



# Serveurs OPeNDAP

---

Hyrax, TDS, ERDAP, *PyDAP*, ~~Dapper~~, ~~GDS~~



# Introduction

Projets: sponsors, équipes, activités

Architecture logicielle

Données lues

Interfaces (WMS, WCS, etc.)

Type de données

URL

Spécificités du serveur Hyrax



# Projets – TDS (Thredds)

UCAR/UniData

10 782 commits, 37 contributeurs

JAVA



# Projets - ERDAP

Bob Simons

36 commits, 2 contributeurs

JAVA



# Projets - PyDAP

PyDAP contributeurs

568 commits, 17 contributeurs

Peu d'activité sur la partie serveur, essentiellement utilisée pour les tests de la partie « cliente ».



# Projets – Hyrax (libdap4, BES, OLFS)

OPeNDAP (NOAA, NASA, NSF, Australian Bureau of Meteorology)

Libdap4: 4089 commits, 12 contributeurs

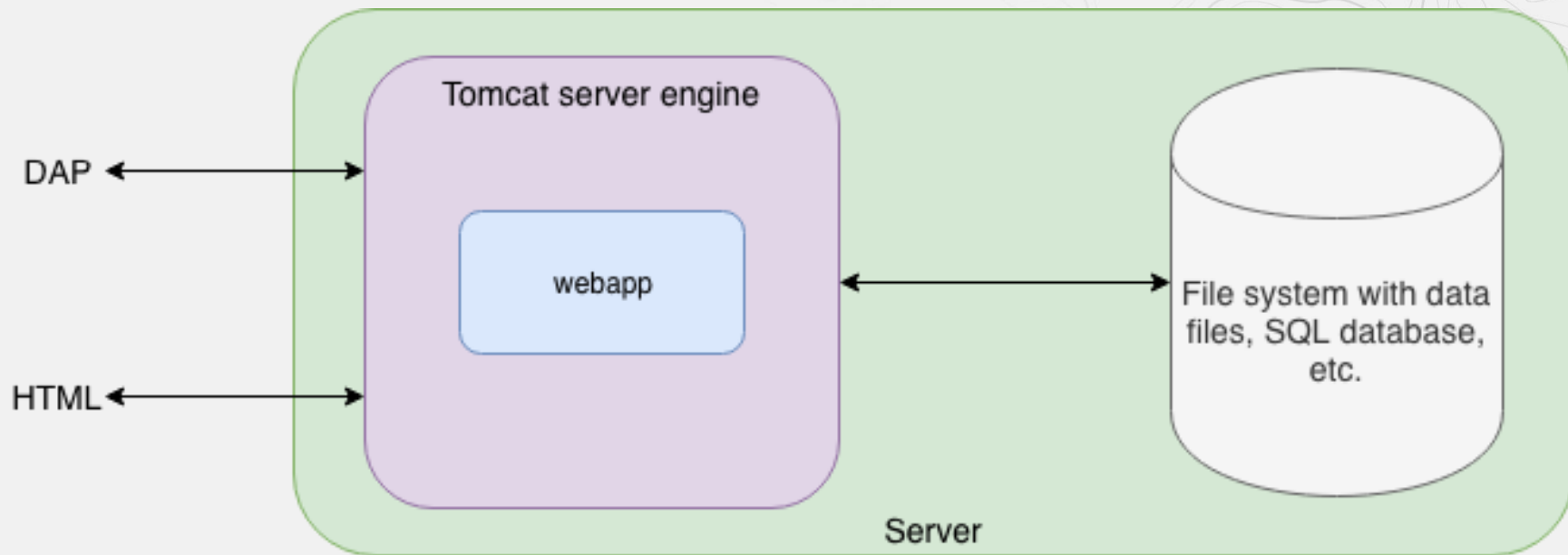
BES: 4923 commits, 14 contributeurs

OLFS: 2858 commits, 6 contributeurs

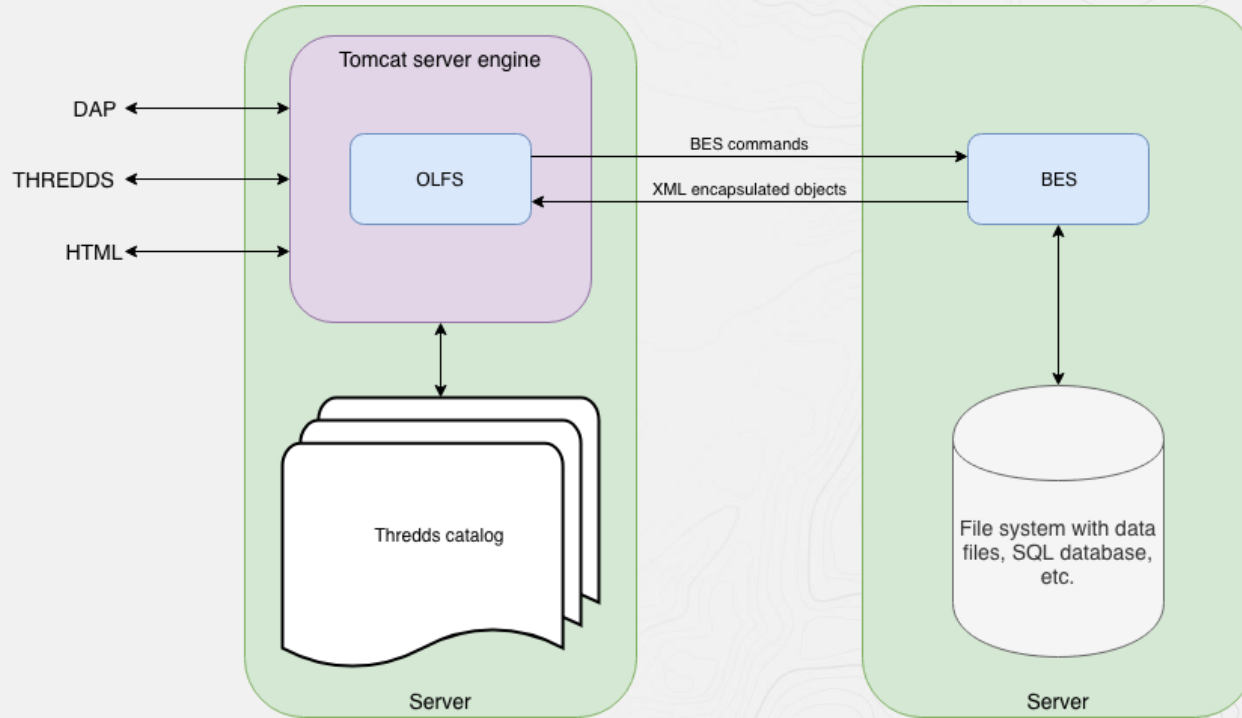
C++, JAVA



# Architecture – Erdap [PyDAP]

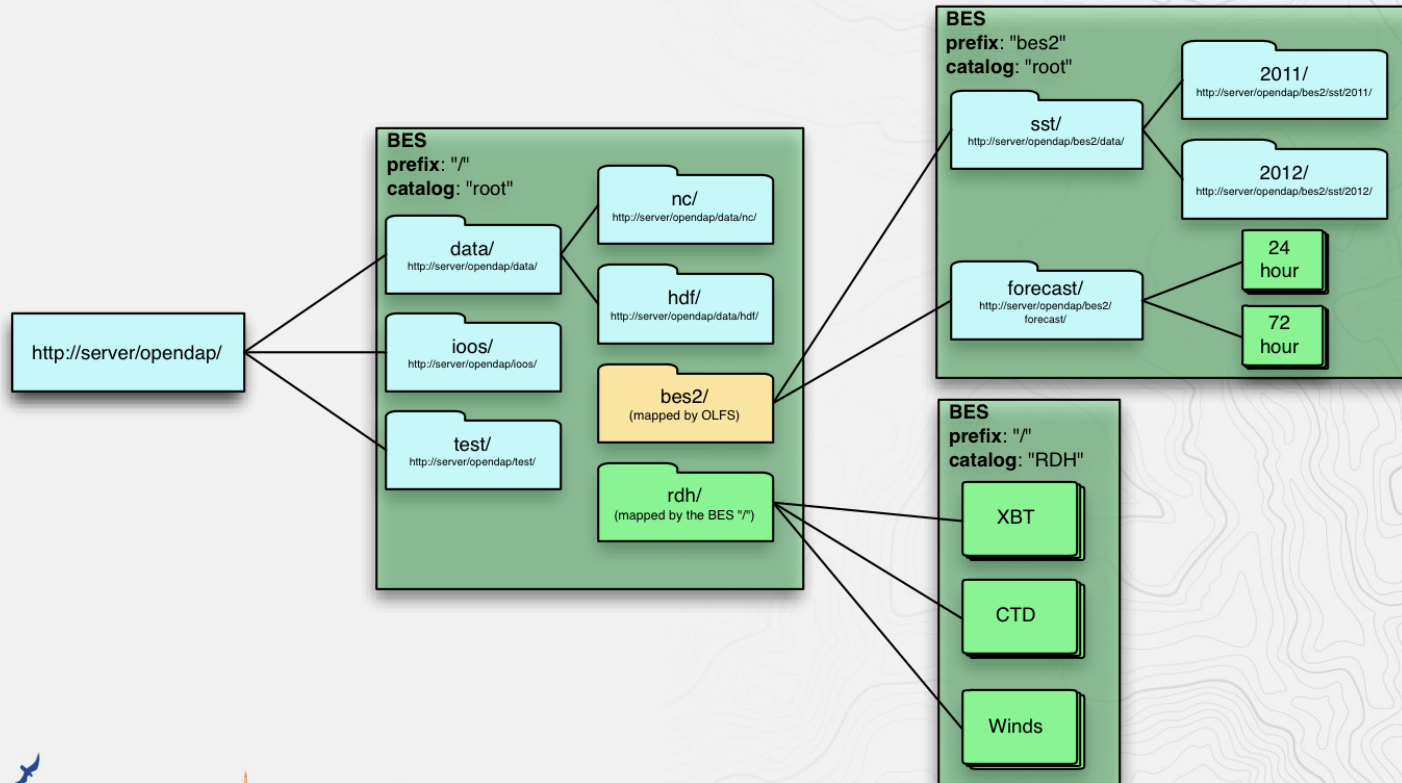


# Architecture - Hyrax





# Architecture - Hyrax



# Données lues - ERDAP

NetCDF-3, NetCDF-4

HDF-4, HDF-5

GRIB

NCML

CSV

SQL (JDBC)

Gateway



# Données lues - TDS

NetCDF-3, NetCDF-4

HDF4, HDF5

BUFR

NCML

Gateway



# Données lues - Hyrax

NetCDF-3, NetCDF-4

HDF-4, HDF-5

NCML

CSV

FreeForm

UGRID (unstructured mesh)

GDAL (GRIB, GeoTIFF, JPEG200)

SQL (ODBC)

Gateway

Extensible...



# Interfaces

## WMS (produits grillés)

- ERDAP (natif)
- Hyrax & TDS : ncWMS - <https://github.com/Reading-eScience-Centre/ncwms>

## WCS

- Hyrax & TDS

## TREDDS catalog

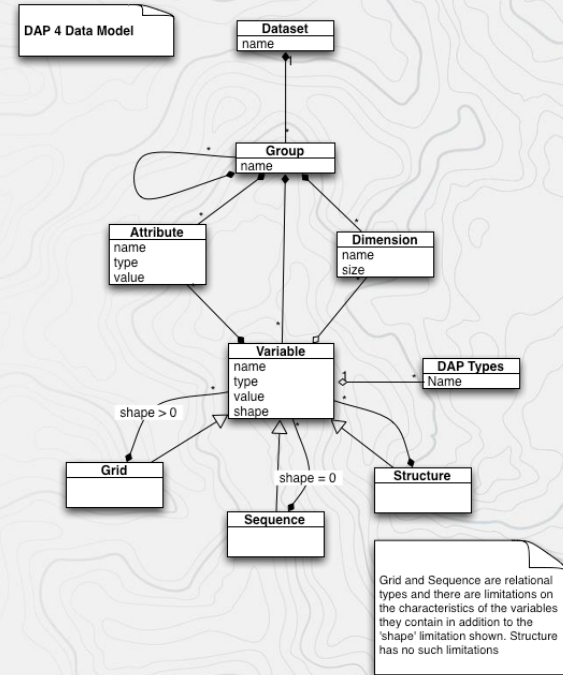
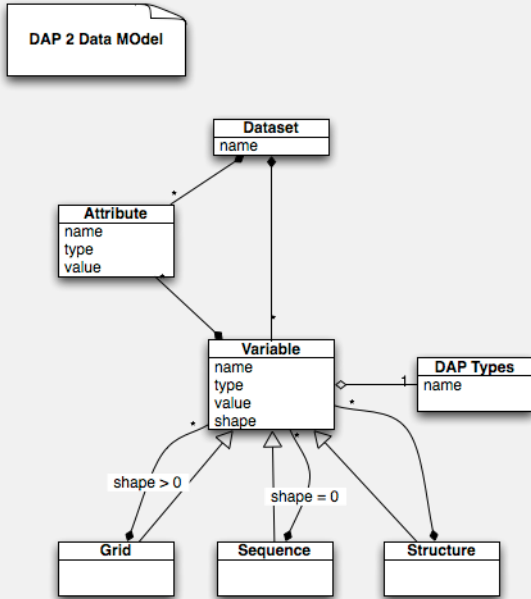
- Hyrax & TDS

## W10n(JSON)

- Hyrax



# Modèle de données



# Données - Tableau

```
Float64 data[height = 5][width = 6];
```

Un tableau est une structure de données indexée unidimensionnelle telle que définie par ANSI C. Les tableaux multidimensionnels sont définis comme des tableaux de tableaux. Un tableau peut être sous-échantillonné en utilisant des indices ou des plages d'indices entre crochets.

Par exemple, `temp[3][4]` indique la valeur dans la quatrième ligne et la cinquième colonne du tableau `temp`. (Comme en C, les index des tableaux OPeNDAP commencent à zéro.)

Un morceau d'un tableau peut être spécifié avec des plages d'indice ; le tableau `temp[2:10][3:4]` sélectionne les indices [2,3,.., 10][3,4] du tableau original `temp`.

Un hyperslab définit une plage d'indice et un pas. Le tableau représenté par `temp[2:2:10][3:4]` n'aurait que cinq lignes ; la valeur du milieu dans la première plage d'indice indique le pas à utiliser pour aller des indices 2 à 10. l'hyperslab [2:3:10] sélectionne les indices 2, 5, 8.



# Données - Structure

```
Structure {  
  Int32 month;  
    Float64 data[5][6];  
} measurement;
```

Une structure est une classe qui peut contenir plusieurs variables de différentes classes. Cependant, bien qu'il implique que ses variables membres sont liées d'une manière ou d'une autre, il ne transmet aucune information relationnelle entre ses membres.

Le type de structure peut également être utilisé pour regrouper un ensemble de variables non apparentées dans un seul ensemble de données.

Le nom de classe "dataset" est synonyme de structure.





# Données - Séquence

```
Sequence {  
    Int32 month;  
    Float64 data[5][6];  
} measurement;
```

$S_{00}$	$S_{01}$	...	$S_{0n}$
$S_{10}$	$S_{11}$	...	$S_{1n}$
$S_{20}$	$S_{21}$	...	$S_{2n}$
.	...	...	.
.	...	...	.
.	...	...	.
$S_{i0}$	$S_{i1}$	...	$S_{in}$

Une séquence est un ensemble ordonné de variables dont chacune peut avoir plusieurs valeurs.

Les variables peuvent être de classes différentes.

Chaque élément d'une Séquence est constitué d'une valeur pour chaque variable membre, donc une Séquence est une sorte d'ensemble ordonné de Structures.



# Données - Grille

```
Grid {  
    Float64 data[lon = 6][lat = 5];  
    Float64 lon[6];  
    Float64 lat[5];  
} measurement;
```

Une grille est une association d'un tableau de N dimensions avec N vecteurs nommés (tableaux unidimensionnels), dont chacun a le même nombre d'éléments que la dimension correspondante du tableau.

Chaque valeur de données dans la grille est associée aux valeurs de données dans les vecteurs associés à ses dimensions ou axes (au sens mathématique du terme).



# Type de données

Type	Intervalle	Protocole
Int8	$[-(2^7), (2^7) - 1]$	DAP4
UInt8	$[0, (2^8) - 1]$	DAP4
Byte	$[0, (2^8) - 1]$	DAP2/DAP4
Char	$[0, (2^8) - 1]$	DAP2/DAP4
Int16	$[-(2^{15}), (2^{15}) - 1]$	DAP2/DAP4
UInt16	$[0, (2^{16}) - 1]$	DAP2/DAP4
Int32	$[-(2^{31}), (2^{31}) - 1]$	DAP2/DAP4
UInt32	$[0, (2^{32}) - 1]$	DAP2/DAP4
Int64	$[-(2^{63}), (2^{63}) - 1]$	DAP4
UInt64	$[0, (2^{64}) - 1]$	DAP4

Type	Intervalle	Protocole
Float32	IEEE	DAP2/DAP4
Float64	IEEE	DAP2/DAP4
String	UTF-8	DAP2/DAP4
URI	RFC 2396	DAP2/DAP4
Opaque		DAP4
Enum		DAP4

~~Date/Time~~



# URLS - Grilles

[http://opendap.cr.usgs.gov/opendap/MOD09Q1.005/h11v04.ncml.json?sur\\_refl\\_b01\[683:1:683\]\[2000:1:2010\]\[2000:1:2010\]](http://opendap.cr.usgs.gov/opendap/MOD09Q1.005/h11v04.ncml.json?sur_refl_b01[683:1:683][2000:1:2010][2000:1:2010])

Nom du produit

Nom de la variable

Temps

Emplacement des tuiles (Emplacements des tuiles (H & V) (les données sont agrégées)

Position des pixels (axes Y et X)



# URLS - Séquences

[http://test.opendap.org/dap/data/ff/gsodock.dat.ascii?URI\\_GSO-Dock.Salinity&URI\\_GSO-Dock.Time>35234.5](http://test.opendap.org/dap/data/ff/gsodock.dat.ascii?URI_GSO-Dock.Salinity&URI_GSO-Dock.Time>35234.5)

```
Dataset {  
    Sequence {  
        Float64 Time;  
        Float32 Depth;  
        Float32 Sea_Temp;  
        Float32 Salinity;  
        ...  
    } URI_GSO-Dock;  
} gsodock.dat;
```



# Clients OPeNDAP

NCL - Scientific data analysis and visualization software

CDAT - Climate Data Analysis Tools

OPeNDAP geturl

Matlab & OPeNDAP Matlab Toolkit

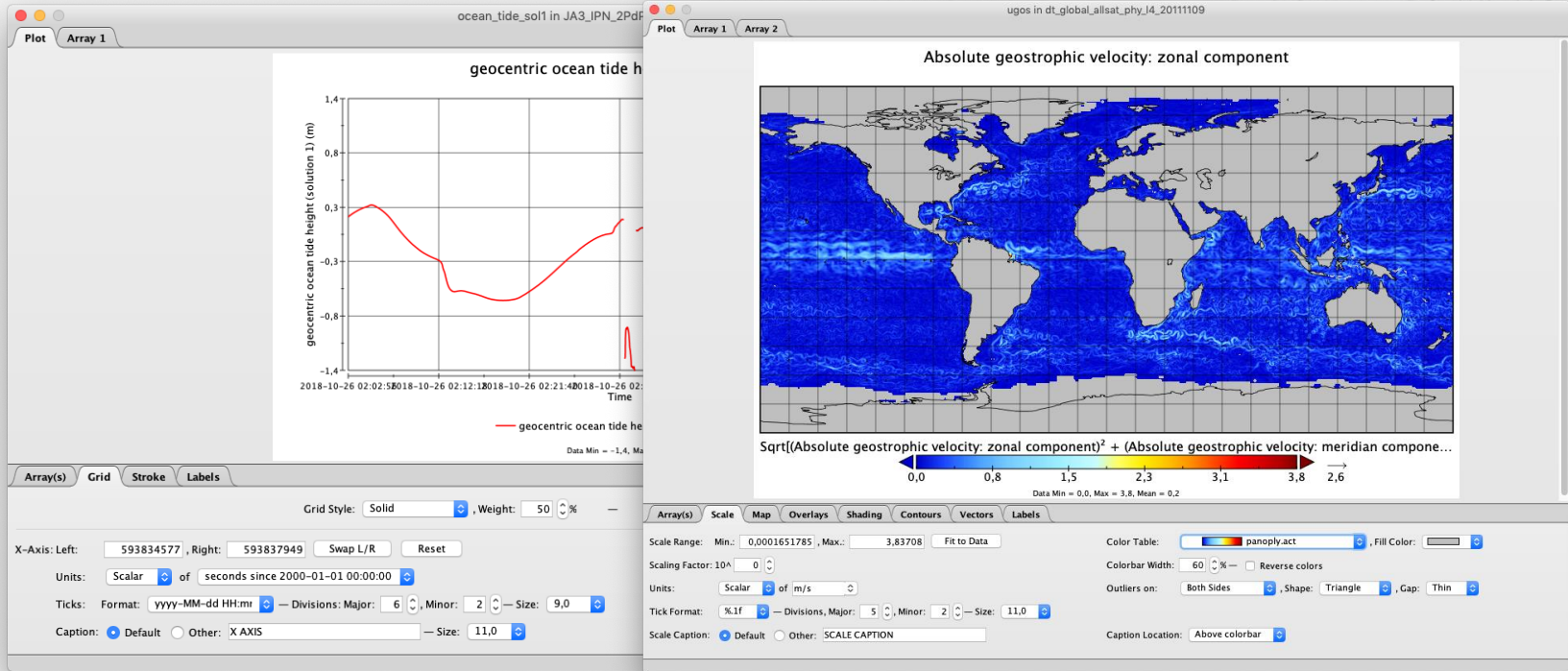
NetCDF

Panoply

Etc...



# Clients OPeNDAP



# Hyrax - ASCII [CSV]

Station<String>	latitude<Float32>	longitude<Float32>	temperature_K<Float32>	Notes<String>
CMWM	-34.7	23.7	264.3	
BWWJ	-34.2	21.5	262.1	Foo
CWQK	-32.7	22.3	268.4	
CRLM	-33.8	22.1	270.2	Blah
FOOB	-32.9	23.4	269.69	FOOBAR





# Hyrax - ASCII [FreeForm]

Latitude and Longitude: -63.223548 54.118314 -176.161101 149.408117

-47.303545 -176.161101 11.7125 34.4634

-25.928001 -0.777265 20.7288 35.8953

-28.286662 35.591879 23.6377 35.3314

12.588231 149.408117 28.6583 34.5260

-63.223548 55.319598 0.4503 33.8830

54.118314 -136.940570 10.4085 32.0661

-38.818812 91.411330 13.9978 35.0173

-34.577065 30.172129 20.9096 35.4705

27.331551 -155.233735 23.0917 35.2694

11.624981 -113.660611 27.5036 33.7004



# Hyrax - ASCII [FreeForm]

```
ASCII_file_header "Latitude/Longitude Limits"
```

```
minmax_title 1 24 char 0
```

```
latitude_min 25 36 double 6
```

```
latitude_max 37 46 double 6
```

```
longitude_min 47 59 double 6
```

```
longitude_max 60 70 double 6
```

```
ASCII_data "lat/lon"
```

```
latitude 1 10 double 6
```

```
longitude 12 22 double 6
```

```
temp 24 30 double 4
```

```
salt 32 38 double 4
```

```
...
```

```
...
```

```
ASCII_output_data "output"
```

```
latitude 1 10 double 3
```

```
longitude_deg 11 15 short 0
```

```
longitude_min 16 19 short 0
```

```
longitude_sec 20 23 short 0
```

```
salt 31 40 double 2
```

```
temp 41 50 double 2
```



# Hyrax - Fonctions serveur

## geogrid

- `geogrid(grid variable, top, left, bottom, right[, expression ...])`

## grid

- `grid(sst,"20>TIME","TIME>=10")`

## linear\_scale

- `linear_scale(var,scale_factor,add_offset,missing_value)`

## bbox

- `bbox(array, min-value, max-value)`

## bbox\_union

- `bbox_union(bbox(a1, min-value-1, max-value-1), bbox(a2, min-2, max-2), ..., "union" | "intersection")`



# Hyrax - Agrégation

Il est souvent souhaitable de traiter une collection de fichiers de données comme s'il s'agissait d'un seul ensemble de données.

Hyrax offre deux façons différentes de le faire :

- il permet aux fournisseurs de données de définir des agrégations de fichiers de manière à ce qu'ils apparaissent comme un ensemble de données unique et
- il permet aux utilisateurs d'envoyer au serveur une requête sur une liste de fichiers et de recevoir une réponse unique.



# Hyrax – Agrégation (serveur)

```
<?xml version="1.0" encoding="UTF-8"?>
<netcdf title="Test of joinNew aggregation using the scan element and dateFormatMark">

  <aggregation type="joinNew" dimName="fileTime">
    <variableAgg name="CGusfc"/>
    <scan location="data/ncml/agg/dated"
      suffix=".nc"
      subdirs="false"
      regExp="^.*/CG[^/]*"
      dateFormatMark="CG#yyyyDDD_HHmms"
    />
  </aggregation>

</netcdf>
```



# Hyrax – Agrégation (client)

```
$ more short_names/d1_netcdf3_variable_subset.txt
&operation=netcdf3
&var=Latitude,Longitude,Optical_Depth_Land_And_Ocean
&bbox="[49,Longitude,50][167,Longitude,170]"
&file=/data/modis/MOD04_L2.A2015021.0020.051.NRT.hdf
&file=/data/modis/MOD04_L2.A2015021.0025.051.NRT.hdf
&file=/data/modis/MOD04_L2.A2015021.0030.051.NRT.hdf
```

```
$ curl -X POST -d @short_names/d1_netcdf3_variable_subset.txt
http://test.opendap.org/opendap/aggregation > d1.zip
```

```
$ unzip -t d1.zip
```

```
Archive:  d1.zip
```

```
testing: MOD04_L2.A2015021.0020.051.NRT.hdf.nc  OK
testing: MOD04_L2.A2015021.0025.051.NRT.hdf.nc  OK
testing: MOD04_L2.A2015021.0030.051.NRT.hdf.nc  OK
```



# Conclusion

Possibilité de séparer le trafic internet des données. Cette fonctionnalité garantit qu'avec un pare-feu correctement configuré, aucun trafic du port 80 n'atteint les machines avec les données diffusées.

Architecture évolutive : plusieurs clients BES pour un même serveur OLFS

Améliorations des agrégations basées sur "NcML" et des agrégations construites à partir des requêtes, où les clients fournissent la liste des fichiers ou ensembles de données à traiter.

Prise en charge de traitements côté serveur, y compris les sous-échantillonnages de grille non structurés

Prise en charge d'un grand nombre de sources de données

Support du protocole DAP4 (modèle de données HDF5)

