Crunching Model Data in the Cloud

Rich Signell, USGS

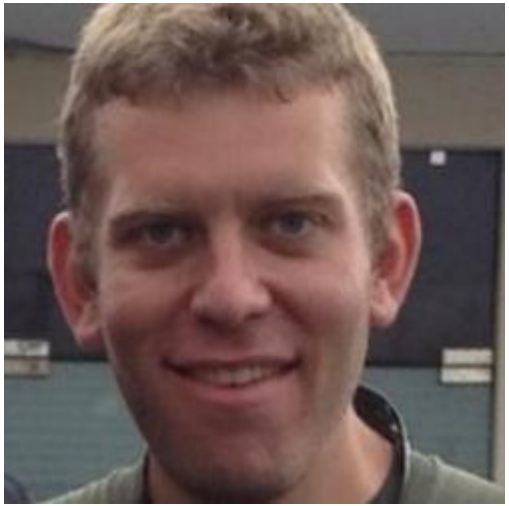
ESIP Summer Meeting, July 17, 2019

Storming the Cloud

Rich Signell, USGS

ESIP Summer Meeting, July 17, 2019

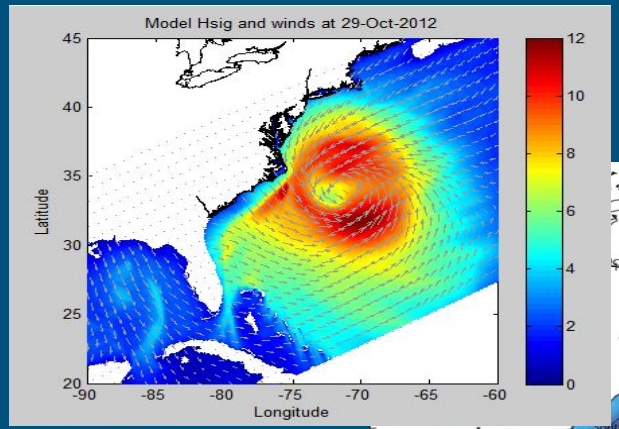
The Story of Dan



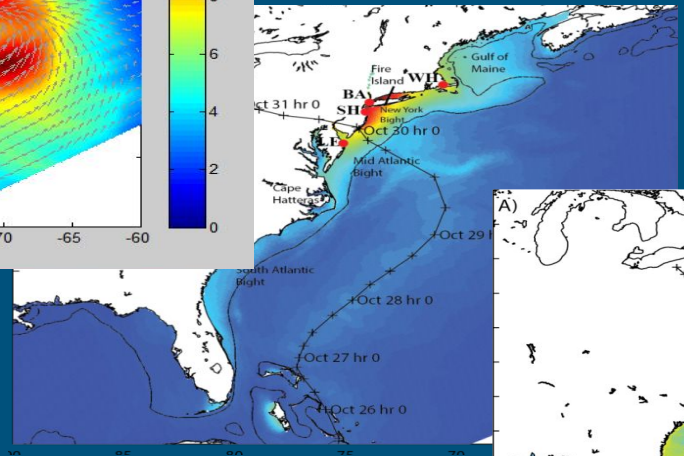
Rich Signell
USGS
ESIP Summer Meeting, July 17, 2019

Sediment Transport Modeling at USGS

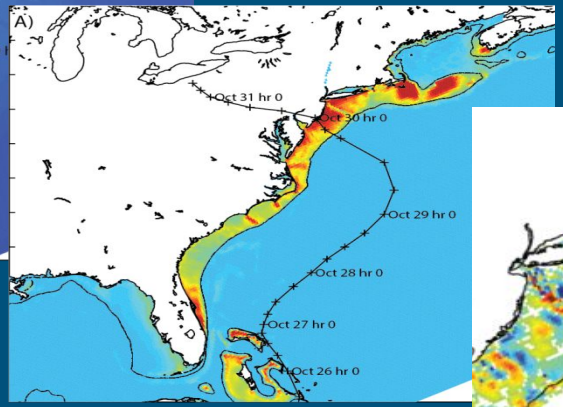
Wind and Waves



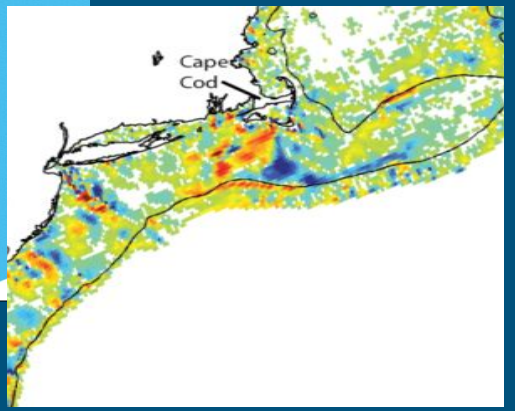
Water Levels



Bottom Stress



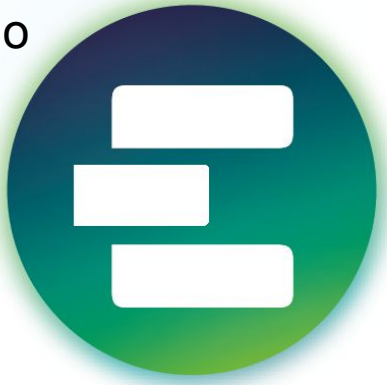
Sea Bed Erosion



COAWST: Coupled Ocean, Atmosphere, Wave, Sediment Transport modeling system (John Warner, USGS)

200TB of NetCDF files

http://pangeo.io



PANGEO

A community platform for Big Data geoscience

Pangeo is an open group. Anyone who agrees with our mission and vision is welcome to join.

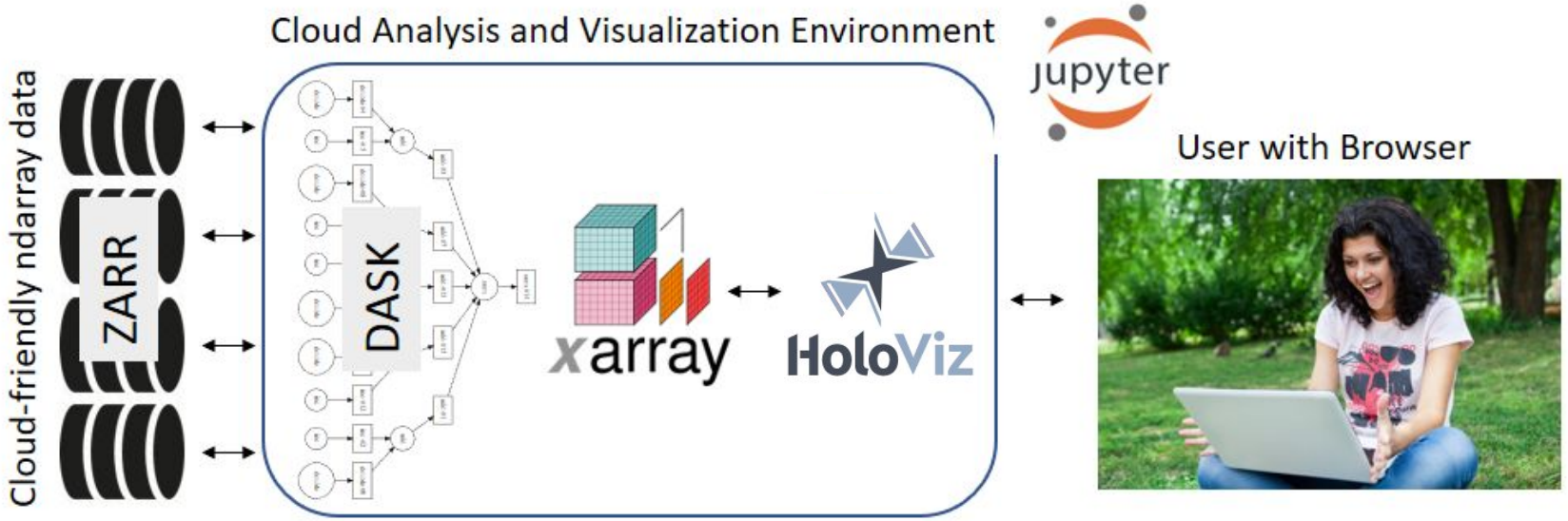
To add your name to the list, fork the source for this site on GitHub, add you details to the file *_data/people.yml*, and submit a pull request. The easiest wa to do this is to directly edit the file on GitHub.

Ryan Abernathey

Lamont Doherty Earth Observatory

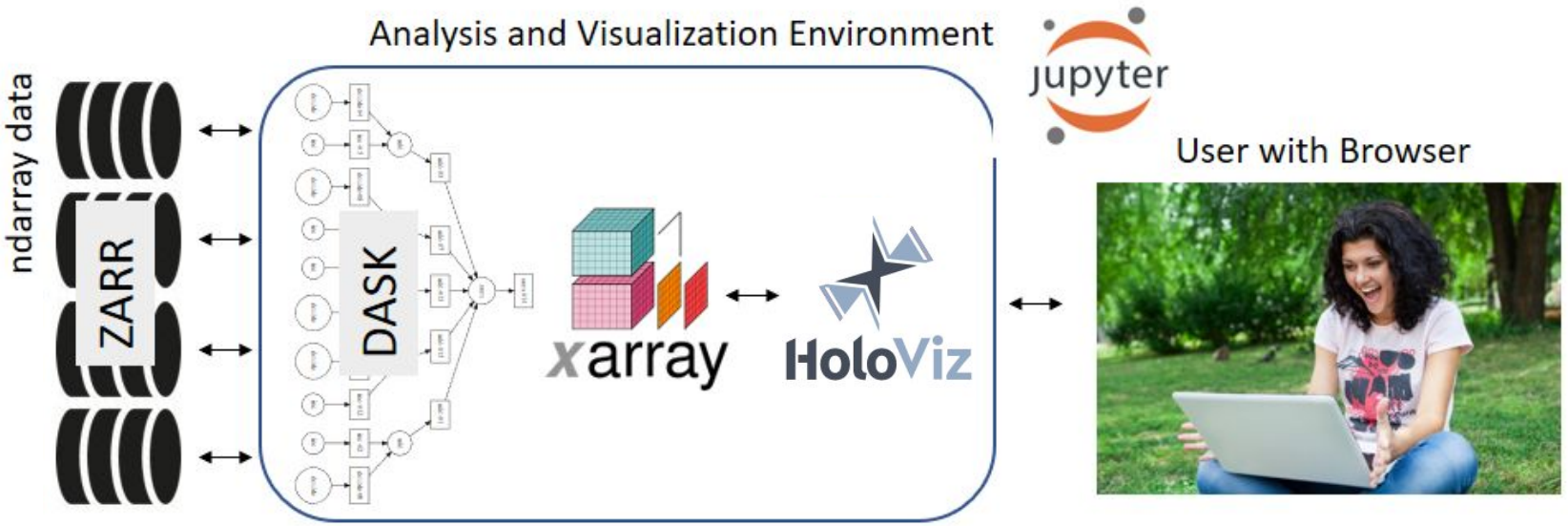
- First ever Pangeo developers meeting? community — 12
  #199 opened 17 days ago by niallrobinson
- How fast can the Met Office's solution pull data from S3? — 16
  #198 opened 17 days ago by mrocklin
- Pangeo use case: Advanced regridding using ESMF/ESMpy/OCGIS/xESMF/Xarray/Dask — 18
  #197 opened 17 days ago by jhamman
- intermittent errors during blosc decompression of zarr chunks on pangeo.pydata.org — 14
  #196 opened 19 days ago by rabernat

Pangeo Cloud Environment



Dask Cluster runs on Kubernetes

Pangeo Laptop/HPC Environment



Dask Cluster runs on HPC or local machine

Run Jupyter on remote machine

On remote machine: $ bash start_jupyter.sh

```
#!/bin/bash
jupyter notebook --no-browser --ip=`hostname` --port=8888
echo "ssh -N -L 8888:`hostname`:8888 -L 8787:`hostname`:8787
$USER@gamone.whoi.edu”
```

On local machine:

```
$ ssh -N -L 8888:gamone:8888 -L 8787:gamone:8787 rsignell@gamone.whoi.edu
```

Dan's Task:
Calculate mean salinity from 700GB dataset

Method #1: Run Python locally, download 700GB data, access Netcdf files locally

Method #2: Run Python locally, access remote data using OPeNDAP

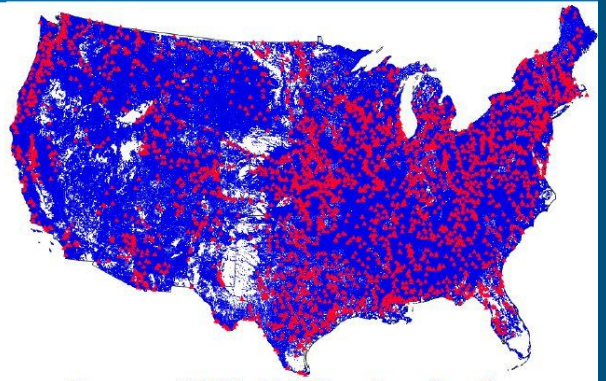
Method #3: Run Python remotely on Sand, access NetCDF files on Sand

Method #4: Run Python remotely on Poseidon, access NetCDF files on Poseidon

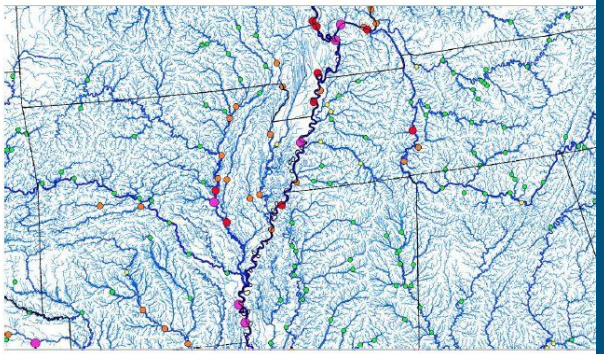
Method #5: Run Python on cloud, access Zarr dataset from Cloud

NWM V1.0 Output (Cosgrove et al)

- **Hydrologic Output**
  - River channel discharge and velocity at 2.7 million river reaches
  - Reservoir inflow, outflow, elevation
  - Ponded water depth and depth to saturation (250 m CONUS+ grid)
- **Land Surface Output**
  - 1km CONUS+ grid
  - Soil and snow pack states
  - Energy and water fluxes
- **Direct-output and derived products** (e.g. stream flow anomalies)



Current NWS AHPS points (red)
NWM output points (blue)



Current NWS River Forecast Points (circles)
Overlaid with NWM Stream Reaches

One month of forcing and output is 15TB

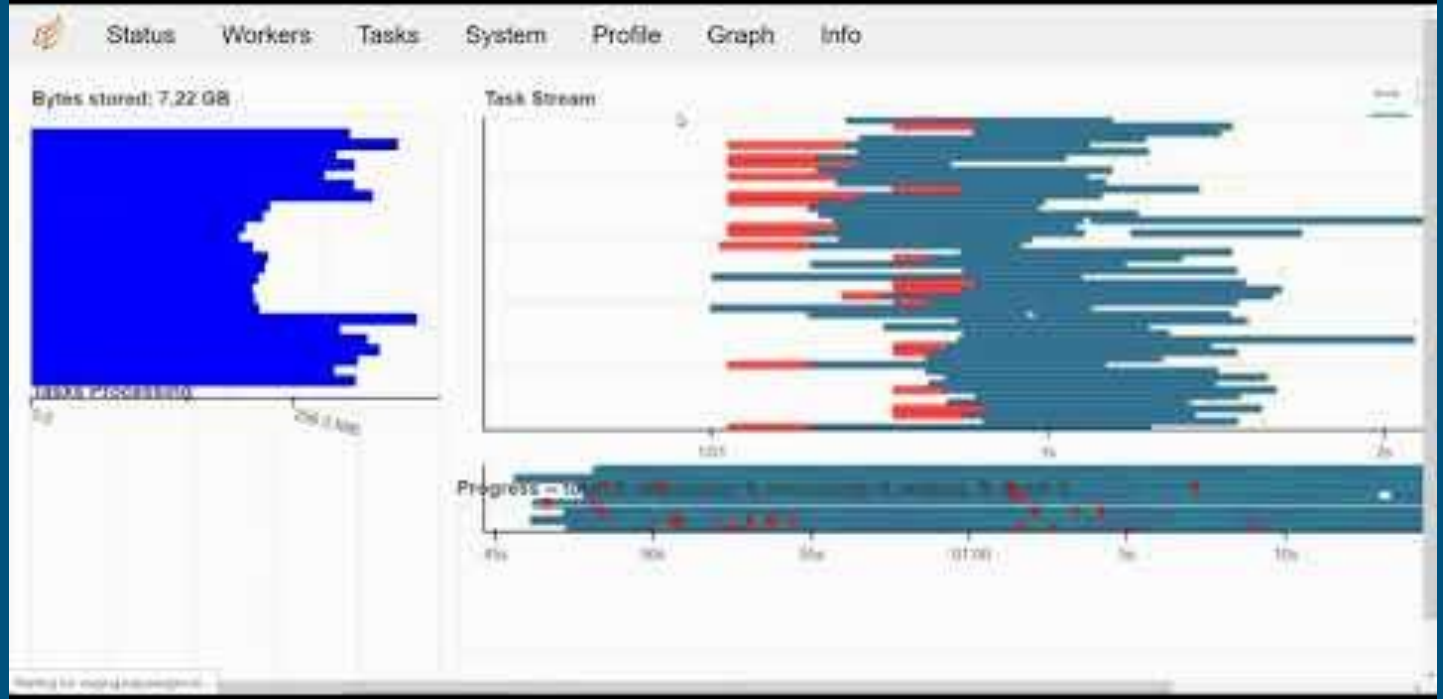
NWM is part of the Big Data Project

Forecast data is being pushed to AWS

(24 year retrospective is available on Open Commons Consortium and AWS)

But there is a problem...

National Water Model Demo



pangeo.esipfed.org/user/rsignell-usgs/notebooks/NWM/Read_Zarr_s3.ipynb

jupyter Read_Zarr_s3 Last Checkpoint: 09/05/2018 (autosaved)

Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help Trusted Python [default]

Read National Water Model (NWM) model data from Zarr

In [1]:
```
%matplotlib inline

from dask.distributed import Client, progress, LocalCluster
from dask_kubernetes import KubeCluster
import xarray as xr
import s3fs
```

In [2]:
```
cluster = KubeCluster()
```

In [5]:
```
cluster.scale(20);
```

In [4]:
```
cluster
```

KubeCluster

Workers 20

Cores 40

Memory 120.00 GB

▸ Manual Scaling

▸ Adaptive Scaling

Dashboard: /user/rsignell-usgs/proxy/8787/status

In [6]:
```
client = Client(cluster)
```

In [7]:
```
# jetstream s3
# url='https://iu.jetstream-cloud.org:8080'
# fs = s3fs.S3FileSystem(client_kwargs=dict(endpoint_url=url), anon=True)
# s3map = s3fs.S3Map('rsignell/nwm/test_week', s3=fs)
```

In [8]:
```
# AWS s3
fs = s3fs.S3FileSystem(anon=True)
s3map = s3fs.S3Map('esipfed/pangeo/NWM/short_term_forcing', s3=fs)
```

pangeo.esipfed.org/user/rsignell-usgs/notebooks/NWM/Read_Zarr_s3.ipynb

jupyter Read_Zarr_s3 Last Checkpoint: 09/05/2018 (unsaved changes)

Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help Trusted Python [default]

In [7]:
```
# AWS s3
fs = s3fs.S3FileSystem(anon=True)
s3map = s3fs.S3Map('esipfed/pangeo/NWM/short_term_forcing', s3=fs)
```

In [8]:
```
ds = xr.open_zarr(s3map)
```

In [9]:
```
ds
```

Out[9]:
```
<xarray.Dataset>
Dimensions:   (time: 168, x: 4608, y: 3840)
Coordinates:
  * time      (time) datetime64[ns] 2018-04-01T01:00:00 2018-04-01T02:00:00 ...
  * x         (x) float64 -2.304e+06 -2.303e+06 -2.302e+06 -2.301e+06 ...
  * y         (y) float64 -1.92e+06 -1.919e+06 -1.918e+06 -1.917e+06 ...
Data variables:
    LWDOWN    (time, y, x) float64 dask.array<shape=(168, 3840, 4608), chunksize=(168, 384, 288)>
    PSFC      (time, y, x) float64 dask.array<shape=(168, 3840, 4608), chunksize=(168, 384, 288)>
    Q2D       (time, y, x) float64 dask.array<shape=(168, 3840, 4608), chunksize=(168, 384, 288)>
    RAINRATE  (time, y, x) float32 dask.array<shape=(168, 3840, 4608), chunksize=(168, 384, 288)>
    SWDOWN    (time, y, x) float64 dask.array<shape=(168, 3840, 4608), chunksize=(168, 384, 288)>
    T2D       (time, y, x) float64 dask.array<shape=(168, 3840, 4608), chunksize=(168, 384, 288)>
    U2D       (time, y, x) float64 dask.array<shape=(168, 3840, 4608), chunksize=(168, 384, 288)>
    V2D       (time, y, x) float64 dask.array<shape=(168, 3840, 4608), chunksize=(168, 384, 288)>
Attributes:
    model_initialization_time:  2018-04-01_00:00:00
    model_output_valid_time:    2018-04-01_01:00:00
```

In [10]:
```
var='T2D'
```

jupyter Read_Zarr_s3 Last Checkpoint: 09/05/2018 (unsaved changes)

Logout | Control Panel

File Edit View Insert Cell Kernel Widgets Help | Trusted | Python [default]

```
model_output_valid_time:    2018-04-01_01:00:00
```

In [11]:
```
var='T2D'
```

In [12]:
```
ds[var].nbytes/1.e9
```

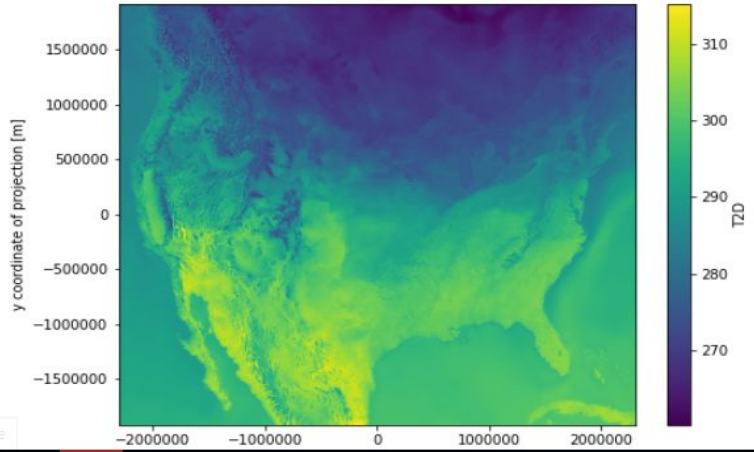
Out[12]:
```
23.78170368
```

In [13]:
```
uvar = ds[var].max(dim='time').persist()
progress(uvar)
```

Finished: 22.1s

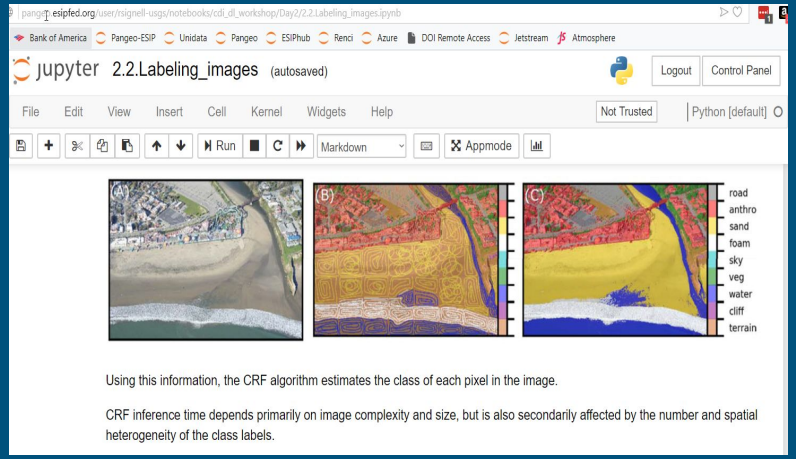
161 / 161		zarr
160 / 160		nanmax-nanmax-aggregate
160 / 160		nanmax-aggregate

In [14]:
```
isub=2
uvar[::isub,::isub].plot.imshow(figsize=(8,6));
```




Status
Workers
Tasks
System
Profile
Graph
Info
Bytes stored: 5.82 GB
0.0
256.0 MiB
512.0 Mi
Task Stream
54s
56s
58s
1/01
2s
4s
6s
8s
Tasks Processing
0
1
2
3
4
5
6
7
Progress -- total: 641, in-memory: 162, processing: 308, erred: 0
zarr
102 / 161
nanmax-aggre...
80 / 160
nanmax-nanma...
91 / 160
getitem
60 / 160

Pangeo is not just for big data...



USGS Deep Learning Workshops
Flagstaff, July 10-12 & Denver, Sep 25-27, 2018

35 Students used ESIP Pangeo on AWS with no issues.
Data was stored on S3, no data moved from Cloud.
Students only needed their web browsers.

```
[ec2-user@ip-172-31-29-161 ~]$ k get pods | grep jupyter
jupyter-30-2d06                  1/1       Running     0
jupyter-abock80                  1/1       Running     0
jupyter-afoxgrover-2dusgs        1/1       Running     0
jupyter-amun0113                 1/1       Running     0
jupyter-bletcher                 1/1       Running     0
jupyter-cjlegleiter              1/1       Running     0
jupyter-collincr                 1/1       Running     0
jupyter-corinacd                 1/1       Running     0
jupyter-couvillionb              1/1       Running     0
jupyter-cto22                    1/1       Running     0
jupyter-dbuscombe-2dusgs         1/1       Running     0
jupyter-debuna                   1/1       Running     0
jupyter-dmillar-2dusgs           1/1       Running     0
jupyter-dvaranka                 1/1       Running     0
jupyter-ebulliner-2dusgs         1/1       Running     0
jupyter-esturdivant-2dusgs       1/1       Running     0
jupyter-hollybeck1               1/1       Running     0
jupyter-jacurtis-2dusgs          1/1       Running     0
jupyter-jennabrown-2dusgs        1/1       Running     0
jupyter-johnsamstone             1/1       Running     0
jupyter-jwfulton-2dusgs          1/1       Running     0
jupyter-khop-2dusgs              1/1       Running     0
jupyter-leonfoks                 1/1       Running     0
jupyter-linhunt                  1/1       Running     0
jupyter-mjcashman                1/1       Running     0
jupyter-nrapstine-2dusgs         1/1       Running     0
jupyter-rsleeter0710             1/1       Running     0
jupyter-rviger-2dusgs            1/1       Running     0
jupyter-ryan-2dlima              1/1       Running     0
jupyter-sarundel                 1/1       Running     0
jupyter-talbertc-2dusgs          1/1       Running     0
jupyter-tmercier                 1/1       Running     0
jupyter-tnwillia                 1/1       Running     0
jupyter-tomtomatron              1/1       Running     0
jupyter-zdefne-2dusgs            1/1       Running     0
```

Pangeo is not just for model output

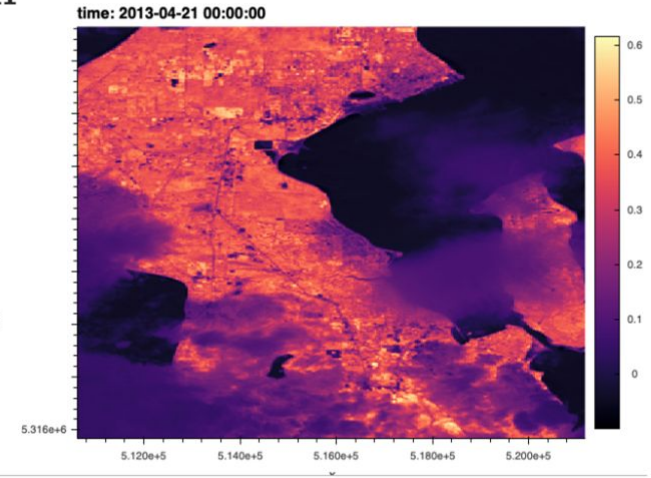
ABOUT PANGEO | WEBSITE

Cloud Native Geoprocessing of Earth Observation Satellite Data with Pangeo

Scott Henderson
Oct 1, 2018 · 7 min read

If you are familiar with satellite imagery you've likely heard that we are entering a "golden era" of Earth Observation. It's true! New satellites are generating Petabyte-scale publicly available archives of imagery at unprecedented rates, enabling new insights and fast global impacts.

Cloud Native Landsat Analysis with Pangeo



Full resolution NDVI calculated from Landsat 8 image from 2013–04–21

We've developed an example Cloud Native quantitative analysis of Landsat 8 satellite imagery. What is special about this example is that the analysis is easily reproduced, scalable, and interactive: 100 Gigabytes of Landsat 8 images covering Washington State (representing the entire archive back to 2013-03–21) are found using NASA's Common Metadata Repository (CMR). Then, using URLs instead of local file paths, the Normalized Difference Vegetation Index (NDVI), a simple landcover

Pangeo is not just for model output

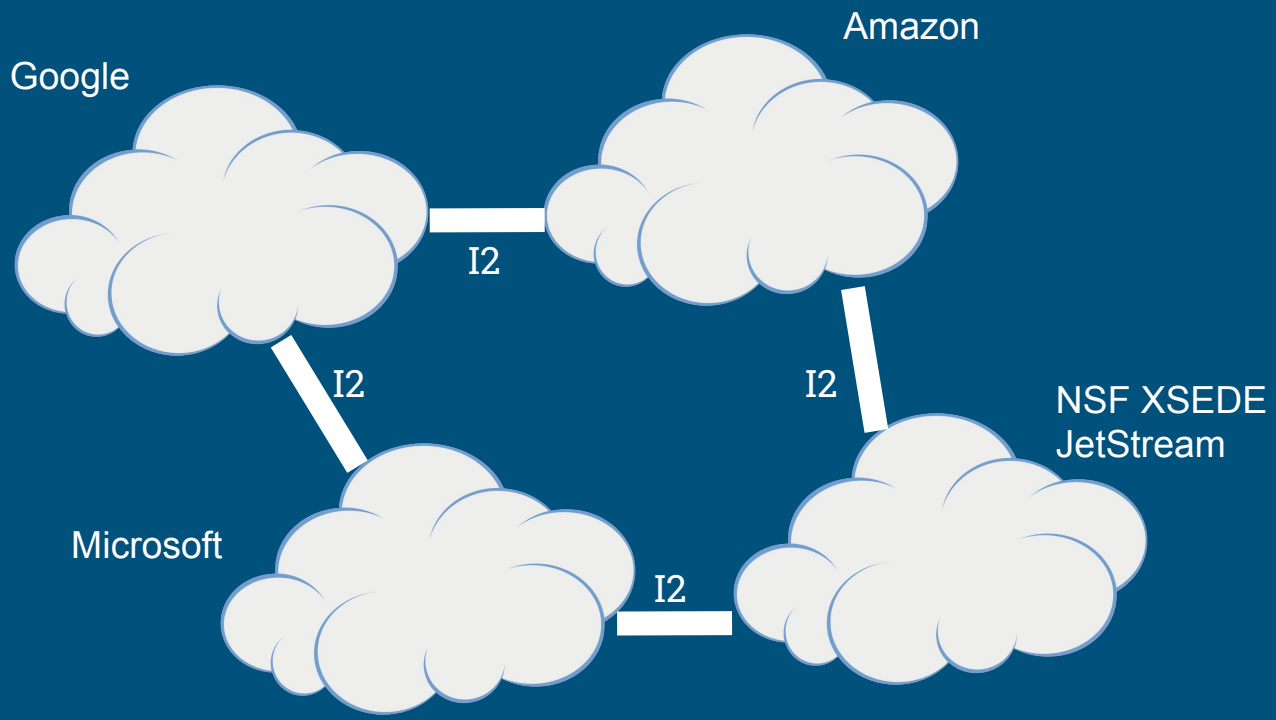
PanNeuro: leveraging a community-based approach for big data neuroscience

BRAIN Initiative PI meeting, April, 2019

Ariel Rokem
The University of Washington eScience Institute

Follow along at: https://arokem.github.io/2019-BRAINI-PanNeuro-slides/

The Future: All data proximate “in the I2 Sense”



Cloud Benefits: Scales with users and demand; Unlimited size datasets; Massive datasets can be crunched with many processors; Users don’t need to buy or maintain fancy hardware, software or fast internet

Cloud Benefits: Scales with users and demand; Unlimited size datasets; Massive datasets can be crunched with many processors; Users don’t need to buy or maintain fancy hardware, software or fast internet