Fast, FAIR, and Scalable: Managing Big Data in HPC with Zarr

Alfonso Ladino-Rincon[1], Stephen W. Nesbitt[1], Deepak Cheerian[2]
[1]University of Illinois Urbana-Champaign, Urbana, IL, USA
[2]Earthmover, NY, USA

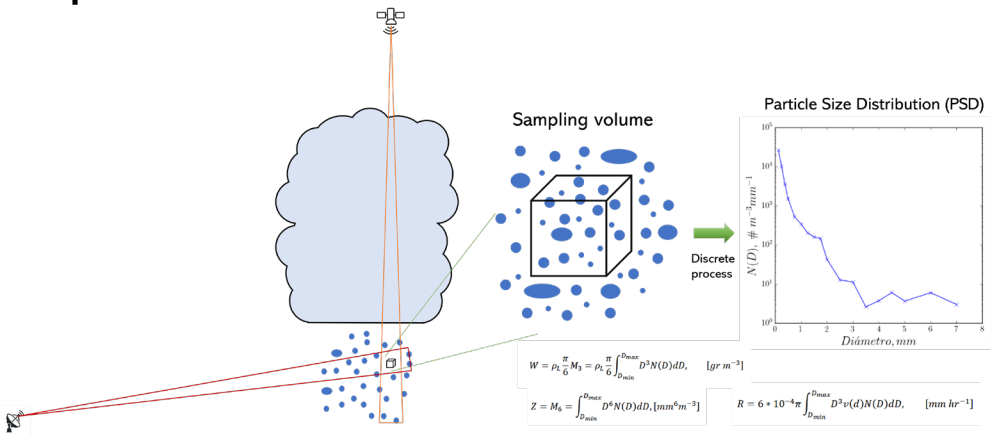
CyberColombia
Bogotá, Colombia | June16th, 2025

About me

Alfonso Ladino, PhD(c).
UIUC

- Cloud and Rain microphysics
- Data Scientist
- ML Enthusiast
- Open-Source contributor - maintainer
- Open Science / Open data advocate

Prior Experience



Open Source Contributions

xarray Xradar PROJECT PYTHIA Raw2Zarr

Affiliations

UNIVERSIDAD NACIONAL DE COLOMBIA

IDEAM

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

earthmover

What to expect in this talk?

Cyber Colombia

MPI +
Big Machines
+
Hardware + C++

Big Data
+ Zarr +
Datacubes
+ Open Data

xarray

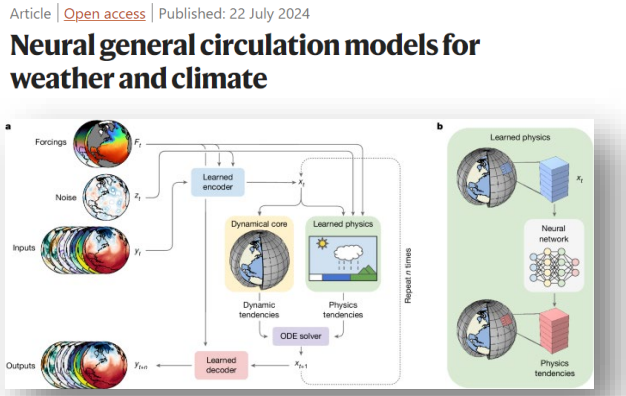
Outline

1. **Challenges with Legacy Formats in HPC**
   - Rapid data growth outpaces NetCDF, HDF5, etc.
   - Poor scalability, cloud access, and collaboration.
2. **Zarr and the Scientific Datacube Model**
   - Cloud-native, chunked, parallel access to arrays.
   - Enables FAIR, scalable, and reproducible workflows.
3. **Real-World Impact: How Zarr Is Transforming Data-Intensive Science**
   - Radar use case: 210× speedup in processing.
   - Zarr + open tools = open data / open science.

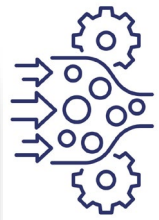
The Explosion of Scientific Data: A Challenge for HPC

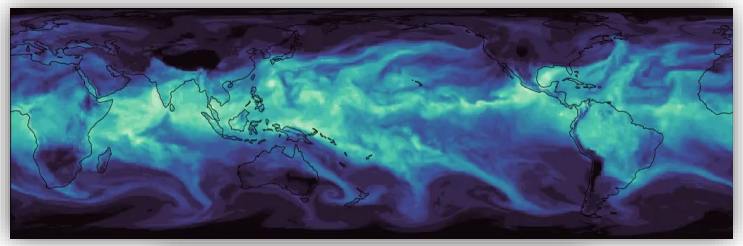
High-performance computing has enabled some of the most important scientific advances

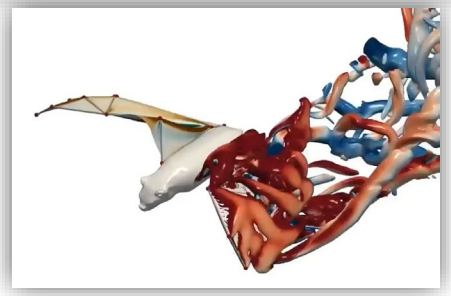
- Run large-scale simulations (e.g., CFD, CMIP6, ML)
- Fine-grained control over compute and memory
- Optimize performance through compilers, MPI, OpenMP



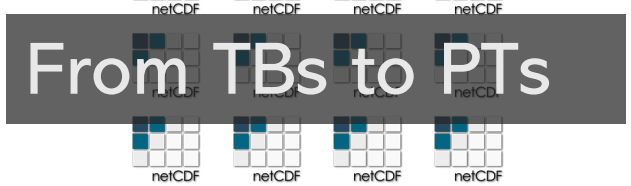
https://doi.org/10.1038/s41586-024-07744-y







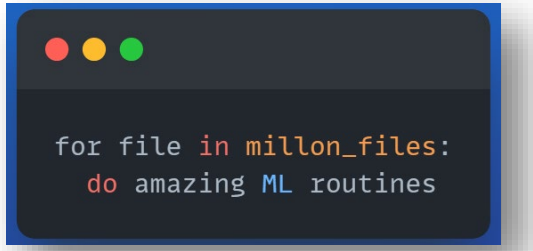
doi:10.1017/jfm.2025.356



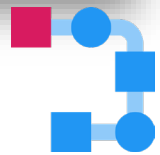
Introducing the Problem with Legacy Formats

Most scientific simulations still archive results in formats designed decades ago — for local disks, single-user access, and small teams. Today, they're choking under cloud-scale data

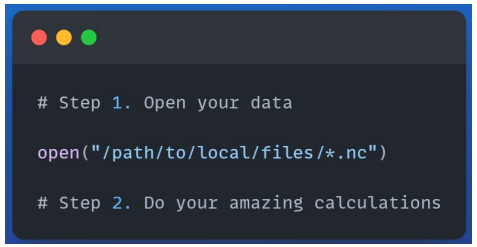
- 🚫 **Monolithic or Millions of Files**
  - Whole-file access required
  - I/O-bounded



- 🔒 **Metadata Locking**
  - Only one process can write; poor parallelism.

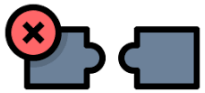


- 🐢 **Hard-to access**
  - Optimized for POSIX
  - Local storage



- ☁️ **Not Cloud-Optimized**
  - Data is not chunked for selective access
  - Data is not Analysis-Ready
  - Does not support concurrent – parallel access

- 🤝 **Poor Interoperability**
  - Hard to integrate with modern Python/ML tools.

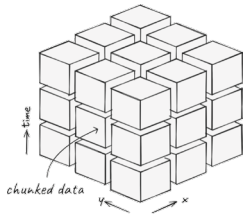


From Files to Datasets: What's Missing?

What modern data formats should support:

- ✅ **Analysis-Ready** – Self described
- ✅ **Cloud-Optimized** – Work well with object storage (S3, GCS, etc.)

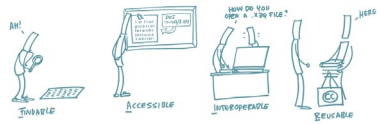
- ✅ **Lazy load** – Metadata
- ✅ **Chunked** – partial queries



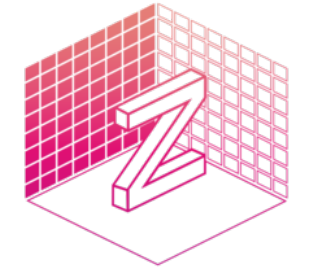
- ✅ **Parallelism** – Safe concurrent reads/writes

- ✅ **Compression** – Save storage without losing speed
- ✅ **Interoperability** – Easy integration with Python, Dask, Jupyter

- ✅ **Openness** – Based on open standards and open source
- ✅ **FAIR Principles** – Findable, Accessible, Interoperable, Reusable



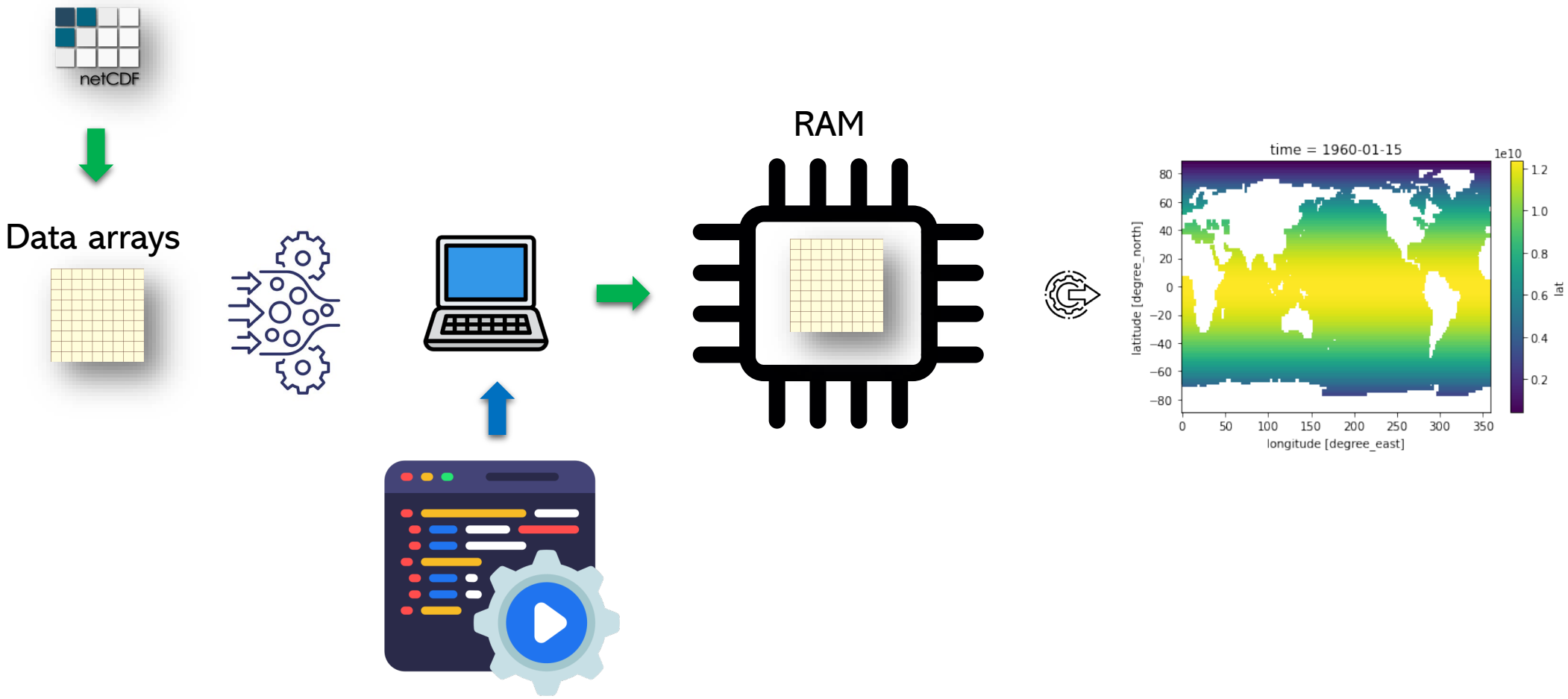
2. Zarr and the Scientific Datacube Model



Zarr

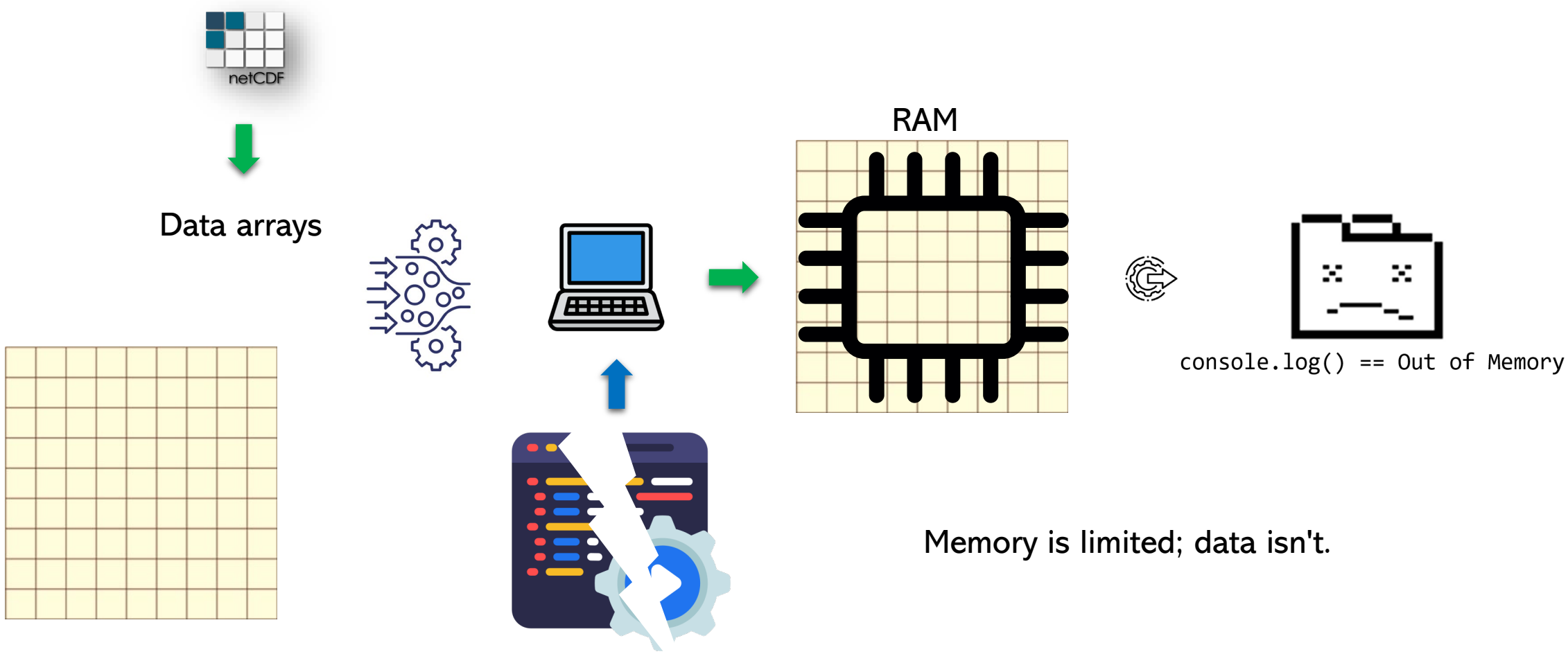
Zarr Starts with a Simple Problem: Memory Limits

To process large data efficiently, we need to rethink how it's stored and accessed.



Zarr Starts with a Simple Problem: Memory Limits

To process large data efficiently, we need to rethink how it's stored and accessed.



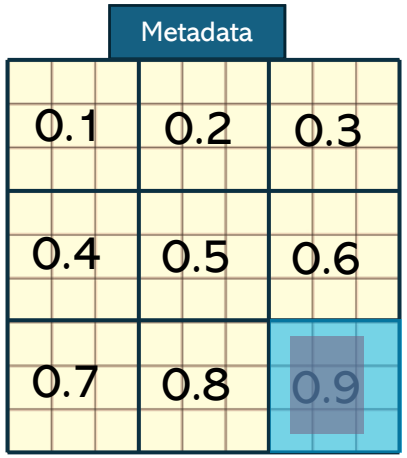
Memory is limited; data isn't.

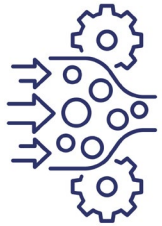
Zarr Starts with a Simple Problem: Memory Limits

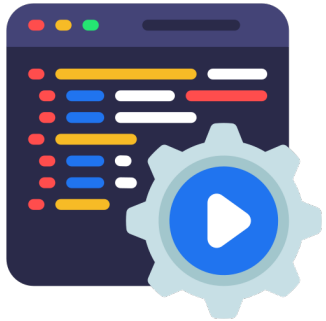
To process large data efficiently, we need to rethink how it's stored and accessed.

Zarr isn't just a format — it's a shift in how we organize, store, and think about scientific data.

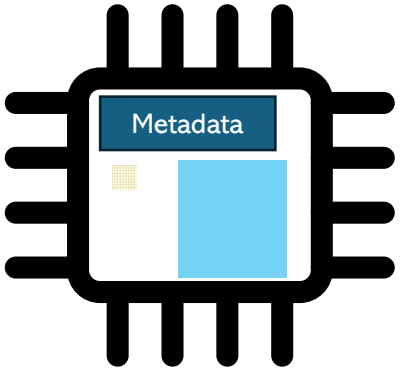
Data arrays

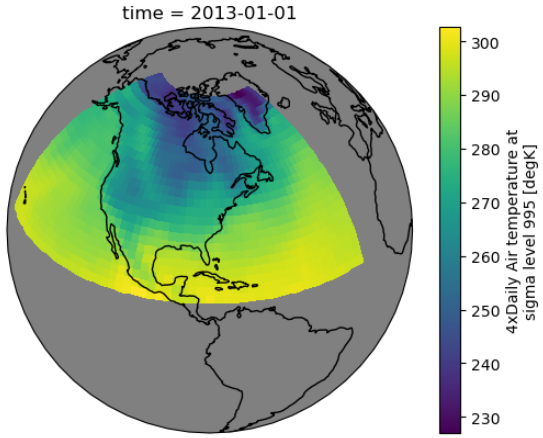






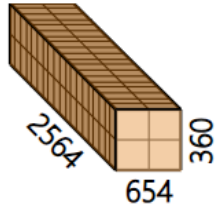
RAM





In memory repr.

	Array	Chunk
Bytes	2.25 GiB	229.92 kiB
Shape	(2564, 360, 654)	(1, 180, 327)
Dask graph	10256 chunks in 2 graph layers	
Data type	float32 numpy.ndarray	



- ✅ Lazy load
- ✅ Chunked
- ✅ Parallelism

What Is Zarr? A Cloud-Native Format for Tensor Data

Zarr is a format for storing large N-dimensional arrays — in chunks — with metadata, designed for fast, parallel access from local or cloud storage.

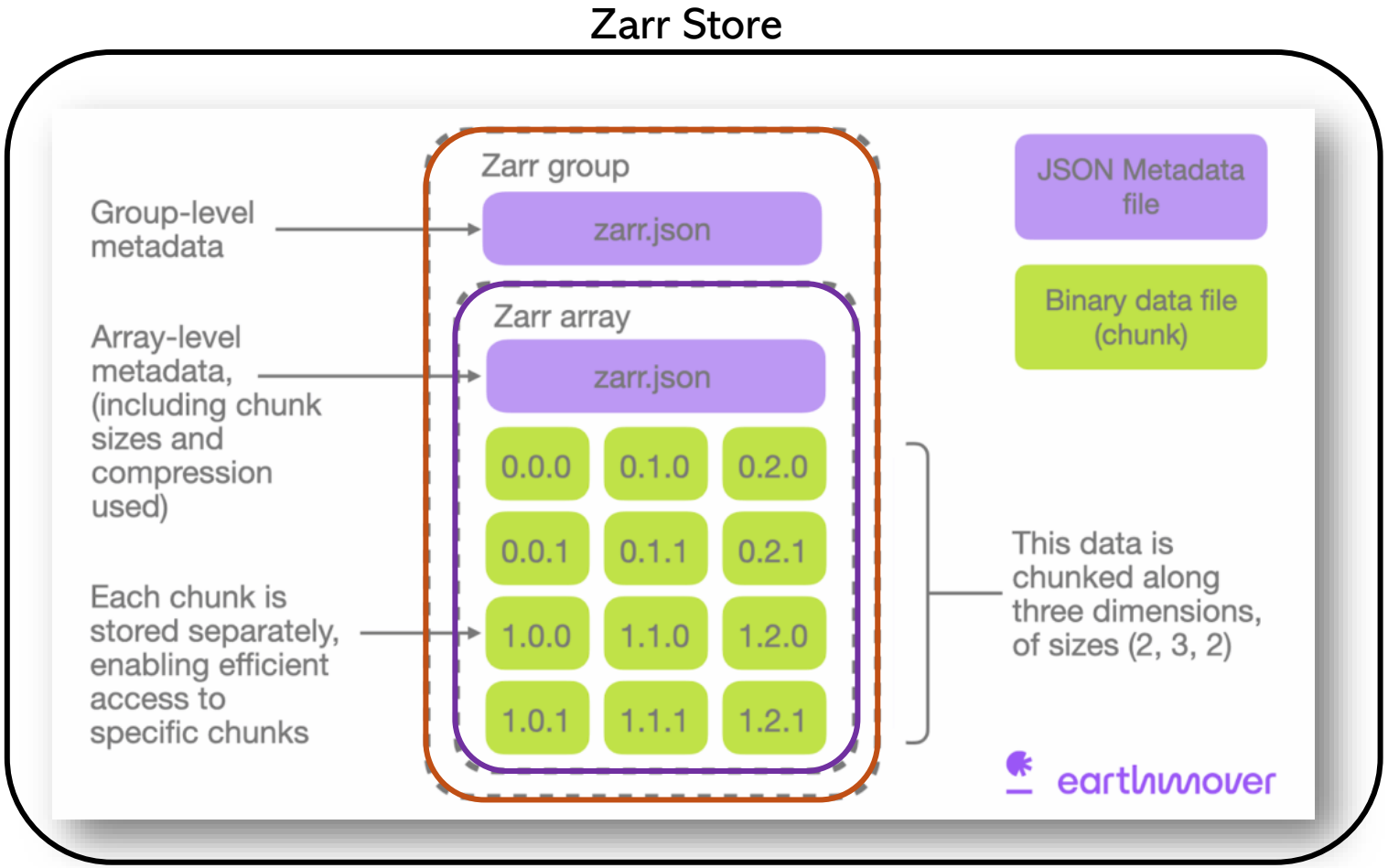
Zarr organizes data in 2 key concepts:

Store:

- A directory on disk (e.g., LocalStore)
- A cloud bucket (e.g., FsspecStore for s3://…)
- An in-memory dictionary (e.g., MemoryStore)

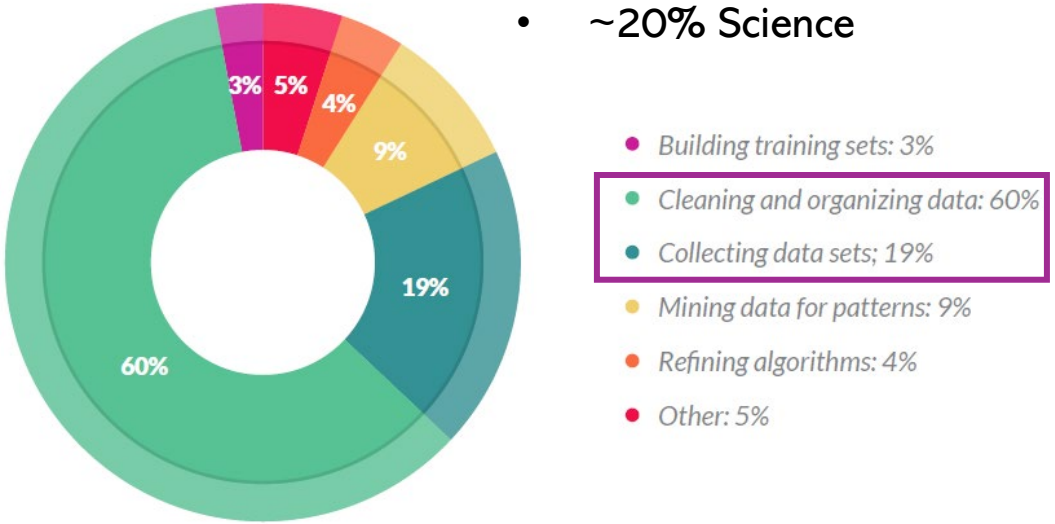
Group:

- A group can contain multiple related Zarr arrays
- It can also hold other groups (nested hierarchy)
- Each group has its own metadata (in zarr.json)



From Storage to Analysis: What Makes Data ARCO?

Analysis-Ready Cloud-Optimized means you can open the dataset, ask a question, and get an answer — no conversions, no preprocessing, no downloading the whole thing



- ~80% Data wrangling
- ~20% Science

How do data scientists spend their time?
Crowdflower Data Science Report (2016)

📦 Analysis-Ready

- ✅ Think in terms of datasets, not individual files.
- ✅ Preprocessed and standardized for analysis.
- ✅ Enriched with metadata and published in catalogs for discovery.

☁️ Cloud-Optimized

- ✅ Stored in Object Storage (not in traditional databases)
- ✅ Self-described (includes its own metadata)
- ✅ Accessible via HTTP
- ✅ Supports parallel and distributed access

Exploring an ARCO Dataset

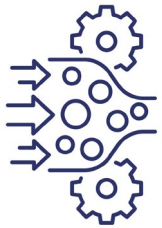
```
import xarray as xr

path = "../raw2zarr/zarr/KVNX.zarr"

dtree = xr.open_datatree(
    path,
    engine="zarr",
    chunks={}
)

dtree
```

Metadata



xarray.DataTree

Only one DataTree object

► Groups: (4)

Multiple VCPs

► Dimensions:

► Coordinates: (0)

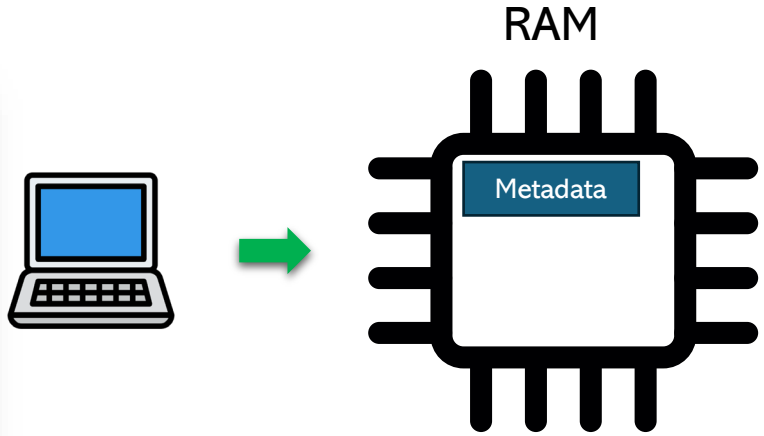
► Inherited coordinates: (0)

► Data variables: (0)

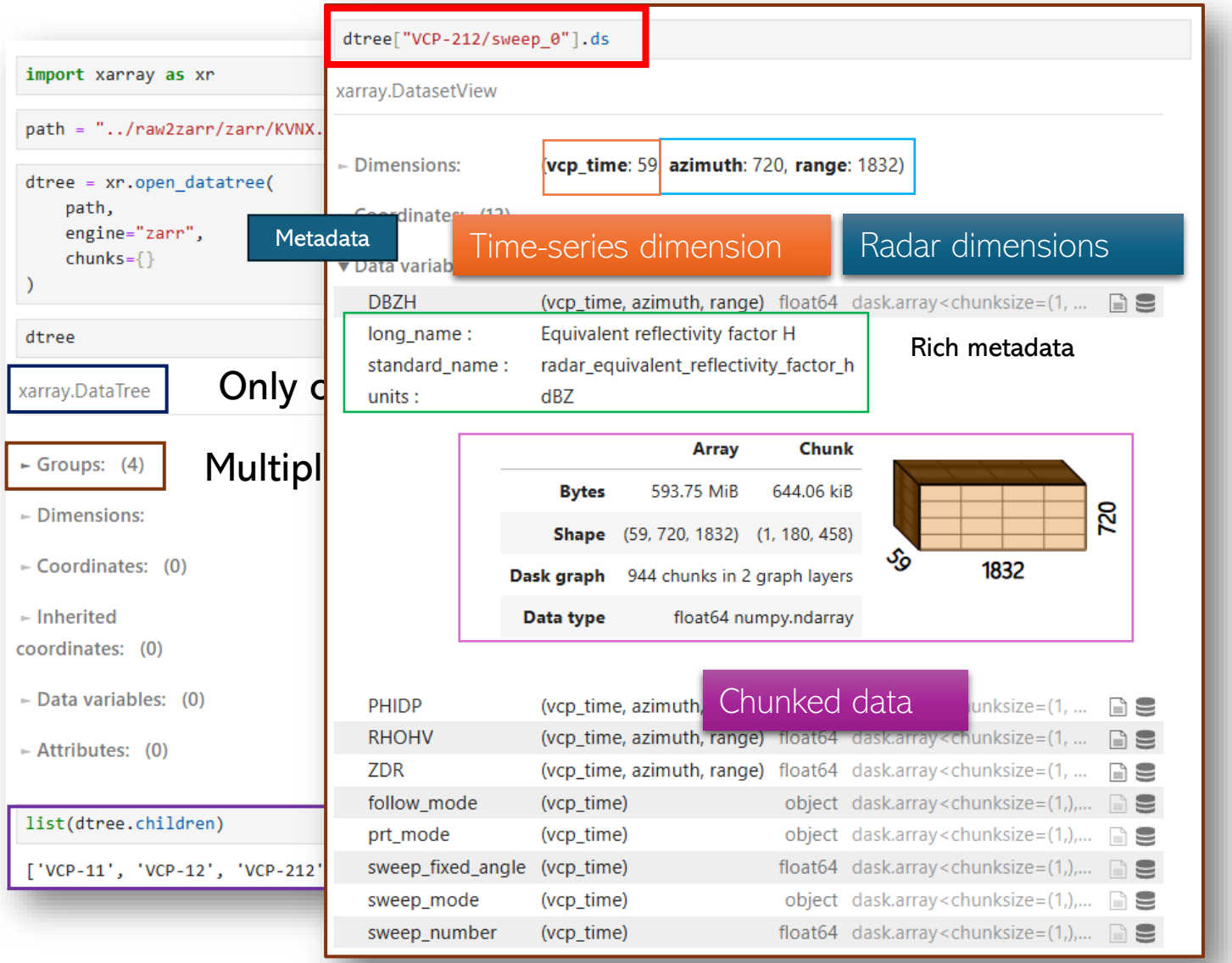
► Attributes: (0)

```
list(dtree.children)
```

```
['VCP-11', 'VCP-12', 'VCP-212', 'VCP-32']
```



Exploring an ARCO Dataset



- ✓ Dataset
- ✓ Self-described
- ✓ Standard-compliant format
- ✓ Analysis-Ready
- ✓ Cloud-Optimized

Exploring an ARCO Dataset

```
import xarray as xr

path = "../raw2zarr/zarr/KVNX.

dtree = xr.open_datatree(
    path,
    engine="zarr",
    chunks={}
)

dtree
```

xarray.DataTree

Only c

- Groups: (4)
- Dimensions:
- Coordinates: (0)
- Inherited coordinates: (0)
- Data variables: (0)
- Attributes: (0)

Multipl

```
list(dtree.children)
```

```
['VCP-11', 'VCP-12', 'VCP-212'
```

```
dtree["VCP-212/sweep_0"].ds
```

xarray.DatasetView

- Dimensions: (**vcp_time**: 59, **azimuth**: 720, **range**: 1832)
- Coordinates: (12)

Radar

- Data variables:

DBZH	(vcp_time, azimuth, range)	float64	dask.array<
long_name :	Equivalent reflectivity factor H		
standard_name :	radar_equivalent_reflectivity_factor_h		
units :	dBZ		

Rich

	Array	Chunk
Bytes	593.75 MiB	644.06 kiB
Shape	(59, 720, 1832)	(1, 180, 458)
Dask graph	944 chunks in 2 graph layers	
Data type	float64 numpy.ndarray	

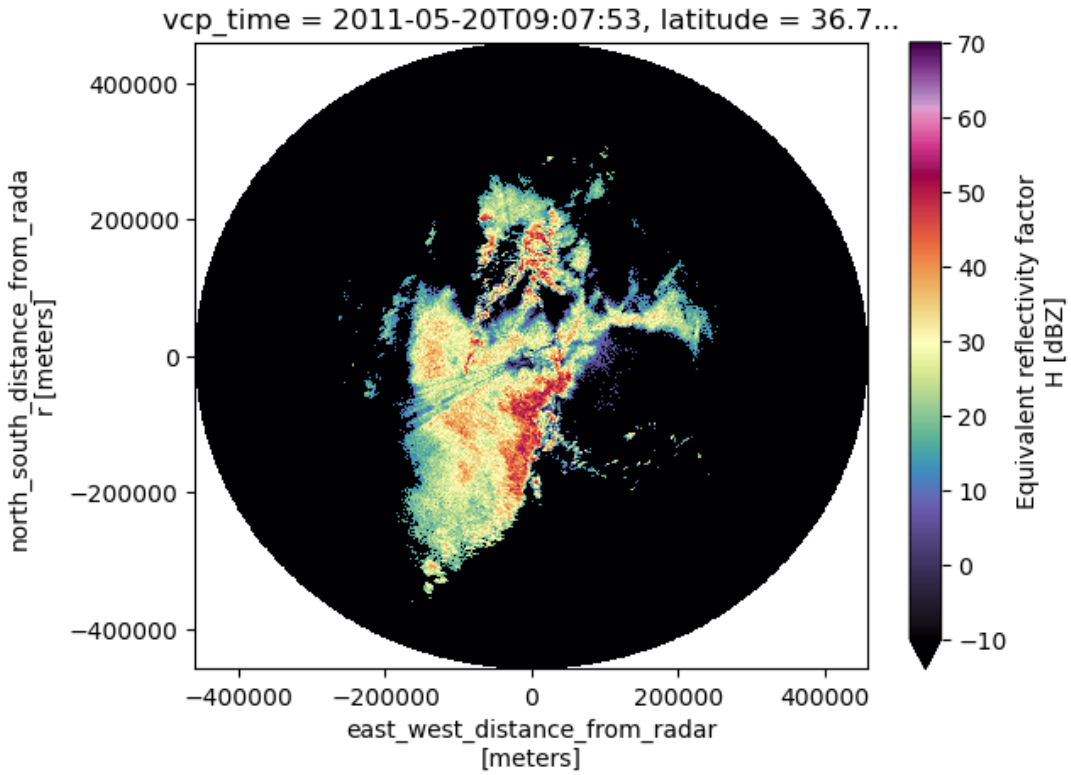
59

Chunked data

PHIDP	(vcp_time, azimuth,		
RHOHV	(vcp_time, azimuth, range)	float64	dask.array<
ZDR	(vcp_time, azimuth, range)	float64	dask.array<
follow_mode	(vcp_time)	object	dask.array<
prt_mode	(vcp_time)	object	dask.array<
sweep_fixed_angle	(vcp_time)	float64	dask.array<
sweep_mode	(vcp_time)	object	dask.array<
sweep_number	(vcp_time)	float64	dask.array<

```
dt_radar["VCP-12/sweep_0/DBZH"].isel(vcp_time=1).plot(
    x="x",
    y="y",
    vmin=-10,
    vmax=70,
    # robust=True,
    cmap="ChaseSpectral",
)
```

```
<matplotlib.collections.QuadMesh at 0x79247b4ae810>
```



Architecture for open, cloud-based, and scalable data

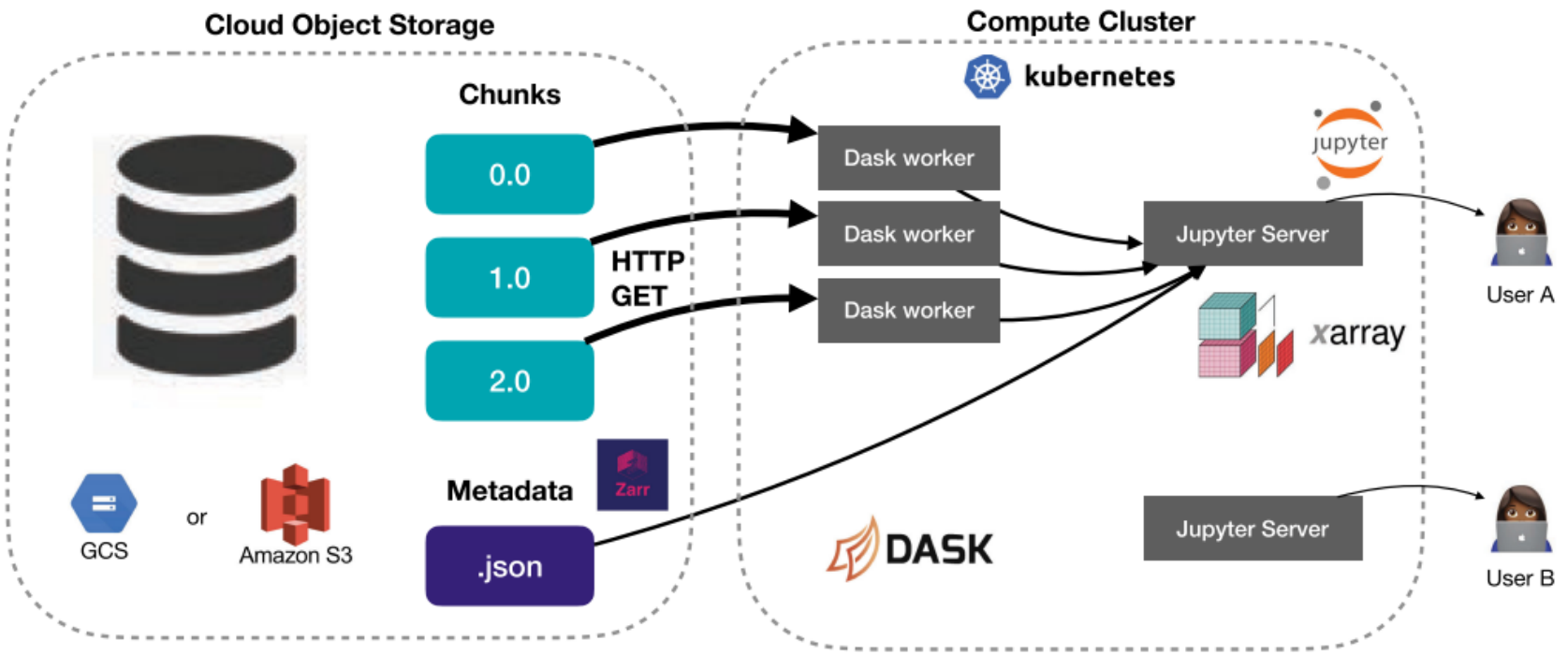
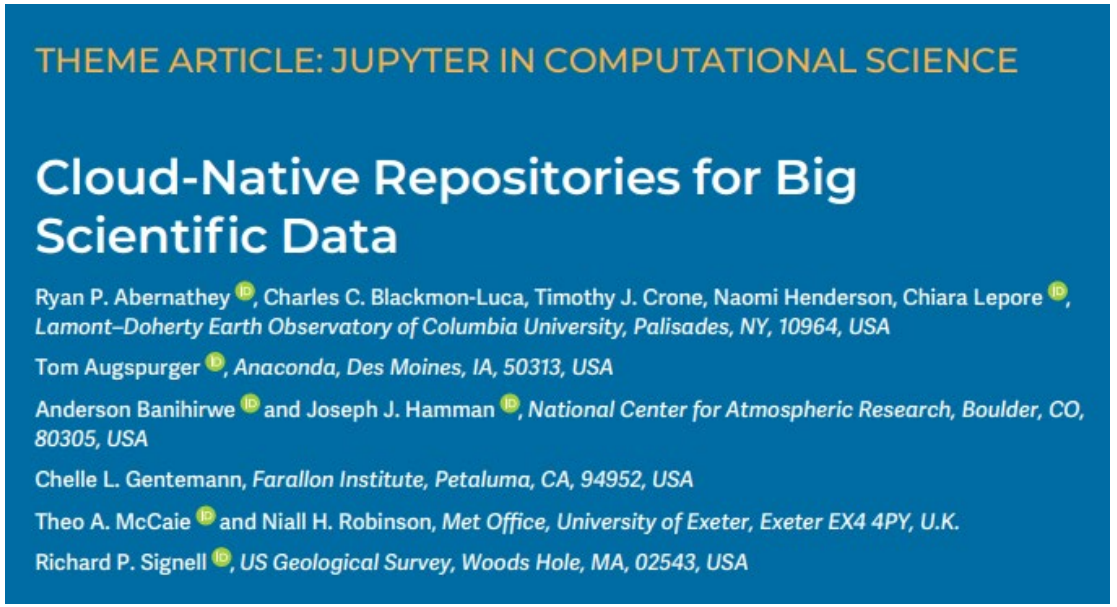


FIGURE 2. Pangeo architecture diagram. The data repository is hosted in cloud object storage (left), in the Zarr format. Compute nodes inside a Kubernetes cluster (right) fetch data and metadata from the object store. Users connect to the system via Jupyter and write interactive data analysis code in Xarray, which dispatches computations on an adaptively scaling Dask cluster.

(Abernathey et al., 2021)

ARCO Datasets Deliver High Throughput

The benchmark uses satellite oceanographic data from NASA's ECCO (Estimating the Circulation and Climate of the Ocean) project.



THEME ARTICLE: JUPYTER IN COMPUTATIONAL SCIENCE

Cloud-Native Repositories for Big Scientific Data

Ryan P. Abernathey, Charles C. Blackmon-Luca, Timothy J. Crone, Naomi Henderson, Chiara Lepore, *Lamont–Doherty Earth Observatory of Columbia University, Palisades, NY, 10964, USA*

Tom Augspurger, *Anaconda, Des Moines, IA, 50313, USA*

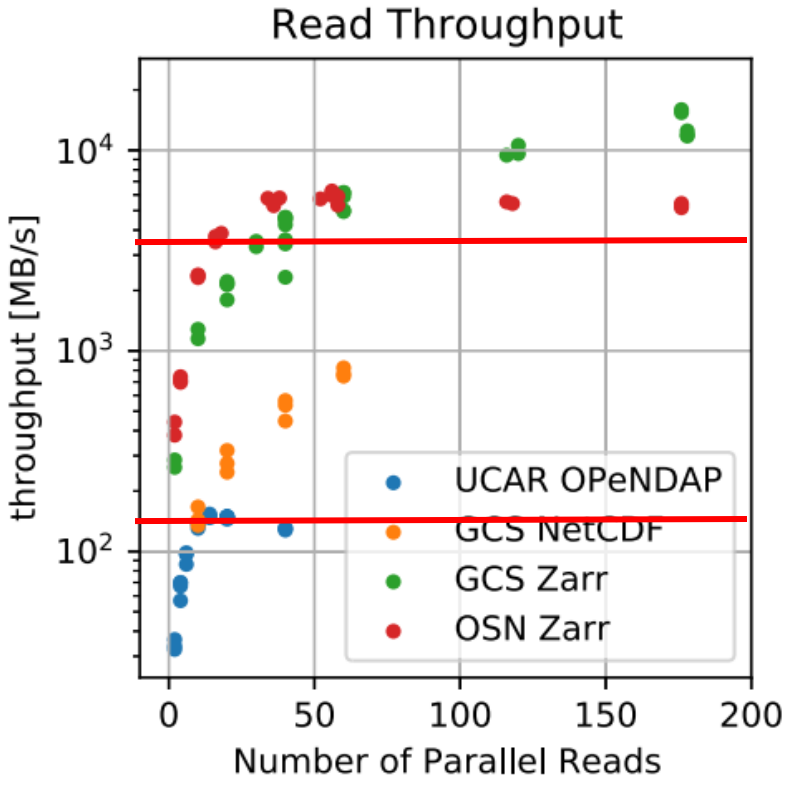
Anderson Banihirwe and Joseph J. Hamman, *National Center for Atmospheric Research, Boulder, CO, 80305, USA*

Chelle L. Gentemann, *Farallon Institute, Petaluma, CA, 94952, USA*

Theo A. McCaie and Niall H. Robinson, *Met Office, University of Exeter, Exeter EX4 4PY, U.K.*

Richard P. Signell, *US Geological Survey, Woods Hole, MA, 02543, USA*

https://doi.org/10.1109/MCSE.2021.3059437



10GB/s

1GB/s

100MB/s

Enables fast analysis over large volumes of scientific data (Big Data)

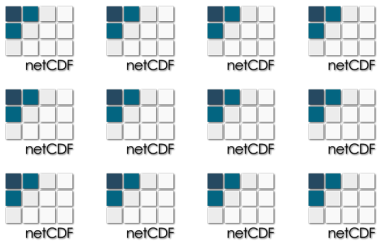
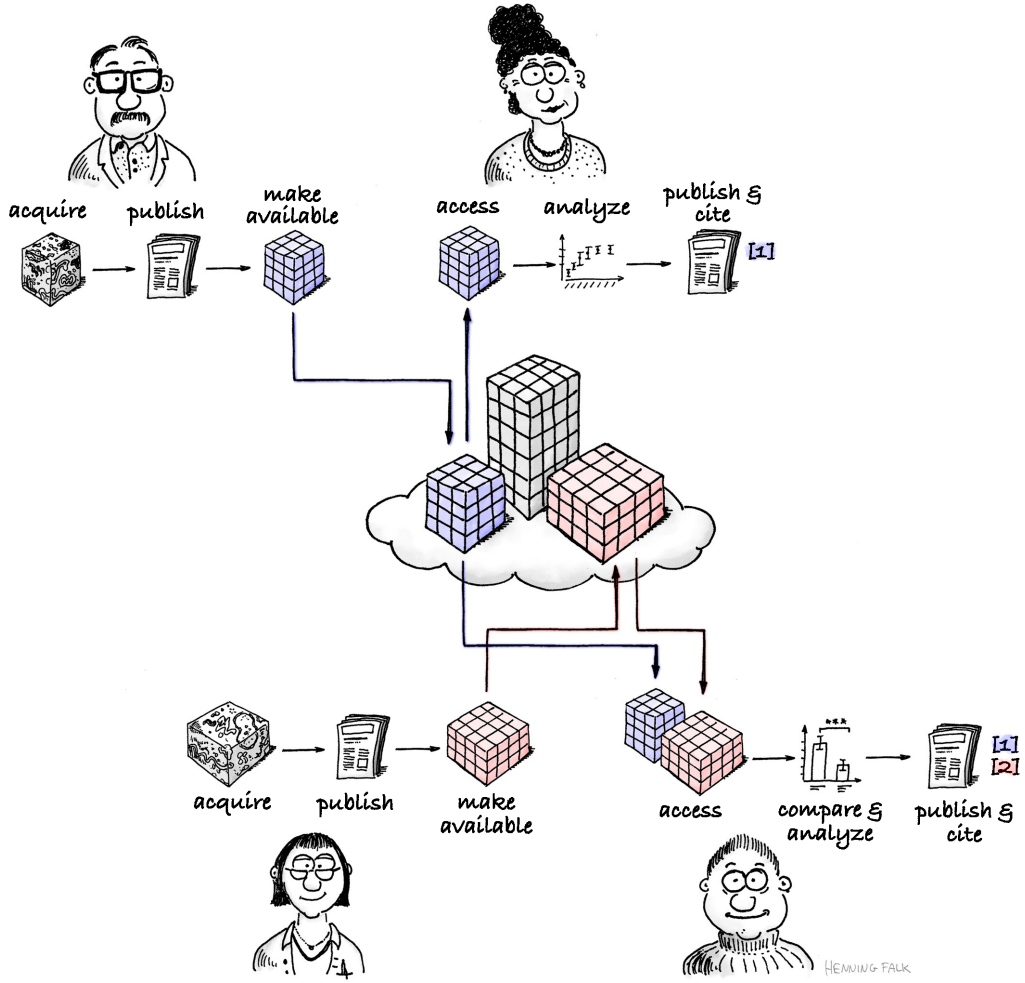


FIGURE 3. Read throughput of different data access methods. Source data available at https://zenodo.org/record/3,829,032. Code available at http://gallery.pangeo.io/repos/earthcube2020/ec20_abernathey_etal/cloud_storage.html.

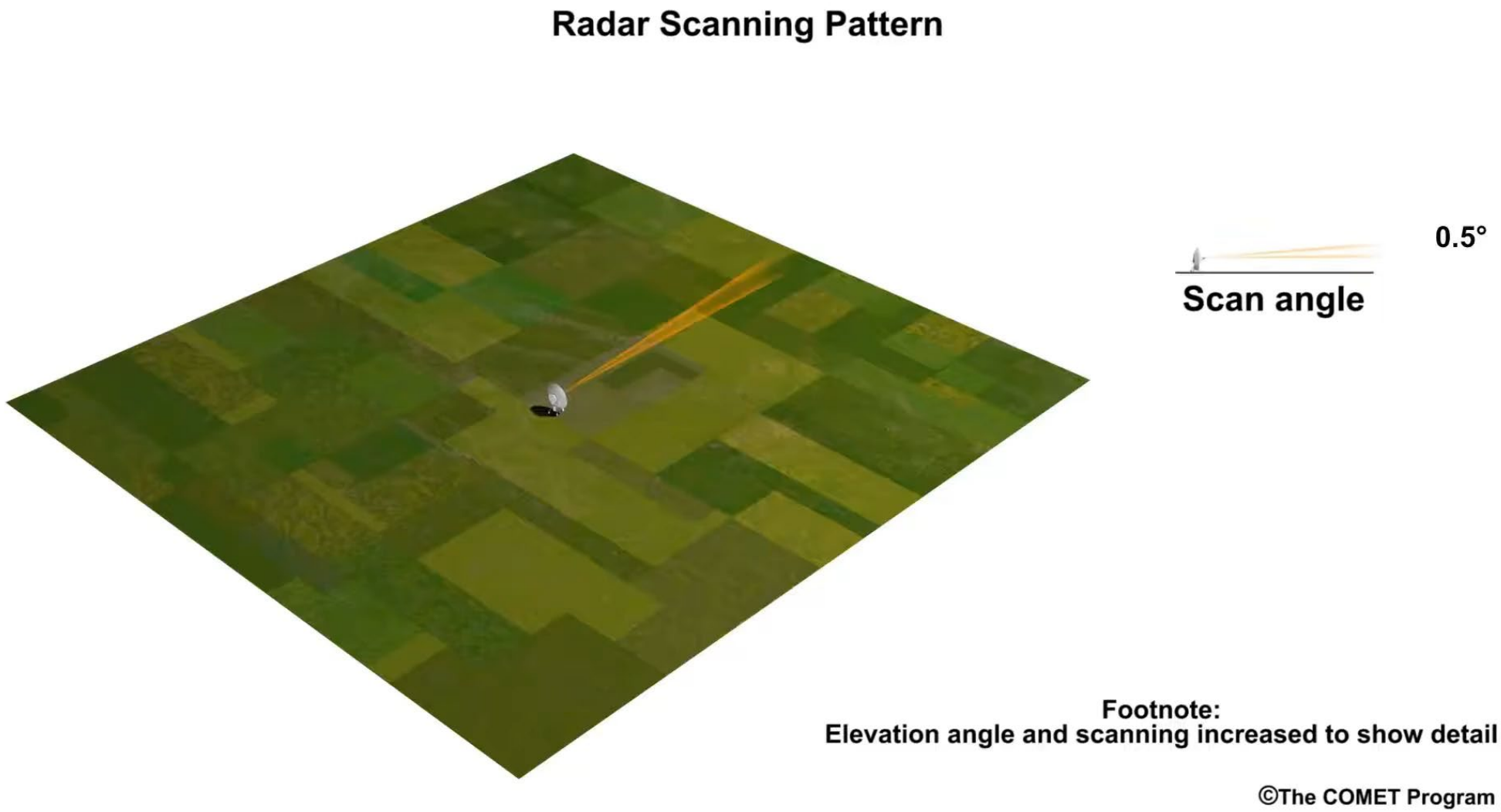
Zarr: Designed for Big Data, Built for Scientists



Courtesy: zarr-illustrations-falk-2022

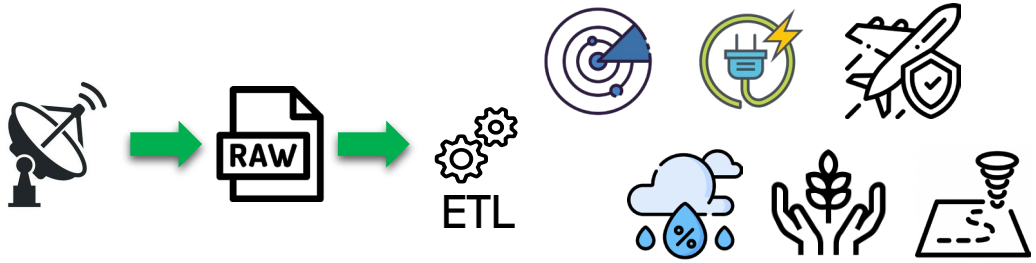
3. Real-World Impact: How Zarr Is Transforming Data-Intensive Science

Weather radar data: High-resolution, real-time, and dynamic



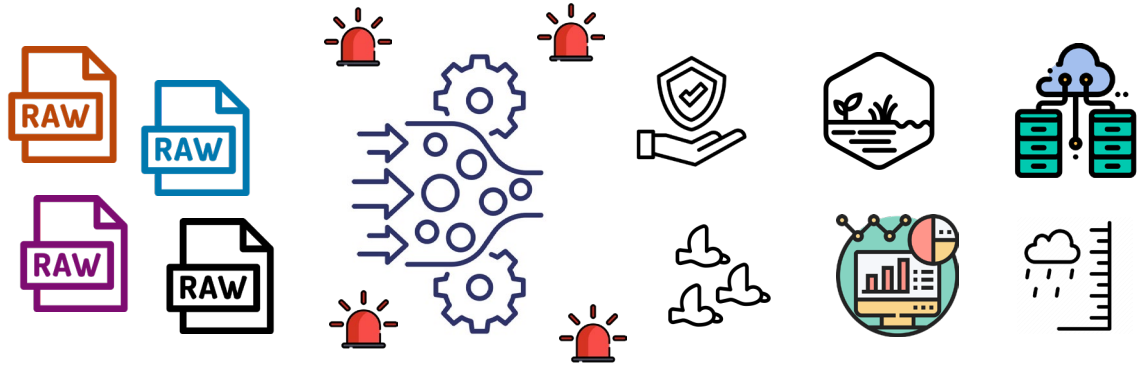
Weather Radars: from “real-time” to long-term applications

“Real-time” products



- Severe Weather Detection
- Precipitation Monitoring
- Marine and Coastal Applications
- Power and Utility Operations
- Public Safety and Emergency Response
- Aviation Safety

Long-term applications (non “real-time”)



- Climatological Studies
- Machine Learning
- Long Term Agricultural Support
- Advanced Meteorological Research
- Ecological and Wildlife Research (bird migration)

Large-scale analysis of radar products demand **extensive input-output (I/O)** operations over data stored in proprietary (binary) formats.

Worldwide distribution of ground-based radars

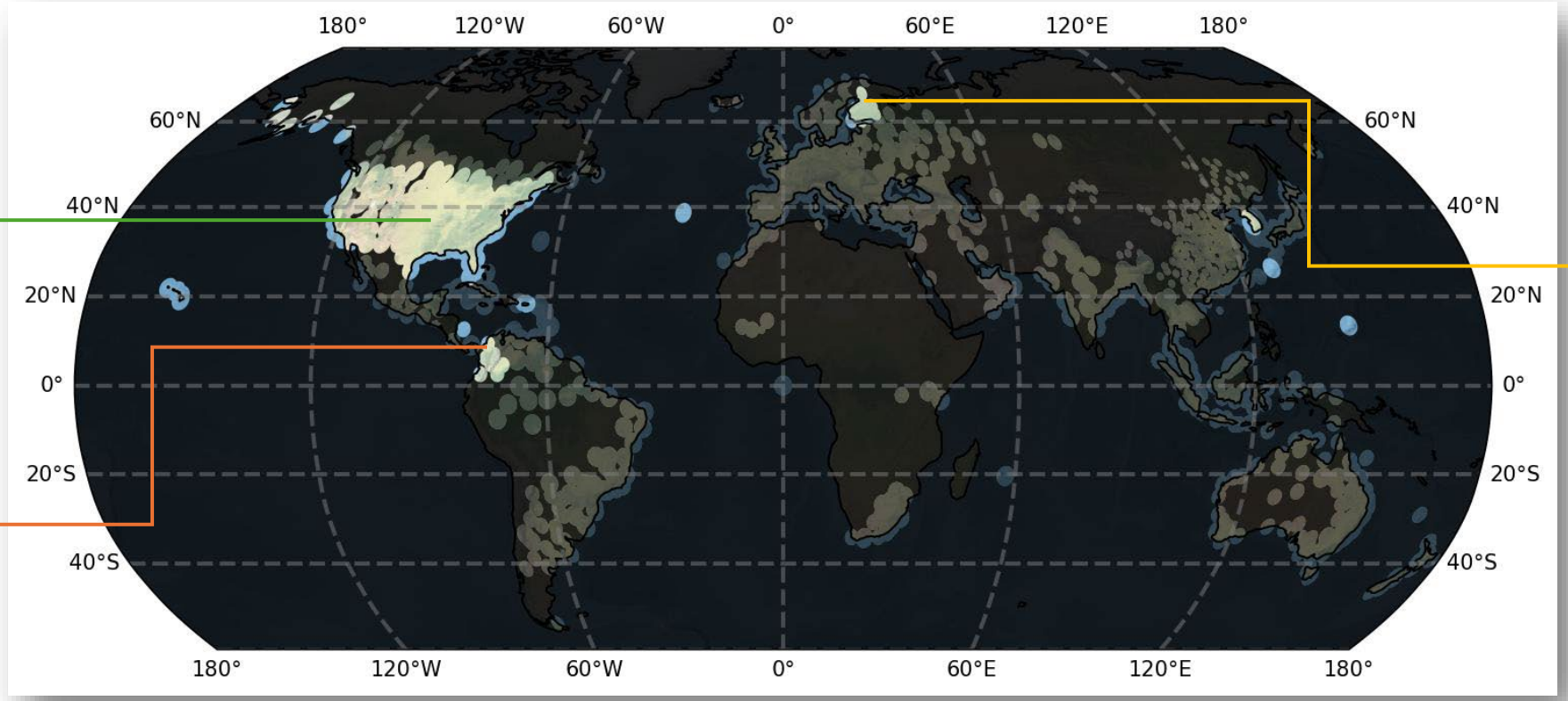
91 Countries (NWS) operate ~809 weather radars (WMO, 2025).

🔓 But only 3 national networks provide open data access

📉 **That's just about 20% of all radars worldwide.**

NEXRAD
~360M files
(870 TB)

Colombian RN
~14M files
(111 TB)



FMI RN
~16M files
(145 TB)

*NWS: National Weather Service
*RN: Radar Network

Public radar data is not fully FAIR-compliant.

🔓 **Open ≠ FAIR**

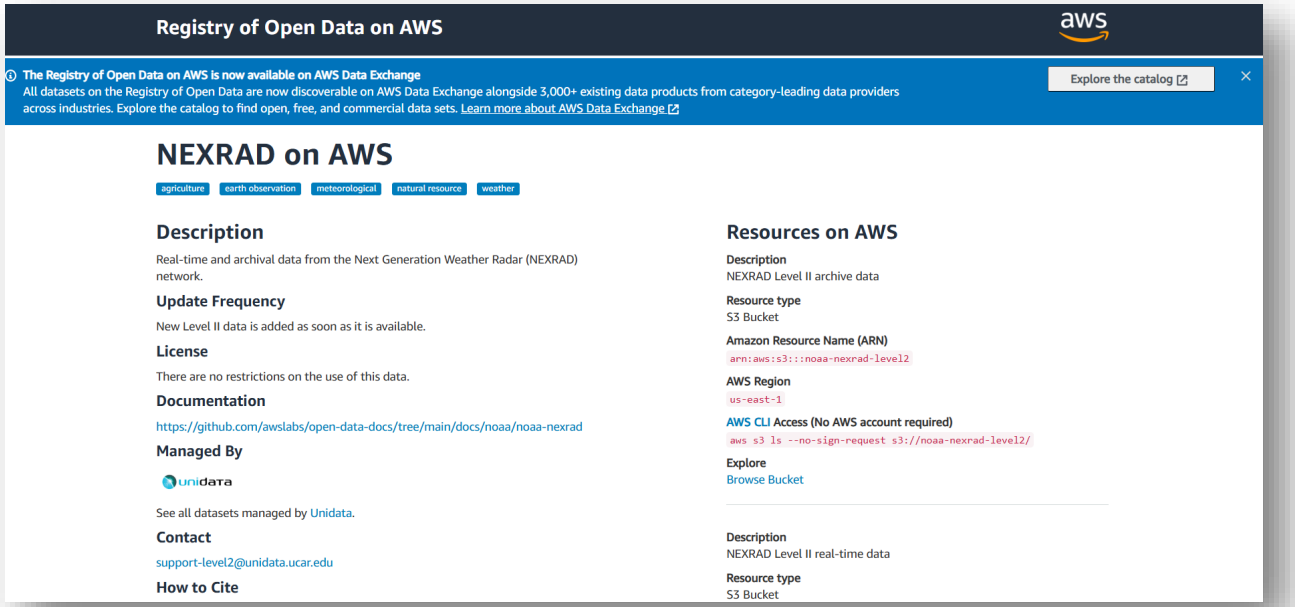
🤔
The rest?

~360 Million Records of Radar Data: The NEXRAD Treasure Trove

NOAA's Big Data Partnership

In 2016, NOAA transferred the **NEXRAD Level II** historical archive to Amazon Web Services (AWS) (Ansari et al., 2018)

- Initially: 1991 to 2016, ~270 TB (~180M files)
- As of Feb 2025: ~870 TB (~360M files)



https://registry.opendata.aws/noaa-nexrad/

Accessible data, but …

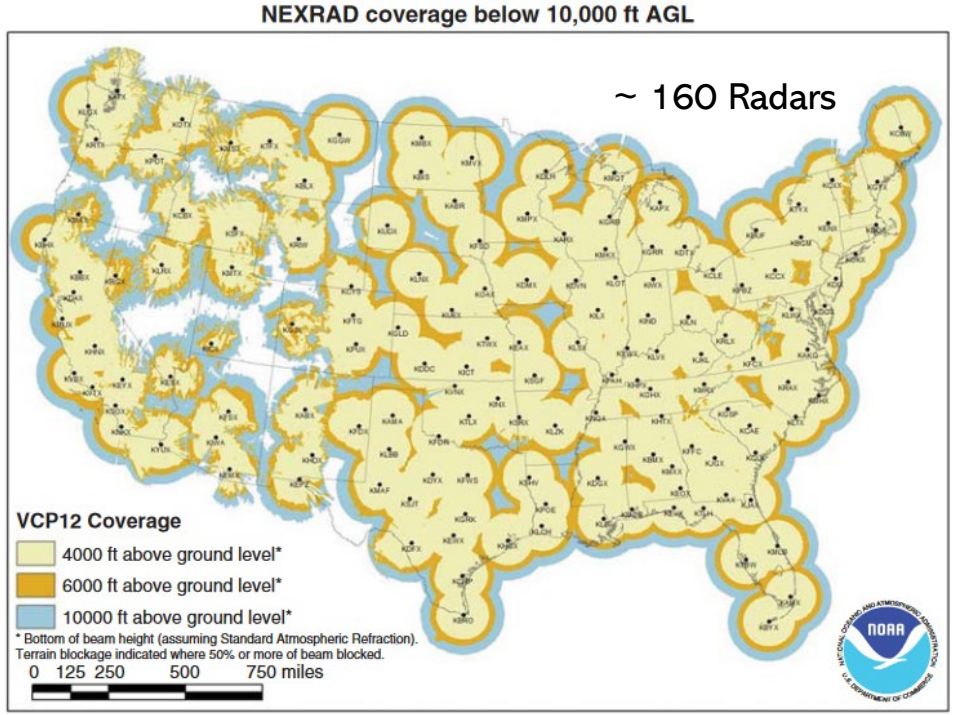


Figure 10.1 Coverage of the WSR-88D network at 4000, 6000, and 10,000 ft altitude above ground level across the continental USA (Courtesy NOAA)

~360 Million Records of Radar Data: The NEXRAD Treasure Trove

NOAA's Big Data Partnership

AWS S3 Explorer | noaa-nexrad-level2 / 2011 / 11 / 11 / KVNX

Hide folders? Folder Bucket 154

50 entries per page Search:

Object	Last Modified	Timestamp	Size
KVNX20111111_000509_V06.gz	9 years ago	2015-09-08 19:15:56	2 MB
KVNX20111111_001452_V06.gz	9 years ago	2015-09-08 19:15:57	2 MB
KVNX20111111_002436_V06.gz	9 years ago	2015-09-08 19:15:58	2 MB
KVNX20111111_003419_V06.gz	9 years ago	2015-09-08 19:16:00	2 MB
KVNX20111111_004402_V06.gz	9 years ago	2015-09-08 19:16:01	2 MB
KVNX20111111_005345_V06.gz	9 years ago	2015-09-08 19:16:02	2 MB
KVNX20111111_010328_V06.gz	9 years ago	2015-09-08 19:16:03	2 MB
KVNX20111111_011310_V06.gz	9 years ago	2015-09-08 19:16:04	2 MB
KVNX20111111_012252_V06.gz	9 years ago	2015-09-08 19:16:05	3 MB
KVNX20111111_013235_V06.gz	9 years ago	2015-09-08 19:16:06	3 MB
KVNX20111111_014218_V06.gz	9 years ago	2015-09-08 19:16:08	3 MB

https://registry.opendata.aws/noaa-nexrad/

Accessible data, but …

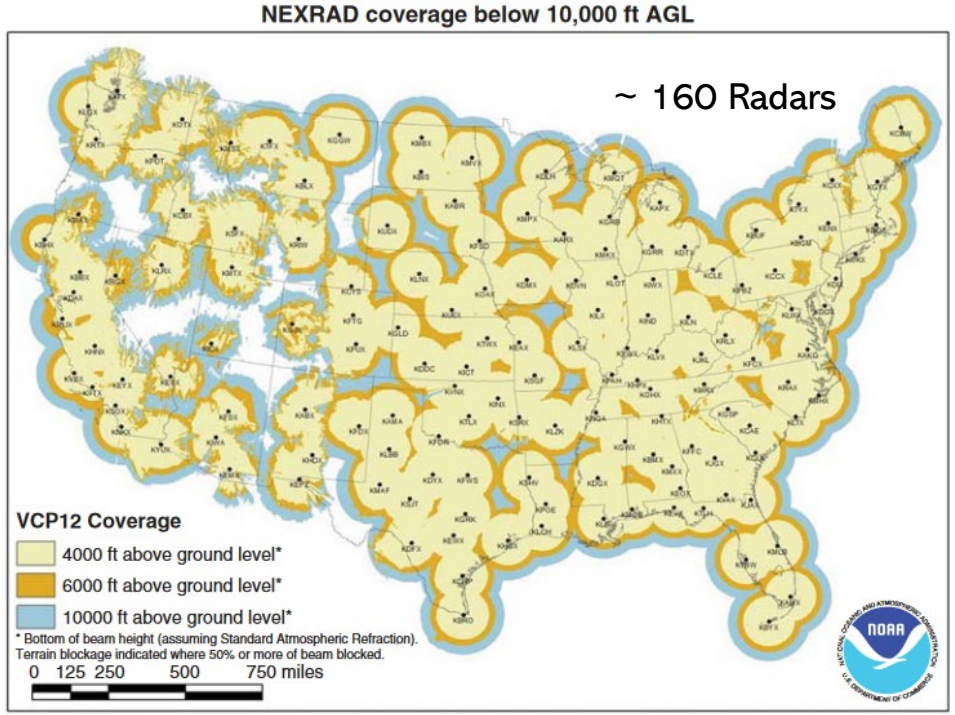


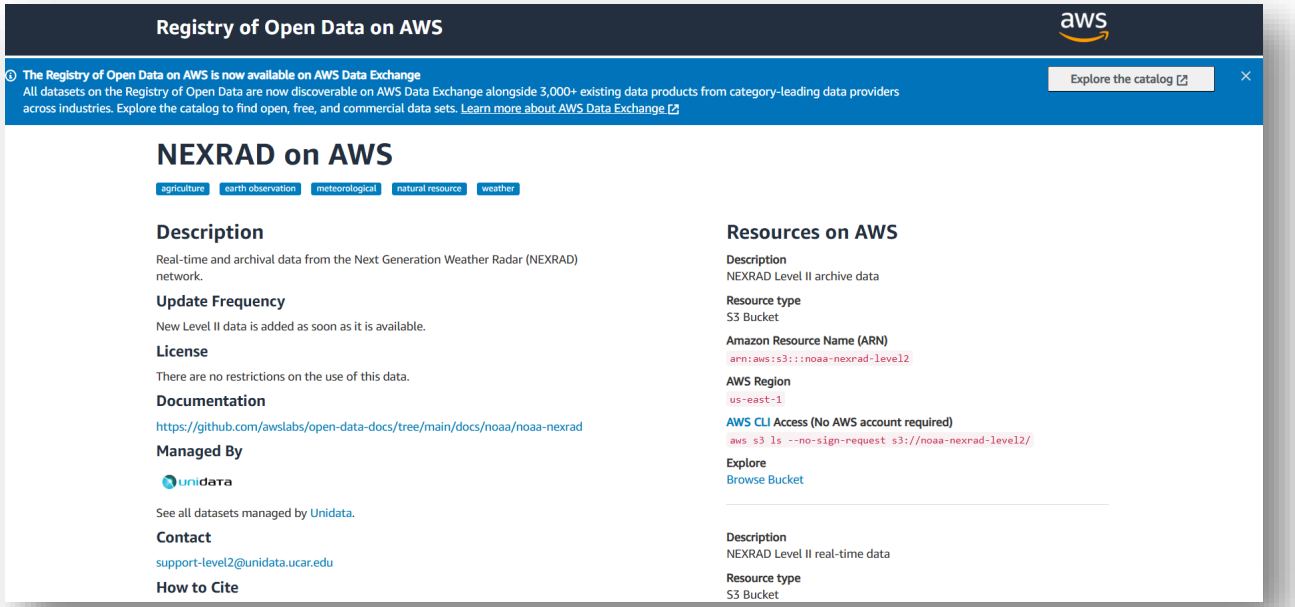
Figure 10.1 Coverage of the WSR-88D network at 4000, 6000, and 10,000 ft altitude above ground level across the continental USA (Courtesy NOAA)

~360 Million Records of Radar Data: The NEXRAD Treasure Trove

NOAA's Big Data Partnership

In 2016, NOAA transferred the **NEXRAD Level II** historical archive to Amazon Web Services (AWS) (Ansari et al., 2018)

- Initially: 1991 to 2016, ~270 TB (~180M files)
- As of Feb 2025: ~870 TB (~360M files)



https://registry.opendata.aws/noaa-nexrad/

Accessible data, but …

- Binary files (Raw format)
- Non-interconnected
- Not FAIR data

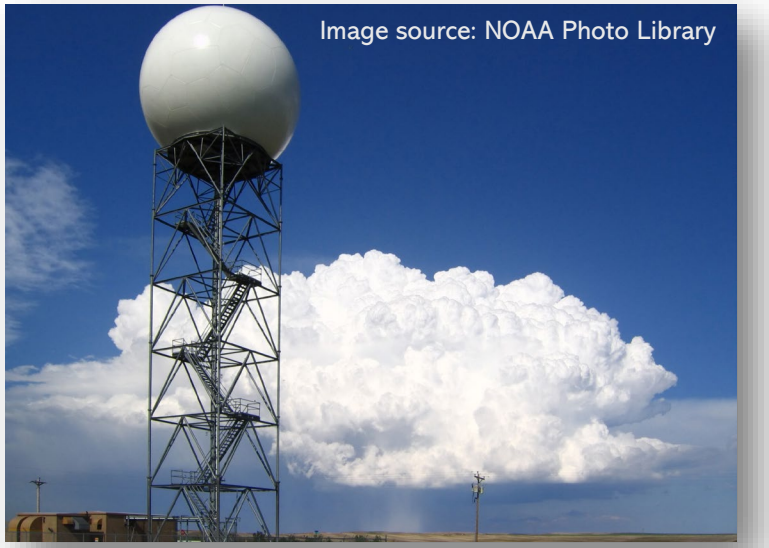
Limits Open Science!

- Accessibility barriers
- Data duplication
- Not interoperable
- Not reusable
- Complicates reproducibility

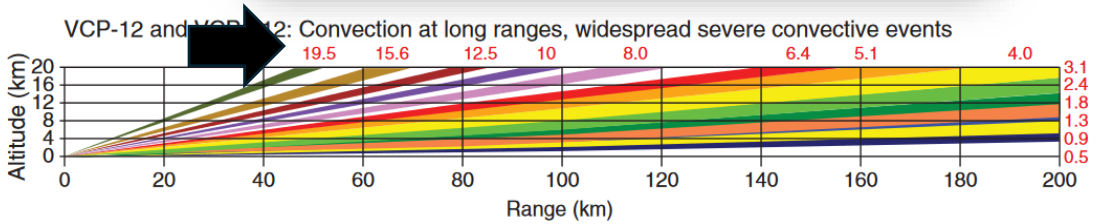
Heterogeneity in radar data: Volume Pattern Coverage - VCPs

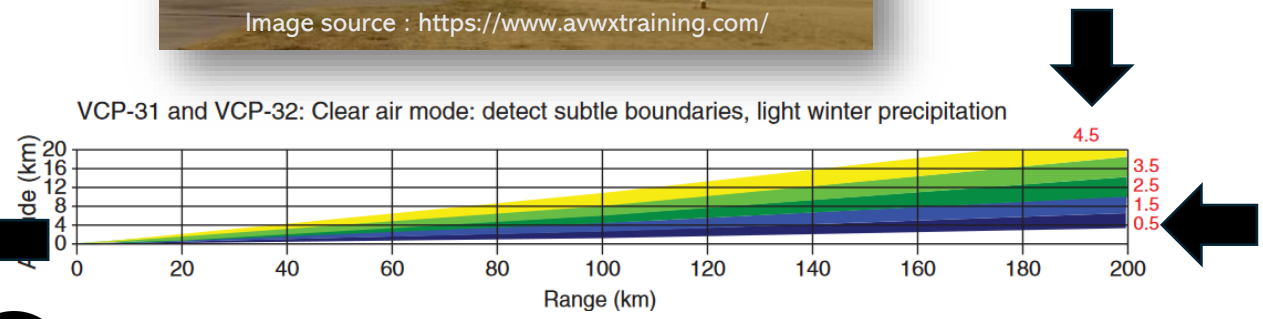
🎯 Different weather, different scan strategies — and that leads to very different data.

Precipitation mode



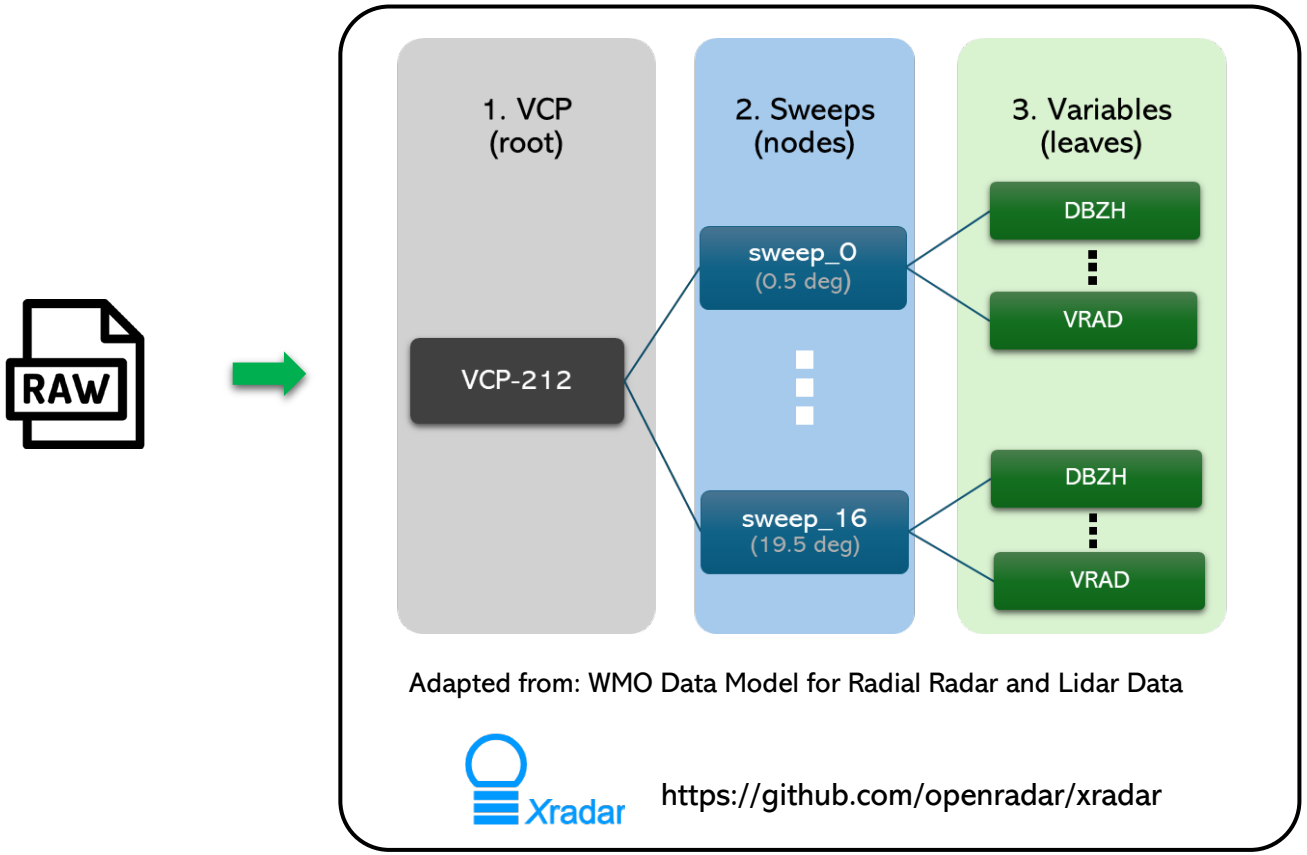
Clear air mode





Bringing Structure to Radar Data: The WMO-FM301 Standard

This hierarchical structure is designed to store radar measurements efficiently adhering to NetCDF-CfRadial structure using Climate and Forecast (CF) conventions.



xarray → DataTree model

Root group:

- Global attributes
- Ancillary variables

Node level:

- Sweep groups
- Sweep attributes

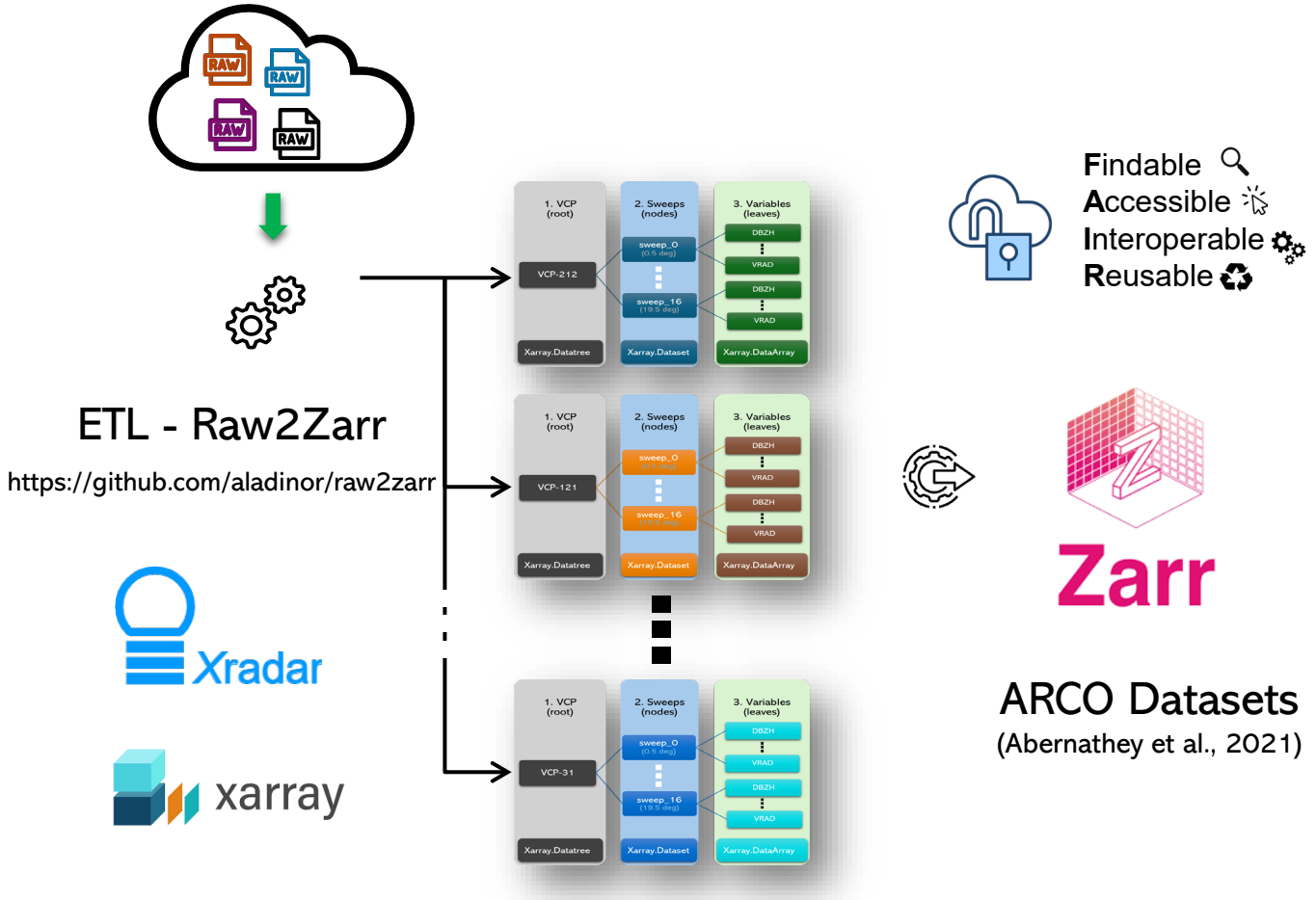
Leaf level:

- Data variables
- Metadata

However, managing thousands of files across multiple VCPs remains a challenge.

From Files to FAIR: Introducing the "Radar DataTree" model

Radar DataTree that aggregates all individual radar files into a single, comprehensive hierarchical dataset for seamless time-series analysis and scalable cloud-based storage, while preserving its unique structure of each VCP.



ARCO Datasets
(Abernathey et al., 2021)

```
KVNX.zarr
.
|-VCP-11
|     |-- sweep_0
|     |-- sweep_1
|     .
|     .
|     .
|     |_ sweep_n
|
|-VCP-12
|     |-- sweep_0
|     |-- sweep_1
|     .
|     .
|     .
|     |_ sweep_n
|
|-VCP-212
|     |-- sweep_0
|     |-- sweep_1
|     .
|     .
|     .
|     |_ sweep_n
```

QVPs: Vertical Radar Insights from 360° Scans

Introduced by Ryzhkov et al. (2016), Quasi-Vertical Profiles (QVPs) extract vertical storm structure from radar scans by averaging data along constant elevation angle (20°) across a full 360° sweep.

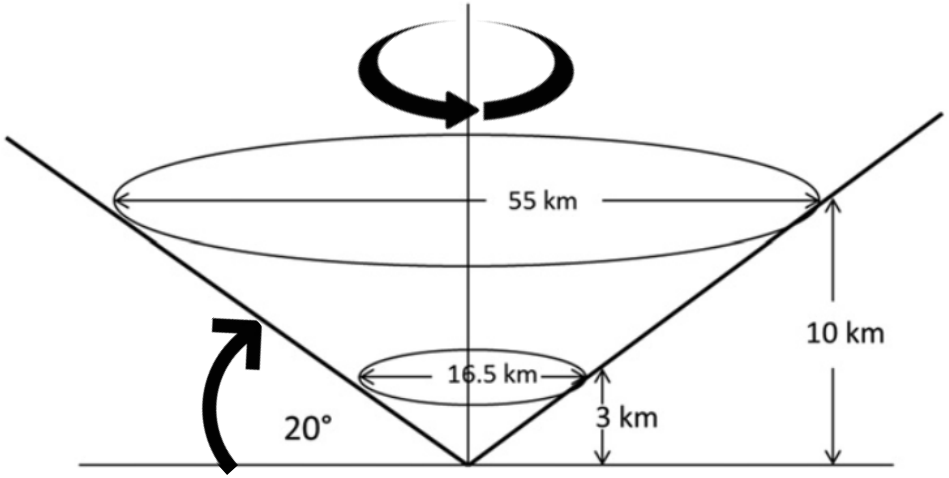


FIG. 2. Conical volume representing azimuthally averaged quasi-vertical profiles of radar variables.

Ryzhkov et al. (2016)

- KVNX (OK) radar data.
- Reproduce Quasi-Vertical Profile (Ryzhkov et al., 2016)
  - May 20, 2011. 10:00 – 12 :00 UTC

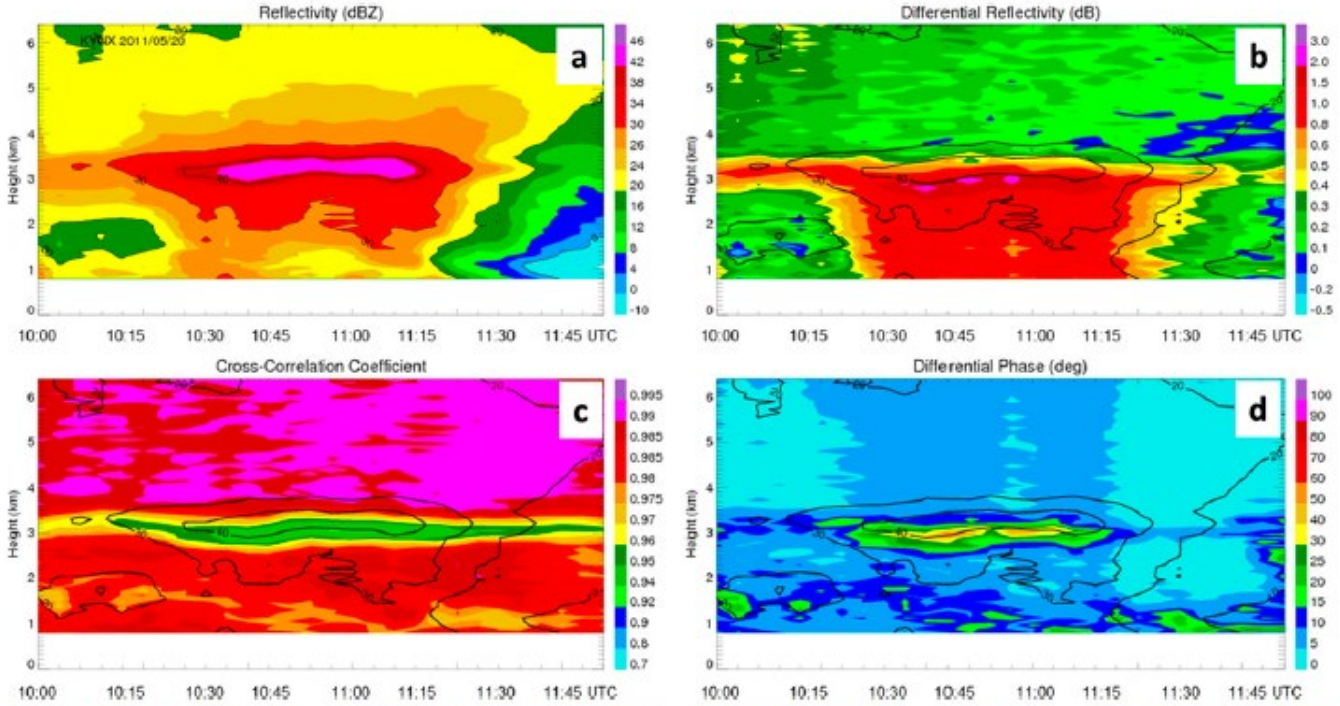
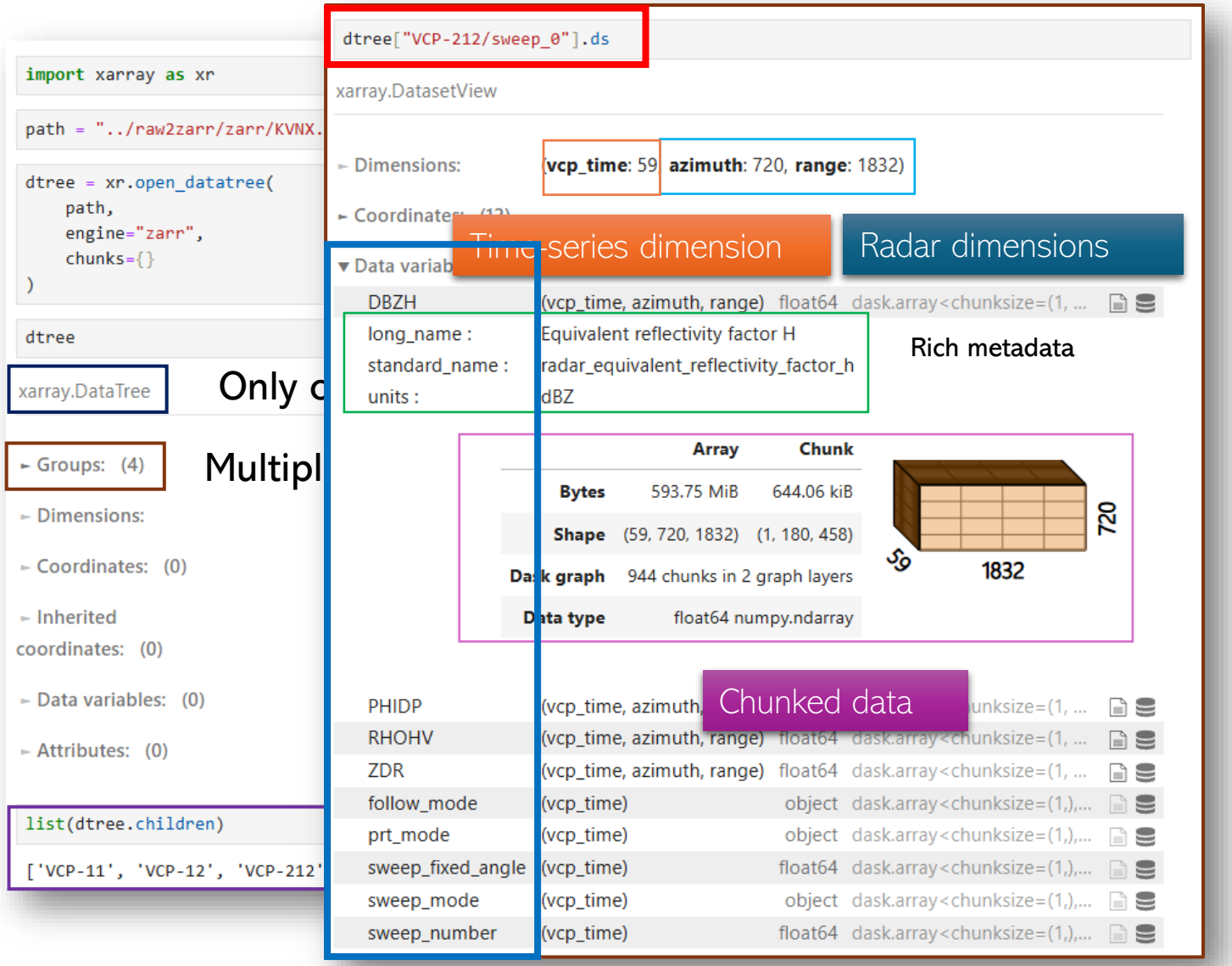


FIG. 4. The height-vs-time representation of quasi-vertical profiles of (a) Z, (b) Z_{DR}, (c) ρ_{hv}, and (d) Φ_{DP} retrieved from the KVNX WSR-88D radar data collected at elevation 19.5° in the case of an MCS observed in northern Oklahoma on 20 May 2011. Overlaid are contours of Z.

Ryzhkov et al. (2016)

Exploring the KVNX Radar DataTree



- ✓ Dataset
- ✓ Self-described
- ✓ Standard-compliant format
- ✓ Analysis-Ready
- ✓ Cloud-Optimized

From Minutes to Seconds: QVPs with Radar Datatree

ARCO-FAIR Radar DataTree model

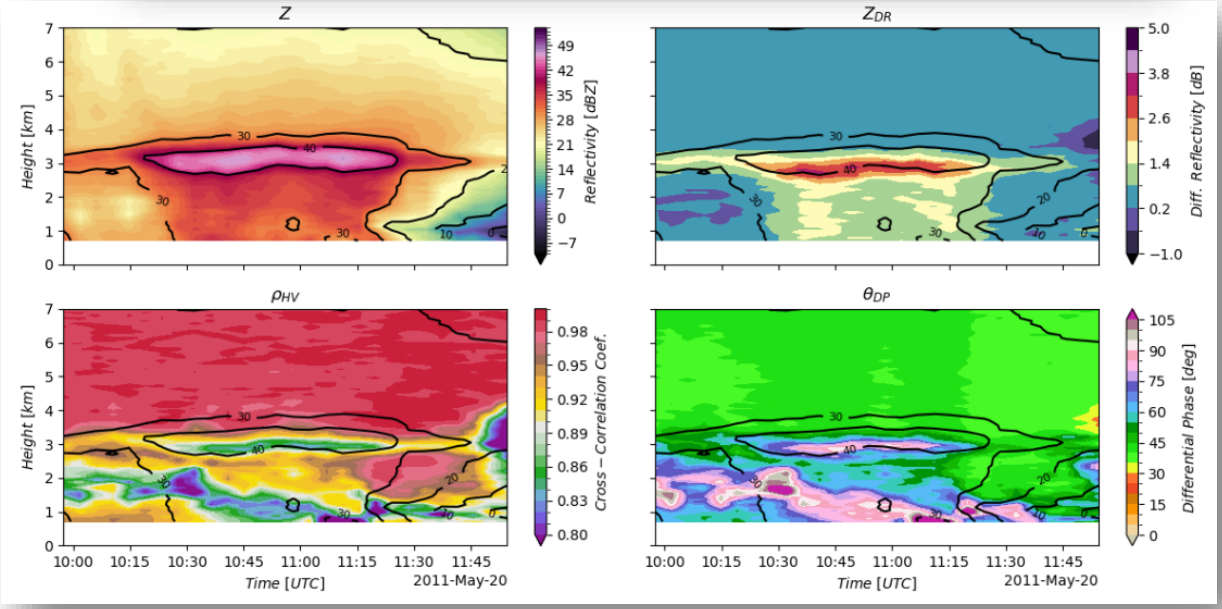
```
%%time
ds_qvp = dtree["/VCP-12/sweep_16"].ds.sel(vcp_time=slice("2011-05-20 09:45","2011-05-20 12:15"))
```

```
CPU times: user 823 µs, sys: 902 µs, total: 1.73 ms
Wall time: 1.73 ms
```

```
%%time
qvp_ref = compute_qvp(ds_qvp, var="DBZH").compute()
qvp_zdr = compute_qvp(ds_qvp, var="ZDR").compute()
qvp_rhohv = compute_qvp(ds_qvp, var="RHOHV").compute()
qvp_phidp = compute_qvp(ds_qvp, var="PHIDP").compute()
```

~1.6s

```
CPU times: user 955 ms, sys: 53.1 ms, total: 1.01 s
Wall time: 1.61 s
```



Download & Process model

```
%%time
ls_dataset = []
ls_time = []
for radar_file in radar_files:
    dtree = nexrad_donwload(radar_file)
    ls_time.append(pd.to_datetime(dtree.time_coverage_start.item()))
    ls_dataset.append(dtree["sweep_16"].to_dataset())
ds_trad_qvp = xr.concat(ls_dataset, dim="vcp_time")
ls_time = [ts.tz_convert(None) if ts.tzinfo else ts for ts in ls_time]
ds_trad_qvp = ds_trad_qvp.assign_coords(vcp_time=ls_time)
```

```
CPU times: user 4min 47s, sys: 59.9 s, total: 5min 47s
Wall time: 5min 38s
```

```
%%time
qvp_trad_ref = compute_qvp(ds_trad_qvp, var="DBZH")
qvp_trad_zdr = compute_qvp(ds_trad_qvp, var="ZDR")
qvp_trad_rhohv = compute_qvp(ds_trad_qvp, var="RHOHV")
qvp_trad_phidp = compute_qvp(ds_trad_qvp, var="PHIDP")
```

~5min 40s

```
CPU times: user 1.19 s, sys: 182 ms, total: 1.38 s
Wall time: 1.28 s
```

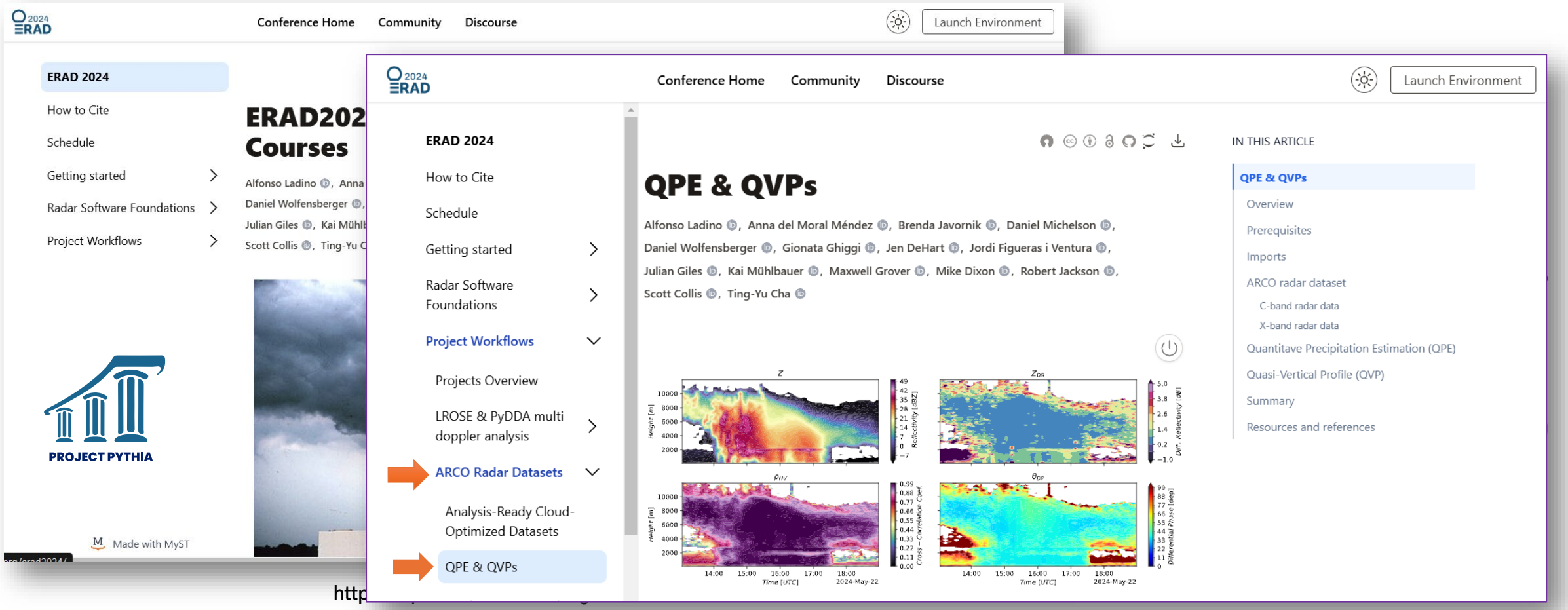
The ARCO-FAIR workflow **~210x** speedup over traditional download-process model

32GB RAM, Core i7 laptop with an SSD.





Empowering Open Radar Science: Lessons from ERAD 2024

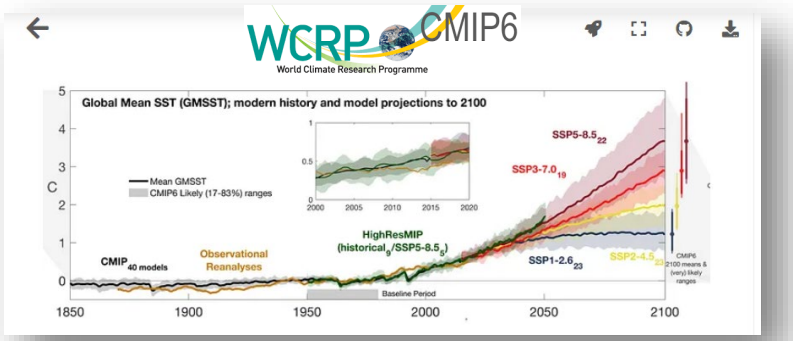


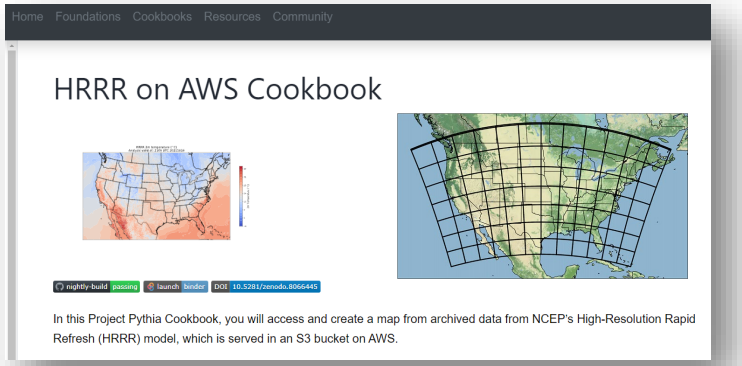
We empowered students to directly engage with the data, reproduce results, and participate in hands-on learning, fostering the next generation of meteorologists and data scientists

Takeaways

1. 🧱 **Traditional formats are not designed for modern, cloud-scale science.**
   - ➢ Even powerful HPC systems are limited when data is locked in monolithic files or scattered across millions of small ones.
2. 🧊 **Zarr enables scalable, parallel access to large multidimensional datasets.**
   - ➢ Its chunked, metadata-rich structure is cloud-native, easy to use, and interoperable with modern tools like Dask and Xarray.
3. 🚀 **The ARCO model turns raw data into usable, reusable infrastructure.**
   - ➢ Analysis-Ready, Cloud-Optimized datasets reduce wrangling time and enable fast, FAIR workflows at scale.
4. 🌎 **Real-world applications show the impact: radar data.**
   - ➢ From Colombia to NEXRAD, ARCO datasets built with Zarr are transforming how we analyze and share big scientific data.
5. 🔓 **The future of scientific computing is open, scalable, and data-first.**
   - ➢ Zarr and ARCO are not just tools — they represent a shift in how science works across HPC and cloud environments.

Zarr Adoption Across Science and Industry





(carbon)plan

→ Zarr is used by CarbonPlan as a storage format for analysis and visualization of climate data.

hhmi | janelia Research Campus

→ Zarr is used extensively within Janelia Research Campus for efficiently storing and accessing large imaging datasets

NCAR
NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

NASA

Google Research

Microsoft

→ Zarr is used by the Microsoft Planetary Computer as a cloud-native storage format for chunked, N-dimensional arrays of geospatial data.

OME

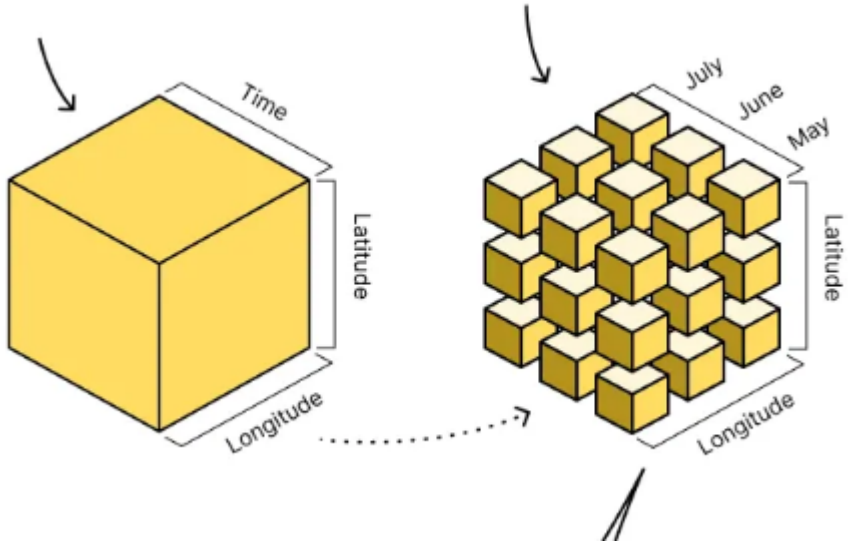
esa

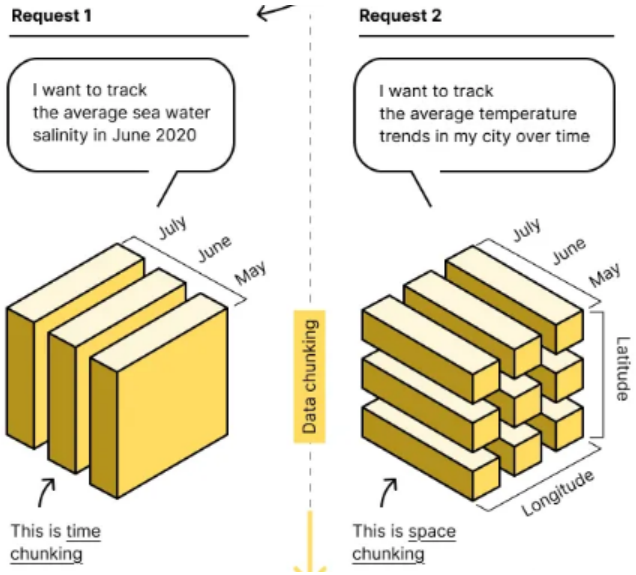
unidata

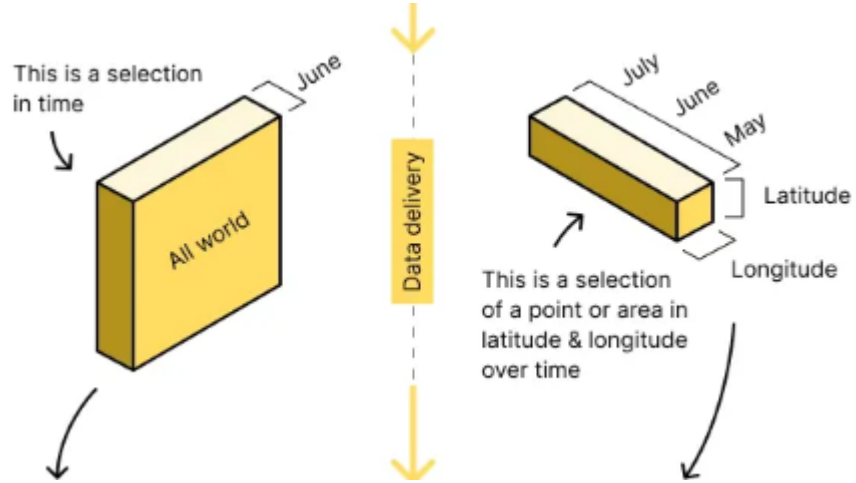
[Bonus Slide]

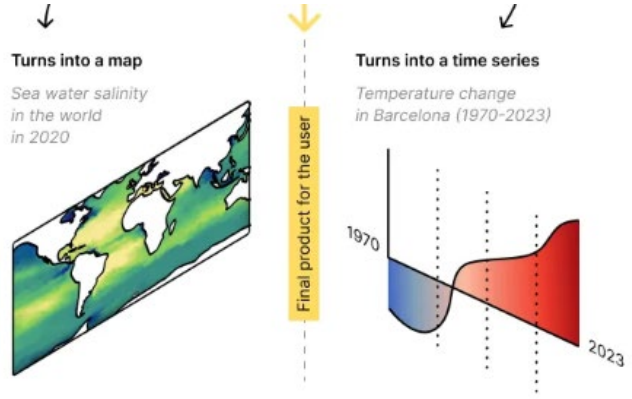
Questions?

alfonso8@illinois.edu









https://blog.lobelia.earth/arco-the-smartest-way-to-access-big-geospatial-data-eaf689eff3c9