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

Parallel Event Selection on HPC Systems

H. Schulz

CHEP Frameworks Mini-workshop 2018

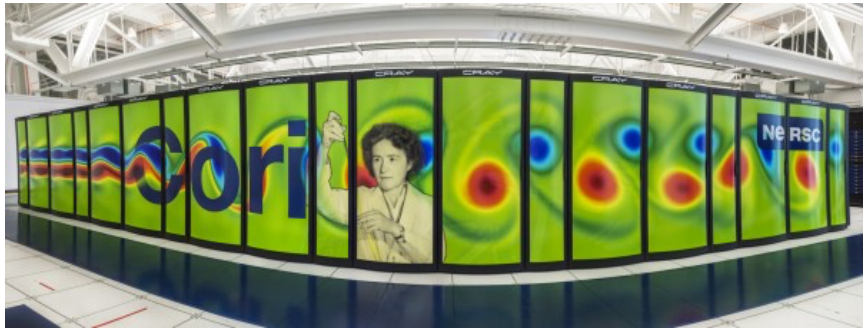
9 July 2018

In partnership with:

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

Toward a distributed object model

- Current and planned HPC machines have large memories and deep memory hierarchies, as well as very fast internal networks, e.g. *Cori*:
  - L1: 64 kB per core
  - L2: 1 MB per tile (2 cores)
  - "L3": 16 GB (MCDRAM, > 460 GB/s bandwidth) per node
  - DRAM: 96 GB (102 GB/s bandwidth) per node; total memory > 1 PB
  - *Burst Buffer*: 1.7 TB/s bandwidth, 288 nodes, 1.8 PB storage
  - Global filesystem: 700 GB/s bandwidth, 29 PB storage
- Future machines will introduce new layers, new technologies (e.g. 3D XPoint)
- We are exploring the design of an object model that leverages these machine features.



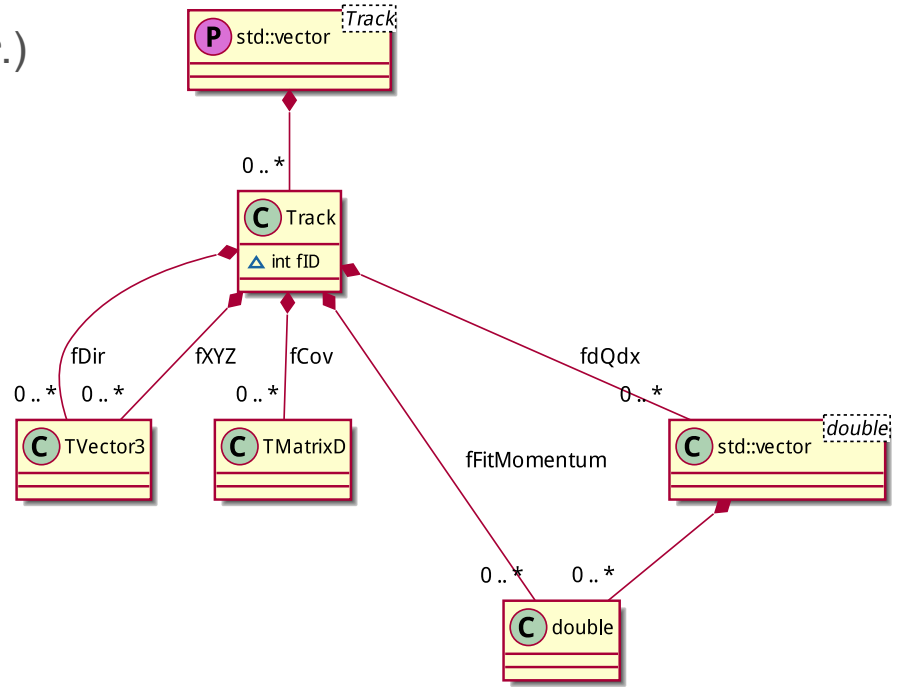
A prototype model user interface

- A common design is to to have levels of data aggregation:
  - Data set / run / subrun / event
  - Blobs of data (*data products*) can be associated with any of these levels
  - At each level of aggregation, the named items are independent, e.g.
    - each event is independent of each other
    - each run is independent of each other run
- Data sets can contain other data sets
  - arbitrary nesting for geometry, alignment, calibrations, …
- Independent items can be processed without need of synchronization
  - data parallel programming, on various scales
  - implicit parallelism
- Distribution of data automated by the data model implementation, not user-visible

Experiment context

- We are using some of the data product classes from LArSoft (used by DUNE, *etc.*)
- We have started with some of the most widely-used classes:
  - `std::vector<recob::Hit>`
  - `std::vector<recob::Track>`
  - `art::Assns<recob::Hit, recob::Track>`,

    which are bidirectional associations between hits and tracks.



Software system context

- We are using existing *art* framework-related software:
  - *gallery*, which provides access to event data in *art*/ROOT files from outside the *art* framework.
  - pre-existing builds of the LArSoft data product libraries
- We are using Docker containers for easy portability of development environment
  - Some of us develop on macOS laptops, others on a variety of Linux installations
  - We will deploy to NERSC (through Shifter) and ALCF (through Singularity)
- Prototype test programs are already running:
  - one program to read from existing *art*/ROOT data files, and to write to the new data store
  - one program to read from the new data store, and verify the integrity of the data

Example of HEPnOS use: reading from the DataStore

```
hepnos::DataStore datastore(configfile);
for (auto it = datastore.begin(); it != datastore.end(); ++it) {
  for (auto const& ds : *it) {
    for (auto const& r : ds.runs()) {
      for (auto const& sr : r) {
        for (auto const& e : sr) {
            std::vector<recob::Hit> hits;
            art::InputTag hit_tag("gaushit", "", "PrimaryReco");
            e.load(hit_tag, hits);
            // .. do something with the vector of Hits ..
        }
      }
    }
  }
}
```

The user interface looks a lot like our current user interface

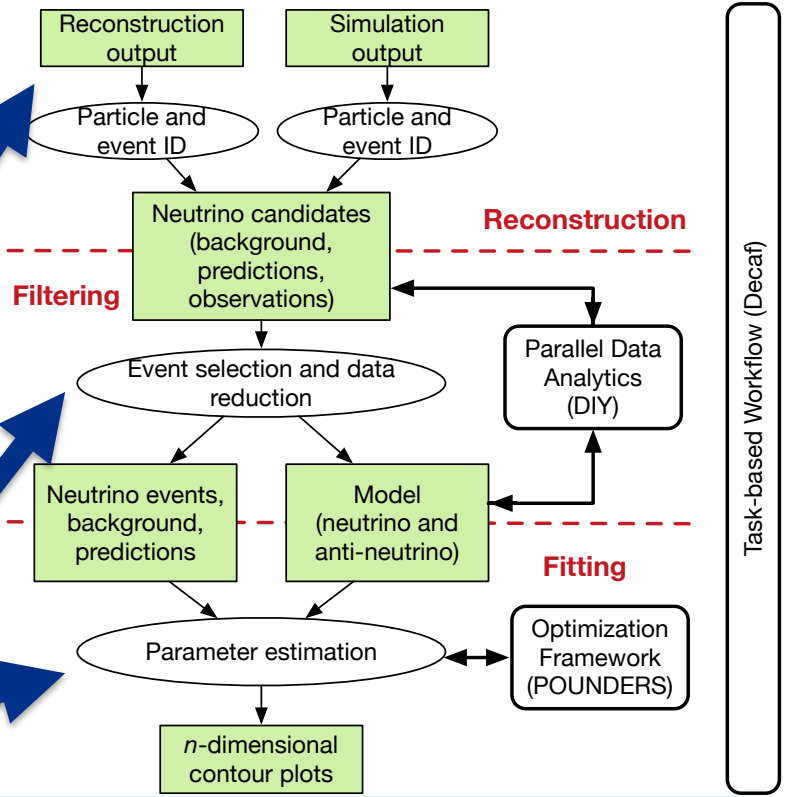
Behind the scenes

- Serialization is done under the covers
  - currently using Boost serialization
  - can be swapped to whatever is most performant (ROOT? Protocol buffers? Others …)
  - For array-oriented data, no serialization is even needed
- Communication uses cutting-edge tools developed for HPC applications
  - e.g. Mercury https://mercury-hpc.github.io
- We are exploring how to put support for MPI parallelism directly into the infrastructure.
  - Will enable our reconstruction/analysis codes to take advantage of data parallelism and distributed programming without looking much different than it does today.

SciDAC-4: HEP Data Analytics on HPC

- Goal: Enable HPC facilities to meet the future HEP data analysis demands.
- Initial targets:
  - NOvA neutrino oscillation measurement fitting procedures and
  - event generator tuning with Pythia, Rivet, and Professor
- Collaboration between DOE Office of High Energy Physics and Advanced Scientific Computing Research (ASCR supports the major US supercomputing facilities)
  - 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)
  - Optimization: S. Leyffer (ANL), J. Mueller (LBNL)
  - Workflow, Data Modeling: M. Paterno (FNAL), T. Peterka (ANL), R. Ross (ANL), S. Sehrish (FNAL)
- J. Kowalkowski – PI (FNAL)

http://computing.fnal.gov/hep-on-hpc/

Neutrino analysis workflow

- Advancements using HPC leadership facilities
  - Analysis using full dataset across all layers, managed using tools and techniques developed by ASCR's **RAPIDS** institute
  - Expand fitting into 4D, 5D, including anti-neutrino data, utilizing multi-dimensional fitting procedures from ASCR's **FastMATH** institute
  - Include all layers in the analysis
- Initial tasks
  - Basic data models for NOvA and LArSoft and datasets mapped into HPC NVRAM-based hierarchical storage systems using ASCR services and tools through the **Mochi** project
  - Demonstration of fast event selection using **DIY**
  - Full automation of Feldman-Cousins analysis prescription with **Decaf**