Pangeo community platform for scientific data processing

OMP 2019/04/02, Guillaume Eynard-Bontemps, CNES/Pangeo
Problems

- Data volume crisis in (geo)sciences
- Software multiplication, non reproducibility
- Many copies of the same datasets
- Local vs HPC vs Cloud
- Technology gap: industry vs academia

Mission

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

Goals/vision

- Foster collaboration around the open source Scientific Python ecosystem:
  - open and collaborative development
  - Welcoming and inclusive culture
- Support the development with domain-specific (geo)science and transverse packages
- Improve scalability of these tools to handle gigabytes to petabyte-scale datasets
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 calculus
- Signal processing
- Thermodynamics
• “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
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**

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
• 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

## Build Your Own Pangeo

<table>
<thead>
<tr>
<th>Storage Formats</th>
<th>HDF</th>
<th>OPeNDAP</th>
<th>Cloud Optimized COG/Zarr/Parquet/etc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ND-Arrays</td>
<td>NumPy</td>
<td>DASK</td>
<td>More coming…</td>
</tr>
<tr>
<td>Data Models</td>
<td>xarray</td>
<td>Iris</td>
<td></td>
</tr>
<tr>
<td>Processing Mode</td>
<td>jupyter</td>
<td>Interactive</td>
<td>Batch</td>
</tr>
<tr>
<td>Compute Platform</td>
<td>HPC</td>
<td>Cloud</td>
<td>Local</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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-jobqueue¹

¹: https://github.com/dask/dask-jobqueue
Welcome to the Pangeo public cloud deployment!

**hub.pangeo.io**

**pangeo.binder.io**

JupyterHub/BinderHub running on the Google Cloud

- Kubernetes for both Jupyter and Dask-distributed
  - Dask-kubernetes
- Exploring/evaluating:
  - Cloud storage
  - User environment customization
  - Data discovery
- Kubernetes Helm-chart (github.com/pangeo-data/helm-chart)
- CI/CD with Hubploy and CircleCI
- Deployments exist on AWS and Azure.
Pangeo vs state of the art

<table>
<thead>
<tr>
<th>Spark</th>
<th>DASK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mature</td>
<td>Less Mature</td>
</tr>
<tr>
<td>Robust</td>
<td>Pretty strong</td>
</tr>
<tr>
<td>JVM/Python</td>
<td>Python only</td>
</tr>
<tr>
<td>Query optimized</td>
<td>Science optimized</td>
</tr>
<tr>
<td>Collections &amp; Dataframes</td>
<td>Collections, DF, Arrays, Futures…</td>
</tr>
<tr>
<td>Python overhead</td>
<td>Python only</td>
</tr>
<tr>
<td>For big tabular data</td>
<td>For science data</td>
</tr>
<tr>
<td>Hadoop/Cloud/HPC</td>
<td>Hadoop/Cloud/HPC</td>
</tr>
</tbody>
</table>

Laptop to cluster
Serverless
NetCDF/TIFF no ingestion
Scales with Dask
Python only
Can build array db with Pangeo (Open data Cube)

I don’t know much about Array DBs!!!
HPC (HAL)
- 500 Tflops CPU
- 460 batch servers / 11K cores
- 8 interactive servers pre/post processing w/ GPU
- 8 PB GPFS / 250TB burst buffer / 100GBs bandwidth
- Low latency network
- GPGPU Nvidia Volta V100, 240Tflops

HPC DRSF (Ktulu)
- 20 Tflops
- 2 interactive servers pre/post processing w/ GPU
- 24 servers / 576 cores
- 120TB GPFS
- Low latency network
HPC use cases in CNES

Two main kinds of processing

**Numerical simulation (HPC)**
- Upstream phase, R&D
- Highly optimized technics
- Fine grain parallelism
*Trends*: multiscale, multiphysics

**Data Processing (HTC)**
- Downstream phase, operation
- Sensors data → scientific data
- Coarse grain parallelism
*Trends*: data volume explosion
JupyterHub and notebooks for interactive computing
- Hub on a VM with qsub access
- Batchspawner, Wrapspawner
- dask.distributed: parallel workers across many HPC nodes
- Xarray for computational toolkit and I/O
- New tool for deploying dask clusters on HPC: dask-jobqueue
  - Start a cluster from a notebook
  - Interactive (or not) distributed computing
  - Auto scaling capabilities
Demos @ CNES

Dask and dask-jobqueue basic example

Some realistic workload

Image processing: NDVI

Going deeper with Xarray
Pangeo Cloud use

Geoscience use cases:
http://pangeo.io/use_cases/index.html

Astronomy with GAIA catalog:
https://github.com/pangeo-data/pangeo/issues/255#issuecomment-427186915

Image processing and visualization
Conclusions

• Pangeo ecosystem greatly facilitates distributed computing and data analysis at scale
• It changes ways of doing it too
• Non monolithic platform built on top of existing Scientific Python stack and new related packages
• Community is always here to help
• Dask more versatile and easy to use than Spark.

Next steps

• Broaden users and use cases at CNES
• Encourage people to get in touch with Pangeo community
• Work in cooperation with others (Ongoing with Ifremer and CLS on SWOT aval data processing)
• Get involved!!
• Pangeo french meeting ont May 23rd

Pangeo wesite and discussions:
https://pangeo.io
https://github.com/pangeo-data/pangeo/issues
https://medium.com/pangeo

Pangeo Example + Binder:
https://github.com/pangeo-data/pangeo-example-notebooks
http://binder.pangeo.io/v2/gh/pangeo-data/pangeo-example-notebooks/master

Dask jobqueue:
https://github.com/dask/dask-jobqueue

Dask simple examples:
https://github.com/dask/dask-examples

My email
Guillaume.Eynard-Bontemps@cnes.fr
Github and binder link:
https://github.com/guillaumeeb/pangeo-tutorial-agu-2018

Dask only tutorial:

Other binder resources:
https://github.com/pangeo-data/pangeo-example-notebooks
https://github.com/pangeo-data/pangeo_ocean_examples
https://github.com/dask/dask-examples