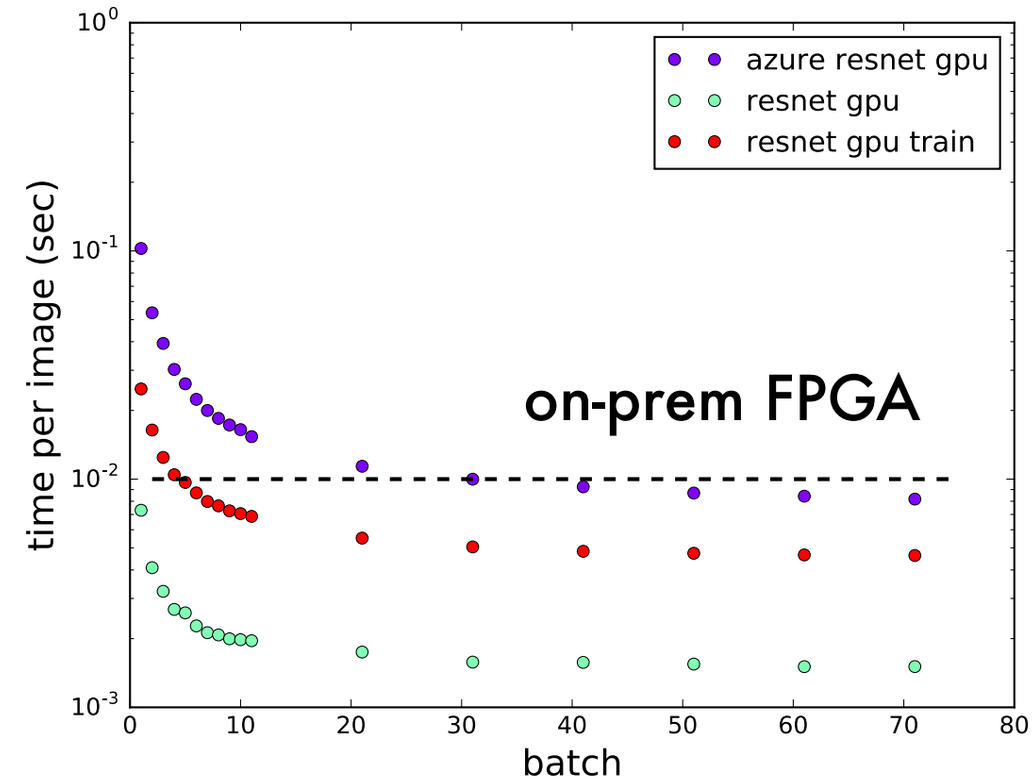


- Above plots for standalone python benchmark using i7 3.6 GHz, TensorFlow v1.10
  - inference time ~ 180 - 500 ms, 2/5 images per second

- Also run local test with CMSSW on cluster @ FNAL:
  - Xeon 2.6 GHz, TensorFlow v1.06
  - 1.75 s/inference

Comparison with GPU





- Tested GPU: NVidia GTX 1080 Ti, connected directly to CPU with PCIe (no gRPC)
- Locally connected GPU gives similar performane as on-prem/remote FPGA co-processors, but:
  - GPUs need large batch size (how to batch?)
  - PCIe vs network

Conclusions

- Current HEP computing paradigms not substainable for future requirements:
  - HL-LHC here as an example but large-scale neutrino experiments (ex: DUNE) sharing similar challenges
- Possible solution: recast physics problems into a **machine learning** problem
  - high phyiscs performance, highly parallelizable and highly supported in industry
- ML algorithms can be accelerated on ML-oriented hardware:
  GPUs, FPGAs, ASICs
- Presented proof-of-concept for acceleration on cloud FPGAs (Microsoft Brainwave)
  - for large computing tasks, **there is > x100 benefit over CPU-only computations**
  - closer clouds and edge solutions also suitable for latency-limited tasks (HLT)
- Work in progress: benchmark other platforms (Google/AWS/IBM) and keep R&D on ML algorithms for computationally intensive physics problems ...

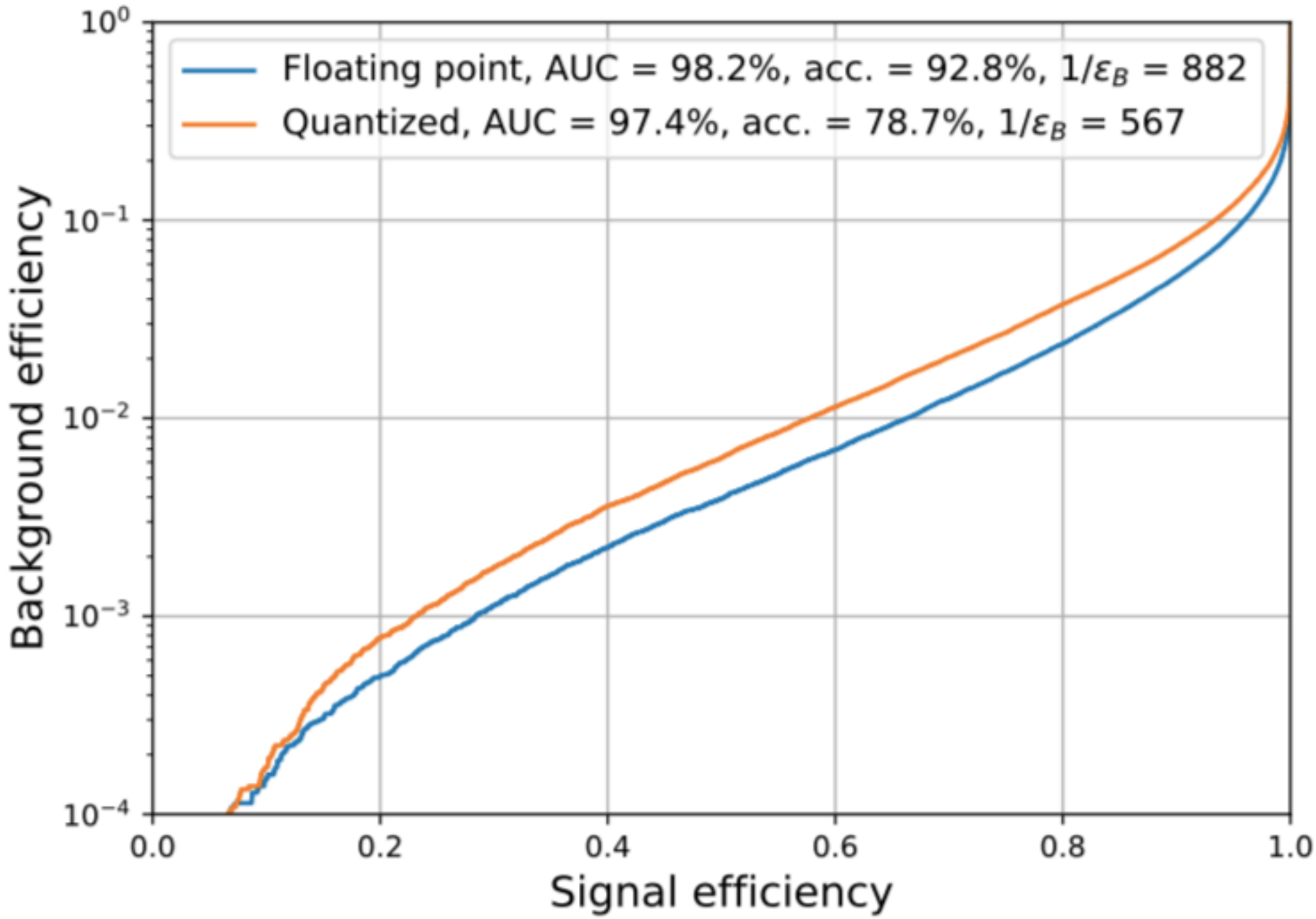
Backup slides

Summary

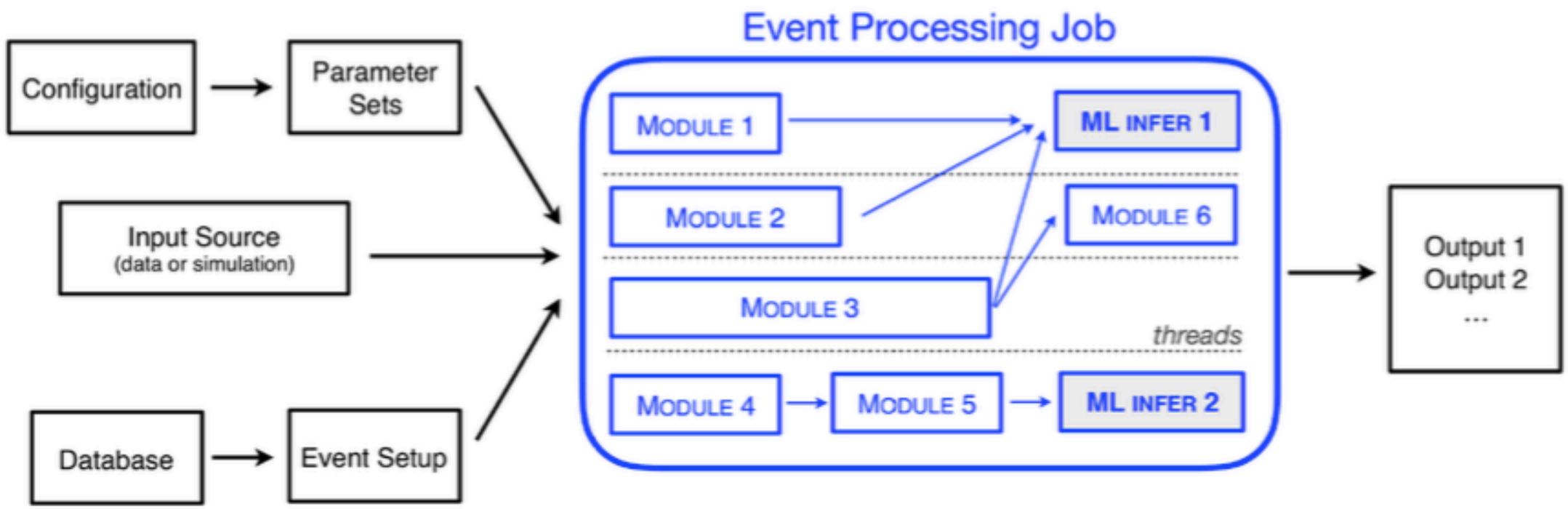
Type	Hardware	⟨Inference time⟩	Max throughput	Setup
CPU	Xeon 2.6 GHz, 1 core	1.75 seconds	0.6 img/s	`CMSSW`, `TF v1.06`
CPU	i7 3.6 GHz, 1 core	500 ms	2 img/s	python, `TF v1.10`
CPU	i7 3.6 GHz, 8 core	200 ms	5 img/s	python, `TF v1.10`
GPU (batch=32)	NVidia GTX 1080	100 ms	10 img/s	python, `TF v1.10`
GPU (batch=1)	NVidia GTX 1080	9 ms	111 img/s	python, `TF v1.10`
GPU (batch=32)	NVidia GTX 1080	7 ms	143 img/s	TF internal, `TF v1.10`
GPU (batch=1)	NVidia GTX 1080	1.5 ms	667 img/s	TF internal, `TF v1.10`
Brainwave	Altera Artix	10 ms	660 img/s	`CMSSW`, *on-prem*
Brainwave	Altera Artix	60 ms	660 img/s	`CMSSW`, *remote*

A factor x175 (x30) speedup for Brainwave on-prem (remote) over current CMSSW CPU performance.

ResNet-50 performance



Particle physics computing model



- Event-based processing
  - Events are very complex with hundreds of products
  - 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