

# High Energy Physics Center for Computational Excellence

Optimizing Data Storage for  
Next-Generation HEP Experiments

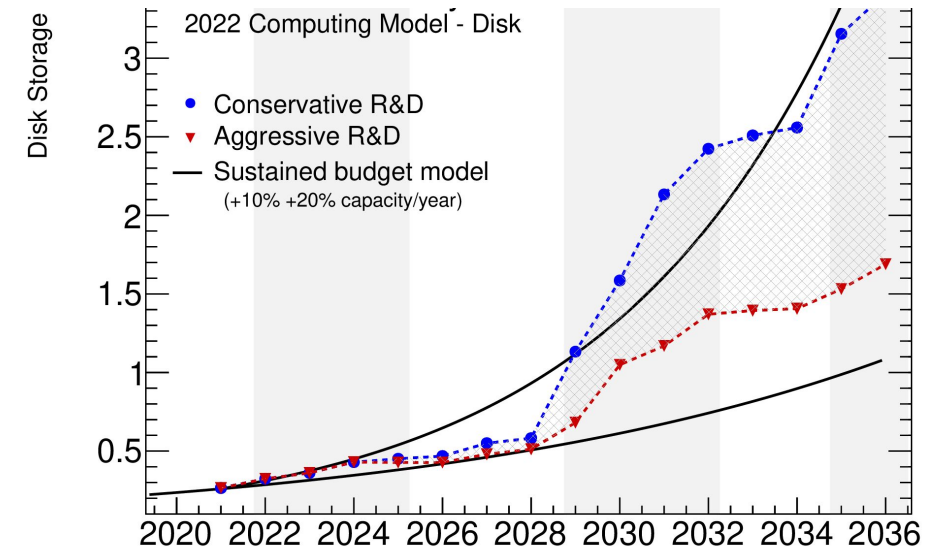
Rob Ross, Peter van Gemmeren  
for HEP-CCE

DOE HEP CCE Review  
December 18 2023

# Optimizing Data Storage for Next-Generation HEP Experiments: Motivation

### Future Storage needs for HEP

- The physics reach of many HEP experiments, particularly the LHC ones, is projected to be limited by available storage resources
  - No opportunistic storage resources
  - But there are efforts by experiments and other projects, e.g., ServiceX, iDDS, HEPnOS
  - And continuation of previous IOS work can help tuning I/O patterns to better suit HPC platforms, e.g. Darshan, HDF5



The expected disk storage needs of ATLAS in Exabytes as a function of time.

ATLAS Collaboration. ATLAS Software and Computing HL-LHC Roadmap. Tech. rep. Geneva: CERN, 2022. <https://cds.cern.ch/record/2802918>.

### How HEP-CCE can contribute

- Building on the expertise and formation of the HEP-CCE/IOS team, we can help address this storage crisis by:
  - Applying Lessons Learned to HEP Experiments, **Darshan** and **HDF5** [\[Report 3.1\]](#)
  - Tracking and aiding the evolution of ROOT I/O, in particular **RNTuple** [\[Report 3.5.1\]](#)
  - Reduced Precision and Intelligent Domain-specific **Compression Algorithms** [\[Report 3.5.2\]](#)
  - **Object Stores** and Strategies for Data Placement and Replication [\[Report 3.5.3\]](#)
  - Optimized **Data Delivery to HPC systems** [\[Report 3.5.4\]](#)

# Applying Lessons Learned to HEP Experiments: IOS

## Optimizing HEP workflow I/O for HPC storage systems

- IOS will continue the current work on Darshan, HDF5 and Data Model

## Mimicking Framework and Darshan (also see IOS status)

- Extend emulation and characterization capabilities from application to the workflow level
- Darshan analysis tools for workflows
  - Refactor PyDarshan to easily allow aggregation and visualization of Darshan data across multiple logs
- Instrumentation of Intel DAOS I/O libraries
  - See later: [\[Report 3.5.3\]](#)

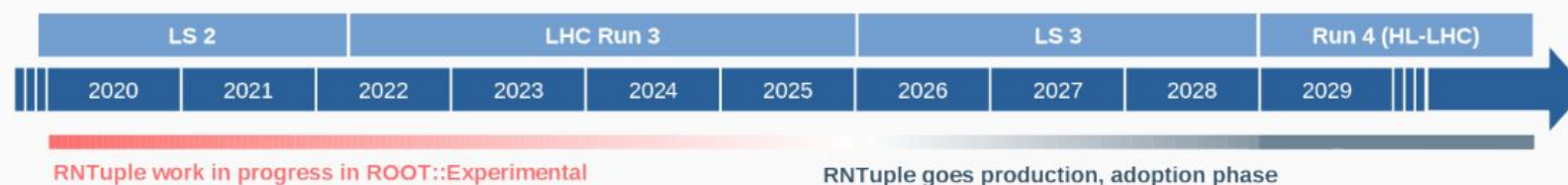
## HDF5 and Data Model

- Study survey findings from work by various experiments: <https://github.com/hep-cce2/GPU-DM>
  - Goal is not to duplicate these efforts, but augment them to make results more generic and alloc common use.
- Take into account storage considerations, such as more limited data model support by HDF5, other backends (or ROOT RNTuple, see later: [\[Report 3.5.1\]](#))
- Implement the generalized approach by experiments in the **Mimicking framework**
- Implement tasks (simplified and generalized) that can be offloaded into GPUs
- Design existing framework that can also be used as a training tool

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

## ROOT: TTree to RNTuple migration

- 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
- **RNTuple** is one of the upcoming **ROOT 7** features, targeting **HL-LHC production deployment**.
  - Comes with modern C++ interfaces
    - `unique_ptr`, templates, better error handling, `std::string`
  - Promises better performance than TTree I/O
    - including **significant storage reduction**
  - Already available in ROOT 6.24
    - in 'Experimental' namespace



# IOS: Tracking the evolution of ROOT I/O, in particular RNTuple

## Prior IOS Work related to RNTuple

### ROOT 7 and RNTuple, [link](#)

- Tuesday Jan 27, 2021
- Speakers: Javier Lopez Gomez; , Philippe Canal (FERMILAB)

### RNTuple studies for ATLAS, [link](#)

- Tuesday Feb 8, 2022
- Speaker: Marcin Nowak (BNL)

## RNTuple::PrintInfo() Example

```
>>> reader = ROOT.Experimental.RNTupleReader.Open('My RNTuple', "rntuple.root");
>>> reader.PrintInfo()
***** NTUPLE *****
* N-Tuple : My RNTuple *
* Entries : 5 *
*****
* Field 1 : PackedContainer<int> (PackedContainer<int>) *
* Field 1.1 : :_0 (std::vector<std::int32_t>) *
* Field 1.1.1 : :_0 (std::int32_t) *
* Field 1.2 : :_1 (Interface) *
* Field 1.3 : m_parms (Parameters) *
* Field 1.3.1 : m_nbits (std::uint8_t) *
* Field 1.3.2 : m_nmantissa (std::uint8_t) *
```



Marcin Nowak, BNL NPSS / ATLAS HEP-CCE/IOS meeting 8 Feb 2022

8



# Experiment Status of RNTuple

## ATLAS and CMS can store most derived analysis product in RNTuple

- Most recent work done by ATLAS team in close collaboration with ROOT experts. Outcome was presented during CHEP @ JLab:

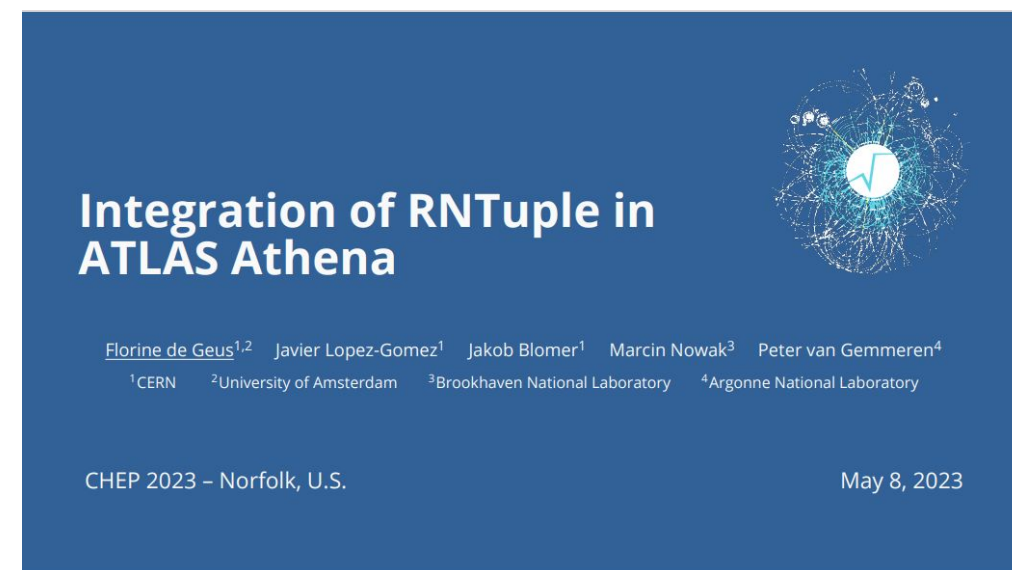
### Integration of RNTuple in ATLAS Athena

### Speaker

Mrs De Geus, Florine (University of Amsterdam)

<https://indico.jlab.org/event/459/contributions/11559/>

- **CMS** has been able to store their **nano-AOD** in RNTuple for a while
  - Flatter data model than ATLAS' DAOD-PHYS



**Integration of RNTuple in ATLAS Athena**

Florine de Geus<sup>1,2</sup> Javier Lopez-Gomez<sup>1</sup> Jakob Blomer<sup>1</sup> Marcin Nowak<sup>3</sup> Peter van Gemmeren<sup>4</sup>  
<sup>1</sup>CERN <sup>2</sup>University of Amsterdam <sup>3</sup>Brookhaven National Laboratory <sup>4</sup>Argonne National Laboratory

CHEP 2023 – Norfolk, U.S. May 8, 2023

# Experiment status of RNTuple

## ATLAS and CMS can store most derived analysis product in RNTuple

- Both experiments see **very significant storage reductions**
- Depending on:
  - Data Model/Data Product
  - Compression Algorithms
  - Storage Parameter
- Preliminary, numbers:
  - CMS nano-AOD: **30-40% reduction in file size**
  - ATLAS DAOD-PHYS: **20-30% reduction in file size**

## On HL-LHC scale, these savings would correspond to ExaBytes of Disk and Tape

- Main contributors to space savings:
  - i. More compact representation of collections and bools
  - ii. Data encoding optimized for better compression ratio (byte-splitting, delta encoding, etc.)

# Data Model support of RNTuple

## Streamlined RNTuple, will not support full C++ data models (as TTree does)

- To achieve better performance than TTree, RNTuple design choices made it more streamlined and reduced support for very complex data model features.

## Complex production data models will need redesign

- HEP-CCE will:
  - provide generalized templates and guidelines for developing data models that can be stored in RNTuple
    - This effort is synergistic with the design of HPC-friendly data model
  - identify possible limitations and coordinate areas of improvement for RNTuple while it is still in the experimental stage

Type	Examples	EDM Coverage		RNTuple Status
PoD	bool, int, float	Flat n-tuple	Reduced AOD	Available
Vector<PoD>	RVec<float>			Available
String	std::string		Full AOD / RECO	Available
Nested vector	RVec<RVec<float>>			Available
User-defined classes	"TEvent"			Available
User-defined collections	"TCudaVector"			Available
stdlib collections	std::map, std::tuple			Avail. / Testing
Variadic types	std::variant, std::unique_ptr			Avail. / Testing
Intra-event references	"&Electrons[7]"			In design
Low-precision floating points	Float16_t, Double32_t	<i>Optimization benefitting all EDMs</i>		Testing
	Custom precision and range			In design
	Precision cascades <small>ACAT'22</small>			In design

Data model support by RNTuple.

Jakob Blomer, Philippe Canal, Axel Naumann, Javier Lopez-Gomez, Giovanna Lazzari Miotto, "ROOT's RNTuple I/O Subsystem: The Path to Production," CHEP, May 2023.

<https://indico.jlab.org/event/459/contributions/11594/attachments/9389/13620/rntuple-chep23.pdf>



# ROOT/HEP-CCE Workshop/Day on RNTuple

## Experts from ROOT and HEP-CCE are co-organizing RNTuple workshop

- Monday 6 Nov 2023
- Participation of relevant experts from the experiments
- ROOT hopes to finalize the first version of the RNTuple binary format by the end of 2023
- HEP-CCE/IOS program plans to host experiments efforts of adopting RNTuple and find common solutions and guidelines
- Conduct API Review

The screenshot shows the Indico event page for "RNTuple Format and Feature Assessment". The event is scheduled for Monday 6 Nov 2023, from 02:00 to 11:00 US/Central. The registration status is "You are registered for this event" with a "Modify registration" button. The participants listed are Jakob Blomer, Marco Clemencic, and Peter Van Gemmeren. Below the event details, there is a section for "RNTuple Feature Discussion with Experiments" from 02:00 to 11:00, with conveners Jakob Blomer (CERN), Peter Van Gemmeren (Argonne National Laboratory (US)), and Philippe Canal (Fermi National Accelerator Lab. (US)). The page footer includes the CERN logo, "Powered by Indico v3.2.5-pre", and links for Help, Contact, Terms and conditions, URL Shortener, and Privacy.

# Reduced Precision and Intelligent Domain-specific Compression Algorithms

## Reduced Precision Storage

- Most experiment HEP data is stored in a compressed format using standard lossless compression algorithms
  - Lossy compression algorithms 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
  - Currently used in CMS nano-AOD
  - Under study for ATLAS PHYSLITE data
- Potential storage savings ~20-30%

## Intelligent Domain-specific Compression

- IOS team has surveyed different tools developed by computer scientists and used elsewhere in science that could find application for some HEP domains:
  - Hybrid Learning Techniques for Scientific Data Reduction with [MGARD](#)
  - Compression of Scientific Data with [SZ](#)
  - Statistical Similarity for Data Compression with [IDEALEM](#)

# Reduced Precision and Intelligent Domain-specific Compression Algorithms

## Challenges

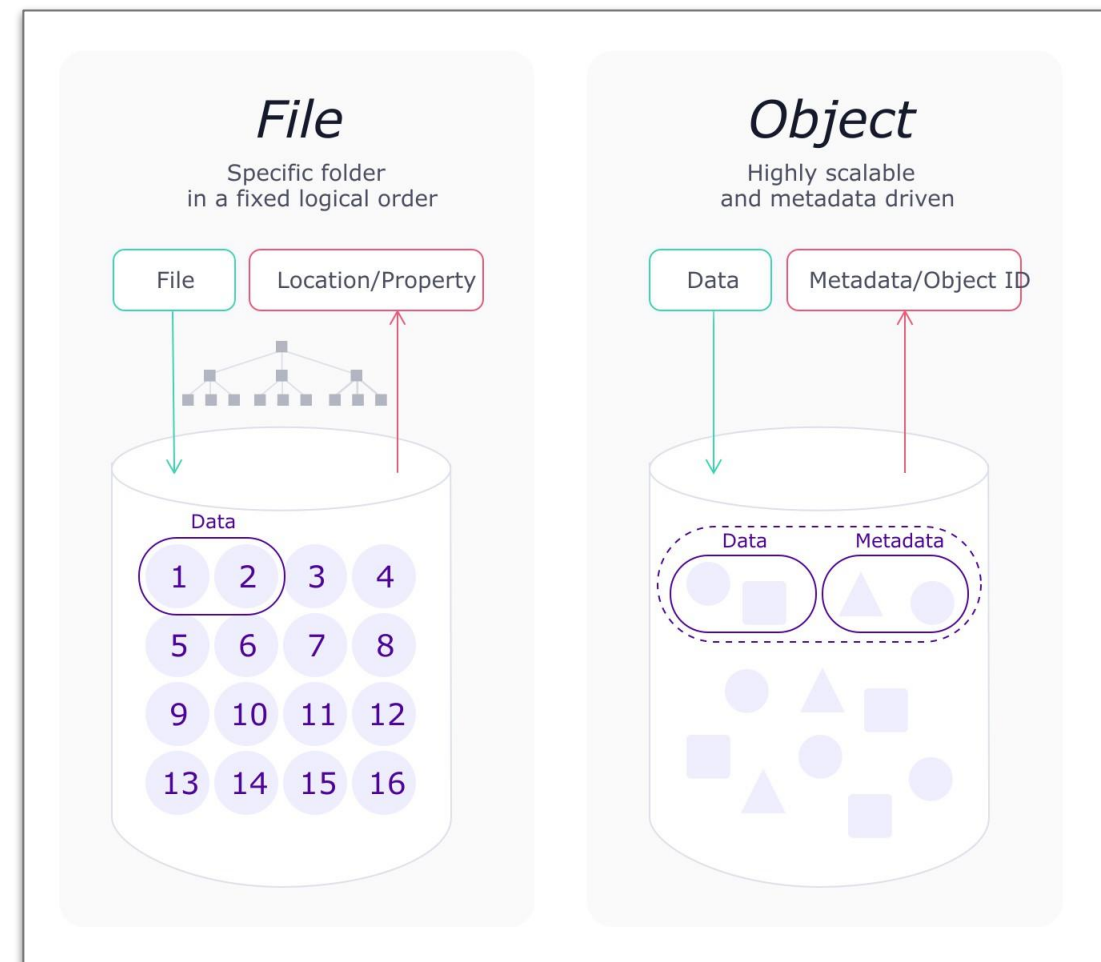
- As data is one of the most important assets of HEP experiments, any reduction of stored content is rightfully scrutinized, but at the same point storing values with precision far greater than their measurement wastes storage and compute resources that could better be used elsewhere

## HEP-CCE Role

- HEP-CCE will carry out research and testing of experiment use-cases in a coordinated fashion
- Intelligent/lossy compression with ROOT resident data may require infrastructure enhancements that would benefit from common developments
- Tools, metrics, and methods for validation and safeguarding of lossy compressed data to not reduce its physics potential can be developed within HEP-CCE

# Object Storage: What is it?

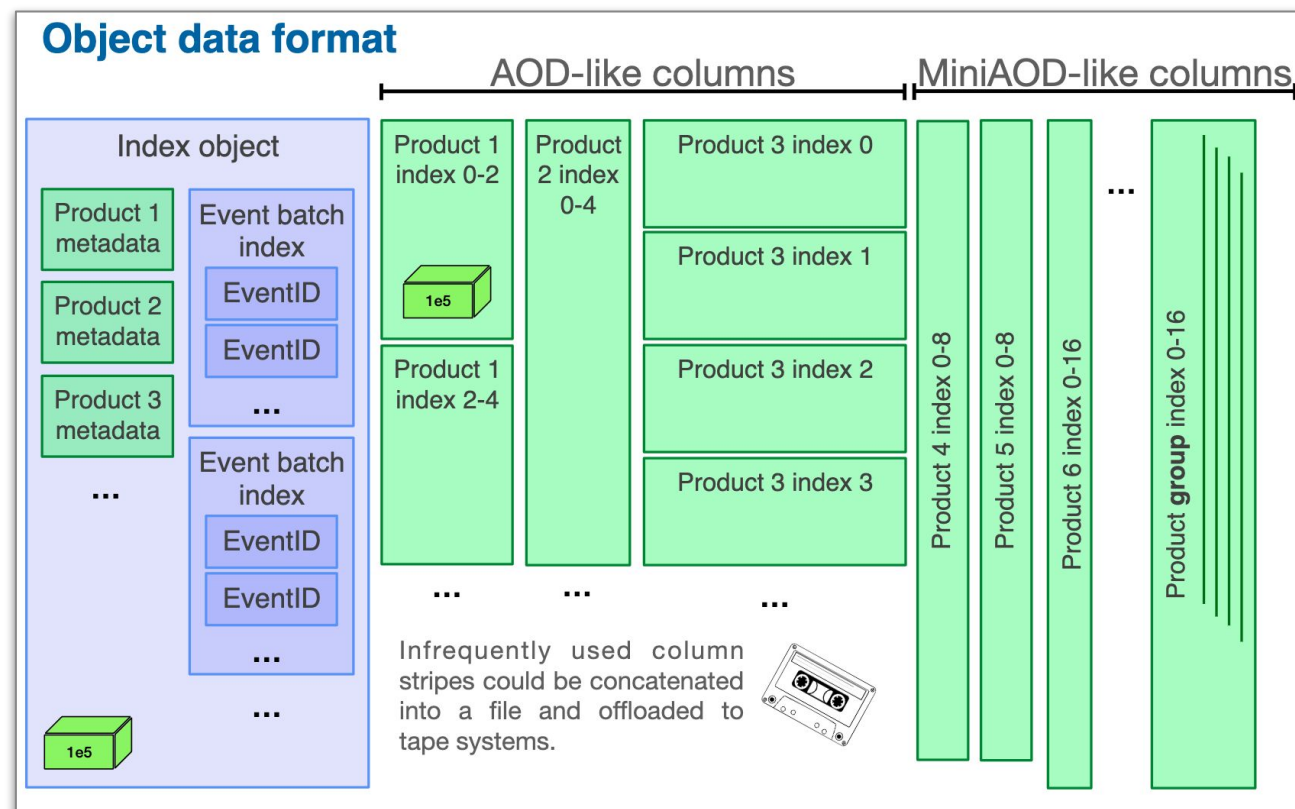
- “Object storage” is a term used to describe a collection of storage technologies that focus on the ability to store, reference, and access large collections of unstructured data
- Unlike a file system, how you find things is generally handled separately (e.g., via a DB)
- There are lots of flavors of object storage being used today in different contexts:
  - **Cloud storage** (e.g., S3): large, immutable objects, HTTP access
  - **Distributed filesystems** (e.g., RADOS): large, mutable and byte-addressable objects with file system access (i.e., Ceph)
  - **HPC storage** (e.g., DAOS): semi-structured, mutable, byte-addressable objects with key/value access



<https://www.scaleway.com/en/blog/understanding-the-different-types-of-storage/>

# Storing ROOT Data in Objects

- 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
  - DAOS
  - HDF5 over objects
  - Data lakes for AI applications



The FNAL team is investigating mapping CMS datasets into Ceph objects. The approach is not specific to Ceph, although different mappings might be more advantageous on specific underlying technologies.

Bo Jayatilaka, Christopher Jones, Nicholas Smith, "Using CEPH Object store with ROOT serialization in CMS", December 2022.

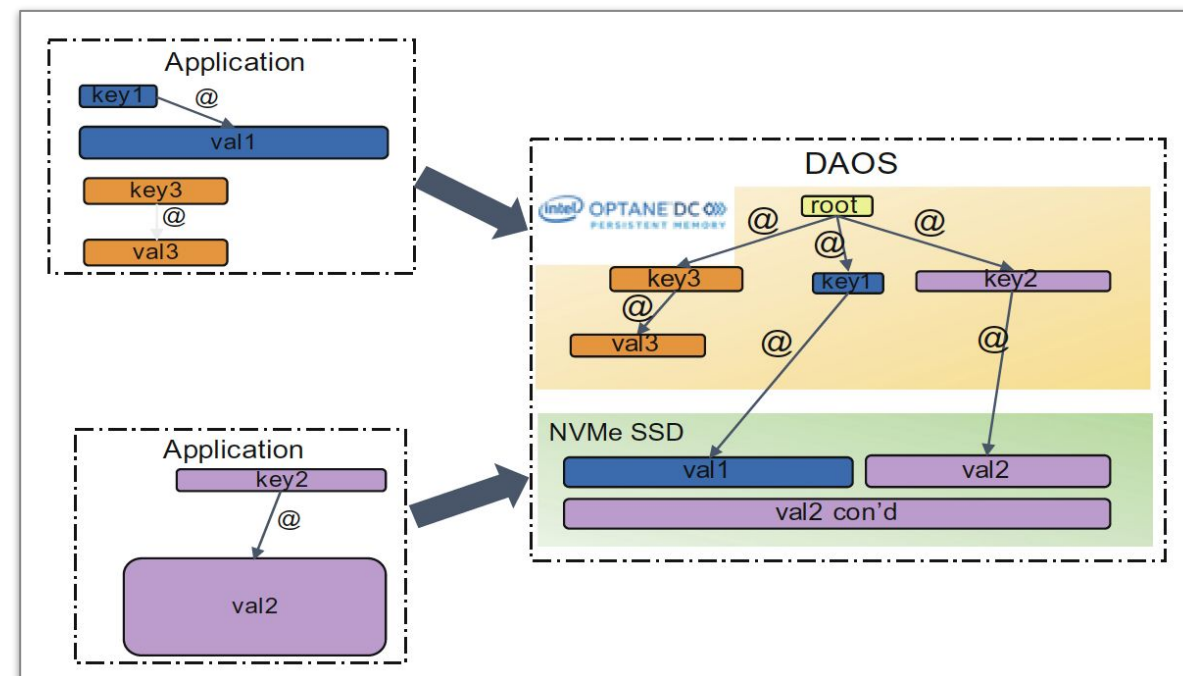
[https://indico.fnal.gov/event/57189/contributions/254706/attachments/162368/214598/n\\_csmith-uscms-objectstoresQ4.pdf](https://indico.fnal.gov/event/57189/contributions/254706/attachments/162368/214598/n_csmith-uscms-objectstoresQ4.pdf)



# 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
  - Custom API, array objects, striping all provide opportunities for optimization beyond a “standard” object store



Example of keys and references employed in a DAOS volume. Array objects preserve record addressability that is incredibly valuable in many HPC contexts (e.g., HDF5 arrays).

Zhen Liang, Johann Lombardi, Mohamad Chaarawi, Michael Hennecke, "DAOS: A Scale-Out High Performance Storage Stack for Storage Class Memory," June 2020.

[https://doi.org/10.1007/978-3-030-48842-0\\_3](https://doi.org/10.1007/978-3-030-48842-0_3)

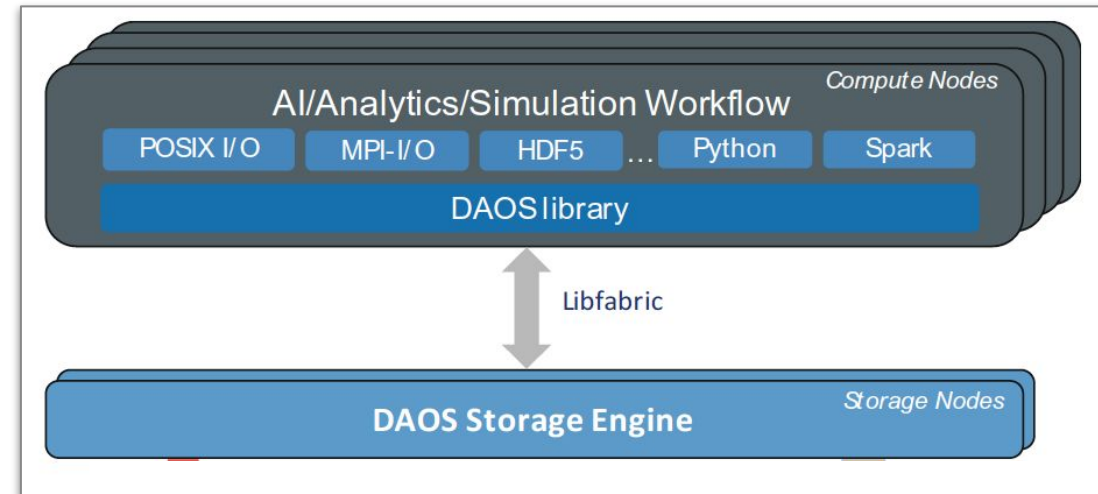
# Object Stores, DAOS, and RNTuple

## 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 will study RNTuple DAOS implementation using Darshan

- Darshan already provides initial support for characterizing DAOS storage access
- 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



A variety of user APIs have already been developed for using DAOS from applications, including POSIX, HDF5, and ROOT.

Zhen Liang, Johann Lombardi, Mohamad Charawi, Michael Henneke, "DAOS: A Scale-Out High Performance Storage Stack for Storage Class Memory," June 2020.

[https://doi.org/10.1007/978-3-030-48842-0\\_3](https://doi.org/10.1007/978-3-030-48842-0_3)

## 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

## Building on the work done by IOS:

- Will continue work on Darshan, HDF5 and Data Model

## Adding new focus on Optimizing Data Storage for next-generation HEP experiments

- This work will leverage our experiences where possible
- Take advantage of all the group's membership and expertise
- Act as forum for HEP experiments

## Improvements will have significant impact to resource needs:

- Storage requirements for HL-LHC and DUNE are very sizeable and much larger than previous experiments.
- New approaches can lead to large savings and it may be worthwhile to invest in previously neglected techniques.
- New technologies allow for possibilities that weren't available in LHC Run 1-3



# Thank you



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). This research used resources at Argonne Leadership Computing Facility, NERSC and BNL Scientific Data and Computing Center.