Marinet2 Bring Your Own Data (BYOD) Workshop
23rd June 2020

MaRINET2

Data access through Virtual Research Environments

Antoine QUÉRIC,

IFREMER

@MaRINET2_EU

Data is :

- converted to an inter-operable format (netCDF),
- compliant with Marinet2 data standards

How should you access to the data from the VRE or your desktop ?

What are we going to talk about ?

Jupyter notebooks / lab : interact with your python code on the go, in your web browser and easily share youur code

Xarray : convenient python module to handle multidimensional data

Intake : virtualize the access to your data for your users

Opening files with Python

Data is stored on your datacenter.

Open one file **or** a collection with **xarray**

Import python modules

In [1]:
```
import warnings
warnings.filterwarnings('ignore')
import xarray as xr
import glob
```

Locate the files on your datacenter

Here we only show some of them, the collection is bigger.

In [2]:
```
files_dir = "/home/datawork-e-marinet/intranet/data/WRRT/2019_Ifremer/Measurements_convertis_CG/*.nc"
glob.glob(files_dir)[0:2]
```

Out[2]:
```
['/home/datawork-e-marinet/intranet/data/WRRT/2019_Ifremer/Measurements_convertis_CG/RRWind_040.nc',
 '/home/datawork-e-marinet/intranet/data/WRRT/2019_Ifremer/Measurements_convertis_CG/RRWind_068.nc']
```

Open the collection of files with xarray

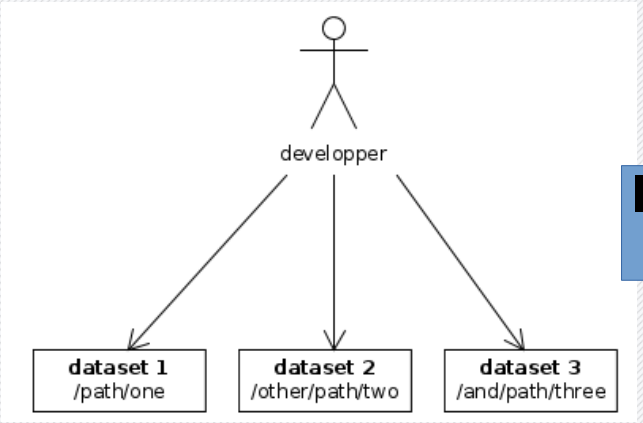
In [3]:
```
ds = xr.open_mfdataset(glob.glob(files_dir))
ds.sortby("TIME")
```

Out[3]:
```
<xarray.Dataset>
Dimensions:          (DEPTH: 1, TIME: 5043097)
Coordinates:
  * TIME             (TIME) datetime64[ns] 2019-06-04T11:26:55 ... 2019-06-18T16:11:14.849999872
Dimensions without coordinates: DEPTH
Data variables:
    DEPH             (TIME, DEPTH) float32 dask.array<shape=(5043097, 1), chunksize=(49261, 1)>
    M1               (TIME, DEPTH) float32 dask.array<shape=(5043097, 1), chunksize=(49261, 1)>
    M2               (TIME, DEPTH) float32 dask.array<shape=(5043097, 1), chunksize=(49261, 1)>
```

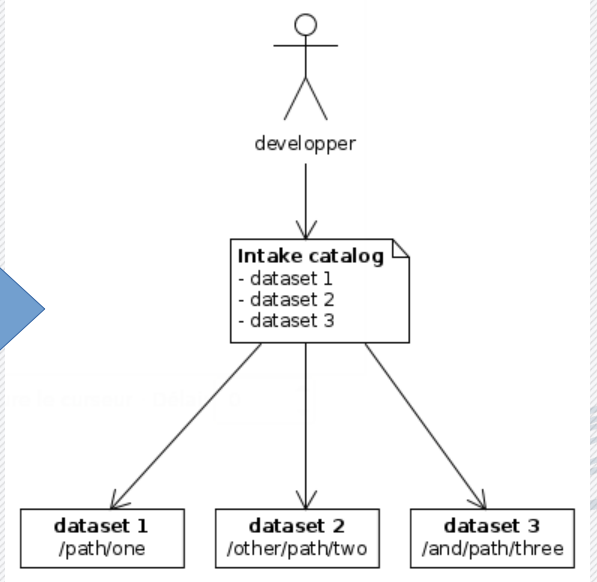
Where are my files, again ?

Handling data in different directories can become painful.

Users do not always know how to load it.



Intake simplifies data access



Intake catalogs in a nutshell

```yaml
plugins:
    source:
        - module: intake-xarray
sources:
    WRRT:
        description: Marinet2 WRRT converted to netCDF
        driver: netcdf
        args:
            urlpath: '/home/datawork-e-marinet/intranet/data/WRRT/2019_Ifremer/Measurements_convertis_CG/*.nc'
            concat_dim: TIME
```

Path to your data

Data loading instructions

Intake catalogs in a nutshell

Import python modules

In [1]:
```
import warnings
warnings.filterwarnings('ignore')
import intake
import intake_xarray
```

Open the catalog :

In [2]:
```
catalog = intake.open_catalog("marinet2_catalog.yaml")
list(catalog)
```

Out[2]:
```
['WRRT']
```

Open the collection of files with xarray

In [3]:
```
ds = catalog.WRRT.to_dask()
ds
```

Out[3]:

xarray.Dataset

- Dimensions: (DEPTH: 1, **TIME**: 5043097)
- Coordinates:

TIME	(TIME)	datetime64[ns]	2019-06-04T11:26:55 ... 2019-06-18T16:11:...

- Data variables:

DEPH	(TIME, DEPTH)	float32	dask.array<chunksize=(49261, 1), meta=np....
M1	(TIME, DEPTH)	float32	dask.array<chunksize=(49261, 1), meta=np....
M2	(TIME, DEPTH)	float32	dask.array<chunksize=(49261, 1), meta=np....

Dask is just an interface that helps user when it comes to parallel computation

Easy data handling with xarray

Create a subset of your dataset

In [13]:
```
ds = ds.squeeze().sortby("TIME") # Removing DEPTH dimension which is not necessary here
subset = ds.where(
    (ds.TIME >= np.datetime64("2019-06-04T11:50:00")) &
    (ds.TIME <= np.datetime64("2019-06-04T12:00:00")),
    drop=True
)
subset
```

Out[13]: xarray.Dataset

- Dimensions: (**TIME**: 26481)

▾ Coordinates:

TIME	(TIME)	datetime64[ns]	2019-06-04T11:50:00 ... 2019-06-04T12:00:00

```
array(['2019-06-04T11:50:00.000000000', '2019-06-04T11:50:00.009999872',
       '2019-06-04T11:50:00.020000000', ..., '2019-06-04T11:59:59.980000000',
       '2019-06-04T11:59:59.990000128', '2019-06-04T12:00:00.000000000'],
      dtype='datetime64[ns]')
```

▾ Data variables:

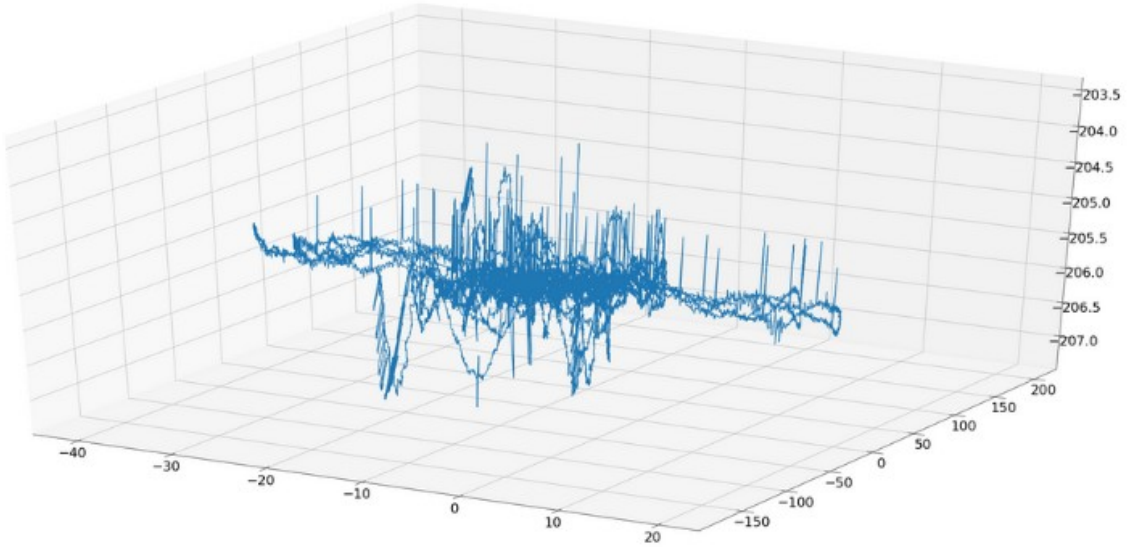
DEPH	(TIME)	float32	dask.array<chunksize=(4880,), meta=np.ndarray>
M1	(TIME)	float32	dask.array<chunksize=(4880,), meta=np.ndarray>
M2	(TIME)	float32	dask.array<chunksize=(4880,), meta=np.ndarray>
Null	(TIME)	float32	dask.array<chunksize=(4880,), meta=np.ndarray>
S	(TIME)	float32	dask.array<chunksize=(4880,), meta=np.ndarray>
P	(TIME)	float32	dask.array<chunksize=(4880,), meta=np.ndarray>
M	(TIME)	float32	dask.array<chunksize=(4880,), meta=np.ndarray>
Amont	(TIME)	float32	dask.array<chunksize=(4880,), meta=np.ndarray>
25m	(TIME)	float32	dask.array<chunksize=(4880,), meta=np.ndarray>
RRWind_X	(TIME)	float32	dask.array<chunksize=(4880,), meta=np.ndarray>
RRWind_Y	(TIME)	float32	dask.array<chunksize=(4880,), meta=np.ndarray>
RRWind_Z	(TIME)	float32	dask.array<chunksize=(4880,), meta=np.ndarray>

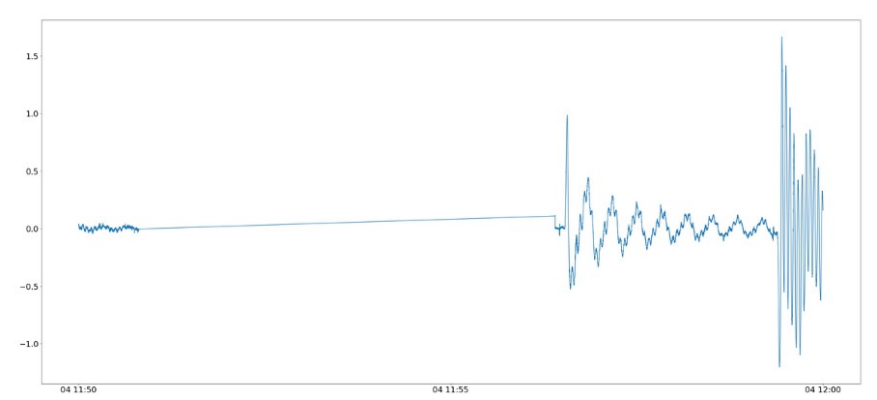
Easy data handling with xarray

Generate plots

In [15]:
```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import numpy as np

plt.rcParams.update({'font.size': 22})

fig = plt.figure(figsize=(40,40))
ax = fig.add_subplot(211, projection='3d')

ax.set
ax.plot(subset.RRWind_X, subset.RRWind_Y, subset.RRWind_Z)

ax2 = fig.add_subplot(212)
ax2.plot(subset.TIME, subset.RRWind_Roll)
```

Out[15]:
```
[<matplotlib.lines.Line2D at 0x2aab21dbbb80>]
```





Perspectives with Intake

Intake provides a **server** mode :

- share your catalog **and** the related data outside your datacenter