

XRADAR.READTHEDOCS.IO



XRADAR: SHARED IO IN THE OPEN RADAR STACK

MAX GROVER

¹Argonne National Laboratory
@mgrover1

Scott Collis¹, Zachary Sherman¹, Robert Jackson¹, Joseph O'Brien¹, Daniel Michelson²

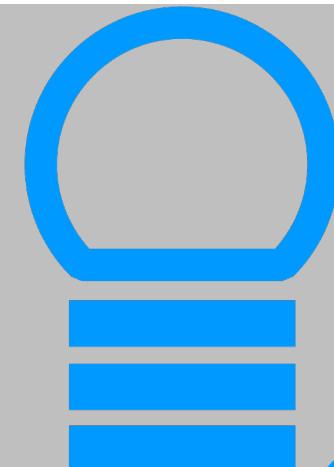
²Environment and Climate Change Canada, WMO Task Team

KAI MÜHLBAUER

University of Bonn Germany
@kmuehlbauer

ALFONSO LADINO

University of Illinois Urbana-Champaign
@aladinor



LET'S TALK ABOUT WEATHER RADAR DATA IN PYTHON – PRE-XRADAR...



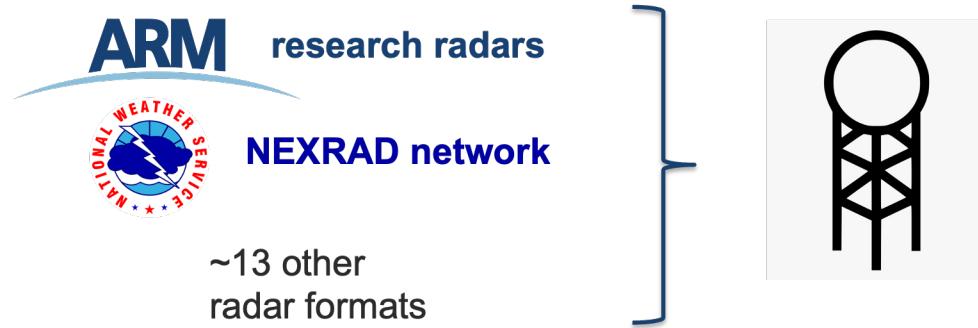
Argonne National Laboratory is a
U.S. Department of Energy laboratory
managed by UChicago Argonne, LLC.



THE PYTHON ARM RADAR TOOLKIT (PY-ART)

Building around the Radar Object

- ▶ Initially created by Scott Collis and Jonathan Helmus (2013), funded by ARM, built to work with weather radar data
- ▶ Over 340 citations of the package (peer-reviewed publications), 1000 monthly users
- ▶ >600,000 package downloads
- ▶ Recent development with array integration, moving to a standard IO library maintained by the broader community

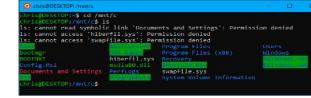


What about other packages??

corrections
retrievals
gridding
graph (viz)
pointers to other libraries

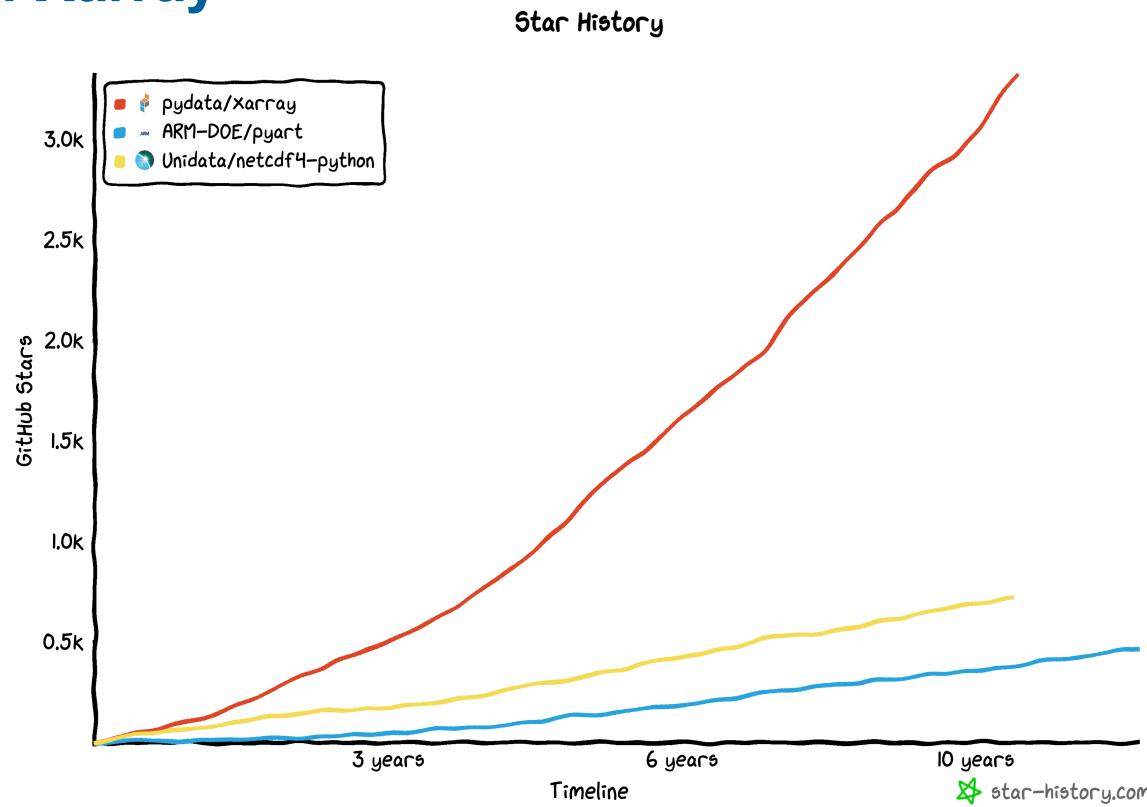
THE OPEN RADAR STACK

On the Road to Interoperability

Data Models	PyART Radar (based on CfRadial 1.4) BALTRAD objects (based on ODIM_H5)	wradlib 
N-D Arrays		
Processing Mode	Interactive 	Batch 
Compute Platform	HPC 	Cloud 
Foundation + Language	 Py-ART, wradlib, PyRAD	 LROSE, BALTRAD

QUESTION: WHY DIDN'T YOU JUST USE XARRAY AS THE DATA MODEL TO START?

The Growth of Xarray



WAIT... 2017???

Xarray could have some traction... (wradlib in 2019...)

Add some support for X-Array in a staged manner. #670

Open 3 of 5 tasks scollis opened this issue on Jul 13, 2017 · 5 comments

scollis commented on Jul 13, 2017 · edited by mgrover1 · Member ...

Creating a Checklist so we can keep track of this:

- Add an xradar branch to develop/test on
- Add helper functions to convert to an xradar object ([↳ Add xradar basic functions #1251](#))
- Move aux_io readers to xradar
- Implement a `radar.to_xradar()` method
- Add a separate xradar object, similar to the radar object, that we can feed metadata + values directly into

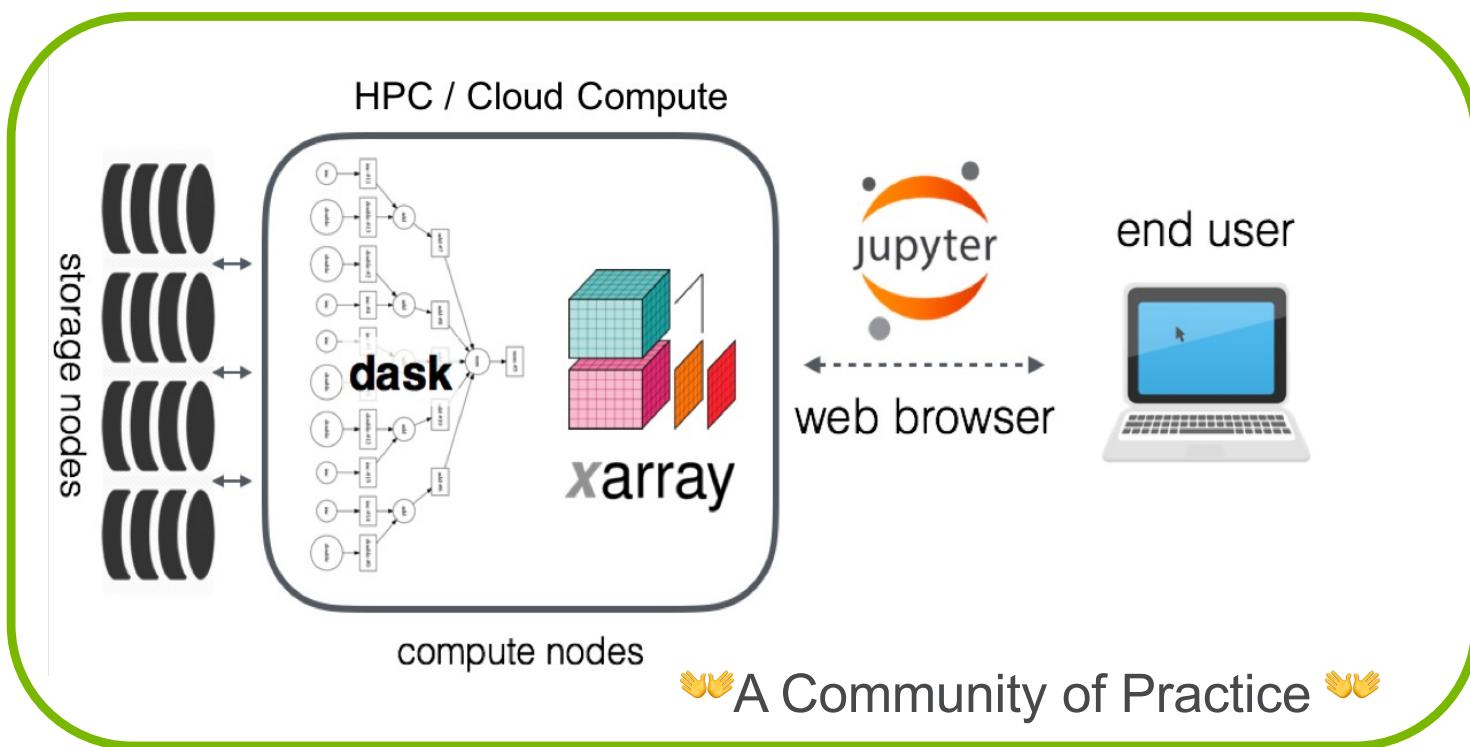
THE GROWTH OF XARRAY AS A DATA MODEL: THE POWER OF BACKENDS AND COMMUNITY



Argonne National Laboratory is a
U.S. Department of Energy laboratory
managed by UChicago Argonne, LLC.

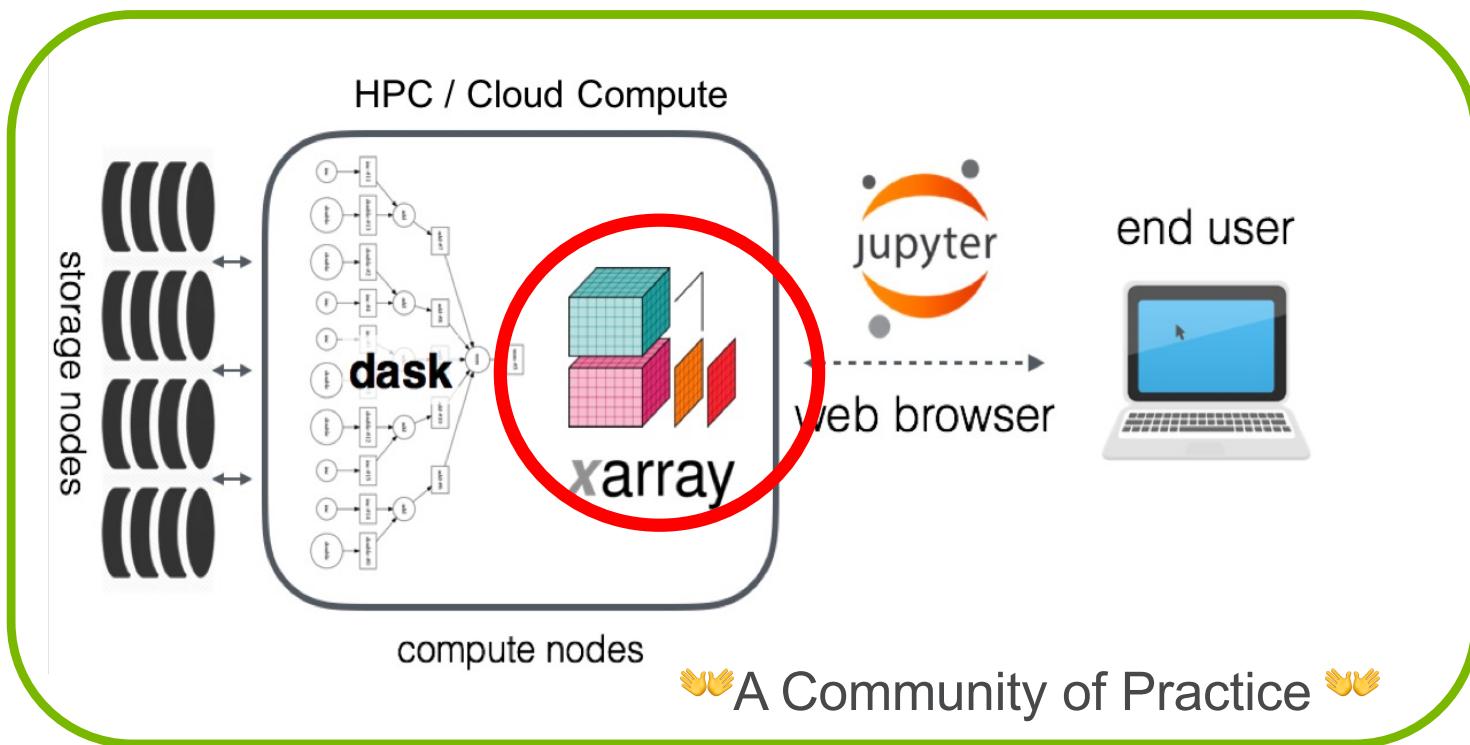
THE TYPICAL PANGEO WORKFLOW

Scaling Climate Data Science



THE TYPICAL PANGEO WORKFLOW

What about the weather radar community...



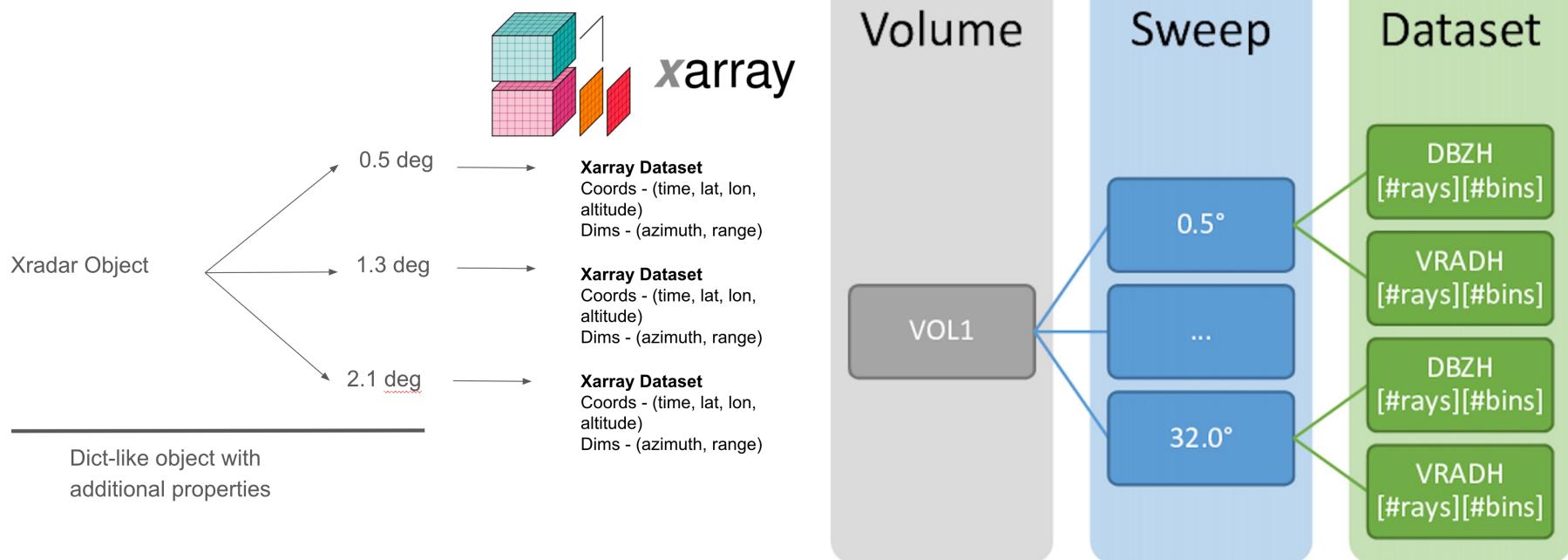
👉 A Community of Practice 👈

9

👀 the open radar community?

THE IDEA: A SHARED PACKAGE FOR IO

The Creation of Xradar



HERE COMES DATATREE

`mamba install xarray-datatree*`

Datatree is a prototype implementation of a tree-like hierarchical data structure for `xarray`.

Datatree was born after the `xarray` team recognised a need for a new hierarchical data structure, that was more flexible than a single `xarray.Dataset` object. The initial motivation was to represent netCDF files / Zarr stores with multiple nested groups in a single in-memory object, but `datatree.DataTree` objects have many other uses.

STEP 1: BACKENDS.

What is a backend: spoiler alert – you use one when you use xarray.



```
import xarray as xr  
  
ds = xr.open_dataset("file.nc",  
                     engine="netcdf4")
```

~~STEP 1~~: BACKENDS.

What is a backend: spoiler alert – you use one when you use xarray.



```
import xarray as xr  
  
ds = xr.open_dataset("file.nc",  
                     engine="netcdf4")
```

STEP 0: BACKENDS.

What is a backend: spoiler alert – you use one when you use xarray.



```
import xarray as xr  
  
ds = xr.open_dataset("file.nc",  
                     engine="netcdf4")
```

STEP 1: COMMON DATA STRUCTURES

The Use of FM301/Cfradial2

Data Model

With the forthcoming standard [FM301](#), which is a subset of [CfRadial2.0](#), as a basis for the xradar data model we can take full advantage of [Xarray](#) and the whole software stack.

We facilitate [datatree.DataTree](#) to bundle the different sweeps of a radar volume into one structure. These sweep datasets are essentially [xarray.Dataset](#) which contain metadata attributes and variables ([xarray.DataArray](#)).

DataTree

The DataTree consists of one global root group and several sweep groups. Optionally, groups containing parameter and calibration can be part of the datatree.

Internal representation: [datatree.DataTree](#)

Supported Backends

- ✓ Cfradial1
- ✓ IRIS/Sigmet
- ✓ Rainbow
- ✓ ODIM_H5
- ✓ Gamic
- ✓ Foruno

→ SOON NEXRAD Level2

► NEXRAD Level3

STEP 2: PROVIDE SOME BASIC FUNCTIONALITY

Plotting Our Data on a Map



```
# Georeference!
radar.xd.georeference(
)
```

► Dimensions: (azimuth: 483, range: 996)

▼ Coordinates:

time	(azimuth)	datetime64[ns] ...		
range	(range)	float32 150.0 300.0 ... 1.492e+05 1.494e+05		
azimuth	(azimuth)	float32 0.0 0.75 1.5 ... 357.8 358.5 359.2		
elevation	(azimuth)	float32 1.104 1.104 1.104 ... 1.104 1.104		
latitude	()	float64 22.53		
longitude	()	float64 120.4		
altitude	()	float64 45.0		
crs_wkt	()	int64 0		
x	(azimuth, range)	float32 0.0 0.0 ... -1.952e+03 -1.954e+03		
y	(azimuth, range)	float32 150.0 299.9 ... 1.491e+05 1.493e+05		
z	(azimuth, range)	float32 47.0 50.0 ... 4.23e+03 4.235e+03		

STEP 3: CONNECT OTHER TOOLKITS

Bringing in the Radar Community



```
# Rain Rate Z(R) Relationship!
radar.wrl.zr.z_to_r(a=200,
b=1.6)
```

```
xarray.DataArray 'RATE' (azimuth: 483, range: 996)
```

```
array([[0.18531333, 0.29432994, 0.31736422, ...,
       nan],
      [0.23131426, 0.31217656, 0.34820113, ...,
       nan],
      [nan, 0.30476677, 0.28758729, ...,
       0.04085606],
      ...,
      [0.00689864, 0.26638007, 0.29809725, ...,
       nan],
      [nan, 0.30195013, 0.31002867, ...,
       nan],
      [0.1167989 , 0.31103662, 0.3222496 , ...,
       nan]], dtype=float32)
```

► Coordinates: (7)

► Indexes: (2)

▼ Attributes:

```
standard_name : rainfall_rate
long_name : rainfall_rate
short_name : RATE
units : mm h-1
```

STEP 4: CONVERT TO ZARR?

A Cloud Optimized Radar Data Format? Easy as `.to_zarr()`



```
# Save to Zarr!
radar.to_zarr("radar.zarr"
)
```

WHAT'S NEXT?

Charting the Road to Open Radar Stack 2.0... including Py-ART

- New Readers
 - NEXRAD (finish up Level2, Level3)
 - CSU CHILL radar
- More Accessors (especially Py-ART)
 - KDP processing, dealiasing, etc.
- Further exploration into a cloud-optimized version of...
 - NEXRAD Level2/Level3 data?
 - Join the discussion on openradar.discourse.group

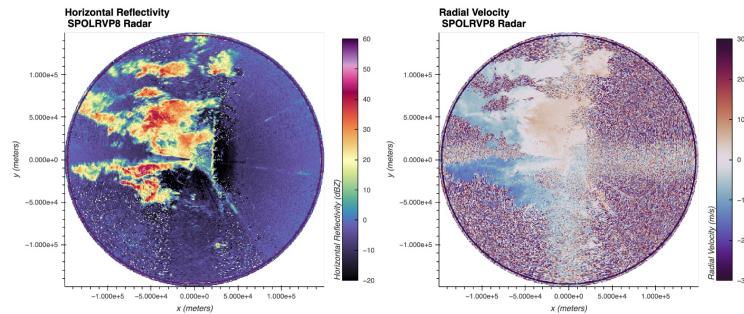


Give it a try!! 

This presentation has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory ("Argonne"). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Biological and Environmental Research's ARM User Facility.

ALSO... CHECK OUT OUR COOKBOOK

With Project Pythia!



Interactive Radar Visualization

Overview

Within this cookbook, we will detail how to create interactive plots of radar data!

1. Reading data with Xradar
2. Creating your first interactive figure with Xradar + hvPlot
3. Combining your plots into a single dashboard
4. Filtering and Checking Data Quality
5. Create a Dashboard to Analyze ZDR Bias

