5B.3

CREATE AN ARCHIVE
WITH THE THREDDS DATA REPOSITORY

Anne Wilson, Thomas Baltzer, John Caron
Unidata Program Center, UCAR, Boulder, CO

1. INTRODUCTION

Web 2.0 refers to the current internet trend of hosted services, such social networking sites, wikis, and folksonomies, whose goal is to promote general creativity, collaboration, and sharing. Some have called “Science 2.0” the scientific corollary, an aspect of the broader “Open Science” movement, which also includes Open Access scientific publishing and Open Data practices. Open access (OA) is free, immediate, permanent, full-text, online access, for any user, web-wide, to digital scientific and scholarly material. Similarly, Open Data is a philosophy and practice requiring that certain data are freely available to everyone, without restrictions from copyright, patents or other mechanisms of control. The experience of a growing number of researchers suggests that this style of science is not only more collegial than the traditional approach, but can also be more productive.

For many years, the Unidata Program Center has been helping universities and research labs acquire and use scientific data, through the use of near real time data streams, visualization and analysis tools, and middleware for data access. With the Unidata THREDDS Data Server (TDS), sites can provide access to a data repository that holds metadata and can serve data via a variety of useful protocols, including HTTP, OPeNDAP, and WCS. Subsetting and aggregation of datasets is supported. The TDS provides a way to publish data, thereby promoting Open Data.

The Unidata THREDDS Data Repository (TDR), the topic of this paper, provides complementary functionality to upload scientific data and other content on demand to a repository where the data can be served via a TDS. The TDR can support both human interactive use and programmatic clients. This paper will discuss TDR functionality and describe the two different applications that employ these two interfaces.

2. TDR FUNCTIONALITY

The THREDDS Data Repository (TDR) is a server that provides client initiated, on demand storage for data of any type to be served by the THREDDS Data Server (TDS). With proper authorization, a TDR client can upload new content and metadata to the server or reorganize repository content. The TDR supports functionality to add, delete, move, and copy a dataset, and to edit associated metadata.

The TDR runs under Tomcat, a popular UNIX-based servlet container. In addition to providing access to the TDR (and, possibly, a co-resident TDS), Tomcat provides security via authentication, role-based authorization, and encryption.

Data files can be uploaded to the server as the body of request to the server. Alternatively, an HTTP URL may be provided, whereby the TDR will perform the data movement. In addition to handling single datasets, the uploaded data file could also be a collection of files, which will subsequently be unpacked into a tree of files. Clients may provide a path to indicate where the data should be placed in the repository, or can simply provide an ID, leaving the TDR to determine a storage location.

*Corresponding author address: Dr. Anne Wilson
PO Box 3000 Boulder Colorado, 80307:
anne@unidata.ucar.edu

Upon successful file movement and unpacking, THREDDS catalogs are created that reflect the new content, hold metadata, and provide TDS access URLs to the data. Access URLs are generated based on information provided by the client, and, in some cases, by cracking open the data file. For example, HTTP access is always provided. OPeNDAP access is provided if the data file can be successfully opened by the Unidata Common Data Model (CDM). WCS access is provided if the data file can be opened by the CDM and the user has indicated that it is a gridded dataset. Gridftp URLs can be provided if the server is configured appropriately.

Clients must have a login in order to upload to a TDR. Clients authenticate to the server via Tomcat's authentication mechanism. The name provided is then used by the TDR as the user name. Data uploaded by that user becomes "owned" by that user. Only that user can perform "write" operations on the data, such as update, move, or delete a dataset, or edit metadata.

The TDR has a set of predefined metadata elements pertinent to scientific data. Currently this includes: temporal and spatial domain, a data type (such as grid, image, point, radial, station, etc.), a data format type (such as BUFR, GEMPAK, HDF4, NetCDF, NEXRAD level II, etc.), title, publisher, summary, and some relevant links. Additional arbitrary metadata elements can be provided. These are entered into the THREDDS catalogs as properties.

Like the TDS, while the UPC does provide specific TDR installations for use by designated projects, the TDR is intended to be deployed by other institutions so that they may create and administer their own repositories. It is expected that a TDR will generally be used with a co-resident TDS, though that is not required. The TDR can retrieve content via HTTP only, and does not yet have the capability to browse catalogs.

3. THE TDR LEAD INTERFACE

The term eScience describes computationally intensive science that is carried out in highly distributed network environments, or science that uses very large data sets that require grid computing. The LEAD cyberinfrastructure project, a NSF large ITR project, requires both. LEAD allows users to run complex applications, such as the running a WRF forecast model or the mining of radar data, providing access to high performance computing capabilities that heretofore were only available to a few.

The TDR is expected to be used in the LEAD cyberinfrastructure project as a public repository where LEAD users can publish data and other content related to their experiments. For this application, the TDR has provided a RESTful web service interface. (Representational State Transfer, or "REST", is a style of software architecture for distributed hypermedia systems. Generally REST refers to any simple interface that transmits domain specific data over HTTP without an additional messaging layer such as SOAP or session tracking via HTTP cookies.)

The REST interface is designed around three types of resources: datasets, containers, which hold datasets and other containers, and metadata.

This interface also supports putting content by ID rather than by providing a path. In this case the TDR structures that portion of repository rather than the client and the data can only be retrieved by ID.

A few sample URLs are:

Create a new container called "Anne":

```
PUT
http://servername:8080/tdr/lead/Anne/
```

(This request has no body.)

Add a container called "wxChallenge" and a nested dataset called "experiment_notes" to the above:

```
PUT
http://servername:8080/tdr/lead/Anne/wxChallenge/experiment_notes
```

(The file is attached as body of the request.)

Add metadata to "experiment_notes" to the above:

PUT

```
http://servername:8080/tdr/lead/A
```

```
nne/wxChallenge/experiment_notes/
metadata.xml
```

(Metadata is attached as the body of the request in XML format.)

Update the dataset called "Anne/wxChallenge/experiment_notes":

```
PUT
http://servername:8080/tdr/lead/A
nne/wxChallenge/experiment_notes
```

(The file is attached as the body of the request.)

Update metadata to "experiment_notes" to the above:

```
PUT
http://servername:8080/tdr/lead/A
nne/wxChallenge/experiment_notes/
metadata.xml
```

(Metadata is attached as body of the request in XML format.)

Get the dataset called "Anne/wxChallenge/experiment_notes":

```
GET
http://servername:8080/tdr/lead/A
nne/wxChallenge/experiment_notes
```

Get the metadata associated with "experiment_notes" in XML format:

```
GET
http://servername:8080/tdr/lead/A
nne/wxChallenge/experiment_notes/
metadata.xml
```

For more information about this interface see the TDR web service interface document at: http://www.unidata.ucar.edu/staff/awilson/private/tdr/leadServiceInterface2.html.

4. THE TDR NEXT GENERATION CASE STUDY INTERFACE

The TDR is being used to implement the Unidata Next Generation Case Study project. In the spirit of Science 2.0, this project is intended to provide a living, community-built case study repository. In this case, TDR usage is interactive - clients interact with the server via a web-based form that is designed around the notion of a case study. Case study designers can upload and organize data and related content in the repository. Links to related supporting material can be provided as metadata. Also, remote content can be cataloged, with the content not being uploaded to the repository. This allows the repository to contain content that is hosted elsewhere, such as complex learning modules.

FUTURE DIRECTIONS

Under the goal of developing useful, working software, the prioritization of future directions will be determined by users. With that said, the following areas will be addressed or are under consideration:

- The user interface will be upgraded to support catalog browsing and handling of arbitrary metadata values and generally improved.

- Role-based authorization will provide permission handling at a finer degree of granularity. For example, a system administration role will be provided. Also, owners may be able to transfer ownership.

- Metadata generation could be expanded, providing more automated harvesting of metadata.

- Storage management could be expanded to provide a variety of storage policies, including remote storage such as on a mass storage device.

CONCLUSION

The TDR expands Unidata's suite of packages by providing functionality to populate a repository with scientific data and other content, thereby further enabling Open Access scientific publishing and Open Data practices, and, we hope, furthering science.

REFERENCES

The LEAD portal:
http://portal.leadproject.org/

The THREDDS Data Server: http://www.unidata.ucar.edu/projects/THREDDS/.

Waldrop, M. M., "Science 2.0: Great New Tool, or Great Risk?" Scientific American, to be published. See http://www.sciam.com/article.cfm?id=science-2-point-0-great-new-tool-or-great-risk.

Weber, J., Etherton, B., and Holmberg, S.O., Building a Framework to Facilitate Interactive and Dynamic Education Case Study Modules, 24th Conference on IIPS, 88th Annual American Meteorological Society Meeting, New Orleans, January 2008.

ACKNOWLEDGEMENTS

LEAD is a Large Information Technology Research (ITR) Grant funded by the National Science Foundation under the following Cooperative Agreements: ATM-0331594 (University of Oklahoma), ATM-0331591 (Colorado State University), ATM-0331574 (Millersville University), ATM-0331480 (Indiana University), ATM-0331579 (University of Alabama in Huntsville), ATM03-31586 (Howard University), ATM-0331587 (University Corporation for Atmospheric Research), and ATM-0331578 (University of Illinois at Urbana-Champaign, with a sub-contract to the University of North Carolina).