The Pangeo Platform: a community-driven open-source big data environment

Joseph Hamman[1], Scott Henderson[2], Anthony Arendt[3], Amanda Tan[4], Dennis Fatland[5], Andrew Pawloski[6], Daniel Pilone[7], Matthew Hanson[8], Tom Augspurger[9], Ryan Abernathey[10], and Richard Signell[11]

[1]National Center for Atmospheric Research
[2]Cornell University
[3]University of Alaska Fairbanks
[4]University of Washington
[5]Unversity of Washington
[6]Element 84, Inc.
[7]Element 84
[8]Development Seed
[9]Anaconda Inc.
[10]Lamont -Doherty Earth Observatory
[11]NOAA

November 24, 2022

Abstract

In this presentation, we will describe the [Pangeo Project](http://pangeo.io), a coordinated community effort with support from NASA, NSF, AWS, Microsoft Azure and Google Cloud, to develop interactive and reproducible open source workflows for discovery, visualization, and quantitative analysis of large datasets used for research in the Earth Sciences. The Pangeo computational platform is based on JupyterHub and deployed wherever the data is stored. Python libraries such as Xarray, Rasterio, and Dask enable distributed parallel computations on HPC and Kubernetes clusters. We will discuss the design concepts central to the Pangeo platform and highlight specific applications using NASA satellite data archives on AWS. We will discuss recent progress in the integration of data discovery tools (e.g. STAC, CMR, Intake) with cloud-native storage formats for multidimensional data types (Cloud-Optimized Geotiff, Zarr, etc.) and highlight how they can be used to construct elegant, robust and reproducible scientific workflows. Finally, we will discuss performance, security, transferability across public cloud platforms, cost to operate, and approaches to encourage a cultural shift in scientific computation through educational events.

The Pangeo Platform

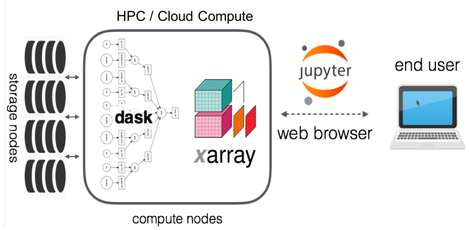
Joseph J. Hamman, NCAR
& The Pangeo Project

AGU Poster #IN11D-0693

The Pangeo Project

- Pangeo is a **community** promoting open, reproducible, and scalable science
- Pangeo is an integrated **ecosystem** of open source software tools
- Pangeo is a community **platform** for Big Data Geoscience

Architecture



The Pangeo Platform assembles a collection of unitary components:

- User Interface (e.g. Jupyter Notebooks)
- Data Model (e.g. Xarray, Iris, or Pandas)
- Parallel Job Distribution (e.g. Dask)
- Resource Management System (e.g. Kubernetes)
- Raw Storage System (e.g. Object Storage)
- High-level Data Broker (e.g. Intake)

Read more about the design principles behind Pangeo in our recent paper: https://arxiv.org/abs/1908.03356.

Community Deployments

The Pangeo Community supports multiple Pangeo deployments on Google Cloud Platform and Amazon Web Services. Log in with a **GitHub** ID:

Google Cloud Platform — hub.pangeo.io & ocean.pangeo.io

aws — aws-uswest2.pangeo.io & aws-useast1.pangeo.io

Microsoft Azure — ooi.pangeo.io

NCAR — jupyterhub.ucar.edu

Deploy Your Own Pangeo

Pangeo can be easily deployed on any Kubernetes or HPC Cluster. Learn more about the setup and deployment process: *pangeo.io/setup_guides*.

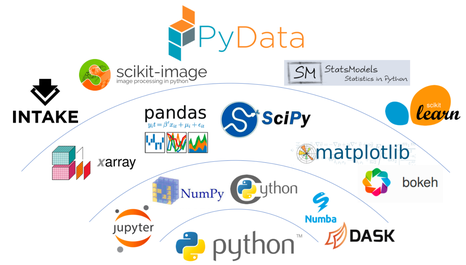
The Pangeo Project is supported, in part, by the National Science Foundation and the National Aeronautics and Space Administration.

PANGEO

A community-driven, open source, big data platform for the geosciences.

Take a picture to learn more about the Pangeo Project

Scientific Python Ecosystem



Pangeo integrates across the thriving open source scientific Python ecosystem. Below, we highlight a few of the core software libraries Pangeo uses.

Interactive computing

jupyterhub — Multi-user gateway to single-user Jupyter Notebooks, *jupyter.org/hub*

lab — Web-based user interface for Jupyter Notebooks, *jupyterlab.readthedocs.io*

Data Search and Discovery

INTAKE — General interface for cataloging and loading data in Python, *intake.readthedocs.io*

SpatioTemporal Asset Catalog — Intake driver for loading collections of Earth Observation data, *intake-stac.readthedocs.io*

INTAKE ESM — Intake driver for loading catalogs of climate model data, *intake-esm.readthedocs.io*

```
[2]: cat = intake.Catalog("catalog.yaml")  # load an Intake Catalog
     ds = cat["gmet_v1"].to_dask()[['pcp']]  # load an xarray dataset from catalog
     display(ds)

     xarray.Dataset
       ▸ Dimensions:      (ensemble: 100, lat: 224, lon: 464, time: 12054)
       ▸ Coordinates:     (4)
       ▾ Data variables:
           pcp            (ensemble, time, lat, lon)  float64  dask.array<chunksize=(1, 366, 224, 46...
       ▸ Attributes:      (6)
```

Data Analysis

DASK — Flexible parallel computing library for analytics, *dask.org*

xarray — N-D labeled arrays and datasets in Python, *xarray.pydata.org*

```
[3]: # Calculate the monthly climatology
     climatology = ds.groupby('time.month').mean()
     climatology

[3]: xarray.Dataset
       ▸ Dimensions:      (ensemble: 100, lat: 224, lon: 464, month: 12)
       ▸ Coordinates:     (4)
       ▾ Data variables:
           pcp            (month, ensemble, lat, lon)  float64  dask.array<chunksize=(1, 1, 224, 464)...
       ▸ Attributes:      (0)

[4]: client = dask.distributed.Client()  # start a Dask Cluster
     client

     climatology.persist()  # start computation in the background
```

Data Storage

netCDF — N-D array-oriented scientific data, *unidata.ucar.edu/software/netcdf*

zarr — Cloud friendly, chunked, compressed, N-D arrays, *zarr.readthedocs.io*

Filesystem-Spec — File-system-like abstractions for remote data, *filesystem-spec.readthedocs.io*