Developing HDF5 for the Synchrotron Community

Nick Rees (Diamond Light Source), Heiner Billich (Paul Scherrer Institute), Andy Götz (ESRF), Quincy Koziol & Elena Pourmal (The HDF Group), Michael Rissi (Dectris), Eugen Wintersberger (DESY)

In May 2012 the Paul Scherrer Institute and Dectris held a workshop for some European synchrotron developers and The HDF Group which identified a number of issues with HDF5, including:

- The single threaded nature of the HDF5 library made it difficult to benefit from the parallel architecture of modern systems.
- Synchrotron detectors used compression to improve data throughput, but this limited the throughput of the library and could not be used with Parallel HDF5.
- A sequence of detector frames could be stored in one HDF5 file but this increased the processing latency since files couldn't be open for read and write simultaneously.

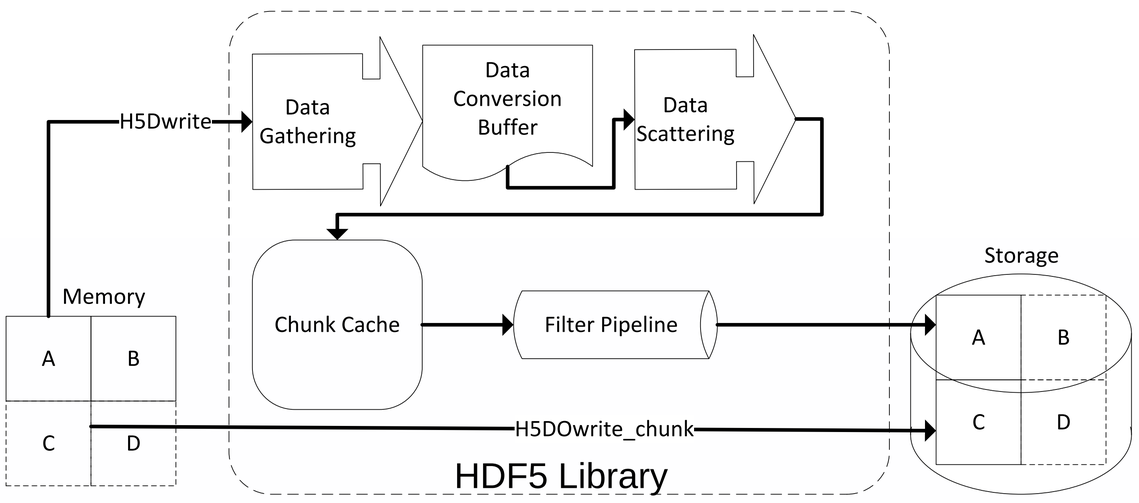
This poster describes what we have done to fix these problems, and how we have worked together to fund development that may have too expensive for a single organisation.

Direct Chunk Writes (Sponsored by Dectris and PSI)

This introduced the H5DOwrite_chunk function in HDF5 which bypasses the internal data flow of chunk management and writes a data chunk directly to disk . This allows a detector driver, for example, to compress data outside the library (maybe using parallel algorithms or hardware accelerators). It also avoids a number of data copies, which limits any dataflow through the HDF5 filter pipeline to ~500 MB/sec on typical processors.

The feature was released in HDF5 version 1.8.11 in May 2013. The implementation and testing required about 5 months of work.

The picture below shows how the new H5DOwrite_chunk routine bypasses much of the processing associated with normal HDF5 chunk management.



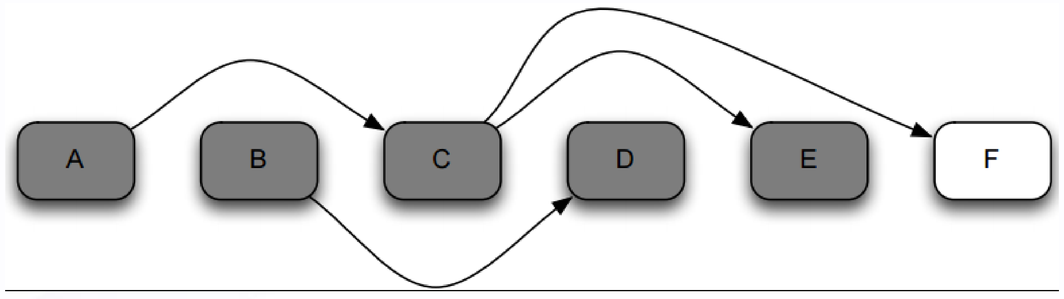
Single Writer/Multiple Reader (Sponsored by Diamond, ESRF and Dectris)

Single Writer/Multiple Reader (SWMR) allows a single process to update datasets in an HDF5 file whilst other processes are reading data from the same datasets. In old (pre 1.10) versions of HDF5 the file was only guaranteed to be consistent if it wasn't open for write. SWMR ensures that the file is always in a consistent state and has the secondary benefit that the file isn't corrupted by an application crash. The consistency is ensured by managing the order in which data and metadata are written to disk (see diagram below).

This algorithm is lock-free and communications are done through the HDF5 file, but it assumes that the underlying file system has POSIX I/O semantics, so that all readers see writes by another process as atomic operations and in the same order that the writer generates them. Some file-systems (notably NFSv3) do not abide by these semantics and cannot be used.

The feature presented significant technical difficulties and required a lot of rework of the HDF5 library architecture. It also triggered changes to the HDF5 File Format introducing new chunk indexing structures with enabled checksums on the metadata items.

The development comprised about twelve months of effort and was delivered as a specialized version of the HDF5 library to DLS, ESRF and Dectris. The HDF Group has been working on merging the feature with the main stream of HDF5 and delivering it in HDF5 release version 1.10.0 at the end of Q1 2016.



An example of how SWMR manages file updates to ensure consistency. A-F are metadata in the file that refer to each other - A is the parent of C, B is the parent of D, and C is the parent of both E and F. Dirty entries are shaded (A-E), and clean entries are not shaded (F). A will not be written until C is clean, C will not be written until both E and F are clean and B will not be written to the file until D is clean.

Dynamically Loaded Filters (Sponsored by DESY)

This allows the HDF5 pipeline filters to be dynamically loaded at run-time. Application domains can develop specialist filters without the need for integrating them within HDF5 or passing support to The HDF Group. We have been developing compression algorithms that are significantly better than the standard HDF5 ones for synchrotron use cases.

Writing, testing and distributing custom HDF5 filters is now a community effort with The HDF Group providing guidance. The HDF Group, DESY and Dectris are working on a repository of the dynamically loaded filters so users can download code and binaries for filters they need (https://svn.hdfgroup.uiuc.edu/hdf5_plugins/), New filters should be registered with The HDF Group: (https://www.hdfgroup.org/services/contributions.html).

The feature was released in HDF5 version 1.8.11 in May 2013

Virtual Data Sets (Sponsored by DESY, Percival detector project and Diamond)

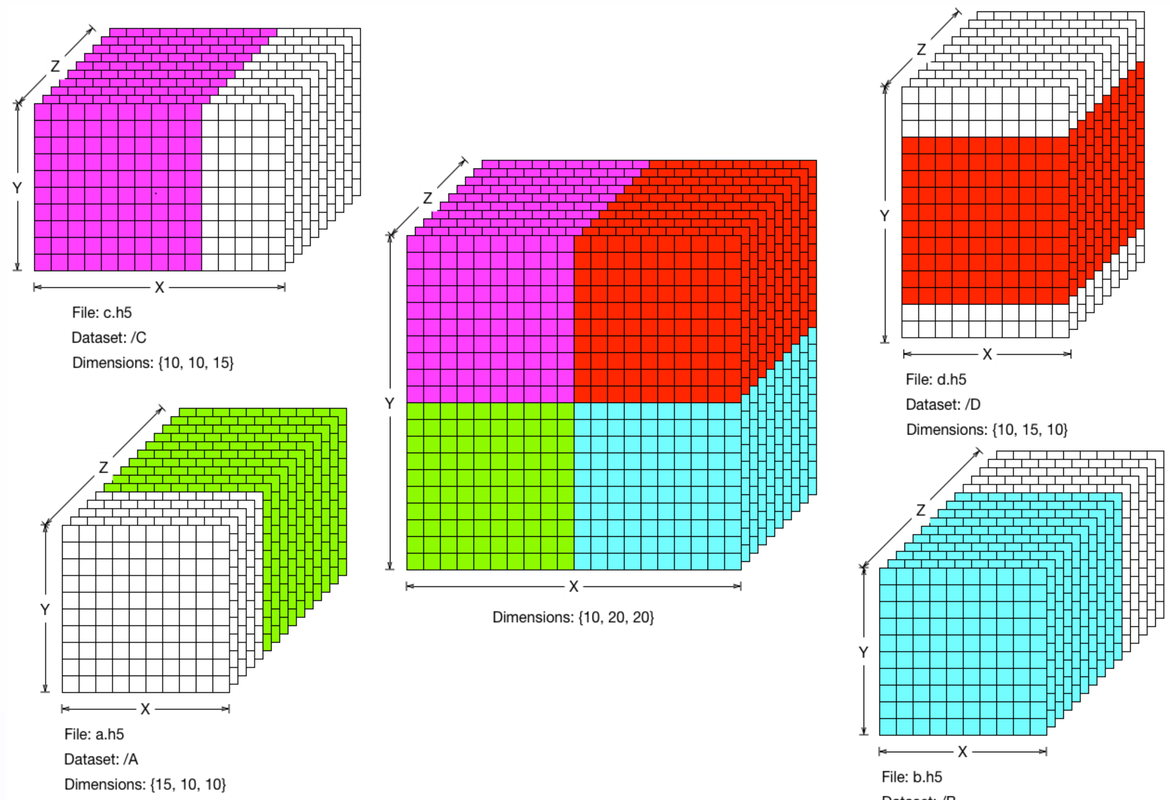
This idea evolved since the PSI workshop. The initial idea was to provide compression in pHDF5 but this would have been complex to implement and was hampered because pHDF5 is not as fast per process as single stream HDF5.

The next idea was to extend the HDF5 soft-link so it could reference a union of datasets from a number of other files (not just one). These files could then be written independently and in parallel and because the writers wouldn't need pHDF5 the datasets could also be compressed. The final design was a generalisation of this and now the parent dataset can be composed of an arbitrary mapping of child datasets.

Each mapping can be any n-dimensional rectangle and gaps can be filled with user specified fill values. Virtual datasets can be used for sparse data, or for combining data from different sources, or for applications that weren't considered when the original files were written. They allow multiple data sources to update an HDF5 virtual dataset simultaneously (provided the writers all write to different child datasets) and combined with SWMR to provide a multiple writer/multiple reader system.

The feature implementation required adding a new storage layout to HDF5 and triggered improvements to the HDF5 library code. The total implementation effort took seven months of the developers' time. Another two months will be spent on merging VDS with the HDF5 mainstream library and preparing the feature for the HDF5 1.10.0 release at the end of Q1 2016.

The picture below shows a simple mapping of four source datasets (around the outside) to a single virtual dataset (centre), but much more complex mappings are possible.



The goal of this poster is to show how we have collaborated to develop common software for the benefit of the entire community. These have been different to many of our other collaborative software developments (e.g. EPICS and TANGO) in that all the development was done commercially rather than in-house. This has created challenges because money rather than manpower had to be committed and this has required a strong motivation and champion for the development. However, it has also proved to be productive because the developments have been delivered largely to time and to the agreed budget.

For more information please visit http://confluence.diamond.ac.uk/display/HDF5DEV/HDF5+Developments or contact Nick Rees at Nick.Rees@diamond.ac.uk