Fermilab | U.S. DEPARTMENT OF ENERGY | Office of Science

HEPnOS: a Specialized Data Service for High Energy Physics Analysis

Sajid Ali *for* SciDAC4 HEPonHPC project
Fermi National Accelerator Laboratory

ESSA 2023 @ IEEE IPDPS

In partnership with:

U.S. DEPARTMENT OF ENERGY | Office of Science | Argonne NATIONAL LABORATORY | University of CINCINNATI | Colorado State University

Introduction

- Present day HEP analyses map data processing tasks onto computer cores by assigning 1 core : 1 file.
- Each file contains many atomic units of data that can be processed independently
  - these are called events.
- The file-based assignment limits the maximum number of cores that can be used for analyzing a dataset.
- The file-based workflows introduce artificial bottlenecks that make our use of HPC resources inefficient.
- HPC clusters have nodes that are connected by **low latency, high bandwidth interconnects**.
- We propose to **remove this bottleneck** by using a distributed data service, HEPnOS that can **harness the interconnects**.
- In this paper, we demonstrate the reading speed and scalability (both weak and strong) of HEPnOS, using sample from a neutrino physics experiment, NOvA.

Background: Neutrino Physics

- Neutrino: elementary particle which holds no electrical charge, weighs a million times less than the electron and passes through ordinary matter with virtually no interaction!
- Abundance: a thousand trillion pass harmlessly through your body every second!
- There are three flavor states of neutrino: electron, muon and tau (and three mass states formed as a linear superposition of the flavor states).
- As a neutrino travels, it may switch back and forth between the flavors. These flavor “oscillations” confounded physicists for decades.

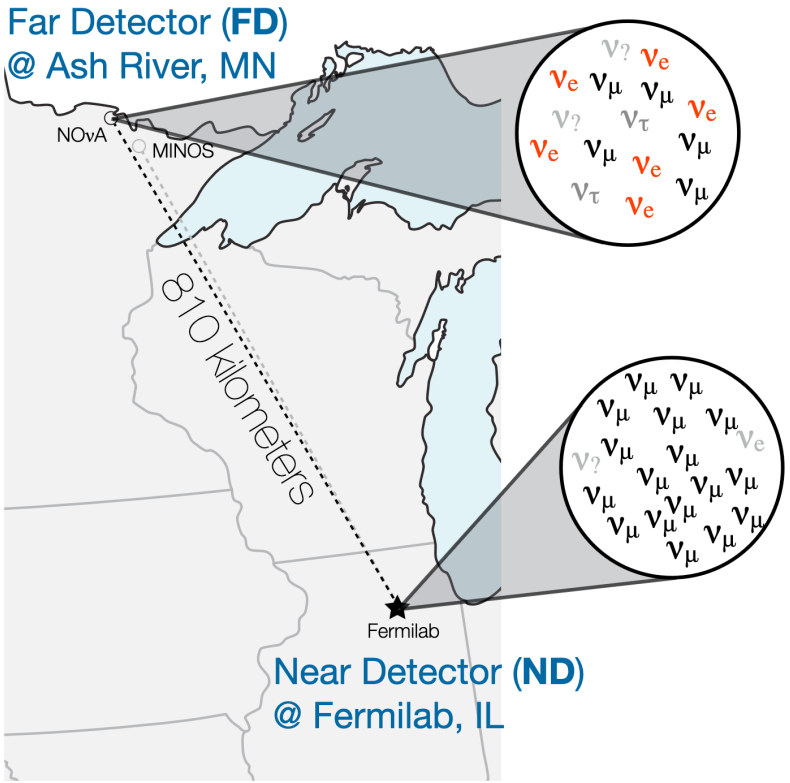
Motivation for NOvA experiment

- The NOvA experiment is constructed to answer the following important questions:
  - What is the relative ordering of the mass states of the neutrinos?
  - Do neutrinos violate charge parity symmetry? (i.e. do anti-neutrinos oscillate the same way as neutrinos)
  - Gather information regarding the oscillation parameters to examine muon-tau symmetry.

- NOvA – (**N**eutrinos from the Main Injector) **O**ff-Axis Electron Neutrino($\boldsymbol{\nu_e}$) **A**ppearance

The NOvA experiment: Overview

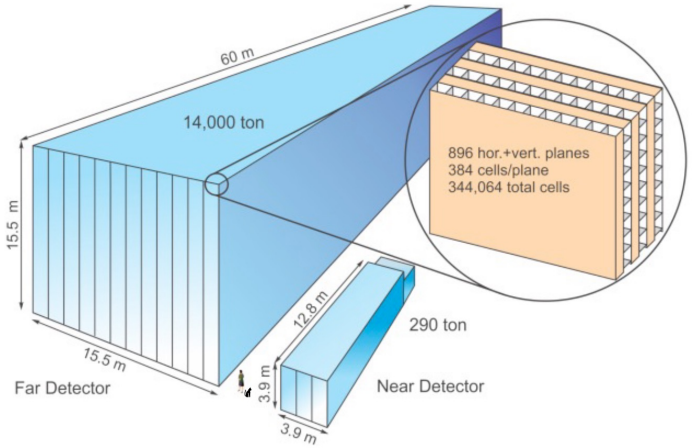
- Fermilab's accelerator complex produces the most intense (muon) neutrino beam in the world and sends it through the earth to northern Minnesota.
- Moving at close to the speed of light, the neutrinos make the 800-km journey in less than three milliseconds.
- When a neutrino interacts in the NOvA detector in Minnesota, it creates distinctive particle tracks.

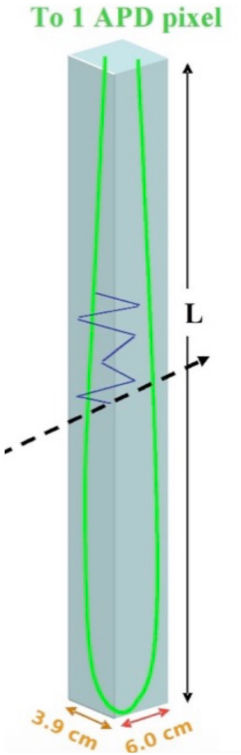


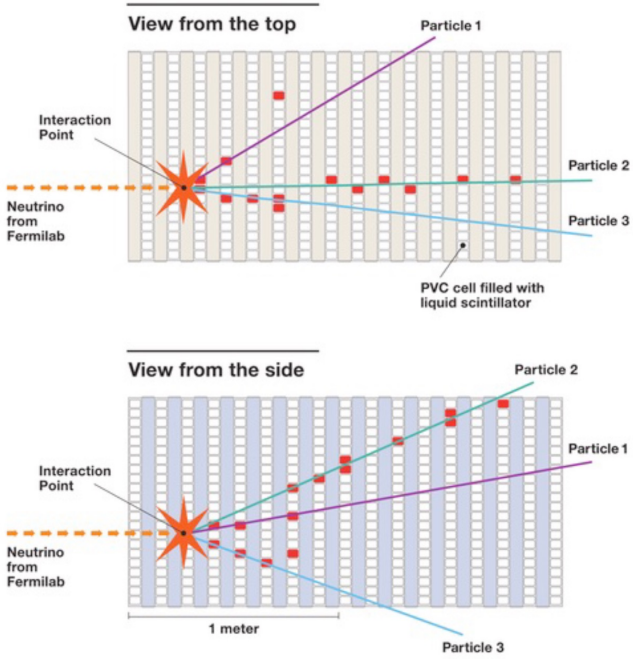
Credit: Maria Manrique Plata, NOvA in 10 minutes, New Perspectives 2022

Scintillator detectors

Detector Overview

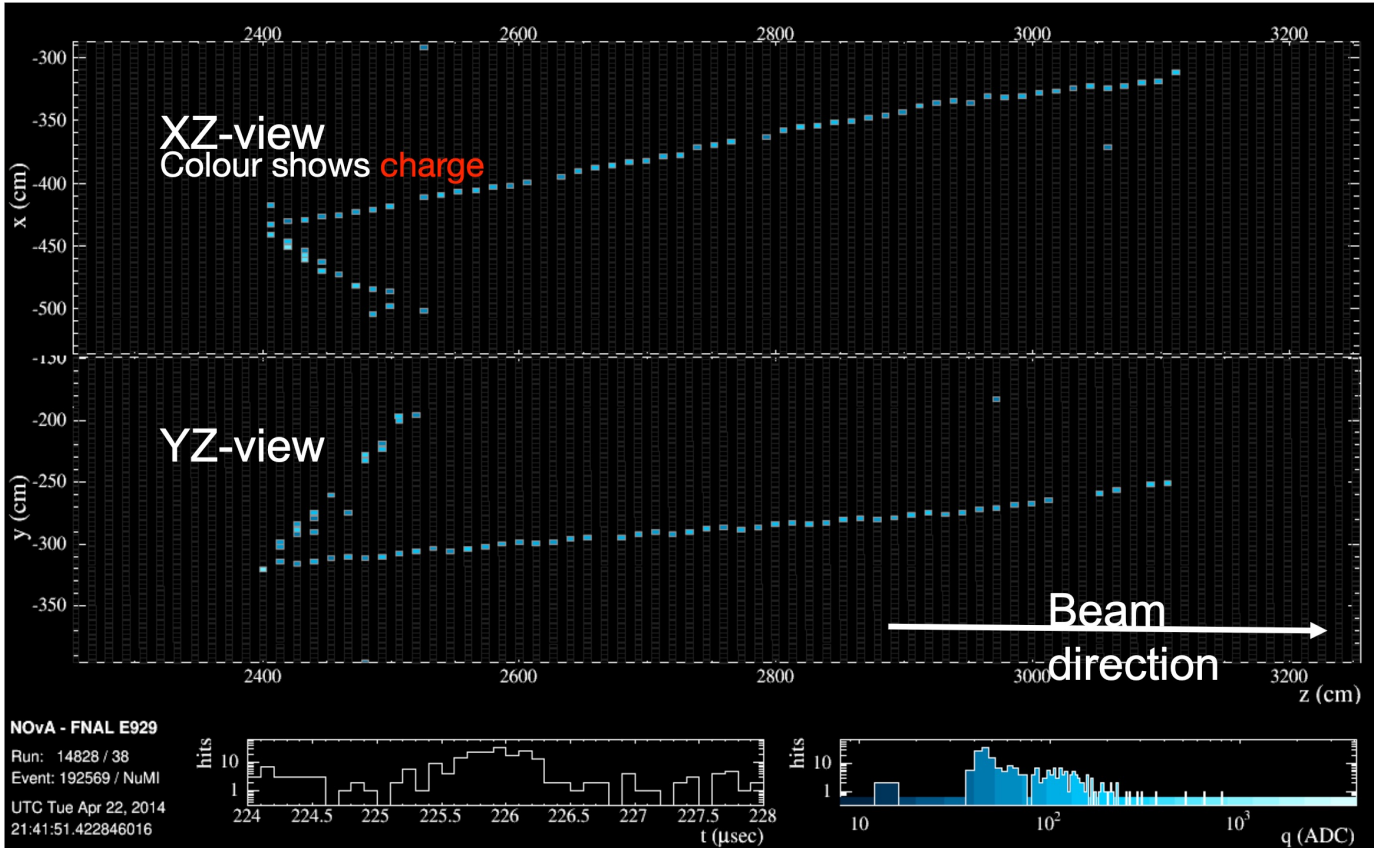




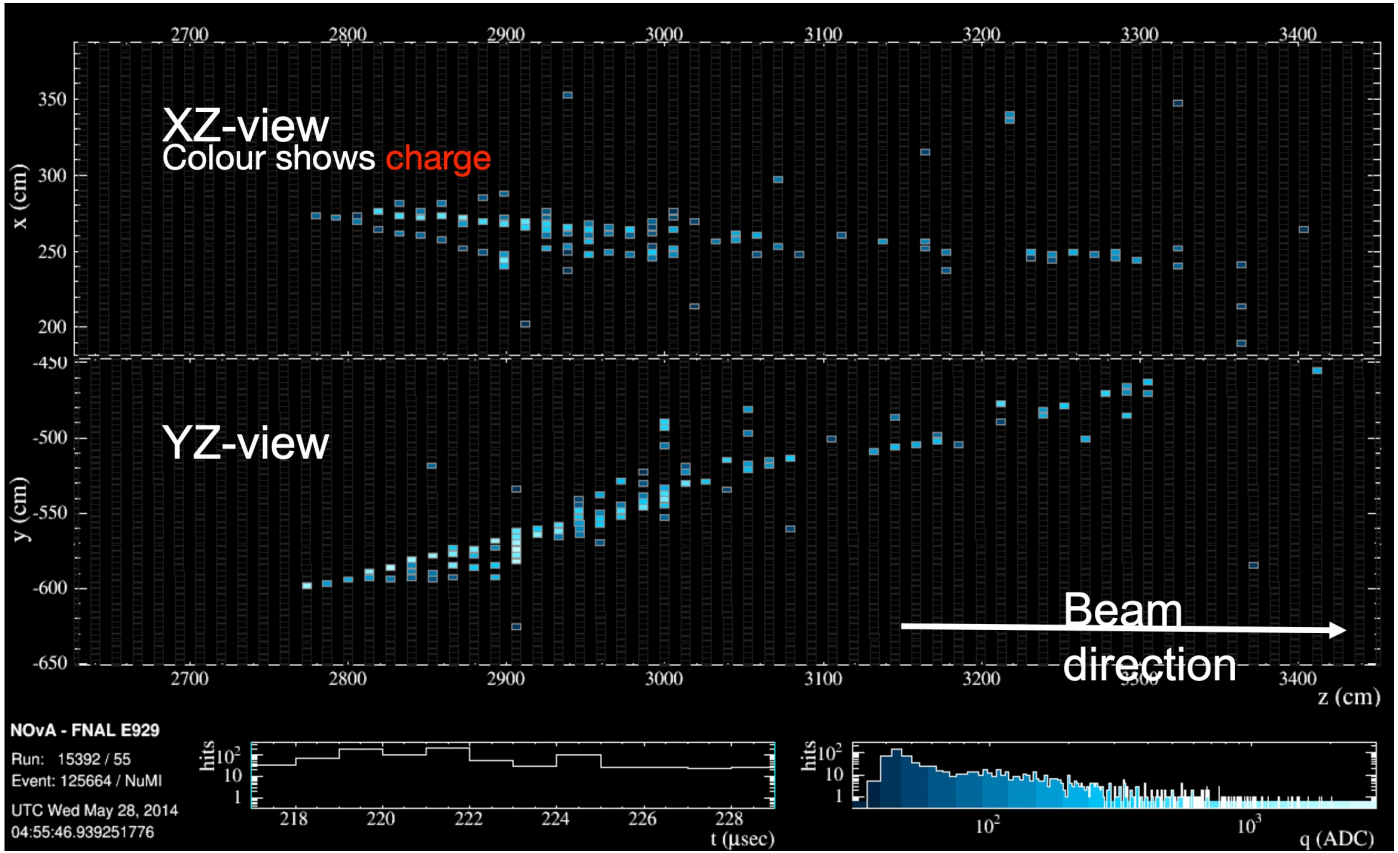


Credit: Maria Manrique Plata, NOvA in 10 minutes, New Perspectives 2022

Muon-neutrino Charged-Current Candidate



Electron-neutrino Charged-Current Candidate



Physics task at hand

- Classify types of interactions based on patterns found in the detector:
  - Is it a muon or electron neutrino?
  - Is it a charged current or a neutral current interaction?

- Classify a detector event by comparing its cell energy pattern to a library of 77M simulated events cell energy patterns, choosing 10K that are “most similar”.

- Compare the pattern of energy (hit) deposited in the cells of one event with the pattern in another event.

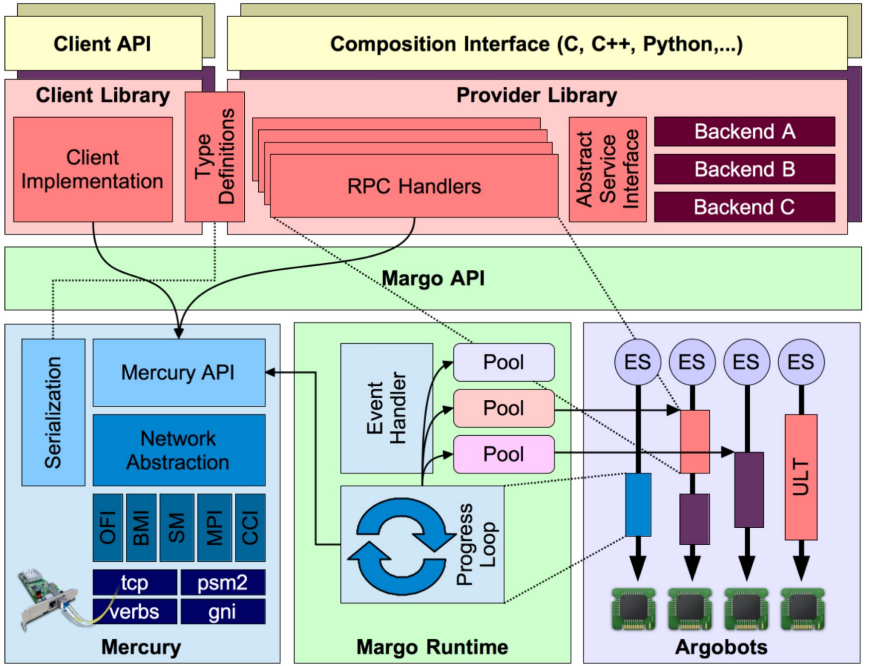
- The “most similar” metric is motivated by an electrostatic analogy: energy comparison for two systems of point charges laid on top of each other.

Goals: Harness HPC resources

- Present day analysis maps the work onto computer cores by assigning each core one ROOT file (which contains many events).
- This limits the maximum number of cores that can be used for analyzing a dataset.
- The goal is to **remove this bottleneck** and allow for faster processing of datasets by **harnessing HPC resources**.
- HPC clusters have nodes that are connected by **low latency, high bandwidth interconnects**.

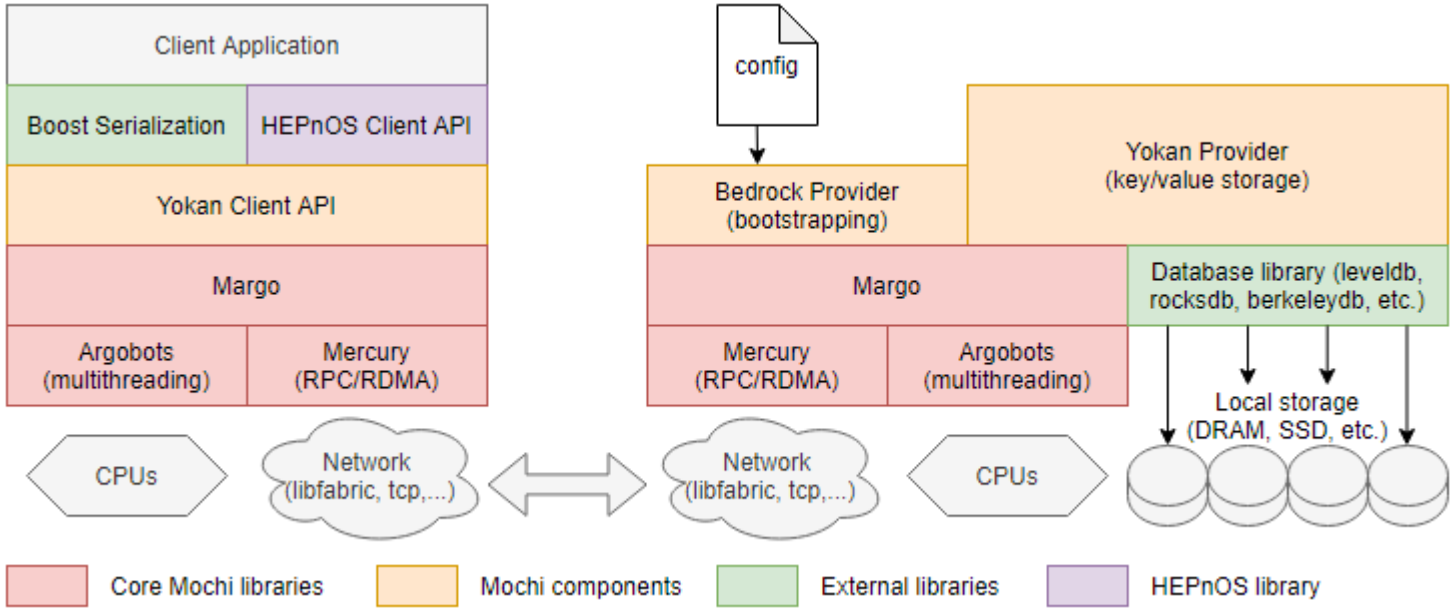
Background: Custom data services with Mochi

- Mochi microservices: a suite of re-usable components for building data services including:
  - *Mercury*: RPC framework that can use a variety of transports, which supports bulk data transfers.
  - *Argobots*: Lightweight user level threads to run tasks in execution streams.
  - *Margo*: Utilities for argobots aware mercury requests.



Anatomy of a data service backed by mochi microservices. Illustration by Matthieu Dorier. J Comput Sci Technol. 35, 121–144 (2020).

High-Energy Physics's new Object Store: Architecture

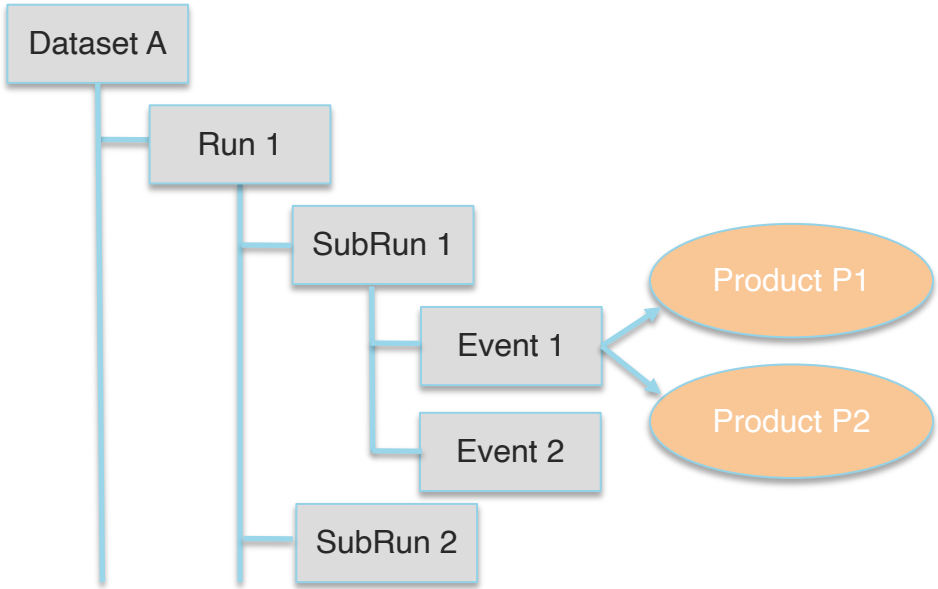


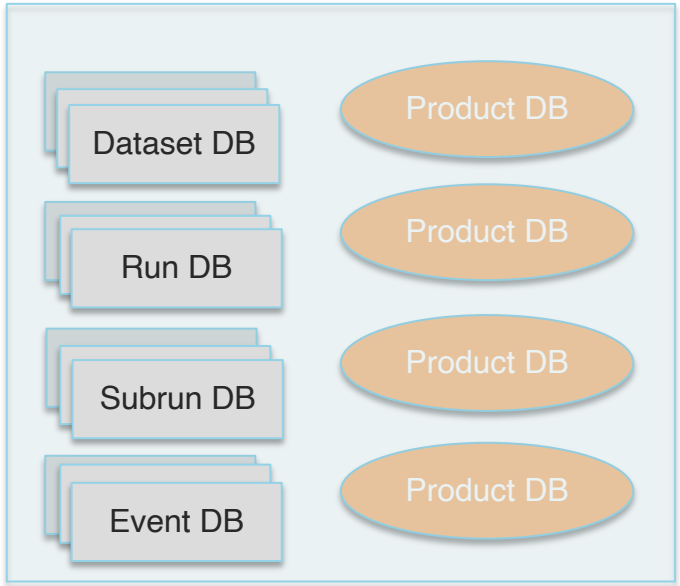
Architecture of HEPnOS: (Left) Client stack, (Right) Server stack. Illustration by Matthieu Dorier.

High-Energy Physics's new Object Store: Features

- Write-once, read-many access.
- Bulk ingest and iterative access.
- Eliminates software artifacts related to the filesystem and grid computing.
- Parallelism expressed at the event level instead of file level, allowing for better load balancing.

High-Energy Physics's new Object Store: Organization





Yokan providers in a HEPnOS server instance

Stored with lexicographic ordering

Stored without ordering

Background on dataset used

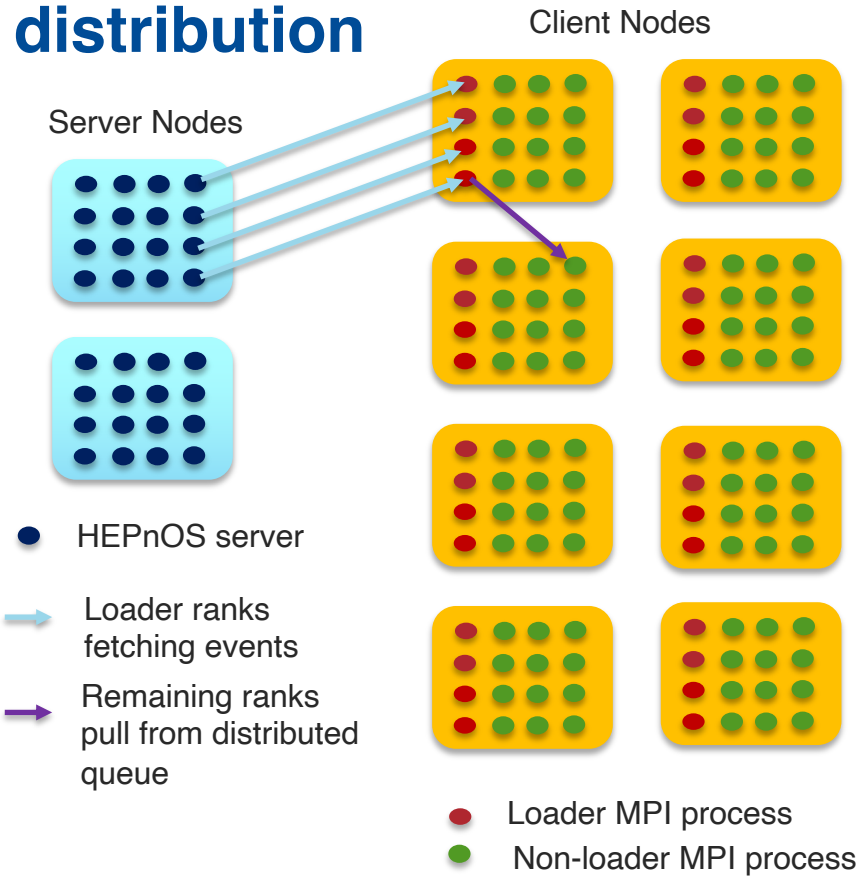
- Data collected from NOvA detectors, where each “spill” of the accelerator is called an “event”. These “events” are further split into “slices” associated with neutrino candidates.
- Performing neutrino candidate selection on these slices as a precursor to fitting model parameters.
- Input data is a collection of ROOT files using the “Ttree” format, also known as the “Common Analysis Format”.
- Using a dataset of 1929 ROOT files, that contain 4,359,414 events and 17,878,347 slices; size: ~0.2TB, representing ~1.1% of the total data (duplicated 4x for scaling studies).

New workflow with HEPnOS

- Set aside some of the compute nodes to run the HEPnOS Server.
- Load the data into the HEPnOS server.
- Call the processing function on “events” on the client nodes, with the HEPnOS Parallel Event Processor.
- Re-run the analysis as needed, without needing to reload data into the server!

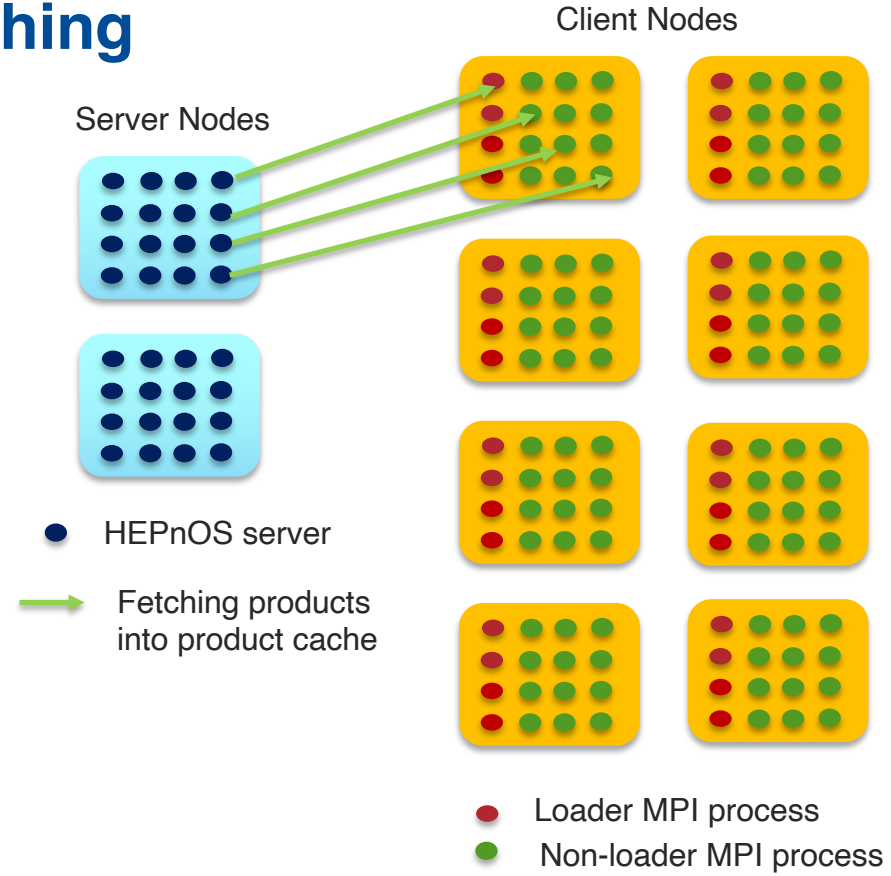
Parallel Event Processor: Task distribution

- Subset of the MPI ranks are designated as “loader” ranks.
- Loader ranks fetch the “events” from the datastore (in batches) and collectively provide a distributed queue.
- All MPI ranks fetch events (in batches) to process.
- This allows for implicit load-balancing at the event level.



Parallel Event Processor: Caching

- Can optionally indicate which product labels need to be pre-fetched into a “product cache” asynchronously.
- The user-function is then called on the events, with the “product cache” used to look up required products before searching for it on the servers.



Experimental Setup

- We tested the conventional file-based workflow with the Python multiprocessing module being used to map files to cores, with cores being idle at larger node counts.

- For the HEPnOS based workflow, we used two storage backends via the Yokan provider:
  - An in-memory backend using the C++ std::map
  - SSD backend utilizing the node-local SSDs via the RocksDB library

- Server configurations for the two backends:
  - In-memory backend: 1 MPI rank per node accessing 64 cores with HT disabled.
  - Node-local SSD backend: 16 MPI ranks per node, each accessing 4 cores with HT disabled.

- Quantity of interest: throughput, measured by number of slices processed per unit time.

Experimental Setup

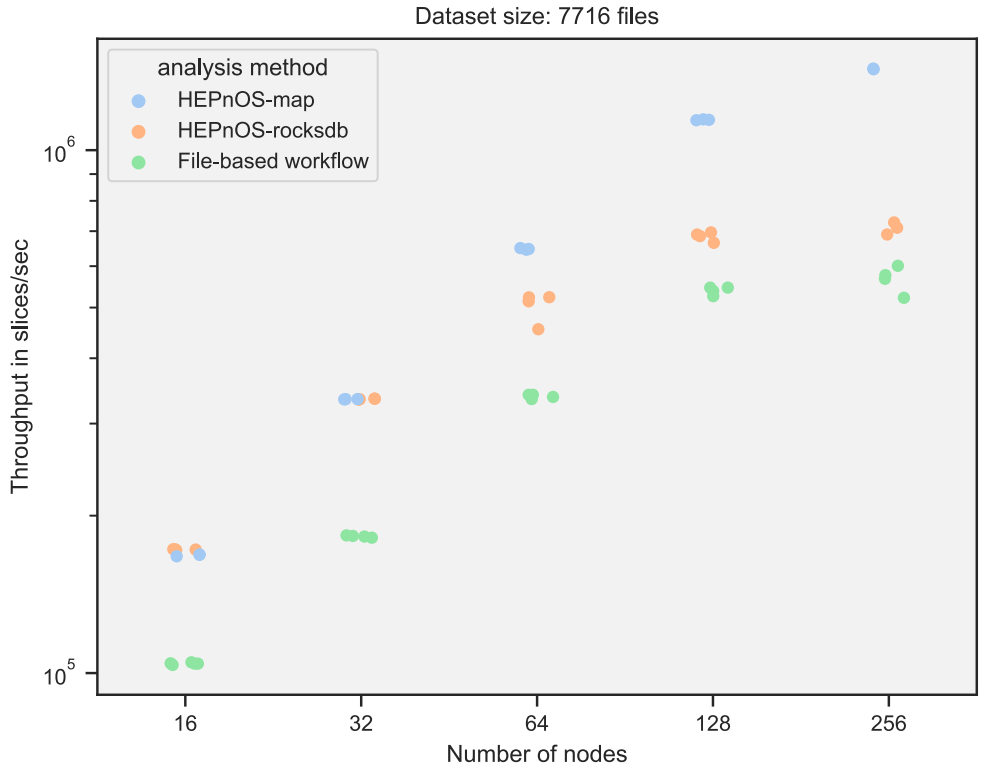
- We use the “theta” cluster at ALCF that consists of Intel Knights Landing nodes connected via a Cray Aries interconnect.

- The server software stack is installed using the *spack* package manager.

- The client stack comes from a SL6 contained image, into which we inject the networking libraries to enable the use of the GNI transport fabric.

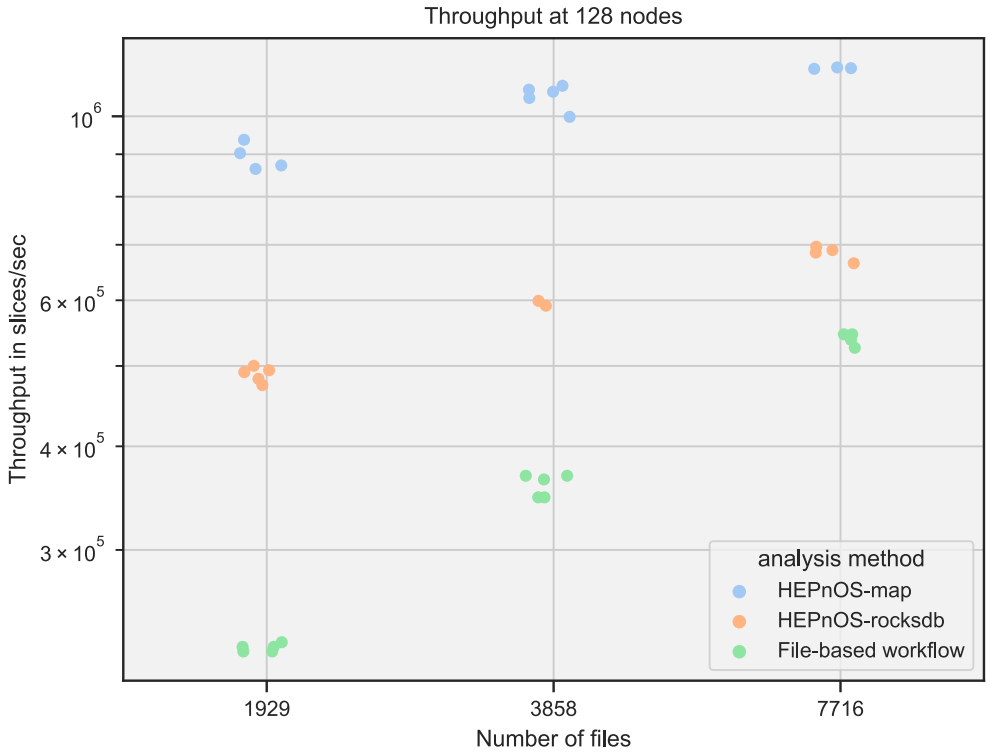
Throughput as a function of # nodes

- Performance of HEPnOS with either backend is better than the file-based workflow.
- In-memory backend of HEPnOS achieves ~85% scaling efficiency at 128 nodes.
- Typical data sizes for this workflow in production would allow for the usage of the in-memory backend.



Throughput as a function of # files

- File-based workflow is unable to harness all the available cores with 1929 files where only ~24% of cores are used.
- By using the HEPnOS-based workflow, we are better able to utilize compute resources.



Summary

- Benchmarked a novel HEP data store for the NOvA neutrino candidate selection workflow.
- Improved throughput for processing slices demonstrated at any number of nodes by **harnessing the interconnects on a HPC system**.
- Demonstrated the ability to **harness workflow parallelism at the event-level** with *HEPnOS*.

SciDAC team

- HEP and ASCR Collaboration
  - LHC and neutrino physics: N. Buchanan (CSU, NOvA/DUNE), P. Calafiura (LBNL, LHC-ATLAS), Z. Marshall (LBNL, LHC-ATLAS), S. Mrenna (FNAL, LHC-CMS), A. Norman (FNAL, NOvA/DUNE), A. Sousa (UC, NOvA/DUNE)
  - FASTMath Optimization: S. Leyffer (ANL), J. Mueller (LBNL)
  - RAPIDS Workflow, Data Modeling: T. Peterka (ANL), R. Ross (ANL)
  - Data science: M. Paterno (FNAL), H. Schulz (UC), S. Sehrish (FNAL)
  - J. Kowalkowski – PI (FNAL)
- Research Associates and Graduate students
  - Steven Calvez (CSU/PD), Pengfei Ding (FNAL), Matthieu Dorier (ANL/PD), Derek Doyle (CSU/GS), Xiangyang Ju (LBNL/PD), Mohan Krishnamoorthy (ANL/PD), Jacob Todd (UC/PD), Marianette Wospakrik (FNAL/PD), Orçun Yıldız (ANL/PD)
- http://computing.fnal.gov/hep-on-hpc/

Acknowledgement

- This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Scientific Discovery through Advanced Computing (SciDAC) program, grant 1013935.

- DoE report number: SLIDES-23-068-CSAID.