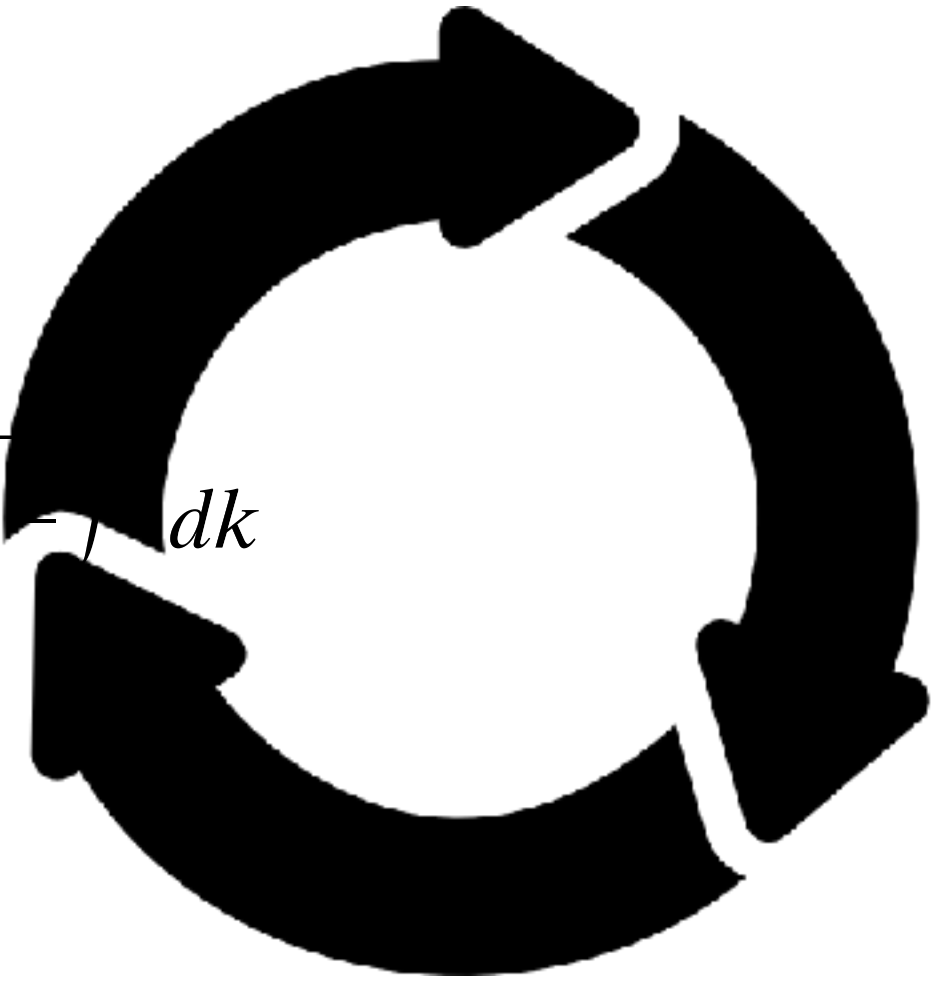
# PANGEO

A COMMUNITY-DRIVEN EFFORT FOR
BIG DATA GEOSCIENCE

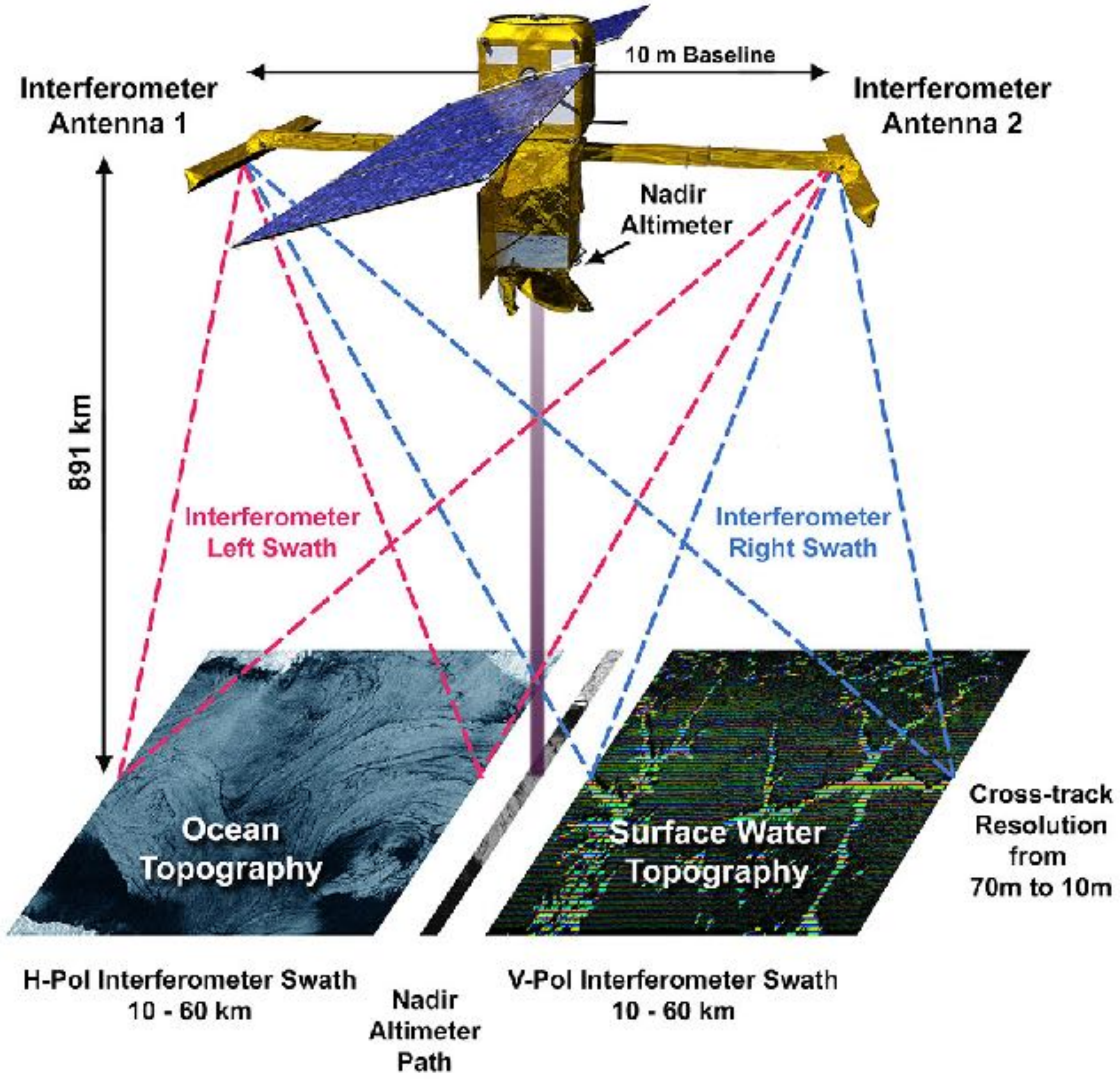
# WHAT DRIVES PROGRESS IN OCEANOGRAPHY?

## New Ideas

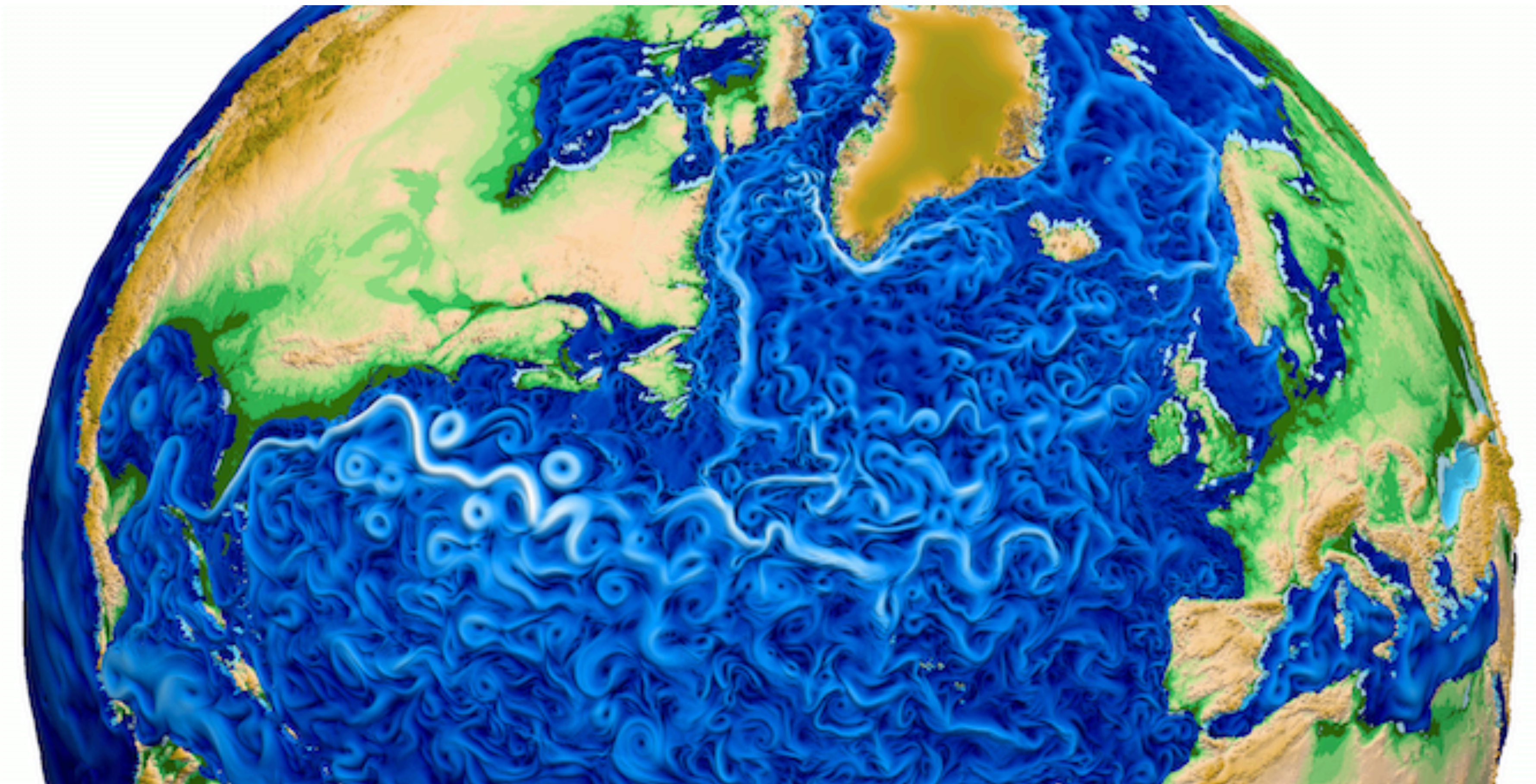
$$E = \frac{\rho_0 |\mathbf{U}|}{\pi} \int_{|f|/|\mathbf{U}|}^{N/|\mathbf{U}|} P_{1\mathrm{D}}(k) \sqrt{N^2 - |\mathbf{U}|^2 k^2} \sqrt{|\mathbf{U}|^2 k^2 - f^2}\, dk$$

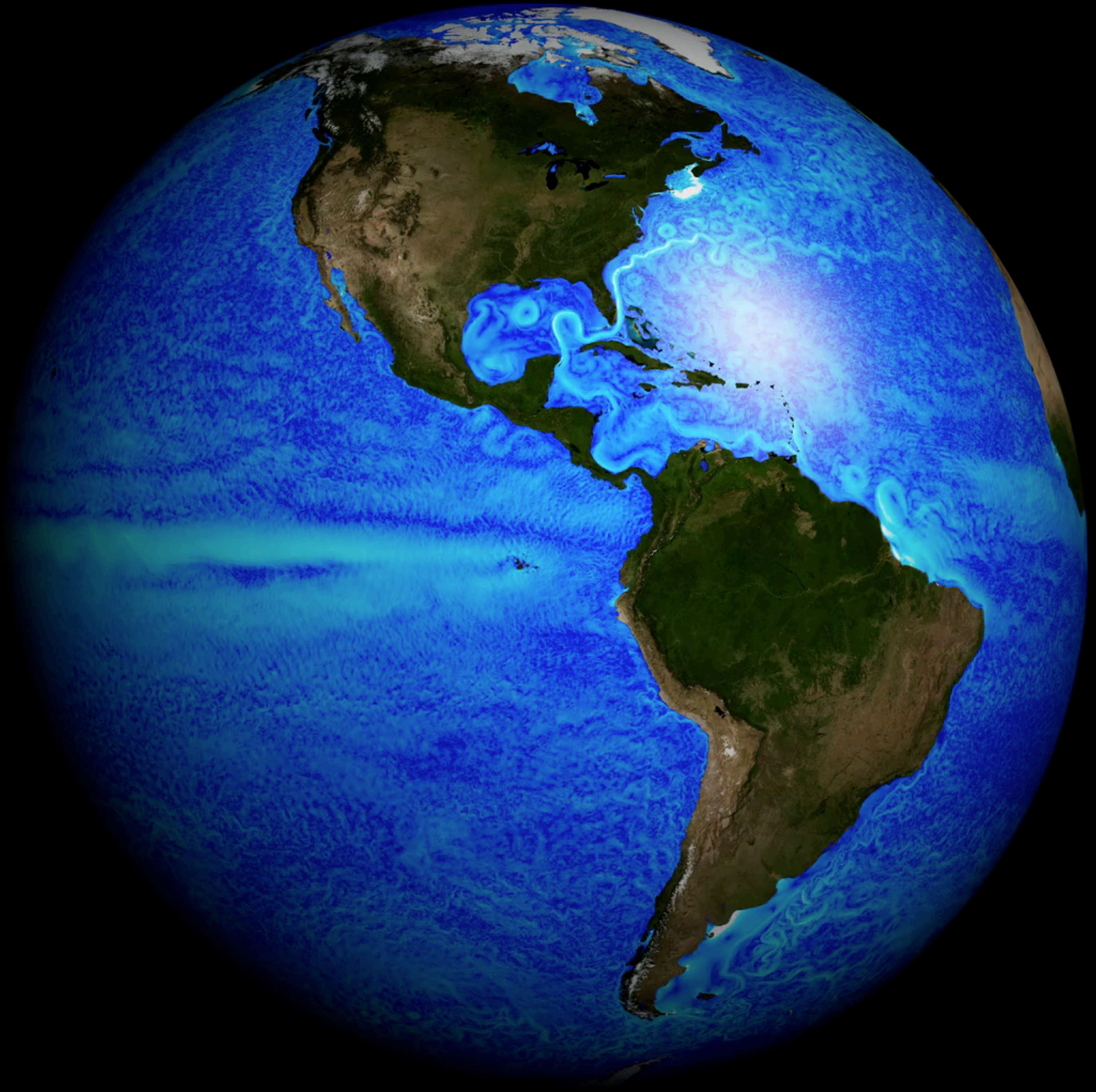


## New Observations




## New Simulations





## MITgcm LLC4320 Simulation

**Grid resolution:**
***1 - 2 km***

**Single 3D scalar field:**
***80 GB***

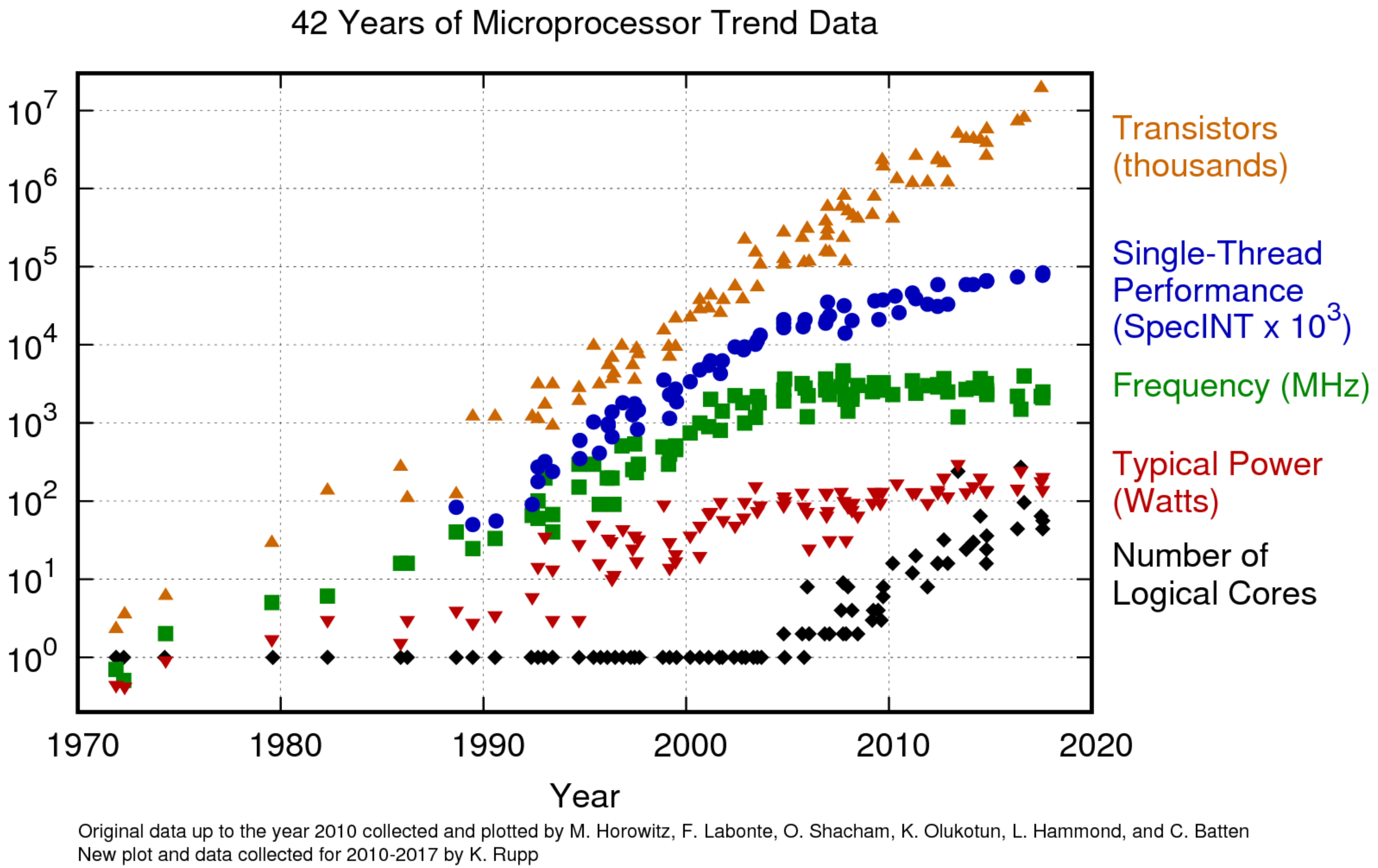
**Output frequency:**
***1 hour***

**Simulation Length:**
***1 year***

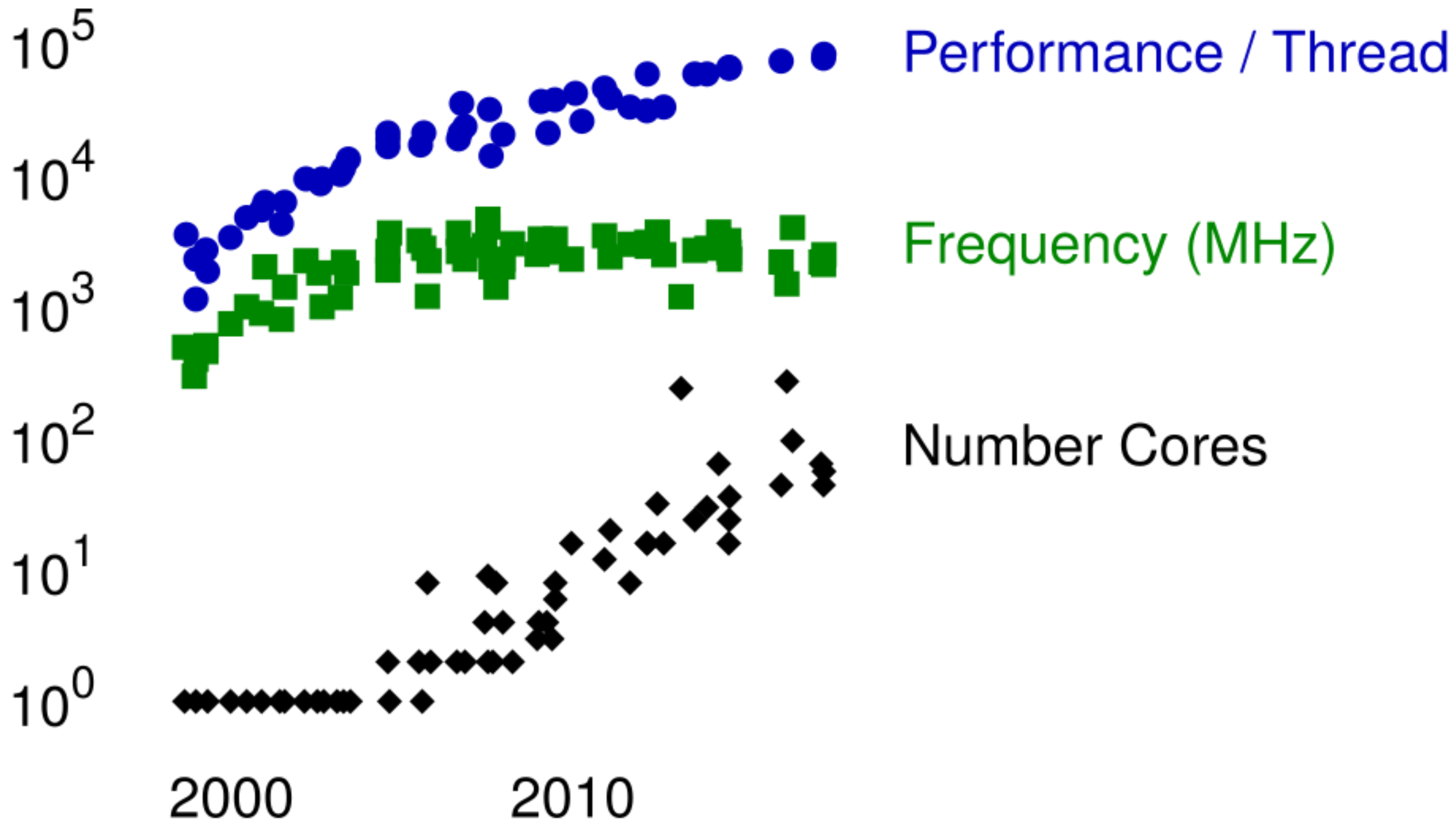
**Output data volume:**
***2 PB***

**Credit: NASA JPL / Dimitris Menemenlis**

# COMPUTER ARCHITECTURE



# COMPUTER ARCHITECTURE

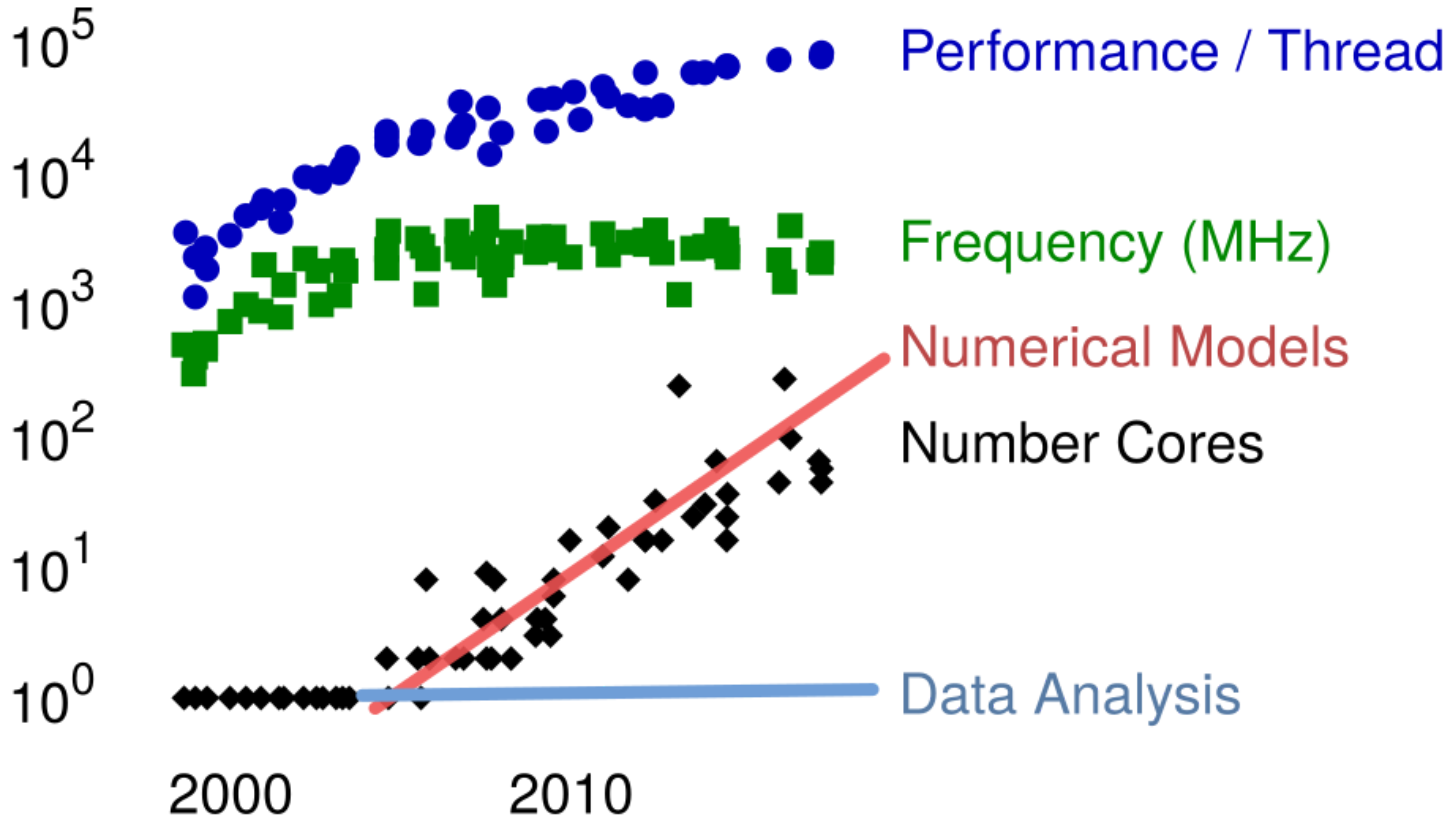


# COMPUTER ARCHITECTURE: SIMULATION


**NASA Pleaides Supercomputer**

# COMPUTER ARCHITECTURE: ANALYSIS AND VISUALIZATION

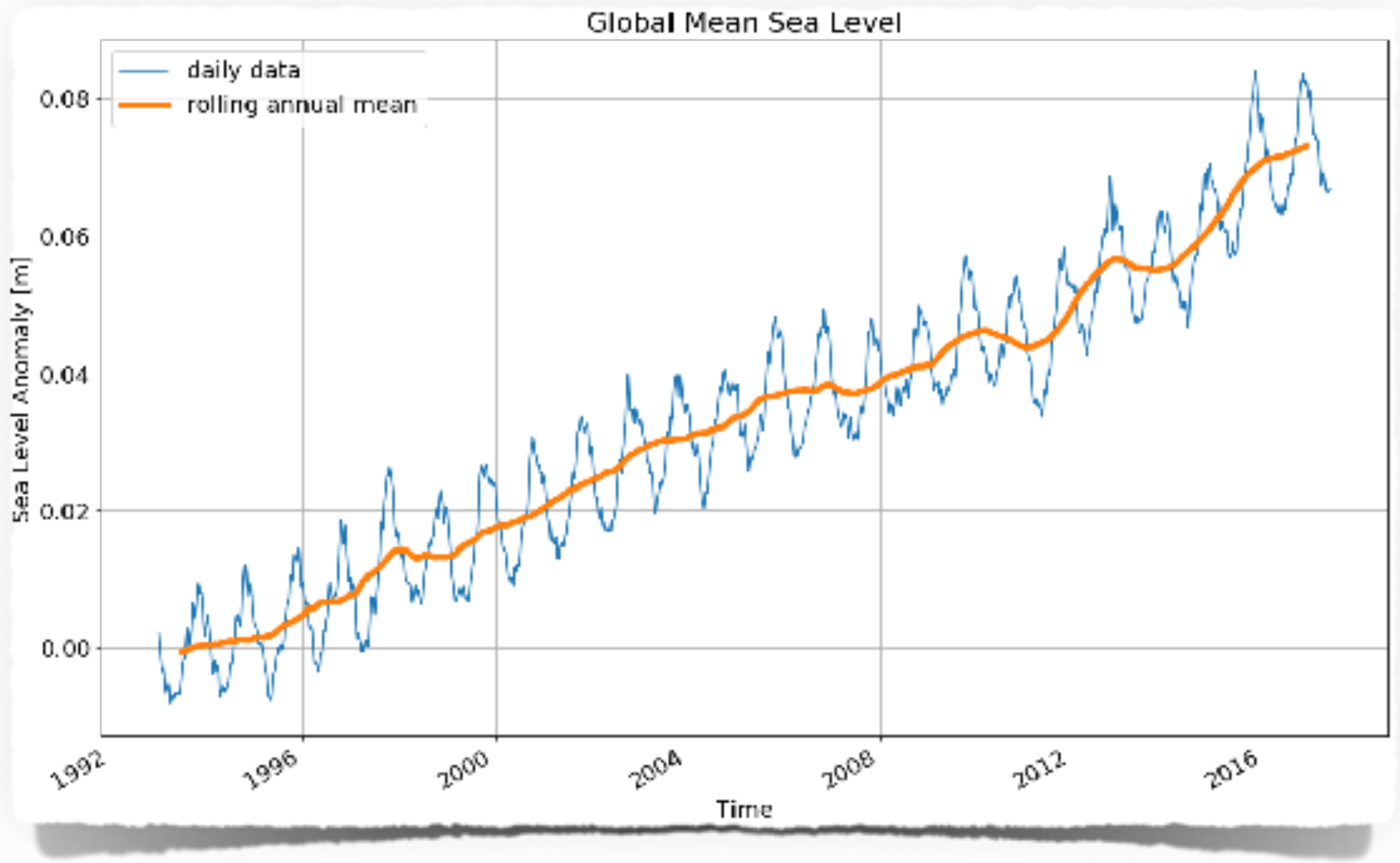
# COMPUTER ARCHITECTURE



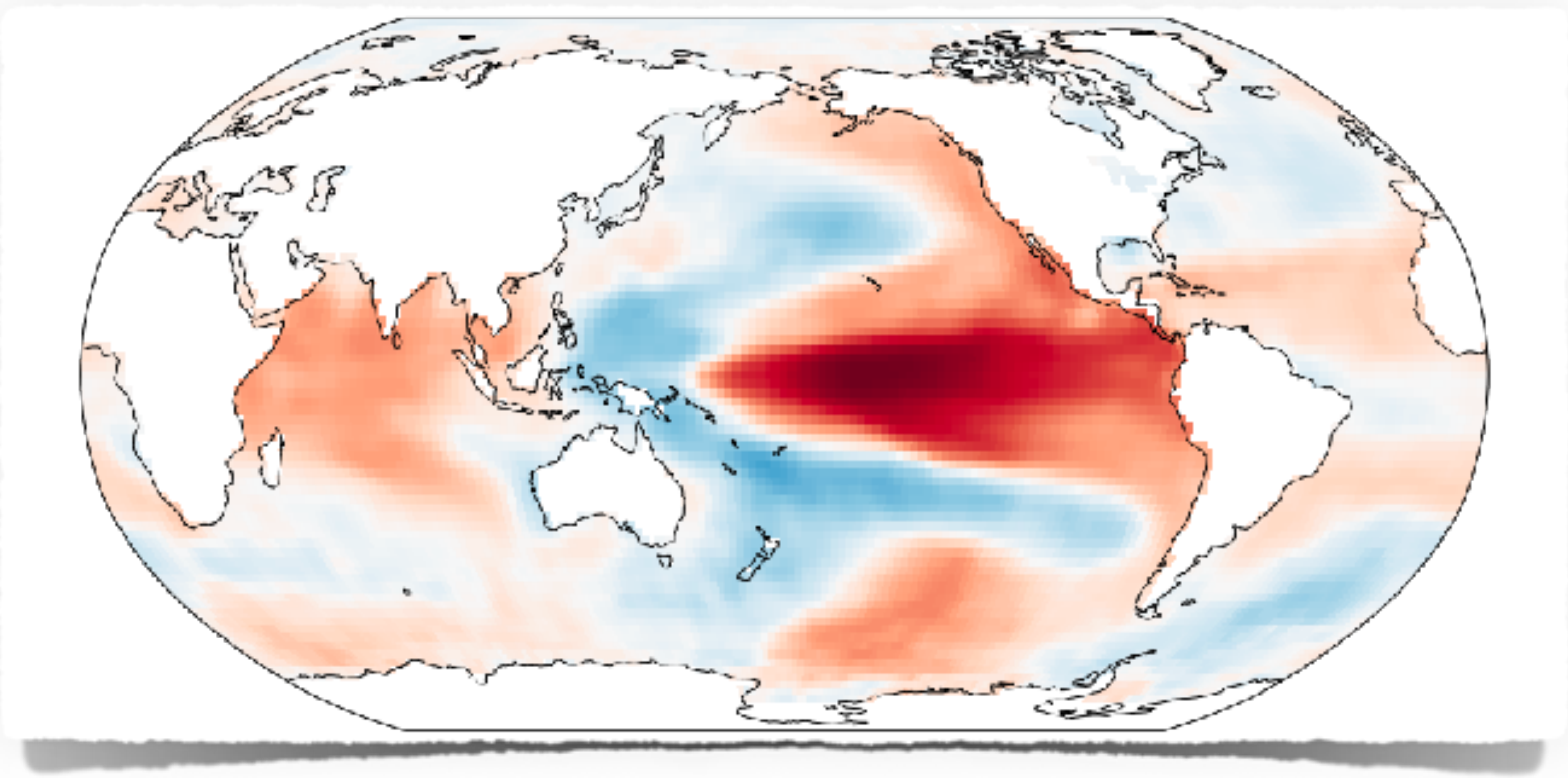
PANGEO

# WHAT SCIENCE DO WE WANT TO DO WITH CLIMATE DATA?

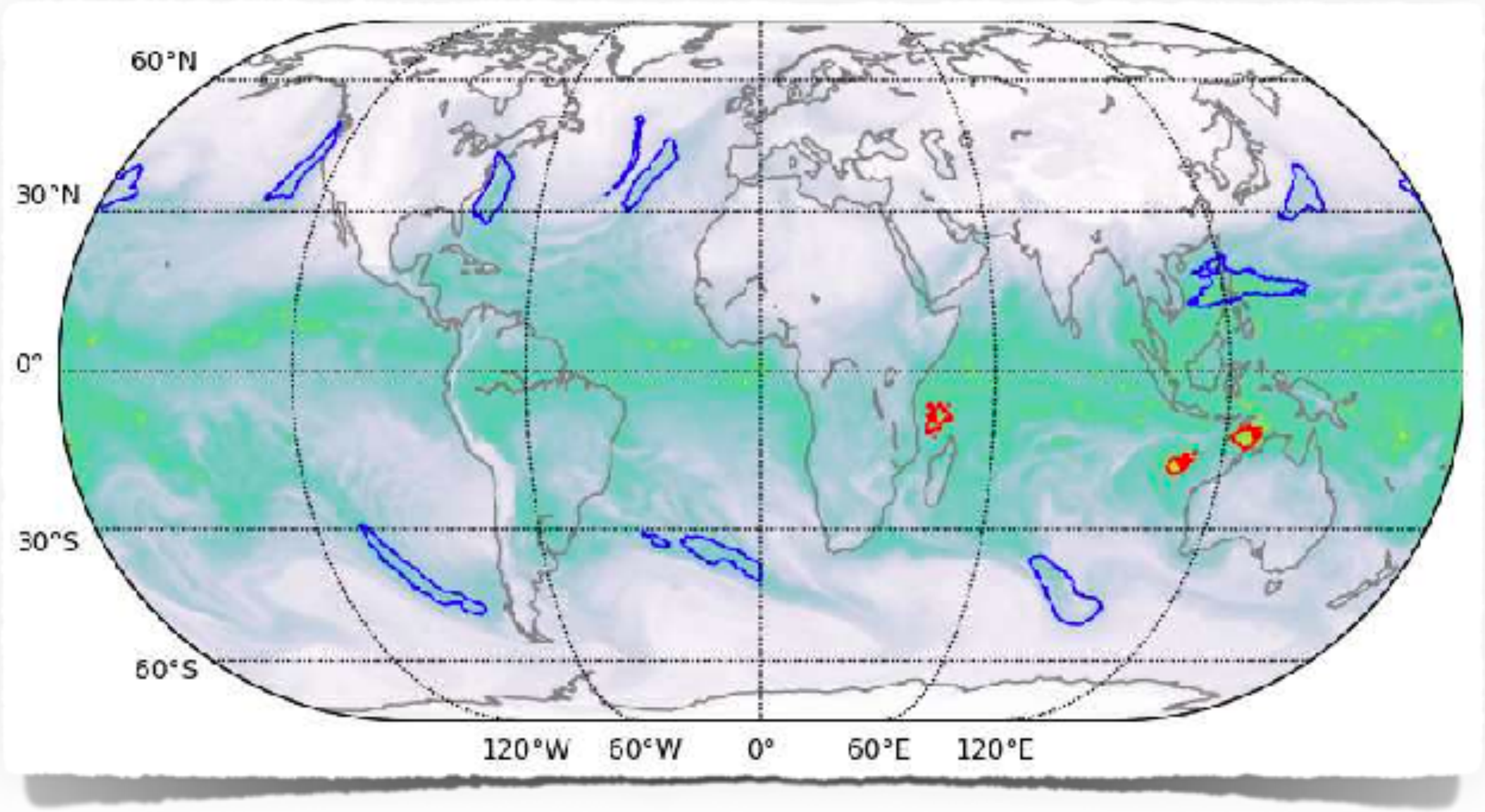
Take the mean!




Analyze spatiotemporal variability



Machine learning!




**Need to process all the data!**

# PANGEO CHALLENGE

*How can we develop* ***flexible*** *analysis tools that meet our community's* ***diverse needs*** *and* ***scale*** *to Petabyte-sized datasets?*

# WHAT IS PANGEO?

*"A community platform for Big Data geoscience"*

- Open Community
- Open Source Software
- Open Source Infrastructure

# PANGEO COMMUNITY


Lamont-Doherty Earth Observatory
COLUMBIA UNIVERSITY | EARTH INSTITUTE

NCAR
NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

Met Office

USGS

ECMWF

GEOMAR

UNIVERSITY of WASHINGTON
eScience Institute

BIDS
BERKELEY INSTITUTE FOR DATA SCIENCE

jupyter

cnes
CENTRE NATIONAL D'ÉTUDES SPATIALES

CSIRO

ANACONDA
Powered by Continuum Analytics

JUPITER™

element84

developmentSEED

Rh RHODIUM GROUP

CLIMACELL
Weather Revealed

HTTP://PANGEO.IO

# PANGEO FUNDING

NSF

EarthCube
TRANSFORMING GEOSCIENCES RESEARCH

NASA

Google Cloud Platform

aws

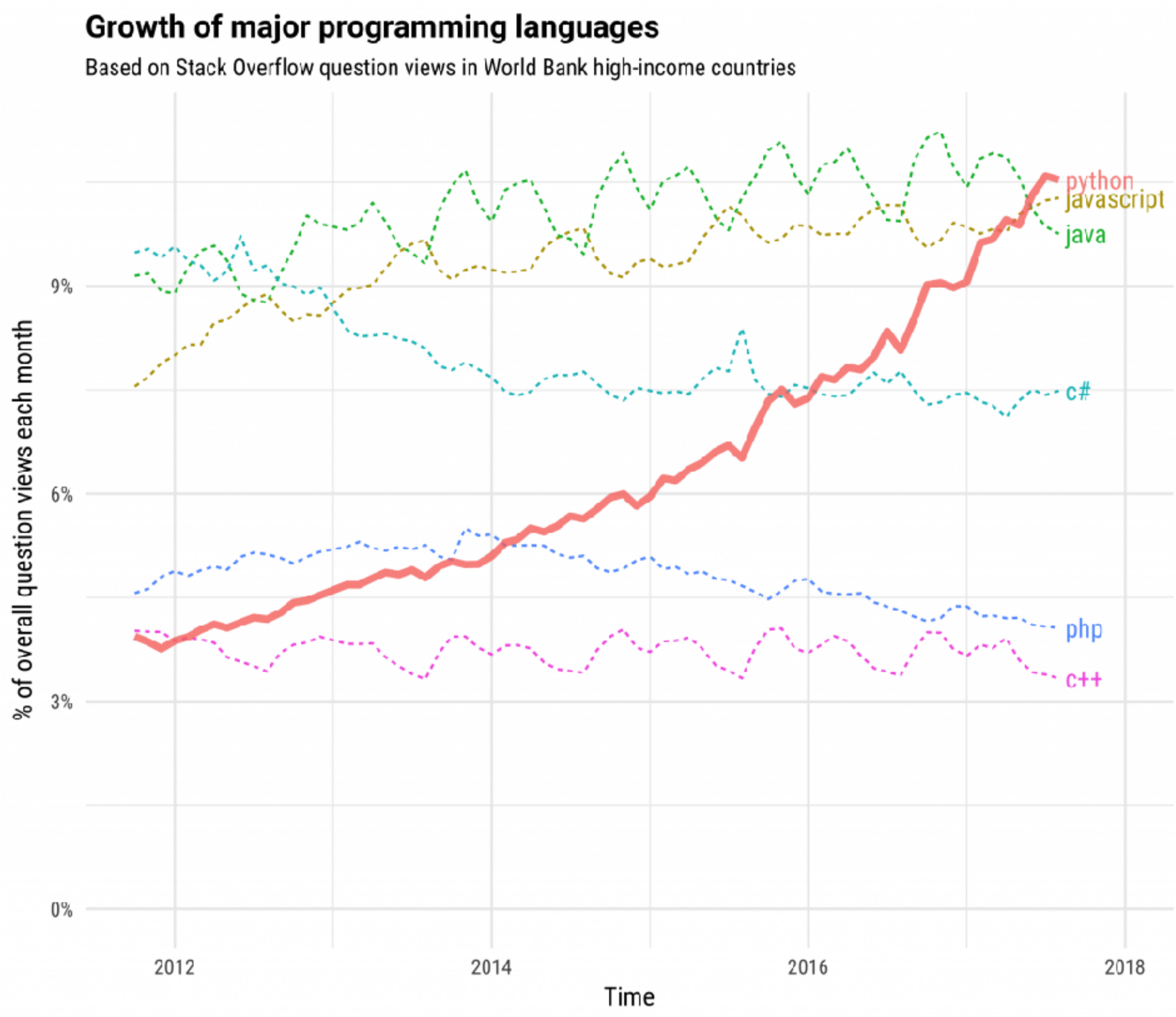
Microsoft Azure

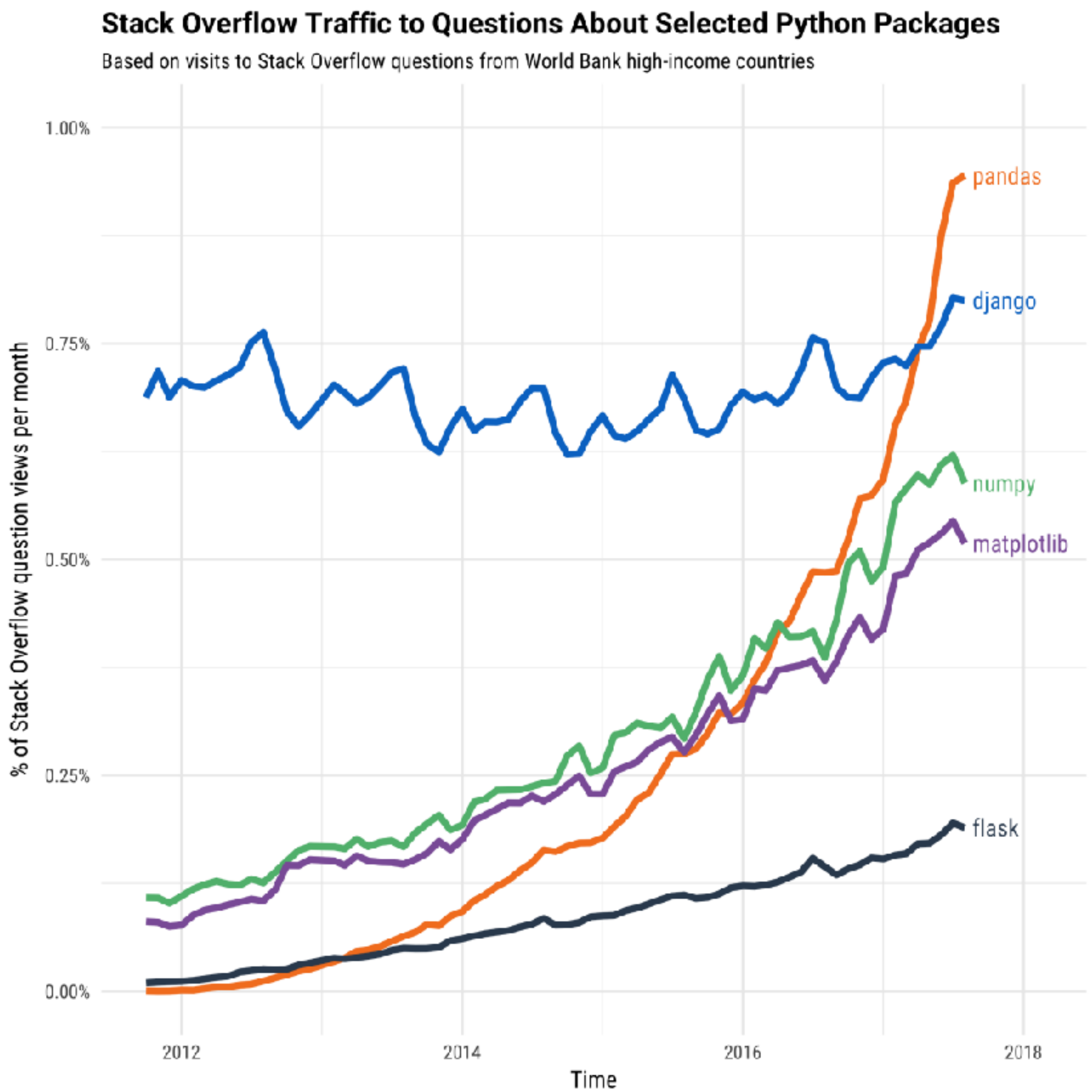
HTTP://PANGEO.IO

# PANGEO SOFTWARE

PANGEO

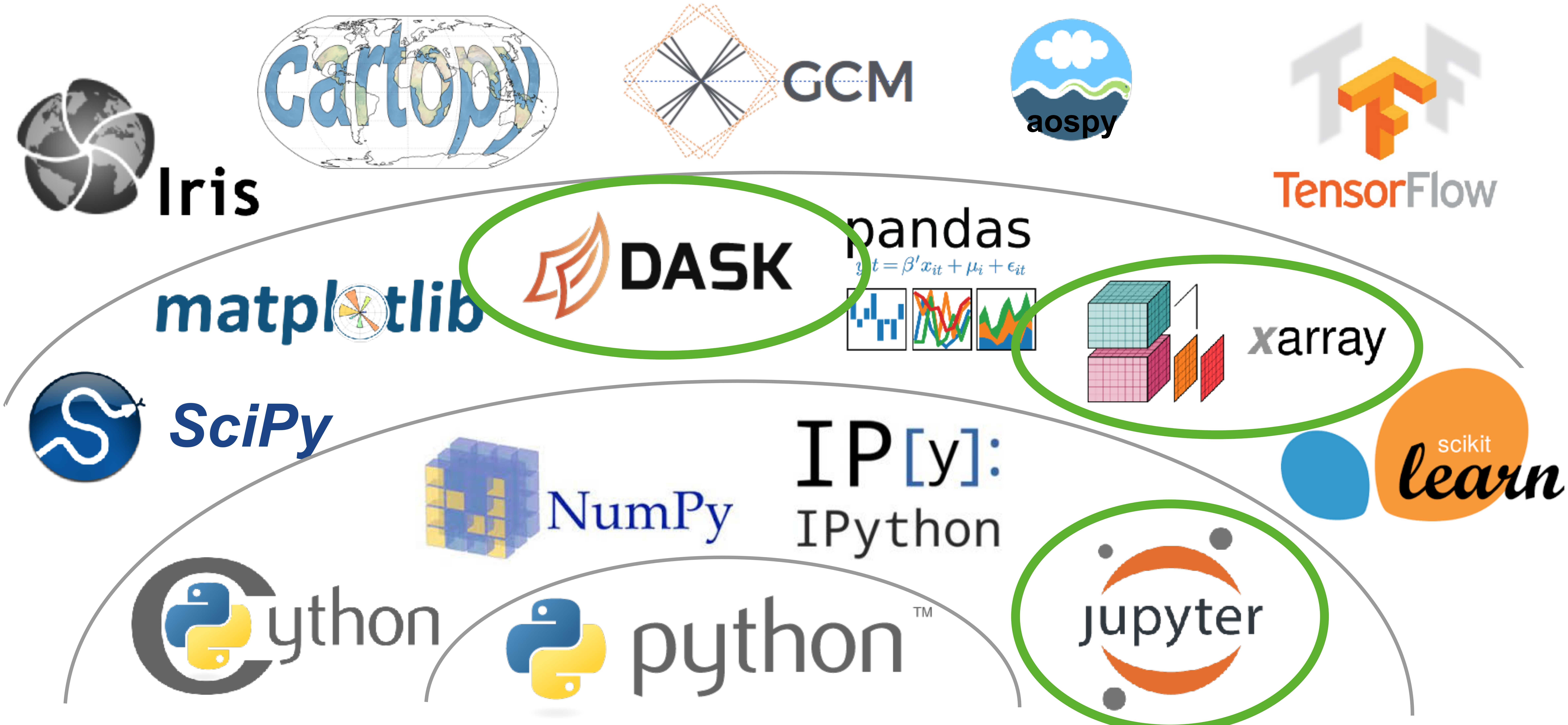
# SCIENTIFIC PYTHON FOR DATA SCIENCE




source: stackoverflow.com

# SCIENTIFIC PYTHON FOR CLIMATE

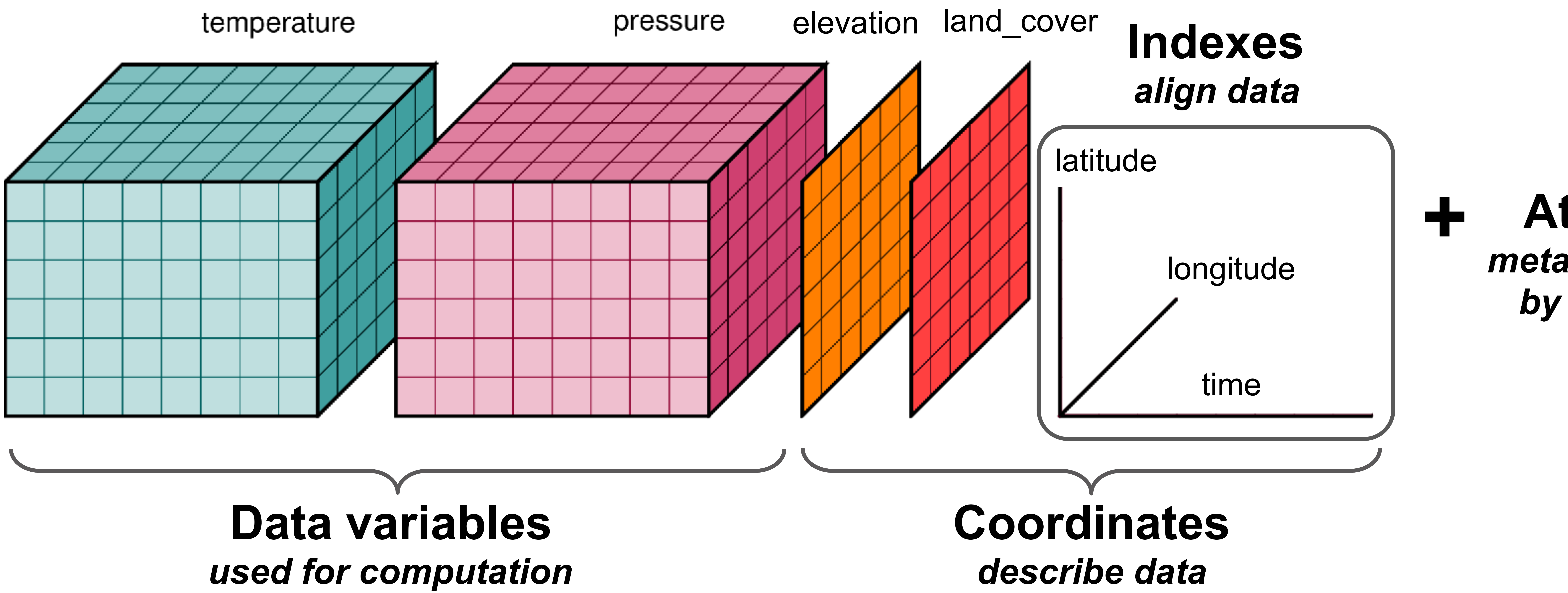



Credit: Stephan Hoyer, Jake Vanderplas (SciPy 2015)

PANGEO

# XARRAY

https://github.com/pydata/xarray

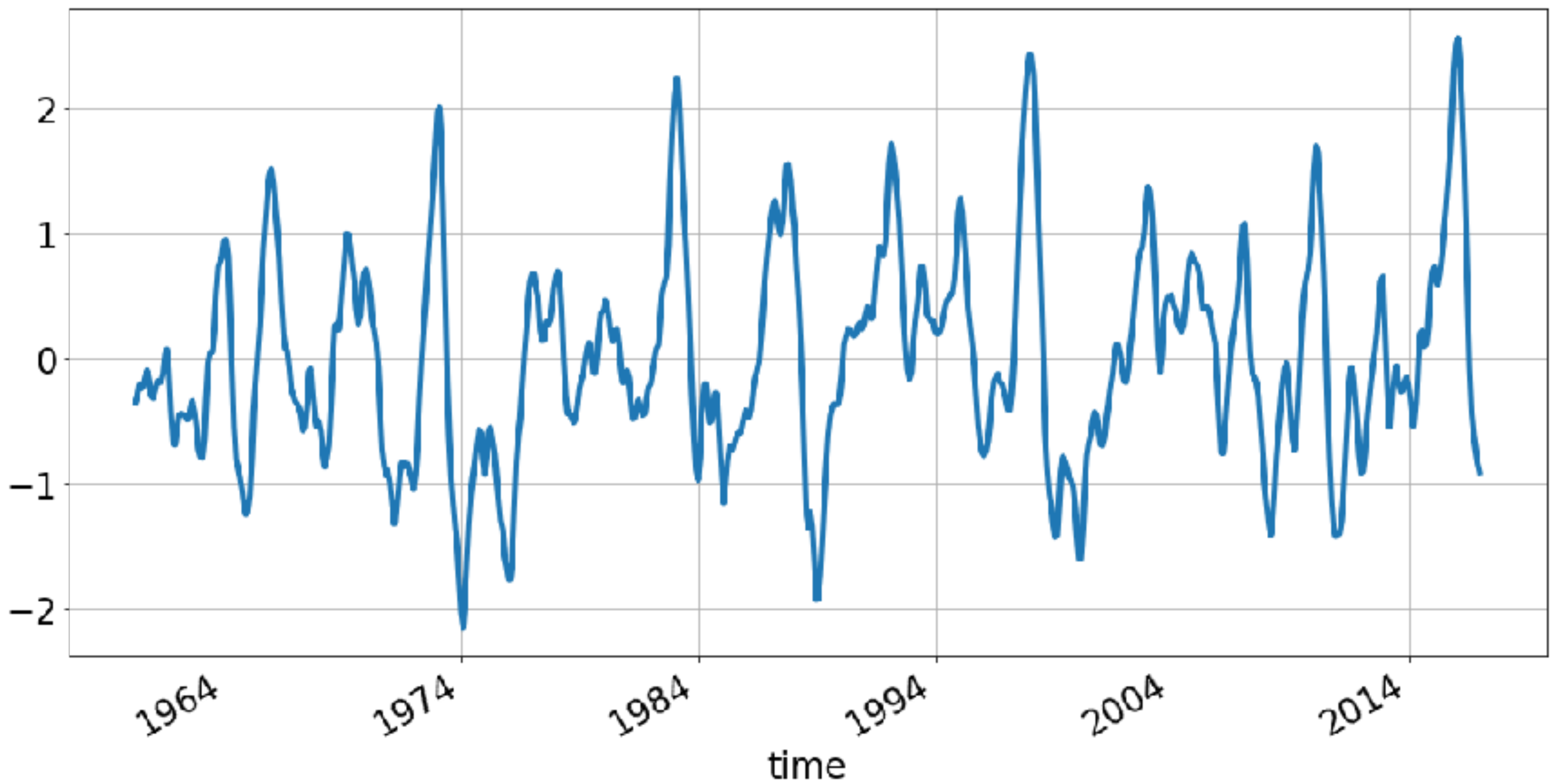



*"netCDF meets pandas.DataFrame"*

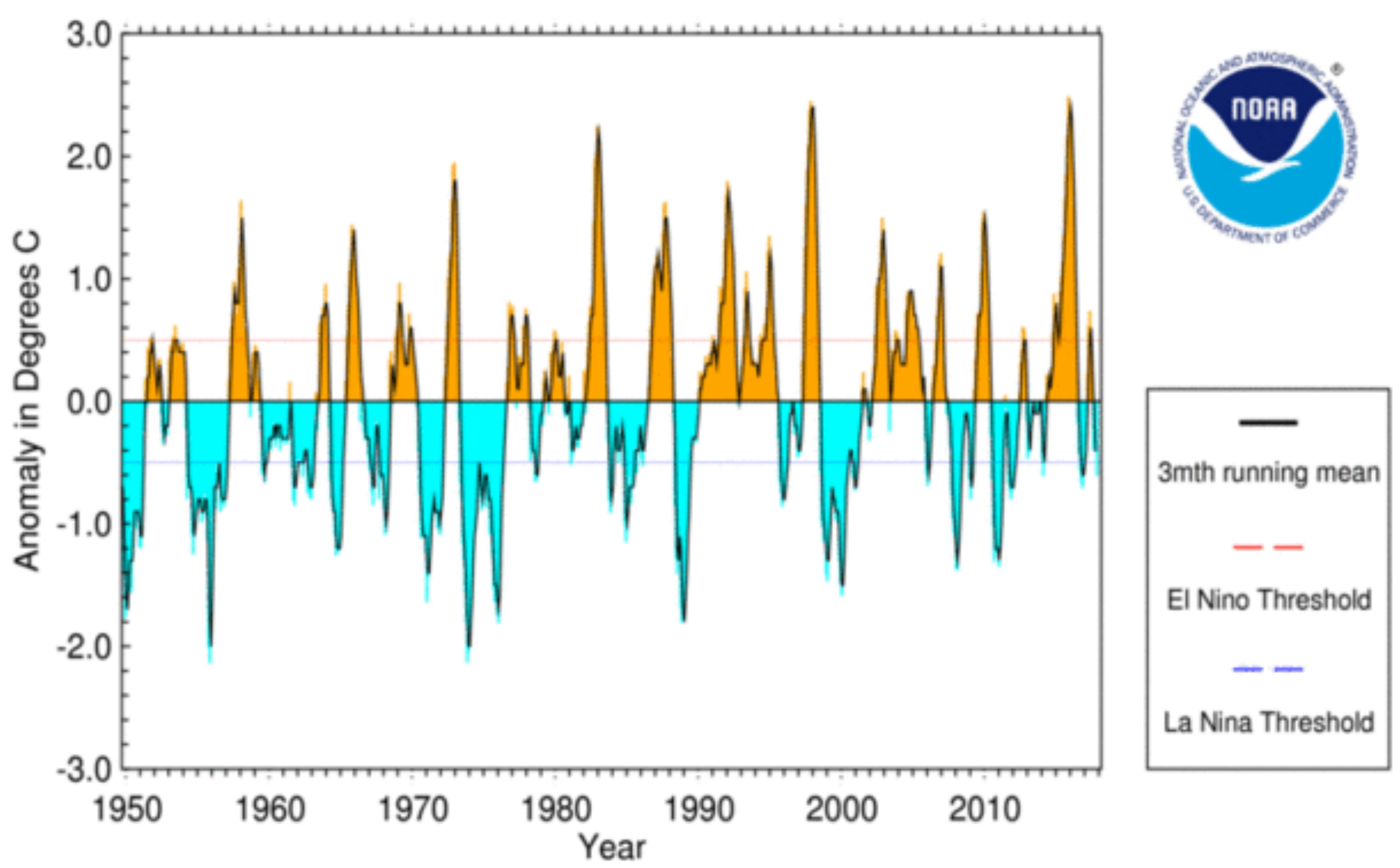
**Stephan Hoyer**
**(Google Research)**

# XARRAY: EXPRESSIVE & HIGH-LEVEL

```
sst_clim = sst.groupby('time.month').mean(dim='time')
sst_anom = sst.groupby('time.month') - sst_clim
nino34_index = (sst_anom.sel(lat=slice(-5, 5), lon=slice(190, 240))
                        .mean(dim=('lon', 'lat'))
                        .rolling(time=3).mean(dim='time'))
nino34_index.plot()
```

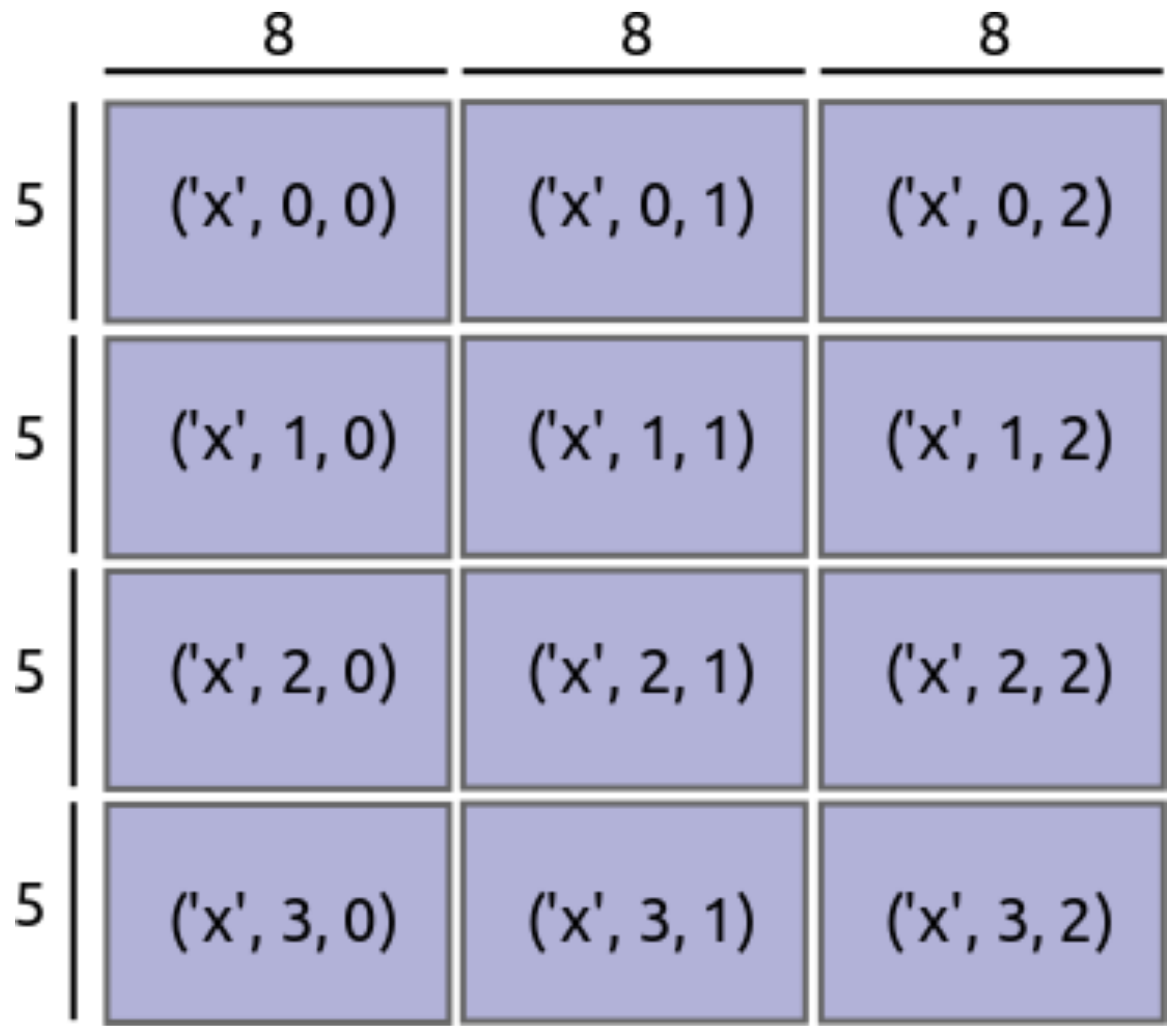



SST Anomaly in Nino 3.4 Region (5N-5S,120-170W)

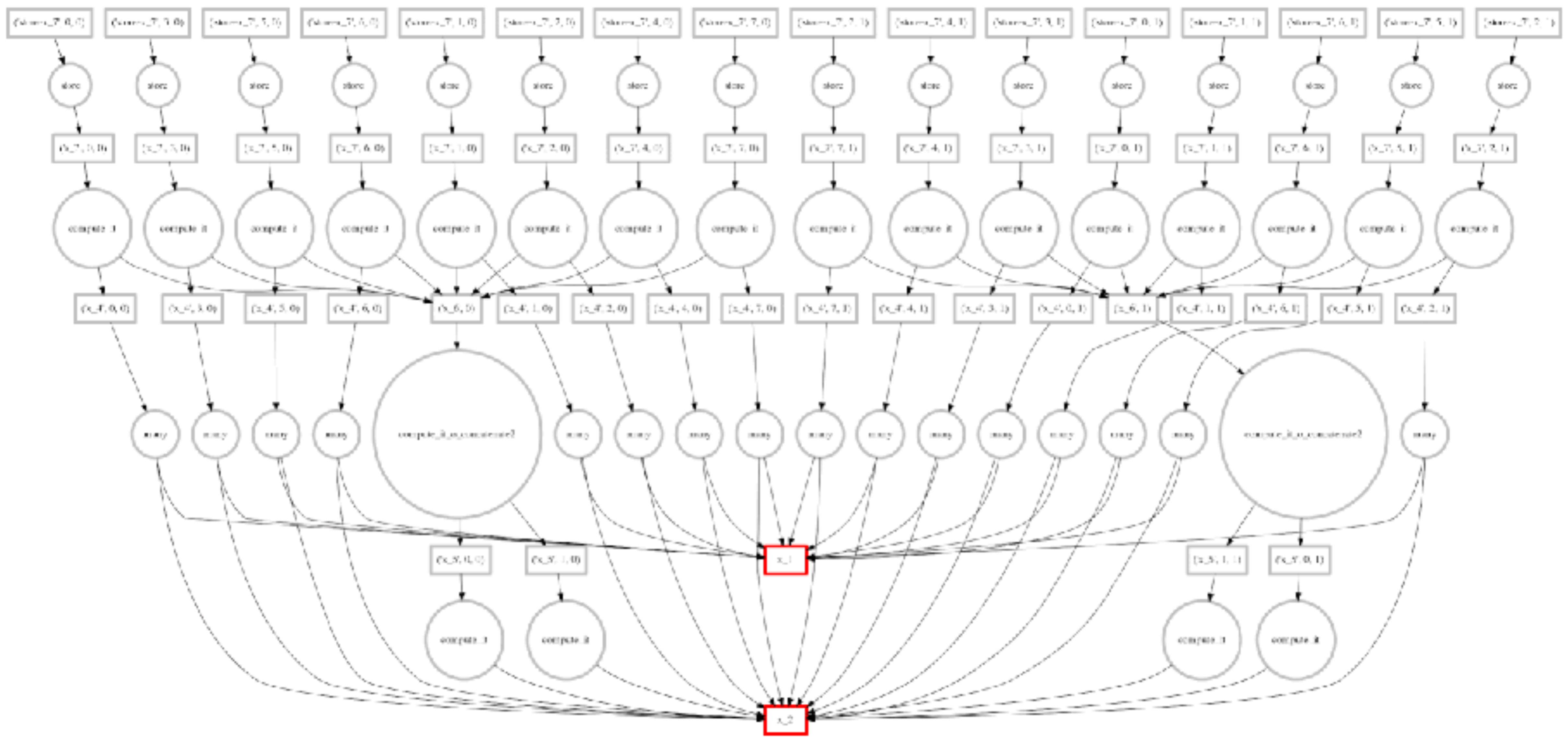


# DASK

https://github.com/dask/dask/




ND-Arrays are split into chunks that comfortably fit in memory

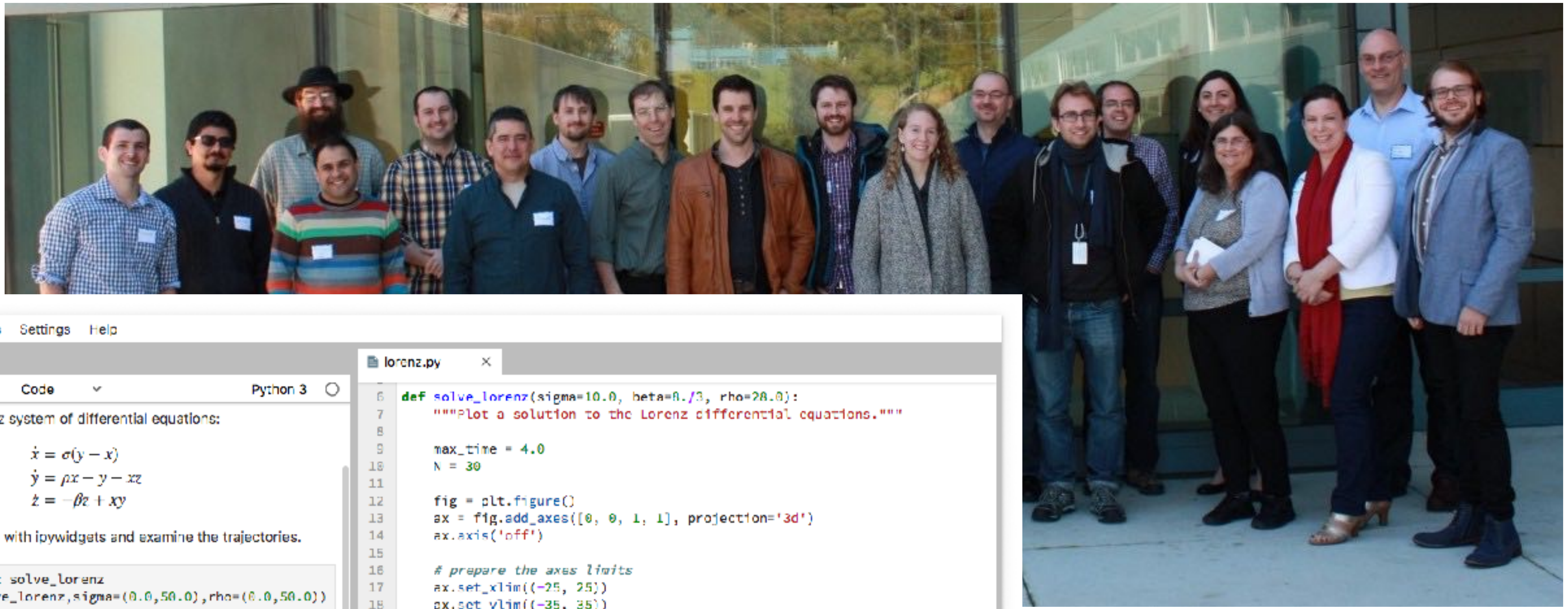


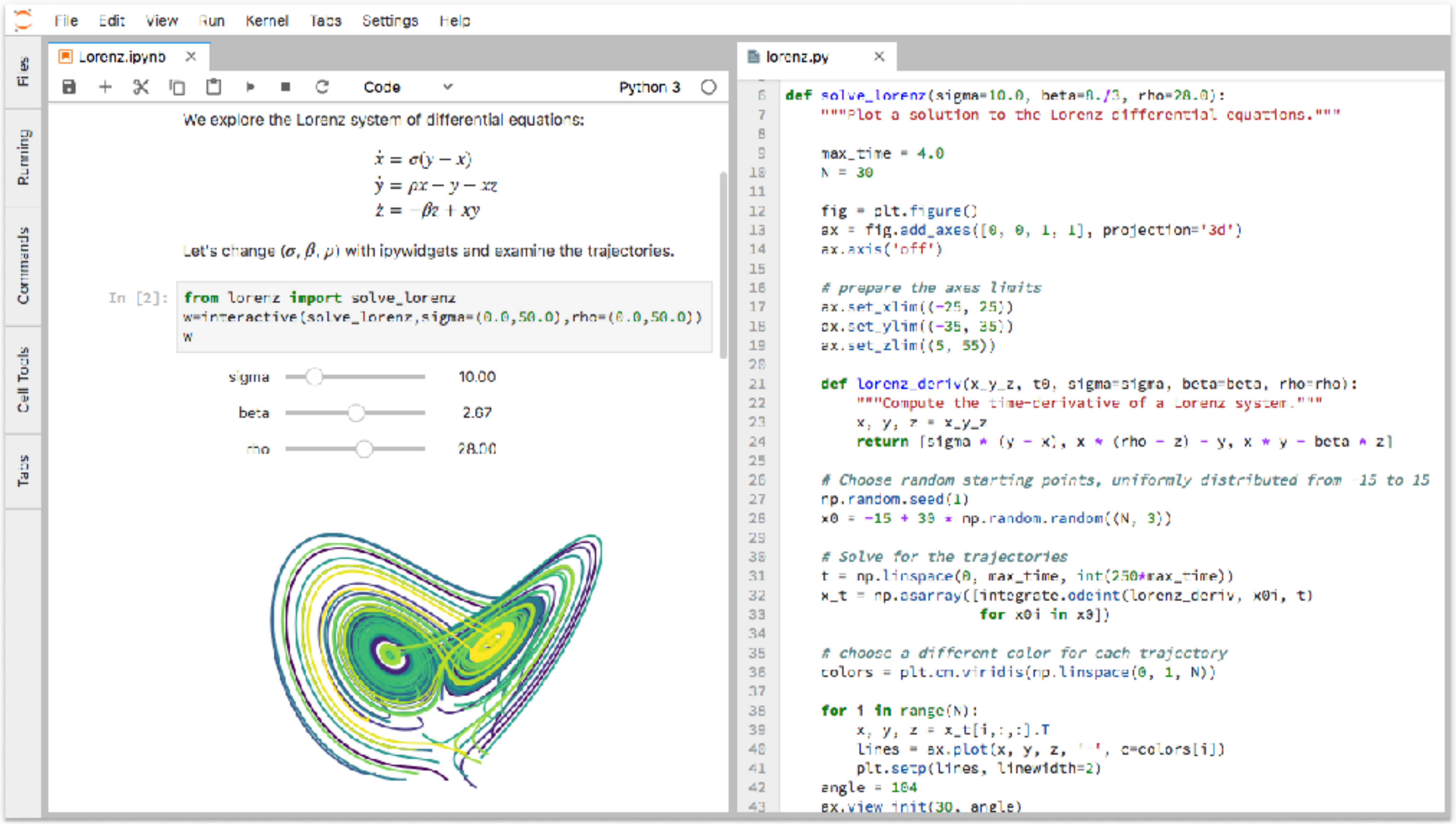
Complex computations represented as a graph of individual tasks.

Scheduler optimizes execution of graph.

**Matt Rocklin (NVIDIA)**

# JUPYTER

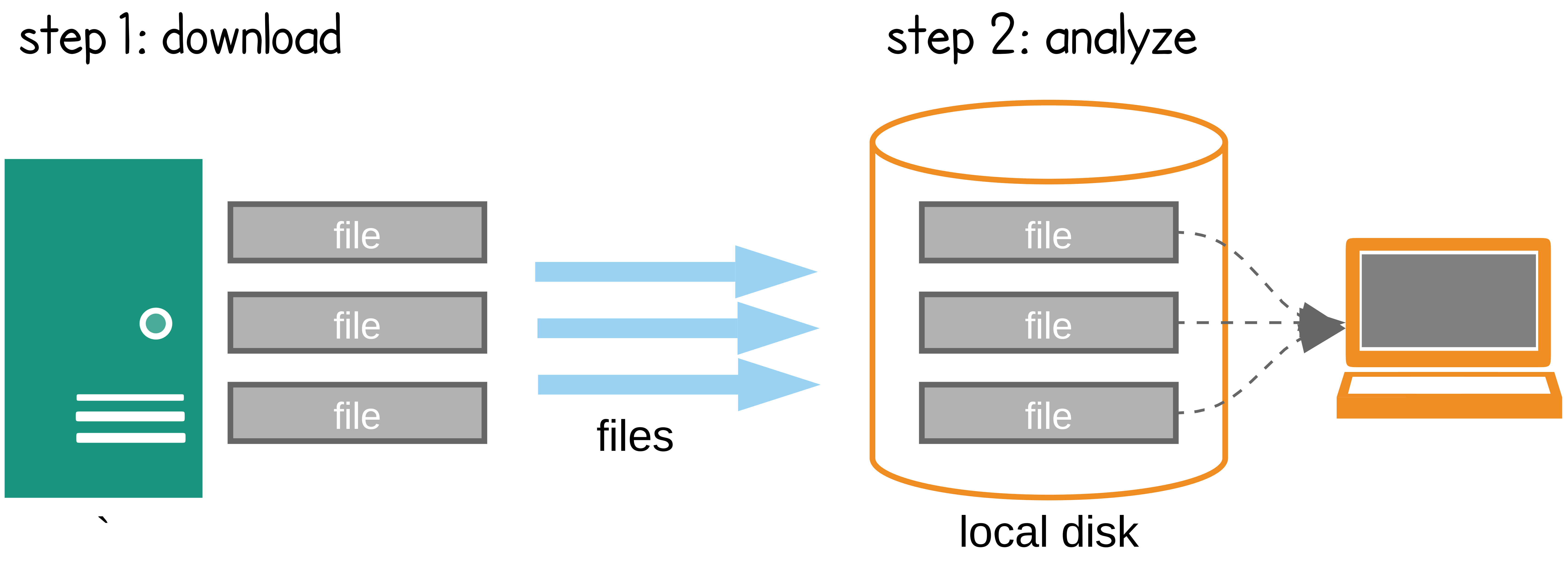





*"Project Jupyter exists to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages."*

# PANGEO INFRASTRUCTURE

# FILE-BASED APPROACH




**Data provider's responsibilities**

**End user's responsibilities**

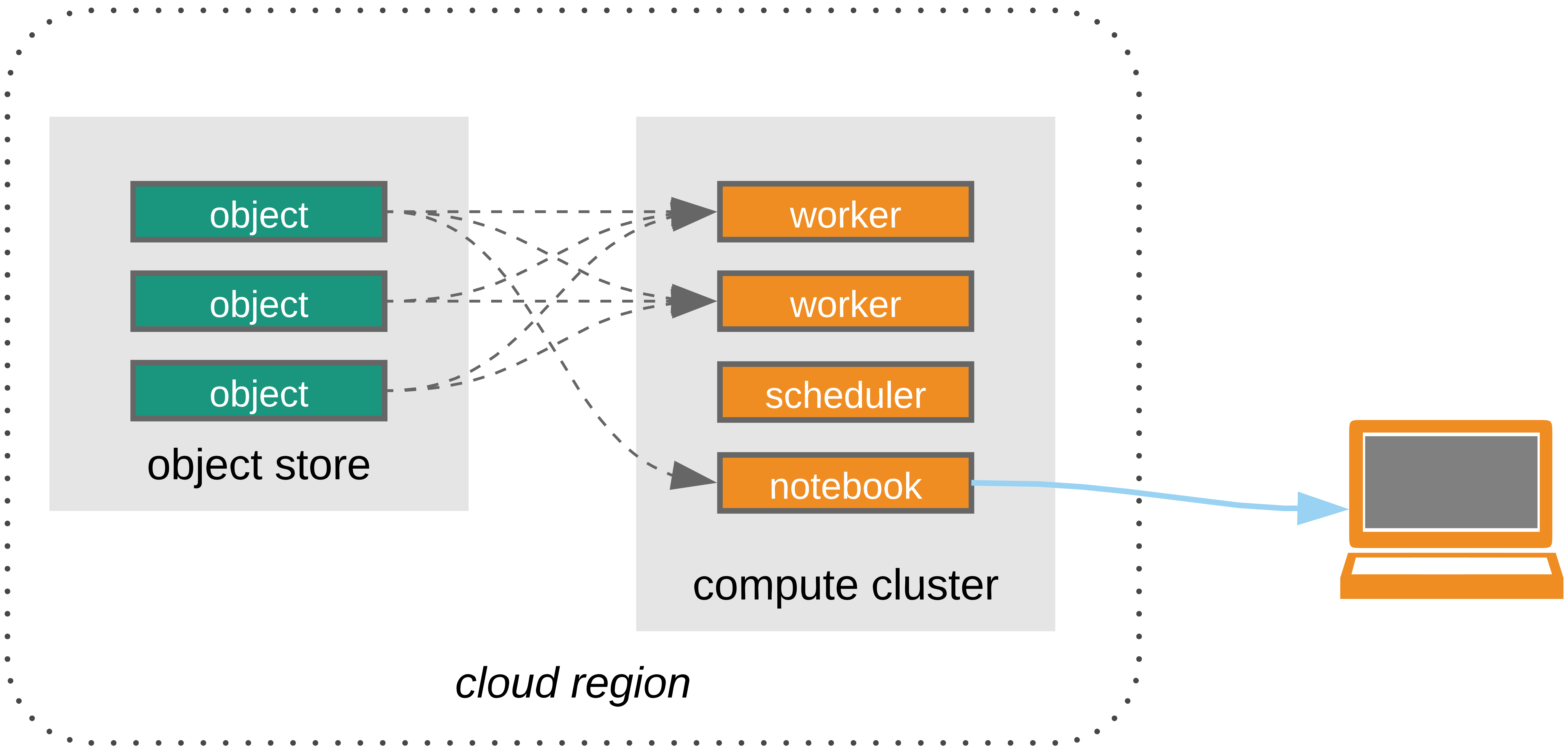
# SERVER-SIDE DATABASE


**Data provider's responsibilities**

**End user's responsibilities**

PANGEO

# CLOUD-NATIVE APPROACH

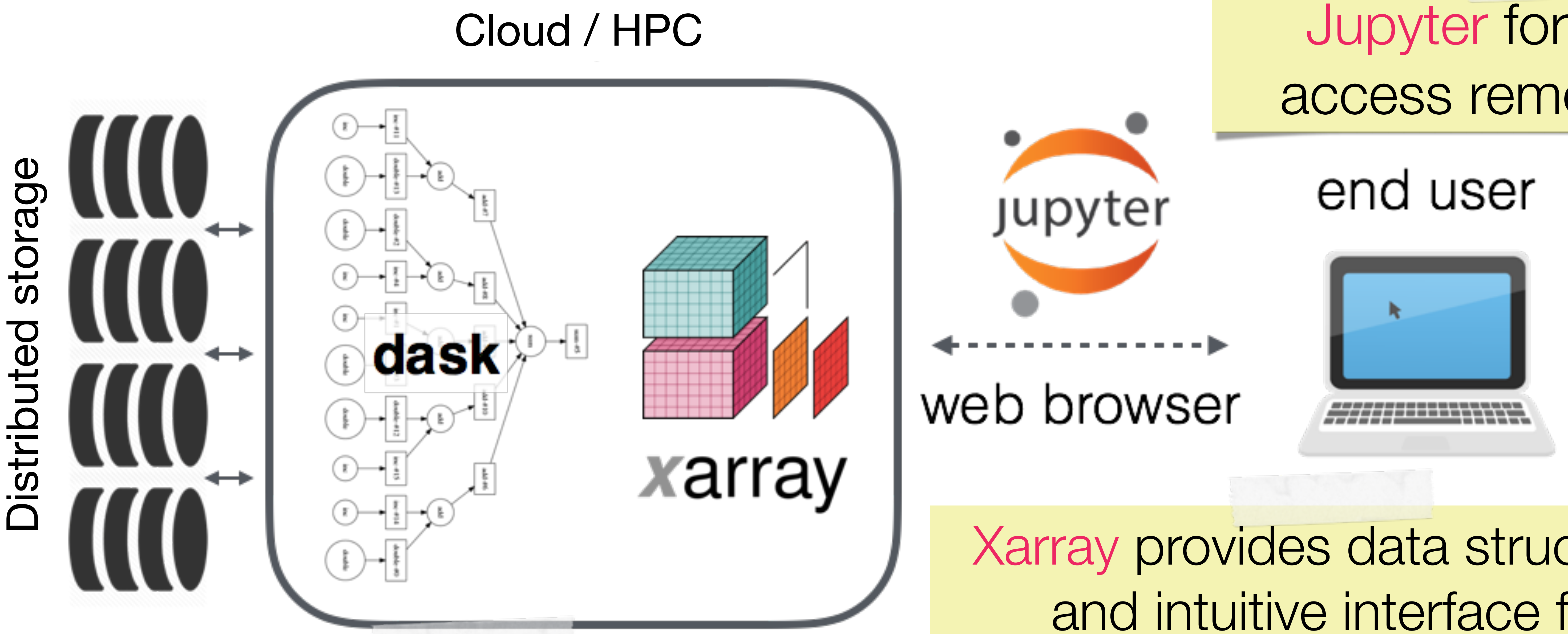



**Data provider's responsibilities**

**End user's responsibilities**

# PANGEO ARCHITECTURE

"Analysis Ready Data" stored on globally-available distributed storage.



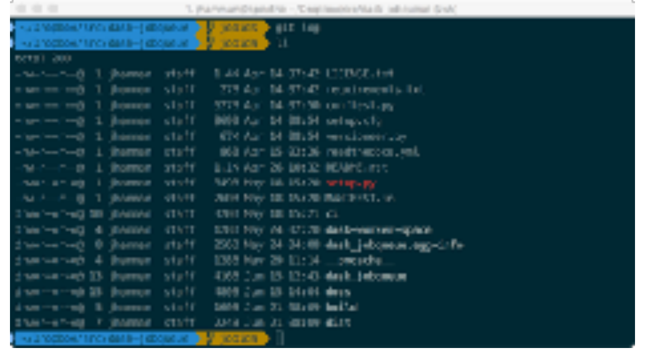

Jupyter for interactive access remote systems

Xarray provides data structures and intuitive interface for interacting with datasets

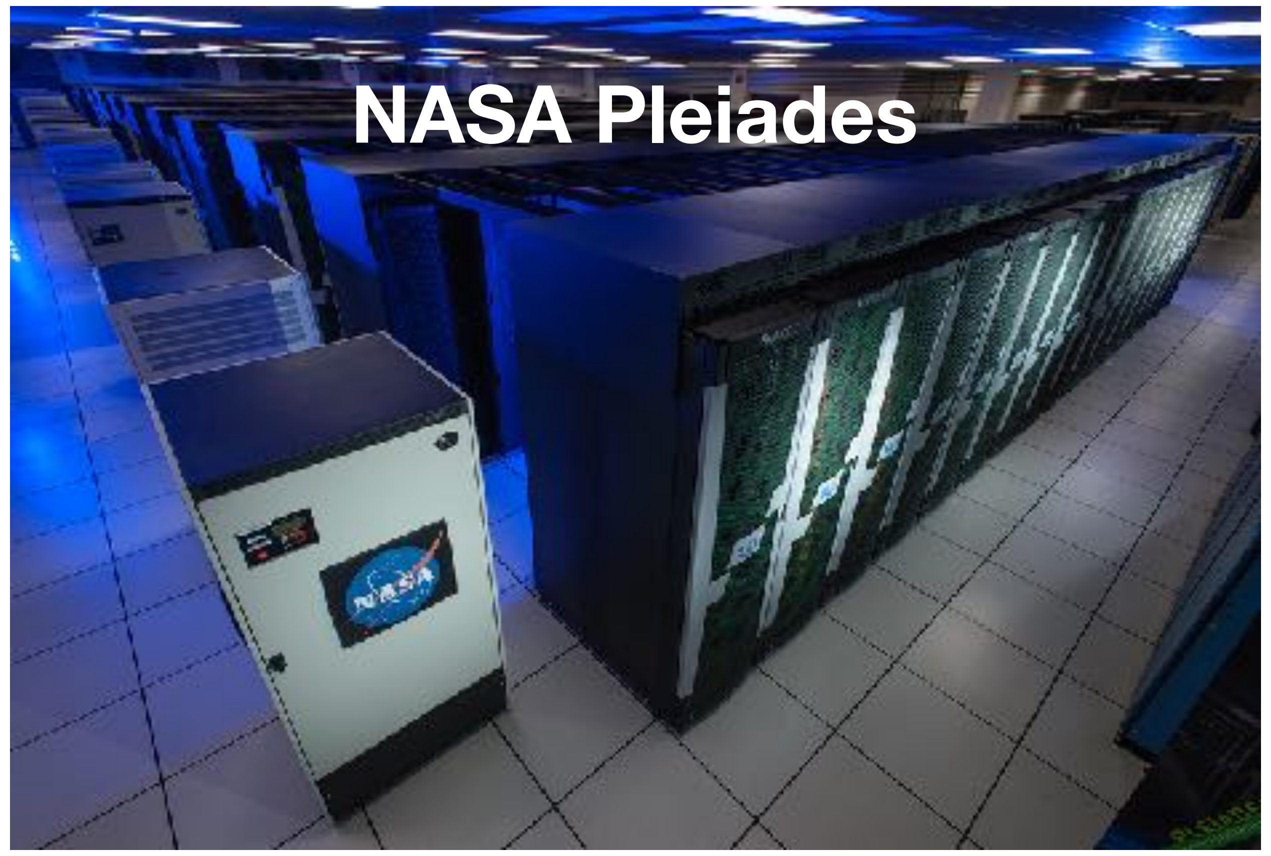
Parallel computing system allows users deploy clusters of compute nodes for data processing.

Dask tells the nodes what to do.

# BUILD YOUR OWN PANGEO

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

# PANGEO DEPLOYMENTS




NCAR Cheyenne

PANGEO.PYDATA.ORG

Over 1000 unique users since March

Google Cloud Platform

Microsoft Azure

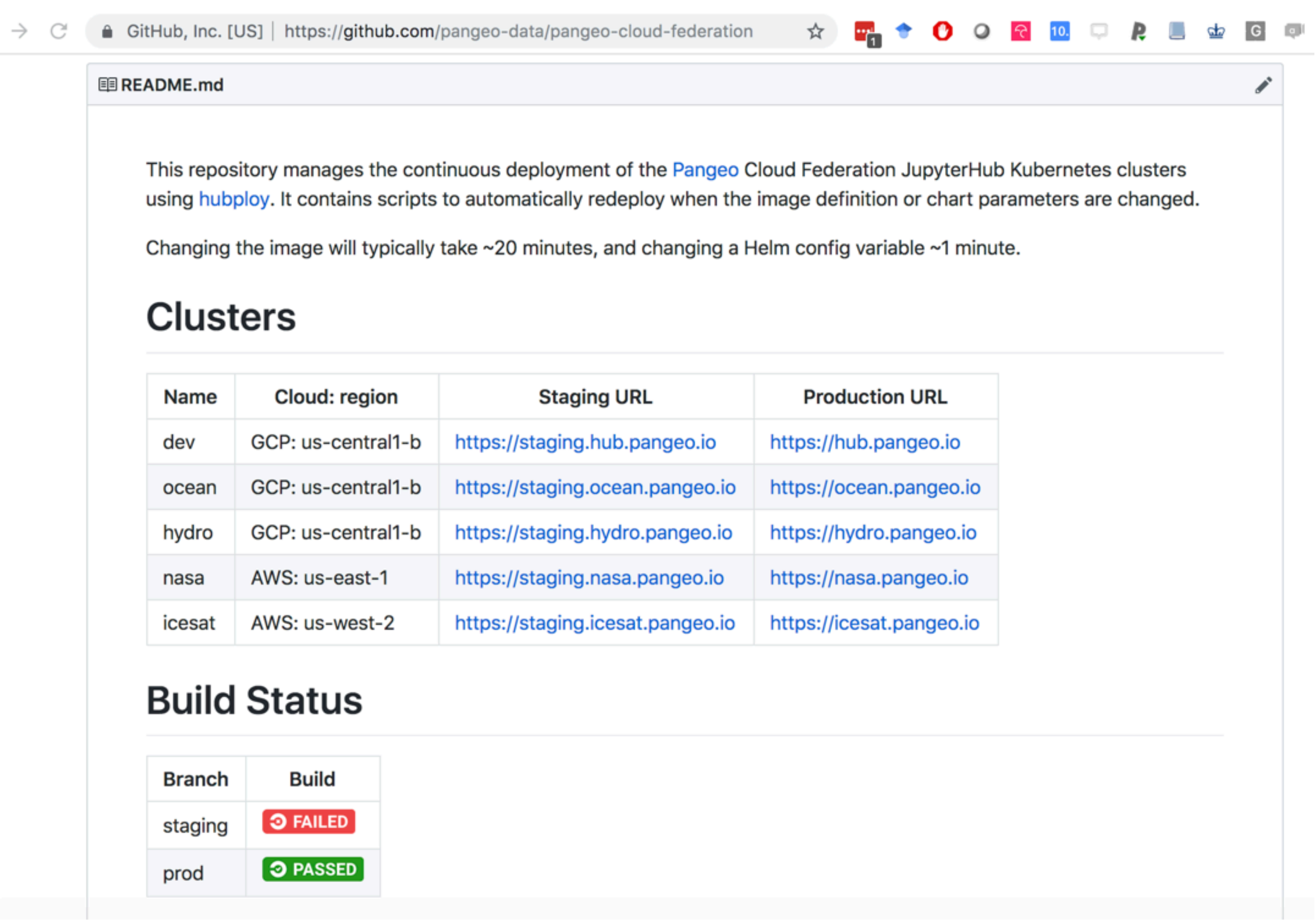
aws

HTTP://PANGEO.IO/DEPLOYMENTS.HTML

XSEDE Extreme Science and Engineering Discovery Environment · NERSC · aws · Azure

| | Government HPC | Commercial Cloud |
|---|---|---|
| Access | ✅ Available to all federally funded projects<br>❌ Available *only* to federally funded projects | ✅ Available globally to anyone with a credit card<br>❌ Authentication is not integrated with existing research infrastructure |
| Cost | ✅ Cost is hidden from researchers and billed by funding agencies<br>❌ Allocations, quotas, limits | ❌ Cost is borne by individual researchers and hidden from funding agencies<br>✅ Economics of scale, unlimited resources |
| Compute | ✅ Homogeneous, high performance nodes<br>❌ Queues, batch scheduling, ssh access<br>❌ Fixed-size compute | ✅ Flexible hardware (big, small, GPU)<br>✅ Instant provisioning of unlimited resources<br>✅ Spot market: burstable, volatile |
| Storage | ✅ Fast parallel filesystems (e.g. GPFS) | ✅ Fast object storage |

# CONTINUOUS DEPLOYMENT



GitHub, Inc. [US] | https://github.com/pangeo-data/pangeo-cloud-federation

README.md

This repository manages the continuous deployment of the Pangeo Cloud Federation JupyterHub Kubernetes clusters using hubploy. It contains scripts to automatically redeploy when the image definition or chart parameters are changed.

Changing the image will typically take ~20 minutes, and changing a Helm config variable ~1 minute.

## Clusters

| Name | Cloud: region | Staging URL | Production URL |
|---|---|---|---|
| dev | GCP: us-central1-b | https://staging.hub.pangeo.io | https://hub.pangeo.io |
| ocean | GCP: us-central1-b | https://staging.ocean.pangeo.io | https://ocean.pangeo.io |
| hydro | GCP: us-central1-b | https://staging.hydro.pangeo.io | https://hydro.pangeo.io |
| nasa | AWS: us-east-1 | https://staging.nasa.pangeo.io | https://nasa.pangeo.io |
| icesat | AWS: us-west-2 | https://staging.icesat.pangeo.io | https://icesat.pangeo.io |

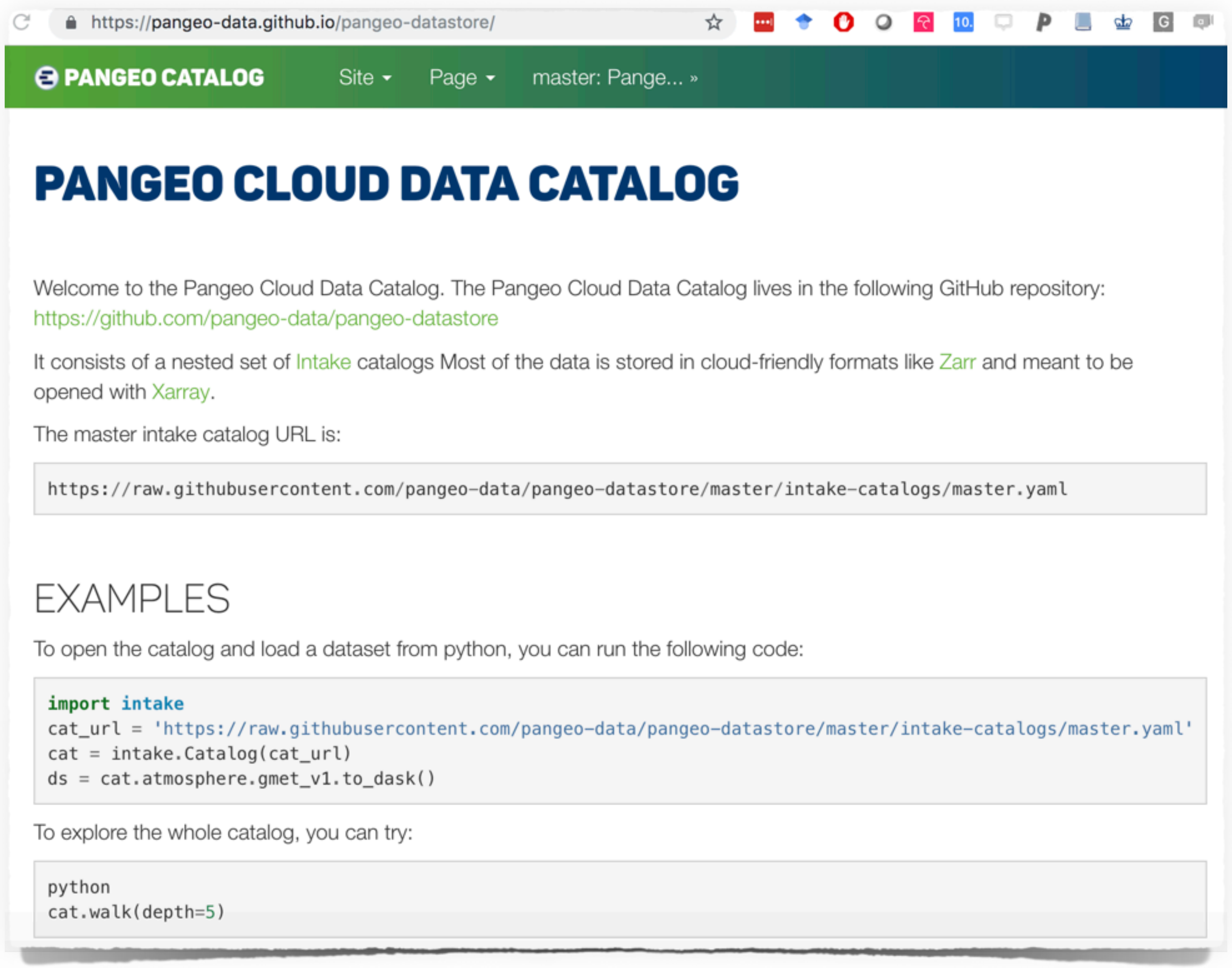
## Build Status

| Branch | Build |
|---|---|
| staging | FAILED |
| prod | PASSED |

- https://github.com/pangeo-data/pangeo-cloud-federation
- Cloud-based clusters managed with helm /kubernetes
- Deployment is completely automated via GitHub / circleci
- Resources scale elastically with demand

PANGEO

# CLOUD DATA CATALOG



https://pangeo-data.github.io/pangeo-datastore/

PANGEO CATALOG Site Page master: Pange... »

## PANGEO CLOUD DATA CATALOG

Welcome to the Pangeo Cloud Data Catalog. The Pangeo Cloud Data Catalog lives in the following GitHub repository: https://github.com/pangeo-data/pangeo-datastore

It consists of a nested set of Intake catalogs Most of the data is stored in cloud-friendly formats like Zarr and meant to be opened with Xarray.

The master intake catalog URL is:

```
https://raw.githubusercontent.com/pangeo-data/pangeo-datastore/master/intake-catalogs/master.yaml
```

### EXAMPLES

To open the catalog and load a dataset from python, you can run the following code:

```
import intake
cat_url = 'https://raw.githubusercontent.com/pangeo-data/pangeo-datastore/master/intake-catalogs/master.yaml'
cat = intake.Catalog(cat_url)
ds = cat.atmosphere.gmet_v1.to_dask()
```

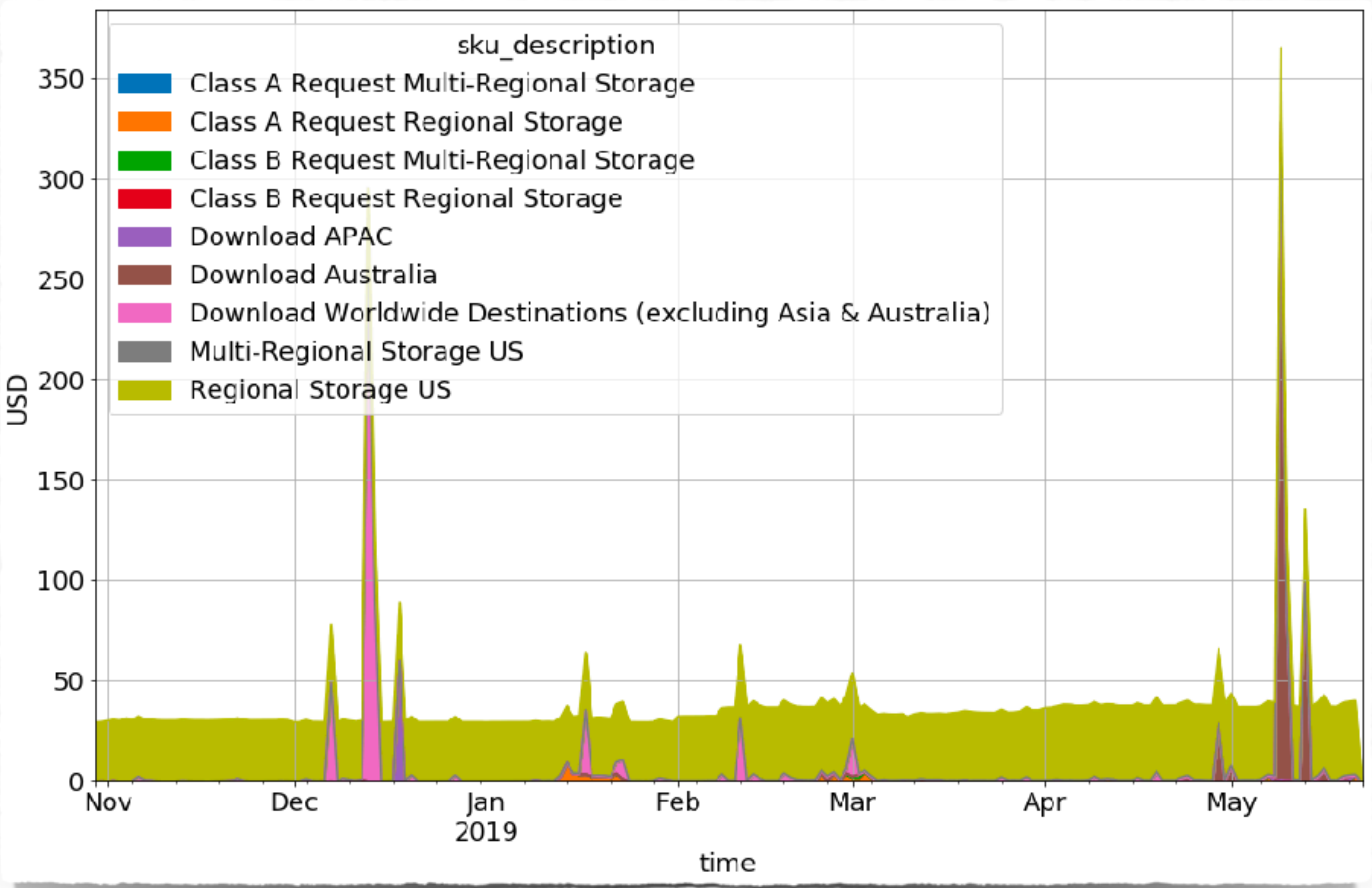
To explore the whole catalog, you can try:

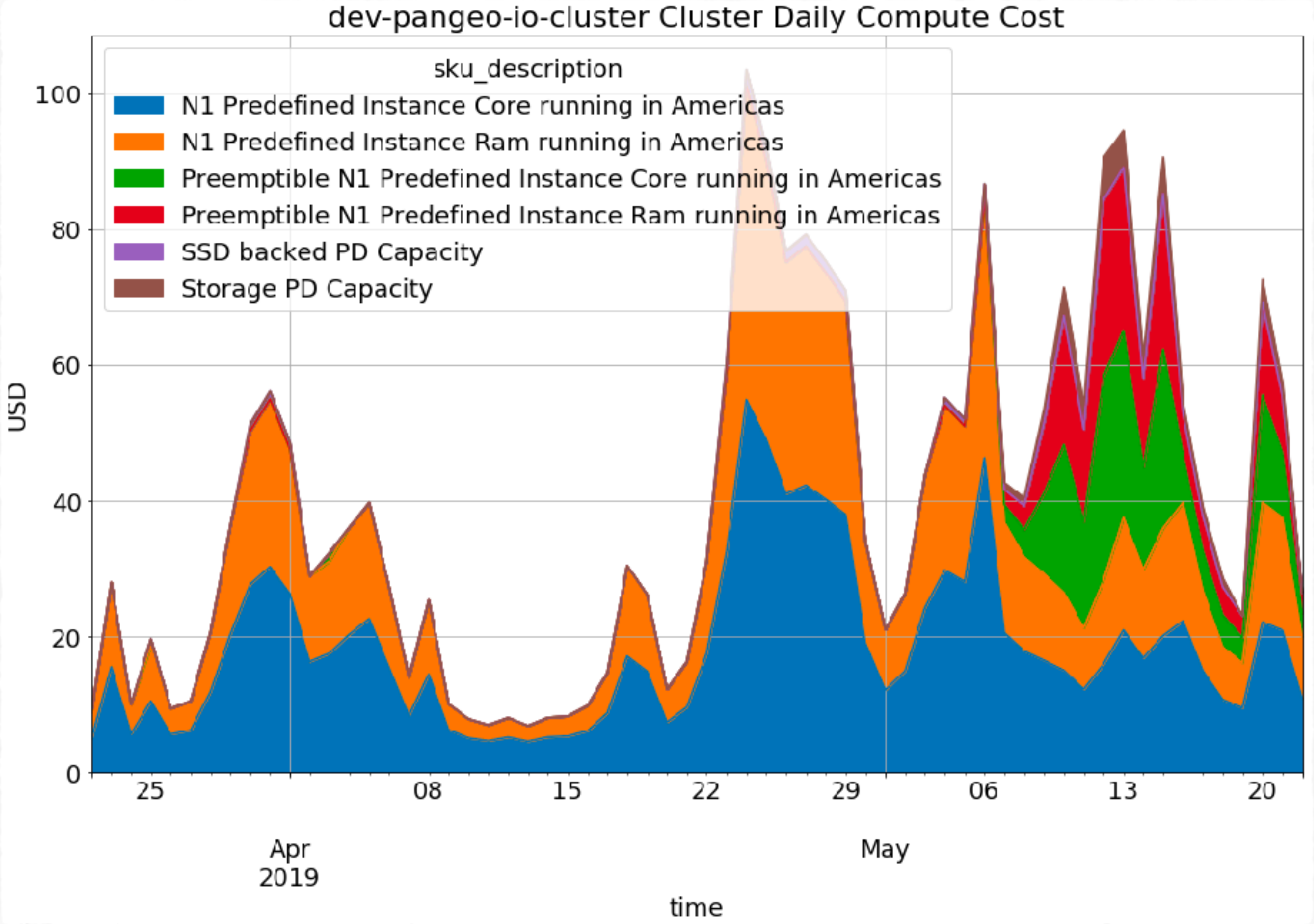
```
python
cat.walk(depth=5)
```

- https://pangeo-data.github.io/pangeo-datastore/
- Datasets stored in **zarr** format (cloud-native HDF-replacement)
- Cataloged using **intake**
- Automated testing of datasets

PANGEO

# CLOUD COSTS



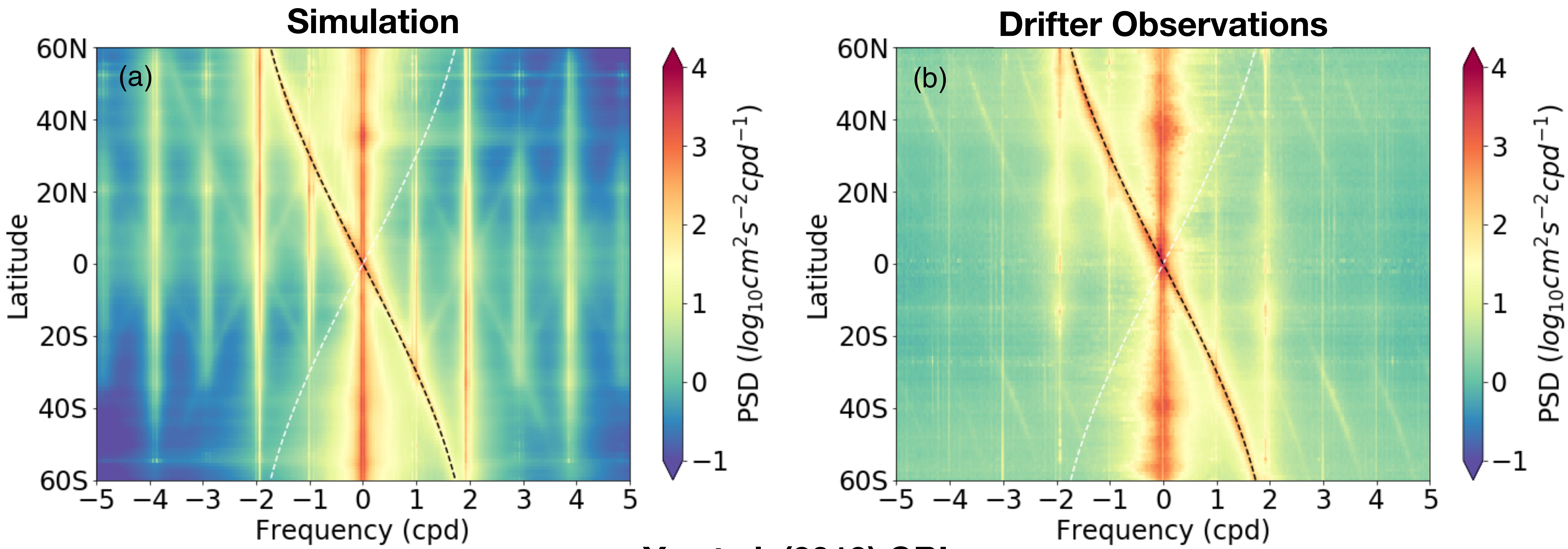



# DEMO

https://tinyurl.com/pangeo-ocean

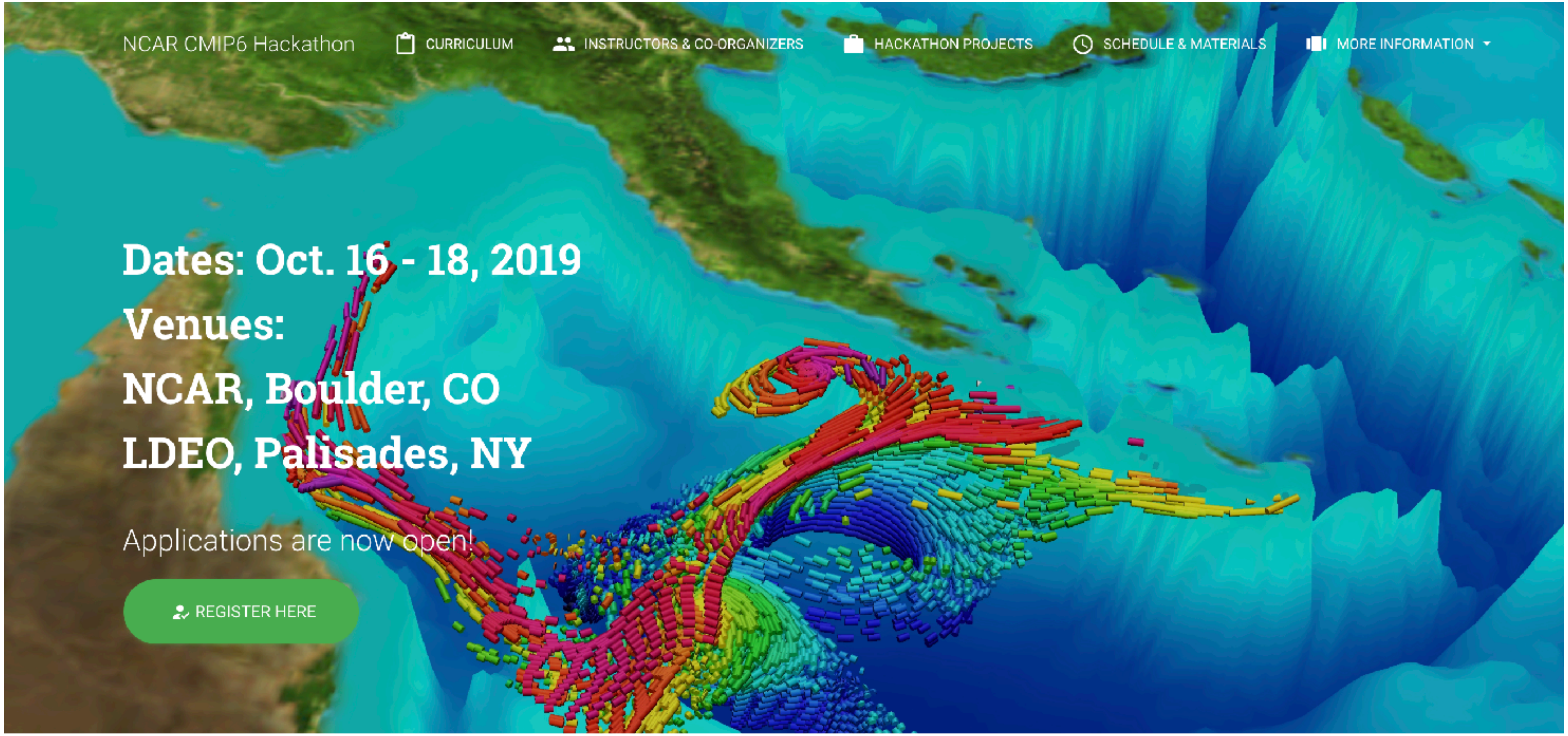
# EXAMPLE CALCULATION

Used Pangeo to process 25TB of data on HPC Cluster




**Yu et al. (2019) GRL**

# CMIP6 HACKATHON




https://cmip6hack.github.io

# HOW TO GET INVOLVED

## HTTP://PANGEO.IO

- Use and contribute to xarray, dask, zarr, jupyterhub, etc.
- Access an existing Pangeo deployment on an HPC cluster, or cloud resources (http://pangeo.io/deployments.html)
- Adapt Pangeo elements to meet your projects needs (data portals, etc.) and give feedback via github: github.com/pangeo-data/pangeo

JupyterLab

https://ocean.pangeo.io/user/rabernat/lab?redirects=1

File Edit View Run Kernel Hub Tabs Settings Help

cesm-pop-highres-ocean

Code Python 3

[5]:
```
<xarray.DataArray 'SST' (time: 14965, nlat: 2400, nlon: 3600)>
dask.array<shape=(14965, 2400, 3600), dtype=float32, chunksize=(1, 2400, 3600)>
Coordinates:
  * time     (time) float64 1.679e+04 1.679e+04 ... 3.175e+04 3.176e+04
  * nlon     (nlon) int64 0 1 2 3 4 5 6 7 ... 3593 3594 3595 3596 3597 3598 3599
  * nlat     (nlat) int64 0 1 2 3 4 5 6 7 ... 2393 2394 2395 2396 2397 2398 2399
    lon      (nlat, nlon) float64 dask.array<shape=(2400, 3600), chunksize=(2400, 3600)>
    lat      (nlat, nlon) float64 dask.array<shape=(2400, 3600), chunksize=(2400, 3600)>
Attributes:
    cell_methods:  time: mean
    grid_loc:      2110
    long_name:     Surface Potential Temperature
    units:         degC
```

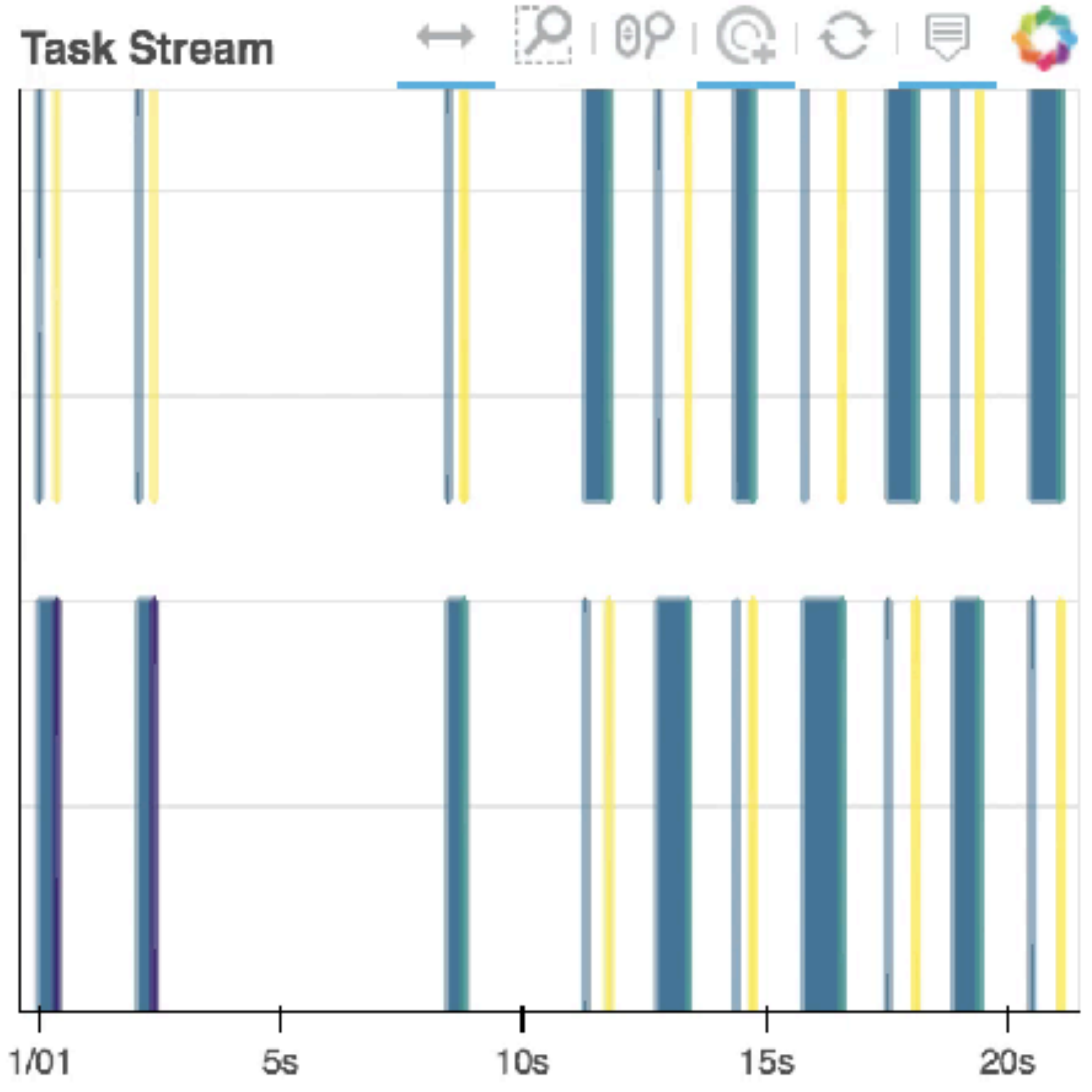
[ ]:
```
%%output holomap='scrubber' fps=1

sst_ds = hv.Dataset(sst, kdims=['time', 'nlon', 'nlat'])
hv_sst = sst_ds.to(hv.Image, kdims=["nlon", "nlat"], dynamic=True)

%opts Image [width=700 height=400] (cmap='magma')
regrid(hv_sst, precompute=True)
```

[ ]:

Dask Task Stream




Dask Progress

Progress -- total: 0, in-memory: 0, processing: 0, erred: 0