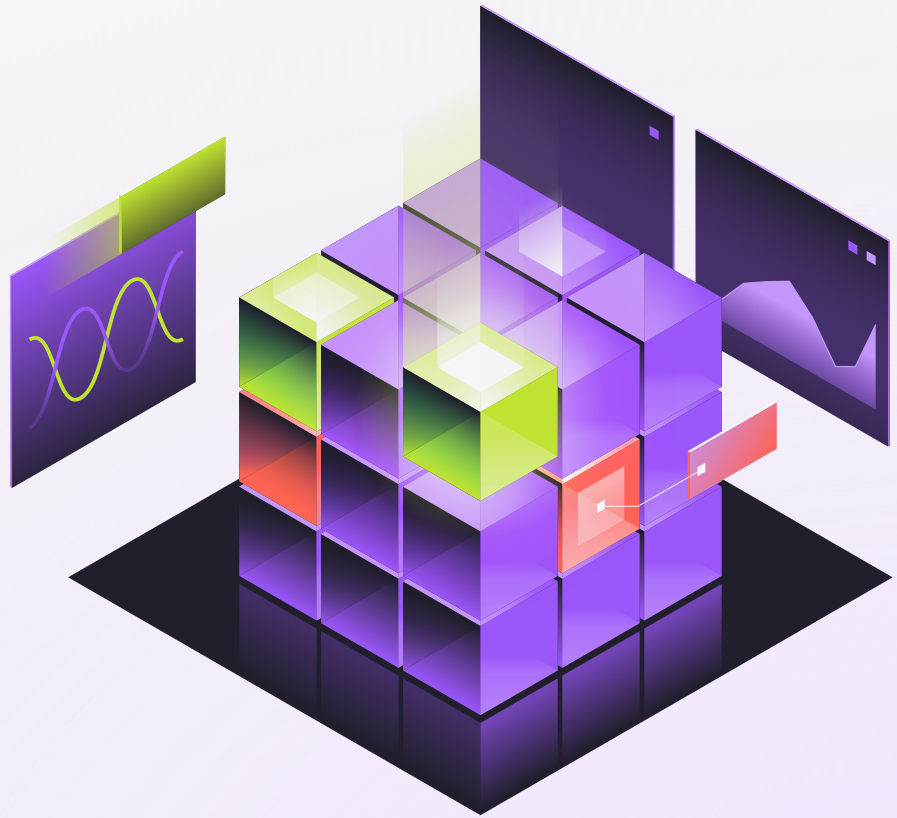
earthmover

zax: cloud-native compute for scientific data

Joe Hamman and Deepak Cherian · Earthmover

CNG 2026 · October 7, 2026



THE PROBLEM

cloud-native compute is too hard

we solved formats

- ✓ ARCO: analysis-ready, cloud-optimized
- ✓ Zarr, COG, Parquet, Iceberg, Icechunk, STAC













we haven't solved compute

THE STATUS QUO

geospatial compute has bad UX

`Xarray → Dask → NumPy → Zarr → fsspec`

- Separation of concerns is good...
- ...but the result is a stack that is tough to operate
- Users have to be experts to make it work
- Chunking: the single biggest source of *pain*
- Orchestration: single biggest source of *toil*; nothing purpose-built, so you roll your own (Airflow/Prefect/Lambda)

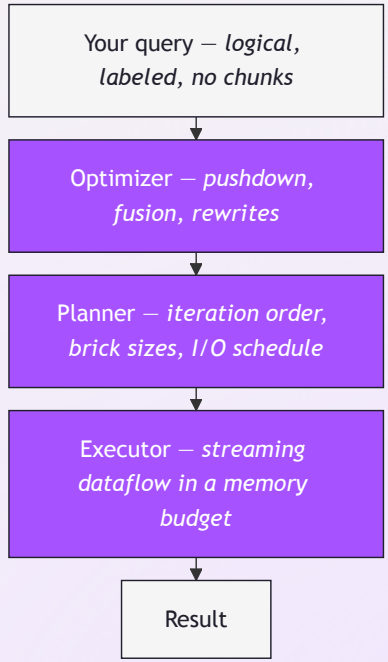
THE CLOSEST THING WE HAVE

`GDAL · NCO · CDO · DuckDB` integrated, one command, does the thing... but mostly CLIs for manipulating files, not query engines for cloud arrays. This is what users ask for.

STEALING FROM DATABASES

designing a modern query engine for arrays

- **Declarative.** Say *what* to compute over named dimensions
- **Logical level only.** No chunk sizes, ever again
- **The engine decides.** Optimized iteration order, chunk sizes, I/O scheduling
- **Predictable.** Run within a memory budget, or fail fast
- **Proven ideas.** Planners, optimizers, cost models, views

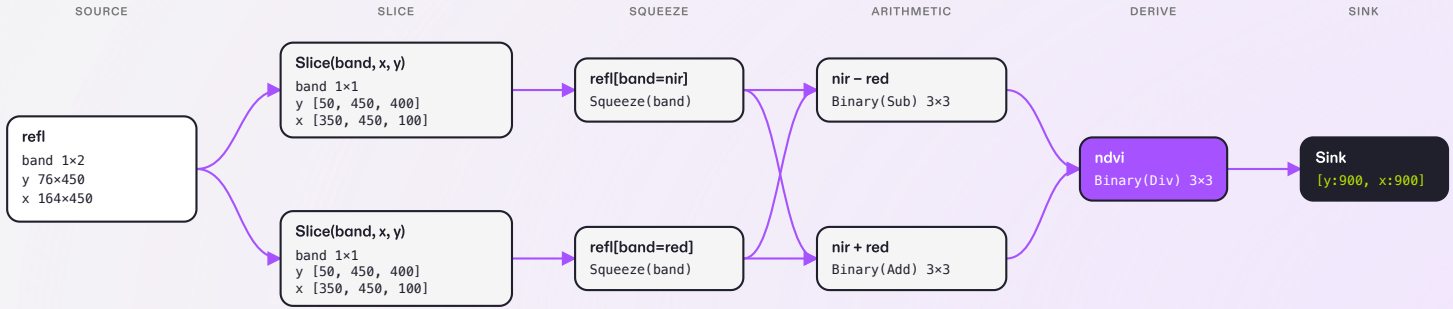


QUERY PLANNER

example ndvi query

```
let nir = field("refl").at("band", "nir");
let red = field("refl").at("band", "red");
let res = ds.derive("ndvi", (&nir - &red) / (&nir + &red));
let box_ = Selection::new()
    .islice("y", 400..1300).islice("x", 1000..1900);
res.select(box_)?
```

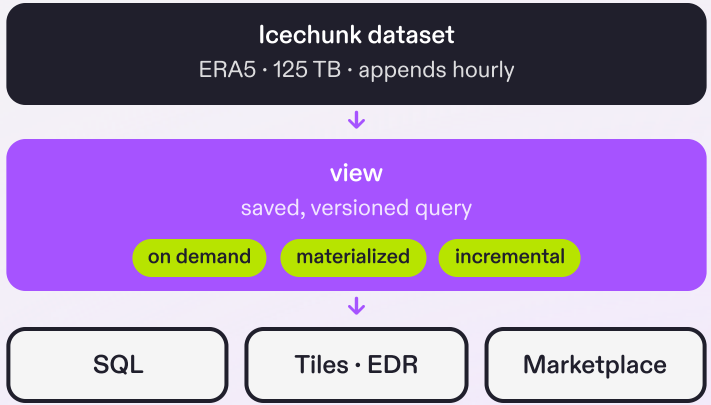
- 37.6 GiB cube: `[band: 2, y: 34200, x: 73800]`
- No chunk sizes in the query
- Selection applied last, **pushed down** into the source read
- Brick grid chosen per node, aligned to disk
- **9 of ~12,500 chunks** read



VIEWS

declarative queries you can keep

- **A saved, versioned query.** Share it, run it again on new data.
- **On demand or materialized.**
- **Incrementally updated.**
- **Treat like any dataset.** SQL, OGC Tiles, the Marketplace
- *Like a GDAL VRT, but more general purpose.*



one array-native core, several interfaces

Zax-SQL
Postgres wire · Flight SQL

Zax-OpenEO
OGC / OpenEO APIs

Zax Server
Xarray-style Python client

Zax Embedded
Rust + Python library

available today in progress

```
SELECT DATE_PART('month', valid_time) AS month,
       AVG(t2m), STDDEV(t2m)
FROM era5.'single/temporal'
WHERE longitude = 45.0 AND latitude = 60.0
GROUP BY month
```

- Any Postgres or Arrow Flight SQL client
- Runs on **125 TB** of ERA5; `WHERE` becomes a chunk selection
- Relational data model is mapped efficiently onto the Zarr data model
- Try it today on Arraylake's free tier

FELT THE PAIN OF COMPUTE IN THIS SPACE?

come find us

- earthmover.io/blog/compute-roadmap
- docs.earthmover.io/compute/sql

