ROOT I/O improvements for HEP analysis

Oksana Shadura (UNL), Zhe Zhang (UNL), Brian Paul Bockelman (Morgridge Institute for Research), Philippe Canal (FNAL), Danilo Piparo (CERN)

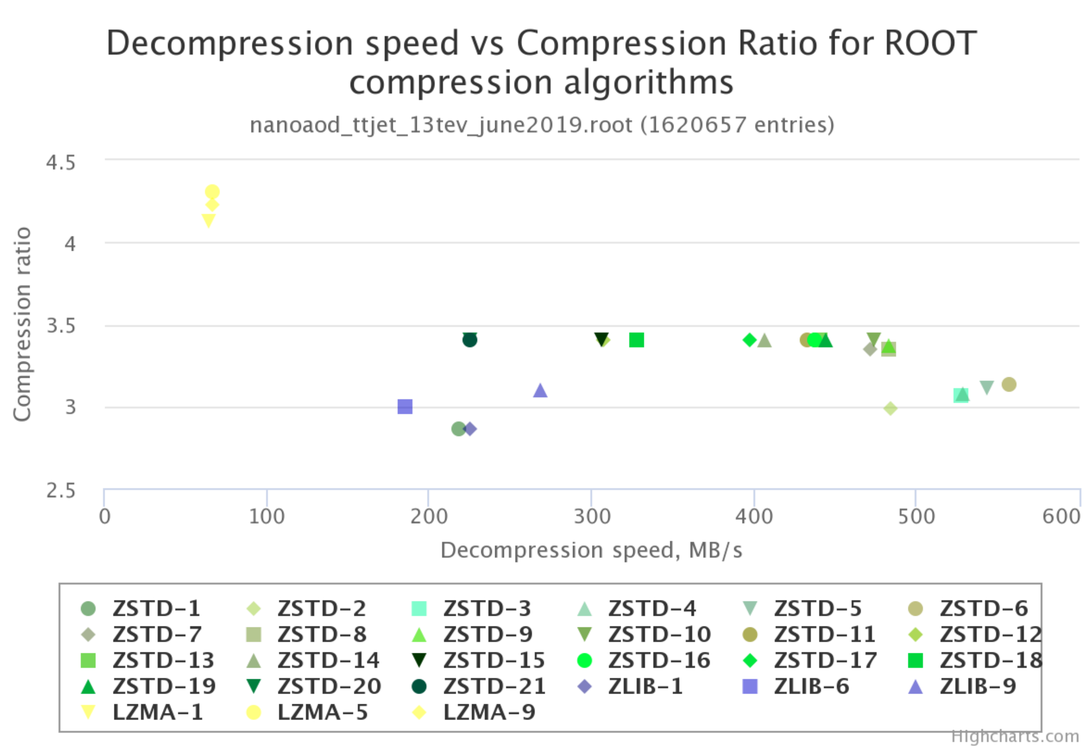
oksana.shadura@cern.ch,bbockelman@morgridge.org

Introduction

ZSTD - a dictionary-type algorithm (LZ77) with large search window and fast implementations of entropy coding stage, using either very fast Finite State Entropy (tANS) or Huffman coding. [Facebook]

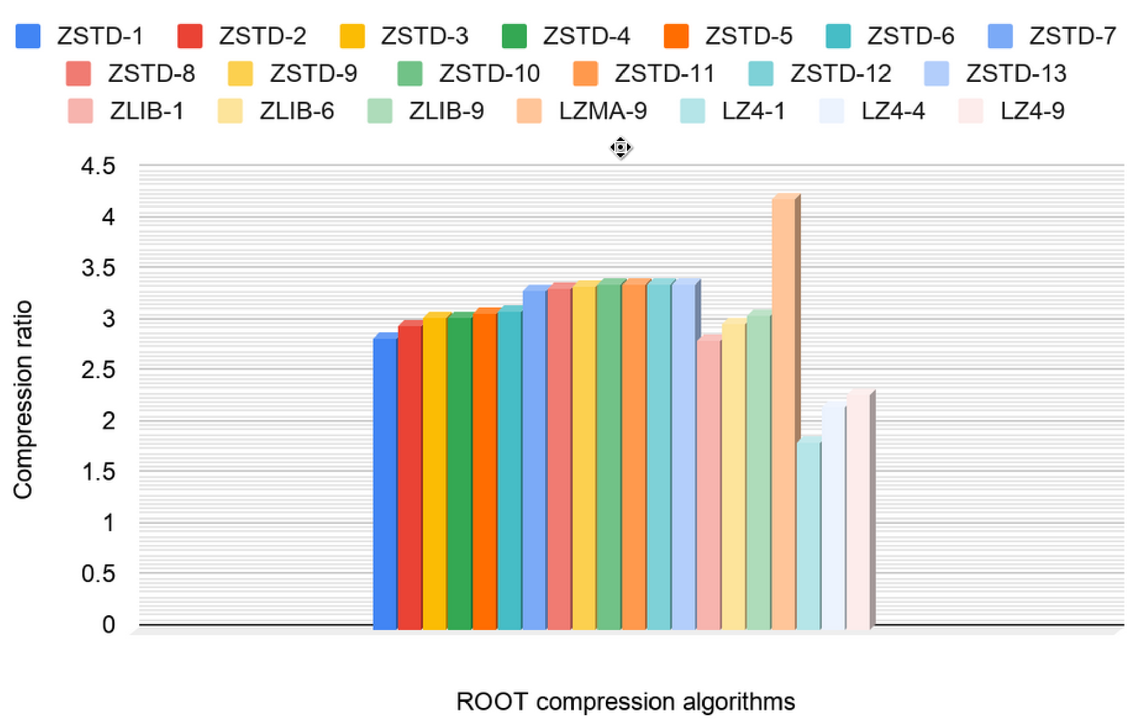
AVAILABLE IN ROOT 6.20!

ZSTD for NanoAOD (compression levels)



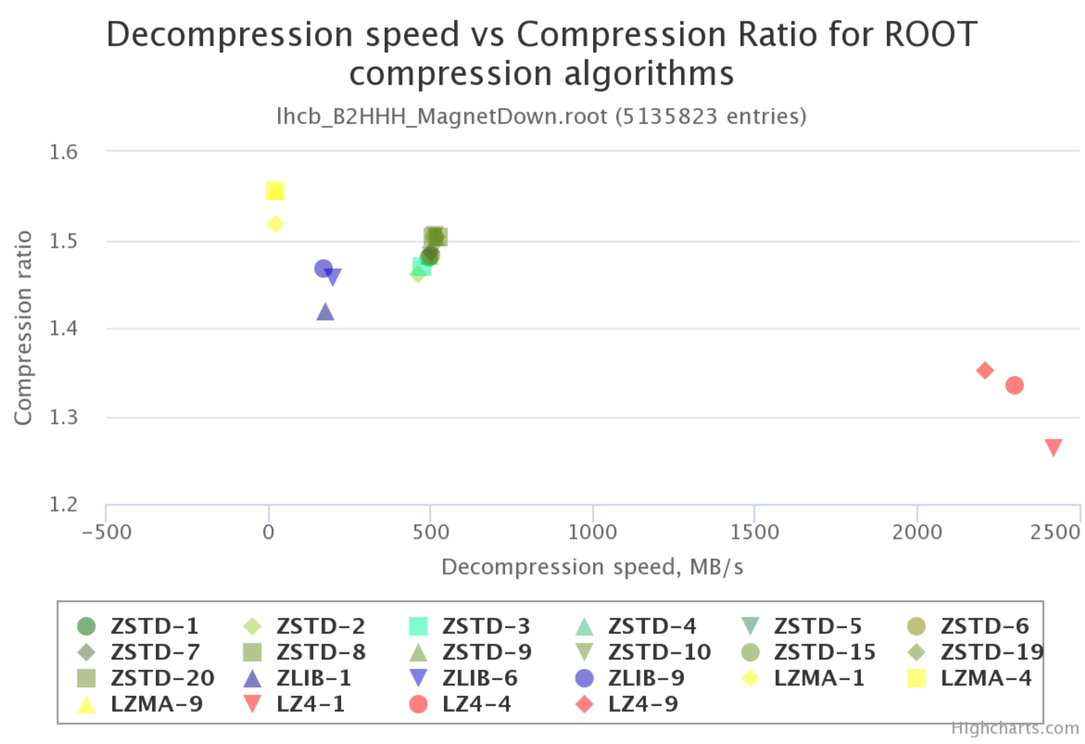
→ Size of file compressed with ZSTD is **20 % bigger**, but it is **6x faster** comparing to LZMA-5!

NanoAOD 2019 compression ratio comparison



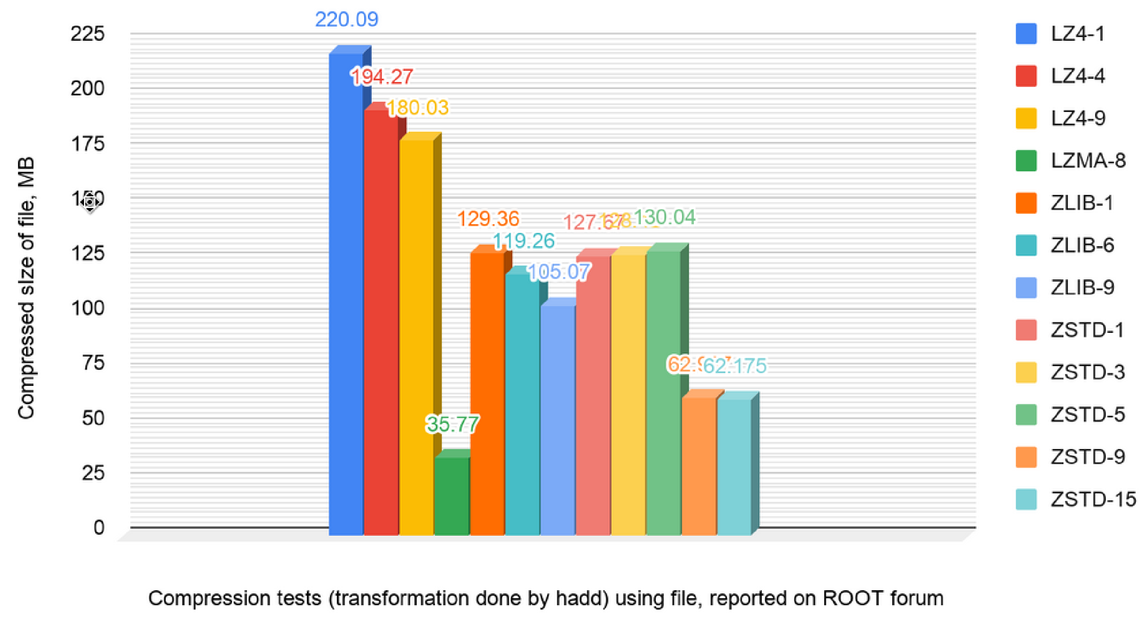
→ A bit better compression ratio then ZLIB;

LHCB compression speed vs compression ratio



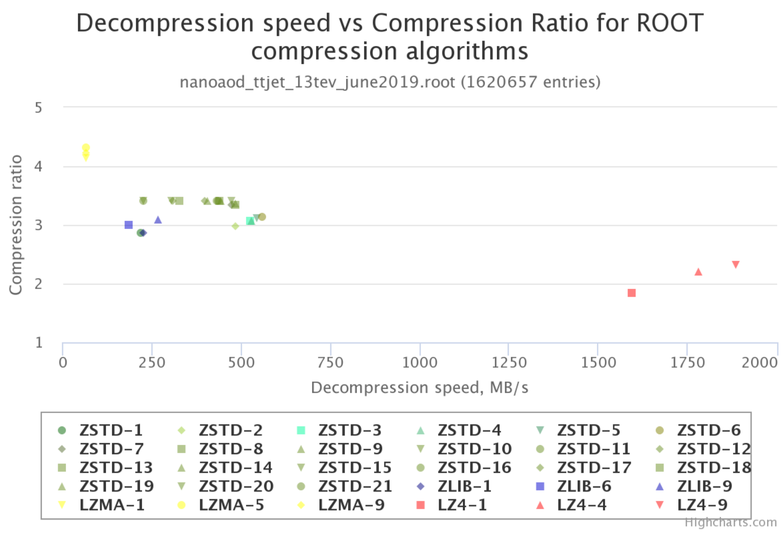
→ For the "flat tree" with a simple structure the best choice could be LZ4: **10x time faster** read speed

Data file with offset arrays



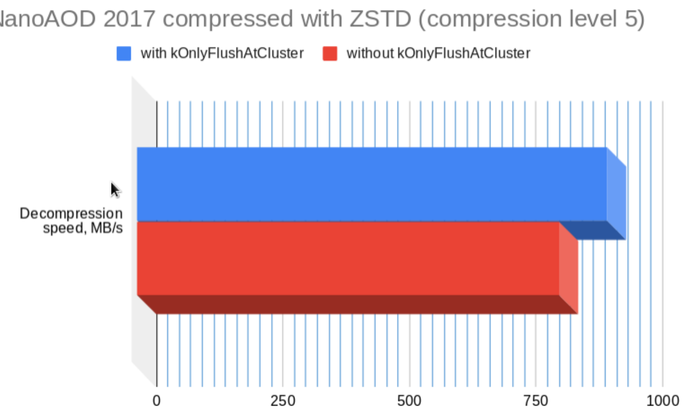
→ ZSTD has no problems with compression of data that contains the byte offset of each event in the branch data (vs LZ4);

CMSSW NanoAOD and MiniAOD (including LZ4)



→ **NanoAOD - using ZSTD** could be a better compromise **between size** of file on a disk and **decompression speed(faster analysis!)**;
→ **MiniAOD - size of file with ZSTD is 10% bigger** than using LZMA, but the time spend in decompressing on readback is **15x less**!
(big thanks to David Lange for MiniAOD measurements)

TTree::kOnlyFlushAtCluster: faster decompression



→ TTrees can be forced to only create new baskets at event cluster boundaries, it simplifies file layout and I/O at the cost of memory (**NanoAOD 2017 size difference was 3.6 %**). Recommended for simple file formats such as ntuples but not more complex data types.

tree->SetBit(TTree::kOnlyFlushAtCluster);

Acknowledgements

This work was supported by the National Science Foundation under Grant ACI-1450323.