CSAI Directorate Postdoc Meeting

Sajid Ali
Research Associate, PSD/DSSL Division
01/10/2023

Overview

➢ **Background**

❑ Accelerator modelling framework development
- Finite difference-based Poisson solvers
- Adopting standards based I/O frameworks
- Configuring CI to streamline development

❑ Novel data analysis frameworks for HEP data analytics on HPC platforms
- Evaluation of framework for neutrino candidate selection
- Adoption of framework for ICARUS workflow modules

❑ Future

Background

- PhD in Applied Physics at Northwestern Univ.; work conducted mostly at Advanced Photon Source and ALCF at Argonne Natl. Lab. Topics include:

  - *X-ray optics modelling and testing:* Software developed within the group, reaching the limits of single node workstations.
  - *Scalable algorithms for x-ray wave propagation*: Initiated a collaboration with the developers of PETSc to develop algorithms for wave propagation.
  - *High-throughput tomography solvers*: Extension of an internship at APS to refactor a first-pass C++ port of a MATLAB tomography algorithm before adding new capabilities.

- Contributed to software used including `PETSc` and `spack`.

- Worked part time at Research Computing Services at Northwestern on proof-of-concept deployment of `spack` installed software tailored to CPU architectures.

Choosing Fermilab

- Interested in Research Software Engineering as a profession, with primary focus on software infrastructure for research as opposed to tenure track research. Number of formal RSE is small, though on the rise.
- - Computational Physics Postdoc at Fermilab
  - 2-year NSF Fellowship at CU Boulder, based upon a research proposal jointly developed with the PI, designed to keep the pipeline for tenure track faculty open during the pandemic
- Chose the FNAL postdoc as the lab has better support for RSE-like careers.

Overview

- ❑ Background
- ➢ **Accelerator modelling framework development**
  - Finite difference-based Poisson solvers
  - Adopting standards based I/O frameworks
  - Configuring CI to streamline development
- ❑ Novel data analysis frameworks for HEP data analytics on HPC platforms
  - Evaluation of framework for neutrino candidate selection
  - Adoption of framework for ICARUS workflow modules
- ❑ Future

Accelerator modelling overview

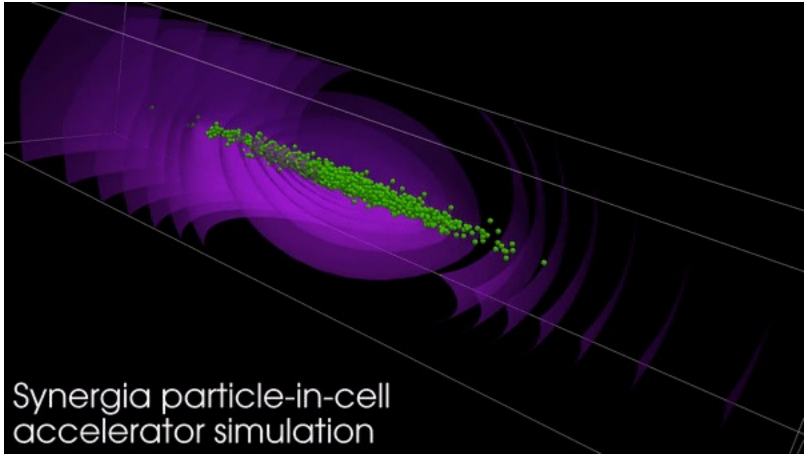
- Accelerators are large and complex devices that provide intense beams.
- Bunched beams of charged particles guided in vacuum pipes with magnetic fields.
- *- Aid in design process*
  - Investigate potential upgrades
  -Validate designs
- Played a crucial role in achieving 700KW+ beam power.
- Ongoing work simulating Booster/Recycler and Main Injector operations for PIP-II era.

Accelerator tunnel: recycler on top, main injector on bottom.

R. Ainsworth et.al. 7th Intl Particle Accel. Conf.; Phys. Rev. Accel. Beams 22, 020404; Phys. Rev. Accel. Beams 23, 121002

Synergia3

- Particle accelerator simulator using the Particle-In-Cell technique developed at Fermilab.
- Typically, we model between 1e10 to 1e11 charged particles by 1e5 to 1e7 macro-particles.
- A collection of macro-particles → bunch.
- A simulator can contain two trains, with each train containing multiple bunches.



https://web.fnal.gov/sites/synergia

J. Amundson et.al. Journal of Computational Physics 211.1 (2006)

Synergia development was developed through the SciDAC-4 ComPASS project funded by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research and Office of High Energy Physics, Scientific Discovery through Advanced Computing (SciDAC) program.

Synergia3

- A bunch is spread over multiple GPUs. Particles are stored as `SoA` (Struct of Arrays) on each `MPI` rank as `Kokkos` views on the GPU. Host mirrors are used when dumping their state during checkpoint hooks.
- Propagation of a beam through a lattice is a “turn”, involving:
  - Independent effects; applied to each particle
  - Collective effects; applied to the whole bunch

Space-charge effects

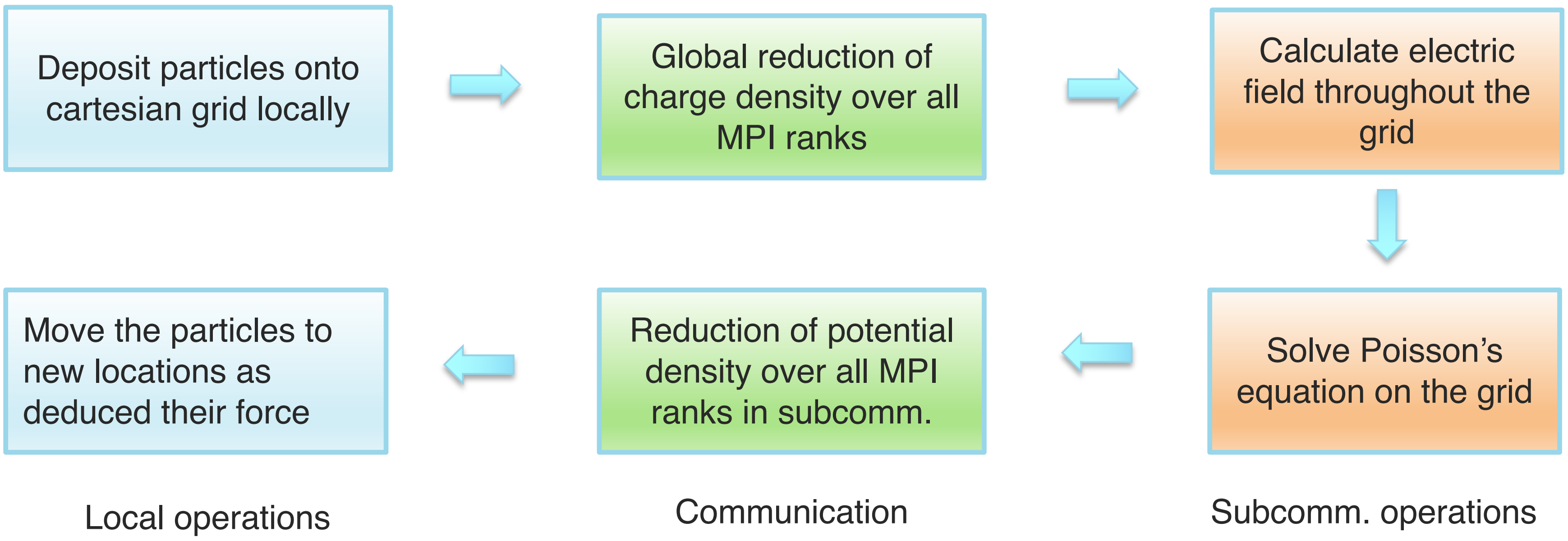
- Space-charge refers to electrostatic repulsion of charges; model via Poisson Eq.

$$\nabla^2\Phi = \frac{\rho}{\epsilon_0}$$ Φ : potential, ρ : charge density, ϵ_0 : constant

- Using all the compute resources available for solving the Poisson eq would introduce a communication overhead.
- Duplicate solve on subsets of compute resources (MPI sub-communicators) to avoid this overhead.

Communication avoiding concurrent solver

Poisson eq. is solved concurrently over multiple sub-communicators, each of which solves the same equation with the same charge density.



Communication avoiding concurrent solver using GPUs

- Space charge solver:

for `bunch` in `bunch-train` do:

all-ranks: deposit particles onto grid on GPU,
deep-copy local charge density to CPU,
`MPI_Allreduce` over all MPI ranks,
deep-copy particles to device;

on each sub-communicator: FFT based solve on GPUs;

all-ranks: deep-copy potential to CPU,
`MPI_Allreduce` over MPI subcomm ranks,
deep copy from CPU to GPU.

Need for a finite difference solver

- The current solver only works for regular boundary conditions.
- We wish to more accurately model the space-charge effects in irregular accelerator elements.
- As a first pass attempt, we have opted to use the finite-difference method for simplicity.

Implementation in Synergia3

- Using `PETSc`'s `DMDA` for (scalable) distributed memory cartesian grid data structures.
- Using Dirichlet boundary conditions to set the potential at the boundary to be 0.
- We plan to apply boundary conditions either by having only a single diagonal entry (= 1) for all points outside the accelerator element (which still gives the matrix with the same dimensions as the corresponding free-space one) or by using `PetscSection` and filling only a subset of the matrix.
- Moved communication routines to `PETSc-SF` to remove redundancy and possibility of using GPU specific communication backends like `NVSHMEM` in the future.
- Since the particles currently live on the GPU, we wish to solve the linear system there.

Current status

- This solver is available as part of synergia3!
- Accuracy tests show that it is on par with the existing rectangular solver.
- Early results show that it is slower on a booster lattice. Investigations to determine the reasons behind this are under way.

Adopting standards based I/O frameworks

- Multiple accelerator modelling frameworks, each with their unique strengths. However, it is challenging to build an end-to-end simulation by combining multiple frameworks.
- “The **openPMD** standard, short for *open standard for particle-mesh data files* is **not a file format** per se. It is a standard for **meta data and naming schemes**.”
- Ongoing work to adopt this to make Synergia3 simulations **interoperable** with other frameworks.

Axel Huebl “openPMD: A meta data standard for particle and mesh based data”

Configuring CI to streamline development

- Adopted GithubActions to build (`ubuntu-22.04`) container images with all dependencies using spack.
- Adopted CI pipelines for synergia3 for Linux/macOS and GCC/Clang using the generated containers.

CI:ubuntu-clang passing | CI:ubuntu-gcc passing | CI:macOS-clang passing | CI:macOS-gcc passing

- Assists in catching bugs early and gives confidence in status of primary git branch.

Overview

- ❑ Background
- ❑ Accelerator modelling framework development
  - Finite difference-based Poisson solvers
  - Adopting standards based I/O frameworks
  - Configuring CI to streamline development
- ➢ **Novel data analysis frameworks for HEP data analytics on HPC platforms**
  - Evaluation of framework for neutrino candidate selection
  - Adoption of framework for ICARUS workflow modules
- ❑ Future

Science use case

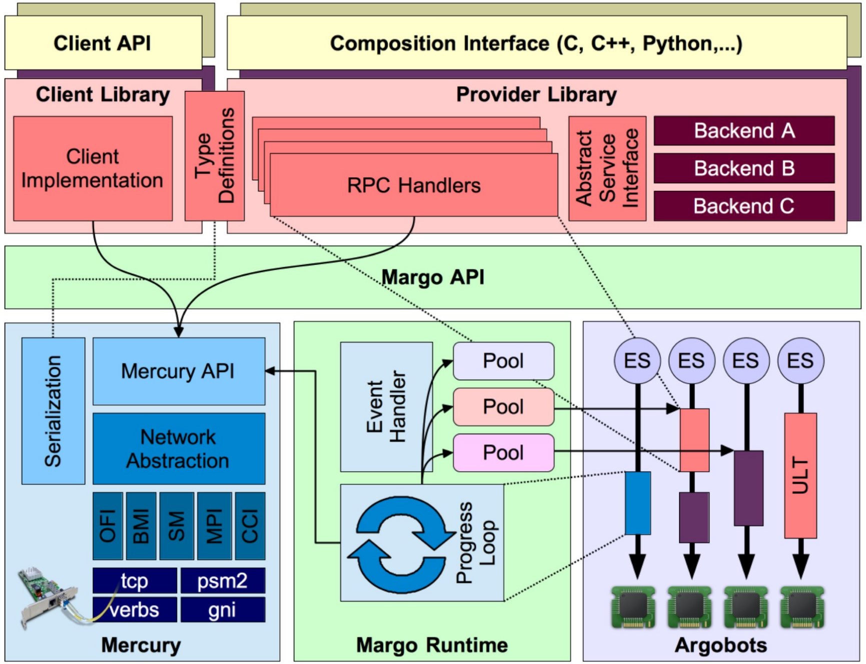
- Data collected from NOvA detectors, where each “spill” of the accelerator is called an “event”. These “events” are further split into “slices” associated with neutrino candidates.
- Performing neutrino candidate selection on these slices as a precursor to fitting model parameters.
- Input data is a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

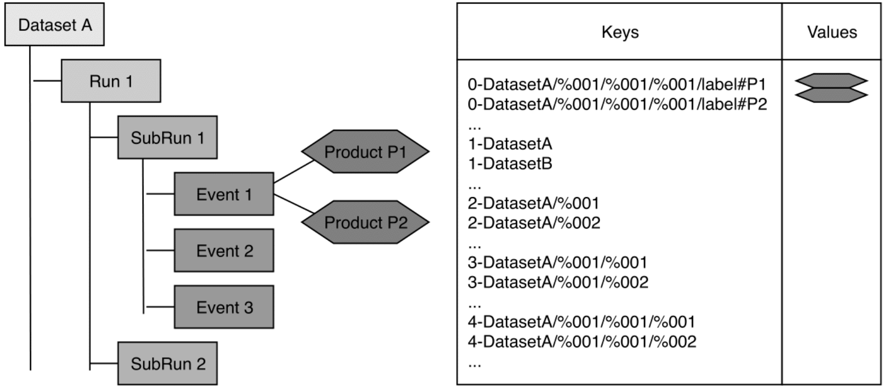


Anatomy of a data service backed by mochi microservices.
Illustration by Matthieu Dorier.

R Ross et al. Journal of Computer Science and Technology. 35, 121–144 (2020)

High-Energy Physics's new Object Store: Features

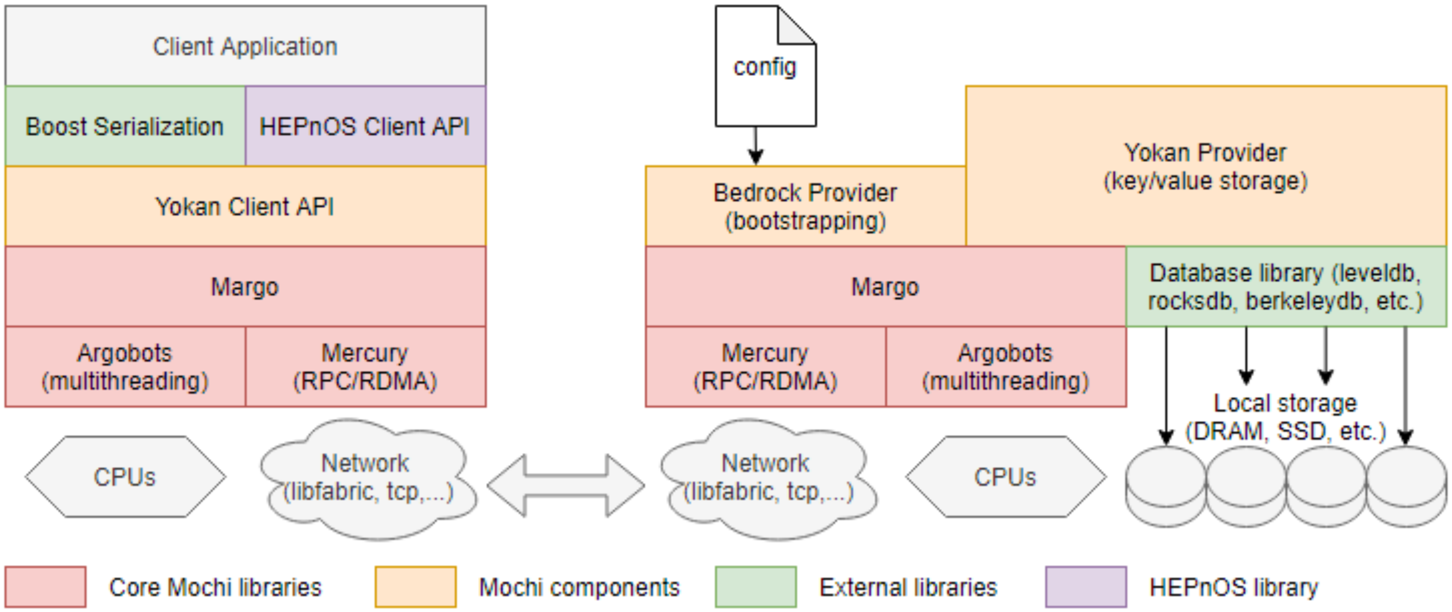
- Write-once, read-many access.
- Bulk ingest and iterative access.
- Eliminates software artifacts related to the filesystem and grid computing.
- Parallelism expressed at the event level instead of file level, allowing for better load balancing.



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

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.

High-Energy Physics's new Object Store: Architecture

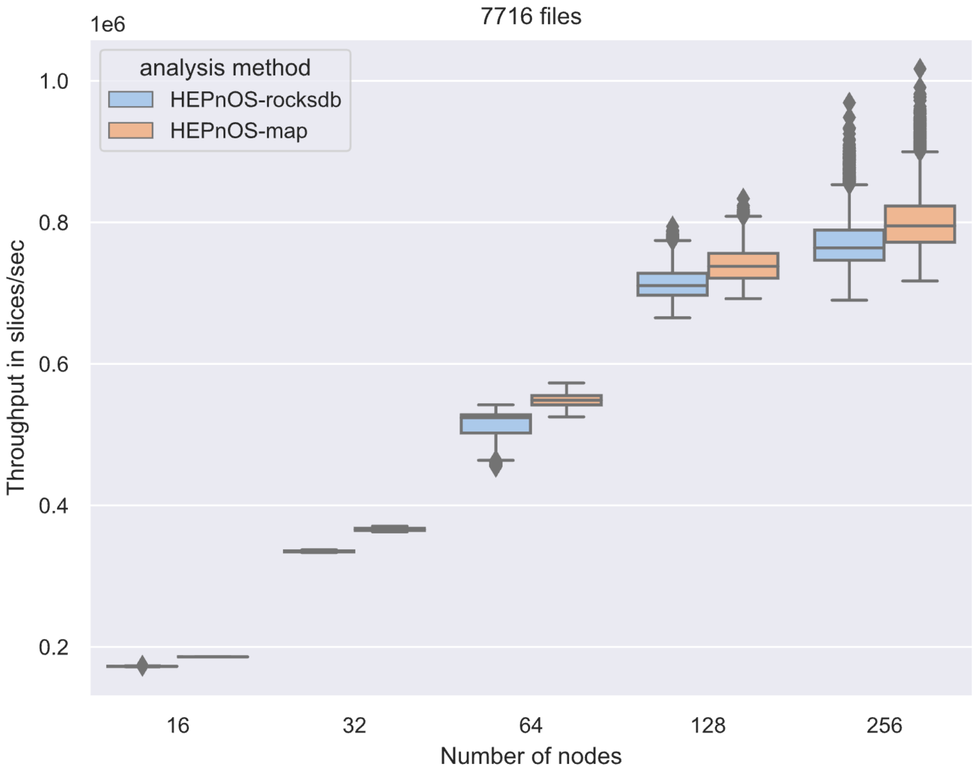


Architecture of `HEPnOS`: (Left) Client stack, (Right) Server stack. Illustration by Matthieu Dorier.

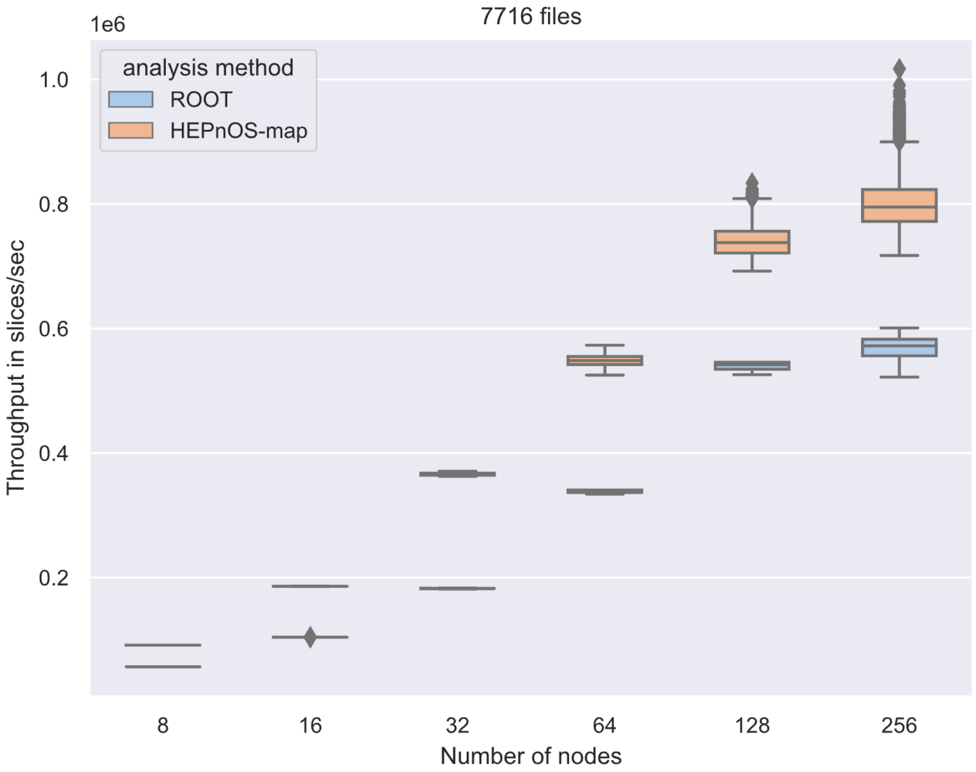
New workflow with `HEPnOS`

- 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!

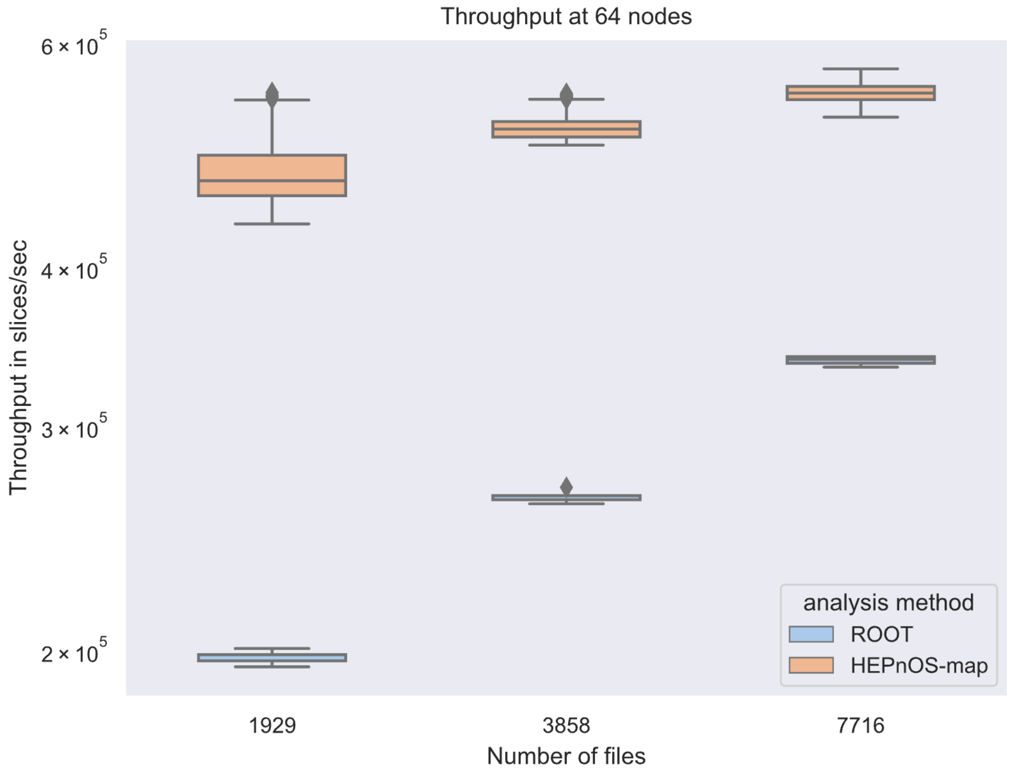
HEPnOS: SSD vs in-memory backend



HEPnOS vs ROOT, constant dataset size



HEPnOS vs ROOT, constant compute resources



Adoption of `HEPnOS` for ICARUS data processing

- Art modules leveraging `HEPnOS` I/O have been written.
- Intermediate data products are now written to `HEPnOS` instead of disk.
- Fixed bugs with incompatibilities in threading frameworks (`Argobots/Intel-TBB`).
- Proof-of-concept works on `csresearch` cluster, tests ongoing on ALCF theta.

Overview

- ❑ Background
- ❑ Accelerator modelling framework development
  - Finite difference-based Poisson solvers
  - Adopting standards based I/O frameworks
  - Configuring CI to streamline development
- ❑ **Novel data analysis frameworks for HEP data analytics on HPC platforms**
  - Evaluation of framework for neutrino candidate selection
  - Adoption of framework for ICARUS workflow modules
- ➢ Future

Future

- Interested in RSE-like roles involving:
  - development/maintenance/evaluation of software tools
  - consulting for use of HPC resources/programming models
  - training on research software development practices
- At Fermilab, “Computer Science Researcher” & “Application Developer” roles.
- At labs with computing facilities:
  - ALCF: Catalyst roles
  - NERSC/LBNL: HPC consultant roles

Acknowledgements

- Accelerator modelling team at FNAL (current): Eric Stern, Marc Paterno, James Amundson; (former) Qiming Lu, P. Lebrun, Q. Lu, A. Macridin, L. Michelotti, C. S. Park, T. Zolkin.

- 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)

- Work supported by the Fermi National Accelerator Laboratory, managed and operated by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy.