Virtualizarr Coordination Meeting

Virtualizarr Coordination Meeting

Meeting URL:

https://numfocus-org.zoom.us/j/88307540201?pwd=Sn2uY0i3wRCqFaZ5oNbJRQrcg1FEyw.1

Next Meeting Agenda (November 14th)

What should we talk about at the next meeting?

-

Nov 14, 2025

Attendees:

- Tom Nicholas
- Aimee
- Sean
- Raphael Hagen
- Max

Agenda:

- VirtualiZarr template pipelines (Sean Harkins juliusbusecke@gmail.com Max Jones)
    - Sean may have developed an example for Microsoft (MET Office UK) at this point. During the meeting we hope to review this template as a group and discuss if it is a helpful approach and what other approaches or considerations should be made.
    - The goal (proposed) for the meeting should be to determine if the templating approach used for the MET Office UK may be useful for others. If yes, determine next steps for it to be used in practice to pressure test it. If no, why and what are the alternatives.
    - What other documentation for using VirtualiZarr in practice would be helpful to folks trying to create and maintain VirtualiZarr stores?

Oct 31, 2025 | VirtualiZarr Coordination Meeting

Attendees:

- Tom Nicholas
- Julia Signell

- Alex Merose / Open Athena / @alxmrs
- Betsy Cannon / Open Athena
- Ryan Williams / Open Athena
- Raphael Hagen / Carbonplan / @norlandrhagen
- Max Grover / Spire
- Gui Castelao / NREL / @castelao

Notes

- Props to max@developmentseed.org for adding virtual-tiff! 🎉

Discussion

- Rich Signell's example that's slow for interesting reasons (no coalesced reads in IC)
    -
- Alex Merose + Betsy Cannon + Ryan Williams: How can VZ support materials science datasets?
    - DFT data
    - Important DFT dataset just came out from Meta + Princeton
        - 500TB in total
        - VASP file format
            - Raster of Human-readable floats
    - Could do with some familiar-sounding features
        - Overviews
    - Currently distributed with Globus
    - Want to zarr-ify - can they avoid copying?
    - https://github.com/materialsproject/pymatgen
    - https://github.com/zarr-developers/VirtualiZarr/issues/218
- Julia: Doing a talk with Chuck Daniels at FOSS4G-NA on Tuesday

Action items

Oct 3, 2025 | VirtualiZarr Coordination Meeting

Attendees:

- Max
- Tom
- Julia
- Sean
- Julius

Notes

- Max
    - Quarterly planning - prioritizing simplification of virtual chunk credential authorization
- Sean

        - Benchmark HRRR virtual zarr dataset vs. actual zarr dataset from dynamic
            - Needed to fix coordinates
            - Initial tests how them as the same; would be different co-located
                - Dynamic - us-west-2
                - Raw - us-east-1
                - Virtual - us-east-1
- Julius
    - PR with additional kwarg for timedelta
        - Looking for feedback on typing approach
- Tom
    - Failing on main (due to zarr change)
    - Starting on inlining array
- Julia
    - Used DMR++ ICE-Sat 2 files, found them weird
    - Used virtual tiff with sentinel-2 COGs
        - Inspired by Emmanuels' registered attribute PR

Discussion

- Simplifying chunk authorization
    - Need a simple way to get all prefixes from a IC store
    - Need a way to simply authorize all chunks
    - Could use earthaccess as a wrapper for IC
    - Would providing an option for anonymous access for Icechunk help?

Action items

- [ ]

Sep 19, 2025

Attendees

- Tom
- Max
- Julia
- Julius
- Raphael

Updates

- Tom
    - Raised https://github.com/zarr-developers/VirtualiZarr/issues/799
- Raphael
    - No VZ updates, just using it in ✨production
- Julius

    - Mostly lurking today. Recent virtualization scripts seem pretty useable/reprocible (https://github.com/virtual-zarr/rasi-icechunk)
- Julia
    - Still refining virtual icechunk stores in STAC. https://github.com/NASA-IMPACT/veda-docs/pull/253 is the latest version
    - The creating STAC items from just the virtual icechunk thing kind of worked
    - I added a reader to xpystac so you can open them easily with the ‘stac’ backend
- Max
    - Just thinking about how https://github.com/zarr-developers/zarr-extensions/pull/21 impacts to VirtualiZarr
    - https://github.com/NASA-IMPACT/dse-virtual-zarr-workshop taught a workshop with Julia
- Sean
    - Worked on region writes to support HRRR.
    - Will do some benchmarking against native chunks in HRRR.

Agenda

- Store arrays in-memory in ManifestStore
    - https://github.com/zarr-developers/VirtualiZarr/issues/799
- We should show documentation/examples for manually setting virtual references as an alternative to xr.to_zarr(..., region=”...”)
- Testing module
    - Checking you’re not propagating metadata from one file erroneously to the entire datacube
    - Dissuade all_close on virtual datasets because it’d load everything

Sep 5, 2025

Updates:

- Tom
    - Thinking about virtual chunks in array lake, probably limit to public data
        - Max: anon buckets with requester-pays support is all most people probably need
- Raphael
    - Updating example docs to the V2 syntax.
        - Could use as inspiration for reducing boiler plate
    - Nice real world use case Virtualizing CESM CMIP runs and then writing real icechunk.

- Julia
    - Working on workshop on representation of virtual zarr in STAC
- Sean

        - Trade off between safety and usability in Icechunk virtual containers has made work a bit tricky
            - https://github.com/earth-mover/icechunk/issues/1184
        - Working on HRRR pipelines
- Max
    - Made a release
    - Inline support

Other issues

- Direct loading of manifest store is undocumented
- What does inline support mean for manifeststore.to_icechunk / to_kerchunk.
    - Documentation would explain parsers and stores first

Aug 22, 2025

Updates:

- Tom
    - Fixed problem with invalid chunk containers by duplicating some IC logic in VZ
    - Released new version with several important bugfixes
    - Currently working on ensuring virtual IC datasets work in Arraylake
    - Submitted VZ + IC + AL abstract to AMS
- Julius
    - Couple of new datasets
        - NASA RASI (https://github.com/virtual-zarr/rasi-icechunk) – very early WIP
        - NASA NLDAS3 (https://github.com/virtual-zarr/nldas-icechunk)
            - This was the dataset which exposed that gnarly bug.
        - Overall v2 is working great for me!
        - Oh the icechunk docs need an update!
- Max
    - Workshop next week - https://github.com/NASA-IMPACT/dse-virtual-zarr-workshop (based on ESIP tutorials)
    - Would like to add external tutorials/presentations to the docs
        - https://www.earthdata.nasa.gov/learn/webinars/accelerating-science-using-virtualized-data-po.daac
        - https://nasa.github.io/ASDC_Data_and_User_Services/PREFIRE/how_to_generate_and_read_virtual_datacube-for-PREFIRE.html
    - Working on Zarr summit
    - Thinking on titiler + Icechunk

- Raphael
    - No contribution updates, but real virtualizarr use case.
        - Ran into an (xarray?) bug: https://github.com/zarr-developers/VirtualiZarr/issues/785
    - Who's going to SatCamp?!
- Julia
    - Nothing really to report. I do have a terminology question though. Do we say "virtual icechunk store" is that a subset of "virtual zarr store"?
    - Should virtual zarr stores be cf-compliant?

Agenda
- Icechunk 2.0?
- What do we want IC's API to look like?
    - https://github.com/earth-mover/icechunk/issues/1184
- ZEP8?
- Julia's STAC examples https://github.com/NASA-IMPACT/dse-virtual-zarr-workshop/pull/8

Questions from Julia's presentation
- Where is the standard for media types and roles? Is the zarr extra file registered?
    - Zarr extension in github.com/stac-extensions/zarr
    - Xaray-assets is an extension that is very xarray specific, but really you should be storing information about the zarr store
- What tool would you use to load the virtual Icechunk store into Xarray?
- How does versioning work?

Action items
- Diagram showing relationship between Virtual Zarr, Icechunk Store, Kerchunk reference file

Jul 25, 2025

Tom and Max are skipping this week due to time-zones / travel

Jul 14, 2025

Attendees
- Tom
- Raphael
- Julia
- Sean

- Julius
- Max

Agenda
- Updates
- Release plan
    - Max wants to release 2.0 this week in time for ESIP next week

Jun 27, 2025

Attendees
- Tom
- Raphael
- Julia
- Chuck
- Max

Agenda
- Updates
    - Tom
        - Post-parser docs updates
        - Various fixes
        - Trying to work out what is feasible to do before SciPy presentation (July 10th)
    - Max
        - No virtualizarr time last week :(will be back next week
    - Raphael
        - [docs] migration guide - possibly outdated with other docs changes
- How close to release?
    - Data types changes have been merged but not released
    - Bug in indexing caused by an xarray change
    - Do we want to warn about behaviour changes?
    - Change parser signature to accept registry
    - Sean to fish out his memorystore caching trick
- Demo at scale?

Jun 13, 2025

Attendees
- Tom
- Sean
- Raphael
- Aimee

- Julia
- Chuck Daniels / Development Seed / @chuckwondo
- Max

Agenda
- Updates
  - [Raphael]
    - No updates, planned Migration guide work with Max this coming Monday.
  - [Sean]
    - Out for the last two weeks
  - [Chuck]
    - Branch (big parser refactor) almost done. Sean is going to have another review. Coordinated w/Max on the issues below to fix build failures that were occurring for Refactor codebase to support a new simplified Parser->ManifestStore model.
  - [Max]
    - Support ManifestArray expansion via None in the indexer
    - Update type-ignore coverage
    - Use numpydoc pre-commit validation

May 30, 2025

Attendees
- Tom
- Sean
- Raphael
- Julia
- Chuck Daniels / Development Seed / @chuckwondo
- Max

Agenda
- Updates
  - Tom
    - Wrote a blog post
    - Didn't do anything on VirtualiZarr itself
  - Sean
    - Almost finished V2 refactoring PR
  - Julia
    - Also blog post
    - Been thinking more about numcodecs - feel like they really need to not be written in Python - https://github.com/juntyr/numcodecs-rs

        - Seems like COGs need a codec that seems like it hasn't been added yet: https://github.com/zarr-developers/numcodecs/issues/583 so I am going to try to open a PR for that
            - The virtual_tiff codecs provides a more COG specific version of this - https://github.com/virtual-zarr/virtual-tiff/blob/main/src/virtual_tiff/codecs.py
            - Most important to register as a formal codec extension in https://github.com/zarr-developers/zarr-extensions/tree/main/codecs
            - Chuck and my VirtualiZarr talk was accepted at FOSS4G-NA
    - Max
        - Registering imagecodecs in zarr-extensions starting with LZW
        - Will update mkdocs PR after refactor is merged
        - Building out more TIFF support - https://virtual-tiff.readthedocs.io/en/latest/#tiff-structure-support
        - Creating a source cooperative repository will example files for TIFF, NetCDF, GRIB
- Agenda
    - Refactor test suite
    - What do we anticipate for the V2 release timeline?
        - Before end of June
    - Lithops
        - https://github.com/lithops-cloud/lithops/issues/1429
    - VirtualiZarr as a runtime translation layer for already-cloud-optimized data?
        - https://github.com/zarr-developers/VirtualiZarr/issues/603
        - Or is this just a special codec for native zarr?
    - Can we virtualize ZIP?

May 16, 2025

Attendees
- Tom
- Sean
- Raphael
- Max
- Chuck Daniels / Development Seed / @chuckwondo

Agenda
- Updates
    - Tom
        - Wrote new pages of documentation
            - Custom readers ("parsers")
            - Scaling

        - Max
            - Virtual TIFF
                - Ready to integrate with VirtualiZarr
        - Raphael
            - In the desert
        - Sean
            - Refactoring readers to use ManifestStore
- How to configure parsers in the 2.0 API https://github.com/virtual-zarr/virtual-tiff/pull/31#discussion_r2092223768
    - Parser definition doesn't need `**kwargs`
- Revisiting ObjectStore vs. Obspec type hints
    - Just type as ObjectStore for now
- Do we need an ObjectStoreRegistry?
    - Motivations
        - DMR++ loadable variables
        -
        - Loading kerchunk data without fsspec or icechunk
    - New constructor method that creates a ManifestStore from one store
- Manifest object names https://github.com/zarr-developers/VirtualiZarr/pull/568#discussion_r2084882074
    - Let's keep Manifest for storage objects and parser instead of backend - rename ChunkManifest to Manifest
- Pass through args between open_virtual_mfdataset and open_virtual_dataset https://github.com/zarr-developers/VirtualiZarr/pull/568#discussion_r2091896659
    - Don't want to stray far from xarray, even if it would technically be better, because familiarity is nice
- Drop variables - https://github.com/zarr-developers/VirtualiZarr/pull/568/files#r2093364604
    - `drop_variable` should pass through to `ManifestStore.to_virtual_dataset()`
    - Individual parser implementations can have their own ways to drop or skip variables

Apr 23, 2025

- Tom Nicholas / Earthmover / @TomNicholas
- Raphael Hagen / CarbonPlan / @norlandrhagen
- Max Jones / Development Seed / @maxrjones
- Sean Harkins / Development Seed / @sharkinsspatial

Agenda

- Updates
    - Tom
        - Focused on CNG talk
        - Deprioritized virtualizarr release
        - Running on lambdas with docker

        - Resolved issues accessing data from source coop
        - Showed that new ManifestStore-based readers are faster
        - Reviewed new SafeTensors reader that Nick from Quansight wrote
        - Working on Zarr tutorial
    - Raphael
        - Focused on CNG talk on GeoParquet
        - Merged Zarr refactor
    - Sean
        - ManifestStore merged into HDF5, it seems like its working!
        - Interested in refactor of codecs to use Zarr V3 syntax directly
        - Interested in imagecodecs wrappers
    - Max
        - https://github.com/maxrjones/virtual-tiff/pull/12
        - Will put in a docs update today for ManifestStore
- Questions
    - Are we blocked on Zarr reader?
        - Just for V2 data but we can merge with only V3 support

Action items

- Tom continues on!
- Max & Raphael address Zarr reader bug
- Max opens a PR for docs on Obstore + ManifestStore
- Sean & Max attend Friday's Zarr python and propose codecs discussion

Apr 18, 2025

- Tom Nicholas / Earthmover / @TomNicholas
- Raphael Hagen / CarbonPlan / @norlandrhagen
- Aimee Barciauskas / Development Seed / @abarciauskas-bgse
- Julia Signell / Element84 / @jsignell
- Devin Rand / Berkeley Earth / @devinrand
- Max Jones / Development Seed / @maxrjones
- Sean Harkins / Development Seed / @sharkinsspatial
- Amit Kapadia
- Alex Merose / OpenAthena / @alxmrs

Agenda

- Updates
    - Tom
        - More refactoring around ManifestStore

        - Blocked by needing Max's PR for generating default object stores
            - Just merged https://github.com/zarr-developers/VirtualiZarr/pull/549!
    - battling docker to deploy lithops at scale
- Sean
    - Will merge https://github.com/zarr-developers/VirtualiZarr/pull/542 after Max's PR is merged.
    - TODO update HDF reader to use direct Zarr v3 codec registry https://github.com/NASA-IMPACT/veda-odd/issues/150
    -

- Raphael
    - Zarr-reader related codec bug/fix/hack?
- Alex
    - Kerchunk may support the OpenMeteo format soon, meaning there may be some cool weather datasets that we could virtualize!
        - https://github.com/fsspec/kerchunk/issues/464#issuecomment-2809977232
        - https://github.com/open-meteo/open-data
- Max
    - Just merged https://github.com/zarr-developers/VirtualiZarr/pull/549
    - Wrapped imagecodecs for Zarr V3 in https://github.com/maxrjones/virtual-tiff, requires a Zarr-Python PR
    - Next up - GeoTIFF tag support in https://github.com/maxrjones/virtual-tiff + Updating https://github.com/zarr-developers/VirtualiZarr/pull/524 to use virtual_tiff + writing docs on how to build a reader
    - Examples of RasterIndex with virtual_tiff + VirtualiZarr
- Amit
    - Making some virtual datasets against HLS datasets and [L2A]
    - Making some non-gridded datasets virtualized, land imaging based
    - Curious about Kerchunk and Zarr v3: what's the motivation for that?
- Devin Rand
    - Research scientist at Berkeley Earth, working on a climate risk product, downscaled CMIP6 models
    -

- CNG?! 🎉
    - Remote attendance? Videos posted?
        - Will ask
    - Will there be birds of a feather for GeoZarr?
        - Trying to make it happen
    - Should we make the dev branch main?
        - Would drop NetCDF support, but would help CNG demos a lot

- Sean: In favor of doing this, might be better received by conference goers, but need to disclose that might not be stable
- Max: From current understanding (having made breaking changes) supports making dev branch main, but release dev-2-alpha.
    - Don't support a regular release due to breaking changes
    - TN: dropped support for Python Zarr v2, which is big.
    - SH: Can keep virtual open ds, going to be a thin wrapper. Can move Tom's base class out and keep a simple func. Top level API should be the same, right?
    - TN: Biggest change is that we need Zarr v3. There is an important breaking change for opening virtual dataset that changes what is loaded by default. Likely, this will lead to less confusing behavior with this change.
        - All the other changes are not pub doc API. e.g. Manifest store didn't exist before; reader wasn't public before, so who cares about making them extensible?
        - It's breaking in teh sense that, w/o the data types PR (no bigendian support, thus no NetCDF 3) – this doesnt' constitute a minor release, but it is up for debate about whether it should me a major release
            - Don't like the idea of making it an Alpha, bc we want to make these changes anyway
            - Alpha arg: when it stopps benign an alpha, then we have NetCDF3 support – right?
                - MJ: Reason for beta/alpha release is that ppl installing would have netcdf support but people can install it from pypi ….
                - MJ: can get use feedback about what the user facing API is.
        - SH: open virtual dataset: do we have to make alterations for ppl passing storage config? Do we prefer to change it?
            - Instead of dict of store config, can we have a default registry model we're using now? Can we give them the ability to use an explicit store? That would be an API breaking change, but it's also the direction we want to go
            - MJ: Agree
        - JS: Idea of making this in the main branch. Spoke about it a few weeks ago. Want to catch ppl who are doing NetCDF 3 things and tell them
        - TN: Does that mean we're leaning for a pre-release?
            - Yes
    - JS: Will the next release of Zarr break VZ?

        - Until we get the Zarr dtypes, NetCDF3 is broken. As soon as we get that, it will change the VZ API and we'll need to fix it.
            - Aimee: I'm assuming this is referring to the changes in https://github.com/zarr-developers/zarr-python/pull/2874/ - tom@earthmover.io Is it the case that the zarr-python API will change such that VirtualiZarr will break using those updates? In what ways is the zarr-python API changing?
        - SH: We don't have int test that catch the [endian tests]. They are inadvertent.
            - TODO(SH): write explicit tests
- Conclusions
    - Have a series of pre-releases using the develop branch, with setting the buffer limit on Zarr python (to be whatever it is now).
        - JS: Buffer limit doesn't really work I think
            - Will get a compatible environment, but if people have a newer Zarr python environment…
            - TN: I think it would work, bc it wouldn't import the internals. One of the main things we've done
    - SH: Try to merge outstanding PRs. Cut a pre-release. Then, write smaller maintenance and housekeeping PRs due to writing the dtypes successfully. Then, cut another pre-release with all of those (so internals are cleaner, but API would be the same). Then, we'll use this in all the talks.
        - SH: What's the best target naming, Max? "Pre-release"?
            - Yeah, esp if we are pre-releasing version 2. Then we can name it whatever we want :)
        - TN: Biggest change: make Xarray optional. That's what the manifest store lets us do.
        - SH: Maybe a little too far to work in all the virtual tiff into VZ now. Maybe it's better to keep that in a separate repo. There's separate upstream blocking things (in Zarr) that we need, plus img codec stuff that we need.
        - MJ: Not convinced whether merging virtualtiff in VZ is better than an optional dep anyways. Shouldn't plan our efforts around this sprint.
            - Want to be able to VZ new types of tiffs.
    - AM: What would be the benefit of removing Xarray in Manifest store
        - TN: VZ is doing two things: mapping file format in teh Zarr model… Manifest store lets us do step one to get to the

Zarr model before going to the step of Concatenating one Zarr store with another Zarr store.
        - Users won't need to know, but devs for new readers of file formats.
    - AM: Resolve the API surface by what can lead to the more performant design. Give developers an escape hatch.
- MJ: [Said something really great but I didn't get it down!]
- SH: Gone full circle: overloading kerchunk, now we're breaking things down into absolute single responsibility.
    - Good idea; it's cleaner
- TN: it should be easier to write a VZ reader than to write a Kerchunk reader; it should be easier to understand the [manifest].
    - Want to get it to a point that it's so nice to use that people write VZ readers first.
- MJ: Maybe virtual tiff can be the first external reader that we build.
    - TN: like that VZ ships with no readers.
    - MJ: Optimizing around package size will be important.
- JS: Entrypoints?
    - TN +JS: Entrypoints are probably not worth it.
    - TN: When entrypoints go wrong, it can be hard to understand why.
- JS: https://github.com/pydata/xarray/pull/10062

Apr 4, 2025

- Tom Nicholas / Earthmover / @TomNicholas
- Raphael Hagen / CarbonPlan / @norlandrhagen
- Aimee Barciauskas / Development Seed / @abarciauskas-bgse
- Julia Signell / Element84 / @jsignell

Agenda

- Updates
    - Aimee: no updates on dev
    - Tom: https://github.com/zarr-developers/VirtualiZarr/pull/522 sets us up to create virtual datasets from manifest store
        - Main question is how long do we keep this development branch for?
        - Zarr data types PR has gotten really intricate and they're talking about breaking it up.
        - People can always install an old version for NetCDF3 support
        - tom@earthmover.io will check with Rich
    - Raphael: .load or loadable_variables?

        - TN: Punted on the decision, loadable_variables still works but can use .load later if we want to
        - TN: Docs could definitely be improved, focusing on stability of the API first
    - TN: Most users should not have to invoke the Manifest Store itself, its more for internal use by readers
        - There are some scenarios, niche, where it may be easier to load the data itself, easy to construct a numpy array but not a manifest array. Ex HDF doesn't want to give you the byte ranges, compact storage layout
        - Same problem with an inlined variable in a kerchunk references file
        - 2 options:
            - Everything a manifest array even if its difficult
            - Adjust manifest store to accommodate both manifest arrays and inline arrays, "in-memory" object store. Up to the reader to handle these cases where data can not be stored as manifest arrays.
    - TN: #2956 DOC: Missing page on layers of Zarr abstractions
        - AB: (wondering if this is correct interpretation) VirtualiZarr's ManifestStore is an implementation of the zarr-python abstract base class (ABC) which enables serialization and deserialization, using the Zarr specification, from other on disk formats (reading supported for certain formats and writing supported for kerchunk and icechunk)
    - Sean: could be a massive take down of NOAA data today? How to do data backup

Mar 21, 2025

- Tom Nicholas / Earthmover / @TomNicholas
- Raphael Hagen / CarbonPlan / @norlandrhagen
- Amit Kapadia
- Aimee Barciauskas / Development Seed / @abarciauskas-bgse
- Max Jones / Development Seed / @maxrjones

Agenda

- Updates
    - Tom
        - Working on changing the behaviour of `loadable_variables` https://github.com/zarr-developers/VirtualiZarr/pull/477/
        - To make it easier to refactor
    - Max
        - Goal is to finish a TIFF reader by end of the month
        - In the meantime, working on ManifestStore and using obstore in the readers
            - Design notes from yesterday - https://hackmd.io/8EsbxaOjSvCpR8r0fnmZaQ

        - Raphael
            - With Tom's help I think we have a fix for a ChunkManifest bug in the Zarr reader - Gonna fix it and try to get it merged for Zarr v3. Then we can add on bits.
            - [For a later add-on PR] The blocking for reading Zarr v2 is creating a viable codec `zarr.core.codec_pipeline.BatchedCodecPipeline` from v2. Might need some advice here @aimee/sean :)

- Qs
    - Need to work out what's going on here
        - https://github.com/zarr-developers/VirtualiZarr/pull/477/#issuecomment-2743822384
        -

Mar 7, 2025

- Tom Nicholas / Earthmover / @TomNicholas
- Raphael Hagen / CarbonPlan / @norlandrhagen
- Aimee Barciauskas / Development Seed / @abarciauskas-bgse
- Max Jones / Development Seed / @maxrjones
- Julia Signell / Element 84 / @jsignell
- Alex Merose / OpenAthena / @alxmrs

Agenda

- Updates
    - Tom
        - Talking to non-geoscience people about virtualizing everything
            - The NWB (neurodata without borders) format is really HDF5
            - MD dynamics field
                - https://omsf-blog.ghost.io/how-geoscience-can-help-fix-molecular-simulations-data-problem/
        - Started ripping out needlessly complicated `loadable_variables` implementation
            - https://github.com/zarr-developers/VirtualiZarr/pull/477
    - Max
        - Pixi vs. uv
        - Berkeley DSE folks were excited about VirtualiZarr, they'd like a hackathon to learn
    - Alex
        - Foot bone MRI data (virtualize??)
    - Aimee
        - Recap of and next steps from Icechunk Demo to NASA
    - Julia

        - Finally finished first version of xarray kwarg default change PR https://github.com/pydata/xarray/pull/10062
- Merging Aimee's work to pin zarr-python>=3
    - This will break some things
        - datetime/timedelta data type, big endian data type - NetCDF3 and FITS
        - These are not yet supported in zarr-python 3.0 (see 3.0 Migration Guide — zarr 3.0.5.dev3+gcf37198 documentation)
    - This will enable cool things
        - Raphael's zarr reader
        - ManifestStore
- ManifestStore vs. Xarray backends
    - Sean brought up yesterday actually more heavily relying on Xarray backends for some readers, still leaving ManifestStore as an option for others, and wanted me to bring it up even though he cannot make it today
- Pixi vs. uv
- Should there be pretty reprs?
- Property based tests? (hypothesis)

Feb 21, 2025

- Alex Merose / Open Athena / @alxmrs
- Raphael Hagen / CarbonPlan / @norlandrhagen
- Tom Nicholas
- Julia Signell / Element 84 / @jsignell

Agenda

- Updates
    - TN: Review and greenlight a few other people's PRs
    - Is it possible to virtualize Parquet? And open pq with Zarr?
        - Related: https://github.com/alxmrs/xarray-sql/issues/4
    - Can do similar stuff with pandas, but can't do it at the file format level.
    - JS: At E84 there's all sorts of geoprocessing work to get LLMs to write Xarray, but writing SQL would be a lot better. It seems like that will only get more common; people will want to do more of that
    - TN: Is that mapping possible? Is that general? People have different types of indexes
        - DBs have multiple indexes. Are the indexes in a DB directly mappable to Xarray indexes?
    - DuckDB rasters
        - https://medium.com/@ahuarte/managing-raster-satellite-imagery-in-duckdb-with-the-spatial-extension-i-d0b87f49b286
    - TN: One is SQL to query arrays. Other is to think Xarray syntax as analogous to SQL.

        - The second would be really powerful idea as its own. To promote Xarray as a query lang. Then the python package would be the ideal …
        - JS: This is like what Ibis does. You can translate SQL to anything dataframe like.
        - Xarray is missing to_sql and from_sql.
        - TN: Latter suggestion doesn't involve sql, it's just promoting Xarray to being something similar to arrays.
        - JS: The thing is, Xarray is python.
        - TN:well, it doesn't have to be.
        - JS: Yeah, but then isn't that different from SQL? Is there a reason to not use SQL?
        - TN: mission dollar question. If it's the same, then you just do translation. But if it's different for arrays than for tabular data, then you need to do something different.
        - Array arrays different enough front ables to not use SQL?
            - QuantSight: XTensor
        - AM: for to/from Xarray/SQL
            - Parser combinator for the translation
    - RH: Have you seen this? Kyle is making a rust based geotiff reader.
        - https://github.com/developmentseed/async-tiff
    - TN: I had a convo with the maintainer of tiff-file
        - Has a lot of stars, but it's a big python file and a big repo with a big readme.
        - AM: http://grugbrain.dev/
        - TN: We need a tiffreader for virtualizarr. Kerchunk one doesn't work out of the box. There is [] that is not zarr v3 compatible.
            - We need a tiffreader. Need to decide if we'll use tiffile or this async tiff thing that devseed people want to do.
            - Tiffile, if we can make it work, will support every variety of tiff under the sun. But, the devseed one will be easier to maintain, probably more performant.
            - VZ doesn't support grib and tiff right now.
            - Consider doing both, letting users choose their favorite (one does everything, one optimized for COGs).
            - Can we get tiffile to work with VZ without any changes? That would be really big.
        - RH: What's happening with Grib?
            - Want someone else to own that reader.
    - TN: have a zarr v3 branch in VZ right now.
        - This makes ZV3 let v3 be the minimum required version
            - There are a couple codex (little endian ones) that don't exist in Zarr yet
            - If merged into main, it would break some stuff

            - There are certain features that can only really be built on this branch. Want to build these features here, but want the branch to exist for as short as possible.
            - Features are fairly common
        - VZ is not that far off from feature completeness.
    - JS: one of the pain points for the Xarray backend entrypoint system is to figure out what the kwargs should be bc you don't have access to the kwarg docs. Is this the best we can do? Is there some way we can have people import
        - TN: Could just never do the string thing; could just pass in the class.
        - JS: I could do the entrypoint thing.
        - TN: As soon as we add the entrypoint thing, we'll have backwards compat concerns.
        - We're nearly ready for this, but need to change the defaults (e.g. with 1d coords).
    - JS: Wanted to do a gut check: I've been sort of slogging through Xarray changing the default kwargs issue. There's not a lot of good ideas in this PR, but did get to some version. It's intentionalyl failing tests. Trying to not be in refactor mode and get it done. Looking at in now, does anyone have opinions to change the default value for a kwarg with a deprecation cycle? Set the default to None. Other way of doing it (done by pandas): making a special type (sentinal value).
        - TN: Xarray does sentinel value in several places
        - JS: Could encode in the sentinel value the old and the new.
        - This PR was tricky bc there were a couple different defaults in the top level fns. The lowest level needs to know if the user pushed in a particular argument. I added these gross kwargs for tracking the state. Added a top lvl arg to concat. Prob not ok. However, if we used sentinel values, then we could set these to minimal or all. How does this strike people?
        - TN: that sounds like a good idea. Consider posting on the OP about what you want to do.
        - https://github.com/pydata/xarray/pull/10062
        - TN: abstracting that s.t. It's clear to devs but invisible to users is good.
        - JS: ideally, these should be private.
        - TN: think this is fine. There should be a sentinel value for data array names.
        - JS: found some old pandas compat, no default thing. Didn't find thing in xarray already.
        - TN: see weird decorators whose purpose is to mgmt deprecation cycle.
        - JS: need to push a lot of this into the code bc of this condition, need to capture places where data [condition] will be different. Only know in certain places in the codebase that are deeply nested.
        - Need to write some roundtrip tests. Currently got existing tests failing in teh right way.
        - TN: this default is one of the biggest footguns when using Xarray. And for VZ.

        - There's a general problem with Xarray that it tries to be too clever with auto alignment.
        - JS: There will be false alarms. Want to make sure there are no missed alarms.

02-07-2025

- Aimee Barciauskas / Development Seed / @abarciauskas-bgse
- Tom Nicholas / [C]Worthy / @TomNicholas
- Raphael Hagen / CarbonPlan / @norlandrhagen
- Julia Signell / Element 84 / @jsignell
- Alex Merose / Open Athena / @alxmrs
-

Agenda

- Updates
    - Tom
        - Released VirtualiZarr 1.3.0! Now everything is stable with released versions of VirtualiZarr/Icechunk/Kerchunk
        - Otherwise working entirely on this FROST thing
    - Raphael
        - Gonna pause zarr reader until ZArray replacement PR (thanks Aimee!)
        - Ripped out zarr chunk manifest reader/writer docs
        - Updated icechunk examples PR (waiting on icechunk/zarr v3 codec issue opened by Matt)
    - Aimee
        - Working through test failures on `ZArray` → `zarr.core.metadata.ArrayV3Metadata` transition
        - Learning a lot about codecs and VirtualiZarr in the process
    - Alex
        - Investigating applying VZ to NWB data via lindi https://github.com/NeurodataWithoutBorders/lindi
            - Ok, by "investigating", maybe I mean "nerd sniping" Tom (for now)
        - HYTRADBOI
    - Julia
        - One open PR that has some ice-chunk roundtrip tests https://github.com/zarr-developers/VirtualiZarr/pull/422

        - I have started looking more at, still scoping out whether it is possible to know what the behavior will be before compute https://github.com/pydata/xarray/issues/8778
- Discussion
    - TN: Thanks everyone for doing the real work in the latest release!
        - We're now in the same state as any other package, e.g. like Xarray or Zarr
        - Thanks for everyone for helping out to get to there
        - Now, we're in a good position to do further refactors; we are in a more stable place
    - TN: Working on FROST
        - Want to talk to AM after call
        - Work with someone trying to publish micro knowledge
    - RH: Weird codecs btw Zarr v2/v3.
        - Pausing zarr reader until ZArray replaced with `ArrayV3Metadata`
        - Pulled out old style Zarr chunk manifest
        - Updated examples to IceChunk (IC)
    - AB: Working on PR to transition away to Zarr class to v3 metadata
        - Q: Is there any deterministic way to know when something is a filter vs a compressor?
            - If we look at the object, we can tell
            - No existing library fn right now.
            - Some of the readers: confused there
                - E.g. one reader is only using filters
                - Is that correct?
        - A: SH: Unsure. Unclear in Zv3 spec
            - Does there need to be a distinction?
            - TN: I thought the whole point was to change bytes-to-bytes codecs; I thought they were defined, just renamed
            - AB: _ is the pipeline
            - SH: Zv3, there's only the codecs property
            - AB: believe it's always expected to be in the order A2A, B2B, A2B (A-> Array, B-> Bytes)
            - …
            - TN: How else could it possibly work?
            - SH: need to abandon Zv2 usage and focus on building the proper codec change in Zv3? metadata array class
                - Readers… should be fairly easy to do
                - When we do the transition from Zv2 metadata spec, if there is an implicit ordering of the filtering and

compressors, what ordering needs to be applied when creating Zv3 codecs

- Q SH: Do we have a [ordering] in the codecs?
    - [Not in Kerchunk]
    - RH: Zv3 is nice, it can unpack a lot of the metadata object
        - Could convert back to Zv2 when needed?
        - AB: Just wrote this today.
            - Usually always goes in the order: filters, serializer, compressor
            - But if there are no filters, there could be None. Same with compressors, but there' always bytes
        - TN: Fair game to tag davis bennet. He wants feedback on the Dr. API so he can change it now if he has to
            - Tag him
- Q SH: on filters/compressors, is that only happening now?
    - It seems like an arbitrary impl detail in IC
    - Was a comment to say "should we change this to match Zv3 convention?"
    - TN: Conceptually, all IC is doing is a KV store to get the bytes out somewhere, then Zarr-Python does the compression checks
        - Instead fo Zarr being a KV store, 1key:1chunk, IC is … for checking out a different version
    - SH: in the ViZ codebase, the only place where compressors/filters are used is in teh []writer
        - RH: can we sunset that?
        - SH: Do we want the ability to write [kerchunk] files?
        - TN: want ability to *read* them.
        - TN: The ice chunk writer doesn't need to know much about them. In the IC case, starting with a v3 manifest array, … theres not much to do
        - SH: The only way for IC writer and Kchk writer, when we split the codec pipeline into 2 separate structures (split btw filters and compressors), we need to go btw each class type for each codec. If it's a bytes codec, etc.

then split across that and make sure the ordering is correct
        - Don’t understand in Zarr the two size how the reader applies those structures
- AB: What said earlier: if we assume all the writers are using manifest array (v3 metadata), then this way of parsing the codecs should work to separate the compressors and filters.
    - SH: That's all I’m saying
    - TN: Do we need to separate them out? If in v3 metadata form, aren’t we basically just putting Z JSOn in there unchanged
    - AB: why does the IC writer need it?
    - SH: IC does that, …
    - AB: require array thing? Q for DB: Shouldn't there be a way, in Zarr, to identify what the compressors or filters are?
    - TN: does Z need a different helper method to accept a [] object.
    - AB: Let's ask DB this question.
    - SH: i have a PR to numcodecs for all the readers
        - All readers across the board
        - HDF reader uses registry finality now. Gets an instantiated concrete class. Used in HDF reader codec chain
        - Problem: Zv3 numcodec; there’s no registry for this. Need to make up the args… build up a chain of Zv3 codecs …
        - Don’t know the time to get the version out. All readers could use this.
    - AB: I’ll try to TAL at this.
- [AM is a good notetaker :)]
- SH: when we use the kerchunk writer, do we have to make the distinction betw the types of the codec change, or … My guess is no
    - In teh IC writer, is it nece to split them?
    - Tom ask Ryan and Matt to change the repr the’r using to align with Zv3.

- TN: not IC responsibility. It's Zarr Python.
- TN: … except virtual ref method. … how easy it is to pass DS through.

- Alex Merose
  - Approval to look at lindi, variant of kerchunk for neurocellular neuroscience people
  - Long term, see how xarray model applied to neuro data
  - Existing - bids, neuro data w/o borders (NWB)
    - Good sign that people in my domain have invented another virtualizarr (lindi)
  - Can we get lindi and virtualizarr …
    - More use of xarray in neuroscience
  - Dandi - spec of how to use bids and NWB
  - Neuro swift - web zarr visualization
  - Hi trad boy - have you tried running a database on it
    - Idea: the topic of comp sci of large systems aren't different fields, their just one big systems field
    - If you look at the hist of op systems, people believed lisp would be an operating system
    - Giving a talk
  - TN: lindi - invented key ideas of kerchunk and some of icechunk, but also interesting things, like hook for h5py library to read native Zarr that I don't think anyone in Pangeo has done before
    - They're so aligned, someone should get these communities to work together
    - There's always just a last 5% of difference that keeps these communities from combining
  - AM: am concerned about linking(?). Wonder if datatree …?
  -
    - TN: doesn't have links either, makes things way more complicacted
    - TN: did realize that icechunk could add links pretty easily, you just need pointers to the same manifest
      - Problem is there is nothing in the zarr spec which arrays or groups or linked to other arrays or groups
      - Prob a deliberate choice because that idea complicates a lot of things
      - Don't have context on how much linking is needed

- AM: an example may be helpful, I have a high level understanding of why links are necessary. A lot of situations of where experiment markers, like start of a video
    - TN: sounds like coordinated inheritance in datatree, data tree understands this but zarr doesn't.
    - AM: links is an implementation detail not a problem
    - TN: if all the linking you're trying to do is link a bunch to one place, then you can do it with datatree and coordinated inheritance. But linking across lots of different nodes.
    - TN: Use case of having one parent, coordinate array in the parent, and 2 child arrays, automatically links from child to parent, but if its a more complicated relationship, that is not what data tree does. That's what arbitrary links in HDF5 do.
    - AM: I do think they need these more complicated relationships. A set of metadata to define the file structure, the file structure lets you define the people/subject being recorded, subject could have multiple sessions. They could have the same experiment for every subject…
    - TN: These neuro people are doing something so similar, it would be a waste not to try to collaborate
    - AM: NWB will run out of funding, may get an extension but don't want to depend on that

- Julia: On VZ
    - Open PR to .. add IC VZ tests in [memory store]
        - TN: want a review?
        - JS: Is it still useful? Can rebase it; didnt' pass last time it was run
            - Thought it was tests we would want eventually
            - Don't think there's no desire to make round-trip tests to just do IC.
            - Still probably useful to have these tests as well.
        - TN: good question. Original ticket assumed Kurchunk would never upgrade
            - Rount trip tests through Kerchunk adn IC is good. Kch is not trustworthy to best round-trip tested
            - If we can't do that [] can't ensure that we can load data

    - In terms of PR, do we wait for [] manifest? Would insure that all codecs are correct, but … [should we do so?]
    - JS: I think more rt tests are better than fewer. They don't take very long.
        - Suspect they all pass but one, which has a bad dtype. (it's hand-written and not real)
    - JS: There's this ticket in Xarray of changing the default kwargs to open mulitfile dataset.
        - "Open the m'f'ing dataset!!"
        - Toying for how to do that for a while. Last comment: only raise warnings if the behavior changes.
        - Suspect that it's not, but could get partially there. Could see multiple vars with the same name, but can't see if they know all the values.
        - It's impossible probably to do exactly what the request was. Thinking was: try to see what's possible to catch and then run the test to see how many failures it is.
    - TN: this would be a big rabbit hole working on it, but it would be the biggest usability improvement in both Xr and Zarr.
    - JS: wanted to do this for a year
    - TN: A great thing to work on!
    - JS: thinking more about stack and VZ. Since the pangeo meeting on wednesday.
        - Probably going to submit a talk in CNG, but will share
        - AB: probably should.
        - Think max after started convo about writing a blog post.
        - JS: I just talked to Max about it.
        - Ab: take notes from teh meeting and write either a blogpost or CNG guide.
        - JS: going to write after HackMD, but ideas were too disjointed.
- TN: want to submit something to VZ on CNG conference.
    - As long as they start at EM, they will pay me to go.

            - RH: want to have a session? Or demo? Working group?
            - AB: EM is going to do an Ice chunk workshop.
            - [alex takes a texting break]
        - AB: probably should ask them what they are planning, what could complement this?
            - TN: could also submit to SciPy a VZ + IC tutorial. SP tutorials are very competitive.
- JS: thinking of stac responses, often saved to geoparquet
    - Instead of hitting the API, causing problems, could it make sense to save them to VZ? If we could read something into XR, say a STAC of COGs, can you, then, write it to VZ? Or, us VZ to write a VZ'd dataset to IC or Kerchunk?
        - TN: not totally sure. VZ+IC is assuming an array dataset
        - JS: that's irrelevant
        - TN: Oh, I see. Well, STAC can point to data that's not array based
        - JS: sure, but if we can read from STAC to XR, then could it be VZ?
            - TN: Yes, but it's a necessary but not sufficient condition. THe other requirement is, is it an IC?, specific codecs. It must be possible to read a single chunk via a single HTTP request. If all are true, then it should be possible to Virtualize.
        - JS: Would be a cool way to get move from STAC to VZ land.
    - TN: FROST is related to this. It's similar to STAC.
        - Q is how do these relate to each other? Can you store STAC into FROST metadata? How do you do that?

01-24-2025

- Tom Nicholas / [C]Worthy / @TomNicholas
- Max Jones / Development Seed / @maxrjones
- Aimee Barciauskas / Development Seed / @abarciauskas-bgse
- Raphael Hagen / CarbonPlan / @norlandrhagen
- Julia Signell / Element84 / @jsignell
- Chuck Daniels / Development Seed / @chuckwondo
- Ayush Nag / University of Washington / @ayushnag
- Alex Merose / Open Athena / @alxmrs
- Sean Harkins / Development Seed / @sharkinsspatial

Agenda

- Updates
    - Max - a bit blocked by lack of datatree support in Zarr, tracked in https://github.com/NASA-IMPACT/veda-odd/issues/31
    - Aimee
        - Using VirtualiZarr to write MURSST to icechunk
            - Lots of encoding differences
            - Appending arrays with different chunk schemes is essential
        - Considering VirtualiZarr -> Zarr V3 update
    - Raphael
        - Martin is making progress on Kerchunk V3
    - Alex
        - Interested in applicability for neural data
            - Lots of formats, typically HDF5, looking into working with Xarray
            - NME dataset
            - https://openneuro.org/
    - Chuck
        - Getting involved in serializing
    - Ayush
        - Working on earthaccess, considering a zarr module for scoping
- Discussion
    - CNGF attendance + VirtualiZarr workshop
        - Alex M, Julia, Aimee and Raphael planning to attend
    - Discussion about kerchunk dependency and zarr-python 3 migration for VirtualiZarr
        - Max's idea - having modular readers so you only need kerchunk / zarr dependency for the specific reader
            - Concern - internal zarray representation - kerchunk's representation may not clearly translate to a zarr v3 array

01-10-2025

- Tom Nicholas / [C]Worthy / @TomNicholas
- Sean Harkins / Development Seed / @sharkinsspatial
- Max Jones / Development Seed / @maxrjones
- Aimee Barciauskas / Development Seed / @abarciauskas-bgse
- Raphael Hagen / CarbonPlan / @norlandrhagen
- Julia Signell / Element84 / @jsignell

- Ayush Nag / University of Washington / @ayushnag
- Nick Byrne / Quansight / @nenb

Agenda

-
-
- Updates
    - Raphael:
        - More Zarr reader progress. I just posted some notes in the PR:
            - https://github.com/zarr-developers/VirtualiZarr/pull/271#issuecomment-2583581531
            -
        - I just tried out the Kerchunk based tiff reader and ran into an error, then read through a bunch of issues to get caught up.
        - Seems like we have some Zarr v3 work to do:
            - https://github.com/zarr-developers/VirtualiZarr/pull/375
        - https://github.com/zarr-developers/VirtualiZarr/issues/374
        -
    - Tom
        - Been working on `open_virtual_mfdataset`
            - https://github.com/zarr-developers/VirtualiZarr/issues/320#issuecomment-2581680451
            - Idea also to use lithops for serverless parallelization of reference generation
                - Works for local lithops executor
                - To try it out at scale on lambdas I need to learn how to make a Docker container first…
        - Also been thinking about what public icechunk data catalogs should look like
            - But looks like Earthmover are ahead of me
                - https://github.com/zarr-developers/VirtualiZarr/issues/320#issuecomment-2581680451
        - Also been looking at supporting ML formats like Hugging Face's safetensors
            - https://github.com/zarr-developers/VirtualiZarr/issues/367
    - Aimee:
        - Working on fill value fix in dmrpp reader just now, and getting an upstream failure guessing related to zarr upgrade, but just happened
            - This where my confusion is last explained: https://github.com/zarr-developers/VirtualiZarr/issues/343#is

suecomment-2581570500

        - Will be trying virtualizarr with latest icechunk (0.1.0a10?) later today for mur sst
    - Julia:
        - Just starting to get back up to speed and figure out where I can slot in.
    - Max
        - Working on the motivation for spending developer time on "Add support for virtual DataTrees in VirtualiZarr" and "Support storing virtual COGs in Icechunk by implementing an aiocogeo-rs backend for TIFF/COG"
            - Inventory all GeoTIFFs in NASA CMR and identify the proportion that are tiled but not fully compliant COGs
            - Solicit user feedback on whether opening a stack of GeoTIFFs are a virtualized Zarr store would be easier than using STAC + xstac or stackstac
            - Explain value of opening COGs as datatrees
    - Sean
        - Completely brain dead after vacation and just trying to organize priorities for work this month on my agenda are
            - Refactor Zarray to internally use a Zarr v2 representation and include a "translator" to allow requesting a v3 structure.
            - Push PRs to fix issues with HDF5 reader discovered during the Pangeo sprint.
            - Work with Ryan to open a new CF codec repo and start to migrate some Xarray decoding logic there.
    - Ayush
        - Working on ManifestStore
            - Load data directly from virtual dataset
    - Nick
        - Wants to write a reader for safetensors
        - But format specification is likely
- Discussion
    - Dependencies
        - Zarr v3 release
            - We want to just depend on zarr-python v3 now
        - Let's also decouple ourselves from kerchunk
        - We need a strategy for this

            - Swap the HDF readers in the test suit to use Sean's non-kerchunk reader
            - Replace v3_codec_pipeline with v2_codec_pipeline
            - Swap
    - Fill value encoding
        - HDF5 fill value is data that has not been created/initialized
        - CF/Xarray fill value is a "no data" value
        - Proposal from Sean: operate like xarray (strip from attrs and create a CF encoding codec), would require implementing CF decoding logic if cf-codecs and codec to attr conversion in xarray
        - Related issues:
            - https://github.com/pydata/xarray/issues/5475
            -

11-15-2024

- Tom Nicholas / [C]Worthy / @TomNicholas
- Sean Harkins / Development Seed / @sharkinsspatial
- Max Jones / Development Seed / @maxrjones
- Aimee Barciauskas / Development Seed / @abarciauskas-bgse
- Moriah Cesaretti / SOFAR ocean moriah.cesaretti@sofarocean.com @moriahc
- Ayush Nag / University of Washington / @ayushnag

Agenda

- Updates
    - Moriah
        - Has been using kerchunk for wave forecasting, interested in virtualizarr
    - Tom
        - Working on fixing TIFF reader
            - https://github.com/zarr-developers/VirtualiZarr/issues/291
        - Giving talk to the UK Met Office next week
    - Sean
        - HDF5 reader?
    - Aimee
        - NASA HPOSS Proposal update - just waiting on some budget information from each company
        - Working through test failures on the Icechunk append PR, then will

        try with MUR SST
    - Max
        - Updating Pythia kerchunk cookbook to use VirtualiZarr
            - Published cookbook - https://projectpythia.org/kerchunk-cookbook/README.html
            - Update to use VirtualiZarr - https://github.com/ProjectPythia/kerchunk-cookbook/pull/69
            - Waiting on two missing features - GRIB reader and appending without loading anything into memory
        - Virtualizing NASA NEX-GDDP-CMIP6 with icechunk output
    - Ayush
        - DMR++ PR merged, now working on earthaccess integration
        - Some old DMR++ have compression metadata order switched, will require updating filters

Discussion

- Tiff support
    - https://github.com/zarr-developers/VirtualiZarr/issues/291
    - https://pypi.org/project/tifffile/2021.6.6/ returns zarr so should be possible to use this instead of kerchunk's tiff reader
- Moriah's use case
    - Hundreds of tbs of NetCDF data
    - Forecast data 150xx every xx
    - Have "weather cubes" which are "tiles" of the globe with byte range references
        - Tom: You could write a custom reader to convert those to valid chunk manifests with VirtualiZarr, like the DMR++ reader
    - Considering writing their own kerchunk javascript library
- Sean's Q about optional dependencies

11-01-2024

- Raphael Hagen / CarbonPlan / @norlandrhagen
- Sean Harkins / Development Seed / @sharkinsspatial
- Ayush Nag / University of Washington / @ayushnag
- Aimee Barciauskas / Development Seed / @abarciauskas-bgse

Agenda

- Updates
- MyPy failures: https://github.com/zarr-developers/VirtualiZarr/issues/274
- [Ayush]
    - PR is ready for merge: https://github.com/zarr-developers/VirtualiZarr/pull/191
- [Sean] -
    - Kyle B has merged (something) in rust reader?
        - Slowly replace fsspec bits in h5py reader
- [Aimee]
    - Progress on appending PR - intermittent test failure?
        - async/sync related?
- [Raphael]
    - Ongoing Zarr reader work. `Fill_value` issue in reader.
        - Issue in how json/kerchunk writes dtype
    - Playing around with ERA5 from ECMWF api -> virtualizarr
        - Works but… ECMWF cred system is bad & they cache netcdf files?
-

10-18-2024

- Raphael Hagen / CarbonPlan / @norlandrhagen
- Tom Nicholas / [C]Worthy / @TomNicholas
- Sean Harkins / Development Seed / @sharkinsspatial
- Ayush Nag / University of Washington / @ayushnag
- Max Jones / Development Seed / @maxrjones

Agenda

- Updates
    - Tom:
        - Icechunk launched, with prototype VirtualiZarr integration
            - But can't merge that PR, because kerchunk doesn't work with zarr-python v3
            - Writes one virtual reference at a time
            -
        - Separating things out
            - Make kerchunk optional
                - https://github.com/zarr-developers/VirtualiZarr/pull/259

                - Splits out readers behind a common interface
                    - https://github.com/zarr-developers/VirtualiZarr/pull/261
        - Raphael:
            - https://github.com/zarr-developers/VirtualiZarr/pull/251
                - Work towards removing kerchunk tests
                - Use kerchunk reader instead of SingleHDFToZarr
                    - Ref for JSON & parquet
        - Sean
            - Reader almost ready - group issue?
            - Filetype -> reader_name. This can be an experimental reader that raises a warning and suggests the Kerchunk HDF reader
        - Max
            - Icechunk exploration + conda esmf fun
        - Ayush:
            - DMR++ work - add support for nested and root groups
-
-
-

09-20-2024

- Raphael Hagen / CarbonPlan / @norlandrhagen
- Tom Nicholas / [C]Worthy / @TomNicholas
- Sean Harkins / Development Seed / @sharkinsspatial
- Aimee Barciauskas / Development Seed / @abarciauskas-bgse

09-06-2024

- Raphael Hagen / CarbonPlan / @norlandrhagen
- Sean Harkins / Development Seed / @sharkinsspatial
- Aimee Barciauskas / Development Seed / @abarciauskas-bgse
- Gustavo Hidalgo / Microsoft / @ghidalgo3
-

Agenda

- Aimee (and Ayush) attended earthaccess hack day where one of the breakout groups was on virtualization, and heard about TEMPO's interest in using VirtualiZarr. There was also discussion about dmr integration

- Scaling up the reference generation examples
    - coiled serverless
        - Raphael's example: https://github.com/zarr-developers/VirtualiZarr/blob/main/examples/coiled/terraclimate.ipynb
        - This has been super easy just adding references to existing functions
    - Cubed
        - https://github.com/zarr-developers/VirtualiZarr/pull/203/files#diff-f4331d23afc25ce33f12c3c712e7f5a0d7d08ff9d0d66345cbbc9f073a283f3c
        -
    - Larger testing dataset?
        - MURSST - chunk structure changes mid-way through dataset
    - Looking at https://github.com/zarr-developers/VirtualiZarr/pull/203
        - Sean mentions serializing to numpy arrays in intermediary steps
            - Can Virtualizarr open serialized numpy arrays into virtual datasets?
        - Tom mentioned cubed solution may be difficult due to offset storage
        - There should be a way to incrementally do the consolidation
- Discussion about codecs / decoding conventions
    - Sean mentioned that attrs are interpreted as CF conventions and offset and scale are applied if found, and then removed from the attrs in the xarray dataset representation
    - It can still roundtrip the data, it reverses
    - Sean wants to put this all in codecs, applications should not have to reimplement this logic

08-23-2024

- Raphael Hagen / CarbonPlan / @norlandrhagen
- Tom Nicholas / [C]Worthy / @TomNicholas
- Max Jones / Development Seed / @maxrjones
- Julius Busecke

Agenda

- Where should readers live?
    - Virtualizarr repo

08-09-2024

- Raphael Hagen / CarbonPlan / @norlandrhagen
- Tom Nicholas / [C]Worthy / @TomNicholas
- Max Jones / Development Seed / @maxrjones
- Aimee B / Development Seed / @abarciauskas-bgse
- Julia Signell / Element 84 / @jsignell
- Ayush Nag / University of Washington / @ayushnag

Agenda

- Update from Tom on V3 hiccup
    - You can add a new section to the JSON metadata file which can be used by the theoretical storage manifest transformer, but it'll be a problem if the metadata is very large because the Zarr spec includes that JSON is the only used file format
    - Metadata scaling with the number of chunks is unique to the V3 spec, but applies to many extensions (chunk manifest, variable length chunks)
    - Davis doesn't think you can redirect from the metadata file to a different file because the metadata is supposed to be self-contained
    - Tom's planning to raise a GitHub issue
- Discussed what file formats Virtualizarr could support
    - Difficult if "chunks" are split into multiple byte ranges
- HPOSS proposal update - coordinating on chunk manifests
- Modularizing DMR++ / VirtualiZarr integration
    - James should send code for determining abundance of DMR++ indexes
- [https://github.com/zarr-developers/VirtualiZarr/issues/124]

- [Raphael] Similar to the pangeo-forge jam-sesh, I might start dropping zoom links when I'm working on Virtualizarr if anyone wants to join.

07-26-2024

Attendance:

- Raphael Hagen / CarbonPlan / @norlandrhagen
- Tom Nicholas / [C]Worthy / @TomNicholas
- Max Jones / Development Seed / @maxrjones
- Aimee B / Development Seed / @abarciauskas-bgse

Agenda

- Nasa proposal todos
    - Not at the moment.
- [Aimee] Anything blocking writing the chunk manifest ZEP?
    - Outstanding questions on the issue are:
        - Does it definitely have to be a storage manifest transformer or can it be a codec?
        - What should the on disk format be?
            - JSON - doesn't scale
            - Parquet - not great if Zarr has to depend on parquet, but parquet enables treating the whole thing like a database of chunks. Useful for chunk-level statistics
            - Zarr - compresses, cleaner to have a format that is the same as actual chunks
            - Readability of json is nice. Zarr metadata is still in json. Should json be invalid?
        - Todos:
            - Continue discussion on chunk manifest issue until crystallised
            - Point out commonality with other zeps of chunk-level information as a scaling bottleneck
- Long-term roadmap issues raised around using zarr to open everything…
    - https://github.com/zarr-developers/zarr-specs/issues/303
    - https://github.com/pydata/xarray/issues/9281
- [Raphael] Advertising / Growing dev group? Should we post anywhere else?
    - VirtualiZarr documentation, in contributing section
- [Raphael]Cloudpathlib -https://github.com/zarr-developers/VirtualiZarr/issues/172
    - lacks https, but might have more consistent handling of cloud paths?
    - We could create a "httppathlib" library that is similar to but separate from cloudpathlib
    - Another option is https://github.com/piskvorky/smart_open
    - What are some of the most challenging issues with fsspec?
        - One is the different ways it handles opening different protocols: https://github.com/fsspec/filesystem_spec/issues/579#issuecomment-1140377495
        - Another is caching

06-28-2024

Attendance:

- Raphael Hagen / CarbonPlan / @norlandrhagen
- Tom Nicholas / [C]Worthy / @TomNicholas
- Julia Signell / Element 84 / @jsignell
- Sean Harkins / Development Seed / @sharkinsspatial
- Aimee B / Development Seed / @aimeeb
- Julius Busecke / Columbia / @jbusecke

Agenda

- https://github.com/zarr-developers/zarr-specs/issues/300
    - Comparison with STAC - STAC has “infinite dimensionality”, no issue with just adding another item
    - More generalizable than backing with sparse arrays (as opposed to numpy arrays)
    - Example use case could be swath data
- Earthmover’s plans on version control
    - If you put your data into arraylake, the format of chunk versions is not open so you lose the ability to understand this data (in open source)
    - They are considering open sourcing this versioning format
    - One method is to create pointers to old chunks in a new version of the store, so there is no duplication for unchanged chunks
    - Compare relative to LakeFS
        - A feature of lakeFS - there is an enterprise level service is user access control
    - Also concerned about scalability. Supposedly arraylake is infinitely scalable, but when it comes to NASA-scale data we have hit S3 and lambda limits a number of times
- Where should Codecs live?
    - Should probably live in zarr so that different zarr implementations can read the data

    - We have to take this deserialization logic in xarray and make it accessible to other zarr implementations
    - How does xarray determine when it’s reading from the zarr (or virtualizarr) store when to do the deserialization
        - Probably in the zarr xarray backend
        - An xarray dataset is input and output from cf decoding steps, so the first xarray dataset is not was is returned to the user
    - Sean “it just works” by creating a numcodecs from the HDF5 codecs information
        - There’s compression, scale and offset

        - Julius is handling the cftime conventions, implicit logic of cf that is captured by xarray
    - It would be ideal to get Kai to pick his brain about encoding/decoding abstractions
    - The big goal here could be to read netCDF data through zarr-python and get all the decoding correct, even for CF conventions
- How should we think about virtualizarr now vs all these upstreaming ideas in the future?
    - We are basically at the MVP, v1.0.0
    - Having “round trip” functionality makes it easier to advocate for this project
    - MVP was to replace kerchunk, with no attempt at upstreaming
        - Mostly at feature parity with kerchunk already
    - The next thing to work on is scaling it out, seeing how it performs
    - Groups (and xarray.DataTree) should be relatively straightforward to implement
        - Tom’s not going to prioritise this himself - someone else who wants it should do it
    - Decision: release version 1 and add issues for improvements, like groups, in future versions
    - Tom will also add explanation of this roadmap plan to the docs
- Julius’ ClimSim dataset on HuggingFace
    - Failcase with the current version of the data: https://github.com/leap-stc/climsim_feedstock/issues/2 (Cannot remotely read files with h5netcdf, but can pull it to local and read it with netcdf4. Is something wrong with files themselves?

06-14-2024

Attendance:

- Raphael Hagen / CarbonPlan / @norlandrhagen
- Tom Nicholas / [C]Worthy / @TomNicholas
- Julia Signell / Element 84 / @jsignell
- Sean Harkins / Development Seed / @sharkinsspatial
- Max Jones / CarbonPlan / @maxrjones
- Anderson Banihirwe / CarbonPlan / @andersy005

Agenda

- Summary of grant applications?
    - Two proposals in progress
        - One with Microsoft
            - Quicker turnaround
        - One to NASA HPOSS
            - Longer-term, but submit in 1-2 months
    - Zarr-python proposal went into F.7 NASA solicitation
- Our use cases

    - Tom: CWorthy OAE dataset
        - Also just learned more about Zarr in general
    - Raphael: General improvement on kerchunk
    - Anderson: CarbonPlan-maps could point to kerchunked data…
        - Tom: Let's raise an issue for this specifically
        - Tom: Can't change chunksize / dtype / compression
            - Anderson: But could use zarr-proxy…
        - Max: Could point maps tools at COGs and not need a tile server
    - Max: Excited for virtualizing geotiffs and COGs
        - Avoid the GDAL / rasterio layer
    - Julia: Replace parts of pangeo-forge?
        - Looking for high-impact places to contribute that would help NASA
    - Sean: Managing large archives that NASA produces
        - Mandate is to provide cloud-optimized access to legacy files
        - Need to generate chunk manifests for legacy archives
            - NASA will never transform all that data into Zarr
            - Friction doing it using kerchunk / pangeo-forge
            - Trying to get recommendations right first time
            - Most don't have DMR++ reference files
                - And DMR++ can only ever focus on netcdf-style data
- Tricky upstream work to flag?
    - Codecs refactor in xarray
    - Xarray indexes work needed to get `combine_by_coords` to work
    - Zarr virtual concatenation ZEP
    - Inspecting files for byte ranges without using kerchunk's readers
        - Pull out into a separate library?
    - Refactor virtualizarr to use v3 "zarr object models"
        - Issue for this?
    - To what extent could ManifestArrays just be zarr-python arrays in the future?
- First release?
    - Tom will hit release
- Channel to co-ordinate?
    - Zulip
    - Put it in the zarr-developers Zulip
    - Max will make a channel
- Any other Q's