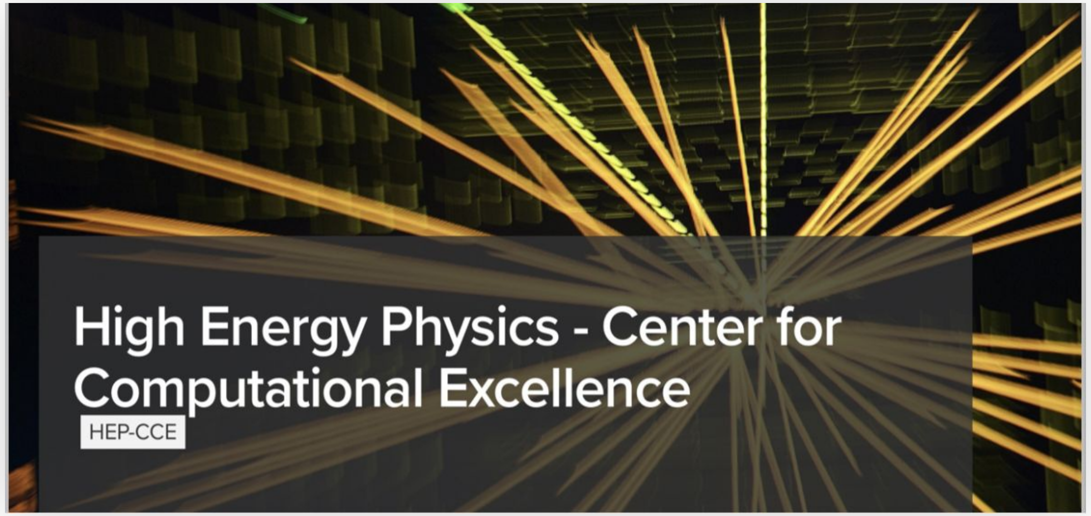
HPC Friendly Data Model and RNTuple in HEP-CCE

Amit Bashyal (ANL), Meghna Bhattacharya (FNAL),
Peter Van Gemmeren (ANL), Saba Sehrish (FNAL)
on behalf of HEP-CCE

March 11, 2024

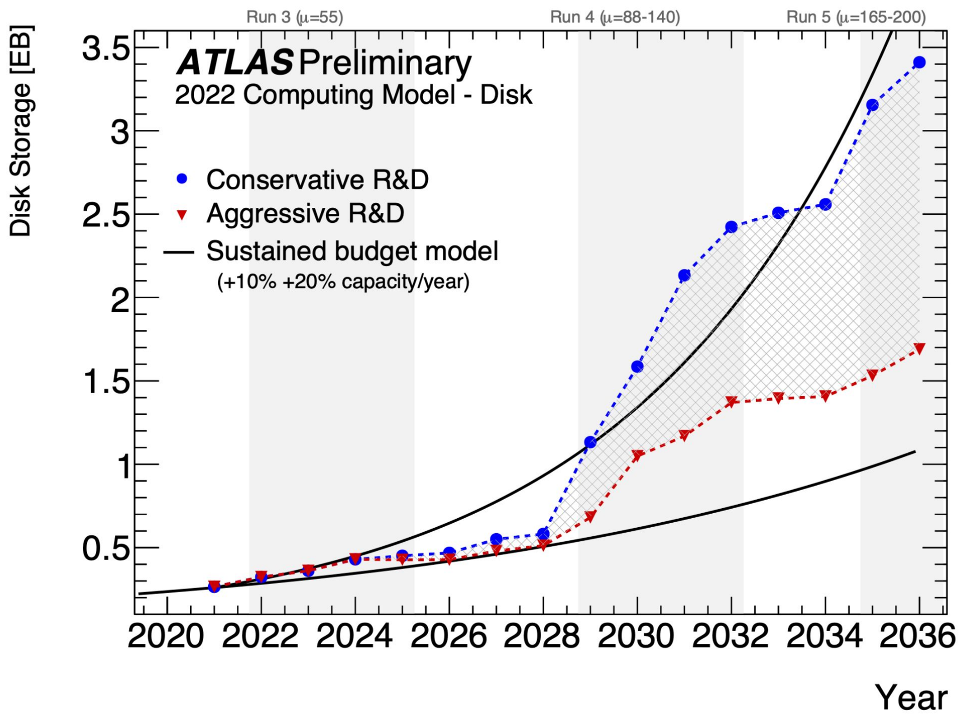


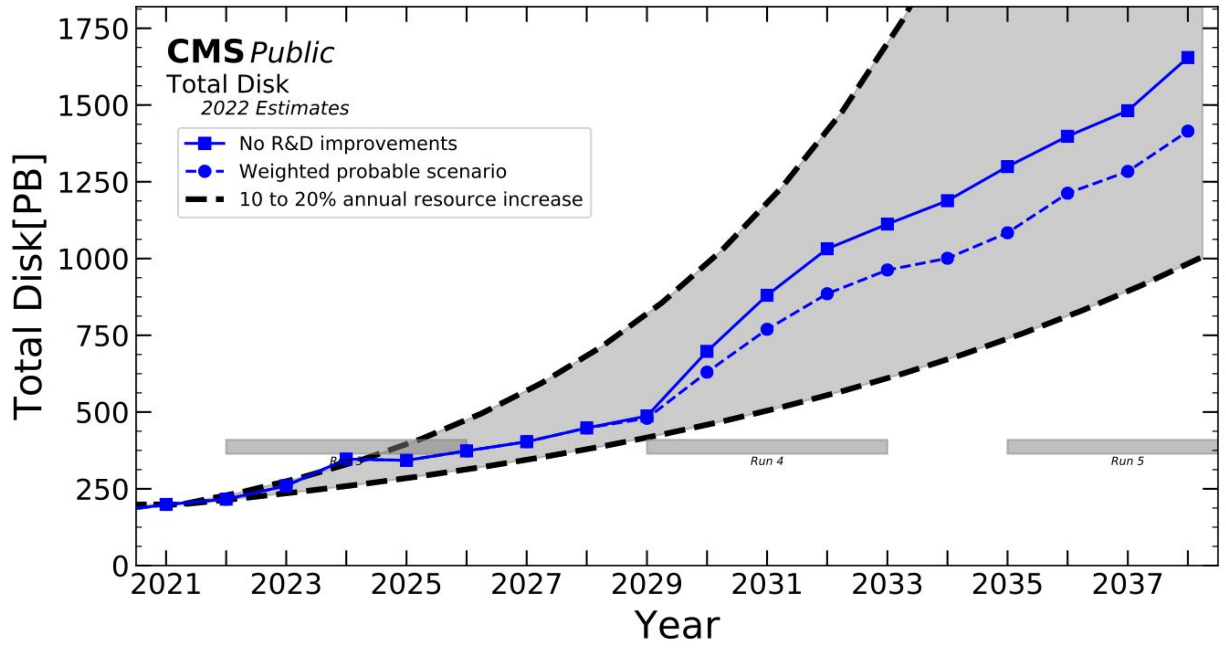
U.S. DEPARTMENT OF ENERGY | Office of Science | Argonne NATIONAL LABORATORY | OAK RIDGE National Laboratory | Brookhaven National Laboratory | Fermilab | BERKELEY LAB Bringing Science Solutions to the World

High Energy Physics-Center for Computational Excellence

- HEP-CCE: Introduction
  - Started as a 3 year (2020-2023) Pilot Project
  - 6 Experiments (Energy, Intensity and Cosmic Frontiers)
  - 5 US National Labs (Started with 4 labs in first iteration)
- First Iteration of HEP-CCE:
  - Address a major issue:
    - Deploying LCF computing facilities to help future HEP computing challenges
  - Portability, event generators etc on HPCs.
    - Developed performance and portability strategies of the HEP software stack to use HPC resources
    - Modify once → Use in multiple HPC systems with different architectures (CHEP 2023)
  - **Input & Output and Storage (IOS)**
    - Study and Development of I/O capability of HEP workflows in the HPC systems
    - Demonstrated the capability of leveraging parallel I/O libraries to write HEP data into HPC native backends like HDF5 (CHEP 2023)
- Successful completion of first iteration
  - HEP-CCE evolved as a base program and expanded its scope

Storage Challenge of the Upcoming HEP Experiments





- Available storage resources can limit the physics reach of HL-LHC era experiments.
- Both ATLAS (left) and CMS (right) require significant research and development efforts to address the storage crisis

HEP-CCE : A Base Program

- New areas of focus to address the requirements of HEP experiments
  - Challenge related to connecting HPC systems with the HEP experiments
    - Leverage the experience gained on first iteration to explore new challenges of future HEP experiments
  - **Challenges of data storage and data management for the future HEP experiments**

- Areas of Efforts
  - Portable Workflows
    - Develop portable workflows that can cover different use cases of future HEP experiments
  - AI/ML applications on HPC platforms
    - Scaling of selected suite of large-scale ML models in the HPC systems
  - Accelerating HEP simulation
    - Use experience from first iteration in accelerating MC simulation using GPUs
  - **Optimizing Data Storage and Data Management (This Talk)**
    - Address the storage challenge of the future HEP experiments by investigating new storage backends and data volume reduction methods

ROOT , TTree and HEP Experiments

- Open source framework 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 last two decades
  - Over Exabyte of data stored in TTree format
- TTree evolved to address experimental needs
  - TTree has been the backbone of HEP computational workflows
  - Supports persistence and I/O of complex experimental data
    - Decades of development to manage HEP complex data needs

- However, TTree architecture predates recent overhaul in C++, modern programming paradigms and evolving computational landscape
  - New storage backend required to enable future HEP experiments to address their computational challenges → RNTuple

RNTuple: Storage Backend for Upcoming HEP Experiments

- RNTuple →New Storage backend in ROOT version 7
- RNTuple and upcoming HEP experiments
  - State of the art, HEP community supported storage and I/O subsystem
  - Address storage & I/O requirements of upcoming HEP experiments
  - Compared to TTree, provides limited data model supports to save on storage
    - ATLAS and CMS report 20-40% saving in their storage (Link)
  - Use of modern C++ standards
    - Adoption of smart pointers, better error handling mechanisms, modern C++ libraries
- HEP experiments have to adopt RNTuple
  - Adopt new RNTuple API
  - May have to change the data model to be persisted in RNTuple
  - HEP-CCE will aid HEP experiments to adopt RNTuple
  - HEP-CCE has been conducting RNTuple API review (Link)
    - Aid the evolution of RNTuple as per the experimental requirements and vice versa

- Bottom line→Future experiments will have to adopt RNTuple to stay state-of-art in the ROOT ecosystem.

Data Models for upcoming HEP experiments

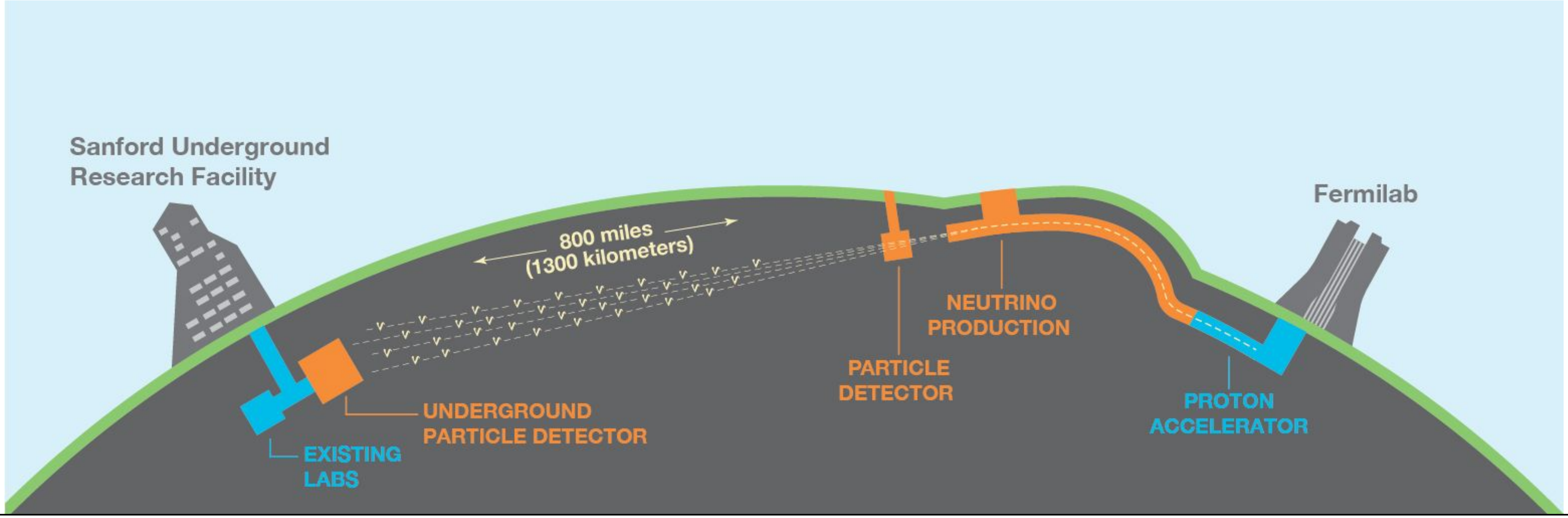
- Future HEP data models have to be:
- HPC Friendly
  - Offloadable into the GPU with little to no modifications
  - Persists in a HPC native storage backend

- Complex C++ HEP data models do not meet these requirements typically

- ROOT State of the Art
  - Persists in RNTuple storage backend

- HEP-CCE and HPC friendly data model design efforts
  - One of the areas of study in second iteration of HEP-CCE
  - Data models of future HEP experiments as candidate to make them HPC friendly
  - Investigate the persistence of data models in RNTuple
  - Generalize the outcome and communicate the deliverable to HEP experiments

DUNE: An Upcoming Intensity Frontier HEP Experiment



- Deep Underground Neutrino Experiment (DUNE): Major Physics Goals
  - Resolve Neutrino Mass Hierarchies
  - Precise Measurement of the delta CP violation in lepton sector
- Based in US and start by the end of this decade.
- Thousands of scientists and engineers from all over the world
- Large event size (Many GBs of Beam Induced and hundreds of TBs from Supernova)
- Tens of PB/year of raw data to be collected (Link)
- Plan to use HDF5 for raw data and ROOT for reconstructed data storage

Proto-DUNE Raw Data and HPC Friendly Data Model

- DUNE detectors use LArTPC technology
  - Generates image like data
  - HPC hardwares are well equipped to analyze image like data
  - DUNE will utilize HPC resources for data production to physics analyses
- Proto-DUNE: Demonstrator experiment for DUNE's LArTPC detectors
  - Simple data model
  - Data written in HDF5 which is a HPC native backend

Attributes to describe metadata

```
Raw Data Group {
        Dataset Fragment 1
        Dataset Fragment 2
        ..
        ..
         }
```

Rough storage layout of Proto-DUNE raw data in HDF5

- Use of HDF5 attributes
- Raw Data is grouped together as Fragments
- Each fragment corresponds to a detector part
- Each fragment consists of payload and headers

Proto-DUNE Raw Data as HPC Friendly Format

- HPC friendly data model design based on survey conducted by HEP-CCE
  - Structure of Arrays (SoA) like design as one of the approaches adopted by HEP experiments to make their data GPU friendly
- Proto-DUNE raw data as SoA
  - Toy MC to create fake Proto-DUNE raw data
  - Use of preprocessor macros to reorganize raw data (Fragments) as SoA
  - Test the persistence of data as SoA in RNTuple

```
struct ProtoDUNERawData {
    uint32_t Fragment1 [frag1_size];
    uint32_t Fragment2 [frag2_size];
    …
    …
    uint32_t SomeScalar;
    …
};
```

Macro reorganizes raw data into SoA

```
==============================================================
NTUPLE:      NTuple
Compression: 404
--------------------------------------------------------------
  # Entries:          10
  # Fields:           23
  # Columns:          11
  # Alias Columns:    0
  # Pages:            61046
  # Clusters:         6
  Size on storage:    79745530 B
  Compression rate:   50.16
  Header size:        391 B
  Footer size:        311279 B
  Meta-data / data:   0.004
```

Proto-DUNE Raw data as SoA in RNTuple

DUNE Analysis Data Format

- DUNE uses Common Analysis Format (CAF)
  - Resolution and size of DUNE detectors → Detailed information, intricate data structure
    - Poses problem for analyzing data with ease and speed

- CAF Data Model
  - Commonly written in ROOT::TTree
  - Later optimized for HDF5
  - Simpler object oriented with multiple level of hierarchies and segmentation
  - Data organized in columnar table format
  - Discard hit by hit (detector level) information with intricate structure
  - Higher-level reconstructed variables from hits are saved for further analysis

- Data Model shared by all neutrino oscillation experiments
  - Upcoming experiments like DUNE (and SBN experiments)
    - CAF should persist in modern ROOT ecosystem that includes RNTuple

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

Conclusions and Future Works

- Demonstrated Proto-DUNE raw data can be written in GPU friendly format
  - Applied lessons learnt in CCE first iteration to adopt SoA like design to make data GPU friendly
  - Showed the persistence of raw data as SoA in RNTuple
  - Future works
    - Look at further optimization of data models for offloading into the GPUs
- Demonstrated the persistence of CAF data model in RNTuple
  - Future works
    - Investigate I/O support in RNTuple
    - Investigate CAF objects ownership in RNTuple
    - Develop selective reading of CAF objects using RNTuple
    - Write CAF data as SoA
- Examples and test frameworks as deliverables for HEP experiments
  - Simple and standalone examples and frameworks to demonstrate
    - Persistency of HEP data model in RNTuple
    - HPC friendly design of HEP data model and persistence in RNTuple
    - Framework designed for heterogeneous computing architectures

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)