

Improving Error-bounded Lossy Compression for Cosmological N-body Simulation

Sihuan Li,^{*} Sheng Di,[†] Xin Liang,^{*} Zizhong Chen,^{*} and Franck Cappello[‡]

^{*} University of California, Riverside, CA, USA

[†] Argonne National Laboratory, Lemont, IL, USA

[‡] University of Illinois at Urbana-Champaign, IL, USA

sli049@ucr.edu, sdi1@anl.gov, xliang007@ucr.edu,

chen@cs.ucr.edu, cappello@mcs.anl.gov

Abstract—Cosmological simulations may produce extremely large amount of data, such that its successful run depends on large storage capacity and huge I/O bandwidth, especially in the exascale computing scale. Effective error-bounded lossy compressors with both high compression ratios and low data distortion can significantly reduce the total data size while guaranteeing the data valid for post-analysis. In this poster, we propose a novel, efficient compression model for cosmological N-body simulation framework, by combining the advantages of both space-based compression and time-based compression. The evaluation with a well-known cosmological simulation code shows that our proposed solution can get much higher compression quality than other existing state-of-the-art compressors, with comparable compression/decompression rates.

I. INTRODUCTION

Scientific simulations of computing in the field of cosmology can produce extremely huge volume of data which needs to be stored in parallel file systems (PFS) in practice. Hardware/Hybrid Accelerated Cosmology Code (HACC) [1] is a typical example, which may produce 20+ PB of data even in one simulation case. How to reduce the cosmological N-body simulation data size significantly is fatal to the success of the cosmological simulations. Existing lossy compressors, however, mainly focus on the data consecutiveness in space, which is not suitable for the particle data because they are often stored based on separate dimensions (x , y , z) such that there are no spatial correlations for the adjacent data points.

In this work, we combine both space-based compression and time-based compression to improve the compression quality for cosmological data. Some challenges need to be resolved. First, the particles across consecutive time steps are not aligned generally, such that they are uneasy to compare across time steps. Moreover, the number of particles may be different between the snapshots at consecutive time steps. Secondly, how to combine space-based compression and time-based compression needs to be studied carefully. Last but not the least, we need to explore how to predict data values accurately with the consideration of strict error controls on demand. We make the following contributions:

- We propose a cosmological data compression model by combining both space and time dimension, with derived optimal intervals for the two types of compression.
- We propose an efficient on-line particle alignment making time-based compression possible for particle data.
- Evaluation with a large-scale HACC simulation shows that our solution is the best in class.

II. FRAMEWORK FOR TIME-SPACE COMPRESSION

The compression principle and workflow during the simulation are illustrated in Figure 1 and Figure 2, respectively.

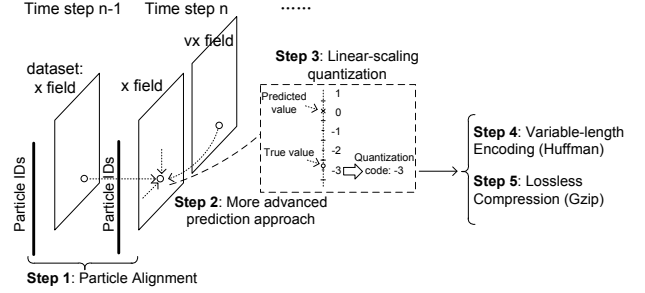


Figure 1. Principle of Our Optimized Error-bounded Compressor

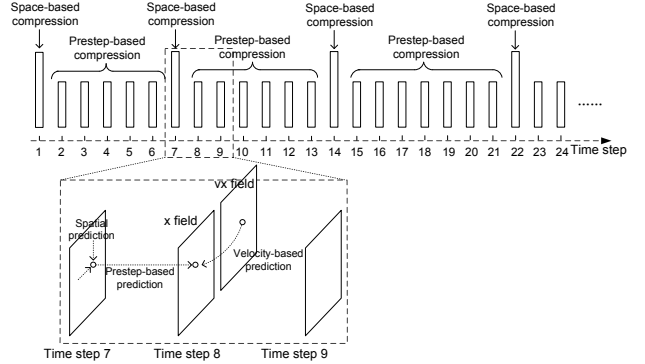


Figure 2. Optimized In-situ Compression by Combining Space-based Prediction and Prestep-based Prediction

III. METHODOLOGY

A. On-line particle alignment

On-line particle alignment is non-trivial because: (1) at most one previous snapshot is usable due to memory limit; (2) it has to be linear time complexity because of linear time complexity cost by compressors. Our proposed algorithm tackles the first challenge by using an alignment indication array instead of swapping. We use radix sorting to tackle the second one. We demonstrate the algorithm in Figure 3. To verify, applying alignment array 010011 to snapshot 0 will get 356 which aligns to snapshot 1. Applying alignment array 001 to dark area of snapshot 1, 356, we get 35 which aligns to snapshot 2.

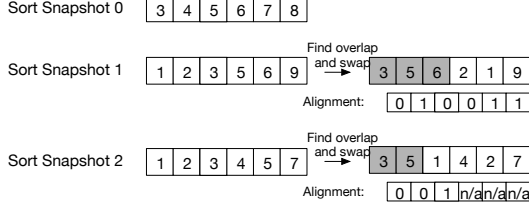


Figure 3. Example of on-line alignment algorithm

B. Optimization of space compression interval

How many time-based compressions should operate between space-based compression is a non-trivial problem.

Assumptions and notations: Assume the particle overlap ratio with previous snapshot is α ($0 < \alpha < 1$); Assume the reciprocal of space based compression ratio is a constant r_s and that of time based compression ratio is a constant r_t ($0 < r_s, r_t < 1$). Without loss of generality, we set the total number of snapshots n , each snapshot has original file size 1, every space based compressed snapshot will be followed by $(k-1)$ time based compressed snapshots.

Theorem 1: Given the above assumptions and notations, the optimal k to minimize the overall compressed file size is either $\lceil k_0 \rceil$ or $\lfloor k_0 \rfloor$ such that $\alpha^{k_0-1}(k_0 \ln \alpha - 1) + 1 = 0$.

Explanation: We calculate the total compressed file size $f(k)$ and prove it has only one global minimum in the domain $k \in [1, +\infty)$.

$$\min_k f(k) \triangleq n \left[r_s + \frac{\alpha}{1-\alpha} (r_t - r_s) \frac{1 - \alpha^{k-1}}{k} \right] \quad (1a)$$

$$\text{subject to } k \in [1, +\infty) \quad (1b)$$

$$k \in \mathbb{N} \quad (1c)$$

C. Optimization of Prediction Accuracy

For velocity, we just use the velocity of the previous snapshot to predict the current one. To improve the prediction accuracy of position variable, we combine velocity and position information in the previous snapshot to predict the position of the current step, by leveraging Newtons law: $\vec{x}_1 = \vec{x}_0 + \vec{v}_0 \Delta t$. We minimize the mean square error (MSE) to get a good estimation of Δt , as formulated below:

$$\min_{\Delta t} mse(\Delta t) \triangleq \frac{1}{n} (\vec{x}_1 - \vec{x}_0 - \vec{v}_0 \Delta t)^2 \quad (2a)$$

$$\text{subject to } \Delta t \in (-\infty, +\infty) \quad (2b)$$

$$\Delta t \in \mathbb{R} \quad (2c)$$

This optimization problem can be solved in linear time.

IV. EVALUATION

We evaluate the position variables and velocity variables separately using HACC data. The HACC users require value-range based relative error for position and point-wise relative error for velocity. We do not compare with the TNG-MF1 trajectory compressor since number of particles is changing. In that case, TNG-MF1 compressor reduces to single-frame compression [8]. We compress 100 snapshots of HACC data

totaling 36GB. To evaluate position variables, we set the value range based error bound to $1e-3$, $1e-4$, $1e-5$, $1e-6$ and compare our compression method (SZ_vlct) with single snapshot based SZ [2], [6] (SZ_s) and time based SZ that only uses previous step position information to predict the current one (SZ_prestep) and ZFP [3], Numarck [5], [7], decimation. In SZ_vlct, to predict position, we fixed the point-wise relatively error bound to be 0.1 for velocity. We plot the rate distortion for variable x in Figure 4. The results are similar for y and z. The figure demonstrates our proposed compression outperforms the other ones by a factor of more than 1.5X. Notice that Numarck and decimation are not able to compress partible data since they are not aligned. We test Numarck and decimation using reordered data that are generated beforehand offline.

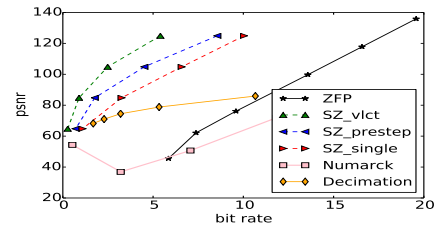


Figure 4. Rate distortion of different compressors on variable x

To evaluate velocity variables, we set the point-wise relative error bound to be $1e-1$, $1e-2$, and $1e-3$ and compare our method (SZ_vlct) with Numarck (Nk), single snapshot based SZ (SZ_s) and FPZIP [4]. To evaluate compression rate, we fixed the value range based absolute error bound for position to be $1e-6$ and the point-wise relative error bound for velocity to be $1e-3$. Decimation and Numarck are omitted because of relatively low compression quality.

	1E-1				1E-2				1E-3			
	SZ_vlct	Nk	SZ_s	FPZIP	SZ_vlct	Nk	SZ_s	FPZIP	SZ_vlct	Nk	SZ_s	FPZIP
Vx	13.90	11.28	9.33	5.67	7.59	5.69	4.91	3.71	4.64	2.97	3.24	2.75
Vy	12.96	10.78	8.66	5.33	7.22	5.49	4.69	3.56	4.45	2.92	3.14	2.66
Vz	11.60	10.08	7.42	4.98	6.69	5.21	4.24	3.40	4.19	2.91	2.93	2.58
					SZ_vlct		SZ_s		ZFP + FPZIP			
					Compression rate		33.76		93.12		67.62	
					Decompression rate		98.80		64.96		65.09	

Figure 5. Compression ratio for velocity and overall compression/decompression rate (MB/s)

For velocity, our compression ratios improve over Numarck by 15% to 56%. For compression rate, our methods slow down because of particle alignment but has better decompression rate.

V. CONCLUSION

We propose a general compression framework combining space-based compression and time-based compression. Several optimization techniques are proposed especially for cosmological scientific simulations based on the compression framework. Evaluation on one of the most popular large scale cosmological simulation (HACC) shows our methods outperform the state of the art.

REFERENCES

- [1] S. Habib, A. Pope, H. Finkel, N. Frontiere, K. Heitmann, D. Daniel, P. Fasel, V. Morozov, G. Zagaris, T. Peterka, et al, "Hacc: Simulating sky surveys on state-of-the-art supercomputing architectures," In *New Astronomy*, 42:4965, 2016.
- [2] D. Tao, S. Di, Z. Chen, and F. Cappello, "Significantly Improving Lossy Compression for Scientific Data Sets Based on Multidimensional Prediction and Error-Controlled Quantization," In *IEEE International Parallel and Distributed Processing Symposium IPDPS2017*, Orlando, Florida, USA, May 29-June 2, pp. 1129–1139, 2017.
- [3] P. Lindstrom. Fixed-Rate Compressed Floating-Point Arrays. *IEEE Trans. on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2674–2683, 2014.
- [4] p. Lindstrom and M. Isenburg, "Fast and Efficient Compression of Floating-Point Data," *IEEE Trans. on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 1245–1250, 2006.
- [5] Yuan, Z., Hendrix, W., Son, S.W., Federrath, C., Agrawal, A., Liao, W.K. and Choudhary, A., 2016, December. Parallel Implementation of Lossy Data Compression for Temporal Data Sets. In High Performance Computing (HiPC), 2016 IEEE 23rd International Conference on (pp. 62-71). IEEE.
- [6] D. Tao, S. Di, Z Chen, and F. Cappello, "In-Depth Exploration of Single-Snapshot Lossy Compression Techniques for N-Body Simulations," in *IEEE Bigdata*, 2017.
- [7] 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)*, pp. 733–744, 2014.
- [8] Lundborg, Magnus and Apostolov, Rossen and Spångberg, Daniel and Gärdenäs, Anders and van der Spoel, David and Lindahl, Erik., 2014. "An efficient and extensible format, library, and API for binary trajectory data from molecular simulations." *Journal of computational chemistry*, 35(3), pp.260-269.