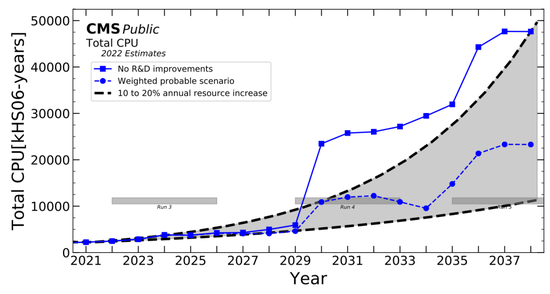
Inference as a Service for HEP ML Models on AMD GPUs Using the SONIC Framework

Ethan Colbert, Yongbin Feng, Mia Liu

Coprocessors to Meet Compute Needs

Computing needs in large experiments are increasing; it's uncertain whether CPU compute capabilities can keep up.



This projection is for computing needs of the CMS experiment during the upcoming HL-LHC (runs 4 and 5) [1]. Blue curves show projected demand, gray area shows projected increase in CPU capability.

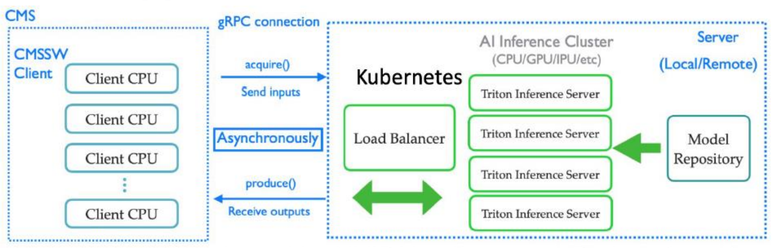
Recent developments in high-throughput computing have focused on **coprocessors**, specialized processors such as GPUs and FPGAs that trade flexibility for efficiency and speed. Effective use of such hardware is an opportunity to shrink the gap.

The "as-a-service" (AAS) approach helps significantly reduce idle time of coprocessors, making their use more cost-effective.

Triton, PyTriton, and SONIC

NVIDIA's Triton Inference Server [2] is a framework for serving inference with machine learning (ML) models. A Triton server can be created from a saved model and simple configs and receive requests via HTTP or gRPC. Inference is thus performed *as a service*.

The SONIC project uses the Triton framework to integrate coprocessors into high energy physics (HEP) workflows [3].



The PyTriton package, also developed by NVIDIA, allows any Python function that takes and returns tensors to be served via Triton. This gives the flexibility to do inference on any hardware available, including AMD GPUs (or other coprocessors). One drawback is that the Python interface is relatively slow – an example of trading speed for flexibility.

HEP ML Models

Two ML models common in HEP were used in this study: ParticleNet [4] and DeepMET [5]. The shapes of the input tensors are listed for both models (*N* = batch size). The models utilize two common ML inference frameworks – ONNX and Tensorflow, respectively.

ParticleNet – ONNX

ParticleNet is a graph neural network (GNN) model trained to perform jet tagging (classification). It takes as inputs reconstructed particles and secondary vertices (SV) within a jet. Inputs can vary in size – shapes shown are those used in this study.

Input Tensor	Shape
Particle Points	[N, 2, 100]
Particle Features	[N, 20, 100]
Particle Mask	[N, 1, 100]
SV Points	[N, 2, 10]
SV Features	[N, 11, 10]
SV Mask	[N, 1, 10]

DeepMET – Tensorflow

DeepMET is a deep neural network trained to estimate the missing transverse energy (p_T^{miss}) of an event. It takes all reconstructed particles as input; input tensors must be zero-padded to 4,500 particles.

Input Tensor	Shape
A	[N, 4500, 8]
B	[N, 4500, 1]
C	[N, 4500, 1]
D	[N, 4500, 1]

Measuring GPU Performance

To obtain a measurement of throughput performance, an instance of the model was created on a compute node equipped with a GPU. Nodes on the Gilbreth cluster at Purdue University and the AMD HPC Fund Research Cloud cluster were used. In this study, inference was performed directly (no Triton server).

We define latency as the time between when an inference call is made and when results are received. For latency ℓ and batch size N, throughput T can then be calculated as:

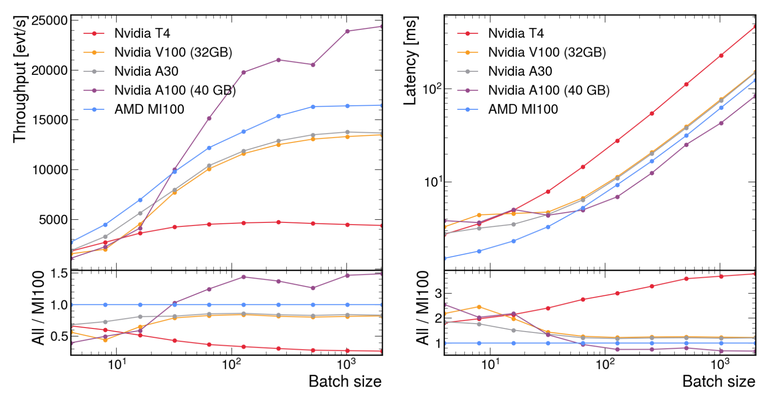
$$T = \frac{N}{\ell}$$

Measurements were made using varied batch sizes, between 4 and 2,048.

For each combination of GPU and batch size, 100 trials were performed. Each trial comprised randomly generating input tensors, recording a timestamp, running inference, and recording a second timestamp. Thus, only the time computing inference with the model is measured as latency.

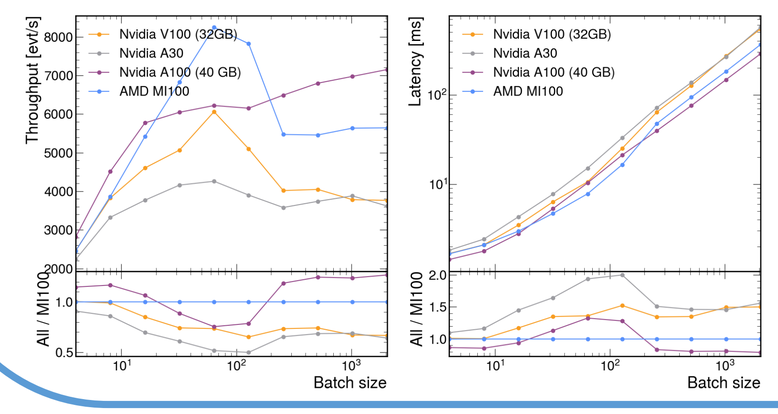
Throughput Results

ParticleNet



All GPUs show expected saturation behavior with ParticleNet – latency increases linearly with batch size once GPU is fully utilized. The AMD MI100 is observed to outperform the NVIDIA V100 and A30.

DeepMET



An interesting peak in throughput is observed with DeepMET; its cause is not yet fully understood. For large batches (after this peak), the AMD MI100 compares to other GPUs similarly as with ParticleNet.

Summary and Outlook

We have demonstrated that Triton inference servers can be used to serve inference on common ML models in HEP using AMD GPUs. Direct-inference tests show the AMD MI100 GPU has somewhat better throughput performance than the NVIDIA V100 and A30.

Further studies are ongoing to measure throughput performance using Triton servers via PyTriton and with a wider variety of AMD GPUs. Later, we hope to develop custom backends to use Triton servers with AMD GPUs in faster C++ environments.

References and Acknowledgements

[1] CMS-NOTE-2022-008
[2] NVIDIA (2024)
[3] CSBS 8, 17 (2024)
[4] Phys. Rev. D 101, 056019 (2020)
[5] Y. Feng, UMD (2020)

Many thanks to AMD for granting us an allocation on the AMD HPC Fund Research Cloud cluster for these studies.