



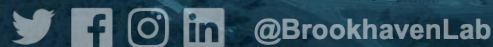
ATLAS RNTuple Latest News

RNTuple Workshop, 2-3 Dec 2024 CERN

Marcin Nowak (BNL)

Peter Van Gemmeren, Alaettin Serhan Mete (ANL)

Tatiana Ovsiannikova (University of Washington)



Current Functionality

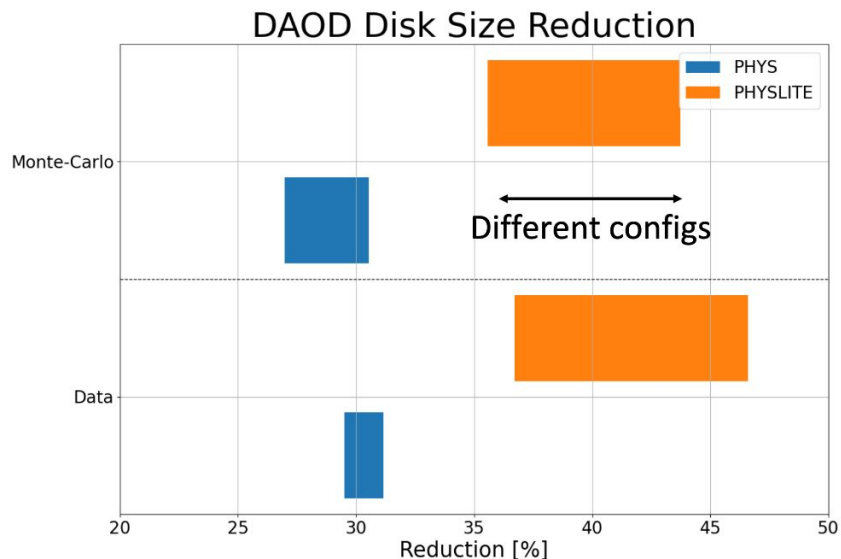
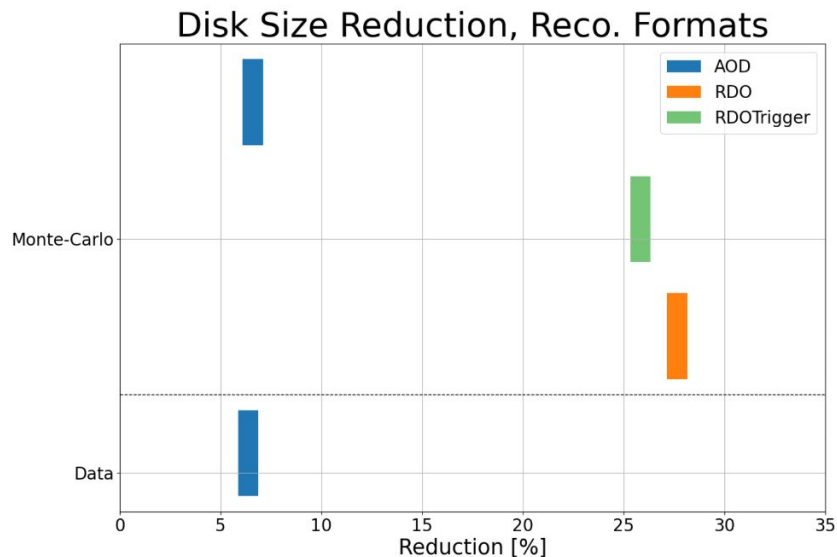
- Athena is the ATLAS production framework for event processing
 - from RDO to DAOD
- Current release (main branch) fully integrated with RNTuple storage technology
 - No functional difference between RNTuple and TTree based jobs^(*) and files
 - All data products (except RAW) are covered
 - (*) One of the workflow configurations is not yet ready - see Peter's talk
 - Using ROOT 6.34 in **dev** and 6.32 in **main** (considering an upgrade soon)
 - Git Athena CI tests (in main) execute a couple of RNTuple unit tests with every MR
 - Nightly ART tests include 8 full event processing RNTuple jobs
- RNTuple support is now a prototype and all RNTuple files considered temporary, for testing
 - We have done performance tests, but not on a very large scale
 - We are preparing for larger production using ATLAS Open Data

The Last Year

- ATLAS thanks CCE for facilitating the API review and for the ROOT team's collaboration
 - Most, if not all, core ATLAS requirements are addressed before/during the API review:
 - e.g., model updating (late attributes), emplacing new values (generating default objects), etc.
 - Still, it was a good opportunity to see what the others are using and cross-check
- Over the course of the API review, we made a number of improvements
 - Using the proper API, e.g., RNTupleReader → RPageSource → RNTupleReader
 - Adopting API changes that result from the review, as well as internal developments
- Focus mainly on the functionality (workflows)
 - But we also looked at a number of optimization aspects:
 - Split vs non-split encoding of fields (more later), etc.
- Analysis adoption of RNTuple has started and seems to be progressing well
 - Some reading code already in the repository

Performance - File Sizes

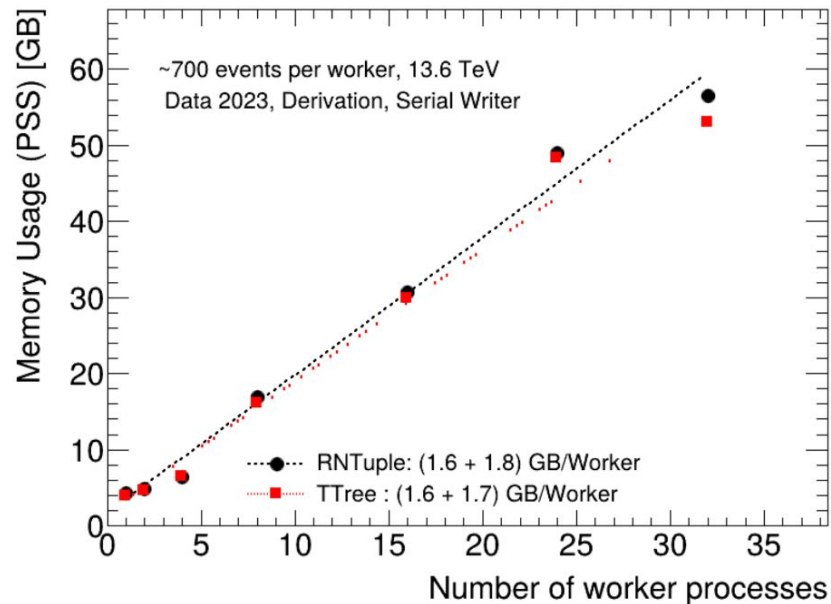
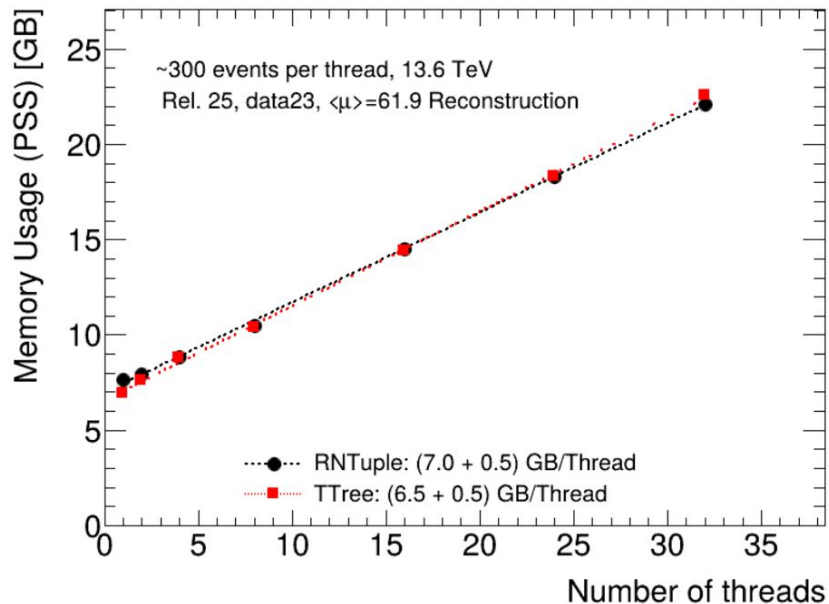
- Really impressive size reduction seen across different data products:
 - Although some data types showed opposite tendencies caused by splitting (see slide #6)



From Tatiana's presentation at CHEP 2024

Performance - Memory Usage

- A small memory consumption increase in multi-process jobs:



From Tatiana's presentation at CHEP 2024

Compression - Split/Unsplit Columns

- Specific columns show a big difference in compression efficiency depending on if they were byte shuffled (split) or not:

TTree

```
* ..... *
*Br 515 :HLTNav_Summary_DAODSlimmedAuxDyn.decisions : *
*      | vector<vector<unsigned int> > *
*Entries : 100000 : Total Size= 239180609 bytes File Size = 35748601 *
*Baskets : 1980 : Basket Size= 131072 bytes Compression= 6.69 *
* ..... *
```

RNTuple

```
HLTNav_Summary_DAODSlimmedAuxDyn:decisions._0._0 [#0] -- SplitUInt32{id:768}
# Elements: 53217539
# Pages: 636
Avg elements / page: 83675
Avg page size: 94475 B
Size on storage: 60086325 B
Compression: 3.54
```

Default (Split)

```
HLTNav_Summary_DAODSlimmedAuxDyn:decisions._0._0 [#0] -- UInt32{id:768}
# Elements: 53217539
# Pages: 661
Avg elements / page: 80510
Avg page size: 34064 B
Size on storage: 22516354 B
Compression: 9.45
```

Unsplit

Missing Functionality (or To-Do List)

- On-the-fly fast merging of RNTuples:
 - What exists now for RNTuple: merging of existing/finalized files
 - What we now use with TTrees: a networked server merging TFiles produced by many concurrent worker processes
- Features needed:
 - Incremental merging - the server keeps the output open and adds incoming data as it is produced by the clients
 - Resetting objects to avoid adding the same data (rows) again
 - Setup based on: [parallelMergeServer](#) and [parallelMergeClient](#)
 - ATLAS actual [example](#) - simplified but with branch synchronization

```
TFileMerger::PartialMerge(TFileMerger::kIncremental | TFileMerger::kResettable)
```

- Indexing
 - Additional indirection through an index allows keeping track of ordering of rows, even when they are merged in a random fashion
- Friend RNTuples
 - For adding extra data to existing RNTuples (event augmentation)

Conclusions

- Current RNTuple implementation provides sufficient functionality for ATLAS
 - There are still desired missing functionality
- Performance tuning and optimization is ongoing
- Next steps for ATLAS involve:
 - Addressing the missing functionality, especially for derivation production (fast merge)
 - Implement control for splitting/not-splitting of individual fields
 - Profiling the performance in all aspects and optimizing the workflows
 - E.g., reproduce all DAOD PHYSLITE OpenData in RNTuple v1.0 for 1-to-1 comparisons w/ TTree
- It was a productive year but more work remains