DAP protocol

A review

Léo Le Lonquer

08/10/24

Overview

1. What is DAP and OPeNDAP ?
2. DAP History
3. DAP Data Model
4. Requesting DAP
5. Tools implementing DAP - Erddap, Thredds, Hyrax
6. Limitations of DAP

What is DAP and OPeNDAP ?

OPeNDAP (Open-source Project for a Network Data Access Protocol) : the organisation

DAP (Data Access Protocol) : the protocol specification

Hyrax : OPeNDAP server implementing DAP

“OPeNDAP” is often used to refer to the protocol or the server. It is good to know which one you are using

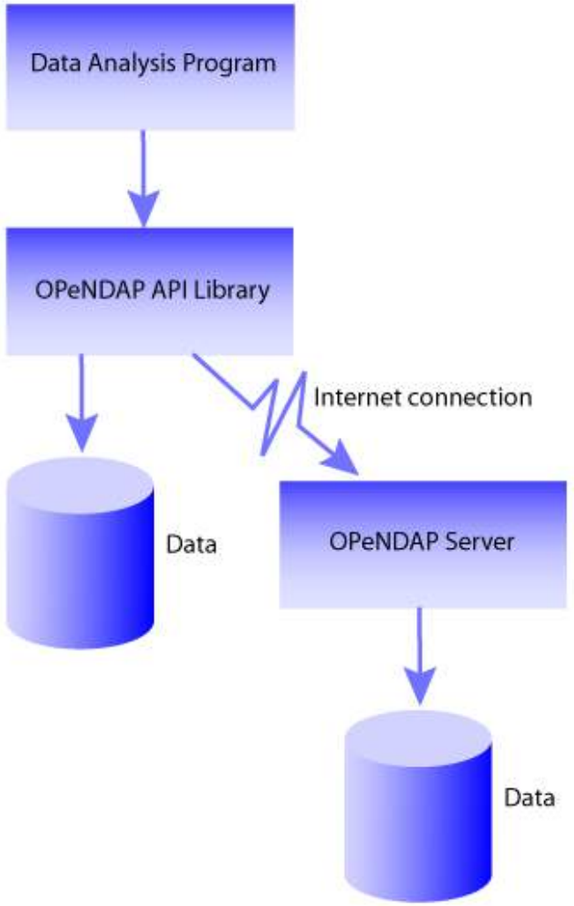
OPeNDAP aims with DAP and Hyrax

Uniform and standardized interface able to

- access **aggregated multidimensional data** stored in different formats - Netcdf, HDF, GRIB
- serve it in the **desired format** - NetCDF, GeoTIFF, JPEG2000, JSON, ASCII - thanks to translation facilities.
- access data as if it was local - “a URL equals a dataset”
- provide a fine-grained access to dataset with subsetting capabilities ““a URL with constraints equals a subset”
- stream dataset as it is updated with new data
- manage metadata for a FAIR and CDM compliance
- act as a middleware between the data, multiple servers and the user
- create a network of interoperable DAP servers

Basic Architecture

OPeNDAP
uses a client/server model



DAP History

1995 : DODS (Distributed Oceanographic Data System) first version of DAP.

The DODS system created and used the dods:// protocol on the Internet. It was created before http://.

2000 : OPeNDAP foundation

2005 : DAP2 (Data Access Protocol) was released, http replaced dods when using DAP

2006 : first version of Hyrax

2014 : DAP4 release (new data model, response model and http use)

DAP Data model (DAP2 & DAP4)

1. **OPeNDAP data model :** an "intermediate data representation" for data sets. This is used to transport data from the remote source to the client.

2. **Dataset Descriptor Structure (DDS):** a description of the shape and size of the various data types stored in some given data set.

3. **Data Attribute Structure (DAS) :** list of attributes of the data such as the units and the name of the variable.

4. **DDX :** an XML version of the DAS and DDS, combined. Nucleus of DAP4 and DMR.

DAP4 Full Specification

DDS Example

Dataset Descriptor Structure

```
Dataset {
   Sequence{
       Int32 day;
       Int32 month;
       Int32 year;
       Float64 lat;
       Float64 lon;
       Float64 O2cal[20];
       Sequence{
       Float64 press;
       Float64 temp;
       Float64 O2;
       Float64 salt;
       } cast;
       String comments;
   } station;
} arabian-sea
```

DAS Example

Data Attribute Structure

```
Attributes {
    Bill {
        String LastName "Evans";
        Byte Age 53;
        String DaughterName "Matilda";
        Matilda {
            String LastName "Fink";
            Byte Age 26;
        }
    }
    Global {
        String Name "FamilyData";
        String DateCompiled "11/17/98";
    }
}
```

DDX Example

```
…
<Array name="longitude">
        <Attribute name="standard_name" type="String">
                <value>longitude</value>
        </Attribute>
        <Attribute name="units" type="String">
                <value>degree_east</value>
        </Attribute>
        <Attribute name="valid_min" type="Float32">
                <value>-180.0</value>
        </Attribute>
…
```

Full example : LPO_GLOBANA_ISAS13_MNTH_TIME_SERIE.ddx

DAP Data model (DAP4 only)

5. Dataset Metadata Response (DMR) : a complete characterization of the associated data source sans content, spelling out its groups, variables, types, dimensions, and attributes as discussed in the preceding two subsections

6. Dataset Services Response (DSR) : describes the various DAP services available for that source, and these always include provision of a DMR and provision of content from the source.

DMR Example (DAP4)

Dataset Metadata Response

HTML version :
http://test.opendap.org/dap/data/nc/coads_climatology.nc.dmr

XML version :
http://test.opendap.org/dap/data/nc/coads_climatology.nc.dmr.xml

DSR Example (DAP4)

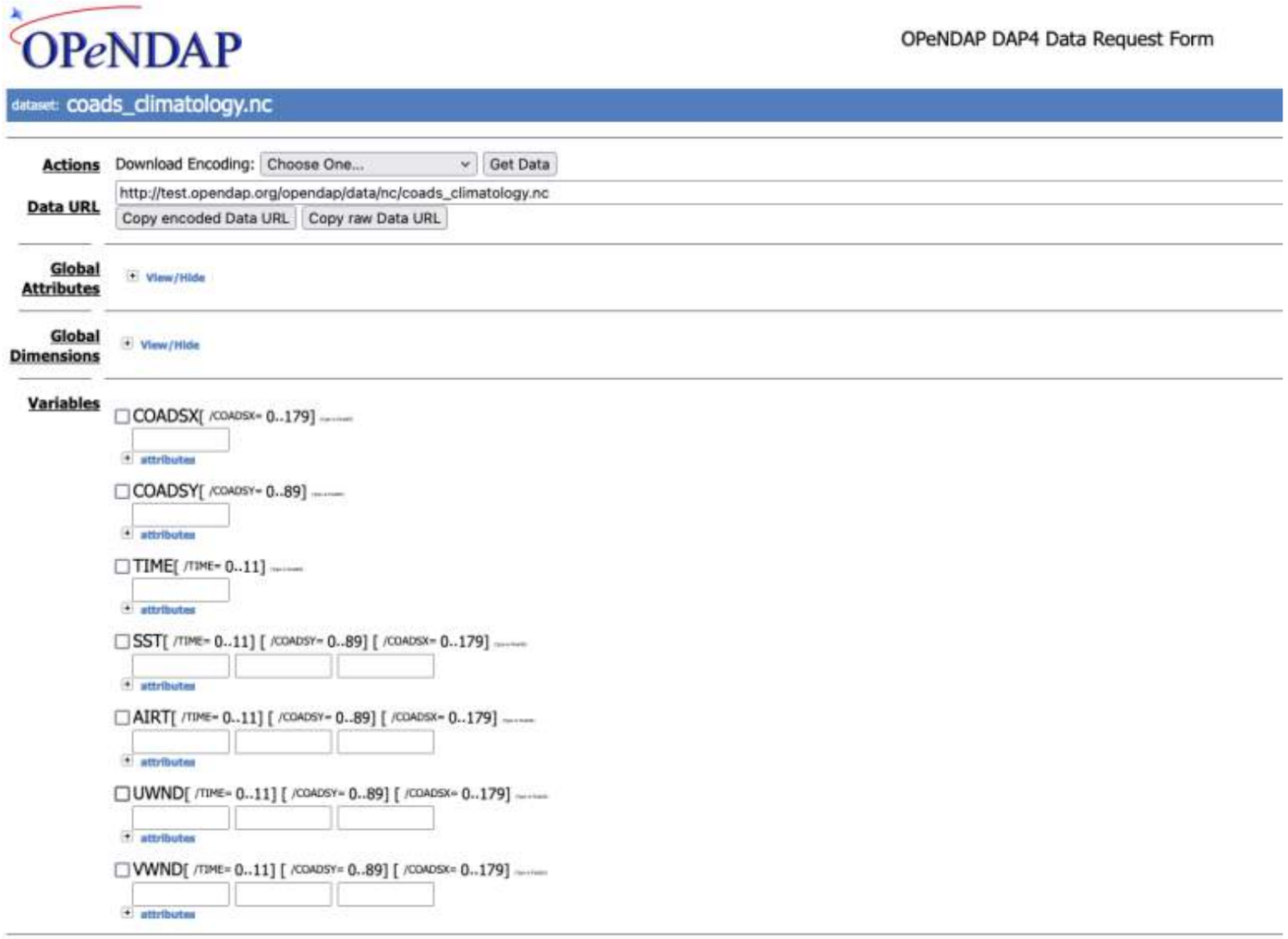
Dataset Service Response

HTML version :
http://test.opendap.org/dap/data/nc/coads_climatology.nc.dsr

XML version :
http://test.opendap.org/dap/data/nc/coads_climatology.nc.dsr.xml

Web Interface

Follow the hyperlink in the image ! ⇒



Constraint Expressions (DAP2 syntax)

Test dataset coads_climatology

- Hyrax interface : http://test.opendap.org/dap/data/nc/coads_climatology.nc.dmr
- Full dataset : http://test.opendap.org/dap/data/nc/coads_climatology.nc
- DDS : .../coads_climatology.nc.dds
- DAS : .../coads_climatology.nc.das

- Request by index : .../coads_climatology.nc.ascii?SST[1][9][178]
- Request by range : .../coads_climatology.nc.ascii?SST[0:1:11][0:1:89][0:1:179]
- Request with coordinates : .../coads_climatology.nc.ascii?COADSX[1],COADSY[2],TIME[3],SST[3][1][2]
- Request with conditions : .../coads_climatology.nc.ascii?SST&TIME<5
- Request with a function : http://test.opendap.org/dap/data/nc/coads_climatology.nc.ascii?geogrid(SST…)
  - ***geogrid*:** Subset applicable DAP Grids using latitude and longitude values.
  - ***grid*:** Subset any DAP Grid object using the values of it's map vectors.
  - ***linear_scale*:** Apply a linear equation to the data returned, including automatic use of CF attributes.
  - ***version*:** The version function provides a list of the server-side processing functions available.

To know more : OPeNDAP User Guide - Constraint Expressions

Usage in Coding

bash and nccopy :

```
test_d4_url="dap4://test.opendap.org/opendap/data/nc/coads_climatology.nc"
nccopy -4 ${test_d4_url} test_coads.nc4
```

python :
(pydap as a dependancy)

```
url = "http://test.opendap.org:80/opendap/data/nc/20070917-MODIS_A-JPL-L2P-A2007260000000.L2_LAC_GHRSST-v01.nc"
import netCDF4 as nc
ds1 = nc.Dataset(url)
import xarray as xr
ds2 = xr.open_dataset(url)
```

See : OPeNDAP Data Access Client Tutorials

Other OPeNDAP Clients

Data-analysis or data-visualization tools such as :

- MATLAB, R, Jupyter Notebooks
- ArcGIS
- Panoply
- GrADS
- Octave
- Ferret
- NCO

See : Interoperability

Thredds - Overview

Tools implementing DAP

- integrated OPeNDAP DAP2 server
- integrated and enriched WMS server and GOdiva browser
- integrated WCS server
- integrated NCSS server
- Filesystem-like web-interface
- Instantaneous startup and availability

Thredds Data Server (TDS) can be seen as a bundle of multiple geospatial tools.

OPeNDAP and Unidata (org behind TDS) are working together for standards.

Thredds - Interface

Tools implementing DAP

Ifremer **Catalog https://tds0.ifremer.fr/thredds/LPO.html**

Dataset

- LPO
  - Climatology ISAS13 (reference period 2002-2012)/
  - Climatology ISAS13 MLD (reference period 2002-2012)/
  - ISAS13 Monthly fields 2002-2012/
  - ISAS13 Monthly fields MLD 2002-2012/
  - Climatology ISAS15 (reference period 2002-2015)/
  - ISAS15 Monthly fields 2002-2015/
  - Climatology ISAS17 (reference period 2002-2017)/
  - ISAS17 Monthly fields 2002-2017/
  - ISAS20 ARGO Monthly fields 2002-2020/
  - ARROW/
  - GLOBMULTI ERA5 GLOBCUR/

Ifremer Thredds Data Server at Ifremer see Info Documentation

Ifremer Ifremer Thredds Data Server

THREDDS Data Server

Catalog https://tds0.ifremer.fr/thredds/LPO_GLOBANA_ISAS13_MNTH/LPO_GLO

Dataset: LPO_GLOBANA_ISAS13_MNTH-MLD_TIME_SERIE

- *Data format:* netCDF
- *Data type:* GRID
- *Harvest:* true
- *ID:* LPO_GLOBANA_ISAS13_MNTH-MLD_TIME_SERIE

Documentation:

- **summary:** LPO_GLOBANA_ISAS13_MNTH-MLD_TIME_SERIE
- http://www.ifremer.fr/lpo/SO-Argo/Products/Global-Ocean-T-S

Access:

1. **OPENDAP:** /thredds/dodsC/LPO_GLOBANA_ISAS13_MNTH-MLD_TIME_SERIE
2. **WMS:** /thredds/wms/LPO_GLOBANA_ISAS13_MNTH-MLD_TIME_SERIE
3. **WCS:** /thredds/wcs/LPO_GLOBANA_ISAS13_MNTH-MLD_TIME_SERIE

Keywords:

- sea_surface_temperature **(CF-1.0)**
- sea_water_salinity **(CF-1.0)**

Variables:

- *Vocabulary* [CF-1.0]:
  - **MLDS_TEMP (degree_C)** = = sea_surface_temperature
  - **MLDS_PSAL (PSU)** = = sea_water_salinity
  - **MLDT_TEMP (degree_C)** = = sea_surface_temperature
  - **MLDT_PSAL (PSU)** = = sea_water_salinity

GeospatialCoverage:

- *Longitude:* -180.0 to 179.5 Resolution=0.5 degrees_east
- *Latitude:* -77.01 to 89.89999999999999 Resolution=0.5 degrees_north
- *Altitude:* 0.0 to 2000.0 meters (positive is **down**)

TimeCoverage:

- *Start:* 2002-01-15T00:00:00Z
- *End:* 2012-12-15T00:00:00Z

Properties:

- aggregation-level = "view"
- update-schedule = "notPlanned"

Viewers:

- Godiva2 (browser-based)
- NetCDF-Java ToolsUI (webstart)
- Integrated Data Viewer (IDV) (webstart)

Thredds - Drawbacks

Tools implementing DAP

- Steep learning curve for configuring dataset

- Issues at Ifremer with historical configurations when upgrading to version 5 (version 4 currently at Ifremer)
  - Is this issue occurring on Hyrax too ?

- Not suitable for datasets with very large number of files out of the box => need for specific tooling
  - what is a very large number of files ? 1.483.232 for OSI-204-b-metop_b
  - At Ifremer, we use a custom tool tdsConfigGen developed by ACTIMAR. One more level of complexity and maintenance.

- Inventory of products difficult to maintain (what is a dataset ?)

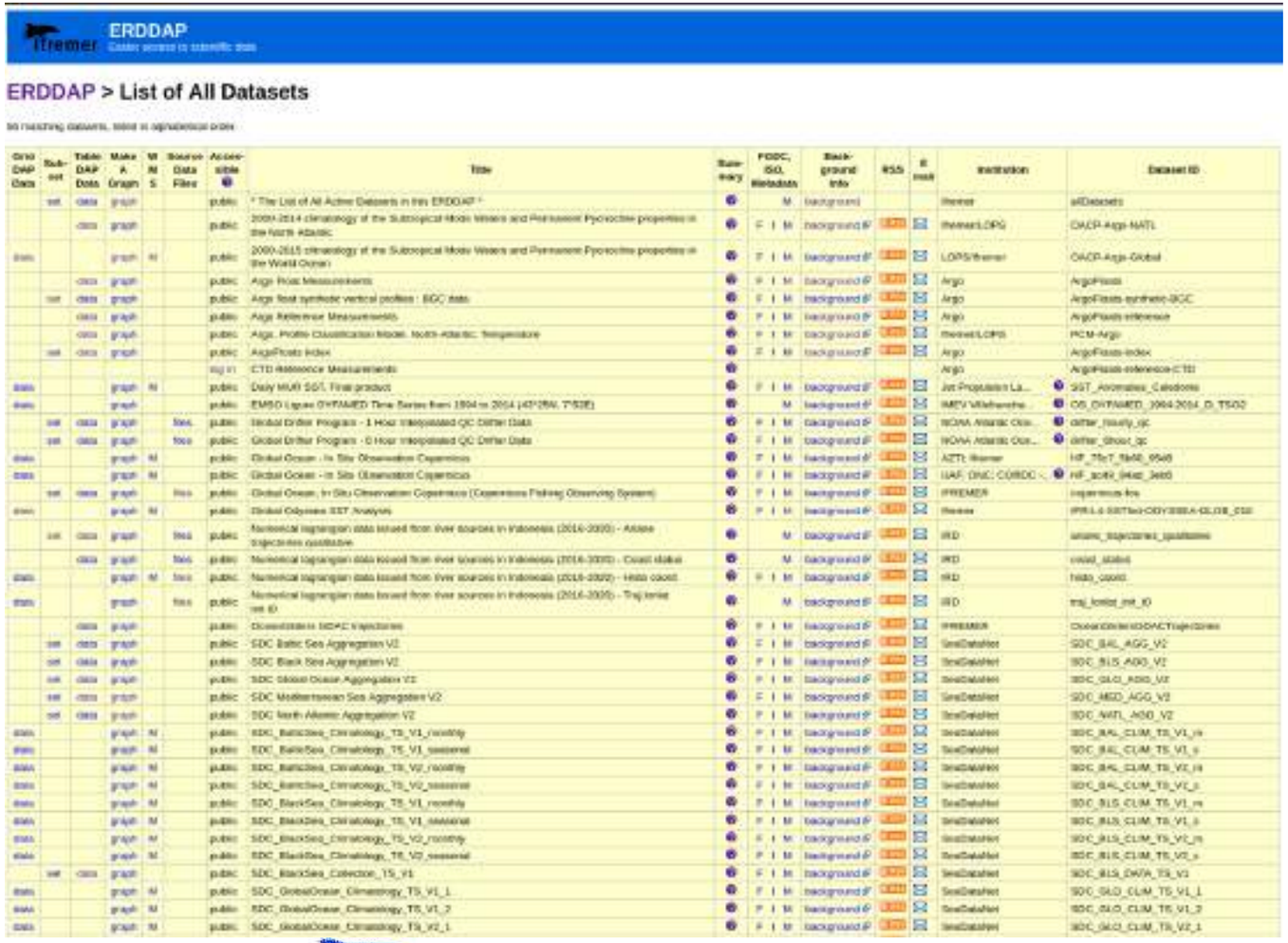
Erddap - Overview

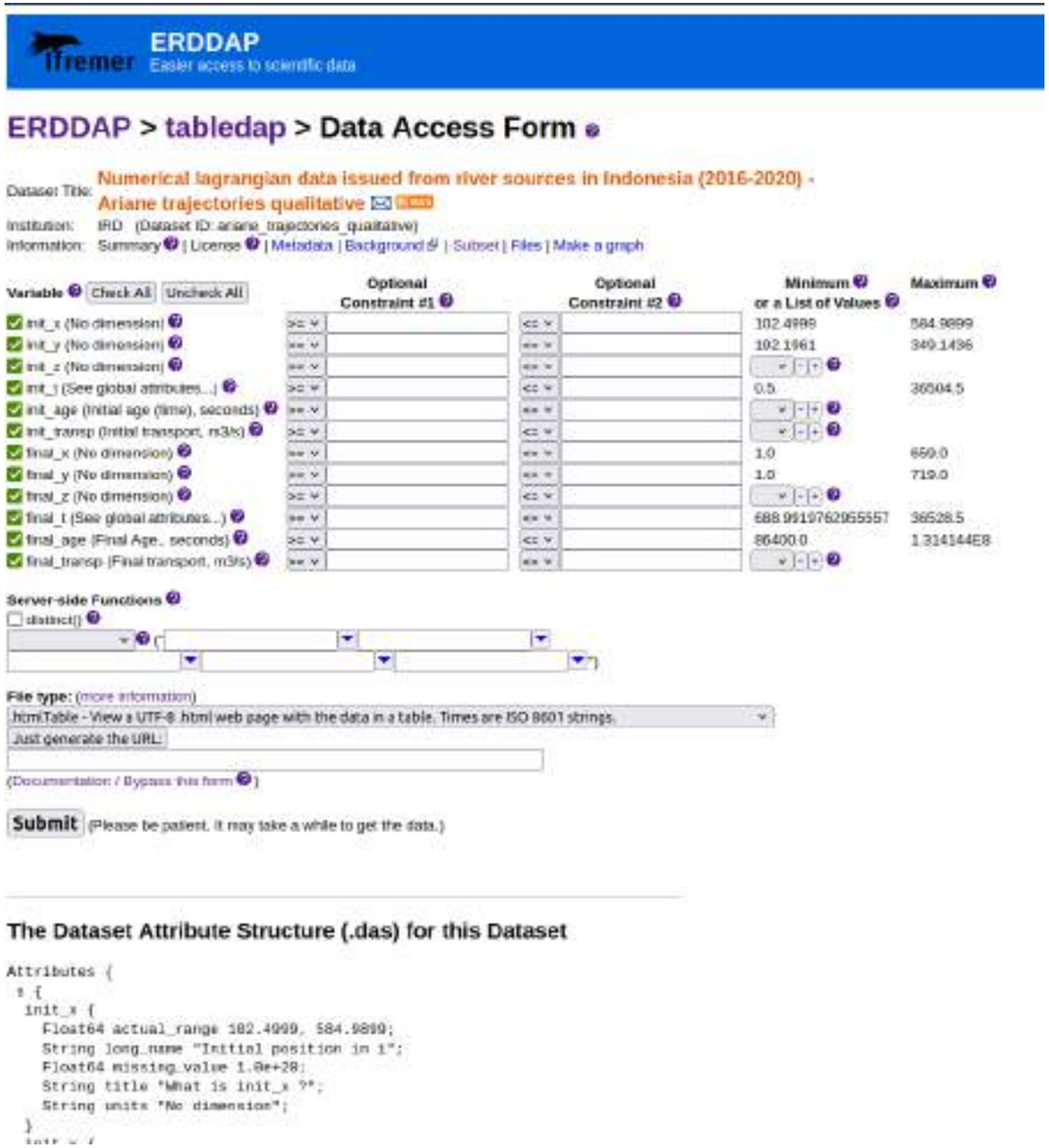
Tools implementing DAP

- Specific implementation of DAP
  - griddap for gridded data
  - tabledap for tabular data
- Rich web-interface, enabling easier DAP request generation
- Basic WMS
- Interface as a list of datasets

Erddap - Interface

Tools implementing DAP





Erddap - Drawbacks

Tools implementing DAP

- Steep learning curve for configuring data
- Aggregation-oriented : no directory scan producing multiple datasets
- Strong cohesion required between files in dataset for configuration
- Can hold datasets with large number of files but not very large (Argo : 18701, test with OSI-204-b-metop_b and its 1.483.232 files didn’t succeed)
- Long startup - need to scan all files in dataset before dataset becomes available. Longer (>12h) if there are large datasets.
- Multiple caveats about Erddap development :
  - after a reboot, need to load web-interface at least once before it starts to index each dataset
  - some confit are deprecated
  - logs are weirdly generated
  - was developed mainly by one developer Bob Simons

Hyrax - Overview

Tools implementing DAP

- Basic DAP4 server
- filesystem-like interface
- now comes with ncWMS2 server alongside
- read Thredds catalogs
- developed by OPeNDAP, so closest to DAP specifications
- now comes with ncWMS2 server in the same docker image (ncWMS2 dependant on DAP)
- not installed at Ifremer

http://test.opendap.org:8080/opendap/data/contents.html

Hyrax - Interface

Tools implementing DAP

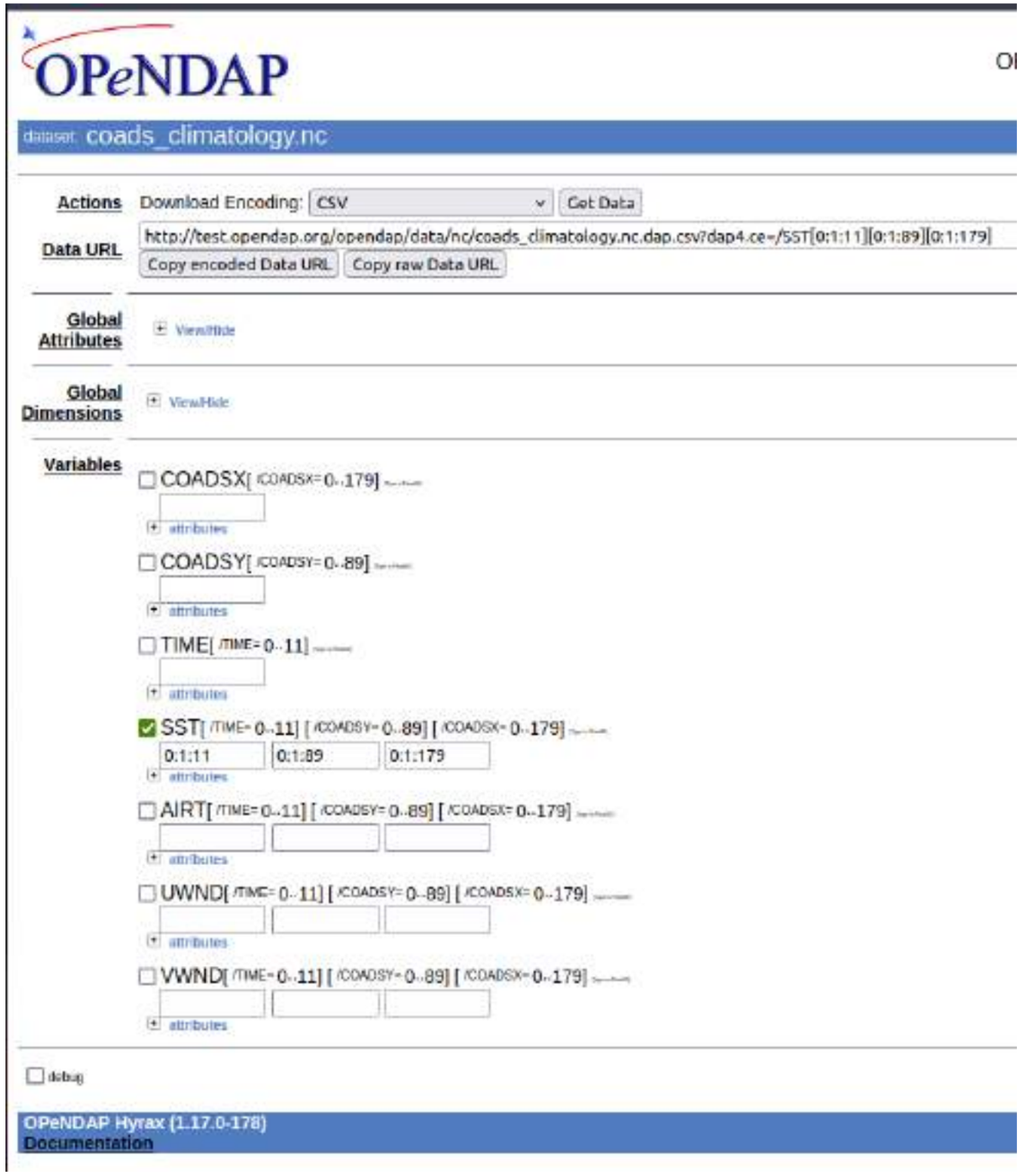
OPeNDAP

Contents of /data/

Name	Last Modified
csv/	2021-12-02T20:21:01GMT
dapreader/	2021-12-02T20:21:01GMT
dmrpp/	2021-12-02T20:21:03GMT
dsp/	2020-12-30T19:57:53GMT
ff/	2021-12-02T20:21:03GMT
gdal/	2021-12-02T20:21:03GMT
hdf4/	2021-12-02T20:21:03GMT
hdf5/	2021-12-02T20:21:03GMT
nc/	2021-12-02T20:21:03GMT
ncml/	2021-12-02T20:21:03GMT
stare/	2021-12-02T20:21:04GMT
ugrids/	2021-12-02T20:21:04GMT

THREDDS Catalog XML

OPeNDAP Hyrax (1.17.0-178)
Documentation



Hyrax - Drawbacks

Tools implementing DAP

- Basic DAP server
- ncWMS2 needs to be configured

http://test.opendap.org:8080/opendap/data/contents.html

DAP Tools Summary

	Erddap	Thredds4	Thredds5	Hyrax
DAP2 (dap, dds, ddx)	✅	✅	✅	✅
DAP4 (dmr, dsr, async)	❌	❌	❌ Not functionnal (see issue and doc)	✅
HTTPServer	✅	✅	✅	✅
/FileAccess	✅	✅	✅	✅
WMS	✅ No transect	✅ ncWMS2 + Godiva	✅ ncWMS2 + Godiva	✅ ncWMS2 + Godiva in same Docker image (configuration needed)
WCS	❌	✅	✅	❓GeoTiff & JPEG services
NCSS*	❌	✅	✅	❌
Static files inventory*	❌	✅	✅	✅
Static files large inventory*	❌	✅	❌ See issue	❓
Lazy dataset loading*	❌	✅	✅	❓
Dataset discovery*	❌	✅	✅	✅
Enhanced data selection interface	✅	❌	❌	❌
Depends on	NOAA Bob Simons	Unidata	Unidata	OPeNDAP

Table footnotes *

- **NCSS :** NetCDF Subset Service

- **Static files inventory :** inventory a list of files to be used by the dataset. Remove the need to scan files at startup. Static files inventory allows for intantaneous availability of the datasets.

- **Static files large inventory :** >1.000.000 files. Maybe it's a good thing that it is blocked.

- **Lazy dataset file scan :** if a file scan is configured, the scan starts only when accessing the dataset url. Otherwise, the server scan each dataset at startup. It is the case for Erddap, meaning it can take more than 12h to start with big datasets.

- **Dataset Discovery :** allows files inside directories to be set as dataset. Erddap only allows aggregated dataset.

Limitations of DAP

- No out-of the box tool for building DAP urls. Building DAP urls is difficult and need to be optimized. Specific set of skills
  - example : argopy library to request Erdapp-hosted ARGO dataset
- no support for Analysis Ready Cloud Optimized (ARCO) formats, columnar formats, that could greatly increase performance
  - parquet, geoparquet, duckDB
- High technical level for maintaining datasets
- Some compatibility issues with netcdf types, see post
- Wall of large datasets => see next slide
- Single server architecture is a limit to serve large datasets
- User feedback ?

The wall of large datasets

Handling large datasets is difficult ! Dataset with huge amount of data or huge number of files

For Thredds and Erddap :

- No workaround with large request : it means reading multiple files.
- no distributed architecture means request depends on 1 server only.
- Thredds v4 with workaround has a higher limit of files than Erddap.

Small files is an issue for every information system :

- holds multiple duplicated metadata
- heavy on the file system
- bad for compression
- process of opening a file is time-consuming

Importance of preprocessing :

- can drastically reduce disk usage
- increase scanning and reading speed
- caveats when dataset is updated, need to find a balance between data freshness and performance

Comparison with alternatives

Alternatives to OPeNDAP :

- NCSS : NetCDF Subset Service (specific to Thredds)
- HTTPServer
- SOS / NCSOS OGC Sensor Observation Service ?
- STAC + S3 ?
- Hyrax in the Cloud/S3/DMR++ (video)

OGC API : Geomatic field

OPeNDAP : oceanographic, satellite fields