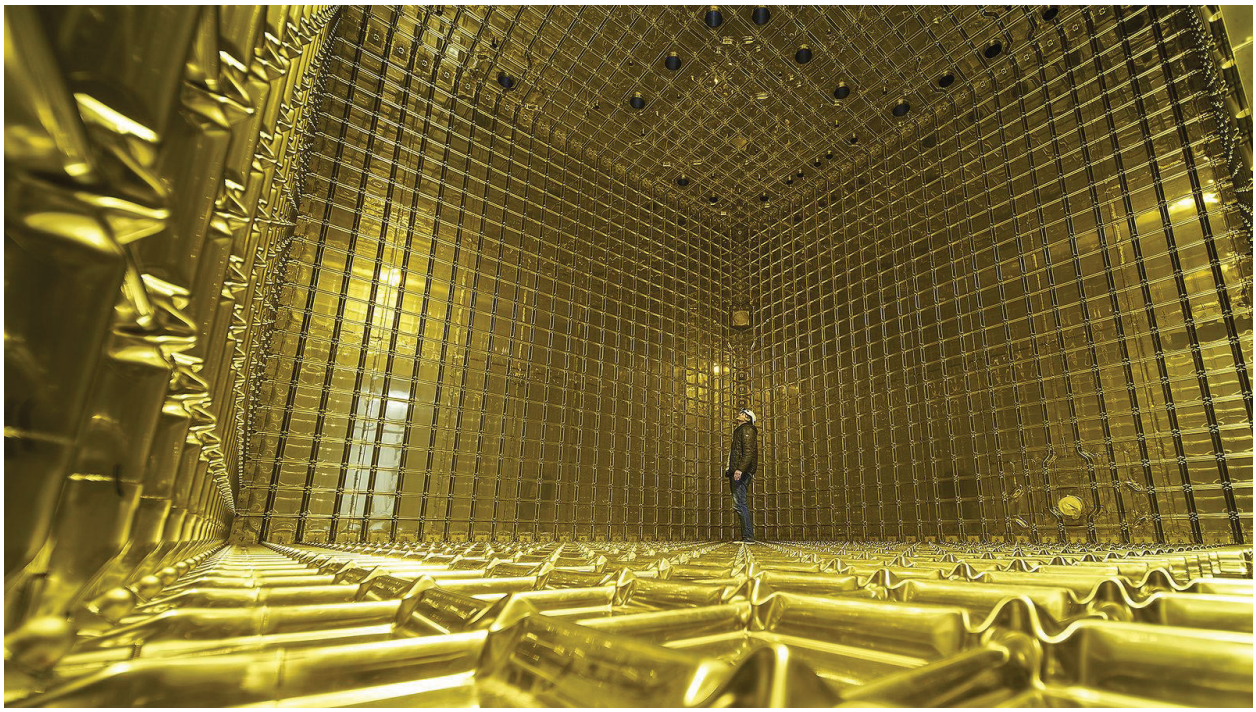
Scaling Inference Using Triton to Accelerate Particle Physics at the LHC and DUNE

Authors: Shankar Chandrasekaran[1], Lindsey Gray[2], Farah Hariri[1], Kevin Pedro[2], Nhan Tran[2], Tingjun Yang[2], Michael Wang[2]
[1] *NVIDIA Corporation*
[2] *Fermi National Accelerator Laboratory, Batavia, IL 60510, USA*
Tags: Triton, Inference, Kubernetes, GPU, scaling, NGC



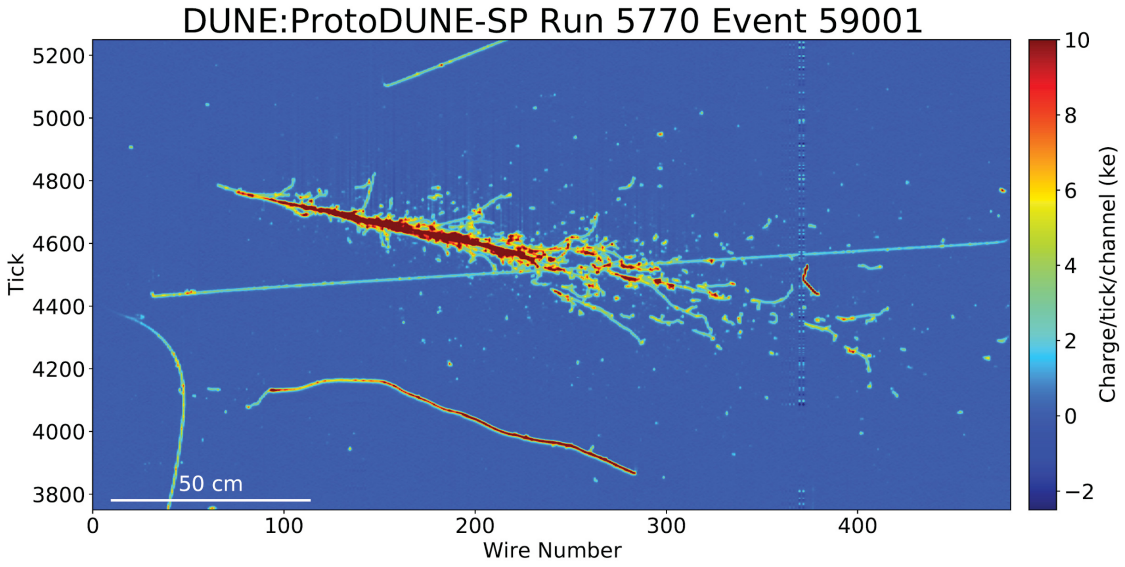
ProtoDUNE detectors for the Deep Underground Neutrino Experiment
[ALT: Provides a pictorial view of the inside of the experiment]
Photo by Maximilien Brice, CERN

Introduction

High energy physics research aims to understand the mysteries of the universe by describing the fundamental constituents of matter and the interactions between them. Diverse experiments exist on Earth to recreate the first instants of the universe. Two examples of the most complex experiments in the world are at the Large Hadron Collider (LHC) at CERN and the Deep Underground Neutrino Experiment (DUNE) at Fermilab. The LHC is home to the highest energy particle collisions in the world and the discovery of the Higgs boson. Detectors at the LHC are like ultra-high speed cameras, capturing the remnants of those collisions every 25 nanoseconds to create a 5D "image" in space, time, and energy. Physicists at the LHC collect huge datasets to find extremely rare events that may give clues about the Higgs boson as a portal to new physics or the particle nature of dark matter. The DUNE experiment sends a beam of particles called neutrinos from the west suburbs of Chicago to an underground mine 1,300 km away in South Dakota, where a massive 40 kton detector is being constructed 1.5 km beneath the earth’s surface to observe these very feebly interacting particles. Studying neutrinos can help us

answer questions such as the origin of matter in the universe and the behavior of core-collapse supernova in the Milky Way galaxy.

These experiments consist of unique and cutting-edge particle detectors that create massive, complex, and rich datasets with billions of events. They require sophisticated algorithms to reconstruct and interpret the data. Modern machine learning algorithms provide a powerful toolset to detect and classify particles, from familiar image processing convolutional neural networks to newer graph neural network architectures. A full reconstruction of these particle collisions requires novel approaches to handle the computing challenge of processing so much raw data. In a series of studies, physicists from Fermilab, CERN, and university groups explored how to accelerate their data processing using the Triton Inference Server.



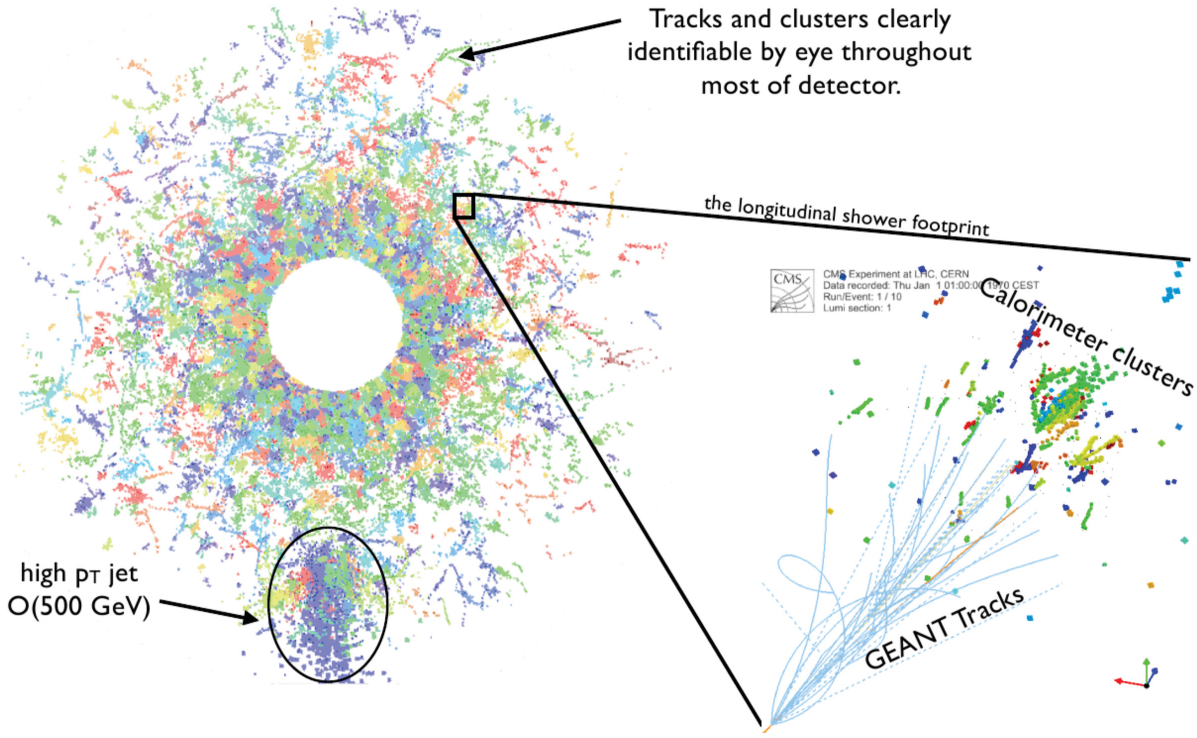
A 6 GeV/c electron event recorded by the ProtoDUNE-SP detector (run 5770, event 59001). The x-axis shows the wire number. The y-axis shows the time tick in the unit of 0.5 μs. The color scale represents charge deposition.
DUNE Collaboration, JINST 15 (2020) P12004

[ALT: To understand how the particle events are captured for ML processing]

As described in this paper, the full offline reconstruction chain for the ProtoDUNE-SP detector is a good representative of event reconstruction in present and future accelerator-based neutrino experiments. In each event, charged particles interact with the liquid argon in the detector, liberating ionization electrons that drift across the detector volume under the influence of an electric field. These electrons induce signals as they pass through and are collected by a set of wire planes at the end of the drift path. Two spatial coordinates can be determined from the different angular orientations of the wires in each plane, while the 3rd coordinate can be determined from the drift time of the ionization electrons. As a result, a very detailed 3D image of the neutrino interaction can be reconstructed. The most computationally intensive step of the reconstruction process involves a ML algorithm that looks at 48x48 pixel cutouts (or patches) representing small sections of the full event in order to identify the particles in them. Importantly, over the entire ProtoDUNE-SP detector, there are thousands of 48 × 48 patches to be classified,

such that a typical event may have approximately 55,000 patches to process. In the following section, we will discuss the performance implications of this process and how using Triton Inference Server allows us to scale the Deep Learning inference.

Similarly, for the LHC, a series of neural networks can be used to process data from low level cluster calibration to electron energy regression to jet (particle sprays) classification.



An illustration of how a similar paradigm is used for the LHC. Hits recorded by the calorimeter system are combined into clusters (zoomed-in section at right), which can then be further combined into higher-level reconstructed particle objects (such as the jet indicated at the bottom left). In simulated events such as this one, the reconstructed clusters can be related to the "truth" information from the simulation software (GEANT) in order to measure the accuracy of the algorithms.

Figure by Lindsey Gray, FNAL

[ALT: To show that the workflow is similar in LHC]

Computing-intensive process

For the ProtoDUNE-SP detector, the reconstruction processing time is dominated by running convolutional neural network inference for the thousands of patches in each event. When running inference on a typical CPU, this consumes 65% of the total time for reconstruction. The current dataset consists of 400TB from hundreds of millions of neutrino events. The team decided to use NVIDIA T4 GPUs to speed up this most computing-intensive process. In the initial trial phase, they used T4 instances on Google Cloud.

In production, thousands of client nodes feed detector data (images) into the reconstruction process. The scale of computing is so large that a distributed worldwide grid of computing resources is needed. This poses challenges to coordinating and optimizing resources shared by different sites worldwide. To cope with these challenges, the team decided to use a novel Inference as-a-service computing paradigm for the first time.

Inference as a service with Triton Inference Server

The team implemented their generic approach, called SONIC (Services for Optimized Network Inference on Coprocessors), for inference as a service using the Triton inference server. This technology is available from the NVIDIA NGC Catalog, a hub for GPU-optimized AI containers, models, and SDKs built to simplify and accelerate AI workflows. The Triton inference server simplifies the deployment of AI models at scale in production. It is an open source inference serving software package that lets teams deploy trained AI models from any framework (TensorFlow, TensorRT, PyTorch, ONNX Runtime, or a custom framework), from local storage, Google Cloud Platform, or AWS S3 on any GPU- or CPU-based infrastructure (cloud, data center, or edge).

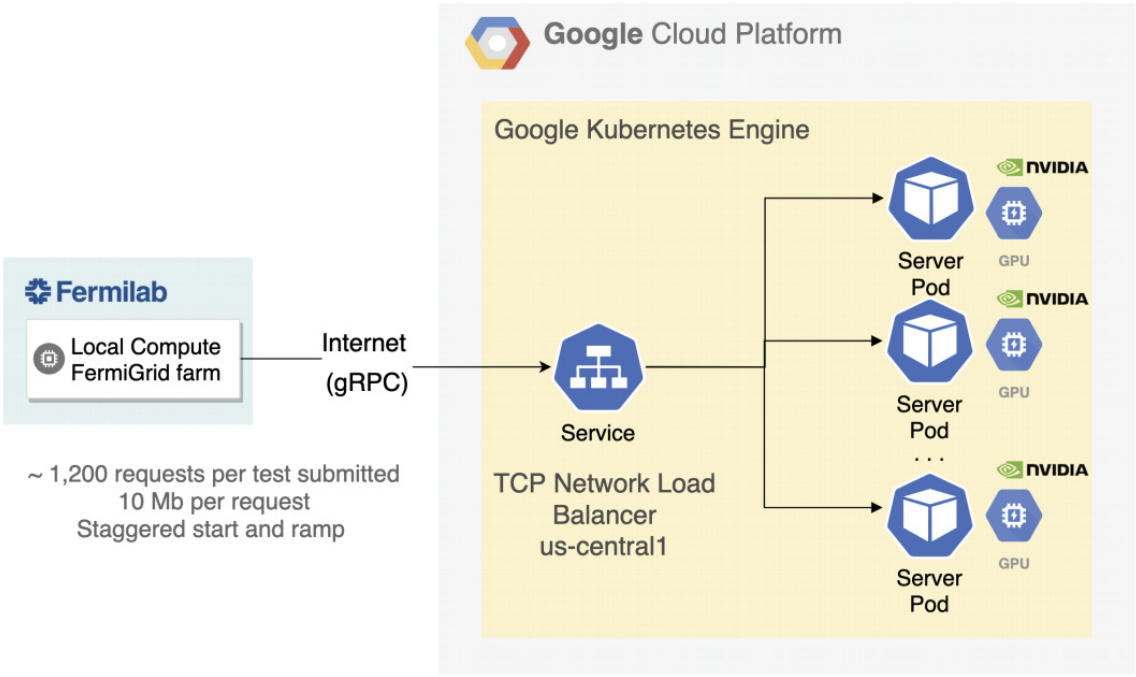
The team deployed the Triton server as a container and used Kubernetes to orchestrate the various cloud resources. Each GPU server in the cluster runs an instance of the Triton server. The clients run on separate, CPU-only nodes and send inference requests using gRPC over the network. Kubernetes handles load balancing and resource scaling for the GPU cluster.

Outcome

Use of T4 GPUs resulted in a 17x speed-up of the most time-consuming ML module of the workflow (track and particle shower hit identification). Overall workflow (event processing time) was accelerated by a factor of 2.7x.

The following are key benefits that the team achieved:

1. The workflow was accelerated without disruption to any of the other algorithms or experiment software.
2. In this deployment, many client nodes sent requests to a single GPU. This allowed heterogeneous resources to be allocated and re-allocated based on demand and task, providing significant flexibility and potential cost reduction.
3. There is a reduced dependency on open-source ML frameworks in the experimental code base. Otherwise, the experiment would be required to integrate and support separate C++ APIs for every framework in use.
4. The Triton software also utilized all available GPUs automatically when the servers had multiple GPUs, further increasing the flexibility of the server. In addition, Triton can execute multiple models from various ML frameworks concurrently.
5. The Triton inference server provides dynamic batching, which combines multiple requests into optimally-sized batches in order to perform inference as efficiently as possible for the task at hand. This effectively enables simultaneous processing of multiple events without any changes to the experiment software framework.



Architecture diagram of the Triton based inference as as service

[ALT: To understand the relative position of the various components in the service]

To scale the Nvidia T4 GPU throughput flexibly, a Google Kubernetes Engine (GKE) cluster was used for server-side workloads. Kubernetes Ingress was used as a load-balancing service to distribute incoming network traffic among the Triton Pods. Prometheus-based monitoring was used for:

1. System metrics from the underlying virtual machine
2. Kubernetes metrics for the overall health and state of the cluster,
3. Inference-specific metrics gathered from the Triton Inference Server via a built-in Prometheus publisher.

All metrics were visualized through a Grafana instance, also deployed within the same cluster. The team kept the Pod to Node ratio at 1:1 throughout the studies, with each Pod running an instance of the Triton Inference Server (v20.02-py3) from the NVIDIA NGC. The throughput was maximized when 68 CPU client processes sent requests to a single remote GPU. (This exact ratio depends on the algorithm and workflow.)

Summary

The offline neutrino reconstruction workflow was accelerated by deploying ML models on Nvidia T4 GPUs with the Triton Inference Server. Triton and Kubernetes helped the team implement inference as a scalable service in a flexible and cost-effective way. Though we focused on a result specific to neutrino physics, a similar result was achieved for the LHC and constitutes a successful proof of concept. These results pave the way for deploying DL inference as a service at scale in high energy physics experiments.

For more details:

- Neutrino (ProtoDUNE) results: arXiv:2009.04509

- LHC (CMS/ATLAS) results: arXiv:2007.10359
- SONIC approach: https://github.com/fastmachinelearning/SonicCMS
- Triton inference server: https://developer.nvidia.com/nvidia-triton-inference-server, NGC Catalog

Acknowledgements: We would like to thank, globally, the multi-institutional team that performed these neutrino and LHC studies - learn more about their work at fastmachinelearning.org.

This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.