EOSDIS
NASA'S EARTH OBSERVING SYSTEM
DATA AND INFORMATION SYSTEM

Creating Cloud-Optimized HDF5 Files

2022 ESIP Summer Meeting

Aleksandar Jelenak
NASA EED-3/HDF Group
ajelenak@hdfgroup.org

This work was supported by NASA/GSFC under Raytheon Technologies contract number 80GSFC21CA001.

Cloud *Optimized* Storage Format

An already existing **file** format where internal structures are rearranged for more efficient data access when in cloud object stores.

Cloud *Native* Storage Format

A data format specifically developed to take advantage of the distributed nature of cloud computing and the key-value interface of object storage systems.

Counts as cloud optimized as well.

Cloud Native

- One **file** consists of **many** objects in cloud store.
- Data reads access entire objects.
- Data writes are allowed.

Cloud Optimized

- One **file** is **one object** in cloud store.
- Data reads access parts of that object (HTTP* range GET requests).

*Hypertext Transfer Protocol

EOSDIS

Why *COHDF5**?

- Least amount of reformatting from archive tapes to cloud object stores.
- Fast content scanning when files are in an object store.
- HDF5** library instead of a custom limited-feature HDF5 file format reader.

*Cloud-Optimized HDF5
**Hierarchical Data Format Version 5

Properties of *COHDF5*

- Cloud-friendly dataset chunk size.
- Optimal chunk compression (compression ratio vs. decompression speed).
- Minimal use of variable-length datatypes. (Important if not using HDF5 library.)
- File space strategy: Paged Aggregation.

Larger Dataset Chunk Size

- 2-10 MiB* chunk size seems preferrable.
- Default HDF5 library *dataset chunk cache* is only 1 MiB but is configurable *per dataset*.
- Appropriate chunk cache size has significant impact on I/O performance.
- Trade-off between chunk size and compression/decompression speed.

*mebibyte (1,048,576 bytes)

Variable-Length Datatypes

- Current implementation of variable-length data in HDF5 files prevents easy retrieval using HTTP range GET requests.
- NASA Earth Science data in HDF5 have little variable-length data, mostly as strings. This may change for certain machine learning training data.
- Avoid these datatypes if not using the HDF5 library .

HDF5 Paged Aggregation

- One of available file space management strategies. Not the default.
- Can only be selected at file creation.
- Existing files must be reformatted with *h5repack* to change file space strategy.

HDF5 Paged Aggregation (cont'd)

- The library always reads and writes file pages.
- Best suited when file content is added once and never modified.
- File metadata and raw data are organized in separate *pages* of specified size.
- Setting an appropriate page size could have all file metadata in just one page.

HDF5 Page Buffering

- Low-level library cache for file metadata and raw data pages.
- Only available for files created with paged aggregation.
- Page buffer size must be an exact multiple of the file's page size.

An Example HDF5 File

- ICESat-2* ATL03 product.
  - File size: 2,458,294,886 bytes (2.29 GiB**)
  - 171 HDF5 groups
  - 1,001 HDF5 datasets
  - HDF5 file metadata size: 7,713,028 bytes
- Repacked original file with two file space page sizes: 4 and 8 MiB.
  - 4MiB version larger by 4,345,472 bytes (+0.18%)
  - 8MiB version larger by 8,539,776 bytes (+0.35%)

*Ice, Cloud, and Land Elevation Satellite-2
**gibibyte (1,073,741,824 bytes)

EOSDIS

Use Case

- With the three files in Amazon S3 (Simple Storage Service), list their content and the dataset chunk file locations.
- A common task to enable alternative access to HDF5 data.
  - HDF Scalable Data Service (HSDS)
  - OPeNDAP (Open-source Project for a Network Data Access Protocol) DMR++ (Dataset Metadata Response++)
  - *kerchunk* (*xarray*/*Zarr* ecosystem)
- Tested using *h5py* 3.7.0 and HDF5-1.13.1 with Read-Only S3 Virtual File Driver (VFD).

Results – AWS S3

File Version	Page Buffer Size	Total Runtime	Page Buffer Stats
Original	N/A	19 min 18.71 sec	N/A
4 MiB Page	4 MiB	22 min 23.86 sec	accesses=6634, **hits=5691, misses=943, evictions=942**, bypasses=0
8 MiB Page	8 MiB	37.1 sec	accesses=6636, **hits=6635, misses=1, evictions=0**, bypasses=0
4 MiB Page	8 MiB	37.979 sec	accesses=6634, **hits=6632, misses=2, evictions=0**, bypasses=0
8 MiB Page	16 MiB	41.042 sec	accesses=6636, **hits=6635, misses=1, evictions=0**, bypasses=0
4 MiB Page	16 MiB	45.977 sec	accesses=6634, **hits=6632, misses=2, evictions=0**, bypasses=0

Results – Local File System

File Version	Page Buffer Size	Total Runtime	Page Buffer Stats
Original	N/A	43.945 sec	N/A
4 MiB Page	4 MiB	44.934 sec	accesses=6634, hits=5691, misses=943, evictions=942, bypasses=0
8 MiB Page	8 MiB	34.885 sec	accesses=6636, hits=6635, misses=1, evictions=0, bypasses=0
4 MiB Page	8 MiB	33.523 sec	accesses=6634, hits=6632, misses=2, evictions=0, bypasses=0
8 MiB Page	16 MiB	36.089 sec	accesses=6636, hits=6635, misses=1, evictions=0, bypasses=0
4 MiB Page	16 MiB	32.423 sec	accesses=6634, hits=6632, misses=2, evictions=0, bypasses=0

Wrap-Up

- HDF5 library supports combining file metadata and raw data bytes into separate internal file pages of configurable size.
- Only available to new files. Existing files must be repacked.
- Using HDF5 page buffer cache when reading such files from cloud object store can significantly improve performance.

Wrap-Up (cont'd)

- Page Buffer size must be at least the size of total file metadata.
- Like with any other cache: The more, the merrier!
- Page buffer statistics is available for fine tuning.
- This is work in progress. ESIP community is invited to try these HDF5 storage options and contribute to developing best practices.

This work was supported by NASA/GSFC under Raytheon Technologies contract number 80GSFC21CA001.

Thank you!

EOSDIS