Impact of RNTuple on Storage Resources for ATLAS Production

Tatiana Ovsiannikova[1,*], *Alaettin Serhan* Mete[2], *Marcin* Nowak[3], and *Peter* Van Gemmeren[2] on behalf of the ATLAS Computing Activity

[1]University of Washington, Seattle, US
[2]Argonne National Laboratory, Chicago, US
[3]Brookhaven National Laboratory, New York, US

Abstract. Over the past years, the ROOT team has been developing a new I/O format called RNTuple to store data from experiments at CERN's Large Hadron Collider. RNTuple is designed to improve ROOT's existing TTree I/O subsystem by improving I/O speed and introducing a more efficient binary data format. It can be stored in both ROOT files and object stores, and it is optimized for modern storage hardware like NVMe SSDs. The ATLAS experiment plans to use RNTuple as its primary storage container in the upcoming High Luminosity-LHC (HL-LHC) data taking. There has been significant progress in integrating RNTuple into the ATLAS event processing framework, and now all production ATLAS data output formats support it. Performance studies with open-source data have shown substantial improvements in space resource usage. The reported study examines the I/O throughput and disk-space savings achieved with RNTuple for various ATLAS data output formats. These measurements will have an important impact on the computing resource needs of the ATLAS experiment for HL-LHC operation.

1 Introduction

The LHC is entering the High-Luminosity phase, during which experimental statistics will increase drastically. During this period, the ATLAS Experiment [1] aims to collect ten times more data than in the previous data-taking period [2]. To address the challenges posed by HL-LHC, ROOT [3] has introduced a new data storage format called RNTuple, which offers significant advantages over the widely used TTree format [4]. RNTuple employs modern and efficient approaches, utilizing the latest C++ features and improvements such as parallel and asynchronous I/O. The new format is expected to improve data throughput and reduce file size compared to TTree. Furthermore, the new format is fully compatible with ROOT's existing I/O infrastructure. RNTuple also provides robust interfaces designed for ease of use, multithreading, and GPU-driven workflows.

Over the past few years, the ATLAS software team, in collaboration with the ROOT team, has developed a new API within the Athena software to support storing all ATLAS production formats storage using RNTuple [5]. ATLAS's typical data processing chain comprises multiple steps, as shown in Figure 1. Except for RAW data, all ATLAS data are stored in

[*] e-mail: tovsiank@uw.edu

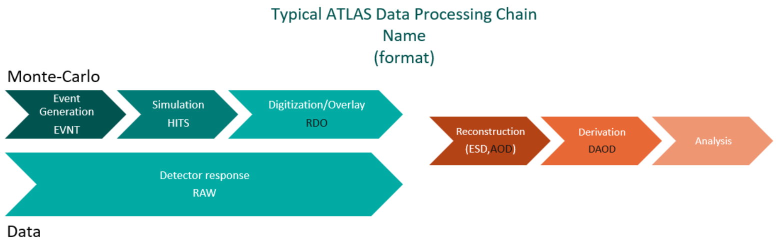

Figure 1: The ATLAS Data processing chain.

ROOT files, broadly categorized into metadata (file-level information like detector conditions) and event data (e.g., tracks, electrons, and muons) [6]. This study evaluates the impact of RNTuple on storage efficiency for the AOD, RDO, RDOTrigger, and DAOD (PHYS and PHYSLITE) formats using real and simulated data. In production workflows, metadata only accounts for a tiny fraction of storage and I/O resources, therefore the study presented here will focus on event data.

2 Storage Comparison Between TTree and RNTuple

The first estimates of size reduction when switching from TTree to RNTuple have been presented by both the ROOT team and ATLAS, limited to ATLAS open data on the DAOD format. ROOT demonstrated significant compression gains using open-source ATLAS data, of up to 40 % [7], while ATLAS reported 20 % size savings for a particular version of their DAOD [5]. This work extends the analysis to reconstruction production formats and multiple DAOD configurations.

Figure 2(a) shows the size reduction for RNTuple compared to TTree for 2023 ttbar data and the corresponding Monte Carlo (MC) reconstruction formats (10k events). The reduction for RDO reaches 20–27 %, whereas for AOD, it is limited to 5–7 %. For the derivation formats, 100k events datasets with varying detector condition were used. The size reduction for DAOD formats is presented in Figure 2(b). Depending on the detector conditions the reduction can vary from 30 to 45 % for the DAOD Physlite and from 20 to 30 % for DAOD Phys.

A detailed breakdown of the inner data structure shows most containers exhibit size reduction (Figure 3), except certain nested vector branches. For the derivation MC format, the largest increase was observed in the Trigger domain (Figure 4).RNTuple's default binary so-called split storage efficiently handles small numbers, combining first, second etc. bytes together, but it struggles with nested vectors containing repetitive data. Based on our feedback the ROOT team is implementing an unsplit storage option to address this issue.

Similar patterns are observed in AOD formats (Figure 5), where a larger number of branches show size increases as shown n the Figure 6. Unlike DAOD, AOD formats contain more complex data structures, which are typically stored as auxiliary and dynamic auxiliary data in the ATLAS Event Data Model. In the default Athena TTree-based AOD format, these complex data structures can be stored within a single column (so-called branch) and compressed together. However, in RNTuple, each auxiliary attribute is stored as a separate field. As a result, for certain branches in the AOD, the original Athena storage approach with

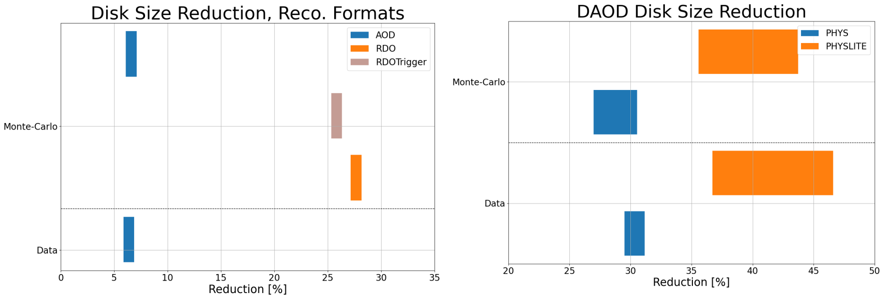

Figure 2: Size reduction using RNTuple vs TTree for MC and Data. (a) The reconstruction production formats. (b) The derivation production formats. The bar width indicates different condition and configuration options for the derivation formats.

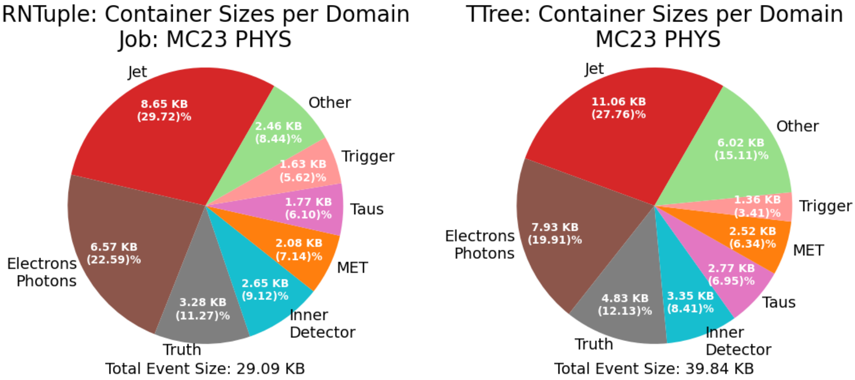

Figure 3: The detailed comparison of DAOD MC 2023 event data size difference. The size per inner TTree (a) and RNTuple (b).

TTree may perform better than RNTuple. To fully understand and address this behavior, additional systematic studies are required to identify the specific causes of these size increases and explore potential optimizations.

3 Memory Performance Studies

The GRID infrastructure imposes strict memory usage limits per worker, making it critical to ensure that RNTuple stays within these constraints. Preliminary memory usage studies were conducted for multi-threaded and multi-process modes, similar to previous Run 3 ATLAS software studies [6].

Figure 7 shows memory consumption as a function of the number of worker processes for derivation production using 2023 data. With serial file output (no parallel compression), a

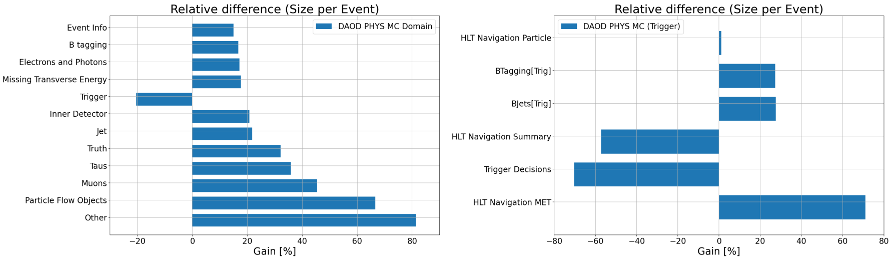

Figure 4: The detailed comparison of DAOD MC 2023 event data size difference. (a) The relative size difference between RNTuple and TTree for inner domains. (b) The relative size difference between RNTuple and TTree for branches (Trigger domain). The negative values indicate that RNTuple is larger than TTree

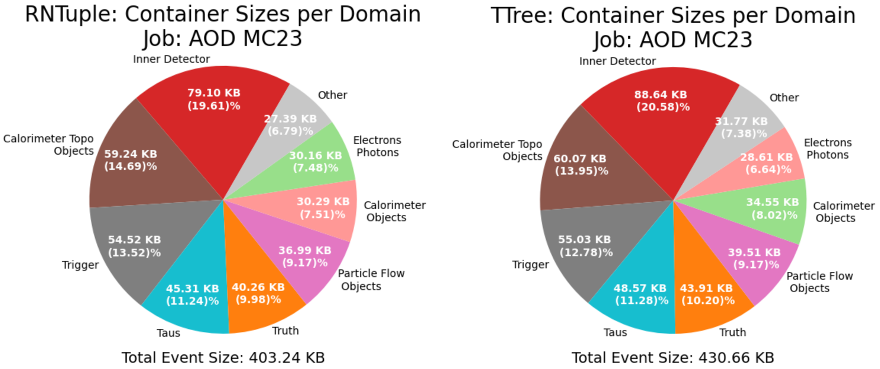

Figure 5: The detailed comparison of AOD MC 2023 event data size. The size per inner TTree (a) and RNTuple (b).

small increase in memory usage (100 MB) is observed. The observed shift in memory usage is more expected than an increase in the slope for this derivation production configuration, as the output file writing remains serial. This observation highlights the need for further systematic studies to better understand and address memory behavior. For reconstruction production, the slope increase is only a few tens of MB, but a 500 MB shift in memory consumption is seen compared to TTree. Despite modest thread/worker scaling, further optimization is necessary to minimize memory usage and comply with GRID limits.

4 Conclusion

The ATLAS software I/O API is now ready to store all main production formats using RNTuple. Significant disk size reductions have been demonstrated when switching from TTree to RNTuple for both derivation and reconstruction formats. The largest reductions

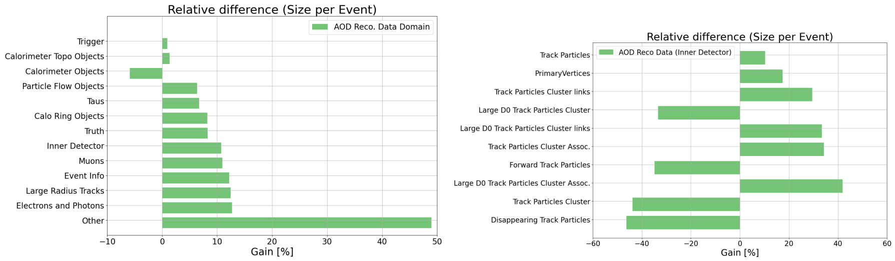

Figure 6: The detailed comparison of AOD MC 2023 event data size. (a) The relative size difference between RNTuple and TTree for inner domains. (b) The relative size difference between RNTuple and TTree for branches (Inner Detector domain). The negative values indicate that RNTuple is larger than TTree

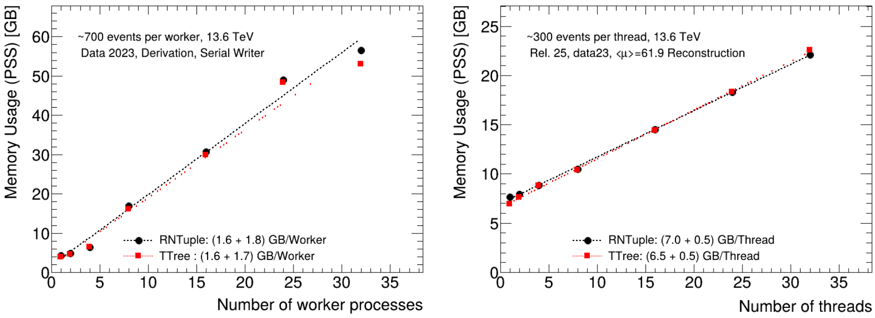

Figure 7: The memory usage as a function of workers/thread. (a) The derivation production data 2023 job. (b) The reconstruction production data 2023 job.

are observed for the RDO and DAOD formats: RDO achieves up to 27 % savings, while DAOD (PHYSLITE) shows reductions of 30–45 %, depending on detector conditions. Improvements for AOD at this point are limited to 5-7 %, however potential for even further improvement has been identified by storing some data in un-split mode (similar to current TTree settings).

Preliminary studies on memory performance show a small increase in memory consumption for both multi-threaded reconstruction and multi-process derivation configurations. While these increases are relatively minor, a better understanding of the memory behavior is needed, along with possible optimizations, to minimize resource usage and comply with GRID constraints.

Overall, the results demonstrate significant improvements in storage efficiency with RNTuple compared to the previous TTree technology. Continued investigations into memory performance and storage optimizations will ensure the successful and seamless integration of RNTuple into ATLAS's production workflows.

References

[1] ATLAS Collaboration, The ATLAS Experiment at the CERN Large Hadron Collider. JINST **3** S08003 (2008). https://iopscience.iop.org/article/10.1088/1748-0221/3/08/S08003

[2] ATLAS Collaboration, ATLAS Software and Computing HL-LHC Roadmap. CERN Tech. Rep. CERN-LHCC-2022-005, LHCC-G-182 (2022). http://cds.cern.ch/record/2802918

[3] R. Brun, F. Rademakers, ROOT - An Object Oriented Data Analysis Framework. Nucl. Inst. & Meth. in Phys. Res. A **389** 81-86 (1997). https://iopscience.iop.org/article/10.1088/1748-0221/3/08/S08003

[4] J. Blomer, P. Canal et al, ROOT's RNTuple I/O Subsystem: The Path to Production. EPJ Web of Conf. **295** 06020 (2024). https://doi.org/10.1051/epjconf/202429506020

[5] A.S Mete, M. Nowak, P. Van Gemmeren, Persistifying the complex event data model of the ATLAS Experiment in RNTuple. ACAT conference proceeding, ATL-SOFT-PROC-2024-002 (2024). https://cds.cern.ch/record/2905189

[6] ATLAS Collaboration, Software and computing for Run 3 of the ATLAS experiment at the LHC. submitted to EPJC, CERN-EP-2024-100 (2024). https://cds.cern.ch/record/2886158

[7] F. de Geus, J. López-Gómez et al, Integration of RNTuple in ATLAS Athena. EPJ Web of Conf. **295** 06013 (2024). https://doi.org/10.1051/epjconf/202429506013