Performance Comparison of Lossless Compression Algorithms on CMS Data using ROOT TTree and RNTuple

Simone Rossi Tisbeni[1], Silvio Donato[2] on behalf of the CMS Collaboration

[1]CERN-NGT, INFN and Università Bologna - [2]INFN and Università Pisa

Introduction

The CMS experiment at CERN generates massive amounts of data requiring efficient storage solutions. With Run 3 producing ~100 kHz of raw collision data and Phase 2 upgrades expected to increase data rates of one order of magnitude, optimized lossless compression algorithms is critical.

This study benchmarks compression performance across different algorithms and compares ROOT's traditional TTree format with the new RNTuple format [1] to identify optimal storage strategies for current and future CMS operations.

Dataset & Hardware specs

Run 3 RAW data: pp collision events at 13.6 TeV. <pileup>=64.5
Phase 2 simDigis: $t\bar{t}$ Monte Carlo at 14 TeV. <pileup>=200

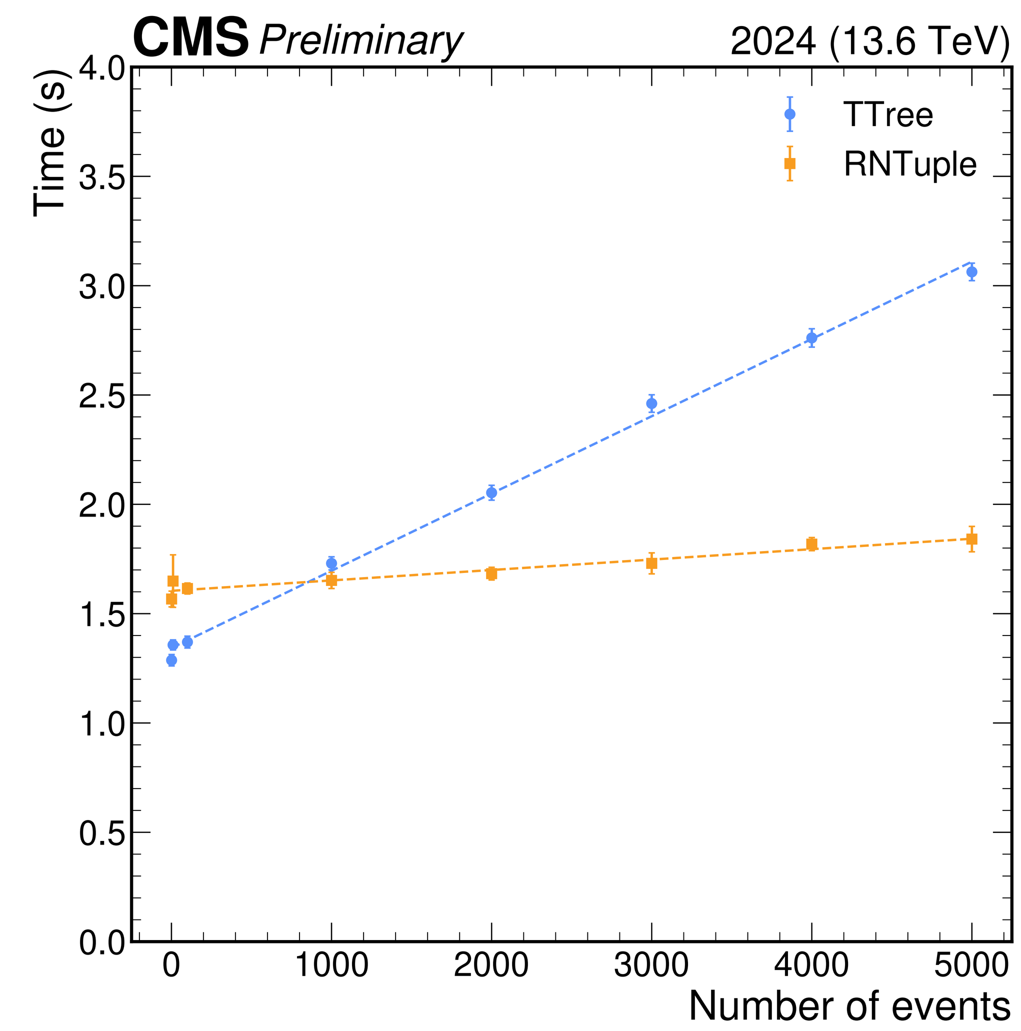
The benchmarks have been run on a node with
AMD EPYC 9534 64-cores 2.45GHz CPU and 180 GB RAM

Reading Timing

The same data was saved in:

- **TTree**: Traditional ROOT format, row-wise storage
- **RNTuple**: Experimental ROOT format, column-wise storage

Read Performance: RNTuple demonstrates significantly faster read times than TTree, with the performance gap increasing for larger event size [2].



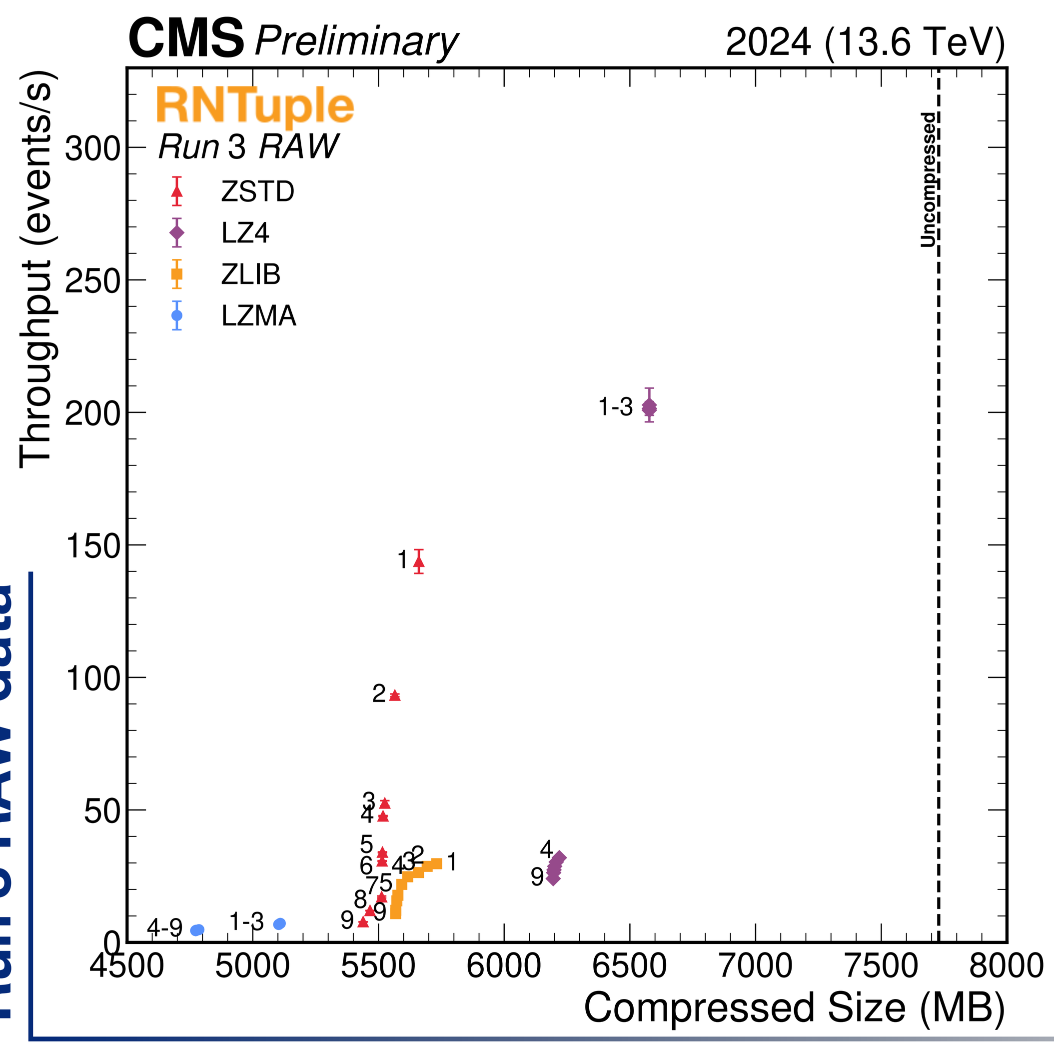
Each measure is averaged over multiple runs after caching
We evaluated four lossless compression algorithms integrated in the CMS Software Framework:

- **ZSTD**: Modern algorithm, balanced performance
- **LZ4**: Fast compression, optimized for speed
- **ZLIB**: Balanced compression ratio and speed
- **LZMA**: Max compression ratio, slow processing

Each algorithm allows varying compression level (1 weakest, 9 strongest)

Compression Comparison

Run 3 RAW data

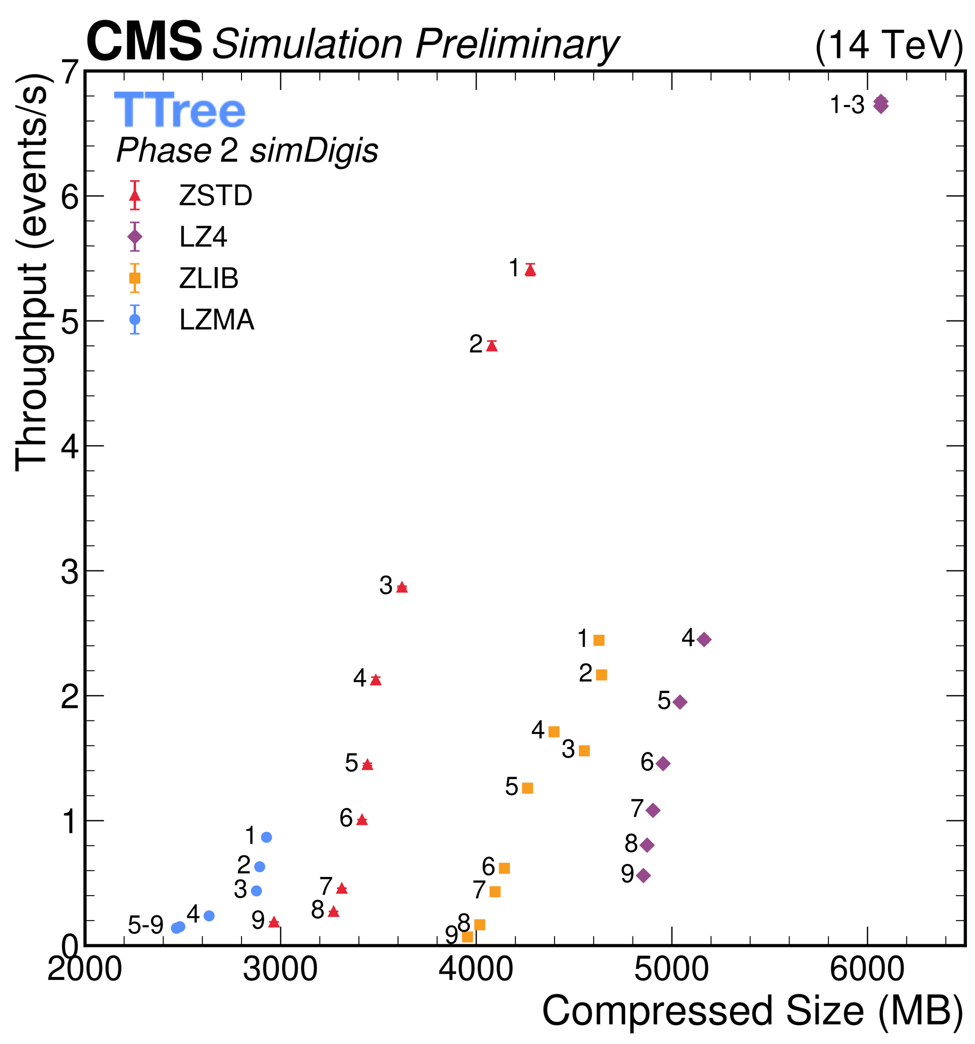


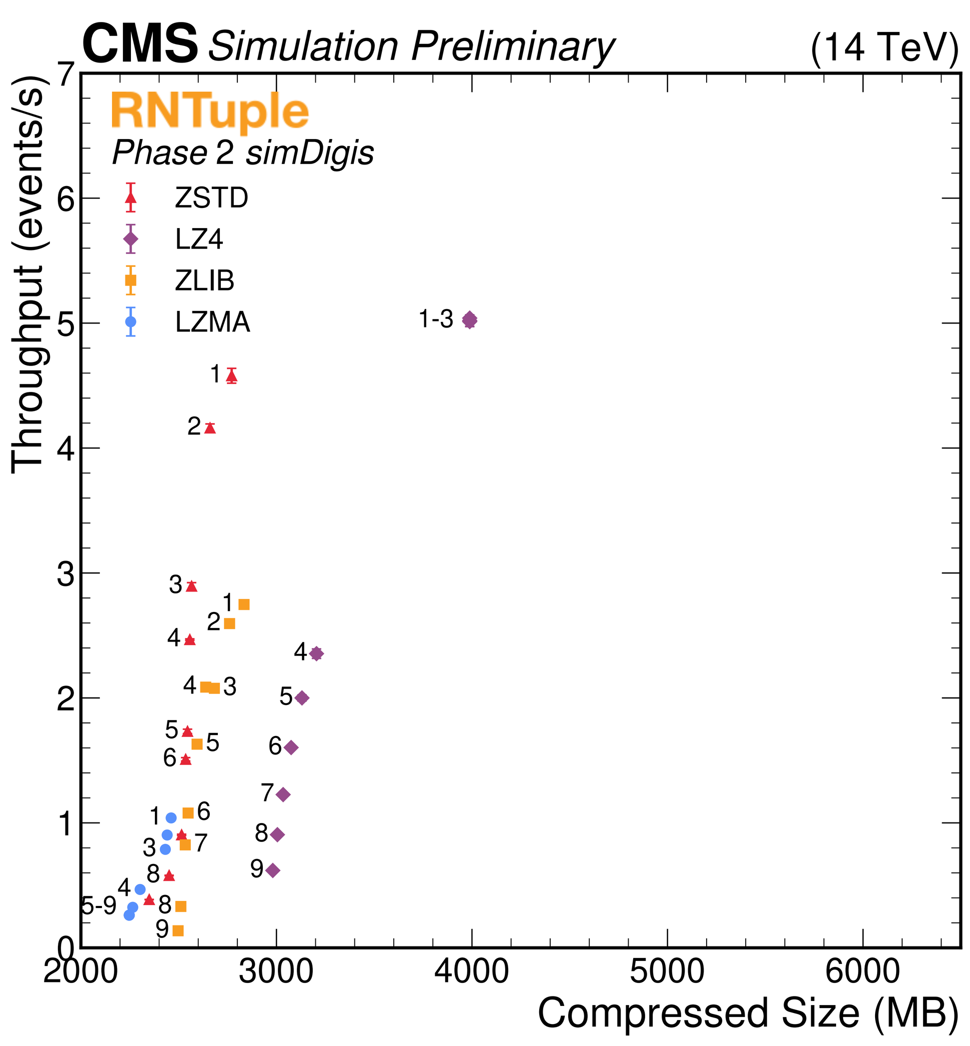
The figures show the compression throughput in **events/s** (reading+compression) as a function of the **compressed size** for each level of the four algorithms [2].

Plots show similar trends for each algorithm: **higher compression levels results in lower throughput and lower compressed size.**

For **Run 3 RAW data**, throughput and compression are comparable in both formats. The lowest levels for TTrees present a slightly higher throughput.

Phase 2 simDigi





For **Phase 2 simDigi** the throughput for TTree is higher than RNTuple for low levels, while is slightly lower for the higher levels [2].
The compression is, on average, **better for RNTuple**.

simDigis compress far better than RAW files because RAW is a **bitstream**, while simDigi stores **structured data** that lossless compressors can exploit [3].

References & Acknowledgements

[1] Jakob Blomer et al. *RNTuple Binary Format Specification 1.0.0.0* Tech. rep. Geneva: CERN, 2024. cds.cern.ch/record/2923186
[2] CMS Collaboration. *Performance Comparison of Lossless Compression Algorithms on CMS Data using ROOT TTree and RNTuple* CMS DP-2025/049 Geneva: CERN, 2025. cds.cern.ch/record/2941436
[3] Next Generations Triggers. *NGT Evaluation Report - 2024* Geneva: CERN 2024 repository.cern/records/nkwjc-7pa06

This work has been partially funded by the Eric & Wendy Schmidt Fund for Strategic Innovation through the CERN Next Generation Triggers project under grant agreement number SIF-2023-004, and by the European Union - Next Generation EU, Mission 4 Component 1 CUP I53D23001070006