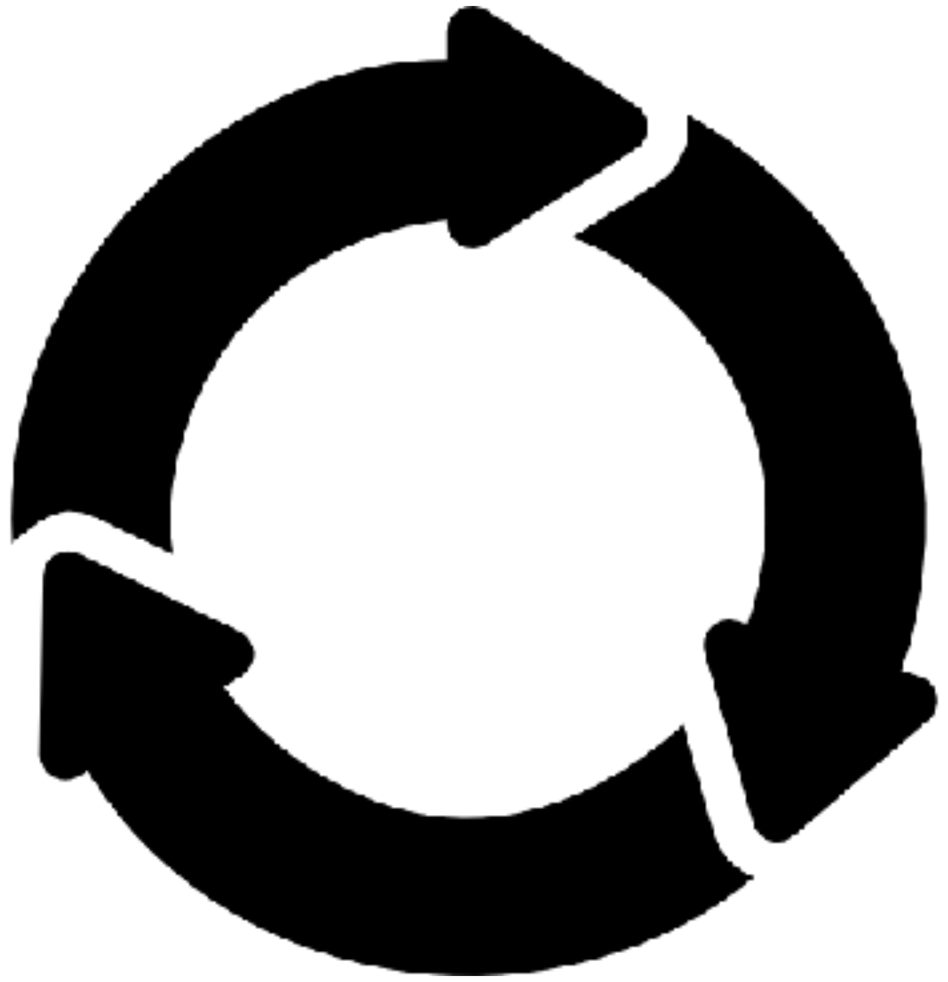
PANGEO

A COMMUNITY-DRIVEN EFFORT FOR
BIG DATA GEOSCIENCE

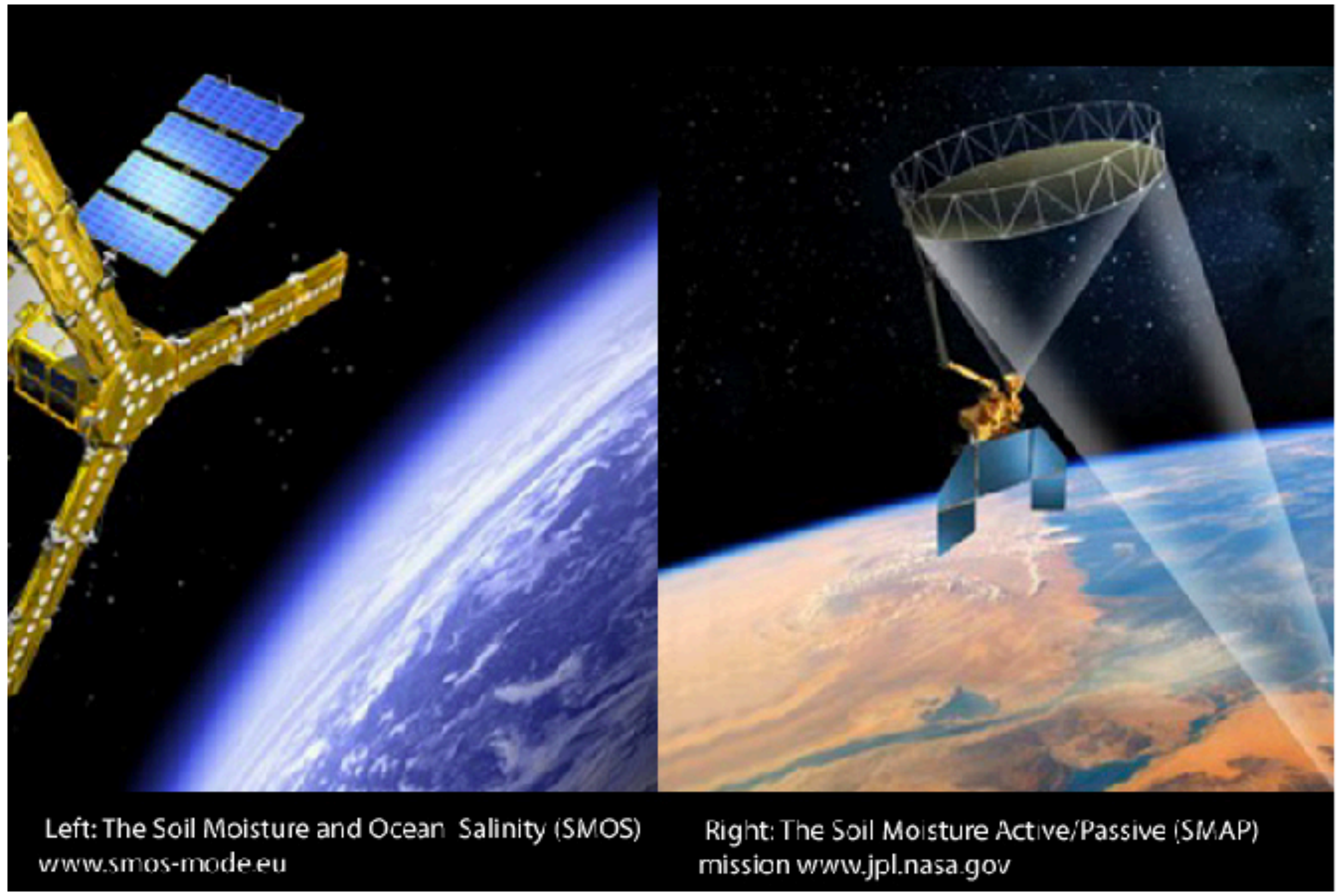
WHAT DRIVES PROGRESS IN GEOSCIENCE?

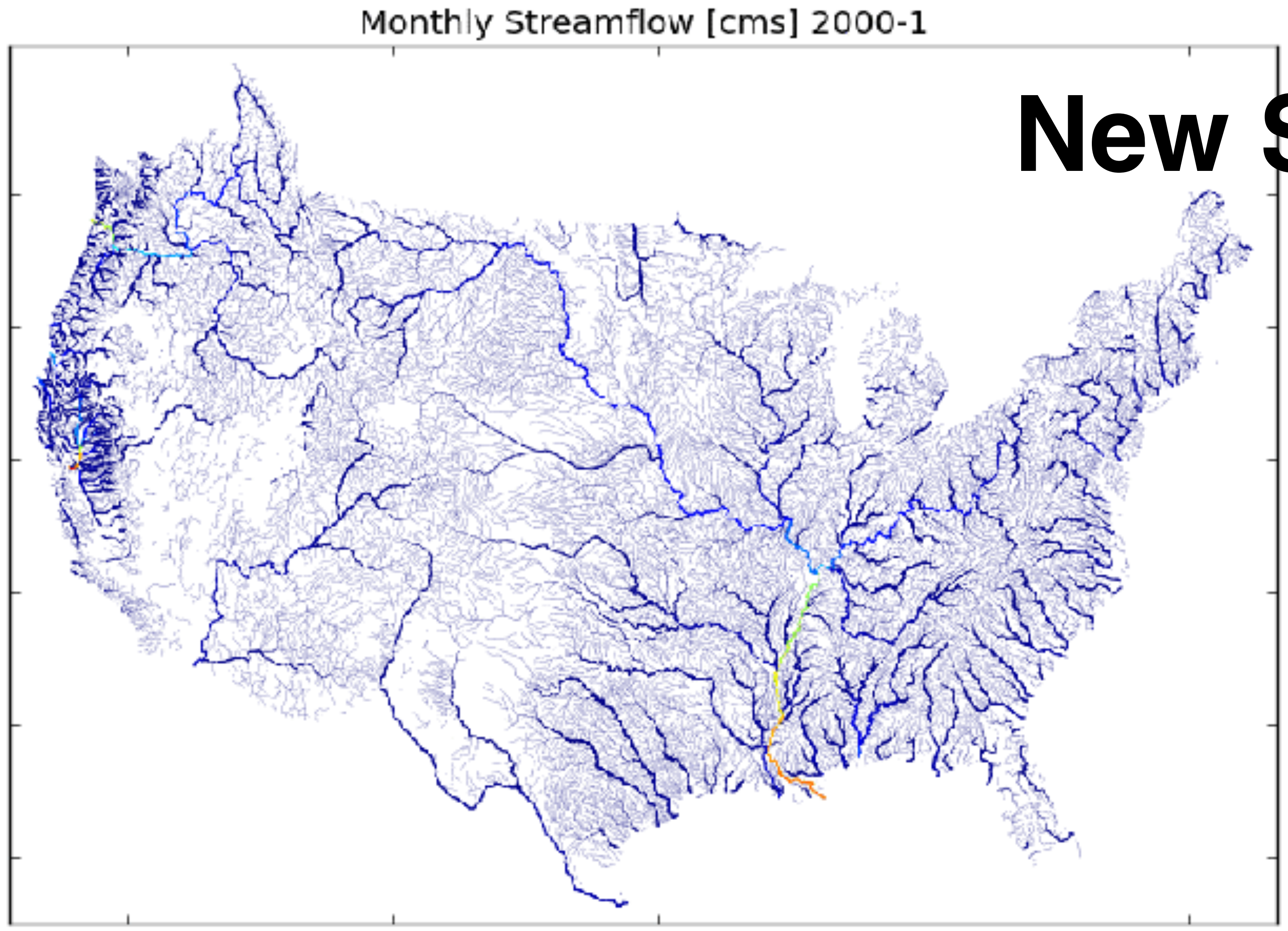
New Ideas

$$q_{liq,z}^{soil} = \begin{cases} q_{rain} - q_{ix} - q_{sx} & z = 0 \\ -K^{soil}\dfrac{\partial \psi}{\partial z} + K^{soil} & z > 0 \end{cases}$$



New Observations





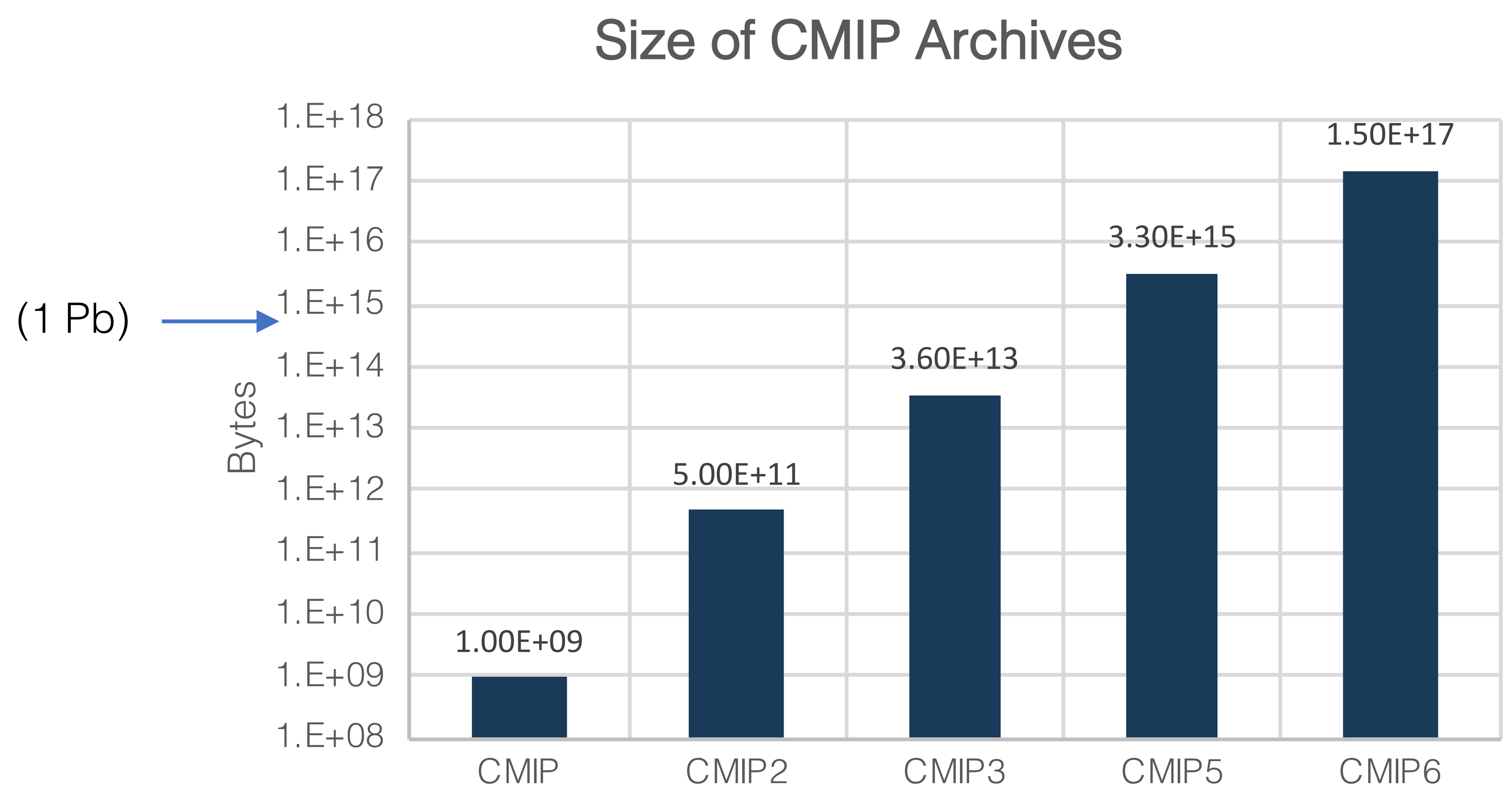
New Simulations

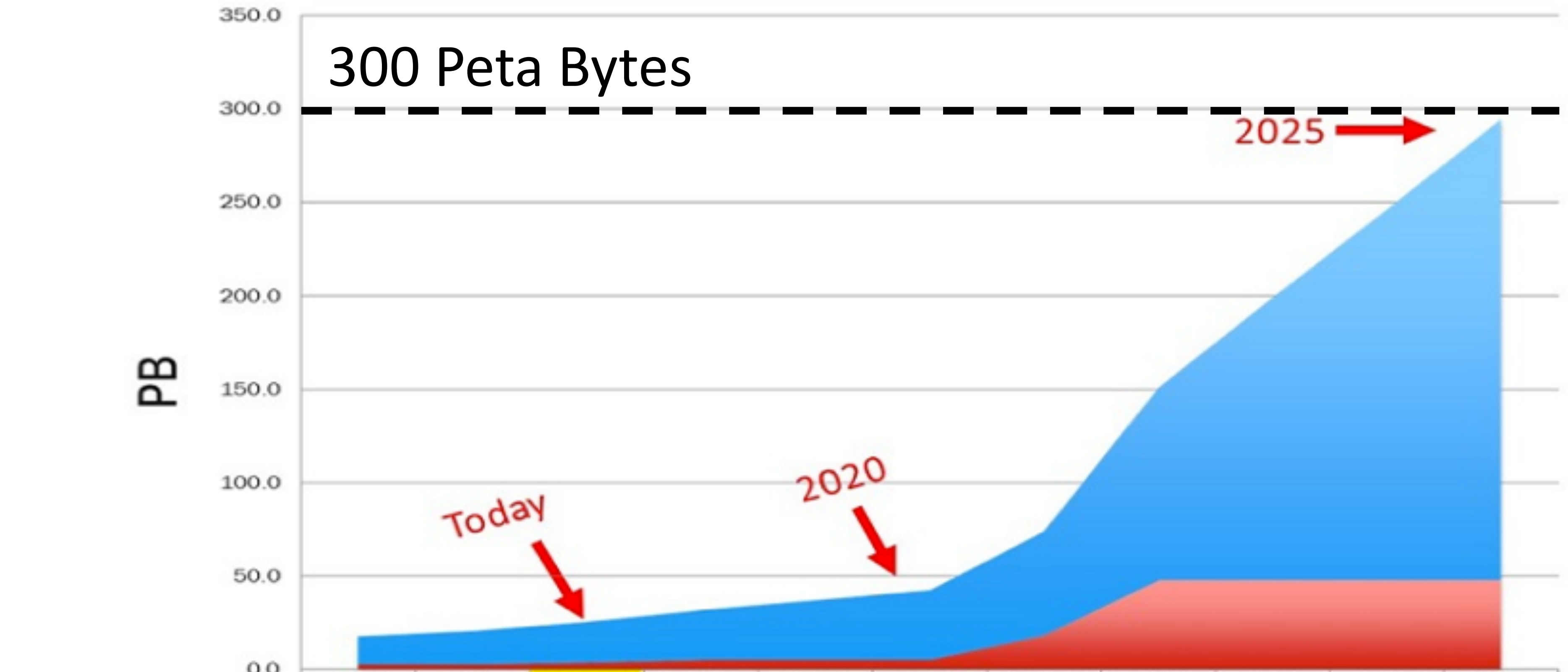
REDUCING TIME TO SCIENCE WITH PANGEO (AN OUTLINE)

- Familiar software ecosystem
- Data-proximate deployments
- Scalability
- Emphasis on next-generation data storage formats for the geosciences
- Demonstration

THE BIG DATA GEOSCIENCE ERA IS NOW

- The geosciences are facing a data volume crisis
- From Earth System Models:
  - Higher resolution
  - More process representation
  - Larger ensembles
  - On track for exabytes by CMIP7





	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

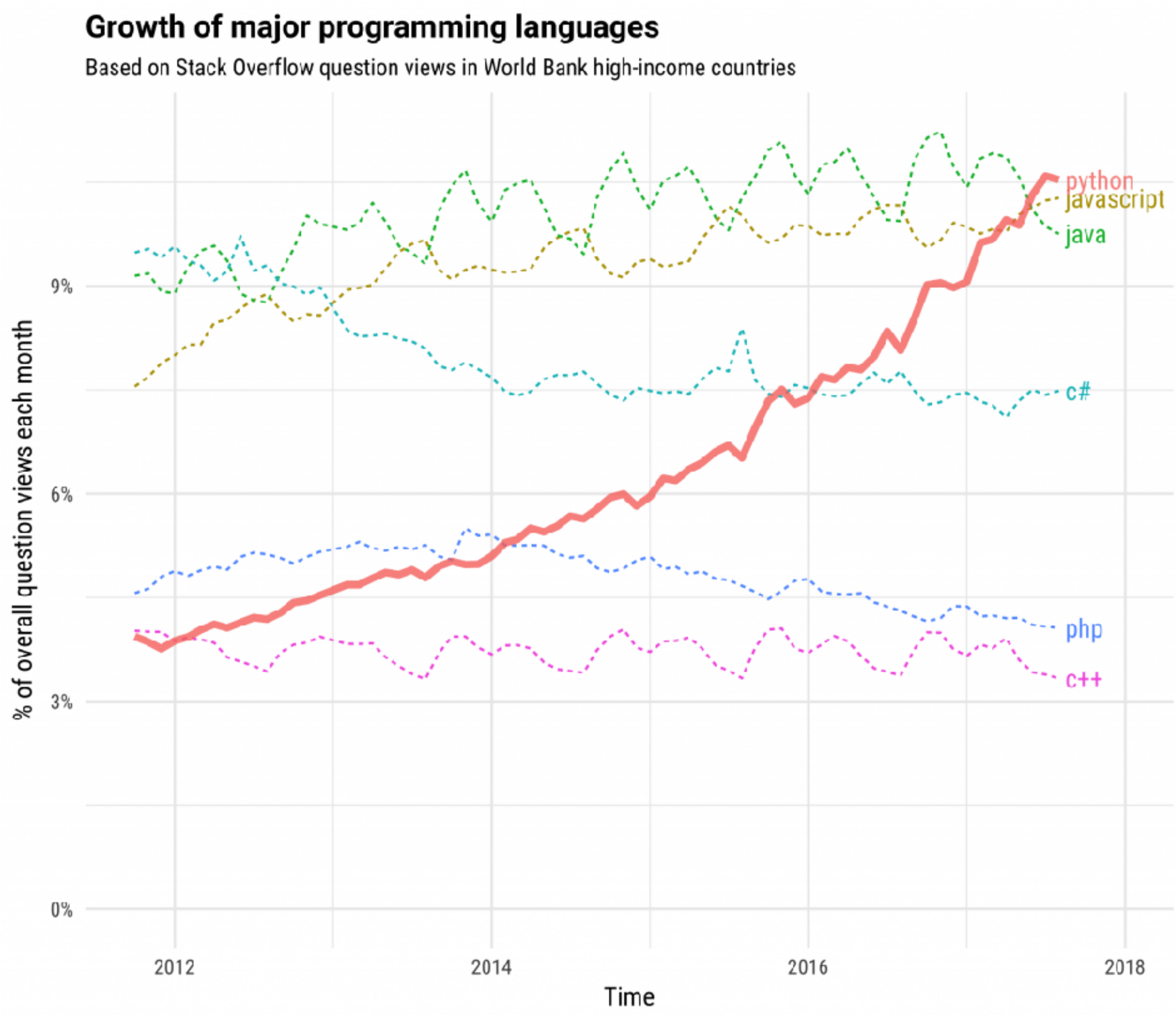
- From Remote Sensing Platforms:
  - New sensors / platforms
  - Continuous observations
  - Multiple versions of derived datasets

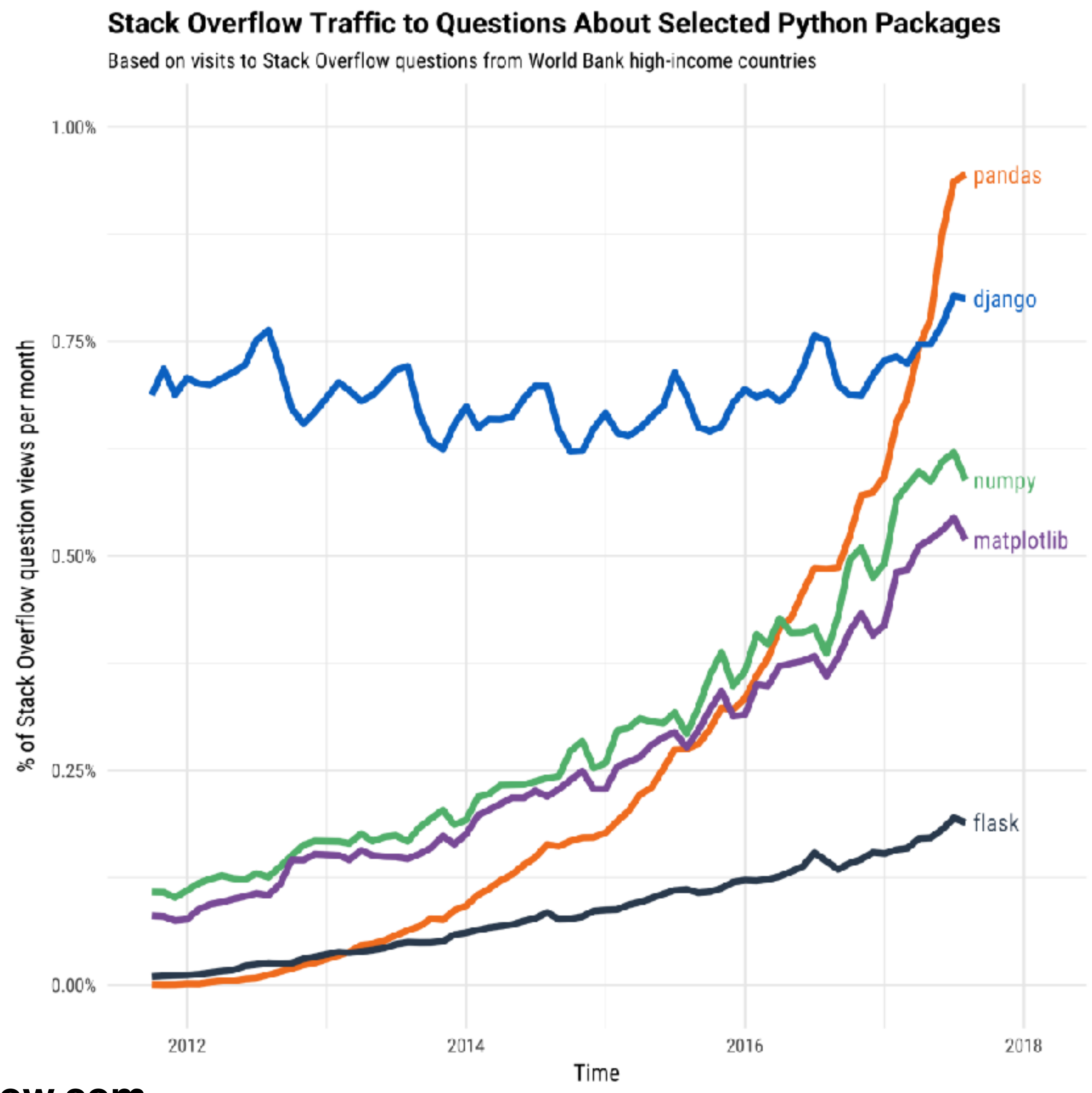
FRAGMENTATION PROBLEMS

1. **Software**
    - Few tangible incentives to share source code (funding agencies, journals)
    - Lack of extensible development patterns; often it is easier to "home grow" your own solution, rather than using someone else's.
    - Result is that most geoscientific research is effectively unreproducible and prone to failure.
2. **Data sprawl**
    - Inefficiencies of many copies of the same datasets ("dark replicas")
    - Lessons learned from the CMIP archives (CMIP3 was duplicated > 30x)
3. **Local vs. High-performance vs. Cloud Computing**
    - Traditional scientific computing workflows are difficult to port from a laptop, to HPC, to the cloud

PANGEO

SCIENTIFIC PYTHON FOR DATA SCIENCE



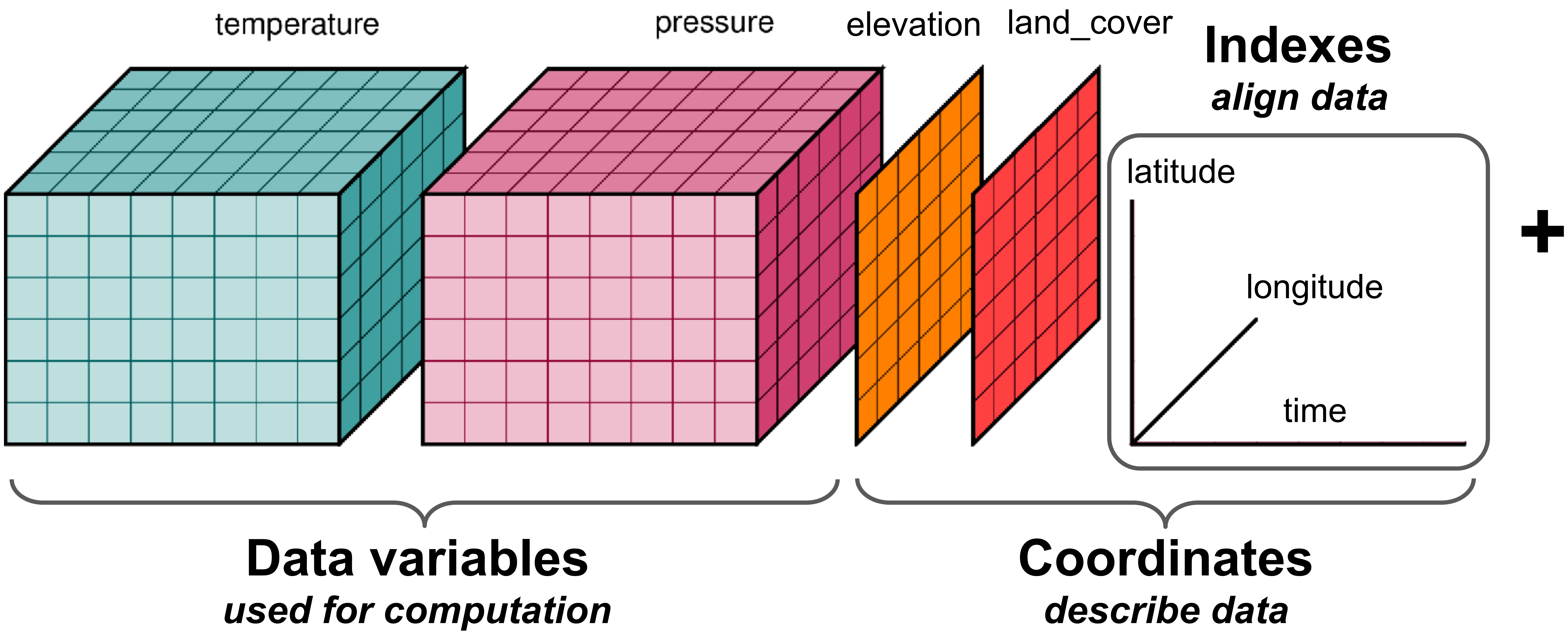


source: stackoverflow.com

SCIENTIFIC PYTHON FOR DATA SCIENCE

Credit: Stephan Hoyer, Jake Vanderplas (SciPy 2015)

XARRAY DATASET: MULTIDIMENSIONAL VARIABLES WITH COORDINATES AND METADATA



"netCDF meets pandas.DataFrame"

Credit: Stephan Hoyer

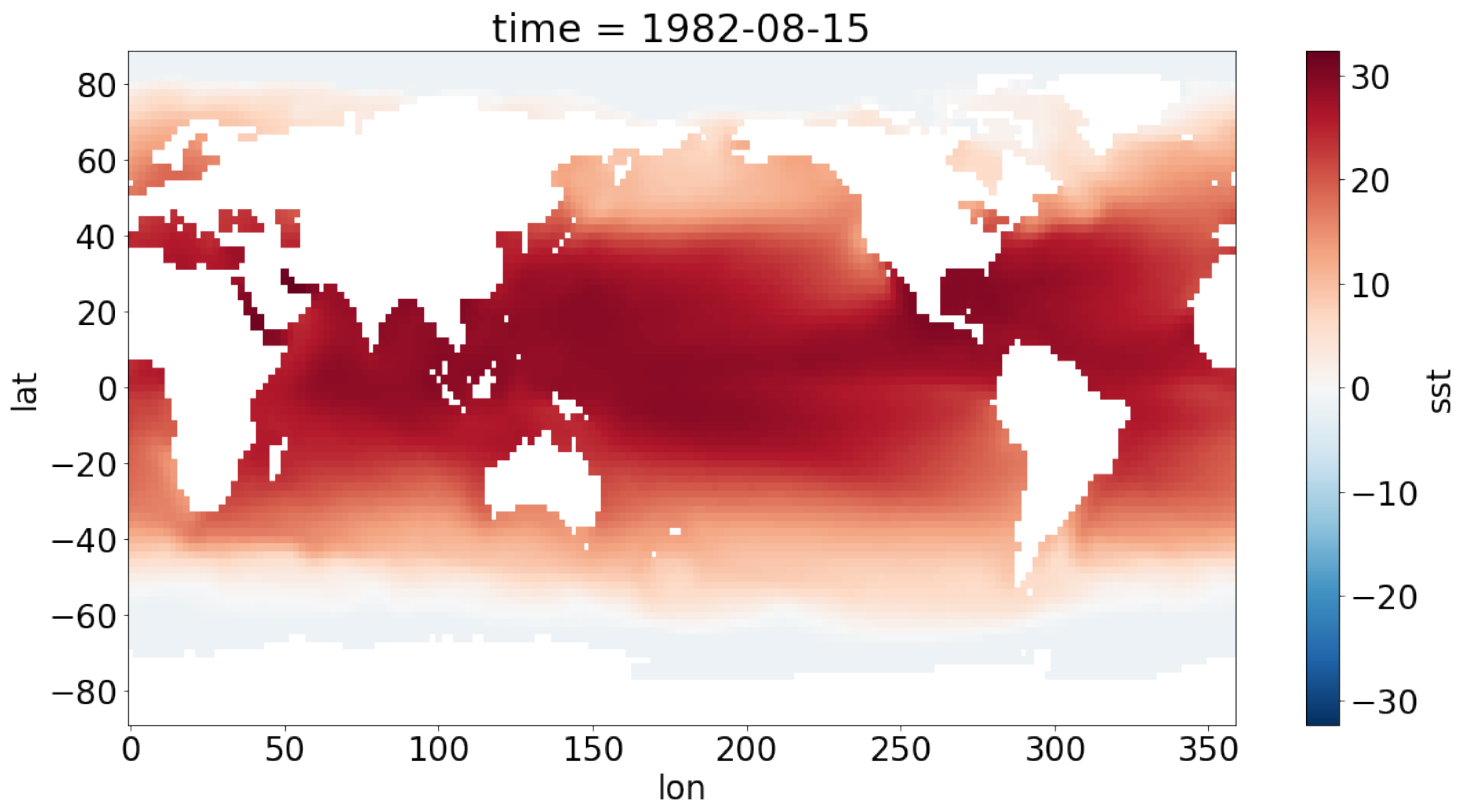
XARRAY MAKES SCIENCE EASY

```
import xarray as xr
ds = xr.open_dataset('NOAA_NCDC_ERSST_v3b_SST.nc')
ds
```

```
<xarray.Dataset>
Dimensions:  (lat: 89, lon: 180, time: 684)
Coordinates:
  * lat      (lat) float32 -88.0 -86.0 -84.0 -82.0 -80.0 -78.0 -76.0 -74.0 ...
  * lon      (lon) float32 0.0 2.0 4.0 6.0 8.0 10.0 12.0 14.0 16.0 18.0 20.0 ...
  * time     (time) datetime64[ns] 1960-01-15 1960-02-15 1960-03-15 ...
Data variables:
    sst      (time, lat, lon) float64 nan nan nan nan nan nan nan nan nan ...
Attributes:
    Conventions:  IRIDL
    source:       https://iridl.ldeo.columbia.edu/SOURCES/.NOAA/.NCDC/.ERSST/...
```

XARRAY: LABEL-BASED SELECTION

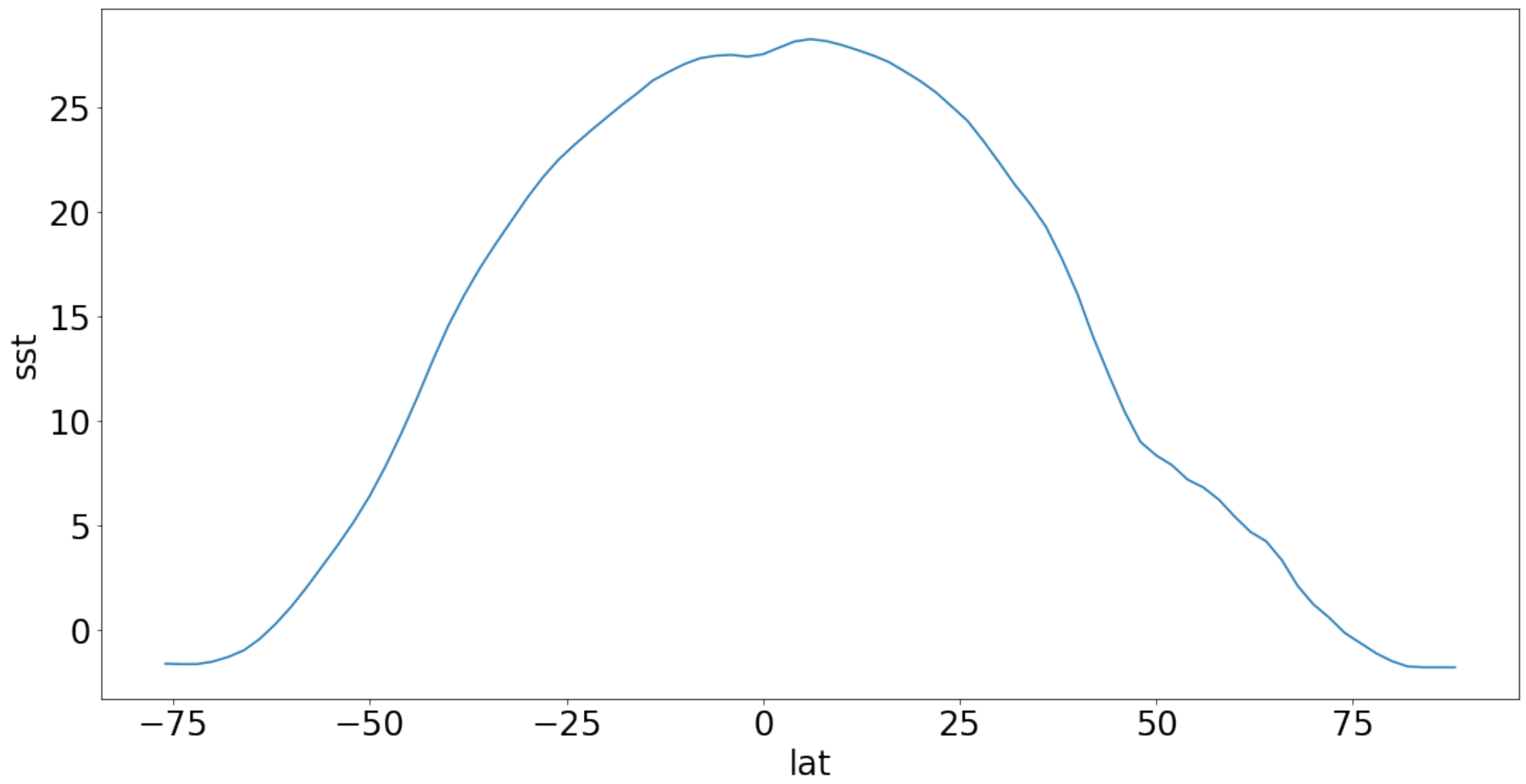
```
# select and plot data from my birthday
ds.sst.sel(time='1982-08-07', method='nearest').plot()
```



PANGEO

XARRAY: LABEL-BASED OPERATIONS

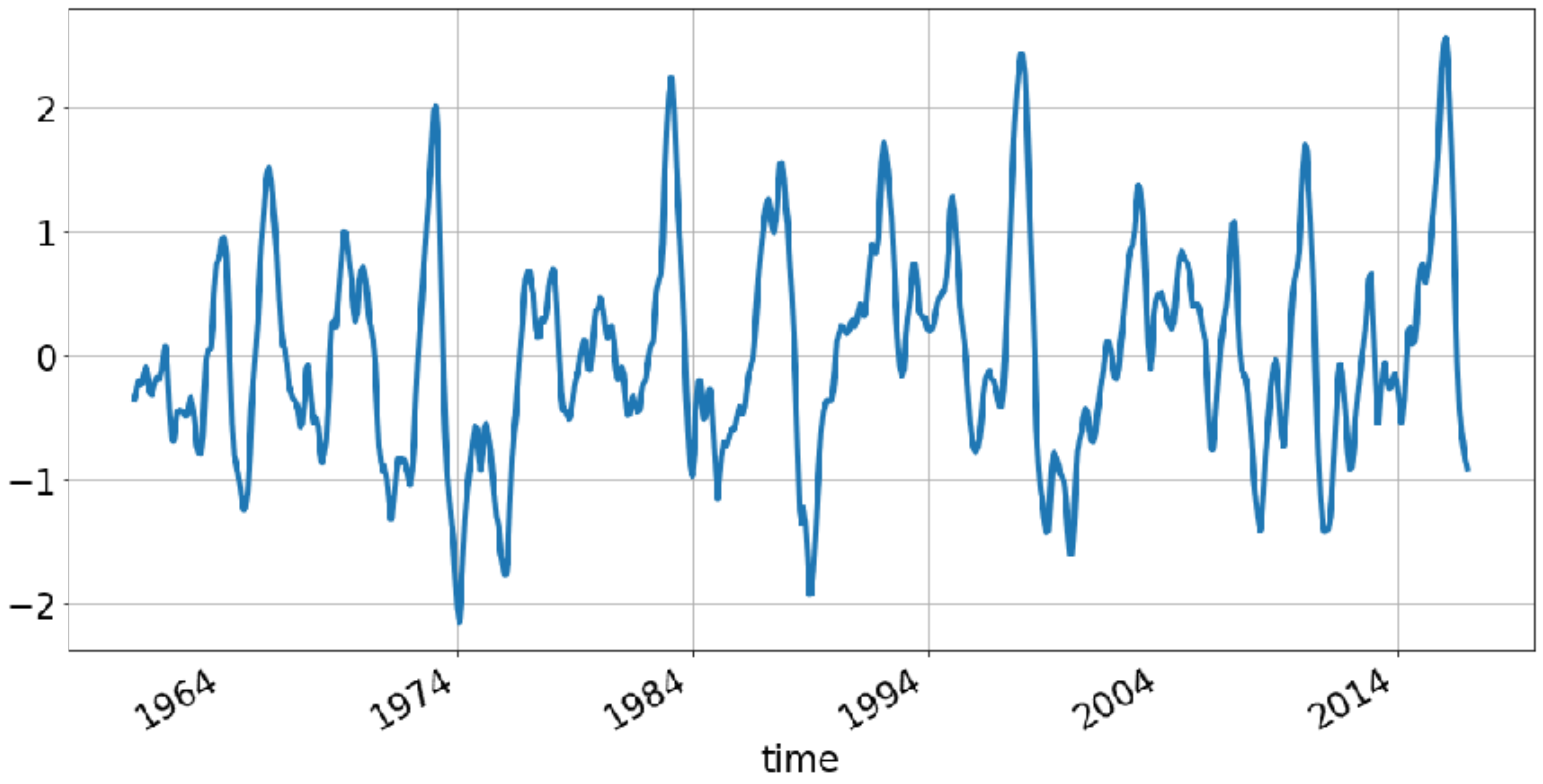
```
# zonal and time mean temperature
ds.sst.mean(dim=('time', 'lon')).plot()
```

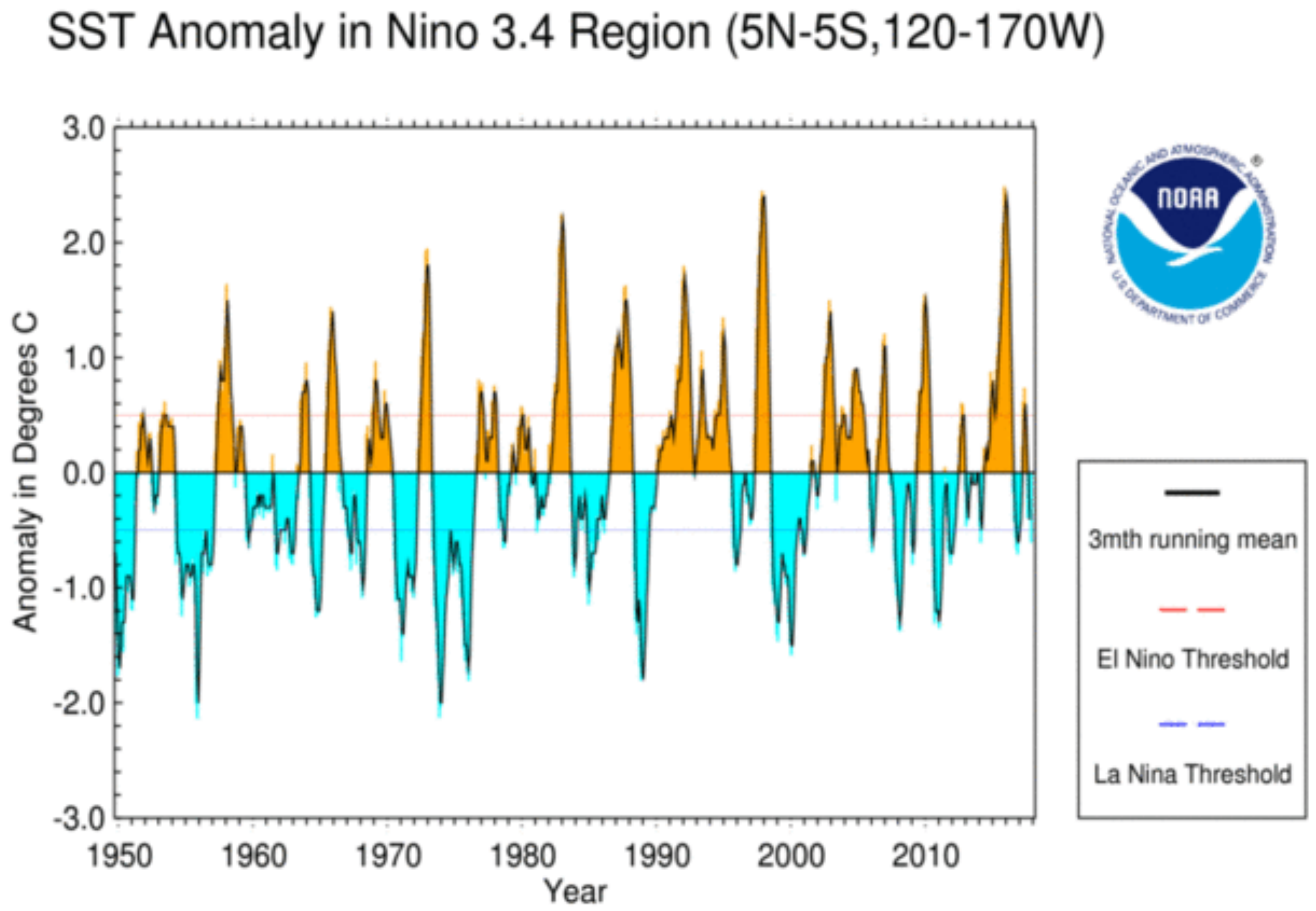


PANGEO

XARRAY: GROUPING AND AGGREGATION

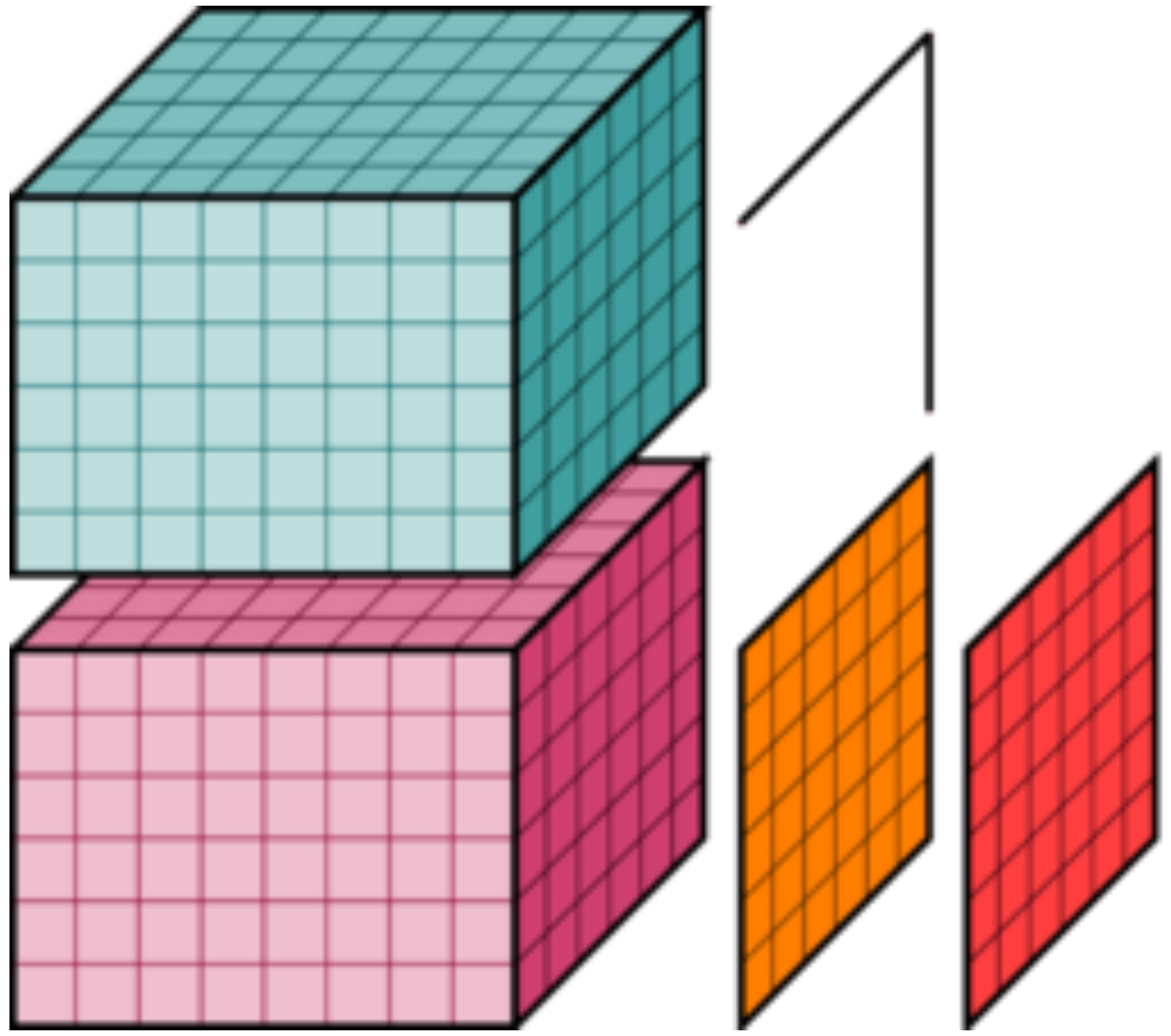
```
sst_clim = sst.groupby('time.month').mean(dim='time')
sst_anom = sst.groupby('time.month') - sst_clim
nino34_index = (sst_anom.sel(lat=slice(-5, 5), lon=slice(190, 240))
                        .mean(dim=('lon', 'lat'))
                        .rolling(time=3).mean(dim='time'))
nino34_index.plot()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





XARRAY

https://github.com/pydata/xarray

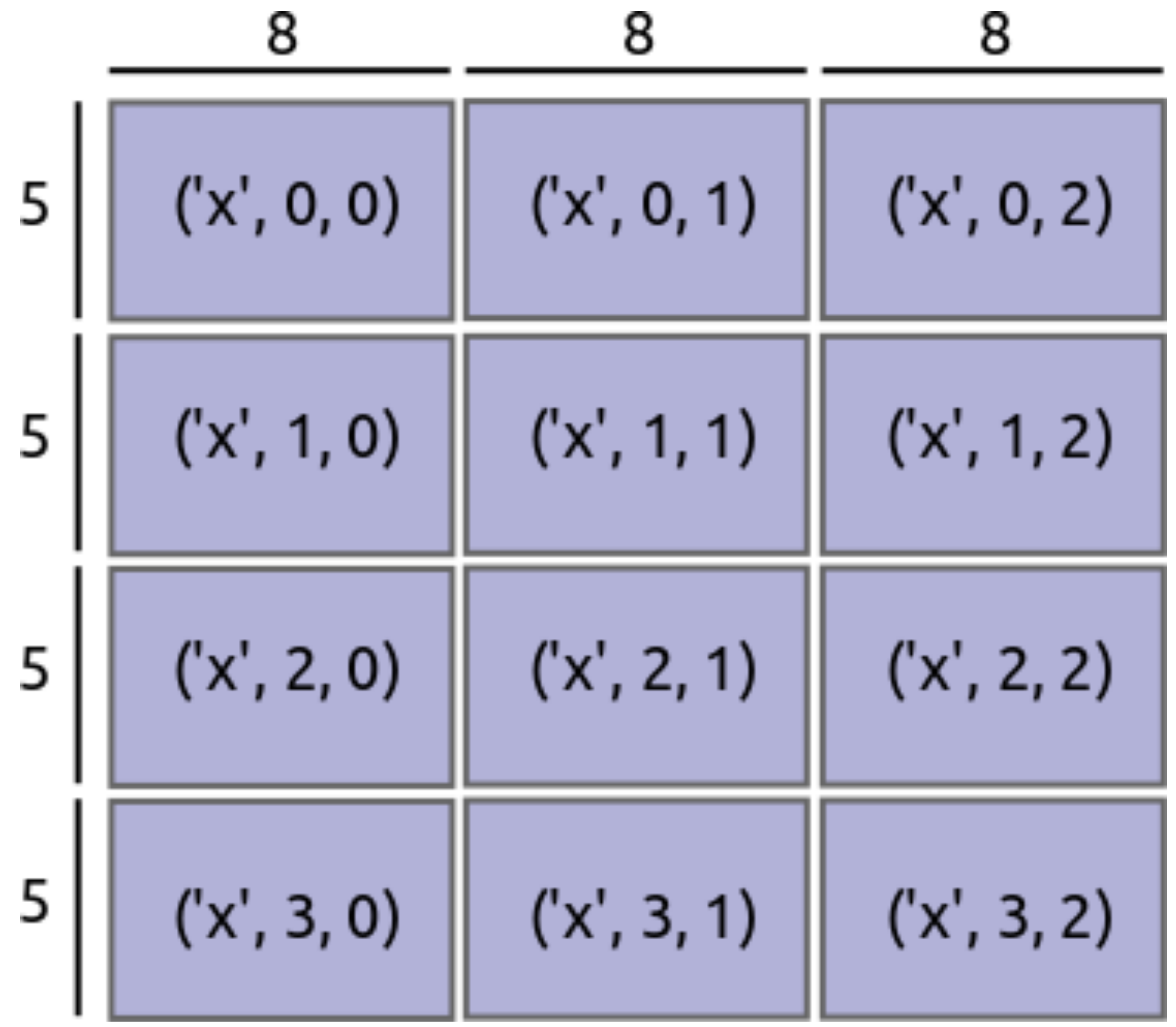


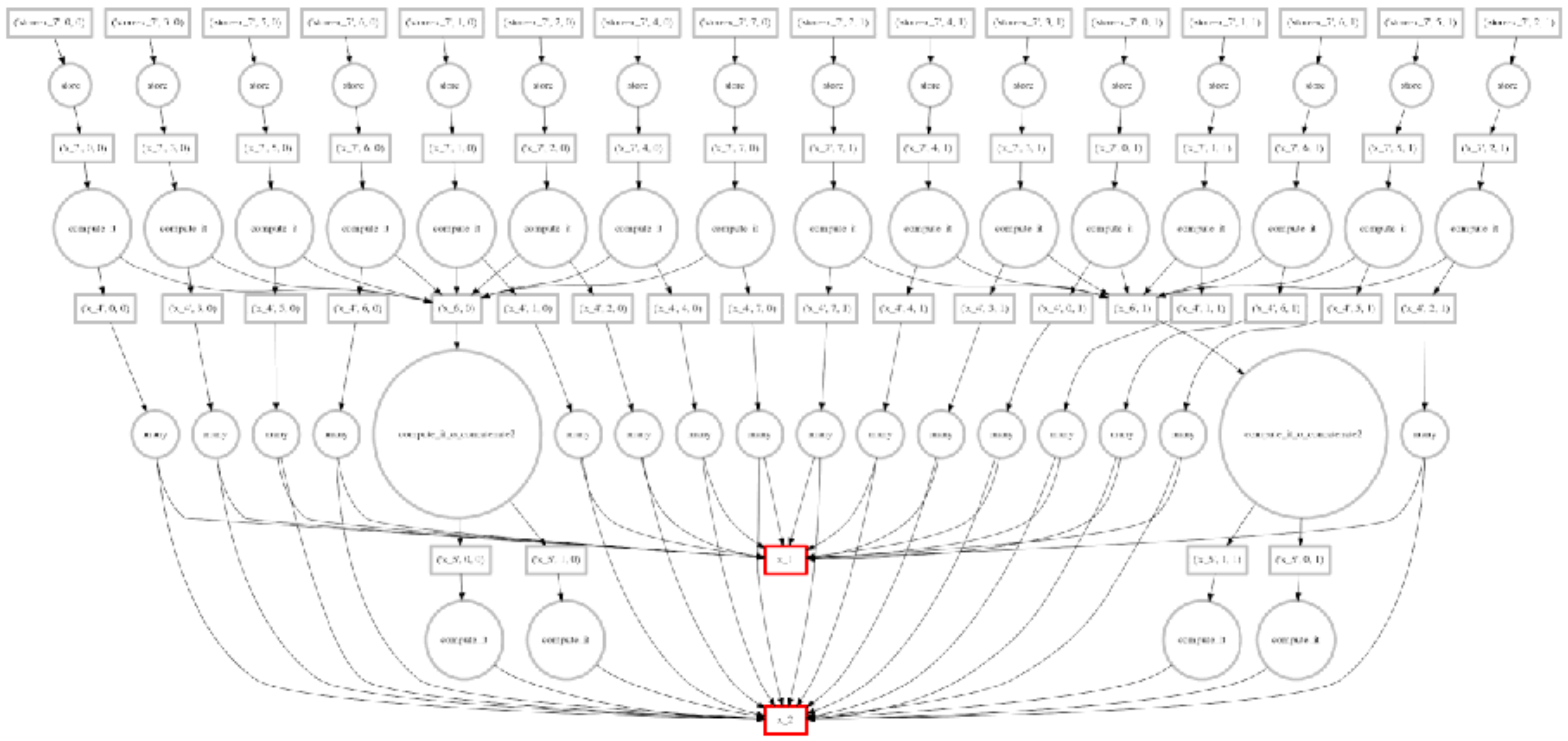
- label-based indexing and arithmetic
- interoperability with the core scientific Python packages (e.g., pandas, NumPy, Matplotlib)
- out-of-core computation on datasets that don't fit into memory (thanks dask!)
- wide range of input/output (I/O) options: netCDF, HDF, geoTIFF, zarr
- advanced multi-dimensional data manipulation tools such as group-by and resampling

PANGEO

DASK

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





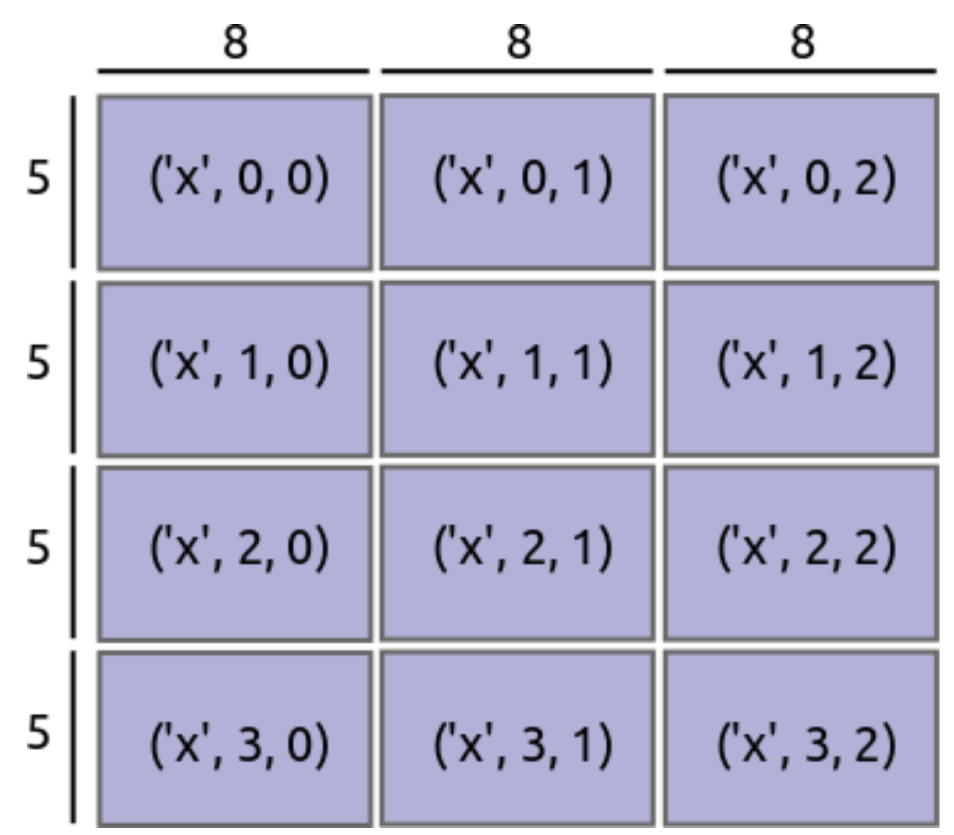
ND-Arrays are split into chunks that comfortably fit in memory

Complex computations represented as a graph of individual tasks.

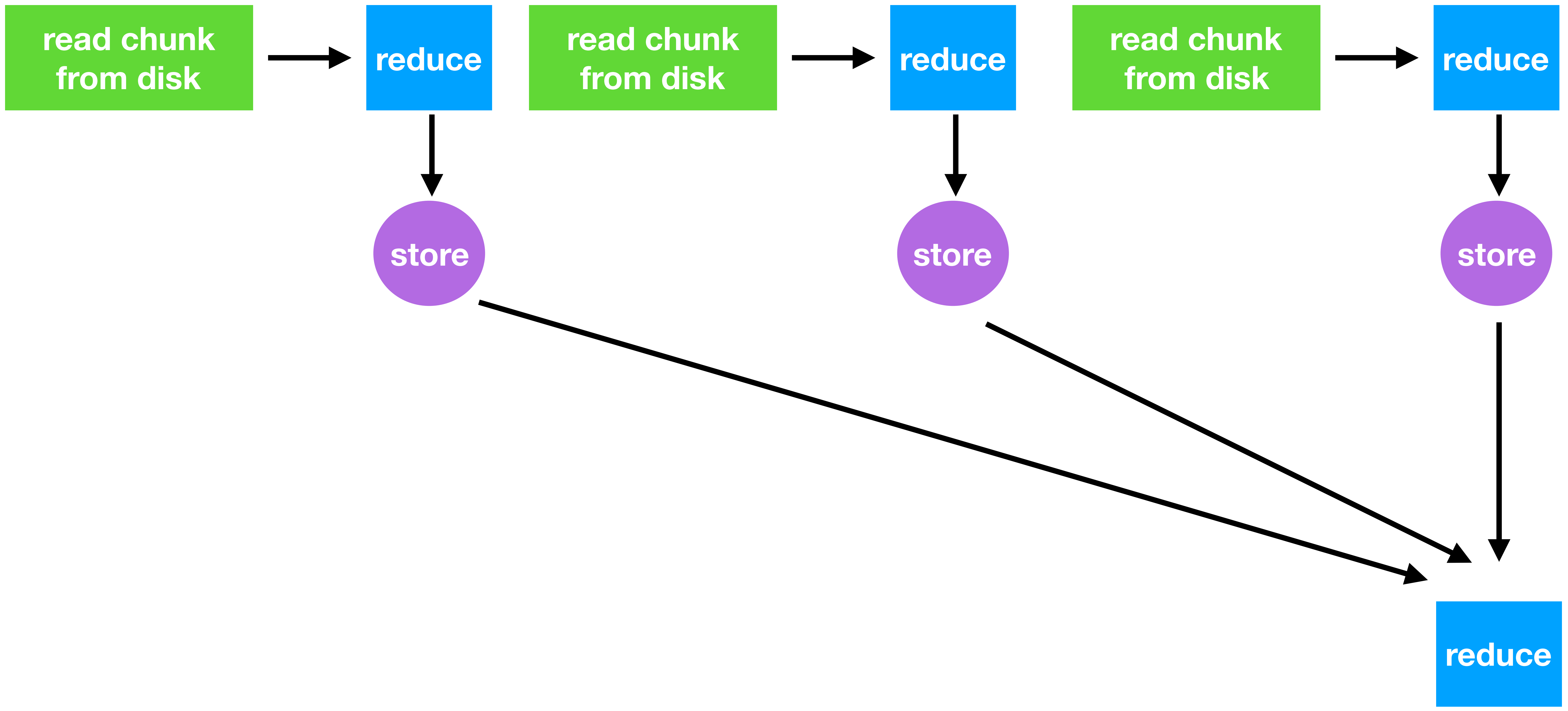
Scheduler optimizes execution of graph.

EXAMPLE CALCULATION: TAKE THE MEAN!

multidimensional array

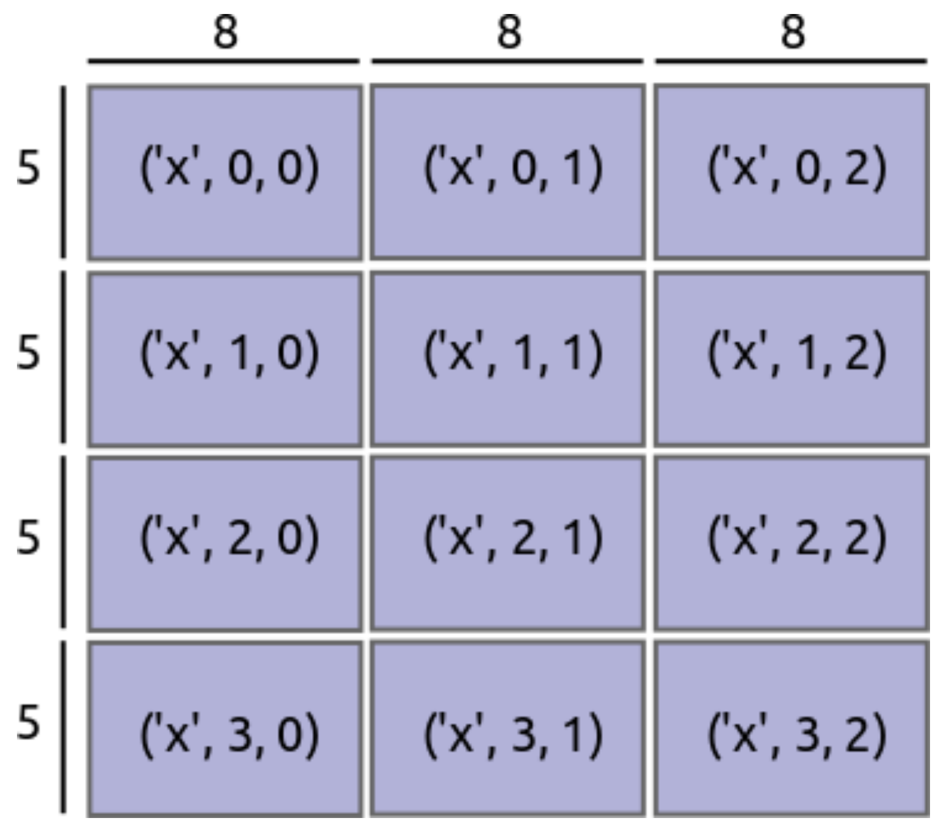


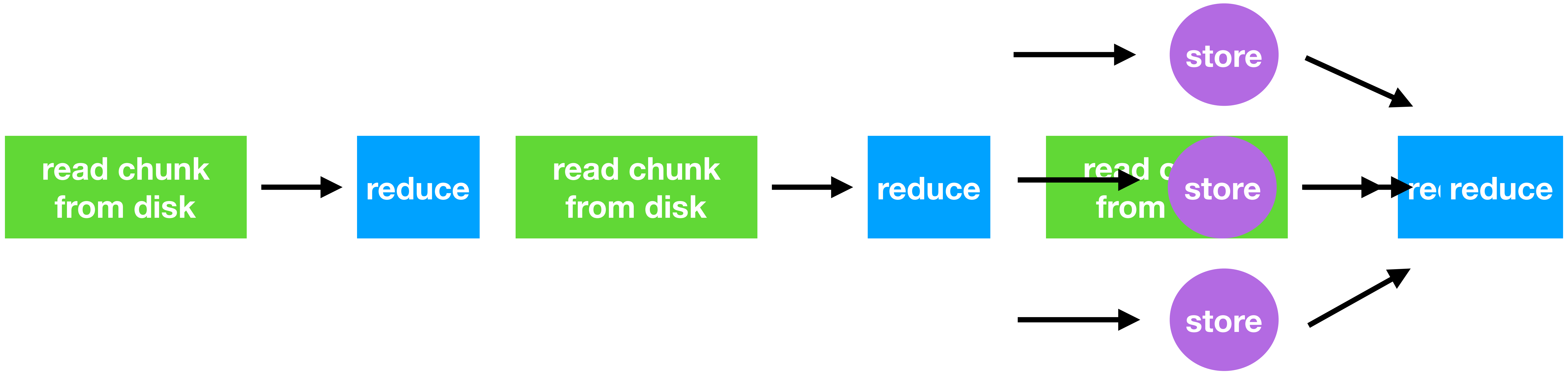
serial execution (a loop)



EXAMPLE CALCULATION: TAKE THE MEAN!

multidimensional array





parallel execution (dask graph)

PANGEO PROJECT GOALS

- 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 to handle petabyte-scale datasets on HPC and cloud platforms.

EARTHCUBE AWARD TEAM

NSF EarthCube Google Cloud Platform

Lamont-Doherty Earth Observatory
COLUMBIA UNIVERSITY | EARTH INSTITUTE

Ryan Abernathey, Chiara Lepore, Michael Tippet, Naomi Henderson, Richard Seager

NCAR
NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

Kevin Paul, Joe Hamman, Ryan May, Davide Del Vento

ANACONDA
Powered by Continuum Analytics®

Matthew Rocklin

OTHER CONTRIBUTORS

Met Office

Jacob Tomlinson, Niall Roberts, Alberto Arribas

Developing and operating Pangeo environment to support analysis of UK Met office products

USGS

Rich Signell

Deploying Pangeo on AWS to support analysis of coastal ocean modeling

Rhodium Group

Justin Simcock

Operating Pangeo in the cloud to support Climate Impact Lab research and analysis

NASA

Supporting Pangeo via SWOT mission and recently funded ACCESS award to UW / NCAR 🎉

BIDS Berkeley Institute for Data Science, Jupyter

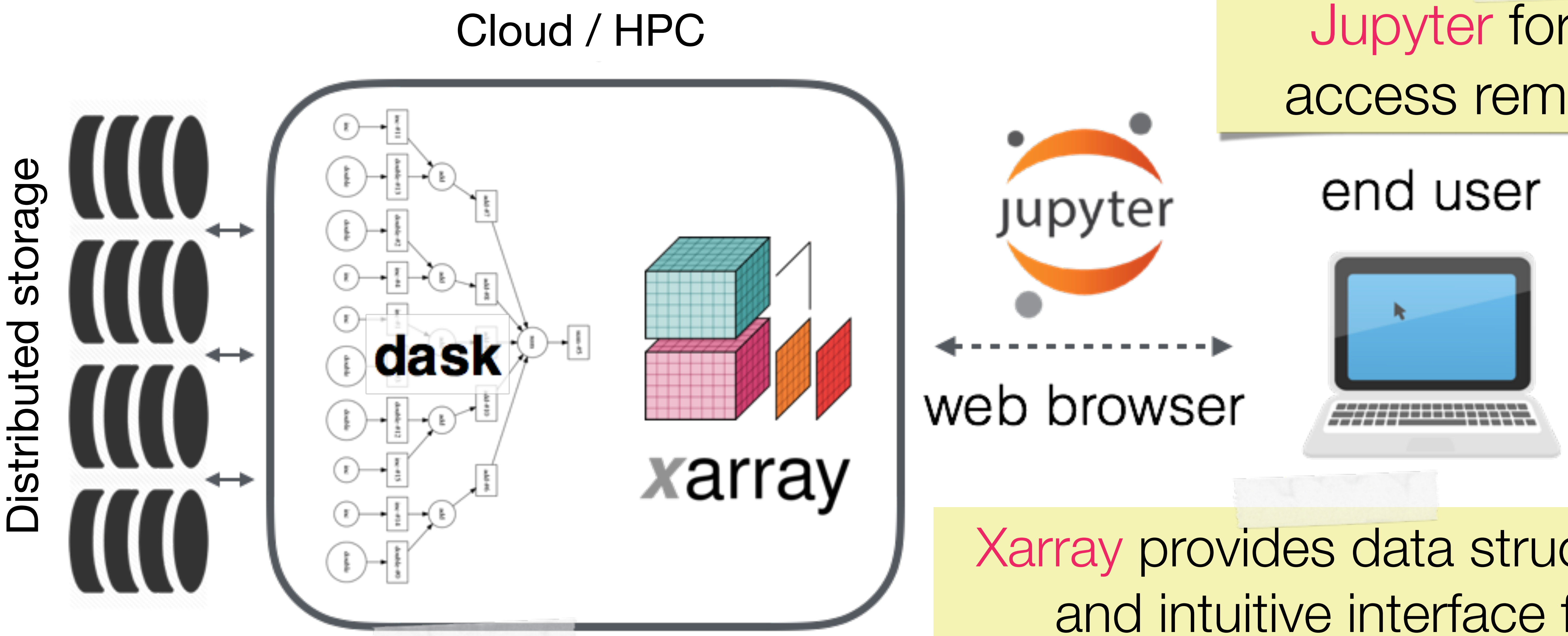
Yuvi Panda, Chris Holdgraf

Spending lots of time helping us make things work on the cloud

GFDL NOAA, Johns Hopkins University, ECMWF

PANGEO ARCHITECTURE

“Analysis Ready Data” stored on globally-available distributed storage.



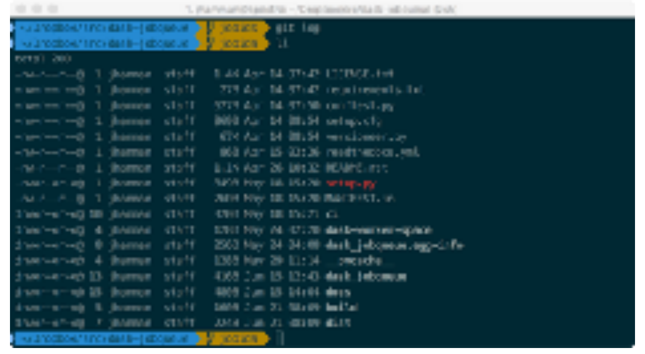
Jupyter for interactive access remote systems

Xarray provides data structures and intuitive interface for interacting with datasets

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

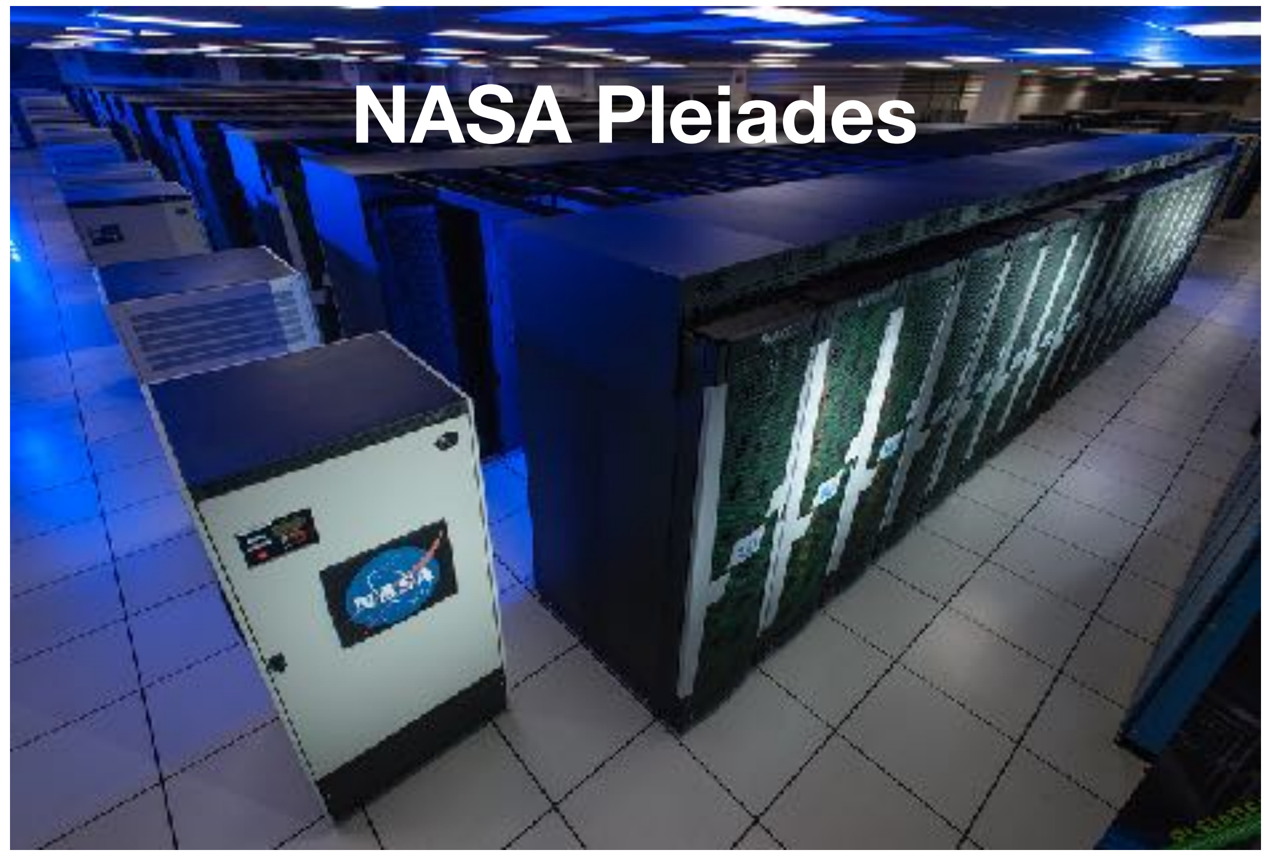
Dask tells the nodes what to do.

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

[HTTP://PANGEO-DATA.ORG/DEPLOYMENTS.HTML](http://pangeo-data.org/deployments.html)



NCAR Cheyenne

(SCALE USING JOB QUEUE SYSTEM)

[PANGEO.PYDATA.ORG](http://pangeo.pydata.org)

Google Cloud Platform

Over 500 unique users since March!

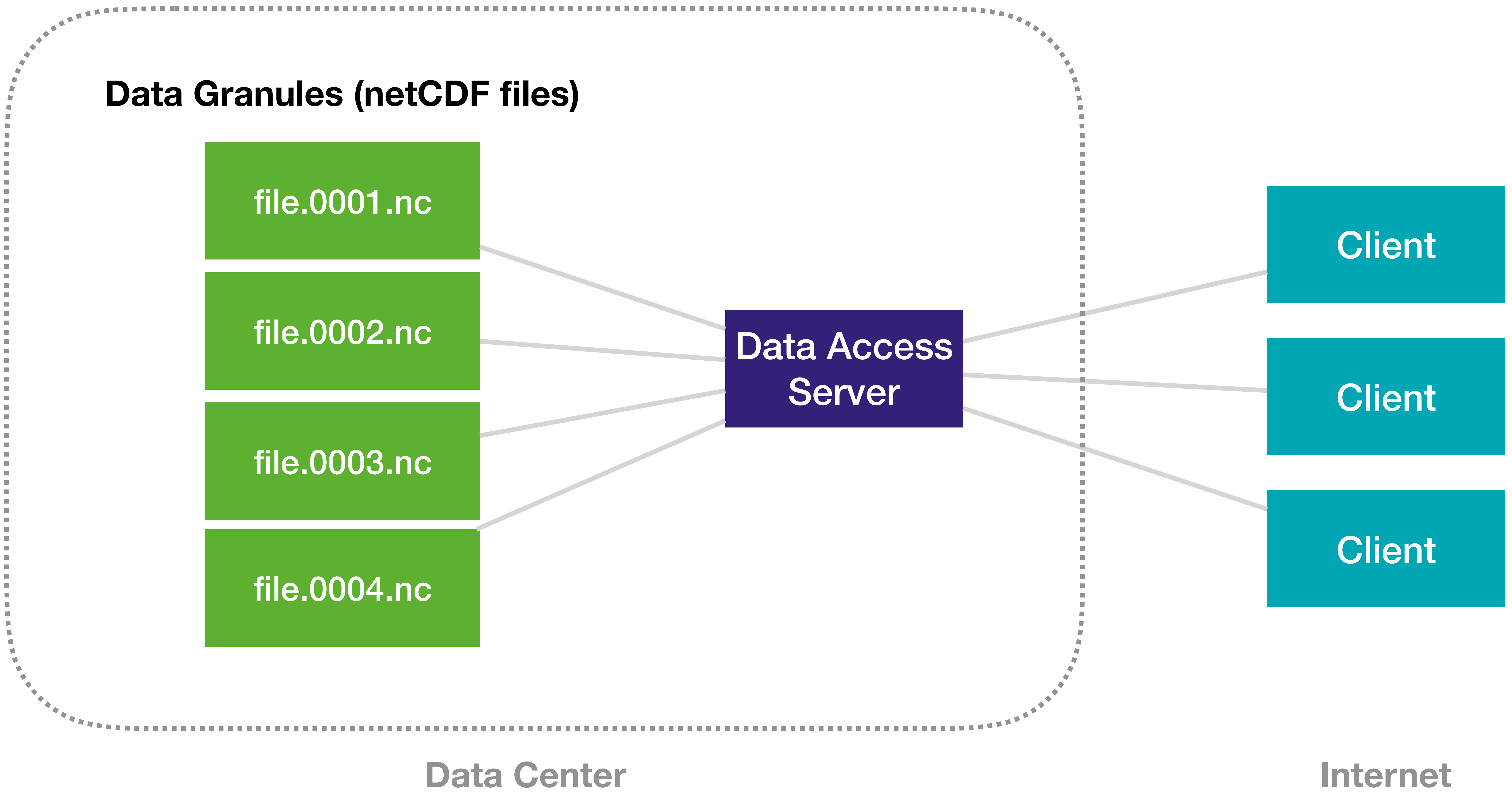
Microsoft Azure

aws

(SCALE USING KUBERNETES)

SHARING DATA IN THE CLOUD

Traditional Approach: A Data Access Portal

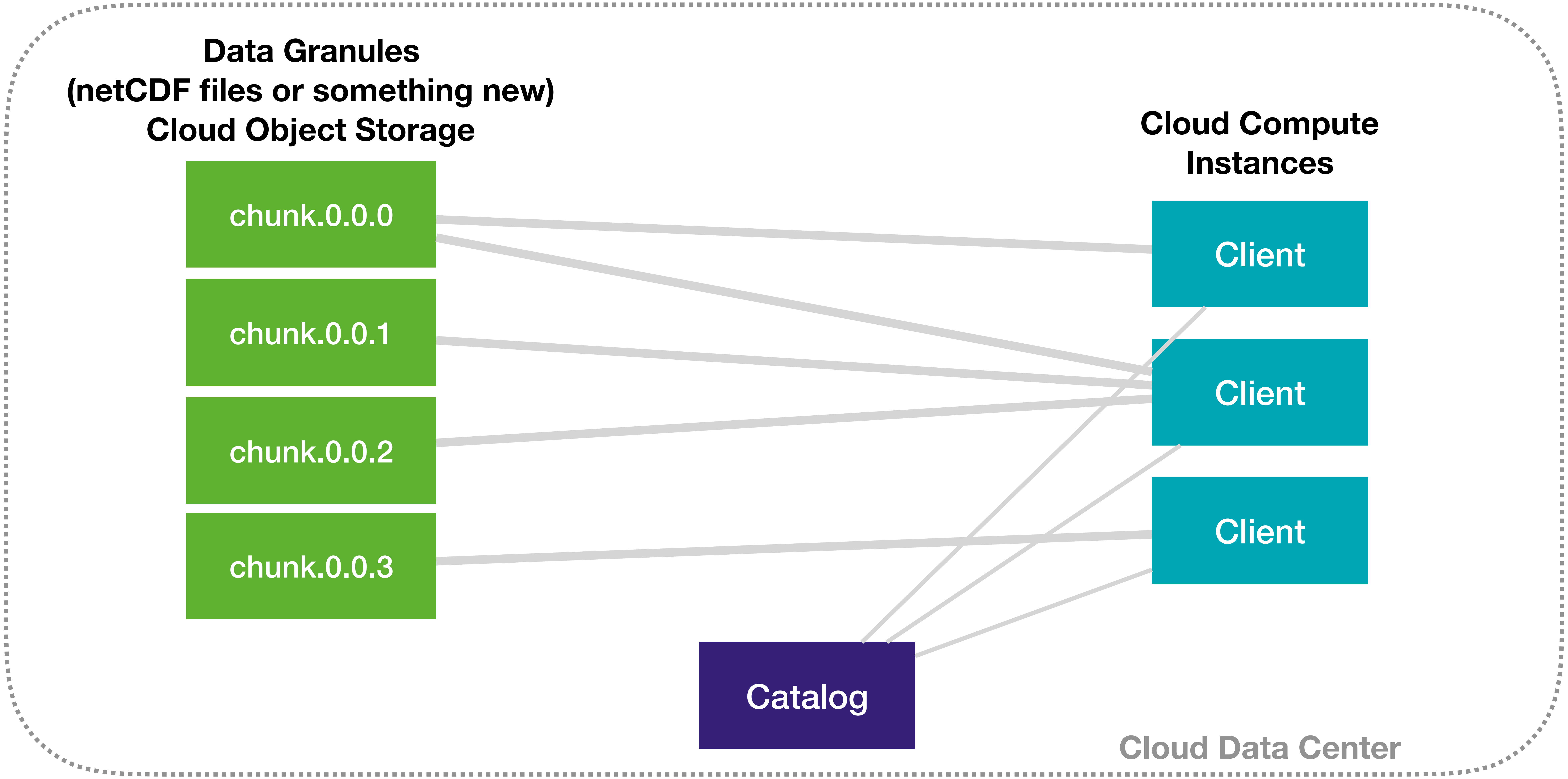


ON-DEMAND ANALYSIS-READY DATA

- **Too big to move**: assume data is to be used but not copied
- **Self-describing**: data and metadata packaged together
- **On-demand**: data can be read/used in its current form from anywhere
- **Analysis-ready**: no pre-processing required

SHARING DATA IN THE CLOUD

Direct Access to Cloud Object Storage



PANGEO

DASK SCALES COMPUTE… CAN THE STORAGE LAYER KEEP UP?

	Cloud Optimized GeoTIFF	HDF + FUSE	HDF + Custom Reader	Build a Distributed Service	New Storage Format (e.g. zarr)
pros	fast, well-established	works with existing files, no changes to HDF lib needed	works with existing files, no complex FUSE tricks	offloads the problem to others, maintains stable API	fast, intuitive, modern
Cons	data model not sophisticated enough	complex, low performance, brittle	Requires plugins to HDF library and tweaks to downstream libs	Complex, introduces intermediary, probably not free	not a community standard

By Matt Rocklin (Anaconda)
http://matthewrocklin.com/blog/work/2018/02/06/hdf-in-the-cloud

HOW TO SHARE A DATASET IN THE CLOUD

https://medium.com/pangeo/step-by-step-guide-to-building-a-big-data-portal-e262af1c2977

- Place your Big Data in cloud object storage in a self-describing, cloud-optimized format.

- Share a public path to your datasets (url/doi/ect)

```yaml
sea_surface:
  description: sea-surface altimetry data from The Copernicus Marine Environment
  driver: zarr
  args:
    urlpath: gcs://pangeo-data/dataset-duacs-rep-global-merged-allsat-phy-l4-v3-alt
    storage_options:
      token: anon
```

(EXAMPLE OF A "INTAKE" CATALOG)

HOW TO GET INVOLVED

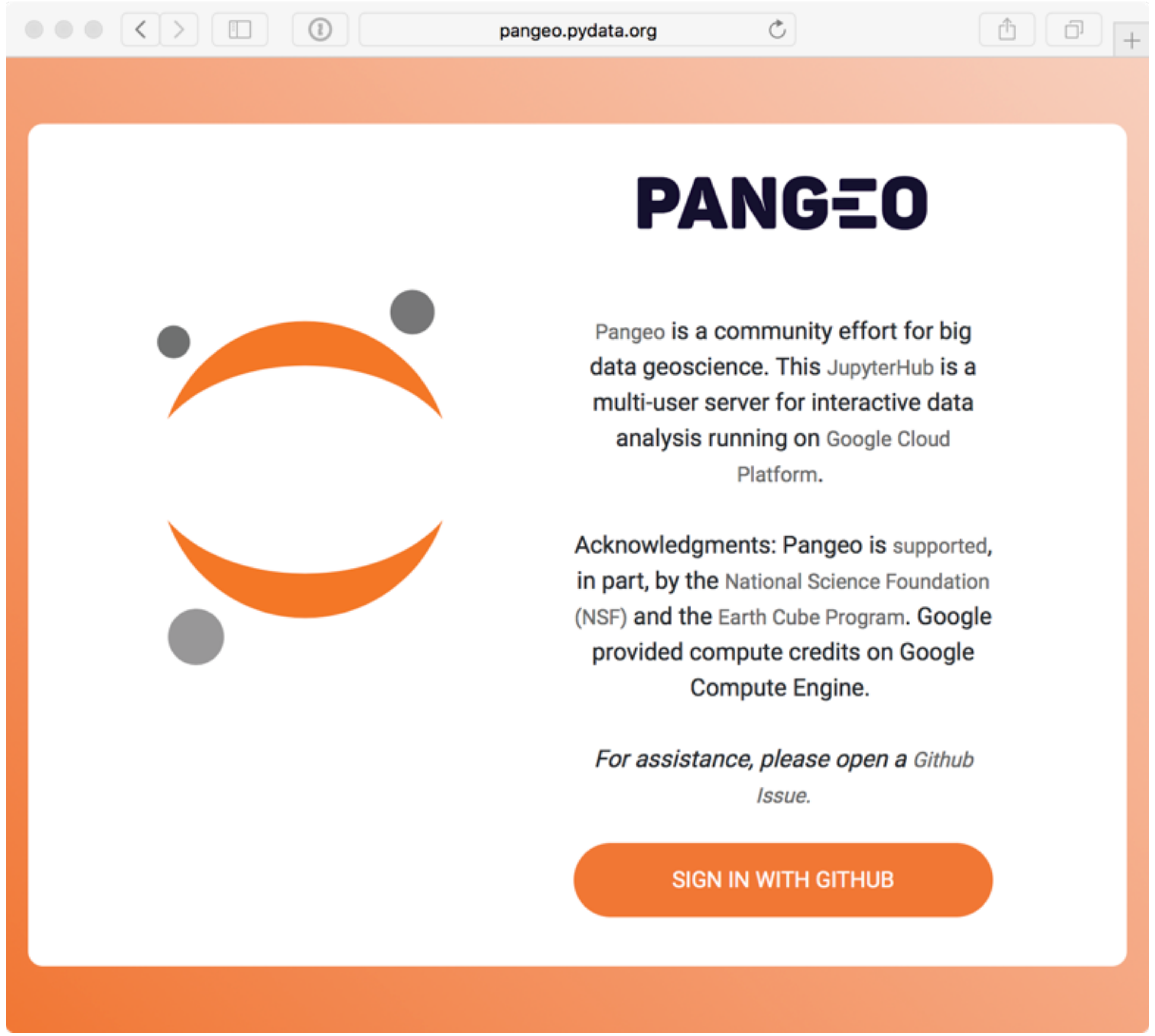
[HTTP://PANGEO-DATA.ORG](http://pangeo-data.org)

- Access and existing Pangeo deployment on an HPC cluster, or cloud resources (eg. pangeo.pydata.org)
- Adapt Pangeo elements to meet your projects needs (data portals, etc.) and give feedback via github: github.com/pangeo-data/pangeo
- Participate in open-source software development!

HANDS ON TIME

- Go to pangeo.pydata.org (requires GitHub credentials)
- Walk through `xarray-data.ipynb`
- Run a few of the examples
- Try some science of your own

(disclaimers about saving data, long term access, security, etc.)



MORE ON CLOUD NATIVE GEOSCIENCE

- **Cloud Native Geospatial Part 2: The Cloud Optimized GeoTIFF**
- **Towards On-Demand Analysis Ready Data**
  - https://medium.com/planet-stories
- **Step-by-Step Guide to Building a Big Data Portal**
  - https://medium.com/pangeo