Using A Cloud -Friendly Data Format in Earth System Models

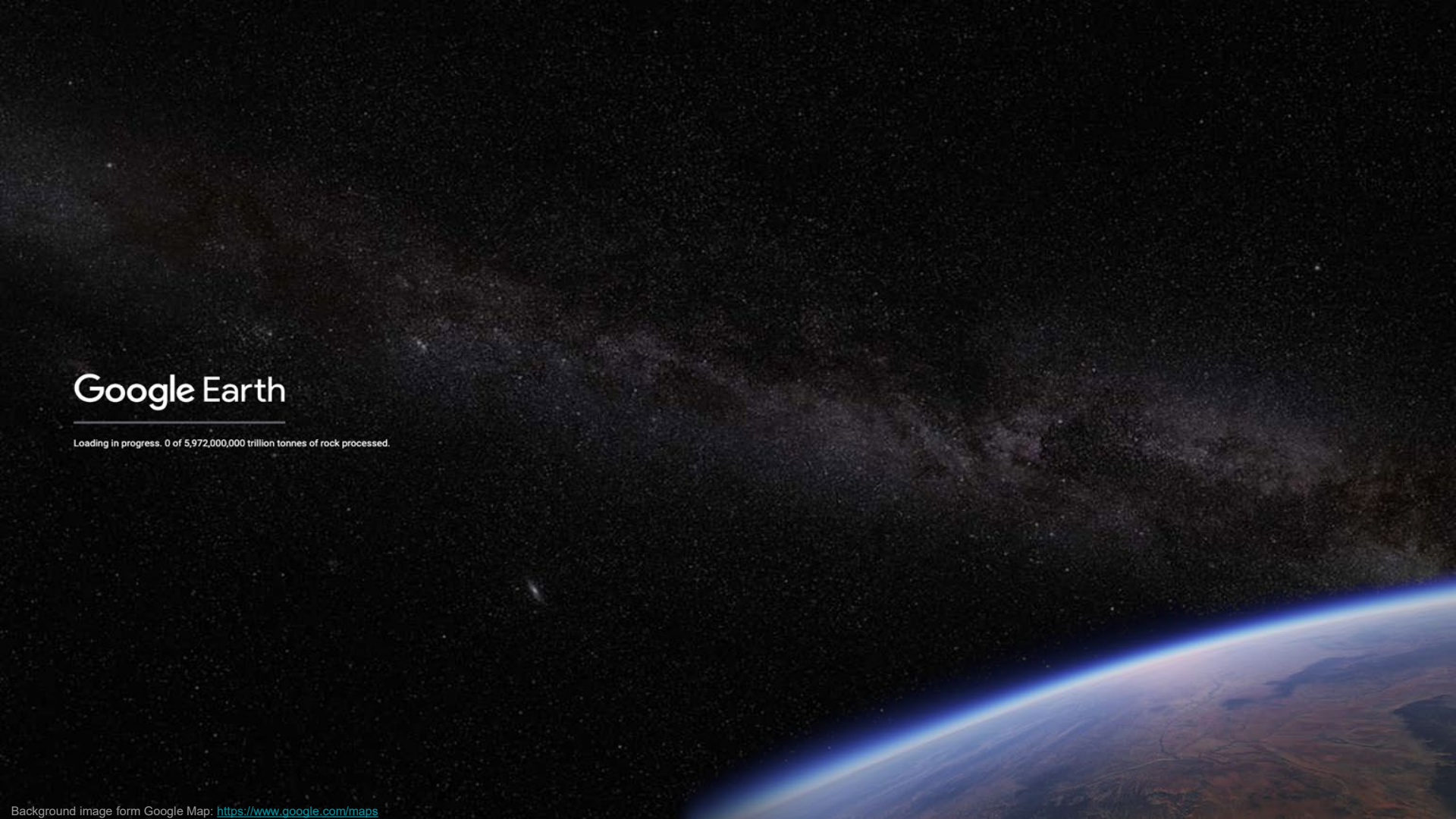
Weile Wei [1 2] Haiying Xu [2] John Dennis [2] Kevin Paul [2]

1 Louisiana State University
2 National Center for Atmospheric Research
Boulder, CO | July 30th, 2019 | weilewei09@gmail.com

This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement No. 1852977.

Outline

1. Background
2. Integrating Z5 into Community Earth System Model (CESM)
3. Performance Analysis
4. Conclusion
5. Future work



Background image form Google Map: https://www.google.com/maps

Community Earth System Model

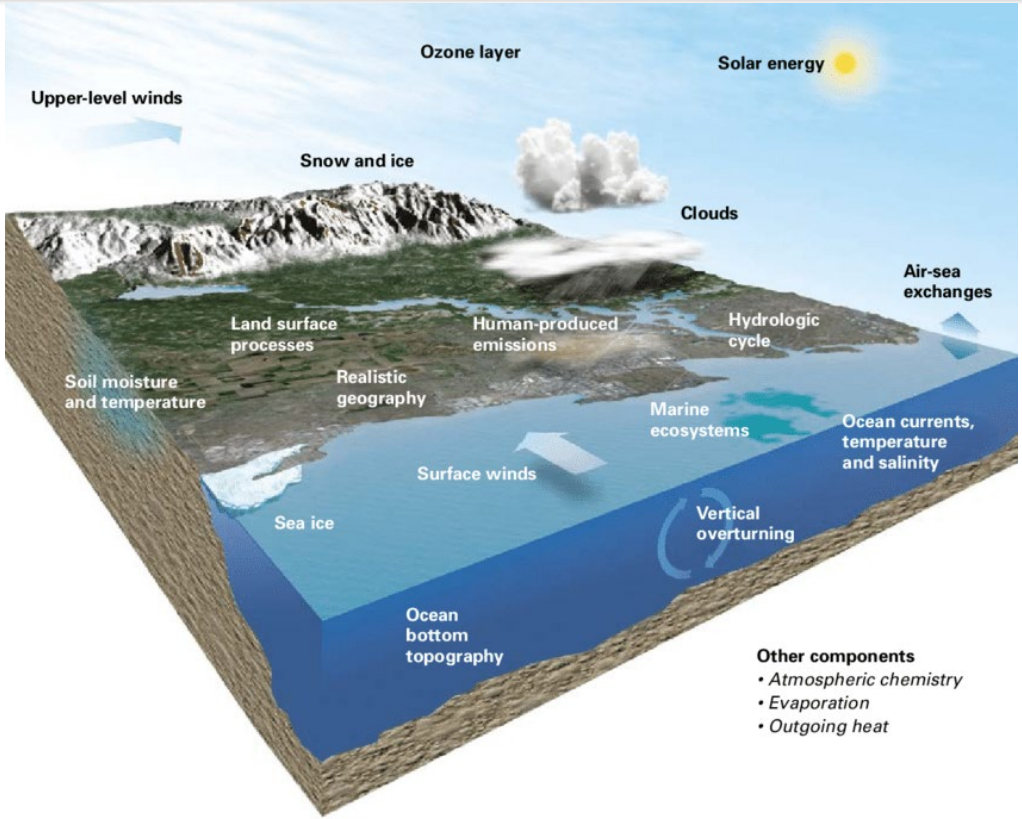


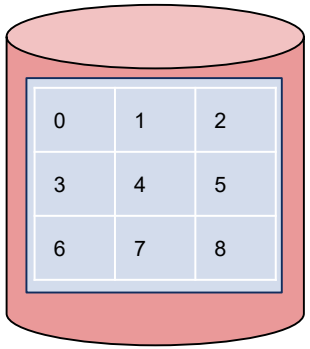
Figure 1. Community Earth System Model

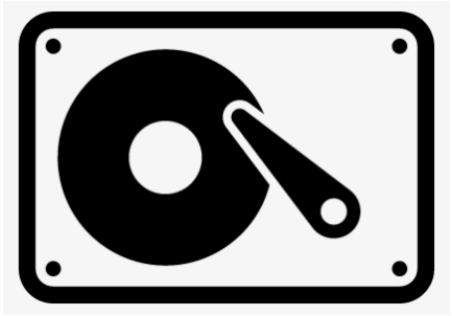
- CESM provides computing simulations of earth's past present, and future climate states
- CESM allows investigation of problems including climate, weather, earth, the water cycle, etc.
- CESM's traditional data format is NetCDF

Image source: National Center for Atmospheric Research, United States of America. Image can be found at http://www2.ucar.edu/sites/default/files/news/2011/CESM_final.jpg.

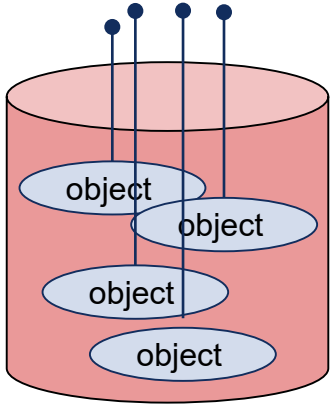
Storage System

Harddrive-based Block Storage (NetCDF)



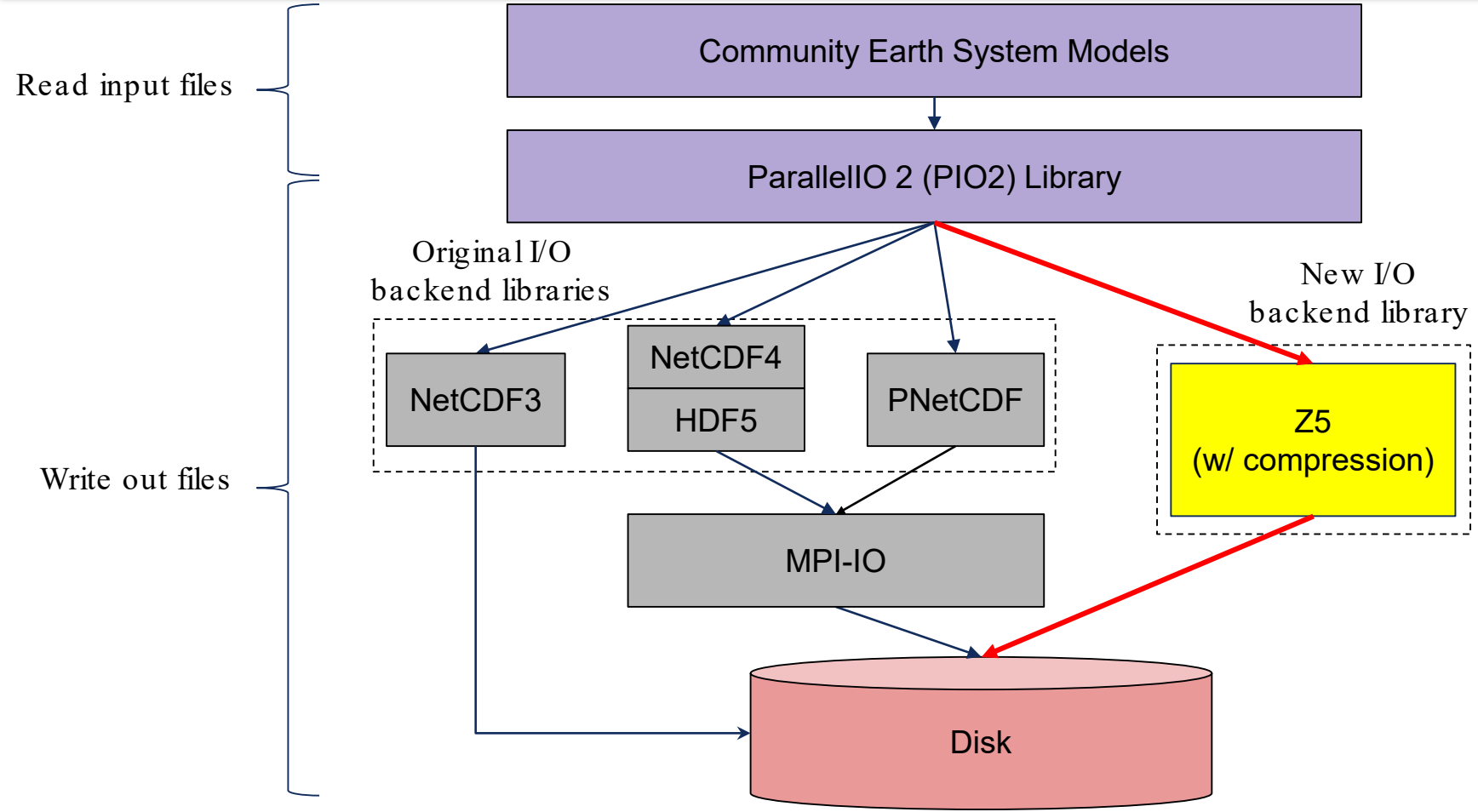


Cloud Object Storage (Z5)



- NetCDF data formats are used in harddrive-based block storage and are difficult to access in object -based cloud storage system. A cloud -friendly data format is needed for the CESM simulation in the cloud.
- NetCDF developers are planning to add Z5 as a new backend.

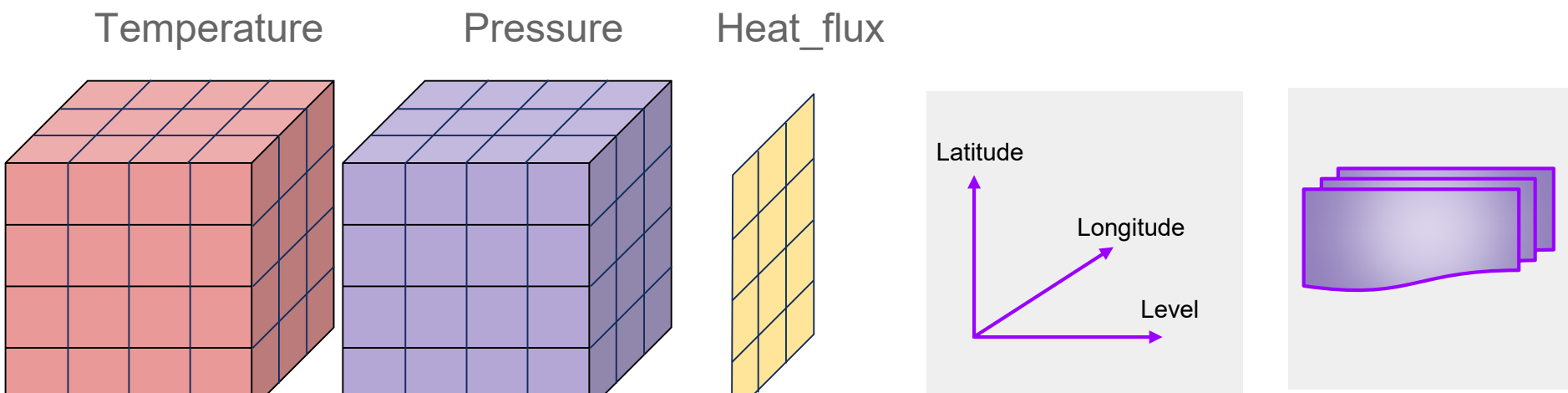
CESM I/O Workflow



Goal

1. Add Z5 to CESM by integrating it into the ParallelIO 2 (PIO2) library.
    - PIO2: A high-level Parallel I/O Library, backed by MPI (Message Passing Interface)
    - PIO2 currently supports NetCDF data formats
    - Z5 is a cloud-friendly data format and a C++ package providing an implementation of compressed, chunked, N-dimensional arrays, designed for use in parallel computing
    - Write C API Wrapper for Z5
2. Analyze the I/O performance via CESM simulation.

Data Model

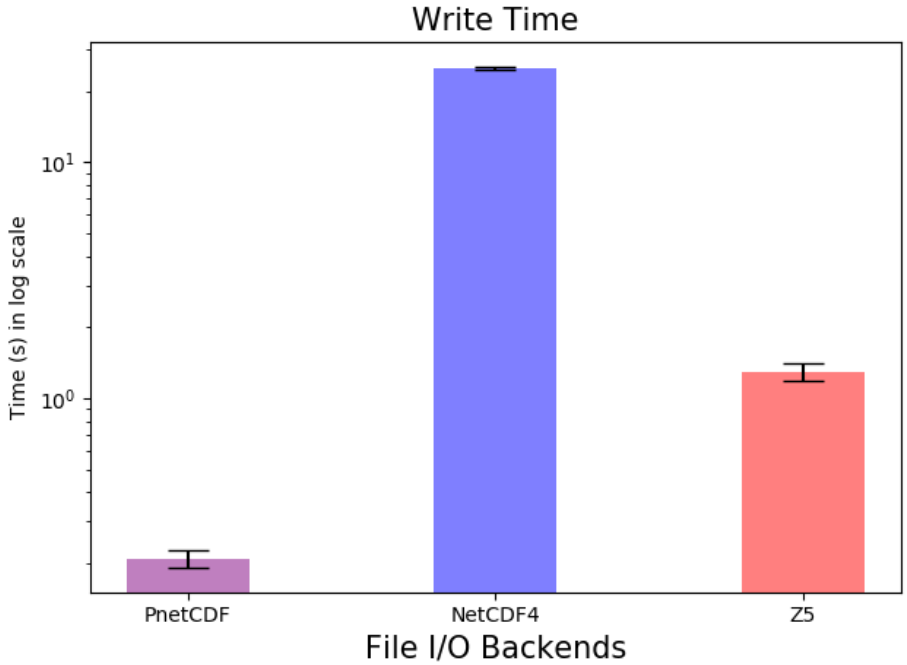


Adapted by https://medium.com/pangeo/step-by-step-guide-to-building-a-big-data-portal-

We add Z5 into PIO2 as an alternate file I/O backend

```
int PIOc_createfile ( int iosysid, int *ncidp, int *iotype, const char *fname, int mode);
        if(file - >iotype == PIO_IOTYPE_NETCDF)
            nc_create(const char *path, int cmode, int* ncidp);
    else if (file - >iotype == PIO_IOTYPE_Z5)
            z5CreateFile ( char * path);
        // ...
int PIOc_def_var ( int ncid, const char *name, nc_type xtype, int ndims,
            const int *dimidsp, int *varidp)
        if(file - >iotype == PIO_IOTYPE_NETCDF)
        nc_def_var(int ncid, const char* name, nc_type xtype,
                                    int ndims, const int* dimidsp, int* varidp);
    else if (file - >iotype == PIO_IOTYPE_Z5)
        // It supports multiple data types: int8, int16, int32, int64, double, float, uint8 ...
        z5CreateFloat32Dataset ( char *path, unsigned int ndim, size_t *shape,
                                    size_t *count, int cuseZlib = 1, int level =
1);
```

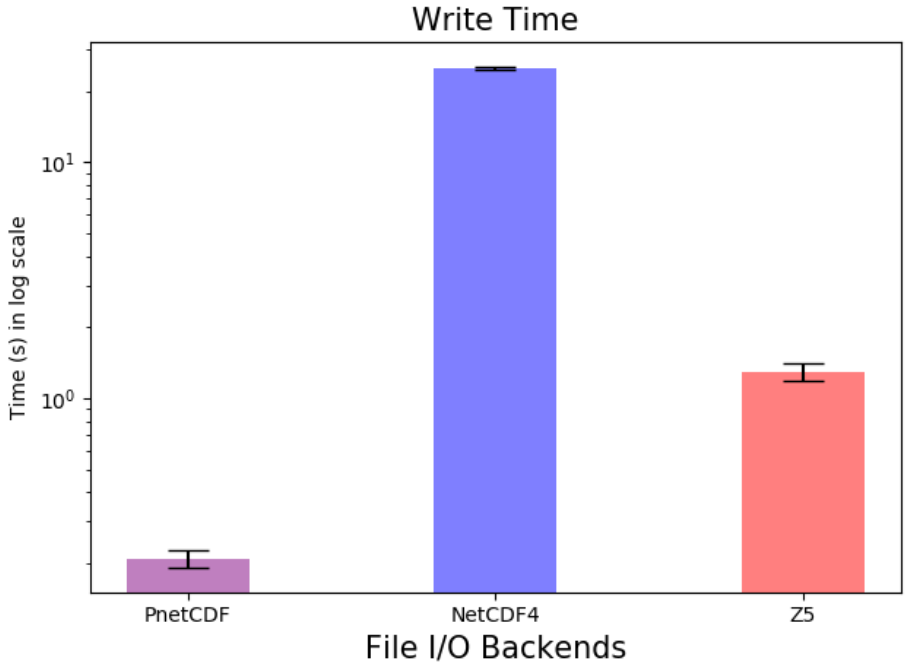
Performance Results in 1 -degree Res. on 30 nodes



Experiment Settings:

The experiment is conducted in 1 degree resolution of CESM simulation on 30 nodes on Cheyenne Supercomputer using Intel compiler. It is similar to the run in production.

Performance Analysis



Preliminary results:

- PnetCDF is fast (~0.2s), however, it does not have compression capability.
- NetCDF4 (~24.9s) has compression but has poor performance
- Z5 is much faster (~1.3s) than NetCDF4 and has compression enabled. Though, Z5 is slower than PnetCDF by 10% in total CESM simulation time.

Project Impact

Codebase Contribution:

1. Contributed **1400+** lines of code for C API wrapper for Z5

https://github.com/kmpaul/cz5test/

1. Contributed **3200+** lines of code for PIO2-Z5 integration

https://github.com/weilewei/ParallelIO

External Impact:

1. Solved Z5 issue C API wrapper for z5 #68

1. 2 accepted pull requests in Z5
   - fix file creation and add nlohmann_json support in CMakeLists #115
   - add writeMetadata for the file handle #114

Conclusion

1. **Low learning curve** : In PIO2, user can reuse same API and workflow to do file I/O with Z5 backend

1. **New I/O backend:** Z5 is a feasible file I/O backend for CESM and is cloud-friendly

1. **Performance:** Z5 has adequate performance to PnetCDF, is much faster than NetCDF and has compression capability.

Future Work

1. To test Z5 supported CESM in cloud services (i.e. AWS S3, Google Cloud, Microsoft Azure)

2. To study the scaling performance of Z5 I/O backend in CESM

Acknowledgement

Many thanks to my mentors Haiying Xu, John Dennis, Kevin Paul!

Many thanks to SIParCS fellows and AJ Lauer, Virginia Do, Eliott Foust, and Blake Lewis!

Q&A

Using A Cloud -Friendly Data Format in Earth System Models

Weile Wei [1 2] Haiying Xu [2] John Dennis [2] Kevin Paul [2]

1 Louisiana State University
2 National Center for Atmospheric Research
Boulder, CO | July 30th, 2019 | weilewei09@gmail.com

This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsor ed by the National Science Foundation under Cooperative Agreement No. 1852977.