HDF5 and Ecosystem: what is new?

ESIP
HDF Town Hall
June 22, 2020
Outline

• HDF update
  • Releases and new features
  • Community engagement
    • Github move
    • Documentation work
    • HDF5 User’s Group meeting
    • The HDF Group Webinars

• HDF5 ecosystem update
  • HDFql
  • H5CPP
HDF update

Releases and new features
HDF4

• **HDF 4.2.15** released in February 2020
  • Added support for
    • macOS 10.13 and 10.14
    • VS2019
    • MinGW (for C library only)
  • Code cleanup: fixed memory leaks and refactored code.

• HDF4 is deprecated in favor of HDF5
  • Use **h4toh5tools** software to convert HDF4 files to HDF5

• The HDF Group will provide HDF4 releases once a year to keep up with the new systems and compilers
  • Known problems: Fortran library build fails with gcc 10

• Migrate to HDF5 now!
HDF5

• Current maintenance releases (1.8.*, 1.10.*, 1.12.*)
  • In all releases we addressed vulnerability issues reported by HDF5 Community
  • HDF5 1.8.21 June 2018
    • Updated C++ APIs; bug fixes in tools
    • The HDF Group will release HDF5 1.8.22 later in the year
    • Consider migrating to HDF5 1.10.6 or HDF5 1.12.0 now unless you see performance degradation for your application
    • Support for 1.8.* releases will be dropped as soon as we address performance issues in 1.10 and 1.12 releases.
  • HDF5 1.10.6 December 2019
    • Added S3 and HDFS read-only Virtual File Drivers (VFDs) (more info on the next slides)
    • Improved performance when creating a large number of objects
    • Addressed several backward compatibility issues when creating files to be read by HDF5 1.8.* versions
  • HDF5 1.10.7 August 2020
    • Performance improvements for handling selections and large number of objects
HDF5 (cont’d)

• Current maintenance releases
  • HDF5 1.12.0 in February 2020 – major release
    • Virtual Object Layer (VOL)
    • External references
  • Performance improvements when working with selections
  • Extensions to file format to support new features
  • Files created by HDF5 1.12.* releases may not be compatible with HDF5 1.10.* and earlier
    • H5repack tool can be used to “downgrade” the version of file format to 1.10 or to 1.8
• Why to move to 1.12.*?
  • Multiple ways to access data in Cloud and in Object Store using VOL and VFD capabilities
    • Minimum or no modifications are required to applications
  • External references along with existing Virtual Datasets provide a flexible way to aggregate data stored in HDF5 files
  • Performance improvements for sequential and parallel HDF5
HDF5 Library Architecture (1.12.0)

HDF5 API and language bindings

Virtual Object Layer (VOL)

Pass-through VOL connectors

Native Connector

Native Connectors

POSIX
MPI I/O
SWMR
S3
HDFS
REST
DAOS
Data Elevator
Hermes

External connectors

HDF5 applications can talk to Cloud using S3 VFD or REST VOL and HSDS server
HDF5 Cloud Solutions

S3 VFD

- Part of the 1.10.6 and 1.12.0 releases
- Read-only
- Usage:
  ```c
  H5FD_ros3_fapl_t ros3_fa = { 1, 0, "", "", "" };  
fapl_id = H5Pcreate(H5P_FILE_ACCESS);  
status = H5Pset_fapl_ros3(fapl_id, &ros3_fa);  
file = H5Fopen (FILE, H5F_ACC_RDONLY, fapl_id);
  ```

- Future work:
  - Improve scalability
  - Make VFD dynamically loaded and file in S3 without application modification
  - Provide write access to S3

REST VOL

- External plugin for HDF5 1.1.2.0
  
  https://github.com/HDFGroup/vol-rest/blob/hdf5_1_12_update/docs/users_guide.pdf

- Supports both reads and writes
- Usage:
  ```c
  RVinit();  
fapl = H5Pcreate(H5P_FILE_ACCESS);  
H5Pset_fapl_rest_vol(fapl);  
file = H5Fopen(filename, H5F_ACC_RDWR, fapl);  
or set environment variable to point to connector; no modification to application is needed
HDF5_PLUGIN_PATH=/<path>/lib  
HDF5_VOL_CONNECTOR=REST
  ```

- Future work:
  - Provide direct access to S3
  - Improve scalability
HDF5 dataset may now store references to objects or to dataset regions that are in another HDF5 file.

Dataset in B.h5 stores references to datasets stored in A.h5
HDF5 Reference to Attribute

HDF5 1.12.0 allows to reference an attribute of a dataset or a group.

Dataset in A.h5 stores references to the attributes on the objects in A.h5 and in B.h5.
Splitter and Mirror VFDs

- Coming in HDF5 1.12.1
HDF5 File Provenance

- HDF5 offers no support for version control and provenance management
- Data provenance is a major concern in cases where
  - Original data must be preserved
  - Changes are tracked and attributed
- General HDF5 “undo” capability is technically challenging but tracking changes per file open/close cycle may be sufficient
  - Changes are stored in the file itself
- Solution: new VFD (called provenance or “onion’)
  - Coming in HDF5 1.12.2
HDF update

Community engagement
HDF5 is moving to Github

- We plan to announce the move after HDF5 1.10.7 release
HDF5 Documentation

• The HDF Group is working on addressing deficiencies of current HDF documentation (e.g., slow, hard to search, multiple locations)
• Go to https://hdf5.wiki/ and help us with requirements gathering!
Want to learn more about HDF5?

• HDF5 Webinars
  - https://www.hdfgroup.org/category/webinar/
  - Announced on HDF-FORUM and Website

• HDF5 User’s Group Meeting
  - October 2020 via ZOOM
  - Interested to present your work? Please contact epourmal@hdfgroup.org

• HDF-FORUM
  - https://forum.hdfgroup.org/

• HDF Helpdesk
  - help@hdfgroup.org
  - Service Desk (requires registration via https://hdfgroup.org )
HDF5 ecosystem update

HDFql and H5CPP – news kids on the block
HDF query language (HDFql)

- HDFql ([https://www.hdfql.com/](https://www.hdfql.com/)) is a high-level language to manage HDF5 data
  - Designed to be as simple and powerful as SQL
    - Reads and writes HDF5 data; supports parallel I/O
    - Has data and metadata querying capabilities
  - Works on Linux, Windows and macOS and supports C, C++, Java, Python, C#, Fortran, and R
  - Easy to learn; portable

```c
hid_t file, group, dataspace, property, dataset;
hsizes_t dimension;
int value[3];
file = H5Fcreate("my_file.h5", H5F_ACC_EXCL, H5P_DEFAULT, H5P_DEFAULT);
group = H5Gcreate(file, "my_group", H5P_DEFAULT, H5P_DEFAULT, H5P_DEFAULT);
dimension = 3;
dataspace = H5Screate_simple(1, &dimension, NULL);
property = H5Pcreate(H5P_DATASET_CREATE);
H5Pset_chunk(property, 1, &dimension);
H5Pset_deflate(property, 9);
dataset = H5Dcreate(group, "my_dataset", H5T_NATIVE_INT, dataspace, H5P_DEFAULT, property, H5P_DEFAULT);
value[0] = 4;
value[1] = 8;
value[2] = 6;
H5Dwrite(dataset, H5T_NATIVE_INT, H5S_ALL, H5S_ALL, H5P_DEFAULT, &value);
```

 Courtesy Rick Noyer rick@hdfql.com
H5CPP

• Website: http://h5cpp.org/
• GitHub: https://github.com/steven-varga/h5cpp
• Download: http://h5cpp.org/download/

• Modern C++17 interface to HDF5
  • Designed on principles of simplicity – uses four basic CRUD-like templates
  • Supports popular Linear Algebra systems
  • Has zero overhead over C APIs
  • No knowledge of HDF5 is required
  • Developed by Steven Varga, independent developer and researcher
h5py

```python
temperature = np.random.random(1024)
wind = np.random.random(2048)
f = h5py.File('weather.hdf5')
f['/15/temperature'] = temperature
f['/15/temperature'].attrs['dt'] = 10.0
f['/15/wind'] = wind
```

H5CPP

```cpp
fvec temperature = arma::randu<fvec>(1024);
fvec wind = arma::randu<fvec>(2048);
auto fd = h5::create("weather.hdf5", H5F_ACC_TRUNC);
auto ds = h5::write(fd, "/15/temperature", temperature);
ds["dt"] = 10.0f;
h5::write(fd, "/15/wind", wind);
```
THANK YOU!

Questions & Comments?