RDA Working Group Proposal: HDF5 External Filter Plugin Working Group

December 13, 2014

1 Charter

One of the key features of HDF5 is the ability to apply compression (or "a compression filter" in the HDF5 terminology) to individual data objects such as datasets and groups stored in an HDF5 file. Four general compression algorithms are included in the HDF5 core library. With increasing amount of data recorded during experiments or generated by computer simulations, compression algorithms become increasingly important in order to reduce the required storage space and to reduce a bandwidth when transferring files. As there is no universal compression algorithm that satisfies the demands of every application, HDF5 provides a capability to add a custom compression filter to address the needs of a specific application.

However, this approach comes with a serious drawback. In order for an application to read a file containing data compressed with a custom filter, the source code of this application must be changed to register the custom filter with the HDF5 library and the application has to be recompiled. This approach becomes unacceptable for applications that are distributed as binary only, especially for the commercial applications such as MATLAB and IDL.

As a solution to this problem The HDF Group (THG) implemented a new external filter interface for the HDF5 library that allows adding and removing filters at runtime without recompiling the code. The filters are installed as shared modules which are loaded by the library on demand. This new approach not only allows commercial applications to access data compressed with a custom algorithm, it also makes life easier for the open source developers who do not have to recompile their code whenever a new filter algorithm becomes available.

One example of where the new external filter interface comes in handy is the Eiger X-ray detector produced by DECTRIS using LZ4 compression. Another one would be the two HDF5 Python bindings `h5py` and `PyTables` that provide custom compression algorithms. However, with the old approach data written by the Eiger detector or by one of the two Python bindings using their own compression algorithms would never be accessible by applications like Matlab or IDL. Using the new external filter interface and after installing the appropriate filter modules, commercial applications can easily access the data compressed with a custom filter just as if it would have been written with one of the internal filters.

The proposed working group will

- Establish standards for how filter code should be organized, tested, documented and distributed.

- Provide the infrastructure for developing and distributing filters code, modules and documentation.
- Provide a set of standard files to be used as benchmarks for every filter so the users can verify their own filter module installations.
- Be an entry point for users who encounter custom filters in files and require the appropriate filter module to access it.

The working group will raise awareness among the developers of the requirement for the long-term accessibility to data compressed with custom filters in HDF5.

2 Value proposition

The external filter interface implemented in recent releases of HDF5 has greatly increased the capabilities to efficiently store data streams with extreme data rates or data volumes in the HDF5 files. However, further development of the custom filters by the Open Source community and assurance of the long-term accessibility to data compressed with custom filters requires technical and organizational framework. The framework should provide a standard for the source code development and distribution by creating the design and development guidelines and recommending the best software development practices. This working group will establish the technological basis and guidelines to render the external filter interface into a sustainable, well documented facility. We expect that it will greatly facilitate the use and deployment of the new compression filters for high-throughput storage of all kind of binary data in HDF5 to the benefit of both academic and industrial research and development.

3 Working plan

It is currently suggested that the source code for the external filters modules will go into a GitHub repository. The working group will

1. Define the standards for the external filter module source code, in particular, for
   - getting source code into the repository
   - the source code structure the module has to obey
   - testing and documentation standards
   - test files
2. Determine the target platforms and architectures for which external filters should be available.
3. Develop a process for external filters registration with the repository (what are the prerequisites).
4. Establish mailing lists for users and developers.
5. Develop a formal process for filter maintenance (for instance, whether or not all filters shall be tested with every new release of the HDF5 library).

6. Set up a process for dealing with abandoned modules in the repository.

The working group will identify other tasks as work progresses.

4 Adoption plan

Currently THG runs a website which lists all registered external filters and basic information about them. THG will stay in charge of the external filters registration. The working group will collaborate with THG on making the registration process and publication of information "developer" friendly.

Possible process for developing and publishing an extrenal filter could look like this:

1. Propose a new filter and obtain a filter identifier from THG to start with the development.
2. Access will be granted to the repository where the developer can deposit the code.
3. Once the filter passes all required tests, the code must be reviewed.
4. If the code passes the review process or after the changes requested by the reviewer have been made the filter can be published in the repository.

This is a draft of a possible publishing process. The details and all the subtle problems within it will be resolved as the working group proceeds.

5 Initial Membership

It is expected that a developer (or an organization), who has registered an external filter with THG, will become a member of this working group. THG will also be a member of this group.