Evidence of Implementation for the HDF5 1.6 file format and I/O Library

Elena Pourmal, Mike Folk*
2006/01/23
Revision 0.0

This document describes available implementations of the HDF5 File Format and I/O Library.

HDF5 was created to address issues of managing and accessing complex and large (order of terabytes and petabytes) datasets and data collections. Due to the complexity of the HDF5 File Format and requirements for the I/O library, there are currently only two implementations available. The implementations listed below are the implementations of the HDF5 File Format and/or I/O Library, as opposed to the successful Open Source and commercial software products built on top of HDF5 (see http://hdf.ncsa.uiuc.edu/tools5.html)

The following list of implementations is complete as of January 21, 2006, to the best of our knowledge.

1. *Reference implementation from NCSA, University of Illinois at Urbana-Champaign*
   The reference implementation of the HDF5 File Format and I/O Library (http://hdf.ncsa.uiuc.edu/HDF5/) consists of approximately 2073 files or about 917,000 lines of the source code. It is available in C language with Fortran 90, C++ and Java wrappers. It took about 15 person years to implement the file format and library, which satisfy the following criteria:
   - *Portability*
     HDF5 files can be read/written on almost all currently available platforms and operating systems. For the complete list see http://hdf.ncsa.uiuc.edu/HDF5/release/platforms5.html
   - *Efficient storage and access*
     HDF5 supports compression and chunking of raw data, which minimize storage and provide efficient access to complex subsets of the raw data; HDF5 metadata caching mechanism supports adept access to an unlimited number of objects stored in an HDF5 file.
   - *Flexibility and extendibility*
     The reference implementation of the HDF5 file format and I/O library is designed to adapt to the new storage and performance requirements.
   - *Scalability*
     Data stored in HDF5 files can be accessed and modified in parallel using the MPI I/O paradigm; this access is scalable (i.e. total I/O speed is proportional to the number of processes HDF5 application uses to access the data)

* NCSA University of Illinois epourmal@ncsa.uiuc.edu, mfolk@ncsa.uiuc.edu
The HDF Group epourmal@hdfgroup.otg, mfolk@hdfgroup.org

- *Well documented*
  Complete documentation is available from the NCSA HDF5 Web page (http://hdf.ncsa.uiuc.edu/HDF5/doc/)

2. *UniData Java implementation*
   According to http://www.unidata.ucar.edu/software/netcdf-java/ version 2.2 of the NetCDF-Java library is an implementation of a Common Data Model (CDM), a generalization of the NetCDF, OpenDAP and HDF5 data models. It is also a prototype for the NetCDF-4 project (http://www.unidata.ucar.edu/software/netcdf/netcdf-4/index.html), which provides the "data access layer" of the CDM, on top of the HDF5 File Format. The netCDF-Java library can only read HDF5 files. Currently this HDF5 reader is in "alpha" release. NetCDF-HDF5 Java APIs will be made public at the time of the NetCDF-4 "beta" release when NetCDF file format is finalized. For more information on the NetCDF-HDF5 Java reader, contact John Caron (caron@unidata.ucar.edu).

* NCSA University of Illinois epourmal@ncsa.uiuc.edu, mfolk@ncsa.uiuc.edu
The HDF Group epourmal@hdfgroup.otg, mfolk@hdfgroup.org