Kerchunk: A Brief Overview

Lucas Sterzinger, PhD

NOAA EDMW Workshop - May 14th, 2024

Disclaimer

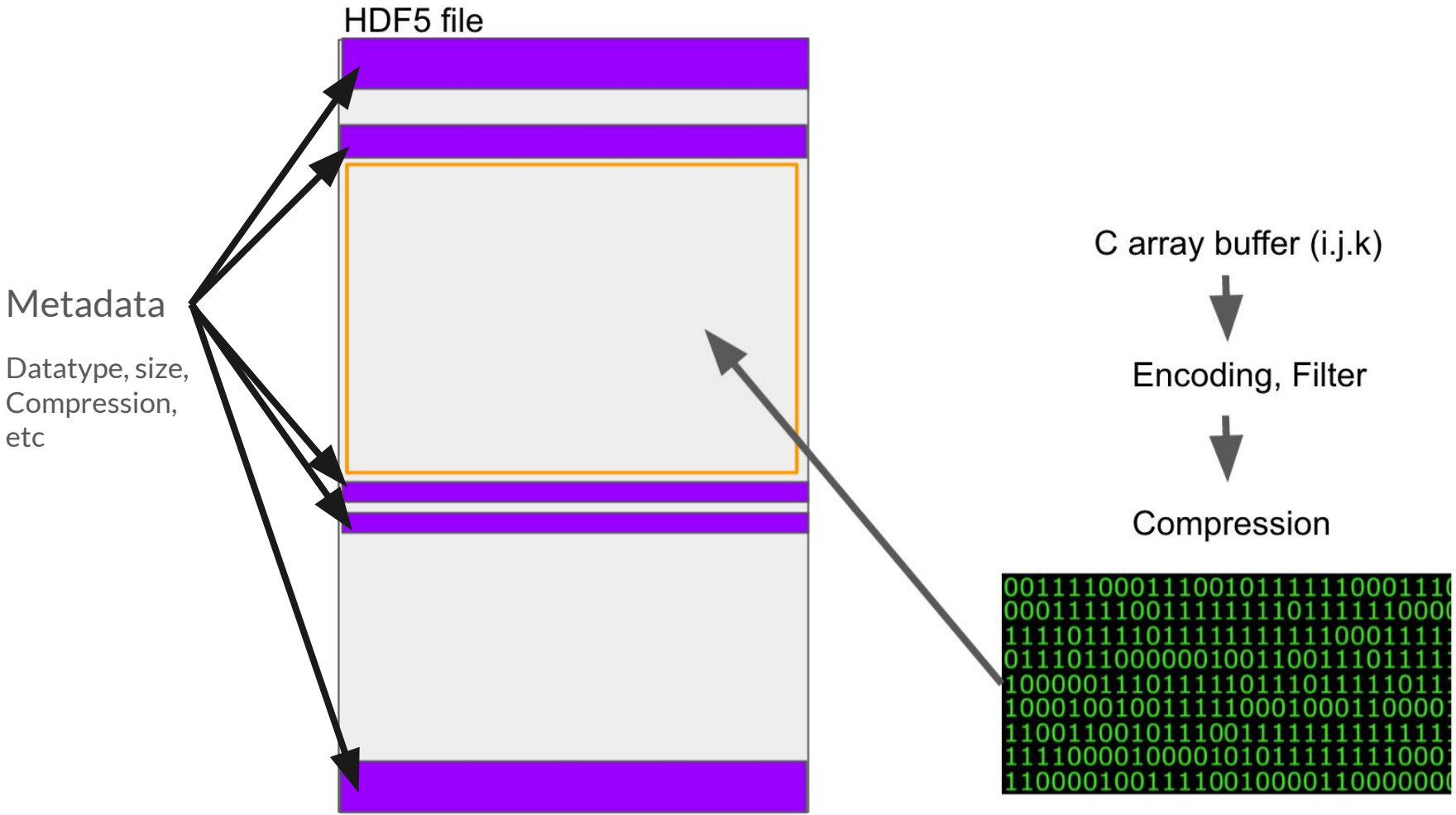
This is not related to any work I currently do at NASA - I was introduced to Kerchunk (then called `fsspec-reference-maker`) via an NCAR internship in 2021. I have tried to remain active in the community in a personal capacity ever since.

Problem description

Image Source - Pangeo/Ryan Abernathy

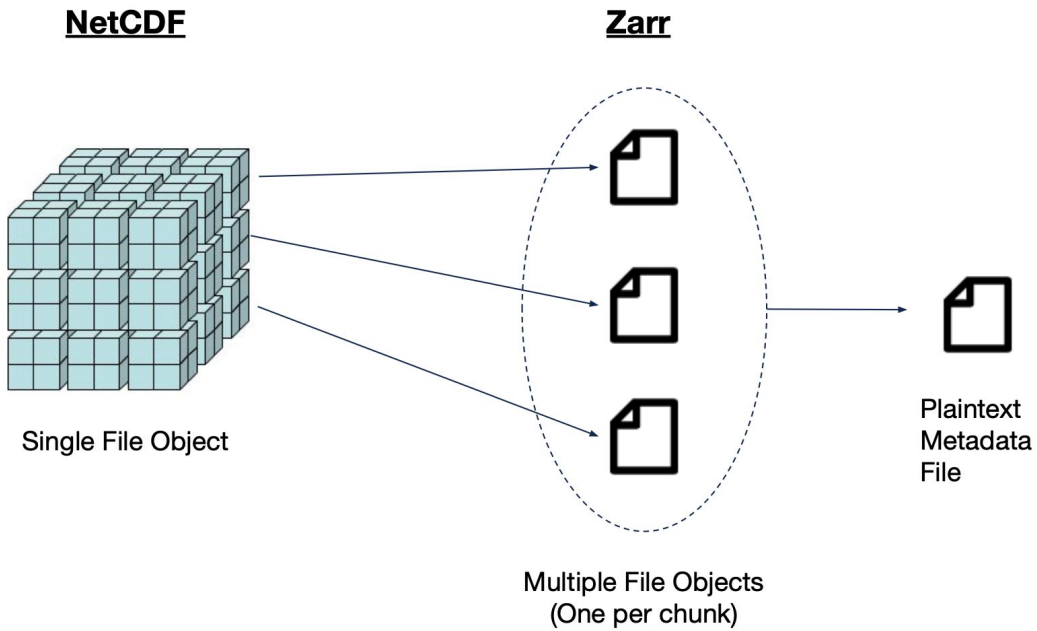
- Standard archival data formats such as HDF5, NetCDF4, GRIB, TIFF are not optimized for cloud access
  - Designed for local filesystems.
  - Metadata is often scattered throughout the file
- Datasets are often split across many individual files making access to large time spans of data more difficult.
- Services such as OPeNDAP and THREDDS exist to mitigate some of these issues, but require server side software in between the client and data

These file formats are difficult to scale to what cloud-native computing and storage can allow


HDF5 file
Metadata
Datatype, size,
Compression,
etc
C array buffer (i.j.k)
Encoding, Filter
Compression

One solution: Zarr

NetCDF vs Zarr

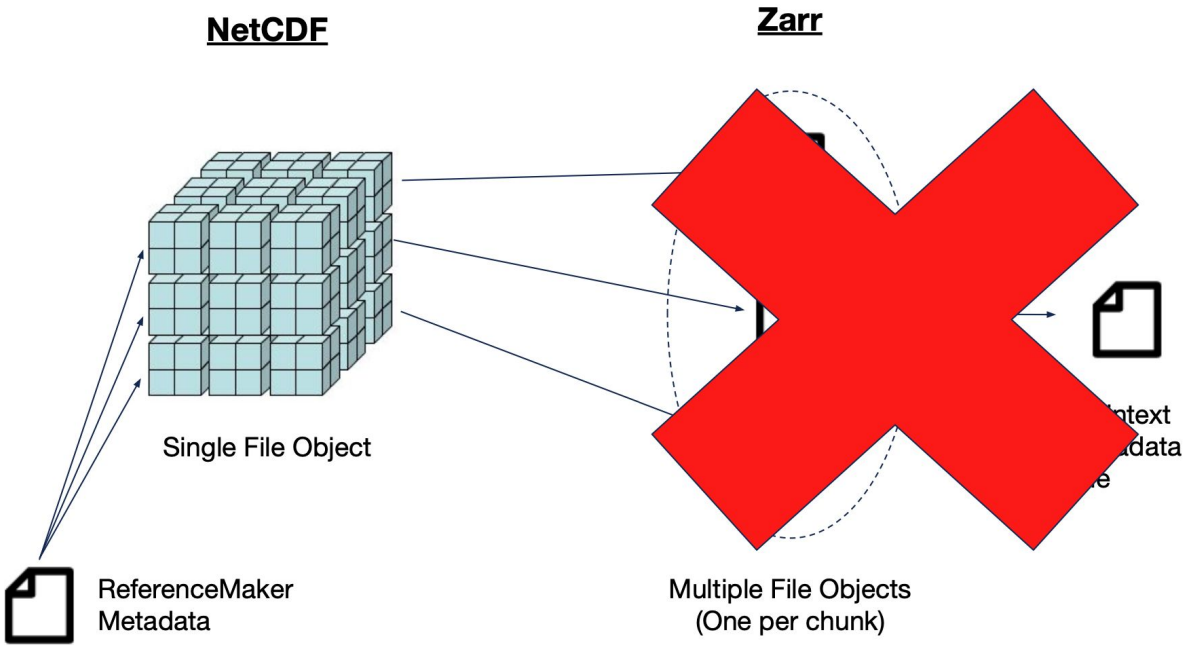


Zarr works by breaking apart data chunks into individual files, and by separating the metadata into plaintext.

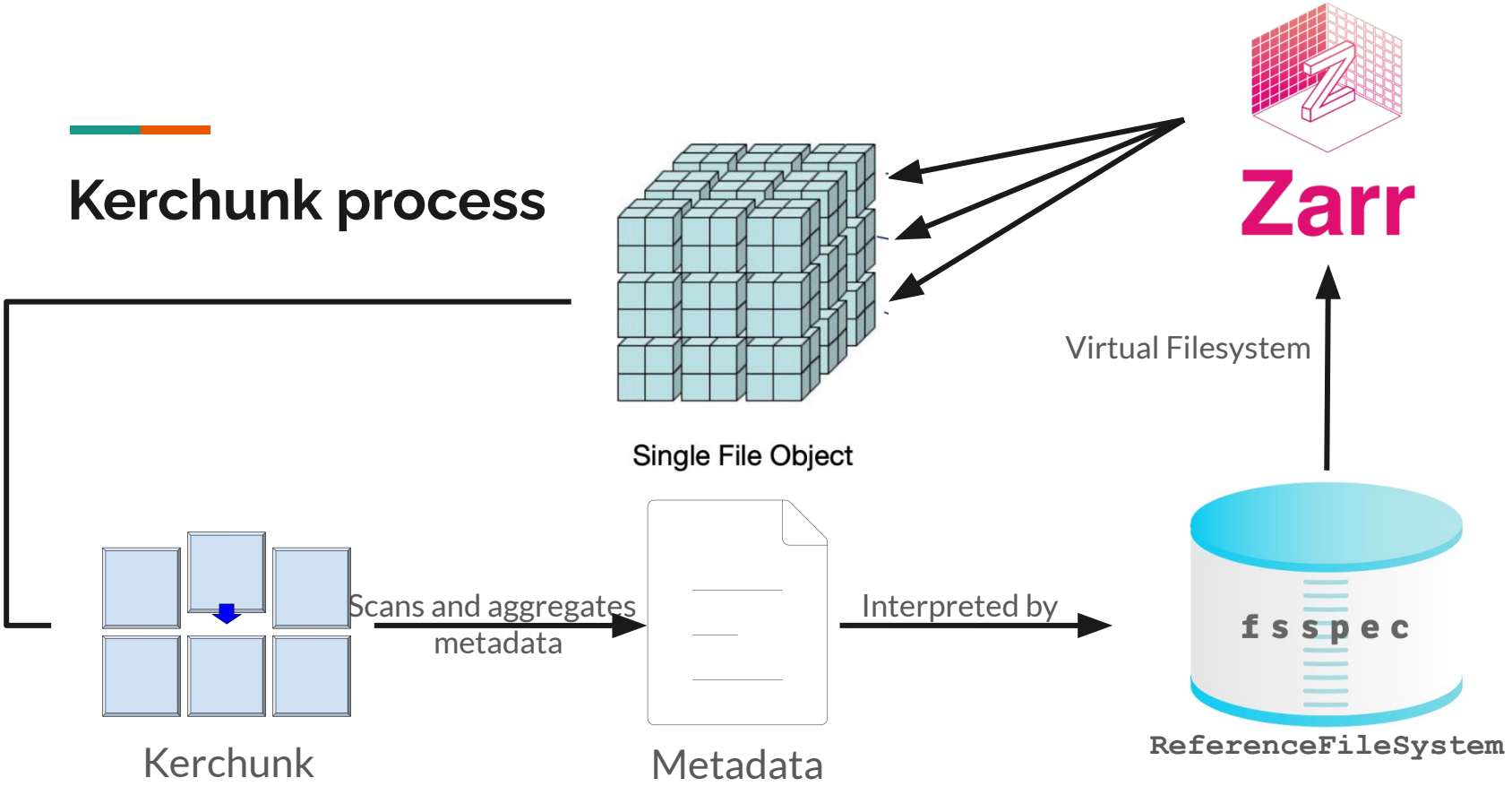
This format allows for rapid and scalable access to data in cloud object storage, but requires converting or duplicating existing data

Another solution: Kerchunk

Kerchunk extracts byte ranges of individual chunks



Kerchunk process




```
with fs.open(u) as infile:
    reference = SingleHdf5ToZarr(infile, u).translate()
```

ReferenceFileSystem allows Zarr to read chunks directly

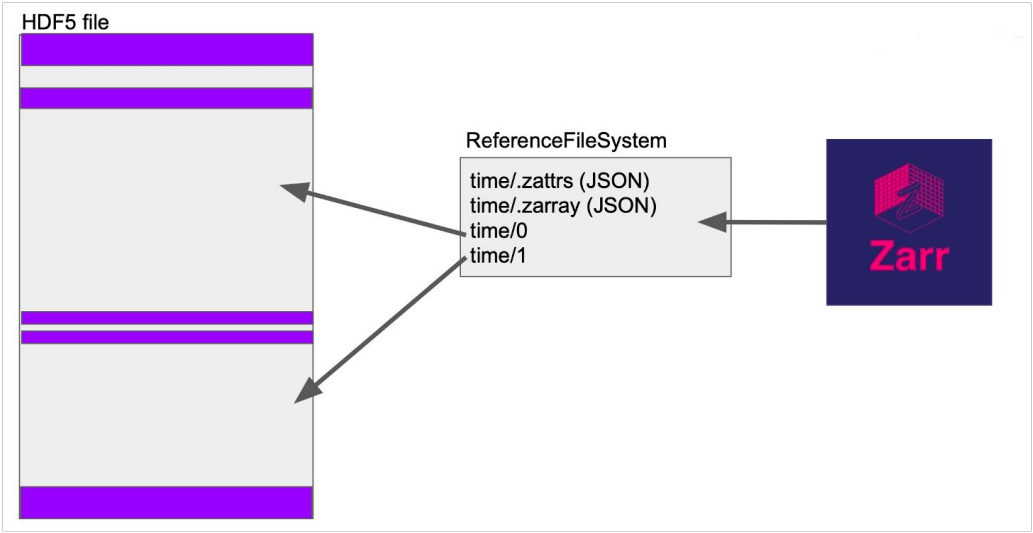
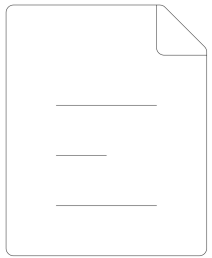


Image courtesy of Martin Durant

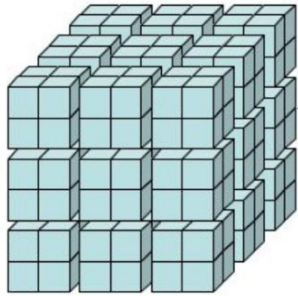
ReferenceFileSystem maps byte ranges to a virtual, Zarr-readable file system.

Zarr can read data chunks directly, bypassing the NetCDF/HDF/GRIB API entirely

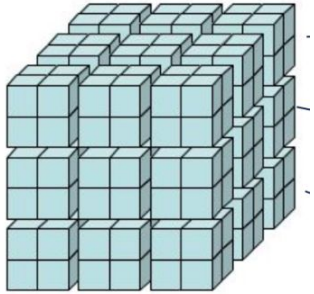
Metadata files can be consolidated, Pointing to many individual files



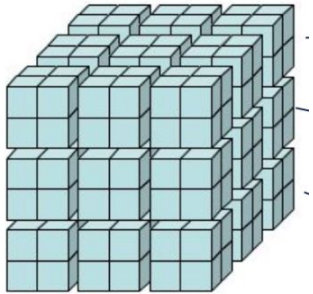
Metadata File



Single File Object



Single File Object



Single File Object

```
mzz = MultiZarrToZarr(
    './jsons/*.json',
    concat_dims='time',
    remote_options=r_opts,
    coo_map={'time' :'cf:time'},
)
```

Quick Demo

Takeaways

- Kerchunk extracts metadata and byteranges of data chunks, allowing for fast understanding + opening of of one or more files without the need to scan through the file itself.
- This extracted metadata, combined with ReferenceFileSystem, creates a virtual Zarr-readable file system. The Zarr API can then access chunks directly (translated into byte range requests by ReferenceFileSystem), opening the dataset into xarray and bypassing the NetCDF/HDF/GRIB libraries/backends altogether.
- Metadata files can be:
    - Combined to create a single reference describing many files
    - Created by anyone with access to the original archival format files
    - Shared and utilized by anyone with access to the original files

The future of Kerchunk

- Zarr v3 Storage Mapper (Kerchunk-like support within Zarr spec itself)

- VirtualiZarr - creating an xarray API for generating Kerchunk metadata

Questions?

Email: lucas.j.sterzinger@nasa.gov

Demo code:
https://github.com/lsterzinger/2024-edmw-kerchunk-demo