EGU 2026 · ESSI2.2 · MON 4 MAY · X4.95

Zarr *at scale*

virtualization, sharding, and performance optimizations for Earth science data

Max Jones[1], Joe Hamman[2], Davis Bennett, Kyle Barron[1], Justus Magin[3]
[1] Development Seed · [2] Earthmover · [3] LOPS, Univ. Brest

1 SHARDING

Many small chunks, few storage objects

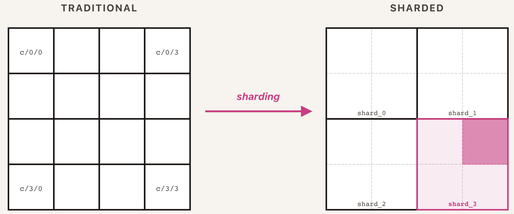
Random access of small chunks with the storage footprint of large ones.

Sharding packs thousands of chunks into a single storage object. Readers fetch only the bytes they need via byte-range requests, so random access stays fast.

- → **HPC filesystems:** escape per-file inode pressure
- → **Cloud object stores:** skip per-object request overhead
- → **Same access patterns:** readers transparently support shards

```python
import zarr

zarr.create_array(
    store="array.zarr",
    shape=(1024, 1024),
    chunks=(16, 16),    # unit of access
    shards=(256, 256),  # unit of storage
    dtype="float32",
)
```



zarr-at-scale.maxjones.dev/sharding

2 VIRTUALIZATION

Read archival files as one Zarr dataset

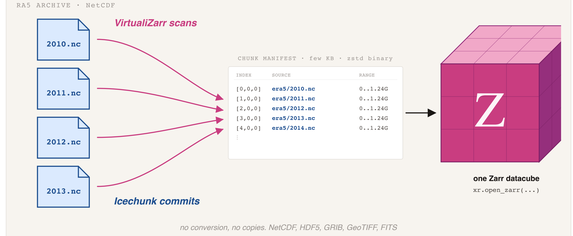
No conversion, no copies. Point at NetCDF4, HDF5, GRIB, GeoTIFF.

VirtualiZarr scans existing files and builds a virtual manifest. Icechunk commits these manifests transactionally. xarray reads across the whole archive as a single Zarr dataset.

- → **No conversion:** expose terabytes of NetCDF/HDF5 without rewriting
- → **Atomic updates:** safe under concurrent writes via Icechunk
- → **Format-agnostic:** NetCDF4, HDF5, GRIB, GeoTIFF, FITS

```python
import virtualizarr as vz
import xarray as xr

vds = vz.open_virtual_mfdataset(
    urls, registry=registry,
    parser=vz.parsers.HDFParser(),
    combine="by_coords",
)
vds.virtualize.to_icechunk(session.store)
session.commit("CMIP6 archive")
ds = xr.open_zarr(session.store)
```



zarr-at-scale.maxjones.dev/virtualization

3 VARIABLE CHUNK GRIDS

One array, many chunk sizes

Match chunk size to data density. Fine where it matters, coarse where it doesn't.

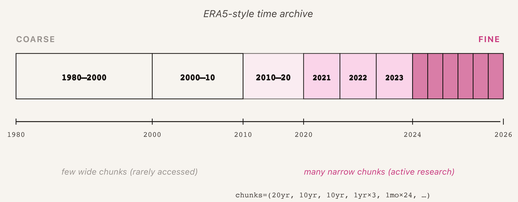
Just landed in zarr-python. A Zarr array can now have non-uniform chunks along any axis. Faster reads where activity concentrates, less storage overall.

- → **Match density:** fine chunks where data is dense, coarse elsewhere
- → **Time archives:** fine for recent observations, coarse for history
- → **Less waste:** no padding sparse regions to fit a uniform grid

```python
import zarr

zarr.config.set({"array.rectilinear_chunks": True})

zarr.create_array(
    store="timeseries.zarr",
    shape=(6760,),                          # one year, hourly
    chunks=[(4096, 2048, 1024, 512, 512, 256, 256, 56)],
    dtype="float32",
)
```



zarr-at-scale.maxjones.dev/variable-chunk-grids

4 IN-BROWSER RENDERING

Render multi-terabyte datasets in the browser

Stream Zarr chunks straight to the GPU. No server, no pre-rendered tiles.

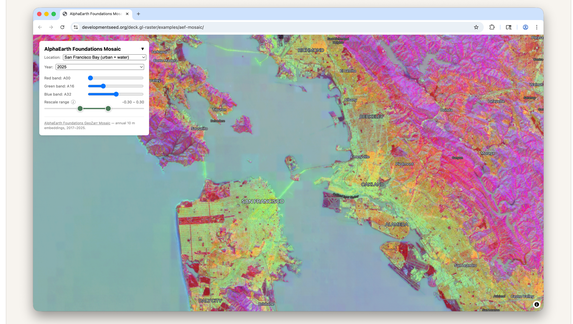
Zarrita.js reads Zarr chunks directly from a browser fetch. deck.gl-raster pushes them onto the GPU. Pan, zoom, recolor a multi-terabyte dataset interactively, on a laptop.

- → **No tile pipeline:** skip pre-render, tile cache, tile server
- → **Cloud-native:** data lives in Zarr on object storage, read directly
- → **GPU-accelerated:** pan, zoom, recolor at interactive rates

```python
import * as zarr from "zarrita"
import { ZarrLayer } from "@developmentseed/deck.gl-zarr"

const store = new zarr.FetchStore(ZARR_URL)
const root = await zarr.open.v3(store, { kind: "group" })
const arr  = await zarr.open.v3(
    root.resolve("/embeddings"), { kind: "array" })

new ZarrLayer({ node: arr, metadata: root.attrs,
                selection, getTileData, renderTile })
```



zarr-at-scale.maxjones.dev/in-browser

EARTH-SCIENCE PROFILE

GeoZarr: open conventions for geospatial Zarr

Composable conventions for CRS, spatial transforms, pyramids, and climate metadata. OGC standardization in progress, target summer 2026.

Try it: `inspect.geozarr.org` · `geozarr-toolkit` · `geozarr-examples`

proj: CRS
EPSG, WKT2, PROJJSON

spatial: Transforms
Affine, grid registration

multiscales Pyramids
Resolution levels for tiling

CF Metadata
Climate & Forecast (NetCDF compat)

GeoZarr
geozarr.org

Jones, Hamman, Bennett, Barron, Magin · EGU26-15196 · CC BY 4.0
max@developmentseed.org · doi.org/10.5194/egusphere-egu26-15196

Companion site
zarr-at-scale.maxjones.dev