SCALING ANALYTICS WITH PANGEO: XARRAY + DASK + ZARR

RYAN ABERNATHEY
CEOS WGISS-49
WHO AM I?

Physical Oceanographer

Ph.D. From MIT, 2012

Associate Prof. at Columbia / LDEO

https://ocean-transport.github.io/

Core developer of Zarr

Core developer of Xarray

Co-founder of Pangeo

Open Source Advocate
WHAT IS PANGEO?

“A community platform for Big Data geoscience”

• Open Community
• Open Source Software
• Open Source Infrastructure
Grass-Roots Adoption

One of the highlights of AGU19 for me was a workshop on @pangeo_data. If you're interested in earth sci + geospatial analysis at scale I can't recommend their tutorial enough. Find it at github.com/pangeo-data/pangeo-tutorials. Clear and concise intro to #xarray, #dask, #geopandas, and #wotnot.

It's taken a while, but I think that the whole xarray/Dask/cloud thing has finally clicked!

I've been working with CESM Large Ensemble data for a few months now - moved onto a Pangeo server and managed to speed up my workflow massively!

@xarray_dev, @dask_dev, @pangeo_data

#cmip6hack is just wrapping up, and has changed the way I think about, and hopefully do, climate model analysis in the future. The @pangeo_data infrastructure makes it all so easy.

#AGU2019 #OpenSource
Inspiration: Stephan Hoyer, Jake Vanderplas (SciPy 2015)
“Project Jupyter exists to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages.”
Data variables
used for computation

Coordinates
describe data

Indexes
align data

Attributes
metadata ignored by operations

“netCDF meets pandas.DataFrame”

https://github.com/pydata/xarray
```python
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()
```
Flexible, general-purpose parallel computing framework.

Complex computations represented as a graph of individual tasks.

Scheduler optimizes execution of graph.

ND-Arrays are split into chunks that comfortably fit in memory.
Flexible, general-purpose parallel computing framework.

Complex computations represented as a graph of individual tasks.

Scheduler optimizes execution of graph.

ND-Arrays are split into chunks that comfortably fit in memory.
Zarr is an open source library for storage of chunked, compressed ND-arrays created by Alistair Miles (Imperial) for genomics research (@alimanfoo); it is now a community-supported standard.

- Arrays are split into user-defined chunks; each chunk is optional compressed (zlib, zstd, etc.).
- Can store arrays in memory, directories, zip files, or any python mutable mapping interface (dictionary).
- External libraries (s3fs, gcsf) provide a way to store directly into cloud object storage.
- Implementations in Python, C++, Java (N5), Julia, Javascript

https://zarr.readthedocs.io/
PANGEO ARCHITECTURE

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

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

Dask tells the nodes what to do.

Cloud / HPC

Jupyter for interactive access remote systems

Xarray provides data structures and intuitive interface for interacting with datasets

Distributed storage

web browser

end user
PANGEO DEPLOYMENTS

NASA Pleiades

NCAR Cheyenne

OCEAN.PANGEO.IO

Google Cloud Platform

Microsoft Azure

aws

HTTP://PANGEO.IO/DEPLOYMENTS.HTML
Live example at
from intake import open_catalog

cat = open_catalog("https://raw.githubusercontent.com/pangeo-data/pangeo-datastore/master/intake-catalogs/ocean.yaml")

ds = cat["sea_surface_height"].to_dask()

ds

Dimensions:
(latitude: 720, longitude: 1440, nv: 2, time: 8901)

Coordinates:

<table>
<thead>
<tr>
<th>crs</th>
<th>()</th>
<th>int32 ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>lat_bnds</td>
<td>(time, latitude, nv)</td>
<td>float32</td>
</tr>
<tr>
<td>latitude</td>
<td>(latitude)</td>
<td>float32</td>
</tr>
<tr>
<td>lon_bnds</td>
<td>(longitude, nv)</td>
<td>float32</td>
</tr>
<tr>
<td>longitude</td>
<td>(longitude)</td>
<td>float32</td>
</tr>
<tr>
<td>nv</td>
<td>(nv)</td>
<td>int32 0 1</td>
</tr>
<tr>
<td>time</td>
<td>(time)</td>
<td>datetime64[ns] 1993-01-01 ... 2017-05-15</td>
</tr>
</tbody>
</table>

Data variables:

| adt   | (time, latitude, longitude) | float64 | dask.array<chunksize=(5, 720, ... |

Array | Chunk |
---|---|
Bytes | 73.83 GB | 41.47 MB |
Shape | (8901, 720, 1440) | (5, 720, 1440) |
Count | 1782 Tasks | 1781 Chunks |
Type | float64 | numpy.ndarray |
ds.sla.hvplot.image('longitude', 'latitude',
      rasterize=True, dynamic=True, width=800, height=450,
      widget_type='scrubber', widget_location='bottom', cmap='RdBu_r')

Time: 1993-01-01 00:00:00
Create and Connect to Dask Distributed Cluster

```python
from dask_gateway import Gateway
from dask.distributed import Client

gateway = Gateway()
cluster = gateway.new_cluster()
cluster.adapt(minimum=1, maximum=20)
cluster
```

**GatewayCluster**

<table>
<thead>
<tr>
<th>Workers</th>
<th>Cores</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0 B</td>
</tr>
</tbody>
</table>

- **Manual Scaling**
- **Adaptive Scaling**

Name: prod.f855cbed758f4a628bfc7190df860ad3

Dashboard: https://hub.binder.pangeo.io/services/dask-gateway/clusters/prod.f855cbed758f4a628bfc7190df860ad3/status
```python
# the computationally intensive step
sla_timeseries = ds.sla.mean(dim=('latitude', 'longitude')).load()

sla_timeseries.plot(label='full data')
sla_timeseries.rolling(time=365, center=True).mean().plot(label='rolling annual mean')
plt.ylabel('Sea Level Anomaly [m]')
plt.title('Global Mean Sea Level')
plt.legend()
plt.grid()
```
Sharing Data in the Cloud Era

Pangeo Approach: Direct Access to Cloud Object Storage
Do we need an API for data access?
**OPENDAP**

✅ Access remote netCDF-style datasets over HTTP

✅ Subset based on coordinates / variables

✅ Load data lazily

❌ Requires a server

**XARRAY + ZARR**

✅ Access remote netCDF-style datasets over HTTP

✅ Subset based on coordinates / variables

✅ Load data lazily

✅ Serverless (only uses S3)
**OPENDAP**

- Access remote netCDF-style datasets over HTTP
- Subset based on coordinates / variables
- Load data lazily

**XARRAY + ZARR**

- Access remote netCDF-style datasets over HTTP
- Subset based on coordinates / variables
- Load data lazily
- Serverless (only uses S3)

× Requires a server
OPENDAP

✅ Access remote netCDF-style datasets over HTTP
✅ Subset based on coordinates / variables
✅ Load data lazily
❌ Requires a server

XARRAY + ZARR

✅ Access remote netCDF-style datasets over HTTP
✅ Subset based on coordinates / variables
✅ Load data lazily
✅ Serverless (only uses S3)
CLOUD STORAGE
THROUGHPUT BENCHMARKS

To Google Cloud Region US-CENTRAL
New climate model data now in Google Public Datasets
Open source scientific python provides a great foundation for scalable earth system analytics (especially in the cloud). **Use it, don’t reinvent it!**

How do we support / sustain open-source foundational software tools? (No agency or lab “owns” these, but they are critical infrastructure.)

**idea:** pay your staff to contribute to Xarray, Dask, etc., don’t just build on top of them

The best way to take advantage of cloud is to give users direct access to analysis-ready data in object storage. Don’t hide it behind an API.
LEARN MORE

http://pangeo.io

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

https://medium.com/pangeo

@pangeo_data
PANGEO CLOUD STACK

- **GCM**
- **climpred**
- **xarray**: High-level API for analysis of multidimensional labelled arrays.
- **DASK**: Flexible, general-purpose parallel computing framework.
- **Zarr**: Cloud-optimized storage for multidimensional arrays.

Kubernetes

Object Storage
ZARR

Example .zarray file (json)

```json
{
  "chunks": [
    5,
    720,
    1440
  ],
  "compressor": {
    "blocksize": 0,
    "clevel": 3,
    "cname": "zstd",
    "id": "blosc",
    "shuffle": 2
  },
  "dtype": "<f8",
  "fill_value": "NaN",
  "filters": null,
  "order": "C",
  "shape": [8901, 720, 1440]
}
```

Zarr Group: `group_name`

Zarr Array: `array_name`
Zarr

Example .attrs file (json)

```
{
    "_ARRAY_DIMENSIONS": [
        "time",
        "latitude",
        "longitude"
    ],
    "comment": "The sea level anomaly is the sea surface height above mean sea surface; it is referenced to the [1993, 2012] period; see the product user manual for details",
    "coordinates": "crs",
    "grid_mapping": "crs",
    "long_name": "Sea level anomaly",
    "standard_name": "sea_surface_height_above_sea_level",
    "units": "m"
}
```