Fermi National Accelerator Laboratory

Fermilab

Supercomputing for Accelerating Neutrino Physics

Elisabeth Petit – Bois
Georgia Institute of Technology – GEM Fellow

Supervisors:
Kenneth Herner
Michael Kirby
Andrew Norman
Fermi National Accelerator Laboratory

Managed by Fermi Research Alliance, LLC for the
U.S. Department of Energy Office of Science
www.fnal.gov

U.S. DEPARTMENT OF ENERGY | Office of Science

Table of Contents

1. Acknowledgements

A huge thank you to my supervisors – Kenneth Herner, Michael Kirby, and Andrew Norman – for being great advisors and mentors. Through them, I was able to connect with the FastML group: a group of researchers who have created the foundation of this work, lent a listening ear to this summer's progress and contributed their expert opinion on how to proceed. There would be nothing to report without everyone's support.

I am also grateful to Fermilab's SIST/GEM committee for successfully putting on a virtual program for the second year in a row while maintaining a quality education and experience.

Finally, I would also like to extend my gratitude to The National GEM Consortium for their role in making opportunities like these possible for myself and many other minorities in STEM looking to conduct research and pursue graduate education.

2. Background

The Deep Underground Neutrino Experiment (DUNE) is an international experiment headquartered at Fermi National Accelerator Laboratory (Fermilab) in Batavia, Illinois. The collaboration grew in an attempt to support ongoing efforts for studying neutrino physics.

DUNE, although still under construction, is an extremely ambitious neutrino physics plan. The completed project will be housed in the United States; however, there is a prototype at CERN known as ProtoDUNE. The final result is expected to consist of two particle detectors and a proton accelerator, powering the most intense neutrino beam in the world. The neutrino beam will travel through the Earth from Fermilab to the Sanford Underground Research Facility in South Dakota, spanning an impressive distance of around 1,300 kilometers total [1].

2.1 Accelerator Physics and Computing

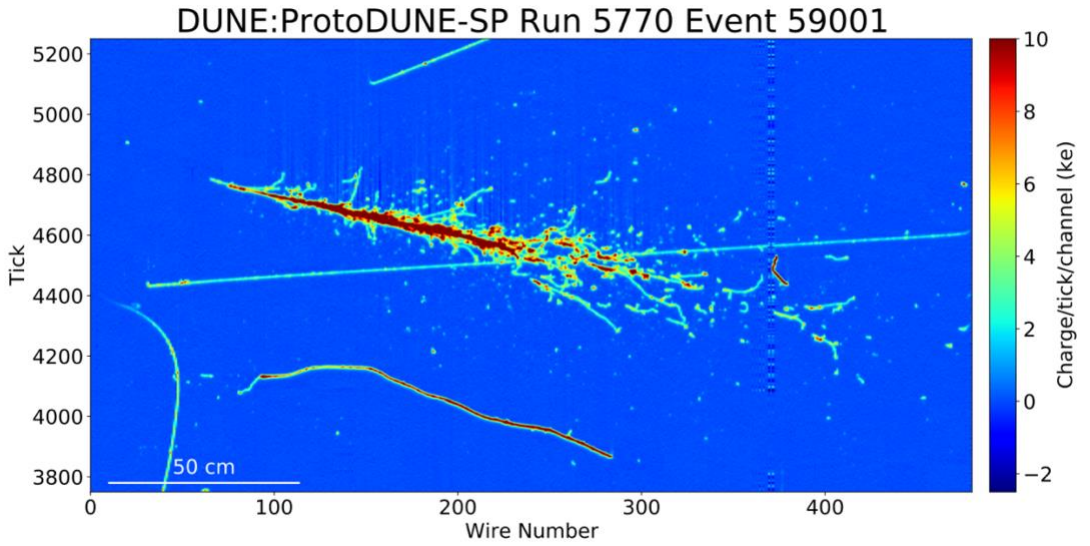

Figure 1: Example of ProtoDUNE Reconstructed event [2]

Particle accelerators – like those that will be used by DUNE – generate millions of events. Naturally, this large number of events require multi-petabyte storage systems. Figure 1 shows a single reconstructed event from ProtoDUNE-SP with color variations indicating amount of energy – or charge – released [2]. There are millions of these occurrences within detectors that may point to new physics.

To identify interesting interactions, computational physicists must use advanced mathematical techniques that can reduce electronic noise, reconstruct particle trajectories, and identify particle interactions. In fact, machine learning inference can effectively determine the types of interactions within a detector. The computation requirements from experiments like DUNE regularly require hundreds of millions of CPU hours to complete, so it is necessary to use strong computing resources, placing a significant strain on available resources and budgets.

2.2 Services for Optimized Network Interference on Coprocessors (SONIC)

To introduce powerful processing power into the particle analysis workflow in a unique way, Fermilab scientists from the FastML group proposed a heterogenous computing framework called “Services for Optimized Network Interference on Coprocessors (SONIC)” [3]. The framework allows for CPUs to communicate remotely with a GPU-equipped NVIDIA Triton inference server. Because multiple CPUs can access the GPU server as required, SONIC can function as a GPU-as-a-service (GPUaaS) model. There also can be limits placed on the server to ensure appropriate request routing to account for overloaded GPUs or heavy requests.

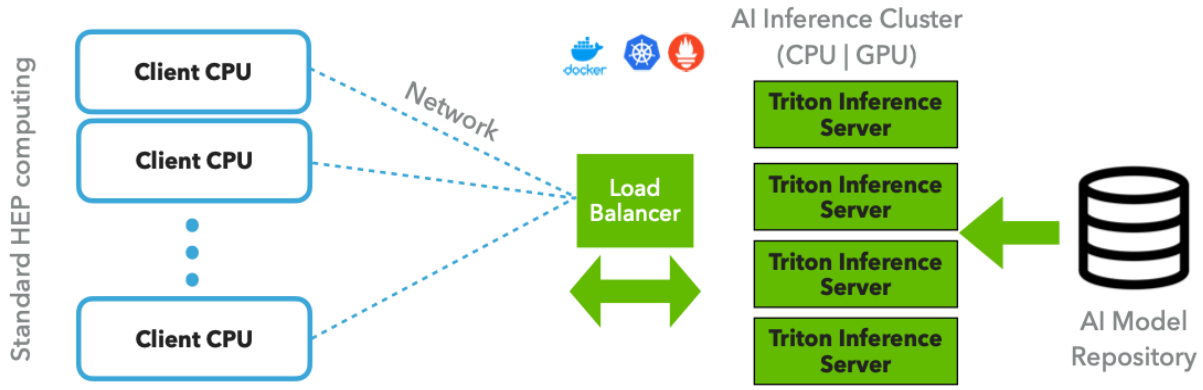

Figure 2: SONIC Workflow Overview [3]

Figure 2 describes the SONIC workflow [3]. Starting from the left-hand side, client CPUs can connect to the NVIDIA Triton inference server, managed by a load balancer. On the server, the cluster determines which ML model to run for the request, performs processing, and returns the response over the network. In the end, we have a workflow of CPUs and GPUs working together to speed up certain computing tasks, all facilitated by the SONIC model.

2.3 Challenges

While SONIC provides a fresh perspective on heterogenous computing, the model comes with a major drawback: network latency. Taking requests over the network can amount to unnecessary delay in processing, especially if the network available is not high-speed or the server is not replicated in a variety of regions. These two challenges can be avoided by localizing the SONIC framework, making use of in-house GPUs, and taking advantage of high-performance computing allocations made available to the lab by government partnerships.

3. Objective

The objective of this project is to speed up the SONIC framework by localizing the GPU server on a supercomputer. We expect speedup most noticeably in the ML inference stage of event processing. During our tests, we will also demonstrate the validity of the foundational SONIC proposal. To accomplish these goals, we will be using resources made available by the National Energy Research Scientific Computing Center, or NERSC [4]. NERSC hosts the Cori Supercomputer at Lawrence Berkeley National Lab.

4. Speed Tests

Understanding how our new approach to SONIC measures against the previous proposal, we must define measures of success and run comparable experiments. There are two metrics we are most concerned with in order to define SONIC's efficiency: average total event reconstruction time and average ML inference time.

The total event time defines the amount of time it takes for the script to completely process an event through four stages:

1. **Event generation** – initial stage of modeling the desired particle interaction.
2. **Geant4** – simulates how particles react when put through different types of matter, or, in this case, different detectors.
3. **Detector Simulation** – reads out wire signals and reconstructs momentum and curvature of particles based on hits.
4. **Reconstruction** – analyzes hits, tracks, and energy of reconstructed particles.

The average ML inference time, on the other hand, focuses on the reconstruction stage. Although it is only a part of reconstruction, it is one of the most computationally expensive. This section is where the SONIC framework shines brightest. The other stages are performed on the CPU while the reconstruction stage involves sending a request to the GPU server in order to analyze event information using a defined ML model.

4.1 FermiGrid Configuration vs. Cori Configuration

FermiGrid is a general purpose batch computing cluster for the Fermilab neutrino and muon experiments consisting of approximately 25,000 cores [5]. In the past, tests have been done using FermiGrid to validate the results shared in the SONIC proposal. We use these results to compare against information retrieved from Cori. It is important to recognize potential key differences in the two systems' configurations to better understand the outputs from the speed tests.

Firstly, FermiGrid is located in Batavia, Illinois and Cori is located in Berkeley, California. This places FermiGrid closer to the Ohio-based Google Cloud server hosting the Triton inference server for SONIC. When pinging the server, Fermilab's server averages a 12.5ms response while Cori receives a response in about 123ms. Because the two ping averages vary greatly, it may be due to a network configuration differences, location, or both.

FermiGrid runs on a variety of Intel 64-bit processors. In our speed tests for Cori, we fluctuate between a Knights Landing and Haswell chips to make our results more robust. The Knights Landing chip is less performant than the Haswell chip.

Finally, we ran different files between the two tests. FermiGrid ran on five data files sourced from ProtoDUNE while Cori ran two Monte-Carlo data files. The files are labeled accordingly in the tables in sections 4.2 and 4.3 in tables 1 through 4.

4.2 Event reconstruction speed test for ProtoDUNE using FermiGrid

The table below describes results after running five ProtoDUNE data files. We record the average total event reconstruction time and the ML inference time for CPU-only executions (Table 1) and GPU-incorporated executions (Table 2).

Table 1: CPU Speed Test Results for FermiGrid

File	Avg total event time (s)	ML inference time (s)
1	844.531	328.192
2	983.785	359.097
3	880.617	323.983
4	840.133	306.585
5	527.301	188.112

Table 2: GPU Speed Test Results for FermiGrid

File	Avg total event time (s)	ML inference time (s)
1	566.278	21.813
2	431.857	20.792
3	382.902	20.694
4	634.182	24.325
5	404.612	20.279

4.3 Event reconstruction speed test for ProtoDUNE using Cori

The table below describes results after running two ProtoDUNE Monte-Carlo files. We record the average total event reconstruction time and the ML inference time for CPU-only executions (Table 3) and GPU-incorporated executions (Table 4).

Table 3: CPU Speed Test Results for Cori			
File	Processor	Avg total event time (s)	ML inference time (s)
6	KNL	1344.94	827.136
6	Haswell	249.03	147.057
7	KNL	1118.84	717.792
7	Haswell	203.805	127.486

Table 4: GPU Speed Test Results for Cori			
File	Processor	Avg total event time (s)	ML inference time (s)
6	KNL	612.638	99.694
6	Haswell	202.339	103.290
7	KNL	456.305	63.572
7	Haswell	264.244	186.394

4.4 Results Discussion

In both FermiGrid and Cori, we can see that the SONIC framework decreases the overall time for event processing and ML inference. The change between the CPU and GPU timing is most noticeable in the ML inference timing with ratios varying between 9.28 and 17.27 for FermiGrid and 8.30 and 11.29 for Cori's KNL processor. When using the Haswell processor, Cori seems to move much faster when processing overall events, but it fails to demonstrate significant speedup between CPU and GPU times. In fact, in file 2, the CPU timings outperform the GPU timings.

What is most peculiar about these results is that FermiGrid seems to have far superior ML inference times around 20s versus Cori's much higher results. It is possible that these metrics tie back to the different configurations of the computer systems. The most probable culprit is the location factor of these servers. As FermiGrid is physically closer to the Google Cloud Triton server, it must be sending and receiving results faster than Cori. However, it is very possible that other factors are related to this significant change.

5. Future Work

The work this summer validated how useful SONIC can be when trying to speedup ML inference in the ProtoDUNE reconstruction workflow. However, we were unable to obtain GPU resources at NERSC to localize the computing framework and investigate how much faster total event processing time and ML inference time could complete. While this is the case, we did uncover unexpected results, showing us that GPUaaS performance relies heavily on the network available.

In the future, we hope to secure the tools necessary to run SONIC on a supercomputer or a computing system equipped with local GPUs. This way, we can see for ourselves how this heterogeneous setup lends itself towards faster analysis times. We also find the timing

discrepancies between Cori and FermiGrid particularly fascinating, so it may be interesting to further investigate why one system outperforms the other in the case that network latency is not the only cause.

6. References

[1] Deep Underground Neutrino Experiment (DUNE) Homepage: https://www.dunescience.org/

[2] B. Abi et al [DUNE Collaboration], JINST 15, P12004 (2020). https://doi.org/10.1088/1748-0221/15/12/P12004.

[3] Wang, Michael, Yang, Tingjun, Acosta Flechas, Maria, Harris, Philip, Hawks, Benjamin, Holzman, Burt, Knoepfel, Kyle, Krupa, Jeffrey, Pedro, Kevin, & Tran, Nhan. *GPU-accelerated machine learning inference as a service for computing in neutrino experiments*. United States. https://www.frontiersin.org/articles/10.3389/fdata.2020.604083/full.

[4] National Energy Research Scientific Computing Center (NERSC) Homepage: https://www.nersc.gov

[5] Computing for Neutrino and Muon Physics: https://computing.fnal.gov/neutrino-muon-physics-computing/