Application of Numerical Accuracy to the Selection of Lossy Compression Error Tolerances

Jon Calhoun, Luke Olson (Advisor), Marc Snir (Advisor)
University of Illinois at Urbana-Champaign
201 N. Goodwin Ave.
Urbana, Illinois 61801, USA
{jccalho2, lukeo, snir}@illinois.edu

ABSTRACT

Long running HPC applications depend on checkpoint restart to recover from failures and utilize multiple time allocations. Memory bandwidth and in particular file system bandwidth continues to be limiters to application performance and scalability. Compression techniques can be used to reduce data size limiting its impact. Lossless compression fails to generate high compression factors for floating-point data. Lossy compression generates noticeably higher compression factors at the expense of adding a small but controllable amount of error into the simulation. In this poster, a methodology for selecting lossy compression error tolerances is presented and evaluated that interprets compression error as numerical error. This interpretation allows defining error tolerances that hide compression error in the high-order truncation error already present in the simulation. This error tolerance selection methodology is shown to add error that does not effect the result of the simulation.

1. INTRODUCTION

Checkpoint restart is fundamental to long running HPC applications. Checkpoints are stored on the parallel file system that is increasingly becoming a performance bottleneck. Next generation HPC system are increasing the size of main memory by 5-10x, but the bandwidth to the parallel file system remains about 1 TB/s. Without increasing I/O bandwidth commensurately, applications may have difficulty scaling due to an increase in I/O time. Combine this with an expected smaller mean time between failure (MTBF) [5], checkpoint restart will become a larger fraction of runtime and put additional pressure on the parallel file system.

Compression techniques can be used to reduce the load on the parallel file system, but traditional lossless compression yields only small reductions in data set size for floating-point data sets [6]. At the expense of a small controllable amount of error being added to the data, lossy compression can yield significantly higher compression factors. Understanding this compression error is critical for effective use of lossy compression.

This poster makes the following contributions:

- provides a methodology to select lossy compression tolerances that do not impact the simulation's numerical accuracy;
- provides the ability to assign order of accuracy to compression tolerances.

2. ERROR TOLERANCE SELECTION

Prior work on using lossy compression to improve checkpoint restart has relied on either significant knowledge of what is happening in the simulation or a trial and error based approach to determine the compression tolerance [3, 4, 2]. The poster's approach exploits the fact that many HPC applications approximate the solution to PDEs or ODEs. Approximations are accurate up to a given tolerance based on the problem's discretization and the truncation error of numerical method used. The level of accuracy given by the truncation error gives us an upper bound on compression error tolerances. A compression tolerance less that truncation error produces results that are numerically identical, but not bit-reproducible to the original. A compression tolerance greater than truncation error will add error into the state variables that can impact results, but does increase the compression factor. This poster selects compression error tolerances that are less than the truncation error of the problem.

To understand truncation error let us look at an approximation of $u(x+h)$ by Taylor Series expansion, method used in deriving and analysis of numerical methods to solve PDEs and ODEs:

$$u(x+h) = u(x) + u'(x)h + \frac{u''(x)h^2}{2} + \mathcal{O}(h^3) \qquad (1)$$

This Taylor Series truncates $u(x+h)$ to second order accuracy. Therefore, any error added that is less than $\mathcal{O}(h^2)$ results in a numerically equivalent approximation to $u(x+h)$ according to the accuracy specified when truncating the Taylor Series.

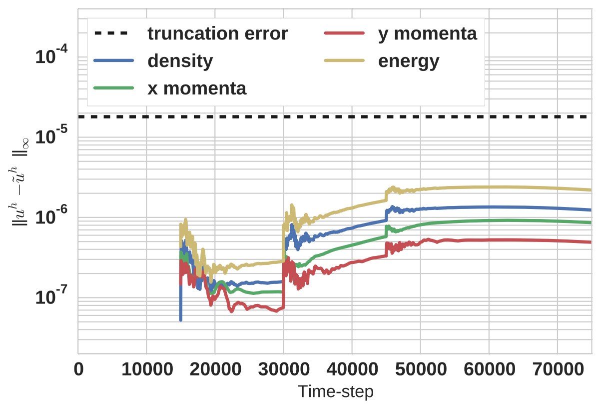

Figure 1: Max-norm between numerical solution u^h and compressed numerical solution $\tilde{u}^h$. There is no error before the simulation restarts at time-step 15000. At each restart (time-step 15000, 30000 and 45000) there is a spike in the error that is partially removed by selection of non-periodic boundary conditions.

3. EXPERIMENTAL RESULTS

3.1 Testing Methodology

The selection methodology is evaluated using the application PlasComCM[1] on Blue Waters solving a 2D Navier-Stokes flow past a fixed cylindrical object with $h_x = h_y = 0.065$. Runge-Kutta 4 advances the simulation and is accurate to $1.8e{-}5$. SZ [1] v1.3 compresses checkpoint files with tolerance $1e{-}6$ and yields an average compression factor of 7x.

3.2 Results

To verify that error resulting from lossy compression does not impact the simulation's results, the simulation is run for 75000 times-steps. At time-step 15000, 30000, and 45000 the simulation restarts from a lossy compressed checkpoint. Figure 1 plots the max-norm between state variables of a simulation restarted from a lossy checkpoint and one restarted from a normal checkpoint. With every restart we see an initial spike in error. This is the error immediately added into the simulation via the lossy compressor. Between the first and second restart, we see reduction in error due to the problem's non-periodic boundary conditions that permit error to escape the domain. After the second and third restart we see an increase in error due to it being accumulated in regions of the domain with little momentum. Areas of low momentum do not permit error to exit the domain quickly. Despite the accumulation, the error in each state variable remains less that the simulation's truncation error. Therefore, both simulations are numerically equivalent.

The checkpointing frequency selected here corresponds to a worst-case scenario for exascale resiliency, MTBF of 18 minutes. Using the system recommended checkpointing interval results in virtually all error leaving the domain before the next checkpoint is taken dramatically reducing error accumulation.

4. CONCLUSION

Lossy compression is emerging as an attractive way to shrink HPC data set sizes for applications that are I/O bound. For lossy compression to be used in practice understanding the error induced is critical. This poster presents the idea of interpreting compression error as numerical error. Selecting an error tolerance less than the truncation error of the simulation allows the simulation to be restarted from a lossy compressed checkpoint without effecting the simulation results. Experiments on PlasComCM show results are numerically equivalent (not bit-reproducible) with error in all checkpointed variables remaining less than the truncation error of the simulation.

Acknowledgment

This work was sponsored by the Air Force Office of Scientific Research under grant FA9550-12-1-0478. This research is part of the Blue Waters sustained-petascale computing project, which is supported by the National Science Foundation (awards OCI–0725070 and ACI–1238993) and the state of Illinois. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications.

5. REFERENCES

[1] S. Di and F. Cappello. Fast error-bounded lossy hpc data compression with sz. In *Proceedings of the 2016 IEEE International Parallel and Distributed Processing Symposium*, IPDPS '16, pages 730–739, Washington, DC, USA, 2016. IEEE Computer Society.

[2] D. Laney, S. Langer, C. Weber, P. Lindstrom, and A. Wegener. Assessing the effects of data compression in simulations using physically motivated metrics. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC '13, pages 76:1–76:12, New York, NY, USA, 2013. ACM.

[3] X. Ni, T. Islam, K. Mohror, A. Moody, and L. V. Kale. Lossy compression for checkpointing: Fallible or feasible? In *Poster Session of the 2014 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '14, Washington, DC, USA, 2014. IEEE Computer Society.

[4] N. Sasaki, K. Sato, T. Endo, and S. Matsuoka. Exploration of lossy compression for application-level checkpoint/restart. In *Proceedings of the 2015 IEEE International Parallel and Distributed Processing Symposium*, IPDPS '15, pages 914–922, Washington, DC, USA, 2015. IEEE Computer Society.

[5] M. Snir, R. W. Wisniewski, J. A. Abraham, S. V. Adve, S. Bagchi, P. Balaji, J. Belak, P. Bose, F. Cappello, B. Carlson, A. A. Chien, P. Coteus, N. A. DeBardeleben, P. C. Diniz, C. Engelmann, M. Erez, S. Fazzari, A. Geist, R. Gupta, F. Johnson, S. Krishnamoorthy, S. Leyffer, D. Liberty, S. Mitra, T. Munson, R. Schreiber, J. Stearley, and E. V. Hensbergen. Addressing failures in exascale computing. *International Journal of High Performance Computing Applications*, 28(2):127–171, May 2014.

[6] S. W. Son, Z. Chen, W. Hendrix, A. Agrawal, W. keng Liao, and A. Choudhary. Data compression for the exascale computing era - survey. *Supercomputing frontiers and innovations*, 1(2), 2014.

[1] http://xpacc.illinois.edu/