Accessing Data Stored in Amazon S3 Using the Hyrax OPeNDAP Server

Fall 2018 AGU

James Gallagher
EED-2 Contractor
jgallagher@opendap.org

Nathan Potter
EED-2 Contractor
ndp@opendap.org

David Fulker
EED-2 Contractor
dfulker@opendap.org

This work was supported by NASA/GSFC under Raytheon Co. contract number NNG15HZ39C.
This document does not contain technology or Technical Data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations.

Outline

- Background
- Optimizations
- Improvements
- Conclusion

Background

- Hyrax can serve data stored on S3 in a way that is competitive with data stored on a spinning disk
- Several approaches are evaluated
- We show that caching metadata, parallel access and connection reuse all provide significant improvements when accessing data from S3

Software Architectures Evaluated

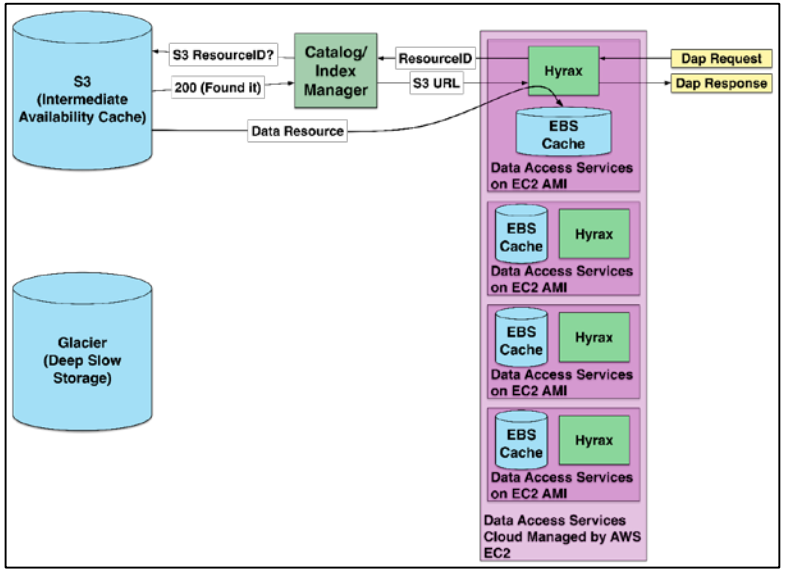
- Caching
- Subsetting
- Baseline - reading from a spinning disk
- *All of these ran in the AWS environment*

Caching Architecture

- Data are stored on S3 as files
- Files are transferred from S3 to a spinning disk cache (EBS, EFS)
- Data are read from the cached files and returned to clients

Advantages: Works with any file, easy to use with legacy software, files easy to obtain, minimal configuration metadata needed

Disadvantages: Initial cost to transfer the whole file, slower than the subsetting architecture



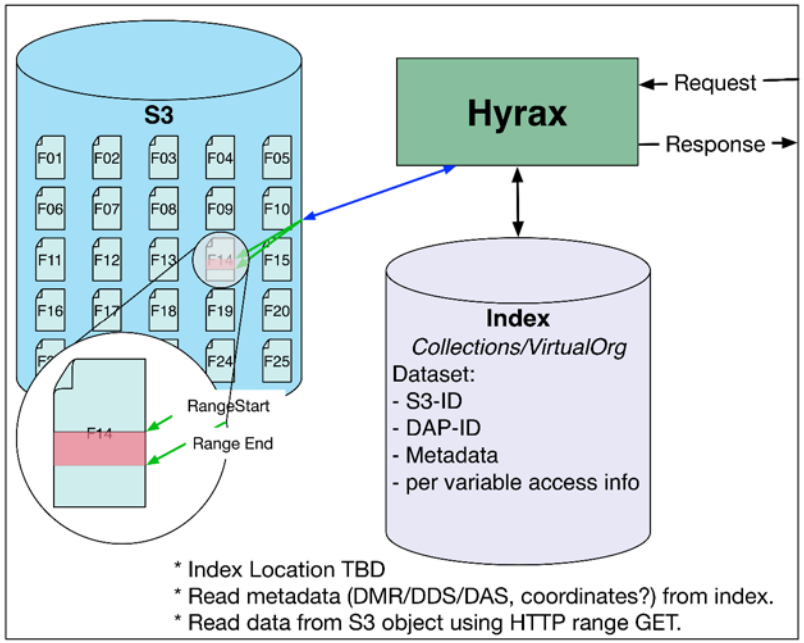
Subsetting Architecture - Virtual Sharding

- Data are stored on S3 as files (HDF5)
- Data are read from S3 by reading parts (virtual shards) of the file

Virtual Sharding: Break a file into virtual pieces. Each shard is defined by its size and position in the file

Advantages: faster than caching, data cache not needed, only data needed are transferred from S3

Disadvantages: effectively a new data format with tricky subsetting issues, more configuration metadata needed



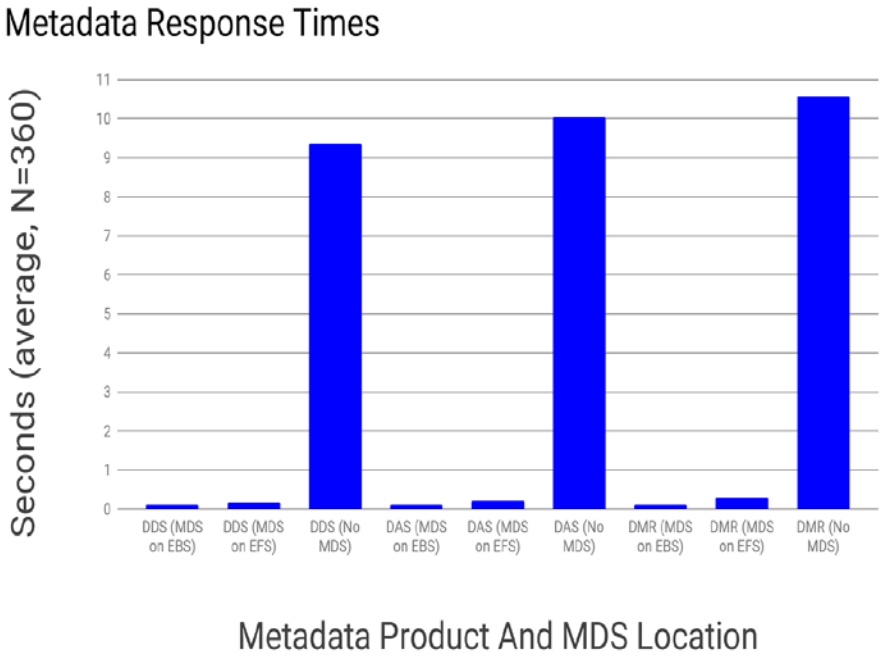
Subsetting Architecture Optimizations

1. Optimized metadata storage
2. Exploit parallel aspects of data access
3. Reuse HTTP 'connections'

Optimize Metadata Storage

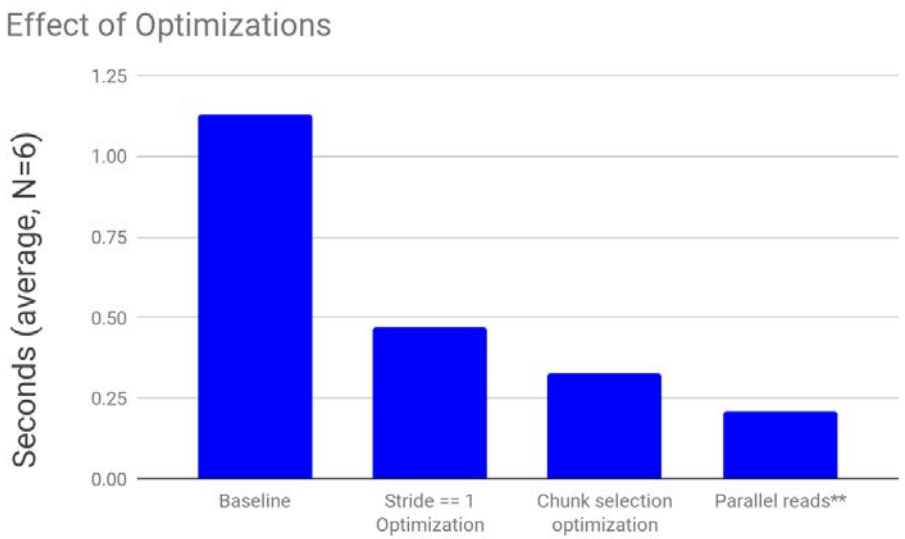
Caching metadata shortens response times

- For data files with O(10^3) variables, two orders of magnitude improvement
- Number of variables and attributes determines time to build a metadata response
- Response time includes time to build and transmit
- The Metadata store holds preformatted responses - they are transmitted without additional encoding
- Objects in the Metadata Store are when building data responses



Subsetting Architecture Optimizations

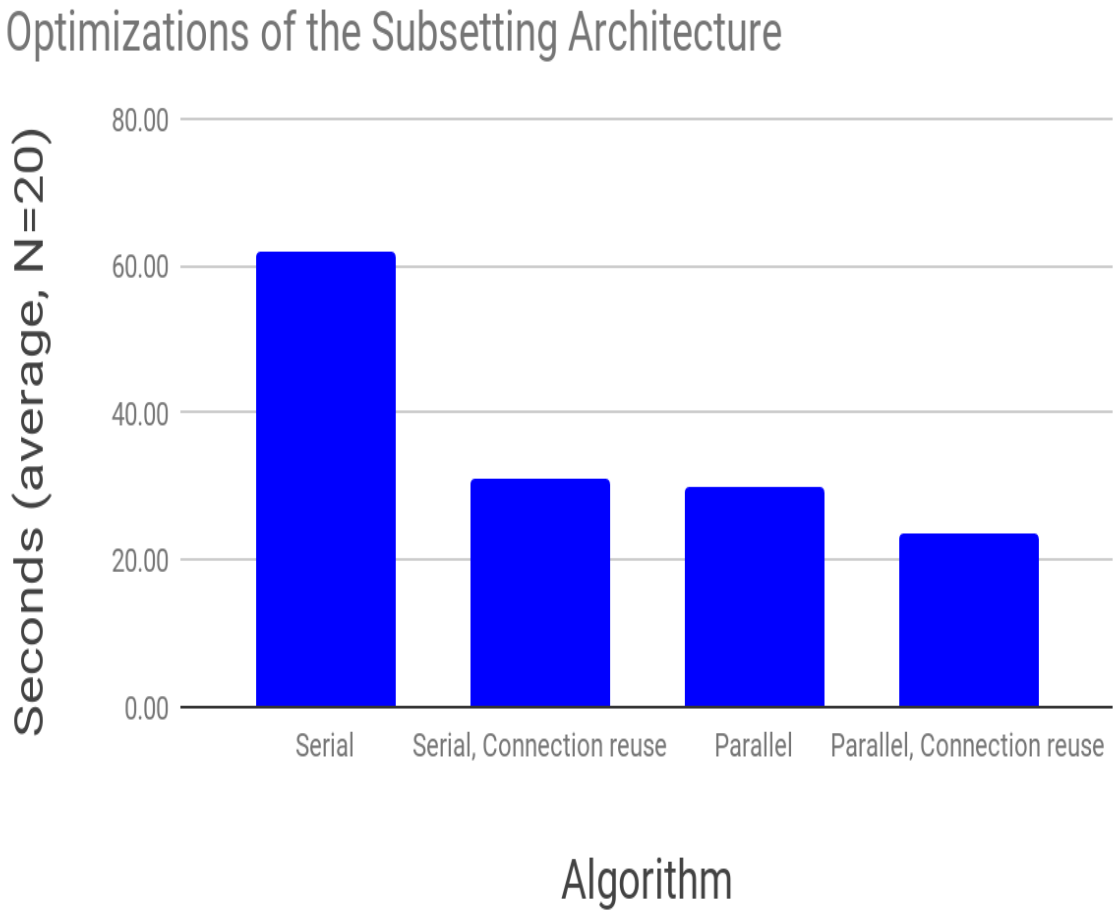
- Greater than 4X improvement
- Special handling of subsetting strides
- Split selection of the virtual shards from transfer and processing
- Parallel transfer of shards minimizes initial costs of transfer



Connection Reuse and Parallelism

Connection, Parallelism reduce S3 transfer times

- Connection reuse provides substantial reduction in transfer time
- Parallel transfers similarly provide reduction in transfer time
- These techniques can be combined for (modestly) increased performance

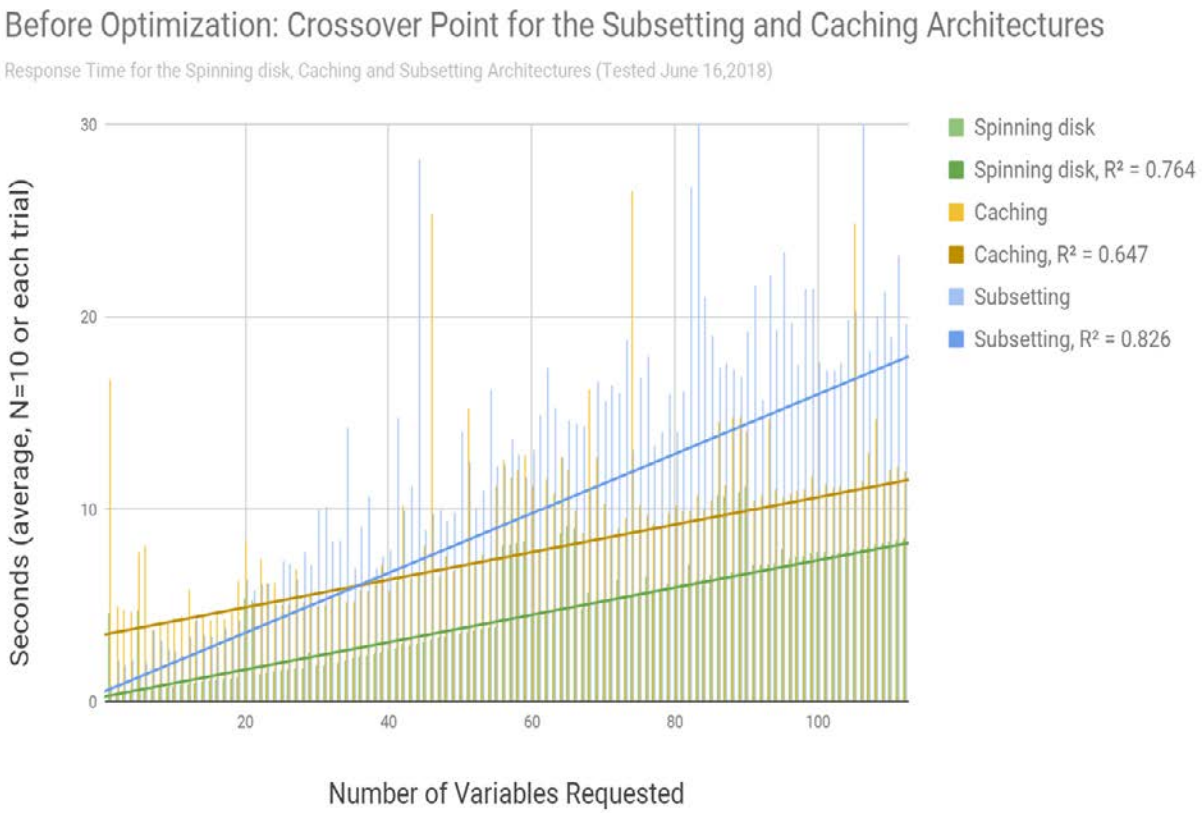


Performance Before Optimizations

Without optimization, caching outperforms the subsetting architecture for some requests*, even though it transfers much more data than needed

*For large HDF5 files with ~1,000 compressed variables, requesting ~40 variables takes longer

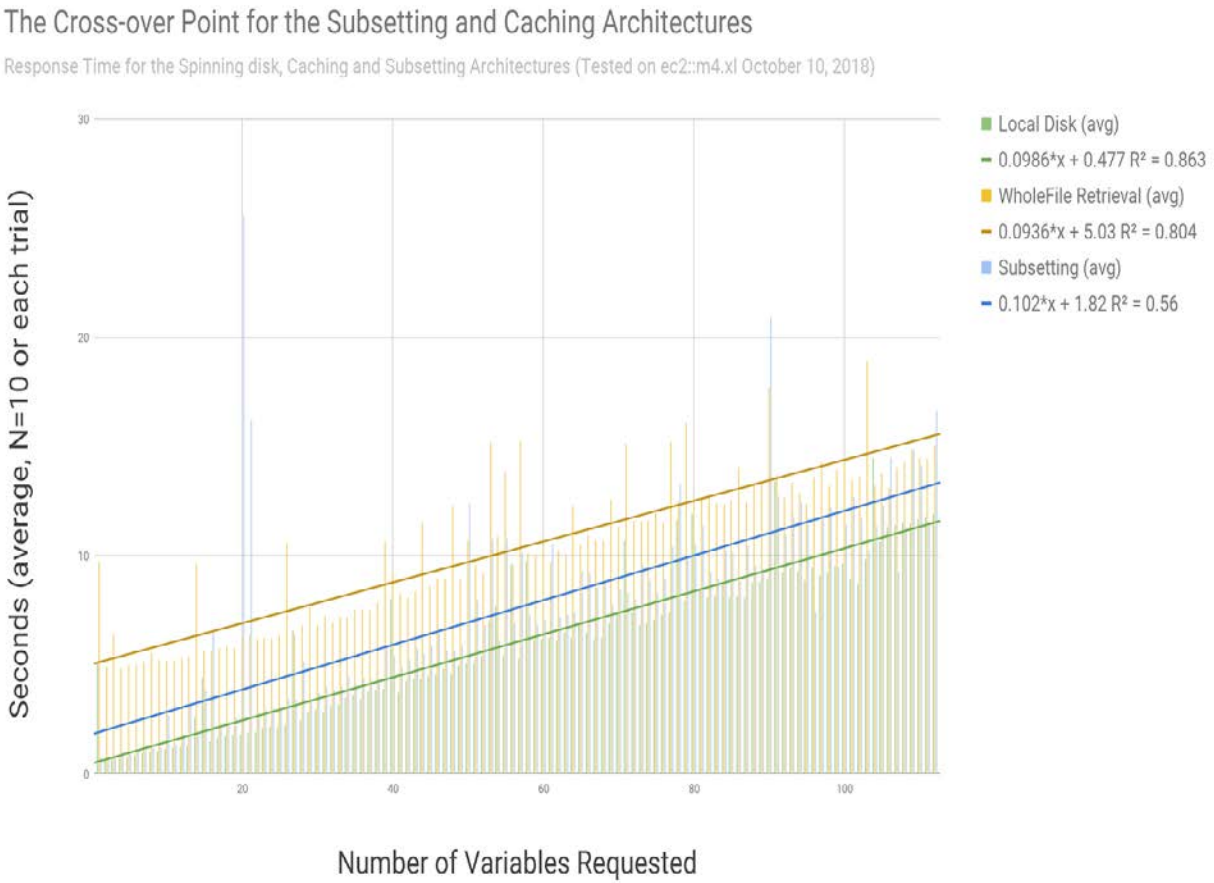
Shown: Caching and subsetting (yellow and blue) and access when data are stored on spinning disk (green)



Performance After Optimizations

After optimization the subsetting algorithm performance exceeds the caching algorithm

Shown: Caching and subsetting (yellow and blue) and access when data are stored on spinning disk (green)



Conclusions

- Optimizing access to S3 can provide large enough performance differences to affect algorithm selection
- The complexity of these improvements is not trivial, so it will benefit users if these optimizations are packaged in a way they can use easily (e.g., a web API)
- These optimizations can be applied to 'legacy' data

This work was supported by NASA/GSFC under Raytheon Co. contract number NNG15HZ39C.

Raytheon

in partnership with

ati

Bellfore Consulting Group

Element 84

Genex Systems

The HDF Group

Innovim The Power of Innovation

Maryland Hawk Corporation

OPeNDAP

REAN Cloud

Red Canyon Engineering & Software

Stinger Ghaffarian Technologies

Valador

EOSDIS