

MuQun Yang\* and Hyo-Kyung Lee  
The HDF Group

## 1. INTRODUCTION

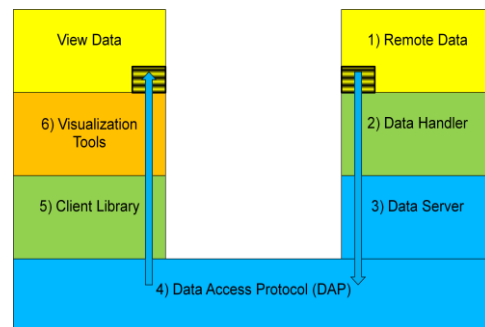
Nowadays most atmospheric scientists are used to downloading Earth Science data via the Internet and then visualizing or analyzing the data on their local machines. As the data volume increases, downloading data becomes an ever more time-consuming procedure. Given that scientists are usually interested in only a small subset of the data in the file, the whole process of fetching the file seems excessively burdensome. The Data Access Protocol (DAP) is a framework that simplifies the retrieval of scientific data by allowing users to browse and select only the subset of data they are interested in.

The OPeNDAP [1] project is an open source project for implementing the DAP. It provides both server and client libraries that can support different scientific data storage formats such as NetCDF [2], HDF4 [3], and HDF5 [4].

A complete OPeNDAP data-processing system, as shown in Figure 1.1, consists of the following components: 1) the remote data in a supported storage format; 2) an OPeNDAP data handler to map the remote data into the DAP; 3) an OPeNDAP data server; 4) an OPeNDAP library to transfer data over the Internet; 5) an OPeNDAP client library to receive the data; and 6) an OPeNDAP visualization and analysis tool built with the client library to visualize and analyze the remote data.

After initial setup, all components of the OPeNDAP system work together automatically and seamlessly, and can deliver subsets of a remote data file to scientists for convenient and quick analysis. The OPeNDAP data-processing system greatly reduces the time that it takes for scientists to access and analyze remote data.

In this paper, we focus on the fifth component of this system. We demonstrate how an HDF5-friendly OPeNDAP client library can help access HDF5 data by retrieving HDF5 objects ignored by other client libraries.



**Figure 1.1.** An OPeNDAP data-processing system; the shaded boxes illustrate that subsets of data can be transferred.

## 2. BACKGROUND

Among the components in an OPeNDAP system, the HDF5-OPeNDAP handler plays the critical role of mapping HDF5 data to DAP so that OPeNDAP clients can retrieve data stored in HDF5 files through the OPeNDAP server. Since DAP and HDF5 don't share the same data model, it is inevitable that HDF5 features that cannot be mapped to DAP will be dropped. However, HDF5 features that can be mapped to DAP, allow many OPeNDAP visualization tools to display and analyze a significant number of NASA HDF-EOS5 files successfully.

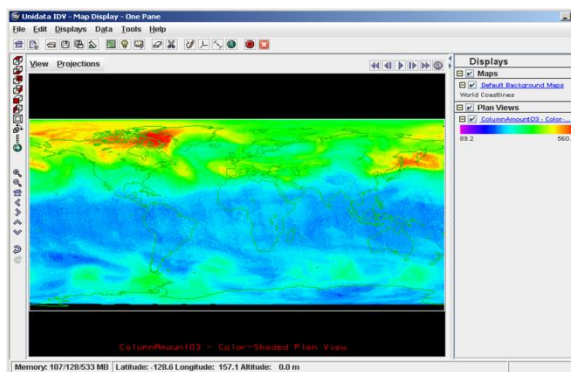
We can use Ferret [5], ODC [6], IDV [7], ncBrowse [8], NCL [9], GrADS [10], IDL [11], and Matlab [12] OPeNDAP clients to demonstrate NASA Aura HDF-EOS5 OMI L3G data served via the HDF5-OPeNDAP handler and an OPeNDAP client library. The HDF5-OPeNDAP handler has been released via OPeNDAP Inc [13]. It has been used by NASA GES DISC to serve NASA Aura OMI Level 3 HDF-EOS5 data [14]. The details about features of the HDF5-OPeNDAP handler are fully explained in "Accessing HDF5 data via OPeNDAP" [15].

---

\*Corresponding author's address:  
MuQun Yang, The HDF Group,  
1901 S. First St., Suite C-2  
Champaign, IL 61820  
E-mail: myang6@hdfgroup.org

Many OPeNDAP visualization tools were initially developed to read and visualize data from local NetCDF files. Later, they were extended to visualize data from remote NetCDF files served by DAP. Such extensions were implemented to use the NetCDF OPeNDAP client library, which shares the same APIs as Unidata's NetCDF library. In the process of extending these tools to serve NetCDF files via DAP, most tools follow the CF conventions [16]—a de facto way to process and share atmospheric science files created by NetCDF APIs. For example, as long as the HDF5-OPeNDAP handler provides DAP output conforming to the CF conventions, the IDV visualization tool, which is built with the NetCDF OPeNDAP client library, can display NASA AURA HDF-EOS5 files via OPeNDAP, as shown in Figure 2.1.

However, there are limitations when using NetCDF OPeNDAP client library to access DAP data. The NetCDF OPeNDAP client library processes the DAP output so that the final representation of the DAP output conforms to the netCDF data model. Any part of the DAP data that does not conform to the NetCDF data model will be either ignored or treated as an error if the data is accessed with the NetCDF OPeNDAP client library. The inability to access data not conforming to the NetCDF data model is a major drawback of the NetCDF OPeNDAP client library. Nevertheless, the NetCDF OPeNDAP client library is widely used by many OPeNDAP visualization tools. Currently, there is no easy-to-use OPeNDAP client library that can make all DAP data available.



**Figure 2.1.** Displaying a NASA AURA OMI HDF-EOS5 file using IDV via the HDF5-OPeNDAP handler configured with the CF option. The physical field is ozone and the unit is DU.

In the next section, we describe the details of a few critical problems that can occur when handling HDF5 files with the NetCDF OPeNDAP client library.

### 3. PROBLEMS HANDLING HDF5 FILES WITH THE NETCDF OPENDAP CLIENT LIBRARY

#### 3.1 Introduction to HDF5 Objects

In HDF5, an object called *group* is used to show relationships among objects. An HDF5 file may be viewed as a rooted directed graph, with at least one group, called the *root group*. An object in HDF5 can belong to different groups. HDF5 uses a *link* to distinguish which group an object is associated with. Besides *group*, the most common HDF5 object is an HDF5 *dataset*. Essentially, an HDF5 *dataset* consists of a data array with some metadata information (called attributes) that describes the array.

One can compare HDF5's *group*, *dataset*, and *link* to a UNIX file system: An HDF5 *group* is like a directory; an HDF5 *dataset* is like a file; and a *link* is like a path to a directory or a file. The path of an object starting from the HDF5 *root group* is called an absolute path of the object; otherwise, it is called a relative path of the object. *Object references* and *region references* are HDF5 datatypes that can be used to access HDF5 objects. Primarily, an *object reference* can be used to access an HDF5 *group* or *dataset*. A *region reference* can be used to access regions of an HDF5 *dataset*.

#### 3.2 Key Problems

The HDF5-OPeNDAP handler with the default configuration maps the essential information in an HDF5 file into the DAP. However, the existing NetCDF OPeNDAP client library cannot interpret all of the HDF5-related information. Of critical importance are the following three key problems.

First, HDF5 allows multiple objects under different groups to have the same name. The HDF5 group information is needed to distinguish among these HDF5 objects. Yet, the output of this essential HDF5 group information must be disabled in the HDF5-OPeNDAP handler, in order to generate DAP data that is understood by the NetCDF OPeNDAP client library. Several OPeNDAP visualization tools built with the library are unable to display any HDF-EOS5 grid data served by the default HDF5-OPeNDAP handler [17], which includes the HDF5 group information.

While disabling output of the HDF5 group information allows more OPeNDAP visualization tools to display most HDF-EOS5 grid data, some problems still occur. In particular, if there are grids with the same name under different groups, ambiguities can occur.

Second, HDF5 object and region references are critical for some communities. In the HDF5-

OPeNDAP handler, all HDF5 references are mapped to DAP *Urls*. *Url*-type data returned by the current HDF5-OPeNDAP handler should be regarded by the client application as an HDF5 object with either an object or a region reference datatype. The *Url*-type data must be de-referenced correctly by the server, in response to a request from the client application via the client library, to get the object or region data it references. The NetCDF OPeNDAP client library does not provide an interface that allows client applications to make the de-referencing request.

Finally, many OPeNDAP visualization tools cannot display HDF-EOS5 swath data. If the current HDF5-OPeNDAP handler provided a way to represent swath data that followed the CF conventions, some tools could possibly display the data. This visualization capability is highly desirable.

#### 4. HDF5-FRIENDLY CLIENT LIBRARY PROTOTYPE

To solve the problems discussed in the previous section, we have developed a new prototype library, the HDF5-friendly OPeNDAP client library. This new library provides the same functionality as the existing NetCDF OPeNDAP client library, and also implements additional APIs to support the retrieval of HDF5-specific and HDF-EOS5-specific information.

In this section, we highlight the benefits of our client library and explain some technical details.

##### 4.1 Client Library Design

We designed our client library by following the NetCDF OPeNDAP client library for two reasons. First, the NetCDF OPeNDAP client library offers C APIs. Second, the C APIs are easy to use and are widely accepted by the Earth Science user community.

We also investigated other options. In particular, there are two C/C++ OPeNDAP client libraries available from the OPeNDAP website: OPeNDAP C API (OCAPI) and DAP++. The OCAPI library is a generic C implementation of the DAP client library [18]. The DAP++ library is a C++ implementation of the DAP client and server [19]. We observed that users familiar with the NetCDF library APIs were reluctant to use the OCAPI library, perhaps because of its complexity. The DAP++ library is not written in C and therefore the visualization tools written in C cannot easily call the DAP++ library.

In designing our library, we investigated two software packages: the NetCDF OPeNDAP client

library and the H5 Lite APIs from The HDF Group.

Our investigation of the NetCDF OPeNDAP client library helped us understand how the library works with DAP++. Furthermore, it convinced us to model our APIs after the straightforward and easy-to-use ones in the NetCDF client library, which is very popular among visualization tool developers.

The H5 Lite APIs provided a rough idea of the minimal set of essential HDF5-related APIs that are needed to retrieve HDF5-specific information.

The following table summarizes the HDF5-friendly OPeNDAP client library APIs and compares them to the NetCDF OPeNDAP client library APIs.

NetCDF OPeNDAP Client Library	HDF5-friendly OPeNDAP Client Library
nc_open	oc_open
nc_close	oc_close
nc_inq	oc_inq
nc_inq_varname	oc_inq_var_name
nc_inq_vartype	oc_inq_var_type
nc_inq_varndims	oc_inq_array_ndim / oc_inq_grid_ndim
nc_get_var_float	oc_get_var_array / oc_get_var_grid

**Table 4.1.** Samples of the HDF5-friendly OPeNDAP client library APIs.

##### 4.2 HDF5 Group and Reference Support

The DAP attribute generated by the default HDF5-OPeNDAP handler to convey HDF5 group information is *HDF5\_ROOT\_GROUP*. This DAP attribute contains all group and dataset names in an HDF5 file. Our HDF5-friendly client library provides an HDF5-specific API that allows a client application to determine if a remote data file is an HDF5 file. The library checks this by identifying the *HDF5\_ROOT\_GROUP* attribute. Another API retrieves the group information stored in the *HDF5\_ROOT\_GROUP* attribute.

For HDF5 references, there is one API called *oc\_hdf5\_get\_var\_url*. It first retrieves the URL string from a DAP *Url*-type variable. It then fetches the referenced data by issuing a request to the OPeNDAP server.

##### 4.3 HDF-EOS5 Swath Visualization Support

The HDF5-friendly client library provides APIs that allow a client application to determine whether a file is an HDF-EOS5 file, and whether it contains swath data. The library identifies an HDF-EOS5 file by checking whether a group called *HDFEOS INFORMATION* exists and detects an HDF-EOS5 swath by checking whether a group called *SWATH* exists.

Geo-location fields are identified by looking for the names of the DAP Grid map variables *XDim* and *YDim*. Many OPeNDAP visualization tools cannot visualize HDF-EOS5 swath data because swath geo-location fields are described by two-dimensional arrays, which cannot be understood by those tools. To solve this problem, we implemented a fast swath-to-grid mapping algorithm inside the *oc\_he5\_swath\_to\_grid* API. Visualization tools can call this API to map a swath array variable to a grid that has one-dimensional geo-location fields. The algorithm that maps swaths to grids is rudimentary and needs further tuning in the future.

## 5. INTEGRATION WITH GRADS

To demonstrate the usefulness of our client library, we have integrated it with a modified version of the GrADS OPeNDAP visualization tool. GrADS is an interactive tool used for easy access, manipulation, and visualization of Earth Science data. It has a programmable interface (scripting language) that allows for sophisticated analysis and display applications [10]. One goal of the integration effort is to demonstrate that at least one popular OPeNDAP visualization tool can use our client library to serve DAP data. Another goal is to evaluate the amount of effort needed to modify the source code of GrADS in order to make it work with our client library.

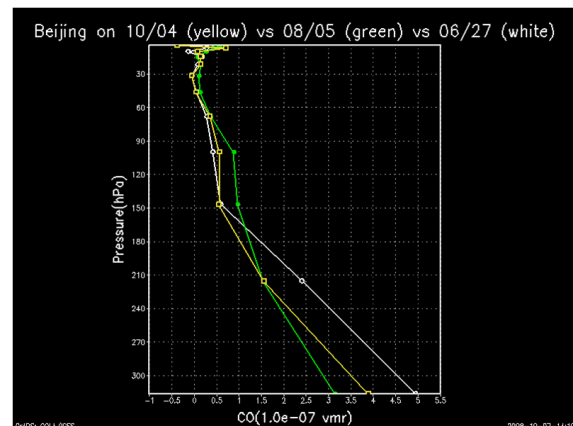
We chose GrADS because it is an open source tool written in C, and because it is easy to add another module inside GrADS to support new command sets without modifying existing modules. For example, we easily added a new command called *ocopen*, which plays a role similar to that of the existing *sdfopen* command in GrADS—whereas the *sdfopen* command calls the NetCDF OPeNDAP client library, the new *ocopen* command calls our library.

Our integration strategy was to understand the remote file access module in GrADS, to distinguish it from the graphics display code, and to add a new file access module that calls our client library. The identification of the existing remote file access was easy since we could trace the API calls made to the NetCDF OPeNDAP client library. The more difficult part was to understand how the graphics display code inside

GrADS works. Once we figured this out, the process was straightforward since our APIs are very similar to NetCDF APIs. Overall, we believe the same integration strategy can be applied to other OPeNDAP visualization tools.

## 6. RESULTS

We can use the modified GrADS tool that calls our HDF5-friendly client library to read and plot NASA AURA MLS swath data via DAP. Figure 6.1 displays a snapshot of the vertical profiles of carbon monoxide (CO) on different dates near Beijing, China. The chosen dates occur before, during and after the 2008 Beijing Olympics games. The amount of CO, reported in units of  $1.0 \times 10^{-7}$  VMR (volume mixing ratio), is shown on the X-axis. Atmospheric pressure, in hector pascals (hPa), is shown on the Y-axis. The CO amounts between 200 hPa (lower stratosphere) and 315 hPa (upper troposphere) indicate measures of pollution. These pressures correspond to elevations of approximately 12 and 9 kilometers above sea level. The plot shows decreasing CO levels as measured by the Aura MLS instrument on the three dates examined, indicating improved air quality during the 2008 Beijing Olympics games [20].



**Figure 6.1.** The vertical profiles of AURA MLS CO data near Beijing, China.

To test how well our client library can reconstruct an HDF5 file served by an OPeNDAP data processing system with a default HDF5-OPeNDAP handler, we also created a simple test application called *dap2h5*. The *dap2h5* test application is linked with the HDF5-friendly client library and writes an HDF5 file.

Figure 6.2 shows the contents of an original HDF5 file in HDFView. Figure 6.3 shows the contents of the file that was written by *dap2h5*. By comparing these two figures, one can verify that the file structure of the reconstructed HDF5 file is identical to that of the original HDF5 file. One should note that many attributes are missing

in Figure 6.3. Reconstructing these attributes may be supported in the future as this prototype tool continues to evolve.

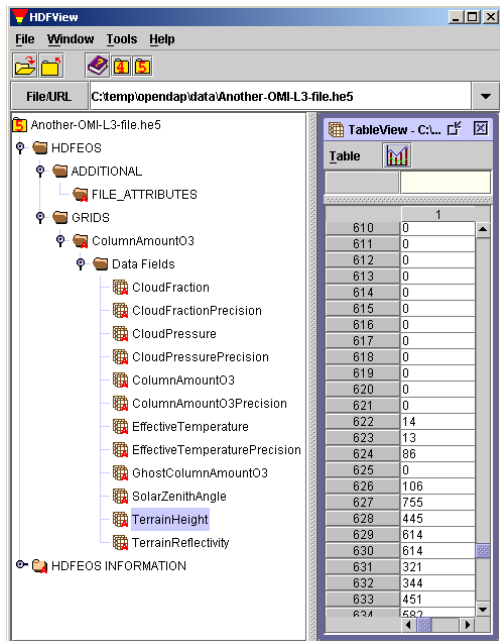


Figure 6.2. An original HDF5 file served by the HDF5 OPeNDAP handler.

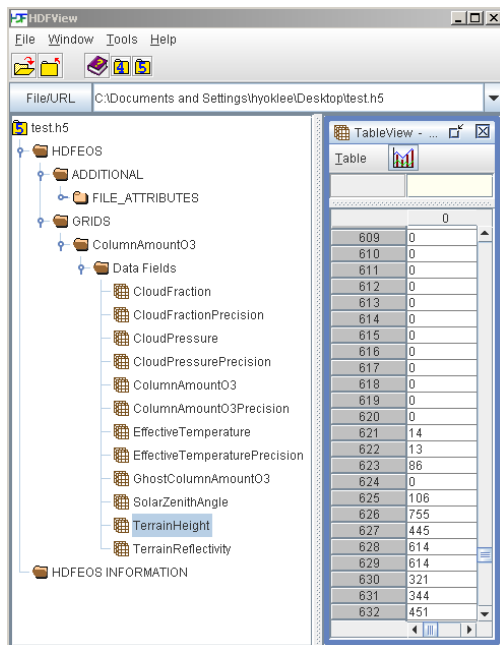


Figure 6.3. A reconstructed HDF5 file using the HDF5-friendly OPeNDAP client library.

## 7. CONCLUSIONS

We have introduced a prototype HDF5-friendly OPeNDAP client library. OPeNDAP visualization tools can use this library to help atmospheric

researchers visualize and analyze additional HDF5 files. For example, one can easily view and analyze an HDF-EOS5 swath with a modified version of GrADS built with our library.

Other features of the HDF5-friendly client library that may benefit applications include support for handling HDF5 groups, and HDF5 object and region references. In summary, our library provides a way for applications to easily access additional HDF5 data via OPeNDAP beyond what is supported by the current NetCDF OPeNDAP client library.

This client library is still a prototype. It is available for download at our website [21]. The website has the GrADS source code that we modified, and the NASA MLS data that we used to demonstrate the results shown in Figure 6.1. The *dap2h5* is also part of the client library distribution, and can be a starting point for further exploration.

If funding allows, we hope to make this prototype a fully tested and supported library. A Java-based, HDF5-friendly OPeNDAP client library would also be very useful, since many OPeNDAP visualization tools are developed in Java.

## REFERENCES

1. OPeNDAP; <http://opendap.org>
2. NetCDF; <http://www.unidata.ucar.edu/software/netcdf>
3. HDF4; <http://hdfgroup.org/products/hdf4>
4. HDF5; <http://hdfgroup.org/products/hdf5>
5. Ferret; <http://ferret.pmel.noaa.gov/Ferret/opendap>
6. ODC – OPeNDAP Data Connector; <http://opendap.org/ODC/>
7. IDV – Integrated Data Viewer; <http://www.unidata.ucar.edu/software/IDV/index.html>
8. ncBrowse – A Graphical netCDF File Browser; <http://www.epic.noaa.gov/java/ncBrowse/>
9. NCL - NCAR Command Language; <http://www.ncl.ucar.edu/>

10. GrADS – Grid Analysis and Display System;  
<http://grads.iges.org/grads/grads.html>
11. IDL;  
<http://www.itvis.com/ProductServices/IDL.aspx>
12. Matlab;  
<http://www.mathworks.com/products/matlab/>
13. HDF5-OPeNDAP Handler;  
[http://opendap.org/download/hdf5\\_handler.html](http://opendap.org/download/hdf5_handler.html)
14. Aura Ozone Mapping Instrument (OM) Global Gridded Level-3 data;  
<http://disc.gsfc.nasa.gov/services/opendap/omi.shtml>
15. Yang, M.; Lee H.; Gallagher, J.; "Accessing HDF5 data via OPeNDAP", 88th AMS Annual Meeting, 2008
16. CF Metadata - NetCDF Climate and Forecast Metadata Convention;  
<http://cf-pcmdi.llnl.gov/>
17. Using DAP Clients to Visualize HDF-EOS5 Grid Data;  
<http://www.hdfgroup.org/projects/opendap/publications/cf.html>
18. OPeNDAP C API (Ocapi) SDK;  
<http://opendap.org/download/ocapi.html>
19. The DAP++ SDK;  
<http://opendap.org/download/libdap++.html>
20. Yang, M.; "HDF5-OPeNDAP Project facilitates use of HDF-EOS Data to track Beijing air quality",  
[http://www.hdfgroup.org/projects/opendap/documentation/HDF5-OPeNDAP\\_Impact\\_2008.pdf](http://www.hdfgroup.org/projects/opendap/documentation/HDF5-OPeNDAP_Impact_2008.pdf)
21. The HDF5-friendly OPeNDAP client library prototype;  
<http://hdfdap.hdfgroup.uiuc.edu:8080/cl.html>

## ACKNOWLEDGEMENTS

The authors would like to thank Dr. Mike Folk at The HDF Group who provided insightful suggestions for this work. We would also like to acknowledge Dr. Robert McGrath at the National Center for Supercomputing Applications at the University of Illinois who was the PI for this project and laid the foundation for the current work.

The authors thank Jennifer Adams from GrADS for testing the HDF5-friendly client library and for providing valuable feedback.

We also acknowledge Dr. Peter Leonard, Daniel Kahn, and Marghi Hopkins at ADNET Systems who kindly provided NASA Aura HDF-EOS5 OMI L3G data for us. Without their gracious help, we could not have successfully demonstrated the use of existing OPeNDAP clients to access HDF-EOS5 data. We also thank the valuable suggestions from Dr. Christopher Lynnes and Dr. James Johnson.

Finally, we would like to thank Ruth Aydt and Herb Morgan, who provided editorial assistance.

This work was supported basing upon the Cooperative Agreement with the National Aeronautics and Space Administration (NASA) under NASA grant NNX06AC83A, NNX08A077A and NNX06AG75A. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NASA.