Zarr access via NetCDF

A checkpoint on the implementation

Manuel Reis (DKRZ)

DKRZ
DEUTSCHES KLIMARECHENZENTRUM

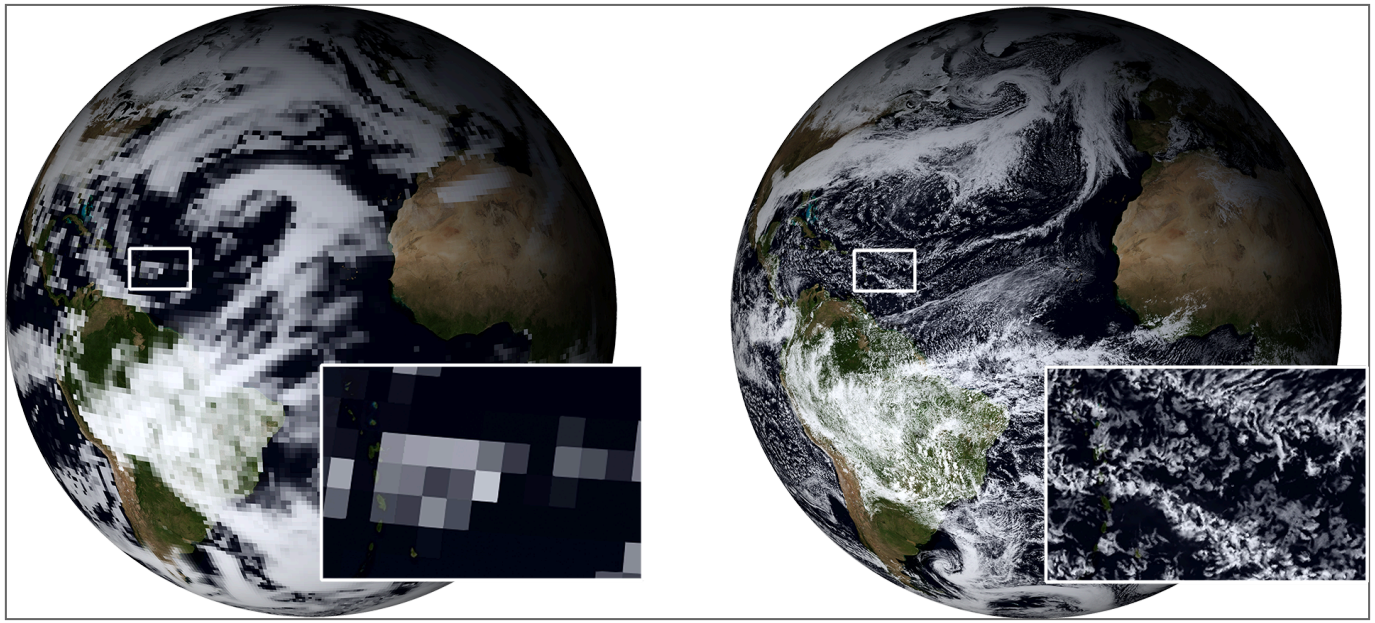
Deutsches Klimarechenzentrum

- National service facility
- Partner for climate research
- HPC
- Data storage
- Services

https://www.dkrz.de/en

DKRZ DEUTSCHES KLIMARECHENZENTRUM

Earth System Modelling[1]



MPI-ESM CMIP HR vs ICON 2.5 km

(data increase)^3 challenges ways to analyze data

1. *source:* https://flo.gitlab-pages.dkrz.de/data-workflows-short/

Data formats

- **NetCDF**: well-established format/API in scientific computing
- Many tools rely on the NetCDF interface
- `netcdf-python` builds on top of it
- **Zarr**: cloud-native, chunked, compressed array format
- Popular for big data (remote access)
- Mostly driven by `zarr-python`
- Goal: Leverage Zarr through the familiar **NetCDF API**

DKRZ DEUTSCHES KLIMARECHENZENTRUM

Why Zarr?

- Scales naturally with data size
- Chunks (& Shards)
- Hierarchy reflected on the storage *“namespace”*
- Distributed I/O patterns
- Widely adopted and well supported in Python
- *Cutting things is easier than gluing things* [1]

1. https://flo.gitlab-pages.dkrz.de/data-workflows-short/#/why-a-single-dataset

DKRZ DEUTSCHES KLIMARECHENZENTRUM

NetCDF4 - HDF5

NetCDF-4 format is HDF5 with **specific rules**

Important not to break backwards compatibility

NetCDF-Zarr Integration

- Provides a **unified interface** for both NetCDF and Zarr data
- **NCZarr**, `Zarr`, **`xarray`**
- Extends support to non-Python ecosystems:
- C
- Fortran

DKRZ DEUTSCHES KLIMARECHENZENTRUM

Tools directly benefiting

- `cdo` - Climate Data Operators
- `ParaView` - Visualization tool

Current Implementation State

- Reading and writing Zarr via NetCDF-4-style API
- On local filesystems and S3[1]
- Limitations:
  - Not optimized for consolidated datasets
  - Zarr3 not addressed yet
  - Many corner cases not covered

1. Implementations derived from **H5FDs3comms**

DKRZ DEUTSCHES KLIMARECHENZENTRUM

Caveats

There are some problems loading valid datasets:

- Produced by other **implementations**
- *“Decorated”* with value-less `attributes`/unexpected `types`
- Misleading/general error messages

DKRZ DEUTSCHES KLIMARECHENZENTRUM

I/O Improvements (remote access)

- Leveraging consolidated metadata
- **Monkey patch** shows up to 37x speedup[1]
- Increase concurrency
- Especially for remote access (`libcurl-multi`)
- Support data stream
- Currently depends on `Content-Length` (allocate buffers)
- Mainly because of HTTP range requests

1. Mostly for `open`

Proof of concept

Zarr as a “protocol”

Zarr can be used as a protocol when datasets are consolidated:

- Only `read` or `write` (no listing/walking needed)
- When data is public, a single HTTP layer can support reading from different services (`S3`, `Swift`, `Apache`)
- Same goes for other protocols supported by `libcurl`

DKRZ DEUTSCHES KLIMARECHENZENTRUM

Zarr as a “protocol” (2)

```
ncdump -h https://s3.eu-dkrz-1.dkrz.cloud/wrcp-hackathon/
data/ICON/d3hp003.zarr/PT6H_mean_z9_atm#mode=zarr
```

```
netcdf PT6H_mean_z9_atm {
dimensions:
    time = 1700 ;
    cell = 3145728 ;
    crs = 1 ;
    soil_level = 5 ;
variables:
    float clivi(time, cell) ;
        clivi:_FillValue = NaNf ;
        clivi:grid_mapping = "crs" ;
        clivi:hiopy\:\:enable = 1UB ;
        clivi:hiopy\:\:nnn = 4 ;
        clivi:hiopy\:\:time_method = "mean" ;
        clivi:long_name = "cloud ice path" ;
        clivi:short_name = "" ;
        clivi:standard_name = "atmosphere_mass_content_of_cloud_ice" ;
        clivi:units = "kg m-2" ;
...
```

Here’s the source code: **https://github.com/mannreis/netcdf-c/tree/zarr-http**

DKRZ DEUTSCHES KLIMARECHENZENTRUM

Zarr as a “protocol” (3)

```
ncdump -h https://swift.dkrz.de/v1/dkrz_948e7d4bbfbb445fbff5315fc433e36a/
hera5/v2/hera5_P1D_hpz7.zarr#mode=zarr
```

```
netcdf hera5_P1D_hpz7 {
dimensions:
    time = 4991 ;
    cell = 196608 ;
    level = 29 ;
    crs = 1 ;
variables:
    float \100u(time, cell) ;
        \100u:_FillValue = NaNf ;
        \100u:coordinates = "latitude longitude" ;
        \100u:levtype = "surface" ;
        \100u:long_name = "100 metre U wind component" ;
        \100u:standard_name = "" ;
        \100u:type = "analysis" ;
        \100u:units = "m s**-1" ;
    float \100v(time, cell) ;
        \100v:_FillValue = NaNf ;
        \100v:coordinates = "latitude longitude" ;
        \100v:levtype = "surface" ;
        \100v:long_name = "100 metre V wind component" ;
        \100v:standard_name = "" ;
        \100v:type = "analysis" ;
        \100v:units = "m s**-1" ;
...
```

DKRZ DEUTSCHES KLIMARECHENZENTRUM

Zarr as a “protocol” (4)

```
ncdump -h https://eerie.cloud.dkrz.de/datasets/
nextgems.IFS_2.8-FESOM_5-production.2D_hourly_healpix128/zarr#mode=zarr
```

```
netcdf zarr {
dimensions:
        time = 10201 ;
        value = 196608 ;
        crs = 1 ;
variables:
        float \100si(time, value) ;
                \100si:_FillValue = NaNf ;
                \100si:paramId = 228249 ;
                \100si:dataType = "fc" ;
                \100si:numberOfPoints = 196608 ;
                \100si:typeOfLevel = "heightAboveGround" ;
                \100si:stepUnits = 1 ;
                \100si:stepType = "instant" ;
                \100si:gridType = "healpix" ;
                \100si:shortName = "100si" ;
                \100si:units = "m s**-1" ;
                \100si:name = "100 metre wind speed" ;
                \100si:cfVarName = "si100" ;
                \100si:missingValue = 9999 ;
                \100si:NV = 0 ;
                \100si:gridDefinitionDescription = "150" ;
                \100si:grid_mapping = "crs" ;
...
```

DKRZ DEUTSCHES KLIMARECHENZENTRUM

Zarr as a “protocol” (5)

```
ncdump -h -n test-ipfs
ipfs://bafybeiadmnra665v3yflqz7ekjq3sgzt2bpb2ytz4dsu34ggf3gxd2nn5m#mode=zarr,ipfs
```

```
netcdf test-ipfs {
dimensions:
        TIME = 9372100 ;
variables:
        float ABSHUM(TIME) ;
                ABSHUM:_FillValue = NaNf ;
                ABSHUM:long_name = "Absolute Humidity" ;
                ABSHUM:standard_name = " " ;
                ABSHUM:strptime_format = "float" ;
                ABSHUM:units = "g/m^3" ;
        float ALPHA(TIME) ;
                ALPHA:_FillValue = NaNf ;
                ALPHA:long_name = "Angle of Attack" ;
                ALPHA:standard_name = " " ;
                ALPHA:strptime_format = "float" ;
                ALPHA:units = "deg" ;
        float BETA(TIME) ;
                BETA:_FillValue = NaNf ;
...
```

`libcurl-8.4.0` has **IPFS support**

Future Directions

- Explore other zarr implementations[1]
- Consider HDF5 Virtual Object Layer
- Support for remote/cloud stores (`http`, `s3`, `ipfs`)
- Trade-off maintainability vs flexibility (`aws-c-s3` vs `libcurl`)
- Uncertainty around Unidata funding [2]

1. **https://zarr.dev/implementations/**
2. **https://www.unidata.ucar.edu/blogs/news/entry/nsf-unidata-pause-in-most**

DKRZ DEUTSCHES KLIMARECHENZENTRUM

Summary

- NetCDF enables Zarr for scientific workflows
- Further improvements
- Dataset interoperability
- Remote access
- Difficult to maintain and keep up
- Scattered implementations from the growing community
- Combine efforts, involve zarr community!?

DKRZ DEUTSCHES KLIMARECHENZENTRUM

Thank You

Special thanks to:

- **Ward Fisher**
- **Dennis Heimbigner**
- **Florian Ziemen**

> **Check out this presentation about optimizing Earth System Modelling data for analysis**

DKRZ
DEUTSCHES
KLIMARECHENZENTRUM


DKRZ
DEUTSCHES
KLIMARECHENZENTRUM


DKRZ
DEUTSCHES
KLIMARECHENZENTRUM


DKRZ
DEUTSCHES
KLIMARECHENZENTRUM

Speaker notes