Free, public, standardized Zarr stores of geospatial data in the cloud for all! Now in Beta.

Christine Smit[1,2], Hailiang Zhang[1,3], Brianna Pagán[1,3], Dieu My Nguyen[1,3], James Acker[1,3], Ashley Heath[1,3], Mahabaleshwara Hegde[1], Long Pham[1]

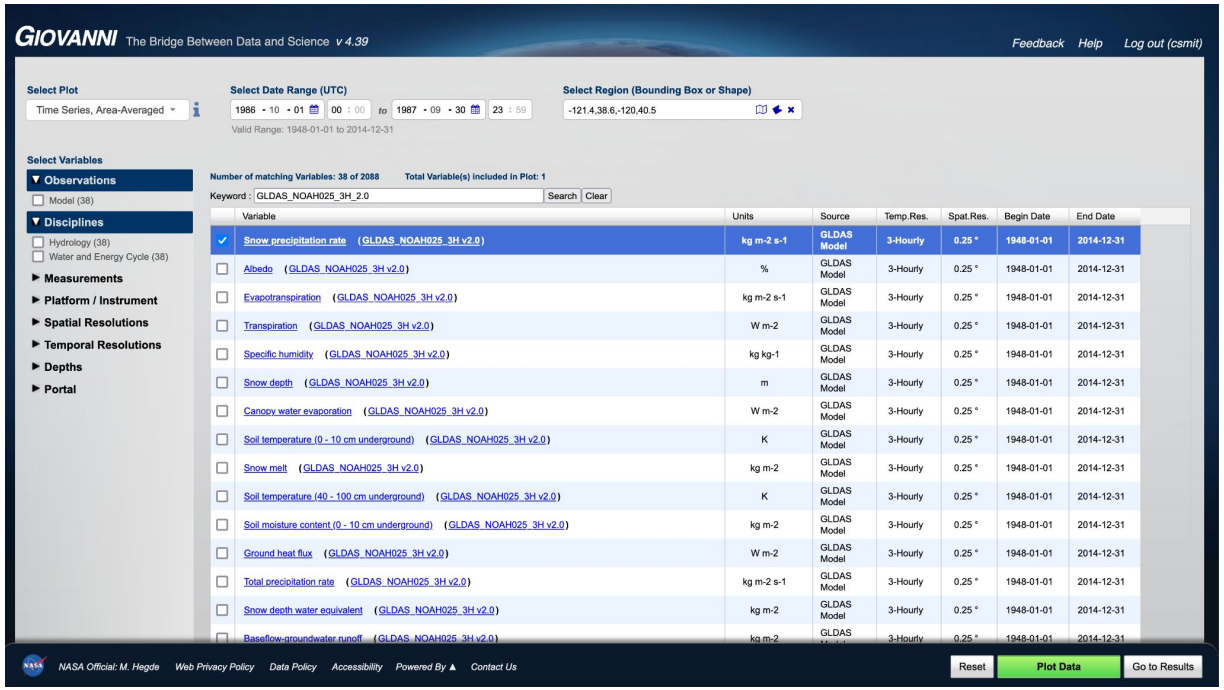
1 - NASA Goddard Earth Science Data and Information Science Center (GES DISC), 2 - Telophase, 3 - Adnet

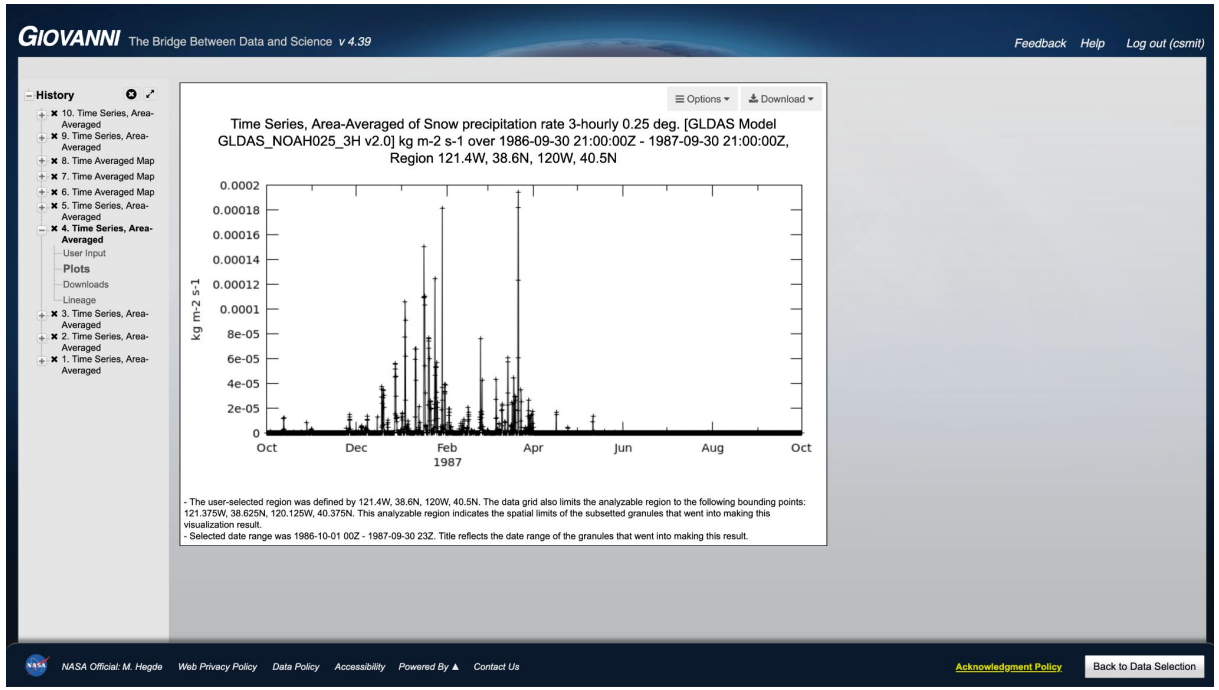
Outline

- A little history
- Data
- Demo!
- Documentation
- Limitations and future plans
- How to contact us

A quick intro to Giovanni

https://giovanni.gsfc.nasa.gov/giovanni



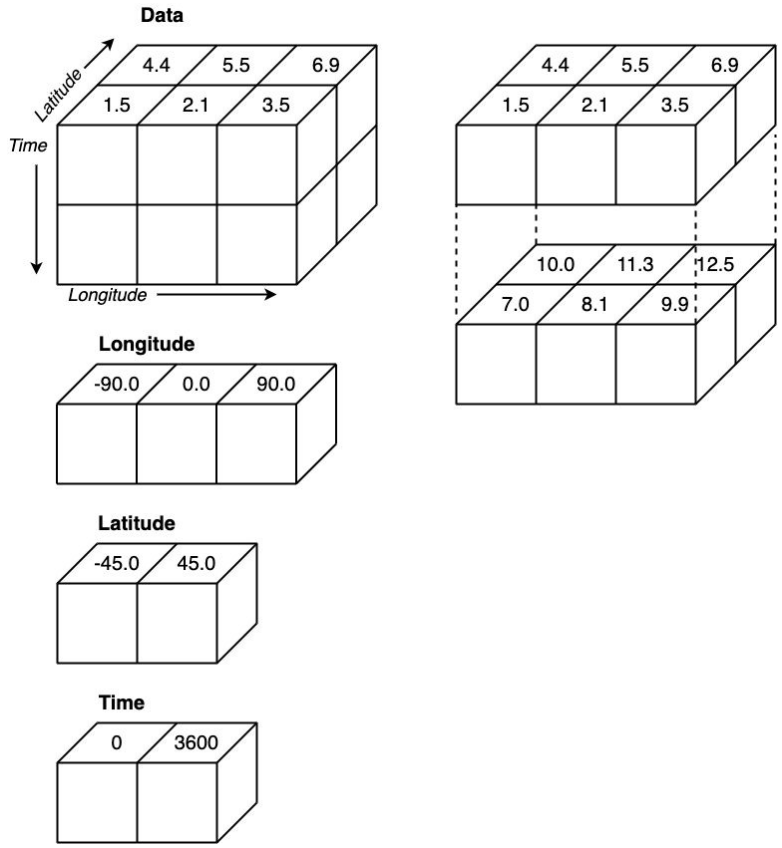


Cloud Part 1: Parquet

~2018-2020

Parquet

Multi-dimensional array with dimension variables



Point data (Data frame)

Longitude	Latitude	Time	Data
-90.0	-45.0	0	1.5
0.0	-45.0	0	2.1
90.0	-45.0	0	3.5
-90.0	45.0	0	4.4
0.0	45.0	0	5.5
90.0	45.0	0	6.9
-90.0	-45.0	3600	7.0
0.0	-45.0	3600	8.1
90.0	-45.0	3600	9.9
-90.0	45.0	3600	10.0
0.0	45.0	3600	11.3
90.0	45.0	3600	12.5

Parquet highlights (~2018-2020)

The good ✅:

- Easiest algorithm writing ever!
- Clean, simple architecture

https://www.youtube.com/watch?v=CbH9SVrPSUA

Parquet highlights (~2018-2020)

The good ✅:

- Easiest algorithm writing ever!
- Clean, simple architecture

The bad ⛔:

- Lack of open source solutions that performed well for our tasks
- Inconsistent performance from AWS Athena

https://www.youtube.com/watch?v=CbH9SVrPSUA

Parquet highlights (~2018-2020)

The good ✅:

- Easiest algorithm writing ever!
- Clean, simple architecture

The bad ⛔:

- Lack of open source solutions that performed well for our tasks
- Inconsistent performance from AWS Athena

The ugly 😖:

- Looming cost problems with AWS Athena
- Difficulty of updating enormous parquet files

https://www.youtube.com/watch?v=CbH9SVrPSUA

Cloud Part 2: Zarr

> 2020

Moving to zarr

The good ✅:

- Supports multi-dimensional arrays natively. Is largely compatible with NetCDF and HDF data models, which is a good fit for our data.
- Easy to update as new observations come in.
- Excellent open source support.

Moving to zarr

The good ✅:

- Supports multi-dimensional arrays natively. Is largely compatible with NetCDF and HDF data models, which is a good fit for our data.
- Easy to update as new observations come in.
- Excellent open source support.

The bad ⛔:

- Making zarr stores public that need to be updated requires additional instrumentation beyond the zarr library.

What we're making public

24 "complete" zarr stores (no new data coming in)

What we're making public

24 "complete" zarr stores (no new data coming in)

- hydrology variables from GLDAS_NOAH025_3H v2.0
  - GLDAS Noah Land Surface Model L4 3 hourly 0.25 x 0.25 degree V2.0
  - Global spatial extent
  - Land-only
  - 1948-01-01 to 2014-12-31

Sidebar: how NASA archives earth science data

Sidebar: how NASA archives earth science data

Product /
collection

Sidebar: how NASA archives earth science data

Granules / files

Product / collection

GLDAS_NOAH025_3H.A19480101.0300.020.nc4
GLDAS_NOAH025_3H.A19480101.0600.020.nc4
GLDAS_NOAH025_3H.A19480101.0900.020.nc4
GLDAS_NOAH025_3H.A19480101.1200.020.nc4
GLDAS_NOAH025_3H.A19480101.1500.020.nc4
GLDAS_NOAH025_3H.A19480101.1800.020.nc4
GLDAS_NOAH025_3H.A19480101.2100.020.nc4
GLDAS_NOAH025_3H.A19480102.0000.020.nc4
GLDAS_NOAH025_3H.A19480102.0300.020.nc4
GLDAS_NOAH025_3H.A19480102.0600.020.nc4
GLDAS_NOAH025_3H.A19480102.0900.020.nc4
GLDAS_NOAH025_3H.A19480102.1200.020.nc4
GLDAS_NOAH025_3H.A19480102.1500.020.nc4
GLDAS_NOAH025_3H.A19480102.1800.020.nc4
GLDAS_NOAH025_3H.A19480102.2100.020.nc4
GLDAS_NOAH025_3H.A19480103.0000.020.nc4
GLDAS_NOAH025_3H.A19480103.0300.020.nc4
GLDAS_NOAH025_3H.A19480103.0600.020.nc4
GLDAS_NOAH025_3H.A19480103.0900.020.nc4
GLDAS_NOAH025_3H.A19480103.1200.020.nc4
GLDAS_NOAH025_3H.A19480103.1500.020.nc4
GLDAS_NOAH025_3H.A19480103.1800.020.nc4
GLDAS_NOAH025_3H.A19480103.2100.020.nc4
GLDAS_NOAH025_3H.A19480104.0000.020.nc4
GLDAS_NOAH025_3H.A19480104.0300.020.nc4
GLDAS_NOAH025_3H.A19480104.0600.020.nc4
GLDAS_NOAH025_3H.A19480104.0900.020.nc4
...

Sidebar: how NASA archives earth science data

Product / collection

Granules / files

GLDAS_NOAH025_3H.A19480101.0300.020.nc4
GLDAS_NOAH025_3H.A19480101.0600.020.nc4
GLDAS_NOAH025_3H.A19480101.0900.020.nc4
GLDAS_NOAH025_3H.A19480101.1200.020.nc4
GLDAS_NOAH025_3H.A19480101.1500.020.nc4
GLDAS_NOAH025_3H.A19480101.1800.020.nc4
GLDAS_NOAH025_3H.A19480101.2100.020.nc4
GLDAS_NOAH025_3H.A19480102.0000.020.nc4
GLDAS_NOAH025_3H.A19480102.0300.020.nc4
GLDAS_NOAH025_3H.A19480102.0600.020.nc4
GLDAS_NOAH025_3H.A19480102.0900.020.nc4
GLDAS_NOAH025_3H.A19480102.1200.020.nc4
GLDAS_NOAH025_3H.A19480102.1500.020.nc4
GLDAS_NOAH025_3H.A19480102.1800.020.nc4
GLDAS_NOAH025_3H.A19480102.2100.020.nc4
GLDAS_NOAH025_3H.A19480103.0000.020.nc4
GLDAS_NOAH025_3H.A19480103.0300.020.nc4
GLDAS_NOAH025_3H.A19480103.0600.020.nc4
GLDAS_NOAH025_3H.A19480103.0900.020.nc4
GLDAS_NOAH025_3H.A19480103.1200.020.nc4
GLDAS_NOAH025_3H.A19480103.1500.020.nc4
GLDAS_NOAH025_3H.A19480103.1800.020.nc4
GLDAS_NOAH025_3H.A19480103.2100.020.nc4
GLDAS_NOAH025_3H.A19480104.0000.020.nc4
GLDAS_NOAH025_3H.A19480104.0300.020.nc4
GLDAS_NOAH025_3H.A19480104.0600.020.nc4
GLDAS_NOAH025_3H.A19480104.0900.020.nc4
...

Data variables / parameters

```
float Swnet_tavg(time, lat, lon) ;
        Swnet_tavg:standard_name = "surface_net_downward_shortwave_flux" ;
        Swnet_tavg:long_name = "Net short wave radiation flux" ;
        Swnet_tavg:units = "W m-2" ;
        Swnet_tavg:_FillValue = -9999.f ;
        Swnet_tavg:missing_value = -9999.f ;
        Swnet_tavg:cell_methods = "time: mean" ;
        Swnet_tavg:vmin = 0.f ;
        Swnet_tavg:vmax = 1041.f ;
float Lwnet_tavg(time, lat, lon) ;
        Lwnet_tavg:standard_name = "surface_net_downward_longwave_flux" ;
        Lwnet_tavg:long_name = "Net long-wave radiation flux" ;
        Lwnet_tavg:units = "W m-2" ;
        Lwnet_tavg:_FillValue = -9999.f ;
        Lwnet_tavg:missing_value = -9999.f ;
        Lwnet_tavg:cell_methods = "time: mean" ;
        Lwnet_tavg:vmin = -309.47f ;
        Lwnet_tavg:vmax = 42.44998f ;
float Qle_tavg(time, lat, lon) ;
        Qle_tavg:standard_name = "surface_upward_latent_heat_flux" ;
        Qle_tavg:long_name = "Latent heat net flux" ;
        Qle_tavg:units = "W m-2" ;
        Qle_tavg:_FillValue = -9999.f ;
        Qle_tavg:missing_value = -9999.f ;
        Qle_tavg:cell_methods = "time: mean" ;
        Qle_tavg:vmin = -73.88426f ;
        Qle_tavg:vmax = 553.9257f ;
...
```

Example: original NetCDF data granule for GLDAS_NOAH025_3H v2.0

xarray.Dataset

▸ Dimensions: (**time**: 1, bnds: 2, **lon**: 1440, **lat**: 600)

▾ Coordinates:

time	(time)	datetime64[ns]	1948-01-01T03:00:00
lon	(lon)	float32	-179.9 -179.6 ... 179.6 179.9
lat	(lat)	float32	-59.88 -59.62 ... 89.62 89.88

▾ Data variables:

time_bnds	(time, bnds)	datetime64[ns]	...
Swnet_tavg	(time, lat, lon)	float32	...
Lwnet_tavg	(time, lat, lon)	float32	

...

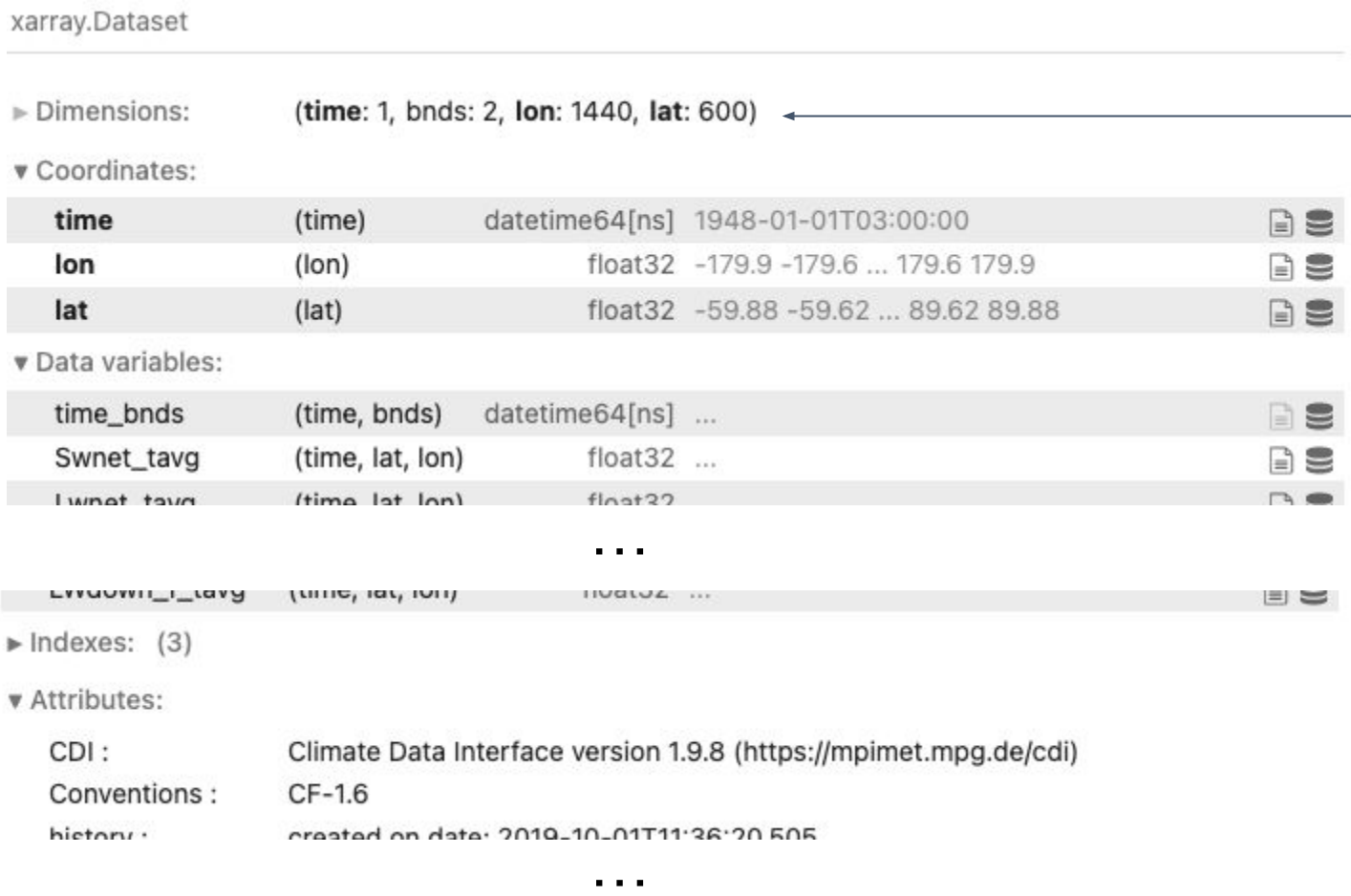
▸ Indexes: (3)

▾ Attributes:

CDI :	Climate Data Interface version 1.9.8 (https://mpimet.mpg.de/cdi)
Conventions :	CF-1.6
history :	created on date: 2019-10-01T11:36:20.505

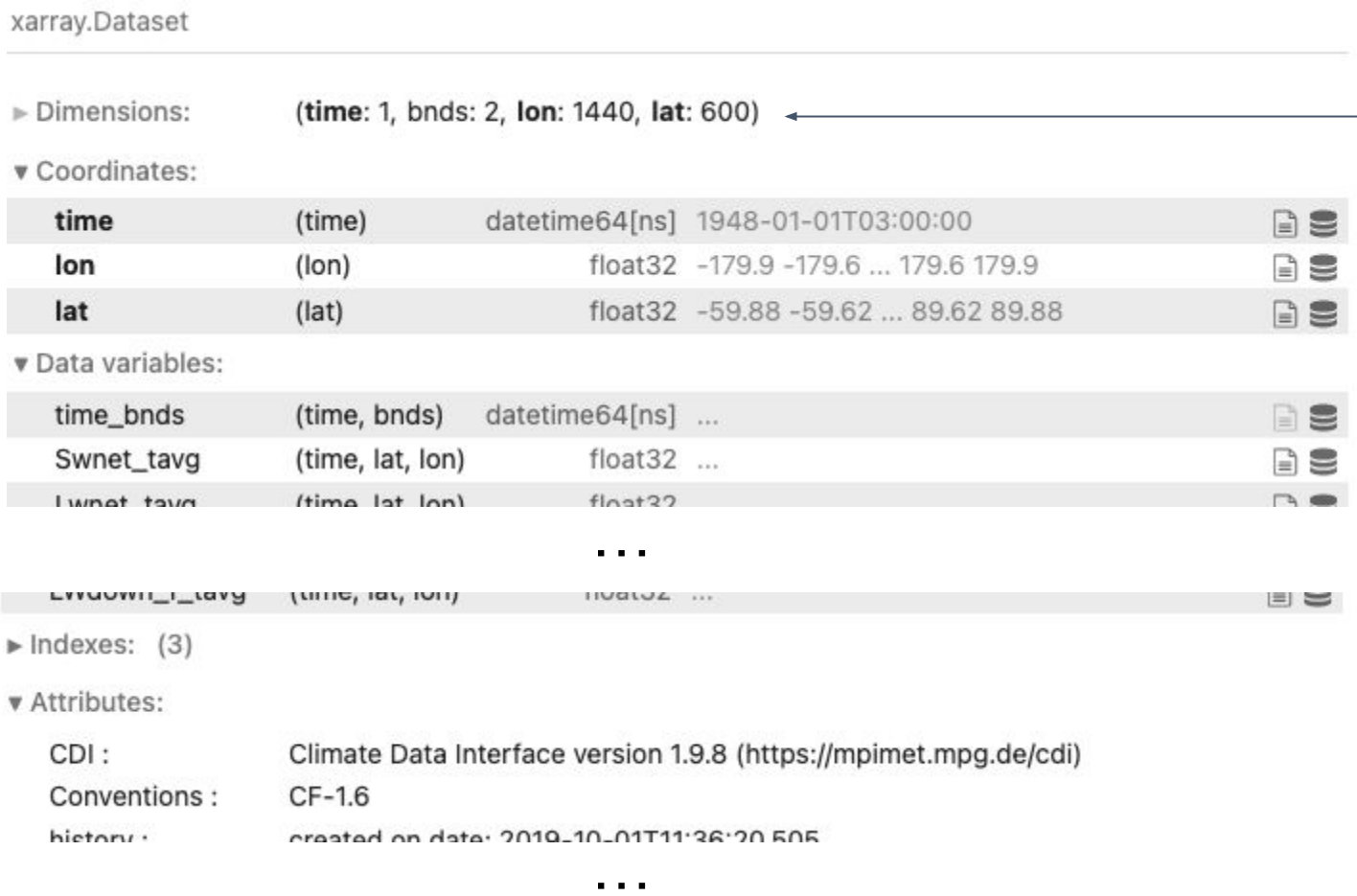
...

Example: original NetCDF data granule for GLDAS_NOAH025_3H v2.0



Data for just a single time slice

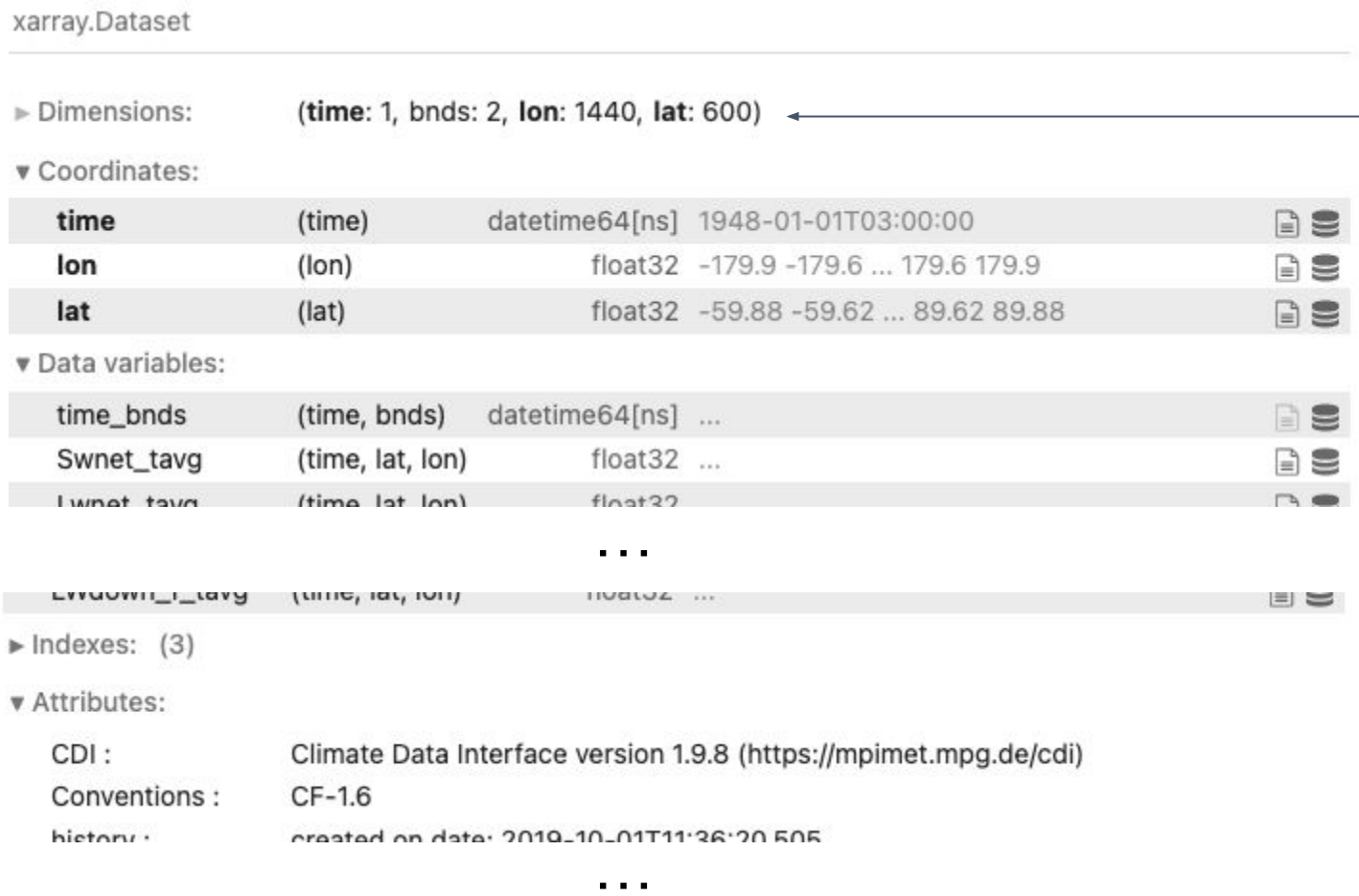
Example: original NetCDF data granule for GLDAS_NOAH025_3H v2.0



Data for just a single time slice

Coordinates

Example: original NetCDF data granule for GLDAS_NOAH025_3H v2.0

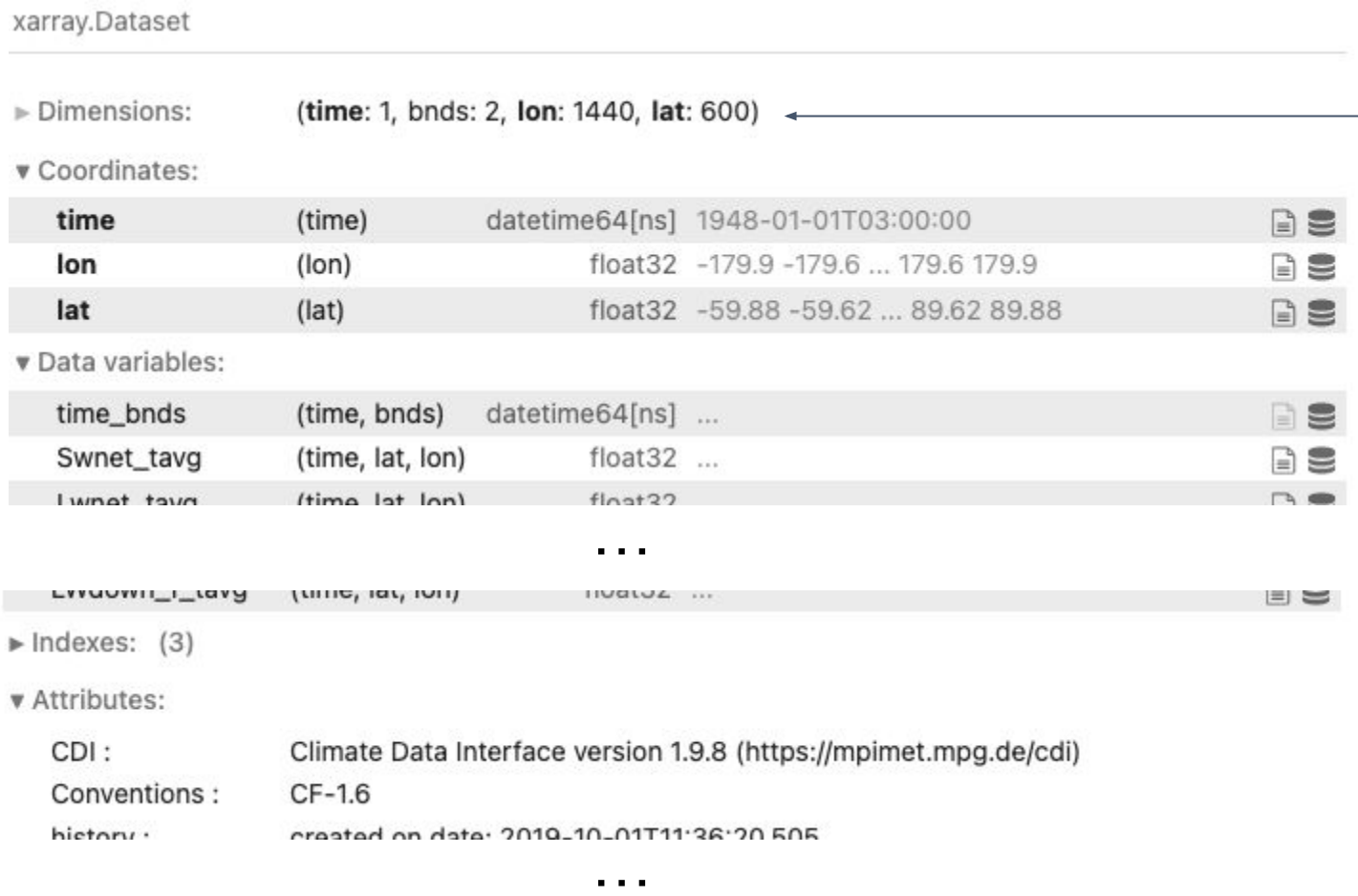


Data for just a single time slice

Coordinates

Bounds for one dimension

Example: original NetCDF data granule for GLDAS_NOAH025_3H v2.0



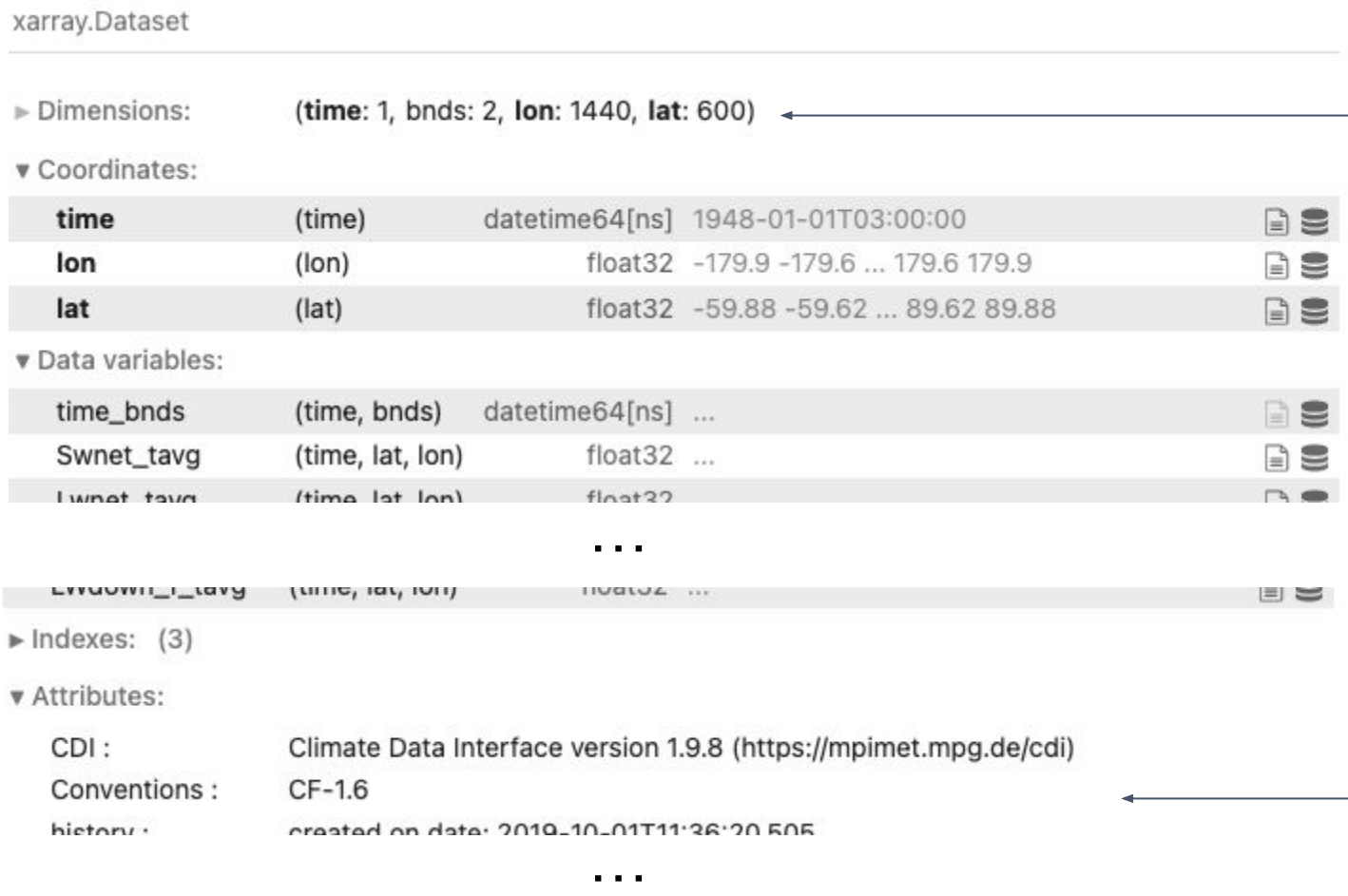
Data for just a single time slice

Coordinates

Bounds for one dimension

36 data variables

Example: original NetCDF data granule for GLDAS_NOAH025_3H v2.0



Data for just a single time slice

Coordinates

Bounds for one dimension

36 data variables

Metadata about granule and product

Zarr store normalization

Zarr store normalization

- Standardize dimensions and bounds:
  - name
  - order
  - units

Zarr store normalization

- Standardize dimensions and bounds:
  - name
  - order
  - units
- Aggregate across time

Zarr store normalization

- Standardize dimensions and bounds:
  - name
  - order
  - units
- Aggregate across time
- Include only one measurement in each zarr store

Zarr store normalization

- Standardize dimensions and bounds:
  - name
  - order
  - units
- Aggregate across time
- Include only one measurement in each zarr store

⇒ Makes the data look "weird" if you know the original product, but makes writing algorithms easier.

Example: Zarr store for Snowf_tavg variable in GLDAS_NOAH025_3H v2.0

xarray.Dataset

▸ Dimensions: (**latitude**: 600, bounds: 2, **longitude**: 1440, **time**: 196000)

▾ Coordinates:

latitude	(latitude)	float64	-59.88 -59.62 ... 89.62 89.88
longitude	(longitude)	float32	-179.9 -179.6 ... 179.6 179.9
time	(time)	datetime64[ns]	1948-01-01T03:00:00 ... 1947-...

▾ Data variables:

latitude_bounds	(latitude, bounds)	float64	dask.array<chunksize=(18, 2), ...
longitude_boun...	(longitude, bounds)	float64	dask.array<chunksize=(36, 2), ...
time_bounds	(time, bounds)	datetime64[s]	dask.array<chunksize=(800, 2),...
variable	(latitude, longitude, time)	float32	dask.array<chunksize=(18, 36, ...
long_name :	Snow precipitation rate		
units :	kg m-2 s-1		

▸ Indexes: (3)

▾ Attributes:

Conventions :	CF-1.10
DOI :	10.5067/342OHQM9AK6Q
product_long_n...	GLDAS Noah Land Surface Model L4 3 hourly 0.25 x 0.25 degree V2.0
product_short_...	GLDAS_NOAH025_3H
product_version :	2.0

Example: Zarr store for Snowf_tavg variable in GLDAS_NOAH025_3H v2.0

xarray.Dataset

▸ Dimensions: (**latitude**: 600, bounds: 2, **longitude**: 1440, **time**: 196000)

▾ Coordinates:

latitude	(latitude)	float64	-59.88 -59.62 ... 89.62 89.88
longitude	(longitude)	float32	-179.9 -179.6 ... 179.6 179.9
time	(time)	datetime64[ns]	1948-01-01T03:00:00 ... 1947-...

▾ Data variables:

latitude_bounds	(latitude, bounds)	float64	dask.array<chunksize=(18, 2), ...
longitude_boun...	(longitude, bounds)	float64	dask.array<chunksize=(36, 2), ...
time_bounds	(time, bounds)	datetime64[s]	dask.array<chunksize=(800, 2),...
variable	(latitude, longitude, time)	float32	dask.array<chunksize=(18, 36, ...
long_name :	Snow precipitation rate		
units :	kg m-2 s-1		

▸ Indexes: (3)

▾ Attributes:

Conventions :	CF-1.10
DOI :	10.5067/342OHQM9AK6Q
product_long_n...	GLDAS Noah Land Surface Model L4 3 hourly 0.25 x 0.25 degree V2.0
product_short_...	GLDAS_NOAH025_3H
product_version :	2.0

Data consolidated across all granules.

Example: Zarr store for Snowf_tavg variable in GLDAS_NOAH025_3H v2.0

xarray.Dataset

▸ Dimensions: (**latitude**: 600, bounds: 2, **longitude**: 1440, **time**: 196000)

▾ Coordinates:

latitude	(latitude)	float64	-59.88 -59.62 ... 89.62 89.88
longitude	(longitude)	float32	-179.9 -179.6 ... 179.6 179.9
time	(time)	datetime64[ns]	1948-01-01T03:00:00 ... 1947-...

▾ Data variables:

latitude_bounds	(latitude, bounds)	float64	dask.array<chunksize=(18, 2), ...
longitude_boun...	(longitude, bounds)	float64	dask.array<chunksize=(36, 2), ...
time_bounds	(time, bounds)	datetime64[s]	dask.array<chunksize=(800, 2),...
variable	(latitude, longitude, time)	float32	dask.array<chunksize=(18, 36, ...
long_name :	Snow precipitation rate		
units :	kg m-2 s-1		

▸ Indexes: (3)

▾ Attributes:

Conventions :	CF-1.10
DOI :	10.5067/342OHQM9AK6Q
product_long_n...	GLDAS Noah Land Surface Model L4 3 hourly 0.25 x 0.25 degree V2.0
product_short_...	GLDAS_NOAH025_3H
product_version :	2.0

Data consolidated across all granules.

Standardized coordinates with standardized names and units.

Example: Zarr store for Snowf_tavg variable in GLDAS_NOAH025_3H v2.0

xarray.Dataset

▸ Dimensions: (**latitude**: 600, bounds: 2, **longitude**: 1440, **time**: 196000)

▾ Coordinates:

latitude	(latitude)	float64	-59.88 -59.62 ... 89.62 89.88
longitude	(longitude)	float32	-179.9 -179.6 ... 179.6 179.9
time	(time)	datetime64[ns]	1948-01-01T03:00:00 ... 1947-...

▾ Data variables:

latitude_bounds	(latitude, bounds)	float64	dask.array<chunksize=(18, 2), ...
longitude_boun...	(longitude, bounds)	float64	dask.array<chunksize=(36, 2), ...
time_bounds	(time, bounds)	datetime64[s]	dask.array<chunksize=(800, 2),...
variable	(latitude, longitude, time)	float32	dask.array<chunksize=(18, 36, ...
long_name :	Snow precipitation rate		
units :	kg m-2 s-1		

▸ Indexes: (3)

▾ Attributes:

Conventions :	CF-1.10
DOI :	10.5067/342OHQM9AK6Q
product_long_n...	GLDAS Noah Land Surface Model L4 3 hourly 0.25 x 0.25 degree V2.0
product_short_...	GLDAS_NOAH025_3H
product_version :	2.0

Data consolidated across all granules.

Standardized coordinates with standardized names and units.

Bounds on dimensions

Example: Zarr store for Snowf_tavg variable in GLDAS_NOAH025_3H v2.0

xarray.Dataset

▸ Dimensions: (**latitude**: 600, bounds: 2, **longitude**: 1440, **time**: 196000)

▾ Coordinates:

latitude	(latitude)	float64	-59.88 -59.62 ... 89.62 89.88
longitude	(longitude)	float32	-179.9 -179.6 ... 179.6 179.9
time	(time)	datetime64[ns]	1948-01-01T03:00:00 ... 1947-...

▾ Data variables:

latitude_bounds	(latitude, bounds)	float64	dask.array<chunksize=(18, 2), ...
longitude_boun...	(longitude, bounds)	float64	dask.array<chunksize=(36, 2), ...
time_bounds	(time, bounds)	datetime64[s]	dask.array<chunksize=(800, 2),...
variable	(latitude, longitude, time)	float32	dask.array<chunksize=(18, 36, ...
long_name :	Snow precipitation rate		
units :	kg m-2 s-1		

▸ Indexes: (3)

▾ Attributes:

Conventions :	CF-1.10
DOI :	10.5067/342OHQM9AK6Q
product_long_n...	GLDAS Noah Land Surface Model L4 3 hourly 0.25 x 0.25 degree V2.0
product_short_...	GLDAS_NOAH025_3H
product_version :	2.0

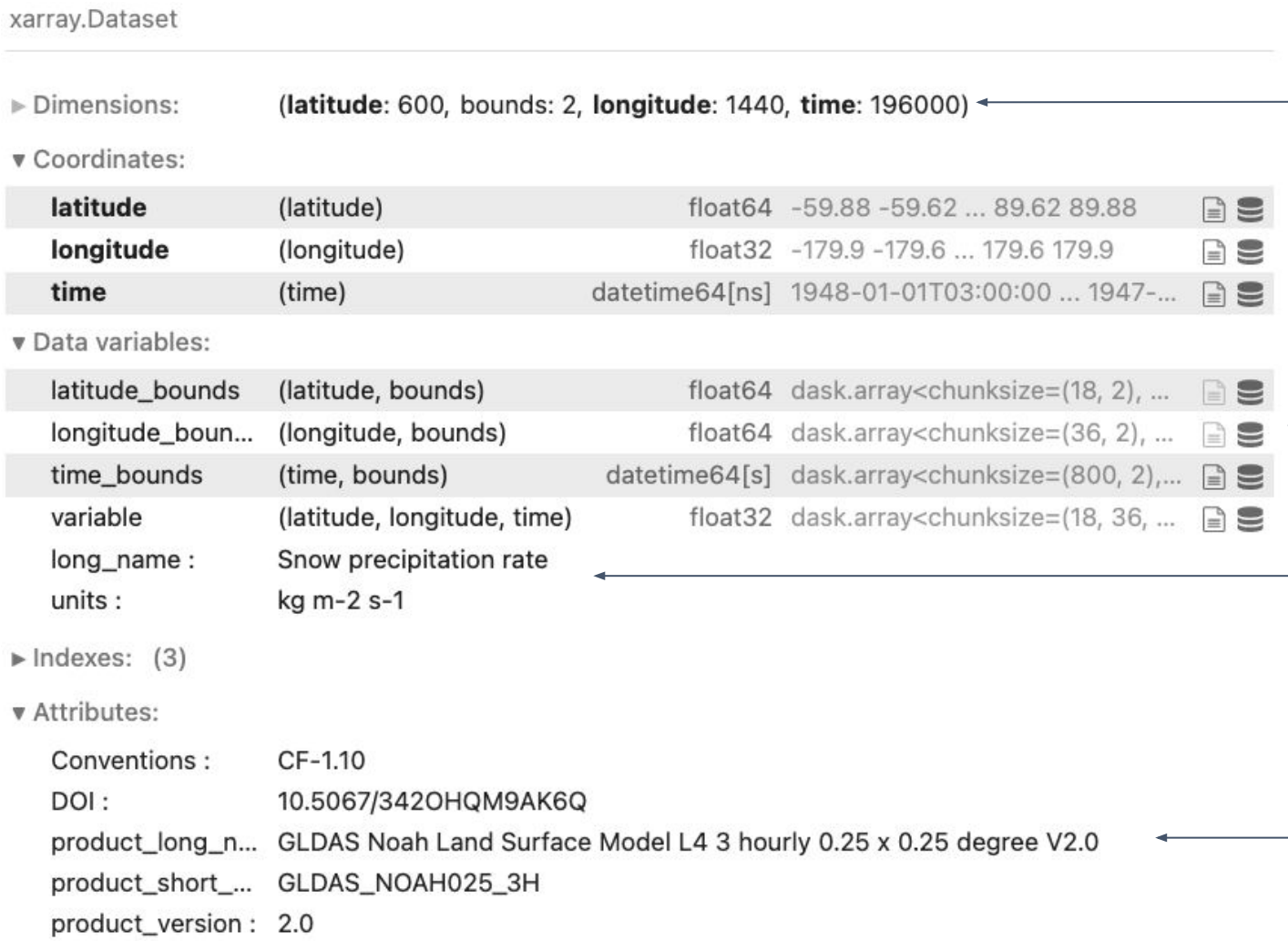
Data consolidated across all granules.

Standardized coordinates with standardized names and units.

Bounds on dimensions

A single data parameter with coordinates in a standard order

Example: Zarr store for Snowf_tavg variable in GLDAS_NOAH025_3H v2.0



Data consolidated across all granules.

Standardized coordinates with standardized names and units.

Bounds on dimensions

A single data parameter with coordinates in a standard order

Metadata tying zarr store to source data product

Chunking dilemmas

Chunking dilemmas

Chunking dilemmas

vs.

Chunking dilemmas

vs.

Chunking dilemmas

vs.

vs.

Chunking dilemmas

vs.

vs.

Chunking dilemmas

vs.

vs.

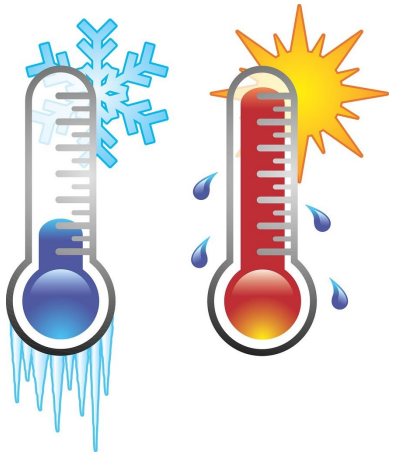
Chunking dilemmas

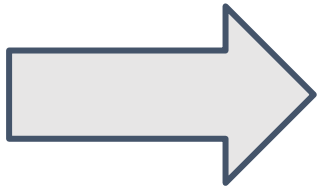
vs.

Compromise

vs.

Demo time!





Demo time!

~ 40 (on prem with pre-downloaded NetCDF files) → ~ 1 second (in AWS with zarr)

Documentation

[How to Search and Access Giovanni Variable Zarr Stores]

Known limitations

Known limitations

- AWS credentials expire after an hour

Known limitations

- AWS credentials expire after an hour
- Data must be accessed from AWS US-West-2

Known limitations

- AWS credentials expire after an hour
- Data must be accessed from AWS US-West-2
- Data discovery is clunky

Known limitations

- AWS credentials expire after an hour
- Data must be accessed from AWS US-West-2
- Data discovery is clunky
- Metadata in attributes is imperfect and may not work with your preferred tool, especially if we are unfamiliar with your preferred tool!

Known limitations

- AWS credentials expire after an hour
- Data must be accessed from AWS US-West-2
- Data discovery is clunky
- Metadata in attributes is imperfect and may not work with your preferred tool, especially if we are unfamiliar with your preferred tool!
- Dimensions need to be re-chunked to improve loading speed with tools like xarray.

Known limitations

- AWS credentials expire after an hour
- Data must be accessed from AWS US-West-2
- Data discovery is clunky
- Metadata in attributes is imperfect and may not work with your preferred tool, especially if we are unfamiliar with your preferred tool!
- Dimensions need to be re-chunked to improve loading speed with tools like xarray.
- Dimensions have fill values, which some libraries do not like. You may need to mask these fill values before plotting.

Future plans

Future plans

- Improve chunking of dimensions to make loading zarr stores with xarray faster

Future plans

- Improve chunking of dimensions to make loading zarr stores with xarray faster
- Make more data public

Future plans

- Improve chunking of dimensions to make loading zarr stores with xarray faster
- Make more data public
- Improve metadata

Future plans

- Improve chunking of dimensions to make loading zarr stores with xarray faster
- Make more data public
- Improve metadata
- Improve discoverability

Future plans

- Improve chunking of dimensions to make loading zarr stores with xarray faster
- Make more data public
- Improve metadata
- Improve discoverability
- Deal with fill values in dimensions

Future plans

- Improve chunking of dimensions to make loading zarr stores with xarray faster
- Make more data public
- Improve metadata
- Improve discoverability
- Deal with fill values in dimensions
- Incorporate community feedback into zarr stores

Tell us what you think!

gsfc-dl-help-disc@mail.nasa.gov

Acknowledgements

- GES DISC Team
- Openscapes
- Luis López, National Snow and Ice Data Center (NSIDC)
- Earth Science Data and Information System (ESDIS) Search and Discovery Team
- Imagery:
  - Mae Mu's pancakes
  - Benson Kua's scones
  - Garry Knight's churros
  - rawpixel's mountains
  - chanellelloyd1's river
  - picryl's thermometers