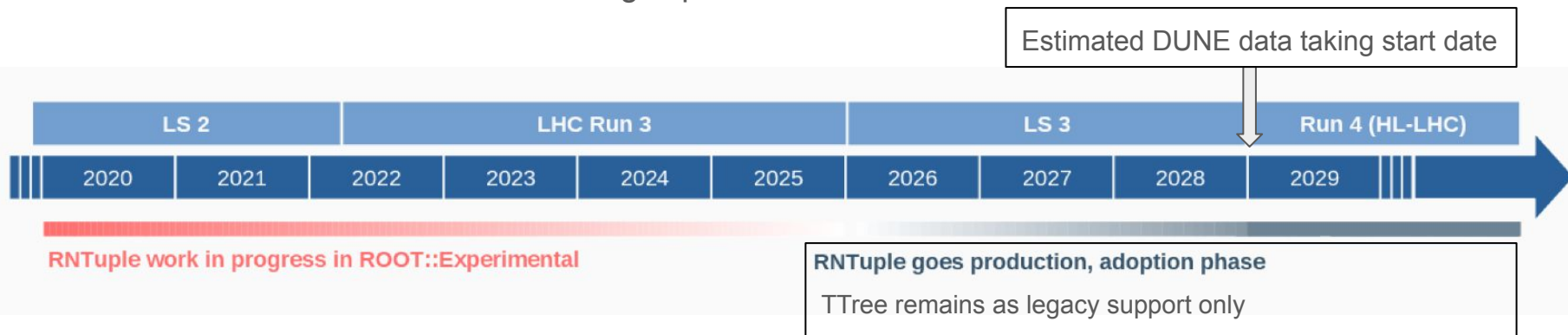


(DUNE Data Processing) Framework and RNTuple

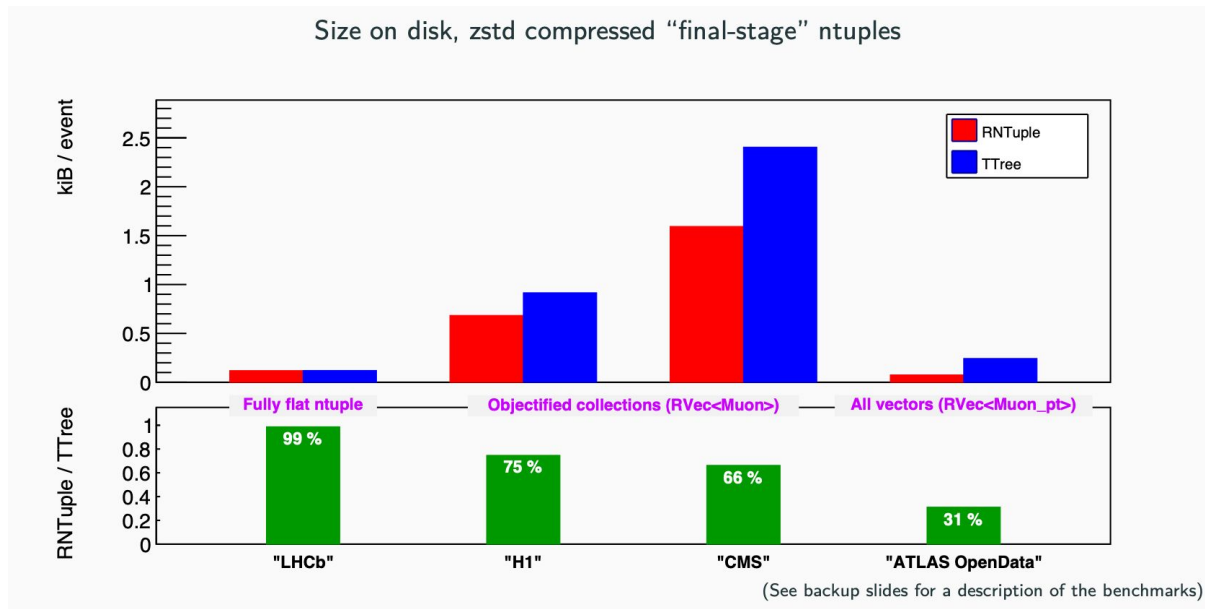
RNTuple Introduction

- I/O subsystem that will replace the current TTree in ROOT.
- Designed for optimized storage and throughput
- Experiments need to adopt RNTuple to keep using state of the art ROOT ecosystem.
 - More relevant for future looking experiments like DUNE



*Taken from [Jacob et al](#)

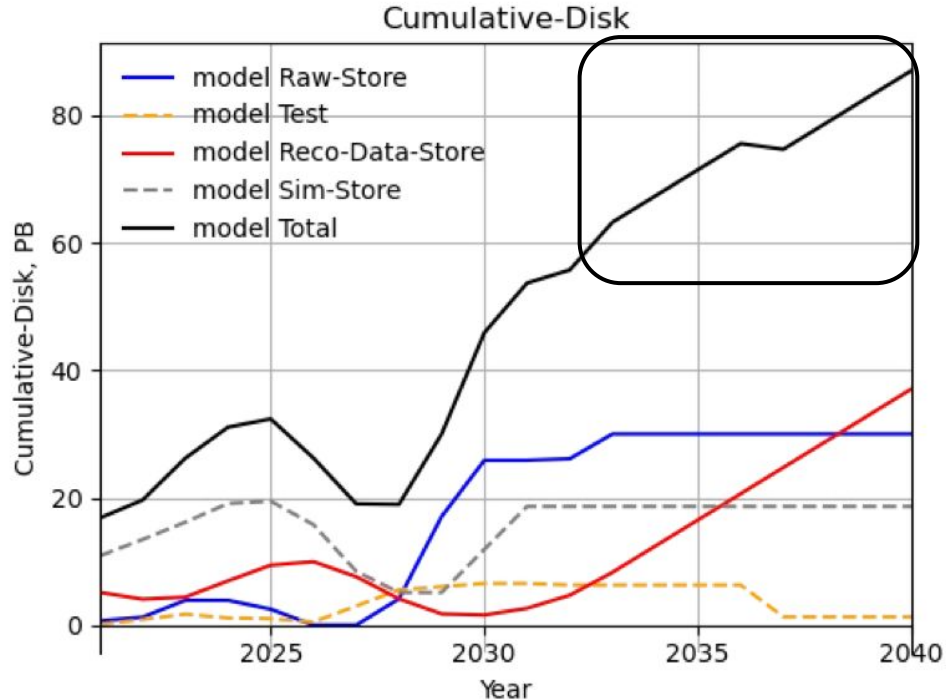
RNTuple Performance : File Size



*CCE work showed up to 30-40% storage saving with DUNE CAF Data persisted in RNTuple.

Note that storage saving depends upon data format/layout and event content. We have not looked at DUNE's data format (savings could be big or...maybe no improvement at all....one more reason for more proactive with RNTuple)

DUNE (Estimated) Storage Requirements



Taken from LBNC 2023 ([Link](#)). Numbers might have changed since then

- CPU Requirements ~2-10% of ATLAS or CMS
- Storage Requirements ~10-20% of CMS or ATLAS.
- DUNE also needs to think about storage savings.
- After 2030s, increase in disk needs will be coming from non-raw data (that will be written in RNTuple)
- Projections here are not based on RNTuple.
- Saving similar to ATLAS or CMS (previous slide) could be huge for DUNE.

I/O Using RNTuple

- For optimized storage and I/O RNTuple has limited support of EDM (compared to TTree)
- Avoid nested classes, pointers, polymorphism in data models if possible.
- Reading and Writing takes place via different APIs.

Writing API

- RNTupleWriter
- RNTupleParallelWriter
- RNTupleWriteOptions

Reading API

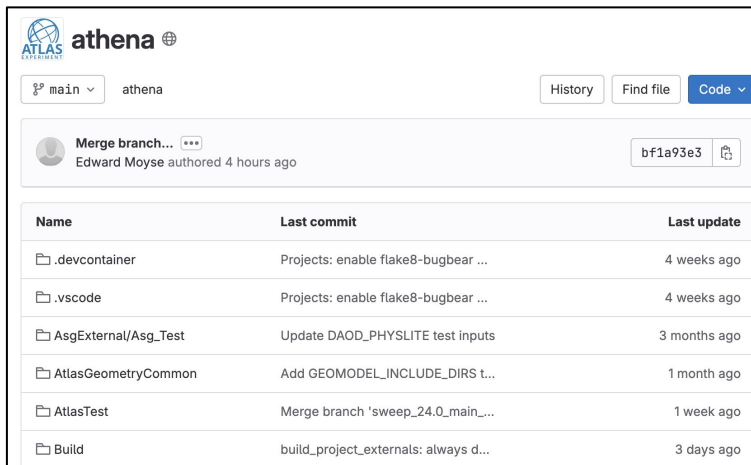
- RNTupleReader
- RNTupleReaderOptions
- RNTupleView

RNTuple API Review

- ROOT and CCE-SOP are currently doing the [API review](#) of the RNTuple
 - Experiments adopt RNTuple in their frameworks
 - Provide feedback to the ROOT developers
 - Feedbacks are integrated in RNTuple API to make RNTuple more streamlined with the experiment frameworks
 - ATLAS and CMS are heavily involved
 - CCE is looking at CAF data model used by DUNE in a limited capacity
 - **DUNE, as one of the major stakeholder needs to be more involved in RNTuple API review**

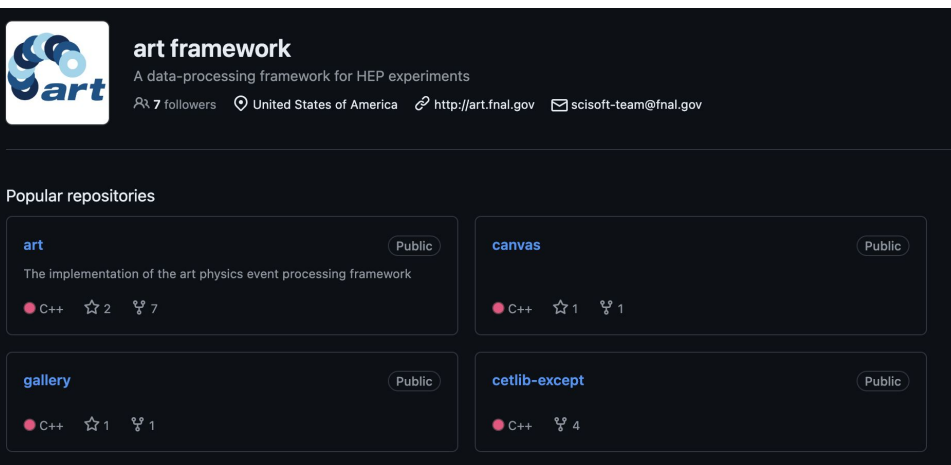
Code Management and Structure for New Event Processing Framework

- Past Experiences
 - *art* : Different components hosted by different repositories under a common suite (Multi-repo)
 - Athena/CMSSW : Framework and its components under single repository (Mono-repo)
- Components of framework like Persistence or Accelerator Integration part
 - Management that gives privileges to the lead author while other authors contribute
 - Contributions as PR. Merge after review
 - Set up automated test build against (latest) rest of the framework periodically



The screenshot shows the GitHub interface for the 'athena' repository. At the top, there's a navigation bar with 'main' selected, 'athena' as the repository name, and buttons for 'History', 'Find file', and 'Code'. Below this, a commit message 'Merge branch...' by Edward Moysé is shown. The main content is a table of repository files and folders.

Name	Last commit	Last update
devcontainer	Projects: enable flake8-bugbear ...	4 weeks ago
.vscode	Projects: enable flake8-bugbear ...	4 weeks ago
AsgExternal/Asg_Test	Update DAOD_PHYSLITE test inputs	3 months ago
AtlasGeometryCommon	Add GEOMODEL_INCLUDE_DIRS t...	1 month ago
AtlasTest	Merge branch 'sweep_24.0_main_...	1 week ago
Build	build_project externals: always d...	3 days ago



The screenshot shows the GitHub interface for the 'art framework' repository. It features the repository name, a description 'A data-processing framework for HEP experiments', and statistics like 7 followers. Below, a section titled 'Popular repositories' lists several sub-repositories: 'art', 'canvas', 'gallery', and 'cetlib-exception', each with its own statistics and public/private status.

art framework
A data-processing framework for HEP experiments
7 followers | United States of America | http://art.fnal.gov | scisoft-team@fnal.gov

Popular repositories

- art** (Public) | C++ | 2 stars | 7 forks | The implementation of the art physics event processing framework
- canvas** (Public) | C++ | 1 star | 1 fork
- gallery** (Public) | C++ | 1 star | 1 fork
- cetlib-exception** (Public) | C++ | 4 forks

Code Management and Structure for DUNE Event Processing Framework

- DUNE's framework is still in R&D
- Different components being developed by different institutions with FNAL taking the leadership
- Code Management Goals:
 - Make sure that all the efforts are in sync with one another.
 - Allow effective contributions from co-authors in the development
 - Allow testing of each components that could depend on other components and core libraries
 - Make sure different efforts are not overlapping one another (there are tools)
 -
- Code Management Strategies
 - Learn from the past experience (*art*, Athena, CMSSW)
 - Communication to come up with an optimal (and pragmatic) strategy
 - Many tools and resources for code management available that did not exist when *art*, ATHENA or CMSSW were being written.
 - Open to other code management platforms
 - ATLAS has changed its platform 3 times (roughly every 7 years) since its inception.

DUNE Requirements and RNTuple

- ATLAS and CMS are heavily involved in RNTuple API Review
- ROOT is aware of ATLAS/CMS requirements
 - Both experiments have submitted suggested API changes/additions for better integration with their frameworks
- DUNE is blind about its framework, data product design.
 - We need to be more involved in RNTuple API review
 - Need to figure what we need and what are already there
- Our absence will make RNTuple friendlier to collider experiments but....
 - But they have different requirements compared to DUNE
- An Example:
 - Basketsize (in Tree) → PageSize (in RNTuple)
 - Certain APIs might be optimized for collider experiments which would not work for DUNE (or even worse would make data processing/persistence complex)

Thank You

The *art* Framework (Important components)

[art_root_io](#) : Manages the I/O of the *art* objects.

[canvas/Persistency](#): Persistence of the *art* objects

[Services](#) : Interaction between *art::Objects/Files* and Framework

art::Event is the fundamental unit of information in the *art* Framework.