



Pangeo community platform and its use at CNES

BiDS 2019, Guillaume Eynard-Bontemps, CNES/Pangeo

Projected NASA Cloud Storage



Problems

- Data volume crisis in (geo)sciences
- Software multiplication, non reproducibility
- Many copies of the same datasets
- Local vs HPC vs Cloud
- Technology gap: industry vs academia

Mission

To cultivate an ecosystem in which the next generation of open-source analysis tools for the geosciences can be developed, distributed, and sustained.

Goals/vision

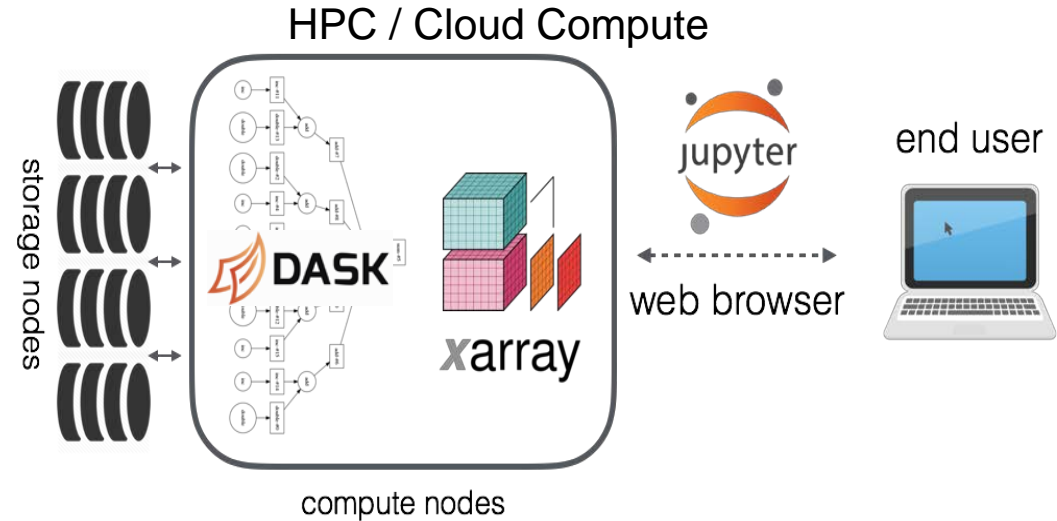
- Foster collaboration around the open source Scientific Python ecosystem:
 - open and collaborative development
 - Welcoming and inclusive culture
- Support the development with domain-specific (geo)science and transverse packages
- Improve scalability of these tools to handle gigabytes to petabyte-scale datasets



EARTH CUBE
TRANSFORMING GEOSCIENCES RESEARCH



- Set of tools that will facilitate science at all scales
- Platform agnostic
- The core of the Pangeo ecosystem includes:
 - **Xarray** (data-model and toolkit for working with N-dimensional labeled arrays)
 - **Dask** (parallel computing)
 - **Jupyter** (interactive computing)
- Extensible: Series of 3rd party packages that build on top of core libraries
- Flexible: Individual components may be swapped in/out



- Examples of 3rd party packages in the Pangeo Ecosystem:**
- Data discovery
 - Regridding and GIS
 - Vector calculus
 - Signal processing
 - Thermodynamics

BUILD YOUR OWN PANGEO

Storage Formats			Cloud Optimized COG/Zarr/Parquet/etc.
ND-Arrays			More coming...
Data Models			pandas $y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$ 
Processing Mode	 Interactive	Batch 	Serverless 
Compute Platform	HPC 	Cloud  Google Cloud Platform	Local 

Spark

vs

DASK

Mature

Robust

JVM/Python

Query optimized

Collections &
Dataframes

Python overhead

For big tabular data

Hadoop/Cloud/HPC

Less Mature

Pretty strong

Python only

Science optimized

Collections, DF,
Arrays, Futures...

Python only

For science data

Hadoop/Cloud/HPC



I don't know much
about Array DBs!!!

VS

databases
(Rasdaman,
SciDB...)

Laptop to cluster

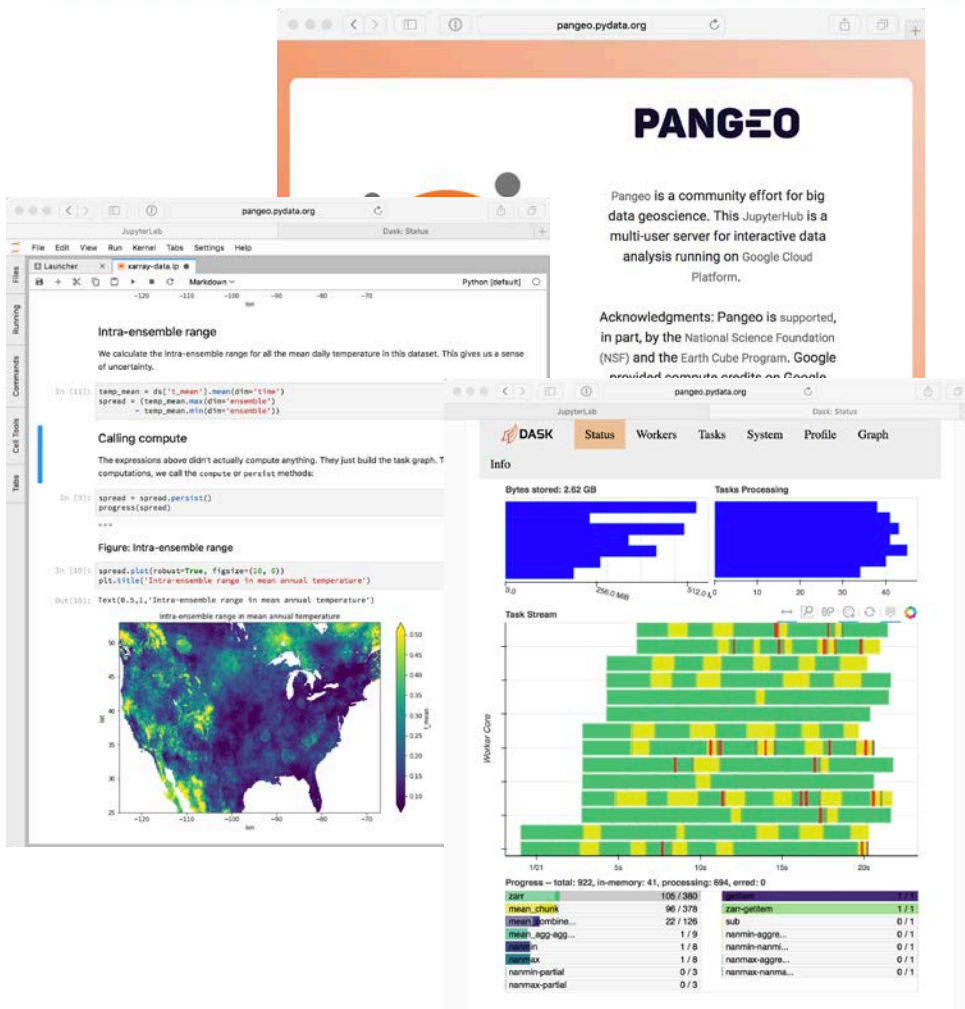
Serverless

NetCDF/TIFF no ingestion

Scales with Dask

Python only

Can build array db with Pangeo
(Open data Cube)



PANGEO

Pangeo is a community effort for big data geoscience. This JupyterHub is a multi-user server for interactive data analysis running on Google Cloud Platform.

Acknowledgments: Pangeo is supported, in part, by the National Science Foundation (NSF) and the Earth Cube Program. Google provided compute credits on Google Cloud.

```
Intra-ensemble range
We calculate the intra-ensemble range for all the mean daily temperature in this dataset. This gives us a sense of uncertainty.

In [11]: temp_mean = ds["temp"].mean(dim="time")
spread = (temp_mean.max(dim="ensemble") - temp_mean.min(dim="ensemble"))

Calling compute
The expressions above didn't actually compute anything. They just build the task graph. To compute them, we call the compute or persist methods.

In [12]: spread = spread.persist()
progress(spread)
+++

Figure: Intra-ensemble range
In [13]: spread.plot(robust=True, figsize=(10, 6))
plt.title('Intra-ensemble range in mean annual temperature')

Out[13]: Text(0.5,1,'Intra-ensemble range in mean annual temperature')
```

Figure: Intra-ensemble range in mean annual temperature

DASK Status Workers Tasks System Profile Graph

Info

Bytes stored: 2.62 GB

Tasks Processing

Task Stream

Worker Core

Progress -- total: 922, in-memory: 41, processing: 694, erred: 0

zarr	106 / 380	zarr-collections	1 / 1
mean_chunk	96 / 378	sub	0 / 1
mean_combine...	22 / 126	nanmin-aggr...	0 / 1
mean_agg-400...	1 / 9	nanmin-nanm...	0 / 1
nanmin	1 / 8	nanmax-aggr...	0 / 1
nanmax	1 / 8	nanmax-nanm...	0 / 1
nanmin-partial	0 / 3		
nanmax-partial	0 / 3		

hub.pangeo.io pangeo.binder.io

JupyterHub/BinderHub running on the Google Cloud

- Kubernetes for both Jupyter and Dask-distributed
 - Dask-kubernetes
- Exploring/evaluating:
 - Cloud storage
 - User environment customization
 - Data discovery
- Kubernetes Helm-chart (github.com/pangeo-data/helm-chart)
- CI/CD with Hubploy and CircleCI
- Deployments exist on AWS and Azure.

CNES Datacenter overview

A photograph of a server room with rows of server racks under blue lighting. The racks are arranged in a long aisle, and the floor is covered with a grid pattern. The lighting is a deep blue, creating a cool and technical atmosphere.

HPC (HAL)

- 300Tflops
- 380 batch servers / 8400cores
- 4 interactive servers pre/post processing w/ GPU
- 6,2 PB GPFS / 200TB burst buffer/ 50GBs bandwidth
- Low latency network
- GPGPU Nvidia Volta V100

HPC DRSF (Ktulu)

- 20 Tflops
- 2 interactive servers pre/post processing w/ GPU
- 24 servers / 576 cores
- 120TB GPFS
- Low latency network

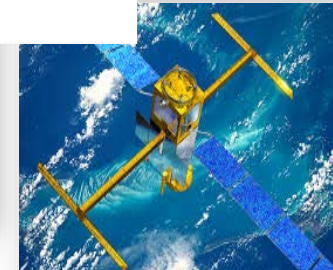
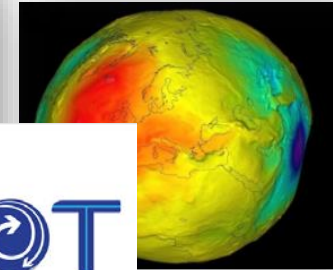
HPC usecases in CNES

Two main kinds of processing

Numerical simulation (HPC)

- Upstream phase, R&D
- Highly optimized technics
- Fine grain parallelism

Trends : multiscale, multiphysics



Data Processing (HTC)

- Downstream phase, operation
- Sensors data → scientific data
- Coarse grain parallelism

Trends : data volume explosion

Demo : Dask and dask-jobqueue basic example



```
File Edit View Run Kernel Tabs Settings Help
/user/eynardbg/ dask_local_vs_cluster_exam sentinel-ndvi-visu-multitemp x
1 from dask_jobqueue import PBSCluster
2 cluster = PBSCluster(cores=4, memory="15GB", local_directory='$TMPDIR',
3 queue='qdev', project='DaskTest',
4 walltime='01:00:00', interface='ib0')
5 cluster.adapt(minimum=2, maximum=16)
6
7 from dask.distributed import Client
8 client = Client(cluster)
9
10 def my_costly_simulation(args):
11     ## Do something
12     return sum(args)
13
14 input_params = [...]
15
16 futures = client.map(my_costly_simulation, input_params)
17 results = client.gather(futures)
18
```

cluster manuellement, ou de manière automatique (Adaptive cluster). On peut aussi configurer ça via des lignes des codes comme montré ci-dessous.

```
[ ]: cluster
Spécification manuelle de la taille du cluster. Le paramètre donné est le nombre de workers Dask.

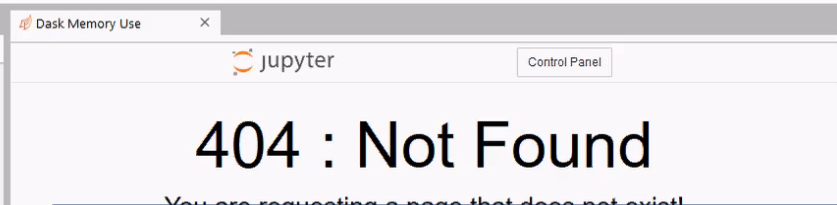
[ ]: cluster.scale(8)
Indiquer qu'on veut un cluster adaptatif (dont la taille variera en fonction de la charge).

Lancer la cellule suivante (ou précédente) doit modifier les informations affichées dans le widget.

[ ]: cluster.adapt(minimum=2, maximum=16)
```

Déclaration du client

Afin que toutes les API de dask (dataframe, delayed, bag ...) utilisent le cluster Dask qu'on a démarré, il est



Takeways:

- No enormous job arrays with short duration jobs. Bigger reservations. Scheduler relief.
- No use of shared storage for syncing results → less bad IOs.
- Simplified high level orchestration, all with Python and Dask

You are requesting a page that does not exist!

Demo : Multi temporal NDVI with Xarray



```
datasets = []
for tif in tif_files:
    try:
        print('loading...', tif)
        ds = create_multiband_dataset(tif)
        datasets.append(ds)
    except Exception as e:
        print('ERROR loading, skipping acquisition!')
        print(e)
```

```
20180215-105300-871_L2A_T31TCJ_D_V1-5/SENTINEL2A_20180215-105300-871_L2A_T31TCJ_D_V1-5_FRE_B
loading... zip:///datalake/S2-L2A-THEIA/31TCJ/2018/02/25/SENTINEL2A_20180225-105018-458_L2A_T31TCJ_D_V1-5_FRE_B
20180225-105018-458_L2A_T31TCJ_D_V1-5/SENTINEL2A_20180225-105018-458_L2A_T31TCJ_D_V1-5_FRE_B
loading... zip:///datalake/S2-L2A-THEIA/31TCJ/2018/02/27/SENTINEL2A_20180227-104236-560_L2A_T31TCJ_D_V1-5_FRE_B
20180227-104236-560_L2A_T31TCJ_D_V1-5/SENTINEL2A_20180227-104236-560_L2A_T31TCJ_D_V1-5_FRE_B
loading... zip:///datalake/S2-L2A-THEIA/31TCJ/2018/03/02/SENTINEL2A_20180302-105023-464_L2A_T31TCJ_D_V1-5_FRE_B
20180302-105023-464_L2A_T31TCJ_D_V1-5/SENTINEL2A_20180302-105023-464_L2A_T31TCJ_D_V1-5_FRE_B
loading... zip:///datalake/S2-L2A-THEIA/31TCJ/2018/03/07/SENTINEL2A_20180307-105016-460_L2A_T31TCJ_D_V1-5_FRE_B
20180307-105016-460_L2A_T31TCJ_D_V1-5/SENTINEL2A_20180307-105016-460_L2A_T31TCJ_D_V1-5_FRE_B
loading... zip:///datalake/S2-L2A-THEIA/31TCJ/2018/03/09/SENTINEL2A_20180309-104022-069_L2A_T31TCJ_D_V1-5_FRE_B
20180309-104022-069_L2A_T31TCJ_D_V1-5/SENTINEL2A_20180309-104022-069_L2A_T31TCJ_D_V1-5_FRE_B
loading... zip:///datalake/S2-L2A-THEIA/31TCJ/2018/03/14/SENTINEL2A_20180314-104014-461_L2A_T31TCJ_D_V1-7_FRE_B
20180314-104014-461_L2A_T31TCJ_D_V1-7/SENTINEL2A_20180314-104014-461_L2A_T31TCJ_D_V1-7_FRE_B
loading... zip:///datalake/S2-L2A-THEIA/31TCJ/2018/03/15/SENTINEL2A_20180315-110609-689_L2A_T31TCJ_D_V1-5_FRE_B
20180315-110609-689_L2A_T31TCJ_D_V1-5/SENTINEL2A_20180315-110609-689_L2A_T31TCJ_D_V1-5_FRE_B
loading... zip:///datalake/S2-L2A-THEIA/31TCJ/2018/03/17/SENTINEL2A_20180317-105355-924_L2A_T31TCJ_D_V1-5_FRE_B
20180317-105355-924_L2A_T31TCJ_D_V1-5/SENTINEL2A_20180317-105355-924_L2A_T31TCJ_D_V1-5_FRE_B
loading... zip:///datalake/S2-L2A-THEIA/31TCJ/2018/03/19/SENTINEL2A_20180319-104019-464_L2A_T31TCJ_D_V1-5_FRE_B
20180319-104019-464_L2A_T31TCJ_D_V1-5/SENTINEL2A_20180319-104019-464_L2A_T31TCJ_D_V1-5_FRE_B
```

In order to concatenate all images in an xarray dataset, we need to get the date for each file. Those dates will be use as a dimension for the xarray dataset.

```
In [23]: list_date = [filename.split('_')[1] for filename in tif_files]

import datetime
i = 0
for date in list_date:
    list_date[i] = date[:4] + '-' + date[4:6] + '-' + date[6:8] + ' ' + date[9:11] + ':' + date[11:13]
    list_date[i] = datetime.datetime.strptime(list_date[i], "%Y-%m-%d %H:%M:%S.%F")
    i += 1
dico = {'date' : list_date, 'tif_name' : tif_files}
pdf = pd.DataFrame(data=dico)
pdf.dtypes
```

Takeways:

- Interactive processing over temporal series with Jupyter/Dask/Xarray
- Interactive visualisation
- Recompute upon user Action
- Statistical analysis
- JPEG2000 is not a good interactive processing format
- Share data in cloud/distributed processing ready format

Conclusions

- Pangeo ecosystem greatly facilitates distributed computing and data analysis at scale
- It changes ways of doing it too
- Non monolithic platform built on top of existing Scientific Python stack and new related packages
- Community is always here to help
- Dask more versatile and easy to use than Spark.

Next steps

- Broaden users and use cases at CNES
- Encourage people to get in touch with Pangeo community
- Work in cooperation with others (Ongoing with Ifremer and CLS on SWOT aval data processing)

Pangeo website and discussions:

<https://pangeo.io>

<https://github.com/pangeo-data/pangeo/issues>

<https://medium.com/pangeo>

Pangeo Example + Binder:

<https://github.com/pangeo-data/pangeo-example-notebooks>

<http://binder.pangeo.io/v2/gh/pangeo-data/pangeo-example-notebooks/master>

Dask jobqueue:

<https://github.com/dask/dask-jobqueue>

Dask simple examples:

<https://github.com/dask/dask-examples>

My email

Guillaume.Eynard-Bontemps@cnes.fr