Tracking as a service

Yuan-Tang Chou
on behalf of the ACTS-as-a-service team
University of Washington

SONIC: Heterogeneous Computing for Science Workshop

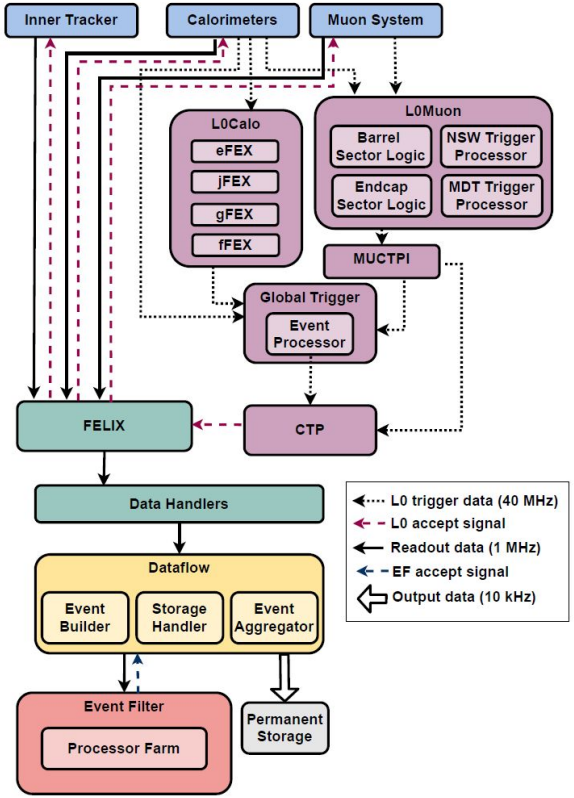
OAC-2117997

Introduction: Event Filter Tracking in ATLAS

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 and FPGAs) with CPU to provide power saving and throughput increase
- The throughput and scalability are the key.
  - Expected to have ~300 k spacepoints with ITK
  - Region-of-interest tracking at 1MHz
  - Full-scan tracking at 150 kHz
  - 2nd Demonstrator in Q3 2024

As-as-service computing model could help here!

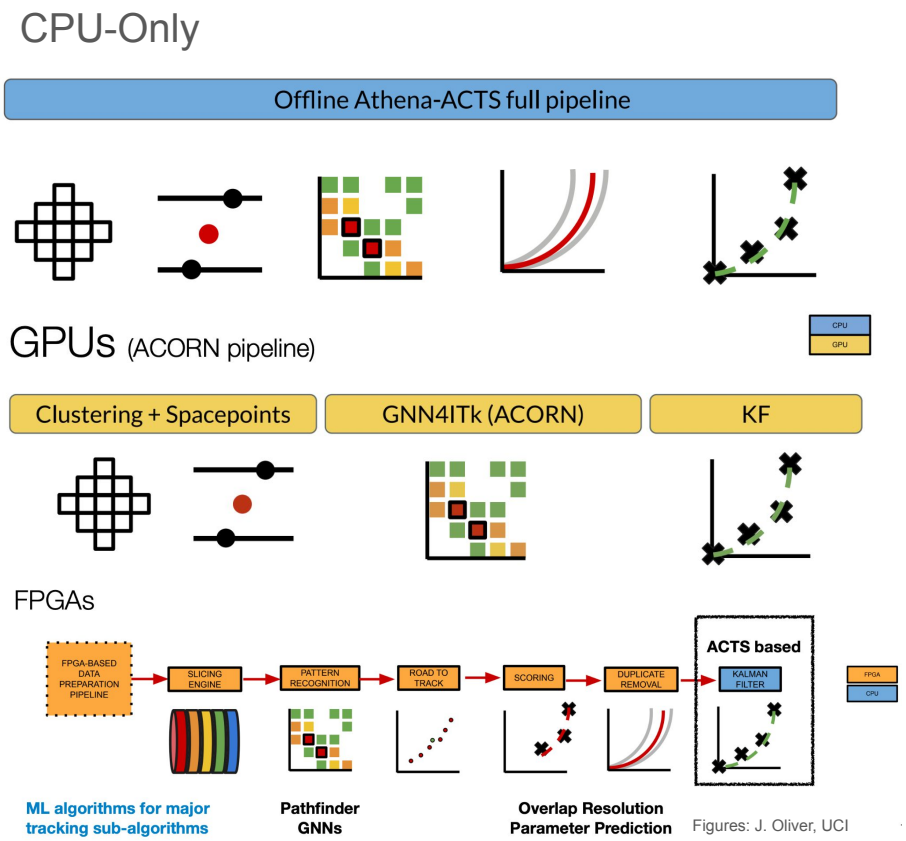


EF Tracking possible pipelines

Currently, there are three possible technology choices for EF Tracking pipelines

- CPU-Only
- GPUs
  - **Track 1:** ACRON
  - **Track 2:** ACORN + Traccc
- **Track 3** FPGA

There are three accelerator tracks which could apply the tracking-as-a-service



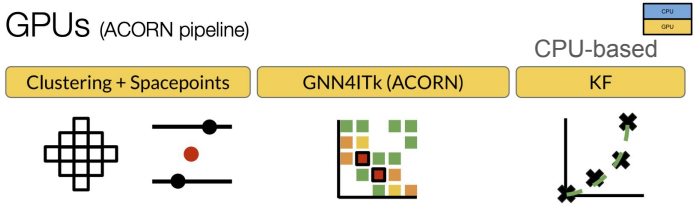
Figures: J. Oliver, UCI

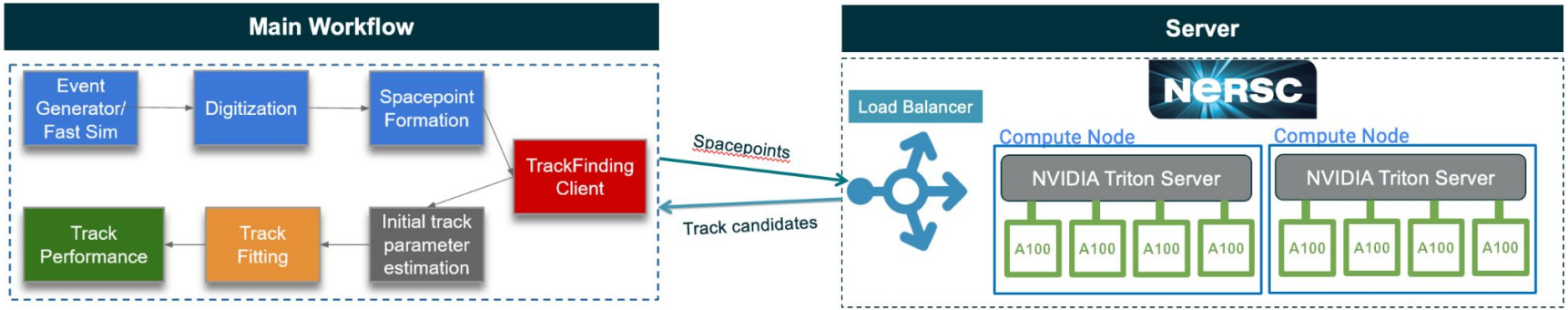
Track 1: ACRON

GPU-based GNN Tracking (ExaTrkX) as a Service

Existing solution from ExaTrkX as a Service

Can be quickly adapted to apply ACORN into Athena for EF Tracking



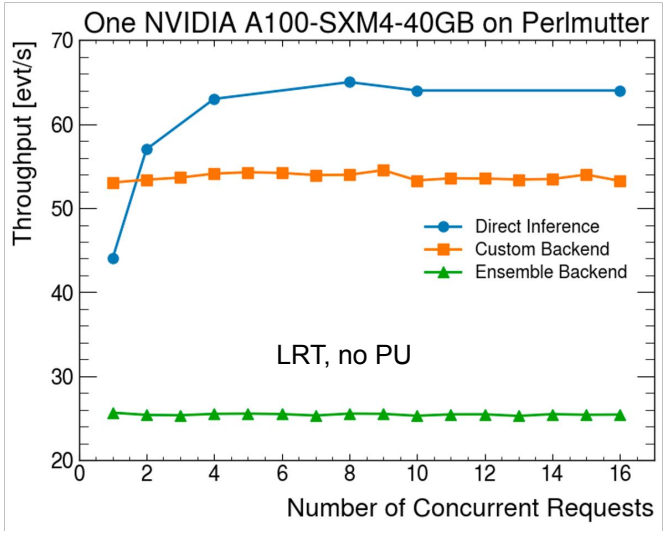


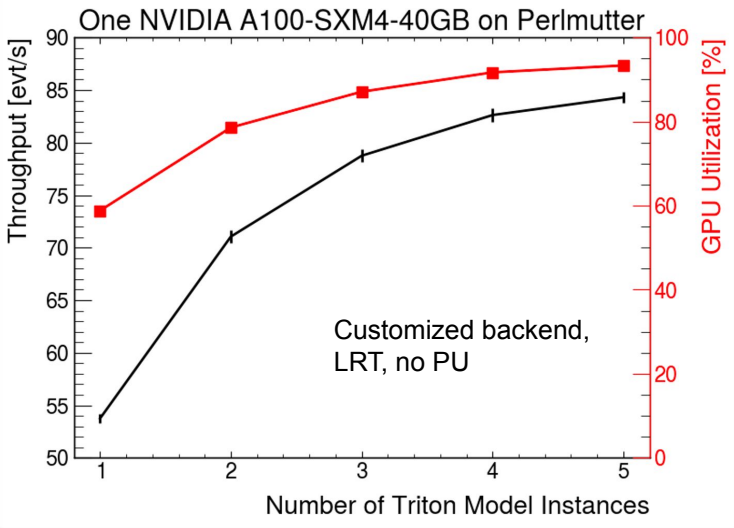
CTD 2022, CTD 2023

arXiv:2103.06995 GNN4ITkTeam / acorn · GitLab - https://gitlab.cern.ch/gnn4itkteam/acorn

Track 1: ACRON

GPU-based GNN Tracking (ExaTrkX) as a Service





- Increasing Triton model instances increases the GPU utilization and throughput
- Customized backend is better than an Ensemble model for a complex workflow like the GNN-based Tracking
- Direct inferences require higher concurrency to reach maximum throughput

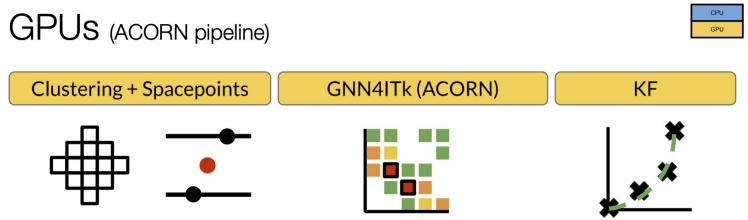
Track 2: ACORN + Traccc

- *Traccc* develops combinatorial KF (CKF) for accelerators
- A promising solution is to to combine ACRON + CKF (GPU), which can have full tracking chain end-to-end on GPUs

We are interested in working on adding this option

Need personpower to work on connecting the two components.

acts-project/traccc: Demonstrator tracking chain on accelerators - https://github.com/acts-project/traccc

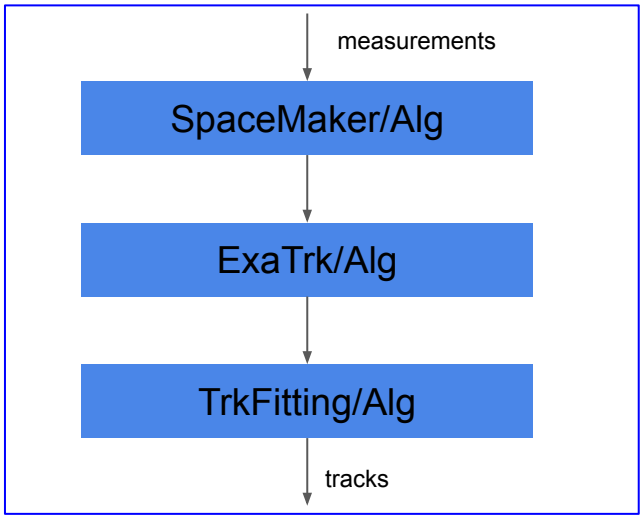


Category	Algorithms	CPU	CUDA	SYCL	Futhark
Clusterization	CCL	✅	✅	✅	✅
	Measurement creation	✅	✅	✅	✅
Seeding	Spacepoint formation	✅	✅	✅	⚪
	Spacepoint binning	✅	✅	✅	⚪
	Seed finding	✅	✅	✅	⚪
	Track param estimation	✅	✅	✅	⚪
Track finding	Combinatorial KF	✅	✅	🟡	⚪
Track fitting	KF	✅	✅	✅	⚪

✅: exists, 🟡: work started, ⚪: work not started yet

Traccc dev status

ACTS with GNN (ExaTraX) Plugin (Direct inference)

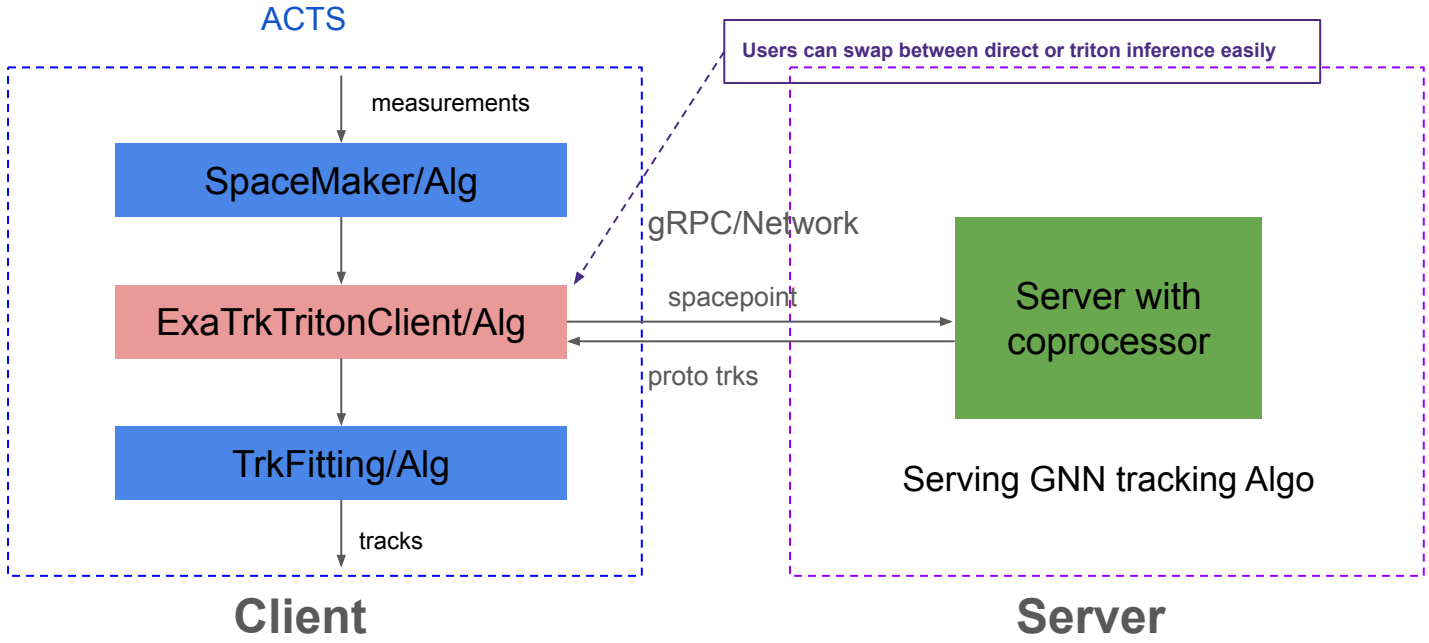


- ACTS contains a full tack reconstruction chain, which will be used in ATLAS for Run 4 offline tracking
- GNN TrackFinding (ExaTrk) can run locally with CPU/GPU
- ACTS TrackFitting still runs only on CPU

Acts - https://github.com/acts-project

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

Client added in ACTS to communicate with Server



Offload more algorithms to coprocessor to increase the throughput → **ACORN + Traccc**

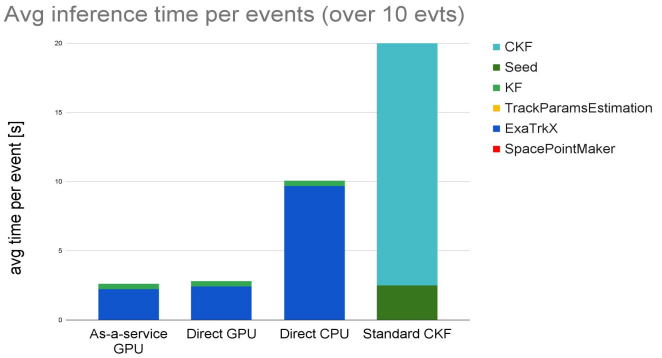
Performance of ACTS-as-a-service

Inference timing studies (ttbar PU=200, ODD)

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

GPU: 1x NVIDIA A100-SXM4-40GB

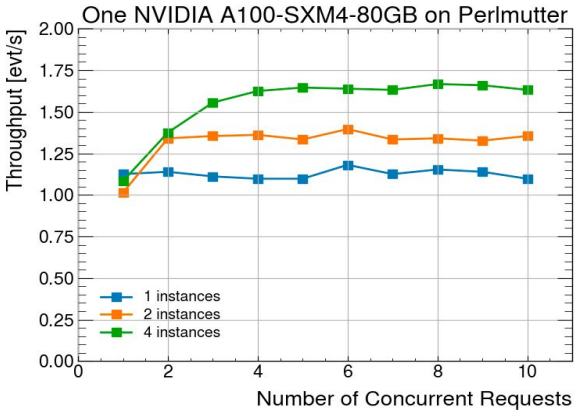
No additional overhead was observed in the as-a-service inference

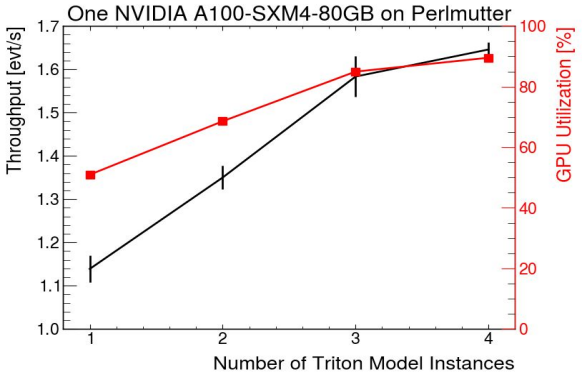


Throughput scaling

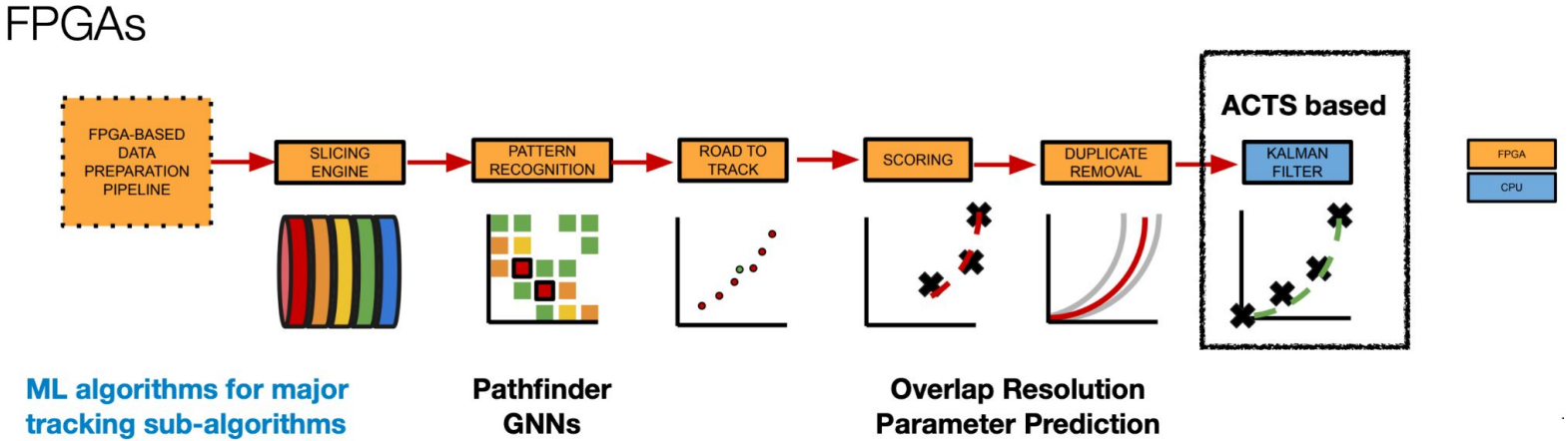
Multiple inference instance run on one GPU

Better utilization and higher throughput for one GPU



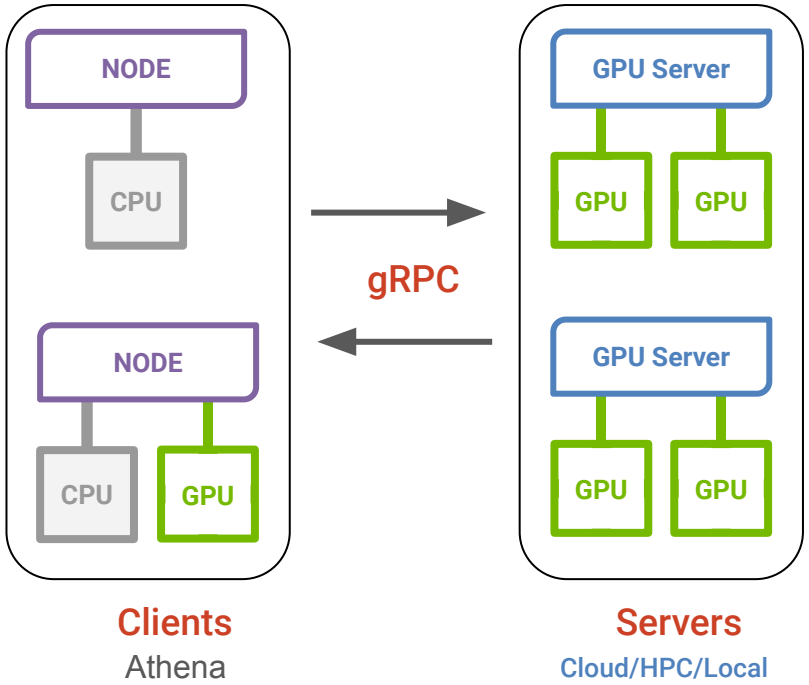


Track 3: FPGA Pipline



- Many ML sub-algorithms to build up the full FPGA pipeline
- Our A3D3 collaborators from UIUC and NYCU is working on porting GNN-based tracking to FPGA
- Both can benefit from the as-a-service approach

Inference as-a-service in Athena



At the end of the date, the chosen option needs to be integrated with Athena

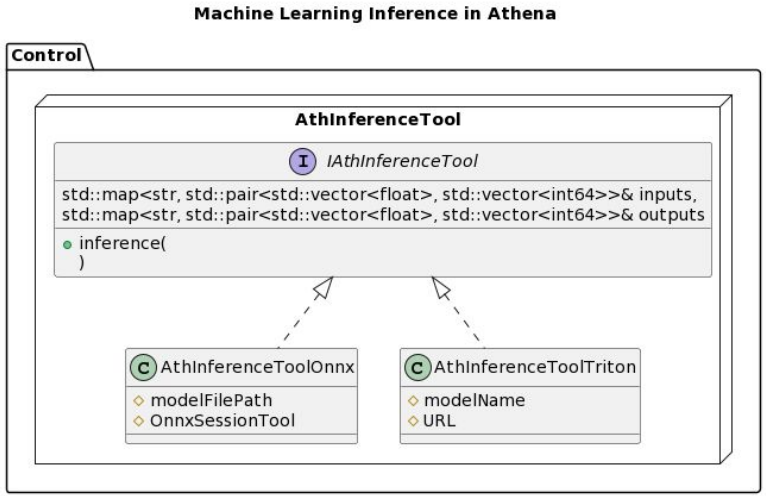
Inference as-a-service provides a way to free Athena from:

- Implementing the algorithms
- Installing their dependencies
- Adapting to algorithm updates
- Supporting them to run on different platforms

But Athena has to:

- Install client dependencies (gRPC and TritonClient)
- Check if the requested model matches the expectation

Inference as-a-service in Athena



- We propose a generic inference interface whose arguments are named-tuples for inputs and outputs,
- Two concrete implementations: one with ONNX and another with Triton.
- Can be swapped through configuration.
- Can apply to both online and offline as-a-service application

For Online tracking: Kubernete approach

For online tracking, local cluster farm at P1 would be the best option

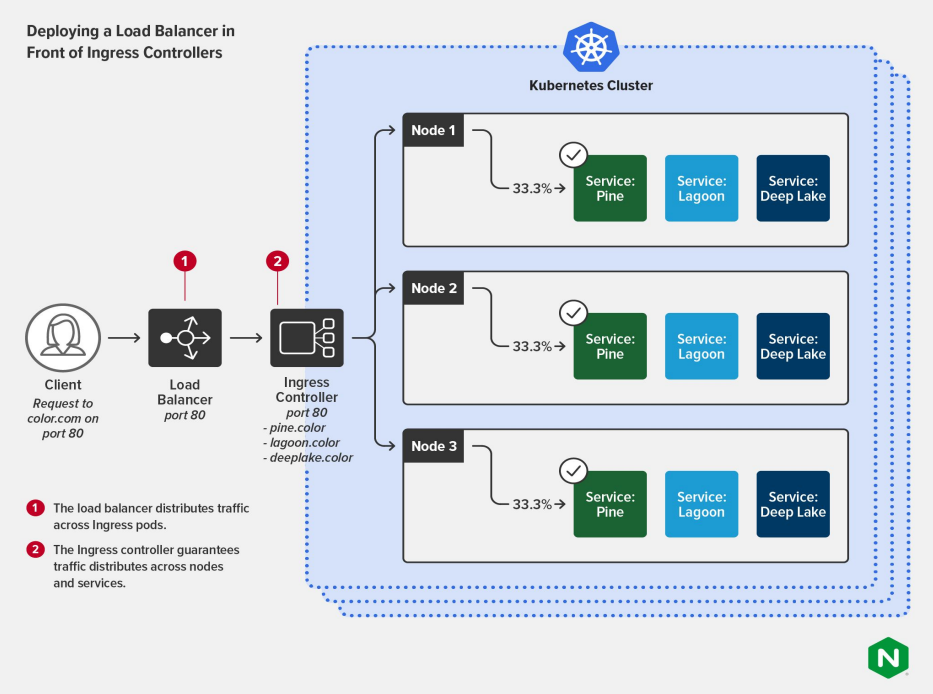


Image credit: https://www.nginx.com/blog/kubernetes-networking-101/

- Load Balancer, a public client-facing endpoint for all services delivered to clients outside the cluster
- Ingress Controller, distributes workloads within the cluster
- Node N, computing nodes that run containerized services

This is the approach industries use to deploy business applications/services.

Need personpower to investigate the feasibility this approach.

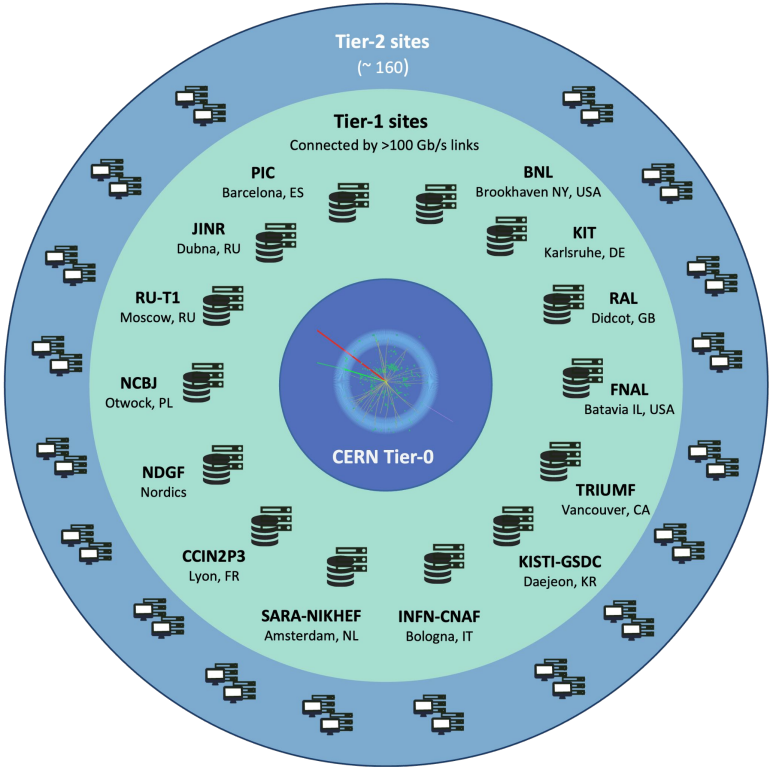
For offline reconstruction:

Scenario #1

Set up a Triton server with all supported models in each Tier 2 cluster. The server will respond to offline Athena IaaS job requests ***within*** *the Tier 2 cluster*. No public endpoint.

Averaged # of request frequency is about 30 requests per hour per Tier 2 cluster.

Maybe a small cluster without complicated *load balancer* is sufficient to respond to those concurrent requests.



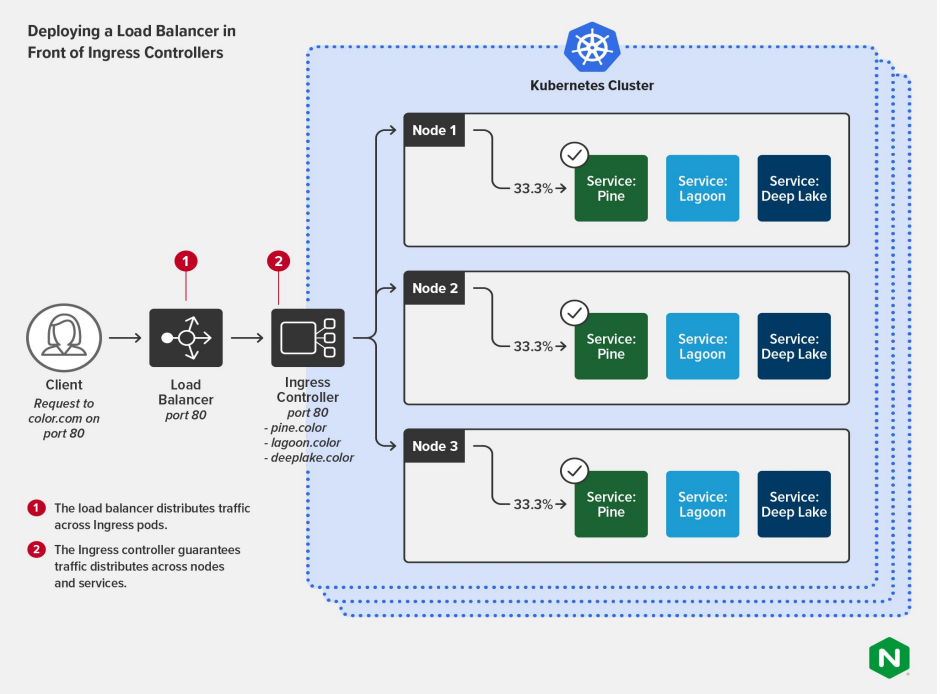
For offline reconstruction:

Scenario #2

Set up a Triton server with all supported models in High Performance Centers through a **public** client-facing endpoint.

Different HPCs have different as-a-service supports. E.g. one cannot run Kubernetes in Perlmutter.

From Athena client's point of view, it should be a matter of a different URL.



Practicalities

- Authentication and security
- Containization choice
- Load balancer implementation
- Model version control
- Dynamically figure out server URL rather than statically assign in job description

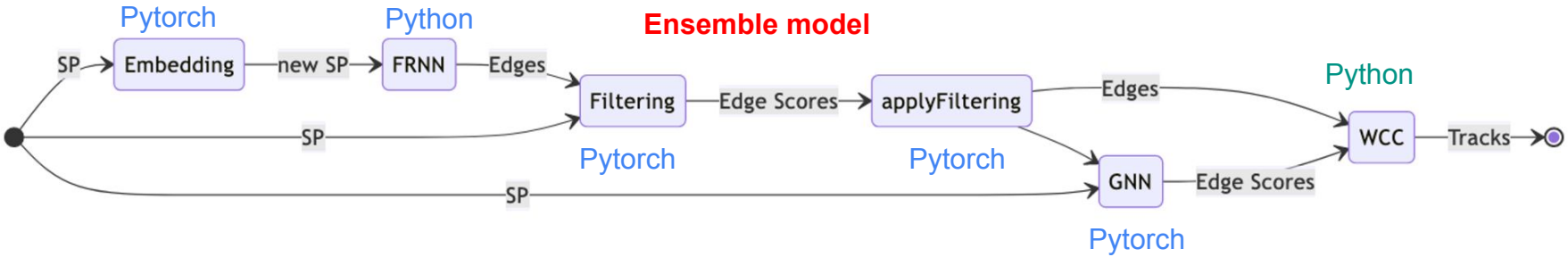
Summary

- ATLAS Phase-II EF Tracking upgrade requires a cost-efficient commercial solution to achieve high throughput for HL-LHC
- The ExaTrkX-as-a-Service based on the NVIDIA Triton server and used ACTS as the first client to use the tracking as a service in ATLAS.
- Integrating Triton (client) into Athena will facilitate the usage of remote & local coprocessors, including GPUs and FPGAs.
- Many options to deploy the Trion server. What is the best option is still an open topic
- This could benefit EF tracking development and aslo offline simulation and reconstruction.

Backup

Ensemble Backend

- GNN-Based Tracking is a complex workflow, consisting of 5 discrete sub-algorithms
- Ensemble scheduling uses greedy algorithms to schedule each algorithms
  - **Pros**: directly use existing Triton inference backends
  - **Cons**: little control with the data flow and algorithm scheduling, increasing the IO operations and latency



Customized Backend

- Customized backend provides means to receive requests from and send outputs to the client.
  - **Pros** : low overhead, full control of data flow and devices;
  - **Cons** : need to write user's own inference code.
- We build customized backends for the GPU-only ExaTrkX inference service and the CPU-only (fallback).