Storing and Processing Complex Data Sets with HDF5

William D. Fraser, Toronto, Ontario, Canada

Downloaded from https://prime-pdf-watermark.prime-prod.pubfactory.com/ at 2025-05-13 via free access

This column is coordinated and edited by William D. Fraser, M.Sc. These articles are not peer-reviewed. The AsMA Science and Technology Committee provides the Watch as a forum to introduce and discuss a variety of topics involving all aspects of civil and military aerospace medicine. Please send your submissions and comments via email to: fraserwdf@gmail.com. Watch columns are available at www.asma.org through the "Read the Journal" link.

Imagine running a complex research project with human and animal subjects evaluated in a variety of experimental and operational settings, involving the collection of a wide variety of anthropometric, biomechanical, medical, physiological, cognitive, and genomic data. The study might extend over years or decades, such as large scale epidemiological studies or long-duration space missions, involve multiple investigators from multiple laboratories and countries, and the investigators that initiate the study may not be the same individuals involved in the analysis and generation of the final reports. Add the further complexity of evolving hardware for data collection, changes in computer architectures for storing the data, and a wide variety of open-source and proprietary software for data manipulation and analysis. The experimental data sets could include extensive quantities of audio, image, and video data, as well as engineering data, such as aircraft parameters, and physical and environmental data such as solar and gamma radiation levels, or air quality. Vast amounts of secondary and tertiary data may be generated from the data analysis or from models and simulations. Lastly, the data should be readily accessible long after the research is complete, to allow other research groups to reanalyze the material, and for legal, legislative, or regulatory requirements.

All of these data types will be associated with a variety of metadata which describe the subjects, the instrument hardware (e.g., amplifier and filter settings), manufacturing data, the investigators themselves, the quality of the data, etc. Large scale studies may also involve the collection and analysis of pre-existing data collected from published literature, along with the corresponding metadata.

For simple experimental studies, data are often stored in multiple binary and text based formats, often specific to particular software applications such as Matlab, Excel, or R. Individual files can be stored in specific directories, with metadata provided in "readme" files or embedded in XML files (or as hardcopy). Relatively small amounts of data can be stored in XML files and other text files, which can be readily shared across systems. However, as the data and metadata requirements increase, the time and effort in labeling, storing, tracking, and documenting the data becomes excessive and storage requirements negate the use of uncompressed binary formats, which may be problematic later if the original software used to generate the file formats is unavailable. One can use SQL databases, but they are not particularly suited for storing very large data sets such as time series and video files, nor do they provide the ability for the inclusion of complex metadata.

HDF5

A number of scientific disciplines that have to deal with very large and very complex data sets have adopted the Hierarchical Data Format (HDF)[11] as the primary database technology, with HDF5 (released in 2002) being the latest incarnation. Developed by the National Center for Supercomputing Applications, it was first released in 1988, and has been extensively used by physicists, astronomers, climate scientists, earth scientists, and material scientists with continuing support from NASA and the U.S. Department of Energy. For example, the Root software, developed by CERN for the storage and analysis of the data collected by The Large Hadron Collider, uses the HDF5 format[9] and it has been adopted by Matlab as its core file format. With the increasing ability of desktop systems to work with very large datasets, life scientists have used the HDF technology often to support collaborative research and data exchange.[1–3,5–8]

Reprint & Copyright © by the Aerospace Medical Association, Alexandria, VA.
DOI: 10.3357/AMHP.4346.2015

HDF5 allows one to store virtually an unlimited amount of heterogeneous data, composed of multiple datatypes, along with the all the metadata (or attributes) in a single object. The HDF5 Application Programming Interface (API) is a collection of open-source modules available for a number of programming languages such C, C++, C#, Matlab, Java, IDL, Python, Fortran, and R for generating, storing, documenting, and extracting data. An HDF5 database resembles a standard computer file system, with its root directory, subdirectories (or folders), and files; however, the databases can be migrated among various computer architectures, without worrying about the particulars of the hardware or operating system formats, such as byte order. An HDF5 is basically a container similar in concept to the video file containers, which store the video and audio channels, along with the codecs (the metadata). Instead of directories and files, HDF5 organizes data by group and datasets (the numerical data in array format). The database root will be displayed as single computer file with the .h5 extension, but will contain the entire group, subgroup, and dataset structure. The root, groups, and datasets have attributes for the storage of the metadata. Groups can have an almost unlimited set of subgroups, and different datasets can store almost any kind of data (**Fig. 1**). Some of the features that make HDF5 databases attractive to anyone dealing with large, complex datasets include:

- Free data model and open-source API that runs on laptops to massively parallel systems with thousands of processors.
- Portable file format, with distributions for Windows, Mac, and all of the varieties of Unix and Linux operating systems.

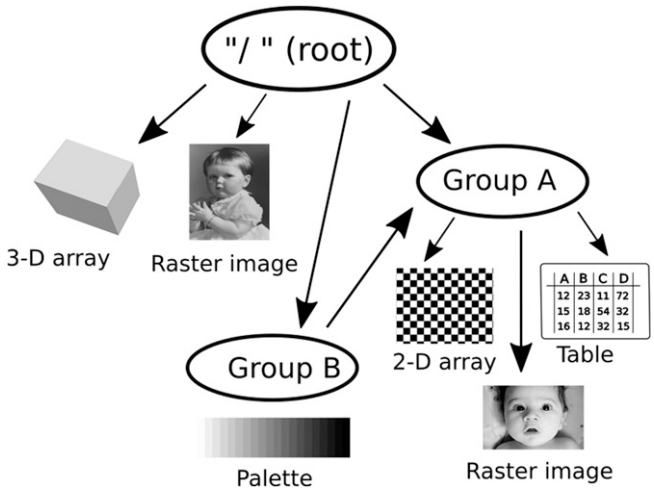

Fig. 1. A schematic of a simple HDF5 database with its root and two groups. The root contains two datasets: a 3-D array of numerical data and a raster image, as well as Groups A and B. Group A contains three datasets: a 2-D numerical array, a table with labeled columns, and a raster image. Group B contains one raster image. The arrow from Group B to Group A indicates that Group B is both an object within the root as well as an object within Group A.

- Database size is only limited by the available physical storage and a single database can be distributed across multiple system file or disks.
- No size restrictions on individual datasets.
- All datasets are mutable.
- Automated generation of specialized metadata, including endianness, size, and architecture.
- Data elements containing other data objects, such as a combination of float, double, and an array of floats.
- Datasets can be associated with more than one group.
- Multidimensional array data storage for integer, float, string, floating point vectors, and bit-length encoded data (including time series, audio, image, and video frames).
- Multiple compression options.
- Extraction of data with SQL-like queries.[10]
- Interactive scripting in documents and electronic notebooks for direct interaction with the datasets.
- Multiple tools and applications for using the HDF5 technology.
- Extensive online documentation, video tutorials, and books for learning to use the HDF5 technology.
- Ability to embed the HDF5 API in custom applications for use by nonprogrammers.
- GUI interfaces for a spreadsheet view of small, two dimensional datasets or subsets of the very large data arrays[4] (**Fig. 2**).
- Attachment of attributes (small named datasets) to groups, subgroups, and datasets, such as analog/digital conversion specifications, literature citations, experimental conditions, and investigator information.
- Extensibility such as inline processing of the data during data storage and extraction, or such as passing time series and image data through digital filters or feature extraction algorithms.
- Efficient analysis of datasets too large for the computer processor memory.
- Python APIs for analysis with multiple processors across multiple computers.

More discussion of the features can be found at the HDF website and YouTube has a number of excellent tutorials and lectures on HDF5 for both the researcher and programmer.

REFERENCES

1. Bernstein HJ, Folk MJ, Benger W. Communicating scientific data from the present to the future. NSF Funded Workshop "Research Data Lifecycle Management"; July 18-20, 2011; Princeton University, Princeton, NJ. New York: Columbia University; 2011.
2. De Macedo DDJ, Von Wangenheim A, Dantas MAR, Perantunes HGW. An architecture for DICOM medical images storage and retrieval adopting distributed file systems. International Journal of High Performance Systems Architecture. 2009; 2(2):99–106.
3. International Neuroimnformatics Coordinating Facility. Electrophysiology Task Force. [Accessed 20 April 2015]. Available from http://www.incf.org/activities/our-programs/datasharing/electrophysiology-task-force.

Downloaded from https://prime-pdf-watermark.prime-prod.pubfactory.com/ at 2025-05-13 via free access

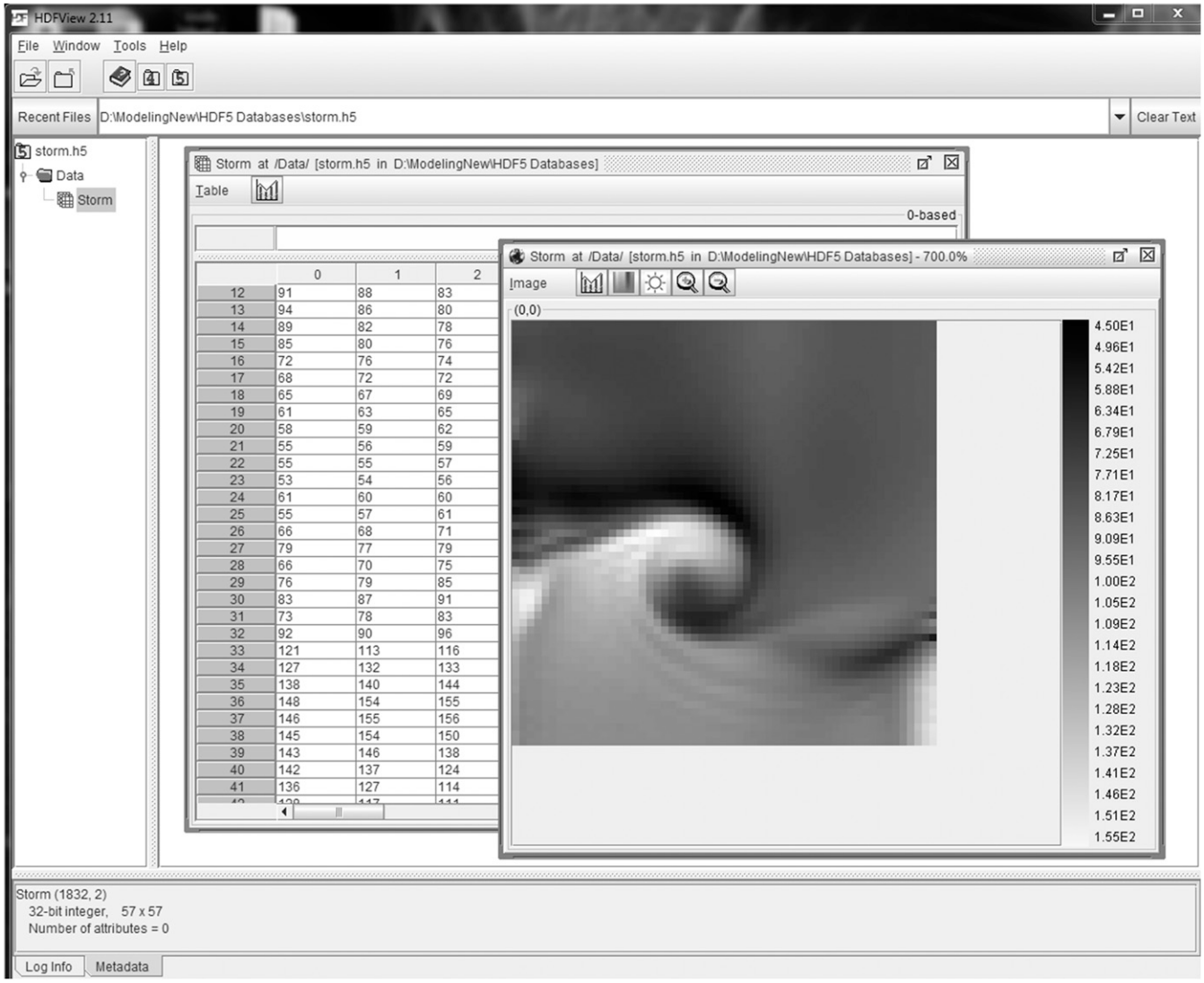

Fig. 2. A screenshot of the HDFView application that provides the ability to view any 2-D slice of a dataset. The dataset is displayed as both a 2-D array of numerical values and the corresponding raster image generated from the numerical values. Data can be entered and edited with the HDFView application.

4. The HDF Group. HDFVIEW. [Accessed 20 April 2015]. Available from https://www.hdfgroup.org/products/java/hdfview/.
5. Karr JR, Phillips NC, Covert MW. WholeCellSimDB: a hybrid relational/HDF database for whole-cell model predictions. Database (Oxford). 2014; 2014. pii: bau095.
6. Loman NJ, Quinlan AR. Poretools: a toolkit for analyzing nanopore sequence data. Bioinformatics. 2014; 30(23):3399–3401.
7. Neuro HDF documentation. [Accessed 20 April 2015]. Available from https://neurohdf.readthedocs.org/en/latest/index.html.
8. Pyl PT, Gehring J, Fischer B, Huber W. h5vc: scalable nucleotide tallies with HDF5. Bioinformatics. 2014; 30(10):1464–1466.
9. ROOT Data Analysis Framework. [Accessed 20 April 2015]. Available from https://root.cern.ch/drupal/.
10. Santhanagopalan V. A metadata based approach for supporting subsetting queries over parallel HDF5 datasets [M.Sc. Thesis]. Columbus (OH): The Ohio State University; 2011.
11. The HDF Group. HDF5 home page. [Accessed 20 April 2015]. Available from https://hdfgroup.org/HDF5/.

Downloaded from https://prime-pdf-watermark.prime-prod.pubfactory.com/ at 2025-05-13 via free access