



Jupyter, Dask : traitement distribué simple et interactif sur HPC avec l'écosystème Pangeo
JCAD 2018, Guillaume Eynard-Bontemps, communauté Pangeo

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 dim, with a strong blue hue.

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

Two main kinds of processing

Numerical simulation (HPC)

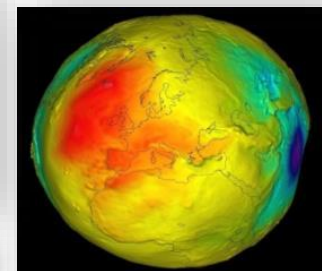
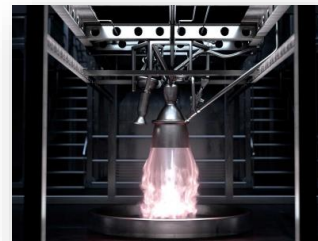
- Upstream phase, R&D
- Highly optimized technics
- Fine grain parallelism

Trends : multiscale, multiphysics

Data Processing (HTC)

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

Trends : data volume explosion



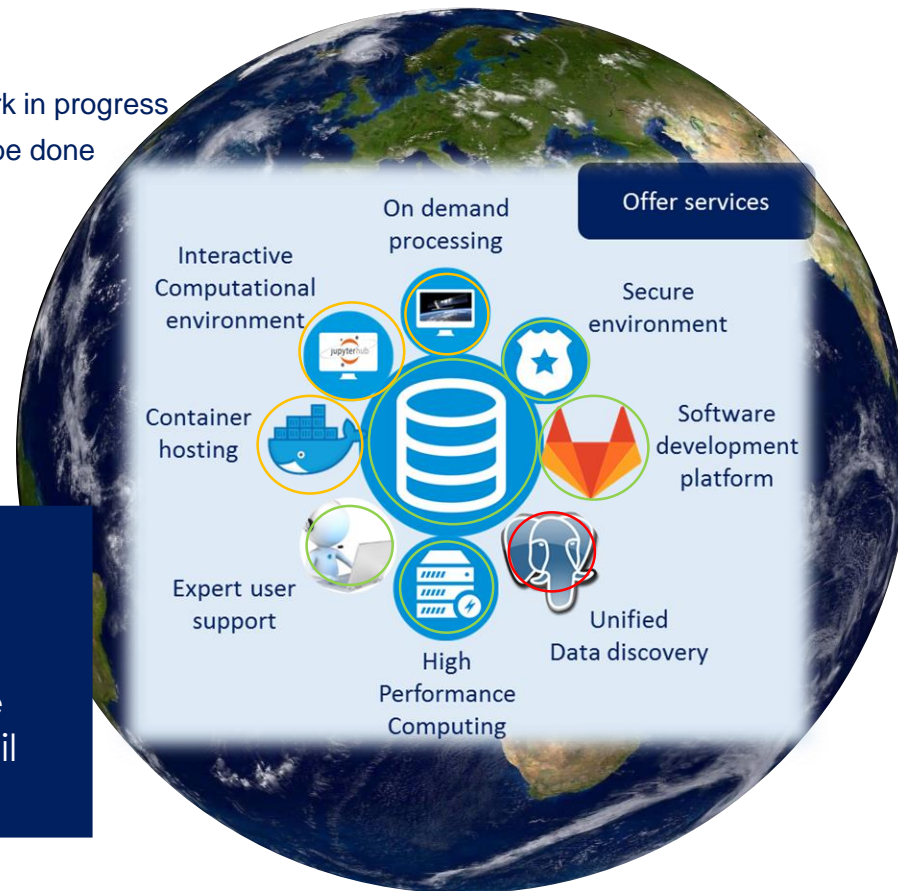
Objectifs

- Casser les silos de données
- Simplifier l'accès aux données
 - Entrée (Sentinel, Landsat, AUX, etc.)
 - Produite (Theia, CFOSat, etc.)
- Performance
- Offre de services complète

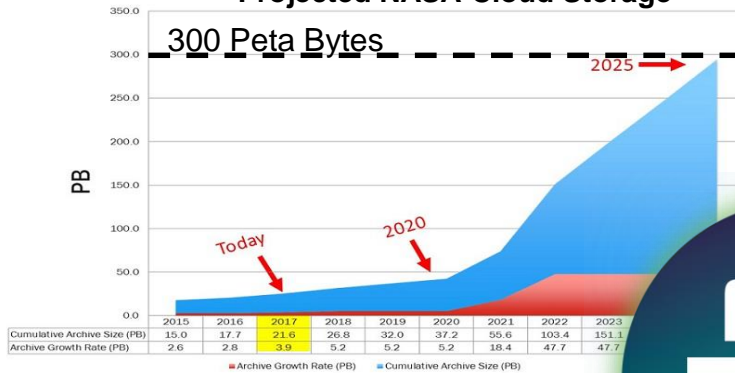
Challenges

- Fournir un environnement de développement standard, simple, flexible et performant (!)
- Favoriser l'adoption des nouveaux paradigmes de développement parallèle (Dask, Spark, ..), le travail des données en mémoire (vs orienté fichier)

- OK
- Work in progress
- To be done



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

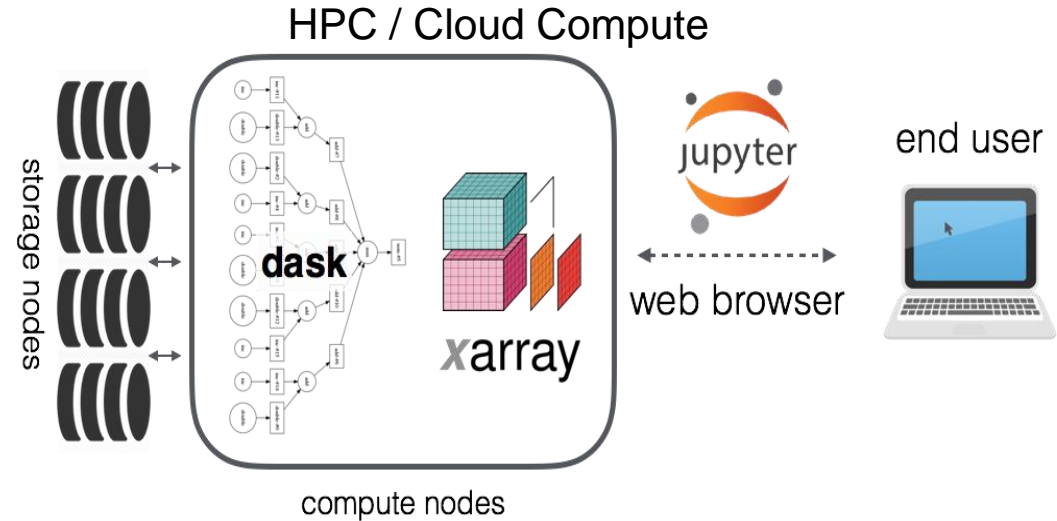
To cultivate an ecosystem in which the next generation of open-source analysis tools for the geosciences can be developed, distributed, and sustained.

Goals/vision

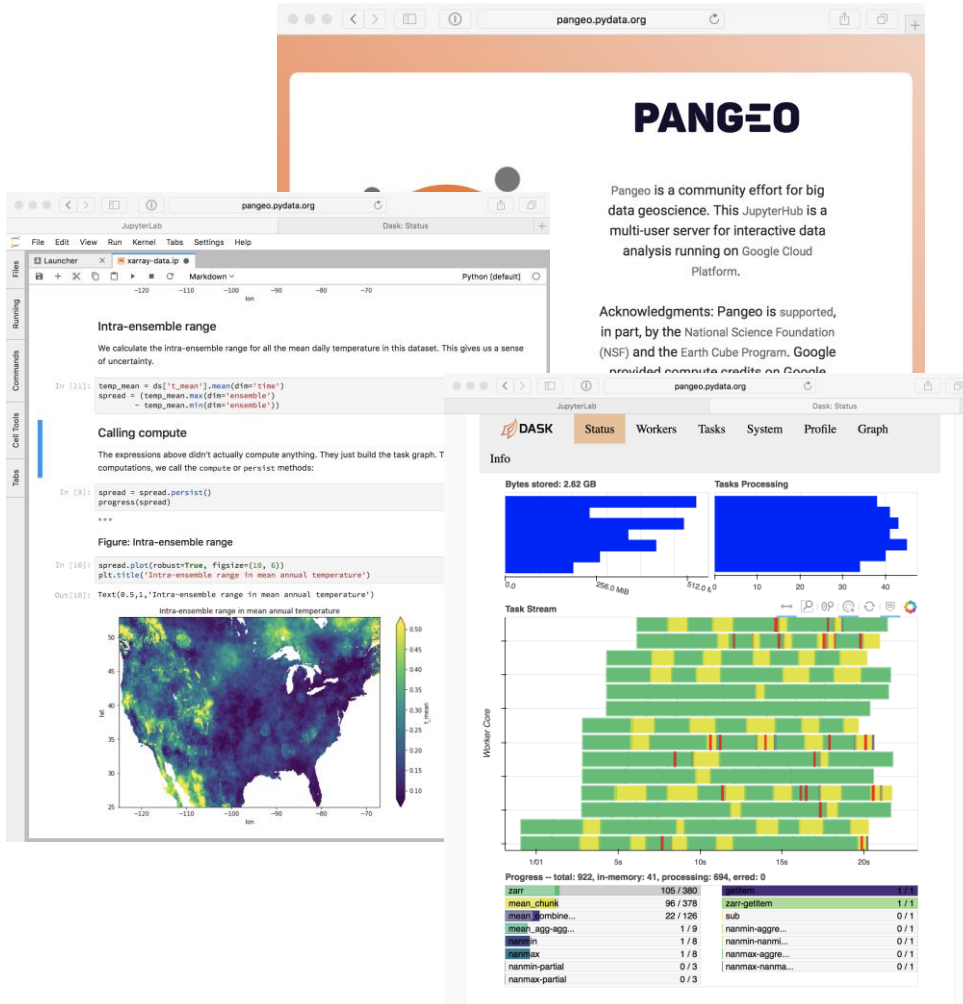
- Foster collaboration around the open source Scientific Python ecosystem:
 - open and collaborative development
 - Welcoming and inclusive culture
- Support the development with domain-specific (geo)science and transverse packages
- Improve scalability of these tools to handle gigabytes to petabyte-scale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



- 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



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.

```
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, we call the compute or persist methods:

```
In [9]: spread = spread.persist()
progress(spread)
***
```

Figure: Intra-ensemble range

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

Out[10]: Text(0.5, 1, 'Intra-ensemble range in mean annual temperature')

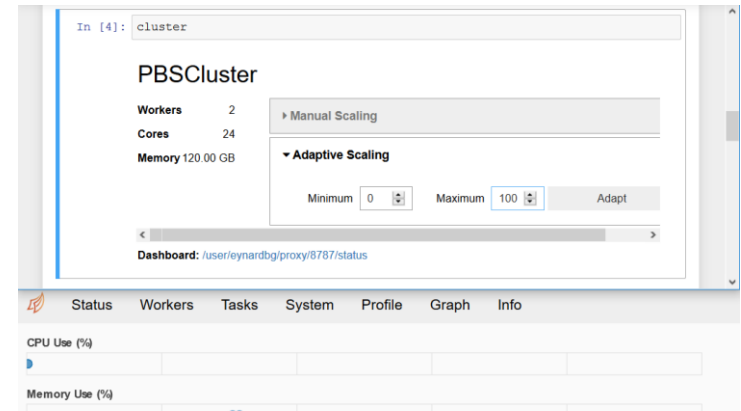
Task	Count	Task	Count
zarr	106 / 390	zarr-getitem	1 / 1
mean_chunk	96 / 378	sub	0 / 1
mean_combine...	22 / 126	nanmin-aggre...	0 / 1
mean_agg-agg...	1 / 9	nanmin	1 / 8
nanmin	1 / 8	nanmax-aggre...	0 / 1
nanmax	1 / 8	nanmax-nanma...	0 / 1
nanmin-partial	0 / 3		
nanmax-partial	0 / 3		

pangeo.pydata.org pangeo.binder.io

JupyterHub/Binder 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)
- Clones of our deployment have been made on AWS and Azure.

- JupyterHub and notebooks for interactive computing
 - Hub on a VM with qsub access
 - Batchspawner, Wrapspawner
- dask.distributed: parallel workers across many HPC nodes
- Xarray for computational toolkit and I/O
- New tool for deploying dask clusters on HPC: **dask-jobqueue**
 - Start a cluster from a notebook
 - Interactive (or not) distributed computing
 - Auto scaling capabilities



The screenshot shows the Dask cluster management interface for a cluster named 'PBSCluster'. The configuration is as follows:

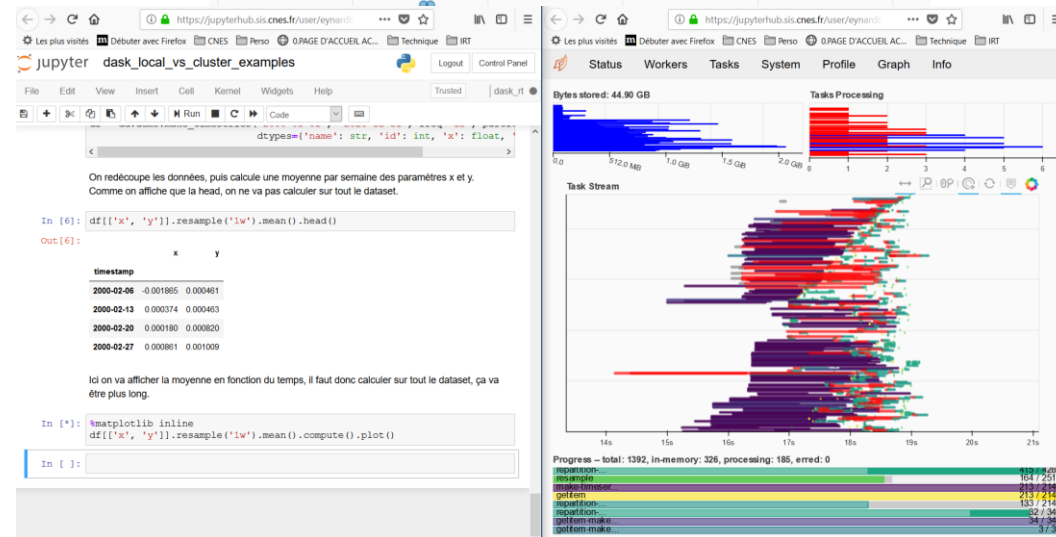
- Workers: 2
- Cores: 24
- Memory: 120.00 GB

Scaling options are visible:

- Manual Scaling: (button)
- Adaptive Scaling: (dropdown menu)
- Minimum: 0
- Maximum: 100
- Adapt: (button)

Dashboard URL: /user/eynardbg/proxy/8787/status

Navigation tabs: Status, Workers, Tasks, System, Profile, Graph, Info



The screenshot shows a Jupyter notebook titled 'dask_local_vs_cluster_examples'. The notebook content includes:

```
dtypes={'name': str, 'id': int, 'x': float, 'y': float}
```

On redcoupe les données, puis calcule une moyenne par semaine des paramètres x et y. Comme on affiche que la head, on ne va pas calculer sur tout le dataset.

```
In [6]: df[['x', 'y']].resample('1w').mean().head()
```

```
Out[6]:
```

timestamp	x	y
2000-02-06	-0.001895	0.000481
2000-02-13	0.000374	0.000463
2000-02-20	0.000190	0.000620
2000-02-27	0.000861	0.001009

Ici on va afficher la moyenne en fonction du temps, il faut donc calculer sur tout le dataset, ça va être plus long.

```
In [*]: %matplotlib inline
df[['x', 'y']].resample('1w').mean().compute().plot()
```

```
In [ ]:
```

On the right side, there are two charts:

- Bytes stored: 44.90 GB**: A horizontal bar chart showing data storage usage over time.
- Tasks Processing**: A Gantt-style chart showing the execution of tasks over time.
- Task Stream**: A scatter plot showing the distribution of tasks over time.

At the bottom, a progress bar shows: Progress - total: 1392, in-memory: 326, processing: 186, erred: 0. Below it are several progress bars for different stages: repartition, merge, splitFrom, repartition, splitFrom, repartition, splitFrom, repartition, splitFrom, repartition, splitFrom.

Dask and dask-jobqueue basic example

Some realistic workload

NDVI_serie Last Checkpoint: il y a 31 minutes (unsaved changes)

Kernel Widgets Help Trusted dask_r

On réalise le calcul de NDVI sur les différentes images de manière distribuée

```
In [9]: ndvITIF = realizeNDVI(big_image_array)
ndvITIF = ndvITIF.persist()
print(ndvITIF)
progress(ndvITIF)

dask.array<truediv, shape=(80, 8100, 9100), dtype=float64, chunksize=(1, 4050, 4550)>
```

Finished: 31 6s

640	getitem
320	truediv
320	sub
320	rechunk-split-rechunk-merge

netcdf_xarray_dask_CFOsat

```
In [*]: %time
dsset(["echo_11a_0*"]).max().compute()
```

Plot

```
In [6]: ds
Out [6]:
```

Going deeper with Xarray

Dask Status

Bytes stored: 3.48 GB

Tasks Processing

Task Stream

Progress - total: 895, in-memory: 150, processing: 442, erred: 0

open_catalog	100 / 100
concatenate	83 / 170
open_dataset	99 / 129

Image processing: NDVI

```
out = regenerate_reducer_for_premiere_image
imshow(ndvITIF[0].compute())
plotlib.image.AxesImage at 0x2abc74f9a3c8b>
```

Geoscience use cases:

http://pangeo.io/use_cases/index.html

Astronomy with GAIA catalog:

<https://github.com/pangeo-data/pangeo/issues/255#issuecomment-427186915>

Image processing and visualization

<https://medium.com/pangeo/cloud-native-geoprocessing-of-earth-observation-satellite-data-with-pangeo-997692d91ca2>

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

Next steps

- Broaden users and use cases at CNES
- Work in cooperation with others (Ongoing with Ifremer and CLS on SWOT aval data processing)

Pangeo wesite 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-examples>

Dask simple examples:

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

My email

Guillaume.Eynard-Bontemps@cnes.fr