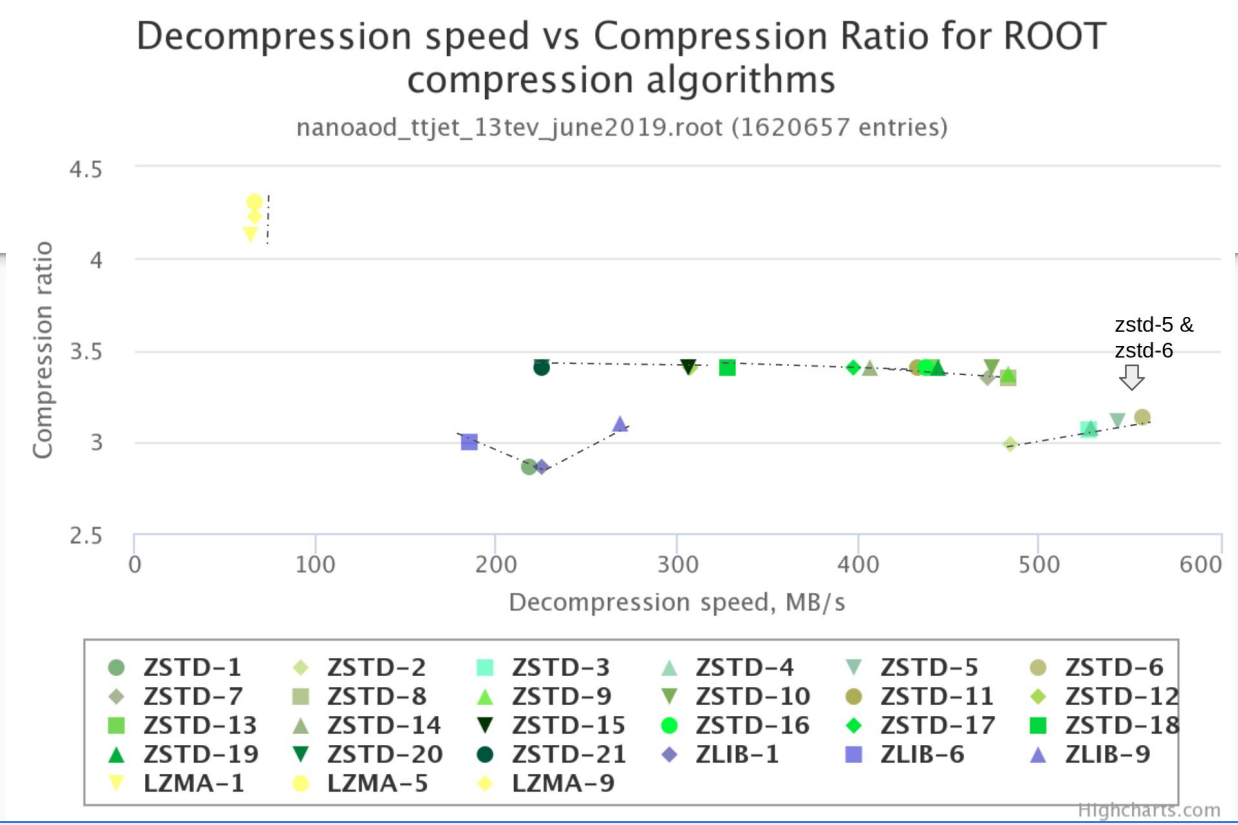
ZSTD compression algorithm in ROOT

Oksana Shadura (UNL)
Brian Bockelman (Morgridge Institute for Research)

ZStandard (or ZSTD)

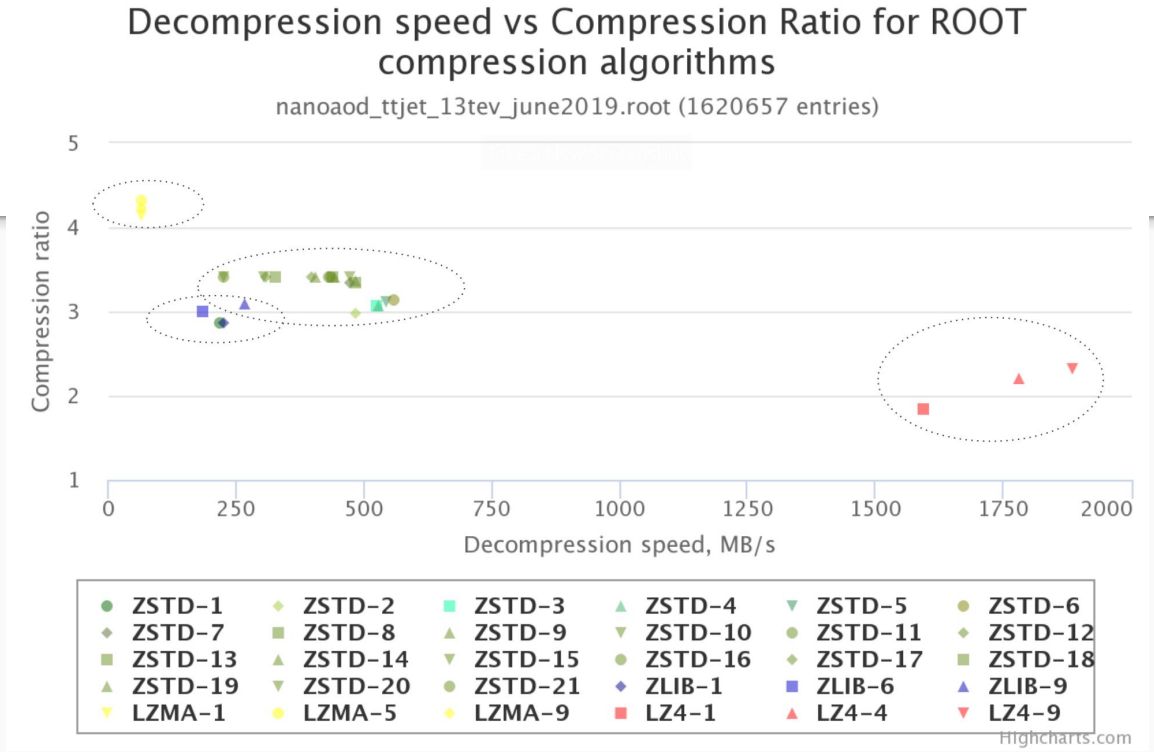
- **ZSTD** - a dictionary-type algorithm (LZ77) with large search window and fast implementations of entropy coding stage, using either very fast Finite State Entropy (tANS) or Huffman coding. [https://github.com/facebook/zstd.git]
- **ZSTD** is extremely flexible in settings
  - It has multiple compression levels (from 1 to 22) and also extra fast compression settings aka negative compression levels
- **ZSTD** will be available in ROOT 6.20.00

ZSTD for NanoAOD (different compression levels)



Better compression ratio and **2x faster decompression then ZLIB**; **6x faster** comparing to LZMA; file compressed with ZSTD *is only 20 % bigger*!

ZSTD for NanoAOD (different compression levels and including LZ4)



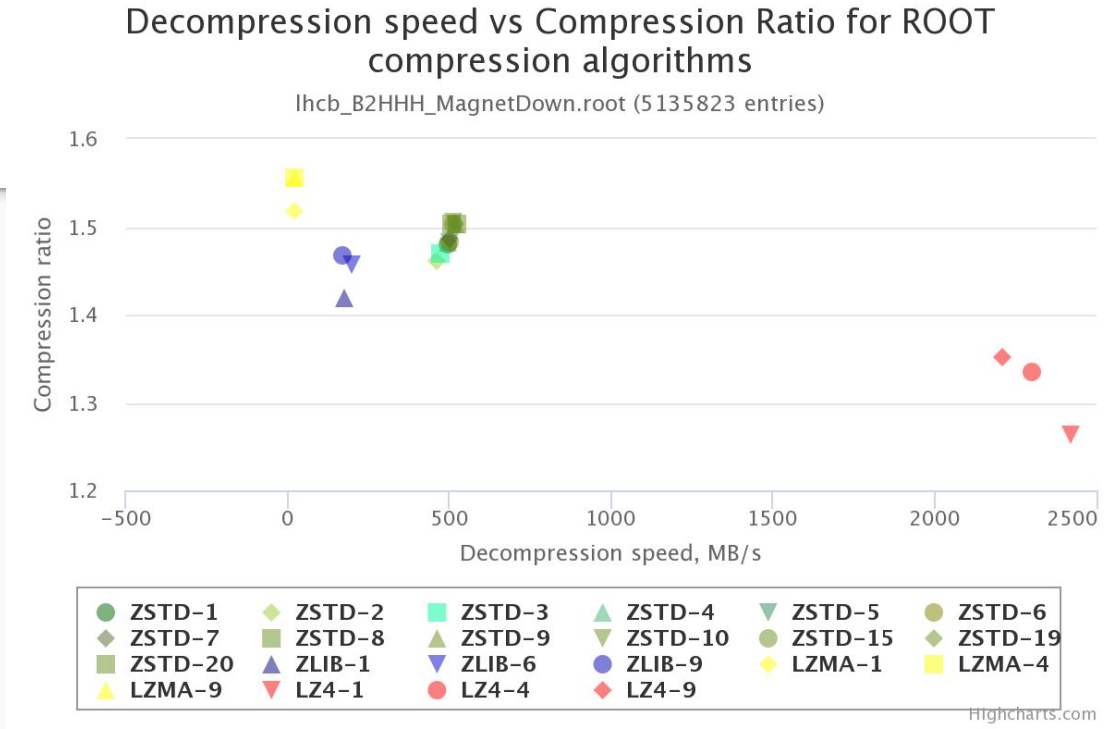
NanoAOD - using ZSTD, it could be **a better compromise between size of file on a disk** and **decompression speed for a faster analysis!**

CMSSW MiniAOD

MiniAOD - **time spend in decompressing on readback is 15x less vs LZMA**, while **size of file with ZSTD is only 10% bigger**!

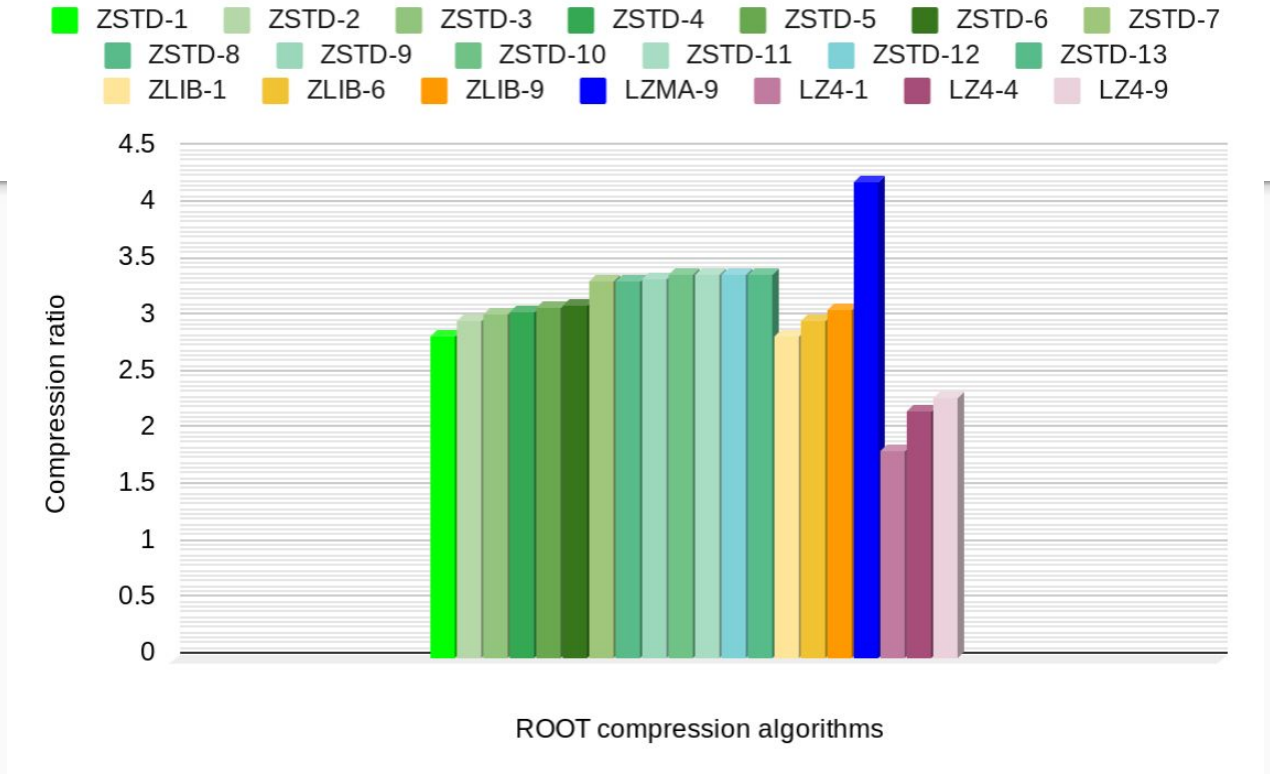
*Big thanks to David Lange for MiniAOD measurements.

LHCB compression speed vs compression ratio



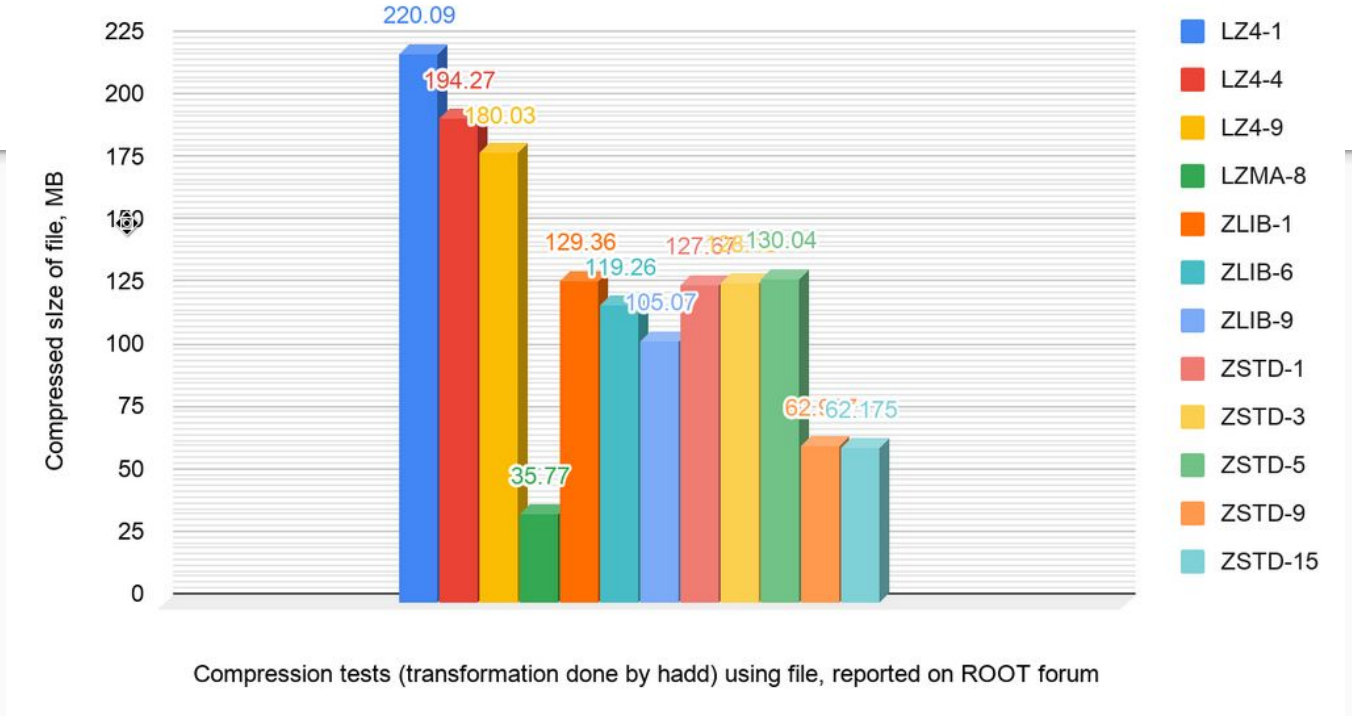
For the very simple ntuples with a simple structure the best choice could be LZ4:
10x time faster read speed

NanoAOD 2019 compression ratio comparison



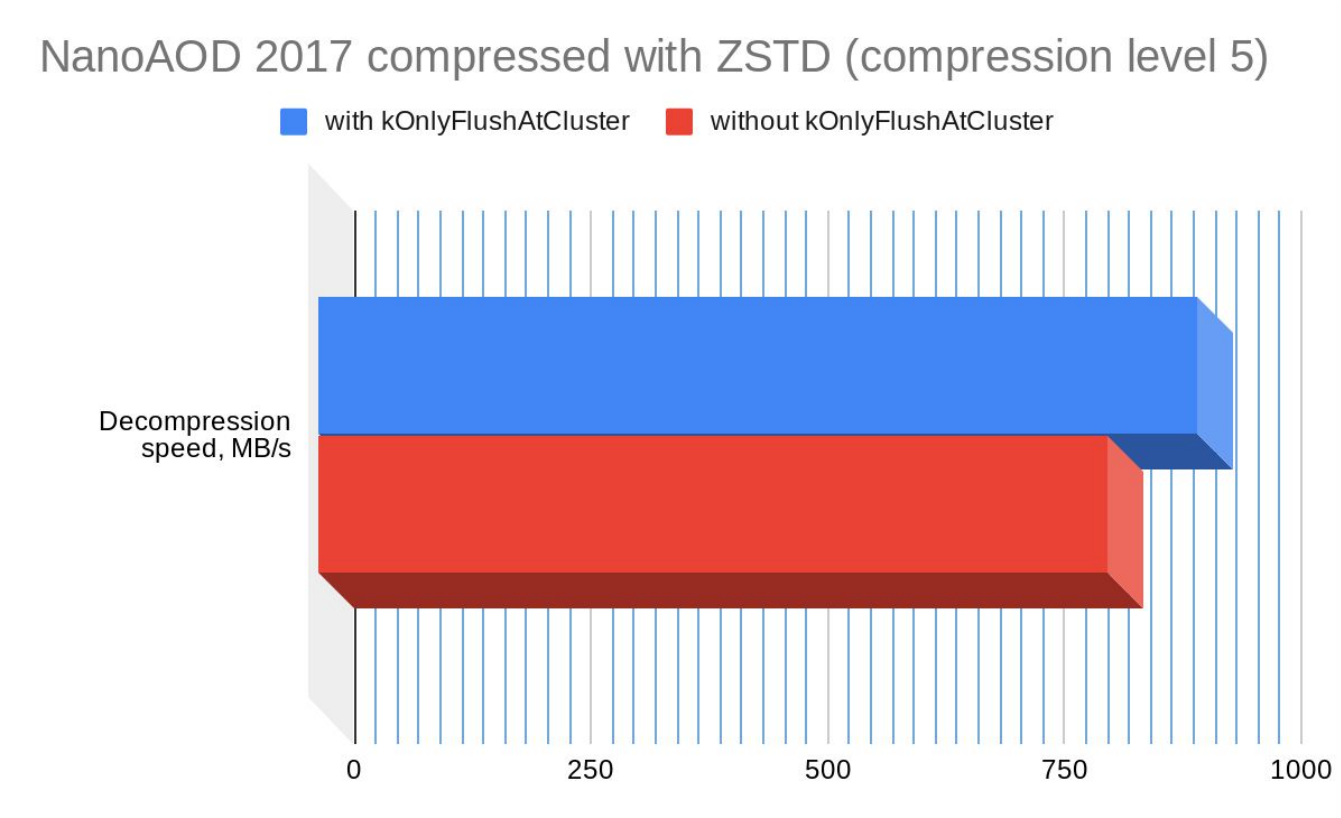
Better compression ratio then ZLIB

Compression of 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)

TTree::kOnlyFlushAtCluster: faster decompression



TTree::kOnlyFlushAtCluster: faster decompression with ZSTD and not only

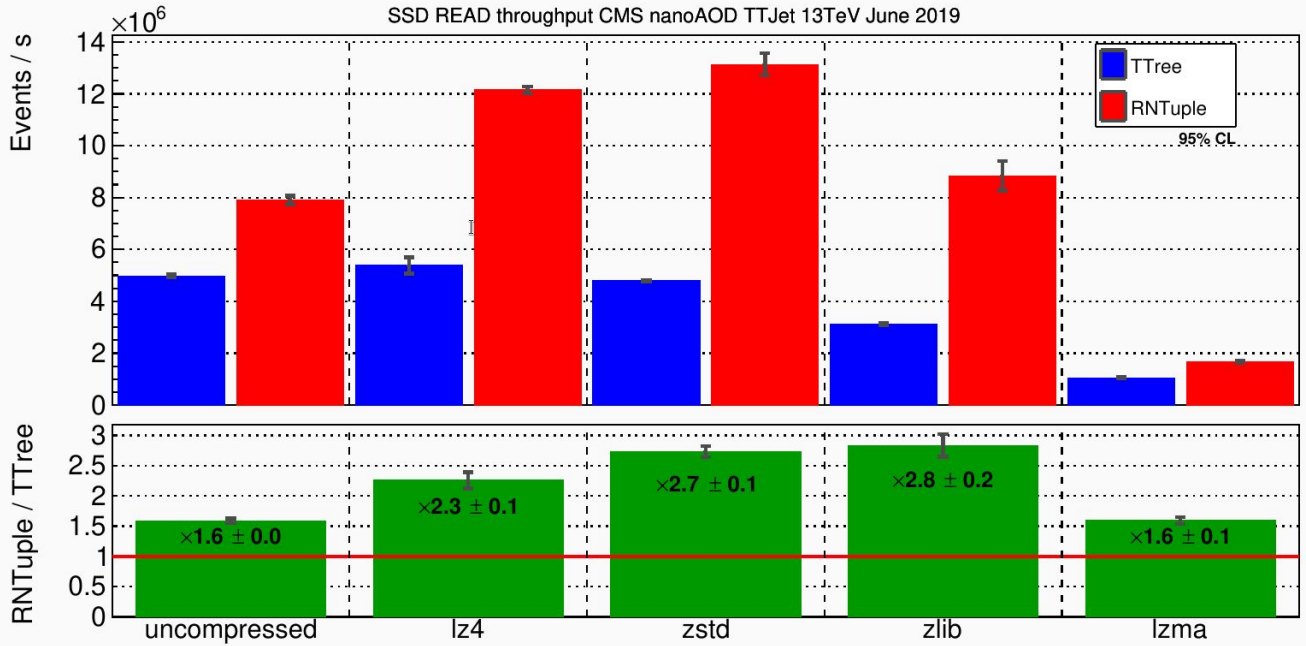
- TTrees can be forced to only create new baskets at event cluster boundaries, it simplifies file layout and I/O at the cost of memory **(using *TTree::kOnlyFlushAtCluster bit for* NanoAOD 2017 size difference was 3.6 %).**
- Recommended for simple file formats such as ntuples but not for more complex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

How to use:

tree->SetBit(TTree::kOnlyFlushAtCluster);

ZSTD for RNtuple (from slides of J.Blomer, CHEP 2019)

Read speed for NVMe SSD



Thank you for your attention!