EOSDIS
NASA'S EARTH OBSERVING SYSTEM
DATA AND INFORMATION SYSTEM

Hierarchical Data Format (HDF) Status Update

2018 Summer ESIP

Elena Pourmal
EED2 Technical Lead
epourmal@hdfgroup.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

- Update on current HDF releases
  - New features
- Moving to HDF5 1.10 series
  - Controlling HDF5 file versioning
  - Taking advantage of HDF compression
- What is coming in HDF5 1.12?
  - Non-POSIX I/O and new defaults
- Getting help with HDF software and data

Current HDF releases

- HDF5 1.8.21 (June 2018)
  - Vulnerability patches
  - Tools fixes
  - Support for Intel Fortran v 18 compiler on Windows
  - *There will be one more maintenance release of HDF5 1.8 version. It is time to move to HDF5 1.10 series!*
- HDFView 3.0 (June 2018)
- For more info see https://hdfgroup.org

Current HDF releases

- HDF5 1.10.2 (March 2018)
  - Vulnerability patches
  - Enabling control over the HDF5 file versioning
  - Enabling compression for parallel (MPI I/O) writes
- HDF5 1.10.3 is coming later this year
  - Parallel compression enhancements
- See https://hdfgroup.org for details

Moving to HDF5 1.10 series

- Controlling HDF5 file versioning
  - HDF5 library is ALWAYS backward compatible
    - New version of the library will always read files created by the earlier versions
  - HDF5 library is forward compatible
    - By default the library will create objects in a file that can be read by the earlier versions of the library
      - HDF5 file does not have a version
      - Versioning is done on an object level

Moving to HDF5 1.10 series

- Q: How one can assure that HDF5 files created by HDF5 library version 1.10 and later will be read by the applications based on HDF5 1.8 and earlier?
- A: Use H5Pset_libver_bounds(*hid_t* fapl_id, *H5F_libver_t* low, *H5F_libver_t* high) in applications
  - Uses file access property to specify the features that can be created by the library specified by the “high” parameter and latest versions of the objects available in the library specified by the “low” parameter.
  - *H5F_LIBVER_EARLIEST, H5F_LIBVER_V18, H5F_LIBVER_V110 , H5F_LIBVER_LATEST*

Moving to HDF5 1.10 series

- Taking advantage of HDF5 compression
  - Compression works for both sequential and parallel (MPI I/O) writes/reads
  - HDF5 supports GZIP and SZIP compressions
    - Open Source and free SZIP from German Climate Computing Center https://www.dkrz.de/redmine/projects/aec/wiki/Downloads
    - Fully compatible with SZIP provided by The HDF Group (encoder is not free for commercial data usage)
  - Multiple third-party compressions available as plugins; see https://portal.hdfgroup.org/display/support/Contributions
  - One compression doesn’t fit all data!

HDF5 Compression

- Using compression with Sentinel Data
  - HDF5 file that was created by converting Sentinel 1 GeoTiff file.
  - File contains one 32-bit integer array with dimensions 20256x25478; dimensions correspond to the number of image strips stored in the original Sentinel 1 GeoTiff file.

Compression	Compression ratio	File size in bytes
No compression	1	2065283096 (2GB)
SZIP	1.062	1944126897
GZIP	1.966	1049969129
SHUFFLE + GZIP	2.192	941879752 (< 1GB)

HDF5 Compression

- Using compression with SeaSat Data
  - HDF5 file contained 3 datasets
  - Table below shows CR for each dataset when using GZIP, SZIP and combinations of SHUFFLE and GZIP
  - Different compressions (highlighted) can be applied to get compression ratio (CR) of 1.9

Compression	CR HH	CR latitude	CR longitude	Total file size in bytes	TCR
Original file (GZIP)	1.167	2.693	2.747	407848072	1
SZIP	1.337	3.789	4.423	317040127	1.29
SHUFFLE + GZIP	1.329	20.049	24.003	216176244	1.89

HDF5 Compression

- Using compression with SeaSat Data
  - Compression and decompression will differ depending on the method
  - Table below shows elapsed times for the h5repack to encode data and h5dump to display data for SeaSat file.

Compression	TCR	Time to compress using h5repack	Time to decompress with h5dump
SZIP	1.29	0:11.34	6:21.98
BLOSC	1.59	0:13.87	6:15.92
SHUFFLE + GZIP	1.89	0:20.91	6:31.29

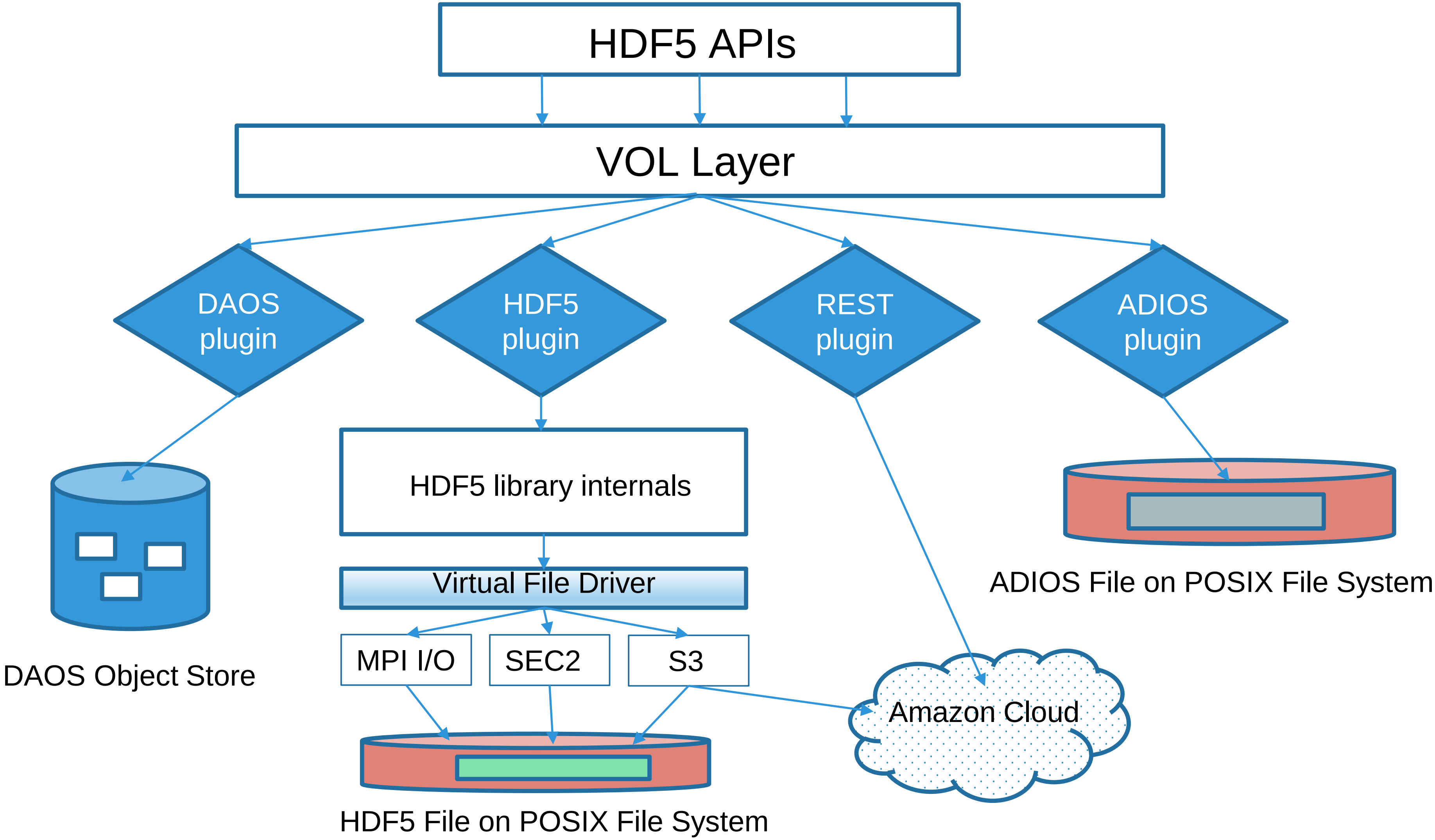
What is coming in HDF5 1.12?

- **New defaults and file format changes**
  - UTF-8 encoding for strings (vs. current ASCII encoding)
  - Setting “low” to *H5F_LIBVER_V18* *(vs. H5F_LIBVER_EARLIEST* in H5Pset_libver_bounds(*hid_t* fapl_id, *H5F_libver_t* low, *H5F_libver_t* high)
    - *Better performance for groups and attributes traversals*
    - *No limitation on the attribute sizes*
  - File format extensions to address misc. file format issues (e.g., 64-bit dataspaces encoding)

What is coming in HDF5 1.12?

- Virtual Object Layer to perform I/O to any storage including Object Storage
  - Plugin architecture for VOL plugins
    - REST VOL plugin
    - VOL plugins in progress:
      - **RADOS: R**eliable **A**utonomic **D**istributed **O**bject **S**tore is part of CEPH distributed storage system.
      - **DAOS: D**istributed **A**synchronous **O**bject **S**torage (DAOS) is an open-source software-defined object store.

Virtual Object Layer



Questions?

EOSDIS

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®