High Energy Physics Center for Computational Excellence

Optimizing Data Storage for Next-Generation HEP Experiments

Saba Sehrish
for HEP-CCE

HEP CCE AHM
July 22, 2024

Five project areas in SOP

Applying Lessons Learned to HEP Experiments, Mimicking Framework, Darshan and HDF5

Conveners: Amit Bashyal (Argonne National Lab), Christopher Jones (Fermilab), Kenneth Herner (Fermilab), Patrick Gartung (Fermilib), Peter van Gemmeren (ANL), Rui Wang (ANL), Saba Sehrish (Fermilab), Shane Snyder (Argonne National Laboratory)

Tracking and aiding the evolution of ROOT I/O, in particular RNTuple 10m

Speakers: Alaettin Serhan Mete (Argonne National Laboratory (US)), Amit Bashyal (Argonne National Lab), Barnali Chowdhury (Argonne National Laboratory), Christopher Jones (Fermilab), Daniel Riley (Cornell University), Kyle Knoepfel (Fermilab), Marcin Nowak (BNL), Matti Kortelainen (Fermilab), Peter van Gemmeren (ANL), Philippe Canal (FERMILAB)

Reduced Precision and Intelligent Domain-specific Compression Algorithms 10m

Speakers: Amit Bashyal (Argonne National Lab), Meghna Bhattacharya, Peter van Gemmeren (ANL)

Object Stores and Strategies for Data Placement and Replication 10m

Speakers: Bo Jayatilaka (Fermilab), Doug Benjamin (Brookhaven National Laboratory (US)), Nicholas Smith (Fermilab), Rob Latham (Argonne National Laboratory), Rob Ross (ANL), Saba Sehrish (Fermilab), Shane Snyder (Argonne National Laboratory)

Optimized Data Delivery to HPC systems 10m

Speakers: Marco Mambelli (Fermilab), Peter van Gemmeren (ANL), Saba Sehrish (Fermilab)

Project 1: Applying lessons learned to HEP experiments, mimicking framework, Darshan and HDF5

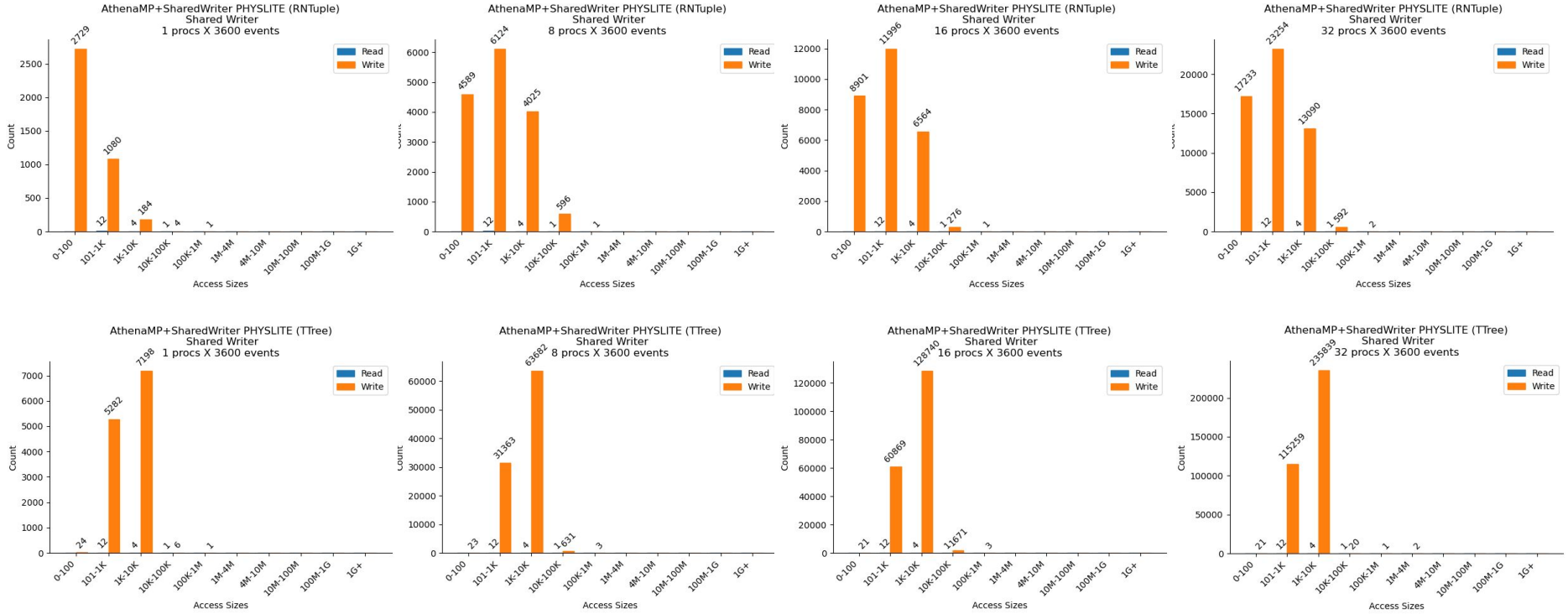
Highlights

- Mimicking framework will continue to be the main source for evaluating different data formats and libraries
  - successfully used for ROOT, HDF5, RNTuple
- Using Darshan to understand IO access and bottlenecks
  - Enhancing tools as needed by the HEP workloads
  - Athena is routinely using Darshan to gain insight into the workflows
    - Fastchian is one of the workflows that get insight with Darshan to assist the ongoing development on HPC
- Plan to use Darshan for optimizing HDF5 usage
  - already identified lack of collective metadata operations

Project 1: Applying lessons learned to HEP experiments, mimicking framework, Darshan and HDF5

- 3600 events/proc, input & output root files

SharedWriter process



Data access size comparison (PHYSLITE)

Project 1: Applying lessons learned to HEP experiments, mimicking framework, Darshan and HDF5

Expanding our use of IO tools to include Drishti

Drishti is a framework to **detect** the root causes of I/O bottlenecks, **map** them into actionable items, and provide guidance and **recommendations** to end-users to optimize their application's I/O performance

Limitations

- Drishti originally takes one I/O profiling log/trace as input for analysis and visualization
- No support for aggregated and multi-file visualization
- In this project these features are required to provide a full view of the I/O behavior

Drishti

Highlights

- Drishti now has a prototype solution to support for multiple Darshan traces from same job
  - Ongoing work for aggregating time-based counters
  - Revisiting the criticality of reported I/O issues when looking at the full picture (e.g., JSON, TXT, log files)
- Exploration of alternatives to support interactive visualization of issues with larger datasets
- Work with HDF5-based outputs to improve IO behavior

TTree

- athenaMP, PHYSLITE
- 8 processes, 3600 events/process
- 11 .darshan files in this job

Relevant Triggered Issues

- Heavy use of STDIO and POSIX
- Possible redundant I/O traffic
  - i.e., total write/read bytes > highest file offset
- 99.77% of file requests are misaligned
- High number (1685204) of small read requests (i.e., < 1MB)
  - 99.99% of all read requests
  - 62.32% are coming from a single .pool.root.1 file
- High number (117554) of small write requests (i.e., < 1MB)
  - 100.00% of all write requests
  - 74.81% are to .pool.root.1
- Write/Read imbalance
  - athenamp_eventorders.txt.Derivation
  - FileManagerLog, AthenaMP.log

RNTuple

- athenaMP, PHYSLITE-RNTuple
- 8 processes, 3600 events/process
- 11 .darshan files in this job

Relevant Triggered Issues

- Heavy use of STDIO and POSIX
- Possible redundant I/O traffic
  - i.e., total write/read bytes > highest file offset
- ↓99.76% of file requests are misaligned
- High number (↑1686256) of small read requests (i.e., < 1MB)
  - 99.99% of all read requests
  - 62.32% are coming from a single .pool.root.1 file
- High number (↓44947) of small write requests (i.e., < 1MB)
  - 100.00% of all write requests
- Write/Read imbalance
  - athenamp_eventorders.txt.Derivation
  - FileManagerLog, AthenaMP.log

Project 2: Tracking and aiding the evolution of ROOT IO in particular RNTuple

RNTuple is one of the upcoming **ROOT 7** features, targeting **HL-HLC production deployment**.
After successfully having served HEP for decades, ROOT's **TTree** container is being replaced by RNTuple:

- HEP has collected well over **1 ExaByte** of data stored in TTree
- ROOT will maintain TTree readability 'forever'
- However, there will be no performance updates to TTree

Recent activities

- Evaluating RNTuple API -- CMS able to store their data (all different formats) using RNTuple
- Storing CAF data using RNtuple

In the AHM

- There will be updates by Amit and Chris on these two RNTuple related projects
- Dedicated discussion session

Project 3: Reduced Precision and Intelligent Domain-specific Compression Algorithms

IOS: Surveyed different compression tools developed by computer scientists (SZ, MGARD, IDEALEM)

- Commonly used in many scientific fields outside of HEP fields
- Could find some applications in the HEP field

SOP: Activities

- Carry out research and testing of compression algorithms with the HEP experiments use case
- DUNE could have a special use case
  - Far Detector is large and homogenous → ~ GBs worth of raw data simple waveform like data per trigger

Parallel effort - approved ANL LDRD project for DUNE's use case

- Highly compressed DUNE raw data with enough fidelity could be written alongside derived data products
- Allows quick inspection of derived data products without having to access original raw data
- Test framework that generates fake raw data which are then compressed using compression tools before writing into RNTuple as blobs along with compression parameters
- Student started working on this project

Project 4: Object Stores and Strategies for data placement and replication

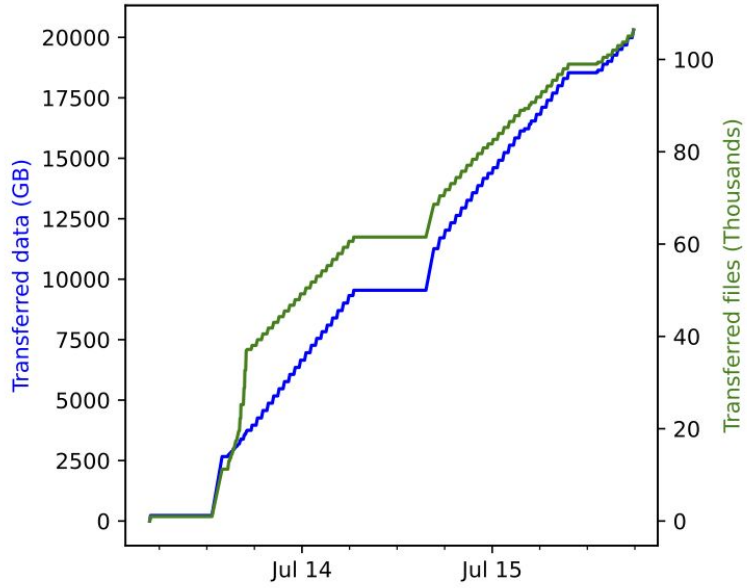
Plan:

- Benchmarking on DOE platforms
  - Talk with Florine de Geus (CERN) on where to get these?
  - Also Analysis Description Language (ADL) benchmarks?
  - Coordinate on scaling, test file and DAOS (DFS and object) configurations
- Darshan and ROOT (and RNTuple) (and DAOS)
  - Capture, persist, analyze TTPerfStats and RNTuple performance data
  - Will need io_uring support for RNTuple file
  - May lead to performance optimization suggestions…
- Other object storage technologies
  - FNAL, BNL continue looking at Ceph
  - Coordinate, share experiences, build shared view of technologies and capabilities, socialize methods to best exploit objects
- **Kickoff during AHM -- dedicated session**

Project 5: Optimized Data Delivery to HPC systems

Real-time DAQ raw data processing using HPC

- SBND, a Fermilab experiment, expressed the need to process its raw data using ALCF HPC systems
  - DAQ node transfers to Fermilab's dCache
  - A Globus DTN connected to Fermilab's CephFS was setup to transfer files to/from ALCF
- Data transfer test
  - ~125k raw files, ~250 MB each (SBND subsystems readouts containing about 20 events). In total ~20 TB was transferred over 2 days
  - Transfer to Argonne simultaneous to transfer from dCache
  - The test on the left had 2 interruptions that will be fixed in production: Globus httpd server crash and Kerberos ticket expiration
  - ~450 MB/s to Argonne sustained for multiple hours, meeting SBND needs
  - Bottleneck in local FNAL transfers from dCache (CephFS or direct)
- Future improvements/changes
  - Use Xrootd for transfers from dCache to DTN
  - Integration with a catalog (SAM or Rucio)
  - Pre-processing (SBND Decoding) on FermiGrid before sending to ALCF



Project 5: Optimized Data Delivery to HPC systems

Data Delivery and Xrootd

- HEP grid computing relies on availability of distributed data access solutions
    - Deliver data from distributed data stores to computing elements
- Xrootd is a high-performance, scalable, fault-tolerant distributed data access solution deeply integrated into the storage solutions of many HEP experiments
    - Major component of storage systems at CERN, Fermilab, and other labs

- In collaboration with the **SLAC team** responsible for Xrootd development
- CCE Phase 2 will evaluate opportunities to optimize the (streaming) delivery of data to object stores and parallel file systems running on HPC systems

Xrootd/streaming for HDF5

- Using our findings on HDF5 data formats and organization, the research proposed here will focus on the development and scheduling required for multi-threaded and serial writes to HDF5 from within the DUNE analysis framework I/O layer
- Additional I/O R&D will focus on optimized streaming delivery of HDF5 data via the xrootd or alternative protocols and tuning the DUNE data model to improve compatibility with HPC data storage systems

Optimizing Data Storage for next-generation HEP experiments: Summary

- Introduced Drishti for IO recommendations in addition to using Darshan
- Using Darshan and Drishti for understanding RNTuple IO access, plan to use for HDF5 as well
- Making significant progress in RNTuple usage, including API evaluation and providing feedback to the ROOT team; two separate talks to cover the progress
- Plan to kickoff object stores related work during AHM
- Continue to work on compression algorithms and learn from ANL LDRD
- Able to transfer SBND data to ALCF at a rate required by the experiment
- Will look into Xrootd use for streaming HDF5