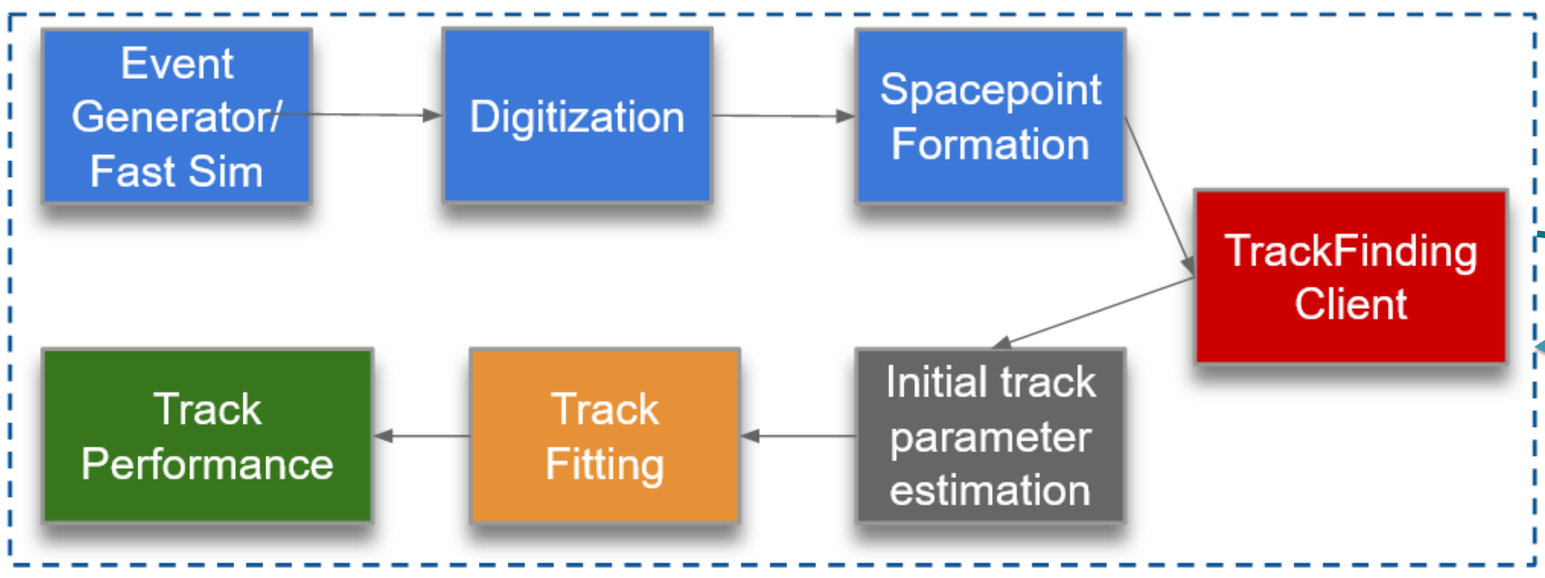
Graph Neural Network-based Tracking as a Service

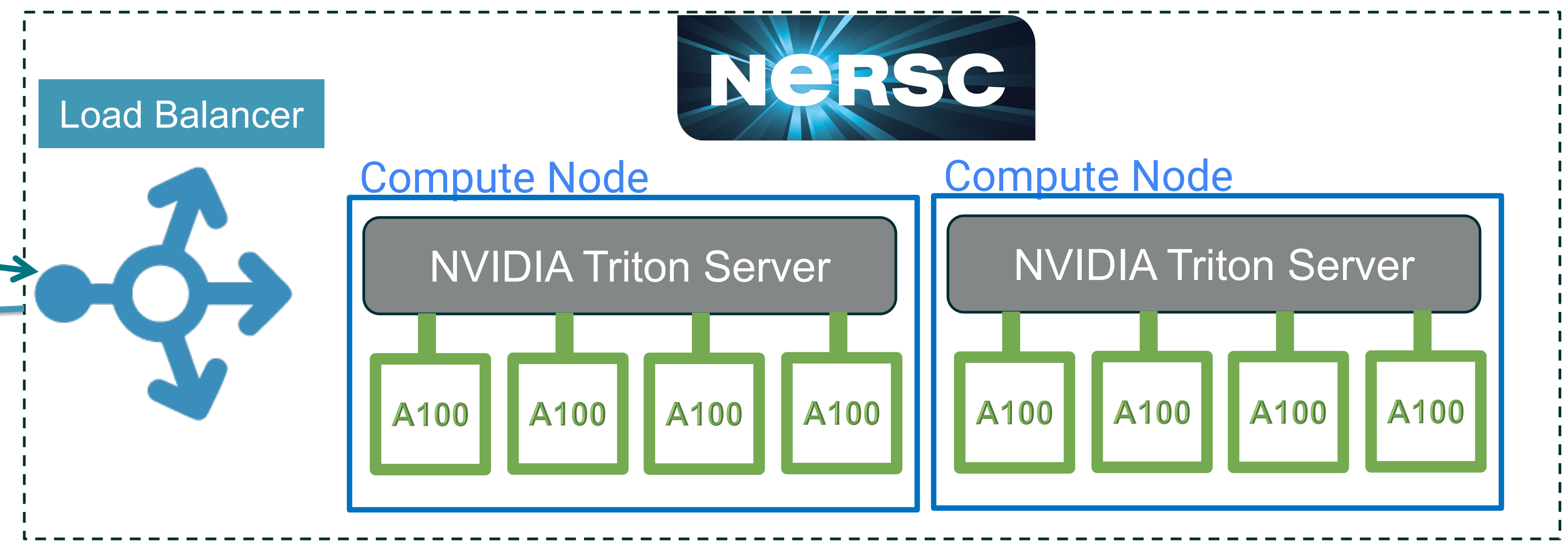
Xiangyang Ju[1], Elham E Khoda[3], Andrew Naylor[2], Haoran Zhao[3], coauthored with Paolo Calafiura[1], Steven Farrell[2], Shih-Chieh Hsu[3], William Patrick McCormack[4], Philip Coleman Harris[4] , Dylan Sheldon Rankin[5], Yongbin Feng[6]

1. LBNL, 2. NERSC, 3. University of Washington, A3D3, 4. MIT, 5. University of Pennsylvania, 6. FNAL

Main Workflow

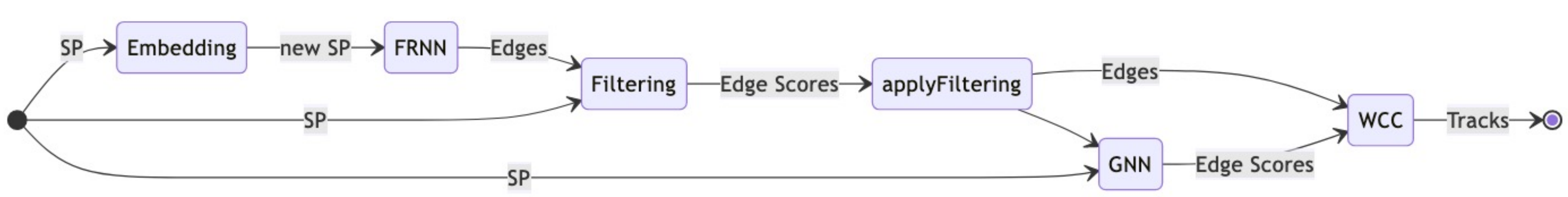


Server



Ensemble Backend

Algorithm	Backend
Embedding	Pytorch
Building (FRNN)	Python
Filtering	Pytorch
GNN	Pytorch
Track labeling (CC)	Python
ExaTrkX Model	**Ensemble**



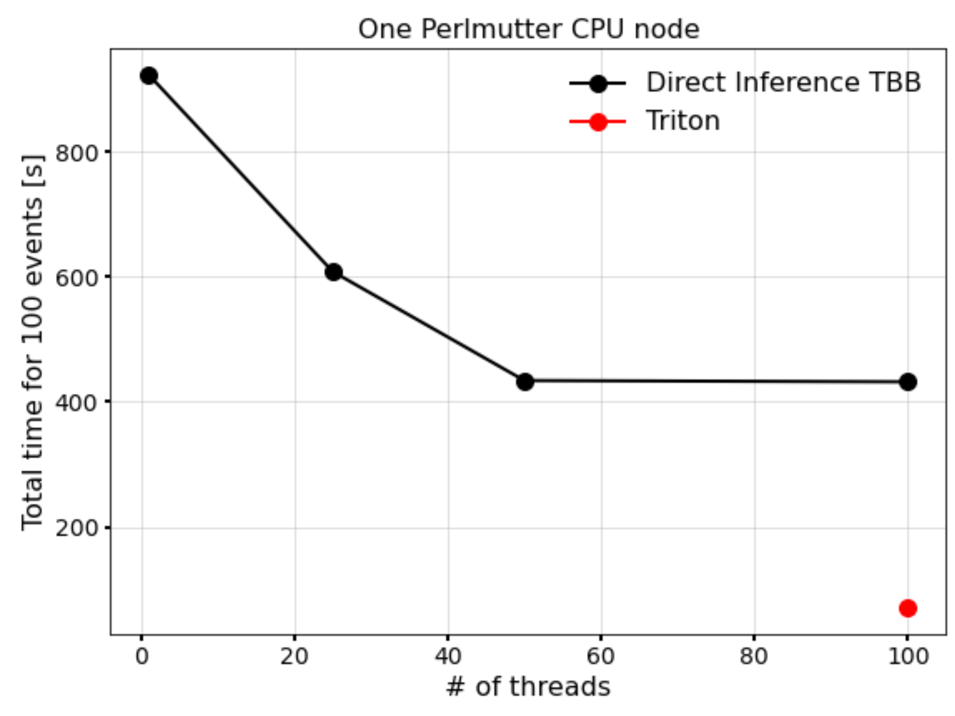
Ensemble scheduling uses greedy algorithms to schedule each model. ***Pros***: directly use existing Triton inference backends; ***Cons***: little control with the data flow and algorithm scheduling, increasing the IO operations and latency → data may be sent to a model in a different device

Customized Backend for CPUs and GPUs

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

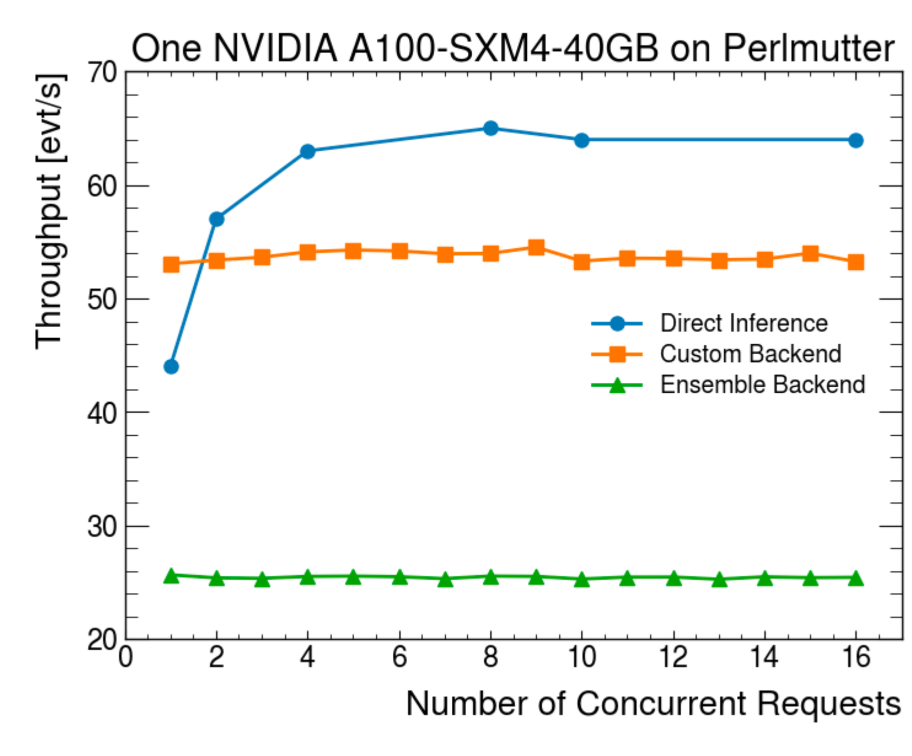
We build customized backends for the CPU-only and the GPU-only ExaTrkX inference service.

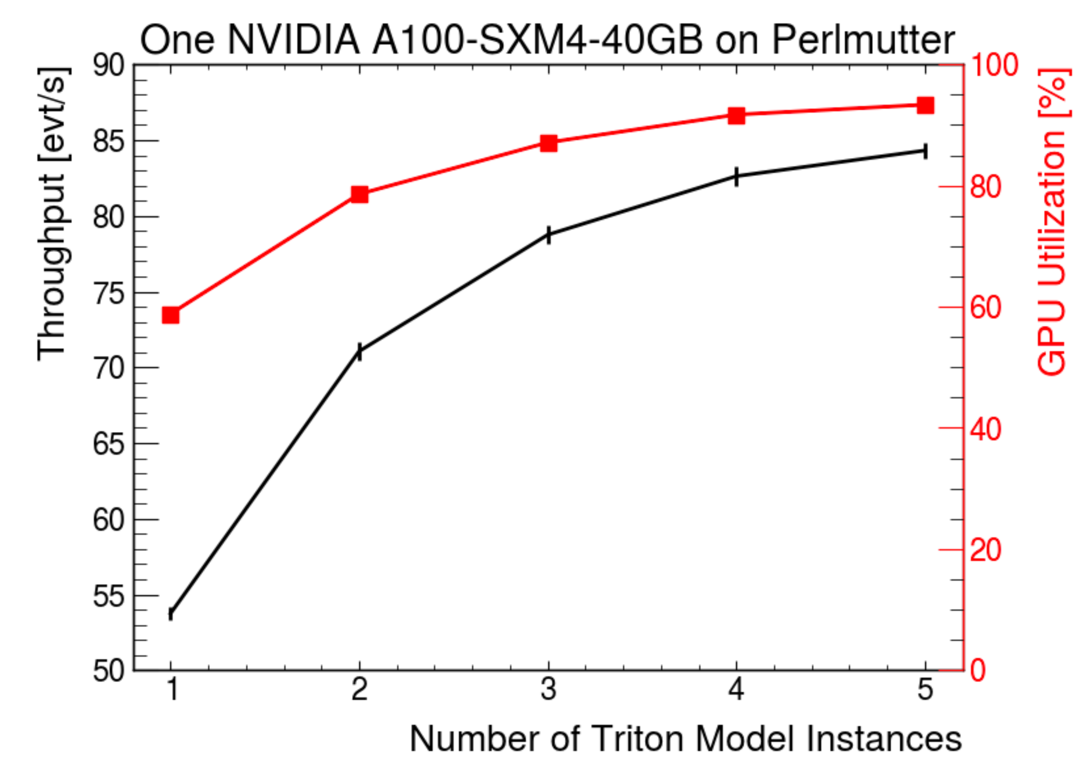
Results on CPU-based GNN Tracking Service



- Perlmutter CPU node: 2x AMD EPYC 7763 CPUs, 64 cores per CPU, 512 GB of DDR4 memory total, 204.8 GB/s memory bandwidth per CPU.
- Triton server better utilizes CPU cores. One possible explanation:
  - The buildEdges step uses the FAISS library, which uses multithreading too. There may be a clash of resource management between the external libraries and the TBB used in the main function

Results on GPU-based GNN Tracking Service





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

Conclusions and Outlook

- We implemented the first customized backend for the GNN-based Tracking as a Service and observed much better performance comparing with our previous ensemble backend implementation.
- We observed that Triton server can yield higher throughput than direct inference with an affordable number of instances (constrained by the device)
- Continue studies in the future
  - Measure the performance with more realistic data and models
  - Evaluate the performance with multiple GPUs and multiple GPU compute node
  - Measure the network latency
  - Estimate resource requirements for online data processing

Acknowledgement:
This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05CH11231 using NERSC award ERCAP0021226
Hsu is supported by NSF award No. 2117997.