Evaluating a novel, HEP distributed data service using NOvA neutrino candidate selection

Sajid Ali *for* SciDAC4 HEPonHPC project
Postdoc / Scientific Computing Division / FNAL

New Perspectives 2022

In partnership with:

Science use case

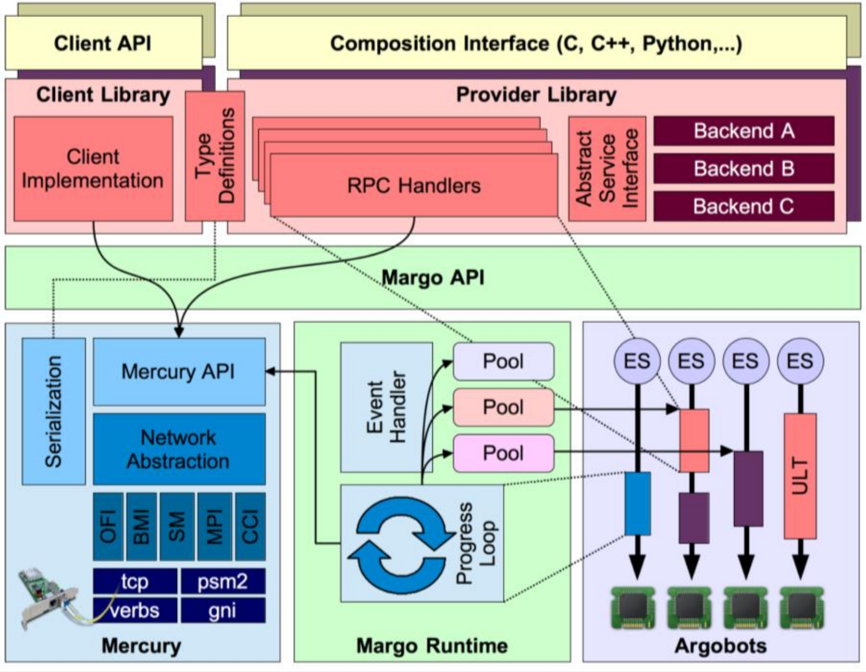
- Data collected from NOvA detectors, where each “spill” of the accelerator is called an “event”. These “events” are further split into “slices” associated with neutrino candidates.
- Performing neutrino candidate selection on these slices as a precursor to fitting model parameters.
- Input data is a 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.

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**.

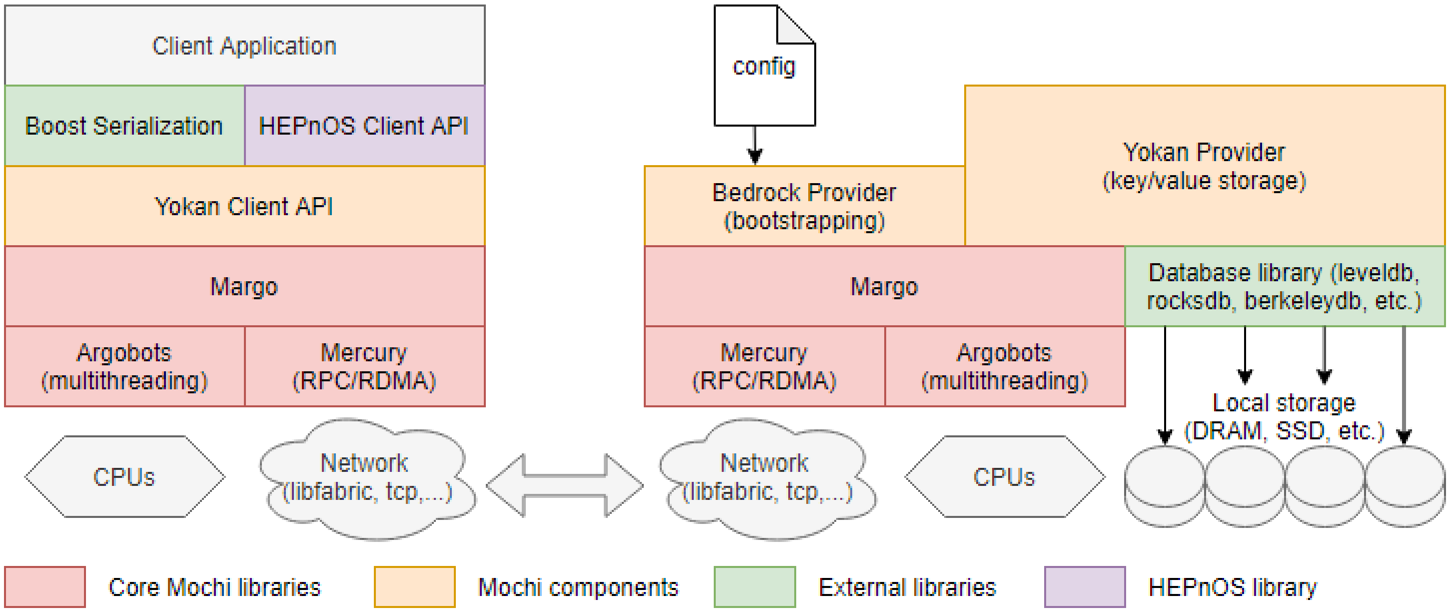
Background: Custom data services with Mochi

- 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.

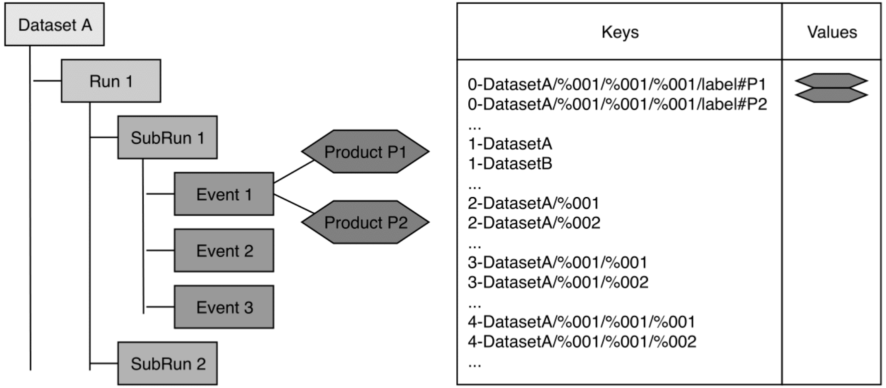
High-Energy Physics's new Object Store: Architecture



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.

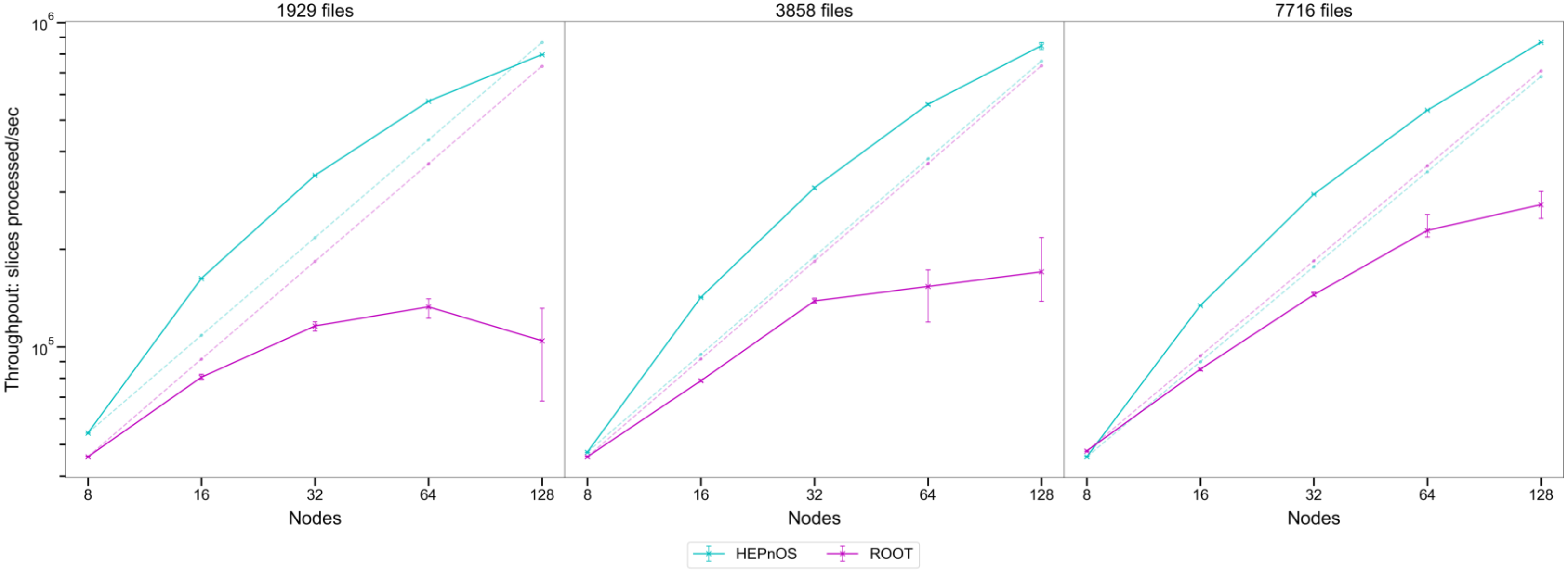


(Left) Hierarchical dataset organization. (Right) Representation in HEPnOS. Illustration by Matthieu Dorier.

New workflow with HEPnOS

- Set aside a small number of nodes to run the HEPnOS Server.
- Load the data into the HEPnOS server.
- Call the processing function on “events” on remaining nodes, HEPnOS takes care of fetching data products from server and passing them to appropriate compute cores.
- Re-run the analysis as needed, without needing to reload data into the server!

Results: Throughput for neutrino slice selection



Throughput of HEPnOS vs ROOT based slice processing. Experiments were run on Theta at ALCF.

Summary

- Demonstrated the use of a novel distributed object store.
- Using events as the basic processing unit instead of files leads to better scaling at larger nodes, as we are no longer limited by # of files.
- Improved throughput for processing slices demonstrated at any number of nodes.

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: FERMILAB-SLIDES-22-082-SCD.