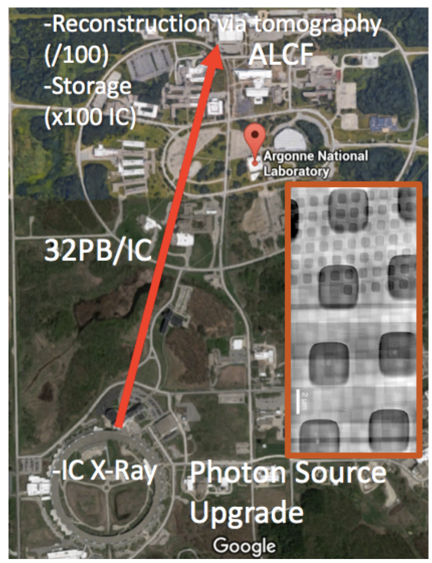
Scalable Compression as a Fundamental Pattern for HPC–Big Data Convergence

Franck Cappello, Sheng Di (ANL)

This white paper covers one element of the plausible pathways to convergence:
1) Common tools and technologies:
    a. Common libraries: numerical; runtime systems; i/o; data analytics;

Scalable data compression is one of the operations that is needed both in scientific simulation and in scientific experiments. But scalable compression is more than a technique; it is one of the fundamental patterns supporting the convergence of HPC and big data.

In every domain where the infrastructure cannot communicate and/or store the generated raw data directly, data compression is a critical data transformation that contributes to satisfying the end-user needs. For example, data compression is already widely used for image and signal compression in many consumer products. Compression is also needed in large-scale data centers (Yahoo compresses emails), and lossy compression is an active research topic for medical imaging and genomic applications.



Compression is becoming more and more relevant for scientific data produced by simulations and experiments. For example, the HACC cosmology code generates 40 PB of data when running a 1 billion particle simulation. In another domain, the RAVEN project proposes to use the Argonne Advanced Photon Source for x-ray tomography of integrated circuits (ICs) that will produce 32 TB of data for each IC. Storing or communicating these raw datasets without significant data reduction is impossible. In the above examples, users need to reduce the data by a factor 10 to 100 to obtain reasonable communication and storage times. The approach of data omission (i.e., storing only 1 data point of 10 produced, 1 snapshot for 10 produced), often used for simulations, is not satisfactory because it impairs the accuracy of the analytics performed from the simulation.

Although compression is critical to evolve many scientific domains to the next step, the technology of scientific data compression and the understanding on how to use it are still in their infancy. The first evidence is the lack f results in this domain: over the 26 years of the prestigious IEEE Data Compression Conferences, only 12 papers [1–13] identify an aspect of scientific data in their title (floating-point data, data from simulation, numerical data, scientific data). The second evidence is the poor performance on some datasets.

Table 1 shows the compression factors of the most effective compressors on datasets coming from simulations in fluid dynamics (Nek5000), shock (FLASH), and climate. SZ [13], ZFP [14], ISABELA [15], SSEM [16], and FPZIP [17] are lossy compressors. SZ, ZFP, and ISABELA are error bounded. We set the error bound to 10^{-6} for these datasets. FPZIP [17] and FPC [18] are lossless. Only NUMARCK [19] has been designed specifically for compression in time. Other compressors perform compression in space. The table shows that compressors achieve excellent compression factors on some datasets. It also shows that some data sets are "hard to compress": compression factors lower than 10 will be of little help in addressing the compression challenges faced by scientific simulations and experiments.

Unfortunately for their users, HACC and APS datasets (our two introductory examples) fall into this category of hard-to-compress data -sets (SZ and ZFP hardly manage to compress them by factors of 3 to 5). Another element demonstrating that scientific data compressors are in their infancy is the lack of scalable compressor. Compressing large datasets in parallel today mainly consists of the concurrent execution of local compressor instances compressing dataset chunks in isolation. Except for pFPC and ISABELA, none of the existing compressors has a parallel implementation performing the compression considering all the data and not isolated chunks.

Table 1: Compression factors of key compressors

Benchmark	SZ	ZFP	ZFP+Gzip	ISA	ISA+Gzip	SSEM[a]	FPZIP-40[b]	Gzip	FPC[c]
Blast2	110	6.48	36.2	4.56	46.2	39.7	22.9	77	11.4
Sedov	7.44	4.42	5.47	4.42	7.44	17	3.43	3.13	1.9
BlastBS	3.26	3.48	3.65	4.43	5.06	8.45	2.43	1.24	1.29
Eddy	8.13	2.5	2.61	4.34	5.18	N/A	2.56	5.5	3.89
Vortex	13.6	4.45	4.77	4.43	4.72	12	3.35	2.23	2.34
BrioWu	71.2	8.1	43.4	5	57.4	35.7	21.9	73	8.5
GALLEX	183.6	36.7	92.7	4.89	33.6	82.4	20.35	34.7	11.37
MacLaurin	116	10.2	14	4.1	5.47	7.44	3.84	2.03	2.08
Orbit	433	31.7	89	4.96	8.43	11.7	3.9	1.8	1.86
ShafranovShock	48	3.68	8.75	4.24	12.2	20.3	19.9	28	7.33
CICE	5.43	2.11	2.16	4.19	4.46	3.83	2.3	2.6	2.67
ATM	3.95	2.3	2.75	3.1	3.7	1.82	1.04	1.36	N/A
Hurricane	1.63	1.19	1.2	2.57	2.65	1.11	2.07	1.16	N/A

Considering that exascale execution and extreme-scale experiments will produce even more data in the near future (2020–2025), investment in lossy compressor technology for hard-to-compress scientific datasets is urgently needed in order to support the communication, storage, and analysis of this data.

Beyond the research on compression, scientists also need to understand how to use lossy compression. The classic features of compressors (integer data compression, floating-point data compression, fast compression and decompression, error bounds for lossy compressors) do not characterize compressors specifically with respect to their integration into a high-performance computing and data analytics workflow.

For example, in the IC imaging application, assuming a lossy compressor capable of a factor of 100 compression, can we perform the tomography and the following data analytics directly from the compressed data? Obviously, if the following steps can work only from decompressed data, large storage and significant decompression time will be needed. If the data needs to be decompressed, can we decompress it only partially to allow for pipelined decompression, reconstruction, and analytics? Note that partial decompression requires random access in the compressed dataset, a capability that is not considered a priority today in lossy compressors (SZ and ZFP provide random access). The same set of questions applies to large scale simulations. If we can avoid data omission and compress the raw dataset by a factor of 100, can the following data analytics steps be performed on the compressed data?

Using lossy compression for scientific data coming from simulations and experiment requires developing new techniques in order to perform data analytics directly on lossy compressed data or to enable the pipelining of decompression with data analytics.

Bibliography

[1] J. Bradley, C. Brislawn, Wavelet Transform-Vector Quantization Compression of Supercomputer Ocean Model Simulation Output, IEEE Data Compression Conference, 1993

[2] V. Engelson, D. Fritzson, P. Fritzson, Loseless Compression of High-Volume Numerical Data for Simulations, IEEE Data Compression Conference, 2000

[3] H. Yokoo, Application of AVL Trees to the Adaptive Compression of Numerical Data, IEEE Data Compression Conference, 1993

[4] J.P. Wilson, Wavelet-Based Lossy Compression of Baratropic Turbulence Simulation Data, IEEE Data Compression Conference, 2002
[5] B.E. Usevitch, JPEG2000 Extensions for Bit Plane Coding of Floating Point Data, IEEE Data Compression Conference, 2003
[6] B. Usevitch, JPEG2000 Compliant Lossless Coding of Floating Point Data, IEEE Data Compression Conference, 2005
[7] P. Ratanaworabhan, J. Ke, M. Burtscher, Fast Lossless Compression of Scientific Floating-Point Data, IEEE Data Compression Conference, 2006
[8] M. Burtscher and P. Ratanaworabhan, High Throughput Compression of Double-Precision Floating-Point Data, IEEE Data Compression Conference, 2007
[9] D. Unat, T. Hromadka III, S. B. Baden, An Adaptive Sub-Sampling Method for In-memory Compression of Scientific Data, IEEE Data Compression Conference, 2009
[10] M. Burtscher, P. Ratanaworabhan, pFPC: A Parallel Compressor for Floating-Point Data, IEEE Data Compression Conference, 2009
[11] K. Sano, K. Katahira, and S. Yamamoto, Segment-Parallel Predictor for FPGA-Based Hardware Compressor and Decompressor of Floating-Point Data Streams to Enhance Memory I/O Bandwidth, IEEE Data Compression Conference, 2010
[12] Y. J. Fan, C. Kamath, Practical Considerations in Applying Compressed Sensing to Simulation Data, IEEE Data Compression Conference, 2010
[13] S. Di and F. Cappello, "Fast Error-bounded Lossy HPC Data Compression with SZ," in Proceedings of IEEE IPDPS16, 2016.
[14] P. Lindstrom, "Fixed-Rate Compressed Floating-Point Arrays," IEEE Trans. on Visualization and Computer Graphics, 20(12):2674–2683, 2014.
[15] S. Lakshminarasimhan, N. Shah, S. Ethier, S. Klasky, R. Latham, R. Ross, and N.F. Samatova, "Compressing the Incompressible with ISABELA: In-situ Reduction of Spatiotemporal Data," in 17th Euro-Par11, pages 366–379, 2011.
[16] N. Sasaki, K. Sato, T. Endo, and S. Matsuoka, "Exploration of Lossy Compression for Application-level Checkpoint/Restart," in Proceedings of IEEE 29th IPDPS15, pages 914–922, 2015.
[17] P. Lindstrom and M. Isenburg, "Fast and Efficient Compression of Floating-Point Data," IEEE Trans. on Visualization and Computer Graphics, 12(5):1245–1250, 2006.
[18] M. Burtscher and P. Ratanaworabhan, "High Throughput Compression of Double-Precision Floating-Point Data," in Data Compression Conference (DCC'07), pages 293–302, 2007.
[19] Z. Chen, S.W. Son, W. Hendrix, A. Agrawal, W. Liao, and A. Choudhary, "NUMARCK: Machine Learning Algorithm for Resiliency and Checkpointing," in Proceedings of IEEE/ACM Supercomputing (SC14), pages 733–744, 2014.