EGU2018-12146

PANGEO

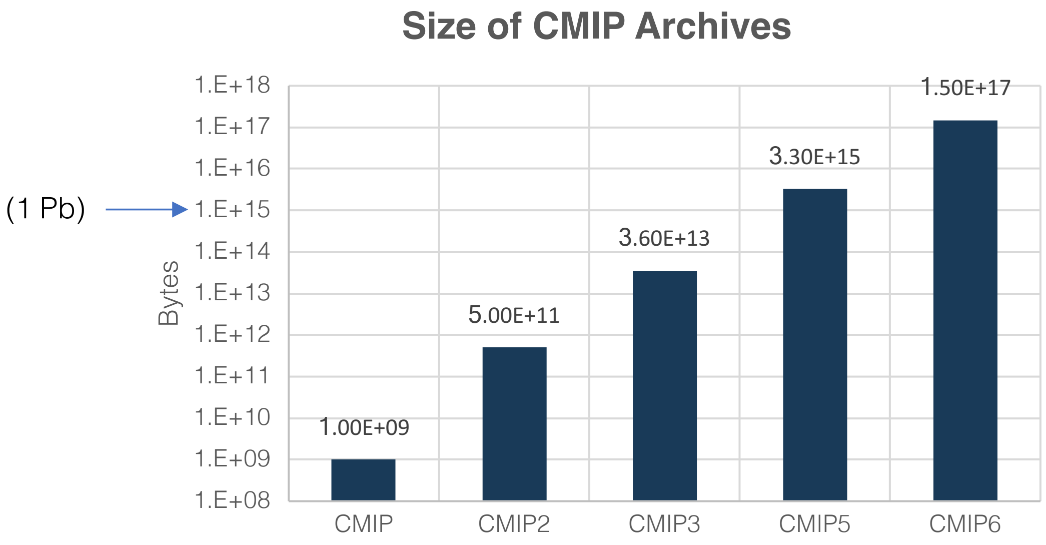
A BIG-DATA ECOSYSTEM FOR SCALABLE EARTH SYSTEM SCIENCE

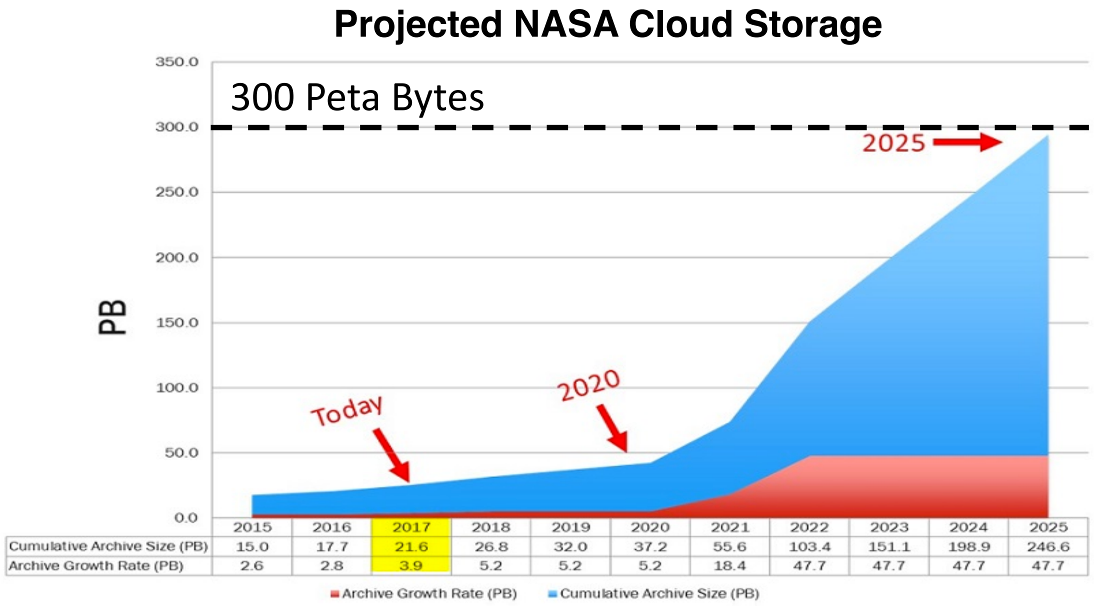
Joseph Hamman[1], Matthew Rocklin[2], Ryan Abernathy[3], and the rest of the Pangeo Community

1. Research Applications Laboratory, National Center for Atmospheric Research
2. Anaconda, Inc.
3. Columbia University / Lamont Doherty Earth Observatory

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


Size of CMIP Archives


Projected NASA Cloud Storage

	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

THE FRAGMENTATION PROBLEM

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
   - 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

GROWING TECHNOLOGY GAPS

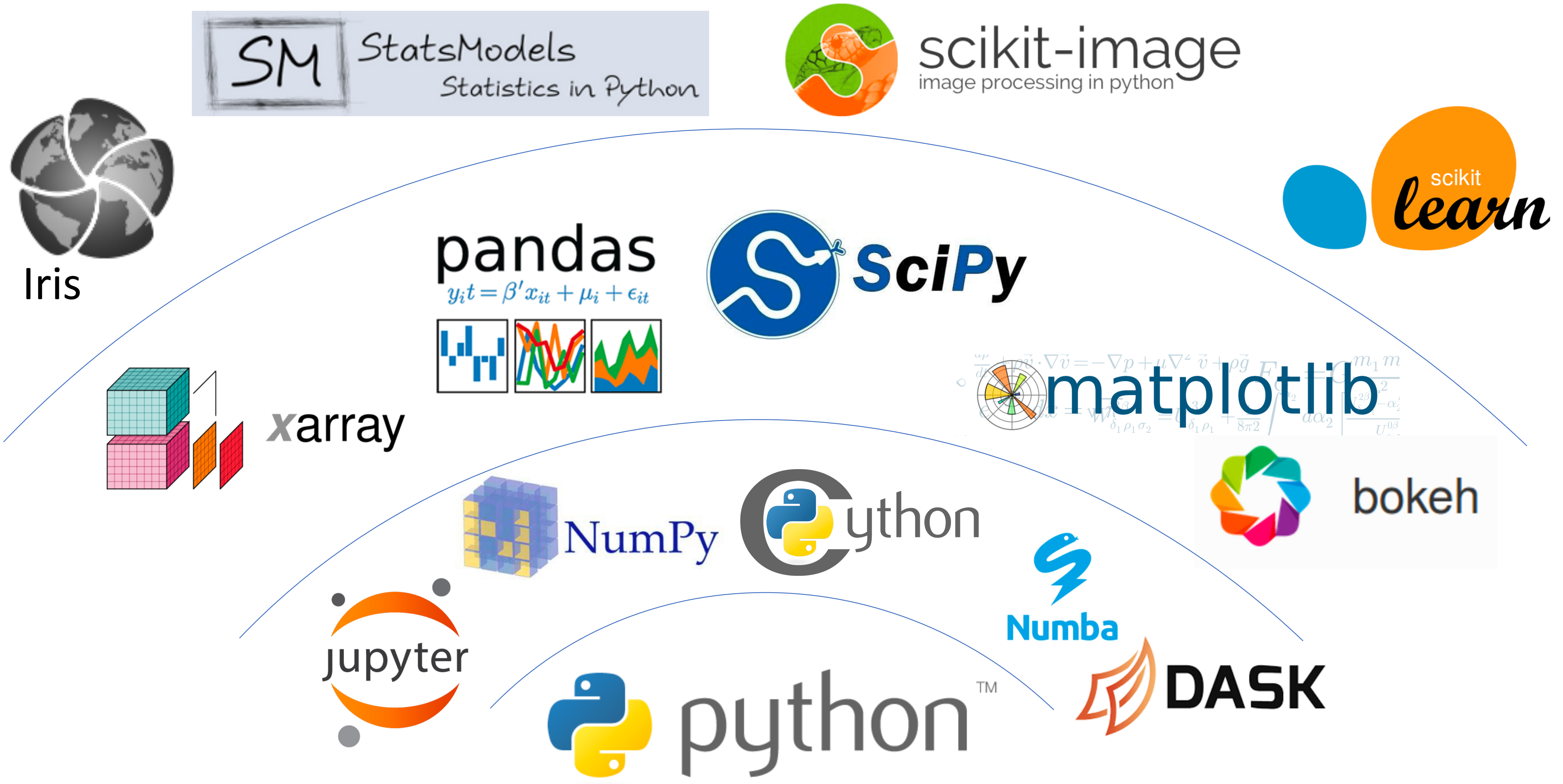
- Most geoscientists do not have specific training in data-science, software development, or computational methods.
- Private data-science industry vs. academia:
  - Leveraging open-source software / data
  - Adoption of big-data tools and frameworks (general understanding of how/where/when to scale)
  - Migration to the commercial cloud

PANGEO IS A COMMUNITY EFFORT FOR BIG DATA GEOSCIENCE

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

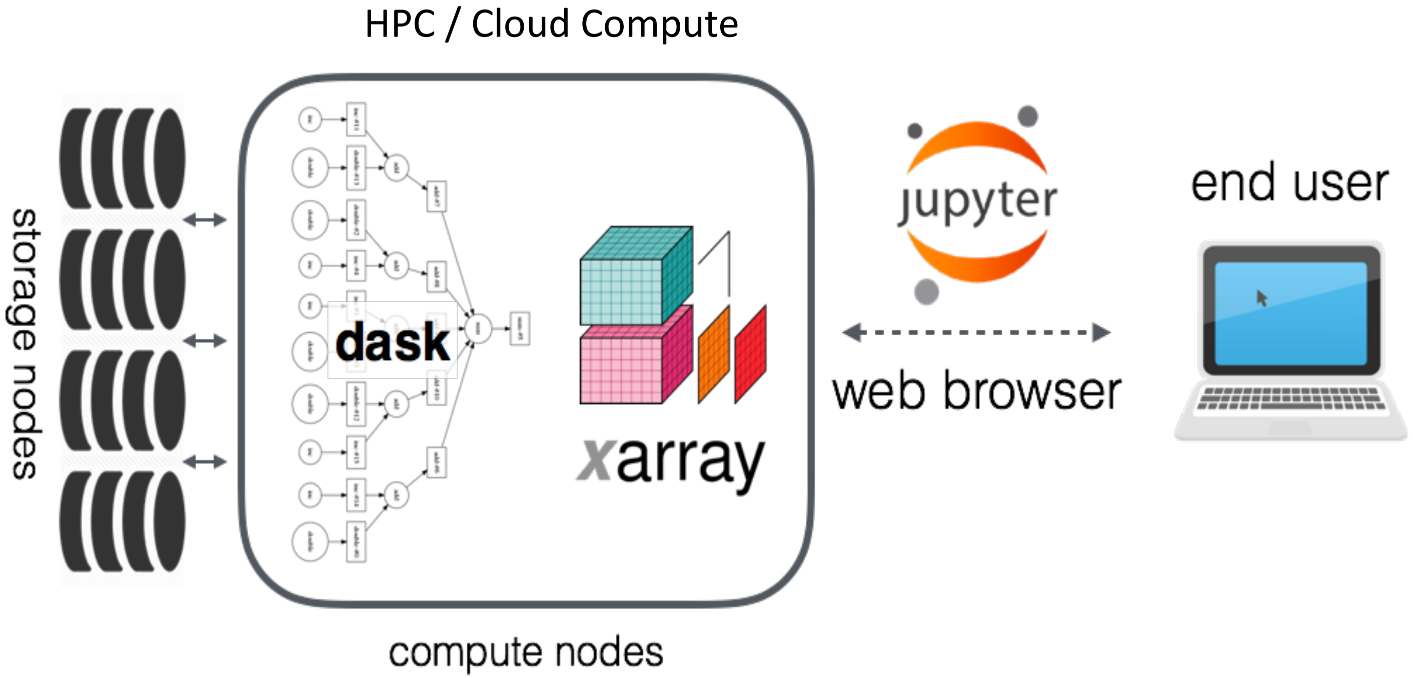
- **Vision**: We envision a collection of related but independent open-source packages that meet specific scientific needs within the geoscience fields. Core attributes of this ecosystem would include:
  - Open and collaborative development
  - Tools for scaling computations from small to very large datasets
  - Frameworks for moving scientific analysis to the data
  - Welcoming and inclusive development culture

PANGEO'S PYTHON BUILDING BLOCKS



PANGEO ECOSYSTEM

- 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

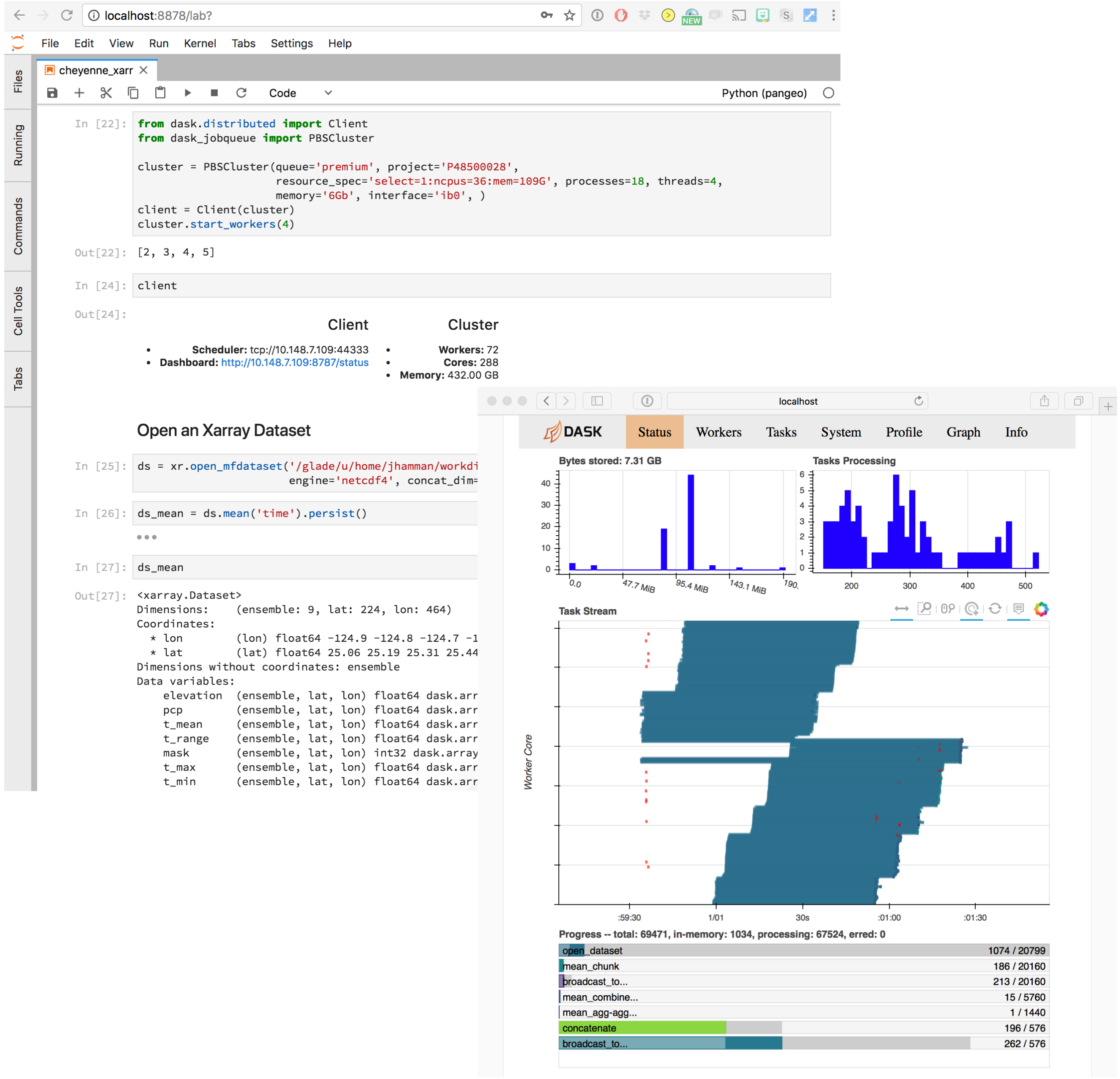


Examples of 3rd party packages in the Pangeo Ecosystem:

- Data discovery
- Regridding and GIS
- Vector calculous
- Signal processing
- Thermodynamics

HIGHLIGHT 1: HPC APPLICATIONS

- NCAR's Cheyenne Super Computer
  - 145,152 processors
  - 52.7 Pb of parallel disk storage
  - InfiniBand high-speed interconnect
- dask.distributed: parallel workers across many HPC nodes
- Xarray for computational toolkit and I/O
- Jupyter notebooks for interactive computing
- New tools for deploying dask clusters on HPC
  - e.g. **dask-jobqueue**[1]



```
In [22]: from dask.distributed import Client
         from dask_jobqueue import PBSCluster

         cluster = PBSCluster(queue='premium', project='P48500028',
                              resource_spec='select=1:ncpus=36:mem=109G', processes=18, threads=4,
                              memory='6Gb', interface='ib0', )
         client = Client(cluster)
         cluster.start_workers(4)

Out[22]: [2, 3, 4, 5]

In [24]: client

Out[24]:
```

Client	Cluster
Scheduler: tcp://10.148.7.109:44333	**Workers:** 72
Dashboard: http://10.148.7.109:8787/status	**Cores:** 288
	Memory: 432.00 GB

Open an Xarray Dataset

```
In [25]: ds = xr.open_mfdataset('/glade/u/home/jhamman/workdi
                                engine='netcdf4', concat_dim=

In [26]: ds_mean = ds.mean('time').persist()
         •••

In [27]: ds_mean

Out[27]: <xarray.Dataset>
         Dimensions:    (ensemble: 9, lat: 224, lon: 464)
         Coordinates:
           * lon        (lon) float64 -124.9 -124.8 -124.7 -1
           * lat        (lat) float64 25.06 25.19 25.31 25.44
         Dimensions without coordinates: ensemble
         Data variables:
             elevation  (ensemble, lat, lon) float64 dask.arr
             pcp        (ensemble, lat, lon) float64 dask.arr
             t_mean     (ensemble, lat, lon) float64 dask.arr
             t_range    (ensemble, lat, lon) float64 dask.arr
             mask       (ensemble, lat, lon) int32 dask.array
             t_max      (ensemble, lat, lon) float64 dask.arr
             t_min      (ensemble, lat, lon) float64 dask.arr
```

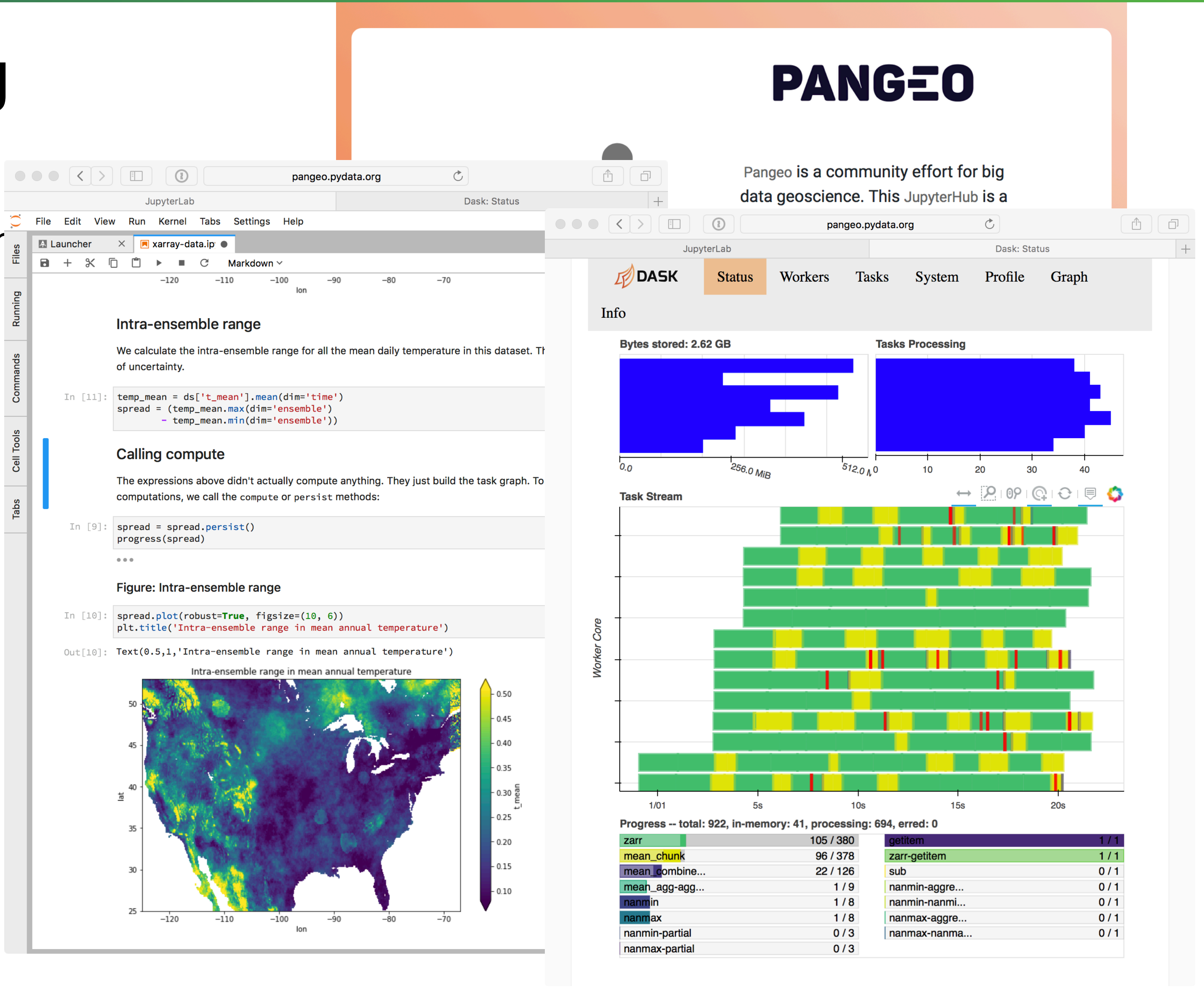
```
Progress -- total: 69471, in-memory: 1034, processing: 67524, erred: 0
open_dataset          1074 / 20799
mean_chunk             186 / 20160
broadcast_to...        213 / 20160
mean_combine...         15 / 5760
mean_agg-agg...          1 / 1440
concatenate            196 / 576
broadcast_to...        262 / 576
```

[1] https://github.com/dask/dask-jobqueue

HIGHLIGHT 2: CLOUD COMPUTING

pangeo.pydata.org

- JupyterHub running on the Google Cloud
- Kubernetes for both Jupyter and Dask-distributed
- 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.



Pangeo is a community effort for big data geoscience. This JupyterHub is a

Intra-ensemble range

We calculate the intra-ensemble range for all the mean daily temperature in this dataset. Th of uncertainty.

```
In [11]: temp_mean = ds['t_mean'].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 computations, 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')
```

```
Progress -- total: 922, in-memory: 41, processing: 694, erred: 0
zarr               105 / 380      getitem            1 / 1
mean_chunk          96 / 378      zarr-getitem       1 / 1
mean_combine...     22 / 126      sub                0 / 1
mean_agg-agg...      1 / 9        nanmin-aggre...    0 / 1
nanmin               1 / 8        nanmin-nanmi...    0 / 1
nanmax               1 / 8        nanmax-aggre...    0 / 1
nanmin-partial       0 / 3        nanmax-nanma...    0 / 1
nanmax-partial       0 / 3
```

CONCLUSIONS AND NEXT STEPS

- **Conclusions**
  - Big-data is requiring us to look for fundamentally different ways of doing our science.
  - Pangeo is an open community effort for big data in the geosciences that is building on top of the existing Scientific Python stack.
  - Rather than building a monolithic platform, we are aiming to develop a ecosystem independent, but related packages.
  - Development on HPC and Google Cloud is happening in concert.
- **Next steps and ongoing efforts:**
  - Benchmark approaches for storing / using geoscience data in the cloud.
  - Further refinement of JupyterHub configuration.
  - Deploy similar JupyterHub interface for HPC platforms.
  - Data discovery.
  - Do more science using Pangeo!

QUESTIONS AND CREDITS

Pangeo Earth Cube Team

- Columbia / Lamont:
  - Ryan Abernathey (lead PI)
  - Chiara Lepore
  - Michael Tippett
  - Richard Seager
  - Naomi Henderson
- National Corporation for Atmospheric Research / Unidata:
  - Kevin Paul (lead PI)
  - Joe Hamman
  - Ryan May
  - Davide Del Vento
- Anaconda Inc.:
  - Matt Rocklin

Community

- Collaborators at:
  - 25+ Universities / Labs
  - 3+ companies
- Join Us!:
  - Website:
    - pangeo-data.github.io
  - Jupyterhub (beta deployment):
    - pangeo.pydata.org
  - Discussion:
    - github.com/pangeo-data/pangeo

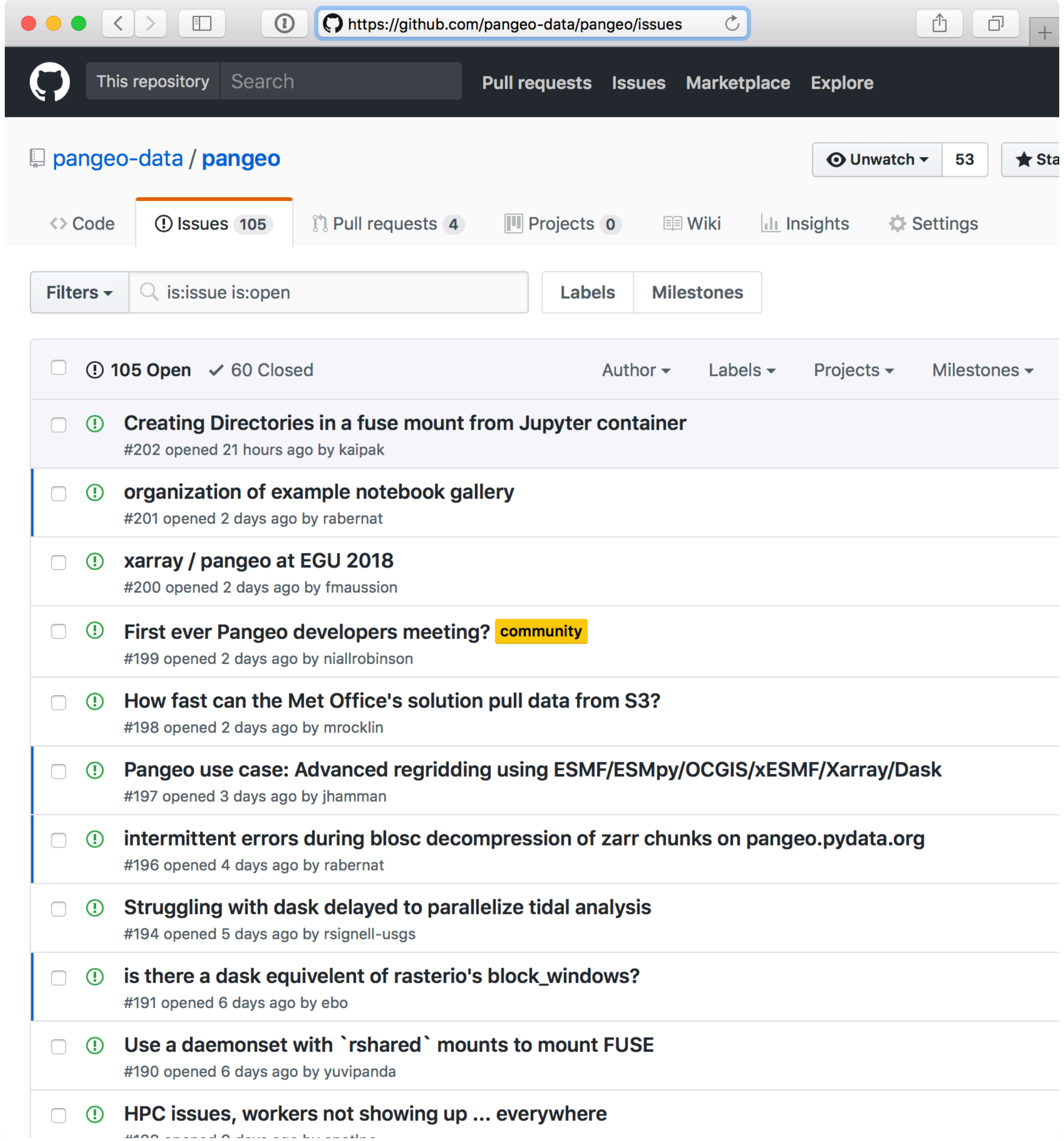
Special thanks to the UK Met Office Team

Met Office

NCAR is supported by the U.S. National Science Foundation.
Pangeo is supported, in part, by the U.S. National Science Foundation and the Earth Cube Program.
Google provided computing credits on Google Compute Engine.

EXTRAS

GITHUB



https://github.com/pangeo-data/pangeo/issues

pangeo-data / pangeo

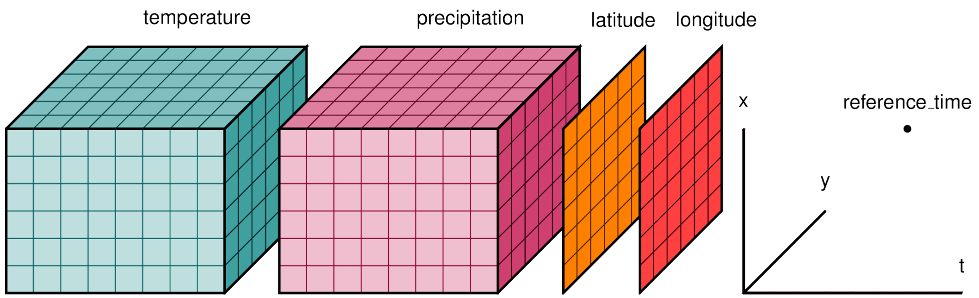
is:issue is:open

105 Open ✓ 60 Closed

- □ Creating Directories in a fuse mount from Jupyter container
  #202 opened 21 hours ago by kaipak
- □ organization of example notebook gallery
  #201 opened 2 days ago by rabernat
- □ xarray / pangeo at EGU 2018
  #200 opened 2 days ago by fmaussion
- □ First ever Pangeo developers meeting? community
  #199 opened 2 days ago by niallrobinson
- □ How fast can the Met Office's solution pull data from S3?
  #198 opened 2 days ago by mrocklin
- □ Pangeo use case: Advanced regridding using ESMF/ESMpy/OCGIS/xESMF/Xarray/Dask
  #197 opened 3 days ago by jhamman
- □ intermittent errors during blosc decompression of zarr chunks on pangeo.pydata.org
  #196 opened 4 days ago by rabernat
- □ Struggling with dask delayed to parallelize tidal analysis
  #194 opened 5 days ago by rsignell-usgs
- □ is there a dask equivelent of rasterio's block_windows?
  #191 opened 6 days ago by ebo
- □ Use a daemonset with `rshared` mounts to mount FUSE
  #190 opened 6 days ago by yuvipanda
- □ HPC issues, workers not showing up ... everywhere

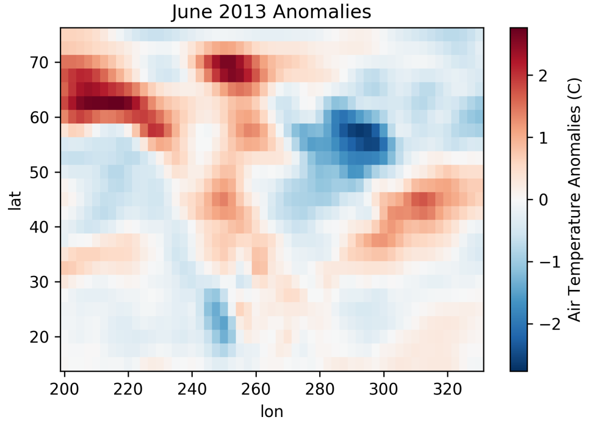
XARRAY

- N-D labeled arrays and datasets in Python
- Data model emulates the Common Data Model (e.g. NetCDF)
- Key features:
  - Label-based indexing
  - Interoperability with core scientific Python packages
  - Parallel computation using Dask
  - Wide range of input/output options
  - Robust data analysis and manipulation toolkit



```python
import xarray as xr

# Load a netCDF dataset
ds = xr.open_dataset('air_temperature.nc')
# Resample daily data to monthly means
ds = ds.resample('MS', dim='time', how='mean')
# Calculate a monthly climatology
climatology = ds.groupby('time.month').mean(dim='time')
# Calculate monthly anomalies
anomalies = ds.groupby('time.month') - climatology
# Plot an example monthly anomaly (June 2013)
anomalies.sel(time='2013-06')['air'].plot()
```

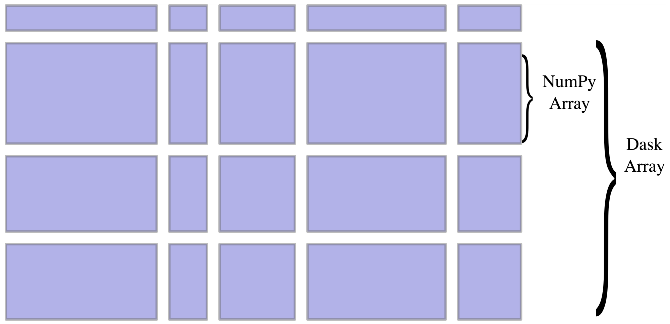


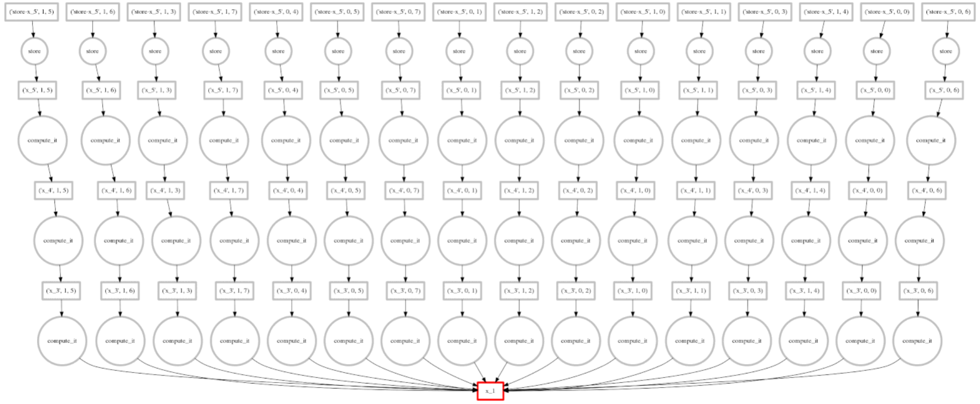
Hoyer, S. & Hamman, J., (2017). Xarray: N-D labeled Arrays and Datasets in Python. Journal of Open Research Software. 5(1), p.10. DOI: http://doi.org/10.5334/jors.148

DASK

- Dask is a flexible parallel computing library for analytic computing
- Parallel arrays allow us to seamlessly scale serial programs and workflows
- Dynamic task scheduling is optimized for computation
- Can be utilized on a single machine or a cluster of machines

Dask arrays coordinate many NumPy arrays arranged into a grid. These NumPy arrays may live on disk or on other machines.



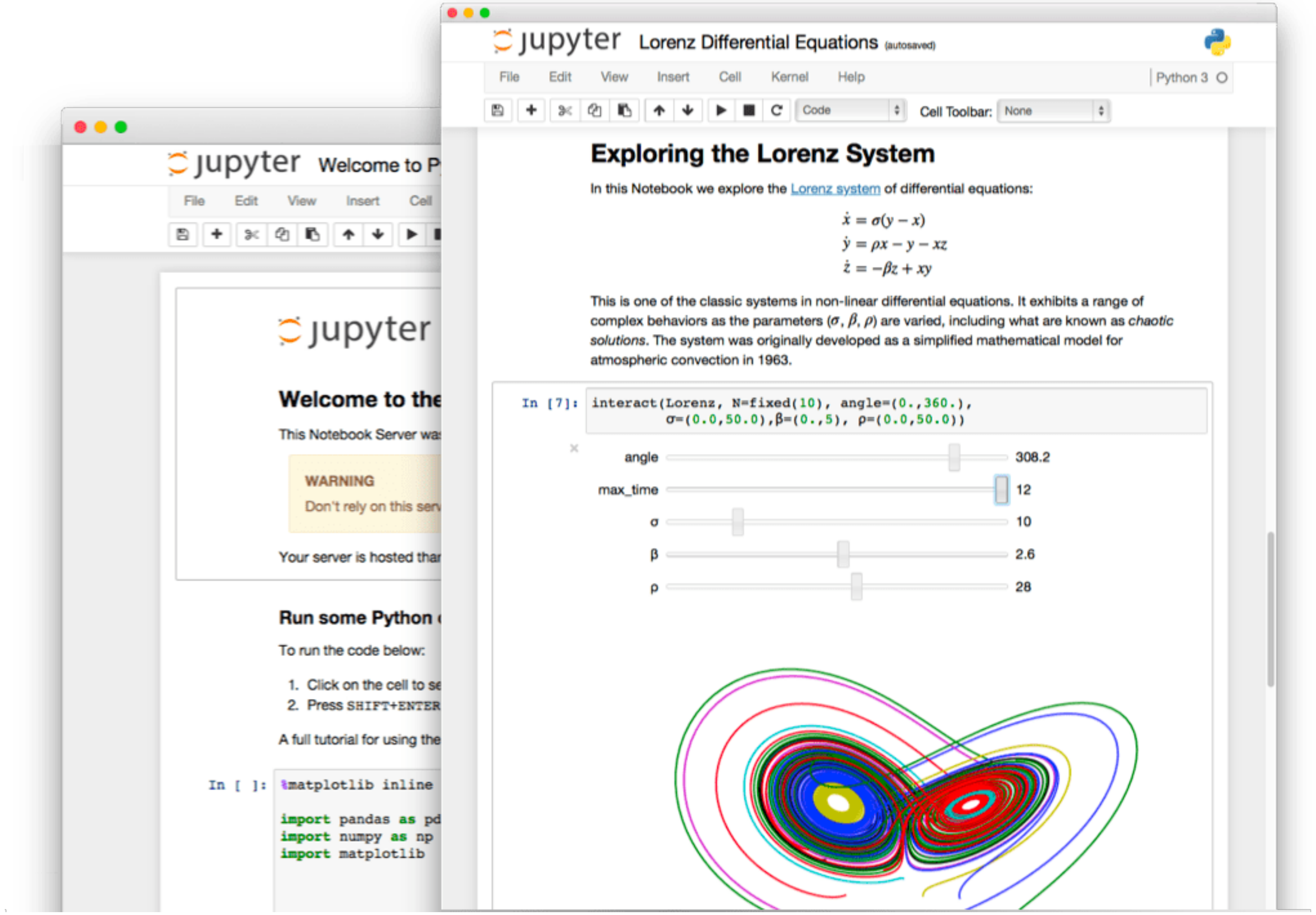


Example of a Dask task graph for a simple, embarrassingly parallel reduction operation.

Source: Dask documentation

JUPYTER

- *"The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and explanatory text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, machine learning and much more."*
- Originally "Python-centric" but has been expanded to include over 40 popular programming languages (e.g. Julia and R)
- Check it out at: http://jupyter.org



jupyter

Welcome to the

This Notebook Server was

WARNING

Don't rely on this serv

Your server is hosted than

Run some Python

To run the code below:

1. Click on the cell to se
2. Press SHIFT+ENTER

A full tutorial for using the

```
In [ ]: %matplotlib inline

        import pandas as pd
        import numpy as np
        import matplotlib
```

Exploring the Lorenz System

In this Notebook we explore the Lorenz system of differential equations:

$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = \rho x - y - xz$$
$$\dot{z} = -\beta z + xy$$

This is one of the classic systems in non-linear differential equations. It exhibits a range of complex behaviors as the parameters (σ, β, ρ) are varied, including what are known as *chaotic solutions*. The system was originally developed as a simplified mathematical model for atmospheric convection in 1963.

```
In [7]: interact(Lorenz, N=fixed(10), angle=(0.,360.),
                 σ=(0.0,50.0),β=(0.,5), ρ=(0.0,50.0))
```

angle 308.2

max_time 12

σ 10

β 2.6

ρ 28