

Zarr in Pangeo

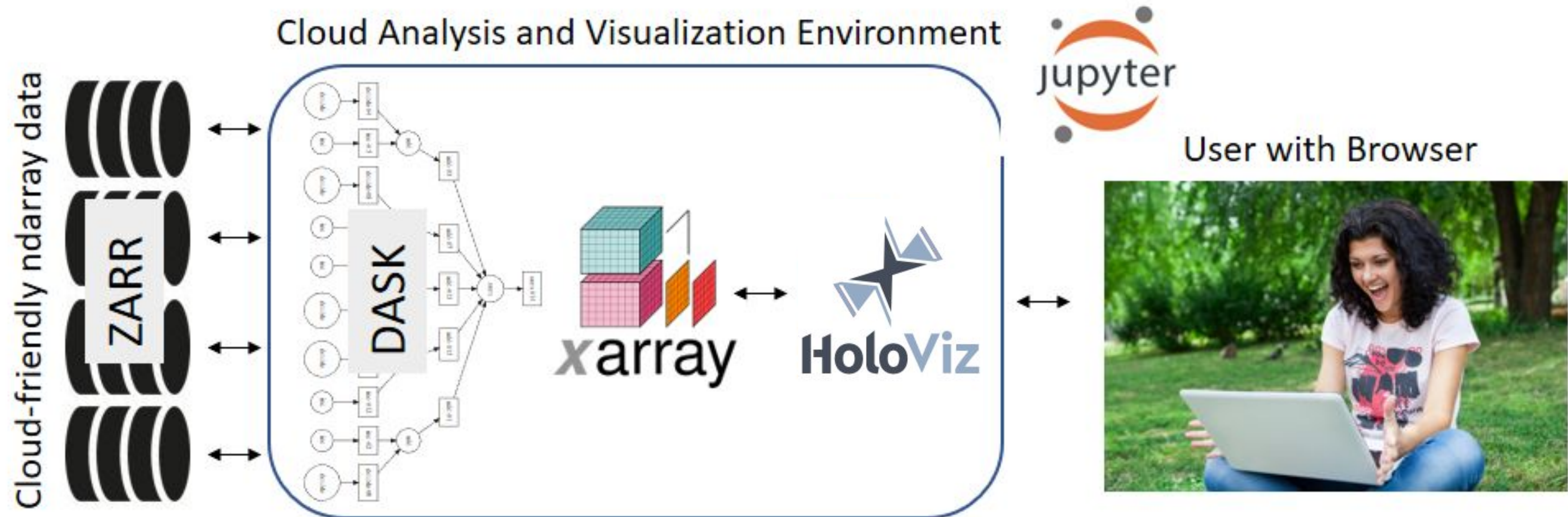
Rich Signell – USGS, for ...
Alistair Miles - University of Oxford
Ryan Abernathy - Columbia University
Joe Hamman - NCAR
Matthew Rocklin – Anaconda, Inc.
... and the rest of the Pangeo team

Email: rsignell@usgs.gov

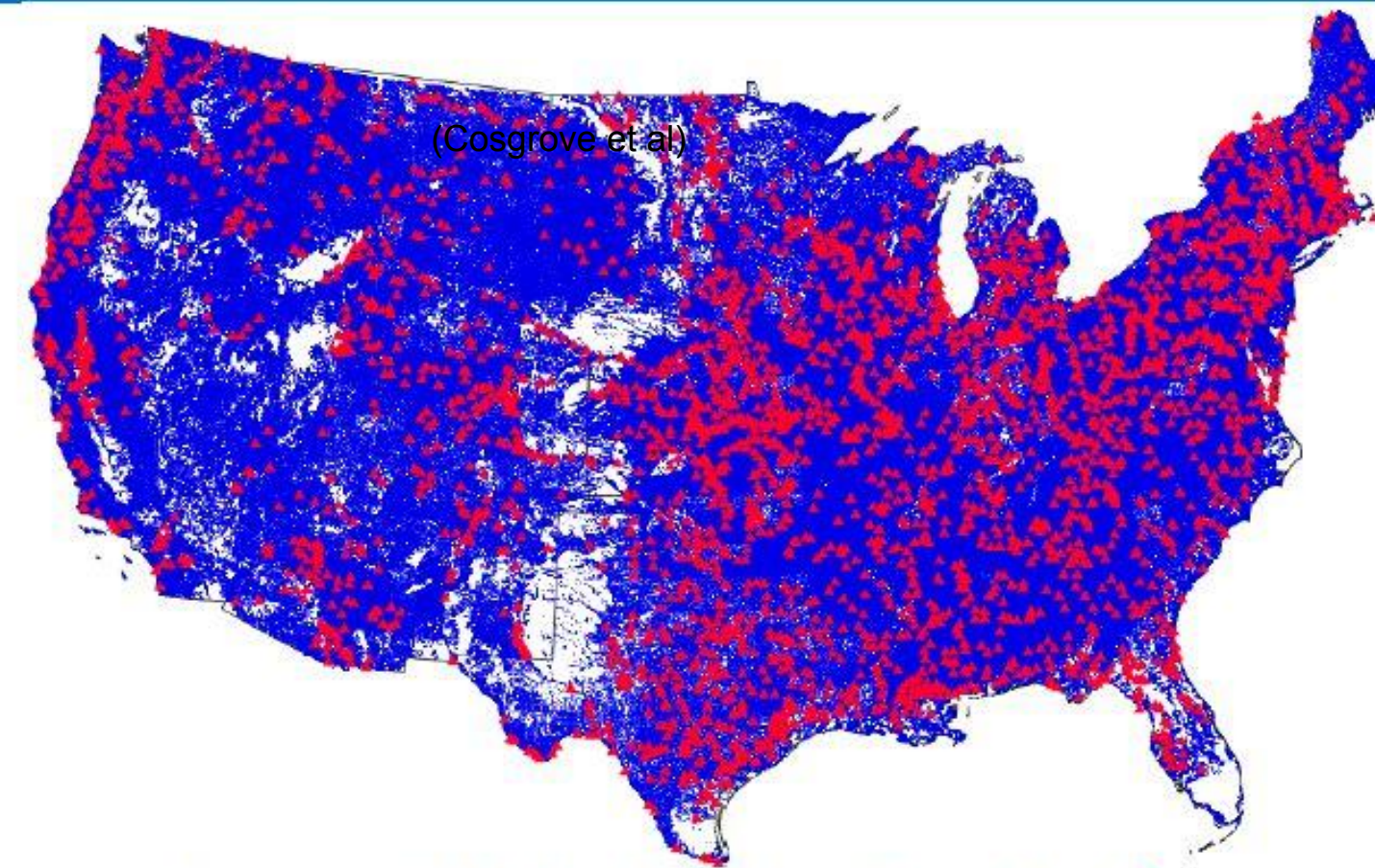
Github: [@rsignell-usgs](https://github.com/rsignell-usgs)

Twitter: [@rsignell](https://twitter.com/rsignell)

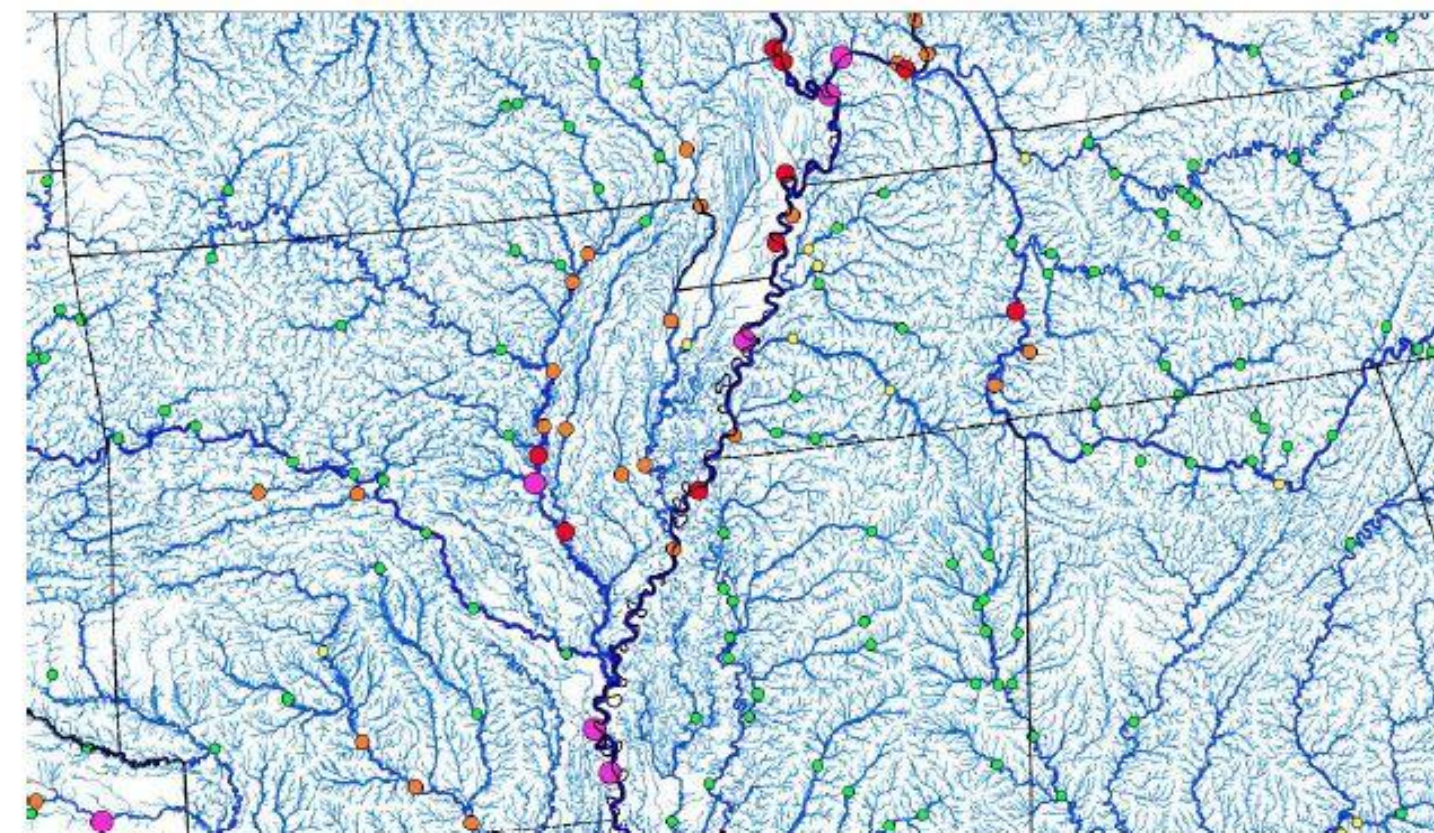
Pangeo Architecture



- **Hydrologic Output**
 - River channel discharge and velocity at 2.7 million river reaches
 - Reservoir inflow, outflow, elevation
 - Pondered 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 4

One month of forcing and output is 15TB



NWM is part of the Big Data Project, with data being pushed to the Cloud:



Forecast data:
s3:noaa-nwm-pds

25 year reanalysis:
s3:nwm-archive

The Problem...

 jupyter read_fuse_dask Last Checkpoint: 31 minutes ago (autosaved) 

File Edit View Insert Cell Kernel Widgets Help Trusted

```
In [12]: var = 'T2D'
```

```
In [13]: ds[var].nbytes/1.e9
```

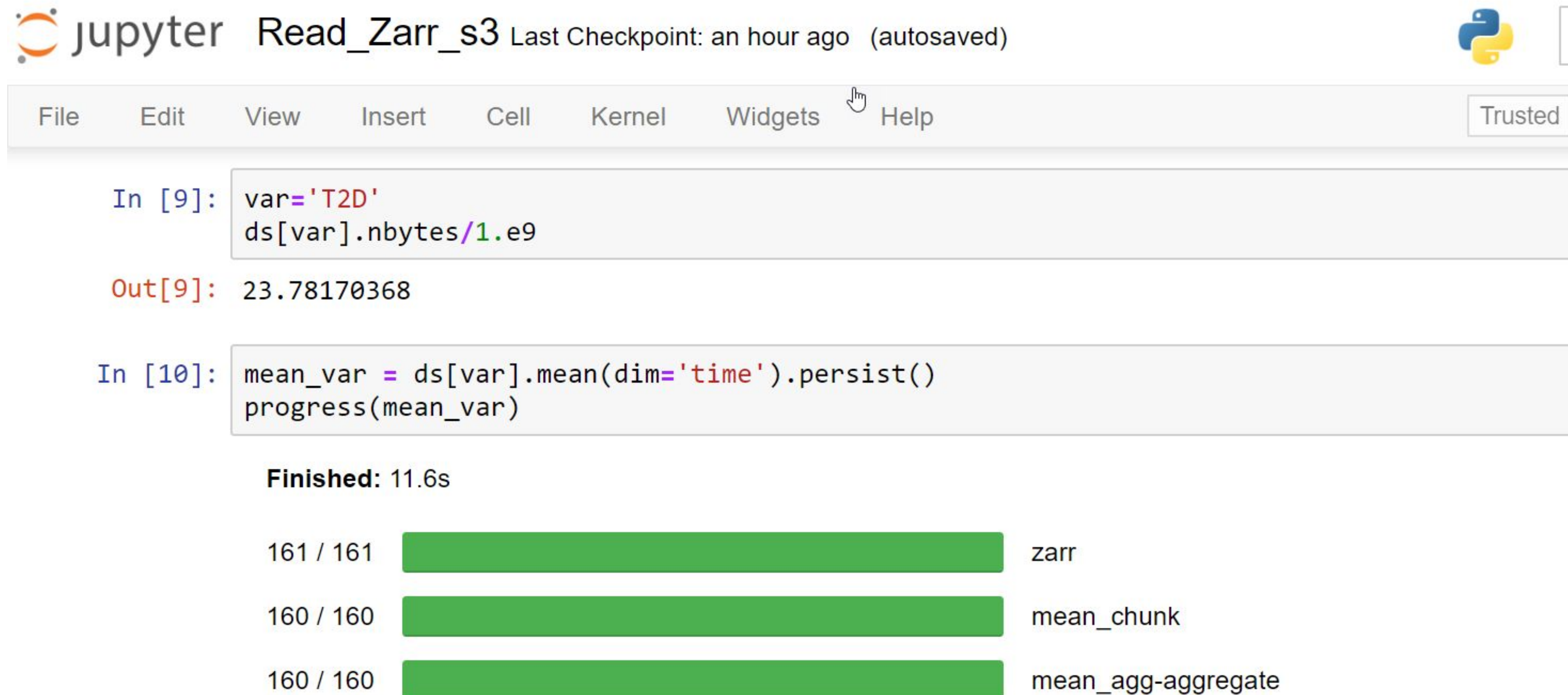
```
Out[13]: 23.92326144
```

```
In [14]: mean_var = ds[var].mean(dim='time').persist()
progress(mean_var)
```

Finished: 1min 52.5s

169 / 169	<div></div>	open_dataset-concatenate
169 / 169	<div></div>	open_dataset
169 / 169	<div></div>	mean_chunk

Zarr provides a Solution...



The image shows a JupyterLab interface with the title "Read_Zarr_s3" and a status "Last Checkpoint: an hour ago (autosaved)". The interface includes a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". A "Trusted" button is visible in the top right. The code area shows two input cells. The first cell contains the code `var='T2D'` and `ds[var].nbytes/1.e9`, with the output `23.78170368`. The second cell contains the code `mean_var = ds[var].mean(dim='time').persist()` and `progress(mean_var)`. Below the code area, a progress bar shows the completion of the task. The progress bar is divided into three sections: "zarr" (161 / 161), "mean_chunk" (160 / 160), and "mean_agg-aggregate" (160 / 160). The total time taken is 11.6s.

jupyter Read_Zarr_s3 Last Checkpoint: an hour ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted

In [9]: `var='T2D'`
`ds[var].nbytes/1.e9`

Out[9]: 23.78170368

In [10]: `mean_var = ds[var].mean(dim='time').persist()`
`progress(mean_var)`

Finished: 11.6s

161 / 161	<div></div>	zarr
160 / 160	<div></div>	mean_chunk
160 / 160	<div></div>	mean_agg-aggregate

For this use case, Zarr is 10x faster

HDF in the Cloud challenges and solutions for scientific data

Multi-dimensional data, such as is commonly stored in HDF and NetCDF formats, is difficult to access on traditional cloud storage platforms. This post outlines the situation, the following possible solutions, and their strengths and weaknesses.

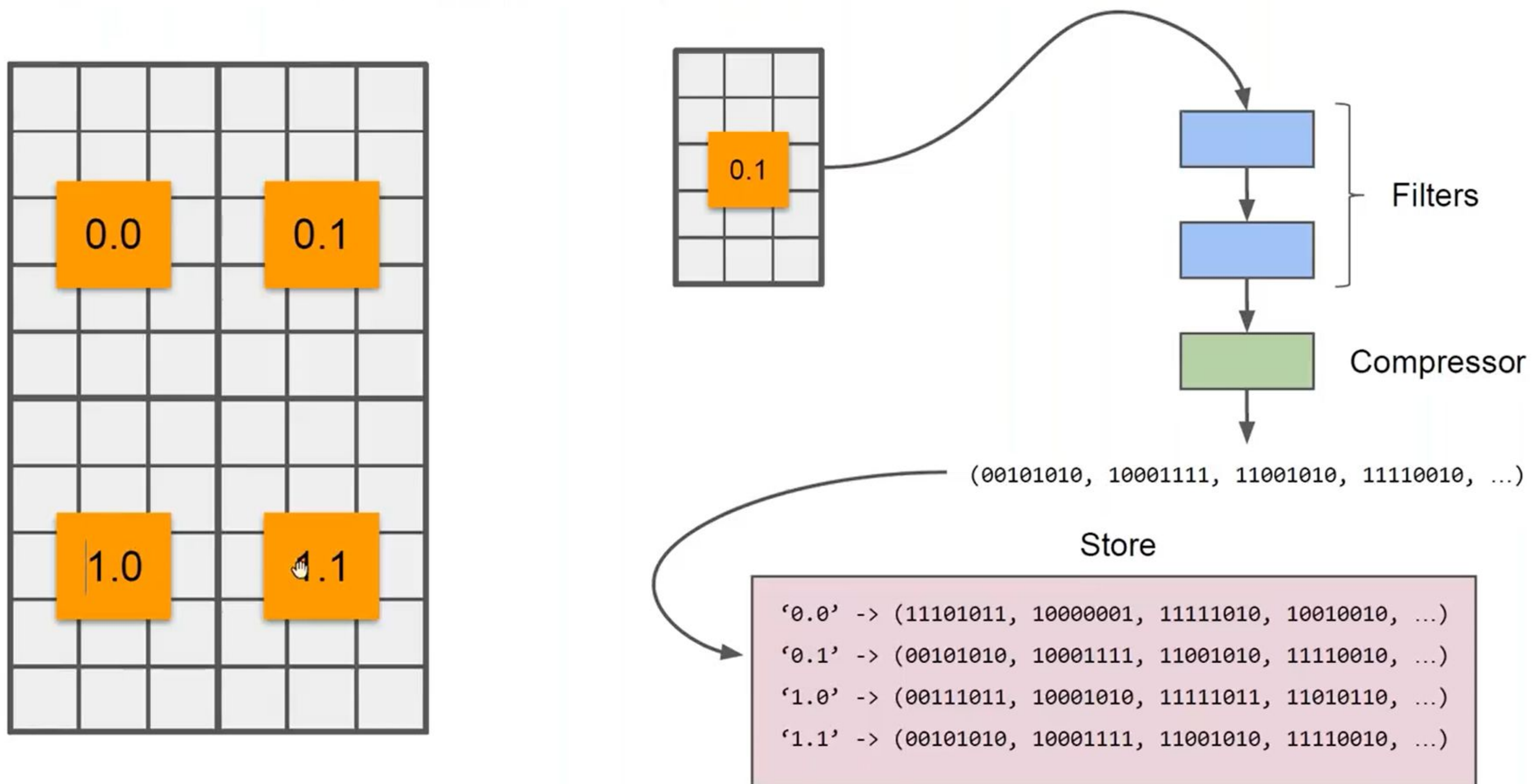
1. **Cloud Optimized GeoTIFF:** We can use modern and efficient formats from other domains, like Cloud Optimized GeoTIFF
2. **HDF + FUSE:** Continue using HDF, but mount cloud object stores as a file system with [FUSE](#)
3. **HDF + Custom Reader:** Continue using HDF, but teach it how to read from S3, GCS, ADL, ...
4. **Build a Distributed Service:** Allow others to serve this data behind a web API, built however they think best
5. **New Formats for Scientific Data:** Design a new format, optimized for scientific data in the cloud

Zarr Format

- Developed to address problems with NetCDF/HDF on cloud storage
- Simple format, clear specification
- Each chunk is stored as a separate binary object
- Lightweight global and variable metadata stored as JSON
- Groups, filters, compression using Blosc
- Free, open-source software
- Read/write using Xarray

Zarr Format

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'.

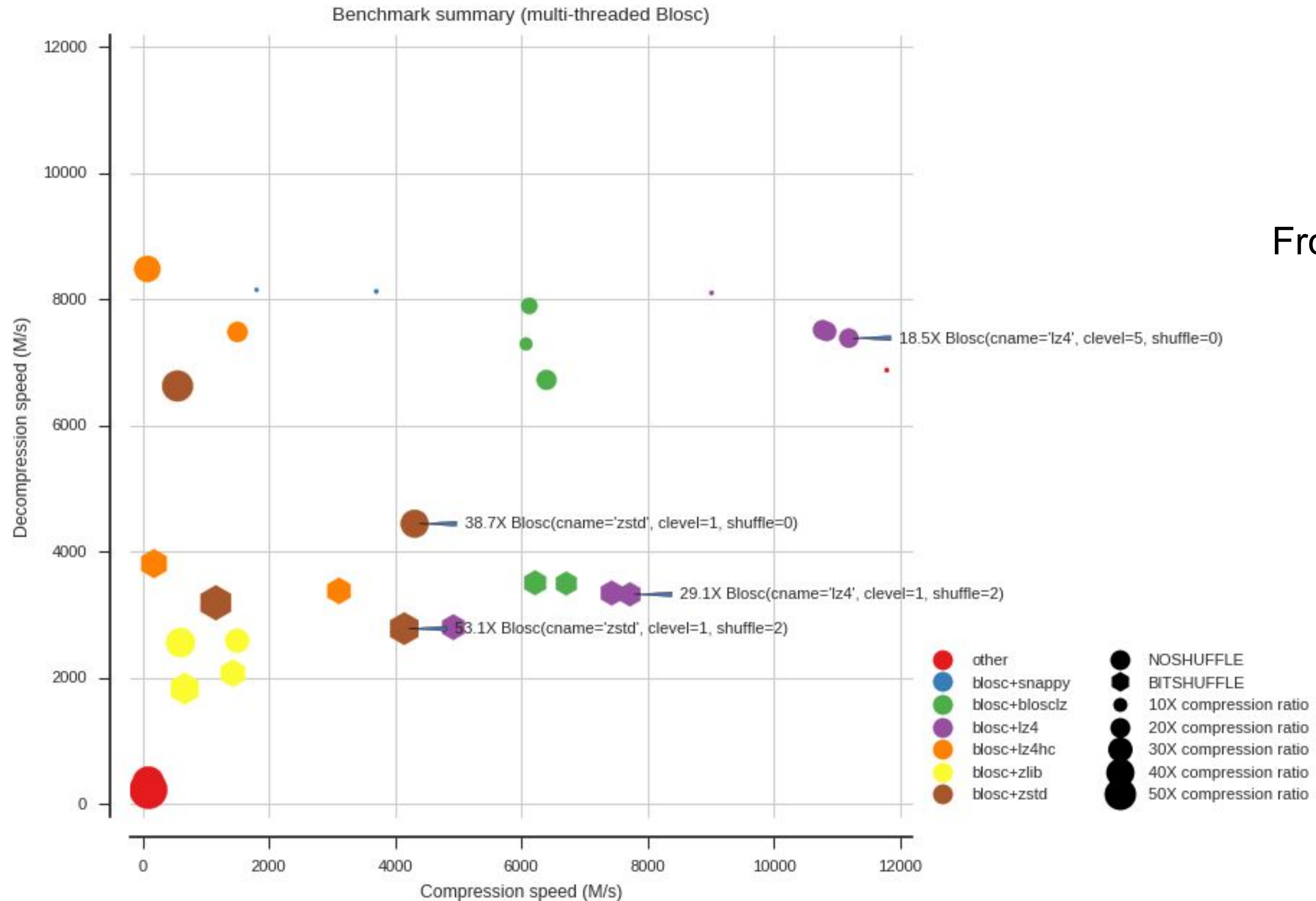


Zarr Format

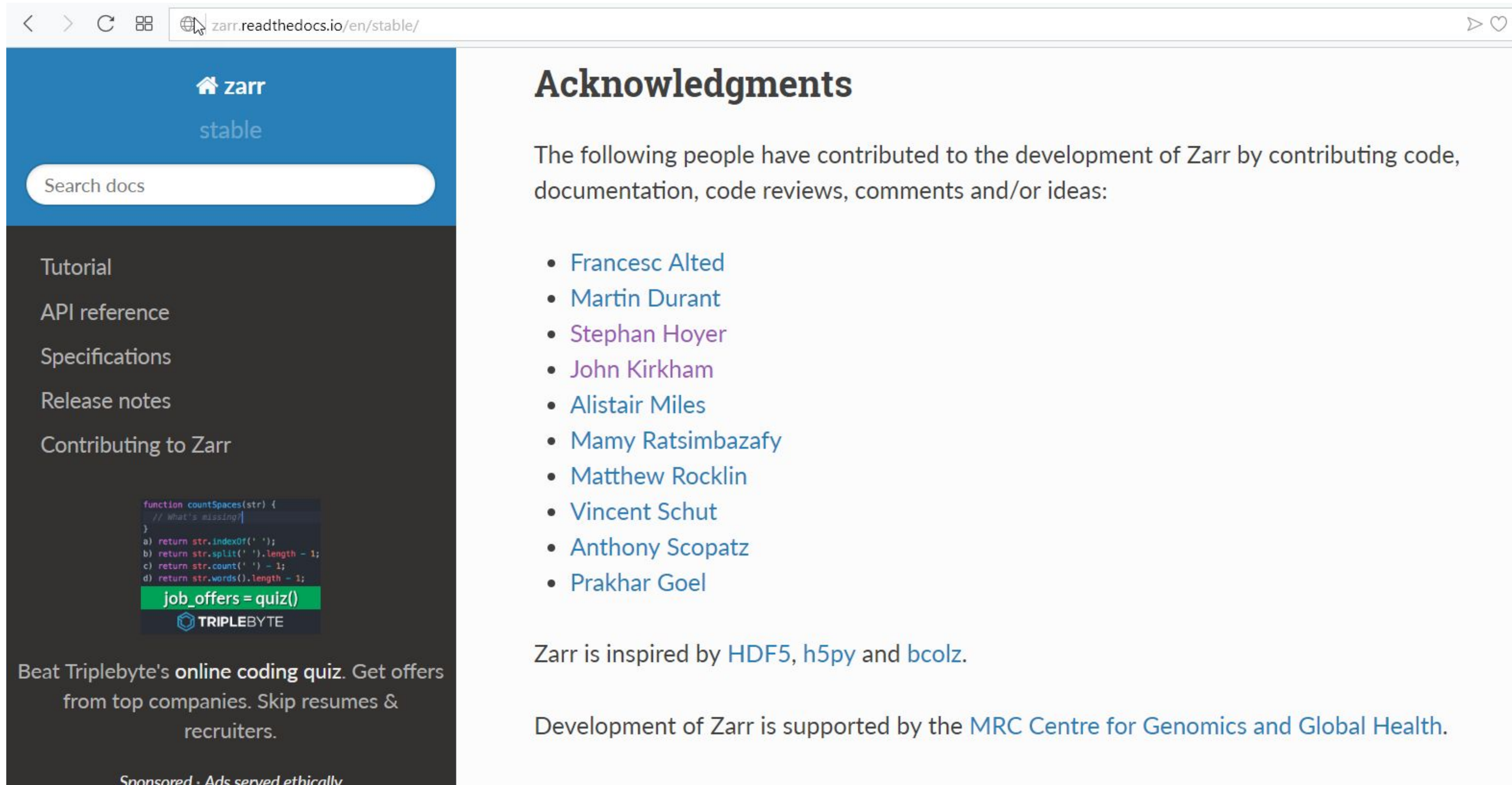
```
more ./adcirc/zeta/.zarray
{
  "chunks": [
    10,
    141973
  ],
  "compressor": {
    "blocksize": 0,
    "clevel": 5,
    "cname": "lz4",
    "id": "blosc",
    "shuffle": 1
  },
  "dtype": "<f8",
  "fill_value": -99999.0,
  "filters": null,
  "order": "C",
  "shape": [
    720,
    9228245
  ],
  "zarr_format": 2
}
```

```
more ./adcirc/zeta/.zattrs
{
  "_ARRAY_DIMENSIONS": [
    "time",
    "node"
  ],
  "coordinates": "y x",
  "location": "node",
  "long_name": "water surface elevation above geoid",
  "mesh": "adcirc_mesh",
  "standard_name": "sea_surface_height_above_geoid",
  "units": "m"
}
```


Compression and Performance



Zarr is community-driven



The screenshot shows the Zarr documentation website. The left sidebar contains a navigation menu with links to Tutorial, API reference, Specifications, Release notes, and Contributing to Zarr. Below the menu is a code snippet for a function named `countSpaces` and a Triplebyte advertisement. The main content area is titled "Acknowledgments" and lists contributors. Below the list, it mentions that Zarr is inspired by HDF5, h5py, and bcolz, and that its development is supported by the MRC Centre for Genomics and Global Health.

zarr
stable

Search docs

Tutorial
API reference
Specifications
Release notes
Contributing to Zarr

```
function countSpaces(str) {  
  // What's missing?  
}  
a) return str.indexOf(' ');  
b) return str.split(' ').length - 1;  
c) return str.count(' ') - 1;  
d) return str.words().length - 1;  
job_offers = quiz()  
TRIPLEBYTE
```

Beat Triplebyte's online coding quiz. Get offers from top companies. Skip resumes & recruiters.

Sponsored · Ads served ethically

Acknowledgments

The following people have contributed to the development of Zarr by contributing code, documentation, code reviews, comments and/or ideas:

- [Francesc Alted](#)
- [Martin Durant](#)
- [Stephan Hoyer](#)
- [John Kirkham](#)
- [Alistair Miles](#)
- [Mamy Ratsimbazafy](#)
- [Matthew Rocklin](#)
- [Vincent Schut](#)
- [Anthony Scopatz](#)
- [Prakhar Goel](#)

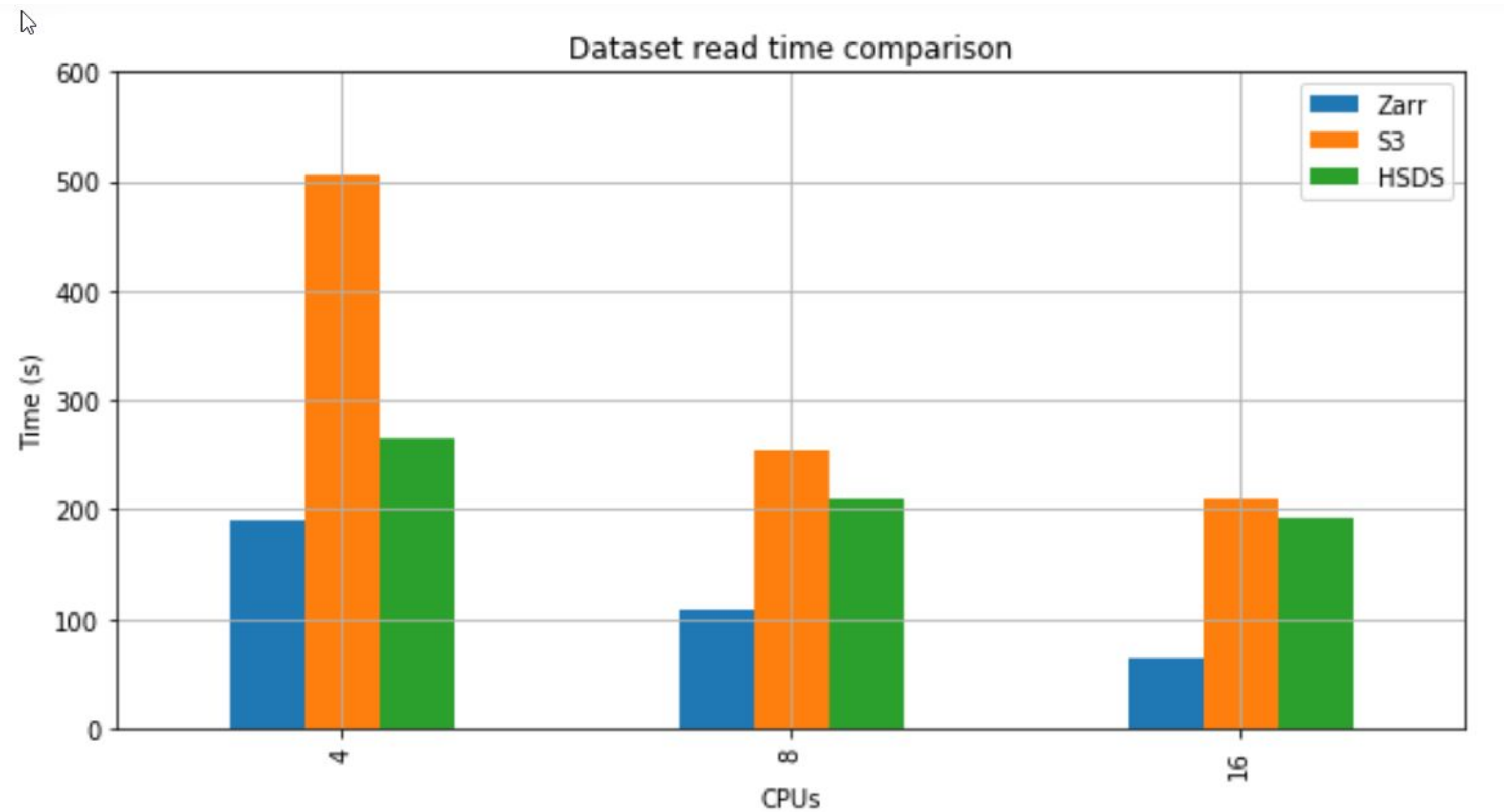
Zarr is inspired by [HDF5](#), [h5py](#) and [bcolz](#).

Development of Zarr is supported by the [MRC Centre for Genomics and Global Health](#).

Zarr Progress

In the last year:

- Representation of NetCDF standardized as a convention
- Unidata formally announced they are developing support for Zarr in the NetCDF C library
- Consolidation of metadata option
- Continued growth of use within the Pangeo community



read 58GB dataset written with chunks, filters, compression



PANGEO

A community platform for Big Data
geoscience

OUR GOALS

1. Foster collaboration around the open source scientific python ecosystem for ocean / atmosphere / land / climate science.
2. Support the development with domain-specific geoscience packages.
3. Improve scalability of these tools to to handle petabyte-scale datasets on HPC and cloud platforms.

<http://pangeo-data.org>

**Zarr - scalable storage of tensor
data for parallel and distributed
computing**



Alistair Miles (@alimanfoo) - SciPy 2019

These slides: <https://zarr-developers.github.io/slides/scipy-2019.html>

<https://www.youtube.com/watch?v=qyJXBlrdzBs>