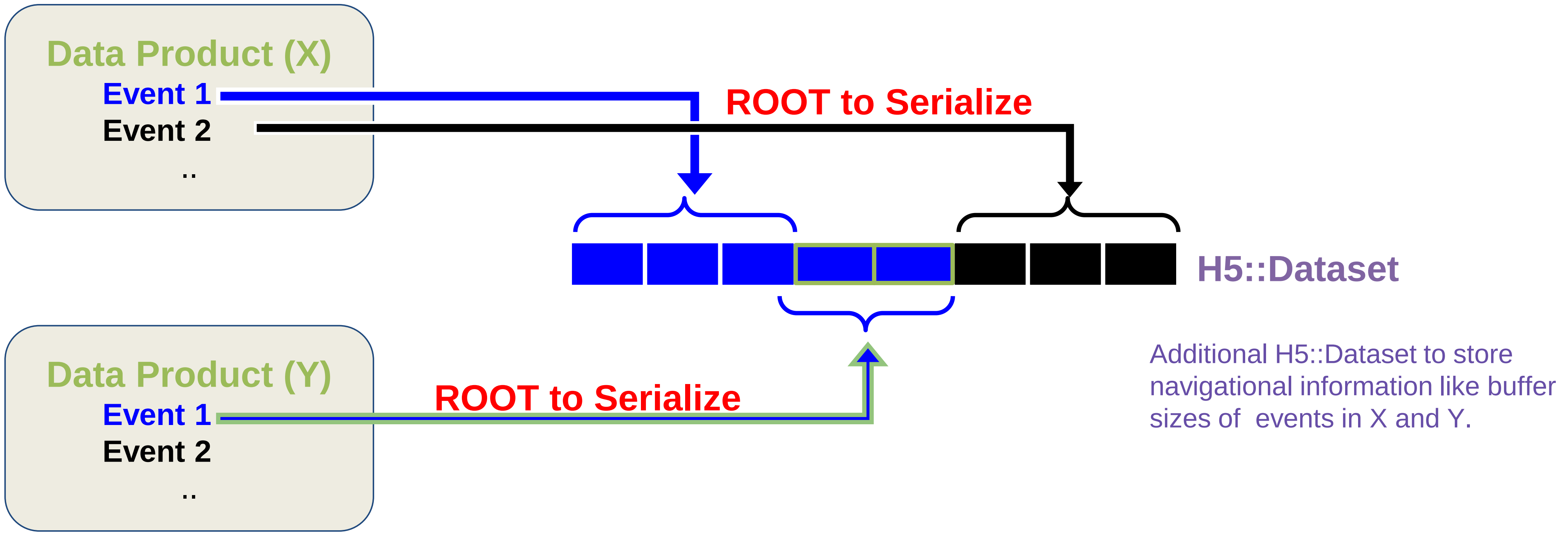
HEP-CCE:
Storage **OP**timization

Peter van Gemmeren (ANL)
On behalf of the HEP-CCE/SOP group

High Energy Physics-Center for Computational Excellence

- Started as a 3 year (2020-2023) Pilot Project now **Base Program**
  - 6 Experiments (**Energy**, **Intensity** and Cosmic Frontiers)
  - 5 US National Labs (ANL, BNL, FNAL, LBNL & Oak Ridge joined)
- Pilot Project of HEP-CCE:
  - Address one major issue: Deploying Leadership Computing Facilities (LCF) to help future HEP computing challenges (Processing Cycles)
  - Activities:
    - **P**ortable **P**arallelization **S**trategies for High-Performance Computing Systems
    - Fine-Grained **I/O** and **S**torage on HPC Platforms, including Data Models and Structures
      - Demonstrated the capability of leveraging parallel I/O libraries to write HEP data into HPC native backends like **HDF5** (CHEP23-Link)
      - Enhance I/O Characterization tool **Darshan** and monitor HEP workflows (CHEP23-Link)

HDF5 as Data Storage Format



Data Products are experiment specific C++ objects usually written in ROOT format.

Use **ROOT** as common tool to serialize C++ objects into byte stream array buffers

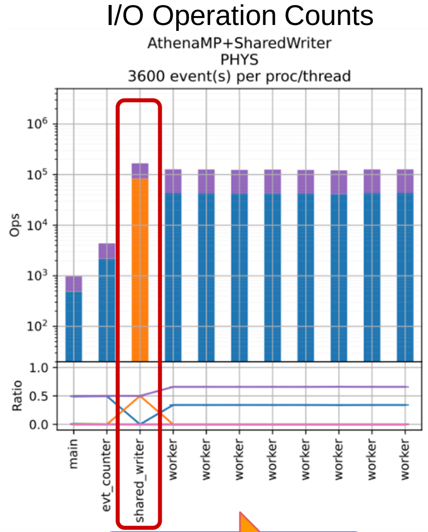
HDF5 Datasets store serialized data products with mapping optimized for parallel I/O. Mapping is independent of experiments.

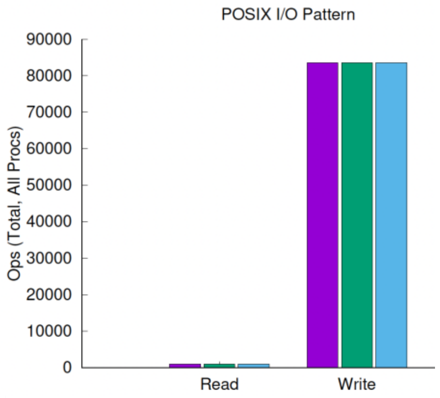
Amit Bashyal (Argonne), others, on behalf of HEP-CCE

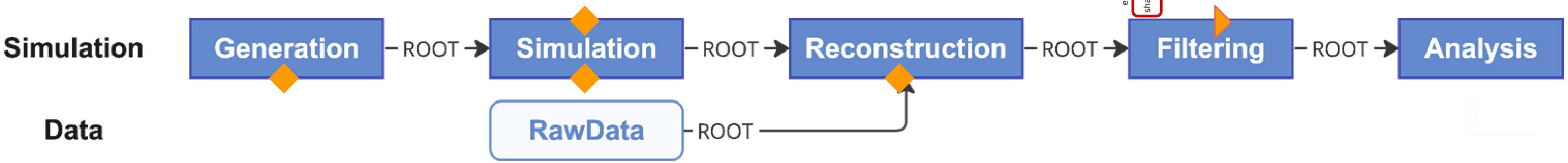
Case study: I/O operations

Broadwell on LCRC@ANL
GPFS

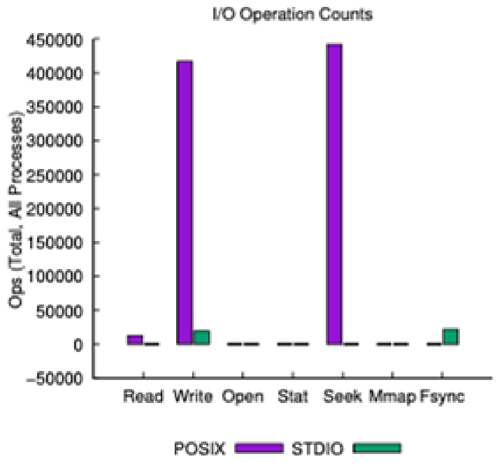
- **Equal number of writes/seeks**
  - Generation & Simulation & Reconstruction & SharedWriter process in Filtering stage at ATLAS (marked)







Haswell on Cori @Nersc
SSD + Lustre
100 events, 16 threads



- **Equal sequential & consecutive I/O**
  - Sequential – next access came somewhere after the last one in the file
  - Consecutive – next access starts with the byte immediately following the last access

Rui Wang (Argonne), others, on behalf of HEP-CCE

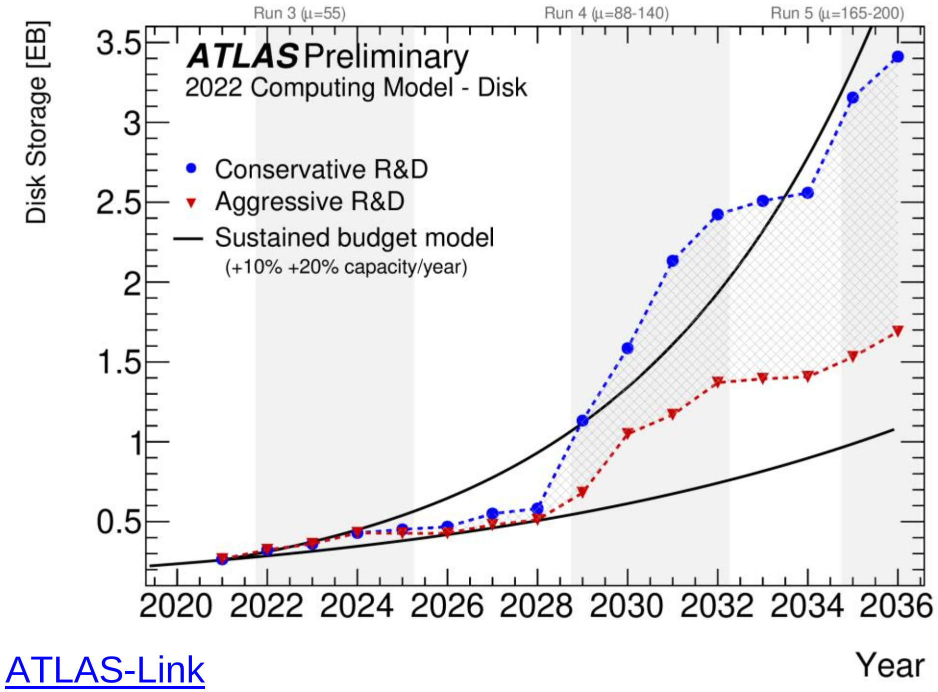
HEP-CCE2: IOS -> Storage Optimization SOP

After successful completion of Pilot Project and D.O.E. Review

HEP-CCE evolved as a Base Program and expanded its scope

Available **storage resources** can limit the physics reach of HL-LHC era experiments

- Optimizing Data Storage and Data Management
  - Investigate new storage backends and data volume reduction methods
    - Tracking and aiding the evolution of ROOT I/O, in particular **RNTuple**
    - Reduced Precision and Intelligent Domain-specific Compression Algorithms
    - **Object Stores** and Strategies for Data Placement and Replication
    - Optimized Data Delivery to HPC systems



ATLAS-Link

ROOT: From TTree to RNTuple

ROOT: HEP Community software used from data processing to physics analysis

- **TTree** as a storage backend that enables HEP experiments to use tools provided by ROOT ecosystem
    - Primary storage backend and I/O subroutine of HEP experiments for decades
    - Over Exabyte of data stored in TTree format
  - TTree evolved to address experimental needs and has been the backbone of HEP computational workflows
    - Now, supports persistence and I/O of complex experimental data
      - Decades of development made TTree outstanding in its support of C++ features
- However, TTree architecture predates recent overhaul in C++, modern programming paradigms and evolving computational landscape

RNTuple, and upcoming HEP experiments

RNTuple: New Storage backend in ROOT version 7

- State of the art, HEP community supported storage and I/O subsystem
  - Address storage & I/O requirements of upcoming HEP experiments
  - Streamlined compared to TTree, provides limited data model support
    - ATLAS and CMS report **20-40% saving in their storage** (CHEP23-Link)
  - Use of modern C++ standards
    - Adoption of smart pointers, better error handling mechanisms, modern C++ libraries
- HEP experiments have to adopt RNTuple to stay current with ROOT
  - Adopt to new RNTuple API
  - May have to change the data model to be persisted in RNTuple

HEP-CCE: Tracking and aiding the evolution of … RNTuple

HEP-CCE will aid HEP experiments to adopt RNTuple

- Co-organized RNTuple Workshop:
  RNTuple Format and Feature Assessment (6-7 November 2023) · Indico (cern.ch)
- HEP-CCE is conducting RNTuple API review:
  Special CCE-SOP tele-conference: RNTuple API Review Kick Off (February 28, 2024) · INDICO-FNAL (Indico)
  - Aid the development of RNTuple as per the experimental requirements
  - Find common guidelines and recipes for experiments frameworks and data models to migrate to RNTuple
- Includes experts from HEP-CCE experiments: ATLAS, CMS, DUNE plus Computing Scientists and is open to everyone.

RNTuple, and experiments status

ATLAS/Athena: Can store all production event data in RNTuple (ACAT24-Link)

- Framework encapsulates persistence technology (TTree) and separates complex Event Data Model from Persistence (T/P separation).

CMS/CMSSW: Capable of storing (analysis) nano-AOD in RNTuple

- Uses some currently unsupported features (e.g. dynamic polymorphism), may store data in un-split mode.

DUNE/art: No significant studies with art & RNTuple yet

- May benefit from developments in CMSSW (from which art was forked).
- HEP-CCE work on RNTuple support for CAF data (ACAT24-Link)

ALICE: Data Model build on Arrow Tables (no complex features), needs Bulk reading (done)

LHCb: Uses flat ntuple (simple), but requires multithreaded I/O (implemented in RNTuple)

Persistifying the Complex Event Data Model of the ATLAS Experiment in RNTuple

Alaettin Serhan Mete (Argonne), Marcin Nowak (Brookhaven), Peter Van Gemmeren (Argonne)

- ATLAS has been using ROOT's TTree storage backend for about two decades
- In LHC Run 4 (2029), ROOT's main I/O subsytem will be RNTuple
  - In a nutshell, a more modern and (compute and storage-wise) efficient technology
- ATLAS has made significant progress for adopting RNTuple for its event data
  - **All applicable ATLAS data formats can we written into RNTuple seamlessly**
  - Both reading and writing are supported on the official software framework (Athena) side
  - Everything is handled by the I/O infrastructure with no change needed for the client code
- Preliminary estimates suggest **20+% storage savings** in some analysis formats
- Getting production-ready still needs a number of key milestones reached:
  - Finalizing/adopting a number of in-progress RNTuple work, e.g., fast merging etc.
  - Updating standalone tools used by the production system for metadata access, file validation etc.
  - Running large-scale stress tests and performing detailed validation studies
- ATLAS will use the rest of Run 3 and the Long Shutdown 3 to deliver these!

CAF Data Model and Persistence in RNTuple

StandardRecord Object

- Event Information
- Incident Beam Related Information
- Generator Level Information
- Reconstructed at Near Detector
- Reconstructed at Far Detector

- **StandardRecord (SR)**: Top level CAF object
- Summary of neutrino event
- Information related to neutrino event as SR member objects

```
============================================================
NTUPLE:      NTuple
Compression: 404
------------------------------------------------------------
  # Entries:        10
  # Fields:         1396
  # Columns:        1091
  # Alias Columns:  0
  # Pages:          138
  # Clusters:       1
  Size on storage:  3729 B
  Compression rate: 2.06
  Header size:      15883 B
  Footer size:      1069 B
  Meta-data / data: 4.546
```

StandardRecord object can be persisted in RNTuple

Amit Bashyal (Argonne), others, on behalf of HEP-CCE

Reduced Precision and Intelligent Domain-specific Compression Algorithms

Most experiment HEP data is stored compressed format using lossless compression, lossy compression are less common

- To reduce storage requirements further, experiments and ROOT are investigating means of reduced-precision storage as much of the data is derived from measurements with inherent uncertainties
- For derived data, **not RAW**
  - Under study for ATLAS PHYSLITE data, Potential **storage savings ~20-30%**
- Need trust-building/safeguarding validators, but may enable keep information down-stream.

IOS team has surveyed different tools developed by computer scientists:

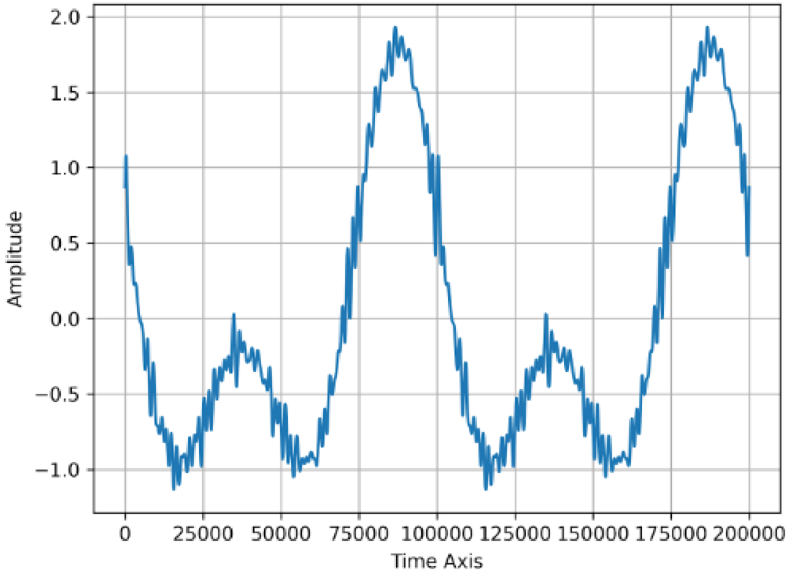
- Hybrid Learning Techniques for Scientific Data Reduction with **MGARD**
- Compression of Scientific Data with **SZ**
- Statistical Similarity for Data Compression with **IDEALEM**

Compression Framework

Working on a **test framework** that generates (or takes as input) raw data, applies intelligent compression using the tools mentioned above and writes compressed data (including in RNTuple).

Perform tests to **measure fidelity and usability** of the compressed data downstream (raw data remains loss-less).

- DUNE FD Trigger raw data will be large waveform (~ few GBs) where some of the above tools could perform well in terms of compression and data fidelity
  - Faster and easier way to inspect data than accessing original raw data
  - Could be useful in some further **ML based analyses** that could be compute intensive but does **not require full precision of the data**
  - Needs collaboration with DUNE stakeholders, compression developers and the ROOT team



Representative waveform like data used to test the compression algorithms

```
original_data [#0]  --  SplitIndex64
  # Elements:          1
  # Pages:             1
  Avg elements / page: 1
  Avg page size:       8 B
  Size on storage:     8 B
  Compression:         1.00
original_data._0 [#0]  --  SplitReal64
  # Elements:          200000
  # Pages:             24
  Avg elements / page: 8333
  Avg page size:       49072 B
  Size on storage:     1177740 B
  Compression:         1.36
```

Waveform data persisted in RNTuple as an array of length 200000 with a loss-less compression of 1.36 by RNTuple.

```
compressed_data [#0]  --  SplitIndex64
  # Elements:          1
  # Pages:             1
  Avg elements / page: 1
  Avg page size:       8 B
  Size on storage:     8 B
  Compression:         1.00
compressed_data._0 [#0]  --  Char
  # Elements:          12115
  # Pages:             1
  Avg elements / page: 12115
  Avg page size:       12115 B
  Size on storage:     12115 B
  Compression:         1.00
```

Compressed (using SZ3) data stored in RNTuple as an array of characters of length 12115.

- Compression ratio of original to compressed data using SZ3 → 99
- Compression ratio using MGARD → 28
- Integration of IDEALEM ongoing.
- Ongoing further optimization in terms of test data features and compression parameters

Amit Bashyal (Argonne)

Object Stores and Strategies for Data Placement and Replication

- Numerous potential advantages for using in HEP:
  - **Reference** rather than copy **upstream data**, saving space
  - Allow **fine-grained versioning**, avoiding replication of unchanged objects
  - Facilitate **user-driven data augmentation**, to subset of events
  - These methods of referencing save storage space
- Object storage activities on HPC side as well, e.g. Distributed Asynchronous Object Storage (**DAOS**)
  - DAOS is an object storage service developed for use on **persistent memory** technologies as a **very high performance** online storage layer
    - Data model includes both key:value objects and array objects
    - Array objects can be used to streamline storage of large multidimensional arrays with record addressability
    - Access can be via POSIX or directly via **custom API**

Object Stores, DAOS, and RNTuple

ROOT’s RNTuple supports DAOS

- Decoupling of namespace operations from data read/write is natural for ROOT data.
- Similar to key–value storage where the key is a UUID, but specifically tuned for low latency / high bandwidth workloads

HEP-CCE is studying RNTuple DAOS implementation using **Darshan**

- Darshan already provides initial support for characterizing DAOS storage access
- Building on: IOS has successfully used Darshan for current HEP workflows using ROOT
- Aligns with, and will benefit from, other activities to understand and tune DAOS use by team members

Outlook

Since becoming base program, HEP-CCE can contribute to a wider variety of challenges, including storage.

Need to ensure to be relevant to our Clients, **the Experiments**, such as ATLAS, DUNE, and CMS

Work together with Computing Science experts and Community Software teams, such as ROOT

Acknowledgement

This work was supported by the U.S. Department of Energy, Office of Science, Office of High Energy Physics, High Energy Physics Center for Computational Excellence (HEP-CCE)