Lossy compression in violent thunderstorm simulations: Lessons learned and future goals

Leigh Orf and Kelton Halbert

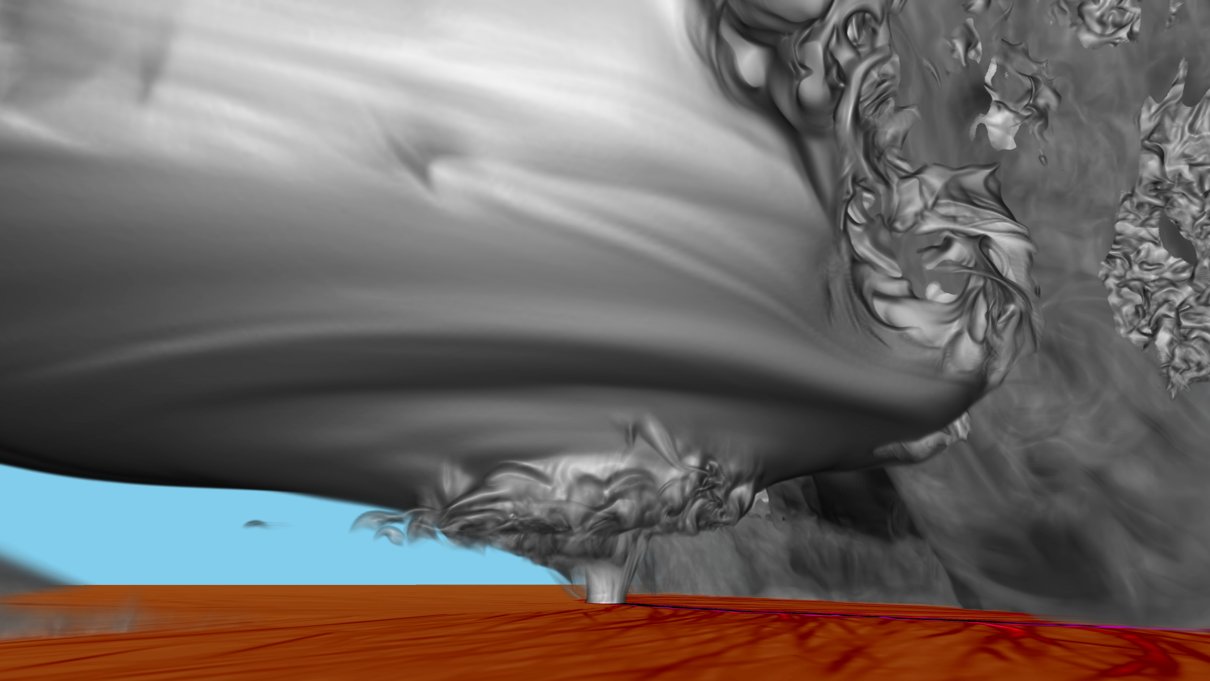
Cooperative Institute for Meteorological Satellite Studies
Department of Atmospheric and Oceanic Sciences
University of Wisconsin - Madison

2022 EGU General Assembly (virtual)
Vienna, Austria

25 May 2022

Atmospheric research goal: Understanding tornadoes in (simulated) thunderstorms

- Our research involves conducting tornado-resolving thunderstorm simulations on supercomputers using a numerical model
- Saved data may be analyzed for years, even for a simulation that only takes days
- Post-processing and visualization will require significantly modest hardware compared to what was required to run the simulation
- High temporal frequency saves of full-resolution data is required in order to capture rapidly changing flow fields associated with tornadoes

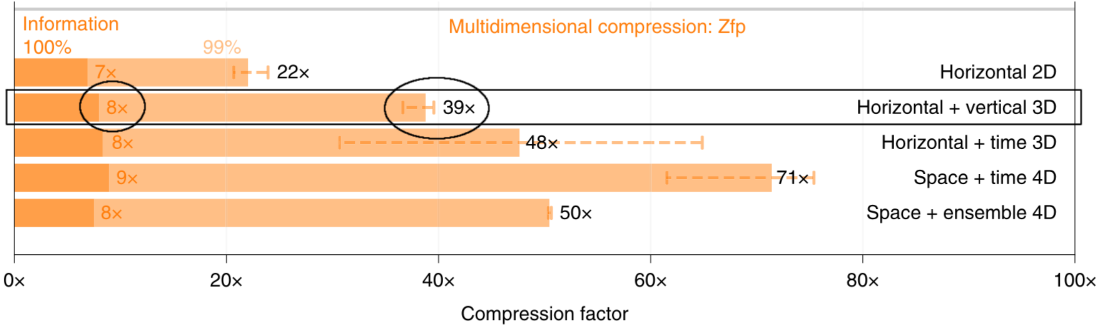


Lack Of a File System (LOFS)

- Default save options for CM1 model do not scale well to large core counts (I/O becomes a severe bottleneck)
- I realized a different approach to I/O was needed, and after several iterations, I developed what I call LOFS (Lack Of a File System, so named because it's a file-based file system)
- **The goal of writing to LOFS is to remove the I/O bottleneck while saving 3D data frequently to enable many years of *post hoc* analysis**

Identifying “false information” in NWP data (Klöwer et al)

Klöwer, M., M. Razinger, J. J. Dominguez, P. D. Düben, and T. N. Palmer, 2021: **Compressing atmospheric data into its real information content.** *Nature Computational Science*, **1**, 713–724. http://dx.doi.org/10.1038/s43588-021-00156-2



- **99% of information is preserved with ∼40:1 compression ratios**

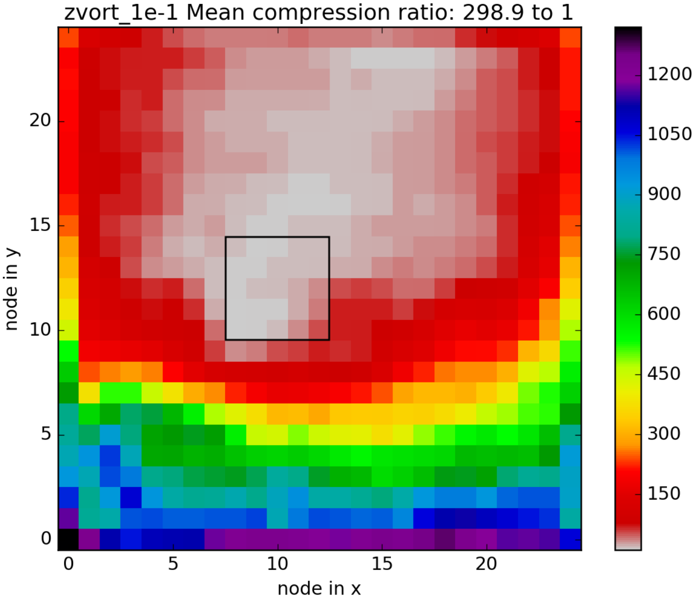
LOFS

- LOFS files are HDF5, written serially, but concurrently to the parallel file system, 1 HDF5 file per shared-memory node per disk save
- Multiple (e.g., 50-100) time levels are buffered before flushing to disk, with files being grown in memory using the HDF5 core driver
- Lossy floating point compression (ZFP) is used for all 3D data, drastically reducing I/O as compared to lossless/uncompressed

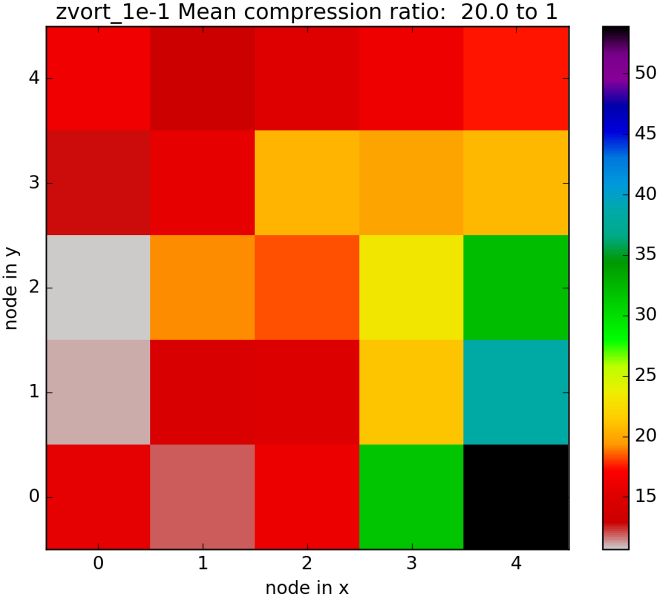
LOFS

- We compress as much as possible, and save only what is needed to calculate all other diagnostic quantities
- Diagnostic calculations are available as read options, calculated on the fly from (ZFP compressed) LOFS saved data
- Utilize multicore architectures for diagnostic calculations, parallelize all calculations in time (embarrassingly parallel) - “high throughput” computing
- This approach is designed to reduce I/O and to minimize the size of the data on disk
- Saving only subdomains saves a LOT of space

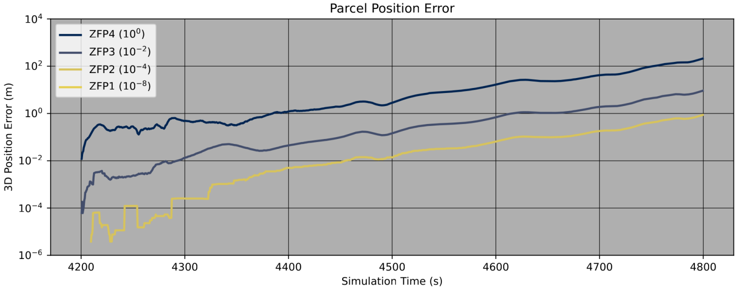
ZFP compression ratios as a function of LOFS file

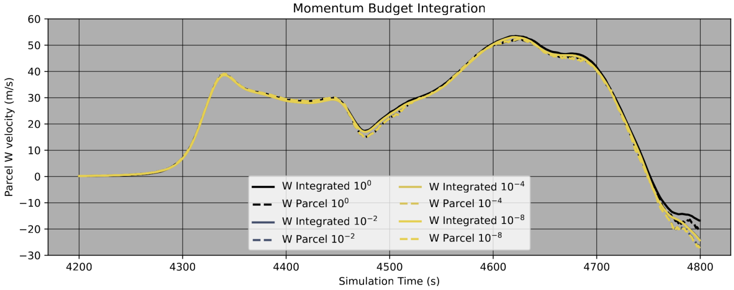


ZFP compression ratios as a function of LOFS file



Error associated with ZFP compression (Lagrangian parcels)


Parcel Position Error


Momentum Budget Integration

Lessons Learned

1. On supercomputers, I/O can easily be a huge bottleneck
2. Dealing head-on with the I/O bottleneck can enable breakthrough scientific discoveries in some situations
3. We modelers should all be using lossy floating point compression!
4. You can get away with a surprisingly high amount of compression and still come to the same scientific conclusions
5. You *can* overdo it, however!!

Future Goals

1. Explore 4D compression with ZFP
2. Find appropriate fixed-accuracy parameter for a wide amount of variables / situations
3. Increase performance for large HPC runs
4. Write Python driver/module for LOFS format (‘import lofs’)

Acknowledgments

Simulations and visualizations conducted on the NSF-sponsored Blue Waters and Frontera supercomputers. Leigh Orf is supported by NSF grants AGS-1832327, OAC-1663954, AGS-2114757 and the Space Science and Engineering Center at the University of Wisconsin.

Collaborators on this and related projects: Dr. Catherine Finley (UND); Dr. Bruce Lee; Dr. Matthew Turk (NCSA); Drs. Lawrence Frank and Vitaly Galinsky (UCSD); Hank Schyma ("Pecos Hank"), storm videographer from Houston; Kelton Halbert, UW grad student

LOFS code, documentation:
https://lofs.io
https://github.com/leighorf/cm1r19.8-LOFS
https://github.com/leighorf/LOFS-read
Peter Lindstrom's ZFP: https://computing.llnl.gov/projects/zfp
See https://orf.media to access high quality video files of many simulations saved with ZFP compressed data.