SONIC

coprocessors as a service for accelerated inference of DL algorithms

Jeffrey Krupa, Philip Harris, Dylan Rankin, Jack Dinsmore, Tri Nguyen, Erik Katsavounidis (MIT)
Maria Acosta Flechas, Burt Holzman, Thomas Klijnsma, Kevin Pedro, Nhan Tran, Michael Wang, Tingjun Yang (FNAL)
Scott Hauck, Shih-Chieh Hsu, Matthew Trahms, Kelvin Lin, Yu Lou, Natchanon Suaysom (University of Washington)
Ta-Wei Ho (National Tsing Hua University)
Javier Duarte (UCSD)
Mia Liu (Purdue University)
+Fast Machine Learning team: fastmachinelearning.org

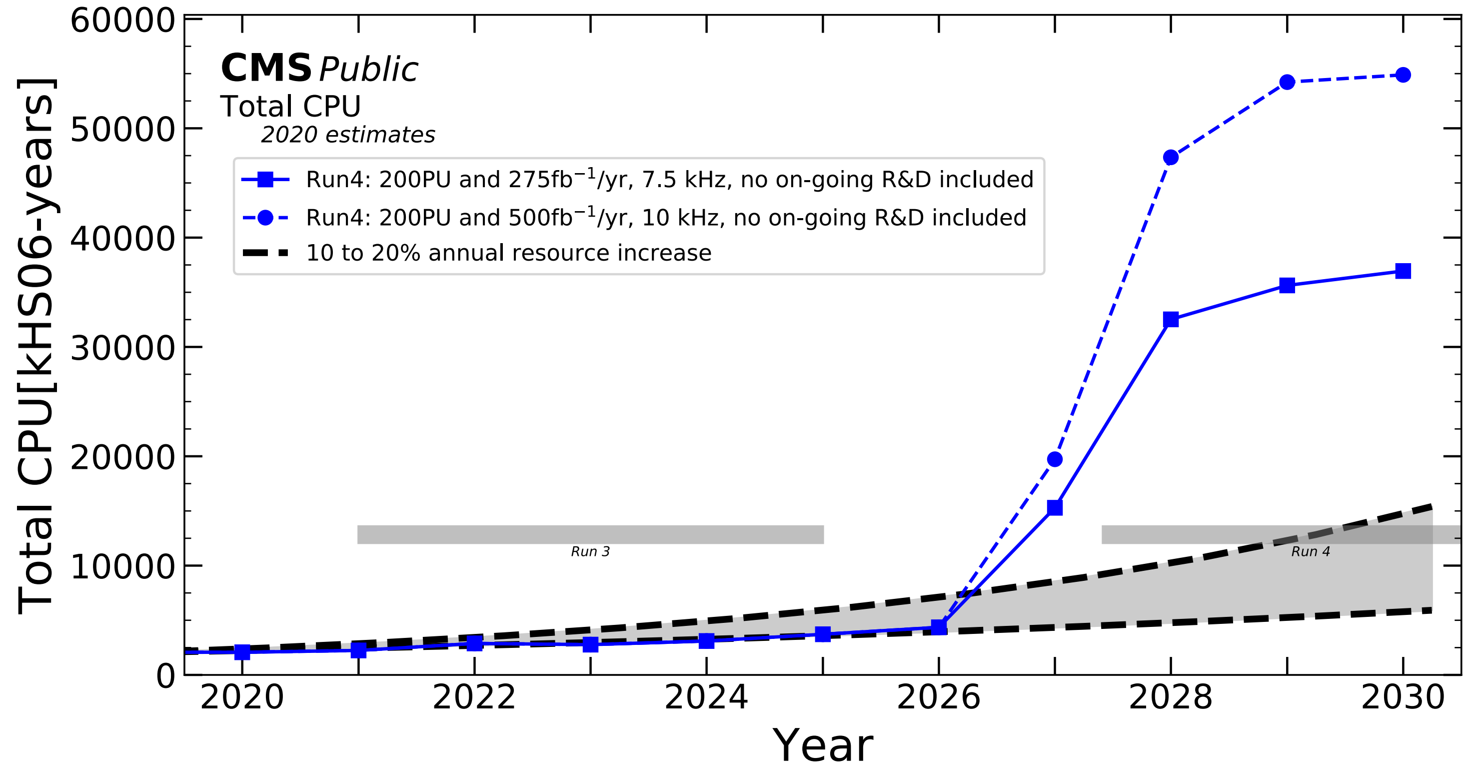
CPAD TDAQ session
March 19th, 2021

Overview

- We present SONIC, a framework for integrating GPUs and FPGAs as a service (aaS) into physics workflows
- We present case studies of integrating GPUs/FPGAs aaS into:
  - LHC experiments: GPU paper, FPGA paper
  - neutrino experiments: ProtoDUNE paper
  - Gravitational waves: LIGO denoising talk

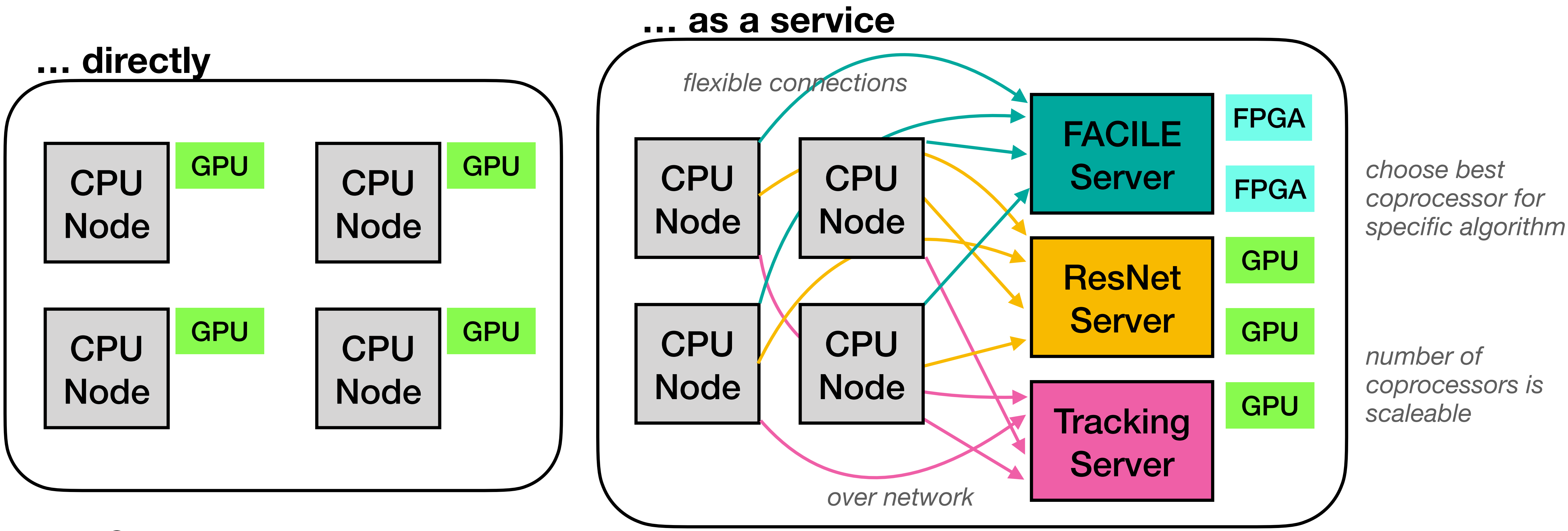
Introduction

- Computing needs at LHC experiments will outpace expected growth in CPU performance



- Compounded by interest in DL algorithms
  - Pervasive in analysis context, but slowly moving to data taking
- Coprocessors (GPUs, FPGAs, …) are a solution to this problem

Connecting to coprocessors…



Communicating with coprocessors as a service:

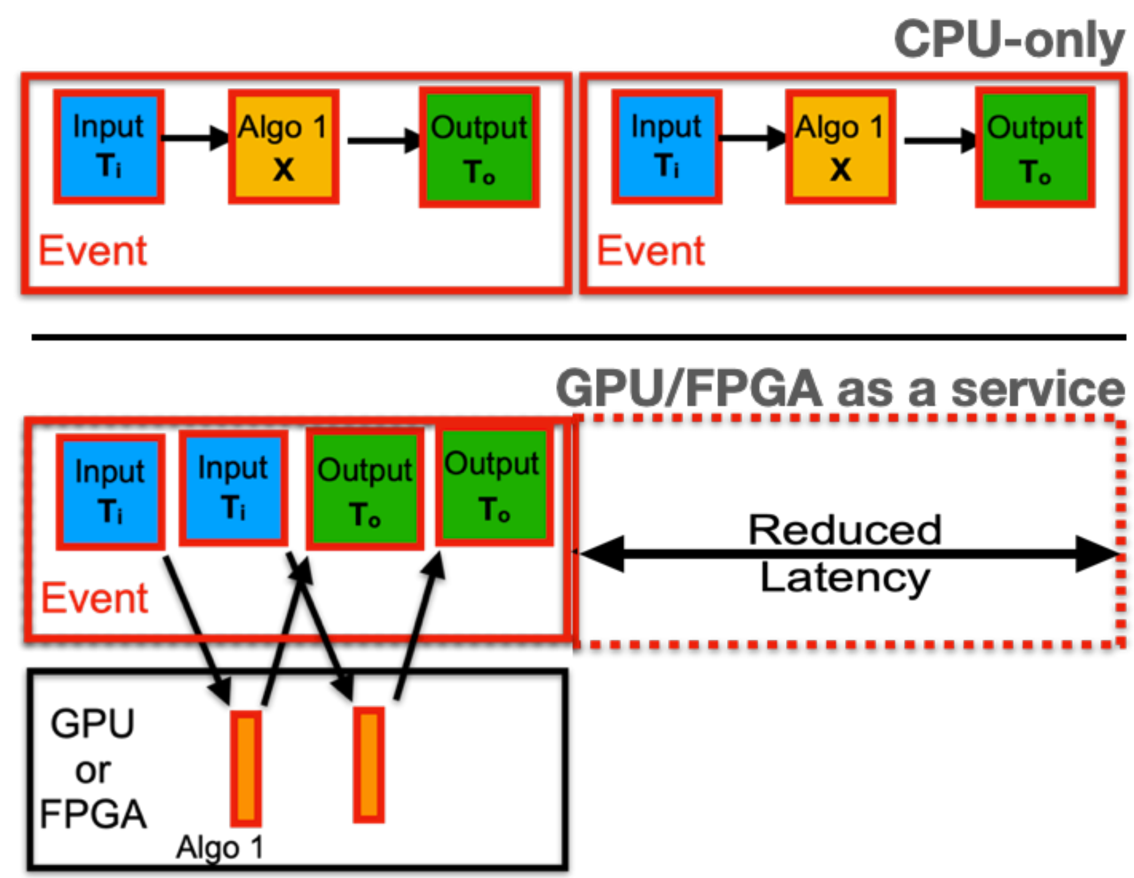
1. Enables integration of coprocessors without larger redesign of computing system
2. Removes burden of writing any algorithm-specific coprocessor code
3. Is heterogeneous friendly
   - Can flexibly configure coprocessor type, number of coprocessors per server, …
   - Many coprocessors to choose from
4. Leverages highly optimized inference tools developed by industry

Considerations: added network load, load balancer, sufficient algorithm speedup

SONIC

Services for Optimized Network Inference on Coprocessors

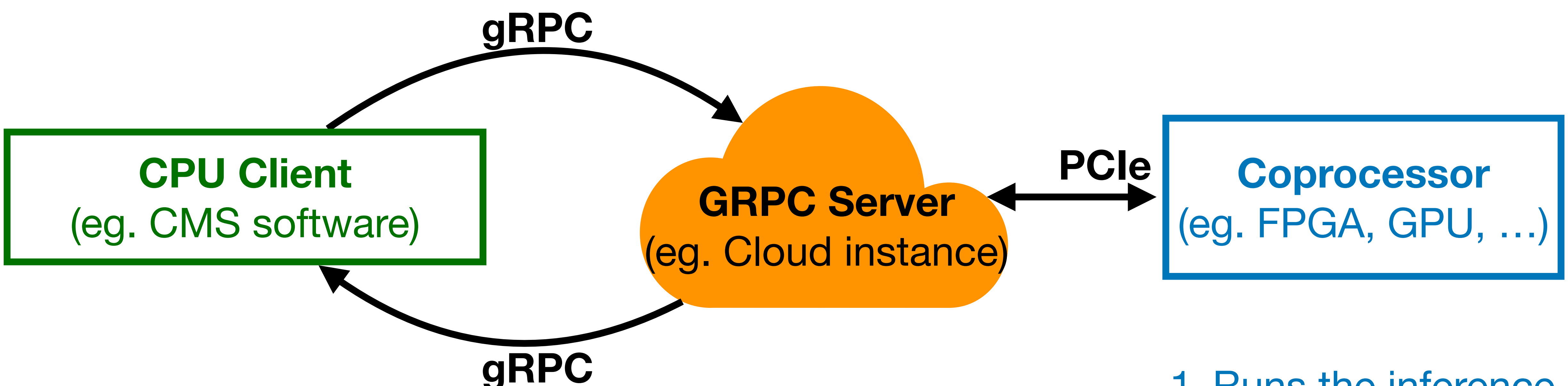
- Integrates as-a-service requests into HEP workflows
- Formats event data for algorithm input
- Makes non-blocking, asynchronous requests
- Works with any coprocessor
- Integrated into CMS software



SONIC

Services for Optimized Network Inference on Coprocessors

- For fast inference we focus on remote procedure call (gRPC) protocol
- Use Triton inference server for inference on NVIDIA GPUs
- Developed custom FPGAs-as-a-Service Toolkit (FaaST) for FPGA

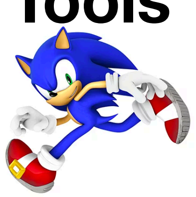


1. Runs the inference

1. Formats inputs
2. Sends asynchronous, non-blocking gRPC call
3. Interprets response

1. Initializes model on coprocessor
2. Receives and schedules inference request
3. Sends inference request
4. Outputs and send results

Tools



Use NVIDIA triton inference server for GPU + Customized GCP Kubernetes

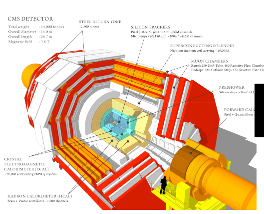
Wrote our own FPGA gRPC inference server

LHC data flow

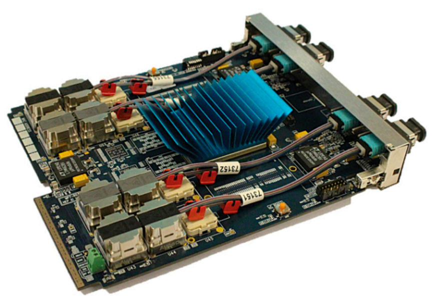
40 MHz ——————————————————————→ **1 kHz**

320 Tb/s — 10 Gb/s

Radiation Hard ASICs



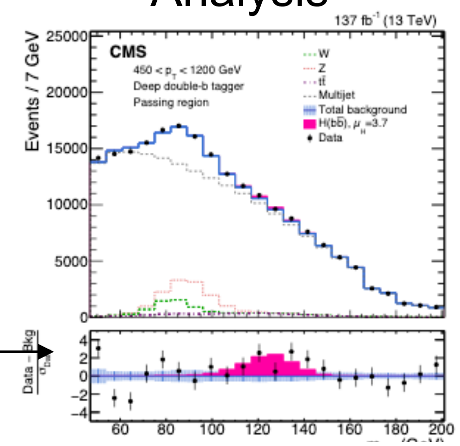
Level 1 Trigger

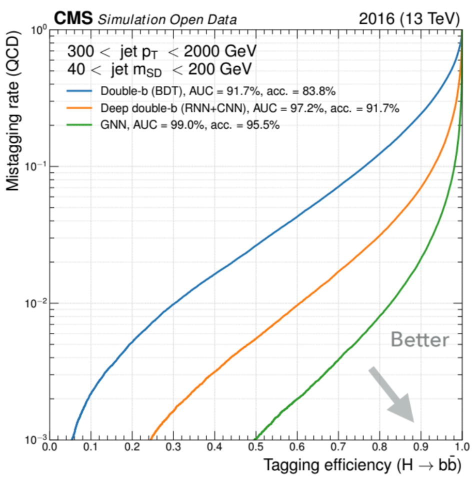


High Level Trigger

Offline reconstruction

Analysis





Fast
10 μs window
L1Trigger

Intermediate
<500 ms window
High Level Trigger

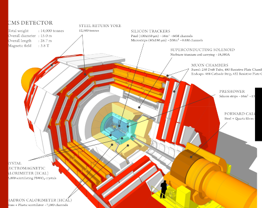
Slow
10 s window
Offline Cluster

LHC data flow

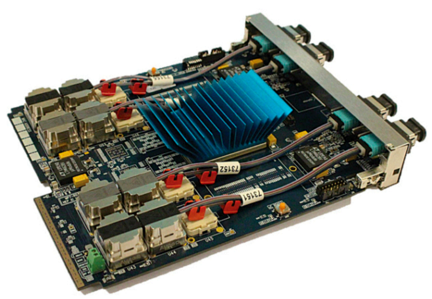
40 MHz → **1 kHz**

320 Tb/s → 10 Gb/s

Radiation Hard ASICs



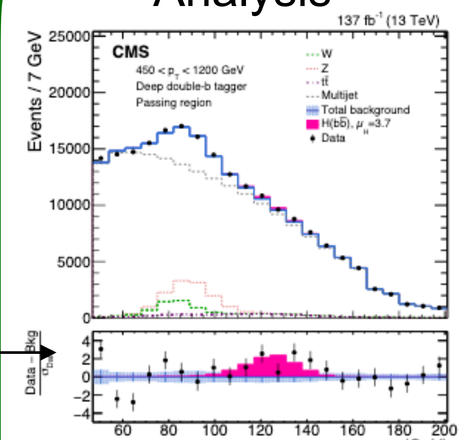
Level 1 Trigger

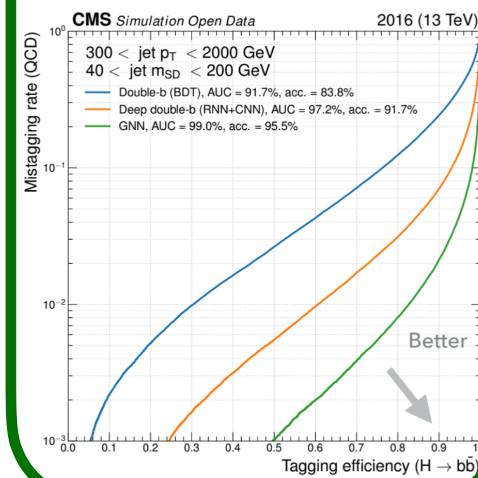


High Level Trigger

Offline reconstruction

Analysis





Fast
10 μs window
L1Trigger

Intermediate
<500 ms window
High Level Trigger

Slow
10 s window
Offline Cluster

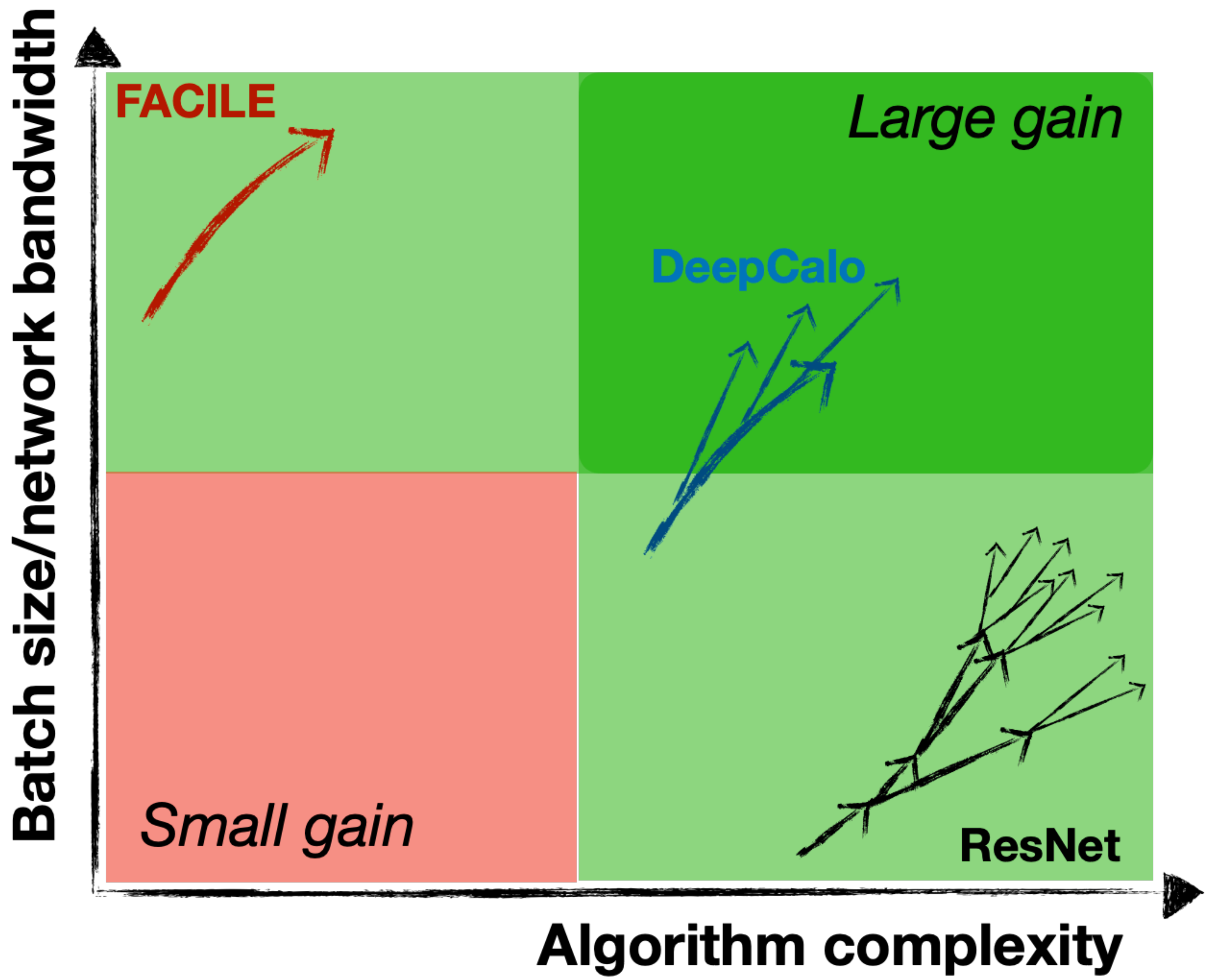
This work focuses on introducing DL+heterogeneity in data taking

DL (+GPUs) is often done on a user-specific basis

See Jim Hirschauer's talk   *See Jennifer Ngadiuba's talk*

Benchmark algorithms for HEP

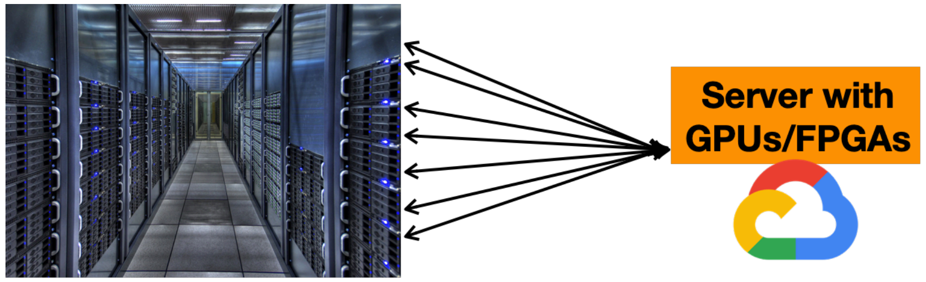
- Gains at large batch and large algorithm complexity/operations
- The algorithm has to be sufficiently sped-up for transfer to not reduce throughput
  - Each algorithm performs as well on physics objects than a corresponding CPU algorithm

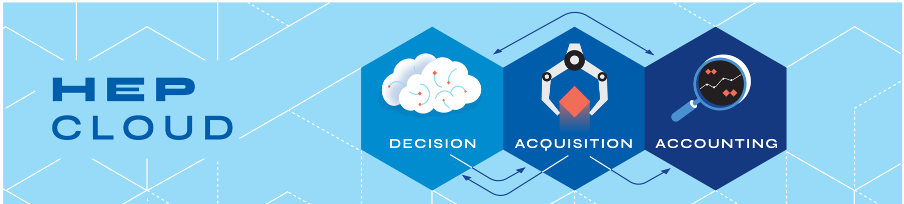


GPU/FPGA aaS	***Gain w.r.t. CPU***
2 ms (GPU) 0.2 ms (FPGA)	**8x (GPU) 80x (FPGA)**
0.1 ms (GPU) in progress (FPGA)	**50x 750x***
1-2 ms (GPU/FPGA)	**500x**

uses dynamic batching optimization

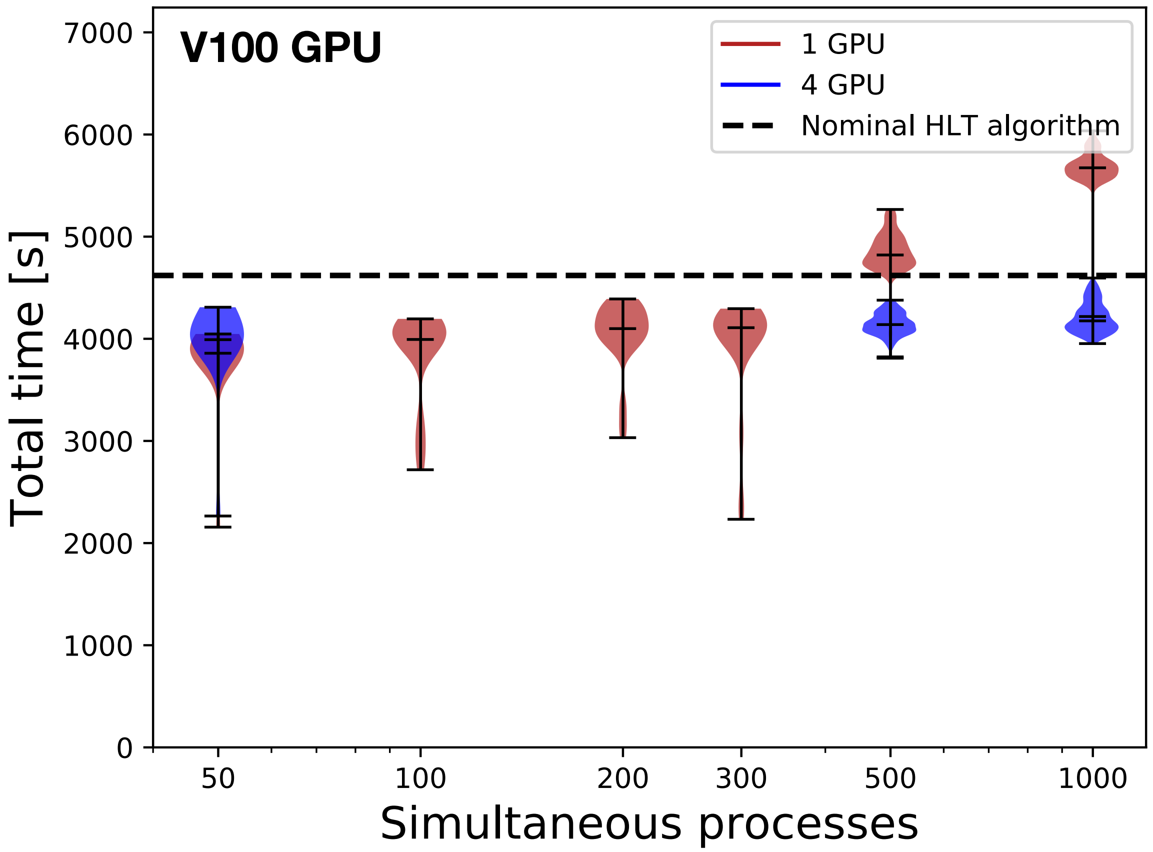
Online reconstruction



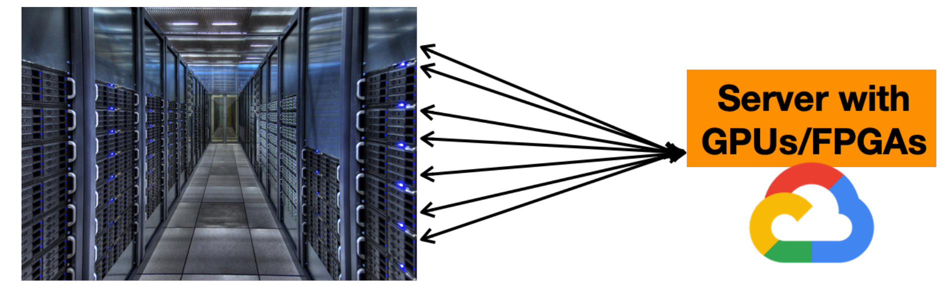


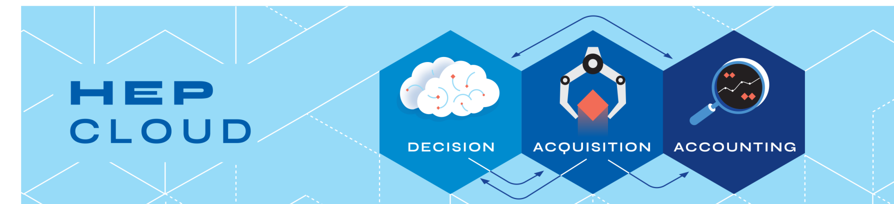
- Simplest point of integration aaS: hadron calorimeter local reconstruction algorithm: low latency, high batch
- Scale test of the CMS High Level Trigger (HLT) in Google Cloud
- HLT instances and server deployed at same site

2007.10359



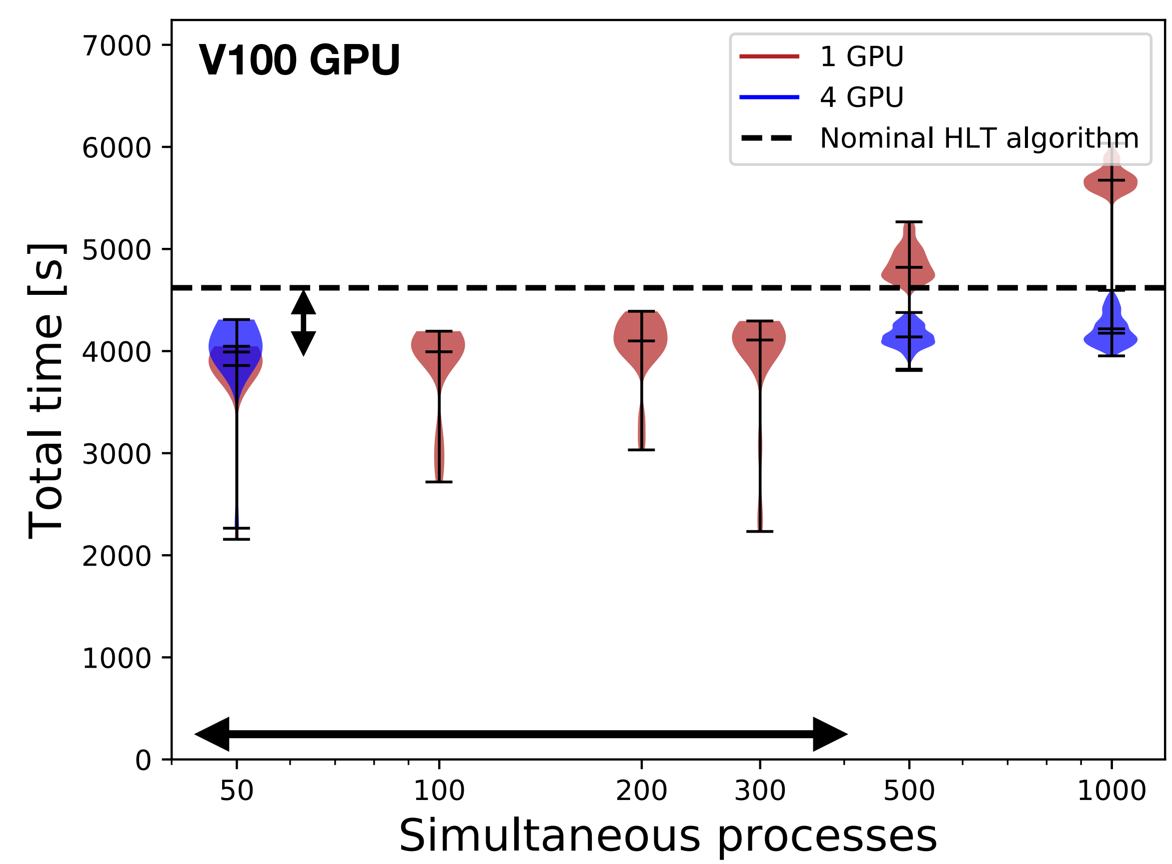
Online reconstruction





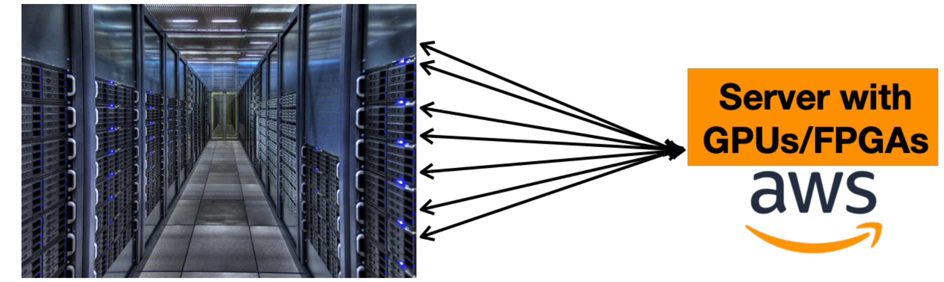
- Simplest point of integration aaS: hadron calorimeter local reconstruction algorithm: low latency, high batch
- Scale test of the CMS High Level Trigger (HLT) in Google Cloud
- HLT instances and server deployed at same site

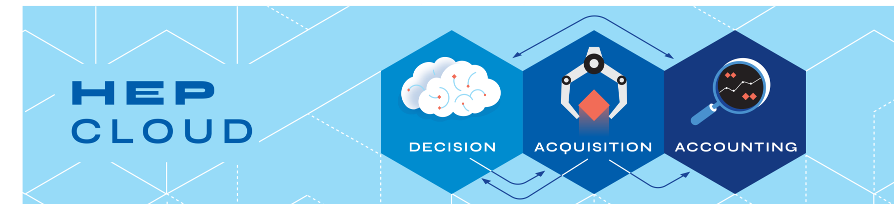
2007.10359



1. 10% reduction in CMS HLT latency
   - Removes HCAL from HLT budget
2. 300 HLT instances can be serviced by a single GPU
3. No network concerns intra-site

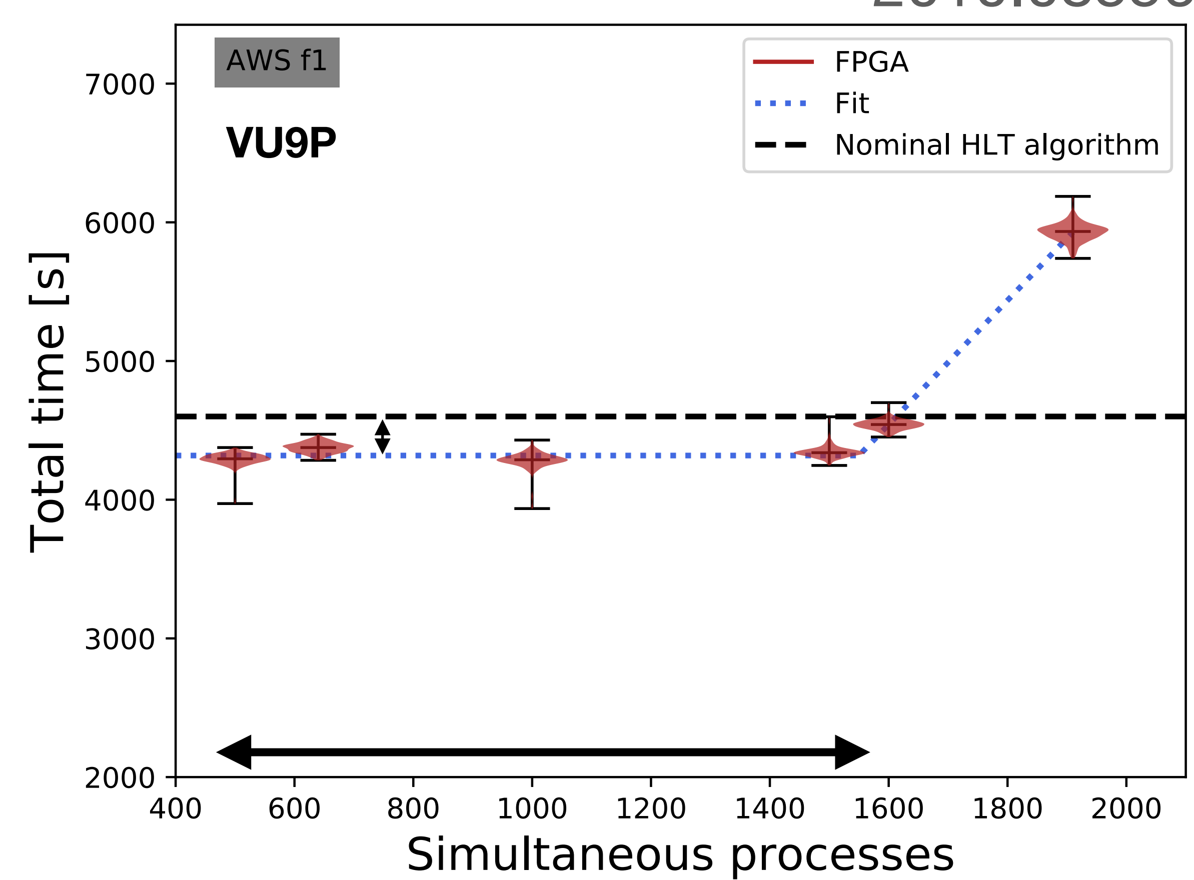
Online reconstruction





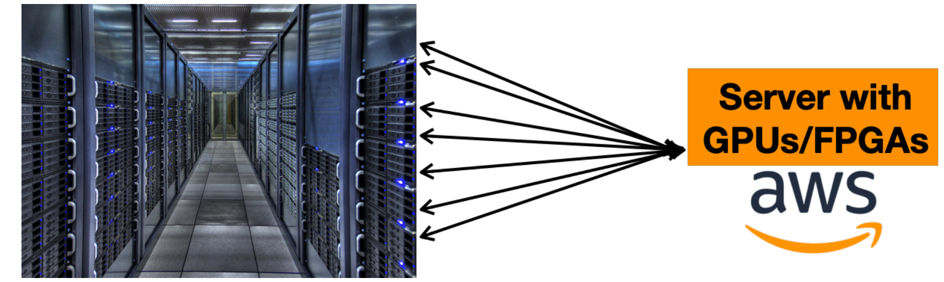
- HLT test with HCAL reconstruction executed on FPGA server
- Uses pipeline of all super logic regions (SLRs) of FPGA
- Developed FPGA-as-a-service Toolkit for FPGA servers
- Limiting factor is 25 Gb/s into FPGA (batch 16000)

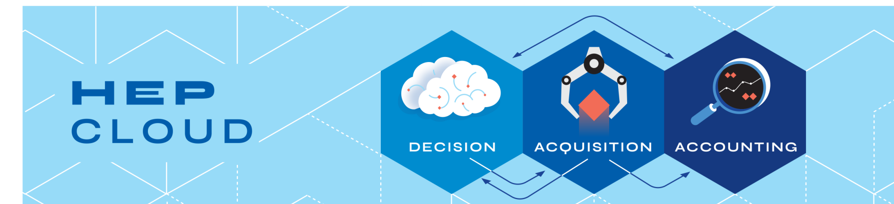
2010.08556



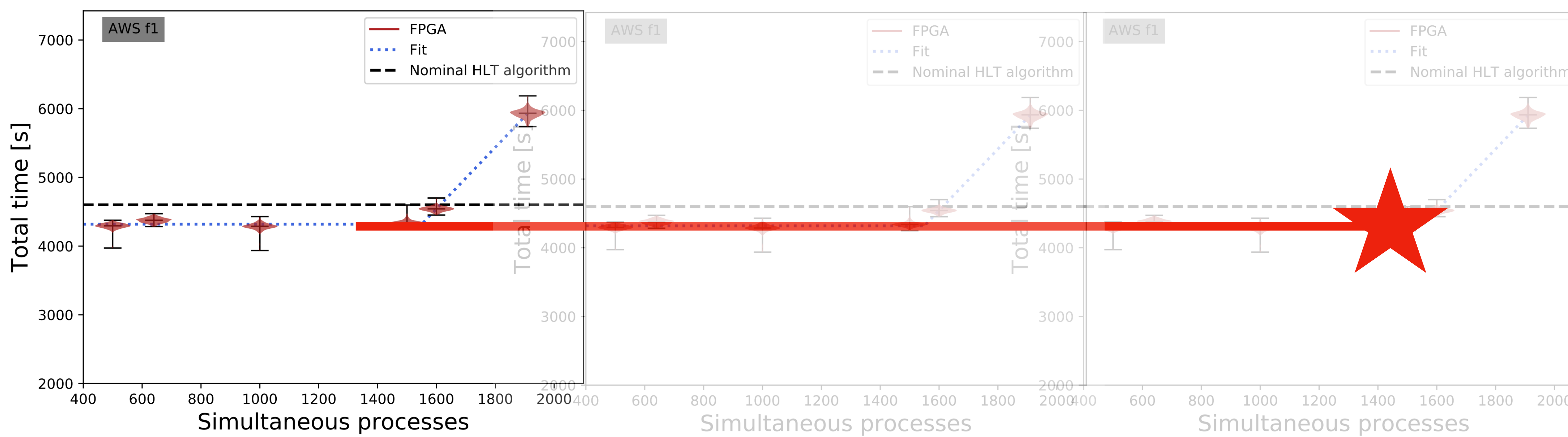
1. Similar 10% reduction in HLT latency
2. 1500 HLT instances can be serviced by a single FPGA

Online reconstruction





- HLT test with HCAL reconstruction executed on FPGA server
- Uses pipeline of all super logic regions (SLRs) of FPGA
- Developed FPGA-as-a-service Toolkit for FPGA servers
- Limiting factor is 25 Gb/s into FPGA (batch 16000)

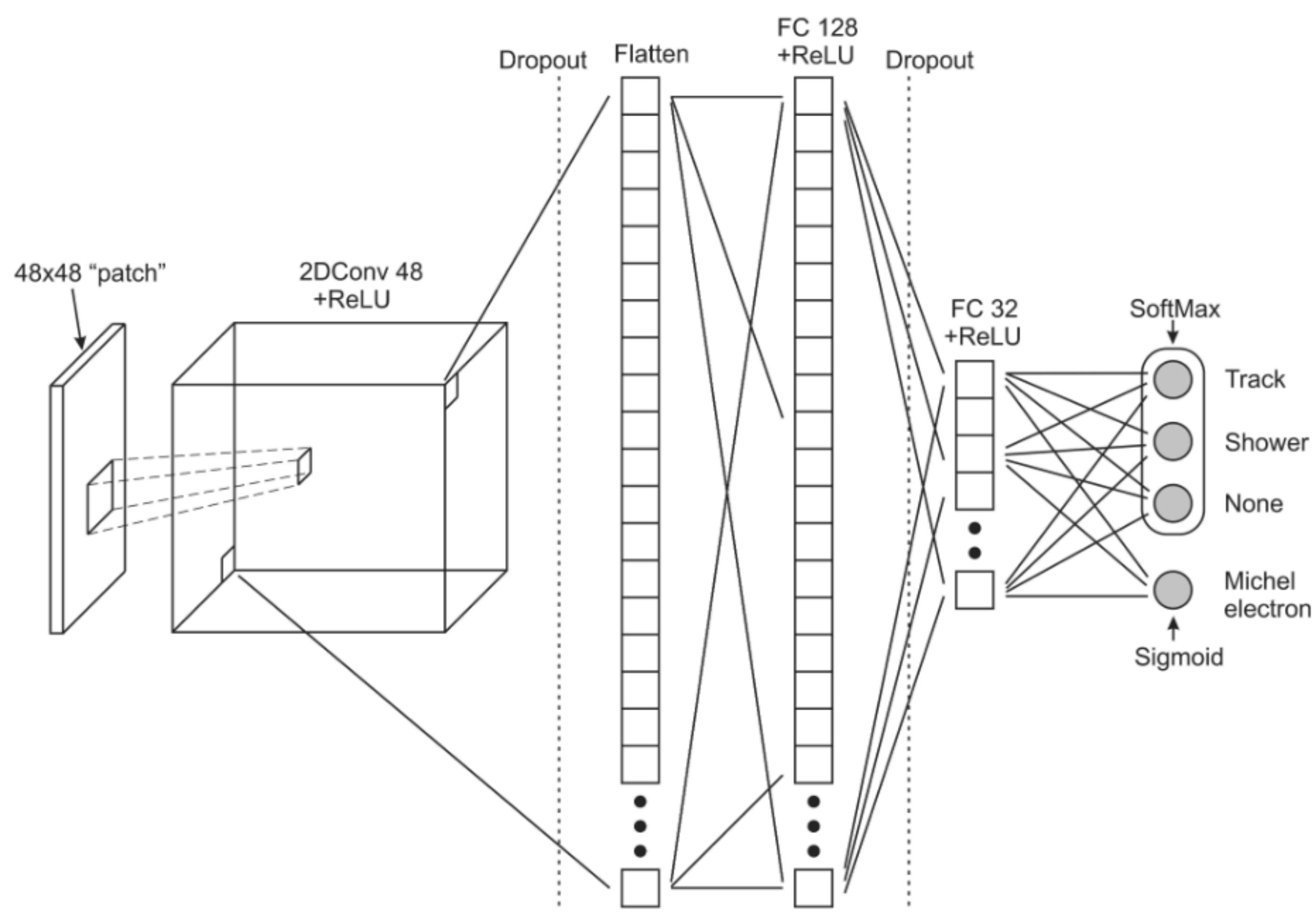


Limit without 25 Gb/s bottleneck is 5500 simultaneous processes

2010.08556

ProtoDUNE

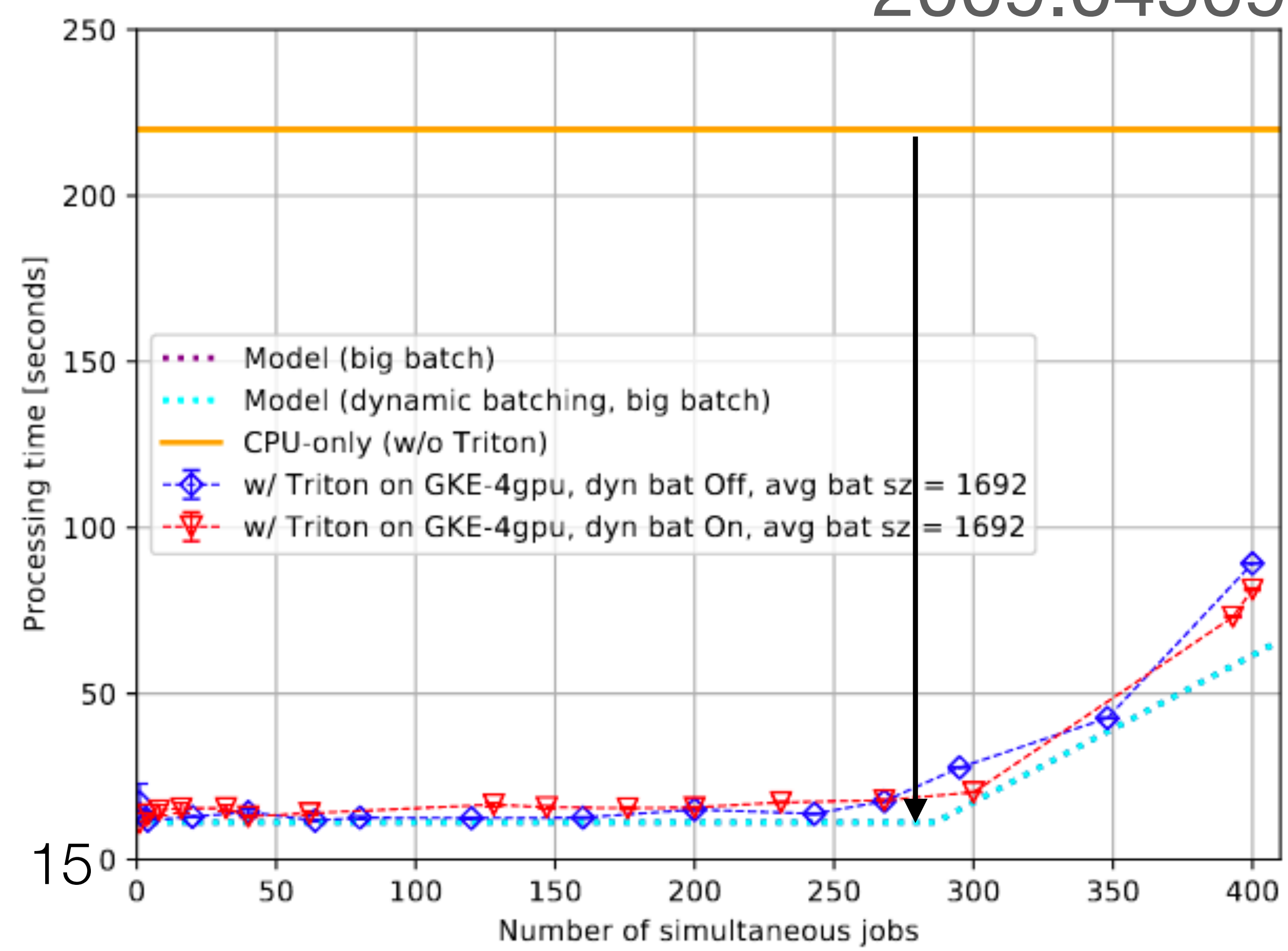
- ProtoDUNE is a testbed for the Deep Underground Neutrino Experiment
- 2/3 of the reconstruction workflow latency is from *EmMichelTrackId*
  - 2D CNN classifies electron as a track, shower, or Michel electron



ProtoDUNE

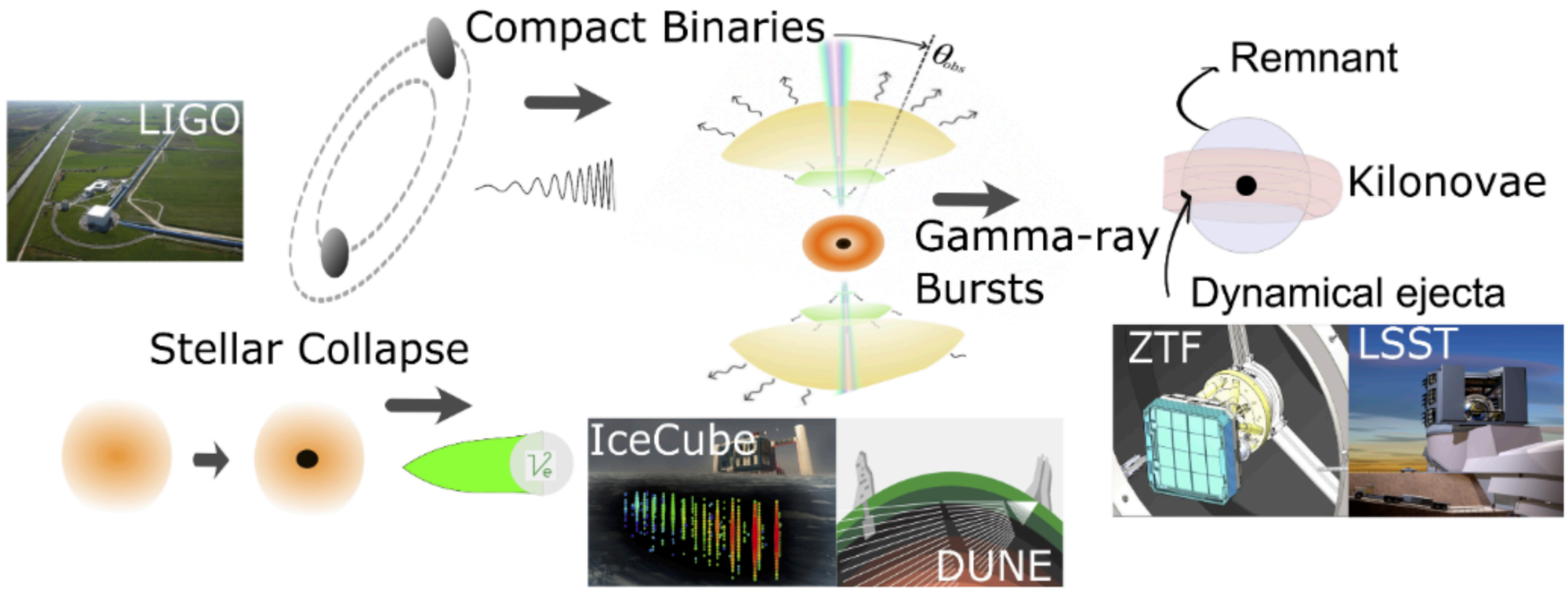
- ProtoDUNE is a testbed for the Deep Underground Neutrino Experiment
- 2/3 of the reconstruction workflow latency is from *EmMichelTrackId*
    - 2D CNN classifies electron as a track, shower, or Michel electron
- Deploying to GPUs as a service reduces algorithm latency by 17x
    - Reduces entire compute by 2.7x
    - Hardware efficient (70 CPU served by single GPU)
    - Related to trigger efforts at DUNE

2009.04509



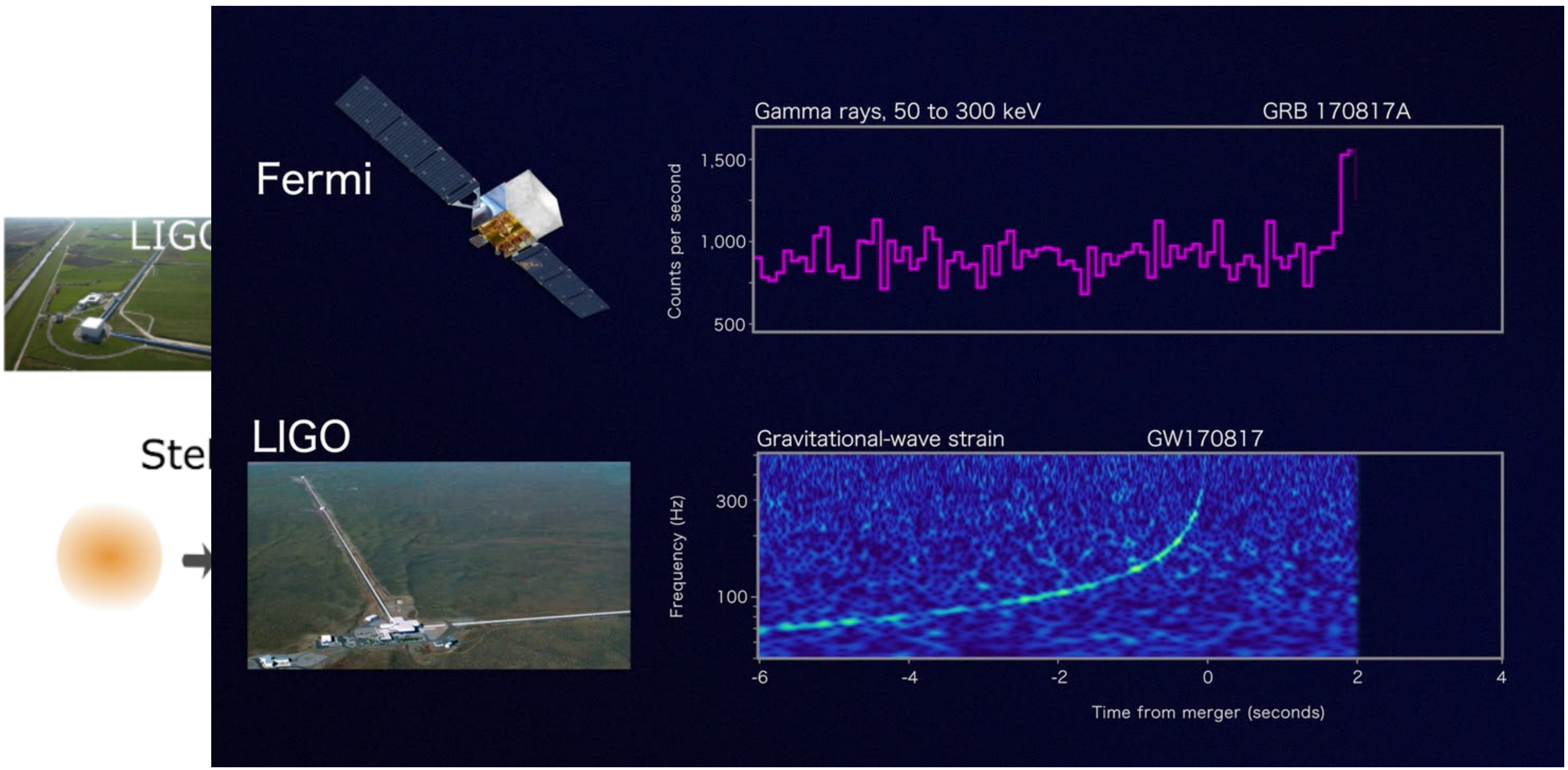
Multi-messenger astrophysics

- Gravitational waves, photons, neutrinos, and cosmic rays carry complementary information about astrophysical events
- Fast inference of LIGO information could help telescopes orient faster



Multi-messenger astrophysics

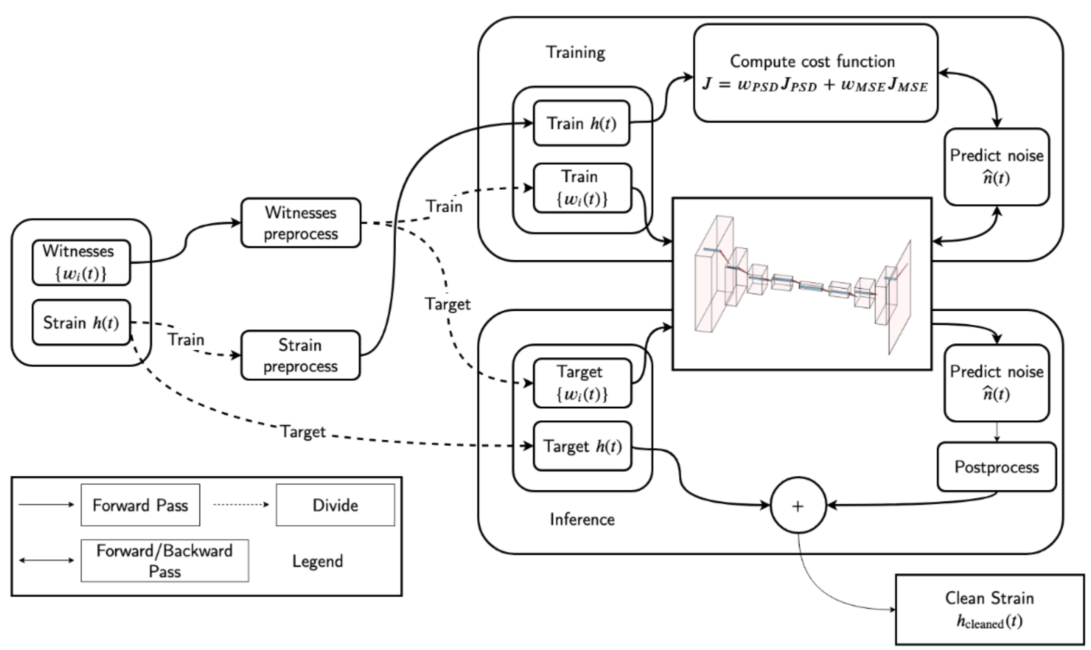
- Gravitational waves, photons, neutrinos, and cosmic rays carry complementary information about astrophysical events
- Fast inference of LIGO information could help telescopes orient faster



Co-incident Gamma Ray Burst and GW

Multi-messenger astrophysics: LIGO

- End-to-end from noisy LIGO strain time series to classification
  - Ensemble of two CNNs
    1. denoising (2005.06534)
    2. binary black hole merger classification (1701.00008)
- Working on a full demonstration of real-time GW processing

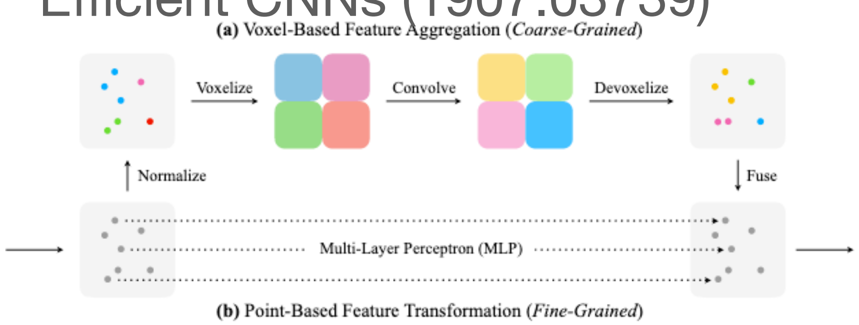


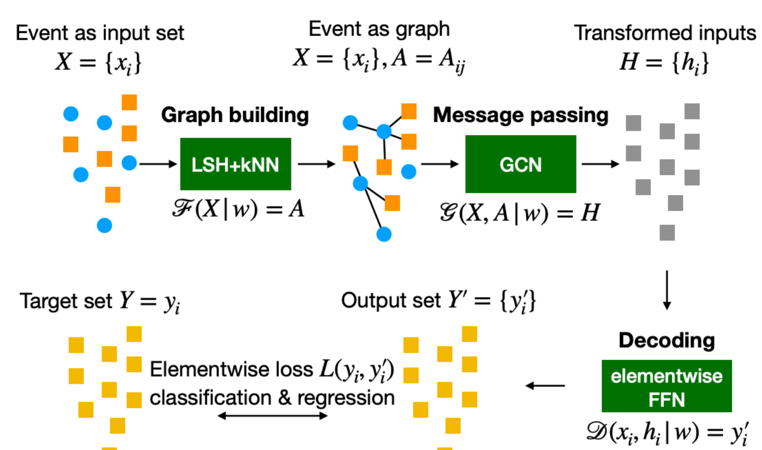
Next steps

- Explore HPCs
- Expand to more physics problems (e.g. clustering, jet tagging) with new architectures (e.g. graph neural networks, particle clouds)
- Investigate new coprocessors (eg Intelligence Processing Unit)

HPC

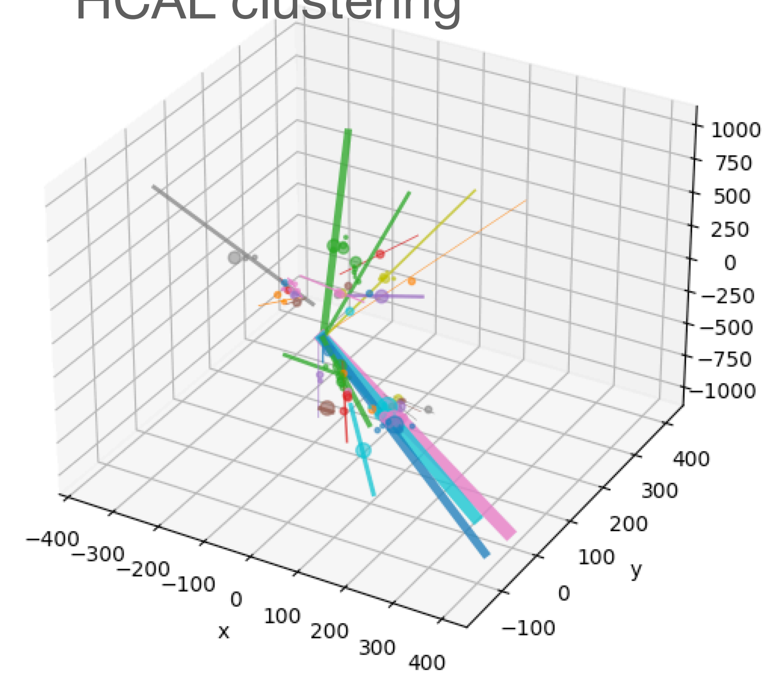
Efficient CNNs (1907.03739)





Graph neural networks

HCAL clustering



Summary

- As-a-service paradigm introduces coprocessors to HEP with minimal changes to pre-existing computing workflows
- SONIC enables user to write simple client code, offloading heavy algorithms onto optimized inference servers with asynchronous call
- FPGA integration added through FPGA-as-a-service Toolkit
- Demonstration of scaled CMS HLT sped-up with hadron calorimeter reconstruction performed on GPUs and FPGAs
- SONIC can serve as a useful tool for online and offline LHC reconstruction
- SONIC framework provides value for other physics experiments, including protoDUNE and LIGO

Thanks!

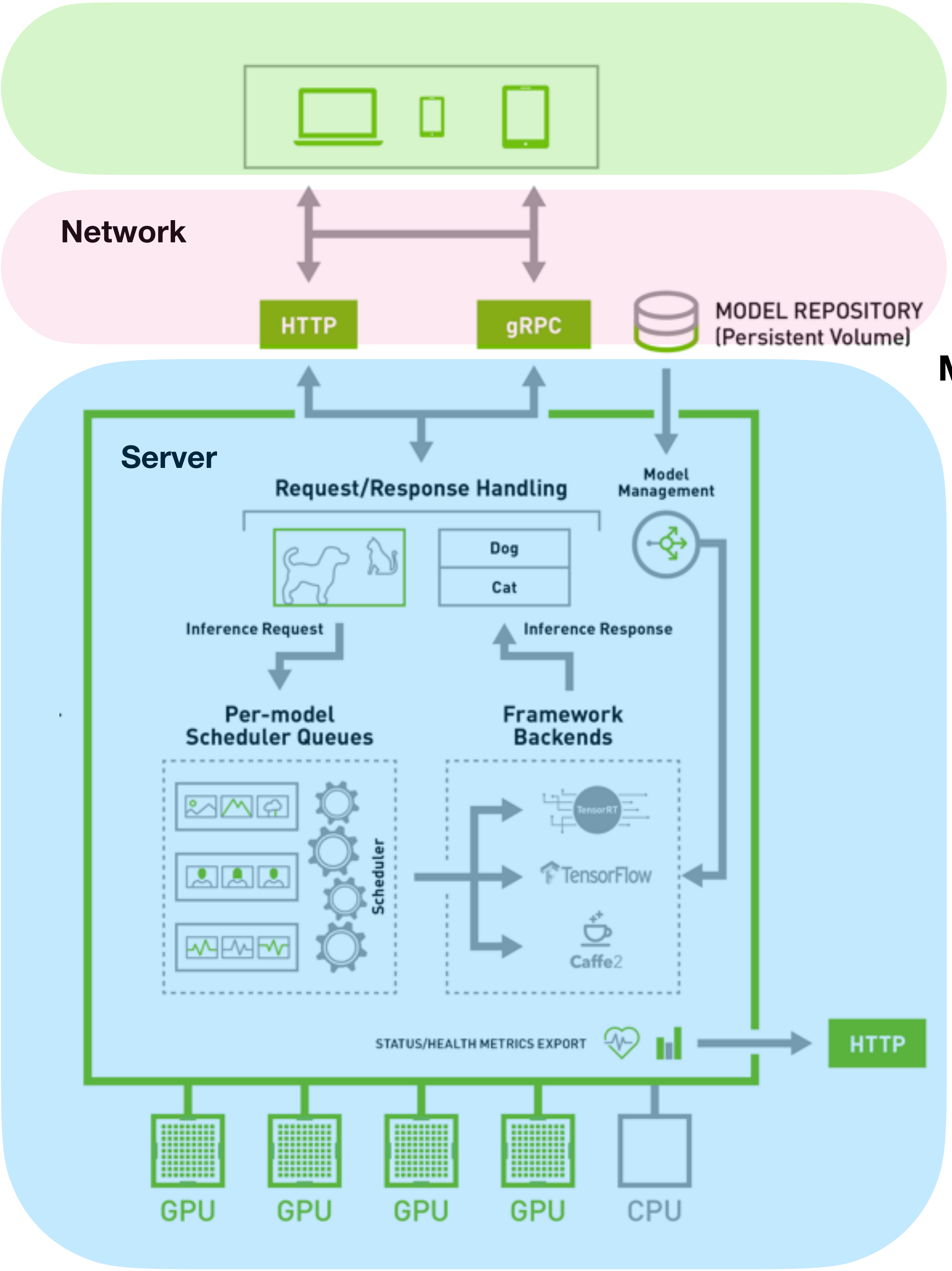
Triton Inference Server

Client sends request over network

Server receives request

Server queues and schedules request

The number of connected GPUs/FPGAs is scaleable; each has an instance of each model



Models are stored in local repository

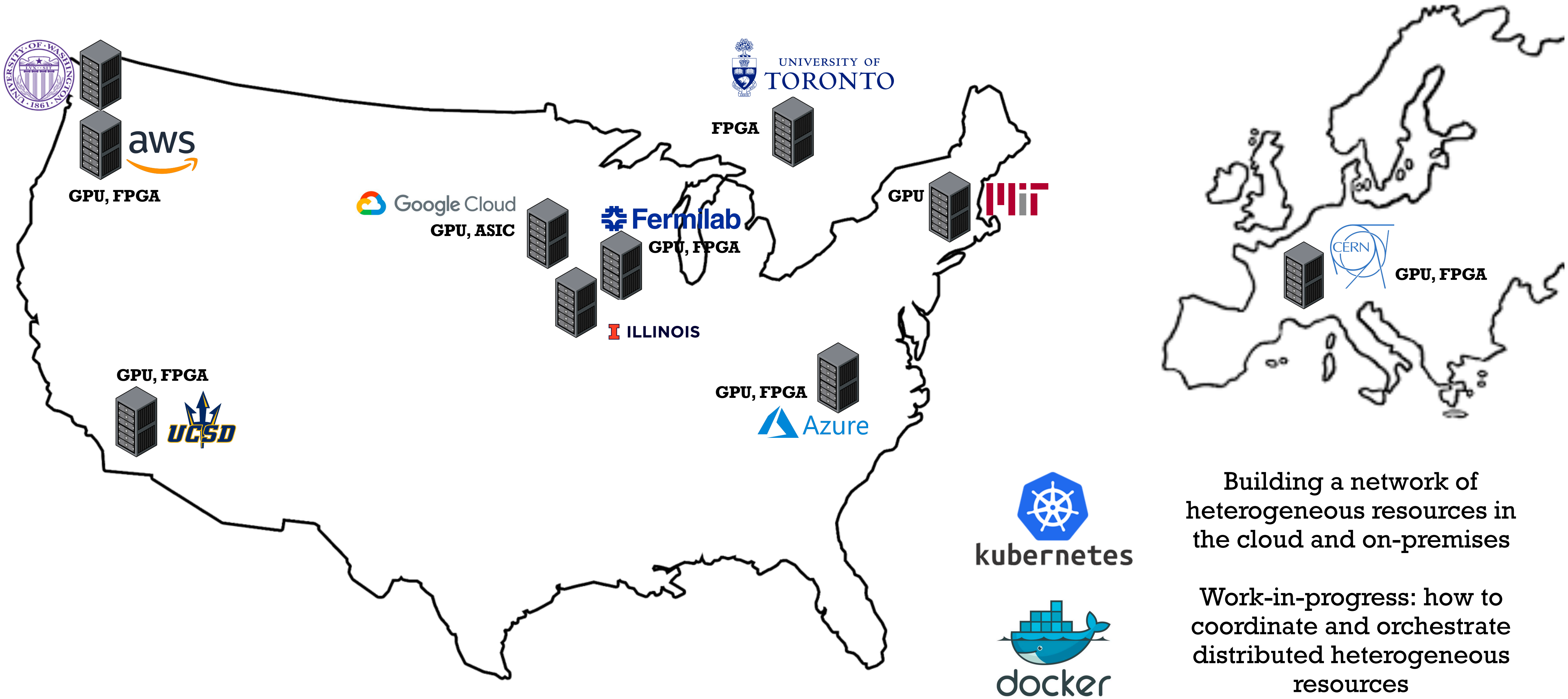
Many model formats (TensorFlow, Pytorch, TensorRT, ...)

Output monitoring information

Tools

Our tools for prototyping CMS reconstruction as-a-service

1. Google Cloud/Amazon Web Services/Microsoft Azure
2. T2/T3 clusters
3. local server/accelerator hardware

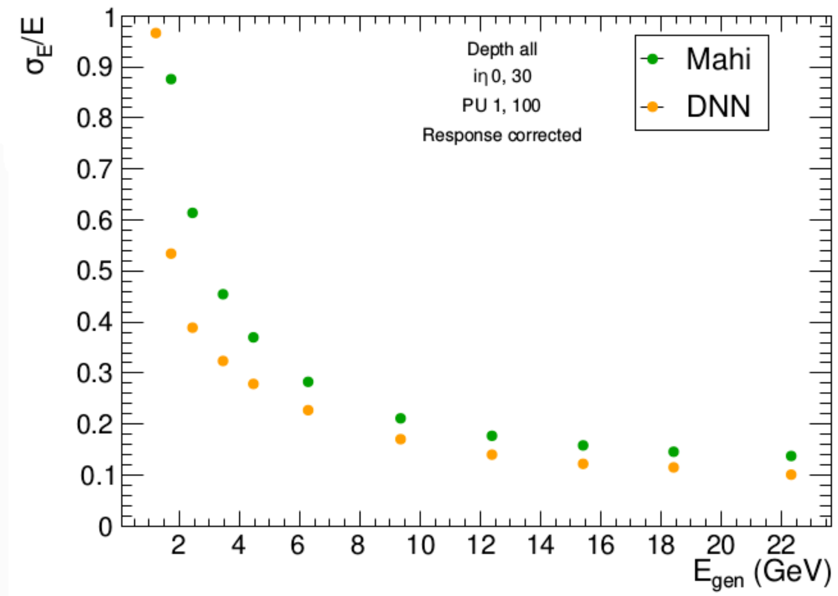


Building a network of heterogeneous resources in the cloud and on-premises

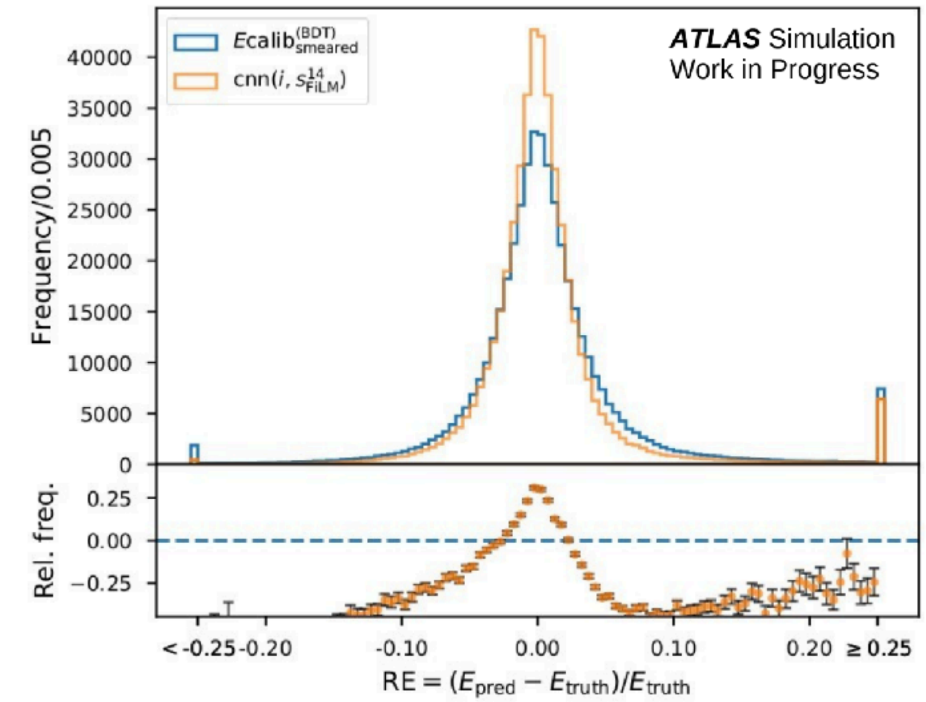
Work-in-progress: how to coordinate and orchestrate distributed heterogeneous resources

We have a wide network of resources, and perform at-scale tests with many different client-servers configurations, with servers both remote and on-site

Benchmark algorithms

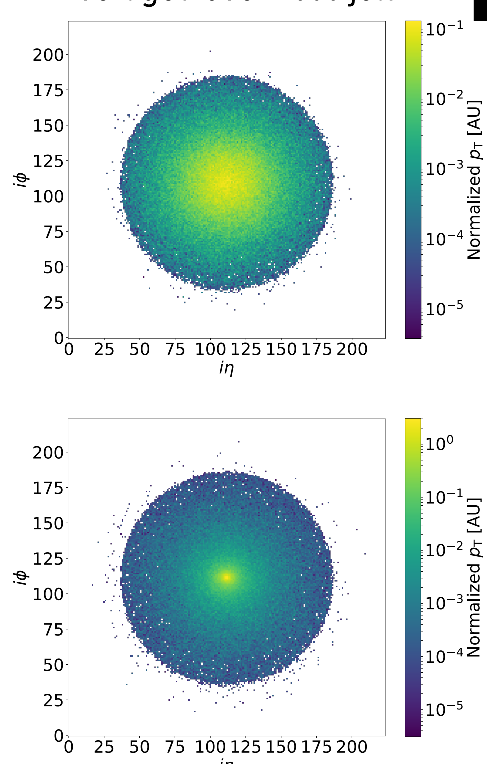


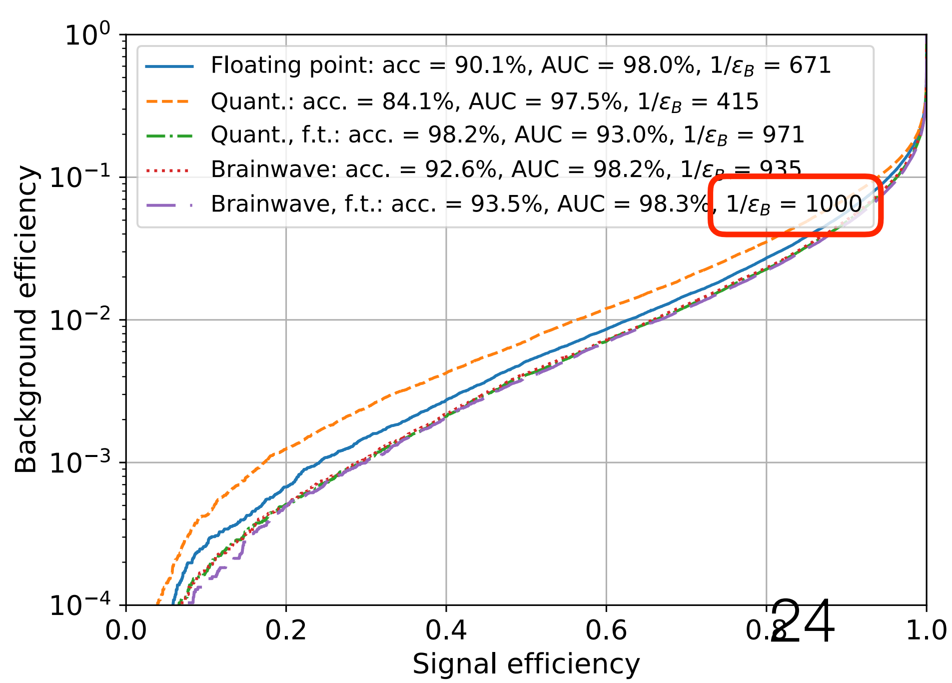
Calorimeter energy regression



ResNet top quark image classification

Averaged over 1000 jets

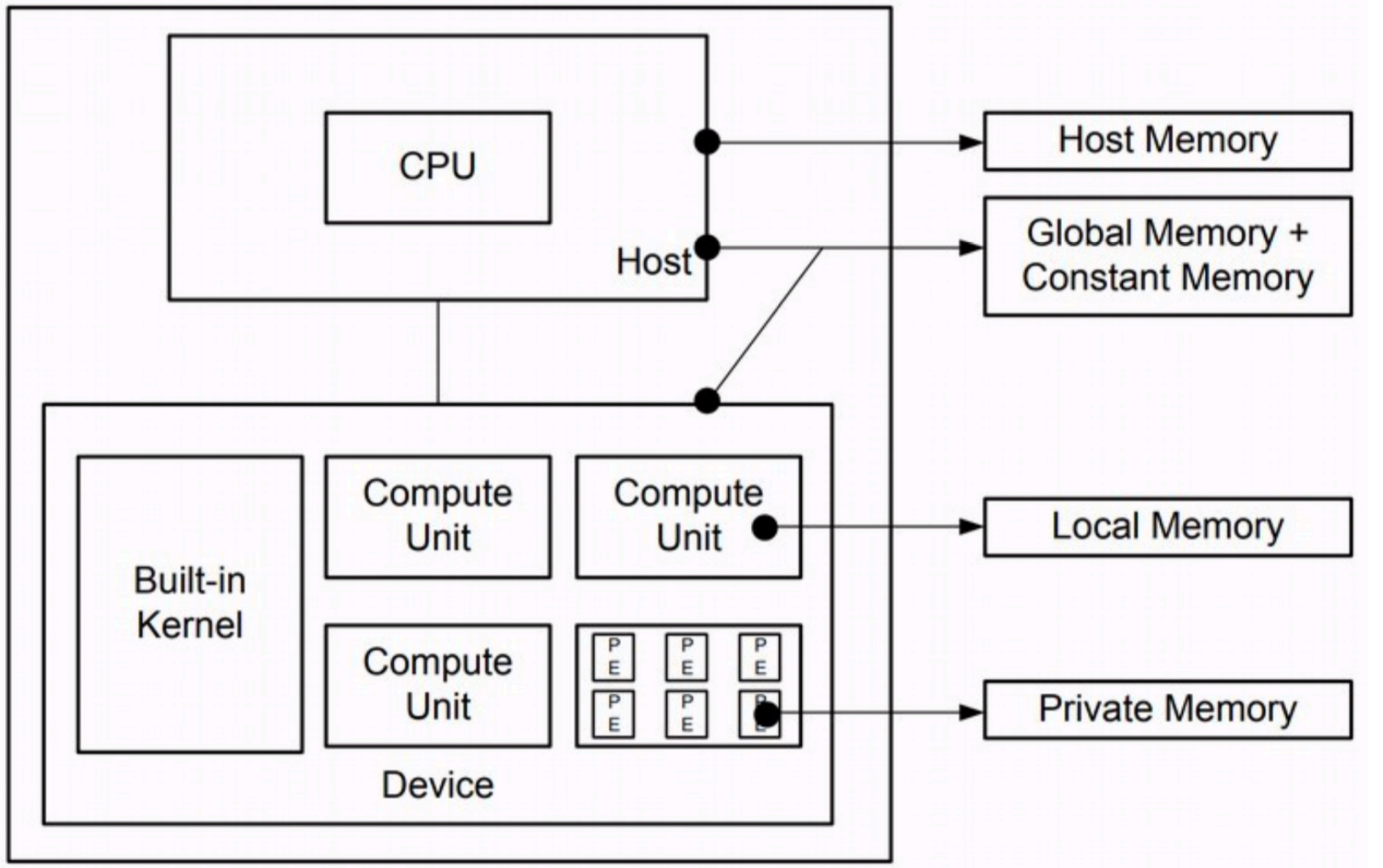




- **FACILE** *(batch 16000)* — **2k parameters**
- **DeepCalo** *(batch 10)* — **2M parameters**
- **ResNet** *(batch 10)* — **10M parameters**

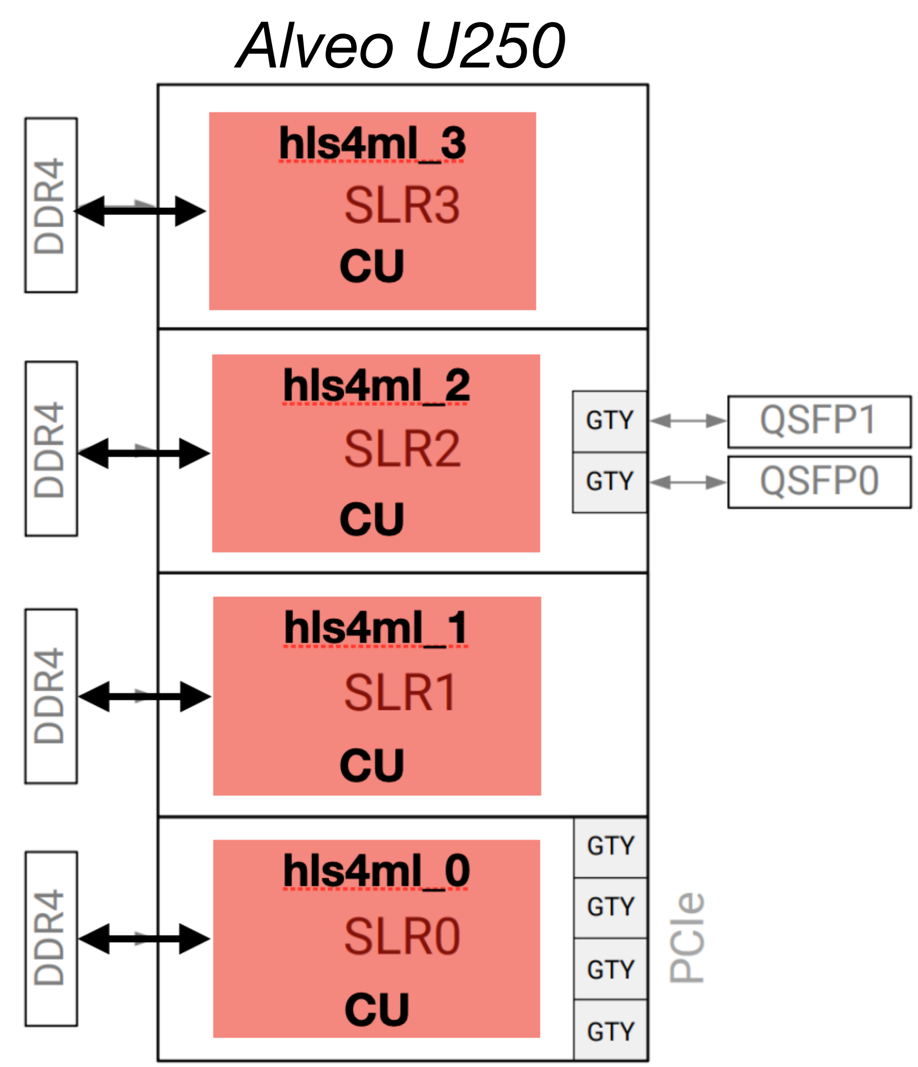
FACILE Server (VITIS + hls4ml)

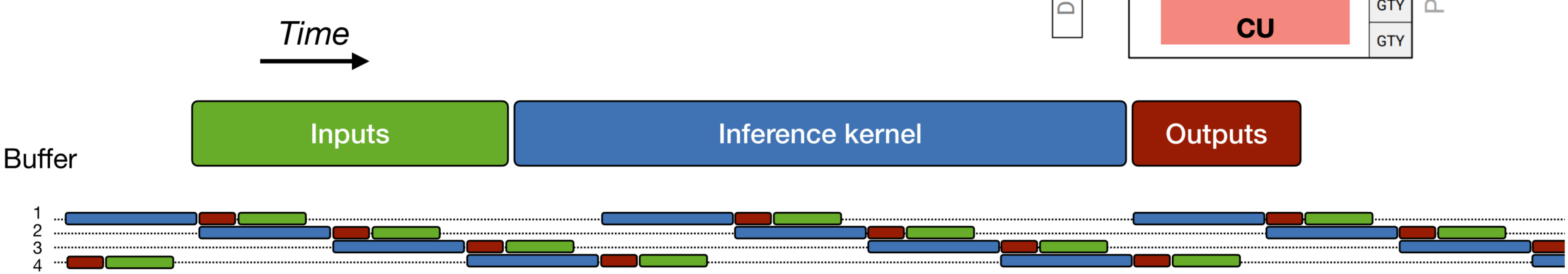
- Use Vitis Accel to manage data transfers, kernel execution
- Basic scheduling:
  - Copy batch 16000 inputs from host to FPGA DDR
  - Run hls4ml kernel
    - Tuned for low latency, pipelined, ~104 ns/inference
  - Copy 16000 batch outputs from FPGA DDR to host
- Server responsible for transferring input to dedicated buffers in host memory
- Set up for Alveo U250, AWS f1



FACILE Server (XILINX VITIS + hls4ml)

- Large amount of server optimization
- Can create multiple copies of hls4ml inference kernel on separate SLRs
- Can create buffer in DDR for multiple inputs, cycle through buffers

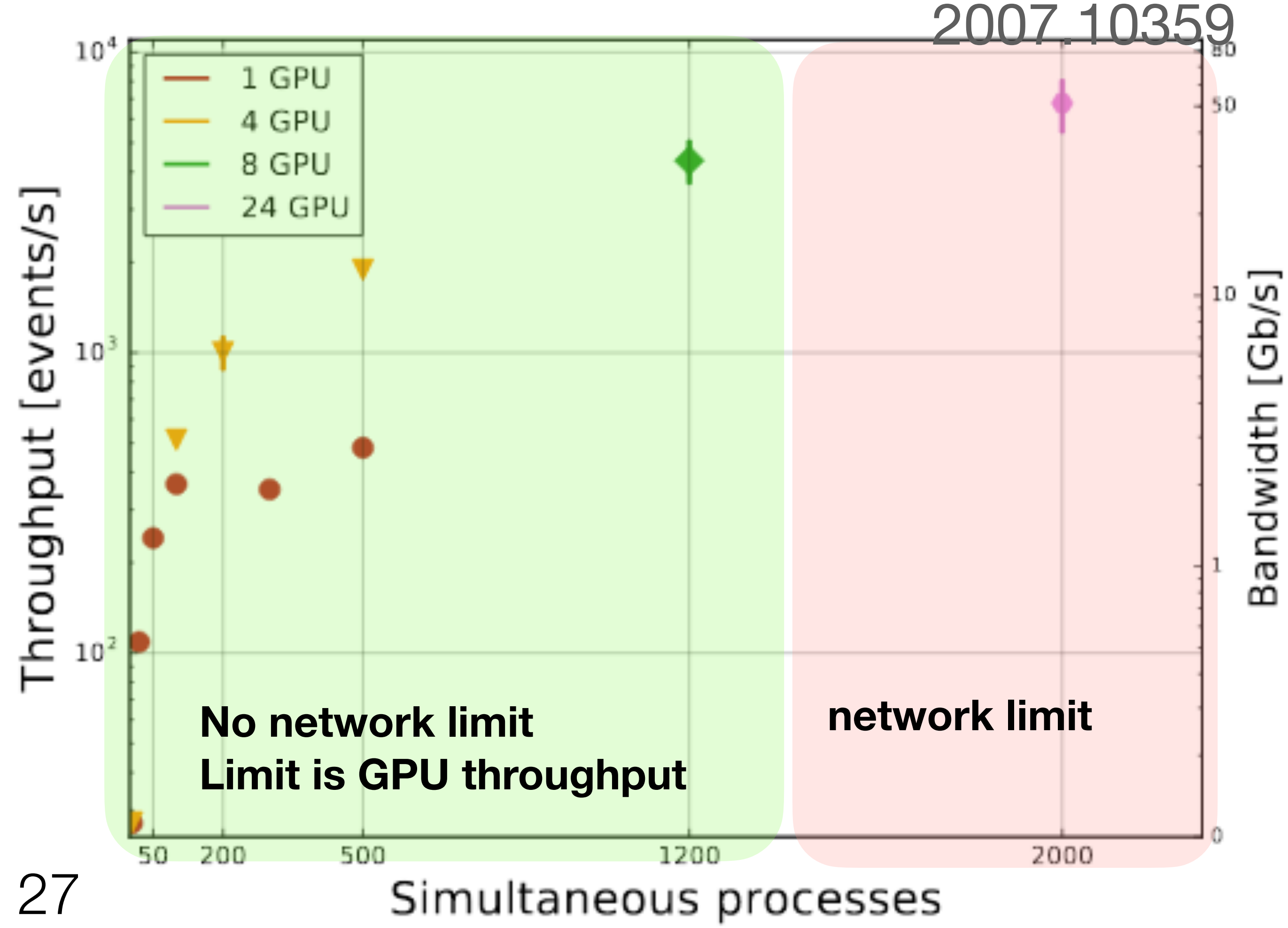




High bandwidth test

- What is the feasibility of remote server operation?
- High bandwidth, long distance test (MIT to Google Cloud in Iowa)
- Throughput scales linearly with number of GPUs
- Tests are stable up to 70 Gb/s (no special links)
    - Far exceeding any realistic use case (offline reco is 10 Gb/s)
    - Custom Kubernetes server to scale up to 24 GPUs

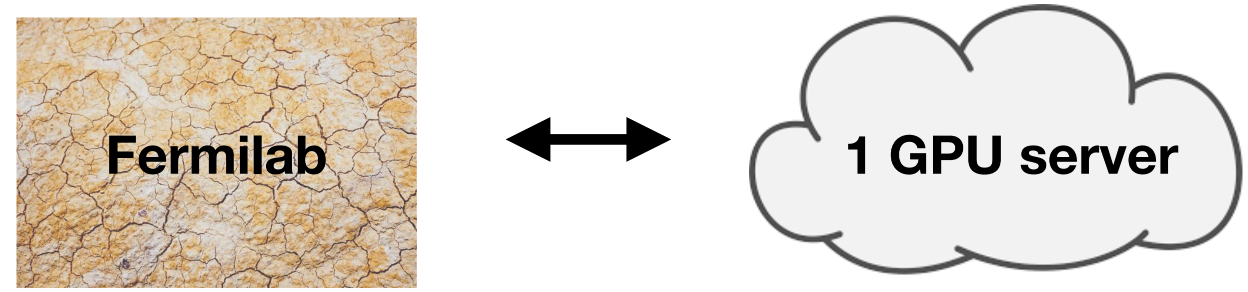
2007.10359



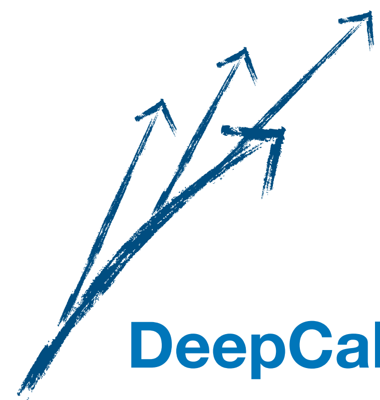
Throughput Tests (GPU)

Google Cloud Platform

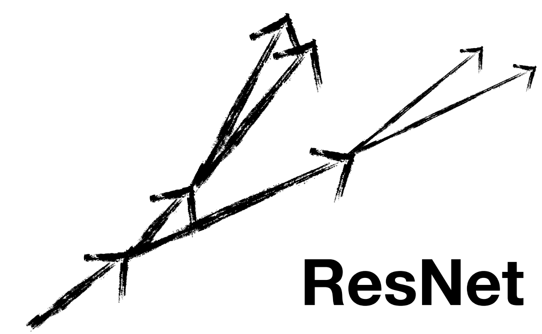
- Inference performed in CMS workflow
- Larger models saturate with fewer clients, lower throughput
- Range of performance for GPUs



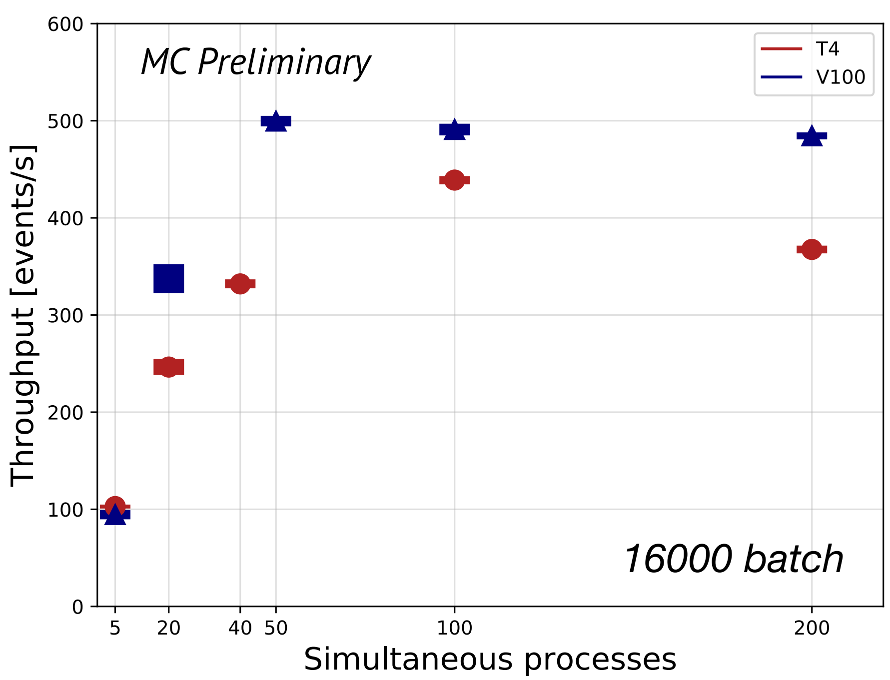
FACILE

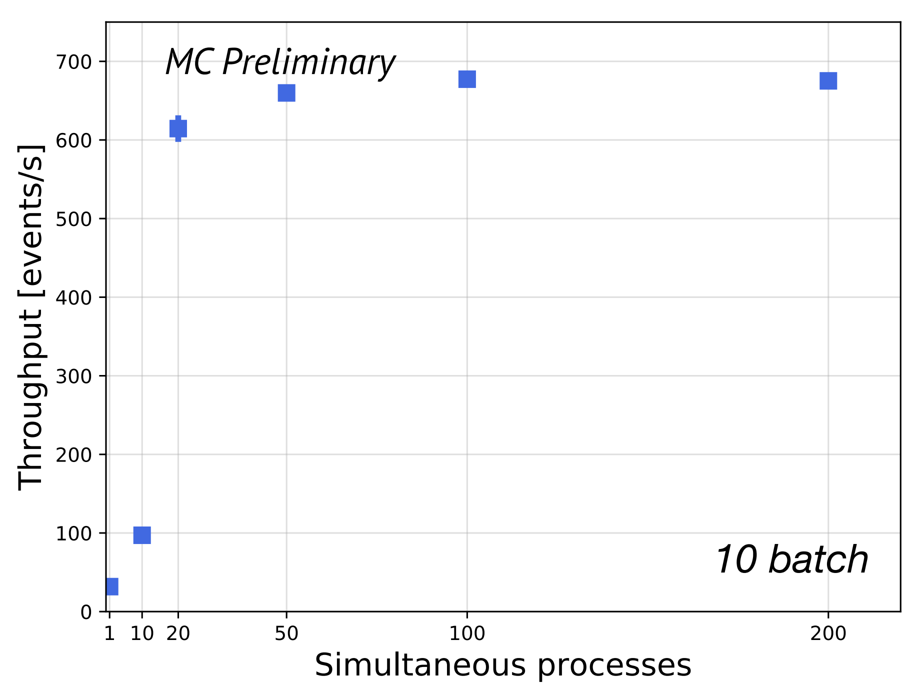


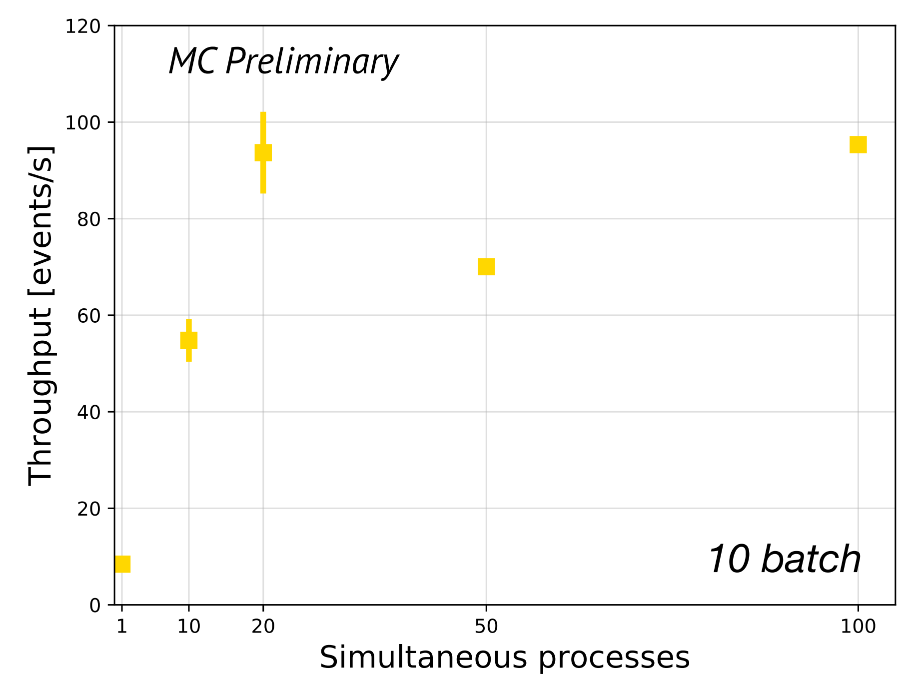
DeepCalo



ResNet

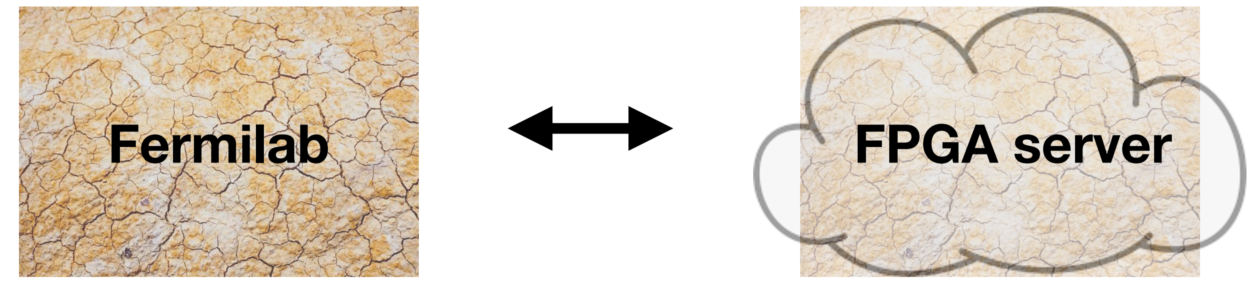






Throughput Tests (FPGA)

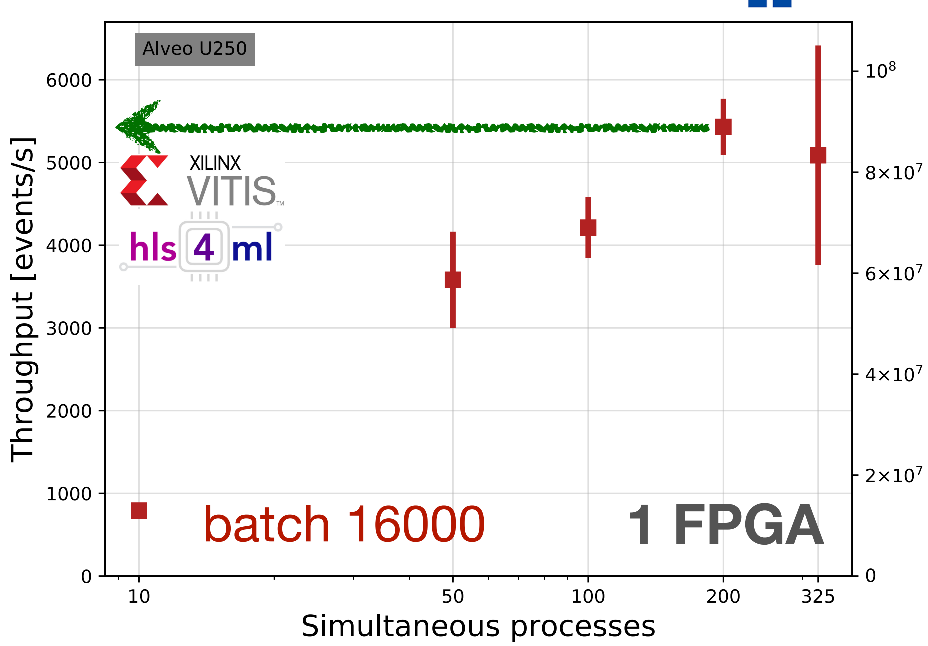
- With small **FACILE** network, server able to process over 5000 events/s



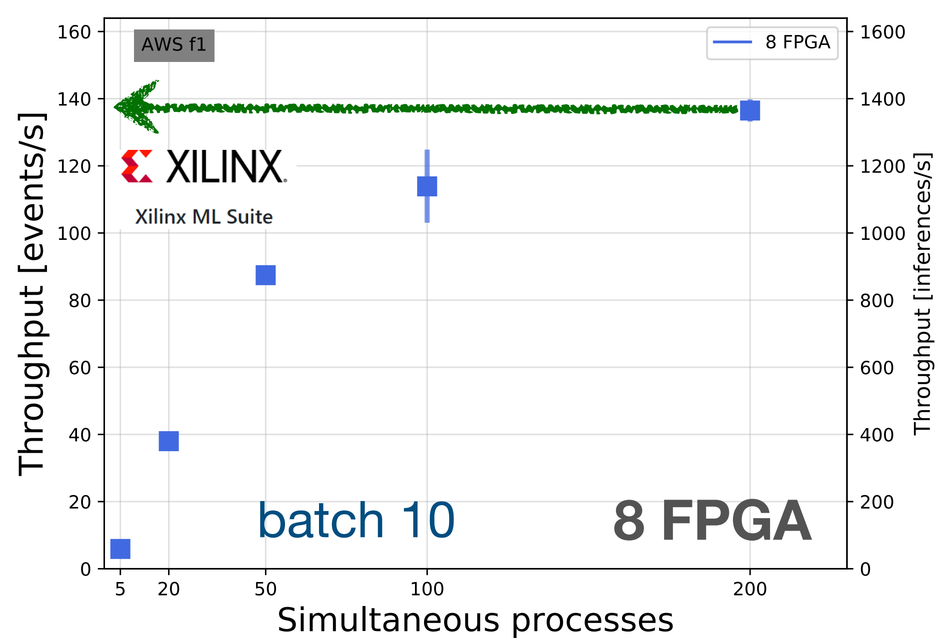
  - Limitation from CPU

- **ResNet** performance depends on hardware/specs

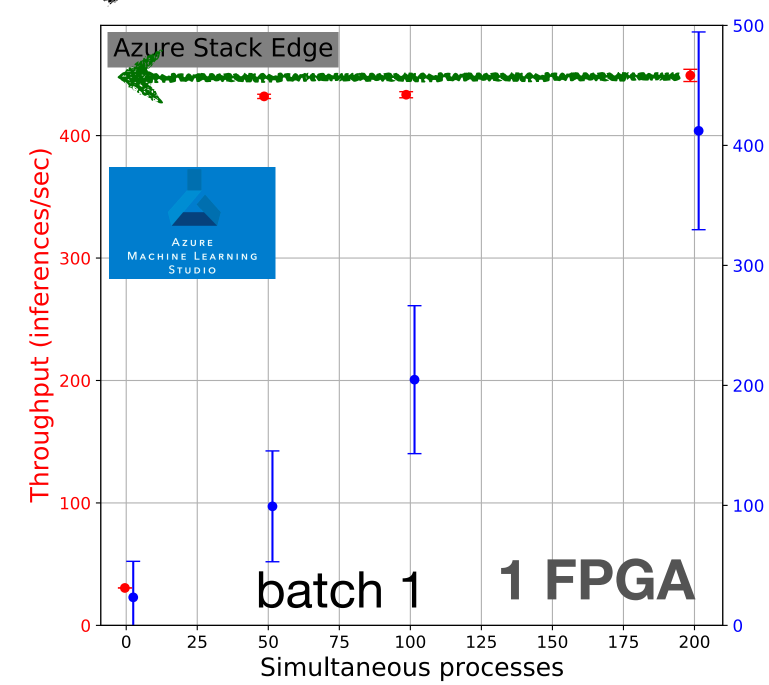
FACILE



ResNet aws

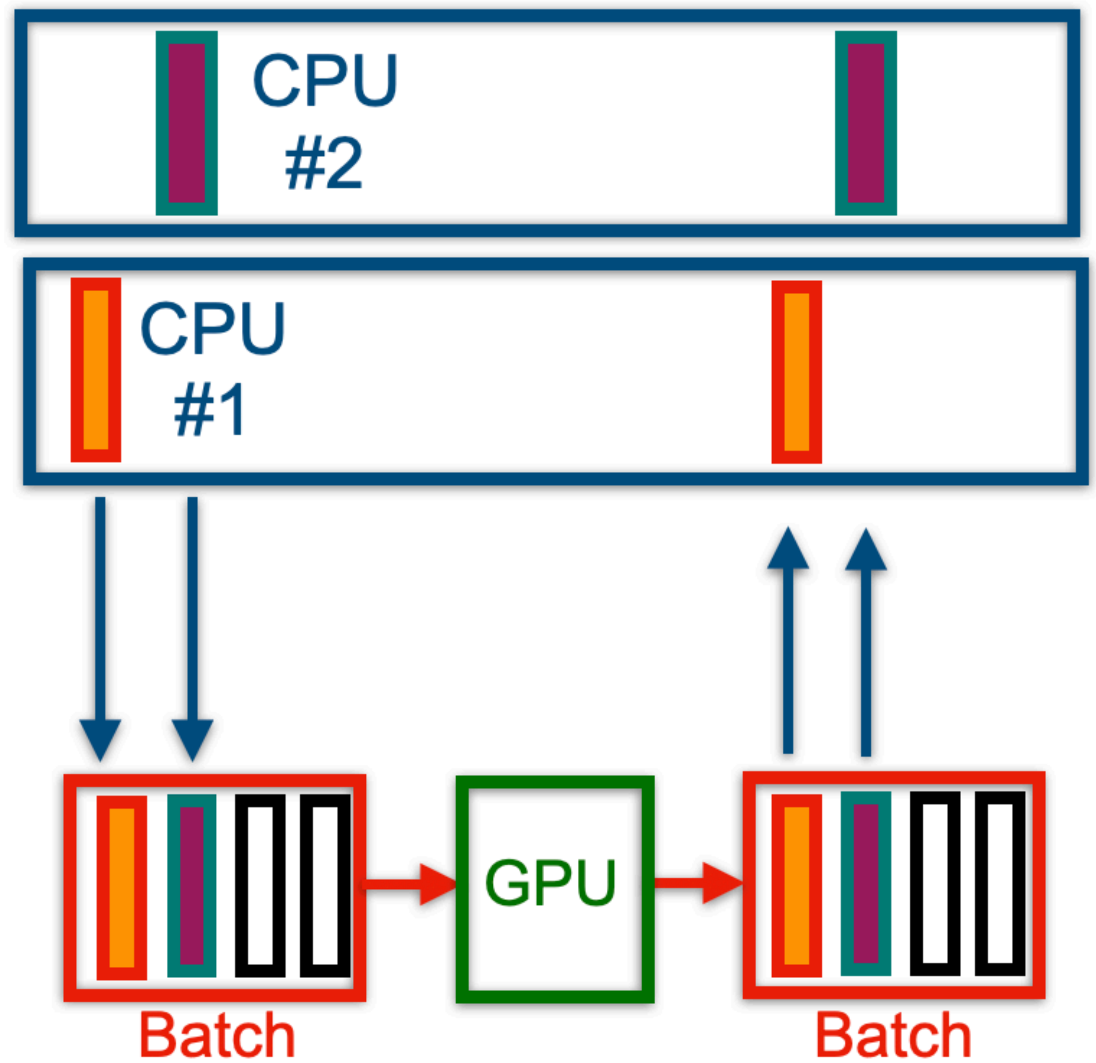


ResNet



Dynamic Batching

- Allows server to wait for requests to build up
- Most beneficial for small-batch algorithms
- Can extend event-by-event processing to multi-event processing
  - Transparent to user
- Single-line change to server configuration

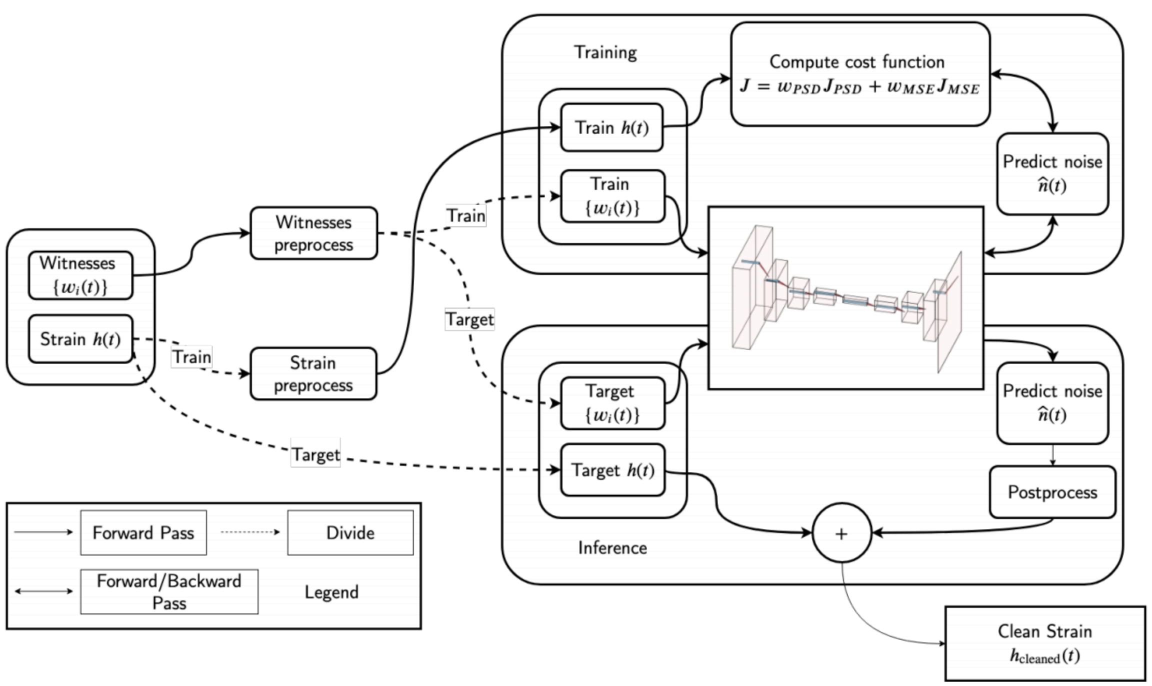


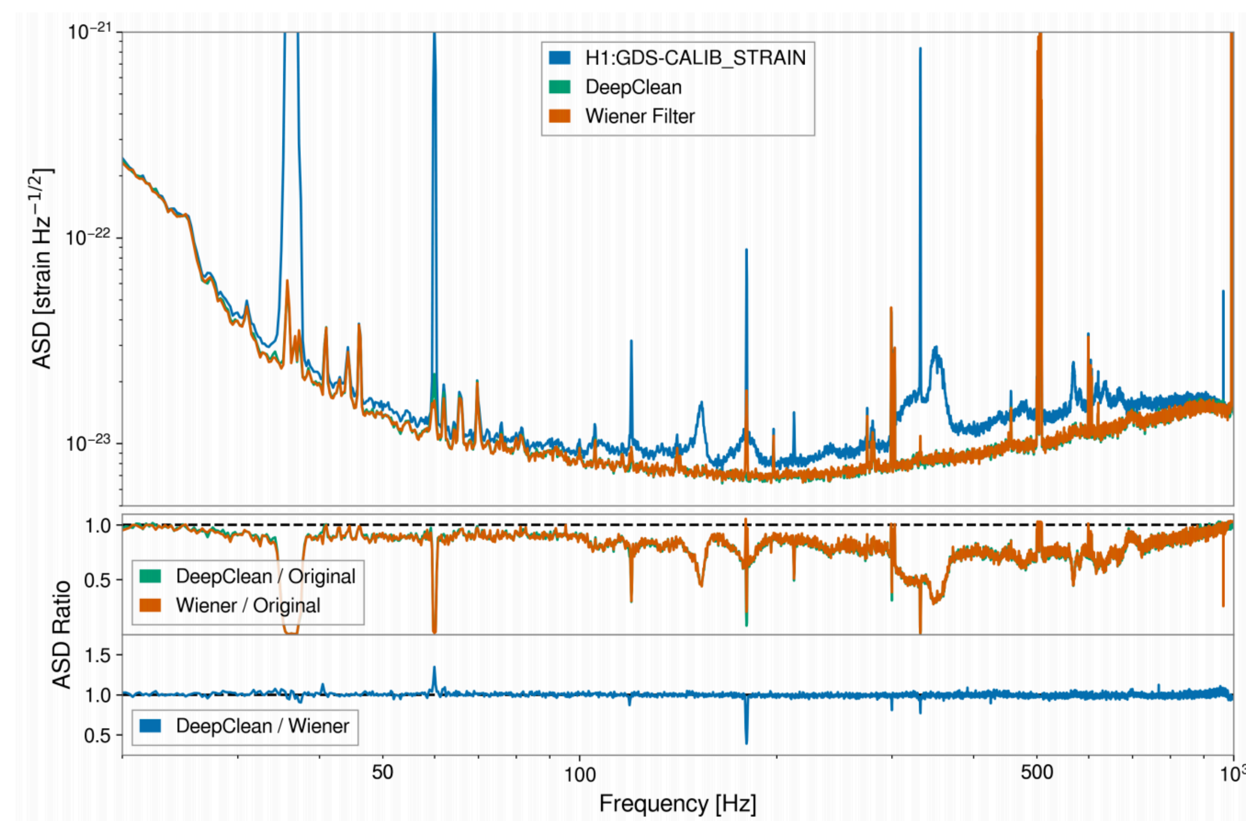

```
dynamic_batching {
  preferred_batch_size: [ 100 ]
}
```

Can also specify max wait time

arxiv/2005.06534

DeepClean





- DeepClean performs at the same level as Wiener Filter