MERGING NETCDF AND HDF5

Russ Rew* and Ed Hartnett
Unidata Program Center, Boulder, Colorado

1. INTRODUCTION

Unidata's netCDF data model, data access libraries, and machine independent format are widely used in the creation, access, and sharing of geoscience data. More recently, NCSA's HDF5 data model, libraries, and format have been used in NASA Earth Science programs and in large computational models that require parallel I/O. Under a two-year award from the NASA Advanced Information Systems Technology (AIST) program, Unidata and NCSA are collaborating to merge netCDF and HDF5 to achieve gains in performance and interoperability.

If the merger is successful, users of netCDF will benefit from support for packed data, large datasets, and parallel I/O. Users of HDF5 will benefit from the availability of a simpler high-level interface suitable for array-oriented scientific data, wider use of the HDF5 data format, and more software for data management, analysis and visualization that has evolved among the large netCDF user community. The overall goal of this collaborative development project is to create and deploy software that will preserve the desirable common characteristics of netCDF and HDF5 while taking advantage of their separate strengths: the widespread use and simplicity of netCDF and the generality and performance of HDF5.

We summarize the current status of the work, including new software that uses the current netCDF interface with the HDF5 format to provide backward compatibility with current data and application software. We also describe enhanced netCDF-4 software under development and how it makes use of advanced features of HDF5.

2. BACKGROUND

Although netCDF (Unidata, 2003) and HDF5 (NCSA, 2003) are two different file formats, they are also much more than that. Each is a data model that organizes a collection of associated abstractions into a high-level view of how to access data. By "high-level", we mean above the level of input-output facilities provided in particular programming languages, so both the netCDF and HDF5 data models have realizations in libraries for use in multiple programming languages. Using a data model makes it possible to deal with data objects such as variables, dimensions, and attributes rather than bits, bytes, and disk blocks. Using a data model appropriate for scientific data access provides advantages similar to using the relational model for database access: a logical view of data independent of how the data is actually stored.

Developed independently, the netCDF and HDF5 data models have nevertheless evolved similar abstractions for thinking about data, although some of these abstractions have different names and different detailed characteristics. For example, each data model has the notion of a named multidimensional array of data elements of the same abstract type: a variable in netCDF and a dataset in HDF5.

The groups who develop and maintain these two data models along with the associated formats, language bindings, libraries, documentation, and support, have recently been awarded resources to attempt a merger that would use a single data format to implement both data models, exploiting their similarities and at the same time enhancing the power and utility of the combined software. Below, we describe in detail how this will be done, and in particular proposed solutions to thorny problems of compatibility with existing programs and data archives.

2.1 NetCDF

Developed at the Unidata Program of the University Corporation for Atmospheric Research (UCAR), netCDF is widely used in earth, ocean, and atmospheric sciences because of its simple data model, ease of use, portability, and strong user support infrastructure. Programming interfaces support access to netCDF data from C, Fortran77, Fortran90, C++, Java, Perl, Python, Ruby, and other languages, as well as a large number of widely used analysis and visualization packages. The Java netCDF interface described by Caron (2003) is notable for support for remote data access using HTTP and OPeNDAP protocols, virtual netCDF datasets specified in XML, and data model extensions for general coordinate systems. Very recent developments include an interface for writing and reading netCDF data on high performance platforms with parallel I/O (pnetcdf, developed by researchers at Northwestern University and Argonne National Laboratory).

2.2 HDF5

Developed at the National Center for Supercomputing Applications (NCSA) with support and involvement from NASA and DOE, HDF5 is particularly suited for high performance computing. HDF5 has built-in support for chunking (partitioning data into fixed-size multidimensional chunks for improved access along multiple dimensions), data compression

**Corresponding author address:* Russ Rew, Unidata/UCAR, PO Box 3000, Boulder, CO 80307;e-mail russ@unidata.ucar.edu. Unidata is sponsored by the National Science Foundation.

and a "virtual file" I/O layer that enables applications to read or write HDF5 data and metadata in a number of different ways. For example, HDF5 supports MPI I/O, efficient parallel I/O on multiprocessor platforms via the Message Passing Interface (MPI) standard. HDF5 is becoming a de facto standard for high-performance scientific data management in government, industry, and academia. Key users include groups at the DOE Laboratories, the ESDIS project, and several groups in the NSF-funded National Computational Science Alliance. HDF5 has also been chosen for distributing NPOESS data products to users.

3. APPROACH

The ultimate goal of the proposed merger is implementation of an enhanced netCDF interface (netCDF-4) using an enhanced HDF5 as its storage layer. NetCDF-4 will extend the netCDF data model while preserving compatibility with existing netCDF data and software. Our approach is to realize this goal in two steps. First, we will implement the current netCDF-3 C interface on top of the current HDF5 (version 1.6.1) library and format. This will verify the soundness of the approach to providing compatibility with both the format and API for current users of netCDF-3. Second, we will implement an enhanced netCDF-4 interface on an enhanced HDF5 library and format, using the experience gained in implementing netCDF-3 on top of HDF5. The benefits of the combination will be demonstrated in advanced Earth science modeling efforts.

The resulting library will still be compatible with current HDF5 data and applications that may continue to use the HDF5 interface to the library directly. The netCDF-4 interface is intended only for access to HDF5 data written through the netCDF-4 or earlier netCDF interfaces. Some HDF5 features may not be supported through the netCDF-4 API, so using the HDF5 library directly will be necessary to make use of such features. Figure 1 illustrates the relationships among applications, interfaces, and data formats for netCDF-3, netCDF-4, and HDF5.

4. MAPPING DATA MODEL ABSTRACTIONS

The netCDF abstractions that must be represented in the HDF5 data model are dimensions, variables, and attributes. In addition, netCDF has a convention for coordinate variables, with the same name as a dimension, that contain coordinate values. To avoid confusion in discussing the mapping of the netCDF data model to the HDF5 data model, we will use lower case for netCDF abstractions and capitalize the names for HDF5 abstractions.

In developing the netCDF-3 on HDF5 prototype, we have investigated several ways of mapping netCDF objects onto HDF5 objects. The current mapping stores all netCDF objects in a single top-level HDF5 Group named "_netCDF". The netCDF dimensions are stored in an HDF5 compound Dataset named "_ncdim_info" that contains netCDF dimension IDs, names, and lengths. NetCDF global attributes (not associated with any variable) are represented as HDF5 Attributes with the same names in the "_netCDF" Group. Each netCDF variable is stored in an HDF5 Dataset with the same name. This Dataset has HDF5 Attributes corresponding to the netCDF variable attributes, and in addition has a special Attribute named "_ncvar_info" of compound type that contains the netCDF variable ID and the list of dimension IDs that specify the variable's shape.

This mapping is natural, but can lead to significant inefficiencies if implemented naively. In particular, since HDF5 has no representation in its data model for a creation order for datasets or, a naive implementation would either not preserve the order in which netCDF variables and their attributes are created, or require accessing extra data to iterate through variables or attributes in the same order as in the corresponding netCDF software. By initially accessing and caching all the associated netCDF metadata on opening a file, it is easy to capture this information for efficient access. Although preserving the creation order of variables and attributes is not important for many applications, it has uses in a few generic utilities that make it a requirement for the netCDF to HDF5 mapping.

The netCDF to HDF5 mapping described above is fixed for use in the current prototype, but some of its artifacts, such as the "_ncdim_info" dataset, may disappear when additions are made to HDF5 to handle this natively. For this reason, the HDF5 format used for the netCDF-3 prototype should be considered experimental and not supported after the netCDF-4 HDF5 format is released.

5. COMPATIBILITY WITH NETCDF-3

Two kinds of compatibility are important to current netCDF-3 users: ability to access data in the current netCDF file format, and ability to continue to run programs that make use of the netCDF library with little or no change other than linking the program to the new library. Much of the effort in developing the first netCDF-3 over HDF5 prototype has gone into implementing both kinds of compatibility.

5.1 Data compatibility

The ability to access old netCDF data is achieved by including the current library as part of the new library, but changing the names of all public functions to avoid name clashes. For each opened file, a pointer is maintained to the appropriate structure of function pointers for either the old netCDF-3 library or the new netCDF/HDF5 library. A file-type test is only necessary on file open, and the correct function dispatch takes place for all subsequent accesses. Both read and write access to old netCDF files are supported. An attempt to invoke netCDF-4 extended functionality on an old netCDF file will result in an error.

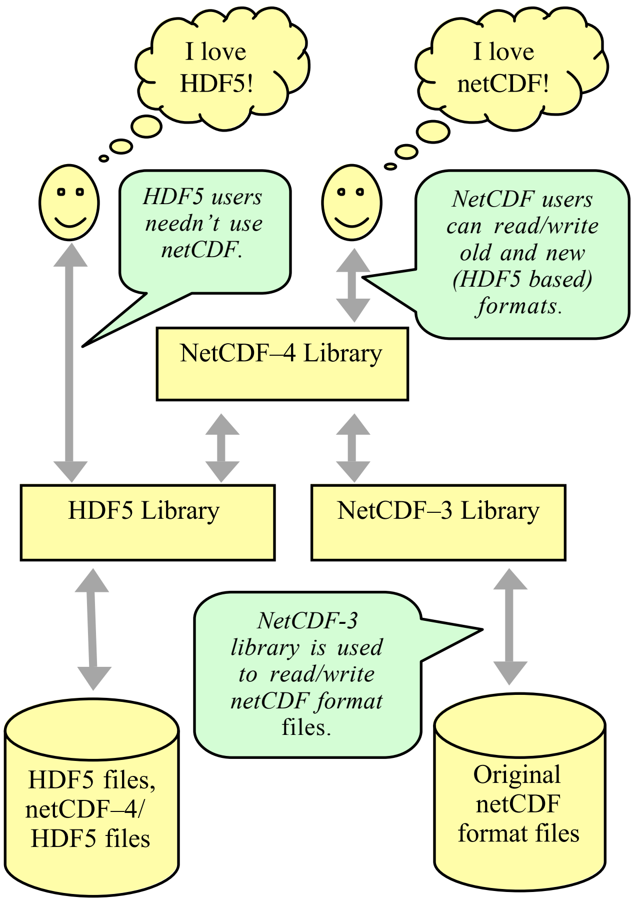

Figure 1. NetCDF-4 Architecture

5.2 API compatibility

A goal for upward compatibility is to provide benefits of using the HDF5-based format by merely relinking existing netCDF applications. In order to achieve this level of compatibility, we are working to ensure that the current extensive test suite for netCDF-3 runs successfully without modification when compiled and linked with the new library. The netCDF create function will, by default, create an HDF5 file, but it will be possible to create a netCDF format file as well. Opening and writing to an existing netCDF file will not change its format. Some new error return codes will be added for netCDF-4, and the netCDF-3 "define mode" restrictions will be considerably loosened, but these changes will not affect API compatibility in most cases.

6. EXTENSIONS NEEDED TO HDF5

Although the HDF5 format is capable of supporting the netCDF interface, with NCSA developers we have determined some additions to HDF5 that may improve how well the netCDF interface can be implemented:

- Shared dimensions (called "scales" in HDF5) to emulate netCDF's shared dimensions directly
- Type conversion facilities, so it is possible to request numeric data as doubles for example, without needing to know whether the data is stored as integers, floats, or doubles
- A new indexing facility that we could use to simplify the representation of object creation order
- Efficiency improvements for cases where current netCDF I/O is faster than HDF5 I/O

The shared dimension feature is an integral part of netCDF. Since it does not (yet) exist in HDF5, we emulate shared dimensions with an extra table of dimension information that includes the dimension name, ID, and length and arrange that each variable stores the IDs of its associated dimensions.

7. NETCDF-4 EXTENSIONS ENABLED BY HDF5

NetCDF-4 features made possible by using the HDF5 library and format include

- Large file support
- Parallel I/O
- Multiple dynamic dimensions
- Packed data, compression
- New data types
- Dynamic schema modifications

The Argonne/Northwestern pnetcdf software also provides parallel I/O and may soon include large file support. It will be interesting to see how netCDF-4 and pnetcdf compare, since they use two very different approaches to provide this functionality to the netCDF user community. NetCDF-4 will provide additional features at the cost of additional library size and interface complexity, but the nature and magnitude of these and other engineering tradeoffs will remain unknown until the project is completed.

8. WORK IN PROGRESS

At the time of this writing, the compatibility approach chosen for the project has proved to be practical and is in use in the netCDF-3/HDF5 prototype, scheduled for completion in early 2004. Most of the unit tests developed for netCDF-3 pass unmodified, and we expect to have all of the tests passing soon. At this point, the public interfaces in HDF5 have proved suitable for implementing all the features and data structures in netCDF-3, but we have identified some enhancements for HDF5 that could further improve the efficiency and maintainability of the netCDF interface.

9. FUTURE PLANS

Testing and tuning the netCDF-3/HDF5 prototype will provide experience needed to develop an enhanced netCDF-4 interface to HDF5 data in 2005. The combined library will provide the scientific community with the benefits of both products. Users of netCDF will benefit from support for packed data, larger datasets, and parallel I/O, all of which are available with HDF5. HDF5 users will benefit from the availability of a simpler high-level interface suitable for array-oriented scientific data; wider use of the HDF5 data format; and the wealth of netCDF software for data management, analysis and visualization.

ACKNOWLEDGMENTS

This work is supported by the NASA Earth Science Technology Office under NASA award AIST-02-0071. We would also like to acknowledge the close collaboration of Mike Folk, Quincey Koziol, and Robert E. McGrath from NCSA and the work of Nancy Yeager from NCSA on an earlier prototype for layering netCDF on HDF5.

REFERENCES:

Li, J., W. Liao, A. Choudhary, R. Ross, R. Thakur, W. Gropp, R. Latham, A. Siegel, B. Gallagher, M. Zingale, 2003: Parallel netCDF: A High-Performance Scientific I/O Interface. SC2003, Phoenix, Arizona, ACM.

NCSA's HDF5 web site: http://hdf.ncsa.uiuc.edu/HDF5/

Unidata's netCDF web site: http://www.unidata.ucar.edu/packages/netcdf/