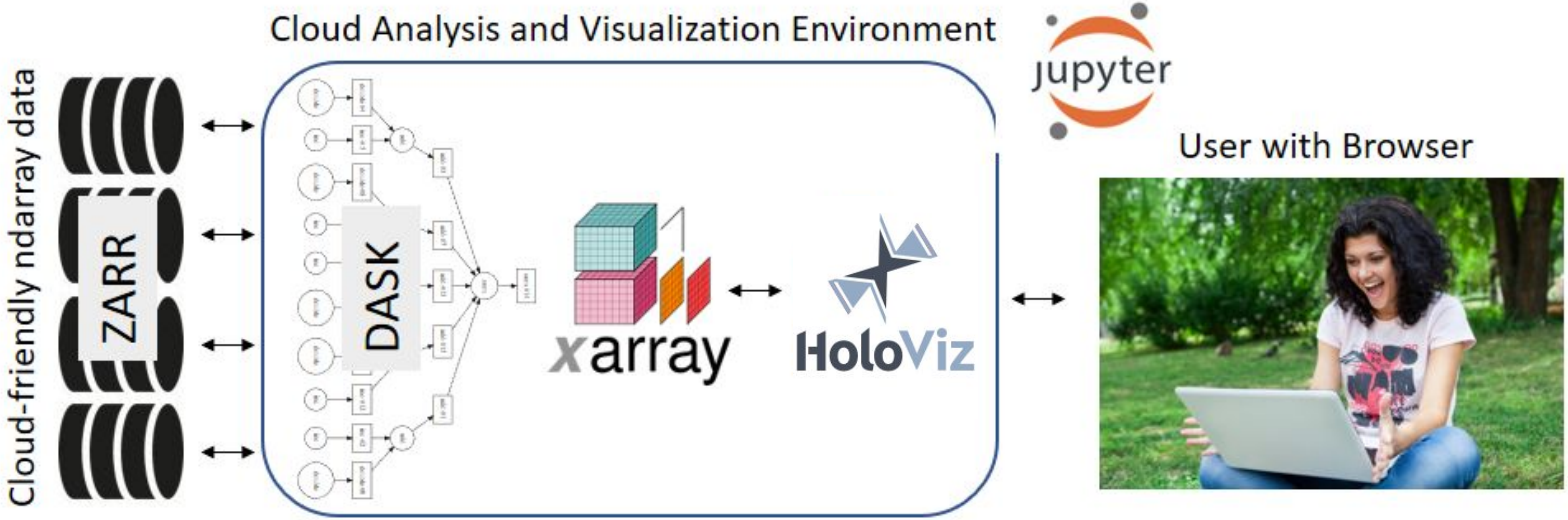
Zarr in Pangeo

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
Twitter: @rsignell

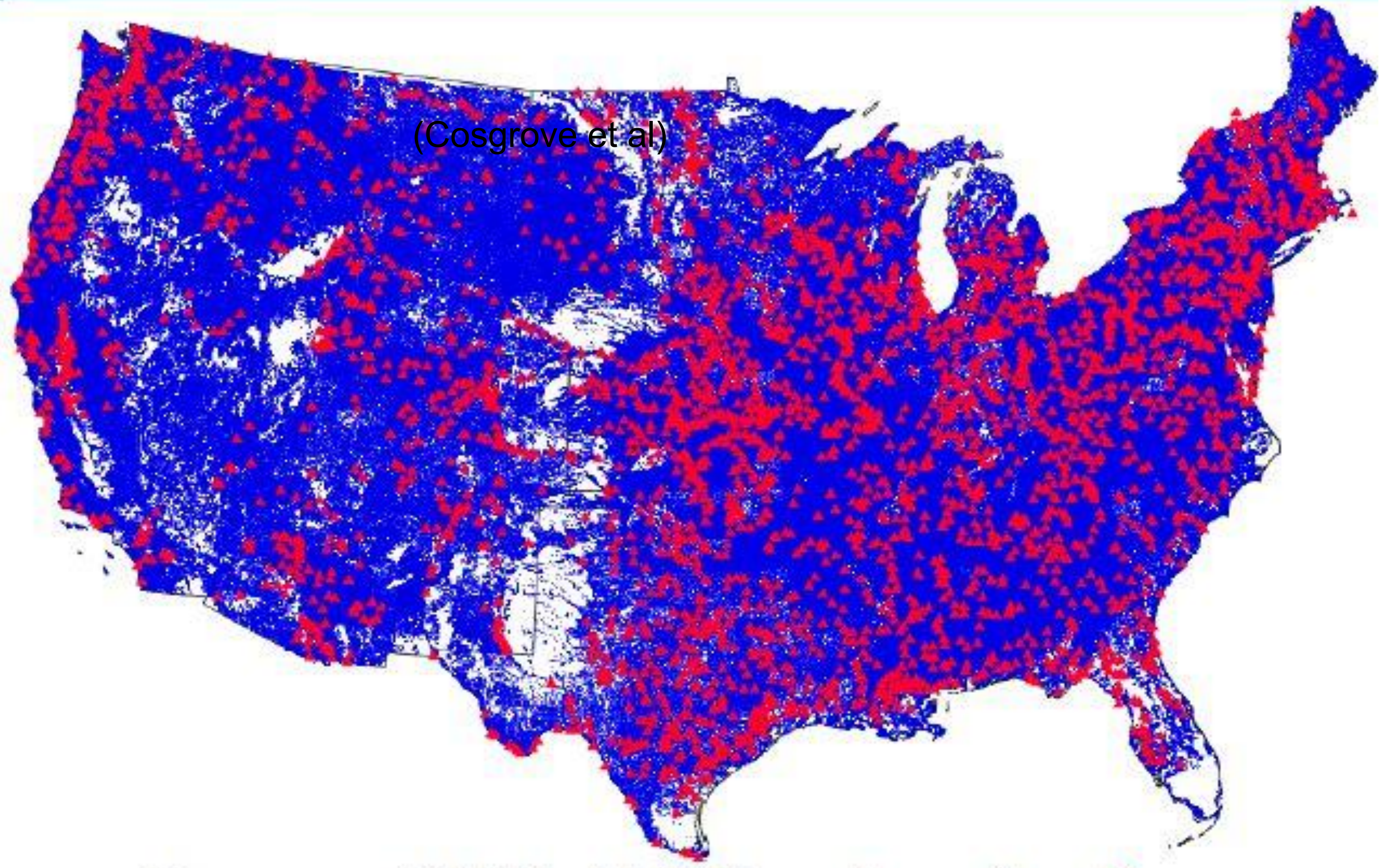
Pangeo Architecture


Cloud Analysis and Visualization Environment
jupyter
User with Browser
Cloud-friendly ndarray data
ZARR
DASK
xarray
HoloViz

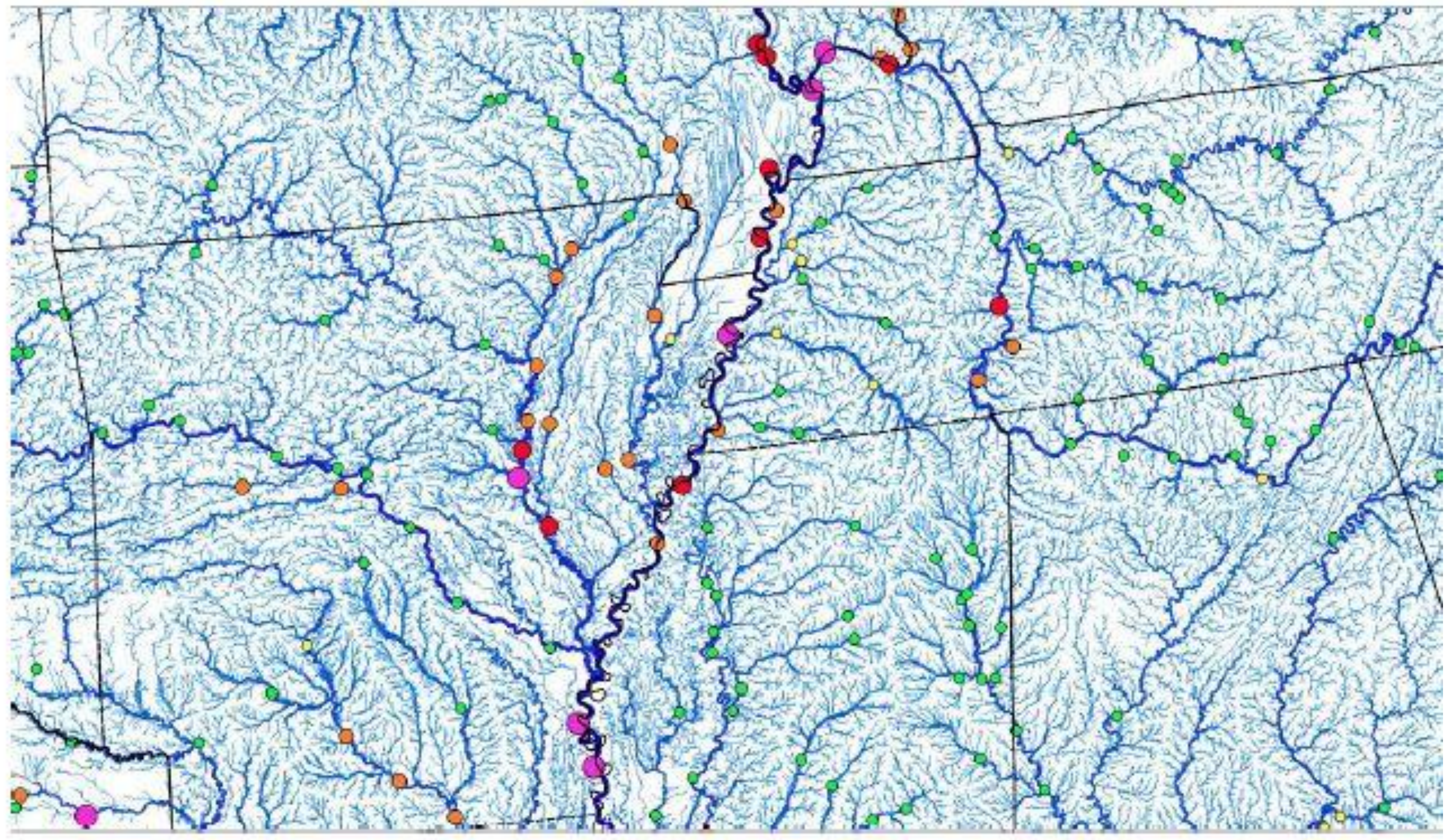
NWM V1.0 Output

- **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)

NWM 1km LSM Domain


(Cosgrove et al)

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



Current NWS River Forecast Points (circles)
Overlaid with NWM Stream Reaches 4

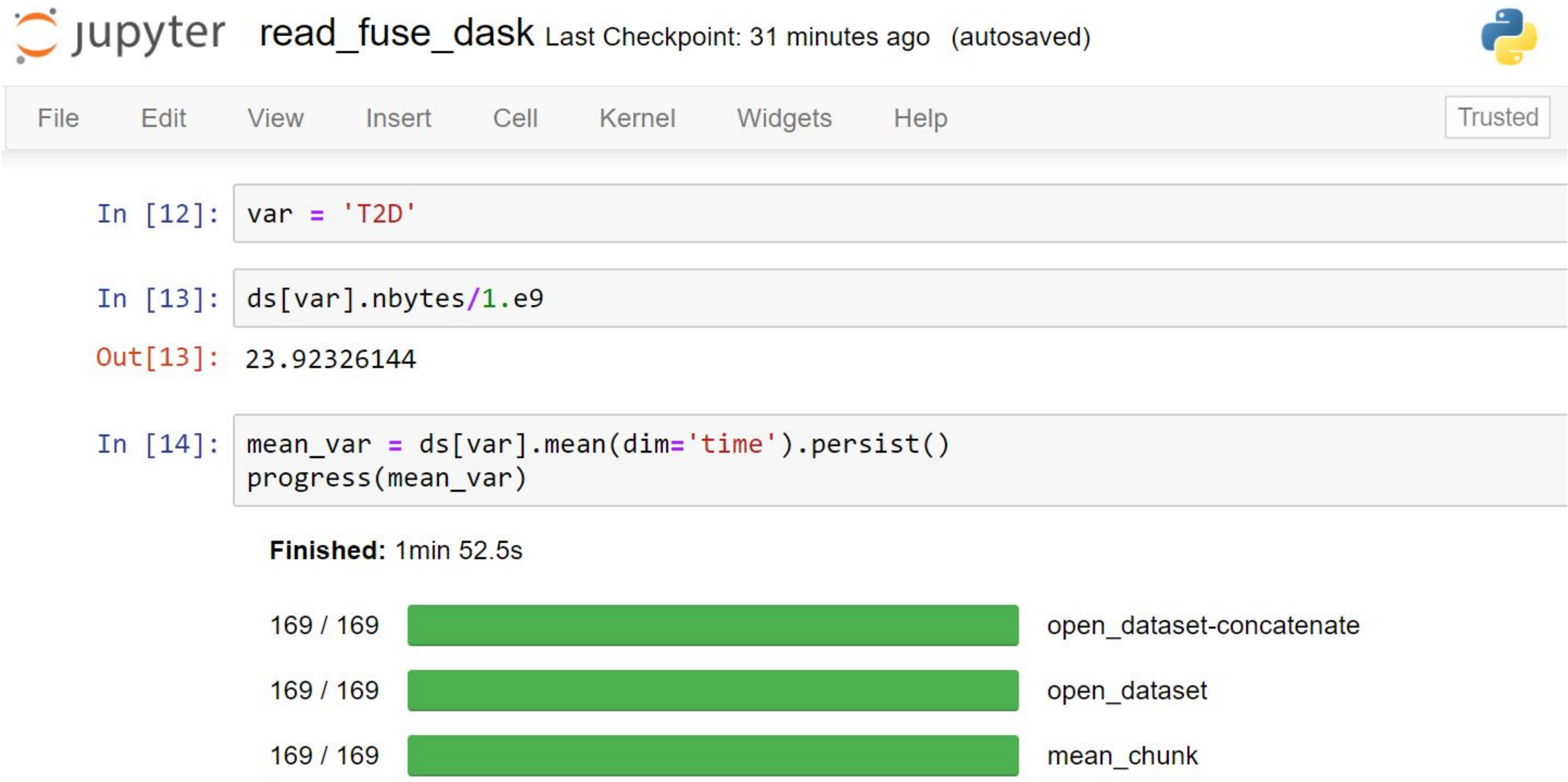
One month of forcing and output is 15TB

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…

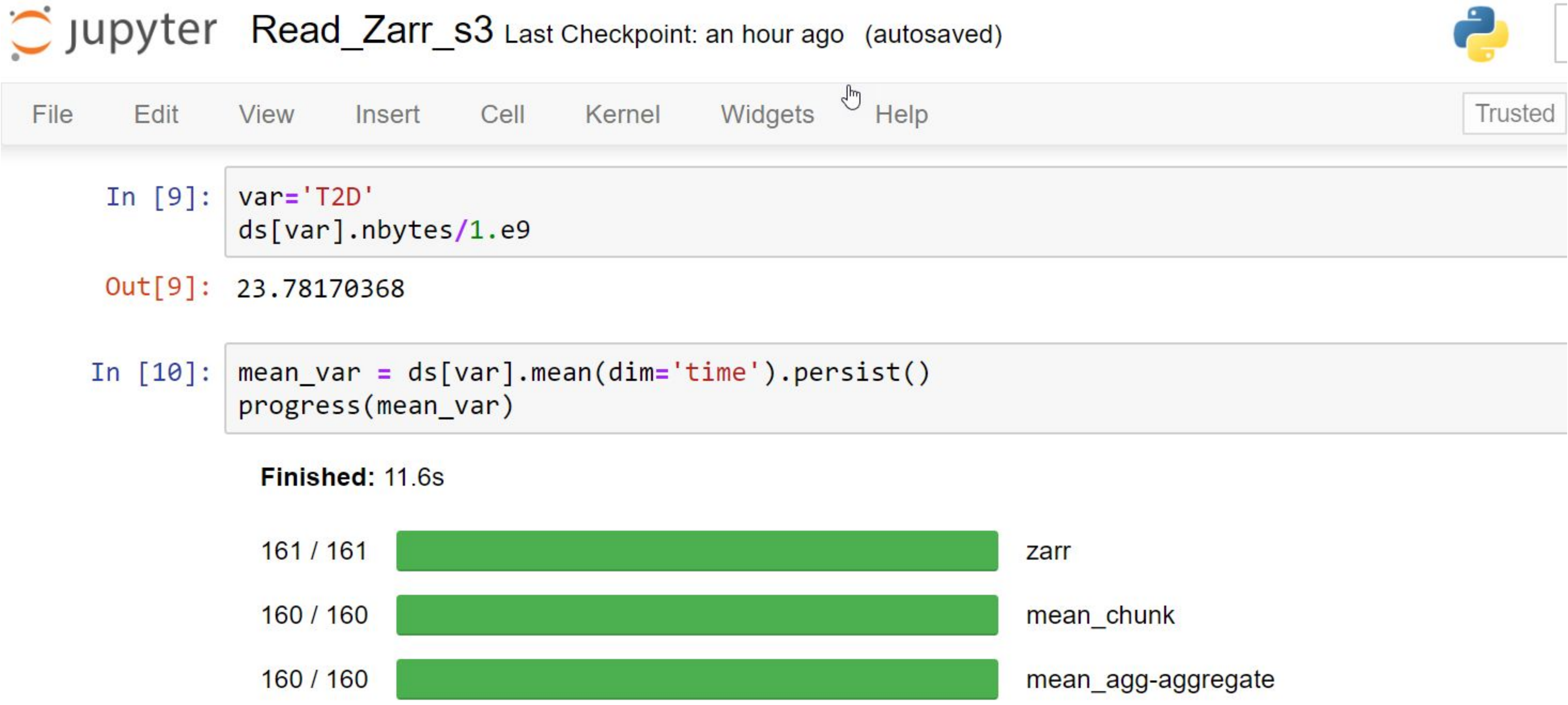


```
In [12]: var = 'T2D'

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  open_dataset-concatenate
169 / 169  open_dataset
169 / 169  mean_chunk
```

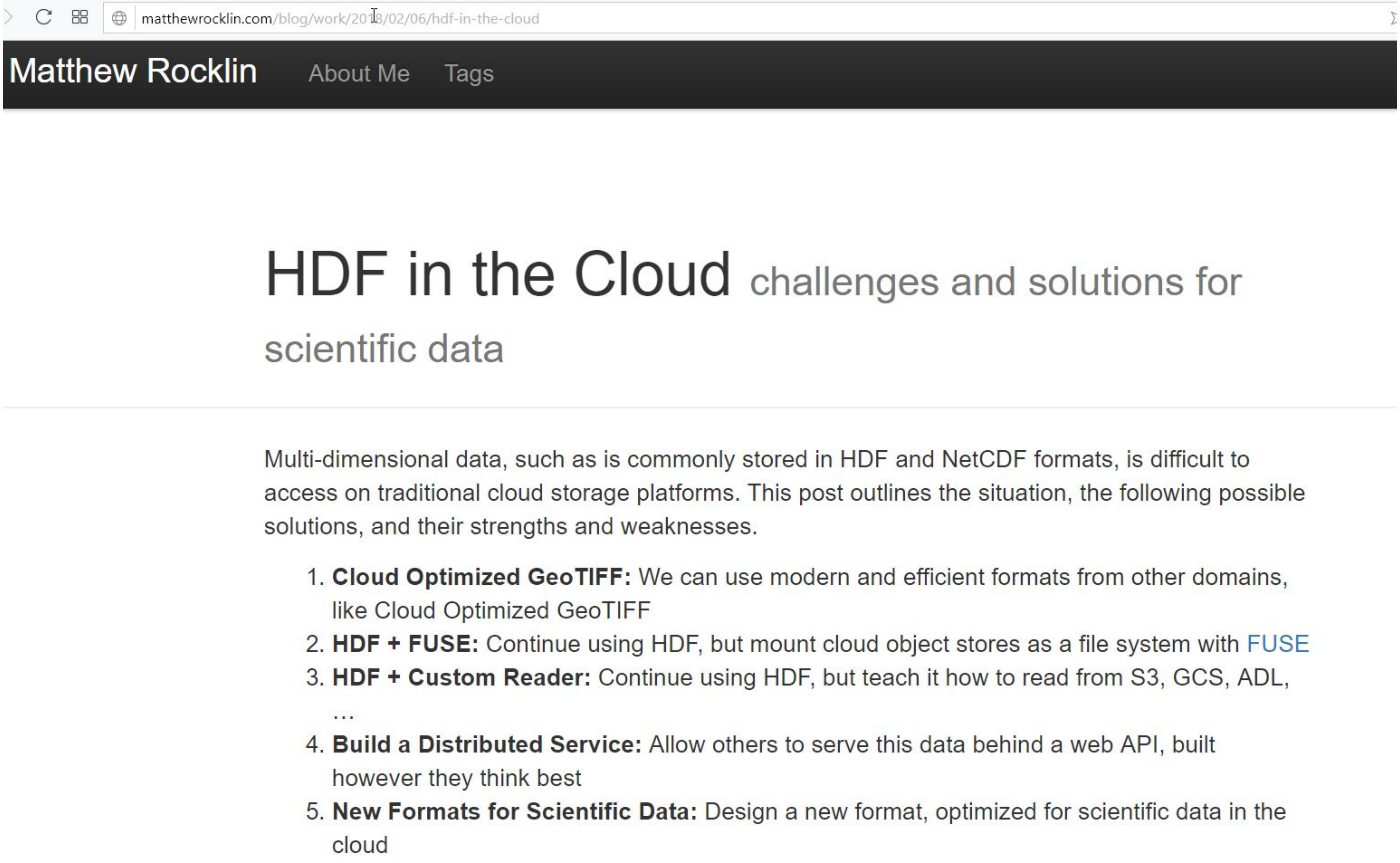
Zarr provides a Solution…



```
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  ████████████████  zarr
160 / 160  ████████████████  mean_chunk
160 / 160  ████████████████  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.

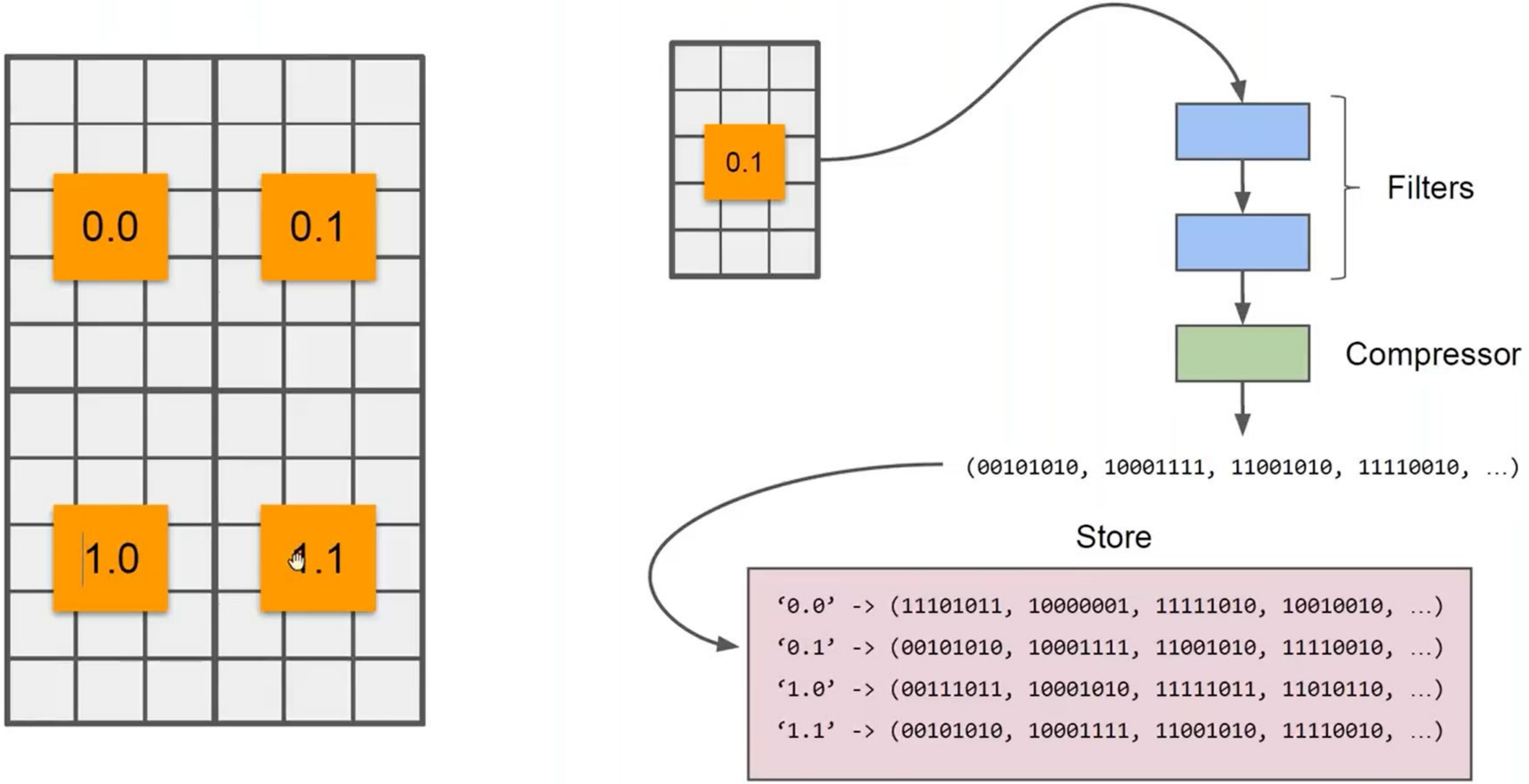
1. **Cloud Optimized GeoTIFF:** We can use modern and efficient formats from other domains, like Cloud Optimized GeoTIFF
2. **HDF + FUSE:** Continue using HDF, but mount cloud object stores as a file system with FUSE
3. **HDF + Custom Reader:** Continue using HDF, but teach it how to read from S3, GCS, 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

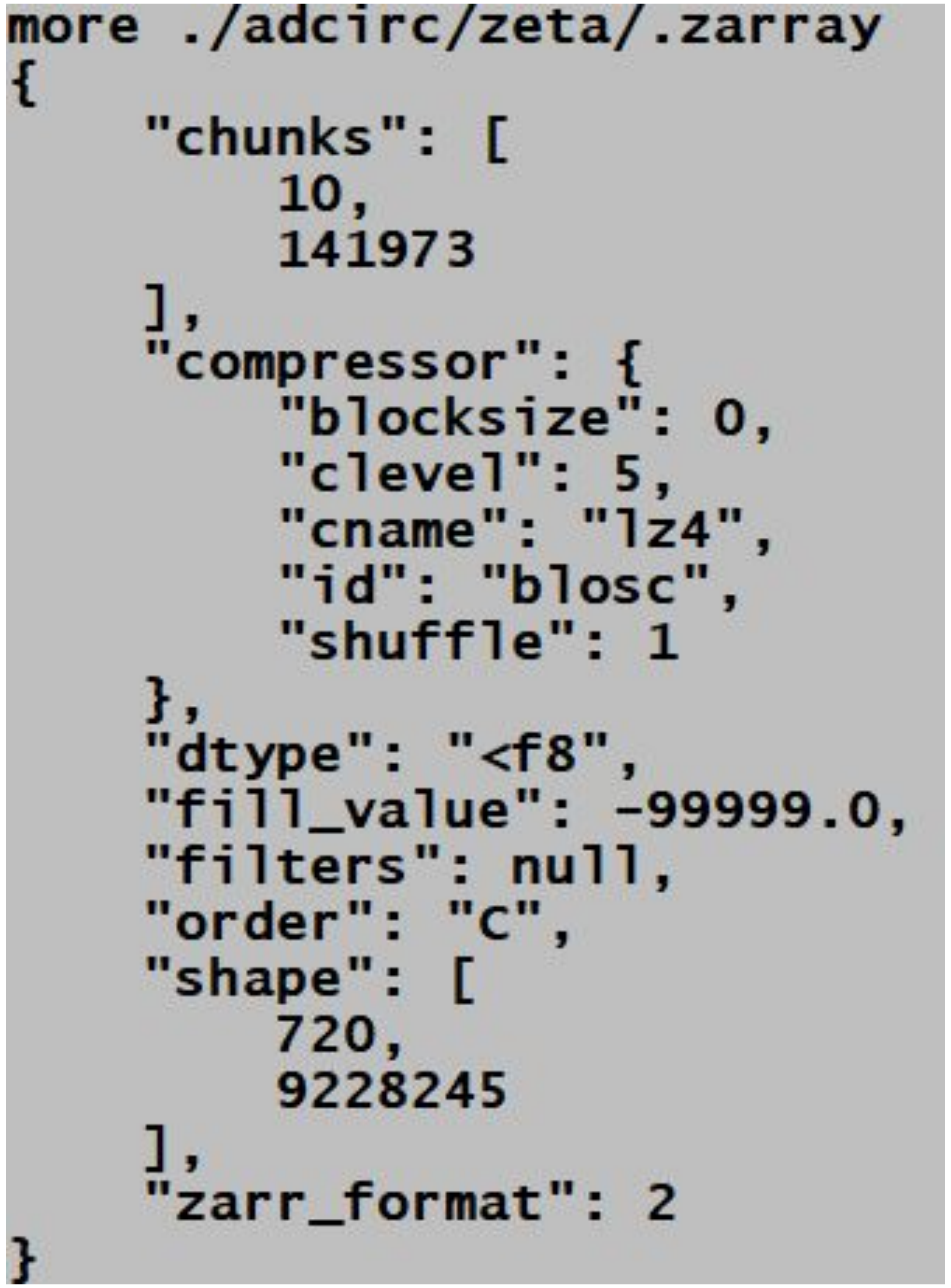
E.g., array with shape (10, 6) and chunk shape (5, 3) has 4 chunks in a 2 by 2 chunk grid, with chunks identified by the keys ‘0.0’, ‘0.1’, ‘1.0’, ‘1.1’.



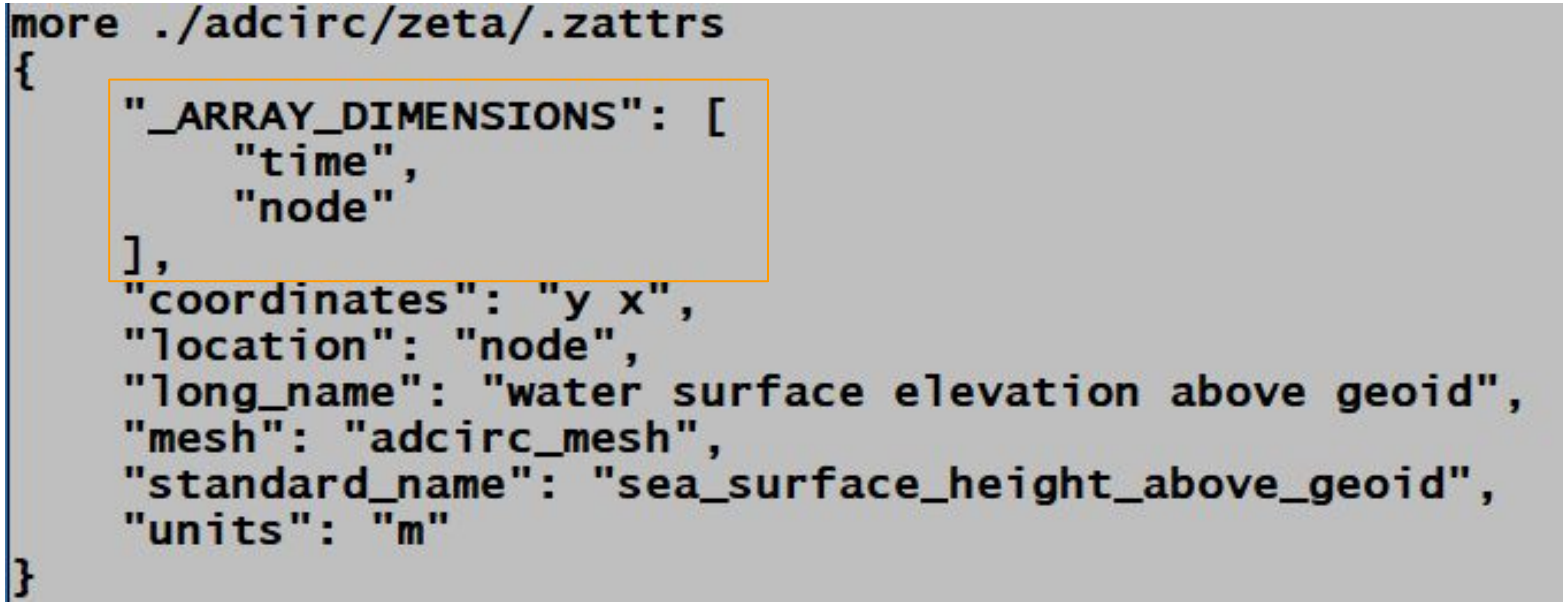
```
(00101010, 10001111, 11001010, 11110010, …)
```

```
‘0.0’ -> (11101011, 10000001, 11111010, 10010010, …)
‘0.1’ -> (00101010, 10001111, 11001010, 11110010, …)
‘1.0’ -> (00111011, 10001010, 11111011, 11010110, …)
‘1.1’ -> (00101010, 10001111, 11001010, 11110010, …)
```

Zarr Format

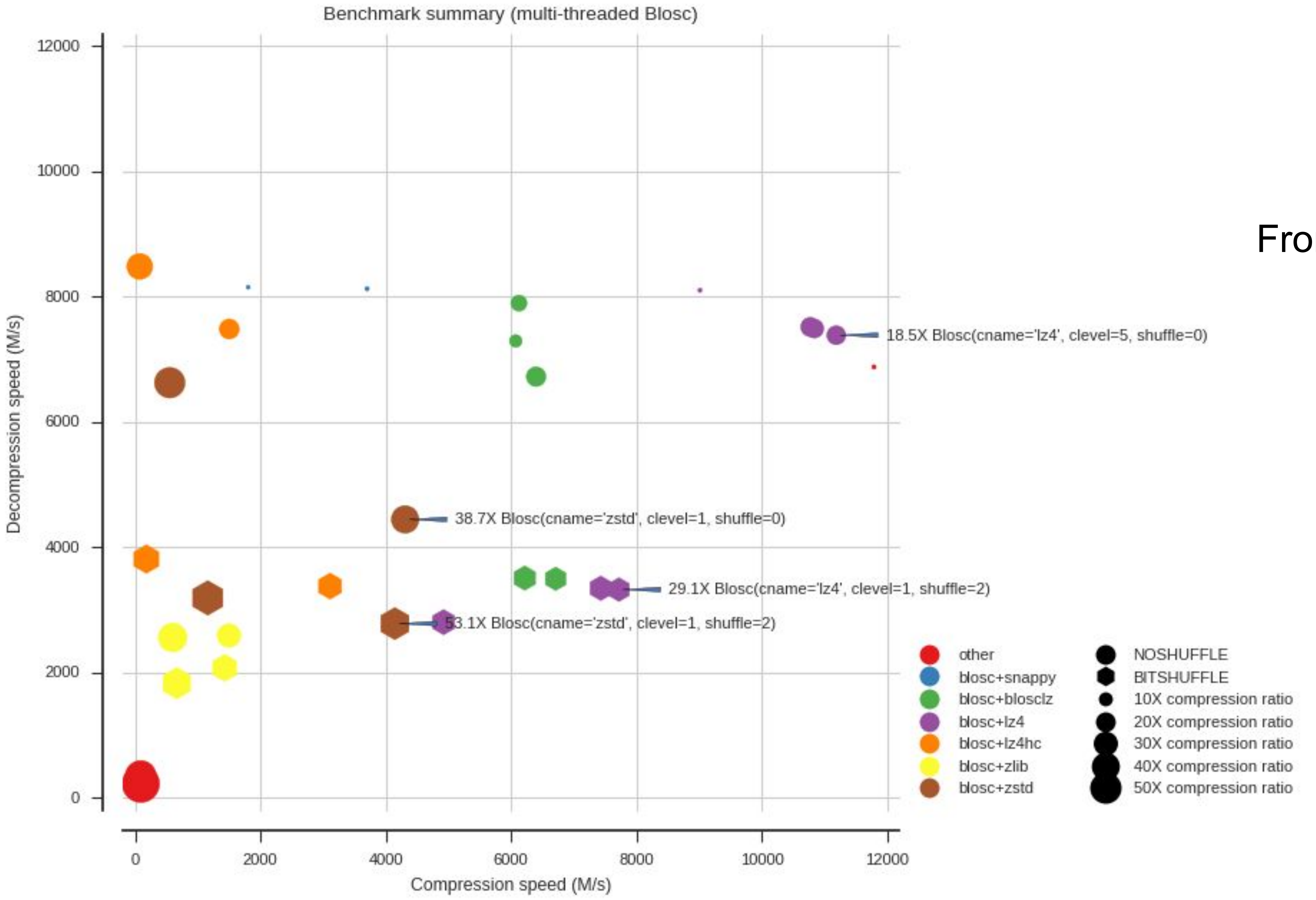


```
more ./adcirc/zeta/.zarray
{
    "chunks": [
        10,
        141973
    ],
    "compressor": {
        "blocksize": 0,
        "clevel": 5,
        "cname": "lz4",
        "id": "blosc",
        "shuffle": 1
    },
    "dtype": "<f8",
    "fill_value": -99999.0,
    "filters": null,
    "order":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



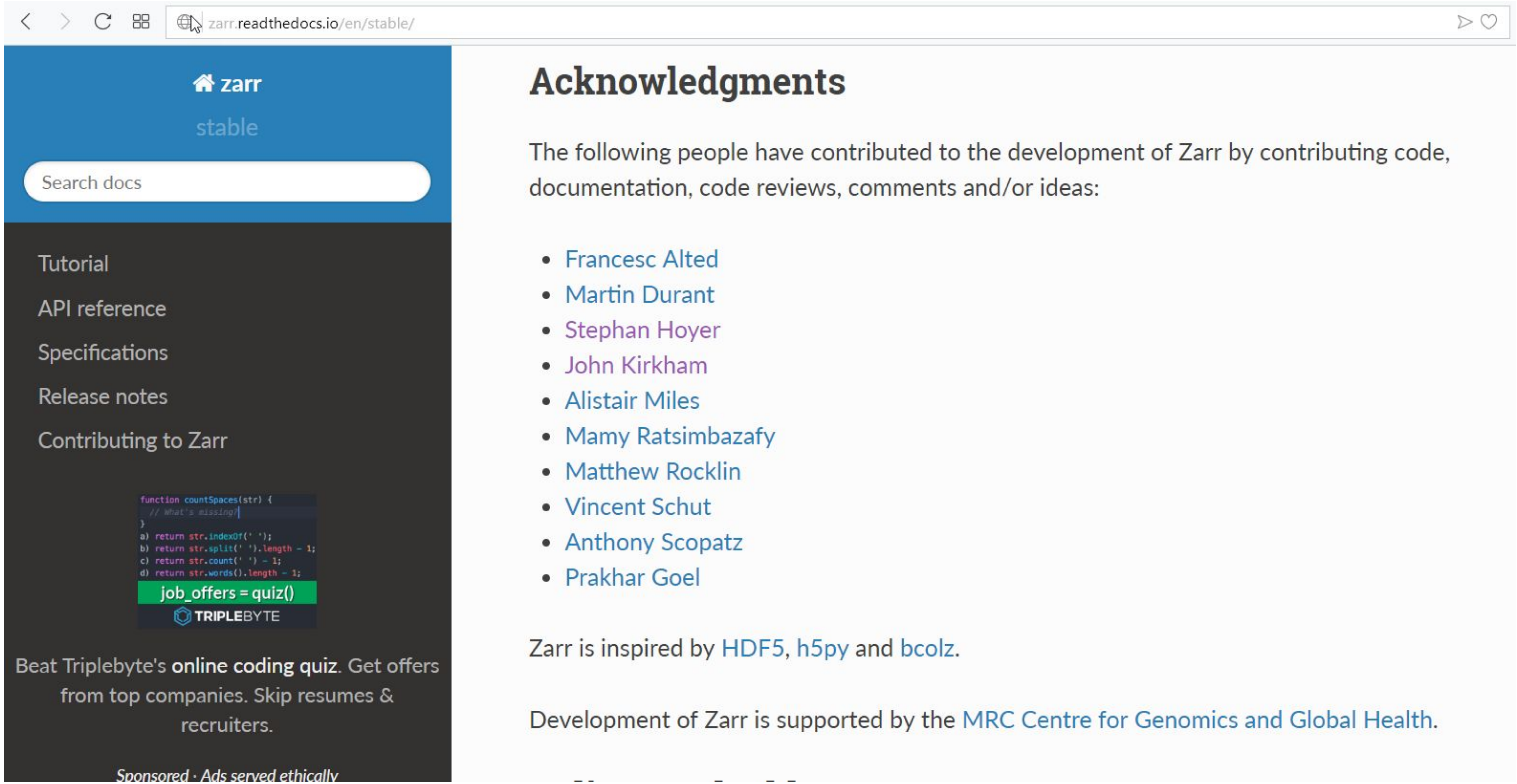
```
more ./adcirc/zeta/.zattrs
{
    "_ARRAY_DIMENSIONS": [
        "time",
        "node"
    ],
    "coordinates": "y x",
    "location": "node",
    "long_name": "water surface elevation above geoid",
    "mesh": "adcirc_mesh",
    "standard_name": "sea_surface_height_above_geoid",
    "units": "m"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

Compression and Performance



From: Alistair Miles' Blog

Zarr is community-driven



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

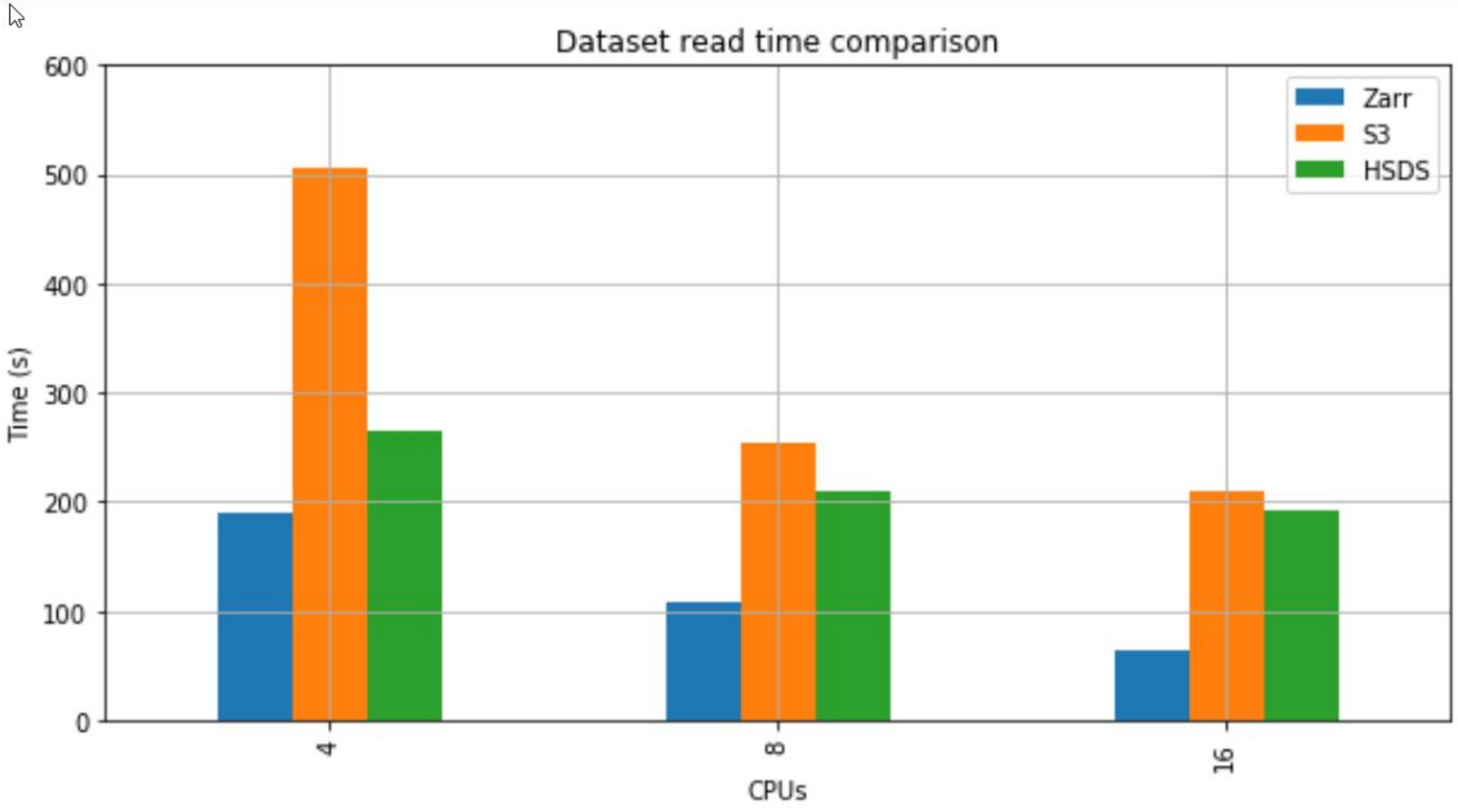
Zarr is inspired by HDF5, h5py and bcolz.

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



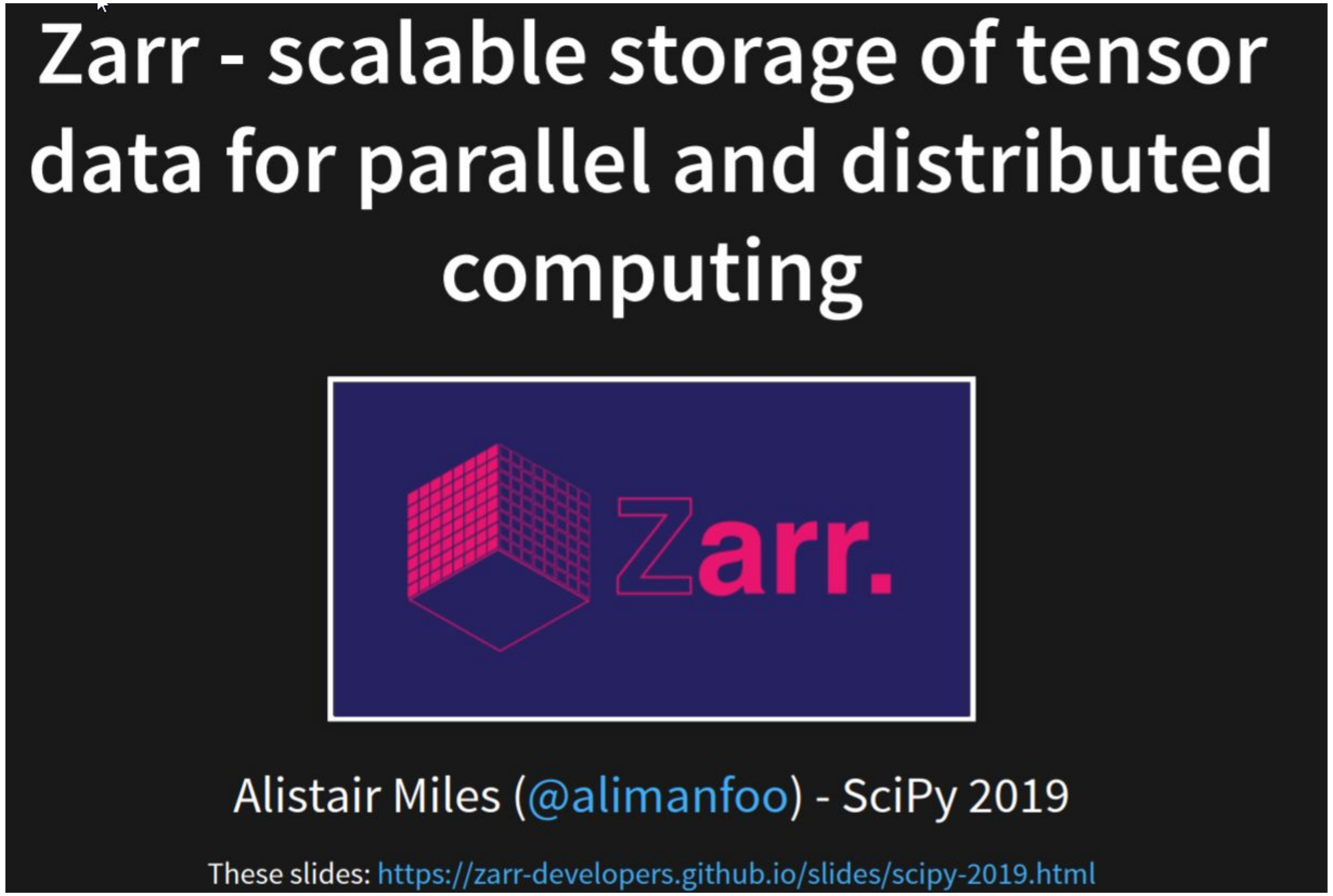
read 58GB dataset written with chunks, filters, compression

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.



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

http://pangeo-data.org https://www.youtube.com/watch?v=qyJXBIrdzBs