The USGS EarthMap Pangeo: Success Stories and Lessons Learned

Rich Signell
Research Oceanographer
U.S. Geological Survey
Woods Hole, MA

with other USGS folks

Jeff Sadler, USGS
Renee Pieschke, KBR
Richie Erickson, USGS

and with help from:

Courtney Owens, USGS
George Rolston, USGS
Sam Congdon, USGS
Harry House, USGS

Scott Henderson, UW
Joe Hamman, NCAR
Amanda Tan, UW

ESIP Winter Meeting
Bethesda, MD: January 8, 2020

EarthMap: Situational Awareness Driven by Sensors + Models

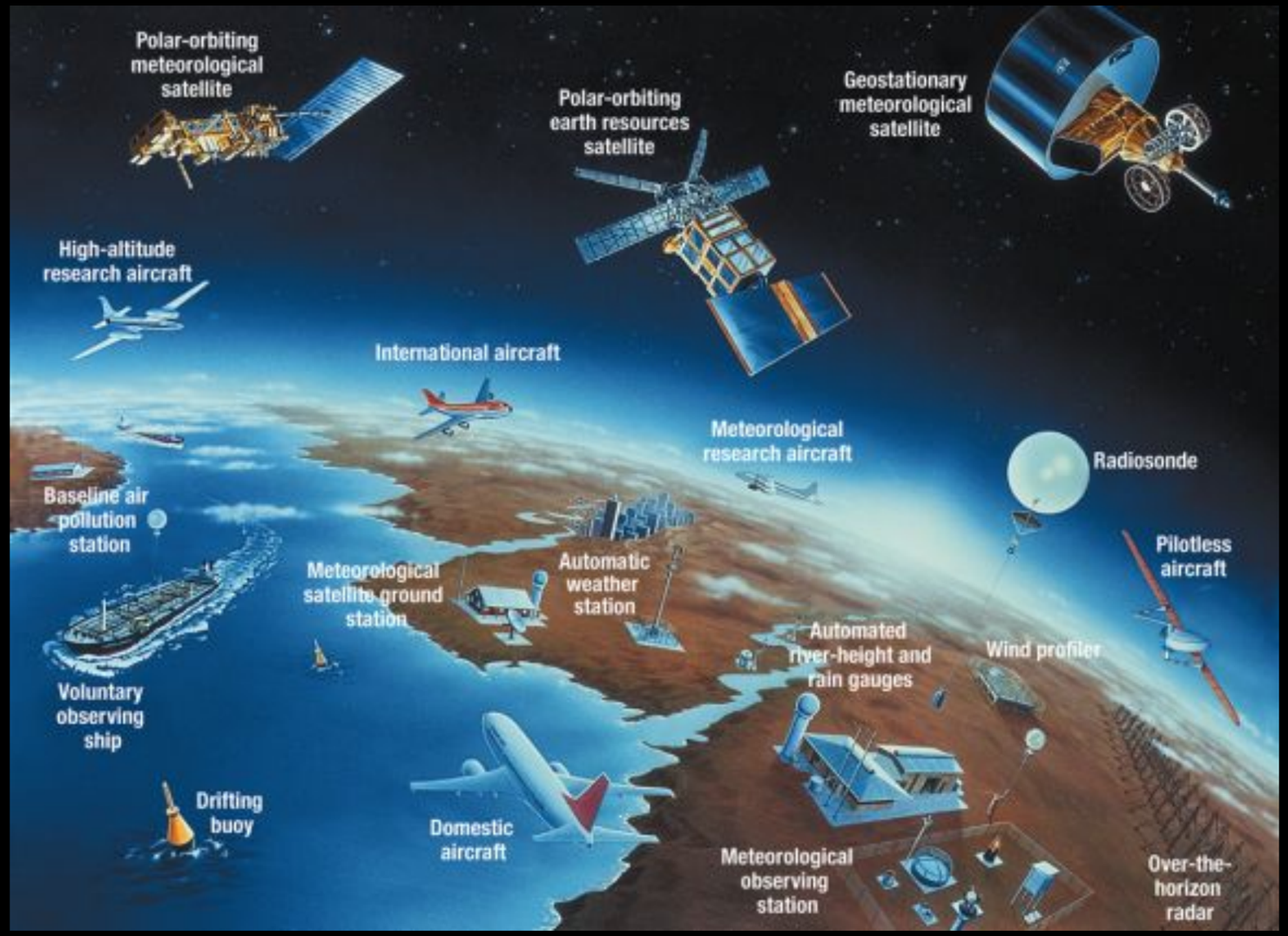


Image Credit: WMO

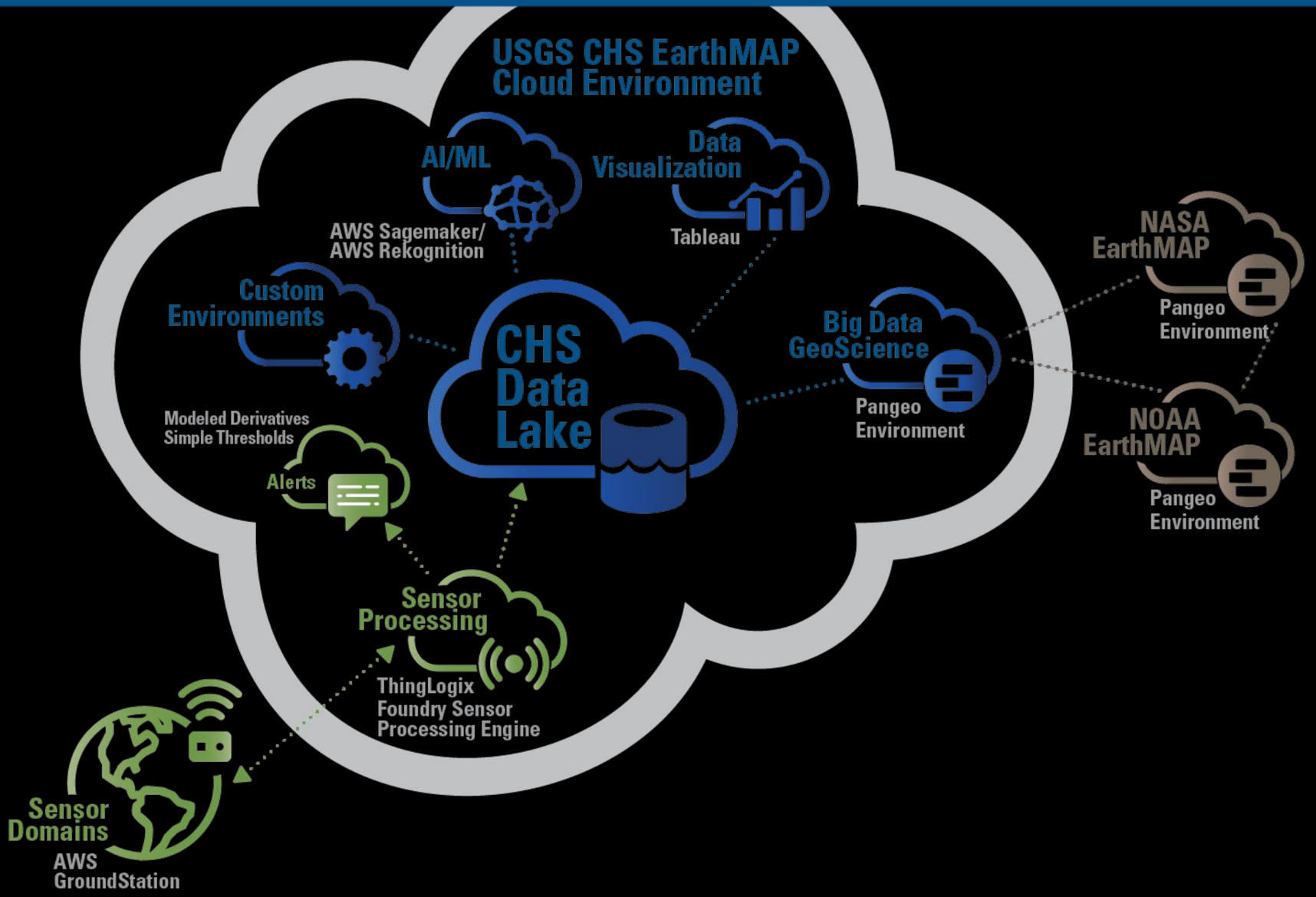
EarthMap: Situational Awareness Delivered Through Smart Devices

StormSense: Automated Flood Alerts Using Integrated Real-Time IoT Sensors (Loftis, VIMS)

Image Credit: Networkworld.com

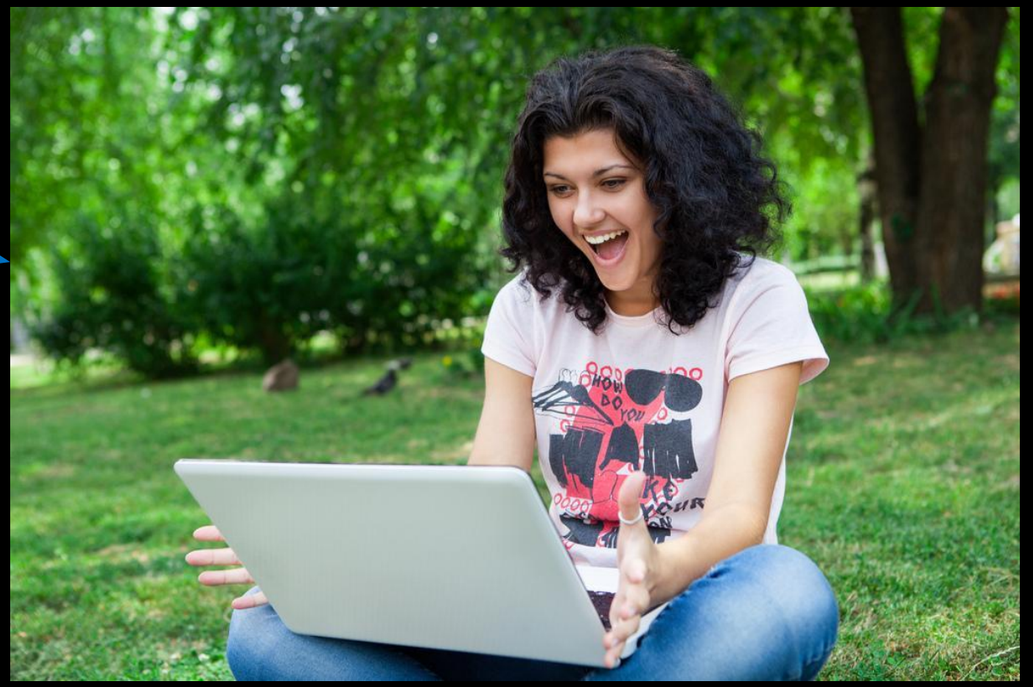
USGS Cloud Hosting Solutions EarthMAP Platform

Simplified Conceptual Phase 1 Architecture



Traditional Model Data Analysis

Model Data Analysis of the Future (available now!)



Pangeo is a Community

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 your details to the file *_data/people.yml*, and submit a pull request. The easiest way to do this is to directly edit the file on GitHub.

Ryan Abernathey

Lamont Doherty Earth Observatory
physical oceanography climate

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

NSF

LAMONT-DOHERTY EARTH OBSERVATORY
THE EARTH INSTITUTE AT COLUMBIA UNIVERSITY

UCAR
UNIVERSITY CORPORATION FOR ATMOSPHERIC RESEARCH

Met Office

Google

ANACONDA

aws

NASA

Microsoft Azure

QUEENS COLLEGE

ESIP

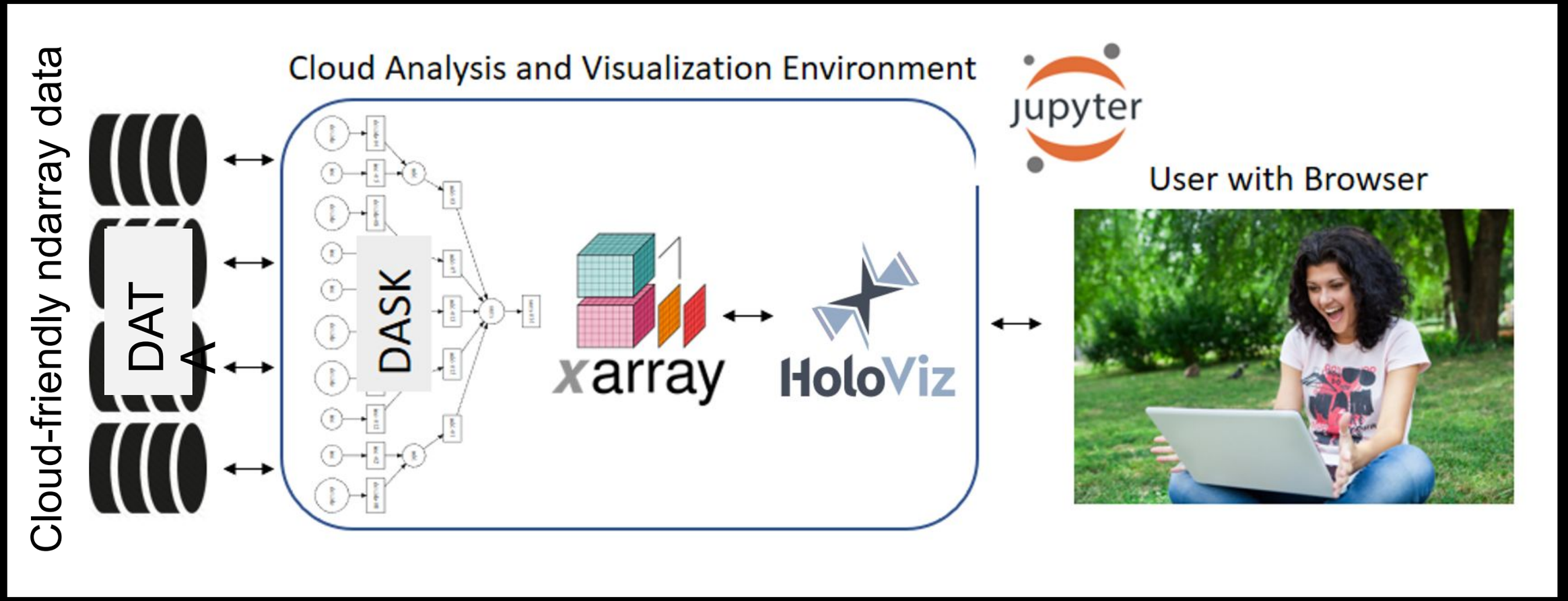
Element 84

UNIVERSITY of WASHINGTON eScience Institute

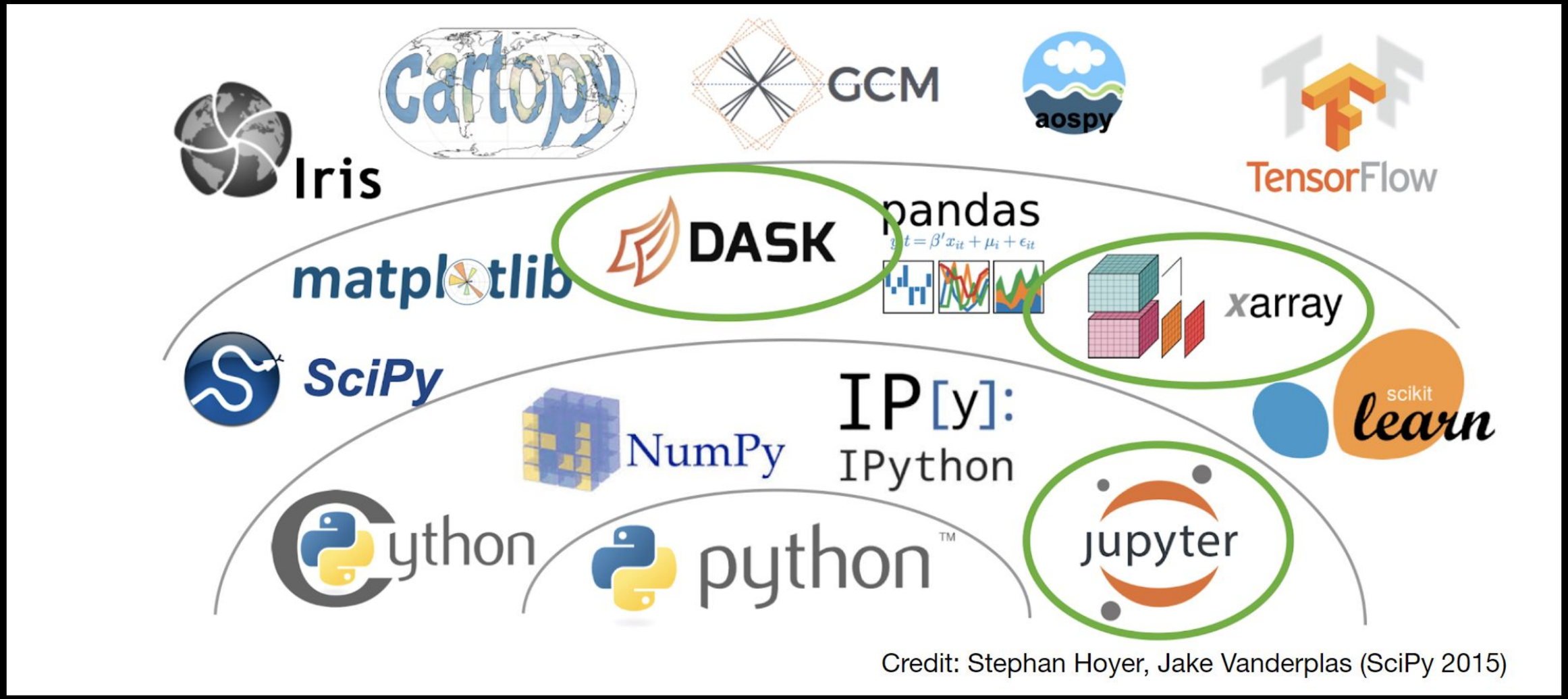
CSIRO

USGS
science for a changing world

Pangeo Cloud/HPC Architecture



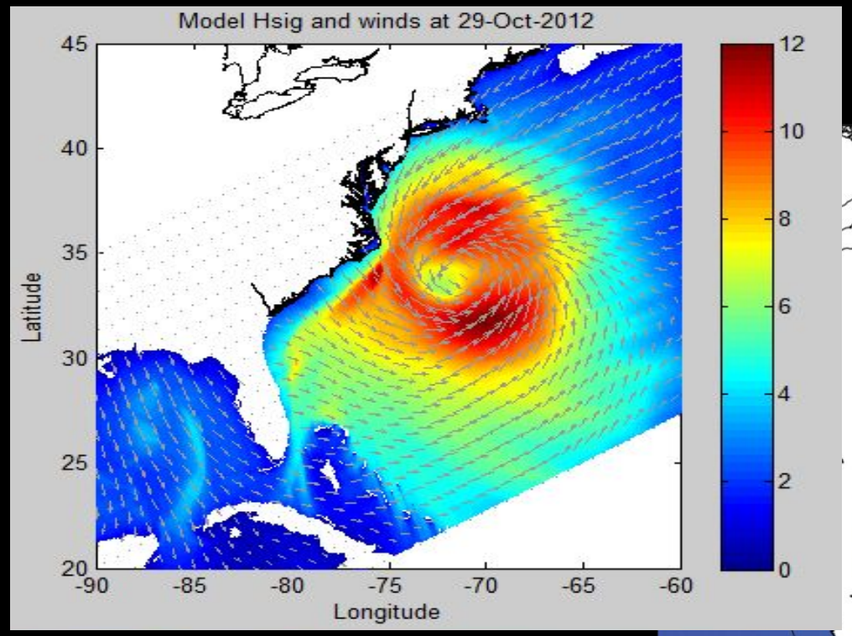
The Broad Python Scientific Ecosystem



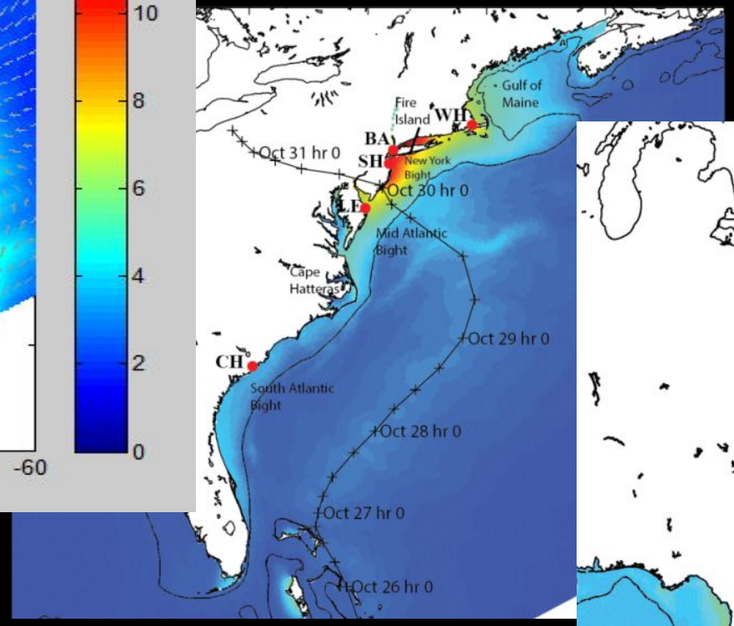
Credit: Stephan Hoyer, Jake Vanderplas (SciPy 2015)

USGS Coupled Ocean Atmosphere Wave and Sediment Transport modeling system

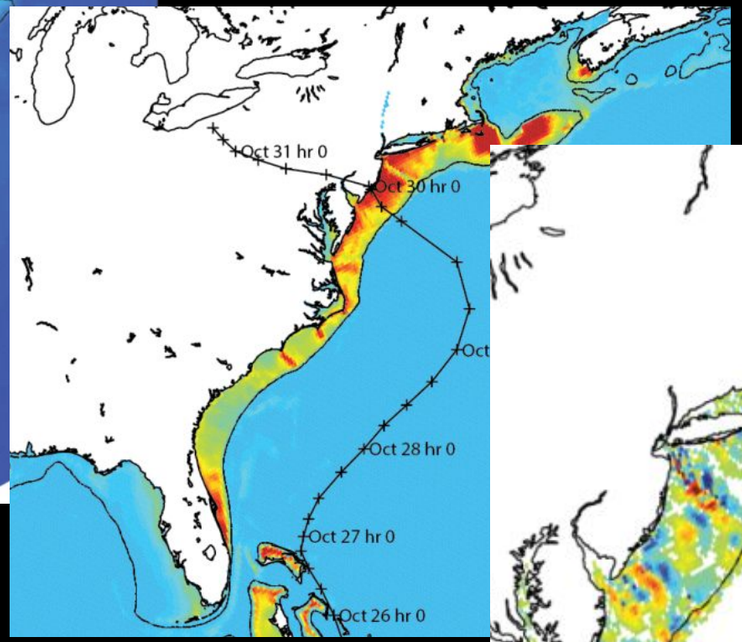
Wind and Waves



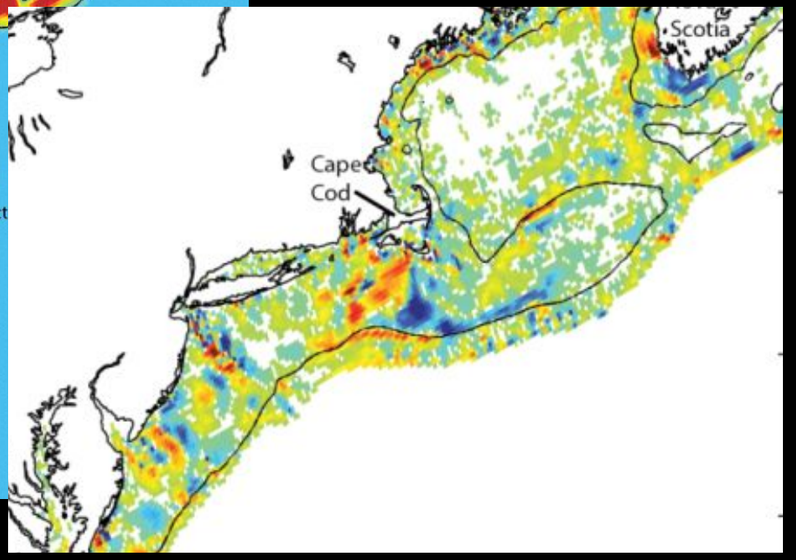
Water Levels & Currents



Sea Bed Stress



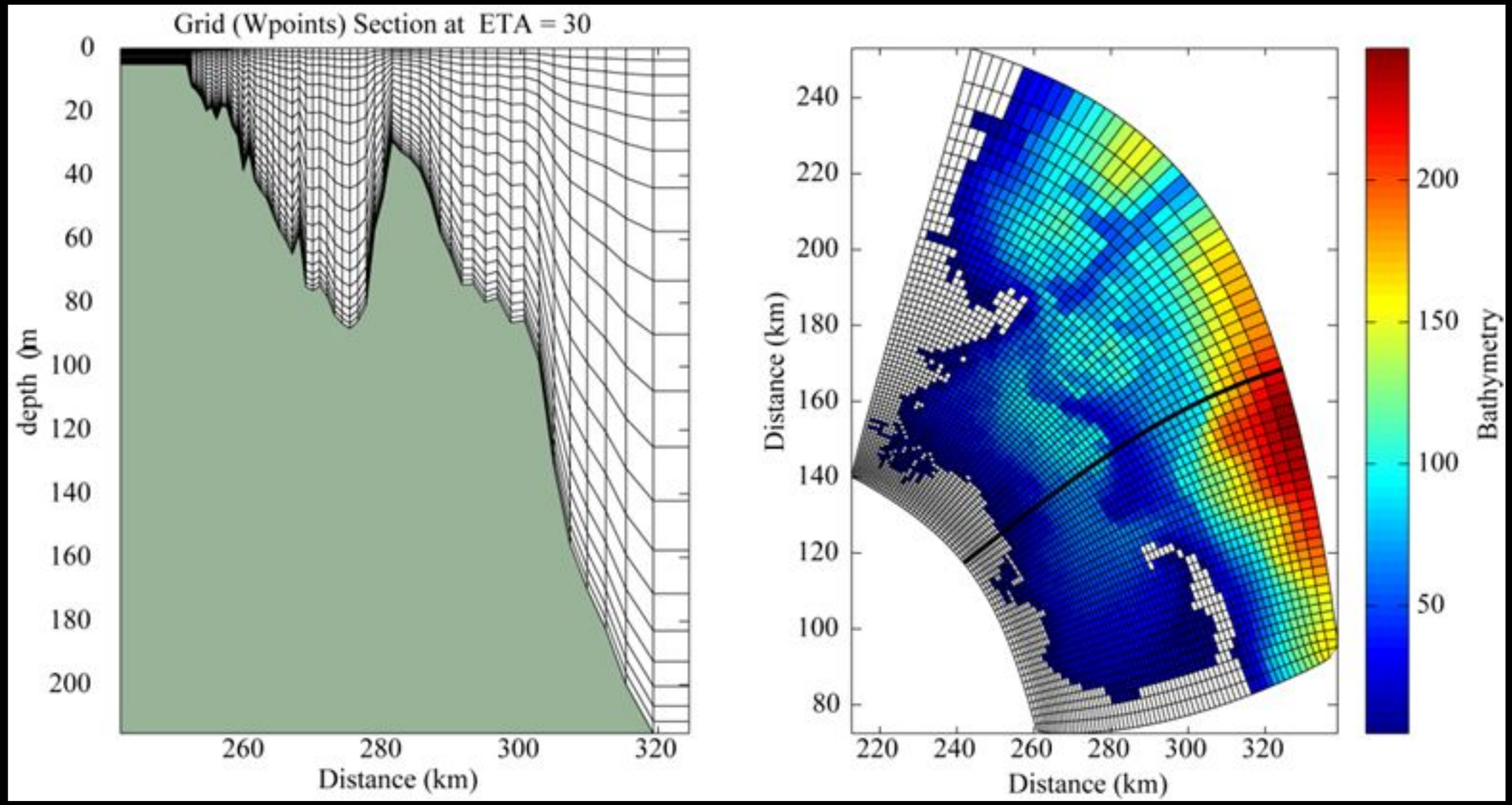
Sea Bed Erosion



~200TB of coastal ocean model output data in 4D (T, Z, Y, X) NetCDF files

600+ users, COAWST development led by John Warner, USGS

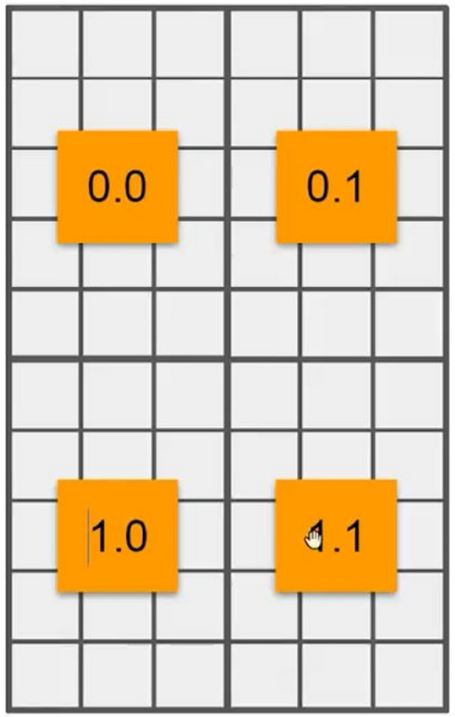
Data is stored in CF-Compliant NetCDF Files

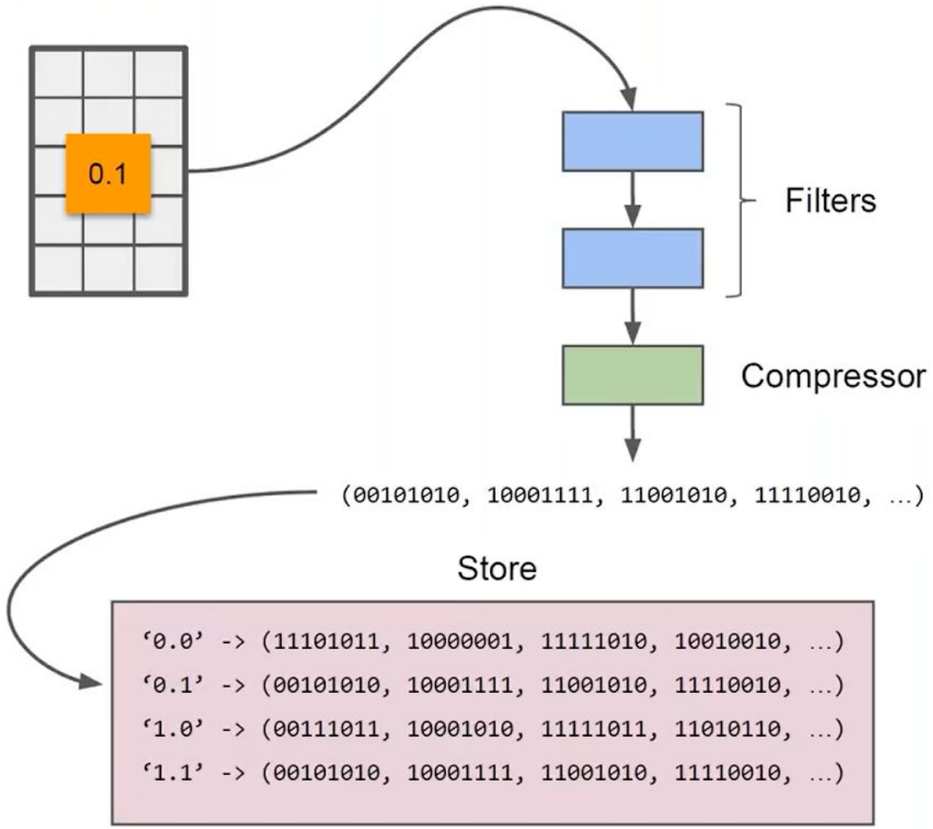


Zarr Format


```
{
    "metadata": {
        ".zattrs": {
            "Conventions": "UGRID-0.9.0",
        },
        "zeta/.zarray": {
            "chunks": [
                10,
                141973
            ],
            "compressor": {
                "blocksize": 0,
                "clevel": 5,
                "cname": "zlib",
                "id": "blosc",
                "shuffle": 0
            },
            "dtype": "<f8",
            "fill_value": "NaN",
            "filters": null,
            "order": "C",
            "shape": [
                720,
                9228245
            ],
            "zarr_format": 2
        }
    },
    "zarr_consolidated_format": 1
}
```


E.g., array with shape (10, 6) and chunk shape (5, 3) has 4 chunks in a 2 by 2 chunk grid, with chunks identified by the keys ‘0.0’, ‘0.1’, ‘1.0’, ‘1.1’.





Convert lots of COAWST NetCDFs to one big Zarr

In theory, this would be a simple as:

```
import xarray as xr
ds = xr.open_mfdataset('*.nc')
ds.to_zarr('all_nc.zarr', consolidated=True)
```

In practice, we usually want to rechunk and xarray has issues with certain NetCDF elements, and it's a bit more complicated....

In [1]:
```
import numpy as np
import xarray as xr
import numcodecs
from dask.distributed import Client, LocalCluster
import glob
```

Build a list of filenames for open_mfdataset

In [2]:
```
infiles = glob.glob('/usgs/vault0/coawst/coawst_4/Output/use/*.nc')
```

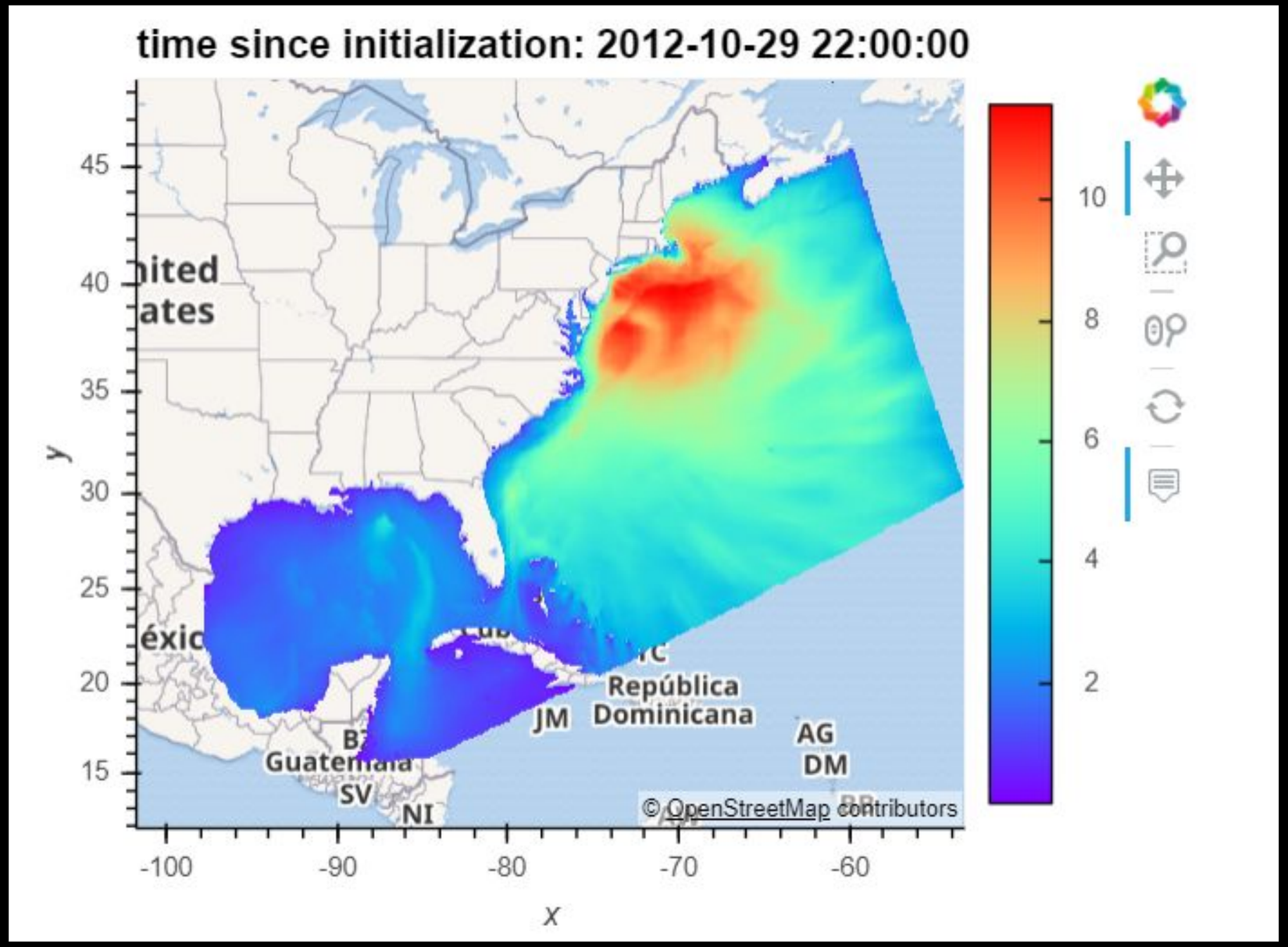
In [3]:
```
outfile = '/usgs/data2/rsignell/data/coawst/zarr_f'
```

In [4]:
```
concat_dim = 'ocean_time'
```

In [5]:
```
infiles.sort()
```

Make sure we sort the files before feeding to `open_mfdataset`

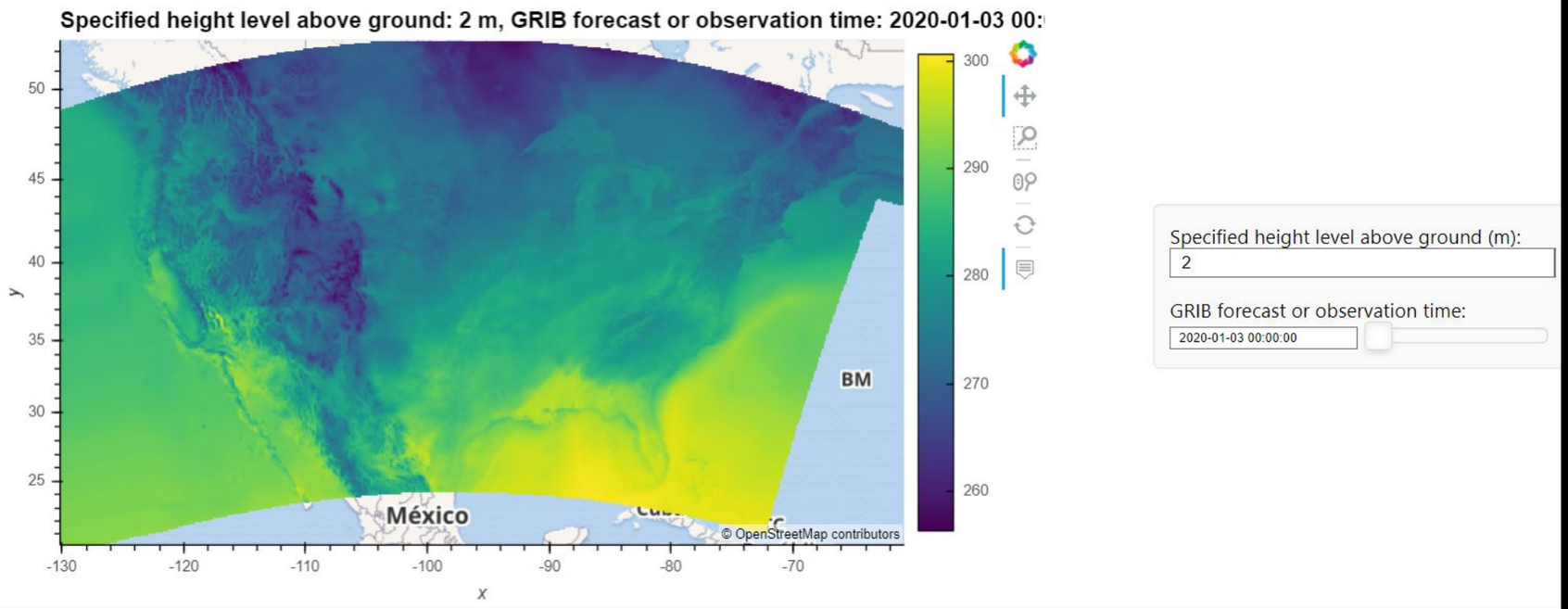
COAWST Forecast Analysis Archive Demo



[Static Notebook on Gist]
[Live Demo on Pangeo (requires login)]

Simple hvplot demo

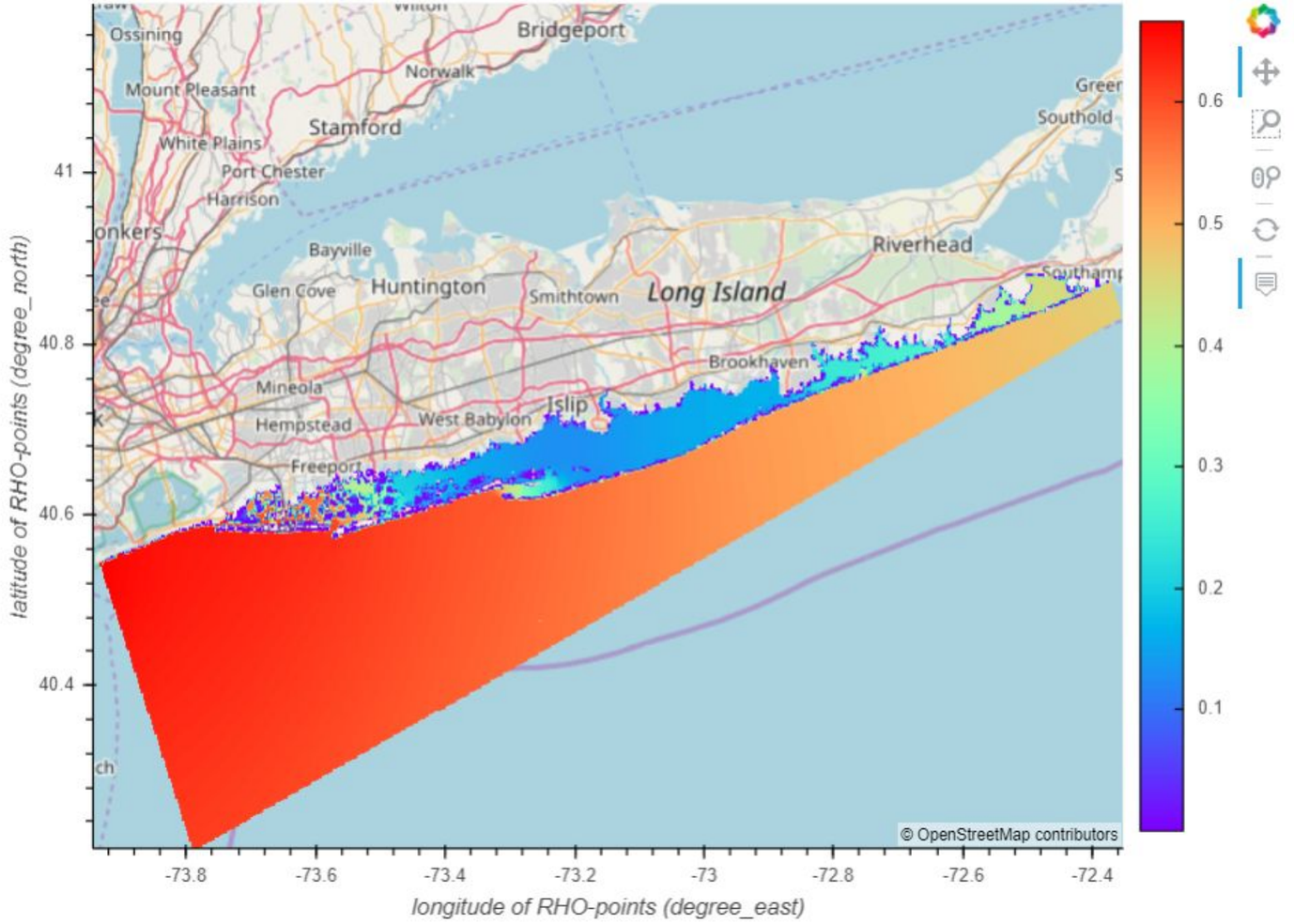
```
ds.Temperature_height_above_ground.hvplot(x='x', y='y', crs=crs, rasterize=True, frame_width=600,
                                          project=True, cmap='viridis', tiles=True)
```



[Static Notebook on Gist]
[Live Demo on Pangeo]

Embarrassingly parallel jobs with Pangeo

```
m2amp.hvplot.quadmesh(x='lon_rho', y='lat_rho', geo=True, rasterize=True, cmap='rainbow', frame_width=600) * gv.tile_sources.OSM
```



1441 time steps

324 x 1542 grid

= 493776 jobs

[Static Notebook on Gist]

Pangeo on AWS

- Kubernetes cluster deployed with Amazon Elastic Container Service for Kubernetes (Amazon EKS)
- Three classes of k8s node pools
  - Core pool: JupyterHub, web proxy (small)
  - Jupyter pool: autoscaling pool for single-user sessions
  - Dask pool: autoscaling pool for Dask workers on premptible (e.g., spot) instances
- Pangeo installed with Helm chart
- Custom environments built with repo2docker at https://github.com/pangeo-data/pangeo-stacks
- Deployed using https://github.com/pangeo-data/pangeo-cloud-federation
- Full deploy instructions at pangeo.io

Addressing Challenges

- Cloud costs: Will cost more in short run. Institutional support: Prior to EarthMap, HPC at USGS funded 100%, Cloud funded 0%. Now pilots can figure out what the real cloud costs are. Storage costs. Disruptors like Wasabi.com?
- Cloud skills: webinars, documentation, notebook examples, workshops, hackathons, road shows, communities like Pangeo and ESIP
- Cloud-optimized data: Zarr language support, NetCDF4/HDF5 read performance
- Pangeo deployment: Pangeo as a service? saturncloud.io, AWS SageMaker?

Scientific Software Environment of the Future

- Learn: pangeo.io
- Follow: medium.com/pangeo
- Discuss: discourse.pangeo.io
- Chat: gitter.im/pangeo-data

Join the ESIP Cloud Computing Cluster, Chaired by Patrick Quinn, Element84