Two High-Performance Alternatives to ZLIB Scientific-Data Compression

Samuel Almeida¹, Vitor Oliveira², António Pina², and Manuel Melle-Franco³

¹ Departamento de Informática, Universidade do Minho, Campus de Gualtar, Braga, Portugal
² Laboratório de Instrumentação e Física Experimental de Partículas, Campus de Gualtar, Braga, Portugal
³ Departamento de Informática, Centro de Ciências e Tecnologias da Computação, Universidade do Minho, Campus de Gualtar, Braga, Portugal

{sam.alm321, manuelmelle}@gmail.com, {vspo, amp}@di.uminho.pt

Abstract. ZLIB is used in diverse frameworks by the scientific community, both to reduce disk storage and to alleviate pressure on I/O. As it becomes a bottleneck on multi-core systems, higher throughput alternatives must be considered, exploring parallelism and/or more effective compression schemes. This work provides a comparative study of the ZLIB, LZ4 and FPC compressors (serial and parallel implementations), focusing on CR, bandwidth and speedup. LZ4 provides very high throughput (decompressing over 1GB/s versus 120MB/s for ZLIB) but its CR suffers a degradation of 5-10%. FPC also provides higher throughputs than ZLIB, but the CR varies a lot with the data. ZLIB and LZ4 can achieve almost linear speedups for some datasets, while current implementation of parallel FPC provides little if any performance gain. For the ROOT dataset, LZ4 was found to provide higher CR, scalability and lower memory consumption than FPC, thus emerging as a better alternative to ZLIB.

Keywords: Data Compression, Scientific Data, ROOT, Parallel Compression, ZLIB, LZ4, FPC.

1 Introduction

Technological developments have lead to increasingly more powerful data processing systems, higher-resolution sensors and higher bandwidth communication networks. Yet systems are getting increasingly farther from Amdahl’s balanced computing system, as the abilities to create, process, distribute and store data have not grown equally. In particular, the availability of multi-core systems has greatly amplified the ratio between the available computational power and the available input/output bandwidth. As the potential of “big data” repositories in our society is explored, this balance between the system’s ability to analyse, to transmit and to store data becomes even more important.
The same applies to the techniques that can affect it, such as caching, pre-fetching and compressing the required data.

Data compression, in particular, has been used for many years to trade computing power for storage capacity as well as for network and storage bandwidth whenever the first more abundant than the later ones. It has been used to improve disk space utilization with compressed packages or file-systems, to speedup WAN traffic over slow links and to increase write bandwidth in some solid-state disks. It also presents a large potential for specialized libraries such as NetCDF, HDF5 and ROOT to store scientific data in more compact forms. Taking into account the 2:1 file size reduction allowed by the LZ77 compression, very significant savings in terms of network bandwidth and disk space are achieved in datasets such those from the LHC’s ATLAS (A Toroidal LHC Apparatus) experiment [5], which start in the multi-petabyte range, and that after several pre-processing stages, the data reaches the analysis applications still in the tens of terabytes range.

In [6] it was identified that in the analysed application the processing time associated with compression was indeed significant, where the ZLIB library took 18% of the total CPU-time to expand data prior to analysis. Some of that overhead can be offset by the reduced time associated with writing and reading less data to disk, as long as compression/decompression bandwidth exceeds storage bandwidth. But modern storage systems with well behaved usage patterns can provide much more bandwidth than ZLIB decompression can handle, both because a single modern disk is already capable of sequentially reading faster than the 120 MB/s decompression bandwidth achieved by ZLIB in a single processor core (Intel Xeon E5620) and because ZLIB does not support multiple threads.

As ZLIB compression and decompression becomes a bottleneck, techniques must be devised to improve performance while maintaining the advantages of compression. This work focuses both alternative compression methods, that could provide higher bandwidth, and implementations with a level of parallelism, that could properly explore multi-core systems. We present a comparative performance study of six compressors, in terms of bandwidth, compression ratio (CR) and scalability, using a group of different numeric scientific datasets. This approach permits assessing the compressor behaviour in the context of ROOT files and a few other application areas where compression can be beneficial.

This work focuses on the study of the serial compressors gzip, LZ4 and FPC and their respective multi-threaded counterparts pigz, lz4mt and pFPC.

2 Background Information

Scientific data compression has been used extensively, and in situ techniques are also on the rise [7][8]. For instance, in the ROOT toolkit [10], compression is used to deal with large volumes of information such as that generated by the Large Hadron Collider’s (LHC) experiments. The approach there is to compress objects with the general purpose ZLIB or LZMA compressors prior to writing the data nodes to disk.