SciDAC Projects

Marc Paterno
SCD Projects Meeting
27 August 2020

HEP Data Analytics on HPC: Interactions with HEP Community

SBND

– Feldman-Cousins algorithm rewritten with vectorization & MPI, Increasing performance more than 300x.

NOvA

– Extensive work on the Feldman-Cousins analysis on HPC architectures. Adopted as primary technique for oscillation results.
– Tools for producing HDF5 analysis formats for physics applications as alternative data format for use in HPC facilities and ML tools.
– Developed PandAna framework for highly scalable analysis.

ICARUS

– Parallel production workflow defined for event processing using *art* & LArSoft frameworks coupled to HEPnOS and DIY, designed for HPC systems

ATLAS analysis

– Enabled H->μμ analysis using HPC systems and our scalable software technology to generate billions of background events

CMS & ATLAS

– Working to integrate Pythia HDF5 tools into CMS infrastructure

DUNE

– Adopted Feldman-Cousins software based on NOvA work

SciDAC: HPC Framework for Event Generation at Colliders

– Developed new event generator framework for unprecedented accuracy, tying Sherpa and Pythia8 together by means of HDF5 and DIY
– Explored application of Machine Learning for Monte-Carlo simulations with Sherpa.

HEP Event Reconstruction with Cutting Edge Computing Architectures

– Partner project in the design and use of the ICARUS workflow.

Theory and Phenomenology

– Leverage closed form solutions to neutrino oscillation problem from FNAL and BNL theory to accelerate computation
– Assisted with Dark Matter Direct detection phenomenology studies
– Helped conduct reinterpretation of ATLAS data to constrain new physics models
– Enabled early-universe research through technical assistance with complex differential equation solving

FASERnu

– Our expertise in generator tuning helped in the successful proposal of the "Fasernu“ experiment to be built at CERN shortly

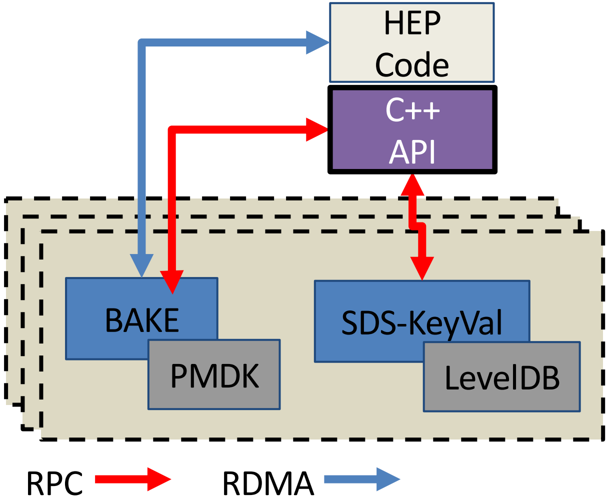
HEPnOS: Fast Event-Store for High-Energy Physics (HEP)

Goals

- Manage physics event data from simulation and experiment through multiple phases of analysis
- Accelerate access by retaining data in the system throughout analysis process
- Reuse components from Mochi ASCR R&D project

Properties

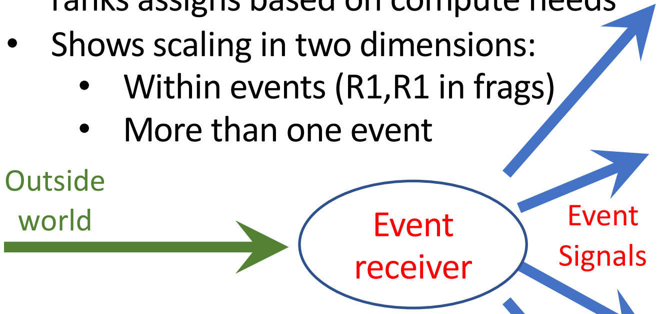
- Read in data at start of run and write results to persistent storage at the end of a campaign
- Hierarchical namespace matching physics concepts (datasets, runs, subruns)
- C++ API (serialization of C++ objects)
- Write-once, read-many

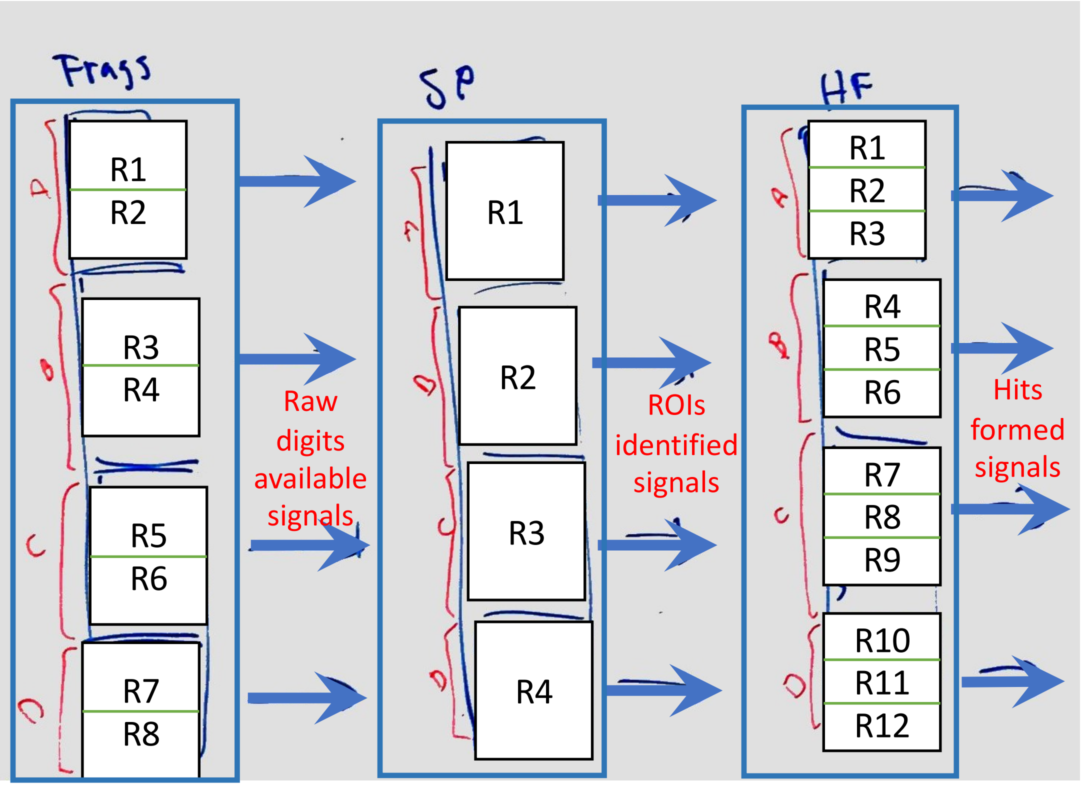


The HEPnOS event store is built using Mochi, a framework for developing specialized data services for use in HPC. Mochi allows use of state of the art libraries (e.g., LevelDB for key/value storage) while providing convenient APIs to scientists.

ICARUS distributed workflow – work in progress (HEPonHPC/HEPreco)

- View as one MPI program
- 3 algorithms: Frags, SP, HF
- 4 channels for events to flow: A,B,C,D
- Each channel has different number of ranks assigns based on compute needs
- Shows scaling in two dimensions:
  - Within events (R1,R1 in frags)
  - More than one event





- Frags has 2 ranks per event
- SP has 1 rank per event
- HF has 3 ranks per event
- A signal means algorithm is done for that event (e.g. R1,R2 have finished)

HEPnOS status

- Getting consistent software builds is a nightmare
  - Some runs native on machine
    - Uses GCC 9.0
  - Other runs in container
    - Some in the container built with GCC 7.3
    - NOvA code had to be built with GCC 6.3
- But we have things running
  - We have demonstrated scaling up to ~100 nodes (~10,000 cores) on KNLs.
- Working with ANL colleagues on fixes to deal with latest upgrade of system tools on Theta (ALCF).

PandAna: An environment for scalable high-level HEP analysis on HPC

Achievements

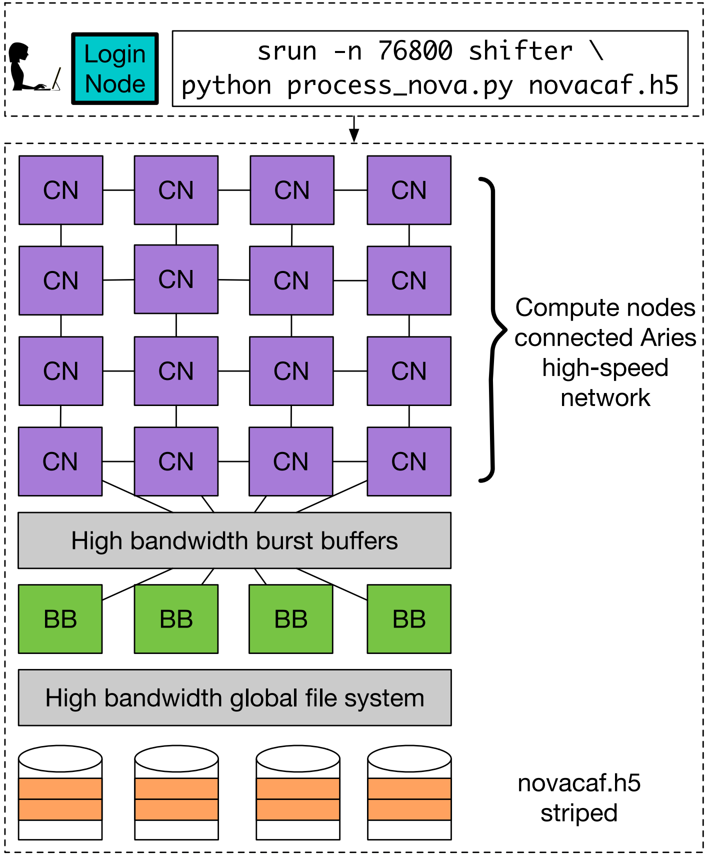
Demonstration of scalable parallelization of an analysis code from NOvA by replacing serial IO mechanism with parallel IO.

Significance and Impact

Allows existing analysis code developed by experimenters to be deployed at HPC sites for processing of large datasets.

Research Details

- Provide an easy-to-use environment for fast and scalable HEP high-level data analysis
  - Users can develop on laptops or local clusters and deploy code to HPC
- **Use HDF5** for fast parallel reading of large amounts of data
- **Use Python** and popular Python data science tools (numpy, pandas)
- **Introducing to HEP the "tidy data" analysis model**, using large data matrices and distributed data parallelism
  - Use MPI for distributed parallelism
  - The parallelism in user code is implicit



PandAna status

- **We have demonstrated scaling (on Cori) up to 64 nodes.**
  - Perfect scaling of in-memory computation, as expected
  - Terrible scaling of reading, but it is fast anyway; much faster than processing (unless we use many ranks).
- **We have trouble getting datasets large enough for performance demonstrations.**
  - We have ~150 GB size data; would like to have several terabytes of data.
  - What we have is suitable for ~1000 MPI ranks, not a large job
- **We have a paper started, but little time to work on it.**
- **We can use more effort on this project (and there is money to support it).**

Accelerating SBNFit Fitting Framework on HPC

Scientific Achievement

Accelerating SBN application and algorithms that calculate Feldman-Cousins corrections used in the sterile-neutrino search to run efficiently on HPC facilities

Significance and Impact

Transforms a memory-limited serial-execution program into an MPI-parallel application that scales up to available compute power of a facility. Factor of 350 single-core speed up from original implementation

Research Details

- Native built of SBNFit application at NERSC to take advantage of specific acceleration hardware
- Algorithms leverage vectorization through Eigen3 yielding a factor of 350 single-core speed up compared to the original implementation
- MPI node-level and thread-level parallelism through DIY and HDF5 using input data and covariance matrix from SBN.
- HDF5 in conjunction with HighFive to write results of the calculation to file.