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: the context, our local HPC deployment, some simple and more complicated use cases. We will then conclude by our overall view of Pangeo.

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

1. PANGEO

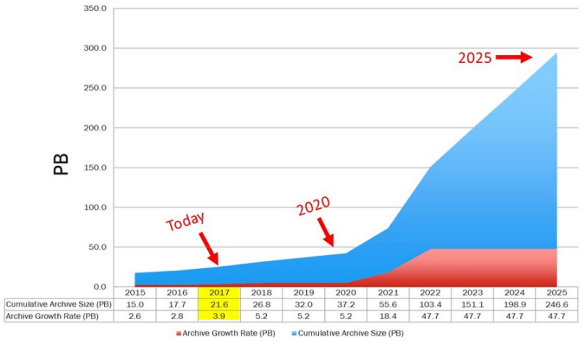
1.1. Motivations, mission and goals

There are several building crises facing the geoscience community:

- Big Data: datasets are growing too rapidly 1 and legacy software tools for scientific analysis cannot 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, climate and eventually other sciences 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.



	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025
Cumulative Archive Size (PB)	15.0	17.7	21.6	26.8	32.0	37.2	55.6	103.4	151.1	198.9	246.6
Archive Growth Rate (PB)	2.6	2.8	3.9	5.2	5.2	5.2	18.4	47.7	47.7	47.7	47.7

Archive Growth Rate (PB) Cumulative Archive Size (PB)

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 and driven openly 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:

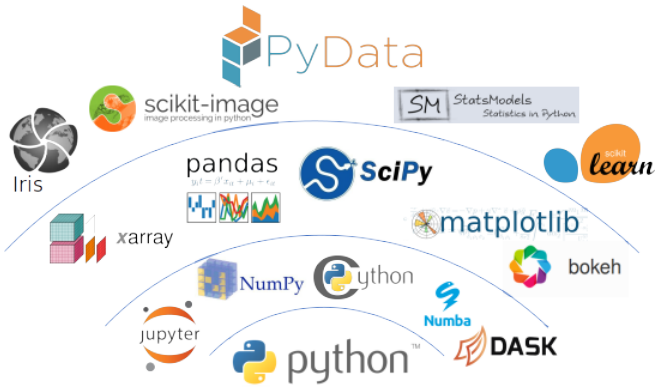


Fig. 2. Pangeo scientific Python ecosystem.

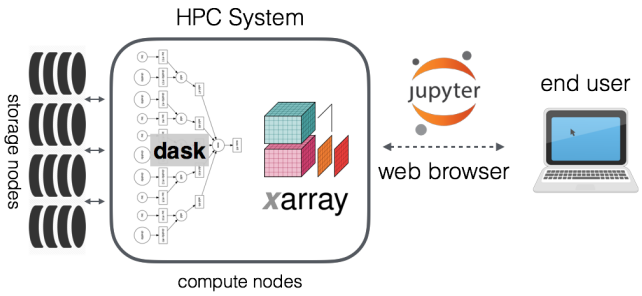


Fig. 3. Pangeo platform main components.

- Jupyter notebooks and Jupyterhub allow interactive computing and analysis from a web application and multiple 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-dimensionnal arrays 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 first 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.

Second element are Dask interfaces to automatically deploy distributed cluster allong various infrastructures are being developed: 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.

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 those: some real science use cases can be found online[5] along their datasets and can be 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 found Pangeo initiative for applying the ecosystem on remote sensing datasets that are being pushed on AWS.
- Astronomy: several scientists are already using the web platform to explore Gaia DR2 data on Google Cloud.
- Neuroscience: there is some on-going work to use Pangeo for analysis on human brain or electrophysiological datasets.

Allowing all this work suppose to dispose of Cloud ready available datasets. 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 [7] for more details on this crutial point. Format like NetCDF or CSV files are not adapted or optimized for use at scale.

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 Pangeo at scale. This is made possible by using Binder tools along with our ecosystem[8]. Binder allows anyone to launch a cloud environment and interactive note-

book from a simple clic on a HTML link, now possibly with the computing power of Pangeo stack.

- Community governance: the project is still young, it needs further organization. A lot of work was initiated 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 those having their own specific software stack. In order to cope with that, it has recently been decided to offer a specific cloud environment for all these different communities.

- In order to achieve the above point, a lot of work is currently being done on simplifying Cloud environment updates, by using mordern tooling for Continous Deployment like Hubploy or CircleCI.

- 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 separate 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 France's 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 overall coordination of Science Data Centers.

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

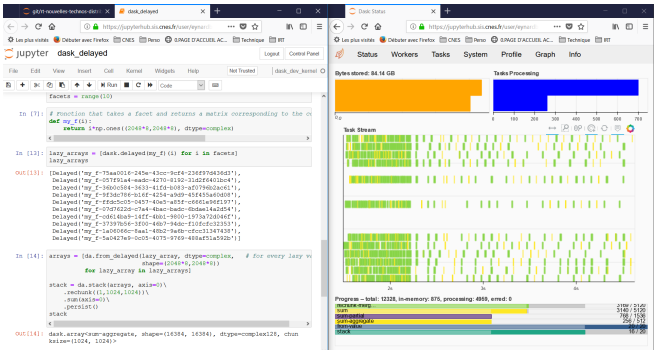


Fig. 4. Notebook and Dask dashboard running with dask-jobqueue on dask array computation.

2.2. Pangeo on our HPC System

CNES main processing platform is a modestly sized High Performance Computer named HAL. It provides some computing resources : about 8000 Intel cores, 6PB storage using IBM Spectrum Scale technology, some NVidia Volta GPU. It uses PBS Pro jobqueue system to schedule the load on compute nodes and handle user and project resource sharing.

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. This way it is easy to configure Jupyter Batchspawner which launch user notebooks using PBS *qsub* command, 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 on HAL. This Python module deployment (alike Xarray or other domain scientific library) is quite simple as it can be done through conda or pip packaging system. A template configuration is proposed to all HAL users. There is also a few commands documented in order to be able to use the python environment and thus create Dask cluster from inside Jupyter, some demonstration notebooks have been shared internally.

2.3. From embarrassingly parallel to more complex workflow with Dask

One important use of our cluster is to do batch jobs: apply a given computation or process to a list of input, which can either be a list of files or a list of parameters. This is usually done with job arrays PBS mechanism. Results are then written into our central storage facility, and a final job will gather them and consolidate them if needed. There are three main drawbacks to this approach:

- When scaling this mechanism to numerous (say undreds of thousands) and short (under one minute) jobs,

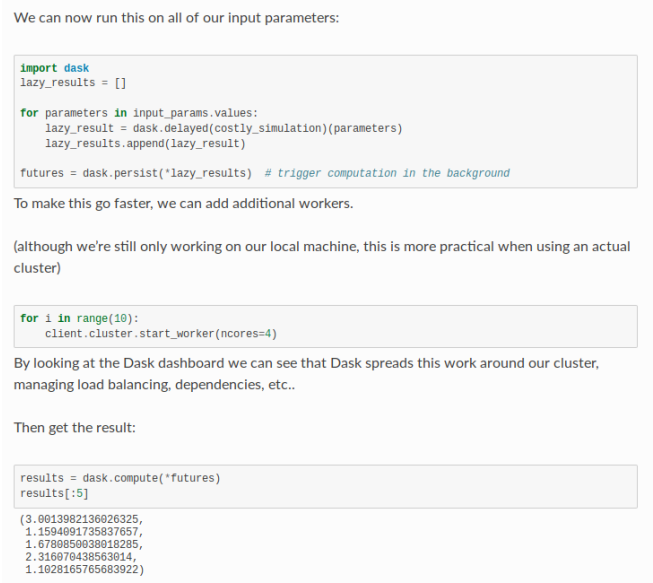


Fig. 5. Embarrassingly parallel workload using Delayed API.

this can lead to PBS scheduler contention and slow responsiveness.

- This often means a lot of bash script and machinery to chain several analysis together, leading in unreadable or unmaintable workflows.

- As results can be really small and are exchanged through a centralized storage system, this also means a lot of load onto the File System and side effects for other users.

Pangeo, mainly through Dask, gives a wonderful solution to all this problems (see 5 for an example). All the workflow, cluster creation, data management parts are handled from python code. Reservation to PBS are only done one time using dask-mpi or by bigger blocks using dask-jobqueue for long running worker process. No need to write or exchange data through disk, all data management is done through memory or network with TCP over Infiniband. This allows to analyse the result of a simulation in the same piece of code where we launched it, and eventually gather only the reduced valuable part for later use. The result of all this is elegant and simple Python code, which can scale easily to thousands of cores and does not stress our HPC cluster main components.

2.4. Interactively simulating remote sensing data through dask array

Another example we implemented with Pangeo was the generation of simulated data from a remote sensing satellite. The main idea of the computation is that we need to generate a lot of big two dimension array (2 or 4 GB each) that represent a part of the simulated output, and them sum all of those together. A first implementation was previously done using numpy and multiprocessing modules, but it was difficult to do the sum in memory, to scale beyond one compute node, and it involved temporary file writing.

Thanks to Jupyter and Dask, we were able to rapidly and interactivly prototype an new algorithm version. Dask solves all of the previous algorithm problems, taking advantage of numpy array chunking and efficient lazy task execution in graphs for chained operations. One problem remains with our solution which is really simple yet: it does not optimize the memory usage, as we need to create all arrays before rechunking and suming them together, but thanks to Pangeo we will fix this soon enough.

3. CONCLUSION

Pangeo is a great ecosystem which enables science at scale on Cloud or on premise infrastructure. We encourage every lab, governement agency, or even industry massive players to take a look at what it proposes. The community is open an always looking for new people to join. At CNES, we decided to focus on this tooling for our shared computing infrastructure, and it already shows its power and its benefit. On going work is to try more and more use cases with Pangeo to identify where it is most useful, and possible limits to the software stack. A collaboration with Ifremer (which has already used Dask[**?**]) on SWOT data valorization will soon help achieve this. We are also trying to actively participate to the community and share our vision and needs to guide by little touches the common effort.

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] Joe Hamman, Ryan Abernathey: Pangeo meets Binder.

[9] S. Fresnay, A. L. Ponte, S. Le Gentil, and J. Le Sommer: Reconstruction of the 3-D Dynamics From Surface Variables in a High-Resolution Simulation of North Atlantic. DOI: 10.1002/2017JC013400.