P2.42

HDF5 – A HIGH PERFORMANCE DATA FORMAT FOR EARTH SCIENCE

MuQun Yang
Robert E. McGrath
Mike Folk
National Center for Supercomputing Applications
University of Illinois, Urbana-Champaign

I. Introduction

HDF[2] is a set of data formats and software libraries for storing scientific data with an emphasis on standards, storage and I/O efficiency. HDF software is open-source and free. HDF4 is based on the original 1988 version of HDF and is backwardly compatible with all earlier versions. HDF4 file size cannot be greater than 2GB and the number of objects inside an HDF4 file must be less than 20,000. The data model of HDF4 is rigid and HDF4 does not support parallel I/O. HDF5 is a new data format and it was first released in 1998. HDF5 limits neither the size of files nor the size or number of objects in a file. HDF5 is based on a more general data model and emphasizes standards and flexible, efficient IO.

In this paper, we will introduce HDF5 as a data format and software. Five sections are included:

- HDF5 data model
- Important HDF5 features
- HDF5 tools
- Other HDF5 products
- HDF5 users

II. HDF5 data model

1. Datasets and groups

The two primary data objects in HDF5 are datasets and groups. A dataset includes a multidimensional array of elements together with additional information describing the dataset. A group is a mechanism for creating and maintaining collections of related objects. Every file starts with a root group.

Besides the array, the components of a dataset include a datatype, a dataspace, a user-defined attribute list, and information regarding special storage options. An attribute list can also be associated with a group.

2. Datatypes and dataspaces

A datatype is a classification specifying the interpretation of a single data element. HDF5 datatypes can be either atomic types or compound types.

HDF5 atomic types can be any of the following:

- standard integer or float
- user-definable scalars (e.g. 13-bit integer)
- variable-length data (e.g. strings)
- pointers: references to objects or dataset regions
- enumerations: names mapped to integers

HDF5 compound types are comparable to C structs. Members of a compound type can be atomic or compound types and members can be multi-dimensional.

An HDF5 datatype can also be stored in the HDF5 file as an independent, named object, called a named datatype.

An HDF5 dataspace contains information *about* a dataset. A dataspace can be defined by the rank and dimension of the array. For the purpose of subsetting datasets, a dataspace can be defined in terms of hyperslabs or as distinct points [5].

3. Attributes

An HDF5 attribute is a small piece of data describing the nature and intended usage of a dataset or group. An attribute has two parts: name and value.

4. Illustrations of an HDF5 file

Figure 1 shows an example of an HDF5 file. Figure 2 shows an example of HDF5 file structure illustrating the main concepts of the HDF5 data model.

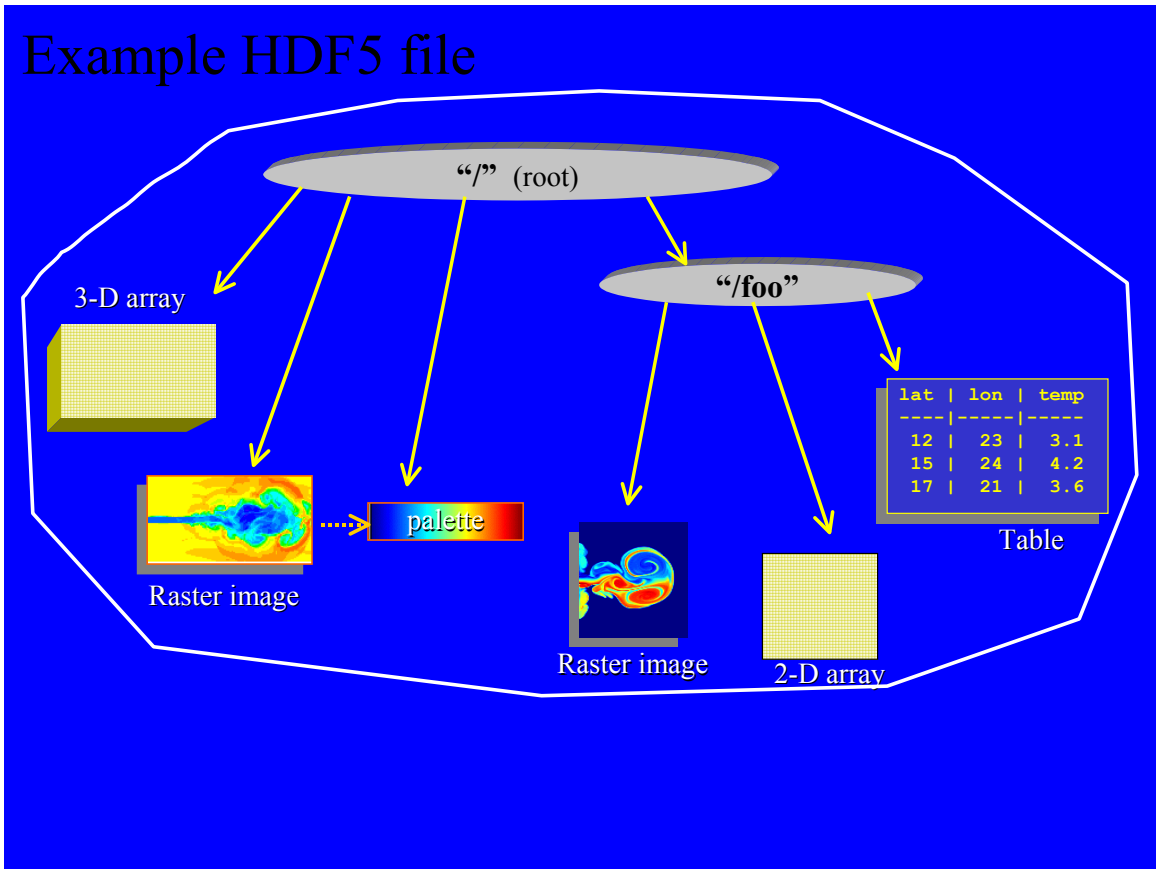

Figure 1: An example HDF5 file (adapted from [2]).

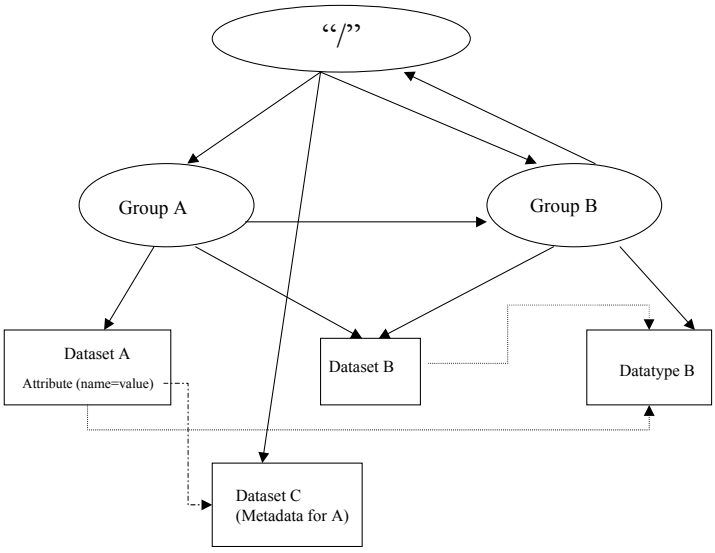

Figure2: Sample HDF5 file structure illustrating the main concepts of the HDF5 data model. The file contains two groups, A and B. Group B is a member of group A and both groups are members of the root group, /, which serves as the point of entry to the file structure graph. The root group itself is a member of group B. Dataset B is a member of groups A and B. Datasets A and B share the same datatype B that is stored in the file as a member of group B. The attribute of dataset A is an object reference and points to another dataset, C, which may be used as meta data (adapted from [3]).

III. Important features of HDF5

1. Specialized data storage

HDF5 provides several specialized data storage options to improve IO performance, storage efficiency and data management.

Figure 3 illustrates several options, including chunking, compression, extendable arrays, and split files. More information can be found at [1][5].

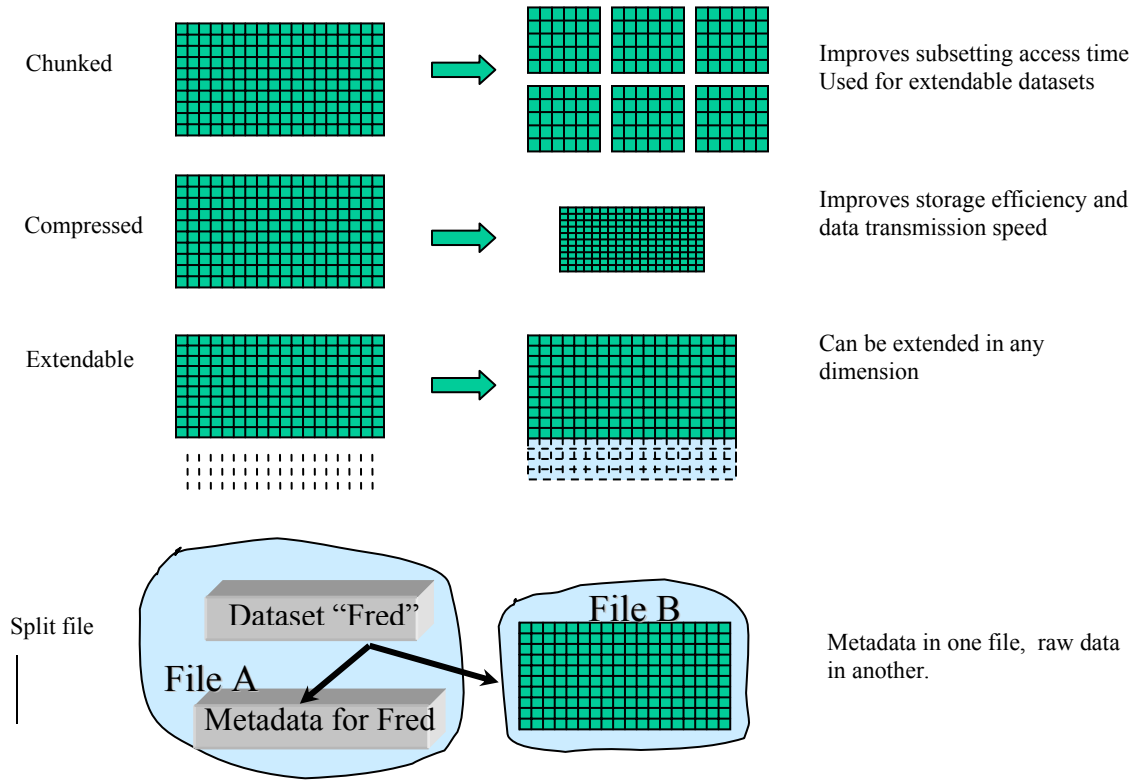

Figure 3: Specialized storage options in HDF5 (adapted from [2])

2. Virtual file driver

HDF5 has a virtual file layer to provide different types of storage, such as a single file, multiple files, local memory, a network protocol, and files on a parallel file system. Furthermore, users can add their own storage type easily through a virtual file driver interface. The network file driver, streamed IO, is an example of such a user-created driver that is included in the standard HDF5 distribution [5]. Figure 4 illustrates how the HDF5 virtual file layer is used.

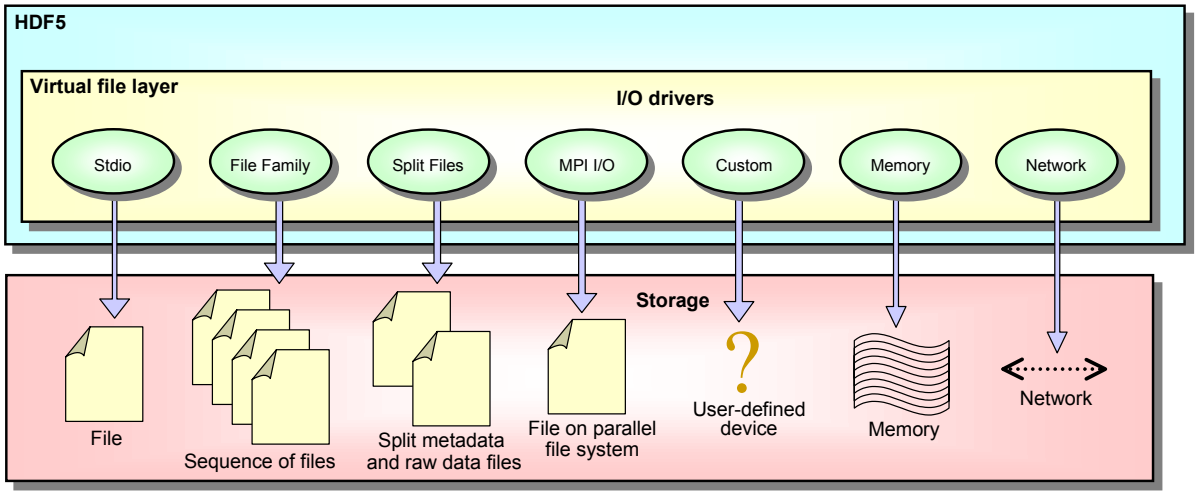

Figure 4: The HDF5 virtual file layer (VFL). Through the use of the VFL, an HDF5 file can be stored as a conventional UNIX file or as multiple files (to overcome a 32-bit system limit on file size); it can be stored as two or more files containing separated meta data and raw data; it can be stored as multiple files on a parallel file system; it can be saved in memory or sent over the network; or it can be handled by another non-standard, user-supplied driver (adapted from [3]).

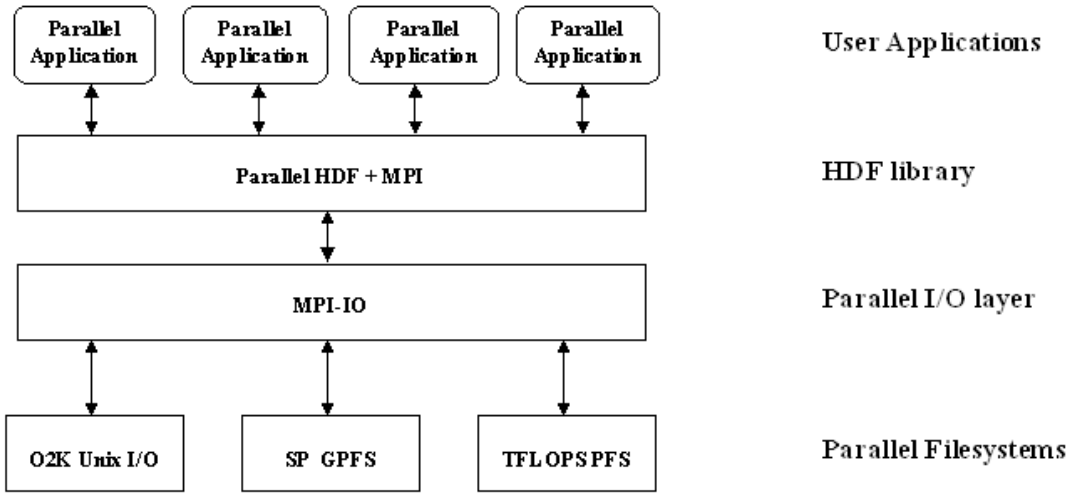

Figure 5: The software stack of parallel HDF5 (adapted from [4]).

3. Parallel HDF5

Another important feature is that HDF5 supports parallel IO through MPI-IO. This allows a parallel application to take fuller advantage of the power of parallel computational environments. Figure 5 shows the software stack of parallel applications using HDF5.

4. External storage properties

The external storage format allows data to be stored across a set of non-HDF5 files. A set of segments (offsets and sizes) in one or more files is defined as an external file list. Only the offset and size of the external file need to be specified.

5. Portability

HDF5 supports C, C++, and Fortran APIs. HDF5 can run on virtually any scientific computing system, including massively parallel systems. An incomplete list of platforms supported is as follows:

- AIX (IBM SP)
- Cray J90, T3E
- FreeBSD
- HP-UX
- IRIX 6.5, IRIX64
- Solaris
- ASCI TFLOPS
- Windows XP, NT 5.0
- Mac OS X

IV. HDF5 tools [6]

The NCSA HDF group also provides several HDF5 tools to help users, developers, and applications better use HDF5.

1. h5dump

h5dump enables the user to examine the contents of an HDF5 file and dump those contents, in human readable form, to an ASCII file. This is one of the most frequently used HDF5 tools.

2. HDFView [7]

HDFView is a Java-based tool for browsing and editing HDF4 and HDF5 files. HDFView allows a user to descend through the file hierarchy and navigate among the file's data objects. The content of a data object is loaded only when the object is selected, providing interactive and efficient access. HDFView editing features allow a user to create and delete HDF objects and attributes and to modify their values.

3. h5repack

h5repack is a command line tool that performs a logical copy of all the HDF5 objects in an input HDF5 file to an output HDF5 file, optionally rewriting the HDF5 datasets with compression and chunking.

4. h5diff

h5diff is a command line tool that compares two HDF5 files and reports the differences between them.

5. h5import

h5import converts data from one or more ASCII or binary files into the same number of HDF5 files.

6. h52gif

h5gif converts an HDF5 file to a GIF file.

7. gif2h5

gif2h5 converts a GIF file to an HDF5 file.

V. Other HDF products

1. HDF5 High-level APIs[8]

The HDF5 High Level APIs consist of a set of functions built on top of the basic HDF5 library. The purpose of this software is to simplify the steps needed to create objects in HDF5. It includes three sets of APIs:

- The Lite API provides general high level functions.
- The Image API provides an easy-to-use interface for managing images and a means of creating and managing standardized image data.
- The Table API provides an easy-to-use interface for managing tabular data and a means of creating and managing standardized tabular data.

2. HDF4 and HDF5 conversion library and utilities [9]

NCSA HDF group also provides a conversion library and a conversion utility to help applications transit from HDF4 to HDF5. The

H4toH5 Conversion Library and the command line conversion utility h4toh5 are based on a document mapping HDF4 file structures and data objects to those of HDF5 [10].

Although the NCSA HDF group encourages the use of HDF5, there is also a utility, h5toh4, that can convert some HDF5 files to HDF4.

VI. HDF users and user support [1]

The list of HDF5 users has grown consistently since its initial release. The HDF5 Library is now used in industrial, academic, and governmental application communities. Two well-known application communities are the NASA Earth Science Data and Information System and the ASCI Data Models and Formats (DMF) Group.

A more complete list of users can be found at http://hdf.ncsa.uiuc.edu/users.html.

The HDF group also maintains a user support Help Desk, which can be contacted by email at hdfhelp@ncsa.uiuc.edu.

References

1. HDF home page: http://hdf.ncsa.uiuc.edu/

2. Mike Folk etc.: HDF5 Overview (slides), http://hdf.ncsa.uiuc.edu/HDF5/papers/presentations/HDF5_overview/index.htm

3. NCSA HDF Group: HDF5 Nomination for the R&D 100 Award 2002, http://hdf.ncsa.uiuc.edu/HDF5/RD100-2002/All_About_HDF5.pdf

4. Albert Cheng: Overview of Parallel HDF5 Design, http://hdf.ncsa.uiuc.edu/HDF5/doc/Tutor/poverview.html

5. HDF5 User's Guide: http://hdf.ncsa.uiuc.edu/HDF5/doc/UG/

6. HDF5 Tools: http://hdf.ncsa.uiuc.edu/hdf5tools.html

7. Peter Cao, Robert E. McGrath: NCSA HDFView http://hdf.ncsa.uiuc.edu/hdf-java-html/hdfview/index.html

8. HDF5 High-level APIs: http://hdf.ncsa.uiuc.edu/HDF5/hdf5_hl/

9. HDF4 to HDF5 conversion webpage: http://hdf.ncsa.uiuc.edu/h4toh5/

10. Mike Folk, Robert E. McGrath, MuQun Yang: Mapping HDF4 Objects to HDF5 Objects, http://hdf.ncsa.uiuc.edu/HDF5/doc/ADGuide/H4toH5Mapping.pdf

Acknowledgements

The authors would like to specially thank Frank Baker at the NCSA HDF group for his great help in editing the abstract.

We would also like to thank all the agencies and companies that have provided support for HDF5 development. Please see http://hdf.ncsa.uiuc.edu/HDF5/acknowledge5.html for our full acknowledgements.