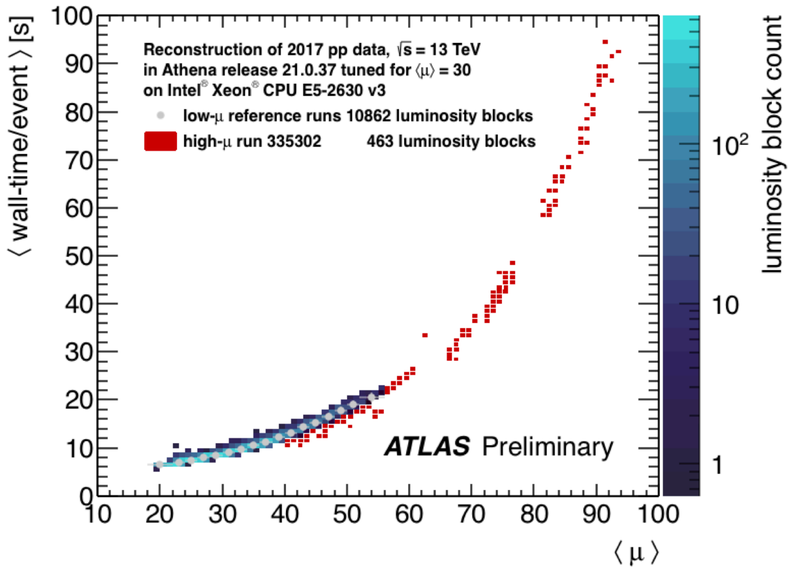
GPU-Accelerated Particle Tracking as-a-Service with the traccc Algorithm

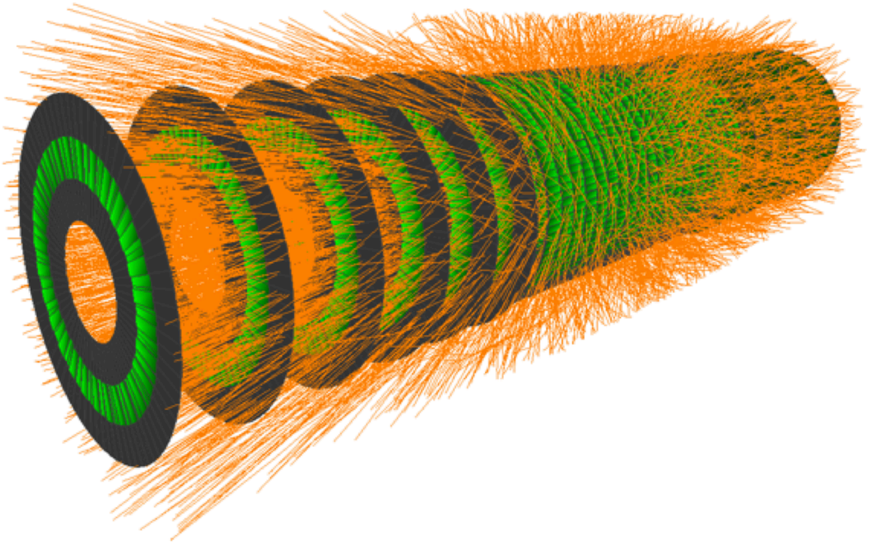
Miles Cochran-Branson*, Yuan-Tang Chou*, Xiangyang Ju**, Haoran Zhao*, Shih-Chieh Hsu*
University of Washington*, Lawrence Berkeley National Lab**
US LUA Annual Meeting
29 November 2024

DE-SC-0023527 PHYS-2117997

Challenges in tracking

- Reconstruction of tracks challenging for HL-LHC events
- Compute time grows super-linearly with μ
- Good environment for coprocessors such as GPUs (GNN tracking, traccc)





(Simulated) High pileup event at HL-LHC as seen by the TrackML detector (arXiv:2103.06995)

Core Questions

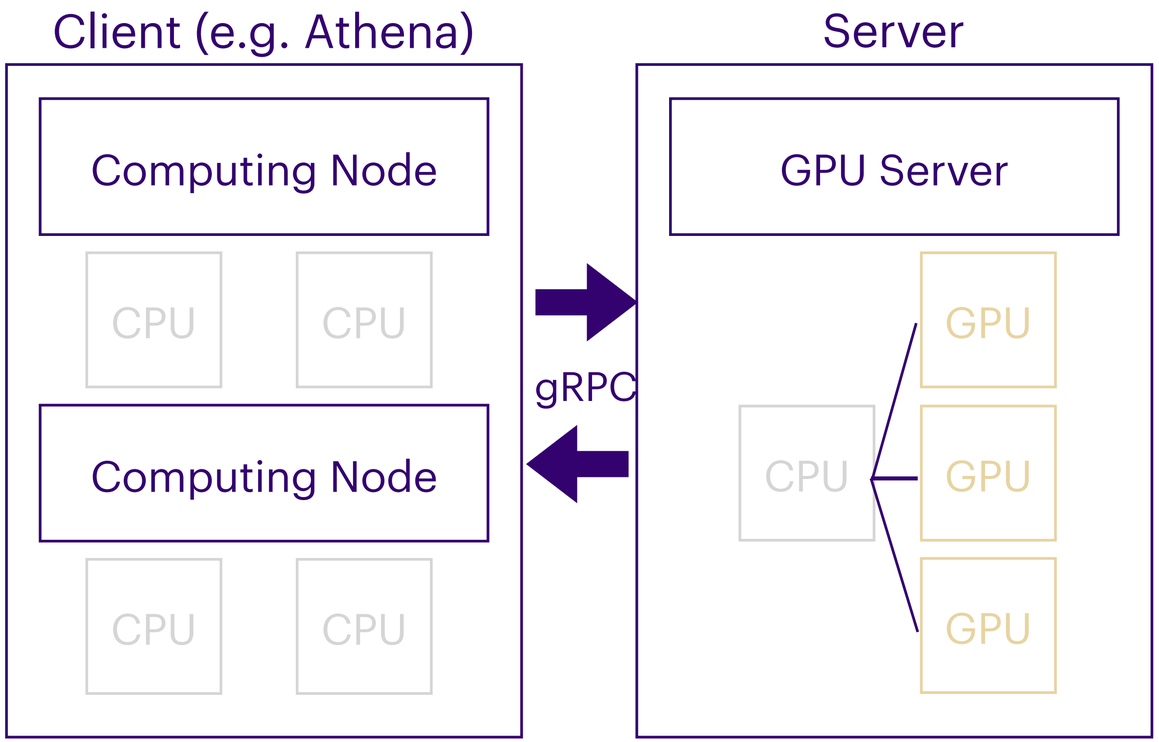
How to deal with expense of coprocessors in a limited budget?

How to enhance scalability of tracking with coprocessors?

How to improve integration in production framework (e.g., Athena)?

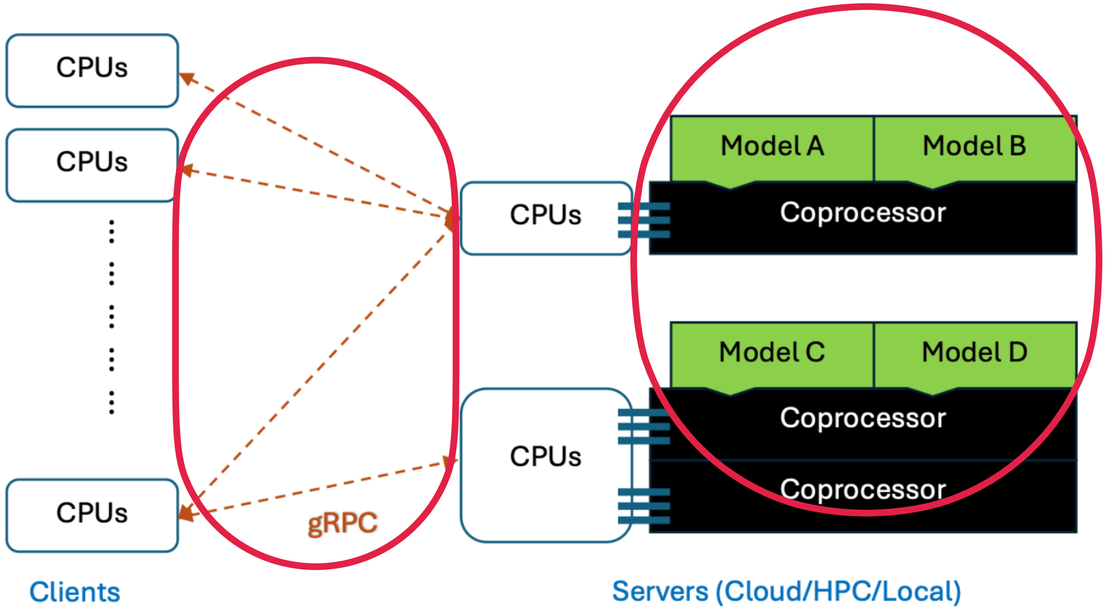
Inference as-a-Service

- Offload expensive coprocessor computation to dedicated server
- Advantages:
  - Can enhance throughput and resource utilization
  - Easier to incorporate inference frameworks in existing frameworks (ATHENA, CMSSW)
- Disadvantages
  - Added latency from data-sending
  - Often need custom backends to deal with server configuration
- Community for aaS in physics: SONIC



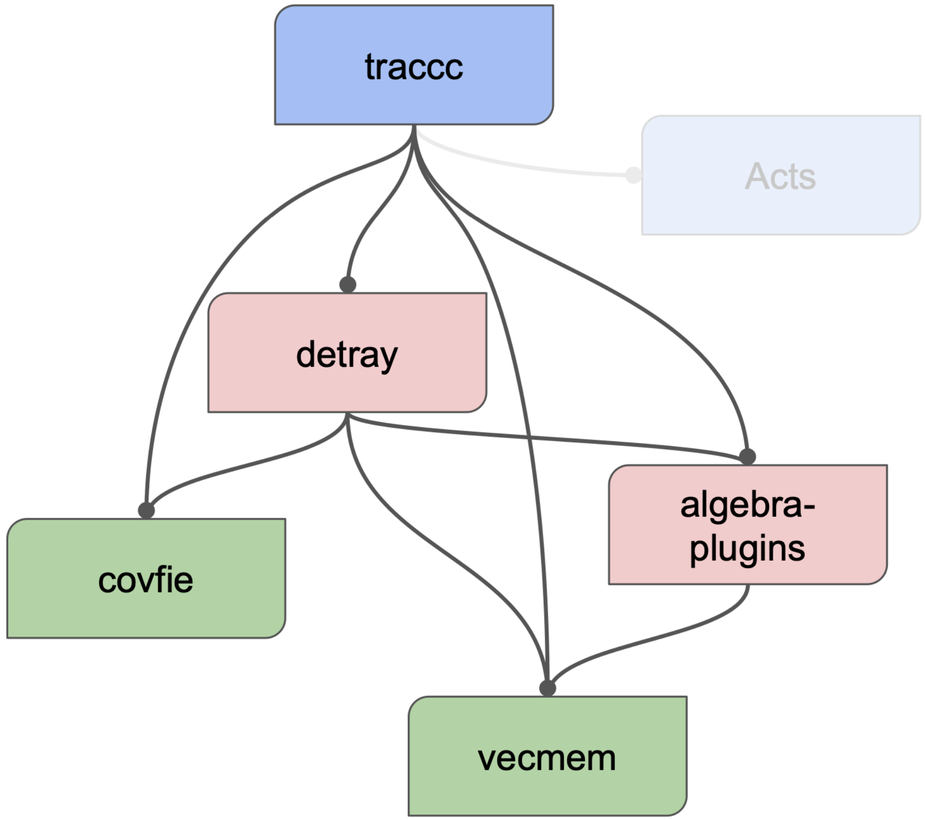
Increase throughput and evaluate performance

- To enhance performance:
  - Load multiple model instances onto server
  - Process multiple concurrent requests
- Metrics to evaluate performance:
  - Throughput
  - GPU utilization (often correlated to GPU FLOPs)
- Metrics measured with Nvidia's perf_analyzer tool
  - Uses nvidia-smi for GPU metrics
  - Measures latency and throughput over a time-window



Traccc

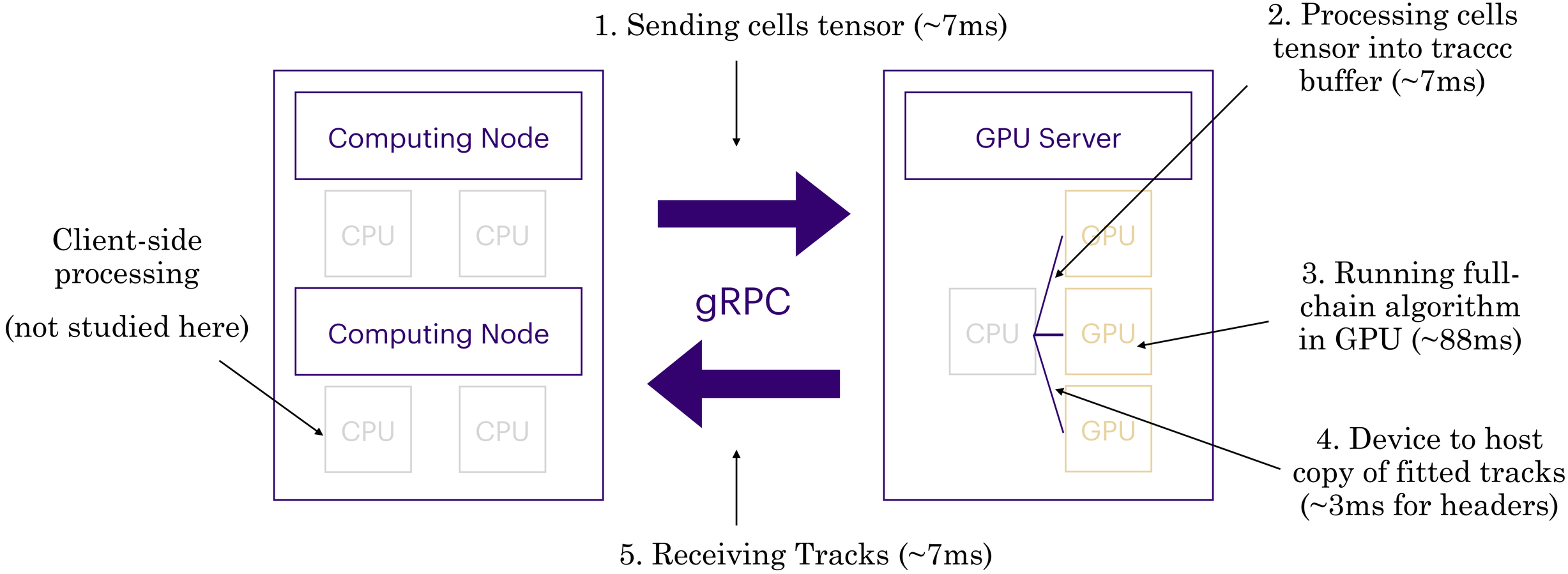
- Demonstrator tracking chain on accelerators
- Set of standalone tools developed outside ACTS framework
  - Currently being integrated in
- Uses ~the same methodology as ACTs
  - Combinatorial Kalman filter for fitting, etc.



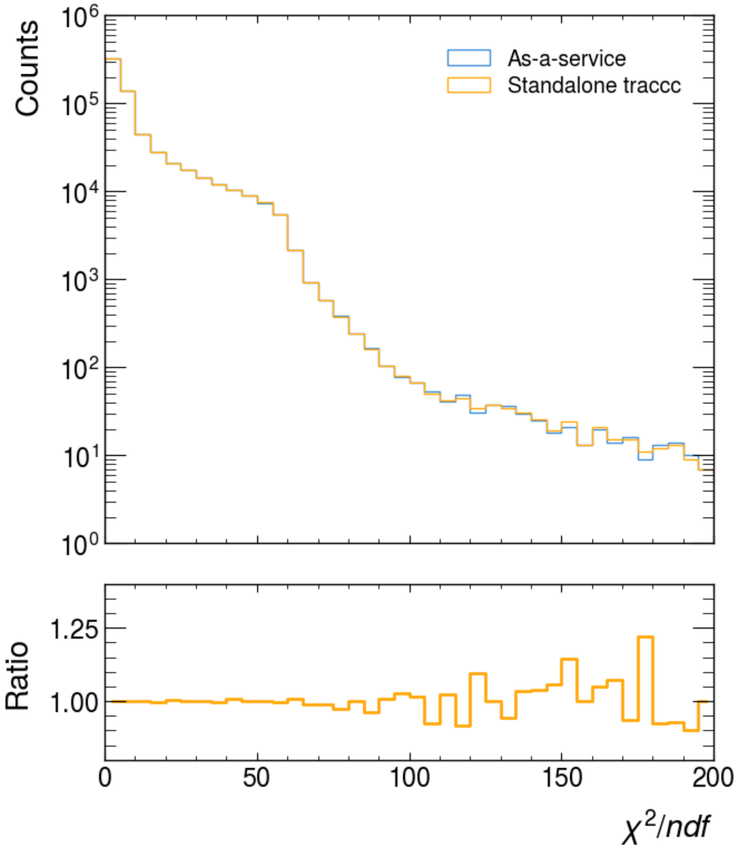
Details of as-a-Service Implementation

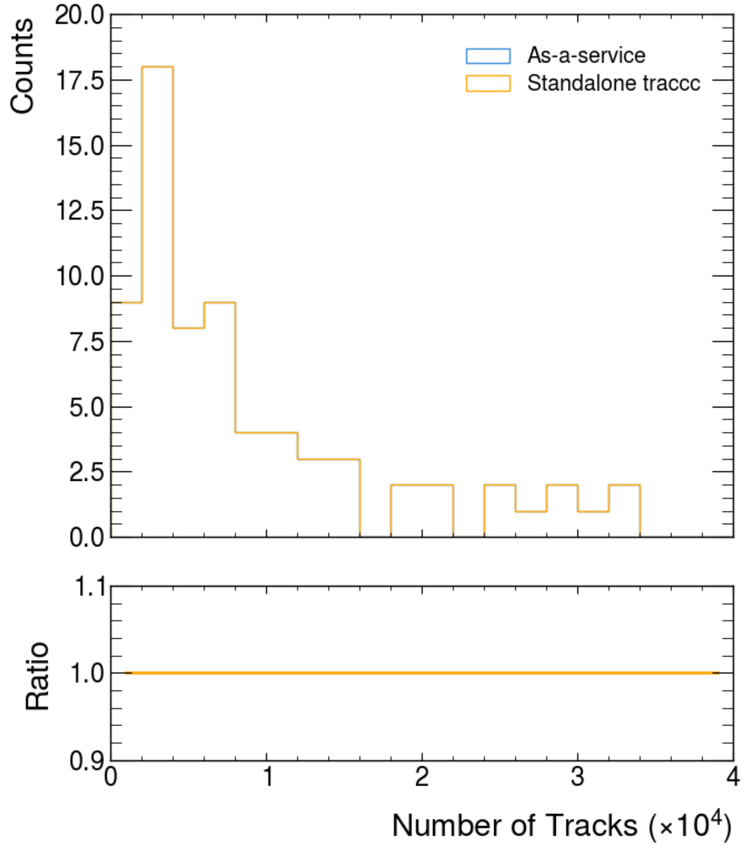
- Components of development
  - Lightweight wrapper to traccc
  - Custom backend to handle client requests
  - Lightweight client (future: Athena)
- Backend IO
  - **Inputs:** tensor of cells
    - Shape: (6 features, Number of cells)
  - **Outputs:** vector of fit results and parameters, i.e. χ^2, ndf, etc.
    - Shape: Number of fitted tracks
- Git repo
  - Using ODD detector, traccc v15

Sources of Latency



As-a-Service matches standalone

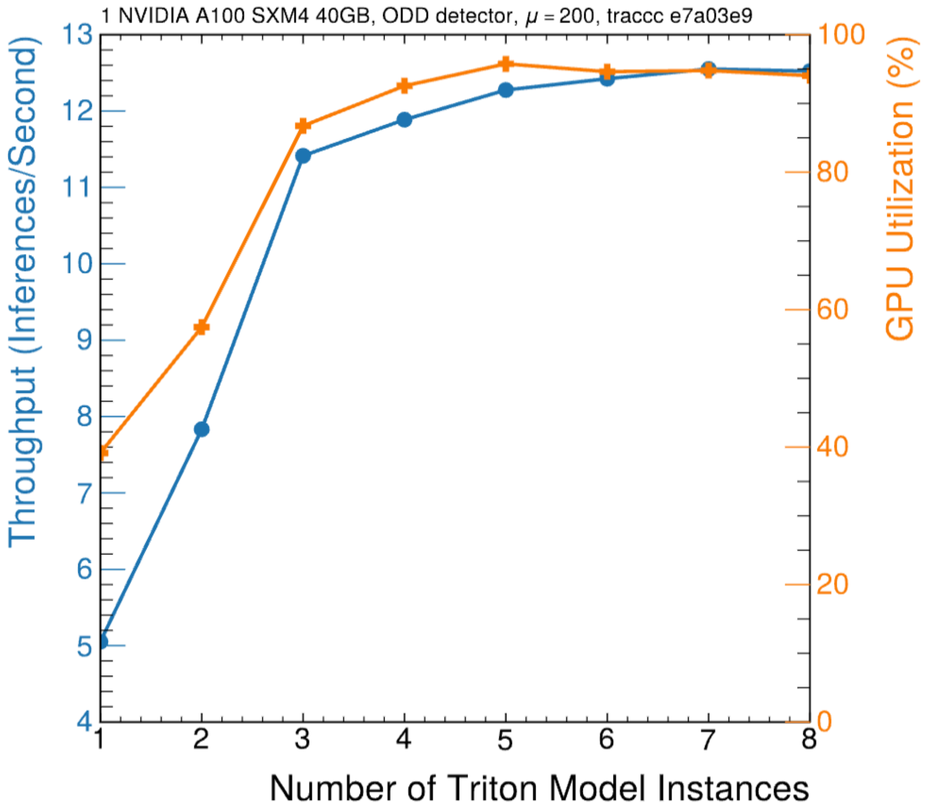




Exact match in number of tracks and ~perfect match in χ^2 distribution

Slight difference in χ^2 distribution from GPU error in fit

Current Performance



Greater than 2x improvement in throughput and increase in GPU utilization to >95% wrt one model instance

Conclusions

- Demonstrated successful implementation of traccc as-a-Serivce
- Improvement in throughput > 2x
- Better resource utilization, GPU utilization increase to > 95%
- **Increase in performance comes (almost) for free!**
  - Only need to develop light-weight wrapper and backend
- Can be applied to offline tracking at the HL-LHC

- In progress
  - Integration in Athena / adapting to use Itk geometry
  - Multi-GPU performance studies and scaling
  - Publication plan: targeting EF tracking paper with traccc performance, or Athena-Triton performance paper

References

Triton

Traccc

Traccc-aaS

SuperSONIC