THE PANGEO BIG DATA ECOSYSTEM AND ITS USE AT CNES

Guillaume Eynard-Bontemps, Joseph Hamman, Ryan Abernathey, Matthew Rocklin, Aurlien Ponte

CNES, NCAR, Columbia, Anaconda, Ifremer

ABSTRACT

Pangeo[1] is a community driven effort for open-source big-data intially focused on the Earth System Sciences. It represents at the same time a collaboration of people and a platform composed of open source scientific python packages like Jupyter, Dask and Xarray. One of its goal is to improve scalability of these tools to handle petabyte-scale datasets on HPC or public cloud infrastructure. In this paper, we will first describe Pangeo: its motivation, community, the underlying technology stack and associated deployments, different applications and the on going work. On a second part, we will present its use in CNES: HPC deployment, some simple and more complicated use cases, and what we are planning to do.

***Index Terms*—** Pangeo, Dask, Jupyter, HPC, Cloud, Big Data, Analysis

1. PANGEO

1.1. Motivations, mission and goals

There are several building crises facing the geoscience community:

- Big Data: datasets are growing too rapidly[1] and legacy software tools for scientific analysis cant handle them. This is a major obstacle to scientific progress.
- Technology Gap: a growing gap between the technological sophistication of industry solutions (high) and scientific software (low).
- Reproducibility: a fragmentation of software tools and environments renders most geoscience research effectively unreproducible and prone to failure.

Pangeo aims to address these challenges through a unified, collaborative effort.

Our mission is to cultivate an ecosystem in which the next generation of open-source analysis tools for ocean, atmosphere and climate science can be developed, distributed, and sustained. These tools must be scalable in order to meet the current and future challenges of big data, and these solutions should leverage the existing expertise outside of the geoscience community.

To accomplish this mission, we have identified three specific goals.

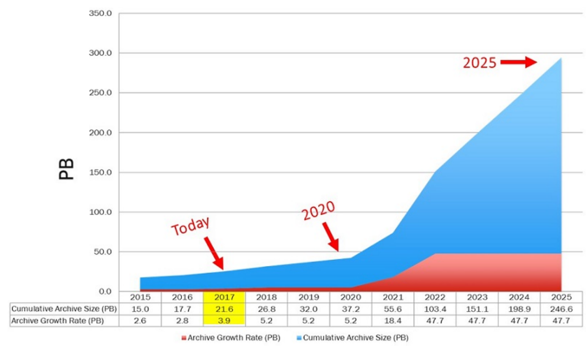

Fig. 1. Projected NASA Earth Observing System Cloud storage[2].

- Foster collaboration around the open source scientific python ecosystem for ocean / atmosphere / land / climate science.
- Support the development with domain-specific geoscience packages.
- Improve scalability of these tools to handle petabyte-scale datasets on HPC and cloud platforms.

1.2. Community

One crutial attribute of Pangeo is to be community driven. The goal is of course to have the most wider and open collaboration as possible. All the effort are made in the open on github[3], any one can join or get involved in the community.

The community is already quite diverse, from academic research, going through government agency and to open source developers. A lot of different nationalities are represented too: from USA of course, but also UK, France or Australia to name a few.

1.3. Technology stack

Pangeo software ecosystem fits directly in the Scientific Python stack, involving well known modules such as Numpy, Pandas, or Sickit-learn[2].

Three python software libraries are at the core of Pangeo:

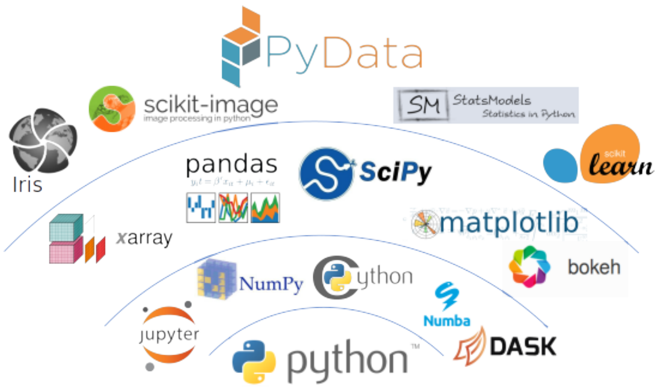

Fig. 2. Pangeo scientific Python ecosystem.

- Jupyter notebooks and Jupyterhub allow interactive computing and analysis from a web application and user handling. They are quickly becoming the standard open-source format for interactive computing in Python.

- Dask is a library for parallel computing that coordinates with Pythons existing scientific software ecosystem, including libraries cited above. In many cases, it offers users the ability to take existing workflows and quickly scale them to much larger applications. Dask-distributed is an extension of dask that facilitates parallel execution across many computers.

- Xarray is the interface for working with big datasets: it offers a Pandas like API for dealing with labelled n-dimensionnasl array at scale.

The github community offers online documentation, scripts and other tooling to link them together in order to deploy a Pangeo platform3 and put the stack on HPC systems or in the public Cloud. The main elements allowing to build and use the platforms along the core libraries are:

- A set of scripts and documentation that allows automatically creating the necessary Cloud Infrastructure. Currently available for Google Cloud Engine, but a lot of work is being done for Amazon Web Services compatibility. There is also a lot of automation that the community is working on using Terraform software or CI/CD tooling.

- Dask interfaces to automatically deploy distributed cluster allong various infrastructures: dask-kubernetes for Kubernetes cluster and so the public Cloud, dask-jobqueue[4] for HPC systems using scheduler such as Slurm, PBS Pro or LSF, and dask-yarn for Big Data infrastrucure.

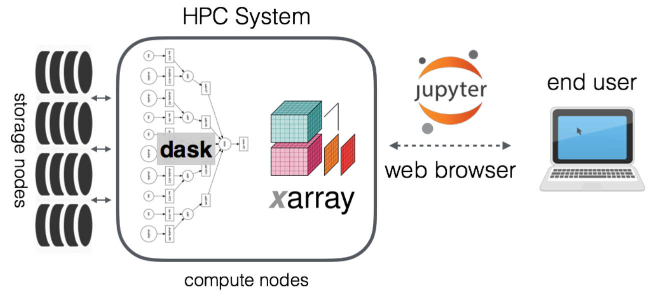

Fig. 3. Pangeo platform main components.

1.4. Applications

Pangeo first scientific domain is earth sciences: ocean, atmosphere, land or climate. There is already a lot of applications of the ecosystem in these domains. Some real science use cases can be found online[5] along their datasets and directly executed from a web browser. We can for example cite Sea Surface Altimetry Data Analysis that explore the increase of global mean sea level over 20 years of data, or also US Precipitation and Temperature Analysis that explore meteorological gridded ensemble precipitation and temperature estimates over the contiguous United States.

But other scientific domains are more and more interested by Pangeo possibilities:

- Satellite imagery analysis: there is a great blog post[6] explaining how to use Pangeo for processing and analysing in real time Landsat imagery. NASA has also decided to fund Pangeo initiative for applying the ecosystem on remote sensing datasets that are being put on AWS.

- Astronomy: several scientists are already using the web platform to explore Gaia DR2 data on GCE.

- Neuroscience: there is on going work to use Pangeo for analysis on human brain or electrophysiological datasets.

The principal point for allowing all this work is to dispose of a Cloud ready available dataset. This means that data must be accessible in a cloud or distributed storage friendly format, like Zarr for multi dimensionnal data, Cloud optimized Geotiff for satellite imagery, or Parquet for tabular data. See [8] for more details on this.

1.5. On going work

The community is very active and working on improving Pangeo possibilities and accessibilities. This includes but does not limit to the following on going tasks:

- Active work for giving everyone the possibility to try to use Pangeo at scale. This is made possible by using

Binder tools along with Pangeo ecosystem[9]. Binder allows anyone to launch a cloud environment and interactive notebook from a simple clic on a HTML link.

- Community governance: Pangeo is still young, it needs further organization. Lot of work has begun on this in the last Pangeo developper meeting this summer.
- As mentioned previously, Pangeo is an ecosystem that can be used by different scientific domains, each of thos having their own software stack. In order to cope with that, it has recently been decided to offer a specific cloud environment for all these different communitites.
- In order to simplify the above point, a lot of work is currently being done on simplifying Cloud environment updates, by using mordern tooling for Continous Deployment.
- As Pangeo is compatible not only with cloud but with different infrastructure, several Dask subcomponents share some logic for deploying clusters. A work to rationalize this logic and separated resources management from Dask scheduling is on going.

2. CNES DEPLOYMENT AND USE CASES

2.1. Context and projects

The Centre National d'Etudes Spatiales (CNES) is the government agency responsible for shaping and implementing Frances space policy in Europe. As such it covers a wide range of subjects: Ariane launcher, Sciences, Earth observation, telecommunication and defence.

On the ground segment processing side, we are involved on several Big Data projects, hosted or not in CNES Data Center:

- Gaia data processing center for some science unit,
- Sentinel product Exploitation Platform, for sharing and online processing of copernicus products,
- Surface Water and Ocean Topography (SWOT) mission, with French processing center hosted at CNES,
- Euclid astronomy mission, for the architecture part.

We are also performing a lot of other processing involving remote sensing data, flight dynamics or other domain on our HPC cluster.

2.2. Pangeo on HPC System

CNES main processing platform is our cluster, named HAL. It is a modestly sized High Performance Computer: about 8000 cores, 6PB storage, some Volta GPU. It uses PBS Pro scheduler to schedule the load on compute nodes and handle user or project priority.

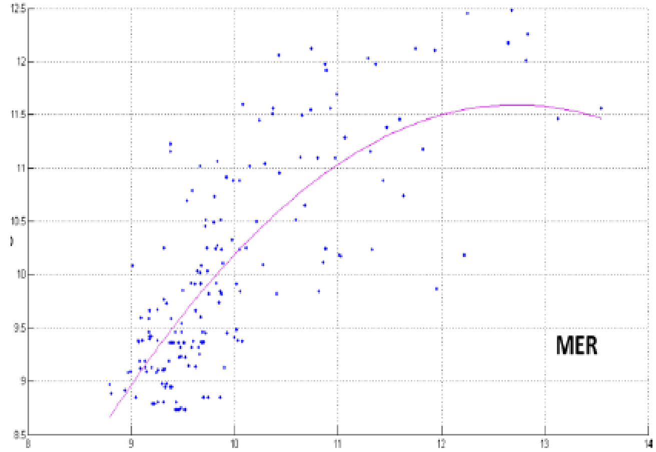

Fig. 4. Notebook and Dask dashboard running with dask-jobqueue.

Pangeo platform has recently been deployed on the cluster, which basically means the configuration of two main components: a Jupyterhub and Dask through dask-jobqueue4.

Jupyterhub has been deployed on a Virtual Machine, which has direct access to HAL cluster through PBS commands, and can also mount its shared file system. This way it is easy to configure Jupyter Batchspawner which launch user notebook using PBS scheduler, alongside Wrapspwaner to be able to select adequate system resources for launched notebook.

Some contributions to dask-jobqueue has been done in order to improve its usability, and then its deployment (along Xarray or other domain scientific library) is quite simple as it can be done through conda or pip python packaging system. A template configuration is proposed to all HAL users. There is also a few commands to issue in order to be able to use the python environment inside Jupyter, and some demonstration notebooks have been shared.

2.3. From embarrassingly parallel to more complex workflow with Dask

Embarrassingly parallel simulation

Before: complex and unreadable batch scripts, launching PBS arrays, writting millions of small results on shared storage.

With Dask: elegant python code, scaling easily, in memory data exchange...

Add some post processing

Add some CSV file generation, build of a dask dataframe, reduction of results, launch of a new simulation...

We can now run this on all of our input parameters:

```python
import dask
lazy_results = []

for parameters in input_params.values:
    lazy_result = dask.delayed(costly_simulation)(parameters)
    lazy_results.append(lazy_result)

futures = dask.persist(*lazy_results)  # trigger computation in the background
```

To make this go faster, we can add additional workers.

(although we're still only working on our local machine, this is more practical when using an actual cluster)

```python
for i in range(10):
    client.cluster.start_worker(ncores=4)
```

By looking at the Dask dashboard we can see that Dask spreads this work around our cluster, managing load balancing, dependencies, etc..

Then get the result:

```python
results = dask.compute(*futures)
results[:5]
```

```
(3.0013982136026325,
 1.1594091735837657,
 1.6780850038018285,
 2.316070438563014,
 1.1028165765683922)
```

Fig. 5. Embarrassingly parallel workload using Futures API.

2.4. Simulating remote sensing data through dask array

Generating a lot of Dask array that needs to be sum. Memory problem, rechunking...

REFERENCES

[1] Abernathey, Ryan; paul, kevin; hamman, joe; rocklin, matthew; lepore, chiara; tippett, michael; et al. (2017): Pangeo NSF Earthcube Proposal. Figshare paper.

[2] Mark McInerney, ESDIS Project Deputy Project Manager/Technical: EOSDIS Cloud EvolutionNASA EOSDIS web site.

[3] Ryan Abernathey et al: Pangeo github project issue tracker.

[4] Joseph Hamman, Matthew Rocklin , Jim Edwards, Guillaume Eynard-Bontemps, Loc Estve, et al. (2018): Scalable interactive analysis workflows using dask on HPC Systems.

[5] Pangeo community: Pangeo use cases.

[6] Scott Henderson, Daniel Rothenberg, Matthew Rocklin, Ryan Abernathey, Joe Hamman, Rich Signell, and Rob Fatland: Cloud Native Geoprocessing of Earth Observation Satellite Data with Pangeo.

[7] Ryan Abernathey: Step-by-Step Guide to Building a Big Data Portal.

[8] Ryan Abernathey: Step-by-Step Guide to Building a Big Data Portal.

[9] Joe Hamman, Ryan Abernathey: Pangeo meets Binder.