HEP-CCE: Scalable IO for Energy and Intensity Frontier Experiments

Taylor Childers & Philippe Canal

- The 1.5-day workshop had 35 registrants with representation from Argonne, Fermilab, Berkeley, Brookhaven, Oak Ridge, Oregon State, Cornell, Marquette, and U. Nebraska. All three DOE ASCR HPC facilities were represented. HEP experiments with current or future large computing needs were represented, including ATLAS, CMS, and DUNE. The core IO developers for these experiments were present including ROOT IO developers which is the software that most HEP experiments use to mediate their file IO disk operations. In addition, IO experts from NERSC, and the Argonne and Oak Ridge Leadership Computing Facilities were in attendance.
- The first morning and some of the afternoon were dedicate to introductory presentations to provide a base of understanding to facilitate the HEP-ASCR exchange. The remaining time was dedicated to topical discussion on how to move HEP toward using shared file systems effectively at HPC facilities.

Rick Zamora (Argonne Leadership Computing Facility) started off the morning with a very nice overview of current IO capabilities on HPCs.

This was followed up with a very well organized dive into ROOT IO structures and behaviors given by Philippe Canal.

Peter Van Gemmeren (ATLAS) & Brian Bockelman (CMS) provided overviews of how the experiments currently perform IO using ROOT tools.

Glen Lockwood (NERSC), Sarp Oral (Oak Ridge), and Kevin Harms (Argonne) provided background on current facilities technology and tools related to shared file systems.

Finally, Jim Kowalkowski (Fermilab) presented ongoing R&D in HEP related to IO performance.

The full schedule is here.

- There is work being done to include parallel IO (via MPI) into the ROOT file IO framework. This includes work done by a recent summer student at Argonne that will be continued by a post-doc at Berkeley.
- ATLAS & CMS rely heavily on the ROOT file IO tools. Given the proper modifications to ROOT, they may benefit.
- HDF5 is a popular file format offering parallel IO capabilities. In machine learning circles, and industry, it has garnered attention.
- The guidance from ASCR facilities was that Object Stores are the leading technology for scalable IO on HPCs and will likely remain so for the next ten years.
- The presentation from Jim indicated the SciDAC for HEP Data Analytics is studying use cases for Object Stores with Intensity Frontier experimental data.

File Formats, HDF5, ROOT, Object Stores.

- HDF5's growing popularity among Machine Learning communities has brought it some attention. A recent update enabled parallel IO operations on HPCs (using MPI).
- The discussion focused on the need for parallel IO capabilities inside ROOT and how those might perform compared to HDF5. Does the community need only one? The general consensus indicated support inside ROOT for HDF5 would be beneficial, particularly for experiments to provide custom outputs for their analyzers. Partial implementations exist with various level of support and could be an opportunity for a funded project to consolidate and productionize into a complete implementation.
- Discussion shifted to Object Stores which saw more engagement from the facilities experts.
- Quick introductory talk on the topic from Glen Lockwood (NERSC), and a follow up from Rob Ross (Argonne/SciDAC-RAPIDS) who collaborated with Jim's SciDAC to build a transient Object Store for the Intensity Frontier.
- Discussion on how the Energy Frontier experiments might benefit from this research. Possible sources of collaboration would be modifying this transient Object Store to run on an HPC and serve event data to/from running ATLAS/CMS software processes. These processes typically open input files as read-only to process events, then create new output files to store completed events. This transient Object Store would be well suited for this task.
- In relation to Object Stores, it was also discussed with the ROOT IO team that support for writing ROOT objects directly to an Object Store would be useful in this context and is a good opportunity for a funded project.

ROOT & Scalable I/O

- Discusssion of changes within ROOT that could enable scalable IO for current experiments that depend on ROOT for file reading and writing, i.e. ATLAS & CMS.
- HEP-CCE summer student, Amit Bashyal, at Argonne working with Taylor Childers to implement an parallel version of the ROOT TFile, TMPIFile, found in this GitHub repository. This was implemented in collaboration with the ROOT IO and ATLAS IO teams with the idea that it could be utilized for parallel IO on HPCs by ATLAS & CMS. The discussion identified this as useful work that would be finalized and included into the ROOT repository very soon after this workshop. There is a postdoc at Berkeley who will help with this. The current implementation supports writing output in parallel, while additional development will be needed to support reading inputs in parallel.
- Thread-level parallel IO support within ROOT was brought up within the context of LHC experiments running multi-threaded event loops. Some work was outlined for collaboration between the CMS and ROOT IO experts to identify and document the necessary tuning required to fix current performance bottlenecks.
- The last topic involved supporting future work related to Machine Learning. Many popular tools such as Tensorflow and PyTorch do not naturally read from ROOT formats. There are currently tools like `root_numpy` which convert ROOT formats into formats these machine learning tools utilize, but this can be a performance bottleneck. Therefore support for ROOT that reads a ROOT ntuple directly into a numpy array was requested and reported by the ROOT IO team to be in development.

ROOT Parallel IO intra-node & cross-node

- MPI-enabled writing is high priority to benefit from two-stage IO as described by Rick Zamora from ALCF. (Driven by ATLAS)
  - Continue TMPIFile testing, implementation
- MPI-enabled reading is also a goal. Need Guidance for implementation. (Driven by ATLAS)
- Thread Performance/Scaling: Lock contention for threaded read/write needs to be addressed in ROOT
  - Continue engaging ROOT IO team with CMS

ROOT Support for other things

- Medium term nice to have ROOT write directly to an object store
- Ease interfacing ROOT with HDF5
- Read directly a TTree into numpy format (in the works)
- Improving ROOT serialization support outside of the TFile context for Object Stores, i.e. streamline management/recording of StreamerInfo (schema description) (partially exists, needs collaboration and development) (ATLAS, Intensity Frontier)
- ROOT Improved compression techniques for structured data, (ECP R&D applicable?

- ALCF presented an interesting I/O profiler, "darshan”
  - https://www.alcf.anl.gov/user-guides/darshan

- The "event service model" is popular because of its inherent event-level load balancing capability

- Object stores enter the HPC scene. NERSC realizes that the parallel file systems' meta-data problem doesn't scale any longer. As of 2025, NERSC plans to have the storage performance tier object storage based with an MPI I/O integration. A POSIX interface will stay for compatibility but at a performance hit. It is not yet clear which native object store interface will be provided, so there might be some devil in the details.
- If we do not get away with one output file per node (CMS ideally wants just that), we probably have to have a ROOT equivalent to MPI I/O collective operations, i.e. the MPI enabled TBufferMerger.

Object Store

- Guidance from Facilities is that Object Stores are going to be the scalable performant IO tool for Exascale. Jim's SciDAC is investigating this for use in Intensity Frontier experiments, need to gauge usability for Energy Frontier.
  - Current model has Object Store inside batch job
  - Energy Frontier should engage with this activity and discuss how this R&D would be used for ATLAS/CMS workflows on HPCs. Local Event Service in ATLAS?
  - How can Object Store level pre-/post-storage processing help improve our workflows? Hot/Cold objects, handling compression on the fly? Metadata handling.
- ROOT over Object Store 'native' interfacing would be helpful.
  - Opportunity for joint R&D project

Possible for new R&D project

- MPI BufferMerger (likely PostDoct @ Berkeley)
- ROOT Files in Object Store
- HDF5 <-> ROOT bridge (consolidation)

Work going on in ROOT Team.

- Bridge ROOT [Tree] <-> NumPy-Array / ML [python] libraries
- RDataFrame over zillions of nodes
- Improved compression and [un]compression speed
- [New] easier out-of-TFile TStreamerInfo management