ADDITIONAL NETCDF COMPRESSION OPTIONS WITH THE COMMUNITY CODEC REPOSITORY

Edward Hartnett [1,2], Charlie Zender[3]

1. CIRES, University of Colorado, Boulder, CO 80309, USA, 2. NOAA/ESRL/GSD, Boulder, CO 80305, USA
3. Departments of Earth System Science and Computer Science, University of California, Irvine, CA 92617, USA

Summary

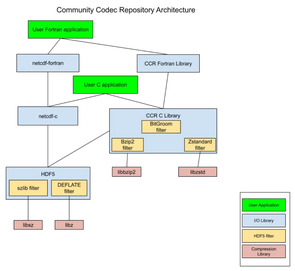
- Since netCDF-4.0 (2008), netCDF can create files with compressed data, using the zlib library.
- Once turned on, the compression/decompression is transparent to the user, but does add time to write and read the data.
- Since netCDF-4.7.4 (2020) compression has also worked with parallel I/O.
- The upcoming netcdf-c-4.8.0 release contains good support for HDF5 filters, which can be used to apply additional compression techniques.
- The recently introduced Community Codec Repo (CCR) project brings new compression filters to netCDF C and Fortran codes.

HDF5 Filters

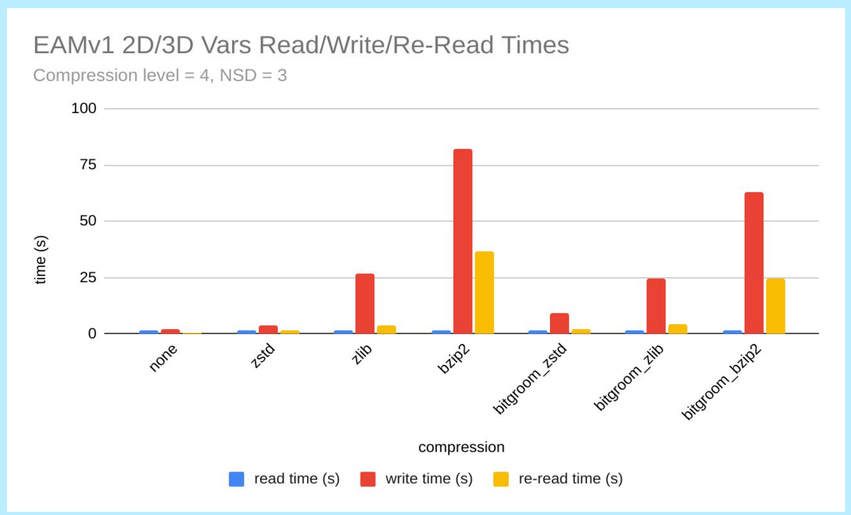
- HDF5 allows chunked data to pass through user-defined filters on the way to or from disk.
- Several filters are built in to HDF5 and are supported in netcdf-c: checksum, shuffle.
- zlib compression is external to HDF5 but supported as a built-in filter with HDF5 and is required in all netcdf-c builds.
- szip compression is supported as a built-in filter if present in HDF5 at HDF5 build-time.
- Generic single filter support was added in netcdf-c-4.6.0 (2018), full support for multiple filters will be in netcdf-c-4.8.0 (2021).

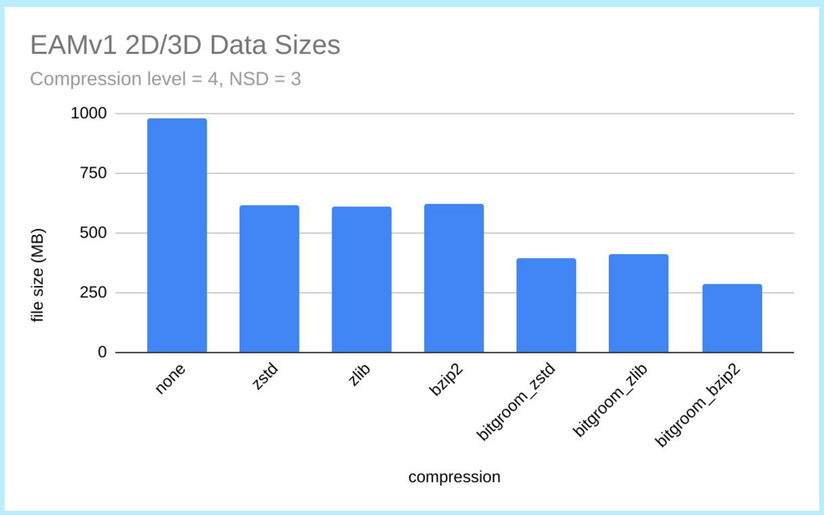
Role of CCR

- CCR provides a curated selection of useful filters, plus the glue code to make them easy to use in netCDF C and Fortran codes.
- CCR provides a single tarball which builds and install the filters and the ccr library, containing the glue code.
- User C/Fortran codes can then turn on additional compression, similarly to how zlib can be turned on for a variable.
- Writers and readers of the data must have the filter installed, or the data cannot be accessed.



Faster, more compressive, and lossy compression are available for netCDF C/Fortran codes, using the CCR project.





https://github.com/ccr/ccr

Available Filters

- The CCR 1.1.0 release contains the following filters:
  - bzip2
  - Zstandard
  - Bitgroom

bzip2

- Like zlib, is a command line tool and a library.
- Burrows-Wheeler block sorting text compression algorithm, and Huffman coding.
- Slower than zlib, but, when used with bitgroom filter, resulted in smallest output.

Zstandard

- Open-source compression library from Facebook.
- Similar compression to zlib, but much faster.

Bitgroom

- Bitgroom is pre-filter for lossy compression.
- For float/double only. Cannot be applied to ints.
- It must be used with another compression filter to reduce data size.
- Sets unneeded bits to 1/0 (alternates, to keep averages the same).
- User specifies number of significant digits to keep, up to 7 for float, 14 for double.

Performance with Climate Data

- For performance test, copy all 2D/3D float vars from a climate data file to new file.
- Then re-read the file.
- Data was January monthly mean output from the first year of the Pre-Industrial control simulation of EAMv1, the atmospheric component of E3SMv1.
- ~1 GB data file with 353 2D vars (1 x 48602), 65 3D vars (1 x 72 x 48602).

Conclusions

- Remember: readers of data must also have filter installed!
- For absolute smallest files, bzip2 + bitgroom. But this is slow to write and read!
- Use Zstandard for much faster zlib-like compression - almost order of magnitude improvement in write speed, 3x improvement in read speed.
- Add bitgroom filter for additional lossy compression.