OPeNDAP's Server4: Building a High Performance Data Server for the DAP Using Existing Software

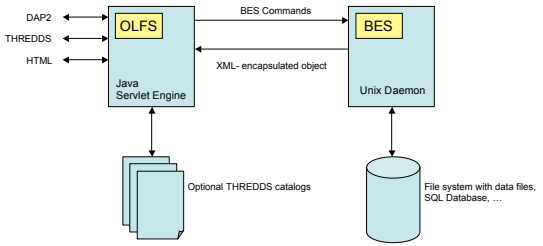
James Gallagher*, Nathan Potter*, Patrick West**, Jose Garcia** and Peter Fox**
*OPeNDAP, Inc., **NCAR/ESSL/HAO

Abstract

OPeNDAP has been working in conjunction with NCAR/ESSL/HAO to develop a modular, high performance data server that will be the successor to the current OPeNDAP data server. The new server, called Server4, is really two servers: A 'Back-End' data server which reads information from various types of data sources and packages the results in DAP objects; and A 'Front-End' which receives client DAP request and then decides how use features of the Back-End data server to build the correct responses. This architecture can be configured in several interesting ways: The Front- and Back-End components can be run on either the same or different machines, depending on security and performance needs, new Front-End software can be written to support other network data access protocols and local applications can interact directly with the Back-End data server.

This new server's Back-End component will use the server infrastructure developed by HAO for use with the Earth System Grid II project. Extensions needed to use it as part of the new OPeNDAP server were minimal. The HAO server was modified so that it loads 'data handlers' at run-time. Each data handler module only needs to satisfy a simple interface which both enables the previously existing data handlers written for the old OPeNDAP server to be used and also simplifies writing new handlers from scratch. The Back-End server leverages high-performance features developed for the ESG II project, so applications that can interact with it directly can read large volumes of data efficiently. The Front-End module of Server4 uses the Java Servlet system in place of the Common Gateway Interface (CGI) used in the past. New front-end modules can be written to support different network data access protocols, so that same server will ultimately be able to support more than the DAP2.0 protocol. As an example, we will discuss a SOAP interface that's currently in development.

In addition to support for DAP2.0 and support for a prototype SOAP interface, the new server includes support for the THREDDS cataloging protocol. THREDDS is tightly integrated into the Front-End of Server4. The Server4 Front-End can make full use of the advanced THREDDS features such as attribute specification and inheritance, custom catalogs which segue into automatically generated catalogs as well as providing a default behavior which requires almost no catalog configuration.



Background

In 1993 the University of Rhode Island began a NASA-funded effort to develop a data system for ocean data that would take advantage of the Internet. Out of that effort, the Data Access Protocol, a server implementing the protocol and several compatible client applications were developed. The data server was based on the Common Gateway Interface (CGI) used for many web-based interfaces, largely because of the simplicity of those types of systems. As the use of the server spread, its limitations became more apparent. At the same time, other issues with the data system became apparent and new work was started to address those issues (such as metadata non-uniformity, searching difficulty and collection organization). Server4 is one aspect of that work. Its goals are to make it easier to solve these problems and to reuse as much of our (and others') existing DAP-compliant software as possible.

The essential features of Server4 are:

❑Two process design can be adapted to a variety of different configurations

❑The server supports several different application protocols for different types of information

❑Formal communication between the processes (to be published soon) will enable other groups to write their own front-ends

Software Reuse

Software reuse is an important part of the Server4 effort. Since the start of work on the DAP, significant effort has been made developing DAP-compliant server software (client software too, although that is not the focus of this poster) and to make a new server we needed to use as much of that work (and knowledge) as possible.

In the OLFS, the THREDDS implementation is 100% from the Unidata THREDDS library. Other elements were written new for this project.

In the BES, the framework itself is an extended version of the Earth System Grid II data server developed by NCAR/ESSL/HAO. It represents about 70% reuse.

The BES Format Modules are about 5% new software with the remaining 95% reused from the Server3 'Data Handlers' (in fact the source code distributions for the format modules still build the Server data handlers too).

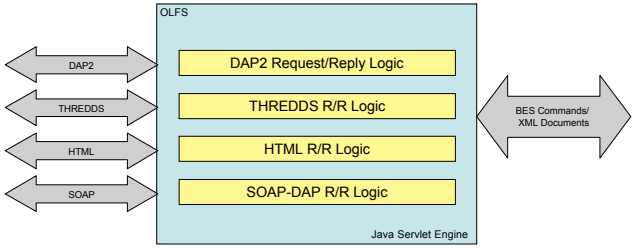
The PPT modules used by both the BES and OLFS are inherited from the ESG II project

The OLFS

The OPeNDAP Lightweight Front end Server (OLFS) is the first of the two processes which make up Server4. The OLFS is implemented in the Java programming language using the Java Servlet (J2EE) technology. The functions carried out by the OLFS are limited to examining the incoming request, passing that request to another process or servlet and packaging the result so that it is a correctly-formatted response. The OLFS does not need to interpret the content of the client requests or BES responses, only how to encapsulated them in different types of MIME, SOAP, or XML documents.

In addition to support for OPeNDAP's Data Access Protocol, version 2 (DAP2), the OLFS provides support for:

- THREDDS data catalogs
- HTML directory pages
- A prototype SOAP interface for DAP (aka DAP4)



The OLFS support for DAP2 is dependent on the functions which are provided by the Back End Server (BES). When the OLFS receives a DAP2 data or metadata request, it recasts that request in terms of commands that can be sent to the BES. The BES's response is then packaged in a MIME document for transmission back to the client program.

THREDDS support in the OLFS takes two forms. First, the OLFS uses a THREDDS catalog to store the bulk of the front end's configuration information. The OLFS also uses the THREDDS Java class library to handle requests for THREDDS catalogs. Data providers can configure the OLFS to return hand written catalogs, automatically generated catalogs, or a mixture of both. The hand-written catalogs are stored on the computer that runs the OLFS; the automatically generated catalogs are built on demand by asking the BES for catalog information. THREDDS catalogs are available as XML documents and as HTML for use with a web browser.

The OLFS also provides an updated version of OPeNDAP Server3 data directories. These directories are HTML pages which can be navigated in a web browser. Unlike Server3's pages, these pages are dynamically generated by the Server4 software, are not dependent on the host computer's web daemon or servlet engine. Clients can switch between the two directory listing types and the OLFS can easily be configured to use one or the other by default.

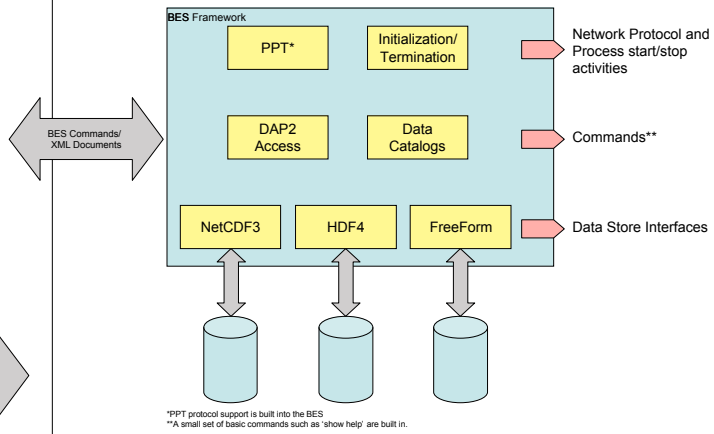
The OLFS also provides an ideal platform to experiment with different data access techniques. The first implemented is a prototype SOAP/DAP4 interface. This interface, although still in a prototype phase, provides several improvements over the older DAP2 interface. It provides a way to request both forms of metadata (syntactic and semantic) in a single XML document, an XML/Multipart-MIME documents response for data and request pooling so that several requests (and their responses) can be pooled and sent to the server in a single network transaction. Other application protocol interfaces are certainly possible and the formalization of the BES communication protocol should simplify their development by groups other than OPeNDAP.

We choose to implement the OLFS using Java Servlets because those provide a good balance between:

- Increased performance
- Ease of installation
- Security features
- Code reuse

By using THREDDS catalogs, Server4 is taking advantage of the development effort of Unidata, Inc. (part of UCAR) and its progress on data organization.

The BES



*PPT protocol support is built into the BES
**A small set of basic commands such as 'show help' are built in.

The BES is a framework for developing high performance data servers developed at the High Altitude Observatory of NCAR/ESSL. The BES's essential feature is that virtually all of its functionality is implemented through *plugin* modules loaded at run-time. This provides a way to combine software components developed for different projects into a new data server.

The BES supports three main types of plugin modules:

- Network protocol and initialization/termination modules
- Command modules
- Format modules

As delivered for use with the OLFS, the BES includes a PPT network protocol module, DAP and data store catalog command modules, and a number of format modules.

The different types of modules are defined by C++ classes which describe the interface that a particular module must implement. Specializations of these classes are compiled to shared-object libraries and loaded by the BES at run-time using information in the BES configuration file.

The BES has more advanced memory management than the Server3 software, which is needed for large data requests.

The BES includes support for SSL authentication, although this is not currently accessible through the OLFS.

The BES is the result of the Earth System Grid II project's needs for a high performance data server which could subset data. By using this software as a basis for Server4, we inherit a platform designed for high performance without the expense of its development. This is higher performance is an important element of Server4.

Availability/Features

❑Beta release of version 1.4 on 18 December 2006, see www.opendap.org/download

❑Coordinated release of data format handlers for:
- ❑NetCDF3 (including large file support)
- ❑HDF4
- ❑FreeForm
- ❑An early version of our HDF5 handler
- ❑Other handlers to follow

Support

❑Use the opendap-tech@unidata.ucar.edu mail list (join at www.opendap.org/mailLists)
❑Or, send a message to support@unidata.ucar.edu.

Presented at the Fall 2006 AGU meeting, San Francisco, CA. Development of Server4 by OPeNDAP funded by NSF with the generous assistance of NCAR/ESSL/HAO.
Reused software components developed using funding from NASA, NOAA and DOE by OPeNDAP, NCAR/ESSL/HAO and others.