National Aeronautics and Space Administration

EARTHDATA

Accelerating Science Using Virtualized Data at PO.DAAC

08/21/2025

Dean Henze[1p], Edward M. Armstrong[1p], Mike Gangl[1], Celia Ou[1]

[1]*Jet Propulsion Laboratory, California Institute of Technology*

The Physical Oceanography Distributed Active Archive Center (PO.DAAC)

[p] *Presenting Authors*

About PO.DAAC

Physical Oceanography Distributed Active Archive Center (PO.DAAC)

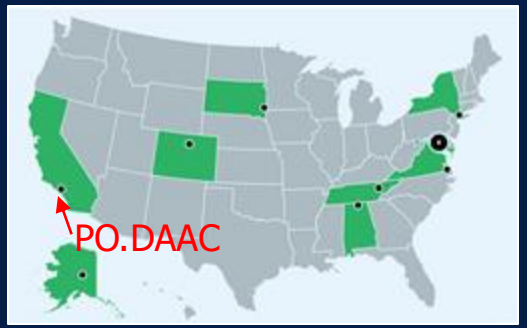
NASA Archive for Physical Oceanography & Terrestrial Hydrosphere Data

Data publication, access, & user support

Our Measurements / Model Output

Glaciers / Ice Sheets ▪ Ocean Circulation ▪ Gravity / Gravitational Field ▪ Ocean Surface Topography ▪ Ocean Temperature ▪ Ocean Heat Budget ▪ Ocean Winds ▪ Salinity / Density ▪ Sea Ice ▪ Surface Water ▪ Ocean Waves

NASA DAACs



Our Missions

ADEOS-II ▪ AQUA ▪ AQUARIUS/SAC-D ▪ COWVR-TEMPEST ▪ CYGNSS ▪ ECCO ▪ GEOS-3 ▪ GHRSST ▪ GRACE ▪ GRACE-FO ▪ ISS-RAPIDSCAT ▪ JASON 1 ▪ JASON 3 ▪ LOCSS ▪ MEASURES-CCMP ▪ MEASURES-MUR ▪ MEASURES-PRE-SWOT ▪ MEASURES-SSH ▪ NSCAT ▪ OMG ▪ OPERA ▪ OSTM-JASON 2 ▪ QUIKSCAT ▪ S-MODE ▪ S-NPP ▪ SAILDRONE ▪ SEASAT ▪ SENTINEL-6 ▪ SMAP ▪ SPURS ▪ SWOT ▪ TERRA ▪ TOPEX-POSEIDON

EARTHDATA

Agenda

Motivation and Introduction to Virtual Datasets

(Dean Henze, 10 minutes)

Basic usage, Cookbook Resources, Benchmarking

(Dean Henze, 15 minutes)

Science Use Case Example

(Edward Armstrong, 15 minutes)

Scope of Usability, Resources, Future developments

(Dean Henze, 10 minutes)

Python Packages to Know

This presentation covers the use of virtual datasets programmatically in Python.

Xarray

- Open and explore netCDF files
- Subsetting capabilities, fundamental scientific computations (statistics, rolling operations, groupby-apply functions, matrix computations).
- Built-in functions work naturally with Dask/Parallel-computing.

earthaccess (NASA Openscapes initiated project) handles Earthdata Login credentials, locates dataset metadata, data file endpoints.

NumPy Fundamental package for math (Python's version of MATLAB's array objects).

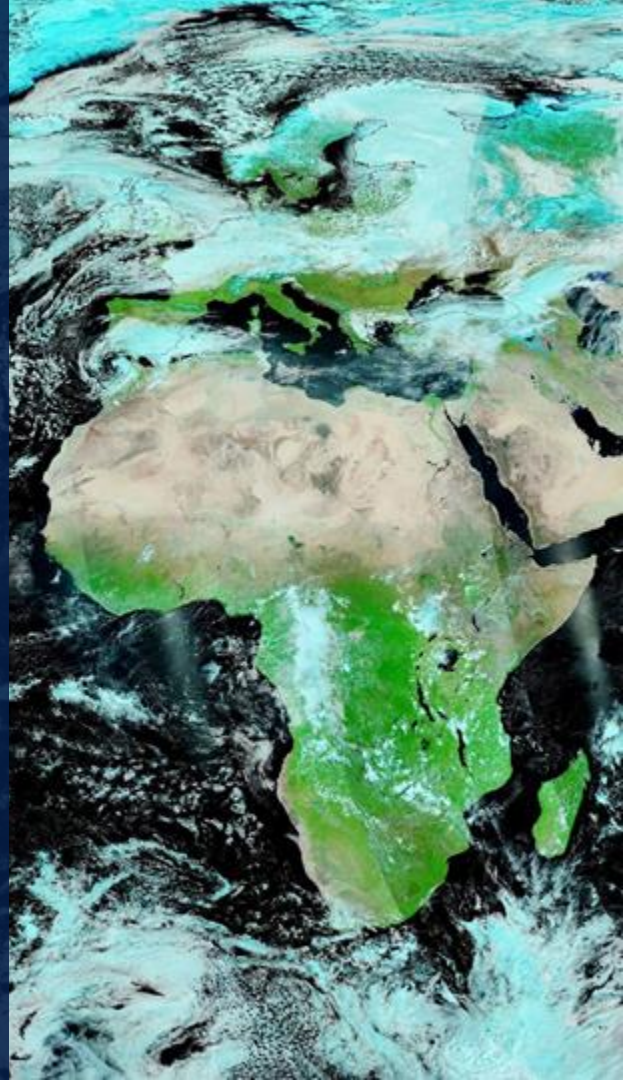
Dask The analog of NumPy but for larger-than-memory arrays and parallel computing.

EARTHDATA

Motivation and Introduction to Virtual Datasets

What is a virtual dataset and how does it simplify/optimize data access?

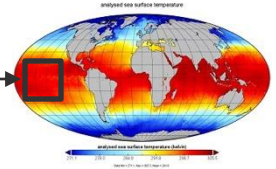
(Dean Henze, 10 minutes)



GHRSST Level 4 OSTIA Global Historical Reprocessed Foundation Sea Surface Temperature Analysis

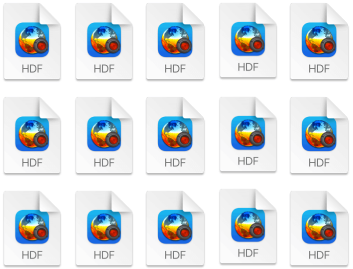
https://doi.org/10.5067/GHOST-4RM02

region of interest



15,340 files, ~ 11TBs

```
for f in files:
    d = open_datafile()
    ...
    ...
```



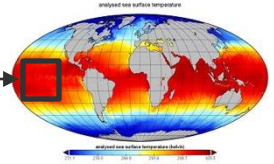
Motivation

- We often have to access data one file at a time.
- This necessitates knowing the filenames, structure, locations, and writing more lines of code to wrangle.
- It could also mean potentially having to downloads 100's of GB's to TB's of data if working programmatically without tools or services.

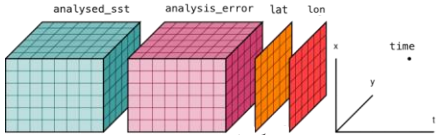
GHRSST Level 4 OSTIA Global Historical Reprocessed Foundation Sea Surface Temperature Analysis

https://doi.org/10.5067/GHOST-4RM02

region of interest



15,340 files, ~ 11TBs




```
Dimensions:           (time: 15339, lat: 3600, lon: 7200)
Coordinates:
  * lat               (lat) float32 -89.97 -89.93 -89.88 ... 89.88 89.93 89.97
  * lon               (lon) float32 -180.0 -179.9 -179.9 ... 179.9 179.9 180.0
  * time              (time) datetime64[ns] 1982-01-01T12:00:00 ... 2023-12-3...
Data variables:
    analysed_sst      (time, lat, lon) float32 dask.array<chunksize=(1, 1200, 2400), meta=np.ndarray>
    analysis_error    (time, lat, lon) float32 dask.array<chunksize=(1, 1200, 2400), meta=np.ndarray>
    mask              (time, lat, lon) float32 dask.array<chunksize=(1, 1800, 3600), meta=np.ndarray>
    sea_ice_fraction  (time, lat, lon) float32 dask.array<chunksize=(1, 1800, 3600), meta=np.ndarray>
```

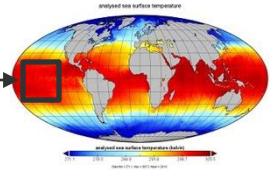
Motivation

The dream is to open the entire dataset in a single line of code, pre-organized by it's dimensions / coordinates...

GHRSST Level 4 OSTIA Global Historical Reprocessed Foundation Sea Surface Temperature Analysis

https://doi.org/10.5067/GHOST-4RM02

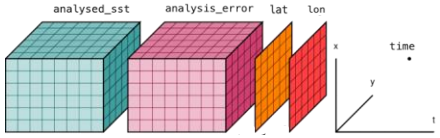
region of interest



15,340 files, ~ 11TBs

Good luck!

ds = xr.open_mfdataset(OSTIA_files)



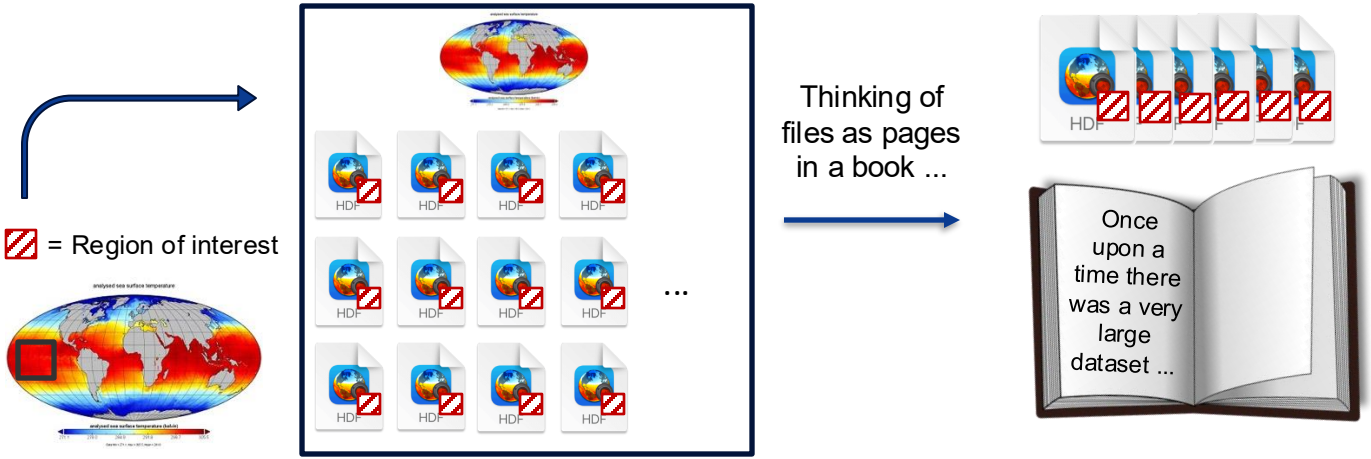

```
Dimensions:           (time: 15339, lat: 3600, lon: 7200)
Coordinates:
  * lat               (lat) float32 -89.97 -89.93 -89.88 ... 89.88 89.93 89.97
  * lon               (lon) float32 -180.0 -179.9 -179.9 ... 179.9 179.9 180.0
  * time              (time) datetime64[ns] 1982-01-01T12:00:00 ... 2023-12-3...
Data variables:
    analysed_sst      (time, lat, lon) float32 dask.array<chunksize=(1, 1200, 2400), meta=np.ndarray>
    analysis_error    (time, lat, lon) float32 dask.array<chunksize=(1, 1200, 2400), meta=np.ndarray>
    mask              (time, lat, lon) float32 dask.array<chunksize=(1, 1800, 3600), meta=np.ndarray>
    sea_ice_fraction  (time, lat, lon) float32 dask.array<chunksize=(1, 1800, 3600), meta=np.ndarray>
```

Motivation

The dream is to open the entire dataset in a single line of code, pre-organized by it's dimensions / coordinates...
But how to avoid opening all the data at once if only a smaller portion (e.g. geographic region) is needed?

Utility of Virtual Data Sets (VDS's)

Analogy to the Table of Contents for a book



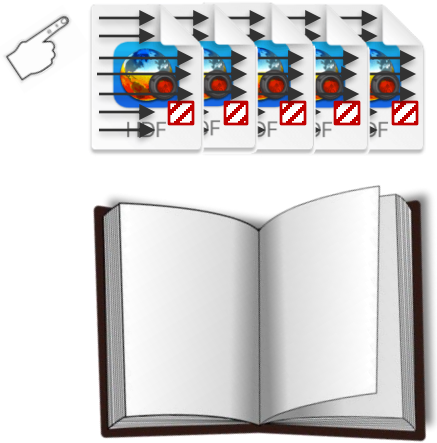
EARTHDATA

Utility of Virtual Data Sets (VDS's)

Analogy to the Table of Contents for a book

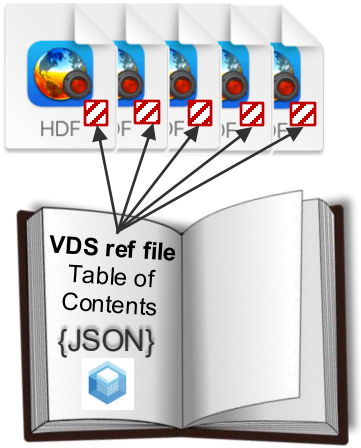
Using the native netCDF files

Read the whole book to get the parts we want



Using the VDS reference file

The VDS reference is a lightweight file acting as a table of contents



EARTHDATA

Utility of Virtual Data Sets (VDS's)

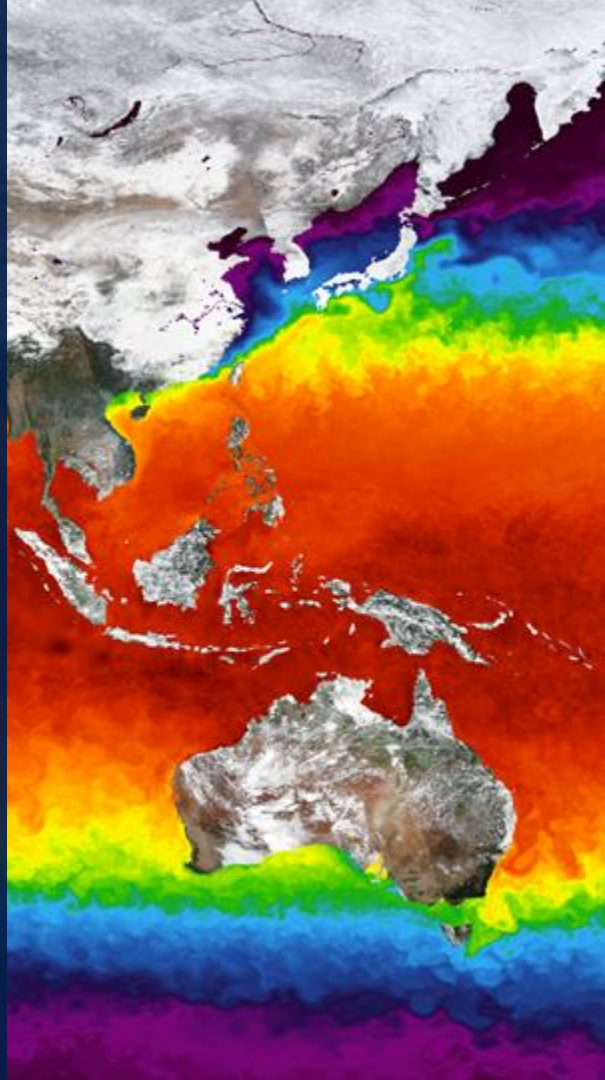
- **Enables "Analysis Ready, Cloud Optimized" (ARCO) Data**
  - Eliminates the need to download massive datasets, prepare them for analysis, and manage local storage.
- **Access in and outside of the Cloud**
  - Being in a cloud environment is not required! Although there are limitations to computation size out-of-cloud and best performance is in-cloud.
- **Improves Performance and Reduces Computation Burden**
  - Fast access and easily tunable sub-setting (order of magnitude faster).
  - Only the desired subset is downloaded, minimizing egress time, computational resources, and costs.
- **Simplifies Data Integration and Reproducibility**
  - Easily work with several data products regardless of their original format or location
  - Simplifies complex interdisciplinary research workflows.

EARTHDATA

Basic usage, Cookbook Resources, Benchmarking

Resources for getting started with VDS's and locating data sets that have them.

(Dean Henze, 15 minutes)



Cookbook Resources and Demo

Homepage
https://podaac.github.io/tutorials/

Chapter on Using Virtual Datasets
https://podaac.github.io/tutorials/quarto_text/UsingVirtualDatasets.html

VDS Starter Notebook
on the Cookbook: https://podaac.github.io/tutorials/notebooks/Advanced_cloud/using_vds_starter.html
on Github: https://github.com/podaac/tutorials/blob/master/notebooks/Advanced_cloud/using_vds_starter.ipynb

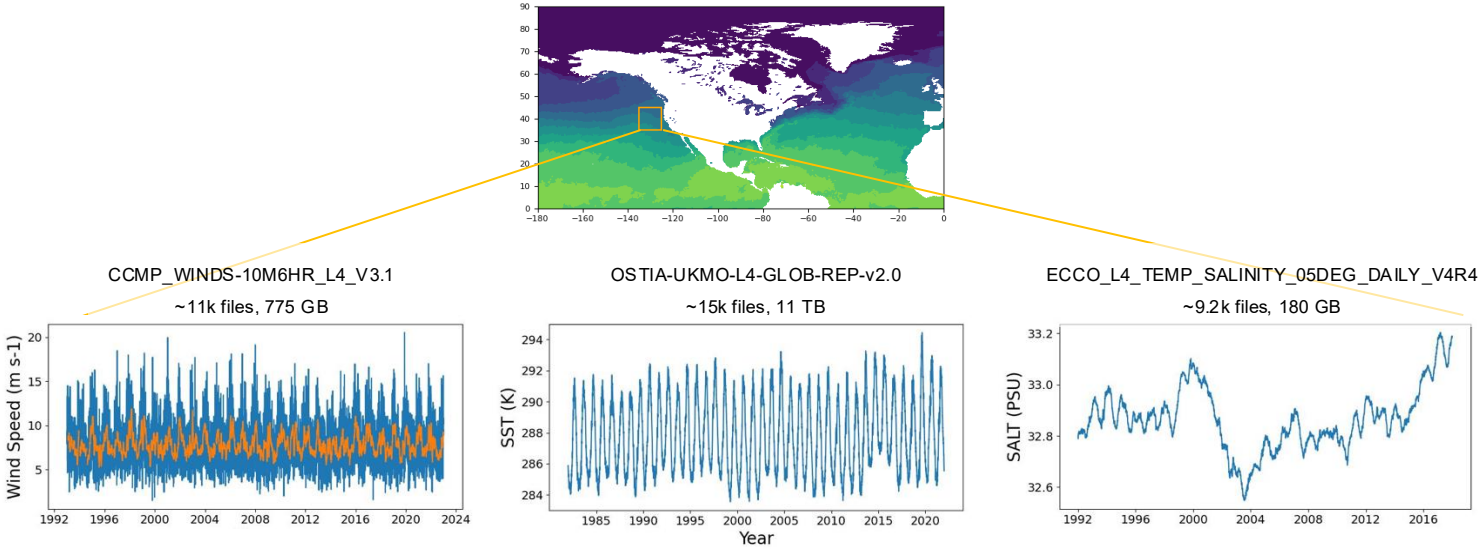
---

Citing Cookbook, DOI *https://doi.org/10.5281/zenodo.10530664*

Benchmarking Results

Sample Analysis

- Compute regional mean time series over 10° x 10° window off USA West Coast for three datasets / variables.
- Time computation spanning 1, 5, 10 , 20 years and full record.

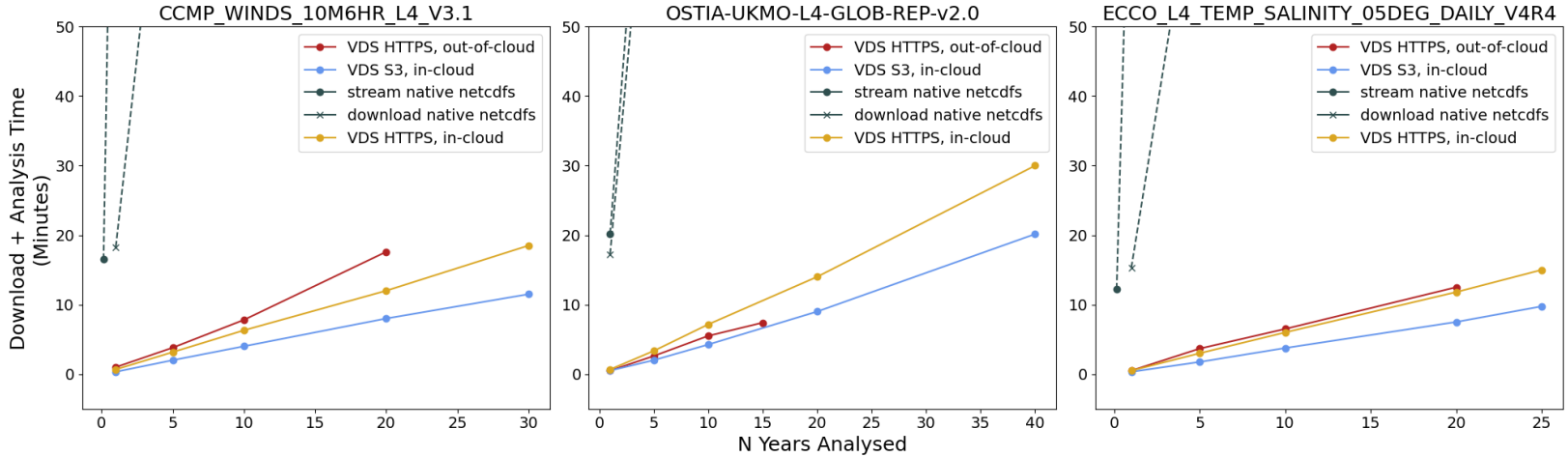


EARTHDATA

Benchmarking Results

Results

- Minimum of an order of magnitude faster than streaming or downloading native netCDF's.
- Recorded time includes any download and analysis times combined.

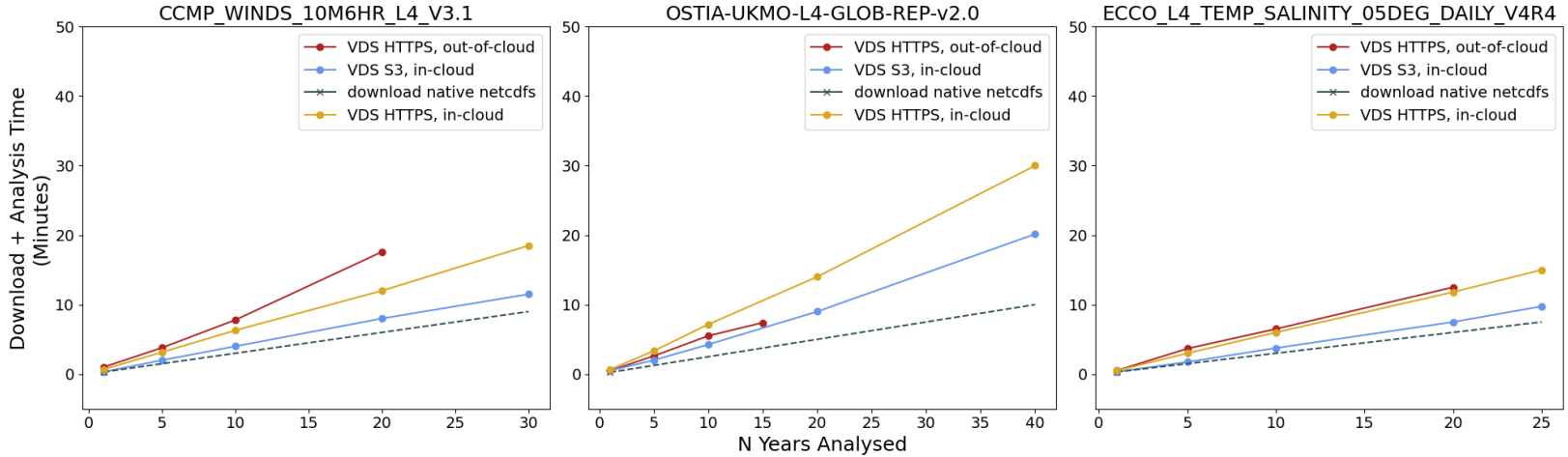


EARTHDATA

Benchmarking Results

Caveat

- Once data are downloaded, the computation completed faster locally than using the VDS's.

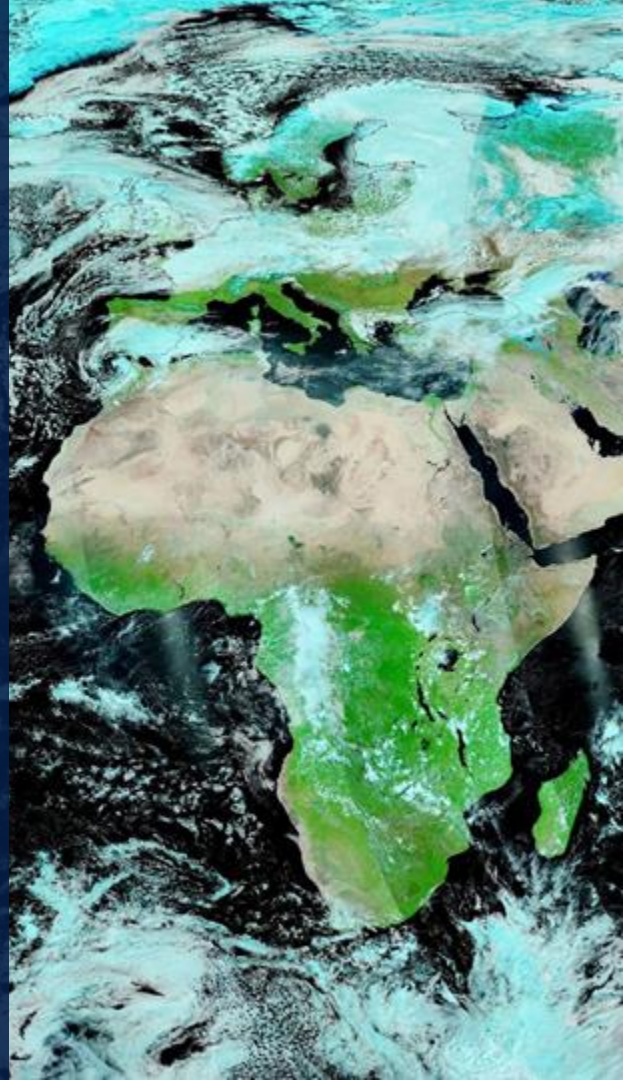


EARTHDATA

Science Use Case Example

Gulf of Tehuantepec Upwelling

(Edward Armstrong, 15 minutes)



Gulf of Tehuantepec Upwelling Example

Demonstrates:

- Interdisciplinary analysis incorporating 4 datasets of **SST, Ocean Wind, Vertical Velocity,** and **Surface Salinity**
- Reading, Re-gridding and Resampling datasets to common intervals
- Clever plots to explore the variability in time!

Notebook on Github:

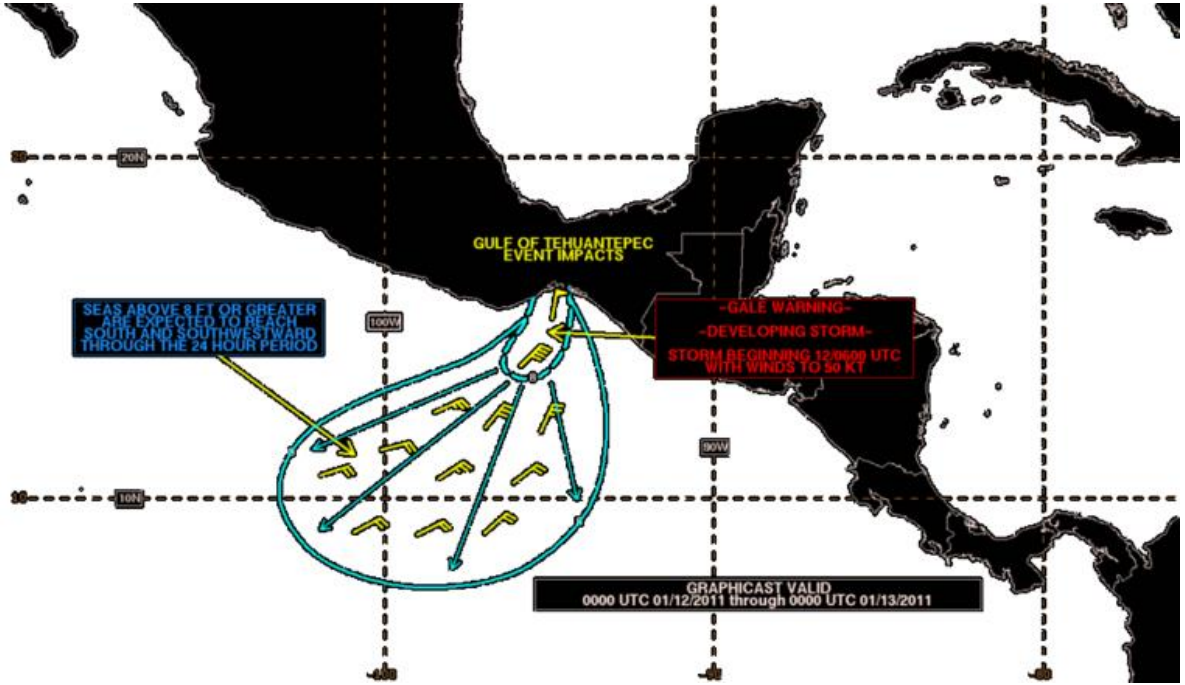
https://github.com/podaac/tutorials/blob/master/notebooks/Advanced_cloud/GulfofTehuantepec_Ocean_Response.ipynb

Notebook on the Cookbook:

https://podaac.github.io/tutorials/notebooks/Advanced_cloud/GulfofTehuantepec_Ocean_Response.html

EARTHDATA

Background - “Tehuantepecer” Wind Event



Causes Ocean Upwelling via changes in Wind Stress Curl !

Image source https://en.wikipedia.org/wiki/Tehuantepecer

EARTHDATA

Wind Stress Curl driven Ocean Upwelling

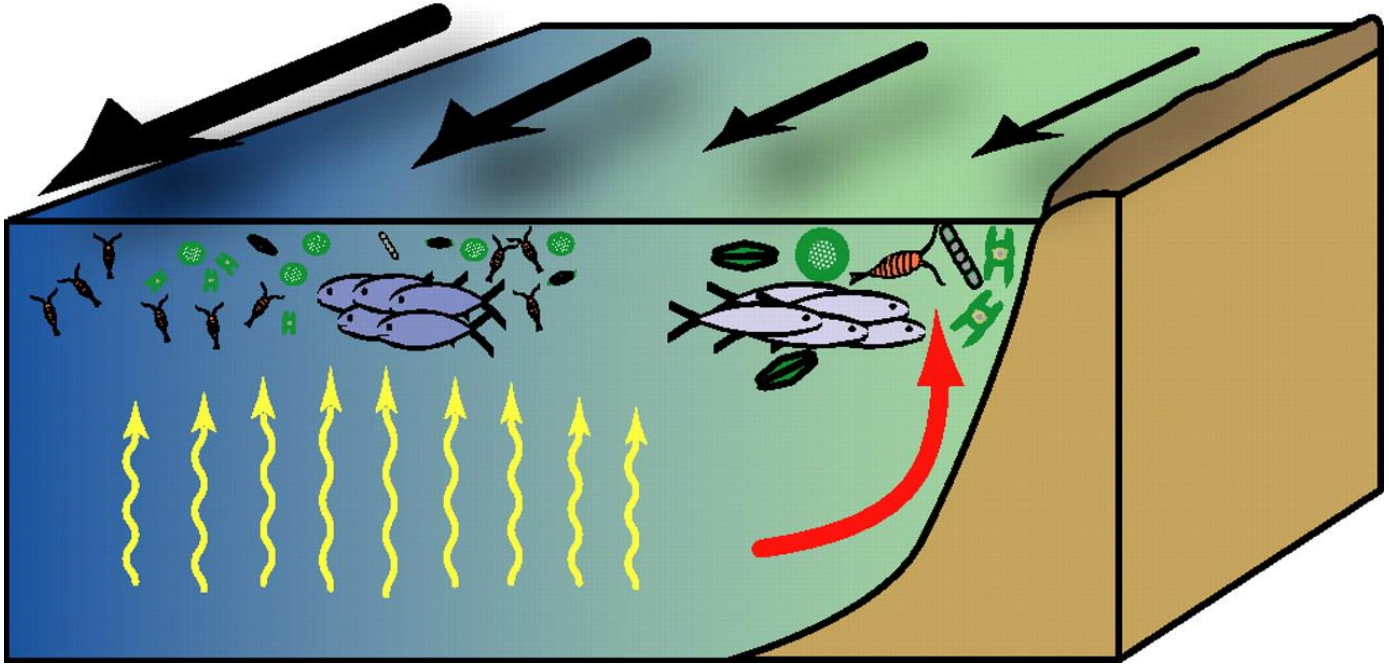
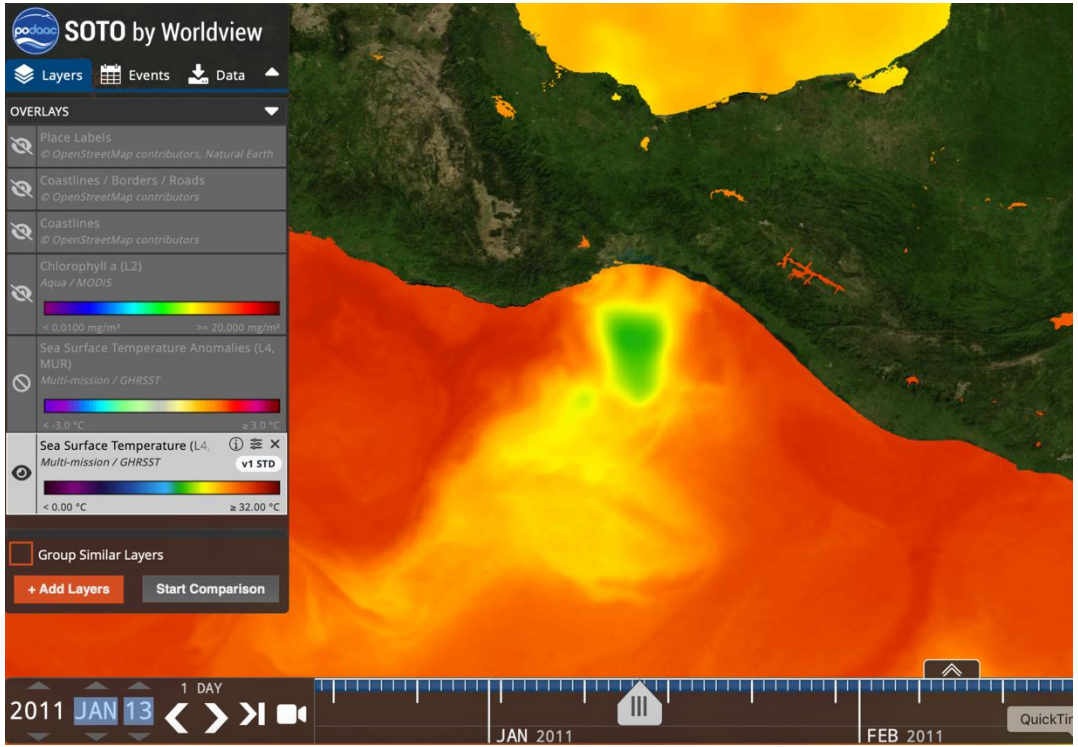


Image Source: Influence of ocean winds on the pelagic ecosystem in upwelling regions. https://doi.org/10.1073/pnas.0711777105

EARTHDATA

SST Ocean Response to Winds

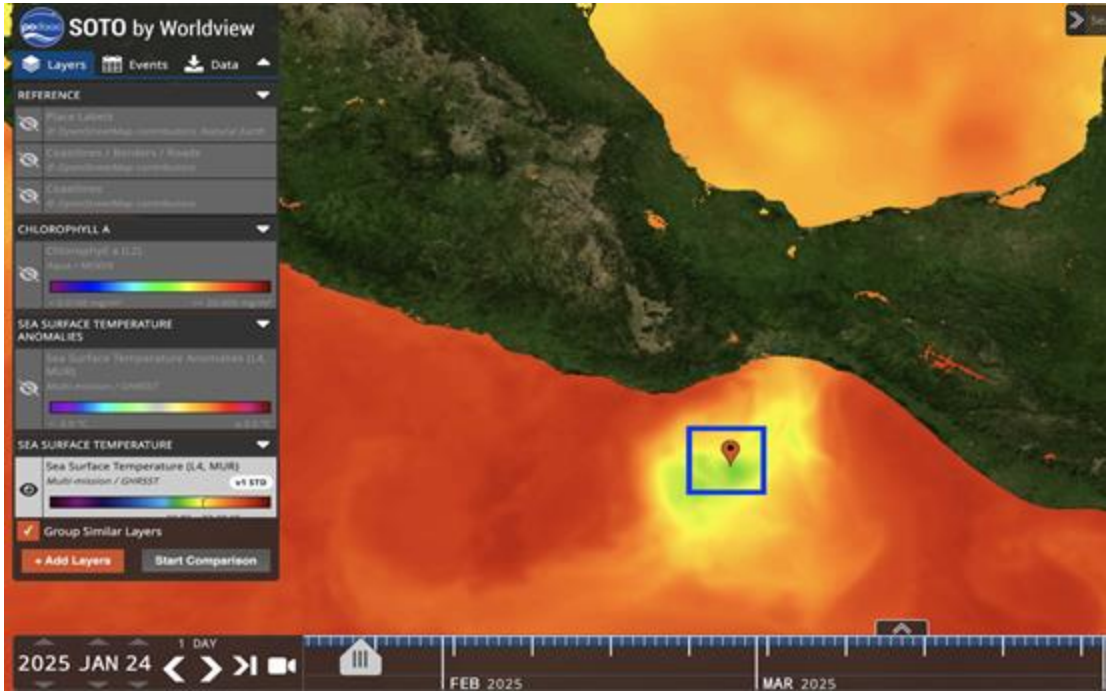


- **Sea Surface Temperature** (SST) response on Jan 13. One day after the wind event started on Jan 11.
- Cooling of nearly 10 ℃ !! Green vs dark Orange.

Image Source: https://soto.podaac.earthdatacloud.nasa.gov/?v=-103.18438961418703,9.823108911088696,-81.49032900055163,19.74671097835717&lg=false&t=2011-01-13-T14%3A16%3A53Z

EARTHDATA

ROI for the interdisciplinary analysis



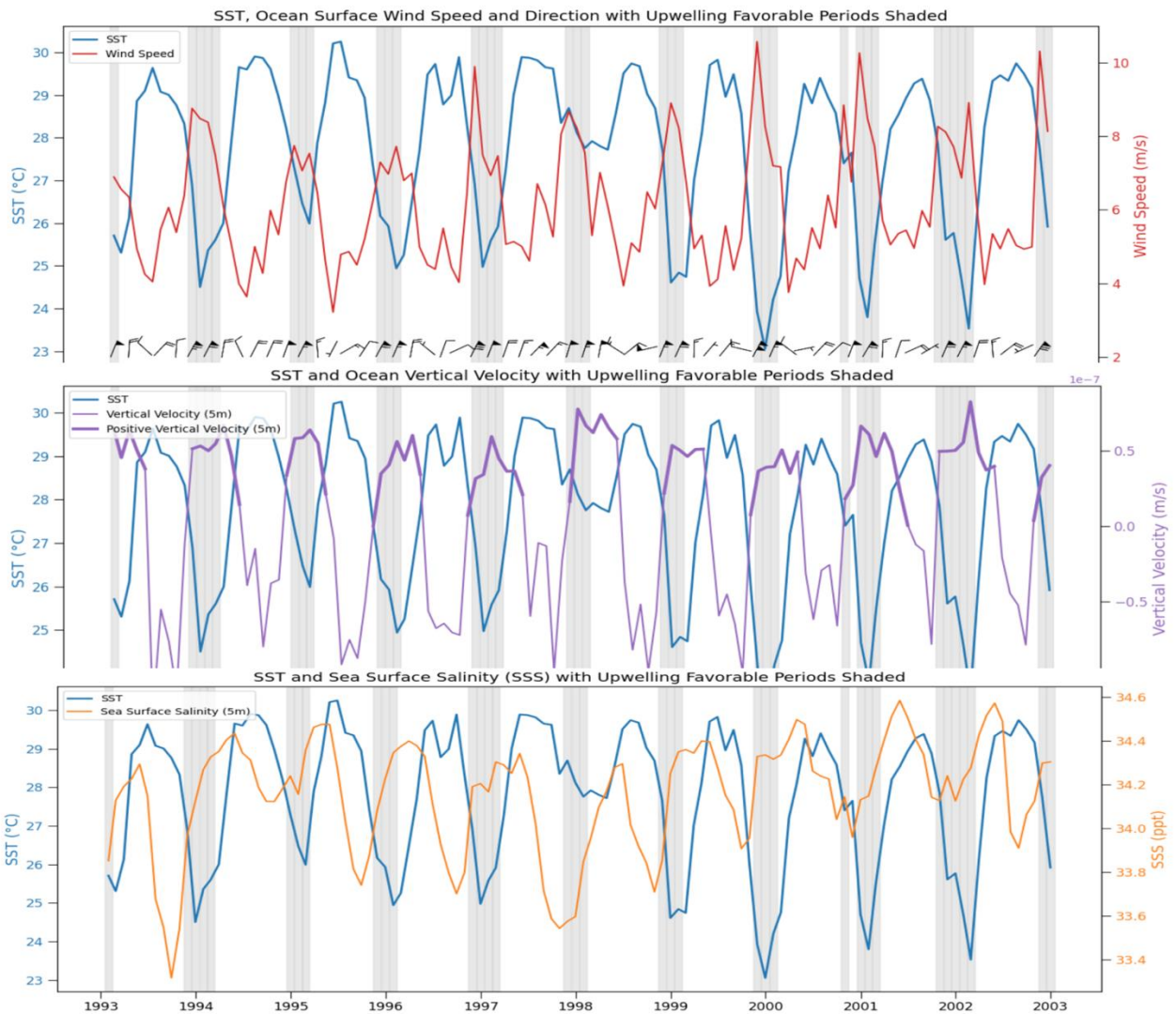
- **Load** the virtual datasets of SST, ocean wind,vertical velocity and salinity
- **Resample** to monthly, one degree intervals
- **Calculate** Wind Direction
- **Overplot** them and examine the relationships and variability
- **Generate** a 41 year SST spatial mean map

….. and now a Jupyter Notebook Demo

EARTHDATA

Summary

SST, **Wind**, **Vertical Velocity, Salinity** & **Upwelling Favorable Periods** in the Gulf of Tehuantepec, 1993-2003

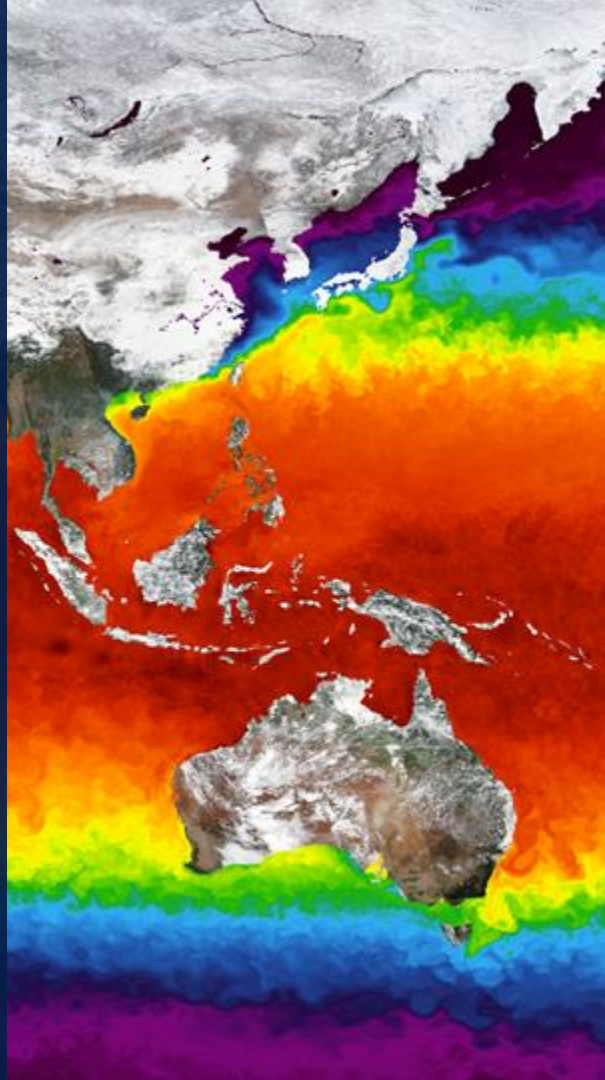


EARTHDATA

Scope of Usability, Resources, Future Objectives

What to keep in mind when using VDS's
and what to keep an eye out for

(Dean Henze, 10 minutes)



Scope of Usability

Check out full list of recommendations on the Cookbook chapter:

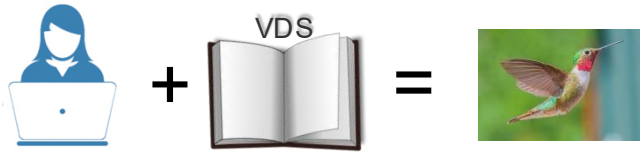
https://podaac.github.io/tutorials/quarto_text/UsingVirtualDatasets.html#guidance-on-scope-of-usability---dont-skip

We will highlight a few points here

EARTHDATA

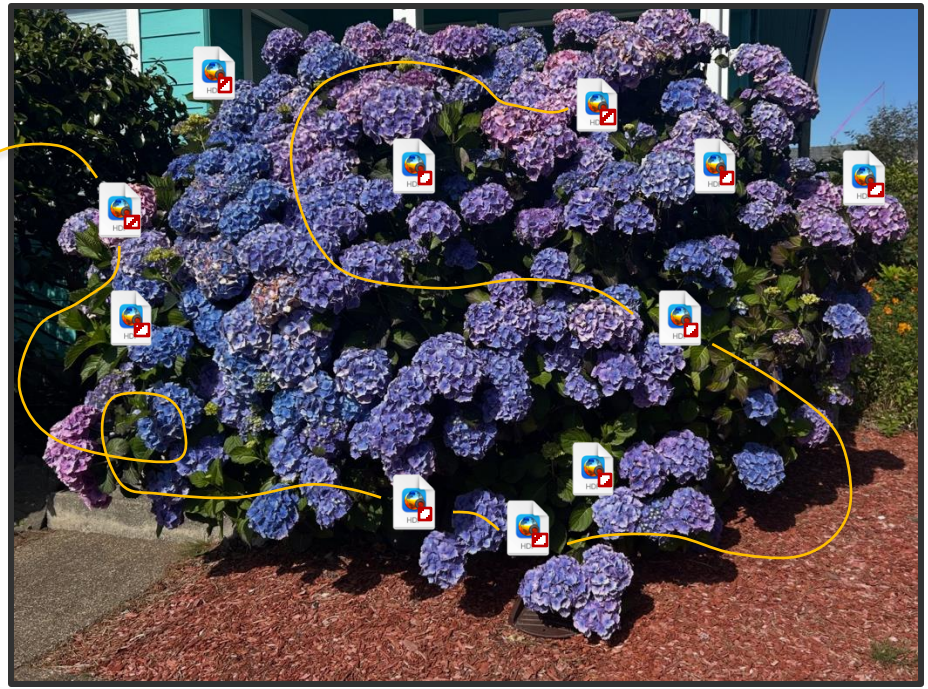
Scope of Usability

Setting Expectations for use without scaled computing



The VDS file allows you to rapidly jump between parts of files to get the needed data, but...

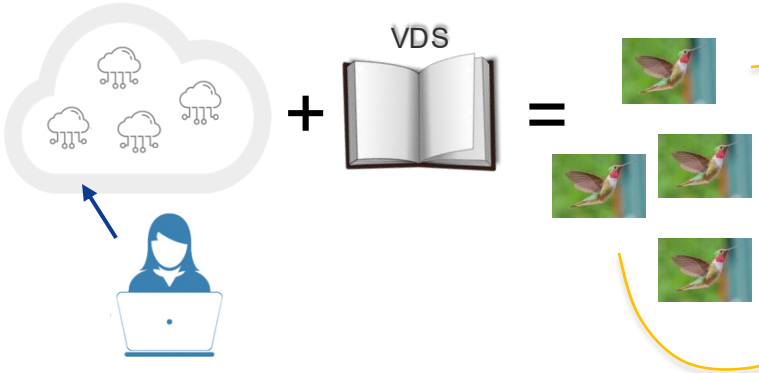
you probably can't access ALL the data at once from a small machine like a laptop.

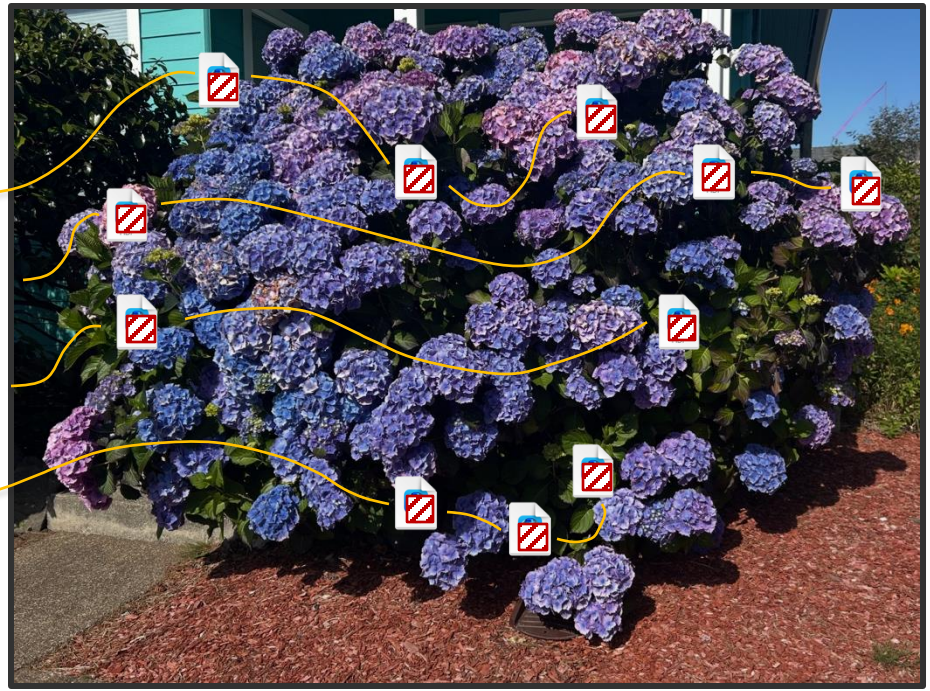


EARTHDATA

Scope of Usability

Setting Expectations for use without scaled computing



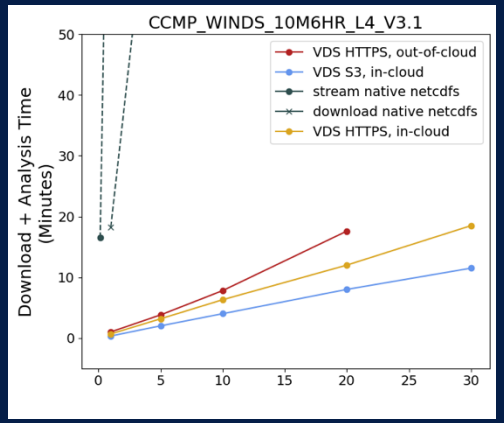


To crunch large fractions (or all) of the data set, you will still need increased computing power / memory (e.g. local computing cluster, or cloud VM's).

EARTHDATA

Scope of Usability

For larger computations we hit a limit with the amount of data able to be accessed at once. For larger computations you will need to be in-cloud.



EARTHDATA

Scope of Usability

These are not new datasets - they are VDS-versions of datasets that have already been published on PO.DAAC. Therefore, for documentation, metadata, citations, etc, please consult the original dataset landing pages (linked to in the table).

If you use VDS’s for your work, please cite the dataset!

EARTHDATA

Resources

The more Xarray + Dask you know, the more power quickly at your fingertips. The Docs are good.

Xarray in 45 minutes:
https://tutorial.xarray.dev/overview/xarray-in-45-min.html

Xarray full documentation, user guide, API, etc:
https://docs.xarray.dev/en/stable/index.html

High-level intro to dask and basic compatibility with Xarray:
https://tutorial.xarray.dev/intermediate/xarray_and_dask.html

Further functionality of Xarray + Dask:
https://docs.xarray.dev/en/stable/user-guide/dask.html

EARTHDATA

Resources

Virtual datasets and cloud-optimized formats, further learning

- Many resources available online.
- High-level overview: Cloud-Optimized Geospatial Formats Guide. For details on the package used to create virtual datasets, check out virtualizarr's read-the-docs page.
- Anyone can create a VDS for a new dataset if they are inclined (and if the underlying netCDF/HDF's are formatted appropriately). There is a bit of a learning curve, you can check out our virtualizarr recipes notebook and references therein to get started.

EARTHDATA

Resources

All links to resources are in our Cookbook chapter

https://podaac.github.io/tutorials/quarto_text/UsingVirtualDatasets.html#resources

EARTHDATA

Future Goals

This technology is actively being developed. Please let us know if you run into issues, we are monitoring this tech to improve it.

We may be updating the Cookbook chapter and notebooks frequently as the tech progresses. Check back periodically.

EARTHDATA

Future Goals

We aim to expand the number and types of datasets offered with VDS capability (open to community feedback):

- Continue expanding L3 / L4 datasets.
- Level 2 products. Prelim tests look good, but highly dependent on file uniformity. SWOT product?...
- Datasets with HDF groups.
- Composite VDS products. For example, a single ECCO VDS with all variables.
- Forward streaming records - updating VDS's as the latest data become available.

EARTHDATA

Future Goals

Integration of VDS search and access capabilities into earthaccess:

```
import xarray as xr
import earthaccess

earthaccess.login()
vds_mapper = earthaccess.get_virtual_reference(shortname)
data = xr.open_dataset(
   vds_mapper, engine="zarr", chunks={},
   backend_kwargs={"consolidated": False}
)
```

EARTHDATA

Summary – Reiterating Utility of VDS's

- **Enables "Analysis Ready, Cloud Optimized" (ARCO) Data**
  - Eliminates the need to download massive datasets, prepare them for analysis, and manage local storage.
- **Access in and outside of the Cloud**
  - Being in a cloud environment is not required! Although there are limitations to computation size out-of-cloud and best performance is in-cloud.
- **Improves Performance and Reduces Computation Burden**
  - Fast access and easily tunable sub-setting (order of magnitude faster).
  - Only the desired subset is downloaded, minimizing egress time, computational resources, and costs.
- **Simplifies Data Integration and reproducibility**
  - Easily work with several data products regardless of their original format or location
  - Simplifies complex interdisciplinary research workflows.

EARTHDATA

Summary – Reiterating Utility of VDS's

Efficiencies in

- Computation time
- Lines of code written
- Data downloaded = $$
- ***Less physical energy used??***

EARTHDATA

Thanks to

Major Contributors to PO.DAAC's VDS Knowledge / Progress

Ayush Nag (past JPL Intern)

Ryan Abernathy, Joe Hamman (Earthmover Team)

Aimee Barciauskas (Development Seed)

Luis Lopez (NSIDC)

Virtualizarr Team (e.g. Tom Nicholas)

Additional Thanks To

Openscapes / earthaccess Community

Coiled Team

Reach Out

Let us know your science use cases, dream data products, troubleshooting issues.

EMAIL
dean.c.henze@jpl.nasa.gov
edward.m.armstrong@jpl.nasa.gov
podaac@podaac.jpl.nasa.gov

WEBSITE
https://podaac.jpl.nasa.gov

EARTHDATA FORUM
https://forum.earthdata.nasa.gov


EARTHDATA
earthdata.nasa.gov
Thank You for Joining
podaac@podaac.jpl.nasa.gov | https://podaac.jpl.nasa.gov