Impact of ROOT compression algorithms on DAOD_PHYS and DAOD_PHYSLITE data format

ATLAS – Italia Calcolo

9.IX.2022

Caterina Marcon

INFN MILANO
Istituto Nazionale di Fisica Nucleare
Sezione di Milano

Motivations

- In the coming runs, the LHC accelerator will provide higher luminosity of particle collisions to the ATLAS experiment:
  - more simultaneous collisions per event → higher demand of disk space to store the events;
  - a larger event rate will require processing.
- The need for data compression has grown significantly → more interest in profiling the compression algorithms provided by ROOT;
- In this presentation, the impact in terms of file size and reading speed of different compression algorithms provided by ROOT on DAOD_PHYS and DAOD_PHYSLITE ATLAS datasets will be presented.

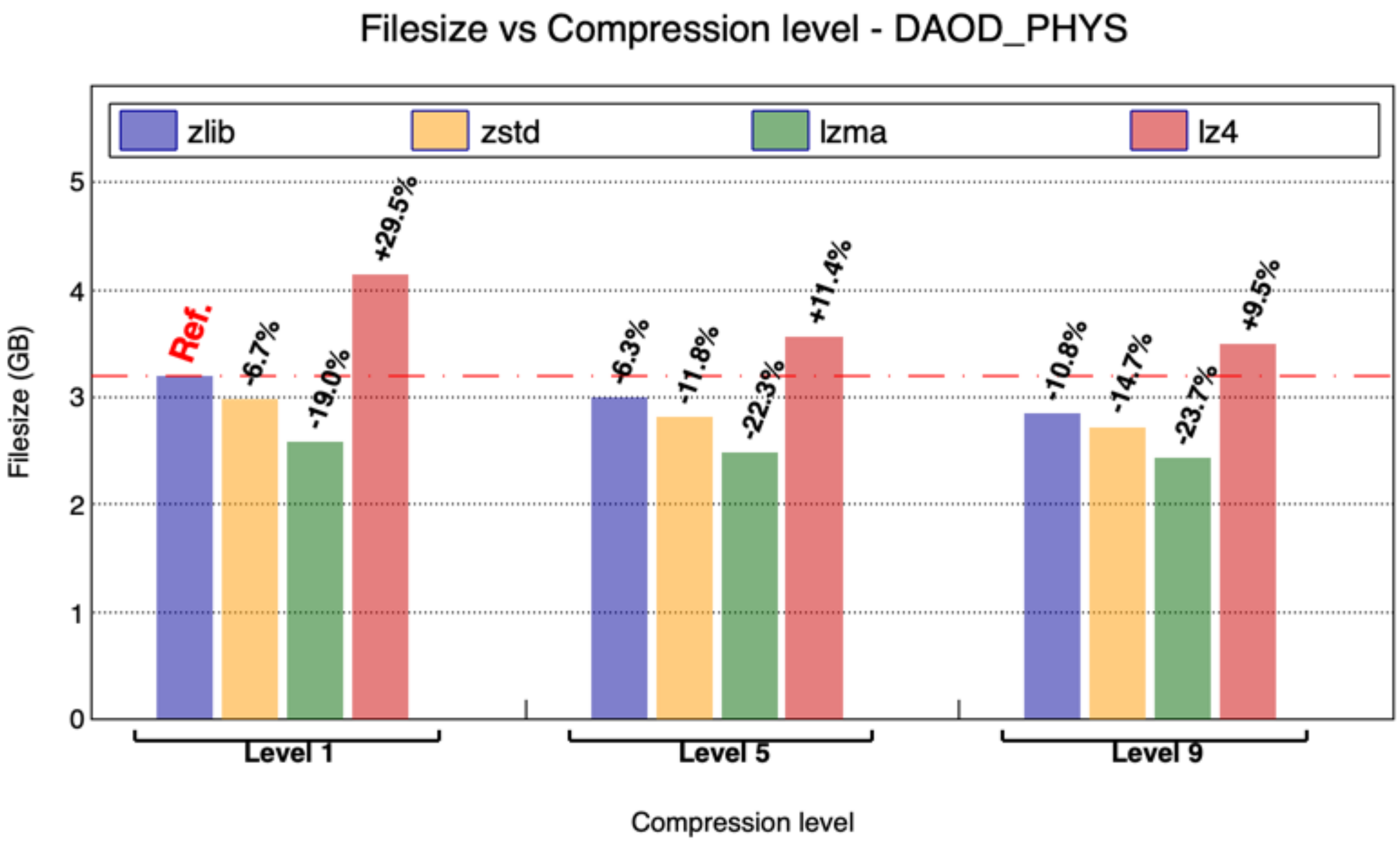
ROOT compression algorithms

- ROOT provides four different compression algorithms:
    - Zlib;
    - Lzma;
    - Lz4;
    - Zstd.
- All these algorithms can be tuned via the compression level option ranging from 1 to 9;
- Higher compression levels offer stronger compression;
- All the algorithms apply lossless compressions -> no validation is needed.

Methods

- Disk-based reading tests allow to collect I/O performance metrics;
- I/O performance metrics are collected via PerfStats (tool provided by ROOT -> access to a range of performance statistics from within the process);
- Reading tests emulate the typical ATLAS data access by reading events from the CollectionTree TTree;
- The CollectionTree object accounts for ~90% of the total file size;
- For file access, the EventLoop backend has been used;
- All tests are carried out using ROOT 6.24, on a dedicated standalone machine;
- Each test has been repeated 5 times and standard deviations are of the order of 3%.

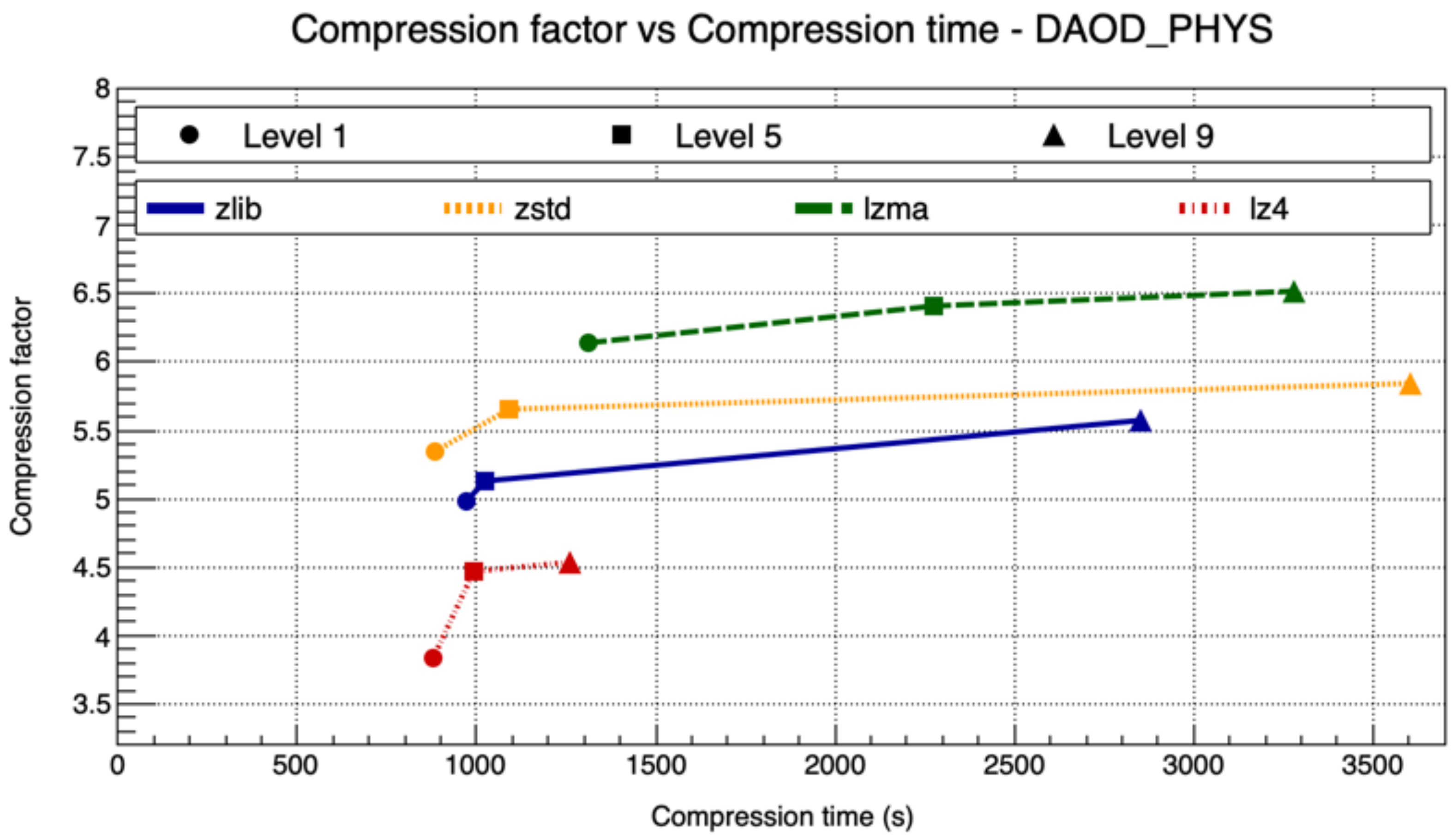
Results: File size VS Compression Level DAOD_PHYS



- The original file size is 15.92 GB;
- As recommended in [1], the zlib level 1 configuration has been considered as the reference performance;
- Lzma provides the best compression (with reductions of more than 20%);
- Lz4 results in the largest files (with increases of up to ~ 30%);
- The file size depends primarily on the compression algorithm and not on the compression level.

[1] https://root.cern.ch/doc/master/structROOT_1_1RCompressionSetting.html

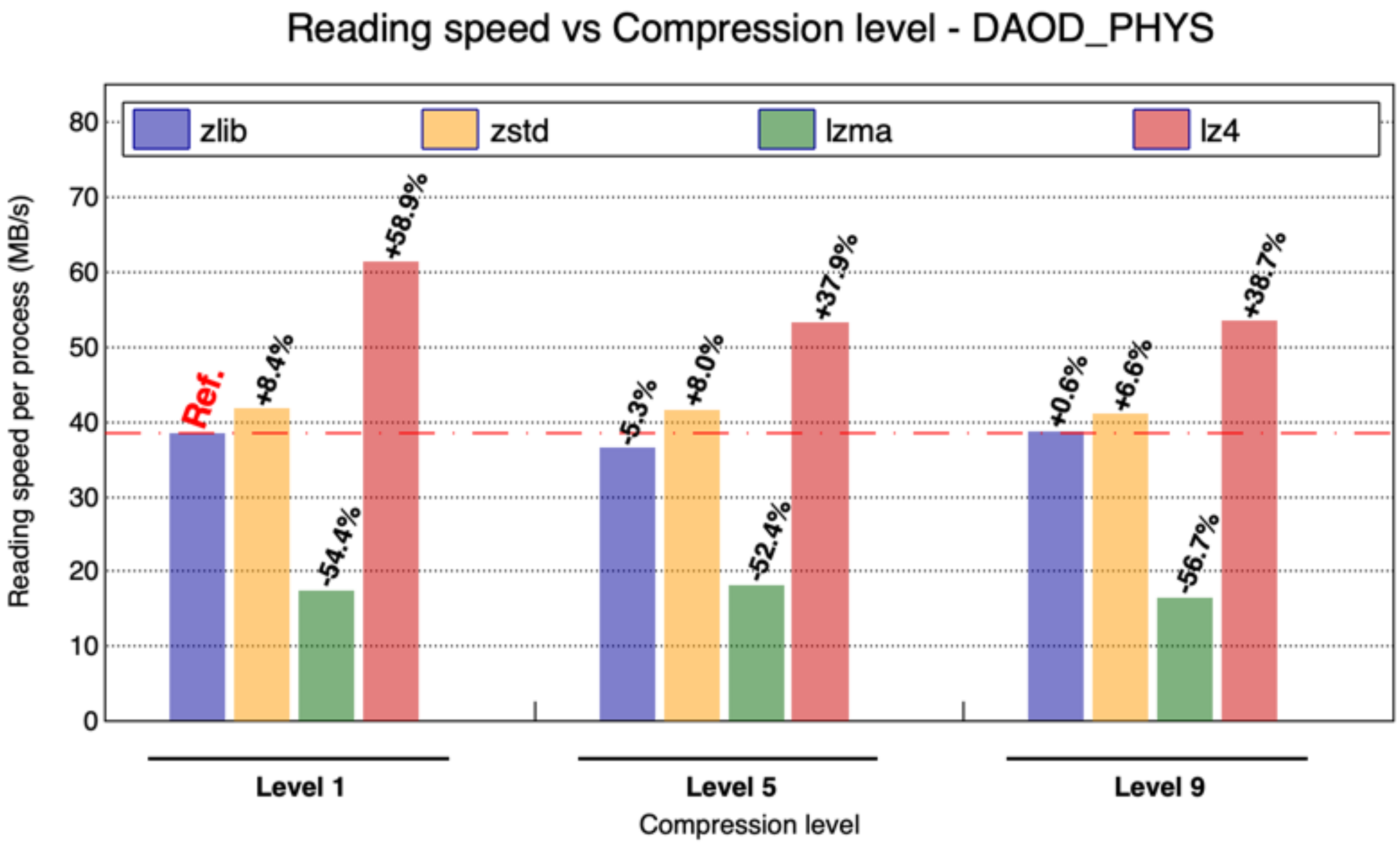
Results: Compression Factor VS Compression time DAOD_PHYS



- Lz4 provides fast compression timse but suffers from low compression factors [1];
- Lzma achieves high compression factors but compression times are slow;
- For Lzma, Lz4 and Zstd, the gain of compression level 9 flattens out → only relevant for cases where file size reduction is the most important metric.

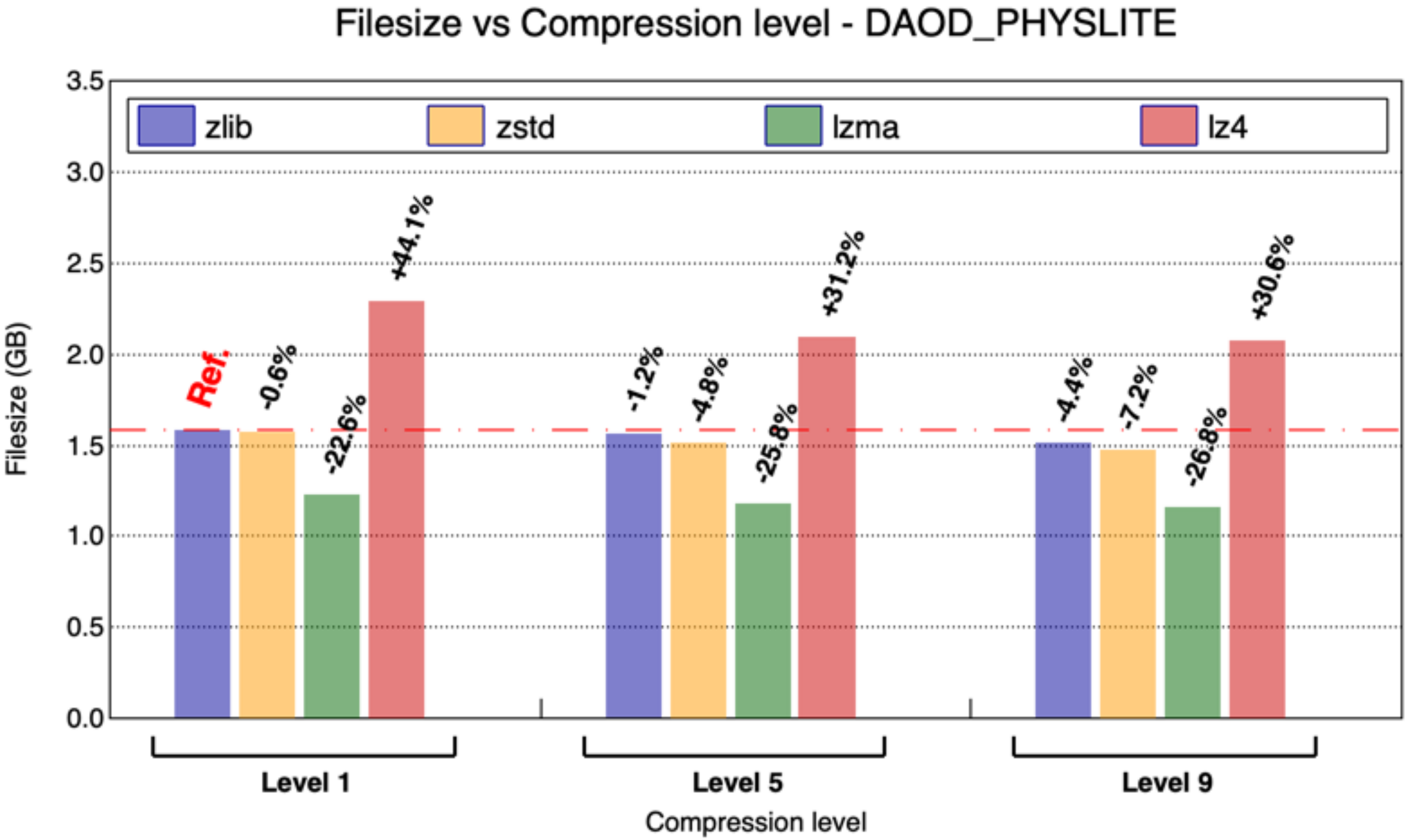
[1] Compression factor = uncompressed data/compressed data

Results: Reading speed VS Compression Level DAOD_PHYS



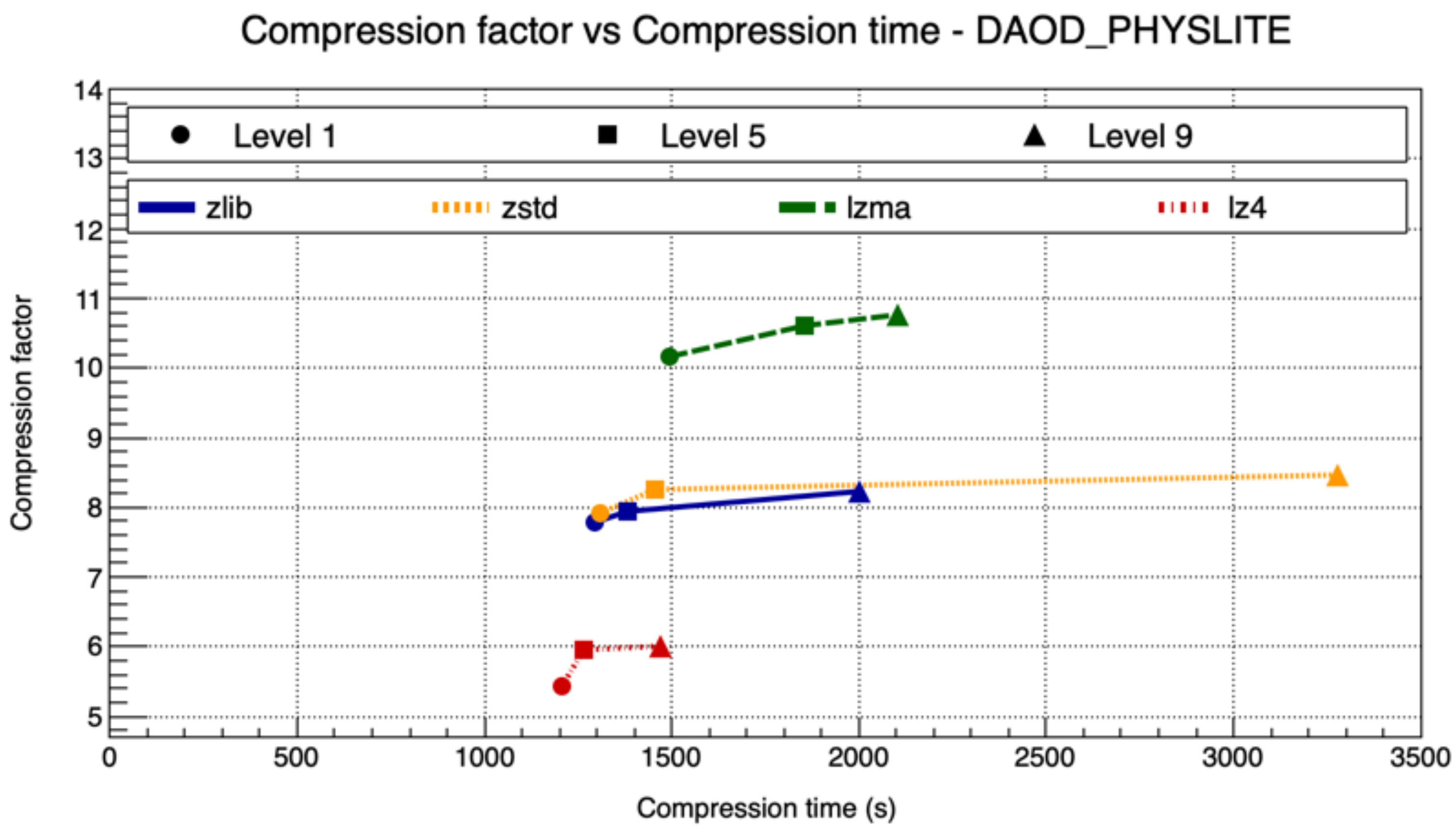
- The zlib level 1 configuration has been taken as reference;
- Lzma has a low reading speed (with degradations of more than 50%);
- Lz4 is the fastest in reading (with a ~60% improvement);
- The reading speed depends primarily on the compression algorithm and not on the compression level.

Results: File size VS Compression Level DAOD_PHYSLITE



- The original file size is 12.46 GB;
- The zlib level 1 configuration has been taken as reference;
- Lzma provides the best compression (with reductions of more than 25%);
- Lz4 results in the largest files (with increases up to ~45);
- The file size depends primarily on the compression algorithm and not on the compression level.

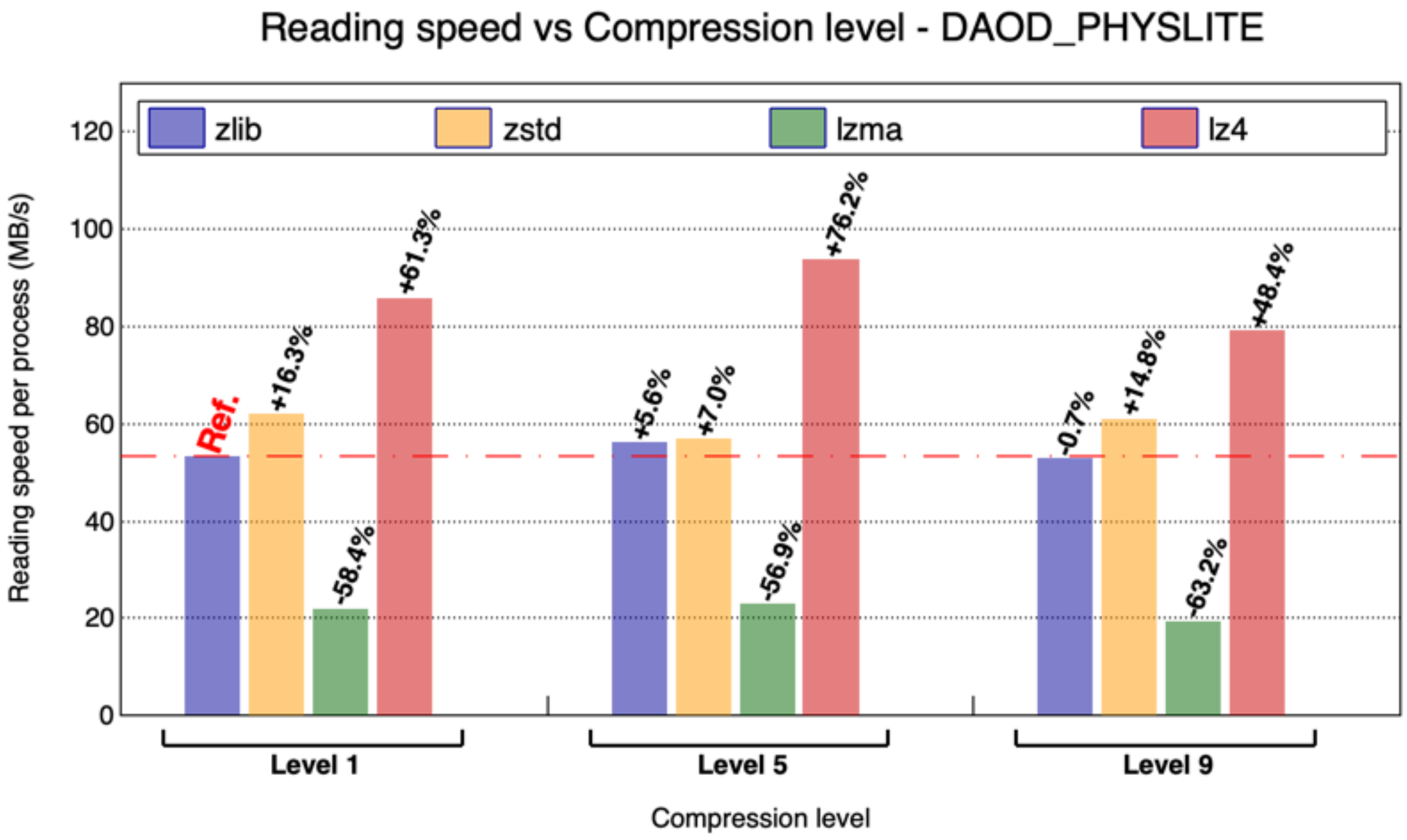
Results: Compression Factor VS Compression time DAOD_PHYSLITYE



- Lz4 provides fast compression times but suffers from low compression factors [1];
- Lzma achives high compression factors but compression times are slow;
- For Lzma, Lz4 and Zstd, the gain of compression level 9 flattens out → only relevant for cases where file size reduction is the most important metric.

[1] Compression factor = uncompressed data/compressed data

Results: Reading speed VS Compression Level DAOD_PHYSLITE



- The zlib level 1 configuration has been taken as reference;
- Lzma has low reading speed (with degradations of more than 60);
- Lz4 is fastest in reading (with more than 70% improvement);
- The reading speed depends primarily on the compression algorithm and not on the compression level.

[1] https://root.cern.ch/doc/master/structROOT_1_1RCompressionSetting.html

Conclusions

- Both the data formats considered show compatible behaviors and are also in agreement with previous tests (with xAOD and DAOD_PHYSVAL);

- For both types of derived files, Lz4 is the fastest in reading but results in the largest files: it should be considered when fast reading is more important than file size reduction;

- In both cases, Lzma provides much better compression at the cost of significantly slower reading speed: it should be considered when file size reduction is the key parameter;

- Compression level 5 could be considered a good compromise.

Outlook

- Investigate the impact that different file storage options provided by ROOT can have:
    - SplitLevel;
    - AutoFlush.

- Investigate the effects of lossy and lossless compressions on a single file.

Computing resources

- CPU: 2x AMD EPYC 7302 (16 Core, 32 Thread)
- 256 GB RAM
- 1.92 TB NVMe SSD (Read: 3000 MB/s, Write: 1500 MB/s)
- CentOS 7