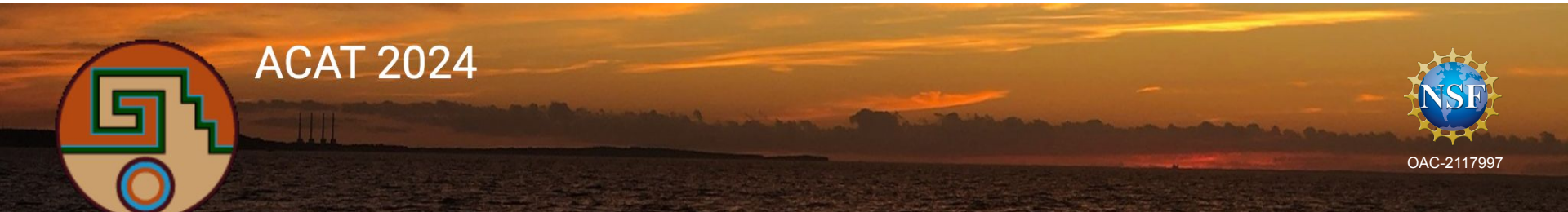
UNIVERSITY of WASHINGTON

a3d3 Accelerated AI Algorithms for Data-Driven Discovery

ACTS as a Service

Yuan-Tang Chou[1], Haoran Zhao[1], Xiangyang Ju[2], Yao Yao[3], Yongbin Feng[4], Elham E Khoda[1], Andrew Naylor[2], Paolo Calafiura[2], Shih-Chieh Hsu[1], William Patrick McCormack[5], Philip Coleman Harris[5] , Dylan Sheldon Rankin[6], Kevin Pedro[4], Miaoyuan Liu [3]

[1] University of Washington, [2] LBNL,[3] Purdue university, [4] Fermilab, [5] MIT, [6] University of Pennsylvania



NSF
OAC-2117997

Track reconstruction in HL-LHC

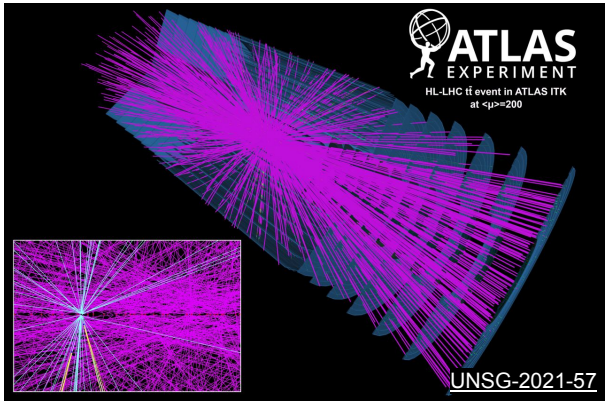
Track reconstruction is expected to be very challenging in high-density environments (HL-LHC).

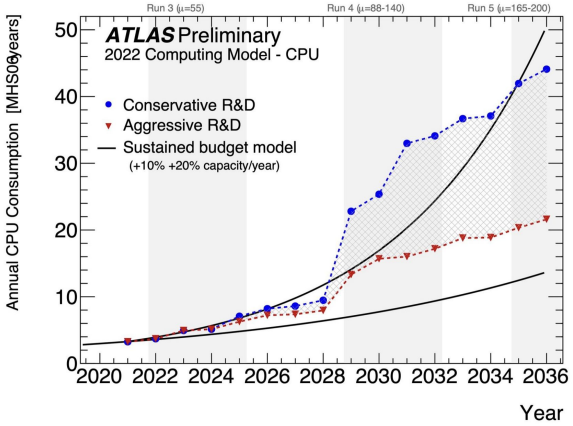
Many great ideas to improve tracking with ML

- LLM tracking by Xiangyang
- ACORN GNN pipeline by Daniel
- ReGAN by Jay
- Learned Clustering by Kilian

All these ML applications can be accelerated by using coprocessors

How can we incorporate coprocessors into our computing model under the limited budget?



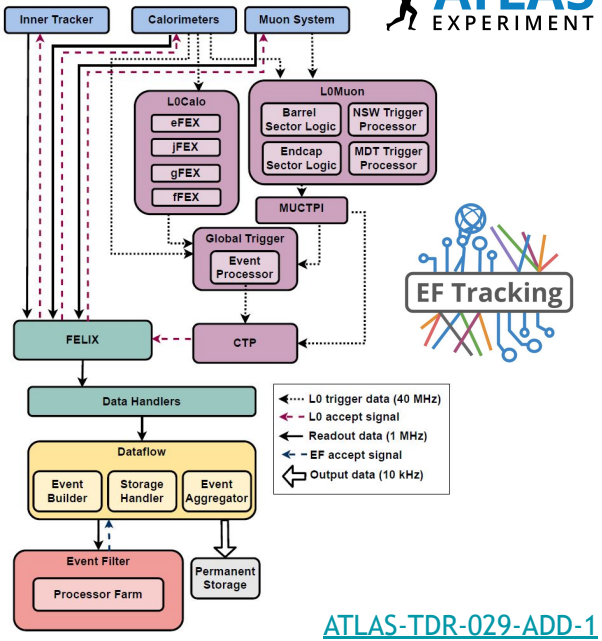


ATLAS Software and Computing HL-LHC Roadmap

Tracking in ATLAS Event Filter

ATLAS is planning to upgrade the current High-level trigger farm to Event Filter with **commercial solution** at HL-LHC

- Heterogeneous devices (e.g., GPUs or FPGAs) with CPU to provide power saving and throughput increase
- **The throughput and scalability are the key**.
  - Expected to have ~300 k space points with ITK detector
  - Region-of-interest tracking at 1MHz
  - Full-scan tracking at 150 kHz



ATLAS-TDR-029-ADD-1

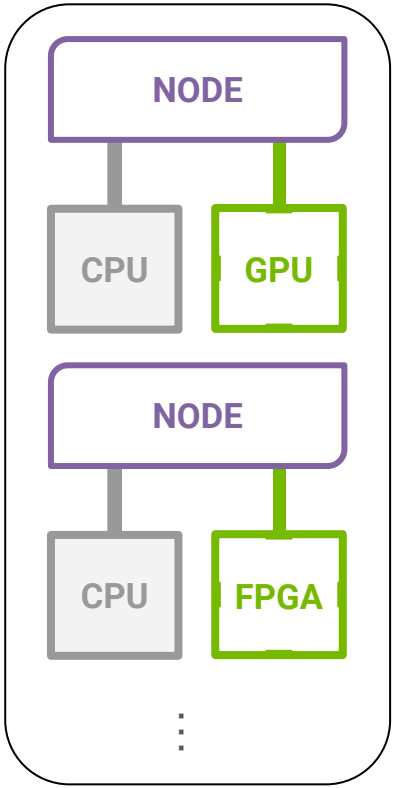
How can we enhance the scalability of the tracking workflow and make the integration much easier?

As a Service Computing Model

Heterogeneous Computing

- The most straightforward way to deploy algorithms on coprocessors is to run on machines with coprocessors
- However, **direct connection** can be inefficient and expensive at scale
- Need to matched the CPU-coprocessor ratio perfectly to fully utilized all the resource

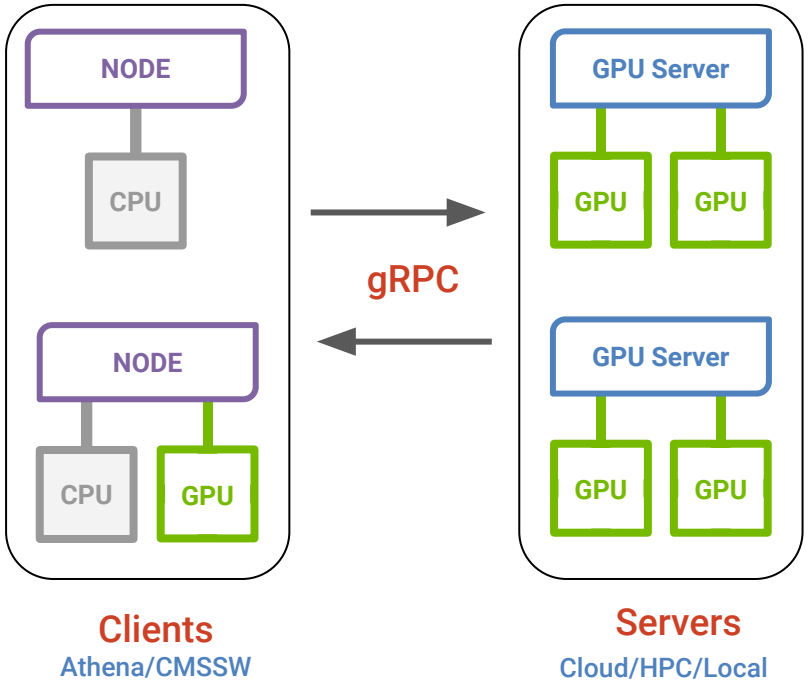
“Direct connection”



As a Service Computing Model

Alternate coprocessor deployment scheme where coprocessor-enabled machines host an inference server and remote jobs send inference requests via network connection

The servers do not necessarily need to be remote; they can be just next to the client machine.



“Direct Connection”

Pros:

- Already have working example [1]

Cons:

- Can be an inefficient use of resources
- Expensive
- Machines without GPUs/FPGAs can’t benefit from coprocessors

“As a Service Computing”

Pros:

- **Factorized** out the underlying backend implementation
- More straightforward to integrate with the **production framework** (e.g. Athena)
- Independent of the underlying **technology choices** and algorithms
- Better scalability and resource utilization (**Reduce cost**)

Cons:

- Adds complexity

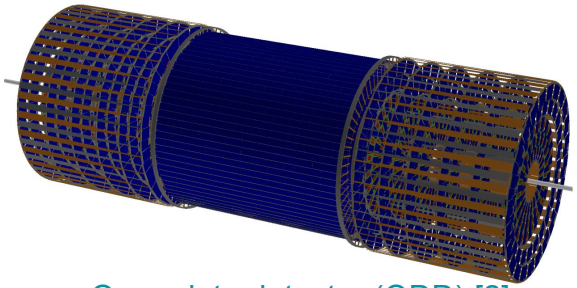
[1] 2004.04334

ACTS as a Service

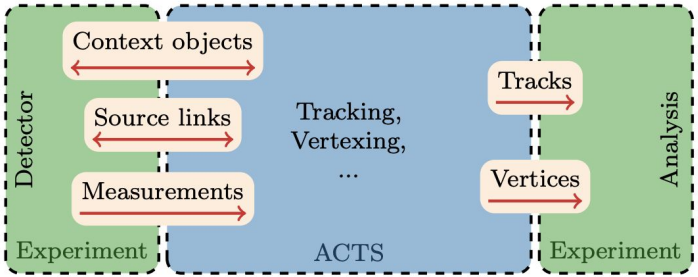
A Common Tracking Software

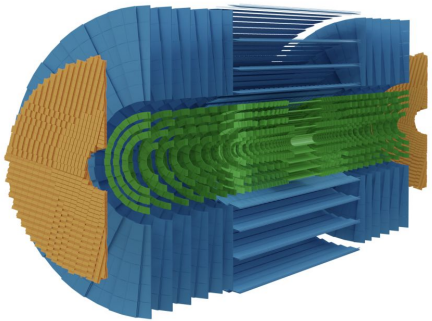
ACTS is an experiment-independent toolkit for charged particle track reconstruction in high energy physics experiments for both production and R&D [1]

Contains a full track reconstruction chain, which will be used in ATLAS for Run 4 offline tracking



Open data detector (ODD) [2]



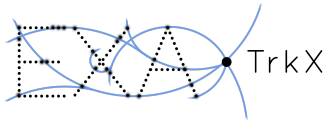


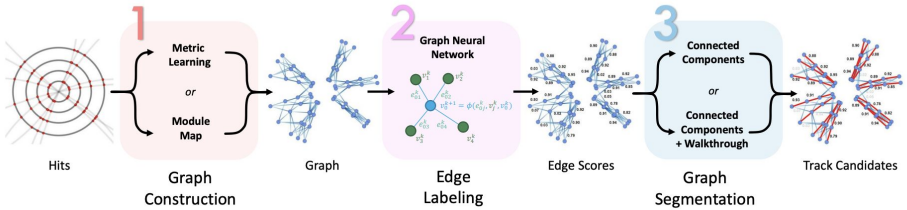
ATLAS ITK

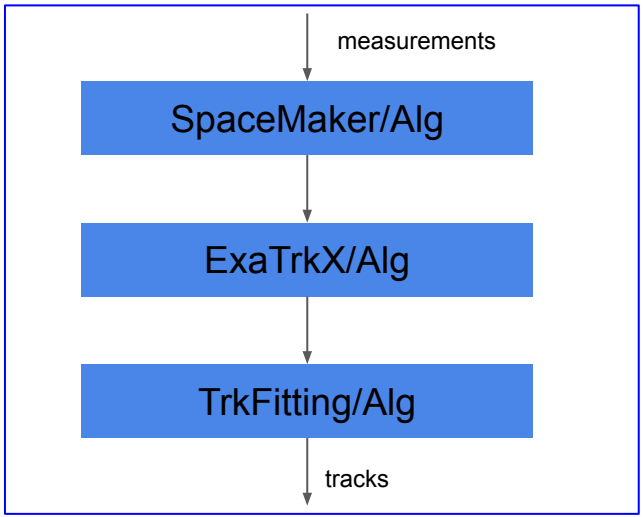
[1] A Common Tracking Software Project, arXiv: 2106.13593
[2] Paul Gessinger-Befurt et al 2023 J. Phys.: Conf. Ser. 2438 012110

ACTS with GNN tracking (ExaTraX)

Direct inference





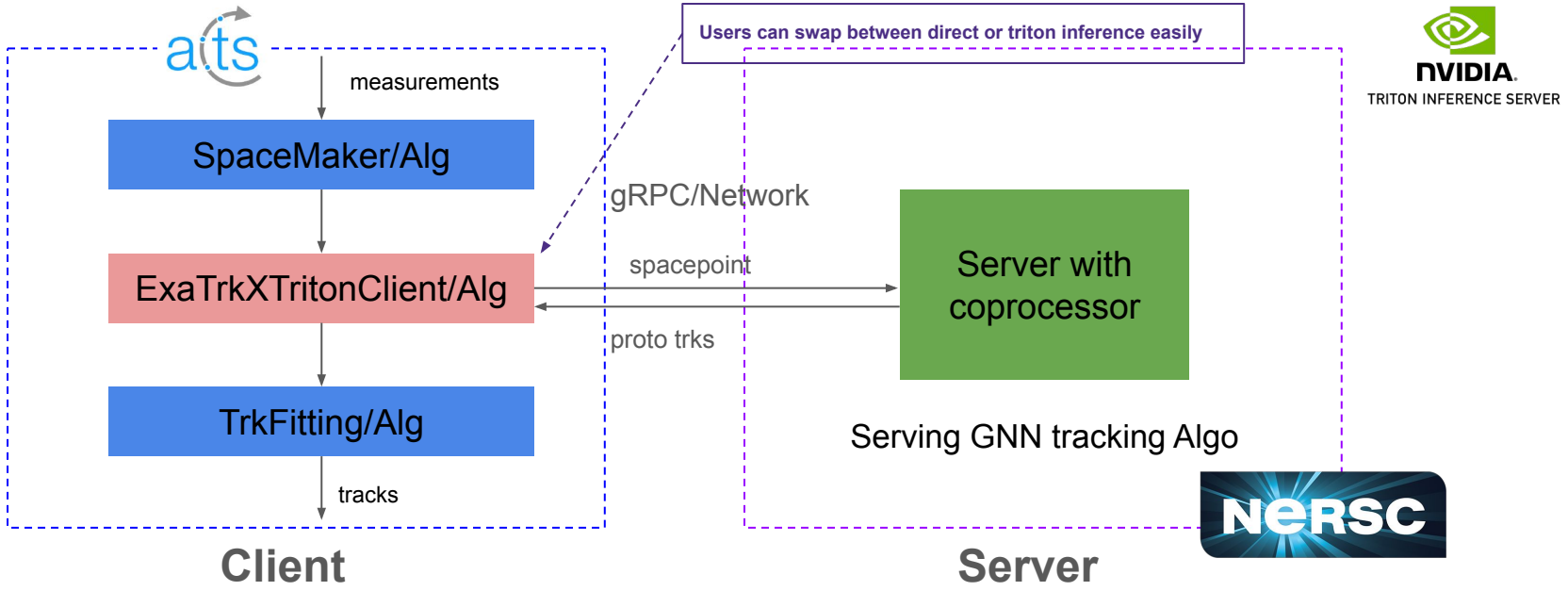


- GNN Track Finding (ExaTrkX) [2] can run locally with CPU/GPU using TouchScript
- Track fitting using Kalman Filter (KF) still runs only on CPU
  - Ongoing effort to port the GPU-based KF developed in traccc project [3]

[1] Acts - https://github.com/acts-project
[2] Performance of a Geometric Deep Learning Pipeline for HL-LHC Particle Tracking Eur. Phys. J. C 81, 876 (2021)
[3] acts-project/traccc: Demonstrator tracking chain on accelerators

Integration of the ExaTrkX-as-a-Service to ACTS

Triton client added in ACTS to communicate with Triton Inference Server

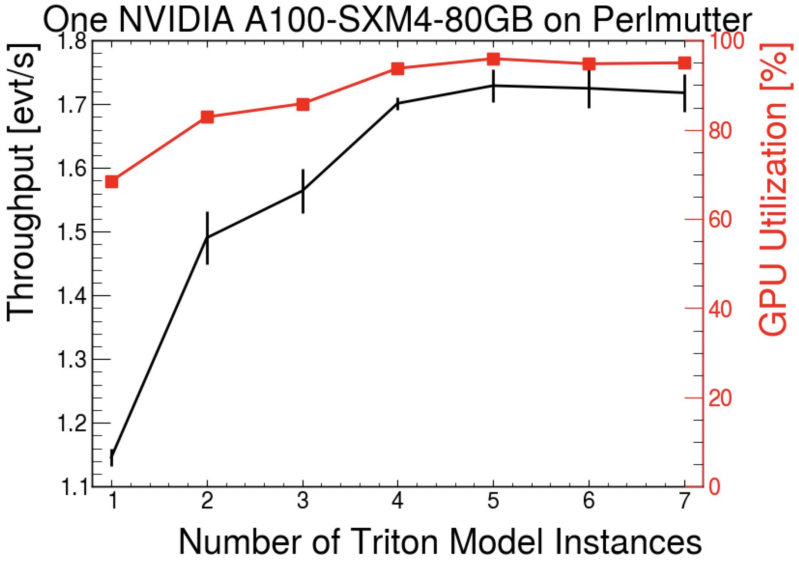


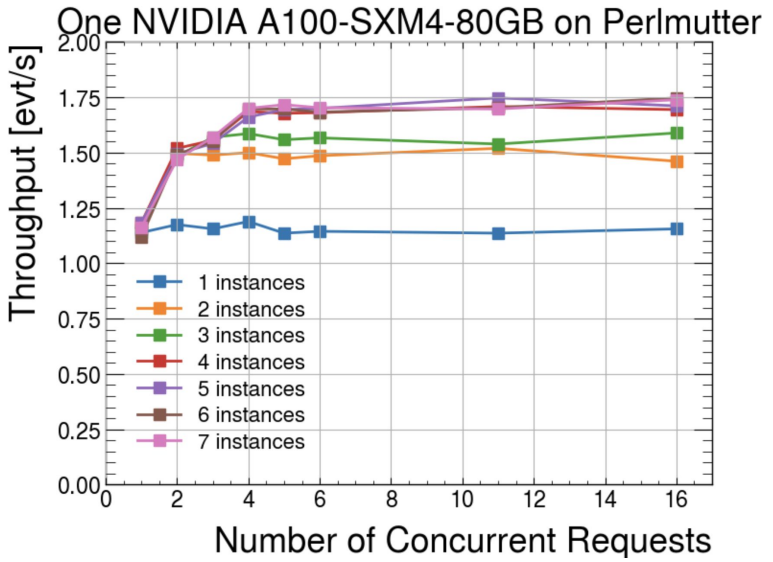
We can offload more algorithm to coprocessor to increase the throughput

Multiple Model Instances Scaling

ExaTrkX custom backend

- Multiple inference instance run on one GPU
- Better utilization and higher throughput for one GPU

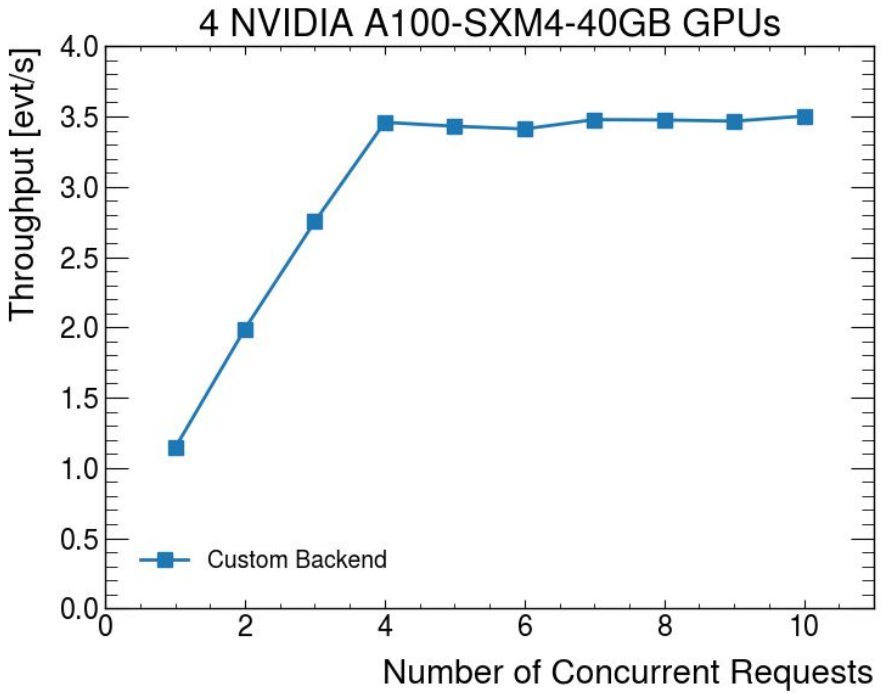




Multiple GPU Scaling

ExaTrkX custom backend

- Test with Triton inference server with 4 GPUs
  - Inference request distributed equally among the GPUs
  - Both client and server are at Perlmutter
- One model instance per GPU
- The throughput scale linearly and saturate with more than four concurrent requests



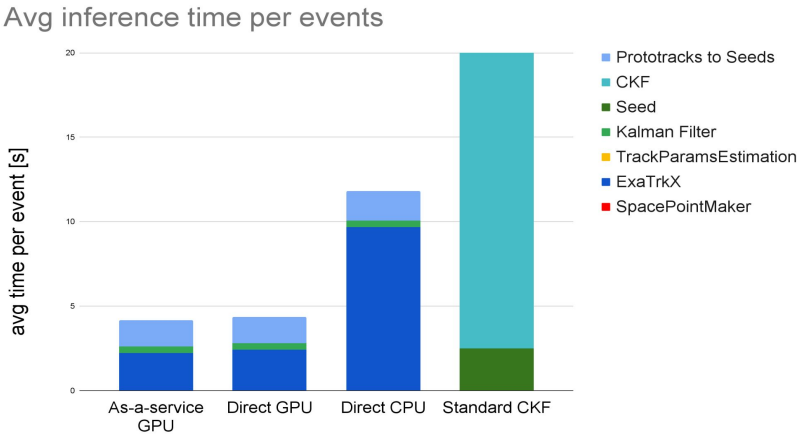
Performance of ACTS as a Service

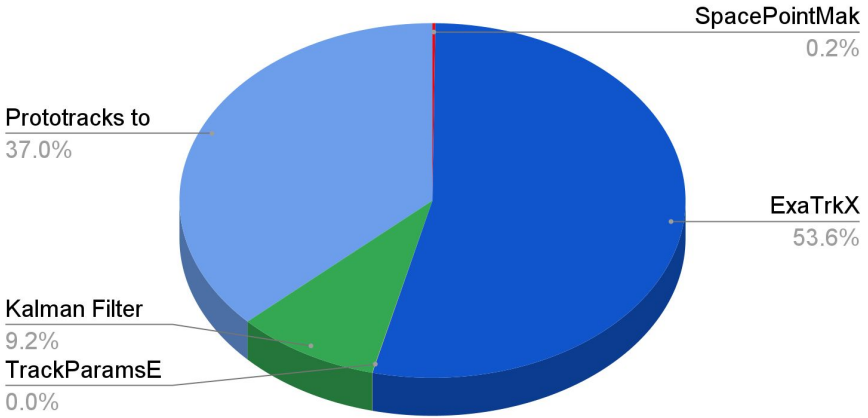
Inference time: (ttbar PU=200, ODD)

CPU: 2x AMD EPYC 7763 CPUs, 64 cores per CPU

GPU: 1x NVIDIA A100-SXM4-40GB (or 80GB)

No additional overhead was observed in the as-a-service inference



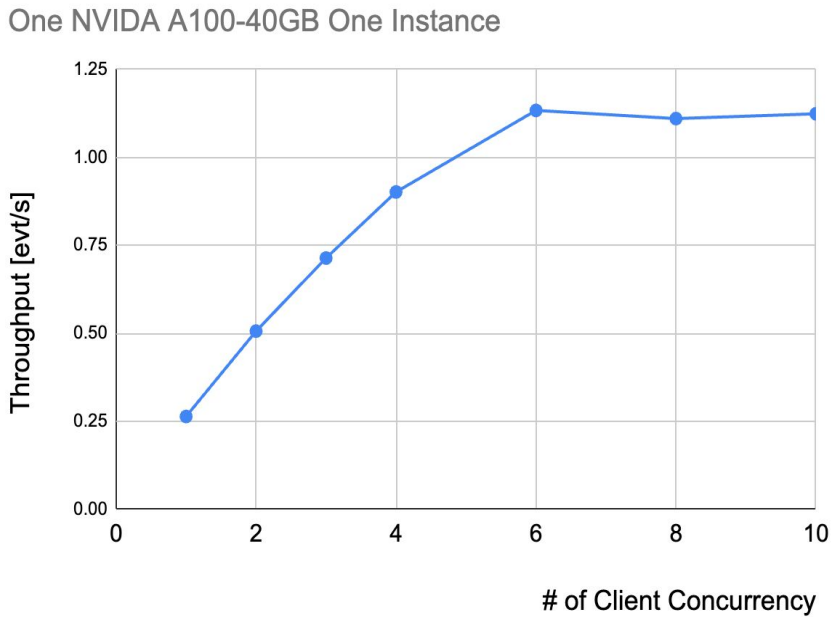


Only ExaTrkX are offloaded to GPUs
~ 53 % of computing time in direct GPU

ACTS as a Service

Measure the throughput considering full ACTS tracking chain

- One GPU with one model instance
- The GPU saturate with more than 6 concurrent request



Network Latency over Remote Inference Servers

- The network latency from using a remote service:
  - Client: Perlmutter at LBNL
  - Triton servers: NRP at UCSD
- The Pods are deployed using K8 cluster
- Network latency: **~ 1.1 s**

Direct distance = ~ 730 km

Yuan-Tang Chou

Kubernete approach

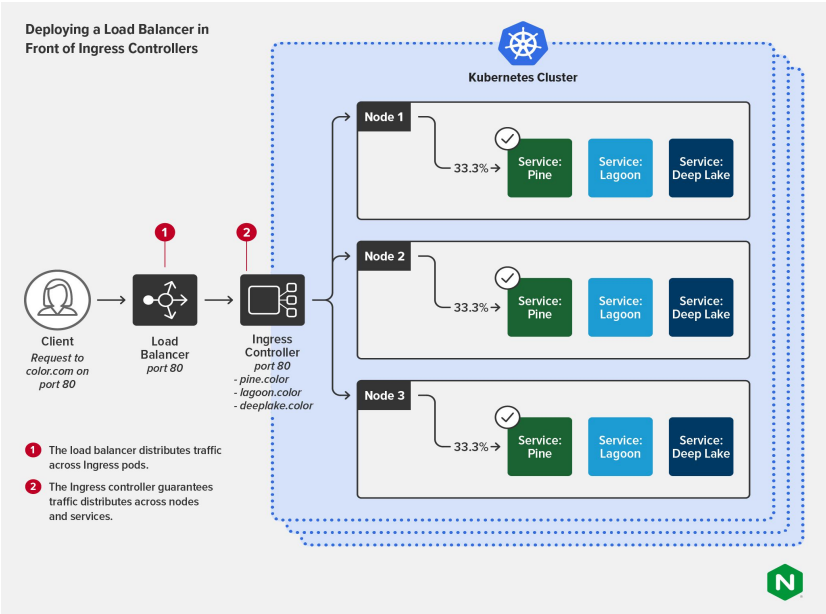


Image credit: https://www.nginx.com/blog/kubernetes-networking-101/

Summary

- We built the ExaTrkX-as-a-Service based on the NVIDIA Triton server and used ACTS as the first demonstrator.
- The result shows no overhead with aaS computing model for GNN tracking with better GPU utilization.

Outlook

An as-a-service computing model could increase tracking throughput regardless of the underlying algorithm/co-processor.

- Optimizable coprocessor-to-CPU ratios (cost-efficient)
- Increased throughput and better scalability
- Flexible technology choice and algorithm design
- Reduce complexity in integration

CMS has shown great advancement in using the as-a-service idea in ntuple production to offload ML algo to coprocessors server. See Yongbin's poster today

We are exploring the possibility of using this technology for offline and online tracking in ATLAS