Parallel Event Selection on HPC Systems

H. Schulz *for* J. Kowalkowski, M. Paterno & S. Sehrish

CHEP 2018

9 July 2018

In partnership with:

Science problem

- Measurement of the neutrino oscillation parameters by the NOvA collaboration.
  - PRL 118, 231801 (2017).
  - Sample of ~27 million reconstructed spills to search for electron-neutrino appearance events.
  - Events are stored in a ROOT n-tuple format
  - ~180 thousand ROOT files.
  - File sizes range from a few hundred KiB to a few MiB; the full dataset is approximately 3 TiB.
- These millions of events were reduced to a few tens of events by the application of strict event selection criteria, and then summarized by a handful of numbers each, which are used in the extraction of the neutrino oscillation parameters.

Computing problem

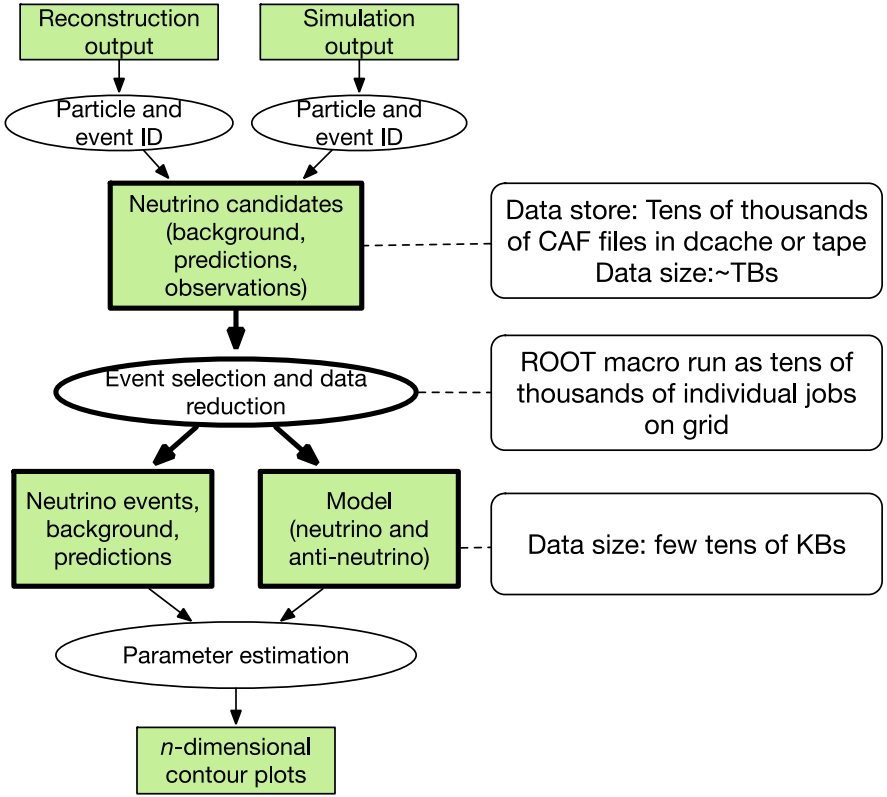
- We want to greatly reduce the time it takes to process analysis-level data.
- Most of the computing resources that will be available to us will be at HPC centers.
- We need to modify our workflows and code to take full advantage of HPC systems.
  - to take advantage of well-established parallel programming tools and techniques
  - to make sure these tools and techniques are sufficiently easy to use

SciDAC-4: HEP Data Analytics on HPC

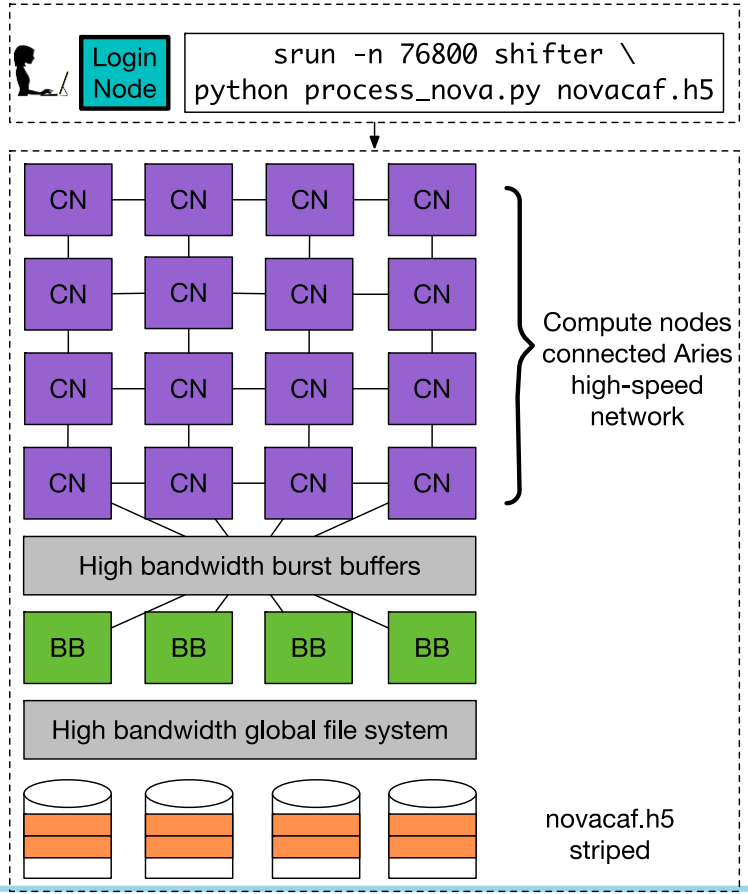
- Goal: Enable HPC facilities to meet the future HEP data analysis demands.
- Initial targets:
  - NOvA neutrino oscillation measurement fitting procedures and
  - event generator tuning with Pythia, Rivet, and Professor
- Collaboration between DOE Office of High Energy Physics and Advanced Scientific 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/

Science context and traditional solution



HPC solution



High-level organization of processing

- We want to minimize *reading*
- We want to minimize *communication* and *synchronization* between processes
  - in MPI programs, a process is called a *rank*.
  - obtain the most data parallelism possible
  - parallelism is *implicit*; user-written code looks just like serial code
- We want to process all data for a given *slice* in a single rank.
  - the *slice* is NOvA's "atomic" unit of processing, like a collider *event*.
  - for data that represent per-slice information, this is trivial
  - for other data, we need to do some work to ensure each rank has the correct data.
- We organize the data into a single HDF5 file, containing many different tables
  - some tables have one entry per slice
  - some have a variable number of entries per slice

Sample table: *sel_nuecosrej*

run	subRun	event	slice	distallpngtop	...35 more…
16433	61	356124	35	nan	
16433	61	356124	36	-0.74013746	
16433	61	356124	37	nan	
16433	61	356125	1	nan	
16433	61	356125	2	423.6337	
16433	61	356125	3	-2.849864	

- This table has one entry per *slice*.
- Remember we are doing *slice-by-slice* selection.

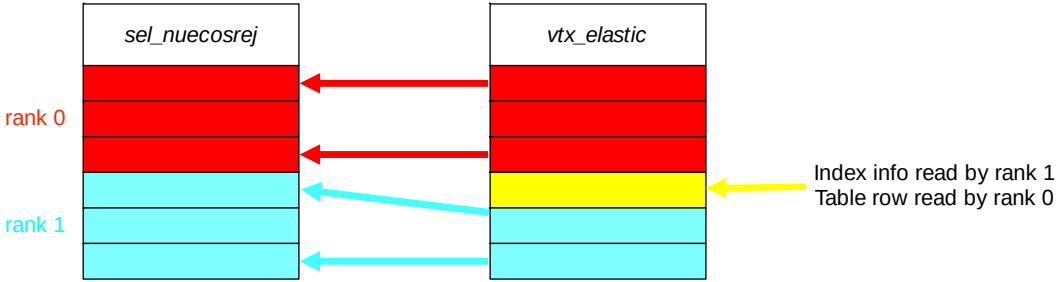
Sample table: *vtx_elastic*

run	subRun	event	slice	vtxid	npng3d	…6 more…
16433	61	356124	35	0	0	
16433	61	356124	36	0	1	
16433	61	356124	36	1	1	
16433	61	356124	36	2	5	
16433	61	356125	1	0	1	
16433	61	356125	3	0	0	

- This table has one entry per *vertex.*
- Some slices have none (and do not appear in this table).
- Some slices have many.

Distributing and reading information

- Each rank reads its “fair share” of index info from each table.
  - identifies which rank should handle which event, for most even balance
  - identifies range of rows in table that correspond to each event (all slices)



- Event “ownership” information distributed to all ranks
  - this assures no further communication between ranks is needed while evaluating the selection criteria on a slice-by-slice basis.
  - perfect data parallelism in running all selection code
- Each rank reads *only* relevant rows of relevant columns from relevant tables
  - all relevant data read by some rank
  - no rank reads the same data as another

Example selection code

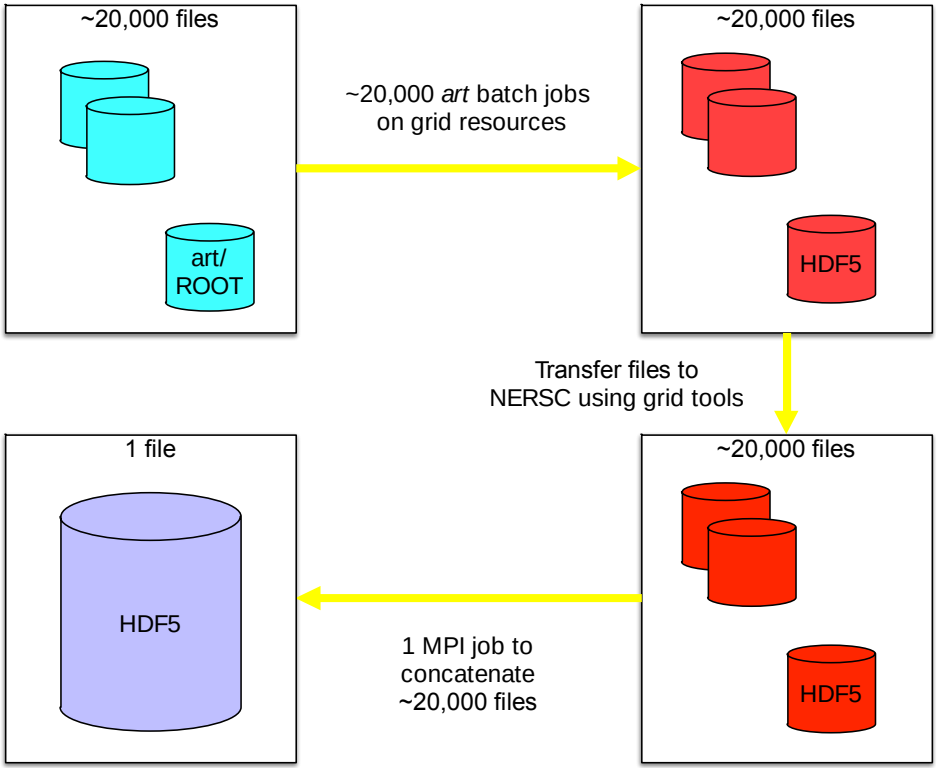
```
def kNueSecondAnaContainment(tables):
  df = tables['sel_nuecosrej']
  return (df.distallpngtop > 63.0) & \
      (df.distallpngbottom > 12.0) & \
      (df.distallpngeast > 12.0) & \
      (df.distallpngwest > 12.0) & \
      (df.distallpngfront > 18.0) & \
      (df.distallpngback > 18.0)
```

- Selection can be done on multiple columns of a table.
- Logical operations are connected by & operator.
- Data parallelism is totally *implicit.*
- Returns an array with one logical value per *slice*.

```
def vtxelasticzCut(tables):
  df = tables['vtx_elastic']
  df['good'] =
    (df.vtxid == 0) & (df.npng3d > 0)
  KL = ['run', 'subRun', 'event', 'slice']
  return df.groupby(KL)['good'].agg(np.any)
```

- *vtx_elastic* table has one entry per *vertex;* may be more than 1 per slice.
- *groupby* combines results for all vertices in one slice.
- Returns an array with one logical value per *slice*.

Workflow for translating old-style data to new-style data



Summary

- NOvA is taking ownership of our HDF “ntuple” production code
  - They will use this in their own future production.
  - Especially interested in using for machine learning; many tools work with HDF5 files.
- We will be doing large-scale performance testing of the code.
  - Similar design processing LArIAT data demonstrate perfect scaling to 76,800 ranks; *read* and *decompress* 42 TB of data in < 20 seconds wall-clock time.
  - We will compare against traditional HEP workflow on many ROOT files.
- We will be comparing performance with C++-MPI implementation.
- Integration with larger workflow that is also part of the SciDAC project
  - use of changes in event selection criteria to evaluation systematic uncertainties in the mixing parameter measurements
  - one integrated MPI program, to take best advantage of HPC platform.

Thanks to our co-workers

- We'd like to thank co-workers not already named part of the SciDAC project.
- From the NOvA collaboration:
  - Steven Calvez (Colorado State University)
  - Derek Doyle (Colorado State University)
  - Alex Himmel (FNAL)
- From the Fermilab Scientific Computing Division:
  - Brandon White