ESDIS
EARTH SCIENCE DATA AND INFORMATION SYSTEM

VirtualiZarr and DMR++

Summer ESIP
22–26 July 2024 Ashville, NC

Ayush Nag[1] *ayushn1@uw.edu*
James Gallagher[2] jgallagher@opendap.org

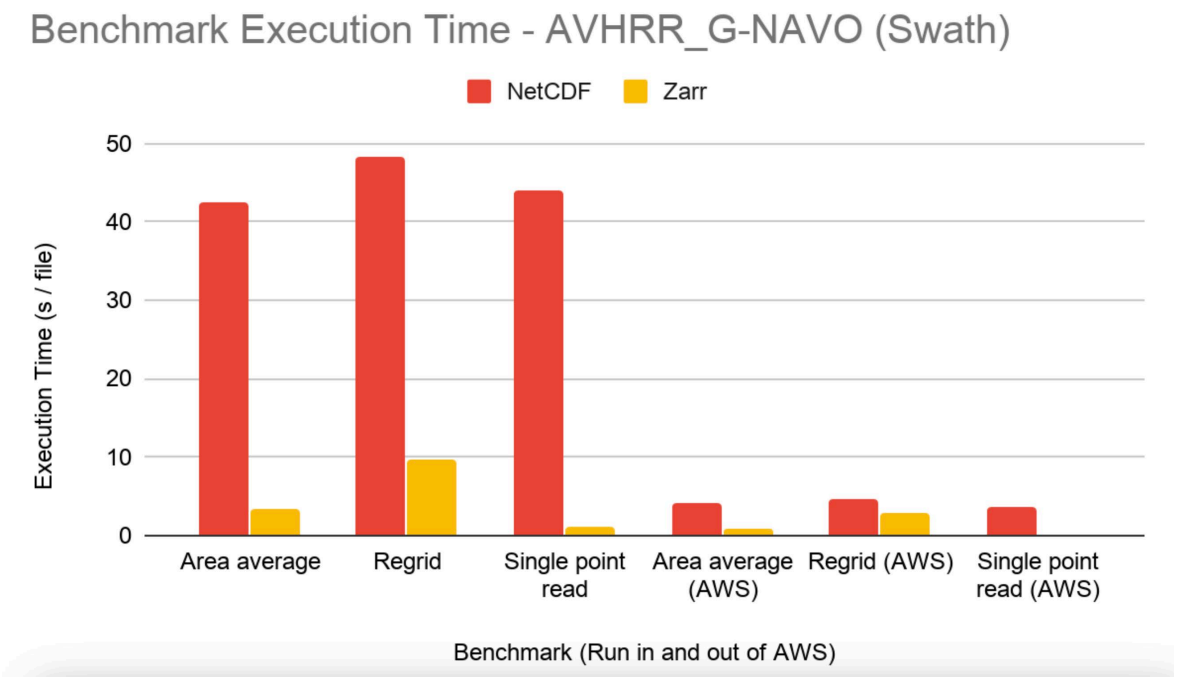
NASA

Agenda

- The problem
- Solutions
- Future directions

The Problem

- SWOT 21 days = ~620 granules
  - SWOT launched Dec 2022 and **SWOT L2 SSH 2.0** has 48,860 granules already
- Reading and concatenating thousands of netCDF files (even on AWS) is **inefficient**
- Exploration, analysis, and visualization are all bottlenecked, since data loading is the first step



Credit: NTRS Task 51 - Cloud-Optimized Format Study

ESDIS
EARTH SCIENCE DATA AND INFORMATION SYSTEM

Solution Using Kerchunk or Zarr

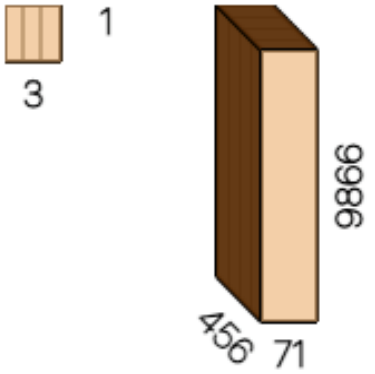
Kerchunk

- Granule metadata (chunk size, byte offsets) stored in separate lightweight ”sidecar” JSON file
- Works alongside existing netCDF remote store

xarray.DataArray 'simulated_true_ssh_karin'

(**cycle_num**: 3, **pass_num**: 456, num_lines: 9866, num_pixels: 71)

	Array	Chunk
Bytes	7.14 GiB	5.34 MiB
Shape	(3, 456, 9866, 71)	(1, 1, 9866, 71)
Count	1369 Tasks	1368 Chunks
Type	float64	numpy.ndarray

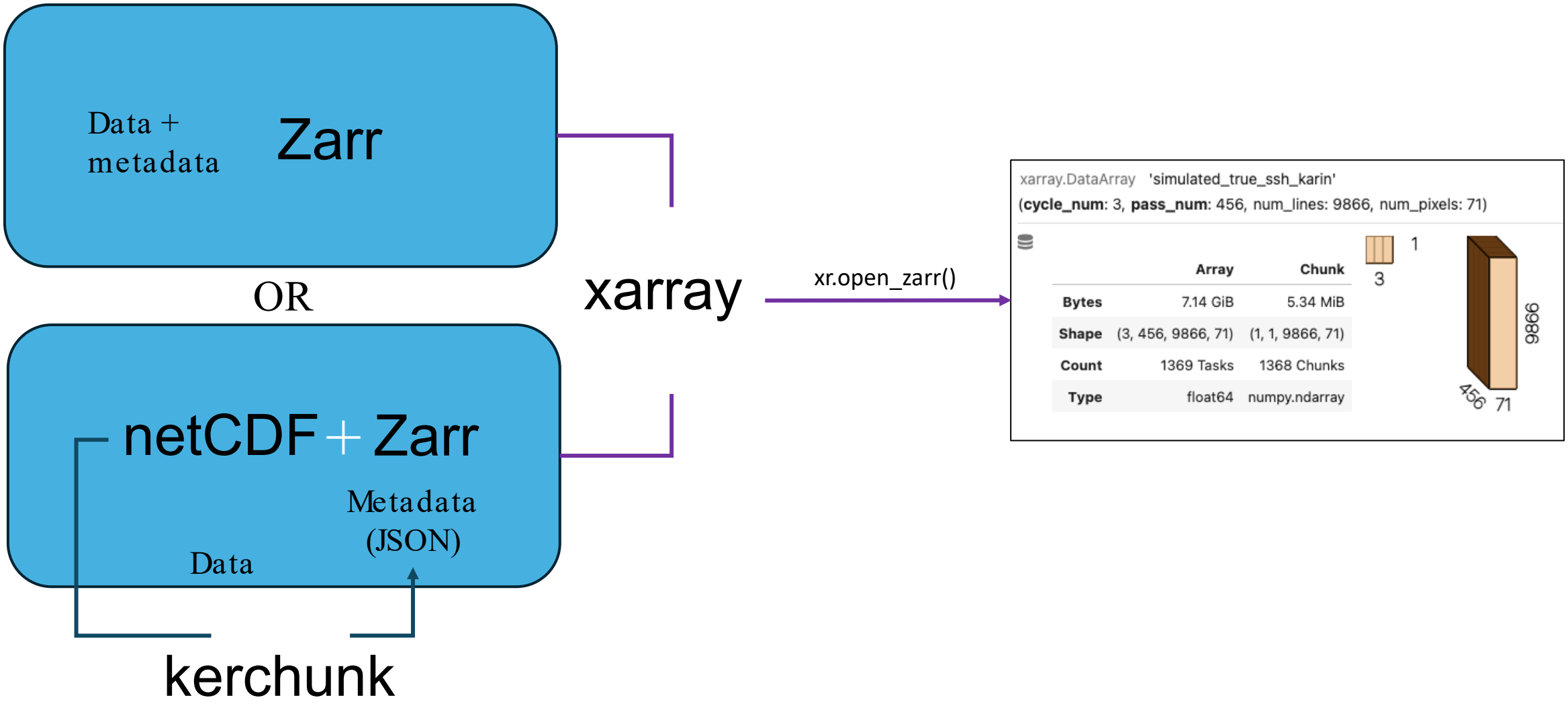


Zarr

- Create N-dimensional arrays with Python numpy
- Chunk arrays along any dimension.
- Store arrays in memory, on disk, inside a Zip file, on S3, …
- However, data must be duplicated into .zarr file

ESDIS
EARTH SCIENCE DATA AND INFORMATION SYSTEM

Solution (visualized)



ESDIS
EARTH SCIENCE DATA AND INFORMATION SYSTEM

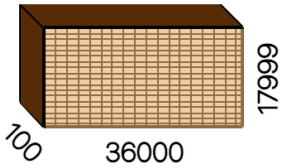
Solution Using DMR++

DMR++

- Granule metadata (chunk size, byte offsets) stored in separate lightweight ”sidecar” DMR++ file
- Works alongside existing netCDF/HDF5 remote store in S3, for example
- Using earthaccess, VirtualiZarr can parse DMR++ and build an equivalent Zarr metadata object
- Xarray can use that to read / subset /aggregate the data
- Using multiple DMR++ documents, aggregations can be formed on-the-fly

xarray.DataArray 'analysed_sst' (**time**: 100, **lat**: 17999, **lon**: 36000)

	Array	Chunk
Bytes	241.39 GiB	7.99 MiB
Shape	(100, 17999, 36000)	(1, 1023, 2047)
Dask graph	32400 chunks in 2 graph layers	
Data type	float32 numpy.ndarray	



▾ Coordinates:

lat	(lat)	float32	-89.99 -89.98 ... 89.98 89.99
lon	(lon)	float32	-180.0 -180.0 ... 180.0 180.0
time	(time)	datetime64[ns]	2017-12-13T09:00:00 ... 2018-03-...

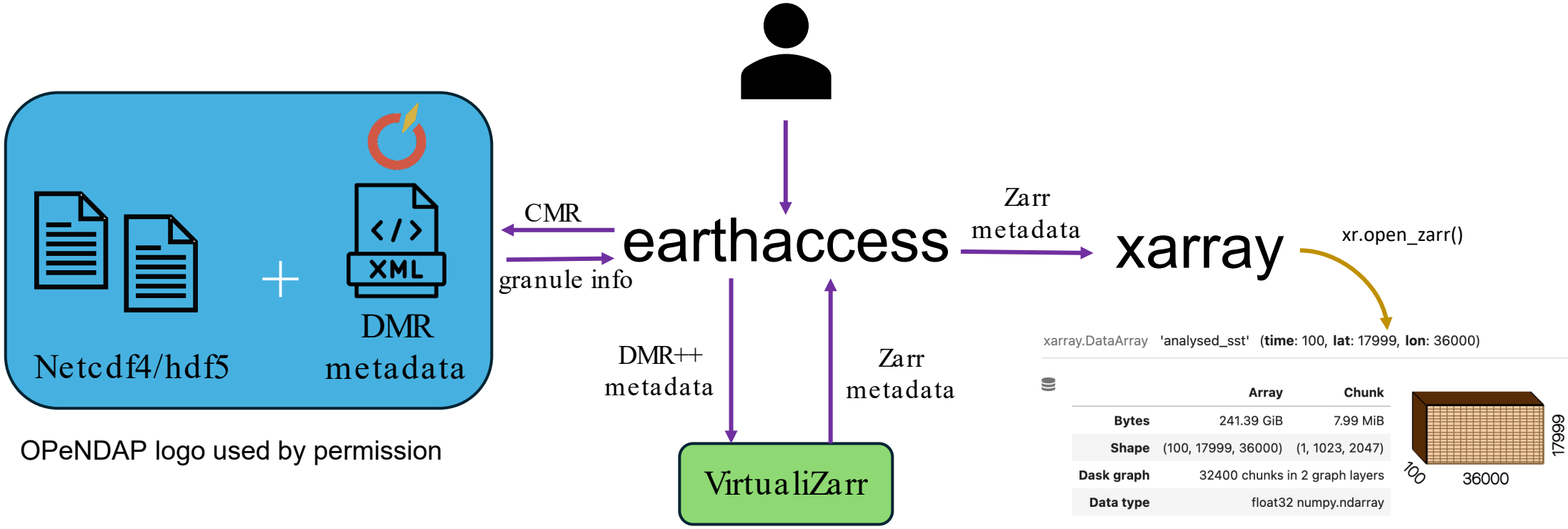
▸ Indexes: (3)

▾ Attributes:

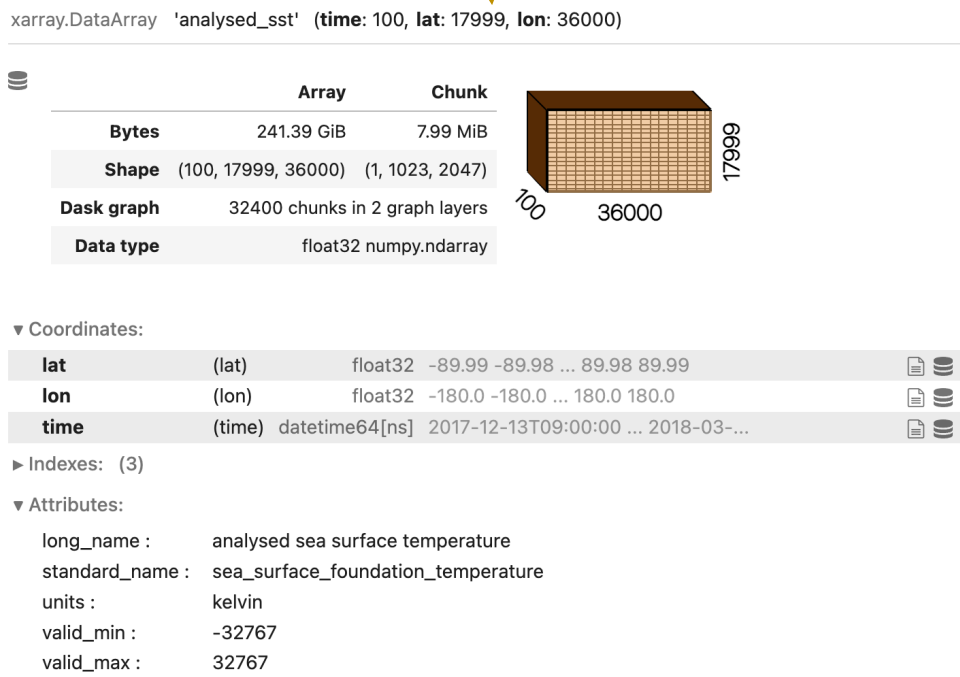
long_name :	analysed sea surface temperature
standard_name :	sea_surface_foundation_temperature
units :	kelvin
valid_min :	-32767
valid_max :	32767

ESDIS
EARTH SCIENCE DATA AND INFORMATION SYSTEM

Solution (visualized)



OPeNDAP logo used by permission



ESDIS
EARTH SCIENCE DATA AND INFORMATION SYSTEM

Solution Implemented

- Query data, load, analyze
  - *earthaccess* uses CMR to find data quickly
- Uses *dask* for parallel AWS reads of DMR++ files
- So far this is a working proof of concept
- Final API is still under construction
  - Support for opening single datasets AND multi-file datasets (concatenation)
  - Support for netcdf4 and hdf5 groups
  - Save concatenated dataset “manifest” to pass to *xarray* directly

[3]:
```
%%time
results = earthaccess.search_data(
    count=100,
    temporal=("2017-12-13", "2023-12-13"),
    short_name="MUR-JPL-L4-GLOB-v4.1"
)
```

```
CPU times: user 108 ms, sys: 18.8 ms, total: 127 ms
Wall time: 1.7 s
```

[4]:
```
%%time
from earthaccess.virtualizarr import open_virtual_dataset
vds = open_virtual_dataset(results, access="direct", concat_dim="time", coords='minimal', compat='override', combine_attrs="drop_conflicts")
vds
```

```
CPU times: user 9.45 s, sys: 783 ms, total: 10.2 s
Wall time: 18.7 s
```

[4]: xarray.Dataset

▸ Dimensions: (**time**: 100, **lat**: 17999, **lon**: 36000)

▾ Coordinates:

lat	(lat)	float32	-89.99 -89.98 ... 89.98 89.99	
lon	(lon)	float32	-180.0 -180.0 ... 180.0 180.0	
time	(time)	datetime64[ns]	2017-12-13T09:00:00 ... 2018-03-...	

▾ Data variables:

analysed_sst	(time, lat, lon)	float32	dask.array<chunksize=(1, 1023, 2047), me...	
analysis_error	(time, lat, lon)	float32	dask.array<chunksize=(1, 1023, 2047), me...	
dt_1km_data	(time, lat, lon)	timedelta64[ns]	dask.array<chunksize=(1, 1447, 2895), me...	
mask	(time, lat, lon)	float32	dask.array<chunksize=(1, 1447, 2895), me...	
sea_ice_fraction	(time, lat, lon)	float32	dask.array<chunksize=(1, 1447, 2895), me...	

▸ Indexes: (3)

▸ Attributes: (0)

[6]:
```
print(f"{vds.nbytes / 1e12} TB")
```

```
1.555113816796 TB
```

ESDIS
EARTH SCIENCE DATA AND INFORMATION SYSTEM

Benefits

- Join a dataset, create a concatenated “view” of the dataset and save that metadata using VirtualiZarr
- Data query engine (*earthaccess*) and data loader are connected → integrated search to access.
- Getting s3 credentials to the DAACs is handled by *earthaccess*

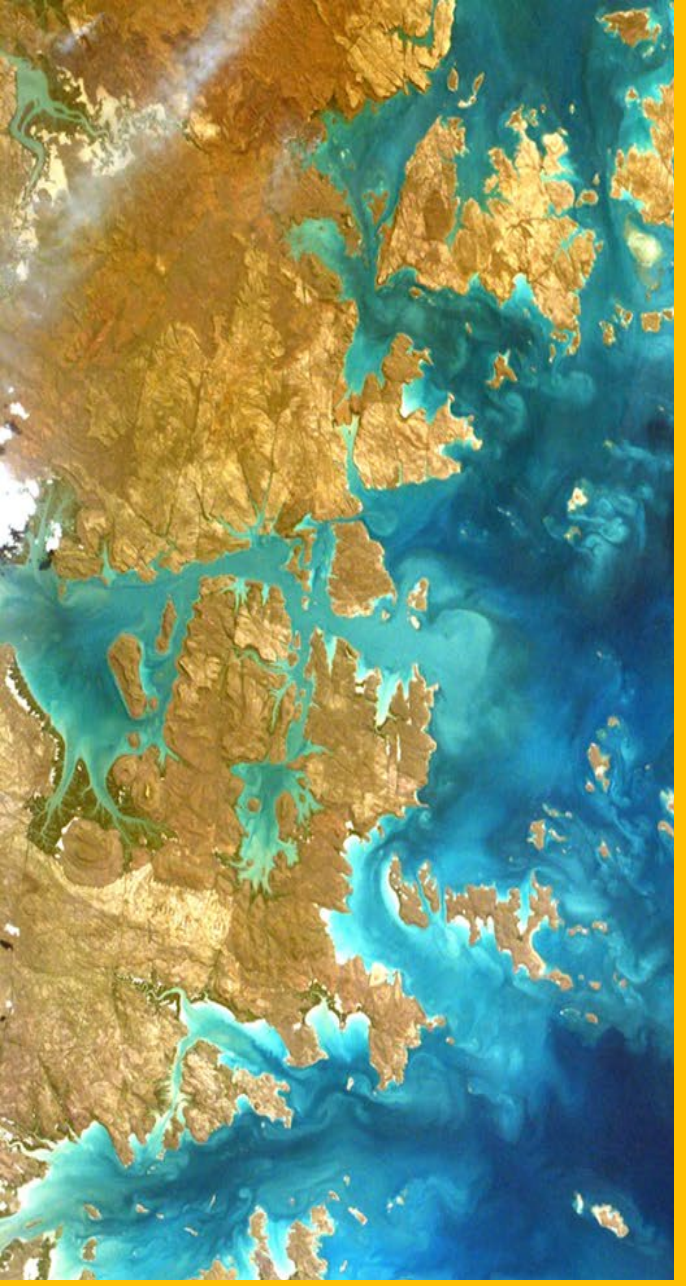
Also

- Current *earthaccess* approach uses *dask* so parallel AWS reads
- DMR++ parser is very fast: only milliseconds of overhead per dataset
- On-the-fly zarr reference creation (translates DMR++)

ESDIS
EARTH SCIENCE DATA AND INFORMATION SYSTEM

Future Steps

- Streamline access to the DMR++
  - Currently it's hard to access
  - Will benefit both VirtualiZarr and ESDIS's internal DMR++ users (Harmony, Giovani, AppEEARS)
- Decompose the DMR++ into pieces
  - Enable selective transfer of DMR++ document's information to improve performance
  - Other performance improvements for support of HDF4 and HDF4-EOS2
- VirtualiZarr → Zarr Specification
  - Any zarr reader (python, JS, C, etc.) will be able to read virtual zarr files



This work was supported by NASA/GSFC under Raytheon Company contract number 80GSFC21CA001

ESDIS
EARTH SCIENCE DATA AND INFORMATION SYSTEM

NASA