

Expanding OPeNDAP

Jason Roberts¹, Glenn Hyland^{1,2}, Andrew Woolf³, David Benda⁴, Steven J. Phipps^{1,5} and Nathaniel Bindoff^{1,6,7}

¹Tasmanian Partnership for Advanced Computing,
University of Tasmania, Private Bag 37, Hobart, Tasmania, Australia, 7001

²now at Australian Antarctic Division,
Channel Highway, Kingston, Tasmania, Australia, 7050

³Science and Technology Facilities Council,
Rutherford Appleton Laboratory, Harwell Science and Innovation Campus,
Didcot, United Kingdom, OX11 0QX

⁴Insight4 Pty Ltd
114 Murray St. Hobart, Tasmania, Australia, 7050

⁵now at CSIRO Marine and Atmospheric Research,
Private Bag 1, Aspendale, Victoria, Australia, 3195

⁶CSIRO Marine and Atmospheric Research,
Castray Esplanade, Hobart, Tasmania 7000

7IASOS, University of Tasmania,

Private Bag 80, Hobart, Tasmania, Australia, 7001

Abstract

We have designed and implemented an OPeNDAP compute server. The modifications presented herein enable a standard OPeNDAP server to perform data-reducing operations at the site at which the data is held, reducing the volume of data transferred over the Internet by up to several orders of magnitude. This new functionality can readily be introduced into a running server. The implementation is also fully extensible, allowing additional functionality to be added as it becomes available.

1 Introduction

OPeNDAP (Open-source Project for a Network Data Access Protocol) is both a protocol and an implementation which allows for ready access to remote data holdings. While OPeNDAP enables access to generic datasets, it has gained particularly widespread acceptance within the earth systems science community. This acceptance is due to its ability to handle self-describing data formats such as netCDF and HDF, and to the ready availability of OPeNDAP-enabled clients. Clients are available for both commercial data analysis and visualisation packages such as Matlab, IDL and Excel, and also for open-source packages such as Ferret, GrADS and NCO. If a user has access to the source code of a non-OPeNDAP enabled client that accesses data through standard library calls, then

it is also possible to enable that client for OPeNDAP by simply linking it against OPeNDAP-enabled versions of the data-access libraries.

When accessing large and remote datasets, OPeNDAP offers significant advantages over the more traditional approach of obtaining a local copy of the file via utilities such as ftp (Roberts et al. 2007). Firstly, users can access remote files using the same analysis and visualisation environment with which they are familiar. They use the same mechanism that they would use to access local files, but specifying a URL rather than a local file path. Secondly, OPeNDAP allows for the sub-sampling of data, in terms of both the variables returned and the spatial or temporal domain of the returned data. This gives the user the ability to select the specific data of interest, without having to transfer entire datasets to a local computer.

While the ability to sub-sample datasets can result in a significant reduction in data volume, the resultant datasets may still be prohibitively large. Earth systems science datasets are often high in spatial resolution, cover a large spatial or temporal domain, or contain a large number of variables. For example, the datasets used in compiling the Fourth Assessment Report of the Intergovernmental Panel on Climate Change (IPCC, 2007) are archived and served by the Program for Climate Model Diagnosis and Intercomparison (PCMDI), and comprise over 32 terabytes of data (PCMDI 2007). These datasets contain the output of an ensemble of computer climate models, each of which was used in turn to produce an ensemble of simulations. Even a single high-resolution simulation can result in a large dataset. For example, the Tasmanian

Partnership for Advanced Computing (TPAC) 1/8° global ocean simulation (Meijers et al. 2007) comprises more than 1.5 terabytes of data.

Typical analysis of earth systems science datasets involve operations that reduce the dimensionality, and hence the associated size, of the data. Operations that fall into this category include averaging, both weighted and unweighted, definite integration, determining the range and variance of the data, and Empirical Orthogonal Function (EOF) analysis. If these operations can be performed at the site at which the data is stored, before being transported across the Internet, then this would result in significant reductions both in the amount of network traffic, and its associated cost, and in the latency to the user. For example, the common operation of temporal averaging on the NCEP/NCAR Reanalysis dataset (Kalnay et al., 1996), which spans the period 1948 to 2006 at six-hourly intervals, reduces the volume of data by a factor of more than 86,000.

The OPeNDAP compute server described herein is one mechanism for performing operations on data before transport over the Internet. The rest of this paper describes the design and implementation of the OPeNDAP compute server. Section 2 introduces the OPeNDAP protocol, Section 3 details the design of the OPeNDAP compute server, Section 4 details the implementation of the OPeNDAP compute server at TPAC, Section 5 discusses some general issues associated with the OPeNDAP compute server, and Section 6 briefly summarises the development of the OPeNDAP compute server.

2 OPeNDAP

OPeNDAP is based on a client/server model, with the client sending a suitably-constructed URL request, and a web server (httpd) delivering the requested data (Sgouros, 2004a,b). This allows for the efficient leveraging of the World Wide Web infrastructure to deliver data transparently to the user. From this perspective, OPeNDAP-enabled analysis and visualisation environments such as Matlab and Ferret can be considered as specialised web browsers (Sgouros, 2004a).

OPeNDAP can handle a number of data formats commonly used within the earth systems science community. These formats include netCDF, HDF, JGOFS, DSP, GRIB and BUFR. These formats are machine-independent and are mostly self-describing, with the file containing both data and metadata.

Sgouros (2004a) gives the general format of an OPeNDAP URL as

Protocol://MachineName/Server/Directories/Filename.Suffix?ConstraintExpression

Both the suffix and constraint expression are optional. The suffix allows for either alternative information or formats to be returned. The constraint expression allows for selection of the data variable, in the case of a file containing more than one variable, and for sub-sampling of the data (Sgouros 2004a,b, Roberts et al. 2007). An example OPeNDAP URL for the TPAC earth systems science digital repository is

<http://dods.acecrc.org.au/cgi-bin/nph-dods/dods-ncep1/pressure/air/air.1948.nc>

Versions 3 and earlier of OPeNDAP are implemented as a Common Gateway Interface (CGI) dispatch script running on an httpd server (Sgouros, 2004b). This dispatch script parses the URL and calls helper CGI applications to respond to the particular type of data request, as specified by the OPeNDAP URL filename suffix.

3 OPeNDAP compute server

OPeNDAP compute operations are selected by a special form of the filename fragment of an OPeNDAP URL. In particular, any filename of the form

```
_calc_<operation>(inputs:<input_args>)(params:<param_args>).nc
```

will request that the operation <operation> be performed on the data specified by <input_args>, using parameters given by <param_args>. An example OPeNDAP compute URL is given below:

```
http://dods.acecrc.org.au/cgi-bin/nph-dods/dods-  
ncep1/pressure/air/_calc_average(inputs:air.1948.nc)  
(params:input_var=air,output_var=avg_air,axis_name=time).nc
```

This will direct the OPeNDAP server to compute the average of the variable 'air' in file 'air.1948.nc' over the 'time' axis, and return the result in variable 'avg_air'

All of the operations implemented to date, with one exception as discussed in Section 5, accept multiple input filenames, using both comma-separated lists and Unix-style wildcards. This ability to operate across multiple files is especially useful because, within many earth systems science datasets, variables are split across multiple files. The NCEP-DOE Reanalysis 2 dataset (Kanamitsu et al., 2002), for example, uses individual files for each combination of variable and year, while long-term numerical climate simulations often have individual files which contain all output variables for one or more decades. An example OPeNDAP compute URL which operates on multiple files is given below:

```
http://dods.acecrc.org.au/cgi-bin/nph-dods/dods-  
ncep1/pressure/air/_calc_average(inputs:air.*.nc)  
(params:input_var=air,output_var=avg_air,axis_name=time).nc
```

This specifies the same operations as in the previous example, but with the averaging performed over all files which match the wildcard expression in the selected directory on the server, eg. `air.1948.nc`, `air.1949.nc` ... `air.2006.nc`. The variable 'air' must exist in all files.

A standard OPeNDAP server can easily be upgraded to an OPeNDAP compute server, without any loss of original functionality. In addition to adding some scripts to the OPeNDAP server (see Section 4), the only modification to the standard OPeNDAP server is the addition of a few lines of code into the dispatch CGI script. These additional lines

of code parse the URL and, if the first six characters of the filename fragment are “_calc_”, transfers control to an external script (the ‘compute parser’) that further parses the URL.

4 Implementation at TPAC

An OPeNDAP compute server has been implemented at TPAC using Python and the Climate Data Analysis Tools (CDAT) (PCMDI 2003, 2004). The compute parser (see Section 3) is a Python script which evaluates the URL, obtains the requested operation (<operation>), and checks whether or not an executable script <operation> exists in the compute server directory. If such a script exists, the compute parser further evaluates the URL <input_args> and <param_args> fragments, generating a unique temporary filename. It then executes the script <operation>, which performs the required operation and saves the result to a unique temporary file. The compute parser modifies the original URL such that it references the temporary file, and then returns control to the dispatch CGI script. This script performs any other specified functions, such as sub-sampling or URL suffix operations, and returns the data to the client.

This design, which dynamically checks for an executable script <operation> at run time, allows for the easy extension of the operations supported. To introduce a new operation into a running server, the system administrator simply needs to copy a suitable script called <operation> into the OPeNDAP compute server directory. There are no restrictions on the format of the script, as long as it can access the underlying data and is callable from Python. For example the script could be a FORTRAN or C executable, a

Matlab M file or a CDAT script. Figure 1 shows the schematic structure of the OPeNDAP compute server.

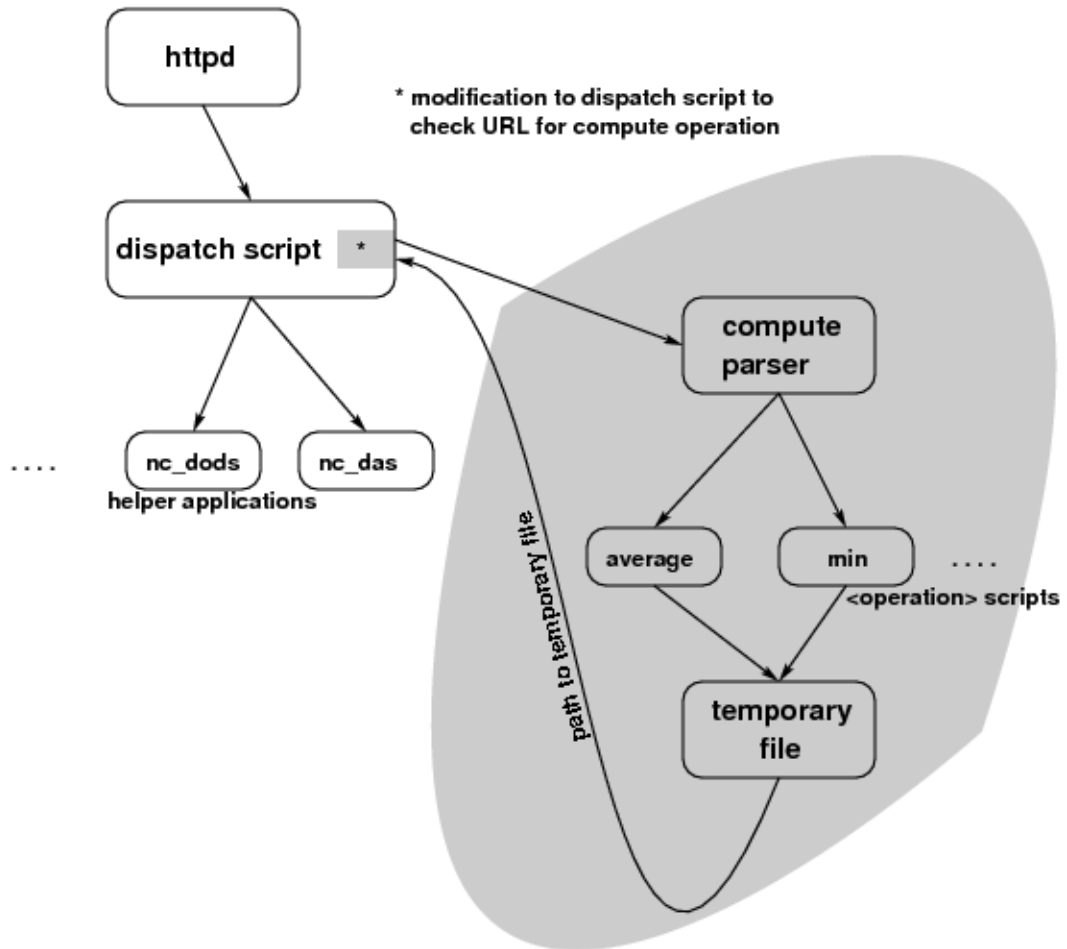


Figure 1: The structure of the OPeNDAP compute server. The shaded areas denote modifications or additions to a standard OPeNDAP server.

Table 1 details the operations initially supported by the TPAC OPeNDAP compute server.

Operation	<Function>	Params:
Average	average	input_var, axis_name, output_var*, axis_weight*
Definite integral	def_integral	input_var, axis_name, output_var*, axis_weight*
Indefinite integral	indef_integral	input_var, axis_name, output_var*, axis_weight*, reverse_axis*
Maximum	max	input_var, axis_name, output_var*
Minimum	min	input_var, axis_name, output_var*

Table 1: Operations initially supported by the TPAC OPeNDAP compute server.

*denotes optional parameters.

The axis_weight parameter defines the weighting function used for the averaging, definite and indefinite integration operations. The default value of this parameter is 'r' (raw) for averaging, and 'm' (metres) for both definite and indefinite integration. The other valid option for axis_weight is 'e' (equal). The metres weighting option will convert angular measurements of latitude and longitude into metres, using a value of 6371km for radius of the Earth, and taking into account the variance of longitudinal spacing with the cosine of the latitude. If the axis is neither a longitude nor latitude axis, then the metres option will revert back to the raw option, which relies upon the spacing of the underlying axis. The output_var parameter is used to name the output variable and, if not specified, will default to the same value as the input_var argument.

CDAT is a freely available data analysis and visualisation environment built on Python (PCMDI 2002, 2003), that is used to implement the operations shown in Table 1. The Climate Data Management System component of CDAT allows for access to single files

and multiple files through a virtual dataset (PCMDI 2003). The CDAT utilities (PCMDI 2002) provide the underlying compute functionality, including the averaging operation used for the average and integration options. The definite integral is implemented as an average without the final division by the sum of the weights, while the indefinite integral is implemented as a series of definite integrals.

Several of the data formats supported by OPeNDAP are self-describing – for example, netCDF and HDF - with the file containing both data and metadata. The metadata includes information which describes the layout of the data (such as the axis names and spacing), the variables contained within the dataset (such as their names, units and valid data range) and the dataset itself (such as the source of the data, and information on the program used to generate the file). If the metadata conforms to a standard, the ability to automate both the interpretation of the metadata and data discovery becomes possible. One metadata standard that has gained widespread adoption within the earth systems science community is the netCDF Climate and Forecast (CF) standard, which specifies, amongst other things, the ordering and labeling of axes. Halliday (2006) gives a detailed description of the CF standard.

The virtual dataset capability of CDAT relies on the underlying dataset adhering to the CF standard. However, the CDAT implementation assumes that the axis label attribute in the metadata is in uppercase, while the CF standard allows for case insensitivity (Halliday 2006). This causes incompatibility with some of datasets in the TPAC digital repository. Whilst this problem could be easily fixed, a more significant problem arises when

accessing multiple files as a virtual dataset. For example, performing a temporal average over all 59 files for a single variable in the NCEP/NCAR Reanalysis dataset would result in a virtual file of 58 gigabytes in size, over which the averaging operation would have to be performed. The alternative, adopted in the TPAC OPeNDAP compute server implementation, is to perform the requested operation on each individual file, and to store the result in a temporary array. The operation is then repeated on this temporary array, after modifying the weighting matrix, if appropriate, to reflect the initial application of the operation. This approach reduces the server memory requirements such that they are only slightly larger than those for performing the same operation on a single file.

5 Discussion

The indefinite integral operation has been included in the initial operations supported by the OPeNDAP compute server for completeness as, for example, it is frequently used in the calculation of oceanic transports. However, this operation does not reduce the dimensionality or size of the dataset, and an implementation of a multiple-file version could cause problems at both the server and client ends, due to excessive memory requirements. Therefore, the indefinite integral operation only supports single files. All other functions shown in Table 1 reduce the dimensionality and size of the dataset, and can support both single and multiple files as arguments.

Due to the modular structure of the OPeNDAP compute server, it is relatively easy to upgrade other OPeNDAP implementations. For example, version 4 of the OPeNDAP server is under development and is based on Java. With the addition of a few lines of

code to parse the URL, to check the first six characters of the filename fragment for the string “_calc_” and to pass control to the compute parser, version 4 installations could be OPeNDAP compute-enabled. THREDDS and Hyrax implementations could be OPeNDAP compute-enabled in a similar manner.

Because the datasets associated with an OPeNDAP server contain metadata, any modification in the data needs to be reflected in modifications to the associated metadata. The code which modifies the metadata is contained within the execution scripts for each individual operation, as the details of the modification are operation-specific. Within the CF standard, the name of each variable is contained in the long_name attribute, and the associated units of the variable are contained in the units attribute. All of the OPeNDAP compute server functions require the alteration of the variable's long_name attribute to reflect the operation performed on the data. For example, performing an average over the time dimension of a variable with a long_name attribute of “Sea Surface Temperature” would modify the long_name attribute to be “time averaged Sea Surface Temperature”. In addition, the two integration operations change the units of the result, and the variable's original units attribute must be multiplied by the units of the axis on which the integration was performed. For example, an indefinite integration of the above “Sea Surface Temperature” (units degrees Centigrade) along an axis called depth (units metres) would result in a long_name attribute of “depth indef integral Sea Surface Temperature” and a units attribute of “degrees Centigrade * metres”.

CDAT can itself be linked against OPeNDAP-enabled libraries, and hence the <input_args> which specify input filenames could be URLs to data held on other OPeNDAP servers. However, the TPAC OPeNDAP server is installed behind an authenticated proxy server. This means that external users can request data from the TPAC OPeNDAP server transparently, but any request for external data that originates from within the authenticated proxy domain requires valid proxy credentials. This causes problems in supporting URLs as valid <input_args>, and hence a non-OPeNDAP enabled version of CDAT has been used. Therefore, while multiple files are supported, all of these files must be local to the compute server. This is a site-specific restriction, rather than a general restriction.

6 Summary

An OPeNDAP compute server has been developed and deployed at the Tasmanian Partnership for Advanced Computing. The implementation required only slight modifications to a basic OPeNDAP server, and maintains complete compatibility with the basic OPeNDAP server. Operations that reduce the dimensionality and size of the dataset have been the initial focus, and these operations support multiple input files. The OPeNDAP compute server is easily extensible for additional operations. While the initial deployment has been in an OPeNDAP version 3 (CGI-based) server, alternative servers such as OPeNDAP version 4 (Java-based), THREDDS and Hyrax can easily be upgraded to a compute server.

Acknowledgments

This research was supported under the Australian Research Council's E-Research Special Research Initiatives funding scheme (project number SR0567397).

OPeNDAP is supplied courtesy of OPeNDAP Inc (<http://www.opendap.org>) and the Climate Data Analysis Tools (CDAT) is courtesy of the Program for Climate Model Diagnosis and Intercomparison (PCMDI) at Lawrence Livermore National Laboratory. The Tasmanian Partnership for Advanced Computing (TPAC) digital repository is an Australian Partnership for Advanced Computing (APAC) and ARC Network for Earth Systems Science funded project.

References

Halliday, K. (2006), CF Conformance Requirements and Recommendations, http://cf-pcmdi.llnl.gov/conformance/requirements_recommendations (accessed 21 June 2007)

IPCC (2007), *Climate Change 2007: The Physical Basis. Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change*, Solomon, S., Qin, D., Manning, M., Chen, Z., Marquis, M., Averyt, K. B., Tignor, M., and Miller, H. L. (eds.), Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 996 pp.

Kalnay, E., Kanamitsu, M., Kistler, R., Collins, W., Deaven, D., Gandin, L., Iredell, M., Saha, S., White, G., Woollen, J., Zhu, Y., Chelliah, M., Ebisuzaki, W., Higgins, W., Janowiak, J., Mo, K. C., Ropelewski, C., Wang, J., Leetmaa, A., Reynolds, R., Jenne, R.,

and Joseph, D. (1996), The NCEP/NCAR 40-Year Reanalysis Project, *Bulletin of the American Meteorological Society*, 77(3), 437-471.

Kanamitsu, M., Ebisuzaki, W., Wollen, J., Yang, S.-K., Hnilo, J. J., Fiorino, M., and Potter, G. L. (2002), *NCEP-DOE AMIP-II Reanalysis (R-2)*, *Bulletin of the American Meteorological Society*, 83(11), 1631-1643.

Meijers, A.J., Bindoff, N.L. and Roberts J.L. (2007), On the Total, Mean and Eddy Heat and Freshwater Transports in the Southern Ocean Hemisphere of a $1/8^\circ \times 1/8^\circ$ Global Ocean Model, *Journal of Physical Oceanography*, 37(2):277-295.

Roberts, J., Hyland, G., Woolf, A., Bindoff, N., Benda, D. and Phipps, S. (2007), An extensible OPeNDAP compute server, *Journal of Research and Practice in Information Technology* (submitted)

PCMDI (Program for Climate Model Diagnosis and Intercomparison) (2007)
<http://www-pcmdi.llnl.gov/sitemap.php> (accessed 21 June 2007)

PCMDI (Program for Climate Model Diagnosis and Intercomparison) (2002) CDAT Utilities Reference Guide Version 3.3, Lawrence Livermore National Laboratory

PCMDI (Program for Climate Model Diagnosis and Intercomparison) (2003) Climate Data Management System Version 4.0, Lawrence Livermore National Laboratory

Sgouros, T. (2004a) OPeNDAP Users Guide Version 1.14, University of Rhode Island and The Massachusetts Institute of Technology

Sgouros, T. (2004b) OPeNDAP Servers Installation Guide Version 1.16, University of Rhode Island and The Massachusetts Institute of Technology