Big Arrays, Fast: Profiling Cloud Storage Read Throughput

Ryan Abernathey[1]

[1]Columbia

November 22, 2022

Abstract

As the size of geoscience datasets grows, scientists are eager to move away from a download-based workflow, where data files are downloaded a local computer for analysis, towards a more cloud-native workflow, where data is loaded on demand over the network. On-demand data loading offers several advantages, including increased reproducibility, provenance tracking, and, potentially, scalability using distributed cloud computing. In this notebook, we demonstrate how to load data on-demand using three different remote data access protocols: - OPeNDAP, the most common, well-established protocol - NetCDF over HTTP, enabled by the h5py library - Zarr over HTTP, a new format optimized for cloud object storage (e.g. Amazon S3) We then conduct a simple benchmarking exercise to explore the throughput and scalability of each service. We use Dask to parallelize reads from each access protocol and calculate the throughput as a function of number of parallel reads. One conclusion is that Zarr over HTTP, coupled with cloud object storage, shows favorable scaling up to hundreds of parallel processes. Finally, we compare the throughput of Zarr over HTTP on a few different clouds, including Google Cloud Storage, Jetstream Cloud, Wasabi Cloud, and Open Storage Network.

Posted on 22 Nov 2022 — CC-BY 4.0 — https://doi.org/10.1002/essoar.10508242.2 — This a preprint and has not been peer reviewed. Data may be preliminary.

Big Arrays, Fast: Profiling Cloud Storage Read Throughput

Ryan Abernathey

As the size of geoscience datasets grows, scientists are eager to move away from a download-based workflow, where data files are downloaded a local computer for analysis, towards a more cloud-native workflow, where data is loaded on demand over the network. On-demand data loading offers several advantages, including increased reproducibility, provenance tracking, and, potentially, scalability using distributed cloud computing.

In this notebook, we demonstrate how to load data on-demand using three different remote data access protocols:

- OPeNDAP, the most common, well-established protocol
- NetCDF over HTTP, enabled by the h5py library
- Zarr over HTTP, a new format optimized for cloud object storage (e.g. Amazon S3)

We then conduct a simple benchmarking exercise to explore the throughput and scalability of each service. We use Dask to parallelize reads from each access protocol and calculate the throughput as a function of number of parallel reads. One conclusion is that Zarr over HTTP, coupled with cloud object storage, shows favorable scaling up to hundreds of parallel processes.

Finally, we compare the throughput of Zarr over HTTP on a few different clouds, including Google Cloud Storage, Jetstream Cloud, Wasabi Cloud, and Open Storage Network.