

Pangeo Forge Coordination Meeting

Meeting URL:

<https://numfocus-org.zoom.us/j/81511924663?pwd=3rLhw0R2jqUmua9hyp2sCg8rdtQXAY.1>

Next meeting agenda

2025-03-12

Attendance:

- Justus Magin / LOPS / @keewis
- Raphael Hagen / CarbonPlan / @norlandrhagen

Agenda

- Maintenance status of the feedstocks?
 - Mostly abandoned. Archive with a warning added to the readme?
- Edit the docs to mention that PGF has been initially designed like conda-forge, but the infrastructure part has been discontinued
 - Good idea
- Major release with the clear warning that this is a breaking change upgrading to zarr-python v3

2025-03-12

Attendance:

- Justus Magin / CNRS / @keewis
-

Agenda

- Merge zarr v3 support? The failing tests are fixed, and CI passes

2025-02-12

Attendance:

- Justus Magin / CNRS / @keewis

- Raphael Hagen

Agenda

-

2025-01-29

Attendance:

- Alex Merosé / Open Athena / @alxmrs
- Justus Magin / CNRS / @keewis

Agenda

- [Alex] Optimizing the DaskRunner in Beam.
 - I think I can make it better / performant.
- [Alex] 📄 A Vision for Pangeo Forge in 2025
- JM: Beam PySpark runner is great
 - Runner is still the main issue for the pipelines.
- JM: people have moved away, and there is less activity overall.
 - Test are failing, for example. Don't have resources to fix it.
 - Had to do with breaking deps
- Julius has still been using PGF, but no longer.
- Look at oct discussion
 - What will be done with PGF now?
 - Can we use this / the fn execution in parallel / the executor (Lithops)?
 - Some pipeline is like rclone (fsspec has some problems)
 - There may be a case that the PGF runner is too complicated, but PGF might be ok.
 - Maybe something simpler, or something that supports different stages.
 - E.g. rclone, or virtual-zarr creation, all w/o beam
 - Then, use a different stage, e.g. xbeam, that is used for say rechunking.
- Rigidity: all the recipes look the same, but the details look different.

2024-11-20

Attendance:

- Julius Busecke / Columbia / @jbusecke
- Justus Magin / CNRS / @keewis
- Raphael Hagen / CarbonPlan / @norlandrhagen

Agenda

- [Julius] I would love to be able to make single-file recipes work ([PR](#) very much WIP)
-

2024-11-06

Agenda

- [Justus] What can we work on to improve the state of pangeo-forge (assuming we keep it)?
 - Version of pangeo-forge-runner that is less magical (callable recipe) and allows additional kinds of executors (e.g. dask, cubed, etc)?
 - Multiple pipelines in the same recipe, with different resources

2024-10-23

Attendance:

- Julius Busecke / Columbia / @jbusecke
- Justus Magin / CNRS / @keewis
- Raphael Hagen / CarbonPlan / @norlandrhagen
- Tim Hodson / USGS/ @thodson-usgs

Agenda

- [Raphael] Goals - What do people here want out of PGF
- [Julius] Keep runner?
 - Can runner be a set of common/helpful suggestions instead of mandatory? It makes testing really difficult
- [Julius] Deprecate [gh-action](#)!
 - <https://github.com/pangeo-forge/deploy-recipe-action/issues/35>
- [Julius] More verbose meta.yaml example
- [Raphael] I added kerchunk and Zarr reading as inputs to OpenWithXarray
- [Raphael] The fsspec caching model is a real PITA.

- Rate limiting and retries (Nathan made a bunch of progress on this)
- Long running jobs with poor resource usage.
- Resource reqs for downloading a bunch of files can differ from rechunking!
- Split 'getting files' into separate pipeline / stage?
- [Julius] Owes an rclone copy/moving stage
 - Maybe we want to recommend a rclone copy stage instead?
 - Can this solve the long-term painpoint of getting data from HPC

Discussions

- Bring your data to your own object-storage, then use pangeo-forge-recipes?
 - Lithops, github CI, beam job etc.
 - Then start recipe with stable inputs
- Non-local beam runners are difficult to setup.
 - For ex. Tim Hodson has been using cubed b/c it's been much easier to setup, IT wise.
- High level pros and cons:
 - Pros:
 - High level similar structure
 - Nice mental model - pluggable and moveable pipeline pieces
 - Tons of IO options
 - Rechunking
 - Cons:
 - In real cases, beam is not very agnostic. Getting runners setup on different clouds is difficult for most.
 - IO issues
- Can cubed / cubed xarray fulfill (eventually) the reqs that pangeo-forge[beam] does?
- Improving pangeo-rechunker [docs]?

ToDo

Props

- [Raphael] - Everyone here for reviving this meeting

2024-04-29

Attendance:

- Raphael Hagen / CarbonPlan / @norlandrhagen
- Sean Harkins / Development Seed / @sharkinsspatial
- Aimee Barciauskas / Development Seed / @abarciauskas-bgse
- Max Jones / CarbonPlan / @maxrjones

Agenda

We discussed the [draft roadmap](#) and made changes which reflected discussion, such as:

- How to organize recipe development
- What to do about pangeo-forge-runner? Nothing for now since it sounds like some people are still using it. We may deprecate it in the future.
- Max asked if anyone has tested OpenWithXarray with Zarr (and then said he would test it)

2024-04-01

Attendance:

- Raphael Hagen / CarbonPlan / @norlandrhagen
- Greg Corradini / DevelopmentSeed / @ranchodeluxe
- Julius Busecke / LDEO / @jbusecke
- Charles Stern / LDEO / @cisaacstern
- Ryan Abernathey / LDEO & Earthmover / @raternat

Agenda

I. Props

- A. [Raphael] Charles for the appending PR! - Thanks!
- B. [Raphael] Greg, Nathan & Julius for deep dive into fsspec/xarray GC
 - +1 from Julius to Greg and Nathan. This was really fun, and has for now unblocked literally all my PGF work!
 - +1 from Charles

II. Agenda

A. [Greg] [next steps](#) on how to handle deadlocks 🙌

- [Greg] Should we use something other than fsspec? Beam native i/o connectors?
- [Ryan] Fsspec is non-trivial to replicate, particularly around sync/async. Can't we just work around the problem by downloading files to workers?
- [Greg] Isn't downloading against the ethos of cloud data? Maybe we don't need async performance, if we're scaled out horizontally anyway?
- [Ryan] Is downloading to a worker's temporary storage really so bad? It's a little slower, but maybe it's an acceptable trade-off?
- [Greg] Downloading introduces possible other concerns (worker storage)... maybe that's not that big of a problem.
- [Ryan] If there's any way to avoid changing i/o connectors, it will be less painful.. `OpenWithXarray.copy_to_local = True` (with `load=True`, possibly)... should just work, and may not be slower if we know we're going to consume the whole file. Heavily subsetting input files would be one case where this would not be inefficient.
- [Julius] Slightly less efficiency in exchange for risk
- [Aimee] Does open locally obviate the need for caching?
 - [Ryan] Sort of different concerns, since this will be a local cache.
 - [Aimee] Could cache be to local disk?
 - [Ryan] Cache needs to be globally accessible to all workers. Really, caching should be out-of-band from Pangeo Forge. Could be Globus, Skyplane, etc. ... if the goal is to get files from a-to-b, then Pangeo Forge might not be the right tool.
- [Raphael] For really big netcdf files, how does memory consumption look with `copy_to_local`?
 - [Ryan] If we pass around xarray datasets that are not loaded, then the data is lazy and need to be materialized on load. But load simplifies this.
 - [Charles] Does serialization of GCS session cause this issue, then?
 - [Greg] Just serialization

B. [Charles] Append mode for StoreToZarr

<https://github.com/pangeo-forge/pangeo-forge-recipes/pull/721>

- [Charles] Question for the group— How much should Pangeo Forge be responsible for “not messing up” appending? Is it the responsibility of Pangeo Forge to make sure logically incoherent data doesn't get written?
- [Ryan] In most of data engineering, SQL INSERT is the de-facto way this

is done. But in Zarr, that is not the case. What kind of operations do people want? What does INSERT mean in this context? Let's start by exposing what xarray knows how to do. And let users make their own paths with this.

- Step 1: make sure encoding is consistent
 - [Charles] I think `ds.to_zarr` which we use for the schema resizing, should have us mostly covered on this.
- Step 2: become aware of the time dimension
 - Pruning files to become aware of where the appending should begin...
 - [Julius] If we can append a specific additional segment of the data, then maybe its the role of a different stage/transform for pruning/selecting that subset
 - [Charles] A simpler feature than pruning, would be a "strict=True" or something, which could help prevent duplicate appends.

C. [Tim] How about UPDATE? If someone messed something up, could we re-write a range?

- [Charles] Should be doable, let's open an issue
- [Julius] This implies in some way knowing data state, possible with a database, but perhaps a little bit hard for Pangeo Forge.
- [Ryan] Small fixes can be done with zarr or xarray directly

Meeting URL: <https://meet.google.com/fne-nuup-hnn?authuser=2>

2024-03-18

Attendance:

- Raphael Hagen / CarbonPlan / @norlandrhagen
- Charles Stern / LDEO / @cisaacstern
- Sean Harkins / Development Seed @sharkinsspatial
- Julius Busecke / LDEO / @jbusecke
- Max Jones / CarbonPlan / @maxrjones

Agenda

- III. [Charles] Append-mode for StoreToZarr PR forthcoming
 - A. [Tim] This would be useful for USGS
- IV. [Nathan] PySpark Runner WIP

- A. <https://github.com/moradology/beam-pyspark-runner>
 - B. If there's a way to run on AWS EMR Serverless, that would be great!
 - C. EMR Serverless can scale to ~1000s of CPUs
- V. [Julius] Is this fsspec hanging issue related to what I've been seeing in CMIP6 rechunking? <https://github.com/pangeo-forge/pangeo-forge-recipes/issues/710>
 - A. [Raphael] We've been observing strange hangs in *StoreToZarr recipes.
 - B. [Nathan] Unfortunately a lot of bugs are really difficult to reproduce! I'm suspecting there may be something happening in fsspec that doesn't play nicely with serialization. Ideally there's a way to reproduce this hanging outside the distributed context, e.g. pickling a file handle and not being able to use it post-serialization.
 - C. [Julius] From CMIP6, a potentially relevant example (which may be related) is caused when complex rechunking is required. Might be able to boil this down into an MRE (for dataflow).
 - D. [Tim] In a blog post (about rechunker, possibly, link forthcoming) by Ryan I've seen re: rechunking, I've seen the strategy of writing an intermediate of store.
 - E. [Charles] In recipes, we don't (/shouldn't) need to write an intermediate store, because we are using GroupByKey (GBK) to materialize the data for each chunk only exactly when it's needed. Maybe the GBK is getting congested somehow causing the hangs... or maybe this is the same issue as the fsspec file handle thing being discussed in the linked issue.
 - F. [Greg] Thinking about ORM systems, which are all lazy, sessions/connections are all managed at execution time. It would be nice if xarray handled this interaction in the same way as an ORM but they probably do not want to
 - G. [Sean] This might be the same thing as we hit 3 years ago, when fsspec switched from sync to async...
- VI. [Max] Questions about nan equality in schema tests and GeoTiff opener
 - A. At CarbonPlan, we want to be in control of FillValues for zarr, for Javascript compatibility.
 - B. Saw a PR from a year ago for honoring encoding, but tests failing because of a dict nan-equality issue.
 - C. [Charles] On GeoTiff, using xarray_open_kwargs should work for specifying rasterio engine for GeoTiffs. Also open PR for _FillValue encoding changes
- VII. [Julius] [This PR](#) is fairly crucial for data management operations at LEAP. Would be super cool if there is some time to discuss options for implementation.
- VIII. [Julius] CMIP6 recipe deployment [broke](#) while I was on vacation. Does anyone have any clues how to fix this?
 - A. Seems like a probable impact of venv management
 - B. [Greg] Will reach out to Julius to sync on this later today probably.

- IX. [Charles] For Julius's context, there was a discussion begun over the last few weeks about simplifying/changing the runner/action paradigm
 - A. [Julius] data-management is broken so could be more easily changed, cmip6 was working before break
- X.
- XI. [Tim] More consolidation into a single repo would be very helpful from a user perspective
 - A. [Charles] Agree

2024-03-11

Attendance:

- Ryan Abernathey / LDEO / Earthmover / @raternat
- Julia Signell / Element 84 / @jsignell
- Raphael Hagen / CarbonPlan / @norlandrhagen
- Greg Corradini / DevSeed @ranchodeluxe
- Charles Stern / LDEO / @cisaacstern
-

Agenda

- XII. "What's next for Pangeo Forge" - Pangeo Showcase seminar by Tim
- XIII. [Ryan] Summary of project history
 - A. Started with ambitious scope!
 - Build a data pipeline / ETL tool
 - And ALSO use that tool to operationalize building cloud-scale mirrors of lots of production datasets
 - B. An evolution
 - Shifting focus from "any" scientist/analyst as core user, towards the more technical engineering user
- XIV. Use cases for Pangeo Forge

- A. [Ryan] One initial impetus for Pangeo Forge was <https://catalog.pangeo.io/>
 - A static data lake and static (intake) data catalog, which was
 - As data got stale, or needed updating, or needing fixing, those dynamic prompted the growth of Pangeo Forge
 - Rather than the artifact being the dataset, the artifact could become the recipe, which can be maintained and extended over time
- B. [Ryan] Here's one other motivation for Pangeo Forge:
<https://cloud.google.com/blog/products/data-analytics/new-climate-model-data-now-google-public-datasets>
 - Naomi Henderson did a lot of heavy lifting early on by doing manual data ingestion with custom scripts.
 - This project has moved to Pangeo Forge to make the ingesting tooling component more maintainable.
- C. [Raphael] At CarbonPlan, my work is focused on building large scale climate datasets on the cloud
 - Pangeo Forge interest as an alternative to homespun xarray + dask workflows.
 - Looking forward, updating / appending in the time dimension
- D. [Charles] My role going forward: trying to stay involved in the project, believe in the model. Appending and continuous updating remains very important. Runners matter a lot.
- E. [Julia, Greg, Nathan] - NASA use case. Use Pangeo Forge to ETL data to DAACs. Kerchunk matters a lot.
- F. [Nathan] - "I think it's a good idea"
- G. [Julius (not here)] - LEAP use case
- H. [Tim] - Large workforce of scientists, moving to cloud, want reproducibility. Traditionally have tackled this through the "hero" model, success tied to individuals developing workflows + scripts. Federal agencies slow to adopt new tech. Problem we're facing: because we don't have this ARCO database set up, falling back into old patterns.
- XV. [Julia] If we look at Conda Forge, then recipe development is the responsibility of "library" i.e. dataset owners.
 - A. [Charles] A great reminder. This was hard to manage when we tried.
 - B. [Nathan] The distributed nature of this work makes it a lot harder
- XVI. [Ryan] A clearly scoped definition of the tool, and who the target users are, will help us move forward.
 - A. One thing that we've learned is out of scope, is cataloging
 - B. Pangeo Forge *might* be best as "just" a data mover (ETL) and not a cataloger
- XVII. [Ryan] Does kerchunk change the need to build multi-TB zarr datasets?

- A. [Raphael] I see kerchunk are more of a niche use.
 - B. [Greg] Brianna's
 - C. [Tim, Nathan] Lots of reasons why real zarr is necessary: source data is not kerchunk-compatible, we want a pyramid, the grid is bad.
 - D. [Nathan] VRTs (as an analogy) are great, but only if the underlying data is good enough to begin with. Otherwise, usefulness is limited.
 - E. [Ryan] From my view, a lot of people seem to still be building real Zarrs in the cloud.
- XVIII. [Ryan] How does NASA's role as an archival vs. ARCO data provider affect their potential use needs of Pangeo Forge?
- A. [Greg] Aimee would be best to speak to this.
- XIX. ...
- XX. [Ryan] Where can this tool go over the next year?
- A. Audience for now is developers: people whose job it is to get zarr data into the cloud. This doesn't preclude offering users
 - [Nathan] Platform independence (i.e. runners) matters.
 - [Greg, Ryan] Developers can be
 - [Tim] Pangeo Forge is too hard to use currently, even for developers. If AWS experience was more akin to the dataflow experience.
 - [Ryan] The way these things become _easier_ is making a SaaS... i.e. Astronomer for Airflow, Prefect, Dagster, etc.
 - [Nathan] i think a full dask runner and a NON portable spark runner would dramatically widen the appeal/reach
 - [Charles] Agree with Nathan. Maybe the middle ground is something closer to "zero to JupyterHub" on commodity cloud infrastructure? Is the middle ground between where we are now and SaaS, terraform that actually works?
 - B. [Ryan] Solving AWS deployment sounds like it should be a major focus

2024-02-12

Attendance:

- Charles Stern / LDEO / @cisaacstern
- Tim Hodson / USGS / @thodson-usgs

- Sean Harkins
- Greg C.
- Aimee Barciauskas / Development Seed / @abarciauskas-bgse

Agenda

XXI. Props

- A. [Charles] All of the effort by Greg, Aimee, Nathan, and Tim! Great to see pangeo forge taking new life!

B.

XXII. Agenda

- A. [Charles] Logistics:

- Meeting link
 - Sean will setup a devseed google meet link (or ask Aimee)
- Schedule (Ryan's request to shift cadence/phase by one week)
 - Present attendees agree! DevSeed maybe has a conflict next week
 - Action point: Once Charles get's a meeting link from Sean/Aimee, he will change the public calendar announcement.

- B. [Charles] The end of Pangeo Forge NSF Funding (Aug 31)

- C. [Greg] GCP account for integration tests – what else is it for?

- Would like to do some comparison with Flink
- Can recipes really be generalized across executors? Maybe not.
- [Sean] Out-of-order contiguous chunks on Flink
- [Charles] Can't give non-Columbia email addresses access to GCP console
- [Sean] DevSeed can provide a path to GCP access for Greg

- D. [Tim] Finally got a zipped Tiff recipe to work! By digging deeper into fsspec

- E. [Tim] But have been running into some tiff/xarray compatibility troubles.

- [Sean] Offered to dig into the details on rasterio/fsspec backend issues
- [Tim] open_dataset doesn't seem to infer spatial metadata, and can't seem to get drop_bands or scale_factor to work as hoped
- [Sean] in the long run, we don't want to defer everything to rasterio, but rather read use tiff as the reader backend

- F. [Sean] Everyone should attend future-of-kerchunk Pangeo community meeting on Wednesday. - <https://discourse.pangeo.io/t/whats-next-for-kerchunk/4005>

- [Sean] Kerchunk is moving fast, so are we ready to evolve the pangeo forge wrapper?
- [Charles] Yes, the group here should absolutely take ownership of evolving/releasing the kerchunk side of pangeo forge. I will make sure

I'm available to support knowledge transfer as needed.

- [Sean] Chunk map/manifest could (and may likely) be built directly into zarr v3, so there may be to ability to do kerchunk-like-things directly in zarr-python. Another issue is just the internal structuring of kerchunk.
- [Nathan] How far are we from the state-of-the-art in Pangeo Forge currently?
- [Sean] The main work is really evolving kerchunk array operations to closely map xarray array operation conventions. Tom Nicholas's ideas on this are what should (and probably will) be implemented eventually.

■

G. [Sean] What is the status of multiprocessing / out-of-order issue with kerchunk?

- [Aimee] wrote a MRE test
- [Nathan] the auto-dask strategy appears to be faithfully implemented, but still the test is failing
- [Aimee] Which tests specifically are failing?
- [Nathan] Let's pair on this
- [Charles] I'll be happy to join
- We're going to pair on this tomorrow (Tuesday!).

H. [Aimee] What other orgs aside from NASA are considering use of pangeo-forge?

- [Tim] USGS is considering, Beam complexity is an open question. Dask is more familiar.

2024-01-15

Attendance:

- Charles Stern / LDEO / @cisaacstern
- Ryan Abernathey / LDEO / @raternat
- Rapahel Hagen / CarbonPlan / norlandrhagen@gmail.com

Agenda

XXIII. Props

- A. [Ryan] Charles for pushing forward on Dask work to make Dask Beam Executor possible
- B. [Charles] Raphael for moving ConsolidatedMetadata

XXIV. Agenda

A. [Raphael] <https://github.com/pangeo-forge/pangeo-forge-recipes/issues/666>

B. Chunk alignment bug

- <https://github.com/pangeo-forge/pangeo-forge-recipes/pull/658>

- Requires someone (probably Charles) to slowly step through `combine_fragments` and figure out what the context is here

C. Appending

- Beam streaming not well supported across different executors, so let's not use that to begin with

- Milestones

- We can write to an existing, pre-initialized Zarr store (rather than creating new each time)

Changes would go here:

https://github.com/pangeo-forge/pangeo-forge-recipes/blob/6e40279b7f5740600b77a348dd48d09f11467564/pangeo_forge_recipes/transforms.py#L351

- Edge cases around chunk alignment

- Maybe automatically populate `target_chunks`?

- Prune inputs based on existing data in target

- Restrict appending to time dimensions

- Make `file_pattern` more explicit about the time range it covers

- (a) Currently, we can't introspect the `FilePattern` to get a temporal range

- Examine the target to sub-select items that need updating from the `file_pattern`

- Goal is idempotency

- Streaming vs batching?

- [Ryan] I've gone back and forth on whether real streaming is necessary; generally, users of Pangeo Forge don't need streaming-level time intervals (generally, update interval is daily)

- [Raphael] HRRR is faster refresh (hourly)

- For `kerchunk`?

- Where should this logic live? Currently, the logic would be different, but ideally that shouldn't need to be

- With Joe and other's work on Zarr Python, adding this as a Zarr V3 extension should be possible

D. AWS ERA5 public dataset is deprecated (provider has stopped maintaining it)

- <https://registry.opendata.aws/ecmwf-era5/>

E. Dask runner update

- Side inputs: <https://github.com/apache/beam/pull/27618>
- Requires Dask Bag groupby fix:
<https://github.com/dask/dask/pull/10734>
 - Charles will clean this up, and present at Dask Demo Day
- Next step after this: combiners via dask bag foldby

2023-12-18

Attendance:

- Charles Stern / LDEO / @cisaacstern
- Tim Hodson / USGS / @thodson-usgs
- Julius Busecke / LDEO / @jbusecke

Agenda

XXV. Props

- A. [Charles] Brian Hulette (from Google) for helping debug DaskRunner GBK issue
- B. [Charles] Florian Jetter (from Coiled) and Jacob Tomlinson (from NVIDIA) for merging upstream fix for GBK issue in Dask Distributed
- C. [Charles] Tim for enthusiasm!
- D.

XXVI. Agenda

- A. [Tim] Working on a little xarray extension for managing diffs:
[thodson-usgs/xbitdiff \(github.com\)](https://github.com/thodson-usgs/xbitdiff)
 - Git vs. Subversion style diffs (this implements the latter style)
 - [Julius] Sounds like a trade-off between compute (for computing the diff on loading time), vs. storage
 - [Tim] Use case: 100s of TB of NCAR model data, alongside equal-sized bias-corrected dataset. Could we save lots of storage space by just storing the diff?
 -
- B. [Charles] DaskRunner is moving along
 - Fixing GBK issue <https://github.com/apache/beam/pull/29802>
 - Dask Demo Day this Thurs (12/21)
- C. [Julius] CMIP6 PGF recipe:
 - We got access to the open dataset bucket [Charles - 😊!!!!]
 - Remaining blockers on CMIP workflow:

- Consolidated metadata and dims -
<https://github.com/pangeo-forge/pangeo-forge-recipes/pull/556>
- Mover/copier stage -
<https://github.com/leap-stc/cmip6-leap-feedstock/pull/67>
 - Will skyplane work?
<https://skyplane.org/en/latest/quickstart.html#python-api> — maybe beam workers don't have enough permissions for this?
 - Beam IO connectors - Sean will look into docs (or ask ChatGPT!)
 - (a) <https://gist.github.com/sharkinsspatial/e7253aa82b0dde8607a09d0a8134c28>
 - ~~Pangeo_forge_recipes.storage._copy_btwn_filesystems~~
 - `gcsfs.GCSFileSystem.mv(..., ..., recursive=True)`
 - Could we make a performance comparison between gcsfs and built-in beam gcsio for this? (Not a high priority, but interesting!)
- Mutlidim chunking bug -
<https://github.com/pangeo-forge/pangeo-forge-recipes/pull/658>
- Side input issue -
<https://github.com/pangeo-forge/pangeo-forge-recipes/issues/663>
 - [Julius] A canonical example of ocean model output
 - [Charles] See my comment on issue discussion
- D. [Julius] Spoke with a former xrip collaborator at NOAA who wants to get started with
- E. Pangeo Forge
- F. [Sean] In January, will be starting with NOAA NCICS (North Carolina Institute for Climate Studies) to build kerchunks for HRRR with Pangeo Forge!

2023-11-20

Attendance:

- Charles Stern / LDEO / @cisaacstern
- Raphael Hagen / CarbonPlan / @norlandrhagen
- Tim Hodson / USGS / @thodson-usgs

-

Agenda

XXVII. Props

- A. [Charles] Julius for a great pangeo showcase about Pangeo Forge + CMIP6
- B. [Charles] Greg, Raphael, Nathan for pushing on Flink docs
- C. [Ryan] Charles for making big updates to the [docs](#)

XXVIII. Agenda

- A. [Tim] fsspec open zip issue, e.g. [thodson-usgs/staged-recipes at us-ssebop \(github.com\)](https://github.com/thodson-usgs/staged-recipes/pull/1)
- B. [Tim] strategize about USGS collaboration
 - Why does “sponsorship” look like, OSN, etc?
 - [Ryan] OSN vs. commercial cloud is a question for USGS IT leadership.
 - [Ryan] NumFOCUS sponsorship or work orders can be made to support Pangeo Forge. Details TBD, but this is a helpful mechanism.
 - Internal mentorship
 - Incremental updates
 - Major recent Xarray improvements in this area:
 - <https://github.com/pydata/xarray/pull/8434>
 - <https://github.com/pydata/xarray/pull/8428>
 - A simple, good starting place, is just appending to the end of the dataset
 - This does not solve idempotency, but is a good place to start
 - Let’s discuss here:
<https://github.com/pangeo-forge/pangeo-forge-recipes/issues/447>
- C. AGU Pangeo Dinner! <https://lu.ma/pangeo-agu-2023>
- D. [Charles] pangeo-forge.org PR forthcoming, to clean up stale content
- E. [Charles] DaskRunner update
 - Dask Demo Day - Dec 21 - With Jacob Tomlinson
<https://github.com/cisaacstern/beam-dask-demo/pull/1>
 - Working with Brian Hulet (from Google Open Source) on this blocking bug - <https://github.com/apache/beam/issues/29365>
- F. [Raphael] Going to get back to finishing up this PR:
<https://github.com/pangeo-forge/pangeo-forge-recipes/pull/636>

- G. [Sean] Discussion of <https://github.com/fsspec/kerchunk/issues/400>

2023-11-20

Attendance:

- Charles Stern / LDEO / @cisaacstern
- Raphael Hagen / CarbonPlan / @norlandrhagen
- Tim Hodson / USGS / @thodson-usgs
- Ryan Abernathey / LDEO / @raternat
-
-

Agenda

XXIX. Props

- A. [Raphael] Charles + Julius for a billion PRs last week
- B. [Charles] Julius for coming to LA!
- C. [Charles] Raphael for getting the BQ stuff factored out of CMIP6
- D. [Ryan] Charles + Julius for the big push from LA!

XXX. Agenda

- A. [Tim] Demo recipe for USGS
 - <https://github.com/pangeo-forge/staged-recipes/pull/262>
 - [Raphael] Happy to run the pruned recipe on my local laptop
- B. [Ryan] Who should Pangeo Forge be accessible to?
 - Beam raises the bar of what's possible with data pipelines
 - But perhaps also raises the barrier to entry for deployment
- C. [Sean] Greg & Sean pushing from DevSeed
- D. [Charles] DaskRunner + cleaning up stale docs are my top priorities for the remainder of the calendar year. Planning to present DaskRunner at Dask Demo Day in December.
- E. [Tim] A basic quickstart would be great! A public repo that can be run with pangeo-forge-runner from a public repo
- F. [Sean] How does xarray know what are coordinate and non-coordinate dimensions? As it relates to kerchunk.
 - <https://github.com/fsspec/kerchunk/issues/375>
 - <https://github.com/fsspec/kerchunk/issues/377>
 - [Ryan] MultiZarrToZarr is just "concat" ... we should be able to view virtual zarrs as an in-memory object which we can manipulate in known

ways. (Opinionated) recommendations:

- Upstream kerchunk into zarr
- Chunk manifest could then be an official zarr extension
- Can the zarr API then be extended to lazily manipulate a set of these virtual datasets
- [Ryan] Xarray infers dimension coordinates by:
 - Dimension names. If there is a variable with the same name as a dimension, then that's a dimension coordinate.
- [Sean] In the netCDF files I'm working with, arrays use dimensions from parent groups 😊
- [Ryan] SlideRule might be a better fit for icesat than xarray
<https://slideruleearth.io/web/rtd/index.html>
- [Sean] Lidar data is a difficult fit into xarray, but Datatree 🌲 is a game changer! "Xarray-datatree, a workaround for nasa's terrible decision making!"

2023-11-06

Attendance:

- Charles Stern / LDEO / @cisaacstern
- Raphael Hagen / CarbonPlan / @norlandrhagen
- Nathan Zimmerman / E84
- Yuvi / 2i2c
- Negin Sobhani / NCAR/ @negin513

Agenda

XXXI. Props

- A. Greg for Flink runner
 - +100000! (Charles)

XXXII. Agenda

- A. [Nathan] EMR Spark Runner
 - Starting on working on docs for Flink Runner (Great idea!! - Charles)
 - EMR serverless has various benefits: workers are provisioned on the fly, versions EMR non-serverless (which requires a persistent cluster).
 - Thinks a custom container would work for pangeo forge
- B. [Negin] [copied from whereby chat] My name is Negin Sobhani, HPC consultant and computational scientist at NCAR. Working on different projects at CISL but come from atmospheric chemistry background. I am interested how the beam runners can be used for some of the datasets at NCAR.

- Flink can probably work on HPC. Presumably will be more effort than Dask would be to deploy, but it's an available option for getting started today.
- C. [Charles] DaskBakery update
 - It's a work in progress! DaskRunner in Beam is less mature than we previously understood.
- D. [Raphael] Sean or Charles or... do you know if there are any examples of using xarray-beam transforms with PGF?
 - No current examples we're aware of
- D. [Sean] What is our roadmap for recipe CI management, discussion and support?
 - Action points:
 - deprecate/archive staged recipes
 - Charles + Sean go through docs and update everything
 - Recommendation for institutions to use a monorepo (document this)

2023-10-23

Attendance:

- Charles Stern / LDEO / @cisaacstern
- Raphael Hagen / CarbonPlan
- Tom Nicholas / [C]Worthy

Agenda

- XXXIII. Props
 - A. [Charles] Greg from DevSeed for Flink deployment efforts
- XXXIV. Agenda
 - A. Dask runner discussion
 - Charles is working on this, and Charles and Ryan are planning to discuss with a Dask maintainer about enlisting some Dask people to help
 - B. CarbonPlan maps + kerchunk data
 - Tom chatted with CarbonPlan about options here. Current maps frontend doesn't support kerchunk data (at all, but multi-resolution especially difficult)
 - C. [C]Worthy

- Dask runner likely a good option eventually, but near-term using native kerchunk + dask (without Beam/Pangeo Forge) is suggested to avoid being bottlenecked by DaskRunner development
- Skyplane was discussed as a possible option for pushing HPC data to AWS Public Data bucket.

2023-09-25

Attendance:

- Charles Stern / LDEO / @cisaacstern - offline in observance of yom kippur today, look forward to reviewing minutes tomorrow!
- Alex Merose / Google / @alxmrs
- Raphael Hagen / carbonplan / @norlandrhagen
- Chuck Daniels / Development Seed

Agenda

XXXV. Props

- A. [Charles] Raphael for keeping the PRs rolling!
- B.

XXXVI. Agenda

- A. [Charles] Aiming to land docs reset this week
- B. [Alex + Raphael] NDPyramid + PGF
 - How will this work with Xarray-Datatree?
 - Hope to have things just work together.
 - Integrated with Kerchunk on NetCDF references. pyramided on different model runs (ensemble).
 - GeoZarr?
 - Were chatting with Brianna about maybe submitting a ZEP about multi scales. Want to have the little bit of NDPyramid as a working example.
 - Can Alex Merose put Raphael in contact with a Googler who is looking into multiscale Zarr?
 - Yes!
 - Alex's team is focusing on integrating NDPyramid to Xarray-Beam.
 - Want to build `xbeam.Pyramid()` for Xarray chunks to Zarr.
 - Can we build this together?
 - PGF Indexes can directly map to Xarray-Beam Keys. Just need to build it.

- Include it as an optional dependency.
- Lots of web viz Carbon plan does is to create pyramids. Would be nice to have a piece of the pipeline.
- C. [Chuck] Using Kerchunk, bug on the Apache Beam side. Proposed solution for the bug.
 - related to pickling. just uses std pickle instead of dill or cloudpickle.
 - did not submit a PR. it's a couple line change, but what are the ramifications? See <https://github.com/apache/beam/issues/28558>
 - Got the non-kerchunk recipe working (Ryan's suggestions). Charles also helped with what to do next.
- D. [Raphael] Trying to run a std pipeline on a large machine. Also had a dead kernel.
 - <https://github.com/pangeo-forge/pangeo-forge-recipes/issues/618> – same issue as Ryan

2023-08-28

Attendance:

- Charles Stern / LDEO / @cisaacstern
- Alex Merose / Google / @alxmrs
- Raphael Hagen / carbonplan / @norlandrhagen
- Sean Harkins / Development Seed /

Agenda

XXXVII. Props

A.

XXXVIII. Agenda

A. <https://github.com/pangeo-forge/pangeo-forge-recipes/issues/583>

- [Charles] Will take a closer look at this today

B. Consolidated metadata

<https://github.com/pangeo-forge/pangeo-forge-recipes/pull/575>

- [Charles] Confused about what to expect here, can punt on this and do consolidated coordinate dims first
- [Alex] ARCO ERA5 doesn't find a major use case for consolidated metadata, but consolidated coordinate dims makes a big difference to loading time

- [Charles] We can move forward with <https://github.com/pangeo-forge/pangeo-forge-recipes/pull/556>
- [Charles] Climsim 13 TB dataset is “done”, but suffers from slow loading due to lack of consolidated coordinate dimensions
<https://github.com/leap-stc/ClimSim/issues/38#issuecomment-1687209517>
- [Alex] Delete operation which is required as part of `_gather_coordinate_dims` is tricky and error-prone, sometimes causing atomic failures.
 - Could coordinates be written in append-mode the first time? Would require locking, etc. :-|
 - But in ARCO ERA5, 10hrs to compute most of the zarr store, and then 2 days to do the GCP delete
- [Sean] How much of this might be ameliorated by the sharding spec (in zarr v3)?
- [Charles] Does sharding provide some solution for distributed writes + locking?
- [Sean] Not sure about writes, but reads are definitely way better
- [Raphael] Let's discuss sharded writes in the zarr-python working group
- [Alex] Ideas for improving performance of `_gather_coordinate_dims`
 - Consolidation is pretty fast
 - Deletion of “old” shards is very slow
 - Possibilities:
 - Offload delete to an out-of-band garbage collection process
 - Leave the original shards in the bucket, but just ignore them on dataset load
 - [Sean] How much could optimizations in the gcsfs side help?
 - [Alex] Gcsfs probably could help, by using GCS batch delete, but somehow the interaction with `xarray->zarr->gcsfs` is not working. To make this work, gcsfs, has to use batch delete, and we need to do this non-blocking (and non-looping). This could be opt-in, because there are lots of situations where we don't want to “trust delete” will happen.
 - [Alex] For now, re-implementing original version is fine. In increasing orders of difficulty:
 - Land things as we have them now, for parity with 0.9.4
 - Add flag to consolidated coords: `delay_delete`. Split overwrite operation into write, and delete, as separate

steps.

- Can we avoid having to delete in the first place?

(a) Distributed writes

(b) ... or write time at schema generation time?

🙄 If this transform knows which coordinates are consolidated (with a kwarg) then distributed writes can be avoided.

C. [Raphael] I started a PR to accept .parquet as a file input or the reference recipe transform.

- [Sean] This will be really useful. @Raphael, can you post link to Pythia tutorial?

- [WIP / feedback welcome] [tutorial](#). 👍

D. [Sean] Updated issue for appending to zarr?

- <https://github.com/pangeo-forge/pangeo-forge-recipes/issues/447>
- <https://github.com/pangeo-forge/pangeo-forge-recipes/issues/37>
- [Charles] I will close 37 after this meeting

E. [Charles] Docs issue

<https://github.com/pangeo-forge/pangeo-forge-recipes/issues/569>

- FastAPI tutorials: inspiration
- Prototype:
<https://github.com/pangeo-forge/pangeo-forge-recipes/pull/563>
-

2023-08-14

Attendance:

- Charles Stern / LDEO / @cisaacstern
- Alex Merosé / Google / @alxmrs
- Raphael Hagen / carbonplan / @norlandrhagen
- Ryan Abernathey / LDEO & Earthmover / @rabernat

Agenda

XXXIX. Props

- A. [Charles] Alex for help with Beam Coders issue.
- B. [Charles] Raphael for work on consolidating coordinate dims.
- C. [Ryan] Charles for implementing rate limiting, one of our most requested features!

XL. Agenda

A. [Ryan] Pangeo Forge / Arraylake integration? What would be needed

- Early on, Pangeo Forge might've been ETL + cataloging tool, but that scope was too broad. Now PF is an ETL tool.
- Arraylake is a cataloging tool.
- Arraylake leverages deeply nested groups.
 - How can we get “please add to this Zarr group” via PF?
 - [Charles]: Right now, when one has this Zarr group object, somewhere it's instantiated with a URI along the way.
 - [Ryan]: There's a repo path, then there's the repo itself.
 - We don't have a URI is the bottom line.
 - [Charles] The current status with PGF Runner, on a high level, is that anything that we want to inject at runtime, ... we can inject Python objects and kwargs at runtime. Anything that's injected at this point does require a PR to PGF runner explicitly requiring an injection into PGF runner. This is not sustainable. In Convo with Yuvi – a great desire is to transition this design into some type of entry point runner system, s.t. a plug in backend can be defined.
 - As it stands now, needs hardcoding. If it's a string, then we can just pass it to the correct point. If we need to instantiate a python object, then we need a way to parse a python object.
 - Do this kind of thing w/ an FSSpec cache. e.g. import gcsfs, gcsfilesystem, pass all the args as strings, etc.
 - As a first step to get it working, would there be a way to say, “need to instantiate this Zarr with these arguments”?
 - Alex: use setup.py / setup.cfg to add custom parameters
 - (a) <https://beam.apache.org/documentation/sdks/python-pipeline-dependencies/#nonpython>

B. [Alex] How would we integrate Xarray-Beam and Pangeo Forge? What would the translation transform look like?

- First, need to understand how the “keys” / “indexes” differ between the two projects
- Then, write a PTransform that translates PFR indexes to xr-beam keys and vice-versa
- [Ryan]: Long term, would like to see PGF and Xbeam merge
 - Need to see a few things surface to make this happen

- e.g. combining irregular chunked data, – does xbeam allow this? The fragments don't all have to be the same size; this is not a blocker.
- what is most different about PGF is that before you even materialize xarray datasets.
- xbeam: starting with open xarray datasets.
- Given the modular nature of beam, there's no reason why you can't plug them together
- PGF introduces an idea of a Schema. In retrospect, it would have been simpler to skip the schema and create an xarray dataset. Thinking was: don't want dask involved.
- Right now we do this thing where if we want to init a template dataset, or just pass around a lightweight version of a ds, we load it with dask to not materialize the data. this is to keep things light. it's been discussed to expose the idea of a template or a schema s.t. you can have a dataset outline to have this outlined w/o dask.
- [Ryan]Start: translate keys to indexes
 - should be able to map an PGF index to a xb keyed dataset
 - Should be able to translate between systems both ways.
 - Getting to this point would reveal a deeper integration.
- Eventual: all Xarray stuff lives in Xbeam. IO lives in PGF. Makes Ryan nervous to make a big change
- [Charles] xbeam does it always need the values of these keys to always be values of the datasets?
 - [Ryan] Yes, and we have that in various stages of our pipeline.
 - it should be able to hand off to xbeam.

C. [Ryan] Status of Beam / Flink on AWS?

- [Alex]: I recommend Apache Spark on [insert cloud here]
 - [Charles] Do various cloud providers' Spark runners support Beam Python SDK? Yuvi raised some concerns about this re: AWS.
 - Sean - Has never seen Beam run successfully on AWS EMR
 - <https://beam.apache.org/documentation/runners/spark/>
- [Ryan] Spark is the more mature batch executor, why are we using Flink? Could we deploy Spark on K8s? Separate question from managed service vs. K8s.
- [Charles]: K8s + FlinkOperator works (better than expected?)

<https://github.com/pangeo-forge/pangeo-forge-runner/issues/19#issuecomment-1664668473>

D. [Charles] ClimSim update

- Concurrency limiting works
- Caching took a long time (change cache bucket expiration)
- Failed because I made chunks too big?
- Will rerunning take too long?

E. [Charles] NASA presentation with Brianna this week (Weds)

- [Raphael] Q: Is this open? → [Charles] I think no?

F. [Charles] Documentation + integration testing

- Linking -runner + -recipes stories
- I like FastAPI + Uvicorn as an example
- Proposal: In ``-recipes`` docs, don't demonstrate Beam-direct deployment
- [Sean] 👍
- [Alex] "Magic" is dangerous, make sure to mention
- [Sean] Runner needs a lot of love. How much of runner should we be yanking out, and focus purely on templating/configuration? Can we get rid of a lot
- [Alex] Weather tools does something similar to runner. Taking advantage of custom pipeline options may be a good idea. Can we use a lower-friction experience of using Beam's native interface(s)? Setup.py would be one such thing.
- [Sean] Is there some kind of standalone dashboard for surfacing errors/logs?
- [Alex] Spark runner may provide something like this. There probably isn't a direct runner dashboard out-of-the-box.
- [Alex] What if it's "just a python script" with a setup.py? Pure python is better than a custom reflection/code generation framework.
 - [Charles] Can injection be replaced with reserved environment variable names populated by a default factory?
- [Sean]
 - https://github.com/google/xarray-beam/blob/main/examples/xbeam_rechunk.py#L35
 - import unittest

Python

```
import unittest

if __name__ == '__main__':
    unittest.main()

## pgf

import pangeo_forge_recipes as pgr

if __name__ == '__main__':
    # This calls argparse and gets all the CLI args for running a recipe
    pgr.main()

##

recipe = PTransform()

if __name__ == '__main__':
    parser = argparse.subparser()
    pgf.main(parser)
```

None

```
# now
pangeo-forge-runner <script>

python <script>.py
```

G. [Raphael] There have been some new recipe proposals on [PFR](#). Is the current web- infrastructure working?

- [Charles] Nope :-/ ... not all, but some of the proposed recipes (e.g. by Ali Chase) are people I've asked to make notes, with the understanding that these would not be run on the "public" instance.

XLI.

2023-07-31

Attendance:

- Rich Signell / USGS / @rsignell-usgs
- Tim Hodson / USGS / @thodson-usgs
- Raphael Hagen / carbonplan / @norlandrhagen
- Charles Stern / LDEO / @cisaacstern
- Alex Merose / Google / @alxmrs

Agenda

XLII. Props

- A. [Charles] USGS crew for pushing enthusiasm on flink + dynamic chunking
- B. [Raphael] Brianna for plugging pangeo-forge at ESIP a bunch

XLIII. Agenda

- A. [Rich] Post our SciPy Sprint to try to install a Flink runner on AWS to run beam pipelines , at the ESIP Summer meeting, Brianna pinged Yuvi and Yuvi fixed [his AWS runner](#). He then sat with Sean Harkins & Rich Signell in a 1.5 hour screenshare session to try to get the AWS runner working on a USGS linux system. Yuvi
 - [Alex] AWS Kenesis - managed flink. Yuvi's K8s/terraform setup is impressive + portable. But also there's the "buy rather than build" philosophy. Fallback if Kenesis doesn't work... could try AWS managed Spark service. (Flink is stronger than Spark for streaming, but Spark makes sense for batch possibly.) Alex suggests perhaps trying to not do devops at this point, just to get things going on AWS. Rather than "depth-first" focus on a single runner (i.e. K8s), try a "breadth-first" exploration of all the options (managed Flink, managed Spark, etc.).
 - [Charles] I like Alex's suggestion. For "regular developers" (myself included) managing clusters dilutes focus a lot.
 - [Alex] Maybe knows someone at AWS who is a leader in Beam.
 - [Rich] Someone at AWS who wants to support the community with Beam would be really valuable.
 - [Alex] In the spirit of breadth-first, how about trying out the Dask Runner? It's new but possibly useful bc USGS folks are more comfortable with Dask.
 - [Alex] Of the managed AWS options, Spark may be the best option.
 - [Rich] Was hoping to not have to learn Flink or Spark. Would be great to only have to know about Dask.

- [Rich] Summarizing the discussion, maybe the best next step is to connect with someone at AWS who has used Beam on managed services. Or follow a tutorial for that.
- [Alex] Kinesis + Flink are primarily streaming platforms focussed on the hard problem of managing a firehose of pub/sub data. Whereas Spark is optimized for batch. AWS tutorials for running beam on managed Spark.
- [Rich] Dask is the most ideal option if it can work.
- [Alex] My Dask Runner v0.2:
<https://github.com/apache/beam/pull/27618>
- [Charles] Let's discuss follow-up as issues (or possible tutorial docs) on:
 - <https://github.com/pangeo-forge/pangeo-forge-runner> ?
 - Or <https://github.com/pangeo-forge/docs> ? 🤔
 - Most focus has been on `pangeo-forge-recipes`
 - Also see:
<https://github.com/pangeo-forge/pangeo-forge-recipes/issues/545>
 - Where execution docs/tutorials should live is an open question, related to integration testing.
- [Alex]: Spark info
 - <https://beam.apache.org/documentation/runners/spark/>
 - Spark is offered as a managed service on *all* major clouds (GCP, Azure, AWS). This may be the one ops thing to standardize across institutions. [:tada: - Charles]
- B. [Charles] cstern@ldeo.columbia.edu - @Tim and @Rich - email me!
- C. [Rich] Tom Nicholas SciPy talk on Cubed
- D. [Charles] Working on [ClimSim](#) + [github action](#)
- E. [Charles] For an upcoming meeting (with more attendees): appending design review <https://github.com/pangeo-forge/pangeo-forge-recipes/issues/447>

2023-07-17

Attendance:

- Ryan Abernathey / LDEO & Earthmover / @rabernat
- Charles Stern / LDEO / @cisaacstern
- Tyler Erickson / VorGeo / @tylere
- Derek O'Callaghan / UCD / @dgocallaghan
- Tim Hodson / USGS / @thodson-usgs

- Raphael Hagen / carbonplan / @norlandrhagen (I'm on a plane and whereby doesn't work). Congrats on the blogpost! - 😊 Thanks Raphael - Charles
-

Agenda

XLIV. Props

- A. [Ryan] Charles for shipping the release and blog post!

XLV. Agenda

- A. `0.10.0` release!
 - <https://medium.com/pangeo/pangeo-forge-is-all-in-on-apache-beam-d7370299405f>
- B. [Derek] What is the current status of staged-recipes ?
 - [Charles] Currently recipe submission infra is unmaintained. We should soon transition to using the GitHub Action.
- C. [Derek] Is the plan still to tranisi
- D. [Ryan] Big picture: transitioning in last year of grant from “what can we build?” to “what will be the most impactful contributions we can make, that can be maintainable/sustainable beyond NSF grant?”
 - [Tyler] Even a lightweight version of a public service (staged-recipes, etc), while laudable, is going to be difficult to maintain without funding.
 - [Derek] A staged-recipes service is a great way to encourage participation, but a working solution is very difficult to maintain in practice.
 - [Tim] Agencies just need tooling.
- E. Release candidates + testing of `-runner` + Action
 - <https://github.com/pangeo-forge/pangeo-forge-recipes/issues/545>
- F. Data catalog stewardship going forward
 - <https://github.com/pangeo-forge/pangeo-forge.org/issues/404>
- G. Ryan Sentinel NDVI pipeline and Arraylake demo
- H. Idea for next meeting: discuss design for append-only pipeline

2023-06-05

Attendance:

- Ryan Abernathey / LDEO / Earthmover / @rabernat
- Julius Busecke / LDEO / @jbusecke
- Charles Stern / LDEO / @cisaacstern

- Alex Merose / Google / @alxmrs
- Tyler Erickson / VorGeo / @tylere
- Brianna Pagan / NASA / @briannapagan
- Katie Brennan/ TGS / @kbren
- Raphael Hagen / carbonplan / @norlandrhagen

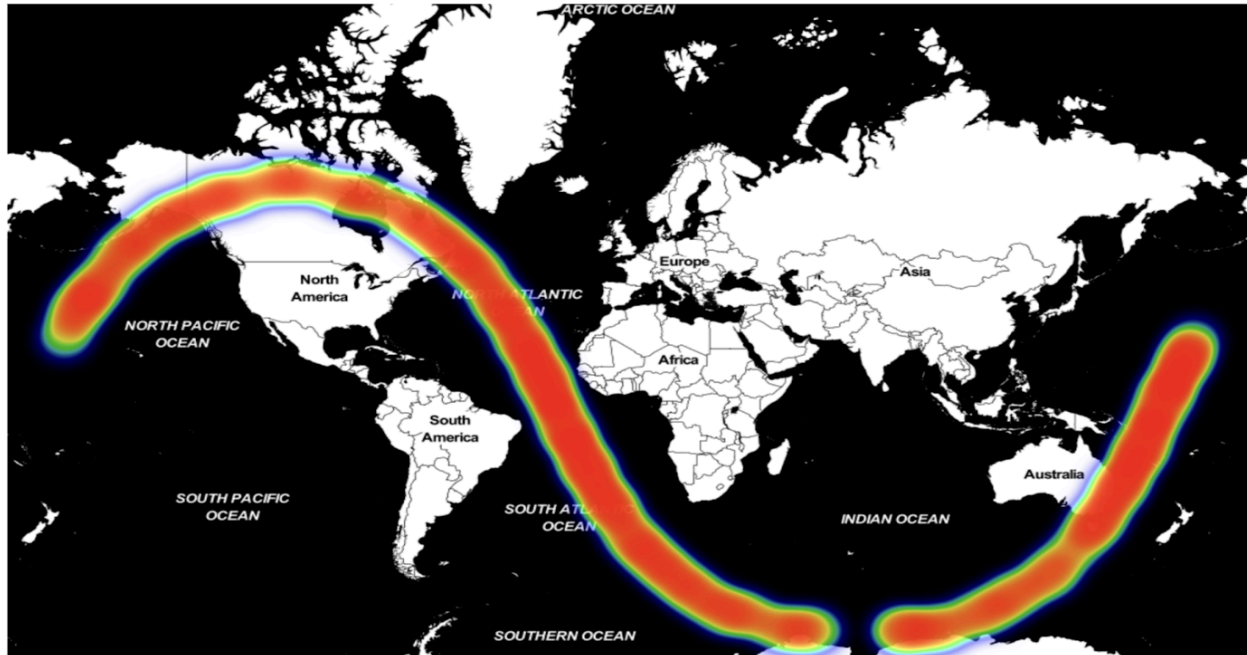
Agenda

XLVI. Props

- [Ryan] Julius for continuing to push CMIP6 data through Pangeo Forge
- [Julius] Charles for continuing to improve the [github action](#) in support of CMIP6 recipes and LEAP data ingestion. Example [‘feedstock’](#)
- [Charles] Brianna for landing the big upcoming Pangeo Forge presentation @ NASA
- [Charles] Alex for offering to work on beam refactor release blocker issue(s)
- [Charles] Katie & Tom at TGS for interest in Pangeo Forge + Dataflow
- F.

XLVII. Agenda

- Walkthrough of LEAP github action from Julius?
- Update on Beam refactor progress?
 - Walked through open `release blocker` issues as a group, and assigned point-person for each open issue, with exception of <https://github.com/pangeo-forge/pangeo-forge-recipes/issues/508>, for which Ryan will follow up with Derek to coordinate a solution.
 - The group agrees that, barring some major unforeseen bug surfacing, we should hold ourselves to a `release blocker` labeling “freeze” starting now, and plan to release `beam-refactor` branch once the current open issues are resolved.
- Alex has a question for Brianna about NASA data!
 - GPM data doesn’t seem to fit the Zarr model
 - Moving satellite
 - You’re thinking of L2 data which is alot harder to fit in - but maybe you can link which GPM you’re talking about?
 - All the zarr data we’re producing at NASA is level 3/4 actually this is why our whole Giovanni tool doesn’t work for L2 data, it is VERY difficult to generalize and visualize L2 data. Not sure how much L2 data GEE has right now. This is a common request from us though, but the assumptions are not very consistent



GPM/2020/02/11/2B.GPM.DPRGMI.CORRA2022.20200211-S174902-E192136.033837.V07A

2023-05-22

Attendance:

- Ryan Abernathey / LDEO / Earthmover / @rabernat
- Charles Stern / LDEO / @cisaacstern
- Katie Brennan / TGS / @kbren
- Alex Merose / Google / @alxmrs
- Brianna Pagán / NASA / @briannapagan
- Tyler Erickson // @tylere
- Thomas Hill / TGS

Agenda

XLVIII. Props

- A. [Ryan] Brianna for pushing on GeoZarr
- B. [Charles] Alex Merose for coordinating GEE meeting (and traveling to attend!) and of course Brianna for being an integral part of the meeting!
- C. [Brianna] Charles for pushing the governance conversation here, organizing a solid meeting with Google folks

XLIX. Agenda

A. Roadmap?

■ Merge beam-refactor

- Charles: Primary working branch for PGF.
 - High priority
 - Currently spread thin working on other things.
 - Suggestion: From here on, will totally focus on closing all beam refactor issues. Want to have merged ASAP.
- Ryan: Why do the Beam refactor? It lost some momentum.
 - Give us more flexibility for the types of recipes that can be supported.
 - (a) pgf v1 was “hard coded” for the NetCDF → use case.
 - (b) found that the exec fw (rolled in house) was not flexible enough; there was endless scope – had to extend fw for each use case
 - (c) deployment: also complicated issue
 - (d) instead of inventing a workflow engine, let’s use one.
 - (e) Beam, b/c
 - (i) Google uses it
 - (ii) Matches PGF exec methods
 - (iii) works in many environments.
 - (f) New Features
 - (i) multi-dim concat for input datasets (got for free, not possible before). Useful for mosaics and fc datasets
 - (ii) flexibility to insert custom stages into the pipeline (not hard coded)
 - (iii) can deploy in GCP dataflow, Spark, Flink, existing big data systems.
 - (g) 98% of the way there. Bugs are getting in the way. Want to fix these before merging.
 - Should we release it, bugs & all?
 - (a) what use cases work vs what doesn't?
 - (i) all the tutorial cases work
 - (ii) CMIP data has a time encoding issue.
<https://github.com/pangeo-forge/pangeo-forge-recipes/issues/508>
 - (b) Integration testing approach:

- (i) Tutorial notebooks insufficient (CI is broken I think)
 - (ii) Need more integration testing that actually runs in CI, to capture these edge cases
- Most-requested features (for beam-refactor):
 - Append-only pipelines (or fixing internal chunks)
 - Streaming (or NRT/continuously updating batch)
 - GeoTiff -> COGS -> GEE pipeline
 - Other runners, on other clouds: Flink, Spark, etc.
 - ...
 -
- Ensure support for beam-refactor in pangeo-forge-runner and <https://github.com/pangeo-forge/deploy-recipe-action>
- Big picture questions:
 - Should we be more explicit that pangeo-forge-recipes is now “just” a Beam SDK? Renaming? Therefore, Should we attempt to upstream pangeo-forge-runner to Apache Beam (or at least make it generic enough that this would be possible) as a general templating / version control -based deployment CLI? (Then, [deploy-recipe-action](https://github.com/pangeo-forge/deploy-recipe-action) also becomes generic.)
 - [Sean] Maybe not a bad direction to go, but would want a timeline for turn-off of public service. Before we advertise this, we’d want proven deployments of non-GCP runners.
 - [Alex] deploy-recipe-action could be really general, and “given” to Apache. But the domain knowledge of pangeo-forge-recipes is much more NumFOCUS-y. We should think more strategically with Pangeo Forge Governance before making decisions here. Google Open Source may be able to offer advice on strategy re: open-source governance.
 - [Ryan] Geo-Beam is a precedent. Xarray-Beam is another precedent. Should we just plan to merge Pangeo Forge Recipes with Xarray-Beam? Ultimately, combining them might make both projects better. Exporting a bunch of high-value, really useful beam transforms is a good place to start. 11 months is the time we have to make this useful enough that others will want to use it and contribute to it (roadmap and developer docs help with this). What is less clear is how to keep the service running.

- Future of the “public” service (compute & storage). Who runs it? Is it single-instance or multi-instance? If multi-instance, how do we avoid
 - [Sean] One downside of moving away from the hosted service is the public discussions of datasets and edge cases. If we run things in a more distributed way, how do we maintain this public discussion and body of knowledge and public discussion.
 - [Tyler] What has the cost of this been?
 - [Ryan] The expense is all personnel.
 - [Charles] Scaling is super hard, I’ve needed to hand-hold
 - [Ryan] Beyond the specific engineering cost, the bigger question is what is the engineering cost and complexity of the initially proposed service? Our NSF grant is 1.5M minus 60% overhead. The project is not funded/scoped to do this. In hindsight, this is probably a 20M engineering feat. What we do have the resources for is to prototype the foundation of an open source toolkit which can eventually become a component of such a service.
 - [Alex] Can this be solved in a distributed fashion at the governance level?
 - [Sean] Sees low fiscal appetite for this at MS, Amazon, etc.
 - [Brianna] Didn’t Google Public Datasets say they want to throw money into this?
 - [Alex] Could write a doc which would make a business case for the existence of lots of ARCO datasets. This could be lucrative for platforms (if the right protocols were in place to provide the data).
 - [Tyler] Radiant Earth seems like their in a good position to broker the conversation with cloud providers, since this is their core mission.
 - [Ryan] Earthmover specifically aims to operate efficient services in the community. That is different from OSS library. That is very different from a public SaaS. The SaaS is very different than a software package. We need to be clear what is this collaborative “thing” we are talking about. Public datasets? That’s a clear point of collaboration for the cloud providers and data providers. A non-profit such as Radiant Earth is possibly

well-positioned to lead the SaaS aspect, as might be a start-up that sees the business case, such as Earthmover. Hunch is that nothing is going to happen without a champion who is going to push this to happen, and that champion is likely to be an individual, not a distributed group.

- B. Possible TGS involvement: documenting GCP Dataflow deployment! (Either manually or via GitHub Action.)
- C. Ryan: Arraylake / Sentinel Demo
- D. Working on Prototype: GPM Level 2B data → EE via Zarr
- E. Governance
- F. GEE Recap
- G.

2023-05-08

Attendance:

- Charles Stern / LDEO / @cisaacstern
- Raphael Hagen / carbonplan / @norlandrhagen
- Katie Brennan/ TGS / @kbren

Agenda

- L. Props
 - A. [Charles] Raphael for leading the charge on kerchunk PR to beam-refactor, which is now merged!
 - B.
- LI. [Charles] beam-refactor status
 - A. <https://github.com/pangeo-forge/pangeo-forge-recipes/issues?q=is%3Aissue+is%3Aopen+label%3A%22+release+blocker%22>
- LII. [Charles] Roadmapping
 - A. I will make a PR for this this week.
 - B. A clear gap is linking the various parts of pangeo forge together
 - Pangeo Forge Recipes
 - Pangeo Forge Runner
 - <https://github.com/pangeo-forge/pangeo-forge-runner>
 - C. In-person meeting with Google Earth Engine (GEE) next week
- LIII. [Katie]

LIV. [Sean]

LV. TGS follow up - cstern@ldeo.columbia.edu , Katie.brennan@tgs.com

2023-04-24

Attendance:

- Charles Stern / LDEO / @cisaacstern
- Raphael Hagen / carbonplan / @norlandrhagen
- Alex Merose / Google / @alxmrs
- Derek O'Callaghan / UCD / @dgocallaghan
- Tyler Erickson // @tylere

Agenda

LVI. Props

- A. [Charles] Raphael for the productive pair sessions on <https://github.com/pangeo-forge/pangeo-forge-recipes/pull/486>
- B. [Ryan] Props to Charles for leading the roadmap exercise

LVII. Review of beam-refactor release blockers:

- A. <https://github.com/pangeo-forge/pangeo-forge-recipes/labels/release%20blocker>
- B. <https://github.com/pangeo-forge/pangeo-forge-recipes/pulls?q=is%3Apr+is%3Aopen+label%3A%22release+blocker%22>
- C. Create an adaptor layer to xarray-beam?
 - Alex has started this
 - What other features should we group in
 - Appending to Zarr
 - Infinite file patterns (streaming)

LVIII. Roadmapping

- A. **Goal:** Translate [these notes](#) into a public-facing document. TL;DR:
 - Release a core end-to-end “product”, namely a functional release of beam-refactor, with a clearly documented way to run/extend it. Details:
 - Release beam-refactor
 - Ensure pangeo-forge-runner can deploy beam-refactor
 - Shore up relationship between pangeo-forge-runner + meta-yaml-schema
 - provide a Github Action for deploying beam-refactor
 - Decide on synchronized release cadence for all these things?

- Any precedents we can use for this?
 - Thus far basically totally overlooked, but now increasingly important.
 - Centralize and clarify documentation (while cleaning up stale repos/pages)
 - Includes deciding what to do with pangeo-forge.org, particularly catalog.
 - (Side note on docs: remove tutorial notebooks in favor of integration-tested scripts.)
 - Goals for extending beam-refactor:
 - Multiple concat dims
 - Dataset fragments?
 - Streaming/glob patterns
 -
 - Future of staged-recipes
 - Github App (“just” useful for this type of public-facing/ “untrusted” deployment environment.)
 - (Governance is a separate / internal discussion for these meetings and subgroups thereof)
- B. **Content:** The roadmap naturally pairs with some type of high-level overview docs (which currently reside in `pangeo-forge-recipes`). It’s difficult/impossible to discuss the roadmap without discussing the relationships of the full stack, including:
- pangeo-forge-recipes
 - <https://github.com/pangeo-forge/meta-yaml-schema> (so far mostly overlooked)
 - pangeo-forge-runner
 - Github Action
 - (Github App)
 - (Pangeo-forge.org)
- C. **Implementation:**
- Option 1: Continue to use roadmap repo, with major overhaul, i.e. <https://github.com/pangeo-forge/roadmap/pull/55>
 - Option 2: New roadmap tab on pangeo-forge.org
 - Option 3: Dedicated pangeo-forge/docs sphinx repo (precedent for this?)
 - Option 4: ?
- D. Discussion:
- [Ryan] If four things need to be released together, they should probably be in a single repository. (pangeo-forge/pangeo-forge). Multiple

packages can also be deployed from a single repo. Also solves docs problem.

- [Alex] +1 on monorepo.
- [Sean] Initially leans more in the direction of -1 on monorepo, but comes to see the value based on discussion.
 - What are examples of deployment pain points?
 - Testing doesn't catch integration issues between these components
 -
 - What would be part of a monorepo?
 - Recipes, runner, meta-yaml-schema
 - And what would be deprecated?
- [Tyler] Keep it as simple as possible for as long as possible
- [Derek] +1 on recipes and runner in the same repo
 - Would it make sense to have notebooks in Pangeo Gallery?
 - [Ryan] Gallery is unmaintained, but yes to Pythia.
- [Tyler] What will governance look like after NSF funding? Others will be looking to that to assess whether they

LIX.

2023-04-10

Attendance:

- Charles Stern / LDEO / @cisaacstern
- Ryan Abernathey
- Raphael
- Tyler // @tylere
- Derek O'Callaghan / UCD / @dgocallaghan

Agenda

LX. Props

- A. Julius (+Charles) for doing lots of CMIP6 recipes and finding useful bugs in the process
- B. Alex Meroze for intro to GEE team
- C.

LXI. Agenda

- A. Report on meeting with GEE team

- Motivated by Simon's (data ingestion lead) interest in integrating open source methods into their data ingestion process. They already use Beam.
- Simon: Narrow goal: can we use Pangeo Forge for ingesting to GEE
 - Simon is a long-term member of the team, always has done ingestion
 - At low level, should be easy
 - Work is not open source because we do a lot of parallelization
 - Common recipe format would be good goal
- 2. They can help us create a CoG-based asset
- 3. How to do incremental updates?
 - These are the main struggle for the team
- Steve: Motivated by sustainability impact
- Mike Jeffries
- My Q: what do their pipelines look like?
 - Mostly fairly simple
 - Landsat: used USGS docker container, 1:1
 - Sentinel 2 -> Dynamic World (multiple products from a single file)
 - Not simply a file transfer; running something on the data
- Alex: weathertools already does many of these things. Would like to contribute it to Pangeo Forge.
- [Tyler] The most important aspect is regularly-refreshing datasets, things which change/evolve over time.
- [Ryan/Tyler] GEE team is probably less interested in the specific features we have so far, and possibly more interested in the community helping write recipes.
- [Charles/Tyler/Ryan/Derek] Simon expressed interest in abstractions that can express pieces of / partial datasets, to aid debugging of large datasets. Tyler confirms that small bugs in large pipelines, requiring restarting, is an issue seen at GEE, including in vector pipelines (e.g. malformed polygons). Slow iteration cycles is one of the major inherent challenges of data engineering (as opposed to frontend engineering). Would "checkpointing" of pipelines be useful? This may be "easier" than in some contexts, insofar as the structure of the datasets is "flat", and replacing "holes" *may* not need awareness of the state of other chunks. Two types of errors we see in PF are source data issues and logical issues in processing. Simon confirms that missing source files is a common issue seen in PF pipelines, and the ability to continue jobs in

this case may be useful. Though Tyler observes there may be situations when error/failure is desired (e.g., to save cost).

B. Roadmapping exercise from Friday

- [Pangeo Forge Roadmapping - April 2023 Notes](#)
- Goal: get this completed and visible on public website. Charles will take lead, anyone else welcome to contribute.

C. Ongoing bugs / feature dev

- Time encoding
<https://github.com/pangeo-forge/pangeo-forge-recipes/issues/508>
- Derek working on documentation
 - Excluded reference recipe
-
- Reference recipe
 - Raphael has time!
<https://github.com/pangeo-forge/pangeo-forge-recipes/pull/486>
- Need tests for
<https://github.com/pangeo-forge/pangeo-forge-recipes/pull/506>

D.

2023-03-27

Attendance:

- Charles Stern / LDEO / @cisaacstern
- Ryan Abernathey
- Tyler Erickson / / @tylere
- Martin Durant / Anaconda / @martindurant
- Raphael Hagen / CarbonPlan / @norlandrhagen
- Julius Busecke / LDEO / @jbusecke

Agenda

LXII. Props

- A. [Charles] Julius Busecke for so much enthusiastic Pangeo Forge hacking last week. (And Yuvi, for his insight to factor pangeo-forge-runner i

LXIII. Agenda

- A. [Charles] Updates on Pangeo Forge Cloud / GitHub App development. Lots of threads:
- Support arbitrary versions of pangeo-forge-recipes (e.g. beam-refactor)
 - Remove postgres database
 - Testing (side effects, side effects, side effects!)

- [Ryan] We're allowed to break production. We don't have customers.
- Bigger questions: what should be an Action and what should be App?
 - [Charles] A main advantage of the App is insulating beam runner (e.g. Dataflow) credentials from recipe contributors.
 - [Ryan] To the group: how do people want to use Pangeo Forge?
 - [Sean] Having an endpoint for job submission and job completion webhook (notification) would be ideal.
 - [Charles] Job completion notification is another benefit of a standalone App, as it provides a public endpoint to post notifications to.
 - [Sean (and others)] Automated notifications would be helpful, especially in the case of continuously updating datasets.
 - [Julius] Notifications of failed jobs matters a lot, considering that I plan to be submitting up to millions of recipes (as a stretch goal).
 - [Ryan] The conclusion seems to be that maintaining the ability to run recipes from an app installed on staged-recipes is valuable... but this should be as thin a wrapper as possible around the public Action(s).
- B. [Julius] - CMIP6 recipes
 - Started to run PGF recipe on LEAP dataflow [Repo](#)
 - Challenges:
 - Query ESGF API to get urls
 - My own duct tape client [pangeo-forge-esgf](#)
 - Been experimenting with [esgf-pyclient](#) (not async)
 - How to handle possibly millions of recipes?
 - Build many many heterogeneous datasets
 - Netcdf3 vs 4 solved by beam-refactor
 - Chunksize should be defined as size (MBs) not fixed time steps - [This](#) was surprisingly easy!
 - How to execute many recipes as a 'local' power user (currently invoking pangeo-forge-runner on my laptop terminal). And how to keep book about what has happened in a similar way as pangeo-forge-cloud?
 - Things kinda work!
 - Mysterious time out errors on dataflow
 - Need to refactor my recipe onto the latest beam-refactor (this [bug](#) has been blocking me there).

- Discussion:
 - How to capture metadata about particular runs?
 - And how to have tests? Which can also be crawled over existing stores, not just triggered on job completion.
- C. [Ryan] Release beam refactor? What is our release checklist looking like?
 - Charles: yes
 - Let's get
<https://github.com/pangeo-forge/pangeo-forge-recipes/pull/486> in first

LXIV.

2023-03-13

Attendance:

- Ryan Abernathey / LDEO / Earthmover / @rabernat
- Tyler Erickson / free-range / @tylere
- Martin Durant / AnAnaconda / @martindurant
- Brianna Pagán / NASA Goddard / @briannapagan

Agenda

LXV. Props

- A. Yuvi did a great demo to NASA internal tech spotlight a few weeks ago, well received and trying to follow up on some threads

LXVI. Beam refactor

- A. [Charles] I am on vacation this week. I have been working on <https://github.com/pangeo-forge/pangeo-forge-orchestrator/pull/234>, which removes the Postgres database from Pangeo Forge Cloud, in line with our recent discussions about reducing maintenance overhead.
- B. [Brianna] made a governance-model branch on the roadmap repo, will submit PR later today summarizing the proposal from convos with Yuvi/Charles
 - <https://github.com/pangeo-forge/roadmap/issues/53> < suggestions from Tyler
- C. [Ryan]
 - Merged <https://github.com/pangeo-forge/pangeo-forge-recipes/pull/425>
 - Updated opendap tutorial - <https://github.com/pangeo-forge/pangeo-forge-recipes/pull/500>
- D. [Martin]
 - A bunch of releases (intake / fsspec)
 - Feature development in Intake

- Working on Kerchunk again (large reference sets in parquet storage)
- E. [Tyler]
 - What's our story on L2 Satellite data
 - Store as COG files -> quick transformation on the fly
 - For visualization
 - <https://developmentseed.org/titiler/>
 - But what about for analytical processing (composites, mosaics) etc?

2023-02-27

Attendance:

- Charles Stern / LDEO / @cisaacstern
- Ryan Abernathey / LDEP @rabernat
- Derek O'Callaghan / UCD / @dgocallaghan
- Martin Durant / Anaconda / @martindurant
- Yuvi
- Martin Durant
- Tyler Erickson
- Raphael Hagen / CarbonPlan / @norlandrhagen

Agenda

LXVII. Props

- A. [Charles] Yuvi and Brianna for preparing an awesome PF demo for NASA (happening today!)
 - https://docs.google.com/presentation/d/1hGaI7v5wqu_3j5WqpkJSZbN_JST6Cq33R2dqIVbCmIHk/
- B. [Ryan] Charles for thinking about the future of Pangeo Forge and making a plan to simplify our architecture
- C. [Ryan] Yuvi for an amazing demonstration of how to use PF as a GitHub action
 - https://github.com/pangeo-forge/GPM_3IMERGDL-feedstock/pull/3/files#diff-681fc284c4579f51fc35fb63fae57c34a8ae6655f6624386e25349aee4cfc70d

LXVIII. Agenda

- A. Martin has made progress on kerchunk + parquet + rust integration
 - <https://github.com/martindurant/rfsspec>
- B. Yuvi and Brianna gave NASA presentation today
 - 96 attendees!
 - Demo worked
 - Talked about the GitHub integration
 - Most of it was demo

- [Can you share materials (slides / code)?]
 - https://docs.google.com/presentation/d/1hGal7v5wqu_3j5WqpkJSZbNJST6Cq33R2dqIVbCmIHk/edit#slide=id.g15b8dde9c50_0_148

C. [Charles] Preparing Orchestrator for the After NSF Times

- Extra props to Yuvi here!!!
- <https://github.com/pangeo-forge/pangeo-forge-orchestrator/issues/233>
- Implications for frontend (no database)
- Actions vs. App
 - Actions can run everything *_except_* production deployments (**for situations where the recipe contributor does not also own the deployment creds**), including pruned test with LocalDirectRunner, linting
 - Actions *can* easily deploy to prod *if the recipe contributor also owns the deployment creds*
 - In situations where some org/institution/company wants to support a beam runner for a group of users (lab members, employees, community, etc.), *without granting them ownership of executor creds*
- What is a bakery? Where does the interface between Actions/App + bakery?
 - Bakery is a beam runner
 - ***new*** Bakery should also run an Agent service to deploy the jobs to the runner. This allows bakeries to “own” their deployment creds, and not have to hand them over to the GitHub App.
 - GitHub App POSTs to Bakery Agent to deploy
- Questions:
 - [Ryan] What becomes of the website? The main feature which is missing is a catalog. Catalogs are very stateful. Catalogs are the essence of state. Something else needs to provide a catalog.
 - [Yuvi] The state has not disappeared, it’s just stored in GitHub. The source of truth lives alongside the recipes, on GitHub. The frontend can cache data from GitHub, but not have the source of truth.
 - [Yuvi] Catalog for end users to find data is one type of state. Another type of state is operational logs. Separation of these concerns will be valuable.
 - [Yuvi] In the conda-forge analogy, Anaconda hosts catalog and

build artifacts (i.e. data).

- [Ryan] Target for **data** are public dataset programs (OSN, AWS, etc.)
 - Home for the catalog(s) remains an open question. Earthmover is building catalog tooling, which may be part of the answer.
 - [Charles/Yuvi] Moving towards a multi-instance world, like Jupyterhub, etc. The pangeo-forge github org becomes a public demonstration, but not the only instance.
- D. [Tyler] (copied from whereby chat) “Have you been in contact with Hamel Husain (<https://www.linkedin.com/in/hamelhusain/>)? He is a great contact for anything to do with GitHub actions (and other GitHub functionality). He also works on nbdev, which has useful tools for cleaning up notebooks before committing to GitHub.”
- E. [Yuvi] GitHub actions for doing test runs + py.test based simple tests
- https://github.com/pangeo-forge/GPM_3IMERGDL-feedstock/actions/runs/4267939582 is an example
 - https://github.com/pangeo-forge/GPM_3IMERGDL-feedstock/pull/3 is the PR with action code
- F. [Ryan] Let’s review the open PRs in in Pangeo Forge Recipes
- [Derek]
<https://github.com/pangeo-forge/pangeo-forge-recipes/pull/487>
 - [Derek] Will create another PR for remaining tasks in
<https://github.com/pangeo-forge/pangeo-forge-recipes/issues/483>
- G. [Derek] Should we have an XarrayZarrRecipe template?
- [Yuvi] More maintainable long term to not have a custom object, and more composable (allows easily injecting custom stages/ptransforms).
 - [Yuvi] The `pf` command line tool may evolve into something that can do some meta-programming `init` functionality to provide some cookie-cutter
- H. [Charles] What should beam-refactor to be released as? 0.10.0 or 1.0.0 ?
- [Ryan] Favors
 - [Ryan] Let’s make a milestone for to release beam refactor

2023-02-13

Attendance:

- Charles Stern / LDEO / @cisaacstern
- Yuvi / 2i2c
- Raphael Hagen / CarbonPlan / @norlandrhagen

- Derek O’Callaghan / UCD / @dgocallaghan
- Alex Merose / Google / @alxmrs
-

Agenda

LXIX. Props

- A. [Charles] Raphael for a great effort getting kerchunk + beam-refactor underway, as well as many other PRs on pangeo-forge-recipes
- B. [Charles] Brianna for continuing brainstorm and efforts on governance models

LXX. Agenda

- A. [Charles] Update on beam-refactor in Pangeo Forge Cloud
 - Nearing the end of setting up dataflow integration testing
<https://github.com/pangeo-forge/pangeo-forge-orchestrator/pull/226>
 - Really happy with how this is working; it makes real PRs, and deploys them to dataflow, like this:
<https://github.com/pforgetest/test-staged-recipes/pull/59>
 - Once this is merged, I can resume getting beam-refactor into production, and use this test as a basis for ensuring nothing is broken
- B. [Charles] The future of Pangeo Forge Cloud. Beginning a brainstorm for what we’d like to get done in the roughly 1 year that remains of NSF funding (in reality it’s a little longer than that)
 - Manage less state
 - Probably no postgres database (we can replace recipe_run objects with GitHub check runs)
 - Probably no catalog?
 - Make compute and storage _much_ more configurable
 - Permissions/auth for certain user groups?
 - Serverless?
- C. [Yuvi] ESDS Tech Spotlight Talk by me & Brianna Feb 26
 - Hopefully show off beam-refactor on AWS
- D. [Derek] Porting documentation/tutorial notebooks to beam-refactor
 - <https://github.com/pangeo-forge/pangeo-forge-recipes/pull/487>
 - <https://github.com/pangeo-forge/pangeo-forge-recipes/issues/483#issuecomment-1422645726>
- E. [Raphael] (beam-refactor) reference recipe Q’s
 - <https://github.com/pangeo-forge/pangeo-forge-recipes/pull/486>
 - Alex provided a helpful live code review of this PR! 😊
- F. [Alex] Beam on Dask: Windowing + discussion

2023-01-30

Attendance:

- Ryan Abernathey / LDEO / @rabernat
- Charles Stern / LDEO / @cisaacstern
- Alex Merosé / Google / @alxmrs
 - I'm having rural internet problems :(
- Sean Harkins / Development Seed / @sharkinsspatial
- Brianna Pagán / NASA / @briannapagan
- Anderson Banihirwe / CarbonPlan / @andersy005
- John Clyne / NCAR / @clyne (clyne@ucar.edu)
- Tom Augspurger / Microsoft / @TomAugspurger

Agenda

LXXI. Props

- A. [Ryan] Brianna for championing [GeoZarr](#)! (Tangential to Pangeo Forge but still huge)
- B. [Charles] Brianna (again!) for leading the Pangeo Forge Governance drafting, along with Yuvi for valuable input on this. Anderson for re-engaging with Pangeo Forge Cloud
- C.


LXXII. Agenda

- A. Big NASA demo: **Feb 27th 1-2PM EST**
 - NASA internal tech spotlight
 - https://hackmd.io/h1FJd0K-R8WamJB_4pV50A
 - <https://github.com/orgs/pangeo-forge/projects/4> <- get beam refactor merged
- B. [Charles] Update on bringing beam refactor to production
 - Weird bugs in current production deployment. So working on integration testing first
<https://github.com/pangeo-forge/pangeo-forge-orchestrator/pull/226>
 - Biggest challenge
 - We have an entire test org for Pangeo Forge Cloud testing
 - How do we have arbitrary numbers of ephemeral github apps?
 - Let's run our own proxy service
- C. When should we merge beam-refactor to main
 - When it passes the end-to-end integration test, resolve in silo'd dev

environment

- Need to get all of the [tutorial recipes](#) refactored (Ryan will open an issue)
- Support for reference recipe

D. John Clyne

- Preparing big NASA proposal
- Building a hybrid cloud
- Wants an letter of support from Pangeo Forge
-  Alex Merose: Apache Beam on Dask- Portable, Scalable, Scientific ...

E. Governance for pangeo-forge

<https://github.com/pangeo-forge/roadmap/issues/51> [Brianna + Charles]

- We are on an exponential curve of complexity, feels very precarious
- Recipes vs the service of pangeo-forge. The open source software or the cloud service
- We need uses cases (Charles also suggested this, using the NASA related work as an example of how should this be steered)

F.

LXXIII.

2023-01-16

Attendance:

- Ryan Abernathey / LDEO / @rabernat
- Charles Stern / LDEO / @cisaacstern
- Derek O'Callaghan / UCD / @dgocallaghan
- Sean Harkins / Development Seed / @sharkinsspatial
- Yuvi / 2i2c / @yuvipanda

Agenda

LXXIV. Props

- (Ryan) Charles for doing what it takes to release a new version of PFR to prod (including lots of frustrating iterations) - debugging in prod is not fun!
- (Ryan) Derek for diving into beam-refactor and helping to surface bugs / feature gaps
- C.

LXXV. Agenda

- Updates on deployment issues
 - (Charles) Things are not totally un-broken yet, specifically deploying dataflow jobs -> reverting everything back w/ PFR 0.9.2

■ What about 0.9.4

- What sort of integration testing would we need to feel more comfortable?
- Whenever we wanted a new version of PFR in production:
 - Upgrade that version in the pangeo docker image forge container
 - But now pangeo docker images have upgraded to python 3.10, but `.to_beam()` compiler in PFR is incompatible
 - So what we should do is: use the feature built into pangeo-forge runner to pass requirements.txt -> use this to upgrade PFR to 0.9.4 at runtime
 - (a) What's the fallback logic for recipes w/o requirements.txt? -> pick an image and freeze it (deprecate pfr version in meta.yaml)
 - (b) More cumbersome requirements (e.g. conda)?
- Yuvi wants to put integration test in Pangeo Forge Runner
 - Ch: that would miss exactly how the orchestrator application is calling runner
 - Would help to move the flink test to AWS directly
- Charles - would like a more comprehensive integration test against real github
 - Monorepo - Ryan likes it, Yuvi doesn't 😊
Great for ts / js, fully integrated applications
- <https://github.com/jupyterhub/mybinder.org-deploy/pull/2476>
- Sean: path forward - **we need integration tests in runner**
...and appropriate mocking of runner in orchestrator
- Pangeo Forge Deploy repo

B. Review of Derek's very helpful issues on beam-refactor

- Derek's recipes require custom process_input functions
- This has raised some issues
 - Make it easier to add custom steps:
<https://github.com/pangeo-forge/pangeo-forge-recipes/issues/473>
 - Support both NetCDF3 and NetCDF4 in a single FilePattern:
<https://github.com/pangeo-forge/pangeo-forge-recipes/issues/472>
 - Time encoding:
<https://github.com/pangeo-forge/pangeo-forge-recipes/issues/465>

PR

<https://github.com/pangeo-forge/pangeo-forge-recipes/pull/471>

^ Ryan will review

- C. Created a Project Board to organize getting beam-refactor merged
<https://github.com/orgs/pangeo-forge/projects/4/views/1> (Yuvi)
- D. Governance discussions

2023-01-02

Attendance:

- Charles Stern / LDEO / @cisaacstern
- Ryan Abernathey / LDEO / @raternat
- Martin Durant / Anaconda / @martindurant
- Yuvi / 2i2c / Berkeley / @yuvipanda
- Brianna Pagán / NASA GES DISC / @briannapagan
- Eric Charles / Datalayer / @echarles

Agenda

- LXXVI. Props
 - A. (Ryan) Charles and Yuvi for their big sprint over the holidays
 - B. (Charles) Yuvi for going on a great pangeo forge roadtrip adventure!
 - C. (Yuvi) Charles for finding me an amazing place to stay at in Santa Monica! Plus time with his cute baby! Also to Brianna for helping me stay with her for 10 days with her amazing dog
- LXXVII. Welcome Eric Charles!
 - A. Coming from Apache open source community (Spark, Hadoop)
 - B. Jupyterlab commiter, Jupyter server contributor, contractor for Quansight, now moving to Anaconda contracting, met many folks at CZI
 - C. Startup called Datalayer (<https://datalayer.io/>) - custom UI built on React.js
- LXXVIII. Pangeo-forge-runner update
 - A. Goal: make it possible to run beam-refactor branch while also being backwards compatible with existing recipes
 - B. Core challenge: how to inject configuration into recipe at runtime (e.g. storage target) - made a hackmd with different option
 - C. Chosen solution: AST rewriting

- Used by pytest: if you write assert in pytest, you're not actually using python to execute the code
- Beam runtime parameter doesn't work in expand, only ParDo. Same architectural problem as putting environment variables deep in library code.

D. <https://github.com/pforgetest/gpcp-from-gcs-feedstock/pull/4/files>

E. <https://github.com/pangeo-forge/pangeo-forge-runner/pull/48>

F. <https://github.com/pangeo-forge/pangeo-forge-recipes/pull/460>

LXXIX. Pangeo Forge Cloud update

A. <https://github.com/pangeo-forge/cloudrun-recipe-handler>

B. <https://github.com/pangeo-forge/pangeo-forge-orchestrator/pull/204>

LXXX. Governance

A. There needs to be a common governing space where multiple stakeholders can operate on a level playing field; goal of governance is not to do day-to-day work but rather to set the framework for collaboration

B. Governance council - not a legal entity. Money doesn't go into it. Code of conduct enforcement.

C. Should we make pangeo-forge-cloud a multi-instance project

D. One specific suggestion:

- Refactor Pangeo governance to include the notion of sub-projects/sub-committees

<https://github.com/pangeo-data/governance/blob/master/governance.md>

- Apply for NumFOCUS sponsorship for Pangeo

- Make Pangeo Forge a sub-project of Pangeo, so that it can receive directed funds via NumFOCUS

E. What does NASA want?

- Want feature X implemented

- Get a contractor to do it
- Direct internal employees to do

- Is the lack of governance currently a blocker for NASA involvement? Maybe not

- Maybe governance isn't a block for NASA, however I think that is primarily due to the existing reputation of the name Pangeo + Ryan, there is a sense of trust in both which makes it easier to consider/adopt internally.

F. <https://github.com/jupyterlab/jupyterlab/blob/master/CORPORATE.md>

G. _Roadmapping_ as a key responsibility of some governing body?

- Compare to: Zarr Steering Council, which sets the project roadmap and

deploys grant resources to achieve goals.

■ Hypothetical:

- Org X (company, institution, agency, etc) wants some features developed for their internal use case
- Org X presents these as an RFC to the Pangeo Forge council, along with any resources they are willing to devote to accomplishing this (grant money and/or developer time)
- (Pangeo Forge council is operating on quarterly RFC review cycles?)
- Pangeo Forge council solicits feedback from all stakeholders. If RFC is approved, council also figures out staffing (including Org X resources and/or other contributions)

H. Next steps:

- Answer: what would a governing body do for Pangeo Forge? (maybe as a PR to `pangeo-forge/roadmap` repo?)
- Then: How would that look as a subcommittee of Pangeo org? (at this point, time to make a PR to Pangeo governance repo?)

2022-12-19

Attendance:

- Ryan Abernathey / LDEO
- Charles Stern / LDEO - apologies I am running ~10 mins late, I'm on my way! (Yay! I was only 4 mins late after all 😊.)
- Derek O'Callaghan / UCD / @dgocallaghan
- Martin Durant / Anaconda / @martindurant
- Alex Merose / Google / @alxmrs

Agenda

LXXXI. Props

A. Ryan:

All the new recipe contributors!

<https://github.com/pangeo-forge/staged-recipes/pulls>

And especially Chris Dupuis for getting involved

B. Charles: Yuvi and Brianna for visiting me!

C. Alex: Cheers to everyone working on the Beam refactor!

LXXXII. Update on orchestrator refactor

- A. <https://github.com/pangeo-forge/pangeo-forge-orchestrator/pull/197>
- B. Outcome:
Currently orchestrator needs to import and do things with pangeo-forge-recipes / pangeo-forge-runner using a subprocess. The problem with this is that the container that is running orchestrator needs every single dependency of pangeo-forge-recipes; can't easily support dynamic installation of recipe runtime dependencies.
Orchestrator PR will give us the ability to have the object which spawns recipes import processes to be configurable. Following the model of the Jupyter ecosystem, LocalSubprocessSpawner will not be used in production; instead will use DockerSiblingContainerSpawner etc.
We can't get Docker daemon on Heroku; so we need to migrate to GCP
- C. Timeline? A lot depends on Yuvi. What is the easiest way to run FastAPI on GCP. Ideally less than a month.
- D. What is the fastest way to get this deployed on GCP?
Answer from Alex: use AppEngine
Maybe you don't need docker daemon directly? Install packages at runtime?
Building containers on GCP:
<https://cloud.google.com/build/docs/build-push-docker-image>
Also, what about cloud functions
<https://cloud.google.com/functions>
- E. pangeo-forge-runner cloud function
 - This can be called by FastAPI by a "CloudFunctionSpawner" – called from Heroku over the network
 - (Charles– brainstorming) Cloud function would run within a base pangeo/forge image and (optionally) dynamically install a `pangeo-forge-runner` and `pangeo-forge-recipes`, and then call `pangeo-forge-runner` with args provided over the network.
 - <https://github.com/pangeo-forge/pangeo-forge-recipes/issues/450#issuecomment-1352453626>
- F. Inspiration for how fast containers can build and start
<https://erikbern.com/2022/12/07/what-ive-been-working-on-modal.html>
- G. Related:
<https://github.com/apache/beam/issues/22349#issuecomment-1315708603>

LXXXIII. Update on Beam refactor

- A. <https://github.com/pangeo-forge/pangeo-forge-recipes/issues/450>
- B. (Charles) Hardest thing was figuring out dynamic `target_url` in the Python SDK context. We need a concise issue description. The answer might be don't use Python SDK and instead run the recipes as a script (which is probably most

canonical Beam approach, and allows passing runtime variables as argparse PipelineOptions).

C. Three things user could give us:

- XarrayZarrRecipe
- CompositeTransform
- Python Script / Module

D. Derek has been porting existing recipes to the new beam-based approach

- Started with very simple recipe with minimal changes
- Now adding a custom transform stage (compute windspeed and direction)
- Recunking not working? It's tested:

https://github.com/pangeo-forge/pangeo-forge-recipes/blob/beam-refactor/tests/test_end_to_end.py

https://github.com/pangeo-forge/pangeo-forge-recipes/blob/beam-refactor/tests/test_rechunking.py

Please open an issue with details.

LXXXIV. Governance

A. Copyright holder and CoC enforcement (community org?) vs. fiscal sponsor (non-profit)

- Open source license makes copyright concerns moot because Apache License supersedes individual copyright claims.

B. How do we move into a multi-stakeholder state? Who makes decisions about what to do with money donated to the project going forward?

- The same body that makes these decisions should enforce CoC
- Regular participants in weekly Coordination meeting are good candidates for such a steering committee, because they/we actually know what's happening!

C. What we want in a fiscal sponsor?

LXXXV. Generalized Kerchunk Reference Recipe status (Alex)

LXXXVI. Test failures with PR (Derek)

A. Seem unrelated to PR, details here:

<https://github.com/pangeo-forge/pangeo-forge-recipes/pull/444>

B. (Charles) Yes, tests are failing for mysterious reasons

<https://github.com/pangeo-forge/pangeo-forge-recipes/issues/451>

LXXXVII. CSV to Parquet recipe status (Derek)

A. <https://github.com/pangeo-forge/pangeo-forge-recipes/issues/94>

2022-12-05

Attendance:

- Ryan Abernathey / LDEO
- Charles Stern / LDEO / @cisaacstern - I'm bringing baby to an appointment and will be about 20 mins late
- Alex Merose / Google / @alxmrs
- Chris Dupuis
- Sean Harkins

Agenda

LXXXVIII. Props

LXXXIX. Alex's thoughts re Earth Engine

A. Weathertools / Weathermover

- https://github.com/google/weather-tools/tree/main/weather_mv : Loads data from raw to BQ / EE
- Data can be represented in lots of ways in BQ: loads raster data as points
- Could be possible to contribute some parts of this to Pangeo Forge

XC. Ryan's Beam documentation:

A. <https://github.com/pangeo-forge/pangeo-forge-recipes/pull/445>

B. https://pangeo-forge--445.org.readthedocs.build/en/445/introduction_tutorial/intro_tutorial_part2.html

C. Next steps

- Bug related to fill_value (encoding)
- Charles will run on Dataflow
- How do we move certain things into custom pipeline options
 - <https://beam.apache.org/documentation/patterns/pipeline-options/>
- Try as many real recipes as possible with the Beam code

Tutorial list:

D. Streaming

- When there are no events, just keep one VM on to respond to events
- Events can be pub/sub, kafka, etc.
- Sean: consolidated metadata in streaming

XCI. (Charles) What's a lightweight pythonic way to programmatically define an official schema for meta.yaml? We want it to be pytest-able and ideally auto/self-documenting. Options include jsonschema Python package (?), traitlets,

pydantic, (these last two are too heavy for this, but just mentioning them for discussion). Others I'm missing?

- A. Related: should this be its own repo? Part of orchestrator? Part of runner?
- B. Ryan: answer - JSON schema (CS: 👍)

2022-11-21

Attendance:

- Ryan Abernathey / LDEO / @raternat
- Charles Stern / LDEO / @cisaacstern
- Derek O'Callaghan / UCD / @dgocallaghan
- Raphael Hagen / CarbonPlan / @norlandrhagen
- Martin Durant / Anaconda / @martindurant
- Yuvi / 2i2c / @yuvipanda
- Alex Meroze / Google / @alxmrs
- Tom Augspurger / Microsoft / @TomAugspurger
- Sean Harkins / Development Seed / @sharkinsspatial

Agenda

XCII. Props

- A. Ryan: props to Charles for stepping back and focusing on what's important with his family...and now starting to get back into the project
- B. Yuvi: Derek for working with and improving the runner

XCIII. Status checks

A. Charles

- Catching up reading various PR threads: please tag me if you need feedback.
- Current focus:
<https://github.com/pangeo-forge/pangeo-forge-orchestrator/issues/195>
- Trying to turn feedback into actionable feedback on orchestrator. Will likely involve traitlets. This will help unblock making logs public
- What mastodon instance should I join?
 - Hachyderm.io :D (CS: 👍)

B. Raphael

- Open PR for GRIB reference recipe (link?)
 - <https://github.com/pangeo-forge/pangeo-forge-recipes/pull/438>

C. Sean

- Kicked off AWS work. Two data science people are building out recipes. PRs for NOAA datasets forthcoming.
- Questions
 - Needs to coordinate with Charles to support writing of data to multiple targets
 - What sort of bakery infrastructure should we be targeting?
 - Questions about append only

D. Alex:

- Dask Beam Runner should work with Pangeo Forge!
 - Prototype with Pangeo Hubs?
- Loading data into GEE with PF

E. Tom

- Did an update to Daymet, picking up the recipe

F. Derek

- Has been working with Yuvi on runner
- Debugging blocked recipes
 - CCMP - issue is similar to issue on Daymet -> aiohttp objects are not serializable
 - Can we reuse cached files across recipes?
 - Charles: I believe we actually already do this (99% sure, but prove me wrong!). Even when reusing a cached file, the recipe still needs to check the cache for the presence of the file, which for a large file list can be time-consuming in itself.

G. Yuvi

- Tried to understand what Arrays are and how to write recipes. Julius has been helpful. Made daymet recipe work.
 - Local

XCIV. Agenda

A. Discussion how to get others using the beam-refactor branch

- <https://github.com/pangeo-forge/pangeo-forge-recipes/tree/beam-refactor>
- Closest thing to an example:
https://github.com/pangeo-forge/pangeo-forge-recipes/blob/beam-refactor/tests/test_end_to_end.py
- Decision to make: do we want to keep the current API, or just have people make a pipeline
 - Do you have to use a pipeline in a context manager? No

- How do we want to package recipes? Pipeline object or script
 - Most of the challenges are in the runner
 - Alex: easiest packaging is a docker file
 - But what about testing images locally
 - Yuvi: build on repo2docker for environments
 - <https://github.com/pangeo-forge/pangeo-forge-runner/issues/27> is the relevant issue I think. runner needs to be split to two - one does cloning, validation, getting repos, setting config, etc, and then the other just does the *submission*.
 - Specific TODOs to move forward with beam-refactor:
 - Try to rewrite tutorial recipes using new syntax
 - Write high-level documentation
 - Refactor existing feedstocks to use new syntax
- B. Brainstorming about future structure and governance of Pangeo Forge
- How should project governance work? Relationship to broader Pangeo?
 - What should be the legal / organizational home to the code?
 - What should be the legal / organizational home to the infrastructure? (backend and data)
 - Single vs. multiple institutions
 - How can related organizations (2i2c, radiant earth, numfocus, dev seed) get involved?
 - Sean: AWS doesn't have a good mechanism for long-term maintenance
 - What is the role of corporate partners (Google, AWS, MS, Earthmover)
 - Framing Question: what does a healthy Pangeo Forge look like 3 years from now?
 - Yuvi: governance wants everyone to be *included*. Infrastructure requires someone to be *responsible*. This is a tension?
 - Alex: Pie in the sky - could IPCC take this
<https://twitter.com/rabernat/status/1583460467929776130>
- C. Alex: Dask on Beam – Let's experiment running PGF?

2022-11-07

Attendance:

- Ryan Abernathey / LDEO / @rabernat
- Julius Busecke / LDEO / @jbusecke
- Alex Merose / Google / @alxmrs
- Raphael Hagen / CarbonPlan / @norlandrhagen
- Anderson Banihirwe / CarbonPlan / @andersy005

- Sean Harkins / Development Seed / @sharkinsspatial
- Derek O’Callaghan / UCD / @dgocallaghan
- Tom Augspurger / Microsoft / @TomAugspurger
-

Agenda

XCV. Props

- A. Ryan: Anderson for continuing to keep the lights on!
- B. Ryan: Raphael for the sick mustache
- C. Anderson: Julius for helping with debugging the mar-greenland and nasa-smap-sss recipes
- D. Alex: Props to Yuvi for helping me with a Github actions issue that will help me fix weather-tools’ build!
- E. Alex: Props to [@pabloem](#) for all the 1:1 time and examples in creating the Dask runner for Beam
- F. Julius: Yuvi for tremendous help with debugging pangeo-forge recipes locally and with dataflow
- G. Julius: Anderson for helping with many different recipes that I
- H. Yuvi: Julius for helping me understand xarray better
- I. Yuvi: Brianna (not here) for advocating hard for pangeo-forge from inside NASA
- J. Raphael: Anderson for debugging my recent recipe
- K. Derek: thanks to Anderson for helping with feedstock

XCVI. Announcements?

- A. Charles' official back-to-work date is Nov. 14. We can expect to start to see him again soon
- B. [anything else?]

XCVII. Status check on infrastructure

- A. Memory issue on orchestrator (<https://github.com/pangeo-forge/pangeo-forge-orchestrator/issues/147>) - orchestrator crashes, no easy way to report back to user, pangeo forge runner executes recipe.py, runs out of memory -> solution, recipe runs inside container with more resources and code isolation
- B. Discovered an issue where, if something goes wrong, no simple way to re-run a feedstock. Will wait until Charles is back. Problem: no easy way to re-deliver payload from Github. Solution: hold on to payload.
- C. Logs - can wait for now, but crucial in the long run. Need to agree on what needs to be published on the front end. Currently publishing just basic dataflow info, not logs.
 - Sean: really curious to hear about what level of log information would

be helpful. Flat-file log dumps + search?

- Julius: first, just show better status. Is failure infrastructure? Code?
- Alex: dependencies is also a source of errors (e.g. eccodes).

Yuvi: <https://github.com/pangeo-forge/pangeo-forge-runner/pull/39> will solve.

Beam can take resource running hints!

- Yuvi: logs are already in stackdriver, but can't make them public
Maybe just put them on storage buckets

XCVIII. Status check on recipes

A. Derek: Offshore winds issue:

- PR link:
https://github.com/pangeo-forge/eooffshore_ics_ccmp_v02_1_nrt_wind-feedstock/issues/2
- https://github.com/pangeo-forge/eooffshore_ics_ccmp_v02_1_nrt_wind-feedstock/pull/3
 - Possible workarounds for the aiohttp serialization issue:
https://github.com/pangeo-forge/eooffshore_ics_ccmp_v02_1_nrt_wind-feedstock/pull/3#discussion_r1010420618

■

B. Alex: ARCO-ERA5

- C. Yuvi: https://github.com/pangeo-forge/GPM_3IMERGDL-feedstock/ has been really positive for talking to NASA folks, as it integrates with CMR and uses earthdata login. Doesn't run from pangeo-forge-cloud yet.

- Trying daymet at
<https://github.com/pangeo-forge/staged-recipes/pull/213>
- Derek: wants to continue to push for local recipe validation via pangeo-forge-runner, some thoughts here:
<https://github.com/pangeo-forge/user-stories/issues/13#issuecomment-1303586954>
 - <https://github.com/pangeo-forge/pangeo-forge-runner/issues/27> is relevant to what you're talking about Derek. The idea would be to have a `pangeo-forge` CLI that would be provided by `pangeo-forge-runner` and that would *depend* on recipes. Otherwise 100% what you are saying :)
 - Would something similar to <https://github.com/conda-incubator/grayskull> be useful here?
 - Alex: handling software supply chain is a key function for the runner. Very hard to install certain packages
 - Everyone should check out <https://modal.com/> to see what a

good UX is like

D. Anderson:

■ New feedstocks w/ datasets

- <https://pangeo-forge.org/dashboard/feedstock/86>
- <https://pangeo-forge.org/dashboard/feedstock/78>
- <https://pangeo-forge.org/dashboard/feedstock/88>
- Q: what is the recipe.yaml spec?
<https://github.com/pangeo-forge/roadmap/blob/master/doc/adr/0002-use-meta-yaml-to-track-feedstock-metadata.md>

Tom: use JSON-schema for this.

- <https://github.com/pangeo-forge/staged-recipes/pull/179> is almost ready. Currently waiting on both Rich Signell to confirm everything is working fine and <https://github.com/pangeo-forge/pangeo-forge-recipes/pull/437> which will enable generating code snippets on the front-end for kerchunk-based datasets

XCIX. Beam on Dask: Ready for early testing of integration with PGF

A. Recent development: Windowing support for the Dask Runner.

- <https://github.com/apache/beam/pull/23913>

B. Giving a talk this Thursday, talking a lot about PGF!

C. Thoughts on planning for

<https://github.com/pangeo-forge/pangeo-forge-recipes/issues/37>

CI. Columbia restrictions on our GCP account are causing problems (we think?)

- A. CloudNAT is required because they restrict external IPs (which IMO doesn't increase our specific security posture in any way)
- B. CloudNAT adds a bunch of extra cost, but also means folks ban us more as outgoing traffic from any number of nodes looks to be coming from a smaller number of IPs (perhaps 1)
- C. No public buckets, can't expose logs either via stackdriver

2022-10-24

Attendance:

- Ryan Abernathey / LDEO / @rabernat
- Alex Merose / Google / @alxmrs
- Martin Durant / Anaconda / @martindurant
- Anderson Banihirwe / CarbonPlan / @andersy005
- Derek O'Callaghan / School of Physics at UCD / @dgocallaghan

- Tom Augspurger / Microsoft / @TomAugspurger

Agenda

CII. Props

- A. Ryan: Anderson for continuing to deliver amazing updates to the Pangeo Forge UI, e.g.
<https://twitter.com/andersy005/status/1582768027489038336>
- B. Ryan: Alex for launching Google's ARCO-ERA5 open dataset (created with Pangeo Forge):
https://twitter.com/al_merose/status/1582792311963934720
- C. Ryan: Alex for pushing forward the Dask runner for Beam and planning to share this at PyData NYC
- D. Anderson: Yuvi for helping with VM connectivity issues:
<https://github.com/pangeo-forge/pangeo-forge-orchestrator/issues/169>
 - Comment from Alex: billing is very different for dataflow as a product (pay per cpu/second)
 - Wasteful to try to parallelize I/O bound tasks (e.g.NOAA server downloads)
 - Alex has an example of rate-limiting
 - Is the system respecting HTTP headers:
 - <https://github.com/pangeo-forge/casm-feedstock/issues/4>
 - We need a "download queue"
 - All cloud providers have a Kafka-compatible clone

CIII. Review of recent dev activity

- A. Ryan is continuing to make progress on Beam refactor (albeit slowly)
<https://github.com/pangeo-forge/pangeo-forge-recipes/pull/425>
 - Groupby Keys / side inputs force pipeline to block
- B. Anderson is looking into ways to expose logs on the front-end so as to make recipe contributors self-sufficient
 - Should the log fetching functionality reside in pangeo-forge-runner?
 - <https://github.com/pangeo-forge/pangeo-forge-runner/issues/20>
 - <https://github.com/pangeo-forge/pangeo-forge-orchestrator/pull/150>
 - Ryan: why don't we re-add logging very carefully into the beam-refactor, being very careful about any potential secrets exposed. Then we can safely expose those logs.
 - Derek: is there some way for the staged-recipe submitter to run Pangeo Forge Recipes locally

This is basically user story 13:

<https://github.com/pangeo-forge/user-stories/issues/13>

- How much information should we expose on the front-end?

CIV. Staged recipe status

A. What are the priorities and blockers on

<https://github.com/pangeo-forge/staged-recipes/pulls>

- Serialization issue when using `preprocess`:
<https://github.com/pangeo-forge/staged-recipes/pull/145#issuecomment-1286331985>

■

B. Derek: Creating offshore wind database
(<https://eooffshore.github.io/datasets.html>); able to upload some of these to Zenodo. Would prefer Pangeo Forge equivalents

- ERA5
- Sentinel-1 Ocean Wind
- etc
- Submitted a few recipes
 - CCMP: <https://github.com/pangeo-forge/staged-recipes/pull/145>
 - ASCAT NRT/REP:
<https://github.com/pangeo-forge/staged-recipes/pull/183>

CV. Should we create a twitter account for Pangeo Forge?

CVI. ARCO-ERA5: Next-Steps

A. <https://github.com/TomAugsburger/era5/blob/main/src/azure/etl.py>

CVII. Dask + Beam runner status

<https://github.com/apache/beam/pull/22421/>

2022-09-26

Attendance:

- Ryan Abernathey / LDEO / @raternat
- Alex Merose / Google / @alxmrs
- Charles Stern / LDEO / @cisaacstern
- Anderson Banihirwe / CarbonPlan / @andersy005
-
-

Agenda

CVIII. Props

- A. (Ryan) Charles for pushing to get all the documentation and other fixes to pangeo-forge-orchestrator tidied up before he goes on leave
- B. (Ryan) Alex for providing constant advice and encouragement on all things Beam
- C. (Ryan) Yuvi for bringing our project into alignment with infrastructure best practices
- D. (Ryan) Anderson for the dataset preview feature!
- E. (Ryan) Brianna for being willing to dogfood Pangeo Forge on real full-sized NASA workloads
- F. (Charles) Yuvi & Brianna for the momentum!

CIX. Update on orchestrator cleanup and documentation (Charles)

- A. To become a full dev on Pangeo Forge, just need to join Pangeo Forge AWS org w/ kms permissions: inside of orchestrator repo, can decrypt everything
 - Using AWS because Columbia AWS accounts have more flexible account granting policy
- B. Goal: leave docs in useable state so that they can be handed over tomorrow
 - Working session on Tuesday?
 - Anderson will meet with Charles before he leaves

CX. How many recipes should we plan to support in a single feedstock? (Charles)

- A. Pangeo Forge Cloud (GitHub App, etc.) works well when the number of datasets build by the feedstock can be held in a human's working memory (e.g. 3-7 datasets)
- B. Alex: running into quota issues? - organization resource limits
- C. Ryan: Is CMIP6 unique? It is a tree structure. Are other feedstocks generating large trees like this? Problems arise here related to the GitHub workflow.
- D. Sean: We could provide guidance on max number of datasets per feedstock.
- E. Ryan: We should define a scope of the open data project. For CMIP in particular, a tree abstraction.
- F. Action: Ryan will open an issue for a data tree abstraction in recipes.
- G. Maybe use pub/sub or a database to manage queue

CXI. Beam / Dask / Flink updates (Alex)

- A. Set up basic dask runner
- B. Flink: Kevin (author of Flink runner) responded on github issue
 - <https://github.com/pangeo-forge/pangeo-forge-runner/issues/19#issuecomment-1258373556>

CXII. Dynamic environments for recipes

- A. Specify arbitrary requirements via requirements.txt or environment.yaml
- B. What about incompatibilities between versions of dataflow / flink and recipe

requirements

- Will flink and dataflow support infinite backwards compatibility?

C. Decision: we will focus on specifying the environment with a text file and not try to persist containers for a long time. Every time a recipe is run, the container will have to be rebuilt.

D. Try to leverage repo2docker caching tech

CXIII. Anderson: can we use Pangeo Forge as a way to catalog existing datasets

2022-09-12

Attendance:

- Charles Stern / LDEO / @cisaacstern
- Ryan Abernathey / LDEO / @raternat
- Martin Durant / Anaconda / @martindurant
- Yuvi / 2i2c / @yuvipanda
-
-

Agenda

CXIV. Props

- (Ryan) Charles for deploying the new back-end into production! Huge milestone
- (Charles) Anderson for jumping into `orchestrator` Issues + PRs! It's so helpful to a set of critical eyes looking at + improving this code.
- (Charles) Yuvi for heroic AWS infra efforts, and defining a path forward for "catalog wrappers" for NASA CMR + ESGF (i.e. CMIP)
- (Charles) Alex for coordinating an in-person meeting opportunity in LA this week.

CXV. (Charles) GitHub App update

- GitHub App is released to prod!
- Triaging
<https://github.com/pangeo-forge/pangeo-forge-orchestrator/issues/94> and issues >= No. 94 this week. Please open more Issues if you find them!
 - This will open a pathway to finally closing (almost a year after it was opened) <https://github.com/pangeo-forge/staged-recipes/pull/88> !

C. Continue labeling/triage of staged-recipes PRs

CXVI. Pangeo Forge Recipes beam refactor

A. No progress last week

B. Deep dive with Alex planned for Wednesday

C. Goal to merge by end of month

- Key Q: will it be possible to dynamically configure pangeo-forge-recipes version in PF Cloud
 - Yes, if we
 - Stop hard coding version in PF runner
 - Orchestrator needs to start using docker!
 - Charles' idea: deploy a fork of GH app that has experiment branch on pforge-test
 - Yuvi's idea: set up multiple virtual envs in the same container that have different version of PF recipes - orchestrator specifies path to PF runner
 - Orchestrator talks to runner through stdout - runner can be in process, docker container, kubernetes pod, etc.

CXVII. Move off Heroku?

A. Not yet. Just have multiple venvs in the same container

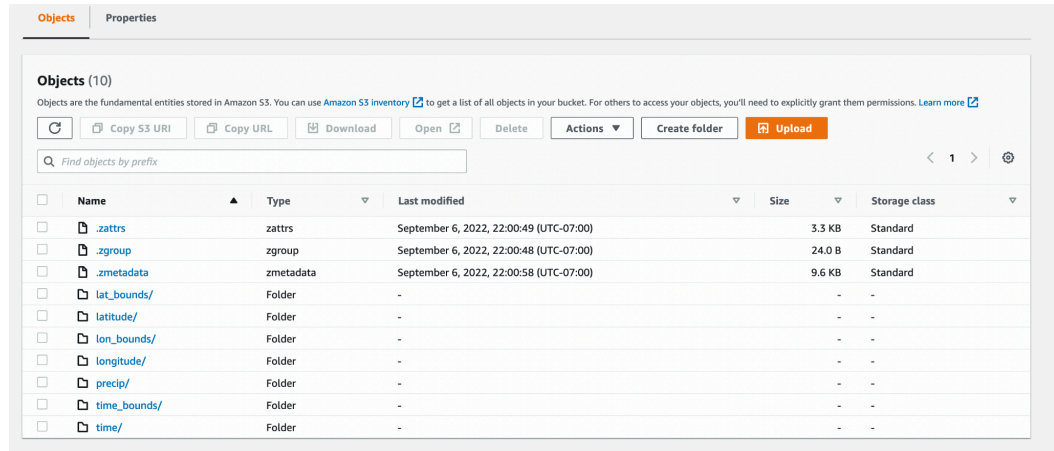
- <https://github.com/pangeo-forge/pangeo-forge-orchestrator/blob/eab98de3e2f30ce758eca2a036b1a1e01bda5cc6/Dockerfile#L40-L41>
- Track this idea in an issue <- Charles
- Sean could tackle the following week <- Sean
- What about the base image
 - Pinned to an arbitrary version
 - Need to unpin
- Longer term, we need to add requirements.txt to support arbitrary package execution in the recipe
 - Downside: pip installing tensorflow is expensive
 - Do we have an issue for this?
 - What about building a container at runtime?
People like this idea!
Someone should open an issue for this <- someone open an issue

- This issue should probably be on `pangeo-forge-runner`

CXVIII. AWS support for pangeo-runner

A. <https://github.com/yuvipanda/pangeo-forge-runner/pull/21> supports Apache Flink! Can run on any k8s cluster, includes integration tests running on github actions with kubernetes. Should be merged soon

- B. <https://github.com/yuvipanda/pangeo-forge-cloud-federation/> has terraform for setting up all AWS infra
- C. I've this infra running and outputting to S3



	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	.zattrs	zattrs	September 6, 2022, 22:00:49 (UTC-07:00)	3.3 KB	Standard
<input type="checkbox"/>	.zgroup	zgroup	September 6, 2022, 22:00:48 (UTC-07:00)	24.0 B	Standard
<input type="checkbox"/>	.zmetadata	zmetadata	September 6, 2022, 22:00:58 (UTC-07:00)	9.6 KB	Standard
<input type="checkbox"/>	lat_bounds/	Folder	-	-	-
<input type="checkbox"/>	latitude/	Folder	-	-	-
<input type="checkbox"/>	lon_bounds/	Folder	-	-	-
<input type="checkbox"/>	longitude/	Folder	-	-	-
<input type="checkbox"/>	precip/	Folder	-	-	-
<input type="checkbox"/>	time_bounds/	Folder	-	-	-
<input type="checkbox"/>	time/	Folder	-	-	-

D.

CXIX. Sean's plans for AWS open data

- A. Run a few demo datasets through existing structure (writing AWS S3 from GCP)
- B. What about secrets? All secrets are committed to orchestrator repo as SOPS-encrypted yaml
- What about redundancy between backend database bakery table and yaml file of secrets
 - In long-run yuvi thinks it should be removed from database to create single source of truth

CXX. Tom's update

- A. PC Tasks:
- <https://github.com/microsoft/planetary-computer-tasks>
- B. Motivation: data has to get to Azure (e.g. Landast), want to get it there as fast as possible
- C. All cloud providers have rich event systems—respond to activities in anything
- D. Respond to storage events: new file is written to bucket, trigger processing
- E. No message queues yet, will get there eventually
- F. Rob has a fancy thing call Azure Durable Functions, ditched that because it didn't scale
- G. Lots of cool stuff about pydantic
- Local workflow yaml definition
 - Pydantic models to do type checking before job submission

CXXI. Let's rename pangeo-forge-runner

<https://github.com/yuvipanda/pangeo-forge-runner/issues/24>

2022-08-29

Attendance:

- Charles Stern / LDEO / @cisaacstern
- Ryan Abernathey / LDEO / @raternat
- Raphael Hagen / CarbonPlan / @norlandrhagen
- Martin Durant / Anaconda / @martindurant
- Alex Merosse / Google / @alxmrs
- Anderson Banihirwe / CarbonPlan / @andersy005

Agenda

CXXII. Props

- A. (Charles) Ryan for a great review of the GitHub App last Friday, and for lighting a fire of enthusiasm to take it to the finish line!
- B. (Charles) Yuvi for amazing design consult on GitHub App, and for creating pangeo-forge-runner which has instantaneously become an key part of the picture
- C. (Ryan) Charles & Anderson for a great workshop with M2LInES & LEAP
- D. (Ryan) Charles for this heroic effort

CXXIII. (Charles) GitHub App live demo

- A. Example:
<https://github.com/pangeo-forge/staged-recipes/pull/177#issuecomment-1230018069>
- B. It's here! The app is not deployed to production yet but I will walk through a development deployment. Points I will highlight:
 - Development iteration is so fast with a local instance + proxy! Hot reload local changes, then replay the event webhook without going back to GitHub.
 - Smooth + robust deployment cycle: local > review > staging > prod. We already had this for the orchestrator, but not for the (soon to be retired) registrar. Will make adding and collaborating on new features so much easier.
 - All secrets are server-side: no configuration of GitHub Secrets to run authenticated job (i.e. Dataflow), which means apps can be installed on "untrusted" repos. Just like readthedocs. Also, GitHub Apps never send

long-lived creds over the network, all auth is using expiring tokens created from a server-side private key.

- SOPS makes server-side secrets management transparent (to maintainers).
- pangeo-forge-runner! So many points to highlight, but here's an important one: makes adding bakery targets (pretty much...) as simple as adding a new YAML config to the ./bakeries subdir on orchestrator. (Right now, the best use case would be for different storage targets—on any cloud. For different runner targets, we'll need to work on adding a new Bakery class to pangeo-forge-runner.)

C. Call to action / invitations:

- Start your own Pangeo Forge Cloud (i.e. GitHub App) in your user account, and run it with a local deployment + proxy. (Just the fact that this is possible points to the big-picture goal of franchising Pangeo Forge Cloud deployments down the line: like binder, there could be many. How provenance + database works in this “decentralized” context? TBD.)
 - What will work (for free):
 - Synchronizing PR commits on a personal GitHub repo into your local database as recipe runs
 - What will work (with a GCP account + gcloud auth login):
 - Deploy recipe runs as dataflow jobs with the /run command
 - If you deploy dataflow-status-monitoring on your GCP project: receive webhooks back to your local instance when dataflow jobs complete
 - What will not work (in a personal account):
 - Creating feedstock repos on PR merge. Not yet sure how to allow GitHub Apps to create repos for you. Some type of OAuth config is required that I haven't figured out yet. Welcome anyone's input on this, it would be really useful for development.
- Sean, let's meet to discuss collaboration on this. One idea is to factor dataflow-status-monitoring into the orchestrator, as you'd previously suggested. (Advantage here would be coupling send/receive functions for webhook data.)

D. Goals: Deploy to prod and retire registrar this week.

E. Notes

- Github App is somewhat live. Last night at ~2:30 am PST, it was installed

on Staged Recipes

- Link here:
<https://github.com/pangeo-forge/staged-recipes/pull/177#issuecomment-1230018069>
- Triggered run of GCCP(?): Status ran, maybe with a wrong url, which was pasted back to the thread.
 - The hard parts of this are done; integration with dataflow, cloud function,
 - We can send a job off to DF, get a status back, and the Github app can orchestrate the process.
- Dev process highlights:
 - Pretty much all of the development was done on a private GH account with a locally hosted FastAPI instance, maybe with a proxy to feed webhooks to the localhost.
 - GH app responded back (p-forge-local-cissac-sterm). Highlight: Happy that dev iteration is really fast. Can literally do something, trigger an event on GH, see an error on FastAPI, then hot reload and reply the webhook with GH.
 - Did this 50 times without taking an action on GH.
 - With credentials, the local instance can do whatever post we want to the local API.
 - This is awesome for debugging small features.
 - This week: try to get this deployed into Prod.
 - Thanks to Yuvi to get locally developed features easily deployed (great dev cycle).
- Highlights: all secrets are on the FastAPI server, not on GH
 - Spoke about with Alex about storing important secrets on GH, you can leak them with a GH action.
 - Now, folks on GH are “firewalled” from the secrets on the FastAPI server.
 - Props to Yuvi demo’ing Mozilla secrets encryption tool (SOPS?).
 - The only env variable needed to be configured is a private key for decrypting.
 - Can get all secrets with just a decryption key.
 - Q: Does this help with issues of recipes that are behind (protos)?
 - No, this is more about the mgmt of secrets to deploy backend.
 - There may be patterns we can copy for the recipes. It’s fundamentally solving the same problem of how to keep

things secret in a public space.

- Pangeo-Forge Runner – Yuvi’s stand-alone CLI
 - anything that involves importing a recipe is suprocessed to PGF runner
 - Sean: Thru PGF runner, it uses a JSON config to know what its storage targets are. When we’re thinking about different storage targets, we can just update the config.
- Call to actions / invitations:
 - Can start (see docs) Pangeo-Forge cloud (your own instance) in your own GH user account (create a local GH app in your account).
 - Then you can run your own local DB
 - Then, you can do most of the things that happen on the “real” or official deployment. But, it’s on your user account.
 - This opens the door to flexible deployment locations, say with Binder. Could it be decentralized?
 - Yuvi has been asking, providing feedback on how we can make this deployable anywhere.
 - You can deploy in own org, or own personal account.
 - You can synchronize PR commits in your local machine...
 - can run slash commands
 - can get notifications back
 - May want to factor back code into orchestrator.
- Q: Self-hosting PGF? (want to privately dev, then publicly release datasets)
 - We probably don’t want multiple versions of the PGF cloud service. That is more of a control plane / service.
 - We do want to support organizations to deploy their own bakery. i.e. the Data Plane of PGF.
 - Host their own runner + storage.
 - Refactoring: make it so the broader community can maintain the control pane software
 - want to ease the dev cycle.
- Think of an App as a CI service
 - A CI services detects, lints, and _ recipes in the repo
 - it can do this in the staged recipes repo, or on prod rpos
 - How we move a contributions from staged to prod as a standalone feedstock is solved.

- The staged recipe to feedstock transition is specific and singular. As long as we have a script that does that right, it doesn't need to live in the app.
 - There are many ways to create a repo. So, the GH action would have to evolve over time, and there could be a fair amt of logic here. Thus, having as much of the logic in a single place is good. The only thing that is unclear how to do is how to create it on a personal GH account
 - Could we use a test organization?
- Being a GH app is better, gives us more flexibility. And! The Dev experience is significantly better.
 - How do we have an open process that meets prod requirements? The best way is to have a good dev experience.
 - To validate: let's have someone else deploy the PGF cloud in test mode. Then we can verify that they can validate the test suite.
 - Docs for making your own deployment: https://github.com/pangeo-forge/pangeo-forge-orchestrator/blob/github-app/docs/development_guide.md
 - This is a healthy thing for our project, for bus factor.
- What are the roadblocks to making this repo public?
 - 1. _ (internet dropped) Heroku doesn't recommend it (gives alerts) Is this a warning or a requirement?
 - 2. Could someone make a competitive PGF?
 - 3. Security. Will trust Yuvi's security audit.
 - Would be nice to make an open GH app for the general community.
 - Wide surface area for an attack because executing arbitrary user code. This code will eventually be executed in a sibling container. There is a whole world of "secure enclaves" which we could draw from.

CXXIV. Beam internal refactor update (Ryan)

- A. <https://github.com/pangeo-forge/pangeo-forge-recipes/pull/405>
- B. Rechunking is implemented; PR soon
- C. How can I test things at scale?
- D. Notes
 - Ideally, XArray-Beam and PGF could be merged in the future
 - There is a clear type of internal PCollection with specific offsets. These could easily be transformed to XArray Beam PCollections. Thus, the handoff could be easy.

- Could eventually change PGF recipes name, given the Beam direction.
- People often ask about the common use case: they just want a reduction of data, but not to perform an ARCO data store.
 - This is what the XArray-Beam layer would do.
 - You could have existing recipes that depend on PGF datasets.
 - Difference btw the two projects
 - In XArray beam, you need to init the dataset outside of Beam.
 - This doesn't work for PGF. Don't know this a-priori, plus we don't want the user to do this.
- Sean: Will attempt to dogfood the Beam project on an AWS project.
 - PTransforms: not directly visible in the list. The FP defines the difference combine dimensions.
 - Right now, if you have more than one Concat dim, it asserts
 - Now, on the beam refactor branch, it will just work.
- Q: Developing all of this with the local beam test runner. Would like to run the pipelines in dataflow.
 - Ryan Abernathey + Alex Merosé will find time to debug the detail.

CXXV. Status: Beam on Dask (Alex)

A. <https://dask.discourse.group/t/distributed-runner-for-beam/784>

CXXVI. GCS Error in Finalization (XarrayZarrRecipe) (Alex) – Ideas to debug?

CXXVII. Idea for AWS deployment: beam on AWS Kinesis (uses Flink)

<https://docs.aws.amazon.com/kinesisanalytics/latest/java/how-creating-apps-beam.html>

Action Items

- Someone else try to deploy the PGF service in test mode via Charles' docs.
 - https://github.com/pangeo-forge/pangeo-forge-orchestrator/blob/github-app/docs/development_guide.md

2022-08-15

Attendance:

- Ryan is unfortunately still stuck in transit and unable to attend 😞
- Charles Stern / LDEO / @cisaacstern
- Raphael Hagen / CarbonPlan / @norlandrhagen
-

Agenda

CXXVIII. Props

- (Charles) Yuvi for riding his motorcycle to SoCal! We are going to hack on Pangeo Forge in person today in LA. :-)
- (Charles) Raphael for making some great headway on maintenance and feature PRs on pangeo-forge-recipes.

CXXIX. Update from Ryan: I am continuing to make good progress on the beam refactor. I worked on this offline from 🇺🇸 for fun. I implemented rechunking on my branch. We are almost there.

- Yay!

CXXX. (Charles) GitHub App refactor

- See last meeting's minutes (below) for details
- Making headway in <https://github.com/pangeo-forge/pangeo-forge-orchestrator/pull/80>
- This remains my top priority, and my goal is to have a minimally functional version deployed by Aug 23 (when we are presenting a tutorial to M2LINES + LEAP).
- Note:** Feedback from Sean, a Makefile for deploying a GitHub app. This is a great idea!

CXXXI. (Sean) Yuvi + Sanjay + Sean are going to have a call soon to discuss logging

CXXXII. (Sean) Questions:

- What is the work plan for Charles + Yuvi today?
 - Connect GitHub App to pangeo-forge-runner to implement recipe test slash command
- How does that affect image/container management?
- When are we going to make the switch in production?
 - (Charles) Goal is within a week! (To be ready for M2LINES/LEAP tutorial on Aug 23.)
- Should we move pangeo-forge-runner ownership to `pangeo-forge` org?

- E. When we deploy to dataflow asynchronously,, how to we tie back for logging?
- F. *** Should we package all communication with Dataflow into the runner? Or do we still need dataflow-status-monitoring as a separate repo?
 - Should `pangeo-forge-runner` package infrastructure? For dataflow-status-monitoring.

CXXXIII. (Sean) Suggestion: <https://webhook.site/#!/0e542c41-ff8e-4346-bda3-2094be72a7be>

2022-08-01

Attendance:

- Charles Stern / LDEO / @cisaacstern
- Alex Merose / Google / @alxmrs
- Raphael Hagen / CarbonPlan / @norlandrhagen
- Yuvi / 2i2c / @yuvipanda
- Joe Hamman / CarbonPlan / @jhamman
-

Agenda

CXXXIV. Props

- A. Anderson for motivating me (Charles) to bring orchestrator back into development mode
 - Ryan + Charles worked on a backed app started in feb, wasn't deployed until last week. With Anderson, it was deployed last week 5 times!
- B. Yuvi for so much helpful feedback, and spearheading `pangeo-forge-runner`
- C. Recipe contributors for their patience as the backend refactor has slowed community engagement/support activities for the time being

CXXXV. Backend refactor (Charles)

- A. Motivation
 - I'll (Charles) be on leave starting in about ~2 months
 - Q: how long will you be gone?
 - ~ 1 month leave, e.g. month of October.
 - will probably be back after a monthish
 - early next year (2023), will take an additional few months
 - In addition, Charles is on vacation this week. Won't have time to do deeper debugging.
 - We've learned a lot from the current (first draft) implementation, and also are hitting major maintenance challenges with it (even for me, who wrote it, let alone others who don't know its quirks)
 - have a good sense of how to make this more robust

- Thanks Yuvi!
- It's time to take the lessons
 - let's make v2 even better

B. Themes

- Two backends (orchestrator + registrar) → one backend application (orchestrator)
 - as of right now is basically only an interface to a Postgres DB
 - Anything that you see _ on github is a separate repo, registrar
 - this does not have any automated deployment.
 - reasons: registrar is a bit messy, and wanted to omit coupling.
 - Theme: want to consolidate backends into orchestrator.
- GitHub Workflows dispatch → GitHub App dispatch
 - this was easy to get started; however, it's brittle due to specific yaml files in all repos we want to integrate with (to the backend)
 - if we change the backend, we'll need to change all the yaml files
 - Thus, as you have often seen, Github apps make this process better.
 - Github app will just be installed in the repo, no yaml for integration
- Notifications/feedback only via PR thread comments → implement Checks API
 - comments by PGF user bot will transition to the GitHub Checks API.
 - This prevents comments thread from being cluttered.
 - e.g. readthedocs.
 - This is less intrusive than the thread.
- No automated deployment (for event handling) → automated deployment
 - Automated deployments, with a clearer PR process, will help move staging into production
- Ad-hoc runtime container story → runtime image
 - worker runtime container has been very ad-hoc up until this point
 - the runner image is part of the official docker imgs repo, which can be maintained there (good to know).
- No way to silo recipe handling/import → roadmap to silo recipe handling/import
 - there's no way to predict what environment a recipe will need.

we can't maintain all possible environments in the python session.

- which is a segue to Yuvi's PGF runner.
- The webapp itself will call a subprocess which will call a recipe. This will use the PGF CLI Util. this gives us a roadmap to silo the imports.
- v1: will not silo, since we won't have the infrastructure set up
 - we can provide an onramp to this
- v2: move to siloing the subprocess into a docker container.
 - common issue w/ staged recipes: provided recipes with reasonable imports (PG ecosystem package, e.g. MetPy) can't be imported on PGF cloud.

C. Goals w/in next two months

- Move all registrar functionality into orchestrator deployment, archive registrar (started in <https://github.com/pangeo-forge/pangeo-forge-orchestrator/pull/80>)
- Completely drop Prefect deployment in favor of Dataflow (exposing logs will require coordination with frontend)
 - everything will be going to DF
- Document maintenance + architecture of orchestrator, hold series of training calls / "office hours" for other project members (ideally beginning early September)
 - Goal: starting in 1 month / 5 weeks; holding some open office hours. Anyone whose interested in being a maintainer of the backend has the opportunity to ask questions.
 - We can refine the docs, and make explicit the missing parts of docs.
 - Docker siloing is not in scope for the next 8 weeks.
- Document backend roadmap, including mid-term goals (a few notes below)
- Q: Will you use the deployments API?
 - this is currently being used in the feedstock
 - there is room for improvements. One to get it to a good state, then leave it open for iteration
- Q: Where are we at for the DF logs making it to the front-end?
 - Spoken about: what level of richness do we need to expose to recipe authors?
 - often errors are infra related
 - Yuvi suggested having log dumps into cloud storage, then

having a holder structure.

- would need to be exposed to infrastructure.
- prototype with Loki; need to make these clusters public IP accessible
- There's some strangeness for how indexes are exposed.
- summary: interim solution.

- Yuvi: will demo an approach as part of pangeo-forge-runner demo
- Alex: suggests exposing `worker-startup` logs to debug dependency issues

D. Mid-term goals (> 2 month time horizon)

- Migrate backend hosting from Heroku → GCP to support running docker daemon for siloed recipe import (h/t Yuvi for this insight)

CXXXVI. `pangeo-forge-runner` (Yuvi)

A. <https://github.com/yuvipanda/pangeo-forge-runner>

- Performs operations on feedstock repos, including fetching from github (or with doi from zenodo, figshare, etc)
- Configurable and pluggable via traitlets (used by the Jupyter & dask ecosystem), can kick off a beam workflow on any number of runners
 - Currently implemented a dataflow runner, will add PortableRunner. Can add Fink later for AWS support
- Would love to show a demo

B. Background: Yuvi was part of the team that helped restart mybinder.org

- Stealing some of the ideas for PGF
- Useful components: clear separation between the thing that will get the data from the thing that will build the image
- this can be useful outside of the web context; makes dev and testing easier.
- looking at logs, etc. is factored out from the github app
- see demo.

CXXXVII. Frontend updates (Anderson)

A. Going through list of issues

B. Haven't had time to spend time in this area.

C. Whenever we're ready to integrate DF logs into the FE, let Anderson know

D. When it comes to new features on the FE, there will be more work that needs to be done on the BE first. Ideally, work would be offloaded to the BE.

E. Q: Right now, the deployment for the orchestrator – can that be automated at all? Would be good if infra was easy to use

- Right now, only Ryan + Charles are the project owners on teh Heroku

account. Changes to the automation (from the main and prod branches, via manual PRs) has to happen with Ryan in Charles absence. Need to remove single points of failure.

- For orchestrator PRs: in addition to the leave,

CXXXVIII. Define dataset schema:

<https://github.com/pangeo-forge/pangeo-forge-orchestrator/issues/5>

- A. Joe: Wanted to call out that Joe will work on this for the next week or two.
- B. Been talking about schemas for a while.
- C. Anderson will work on visualizing Zarr datasets on the FE.
- D. If there's interest in, at the meta.yml level or recipe.py level, having recipe devs specify some basic target dataset schema, as a check on the final output, we have tooling that might fit there.
- E. A major weakness in the CI github automation is that we print the dataset to the user and ask if it looks good.
 - would be excellent to add programmatic validation to this.
- F. Idea: XArray diff?
- G. XArray Schema will move closer to the Pandera project
 - it will report all the schema violates; it batches it
 - will give a readout when it doesn't validate
- H. next version of Xarray schema will have rountripping with YAML and JSON.

CXXXIX. (Alex) Dask runner in beam works as a minimal prototype. Will continue to test

2022-07-18

Attendance:

- Charles Stern / LDEO / @cisaacstern
- Sean Harkins / Development Seed / @sharkinsspatial
- Anderson Banihirwe / CarbonPlan / @andersy005
- Yuvi / 2i2c / @yuvipanda
- Raphael Hagen / CarbonPlan / @norlandrhagen
- Martin Durant / Anaconda / @martindurant
- Joe Hamman / CarbonPlan / @jhamman
- Julius Busecke / LDEO / @jbusecke

Agenda

CXL. Props

- A. Anderson for making the frontend shine for SciPy
- B. Ryan for an awesome SciPy talk
- C. Yuvi for jumping on board! And already making waves with

<https://github.com/pangeo-data/pangeo-docker-images/pull/355>

- CXLI. Administrivia - Repo decision for Dataflow logging infrastructure.
 - A. Loki infrastructure is too much (designed for indexing, not retrieval)
 - B. Logging artifacts in google cloud storage
 - Charles action item: move GPCP logs to OSN, as reference artifact for frontend browser
 - C. Sean will work on pushing the Loki deployment code to github, and we can use this a starting place for public logs!
- CXLII. Contributing to the orchestrator / Extending api.pangeo-forge.org
 - A. Testing infrastructure
 - B. Repository access
 - Write access is restricted
 - <https://github.com/pangeo-forge/pangeo-forge-orchestrator/issues/67>
 - C. Action Item: Charles and Anderson will meet to discuss testing.
- CXLIII. Credentials discussion:
 - A. What is the plan?
 - B. Yuvi has suggested SOPS for users passing credentials
 - C. (Sean) User-passed credentials present a reproducibility problem if contributors disappear.
 - D. (Anderson) +1 for the idea of
 - E. (Sean) Pangeo Forge accounts might get rate-limited! But we can ask for special access to commonly used services.
- CXLIV. CMIP6 recipe discussion
 - A. Feature requests:
 - Repeat on fail
 - Don't execute on existing (successfully executed) recipes
 - B. (Charles) This will be easier to collaborate on PRs for, once `registrar` code is moved into `orchestrator`
- CXLV. Summarize SciPy sprints
 - A. Recipe contributions!
 - B. Ryan worked on Beam refactor
 - C. Yuvi reviewed web app architecture, encouraging consolidation of `registrar` and `orchestrator` apps
 - D. Above-linked docker images PR was made! And lingering questions about how to provide users a matching local conda environment.
- CXLVI. Martin update
 - A. Netcdf3 work is progressing! (for kerchunk)
 - B. Wading into the bottomless pit of grib2. (for kerchunk)
 - C. Fsspec: plans to upstream some things from DVC (data version control) into

fs.spec. An fs.path will mirror os.path.

CXLVII. Sean update on AWS

- A. Cost questions of Dataflow -> S3 writes
- B. Switchable storage targets
 - Charles action item: this will be easy to collaborate on PRs for, once `registrar` code is moved into `orchestrator`

2022-06-22

Attendance:

- Charles Stern / LDEO / @cisaacstern
- Alex Merose / Google / @alxmrs
-
-

Agenda

CXLVIII. Props

- A. Anderson for a huge effort on website styles
<https://github.com/pangeo-forge/pangeo-forge.org/pull/99>
- B. Sean for figuring out and creating
<https://github.com/pangeo-forge/dataflow-status-monitoring>
- C. Charles for pushing staged recipes forward while also doing tons of back-end work

CXLIX. Staged recipe discussion: why are so many runs crashing? Is this worth trying to fix?

- A. Simplest option: just increase worker memory
 - Can go up to 8 GB without redeploying
 - Going past this involves extra work
 - Sean could help; would need a service account
- B. Should we also try the dask-inside-prefect option?
- C. One possible issue for failing recipes: reuse of cached metadata
 - (Charles) Question: is metadata re-cached if the input file exists?
 -

CL. Beam refactor

- A. Registrar work is ongoing:
 - Dataflow registration:
<https://github.com/pangeo-forge/registrar/pull/48>
 - Test recipe: <https://github.com/pangeo-forge/staged-recipes/pull/138>

- Registrar is currently run from a docker container, so I've added a separate `staging` container in which this work is happening. The `staging` container can be selected by marking a recipe PR with the `dev` label.
- How can we align with beam best practices?
 - Should we use templating?
 - <https://cloud.google.com/dataflow/docs/concepts/dataflow-templates>
- Specific idea: can we use isolated conda / venv environments rather than Docker containers to isolate recipe environment
- Alex (rightfully!) hesitant about layers of containers. Charles + Alex will review offline.
- Should we just specialize to dataflow and not worry about generalization
 - No
 - But let's write down our requirements for executors:
 - Be able to execute beam pipeline
 - Submitting jobs from registrar
 - Getting status updates via webhook
 - Gettings logs somehow

B. Ryan's progress on internal refactor

- <https://github.com/pangeo-forge/pangeo-forge-recipes/issues/376>
- Need some help
 - How to correctly use runtime options:
 - <https://github.com/pangeo-forge/pangeo-forge-recipes/pull/379/files#r904051155>
 - Needed for storage target
 - What strategy to use for writing data? Leverage xarray-beam or keep our own zarr writer.

CLI. Dask runner for Beam

A. Brainstorming w/ Google + Beam people last week

- https://docs.google.com/document/d/1Awj_eNmH-WRSte3bKcCcUIQDiZ5mMKmCO_xV-mHWAak/edit#

B. Alex and Charles hacking on this today

2022-06-06

Attendance:

- Ryan Abernathey / LDEO / @rabernat
- Charles Stern / LDEO / @cisaacstern
- Raphael Hagen / CarbonPlan / @norlandrhagen
-
- Justus Magin / Ifremer / @keewis
- Sean Harkins / Development Seeed / @sharkinsspatial
- Tom Augspurger / Microsoft / @TomAugspurger

Agenda

CLII. Props

- A. (Charles) Anderson & Joe for help understanding Javascript, Sean for a bunch of helpful chats
- B.

CLIII. Review User Stories and Milestones

- A. <https://github.com/orgs/pangeo-forge/projects/3/views/1?sortedBy%5Bdirection%5D=asc&sortedBy%5BcolumnId%5D=Milestone>
- B. Based on user stories, we haven't made much progress recently.
 - Does our system need revising?
 - Are we actually working on other things?
 - (Anderson) I've been experimenting with <https://datahubproject.io/> to see if this can be used as part of the leap-data library (which will integrate with pangeo-forge in the future)
 - Tried to write a custom datasource by reading Xarray dataset
 - Tried importing catalog CSVs into datahub; did not go well, could not actually browse the data
 - Has there been any activity on the front-end side?

CLIV. Feedback from Justus

- A. Mostly worked with HDF reference recipe
 - Was fairly easy to get the recipe to work
 - In the tutorial, there was use of `pattern_from_file_sequence` deprecated

method

- Had trouble with default intake catalog that was created
- Produced dataset had very small chunks; wanted catalog to expose larger chunks, can be done via intake catalog
- How to execute on HPC?
 - Dask jobqueue or dask-MPI
 - Say this on our website

CLV. Update on Beam Refactor

A. Ryan shares latest progress

<https://github.com/pangeo-forge/pangeo-forge-recipes/pull/375>

B. Brainstorming which PTransforms are needed

<https://github.com/pangeo-forge/pangeo-forge-recipes/issues/376>

CLVI. Discussion of Beam deployment landscape

A. Dask beam runner

<https://github.com/apache/beam/issues/18962>

B. Dataflow on GCP vs. generalize Flink for all providers?

<https://github.com/pangeo-forge/pangeo-forge-recipes/issues/256#issuecomment-1145316132>

- How much developer time should we devote to AWS & Azure Bakeries, vs. getting the Beam refactor running on GCP Dataflow? Or, more broadly, how much effort should we devote to “Bakeries” vs. running pangeo-forge-recipes on managed services?
- Sean has been researching Flink on Kubernetes deployment
- Martha at Pachama has some experience with Beam & Dask

C. Registration/Orchestration items:

- Job registration & conclusion state webhooks:
<https://github.com/pangeo-forge/registrar/issues/45>
- User-facing logs:
<https://github.com/pangeo-forge/pangeo-forge.org/pull/88>

2022-05-23

Attendance:

- Ryan Abernathey / LDEO / @rabernat
- Charles is on travel today
- Alex Merosé / Google / @alxmrs
- Rachel Wegener / U of Maryland / @rwegener2

- Tom Augspurger / Microsoft / @TomAugspurger
-

Agenda

- CLVII. Props
 - A. Charles and Sean for tackling the thorny issue of versioned Pangeo Forge docker images
- CLVIII. Review User Stories and Milestones
 - A. <https://github.com/orgs/pangeo-forge/projects/3/views/1?sortedBy%5Bdirection%5D=asc&sortedBy%5BcolumnId%5D=Milestone>
 - B. Can we identify a lead for each item?
 - C. Can we harmonize what people are actually working on with the user stories and milestones?
 - There is also technical debt - do we need a tracker for technical debt?
 - Container versioning is an example
- CLIX. Planning for big refactoring sprint tomorrow
 - A. [Opener refactor pangeo-forge/pangeo-forge-recipes#245](#)
 - B. [Should we just adopt xarray-beam as our internal data model? pangeo-forge/pangeo-forge-recipes#256](#)
 - C. Advice from Alex
 - Run stuff on direct runner
 - D. Beam Executor for Dask
- CLX. New Google 20% collaborator
 - A. Koki: lots of OS dev experience
 - B. Original proposal was issue 16: <https://github.com/pangeo-forge/pangeo-forge-recipes/issues/63>
 - C. Also Beam stuff
- CLXI. Thoughts on COGs in PF
 - A. COG workflows are more file based -> more embarrassingly parallel
 - B. Input sources logic is potentially similar
 - C. Having a record of the files you need to process / state / error handling is useful
 - D. Gdal translate
 - E. Sean: a lot of logic is consolidated into stactools; but still don't have a portable orchestrator system
 - <https://github.com/stac-utils/stactools>
 - <https://github.com/stactools-packages>
 - F. So much work is transforming scientific datasets that users bring into a

cloud-optimized format

G. Metadata is a big difference btw COG and Zarr

H. Cirrus geospatial pipeline (Sentinel2 on AWS)

2022-05-09

Attendance:

- Ryan Abernathey / LDEO / @raberna
- Charles Stern / LDEO / @cisaacstern
- Anderson Banihirwe / CarbonPlan / @andersy005
- Raphael Hagen / CarbonPlan / @norlandrhagen
- Joe Hamman / CarbonPlan / @jhamman
- Alex Merosse / Google / @alxmrs
- Wei Ji Leong / Byrd Polar Climate Research Center / @weiji14

Agenda

CLXII. Props

- A. Ryan for spearheading User Stories
- B. Joe and Anderson for the frontend tour
- C.

CLXIII. Review of user stories framework:

- A. For creating user stories: <https://github.com/pangeo-forge/user-stories/issues>
- B. For tracking all important issues across the project:
<https://github.com/orgs/pangeo-forge/projects/3/views/1>
- C. Joe: how to handle user stories that are basically just feature requests?
 - Example of a feature request: concat dims
 - Example of something more complex: reach into dask / prefect logs to debug a recipe
 - Possible resolution: triage users stories that are pure PF-recipes and bugs into PR-recipes issues
 - Add example of good user story to template
 - What is the scope of user stories?
 - Too granular makes too much overhead

- Too broad is not actionable
 - What problem do they solve? Understand the voice of the user. Practice putting ourselves in users' shoes.
- D. Question: how do we categorize issues into milestones *across repos*? This seems like the key mission feature?
- Maybe leverage github discussions
- E. Question: how are we going to prioritize?
- How to tie prioritization to metrics?
 - Currently tracking: Feedstocks / Recipes Runs / Datasets
 - Number of datasets accessed
 - Recipe developers/maintainers
 - Having a broad range of scientific fields covered?
 - What if we have specific metric goals / KPIs
 - OKRs? <https://en.wikipedia.org/wiki/OKR>
 - Question: approximately how long between milestones?
 - Alex: Milestones should maybe not be linked to time but to feature sets / user stories (link to article)
 - <https://www.rubick.com/milestones-not-projects/>
 - It's very useful to be both a user and contributor

CLXIV. Releasing `pangeo-forge-recipes` this week

- A. Start work on bakery image and related automations, or manual process for now? <https://github.com/pangeo-forge/user-stories/issues/9>
- Sean says we should do automation now
- B. Sean: question about version compatibility issues
- We need to pin a prefect version
 - Do the pangeo notebooks image releases drive PF image releases
- C. Charles is assuming we only release 1 version of PFR per pangeo notebook image
- Meta.yaml notebook image is being ignored
 - What about adding pangeo forge recipes to pangeo notebook?
 - What other dependencies do we have that are not in pangeo notebook?
 - <https://github.com/pangeo-data/pangeo-docker-images/blob/master/pangeo-notebook/conda-linux-64.lock>
 - TODO make a pangeo-forge-image on <https://github.com/pangeo-data/pangeo-docker-images>

CLXV. Do we need a user-facing "office hours" for recipe contributors?

- A. 1 or 2 hours a week would be useful because debugging async is slow
- B. Charles: moment of panic regarding dead kubernetes clusters 😬; need to get

higher **recipe success rate**

Follow up:

- Joe make a PR to update the user stories template:
https://github.com/pangeo-forge/user-stories/blob/main/.github/ISSUE_TEMPLATE/user-story.yml
- Everyone: think about what metrics are important to you. Let's orient our process around these.
- Sean: will add pangeo forge image to pangeo docker images

2022-04-25

Attendance:

- Charles Stern / LDEO / @cisaacstern
- Alex Merose / Google / @alxmrs
- Rachel Wegener / U of Maryland / @rwegener2
- Joe Hamman / CarbonPlan / @jhamman
- Anderson Banihirwe / CarbonPlan / @andersy005
- Raphael Hagen / CarbonPlan / @norlandrhagen
- Yuvi Panda / 2i2c / @yuvipanda
- Martin Durant / Anaconda / @martindurant
- Ryan Abernathey / LDEO / @raternat

Agenda

CLXVI. Props

- A. Charles and Rachel for Pangeo Forge tutorial last week at OGC event! 🥂
- B. Charles for shepherding so many new recipes through staged-recipes

CLXVII. Project coordination stuff

- A. Old contractors (Anaconda + Dev Seed) are now officially back!
- B. New contractor (CarbonPlan) is on board
- C. How should we be coordinating / communicating?
 - Do we want to get back into the two-week-sprint structure?
 - Do we want to start doing agile stuff like user stories and backlogs
- D. Review the old project boards

- <https://github.com/orgs/pangeo-forge/projects/1>
 - <https://github.com/pangeo-forge/staged-recipes/projects/1>
 - Why did we stop using these?
- E. Do we need to introduce new management / coordination / communication tools into our workflow?
- Slack?
 - <https://linear.app/> ?
 - GitHub Projects beta? <https://github.com/features/issues>
- F. Discussion
- Alex: hardest part is creating a process that meets everyone's needs (organize work around milestones)
 - Use project board for agenda
 - Requires someone's dedicated time to maintain the process
 - Milestones are more short term
 - High level roadmap: <https://github.com/pangeo-forge/roadmap> - long term vision
 - Action item: update and re-create the github projects (Ryan)
 - Product direction / product vision

CLXVIII. Merged feedstocks update (Charles)

- A. Each feedstock has a /deployments page where production recipe runs are tracked. Clicking on the `Deployed` link for a given deployment on that page links to the corresponding pangeo-forge.org page.
- B. https://github.com/pangeo-forge/WOA_1degree_monthly-feedstock
- Success!
- C. <https://github.com/pangeo-forge/cmip6-feedstock>
- Success! (More features required to realize full potential; specifically, dictionary object support in Registrar & not re-running all recipes on push event. The latter probably requires recipe instance hashing.)
- D. <https://github.com/pangeo-forge/riops-feedstock>
- Failure due to recipe issue; tracked in Issue, PR forthcoming
- E. <https://github.com/pangeo-forge/noaa-coastwatch-geopolar-sst-feedstock>
- Success! (following many failures, presumably due to Prefect graph size)

CLXIX. Prefect graph size issue (Charles)

- A. <https://github.com/pangeo-forge/pangeo-forge-recipes/issues/208>
- B. Idea: keep using prefect, but call dask from inside the prefect task
TODO: new executor (dask inside prefect)
- C. What about using [kbatch](#)?
- No UI for logs
- D. Argo workflows?

- CLXX. Beam Kerchunk prototype (Alex)
- CLXXI. Carbonplan Work
- A. Punch list for existing frontend
 - B. How is the frontend now working? Bugs? Missing info?
 - C. LEAP data library
- <https://github.com/leap-stc/leap-data-library/issues/1>
- CLXXII.

2022-04-11

Attendance:

- Charles Stern / LDEO / @cisaacstern
-
- Martin Durant / Anaconda / @martindurant
- Rachel Wegener / U of Maryland / @rwegener2
- Tom Augspurger / Microsoft / @TomAugspurger

Agenda

- CLXXIII. GitHub integration update (Charles)
- A. Approaching a testing milestone
- <https://github.com/pangeo-forge/registrar/pull/16>
- Welcome reviews on 🙌 this PR, particularly the mocking strategy, which is described in the README:
- https://github.com/pangeo-forge/registrar/pull/16/files?short_path=b335630#diff-b335630551682c19a781afebcf4d07bf978fb1f8ac04c6bf87428ed5106870f5
- Note: This repo is private pending security review. Open to thoughts of when/who can do this!
- B. More features coming now that we have tests
 - C. GitHub App migration on the near horizon
- CLXXIV. How much memory should workers have? (Charles)
- A. <https://github.com/pangeo-forge/staged-recipes/pull/127#issuecomment-1095228244>
 - B. Solutions:
 - Short term: update registrar to give more memory to dask workers
 - Long term: issue to track idea of dynamic memory heuristics
- CLXXV. Zcollection:

- A. <https://discourse.pangeo.io/t/zcollection-a-library-for-partitioning-zarr-dataset/2359>
- CLXXVI. Chunk referencing scheme - Zarr group file, kerchunk, stac to reference COGs - what do we call it?
- A. <https://fsspec.github.io/kerchunk/spec.html>
- B. “References Specification” - each key points to a chunk of data, “Reference Set”
- C. “Reference filesystem”
- D. “Chunk Index”
- E. STAC is different from Chunk Index: the STAC items are generally “standalone” while the chunks can only be interpreted along with other metadata
- F. Should we pursue the “Zarr Chunk Index” on existing file formats (NetCDF, GRIB) as an OGC standard
- CLXXVII. Pushing zarr forward?
- A. Implementation council
<https://github.com/zarr-developers/governance/pull/17>
- B. ZEP: <https://github.com/zarr-developers/governance/pull/16>
- C. <https://github.com/zarr-developers/zarr-specs/pull/134>
- D. Martin wants:
- Support for awkward array (chunks not strictly regular)
 - Asynchronous access to multiple variables
- E. Alexy S??
- CLXXVIII. Tom - working on ERA5
- A. Follow up: let’s talk to Alessandro Amici (B-Open)

2022-03-28

Attendance:

- Charles Stern / LDEO / @cisaacstern
- Alex Merose / Google / @alxmrs
- Ryan Abernathey / LDEO / @ravernat
- Raphael Hagen / CarbonPlan / @norlandrhagen
- Joe Hamman / CarbonPlan & NCAR / @jhamman
- Sean Harkins / Development Seed / @sharkinsspatial
- Anderson Banihirwe / CarbonPlan / @andersy005
- Justus Magin / Ifremer / @keewis

- Tom Augspurger / Microsoft / @TomAugspurger

Agenda

- CLXXIX. Docs update (Charles)
- A. Docs restructured to distinguish Pangeo Forge Recipe vs Pangeo Forge Cloud
https://pangeo-forge.readthedocs.io/en/latest/pangeo_forge_cloud/index.html
- CLXXX. Staged recipes activity (Charles)
- A. <https://github.com/pangeo-forge/staged-recipes/pull/127>
 - B. <https://github.com/pangeo-forge/staged-recipes/pull/129>
 - C. <https://github.com/pangeo-forge/staged-recipes/issues/125>
- CLXXXI. State of GitHub integration (Charles)
- A. Testing remains in progress, aim is to finish this week
<https://github.com/pangeo-forge/registrar/pull/16> (private repo)
 - B. Finishing testing will allow refactor to implement running full recipes from merged feedstocks
- CLXXXII. Progress with Globus (Ryan)
- A. <https://discourse.pangeo.io/t/pangeo-globus-labs-meeting-and-discussion/2308/8>
 - B. <https://github.com/pangeo-forge/pangeo-forge-recipes/issues/222#issuecomment-1080830165>
 - C. <https://github.com/MLMI2-CSSI/foundry/issues/172>
 - D. <https://funcx.org/>
- CLXXXIII. Investigating lithops
- A. <https://lithops-cloud.github.io/docs/>
 - B. <https://github.com/lithops-cloud/lithops/issues/907>
 - C. vs Beam?
 - D. (Sean) Did not see log aggregation as a batteries included feature of lithops. We'll need a solution for this to provide recipe developers infrastructure generated errors.
 - E. (Alex) Serverless may be challenged by data movement intensive applications. Task model and dataflow model are different. Which model should Pangeo Forge follow? If dataflow, Beam may be a good choice. And, Beam may be able to serve as an API for Dask.
 - F. (Ryan) We should raise possibility of Beam API for Dask with Dask team, which may be possible given High Level Graph, Blockwise, etc.
 - G. (Sean/Ryan) Lithops supports Kubernetes clusters, etc.; serverless as a billing model (per function invocation) rather than an infrastructure constraint.
 - H. (Alex) How much would an example task cost to run in a task model executor

vs. a dataflow model executor. Hypothesis: dataflow is more compute efficient

I. (Ryan) TODO: Let's implement a Lithops executor.

J. (Sean) TODO: This can be paired with a Lithops bakery.

CLXXXIV. Front-end roadmap?

A. (Ryan) Goal: LEAP data catalog, browsable on Pangeo Forge website.

B. (Joe)

- Some basic functionality & fine tuning still needed
 - Logging pagination, and pulling logs from bakeries
- Catalog page improvement (from list to actual catalog)
- Inspection/visualization on top of the catalog (might be STAC Browser, might be something new/different)

C. (Ryan) Social features: not just a static platform; a collaboration platform with user accounts.

D. (Sean) A way to try out data snippets in a notebook environment.

E. (Alex)

- <https://blog.jupyter.org/jupyterlite-jupyter-%EF%B8%8F-webassembly-%EF%B8%8F-python-f6e2e41ab3fa>

F. Awesome Xarray REPL by Anderson:

<https://github.com/xarray-contrib/xarray.dev/pull/148>

G. (Ryan) Fsspec async may be a blocker for JupyterLite

H. (Charles) "Netflix for climate data"?

I. (Joe) Two steps:

- Having data online
- What do you do once you have some data

J. (Ryan) The LEAP knowledge graph

- Pangeo Forge should be the linked-to node for data in this graph (as Hugging Face may be, for trained ML models)

CLXXXV. How can we help Justus?

A. Stuck on hpc 😞

B. French ocean models (MARS, NEMO), satellite data (L2)

C. Lots of datasets that need cleaning / organization

D. Lots of small files

E. Tried using kerchunk; but still need cleaning

F. Want to use catalogs

CLXXXVI.

2022-03-14

Attendance:

- Charles Stern / LDEO / @cisaacstern
-
- Max Grover / Argonne National Lab / @mgrover1
- Joe Hamman / CarbonPlan & NCAR / @jhamman
- James Munroe / MUN / @jmunroe
- Martin Durant / Anaconda / @martindurant
- Hema Muni/ CUNY Queens College/ @hemasphere
- Alex Merose / Google / @alxmrs
- Sean Harkins / Development Seed / @sharkinsspatial
- Rachel Wegener / U of Maryland / @rwegener2

Agenda

CLXXXVII. State of GitHub integration (Charles)

A. Examples:

- <https://github.com/pangeo-forge/staged-recipes/pull/122>
- <https://github.com/pangeo-forge/staged-recipes/pull/119>

B. WIP:

- Docs: <https://github.com/pangeo-forge/pangeo-forge-recipes/pull/328>
- Testing: <https://github.com/pangeo-forge/registrar/pull/16>

C. Dispatch: currently GitHub Workflows, I will be exploring GitHub App migration

D. Runners: GitHub Actions (running Dockerized `click` app), curious on this group's feedback re: alternatives:

- Google Cloud Run?
- Bakeries themselves? An existing Kubernetes cluster we have.
 - Sean: (Q): How do we register and spin up containers?
 - Joe: (A): Webhooks from a GitHub App which POST to a bakery endpoint.
- Another option: Prefect
 - there's nothing to say that we can't create a task that calls prefect to run something specific
 - we could have one task to coordinate within the bakery, for example.
-

CLXXXVIII. State of Apache Beam model (Alex)

A. <https://github.com/pangeo-forge/pangeo-forge-recipes/issues/256>

B. Beam represents a shift from task model -> streaming dataflow model, and may

therefore allow dropping certain steps from recipes (e.g. caching inputs)

- C. Question: How would we support both Beam and other execution models, such as Dask? Could the answer be implementing a Beam runner for Dask?

- This idea would allow running Beam graphs on Dask (*not* running Dask graphs on Beam).

- D. Apache Beam is itself a sophisticated compiler. So does it make sense to offload compilation to Apache Beam (and factor it out of Pangeo Forge Recipes)?

- E. <https://beam.apache.org/contribute/runner-guide/>

- F. Question: How does this proposal affect Bakery infrastructure?

- Possibly does not affect infrastructure setup, if Beam runner for Prefect flows.
- Question: What problem does the Bakery solve? Horizontal scaling of Prefect, Dask, etc.

CLXXXIX. (James Munroe) Where does the API stand in terms of contributions?

- A. Is the API stable enough to teach others (API)?

■

- B. How does this scale beyond two timesteps? (And without a human in the loop.)

- A: Modeled after conda forge: once everyone is satisfied that the two time-steps worth of data, merging the PR would create a stand-alone repo for that recipe (in the form of a feedstock repo).
 - As in the case of Conda forge, the submitter of the PR would be the maintainer of the feedstock repo
 - If we end up with too many PRs on staged recipes; then we can reevaluate the system. Right now, this is not the bottleneck.
 - The slash command to run the recipe would yield over to the controller of the feedstock.
- From the top of the hour: GH actions + workflows... they (don't) give de facto access to credentials. The reason that a component hasn't been implemented is that we can't provide fine-grained enough permissions within our current setup.
- From the UX, this is all stabilized. In terms of getting learners from the PR stage to the feedstock stage... should be set-up before a hackathon before April.
- Closing the loop on a) setting up the backend and getting you as a maintainer on the repo and b) getting you set up with docs so you can self-serve should be done by the end of the month (~2 weeks)
 - Thanks for tire-kicking!

CXC. (Max Grover) New role at Argonne (congrats!) and making some connections between Globus and Pangeo Forge.

- A. <https://discourse.pangeo.io/t/pangeo-globus-labs-meeting-and-discussion/2308>

2022-02-28

Attendance:

- Ryan Abernathey / LDEO / @raternat
- Charles Stern / LDEO / @cisaacstern
- Martin Durant / Anaconda / @martindurant
- Alex Meroze / Google / @alxmrs
- Joe Hamman / CarbonPlan & NCAR / @jhamman
- Hema Muni/ CUNY Queens College/ @hemasphere
- Sean Harkins / Development Seed / @sharkinsspatial
- Tim Crone / LDEO / @tjcrone
- Dax Soule/ CUNY Queens College/ @daxsoule
- Raphael Hagen / CarbonPlan / @norlandrhagen
- Rachel Wegener / U of Maryland / @rwegener2

Agenda

- CXCI. Update on website stuff (Joe)
- A. <https://pangeo-forge-org.vercel.app/>
 - B. <https://github.com/pangeo-forge/pangeo-forge.org>
 - Provide functionality feedback
 - Keep the design notes to yourself for now!
 - C. Remove “docs” page; link to <https://pangeo-forge.readthedocs.io/>
- CXCII. Rundown of preparations for OSM tutorial (Charles & Rachel)
- A. <https://docs.google.com/document/d/1YO2opxp4lA-Yavi6RtS3gTYz50EFKDBxUAt7S-p96pE/edit#heading=h.zdpogbtuivr>
 - B. Example of `pangeo-forge-bot` behavior (WIP). Begin reading this `staged-recipes` PR thread from here:
<https://github.com/pangeo-forge/staged-recipes/pull/66#issuecomment-1048578240>
 - C. <https://github.com/pangeo-forge/sandbox>
- CXCIII. Documentation Cleanup
- <https://github.com/pangeo-forge/pangeo-forge-recipes/issues/302> &
- <https://github.com/orgs/pangeo-forge/projects/2>

CXCIV. Martin is working on Kerchunk

- A. Reference fs in fsspec supports multiple different backends in one filesystem
- B. Kerchunk for netcdf classic
- C. How to keep kerchunk and pangeo-forge-recipes aligned?
 - “Combine” functionality is under works
 - Will create longer and more complicated set of parameters

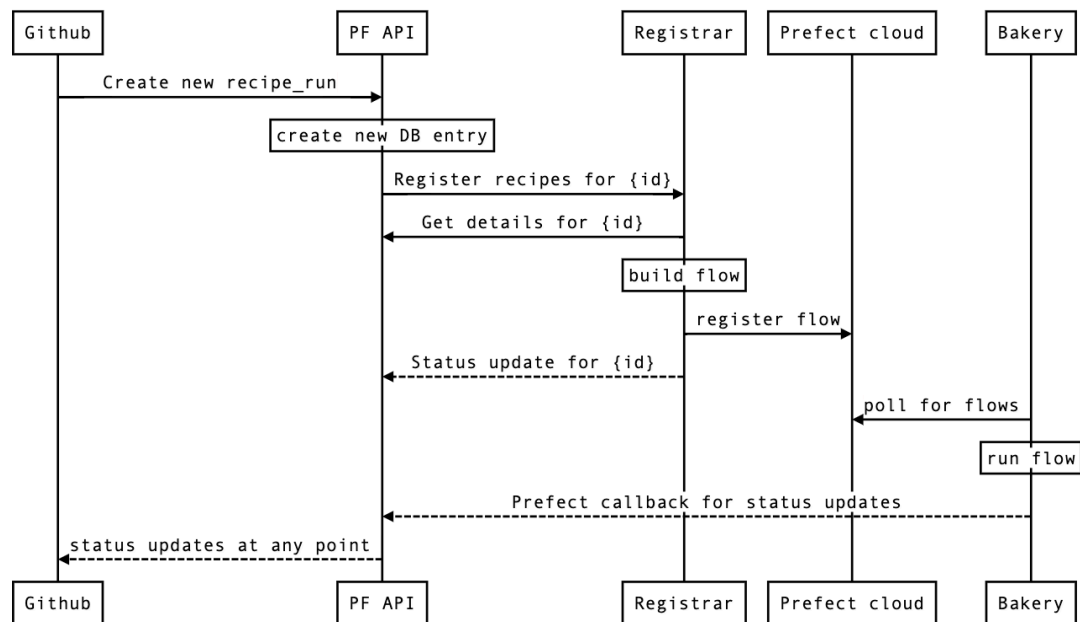
2022-02-14

Attendance:

- Ryan Abernathey / LDEO / @raternat
- Charles Stern / LDEO / @cisaacstern
- Alex Merosé / Google / @alxmrs
- Joe Hamman / CarbonPlan & NCAR / @jhamman
- Rachel Wegener / U of Maryland / @rwegener2
- Hema Muni / CUNY Queens College/ @hemasphere
- Tim Crone / LDEO / @tjcrone
- Douglas Rao / NCICS/NC State University
- Dax Soule/ CUNY Queens College/ @daxsoule
- Tom Augspurger / Microsoft / @TomAugspurger

Agenda

- I. Update on overall cloud orchestration architecture
https://hackmd.io/mn0_q_ymSnCGvDjPVxh5zw



II. Update on “Registrar” progress (Charles)

A. What questions can this group help with?

1. Talk through
2. Review PR?
3. Is register always immediately followed by run? No
 - a) Registration is useful to “check” the flow, i.e. as a linter
4. Reminder: every recipe specifies its pangeo_forge_recipes version in meta.yaml

B. Where does rate limiting happen?

1. Limit number of requests people can make

III. Update on OSM Tutorial development (Rachel)

A. <https://github.com/pangeo-forge/pangeo-forge-recipes/pull/281>

B. How can we help?

1. Review docs and try examples

C. Should we have more of a sandbox? YES!

1. Thebe! <https://thebe.readthedocs.io/en/latest/>
Could be used to get users experimenting with recipes
2. How to jump from sandbox to actual recipe?

D. What does the user have to do to transition to cloud

1. Meta.yaml & customization of targets

E. “Dry runs” / “damp run”

IV. Front End Update (Joe and Anderson)

A. <https://www.figma.com/proto/RtjB4e6hTVnOSw618xaHWU/Pangeo?node-id=1>

[9%3A80&scaling=min-zoom&page-id=14%3A452](#)

- B. Alex: test for accessibility
- C. Need to track recipe_run.id to prefect flow run ID
- D. Can orchestrator broker access to logs? Prefect?
- E. TODO: Add “status” table to data model, so we can observably track a sequence of statuses along with associated links (i.e. “running: logs_link”, “completed: data_link”)
- V. Beam Discussion: Task Orchestration vs Dataflow
- VI. [YOUR ITEM HERE]

2022-01-31

Pretty important meeting. Need to walk through orchestration flow now that the API is up.

Attendance:

- Ryan Abernathey / LDEO / @raternat
- Aimee Barciauskas / Development Seed
- Sean Harkins / Development Seed
- Joe Hamman / CarbonPlan / @jhamman
- Rachel Wegener / U of Maryland / @rwegener2
- Hema Muni/ CUNY Queens College / @hemasphere
- Charles Stern / LDEO / @cisaacstern
- Tim Crone / LDEO / @tjcrone
- Alex Merosse / Google / @alxmrs
- Dax Soule / CUNY Queens College / @

Agenda

- I. OSM tutorial is March 4, 11:30ET
- II. Present new API / CLI (Ryan & Charles)
 - A. <http://api.pangeo-forge.org/docs>
 - B. <https://github.com/pangeo-forge/pangeo-forge-orchestrator/>
- III. Group whiteboarding session to figure out how to plug things back together
 - A. Relevant pieces
 - 1. Staged-recipes [Recipe test workflow](#) calls...
 - 2. [Recipe-prefect-action](#) calls...
 - 3. [Pangeo-forge-prefect](#)
 - B. Should we deprecate [bakeries.yaml](#)?

<https://github.com/pangeo-forge/roadmap/issues/47>

1. It seems like 95% of the content is internal “bakery config”. If so, that should live with the bakery!
 - a) Prefect flow needs all of this information in order to package flows
 - b) Maybe use a single repo (+ repository dispatch) for registering flows - **would this be feasible to do before OSM?**
 2. The backend database should store only enough information about bakeries to enable registration of flows and figure out where the final datasets live—stuff the other parts of the system needs to know about
- C. Can we enumerate all the “states” that a recipe_run can be in?
1. Queued: still being processed by github actions
 2. Registered: action successfully registered the flow
 3. Running: prefect cloud saw the registration and started the flow
 4. Finished
 5. _we could fail at any of those points for different reasons
 6. Suggestion: leverage “model checking” and “formal verification”
links TBA
 - a) Example model-checking walkthrough:
<https://www.hillelwayne.com/modeling-deployments/>
 - b) <https://learntla.com/introduction/>
- D. At what points should the automation call the API?
1. Before flow registered?
 2. After flow registered?
 3. Should prefect cloud call the api? Via [automations](#)?
 4. When dataset is created
- E. Is the API only for logging? Or should it actually “do” things? If so, what should it do? For example
1. Update github webhooks / check runs?
 2. Actually register flows? (Probably not; currently needs to happen from gh actions runtime)
- F. Should pangeo-forge-perfect and the “orchestrator” CLI be merged? If so, what is the scope of the combined library? Who uses it?
- G. Should we become a Github App
- <https://github.com/pangeo-forge/roadmap/issues/48>

Q from Alex: how will people debug?

- Direct access to logs as much as possible

Should we remove prefect dependency?

- Prefect is basically just for launching jobs
-

2022-01-03

This is likely a very short meeting

Attendance:

- Ryan Abernathey / LDEO / rpa
- Aimee Barciauskas / Development Seed
- Joe Hamman / CarbonPlan / @jhamman
- Raphael Hagen / CarbonPlan / @norlandrhagen
- Tom Augspurger / Microsoft / @TomAugspurger
- Sean Harkins / Development Seed / @sharkinsspatial
- Alex Merose / Google / @alxmrs
-

Agenda

- IV. Team check in-how is everyone doing in the new year?
- V. Big Picture
- VI. Prefect account approved!
- VII. Update on contracts:
 - A. Both Anaconda and Dev Seed contract renewal process has begun
What actions are needed to move forward?
Ryan email Kim + LDEO to make sure things are moving forward
 - B. MS Pangeo Forge / Azure contract has closed
- VIII. Any other technical items?
 - A. CLI: <https://github.com/pangeo-forge/pangeo-forge-orchestrator/pull/6>
 - 1. Some overlap / duplication between what prefect provides vs our database
 - B. Apache Beam as internal data model?
<https://github.com/pangeo-forge/pangeo-forge-recipes/issues/256>
 - 1. Xpersist with prefect:
<https://xpersist.readthedocs.io/en/latest/how-to/use-xpersist-with-prefect.html>
 - C. Streaming / reactive updates
 - 1. <https://github.com/pangeo-forge/pangeo-forge-recipes/issues/37>

- a) For COGs, working on reactive model: data shows up in blob storage, needs to get immediately updated
 - b) How do we inform users about updates
 - D. MS Pangeo Forge for Azure issues summary sharing
 - 1. Summary document?
 - E. weather-dl: <https://github.com/google/weather-tools>
- IX. ESIP add'l speakers for
<https://2022esipjanuarymeeting.sched.com/event/qkoV/unlocking-arco-analysis-read-y-cloud-optimized-data-transformation-in-practice>

2021-12-20

Attendance:

- Charles Stern / LDEO / @cisaacstern
- Kevin Paul / NCAR / @kmpaul
- Alex Merose / Google / @alxmrs
- Anderson Banihirwe / NCAR / @andersy005

Agenda

- I. (Kevin) Update from NCAR
 - Has encouraged Anderson (at NCAR) to kick the tires on Pangeo Forge
 - Hoping to identify a HPC use case to build greater consensus at NCAR around using and supporting Pangeo Forge
 - (Sean) Parallels to Development Seed's work with Goddard
 - (Kevin) Many individual NCAR scientists make one-off runs of CESM, and there may be a use case in getting some of those runs out to the cloud.
 - Currently there is no HPC CI service, but were something like that to exist, there could be a simple channel for mirroring HPC data to cloud
 - (Anderson): Coupled nature of Pangeo Forge zarr conversion and writing to destination could be inconvenient/problematic. Might be preferable to convert to local zarr store, then transfer.
 - (Sean) This use case could be handled with existing Pangeo Forge, working with fsspec's LocalFilesystem
 - (Charles) Next steps: suggests Anderson open a GitHub Issue to discuss implementation details

- (Anderson) Will follow up with Max Grover regarding his previous work on this. It seems Max paused on Pango Forge not because it didn't work, but just because he was up against a tight deadline.
- II. (Tom) NASA CMIP6 downscaling project update
- III. STAC
 - (Anderson) What are best practices around STAC?
 - (Kevin) CEDA's work is a useful reference
 - <https://github.com/TomAugspurger/xstac>
 - (Anderson) Will the end user understand STAC?
 - (Charles) What to do about non-calendar time dimensions?
 - (Tom) If there are non-Gregorian calendars, they can be converted during catalog object generation?
 - Maybe we can't use STAC for certain model output due to inflexible reqs in the spec.
 - pystac will refuse data that doesn't have a datetime (e.g. parsable to a python datetime object)
- IV. Sanity Check: Source of Bug
 - ``RuntimeError: There is no current event loop in thread 'Thread-16'.` from asyncio / tornado ...`
 - Happy to take this offline (Alex)
- V. Review of miraculous fix to h5py hanging issue
<https://github.com/pangeo-forge/pangeo-forge-recipes/issues/177>

2021-12-06

Attendance:

- Charles Stern / LDEO / @cisaacstern
- Ryan Abernathey / LDEO / @raternat
-
- Martin Durant / Anaconda / @martindurant
- Tom Augspurger / Microsoft / @TomAugspurger
- Alex Merose / Google / @alxmrs
-

Agenda

- VI. Opener Refactor
 - <https://github.com/pangeo-forge/pangeo-forge-recipes/pull/245>

- <https://hackmd.io/@ravernat/rk7IWCjYY>
- Martin
 - write down what requirements you have for the “Opener”?
 - Caching - what is needed?
 - Remote opening
 - “Slugifying”
 - Sean’s comment: making input caching a recipe pre-stage
 - What if we had a CacheRecipe?
- VII. Two questions raised by [CMIP6 derived datasets](#) (Charles + Julius)
 - Where might `builders` live?
 - Position on derived datasets in Pangeo Forge more generally? These are not archival transformations, but still of great value as a shared resource. Storage space may be small, but computational (& cognitive) burden may not.
 - Notes
 - “Opener” would not be able to dynamically figure out which inputs are there
 - Question: is there a role for Pangeo Forge as a generic workflow documentation engine?
- VIII. Ongoing development notes (Charles)
 - Recipe run [SQL database](#)
 - Plans for `Pipeline` to inherit from `OrderedDict` to allow manual execution via named stages ([brainstorming thread](#))
 - We now test against [dev versions of upstream dependencies](#)
- IX. Pangeo Forge [AGU Presentation December 17](#) (Charles)
- X.

2021-11-22

Attendance:

- Charles Stern / LDEO / @cisaacstern
- Alex Merose / Google / @alxmrs
- Ryan Abernathey / LDEO / @ravernat
- Rachel Wegener / U of Maryland / @rwegener2
- Martin Durant / Anaconda / @martindurant
- Raphael Hagen / CarbonPlan / @norlandrhagen

Agenda

- XI. Orchestrator update (Charles)
 - Pivoting to first make `recipe_run` SQL database
 - <https://github.com/pangeo-forge/pangeo-forge-orchestrator/pull/6>
- XII. To_generator method
 - <https://github.com/pangeo-forge/pangeo-forge-recipes/pull/238>
- XIII. Beam executor!
 - Beam runs on many different runtimes (Spark, Dataflow, etc.)
 - Dataflow performs operator fusion; fuses multiple steps into a single task
 - Our tests use the direct runner; different runners could have different optimizations
 - <https://github.com/pangeo-forge/pangeo-forge-recipes/pull/240>
 - small fix: <https://github.com/pangeo-forge/pangeo-forge-recipes/pull/243> ←
- XIV. Paper update (Charles)
 - Revisions submitted! Alex contributed Beam section and joining as co-author.
- XV. ERA5 meeting with ECMWF
 - <https://docs.google.com/document/d/1PI1g1w79ZbeFmdZGVgqjH2Vz8lS0l6dCqvuFsb8fjBc/edit>
 - We will do a survey to gather requirements
 - MARS vs. CDS API - how to access data?
 - <https://github.com/pangeo-forge/pangeo-forge-recipes/issues/242>
 - Alex is going to release this ERA5 downloading tool
- XVI. Ryan met with Jason Hickey
 - <https://cloud.google.com/blog/topics/sustainability/thoughts-on-cloud-and-climate-change-from-cop26#:~:text=Thursday%2C%20November%2011%2C%202021>
- XVII. Scan of “staged-recipes”

2021-11-08

Attendance:

- Charles Stern / LDEO / @cisaacstern
- Ryan Abernathey / LDEO / @raternat
- Raphael Hagen / CarbonPlan / @norlandrhagen
- Alex Merosé / Google / @alxmrs
- Tom Augspurger / Microsoft / @TomAugspurger

-

Agenda

- XVIII. Orchestrator update (Charles)
 - <https://github.com/pangeo-forge/pangeo-forge-orchestrator/pull/1>
- XIX. Paper revisions due tomorrow (Charles)
 - <https://github.com/pangeo-forge/roadmap/issues/38>
- XX. Filecoin update (Charles)
 - <https://github.com/pangeo-forge/roadmap/issues/40>
 - Should we be trying R2
 - Compile Pangeo Forge to [WASM](#)?
 - <https://pyodide.org/en/stable/>
- XXI. Beam executor PR!
 - <https://github.com/pangeo-forge/pangeo-forge-recipes/pull/225>
 - Mention Beam executor in the paper? YES
 - Integrate with Xarray-Beam?
- XXII. Era5 2 Zarr
 - Compare with: NCEP reanalysis, MERRA (NASA)
 - <https://github.com/pangeo-forge/staged-recipes/issues/92>
 - Alex has a project called weathertools (CLI), can get data faster
- XXIII. Debugging memory usage
 -

2021-10-25

Attendance:

- Ryan Abernathey / LDEO / @rabernat
- Alex Merose / Google / @alxmrs
- Charles Stern / LDEO / @cisaacstern
- Raphael Hagen / carbonplan / @norlandrhagen
- Martin Durant / Anaconda / @martindurant
- Rachel Wegener / U of Maryland / @rwegener2
- Sean Harkins / Development Seed / @sharkinsspatial

-

Agenda

- XXIV. Pangeo Forge Recipes update (Ryan)

- Big internal refactor using “recipe context” approach:
<https://github.com/pangeo-forge/pangeo-forge-recipes/pull/219>
 - Follow up items
 - Expose recipe functions as public API?
<https://github.com/pangeo-forge/pangeo-forge-recipes/issues/223>
Are we okay with deprecating `recipe.do_something()` syntax in favor of `do_something(config=recipe)`?
 - Is now the right time to start adding more unit tests?
<https://github.com/pangeo-forge/pangeo-forge-recipes/issues/173#issuecomment-951109080>
 - Use more granular stages in recipes
<https://github.com/pangeo-forge/pangeo-forge-recipes/issues/224>
- Use of fsspec references for opening inputs:
<https://github.com/pangeo-forge/pangeo-forge-recipes/pull/218>
- Release 0.6.1 soon

XXV. CLI Update and Demo (Charles)

- Big picture proposal: `pangeo-forge-orchestrator`
 - A single, pip-installable Python package for orchestrating all of Pangeo Forge cloud automation. Proposal to subsume the various other repositories (imperfectly) documented in the CI call stack flowchart:
<https://raw.githubusercontent.com/pangeo-forge/flow-charts/main/renderers/ci-flow-with-callstack.png>
 - Implementation aspects
 - Data model - `pydantic`: Parse metadata artifacts from YAML, JSON, databases, etc. into Python dictionaries, with validation
 - Interfaces - CLI (Typer) & API (FastAPI)
- Current status
 - Data model draft & questions
 - “Components” (open to naming feedback) - Python object representations of Recipe, Feedstock, Catalog, and Bakery metadata
 - StorageOptions - how to represent
 - BuildLogs - we’ll need to define a spec (JSON-based?)
 - typer CLI
 - Papermill for automated Jupyter Notebook execution
- Next steps (after Minimal CLI PR)
 - Refactor automation (e.g. `pangeo-forge-prefect`,

feedstock-creation-action, etc.) into this package

- XXVI. Issues for launch
 - Still waiting on prefect enterprise account approval
 - Prefect bug related to large graphs
 - New prefect API, need to evaluate
- XXVII. Update on Google 20% projects (Alex)
 - Every 6 months, Google has a climate fair (“Anthropocene”) to work on products to address climate change (mostly internal / confidential)
 - Trying to recruit Googlers to spend their “free day” on Pangeo Forge
- XXVIII. Update on Frontiers in Climate paper revisions (Charles)
<https://github.com/pangeo-forge/roadmap/issues/38>
- XXIX. Great slide from today’s ARD21 virtual conference (Charles)
<https://twitter.com/iazuleta/status/1452679193406238723>

2021-09-27

Attendance:

- Charles Stern / LDEO / @cisaacstern
- Raphael Hagen / carbonplan / @norlandrhagen
- Sean Harkins / Development Seed / @sharkinsspatial
- Ryan Abernathey / LDEO / @ravernat
- Martin Durant / Anaconda / martindurant
- Alex Merose / Google / @alxmrs
- Tom Augspurger / Microsoft / @TomAugspurger

Agenda

- XXX. Paper submitted to Frontiers in Climate
 - Flowcharts used as paper figures: <https://github.com/pangeo-forge/flow-charts>
 - Welcome any PRs, particularly on the detailed CI workflow w/ call stack
 - **Action point:** Post a preprint. (Martin: suggests a preprint server where figures and text are hosted separately, so figures can be reused.)
- XXXI. OOS Abstract Due Wednesday
 - 2 hr hack session - Rachel and Charles working on this
- XXXII. Public launch prep: <https://github.com/pangeo-forge/roadmap/issues/33>
 - <https://github.com/pangeo-forge/pangeo-forge-recipes/issues/177>
 - Ryan working on reference maker alternate pathway
 - Charles working on h5py reproducer

- <https://github.com/pangeo-forge/pangeo-forge-gcs-bakery/issues/19>
 - Waiting on Columbia purchasing approval of Prefect contract
- <https://github.com/pangeo-forge/pangeo-forge-catalog/issues/1>
 - (Charles) My inclination is to wait until GCP is live
- Intro tutorial in progress:
<https://github.com/pangeo-forge/pangeo-forge-recipes/pull/206>
 Maybe review from Sean?
- <https://github.com/pangeo-forge/pangeo-forge-recipes/issues/208>
 - Prefect currently a non-starter for large time dimensions
 - How large is very large? 20 years @ 30min interval. Memory usage linear to inputs?
 - Some of the issue is in Dask scheduler itself ... recipe serialization size issue persists (recipe serialization should not crash 20 GB scheduler)
 - Workaround: batch input caching, add keyword like **inputs_per_cache_task**
 - Prefect cloud concurrency tags:
<https://docs.prefect.io/orchestration/flow-runs/concurrency-limits.html#task-run-limits>
 - Have shared context between tasks made explicit in an object
 - Beam in Rechunker:
 - <https://github.com/pangeo-data/rechunker/pull/99>
 - Lo-fi example of stateful approach:
<https://github.com/pangeo-data/rechunker/pull/99/commits/129f4cd2c78056bc738f31d9752f81be4666ea47#diff-058f8c4999d258ec892741543195d05ef7441104cfc7c51f50af60e7aff0ae7bR77>
 - Explanation:
<https://github.com/pangeo-forge/pangeo-forge-recipes/issues/169#issuecomment-917298775>

XXXIII. Bug discussion: credentials are not making it into prefect flows

- <https://github.com/pangeo-forge/pangeo-forge-recipes/issues/207>
- Probably caused by
<https://github.com/pangeo-forge/pangeo-forge-recipes/pull/167>

XXXIV. Secrets proposal ADR

- <https://github.com/pangeo-forge/roadmap/pull/36>
- <https://cloud.google.com/secret-manager/docs/creating-and-accessing-secrets>

XXXV. Discussion of pangeo-forge-recipes versioning strategy:

- /run-recipe-test is hard coded to certain package versions
<https://github.com/pangeo-forge/staged-recipes/issues/78#issuecomment-921333480>

- Automatically build images
 - <https://github.com/pangeo-forge/pangeo-forge-bakery-images/issues/7>
 - Ideally we would have auto-built images
 - Should be able to dynamically specify whatever pangeo-forge-recipes version we want; the main issue is the prefect agent version
 - Should not be hard to get going
 - <https://github.com/pangeo-bot/dispatcher/blob/main/.github/workflows/watch-pangeo-dask-feedstock.yml>
 - Should we just pip install pangeo-forge-recipes?
- XXXVI. Dataset versioning ADR change:
 - <https://github.com/pangeo-forge/roadmap/pull/34>
- XXXVII. Bizzaro pangeo-world update: `FilePattern`s (mostly) integrated with XArray-Beam
 - <https://github.com/google/xarray-beam/pull/31>

2021-09-13

Attendance:

- Charles Stern / LDEO / @cisaacstern
- Ryan Abernathe / LDEO / @rabernat
- Martin Durant / Anaconda / @martindurant
- Sean Harkins / Development Seed / @sharkinsspatial
- Chris White / Prefect / @cicdw
- Raphael Hagen / Carbonplan / @norlandrhagen
- Alex Merose / Google / @alxmrs
- Tom Augspurger / Microsoft / @TomAugspurger

Agenda

- XXXVIII. Organization Prefect account for Columbia GCP Bakery
 - <https://github.com/pangeo-forge/pangeo-forge-gcs-bakery/issues/19>
 - Prefect has non-profit / academic programs
 - We will get a 100% free enterprise account
 - Next steps:
 - Have to sign something (Ryan)
 - Pretty big product changes coming
 - Prefect server is getting simplified

- More dynamic workflow logic
- Looking for feedback on product directions
- Q/A
 - Map tasks count as tasks
 - Outstanding issue: low-level memory growth issues with Dask executor
 - Opportunity for group call with Prefect + Dask team to diagnose
 - Log verbosity issue
 - Fully open prefect cloud in read-only
 - Easiest way right now: public username / password (with read only permission) for Prefect dashboard
 - Option 2: Write logs to third party logging service, and then publish open link to that service
 - Pangeo Forge value to Prefect:
 - Share CI workfl

XXXIX. Public launch minimal requirements

<https://github.com/pangeo-forge/roadmap/issues/33#issuecomment-916063521>

- Active work this week on GitHub Actions for interactive PR feedback and STAC

<https://github.com/pangeo-forge/staged-recipes/pull/83>

 - Deploy bakery
 - Get contribution process nailed down
 - Some catalog solution
- Conda forge staged-recipes commit history:

<https://github.com/conda-forge/staged-recipes/commits/main>
- How to be tracking pangeo-forge versions
 - Is bakery image required?

<https://github.com/pangeo-forge/pangeo-forge-gcs-bakery/issues/19#issuecomment-916365455>

No
 - Automatically generate docker images

<https://github.com/pangeo-forge/pangeo-forge-bakery-images/issues/7>

 - Can pangeo bot help with this?

<https://github.com/pangeo-bot/dispatcher/blob/main/.github/workflows/watch-pangeo-notebook-feedstock.yml>
 - <https://github.com/pangeo-forge/bakery-database/blob/main/bakeries.yaml>
 - To update meta.yaml / bakeries.yaml, only have to touch pangeo-forge-prefect

XL. Deadline (Sept 24) approaching for Frontiers in Climate paper

<https://github.com/pangeo-forge/roadmap/issues/24>

- Charles will circulate draft this week, all invited to be co-authors
- Diagrams

- Tools
 - <https://www.lucidchart.com/>
 - Google slides
 - <https://hackmd.io/s/features#Flow-Charts>
- Diagrams
 - Architecture - how things plug together - Charles will draft
 - Contribution process
 - Algorithm
 - (1) Cache inputs
 - (2) Prepare target
 - (3) Store chunks
 - (4) Finalize target
 - Benchmarks .. might not have them by the 24th
- Other figures
 - What is analysis-ready data? Screenshots of Xarray / Zarr
 - Visualization of data from Pangeo Forge (towards the end)
- ARCHITECTURE.md?
 - <https://matklad.github.io/2021/02/06/ARCHITECTURE.md.html>
 - example: <https://github.com/ellisk42/ec/blob/master/docs/software-architecture.md>

XLI.
 XLII.
 XLIII.

2021-08-30

Attendance:

- Ryan Abernathey / LDEO / @rabernat
- Charles Stern / LDEO / @cisaacstern
- Martin Durant / Anaconda / @martindurant
- Raphael Hagen / CarbonPlan / @norlandrhagen
- Joe Hamman / CarbonPlan / @jhamman
- Sean Harkins / DevelopmentSeed / @sharkinsspatial
- Alex Merose / Google / @alxmrs
- Rachel Wegener / U of Maryland / @rwegener2

Agenda

XLIV. NSF Annual Report due in 2 days:

<https://docs.google.com/document/d/1SrnpEgBMEgnVoE6NkaZSL-RSGC8R5Jg0h46vKWvdoN6k/edit#>

Feedback welcome

- CITATION.cff question: For `roadmap` `pangeo-forge-recipes` & `pangeo-forge-prefect` repos: Authors from **Insights > Contributors** ?

Question: we have \$100K per year in cloud computing costs, still totally unspent. Should we rebudget to allow more subcontract work? If so, what contracts?

- Anaconda / fsspec?
- Front-end work?
 - DevSeed has a lot of FE resources but they have a very involved process
 - Who is the target
- 2i2c for JupyterHub / Binder integration?
- Other ideas?
 - Documentation 👍
<https://github.com/pangeo-forge/pangeo-forge-recipes/issues?q=is%3Aissue+is%3Aopen+label%3Adocumentation>
 - Diagrams / design
- Need to coordinate with Tom / MS

XLV. Update on bakeries / github actions

- GCS Bakery is ready to deploy:
<https://github.com/pangeo-forge/pangeo-forge-gcs-bakery/issues/19>
- Log aggregation + search PR
- GitHub workflows are all plugged!
- Q: what information do bakeries expose publicly / privately?
 - Tried to use prefect log tools to ship all dask worker logs to prefect dashboard; but our logs are too verbose, interleaved by workers!
 - Alex has made log searching / aggregation available to admin
- Q: how do we make a bakery talk to OSN?
 - Should be possible. Need to check.
 - https://pangeo-forge.readthedocs.io/en/latest/cloud_automation_user_guide/bakeries.html

XLVI. REST API Proposal

<https://github.com/pangeo-forge/roadmap/pull/31>

- Pangeo Forge Prefect Account - GraphQL API agents to capture and POST bakery run data

- Let's push on Prefect to get a public view of Flow run. Users will want to see the progress of their recipe runs.
 - A *rich* visual user experience is important, with a social experience (starring, commenting). Anaconda Nucleus is a related product. Conda Forge itself doesn't have web UI, but Anaconda does provide a searchable package catalog.
 - GitHub comparison: offers a GUI for the beginner to get started with technologies.
 - User experience and contributions of thumbnails/plots, etc.
- XLVII. Co-authorship invitation: Frontiers In Climate paper
<https://github.com/pangeo-forge/roadmap/issues/24>
- XLVIII. PR Discussion
- Documentation Page Outline:
<https://github.com/pangeo-forge/pangeo-forge-vue-website/pull/8>
 - HDFReferenceRecipe
<https://github.com/pangeo-forge/pangeo-forge-recipes/pull/174>
 Discussion question: *do we want to have optional dependencies?*
 Would be useful to be able to open and examine recipes with minimal dependencies.
 pip install pangeo-forge-recipes[full]
 Martin: yes! It would be useful
 Sean: worker images assume ancillary packages included in pangeo-docker-images
 - <https://github.com/dask/dask/issues/7547#issuecomment-906939846>
 - Query string secrets:
<https://github.com/pangeo-forge/pangeo-forge-recipes/pull/167>
 - Go back to "pipelines" model
<https://github.com/pangeo-forge/pangeo-forge-recipes/pull/192>
 Related rechunker issue:
<https://github.com/pangeo-data/rechunker/issues/92>
 - Is it possible to write a test for serialization problems?
- XLIX. [add]
 L. [more]
 LI. [items]
 LII.

2021-08-16

Attendance:

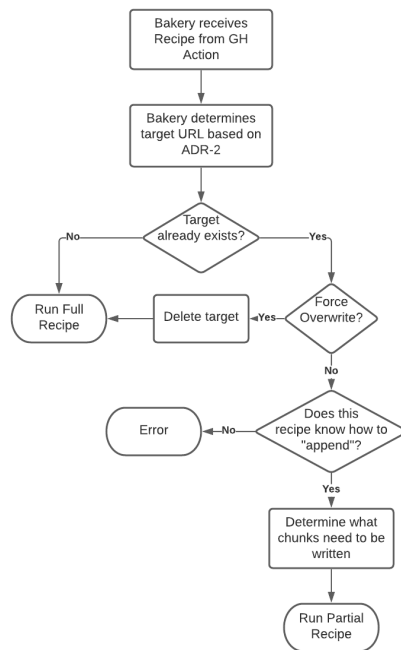
- Ryan Abernathey / LDEO / @rabernat

- Charles Stern / LDEO / @cisaacstern
- Sean Harkins / DevSeed / @sharkinsspatial
- Joe Hamman / CarbonPlan / @jhamman
- Martin Durant / Anaconda / @martindurant
- Raphael Hagen / CarbonPlan / @norlandrhagen
- Alex Merose / Google / @alxmrs
- Aashish / KitWare / @
- Tom Augspurger / Microsoft / @TomAugspurger

Agenda

LIII. Update on bakeries / github actions

- GCS bakery is currently in a private repo
- Overwriting/appending: <https://github.com/pangeo-forge/roadmap/issues/29>



- Martin: maybe want to keep the cache / metadata targets around
- Just using one bucket for everything now (cache / metadata / target)
- Should we customize retention policies?
- How to handle this stuff?
 - “Pre-flow-registration step” possible with pangeo-forge-prefect: <https://github.com/pangeo-forge/pangeo-forge-prefect>
 - If post-build cataloging requires a container to run (it will), that will also be a new action (perhaps not a full flow) that needs to be registered with pangeo-forge-prefect. (Cataloging will require

access to bakery secrets, to get write access to catalog bucket.)

- Challenge: currently we do things asynchronously, but target-check should be synchronous

- Do we ever want to delete an active target? Or just update to a new version

LIV. Discussion of large NetCDF file challenges

- Many datasets are distributed with very large (e.g. 20 GB) netCDF files; simply opening these files with xarray + h5py directly from object storage is always very slow (several minutes)

- Examples

- <https://github.com/pangeo-forge/staged-recipes/pull/68>
- <https://github.com/pangeo-forge/staged-recipes/issues/51>

- Possible solutions

- Use fsspec reference maker to open the files as virtual zarrs instead. But they still have to be scanned. How does the total runtime change in this scenario.
- Bypass xarray and use h5py directly. Could be implemented in pangeo-forge-recipes
- Implement chunked writes in pangeo-forge-recipes (distinct from subset_inputs)
- Pursue upstream fixes in xarray + h5py. Starts with very detailed benchmarks.
 - h5py opening issues also arise in recipes with small inputs, e.g. <https://github.com/pangeo-forge/pangeo-forge-recipes/issues/177>, for which each input file is only 80 MB.
- One (short-term) workaround for large input files is to subset them into smaller netCDFs before running the recipe; for example: <https://github.com/pangeo-forge/staged-recipes/pull/56#issuecomment-899187300>
- TODO: make a benchmarking notebook
 - Collect problematic files and make them public
 - Time out carefully
 - (1) Xr.open_dataset
 - (2) h5py.Dataset + data[:] (decoding / encoding)
 - (3) Reference fs
 - (a) Time creation
 - (b) Time subsetting

LV. Documentation updates!

- Newly organized <https://pangeo-forge.readthedocs.io/en/latest/>
- Next step: <https://github.com/pangeo-forge/pangeo-forge-vue-website/pull/8>

- LVI. Reference FS PR!
- <https://github.com/pangeo-forge/pangeo-forge-recipes/pull/174>
 - Ryan implement filepattern
- LVII. Discussion of Beam executor status
- <https://github.com/pangeo-forge/pangeo-forge-recipes/issues/169>
- “Pipelines” model seemed to not work well:
 - <https://github.com/pangeo-data/rechunker/issues/92>
 - Tom removed intermediate step of creating pipelines...but we could bring it back
 - Might be at the stage where we want recipes to define their own steps (stages)
 - everything might be there w/ `__iter__()`
 - https://github.com/pangeo-forge/pangeo-forge-recipes/blob/master/pangeo_forge_recipes/recipes/base.py#L177
 - it yields stages of the recipe (right now 4)
 - we can build off of that
 - Xarray beam is using FilePatterns API!
<https://github.com/google/xarray-beam/pull/31>
 - Example in production:
https://github.com/google/xarray-beam/blob/main/examples/era5_rechunk.py
 - Possible path to beam integration
 - Fix memory issues in rechunker serialization:
<https://github.com/pangeo-data/rechunker/issues/92>
 - Refactor rechunker beam executor to use pipelines API
 - Put pipelines back into pangeo-forge-recipes
- LVIII. Discussion of deprecating `_copy_bt看_filesystems`
- <https://github.com/pangeo-forge/pangeo-forge-recipes/pull/179>
 - https://github.com/intake/filesystem_spec/pull/723
- LIX. Discuss supporting multiple ConcatDims
- <https://github.com/pangeo-forge/pangeo-forge-recipes/issues/140>
- LX. Start a new issue: rename ``store_chunk`` to disambiguate from ``target_chunks``
- LXI.
- LXII.

2021-08-02

Attendance:

- Sean Harkins / DevSeed / @sharkinsspatial
- Joe Hamman / CarbonPlan / @jhamman

- Marti Durant / Anaconda / @martindurant
- Aimee Barciauskas / Development Seed / @abarciauskas-bgse
- Alex Merose / Google / @alxmrs
- Alex Bush / Development Seed
- Rachel Wegener/ U of Maryland / @rwegener2
- Ryan Abernathey / Columbia / @rabernat

Agenda

LXIII. Update on bakeries / github actions

- Whole test flow functionality is working (slash command for recipe test)
<https://github.com/pangeo-forge/staged-recipes/pull/36#issuecomment-885734110>
 - Maybe the prefect automations is not the right solution for webhooks
- Need to finalize feedstock recipe generation

LXIV. STAC update

- Charles' STAC TODO list
<https://github.com/pangeo-forge/pangeo-forge-catalog/issues/1>
- Tour of the new STAC catalog stuff from Charles
<https://github.com/pangeo-forge/pangeo-forge-vue-website/pull/6>
- Do we want to convert metadata into Datacube extension format? Or just copy and render `.zmetadata` as-is?
<https://github.com/radianteearth/stac-browser/issues/87>
 - How much / what type of coordination should we be attempting with Zarr and STAC communities on this subject?
 - (Martin) `xarray` HTML view is an existing option
 - (Ryan) Charles Blackmon-Luca previously made a Zarr metadata vue component
 - (Sean) Datacube provides dimensional search functionality, so it should be included
 - (Ryan) Zarr metadata will be helpful for non-geospatial datasets such as astronomy, plasma physics, etc.
 - Remaining questions: Where to run catalog-generator (based off crawling, or in CI with additional inputs?). Latter is probably the way to go.

LXV. Review of Rachel's documentation post

<https://github.com/pangeo-forge/roadmap/issues/25>

LXVI. Discussion of Pangeo Forge paper

<https://github.com/pangeo-forge/roadmap/issues/24>

LXVII. Finalize workflow sequences for feedstock repositories.

- The proposed workflow logic was described here <https://github.com/pangeo-forge/roadmap/blob/master/doc/adr/0001-github-workflows.md> but we should probably review for a consensus decision.
 - There is also interdependence on for determining whether new release tags should update or replace existing stores.
 - And we should consider issuing some guidance on how different chunking strategies for the same dataset should be logically organized. Should they be different recipes in the same feedstock or entirely separate feedstocks? How should PR's to distinct recipes affect re-processing of unchanged recipes?
- LXVIII. Discuss supporting multiple ConcatDims
<https://github.com/pangeo-forge/pangeo-forge-recipes/issues/140>
- LXIX. Discussion of Beam executor status
<https://github.com/pangeo-forge/pangeo-forge-recipes/issues/169>
- LXX. Working on water-model 370k+ HDF files with reference-maker

2021-07-19

Attendance:

- Charles Stern / LDEO / @cisaacstern
- Sean Harkins / DevSeed / @sharkinsspatial
- Alex Merose / Google / @alxmrs
 - Recently joined “the project” full time (Anthromet – weather infra team)
 - Wants to coordinate more closely with Pangeo Forge
- Ryan Abernathey / LDEO / @rabernat
- Max Grover / NCAR / @mgrover1
- Tom Augspurger / Microsoft / @TomAugspurger

Agenda

- High level review of project status
 - Bakeries
 - Bakery infrastructure for AWS and Azure
 - Coordinating outstanding issues with worker memory and Prefect
 - Prefect is going to start working more closely with Coiled
 - Nothing for Google Cloud
 - Ciaran has left DevSeed
 - GitHub automation
 - GitHub action in staged-recipes

(outstanding PR:)

- Does the recipe registration portion
- GitHub action for creating feedstock repositories is pending
 - How do we want things structured within the repo itself?
 - What are the workflows for the feedstock repositories?
 - Need a new ADR to sort things out
- Current workflow registers the “pruned” version of the recipe; don’t have a mechanism to check the successful run of the recipe. Considering prefect webook machinery.
- Staged recipe progress
 - Spending time on the SWOT model intercomparison project
 - <https://github.com/roxyboy/SWOT-AdAC-ocean-model-intercomparison>
 - https://github.com/pangeo-data/swot_adac_ogcms
 - Challenging to get data from European FTP servers
 - FTP protocol is old
 - `copy_between_filesystems` function can have an alternate path added to it which does not require the added random access complexity, if the only thing we want to do is read from beginning to end
 - 100s of GB of data has been written
 - Mentoring anyone who wants help with Recipes
 - CMIP6 remains a big challenge; working with Diana Gergel
- Pangeo_forge_recipes package
 - Just completed the big refactor to fix serialization (thanks tom!)
 - Subsetting is kind of working
 - H5py / HDF5 issue (link: <https://github.com/pangeo-forge/pangeo-forge-recipes/issues/170>)
 - Since we are working with remote files, it doesn’t matter if it’s not closed -- easy to work around / monkey patch
- Website
 - Not realistic to expect that this will be done soon by DevSeed
 - But we have notes / documents from Rachel → Sean to distil into an issue
 - Deadline for doing something: Charles’ talk in August
- Catalog
 - MS has a lot of stuff in the works, unclear if / when it will be open-sourced
 - Tool for converting a “collection template” to a STAC catalog (inserts

- descriptions, fixes hrefs, etc.)
 - Tool for converting a STAC catalog to HTML.
 - Need to decide whether to wait until that's open-source or write our own.
- Reference datasets:
 - <https://medium.com/pangeo/fake-it-until-you-make-it-reading-goes-netcdf4-data-on-aws-s3-as-zarr-for-rapid-data-access-61e33f8fe685>
- Discussion of priorities for next phase of work
 - Discussion of priorities for remaining DevSeed contract funds.

“We have 60 hours remaining on the contract and I estimate that the above tasks will take less than 15 hours to implement. Given the overlap in a lot of the work we have been billing hours for our contract with Microsoft when feasible for development work that is useful to the wider pangeo-forge community effort. We have a lot of outstanding items to complete on the Microsoft side but our priorities include”

 - Assisting with STAC specification work to expand the Datacube extension to support zarr archives.
 - Updating the Arturo STAC API used at Microsoft to support collection level faceted search.
 - Creating reproducible tests to illustrate and diagnose Dask worker memory management issues with Prefect's task submission model.
 - Making open source contributions to Prefect to improve labeling dynamic infrastructure resources with flow execution information for cost tracking and debugging.
 - Improving debug log search for Dask workers and coordinating with the Prefect team to optimize log shipping where feasible.
 - There are lots of other Microsoft specific things we have outstanding but these are probably the current top priorities. The main question is what you would like us to prioritize with the remaining hours we have for the Columbia contract and going forward what you think we should be prioritizing in our future Microsoft work to get the best results for the wider pangeo-forge community.
 - **Top priorities decisions:**
 - Handling of recipe-level secrets
 - Implementation of GCP bakery
 - Documentation of bakery deployment process: how do I add my bakery to the federation
- Pangeo Forge Recipe features in progress:
 - Secrets in query string:
 - <https://github.com/pangeo-forge/pangeo-forge-recipes/pull/167>
 - Remove ad-hoc stage skipping:
 - <https://github.com/pangeo-forge/pangeo-forge-recipes/pulls>

- Subsetting:
 - <https://github.com/pangeo-forge/pangeo-forge-recipes/pull/166>
 - (Charles) Current draft not work for eNATL60? Reproducible Binder link: <https://github.com/cisaacstern/subset-enatl60>
 - Xref: <https://github.com/pangeo-forge/pangeo-forge-recipes/issues/163>
 - (Martin) h5py closes files unpredictably and possibly in the wrong thread
 - (Ryan) Can we fix this upstream in h5py? That would be ideal. Second best: patch in pangeo-forge-recipes
 - (Martin) If it comes to it, h5py files can be patched to be unclosable (has done this successfully before).
 - (Charles) Can start an issue with Ryan's suggestion for a minimal example: a FilePattern that opens the same file twice.
- run-recipe-test ci action:
 - <https://github.com/pangeo-forge/staged-recipes/pull/61>
- Chat with Alex about Google involvement
- https://twitter.com/pangeo_data/status/1417183130805485568
 - `XarrayVirtualZarr` recipe would be the pangeo-forge-recipes equivalent
- Website / Catalog
 - (Charles): Feedback on current direction?
 - STAC objects: <https://github.com/TomAugspurger/xstac>
 - Backend: <https://github.com/stac-utils/stac-fastapi>
 - Frontend: <https://github.com/pangeo-forge/pangeo-forge-vue-website>

2021-06-21

Attendance:

- Tom Augspurger / Microsoft / @TomAugspurger (will be late)
- Ciaran Evans / Development Seed / @ciaranevans (Not attending, leaving update below)
- Charles Stern / LDEO / @cisaacstern
- Sean Harkins / Development Seed / @sharkinsspatial (I'll be joining approx 15 minutes late.)
- Max Grover / NCAR / @mgrover1
 - mgrover@ucar.edu
- Martin Durant / Anaconda / @martindurant
- //

Agenda

- Recipe Serialization Update (Tom)
 - PR at <https://github.com/pangeo-forge/pangeo-forge-recipes/pull/160>, still some work to do.
 - filepatterns_from_sequence helper is likely to be removed
- Bakery Development (Ciaran)
 - We've moved to using a central base image for all bakery containers <https://hub.docker.com/repository/docker/pangeo/pangeo-forge-bakery-images>
 - Currently working on moving the Azure & AWS Bakeries over to using the centralised image over repo-specific images
 - Also working on further configuration of the Azure AKS/Prefect setup
 - Refactoring work happening on Bakery Database to include the changes to image setup and adding the Azure Bakery
- Dask worker memory leaks with Prefect (Sean)
 - I'm still working with the Prefect team to diagnose this issue.
 - Tom did not report this issue when testing with K8s in Azure so we are trying to isolate if this is related to a specific Dask/Prefect version combination or a Fargate execution environment.
 - Currently unable to test as the NOAA OISST download server is unavailable :{
 - **Question from Charles:** Sean, at what point should we move to manual execution for OISST? (Per [this comment](#), Hillary needs this by July 1.)
- `socket.timeout` raised when accessing source file server, [as recorded here](#). (Charles)
 - Increasing timeout from 30 to 300 seconds doesn't appear to resolve the issue
 - Difficult to record actual elapsed time because a lot is happening behind the scenes when `source.read(BLOCK_SIZE)` is called from storage.py.
 - Larger blocks: might be better so that we don't have to walk to a far-off offset in the file.
- Cataloging implementation questions (Charles; may need to continue offline w/ Tom)
 - Is this basic structure correct? Crawl buckets to populate a database → generate STAC objects from database → render STAC objects to HTML?
 - Is most relevant info for STAC entries extracted from .zmetadata?
 - [TomAugspurger/xstac: STAC from xarray \(github.com\)](#) (generate a STAC collection from xarray dataset)
- Suggestions for running single-threaded caching jobs for manual recipe execution without keeping my laptop open? (Charles)

- Some ideas here: <https://discourse.pangeo.io/t/pangeo-batch-workflows/804>

2021-06-07

Attendance:

- Charles Stern / LDEO / @cisaacstern *not* able to attend today
- Ryan Abernathey / LDEO / @rabernat
- Tom Augspurger / Microsoft / @TomAugspurger (will be late)
- Martin Durant / Anaconda / @martindurant

Agenda

- I. From Charles: Hi all, sad to miss today's meeting. Just wanted to say thanks for making this project and community so great. It feels like we're building momentum every week and it's awesome to see the tools already being put to real, scientific use, e.g. <https://github.com/pangeo-forge/staged-recipes/issues/14#issuecomment-855263961>. Wishing everyone a great week, and I'll look forward to seeing you around GitHub :-)
- II. Discuss memory overflow problem for large recipes
 - A. Diagnosing the problem
 1. <https://github.com/pangeo-forge/pangeo-forge-recipes/issues/151>
 2. <https://github.com/pangeo-forge/pangeo-forge-recipes/issues/116>
 3. Two areas of memory leaks:
 - a) Recipe serialization - large memory overruns in the container where the prefect flow is initialized. (Agent spawns new TaskDefinition to actually read the flow)
 - b) Something happening at the dask level with rapid task execution and submission. Memory increase regardless of task contents.
 - B. Possible fix approach
 1. <https://github.com/pangeo-forge/pangeo-forge-recipes/pull/153>
- III. Pangeo-forge.org website updates
 - A. PR updating content:
<https://github.com/pangeo-forge/pangeo-forge-vue-website/pull/4>
- IV. STAC Update (Tom)
 - A. [Status update](#)
 - B. STAC Storage: <https://github.com/stac-extensions/storage>
 - C. Backend?

1. DevSeed might not have enough time to do cataloging under Columbia contract
2. <https://github.com/stac-utils/stac-fastapi> 👉 + pg backend
3. <https://github.com/radiantearth/stac-api-spec/blob/dev/fragments/filter/README.md>

From Richard Scott Email

It looks like the hard work has already been done. The query extension to the STAC api-spec is deprecated in order to follow the OGC CQL (Common Query Language) and there is a filter extension on the dev branch of the api-spec which details how this would work:

<https://github.com/radiantearth/stac-api-spec/blob/dev/fragments/filter/README.md>

It looks to me like this is exactly what we are after:

/queryables – returns the list of available facets for searching and each facets json-schema document can expose the acceptable values to UIs

/queryables is available at both the root level, individual collection and there is growing support for a cross collection implementation (akin to the opensearch description document where all “queryables” are available and are restricted as your search restricts)

<https://github.com/openeospatial/ogcapi-features/issues/576>

V. Azure Bakery Update (Ciaran / Sean / Tom)

A. Infrastructure in place

<https://github.com/pangeo-forge/pangeo-forge-azure-bakery>

B. Ready to start testing recipes (hitting [fsspec](#) / [adlfs](#) caching issue)

C. <https://github.com/pangeo-forge/pangeo-forge-azure-bakery/issues/4>

D. <https://github.com/dask/adlfs/issues/230>

E. CloudFiles: <https://github.com/zarr-developers/zarr-python/issues/767>

F. Integration Testing:

<https://github.com/pangeo-data/pangeo-integration-tests/issues/1>

VI. ANYTHING ELSE?

2021-05-24

Attendance:

- Ryan Abernathey / LDEO / @rabernat
- Charles Stern / LDEO / @cisaacstern
- Martin Durant / Anaconda / @martindurant

- Max Grover / NCAR / @mgrover1
- Rachel Wegener / DevSeed / @rwegener2
- Martin

Agenda

I. New biweekly meeting structure:

<https://github.com/pangeo-forge/roadmap/issues/21>

- The new meeting structure was discussed and all agreed to continue with the suggestions made in the above-linked Issue #21.

II. Intermittent hanging issues

<https://github.com/pangeo-forge/pangeo-forge-recipes/issues/144>

Discussion goal: develop a plan for how to resolve these issues once and for all

- Sean interested in best logging practices to surface these issues
- Ryan added “timeout” mechanism to pytest
- Martin highlights the challenge of introspecting CI and some possible paths forward
- Ryan wonders if moving async into pangeo-forge-recipes would surface more helpful errors. Hypothesis: rewriting `_copy_between_filesystems` as async could speed things up.
- https://github.com/intake/filesystem_spec/blob/master/fsspec/asyn.py#L179
 1. Options: await `fs._get_file` or `_get`
 2. Await `fs._cat_file(, start, end)`
- Sean suggests decorating tasks with a “task-type” annotation when they are transformed into Prefect flows
 1. Ryan highlights that this may help with concurrency
- Ryan suggests that, if we want to go “all-in” on Prefect, we could abandon the rechunker executor backend pipeline in favor of an implementation directly in pangeo-forge-recipes
- Tom suggests that serialization issue may be easier to address if executors were implemented directly in pangeo-forge-recipes
- Sean: “the good news is that this is not universal”; more difficult to track down, but not found in all circumstances. He’ll increase fsspec logging level to DEBUG
- Martin echoes that `fs._cat_file` (see note above) may have an advantage over `fs.read` for cloud-to-cloud file transfer. Figuring out if the hang is on read or write is a good place to start.
- <https://github.com/pangeo-data/pangeo-integration-tests/issues/1>
 1. Martin agrees these tests are useful, for reassurance
- “Self hosted runners”: Yuvi’s suggestion, which is worth considering further

III. Improving contribution workflow documentation:

<https://github.com/pangeo-forge/staged-recipes/issues/32#issuecomment-843715983>

Discussion goal: align on a plan for coherent messaging about the broader Pangeo Forge project and a timeline for releasing the top-level website.

- Rachel: where to land for new users, categorizing those users into groups, and the importance of middle-ground introductory materials
 - Ryan: front-end budget is with DevSeed; perhaps text on a website doesn't require a designer
- IV. Thoughts on having an "Intake-Zarr" recipe?
- Motivation - currently have catalogs of CESM model output, would there be an easy way to plug this into the current API?
 - Would replace need to specify pattern with merge and concat dim (already in the catalog)
 - <https://github.com/pangeo-forge/pangeo-forge-recipes/pull/141>
 - Ryan suggests using an Intake catalog to represent file patterns internally
 - Unanswered question from Martin: How to represent the source file extension? (.zarr vs .json vs .yaml)
 - Max - taking a look at this, focusing on using catalog of CESM files...
- V. File path ADR
- <https://github.com/pangeo-forge/roadmap/blob/0ee6f5192f8e0598a87081b7d518d9a256b66198/doc/adr/0003-standardize-storage-target-layout.md>
 - From experience, Ryan felt this was too prescriptive in its current form

2021-05-10



Attendance + 60 second update:

- Ryan / LDEO / @rabernat
- Charles Stern / LDEO / @cisaacstern
- Tom Augspurger / Microsoft / @TomAugspurger
- Rachel Wegener / Development Seed / @rwegener2
- Sean Harkins / Development Seed / @sharkinsspatial
- Martin Durant / Anaconda / @martindurant
- Max Grover / NCAR / @mgrover1
- Tom Nicholas / Columbia / @TomNicholas
 - Lurking

Agenda

- I. Introduce Charles Stern
- II. Detailed discussion of recipe submission workflow:

<https://github.com/pangeo-forge/staged-recipes/pull/28#issuecomment-831465586>

- A. Want to require some manual approval before executing anything
- B. Probably require some kind of “development” bakery for executing the test recipes.
- C. Proposal: two stages of CI:
 - 1. Static analysis (flake8, black, etc.) runs on Github Actions
 - 2. Manual approval: A pangeo-forge maintainer requests a “flow run” that executes the recipe on a *subset* of the data on a “development bakery”.
 - a) What gets written out? We want *some* publicly available data.
 - b) Add an intermediate prefix `s3:///staged/<recipe-name>/...` and a policy will automatically delete that prefix.
 - 3. Feedback from the run
 - a) We want some way to expose the result of the flow run to the Github PR. Maybe through a bot? Maybe through a webhook?
 - 4. Do we need this much validation at this stage?
 - a) Or can some stuff be done after creating the new repository?
 - b) Probably worth doing at PR time. Need to do it at some point.
- III. Pangeo Forge Recipes renaming and release
 - A.  renamed package!
 - B.  release problems
 - 1. GH checkout action might not have tag available
- IV. Update on data that has been stored already / cataloging discussion
- V. STAC + Zarr again (Tom)
 - A. Legacy Pangeo catalog based on intake
<https://catalog.pangeo.io/>
 - B. STAC cataloging works well for COG-style imagery datasets, e.g.
<https://planetarycomputer.microsoft.com/catalog>
 - C. But we want to support much broader data schema
 - D. Clear path to supporting collections in stac catalogs
 - E. STAC “item”: single snapshot in space and time
 - 1. Maps well to single satellite image
 - 2. But what about a larger ND datacube?
 - 3. Implement a search that returns a “virtual item”?
 - F. [ESGF / STAC Meeting Agenda](#)
 - G. Is STAC the core “internal” database?
- VI. Dask executor memory problem
<https://github.com/pangeo-forge/pangeo-forge-recipes/issues/116#issuecomment-831553226>

VII. [PLEASE ADD ANYTHING YOU WANT!]

2021-04-25

Attendance + 60 second update:

- Ryan / LDEO / @rabernat
 - Vacation + Pangeo Forge Recipes
- Ciaran / Development Seed / @ciaranevans
 - Played around with github action to respond to PRs in staged recipes
 - Spinning up bakery
- Sean / Development Seed / @sharkinsspatial
 - Github action is fully implemented
- Tom Nicholas / Columbia / @TomNicholas
 - Lurking
- Rachel / Development Seed / @rwegener2
 - Spinning up
- Aimee / Development Seed / @abarciauskas-bgse
 - Lurking
- Max / NCAR / @mgrover1
 - Interested in globus
 - Use case for CESM moving to amazon
- Martin / Anaconda / @martindurant
 - Releases
 - References in parquet or zarr format
 - Fastparquet accel

Agenda

- Switch to Zoom? (people have complained about whereby)
 - Google meet?
- Bakery update (devseed)
- GitHub action (pangeo-forge-prefect)
<https://github.com/pangeo-forge/pangeo-forge-prefect>
 - Discussion questions:
 - Assuming that internal writing / cache operation should use the private endpoint
 - PRs will only have a single meta.yaml

- What about package versions?
 - PipWorkerPlugin is not compatible with prefect flow serialization
 - Need separate images for each pangeo_forge version
 - But logging is working!
- Google Earth Engine coordination issue
 - <https://github.com/pangeo-forge/roadmap/issues/16>
 - Post a specific recipe idea in staged-recipes/issues
 - Respond to the GEE query
- Big recipe refactor
 - <https://github.com/pangeo-forge/pangeo-forge/pull/101>
 - FilePattern object
 - Simpler Recipe class
 - “Recipe Box”
- Release schedule
 - Rename pangeo_forge to...
 - pangeo_forge_recipes
 - pangeo_forge[recipes]
 - Can't really release because we depend on an unreleased xarray feature
 - <https://github.com/pydata/xarray/pull/5065>
 - Also rechunker needs a new release
 - After these are resolved, let's release very often
 - Dependency hell: s3fs botocore version / prefect boto incompatible
- Charles Stern accepted the Columbia job! <https://cstern.io/>

2021-04-12

Attendance + 60 second update:

- Ryan / LDEO / @rabernat
 -
- Sean / Development Seed / @sharkinsspatial
 - Linting stuff for bakeries
 - Bakery stuff
- Martin / Anaconda / @martindurant
 - Releases of fsspec etc.
 - Working on reference maker code (multifile support)
- Ciaran / Development Seed / @ciaranevans

- Ditto on @seans
- Tom / Microsoft / @TomAugsburger
 - Kickoff meeting with devseed happened!
- Tom Nicholas / Columbia / @TomNicholas
- Aimee Barciauskas / Development Seed / @abarciauskas-bgse
 - Lurking
- Charles Stern / @cisaacstern
 - Migrating montane snowpack project to Xarray
- Chiara Lepore/LDEO/@chiaral
 - Update on GEFSv12 recipe
 - <https://github.com/pangeo-forge/staged-recipes/issues/17>
- Max Grover / NCAR / @mgrover1
 - None - sorry for being late.. Thesis defense this week
 - Working on globus related issue

Agenda

- Bakery update (devseed)
 - Been using an internal base docker image
 - Consider using [conda store](#)?
 - Environment pinning discussion
 - Don't want to force recipe to specify all dependencies
 - But do want to pin dependencies on execution
 - What about people who want to add new runtime dependencies to a recipe?
 - Should this be in meta.yaml
 - Should we create a binder "sandbox" for recipe development?
 - What are CPU / memory / local storage reqs?
 - 1 CPU / 4 GB ram
 - 10 GB local file storage
- Discussion questions:
 - <https://github.com/pangeo-forge/pangeo-forge/issues/95>
 - <https://github.com/pangeo-forge/pangeo-forge/issues/93>
 - <https://github.com/pangeo-forge/pangeo-forge/issues/99>
- When do we start on the github workflow / action?
 - <https://github.com/pangeo-forge/roadmap/blob/master/doc/adr/0001-github-workflows.md>
- ECMWF meeting (ERA5)
- Review pending ADRs

2021-03-29

Attendance + 60 second update:

- Ryan / LDEO / @rabernat
 - Running real workloads with pangeo forge
- Sean / Development Seed / @sharkinsspatial
 - Bakery stuff
- Martin Durant / Anaconda / @martindurant
 - PF has exposed some corner cases for fsspec, asyncio, etc.
- Ciaran / Development Seed / @ciaranevans
- David Brochart / QuantStack / @davidbrochart
- Joe Hamman / CarbonPlan / @jhamman
- Max Grover / NCAR / @mgrover1
- Tom Augspurger / Microsoft / @TomAugspurger
-

Agenda

- Architecture Decision Records (<https://github.com/pangeo-forge/roadmap/pull/12>) as a way to communicate stuff.
 - What is the process for reviewing / merging ADRs?
 - One review :+1: required
 - There are three pending ADRs
 - Meta.yaml <https://github.com/pangeo-forge/roadmap/pull/11>
 - Storage Layout: <https://github.com/pangeo-forge/roadmap/pull/13>
 - Bakery database: <https://github.com/pangeo-forge/roadmap/pull/14>
- Bakery Update
 - Prefect agent
 - Dask cluster
 - How to expose dashboard publicly? (issue link)
 - Logging issues with prefect agent? (issue link; related to <https://github.com/pangeo-forge/pangeo-forge/issues/84>)
 - Tom: define level-to-quantity relationships
 - Where to store flows? S3? GitHub?
 - What dependencies are needed by workers
 - <https://github.com/pangeo-data/pangeo-docker-images>
- Pangeo Forge PRs pending
 - Will be renamed pangeo_forge_recipes
 - Tests are still hanging: <https://github.com/pangeo-forge/pangeo-forge/pull/85>
 - Write with Zarr: <https://github.com/pangeo-forge/pangeo-forge/pull/86>

- Copy input to local: <https://github.com/pangeo-forge/pangeo-forge/pull/87>
- Related xarray PR: <https://github.com/pydata/xarray/pull/5065>
- Ryan has been battle testing real recipes at scale
 - SODA: <https://github.com/pangeo-forge/staged-recipes/issues/23>
 - eNATL60: <https://github.com/pangeo-forge/staged-recipes/pull/24>
 - Discovered some dask problems: <https://github.com/pangeo-forge/pangeo-forge/issues/88>
 - Sounds like <https://github.com/pangeo-data/pangeo/issues/788>
 - <https://github.com/pangeo-data/pangeo/issues/788#issuecomment-699166147>

2021-03-15

Attendance + 60 second update:

- Ryan / LDEO / @rabernat
 - STAC ESGF integration meeting last week: [ESGF / STAC Meeting Agenda](#)
- Ciaran Evans / Development Seed / @ciaranevans
 - Working on bakery repo
- Martin Durant / Anaconda / @martindurant
 - Working on fsspec reference maker spec; PR merged ([link](#))! Implementation [PR](#)
 - Will now develop a recipe example
 - Fixing async issues in fsspec ([link](#) and discussion; see also branch ioloop_message2)
- Aimee Barciauskas, Sean Harkins / Development Seed / @abarciauskas-bgse, @sharkinspatial
 - Project management stuff
- Tom Augspurger / Microsoft / @TomAugspurger (will be 10 minutes late)

Agenda

- Check in on overall coordination / communication
 - Design documentation will live in the “roadmap” repo
 - Follow a certain format
- Bakery update!
- Meta.yaml spec
 - <https://hackmd.io/Y27DLJSSQlaMujq2LISKWA>
 - Questions
 - Where should this live
 - Validator?

- Rename “pangeo_forge” repo:
 - <https://pangeo-for.ge/>
 - <https://catalog.pangeo.io/> this is A catalog
- Recipe PR discussion
 - <https://github.com/pangeo-forge/pangeo-forge/pull/81>
 - <https://github.com/pangeo-forge/pangeo-forge/pull/78>

TODO

- Document storage target bucket layout

2021-03-01

Attendance + 60 second update:

- Ryan / LDEO / @rabernat
-
- From Google:
 - Steve Greenberg (sgreenberg@google.com)
 - 80% in cloud on AI / analytics; looking for climate impact
 - Alex Rosengarten / @alxrsngrtn
 - Dave Lowell (dlowell@google.com)
 - 20% project; using ECMWF weather data to help with internal needs in support of wind power data; metnet
- Ciaran Evans / Development Seed / @ciaranevans
 - Starting on AWS bakery soon
- Martin Durant / Anaconda / @martindurant
 - Will start on recipe for fsspec reference maker
- Aimee Barciauskas / Development Seed / @abarciauskas-bgse
- Filipe Fernandes / IOOS / @ocefpaf
- Tom Augspurger / Microsoft / @TomAugspurger

Agenda

- Update on recipe improvements
 - <https://github.com/pangeo-forge/pangeo-forge/pull/78>
- Http testing stuff
 - <https://github.com/pangeo-forge/pangeo-forge/pull/76>
 - Not really sure what the underlying problem is
- Brainstorm recipe metadata spec
 - See discussion here:

https://github.com/pangeo-forge/staged-recipes/pull/20#discussion_r5688535

[52](#)

-

2021-02-15

Attendance + 60 second update:

- Ryan / LDEO / @rabernat
 - Multiple variables PR
 - Working on uneven chunks
- Aimee Barciauskas / Development Seed
 - Updating example in example-pipeline repo
 - There are some staged recipes interested in working on (GPM IMERG, HRRR, GEFS?) this was a first step
- Ciaran Evans / Development Seed / @ciaranevans
 - Mainly on other projects
- David Brochart / QuantStack / @davidbrochart
 - Working on HTTP CI issue:
<https://github.com/pangeo-forge/pangeo-forge/pull/67>
- Tom Augspurger / Microsoft / @TomAugspurger
- Chiara Lepore / LDEO / @chiaral
 - Looking more closely at GEFS

Agenda:

- I. “Steel thread” concept for Pangeo Forge
 - A. Link to whiteboard: https://miro.com/app/board/o9J_lUWGr_U=/
 - B. Pull request to staged-recipes
 1. Can this be run just on Github Action?
 - a) Some of these require a decent amount of compute, memory?
 - b) We should always be able to do just a subset
 - c) We *probably* want to write real data to somewhere, and let reviewers download it, examine it.
 - C. Environment file?
 1. Many recipes will be covered by the “default” environment of pangeo-forge and its dependencies.
- II. HTTP in CI question

2021-02-01

Attendance + 60 second update:

- Ryan / LDEO / @rabernat
 - organizing!
- Julius Busecke / LDEO / @jbusecke
 - Looking around
- Martin Durant / Anaconda / @martindurant
 - Reference maker
- Sean Harkins / Development Seed / @sharkinsspatial
 - Setting up planning for bakeries
- Tim Crone / LDEO / @tjcrone
 - Working with several groups trying to move data into cloud
- Ciaran ["KEERAN"] Evans / Development Seed / @ciaranevans
 - Working on connecting to staged recipes
- Tom Augspurger / Microsoft / @TomAugspurger
 - Figuring out how MS can be more officially involved
- David Brochart / QuantStack / @davidbrochart
 - Making progress on GPM IMERG
- Anderson Banihirwe / NCAR / @andersy005
 - Wants to help out with website!
- Joe Hamman / NCAR & CarbonPlan / @jhamman

Agenda

- I. Project Board overview and status update
 - A. Recipe Implementation project board:
<https://github.com/pangeo-forge/staged-recipes/projects/1>
 - B. software development is tracked in:
<https://github.com/orgs/pangeo-forge/projects/1>
- II. Discussion of project coordination / communication
 - A. Claim issues via github
- III. Discussion of design questions
 - A. <https://github.com/pangeo-forge/pangeo-forge/issues?q=is%3Aopen+is%3Aissue+label%3A%22design+question%22>
- IV. Bakery Design
 - A. Focus on Prefect dask executors 👍
 - B. Dependencies
 1. Traditionally have been making very large catchall docker images
 2. Alternative is to have each recipe specify its own dependencies (docker)

- container)
 - 3. What about environment.yaml? (repo2docker approach)
 - 4. <https://github.com/pangeo-data/pangeo-docker-images/>
 - C. Going to make bakery infrastructure definition cloud-provider specific
 - 1. Contrast from dask-cloudprovider (ephemeral)
 - 2. Rather than build a kubernetes-specific implementation for all clouds, use the cloud provider's specific tooling
 - a) Use terraform
 - b) Containers / networking will be different for different providers
 - 3. Tom: GCP and Azure work well with terraform
- V. Github Workflows

2021-01-18

Attendance + 60 second update:

- Ryan / LDEO / @rabernat
- Filipe Fernandes / IOOS / @ocefpaf
 - Still lurking; waiting to put up recipe
- Tim Crone / LDEO / @tjcrone
 - Not much yet; looking for next steps
- Charles Blackmon-Luca / LDEO / @charlesbluca
 - Sorting out handoff of work responsibilities
- Chiara Lepore / LDEO / @chiaral
 - I am interested in making a copy of the AWS copy of GEFSv12 data
 - <https://noaa-gefs-retrospective.s3.amazonaws.com/index.html#GEFSv12/reforecast/>
 - right now is in grib2 format (which can be opened very slowly with rasterio). The idea is to create a copy in zarr format.
- Aimee Barciauskas / Development Seed / @abarciauskas-bgse
 - Just wrote a proposal to NOAA to do HRRR! (also grib)
 - Rich included a staged-recipes for this
 - <https://github.com/pangeo-forge/staged-recipes/issues/15>
 - Contract in progress
- Joe / CarbonPlan / @jhamman
 - Thinking about learning the ropes of STAC for big remote-sensing ML project; has used but not created STAC catalogs
 - Good conversation with hydroshare; writing a proposal to do cloud-optimized data for hydrologic sciences

Agenda

- VI. Update on refactor (Ryan)
 - A. Recipe
 - B. Pipeline
 - C. Executors
 - D. <https://github.com/pangeo-forge/pangeo-forge/pull/27>
- VII. Organize communication channels for Pangeo Forge
 - A. Too many repos!
 - B. People don't know where to look to follow project
 - C. Do we need a GitHub project board?
 - D. Discourse post / blog post?
 - 1. When is the right time to engage the broader community?
- VIII. Brainstorming of GitHub issues to define next tasks
 - A. Implement multivariable recipe
 - 1. Needs an example (maybe TerraClimate)
 - 2. Discuss how to refactor recipe module
 - B. Start implementing recipes from backlog:
 - 1. <https://github.com/pangeo-forge/staged-recipes/issues>
 - 2. Figure out if recipe can be accommodated by existing features?
 - a) Yes? Great - go for it!
 - b) No? Issue / PR to pangeo-forge for needed feature
 - C. Create documentation site for pangeo-forge
 - 1. Will be useful to host tutorials
 - 2. pangeo-forge/pangeo-forge
 - D. Develop linting / validation for recipes
 - 1. Can the recipe be instantiated?
 - 2. Randomly download inputs?
 - a) What if they are really big
 - 3. Randomly store chunks?
 - E. Create stub for bakery project
 - 1. Repo
 - 2. Basic class
 - F. Get started thinking about GitHub workflows / bots automation
 - G. Flesh out Pangeo Forge website
 - 1. <https://pangeo-for.ge/>
 - H. More executors
 - I. What about versions of recipe vs. execution

1. Recipe identified by github org + github repo + ref
 2. But execution is context dependent: when bakery executes a recipe, it gets a specific version ID
 3. What metadata tags will we embed in datasets?
- J. What infrastructure do people want to use?
1. OSN (S3 compatible)
 2. Wasabi

2020-12-21

Attendance + 60 second update:

- Ryan / LDEO / @rabernat
- Charles Blackmon-Luca / LDEO / @charlesbluca
 - Looking into intake-stac to allow support for opening of Zarr datasets
- Tom Augspurger / Microsoft / @TomAugspurger
 - Looking into performance of netCDF → Zarr: [Netcdf to Zarr best practices - Data - Pangeo](#)
- Aimee Barciauskas / Development Seed / @abarciauskas-bgse
 - New laptop 🙄
 - Recipe for IMERG
- Filipe / IOOS / @ocefpaf
 - Not much / helping Rich
- Rich Signell / USGS / @rsignell-usgs
 - HRRM - hourly forecast model; trying to construct a “best timeseries” for forcing ocean models (pick and choose between different forecast members). Implementing the traditional method of converting grib2 to netcdf and then using rechunker, but *thinking* about creating metadata for fsspec’s new referenceFileSystem backend that simply accesses chunks from appropriate files to achieve the same goal.
- David Brochart / QuantStack / @davidbrochart
 - Works at quantstack, xtensor Zarr
 - Here to listen; maybe implement Zarr GDAL driver
- Tim Crone / LDEO / @tjcrone
 - Finishing up teaching course; helping students share data via Google Drive; would like more gdrivefs capabilities
- Martin Durant / Anaconda / @martindurant

- Release of fsspec / s3fs / fastparquet / intake
- Working on “derived dataset” idea for intake
- Julius / LDEO
 - Lurking

●
I.

II. 2020-12-21

III. Attendance + 60 second update:

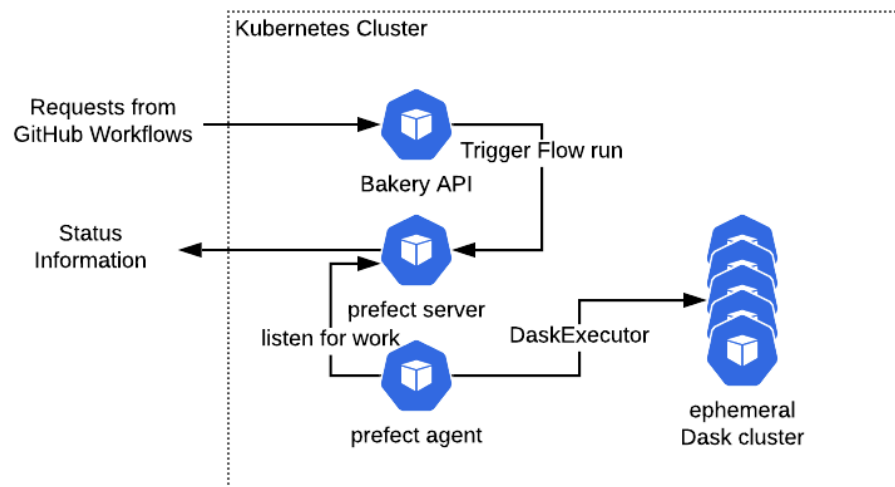
IV. Ryan / LDEO / @rabernatal discussion questions:

A. Do we agree about the canonical Recipe execution sequence?

<https://github.com/pangeo-forge/pangeo-forge/pull/27>

This specifies the interface between Recipe and Bakery. It's a crucial abstraction.

1. Design of interface:
 - a) Prepare
 - b) Cache_inputs
 - c) Store_chunks
 - d) Finalize
 2. Tim: what about more fine-grained dependencies between store_chunk and cache_input
 3. Rich: can we call subprocesses (wgrib2) from store_chunk? Tom: we can process these on remote workers on their own remote filesystems
 4. What about post-processing hooks?
 5. Julius: what about post-processing stuff like trend / anomaly / climatology
 6. How to put rechunker as part of a recipe
 - a) Allow N parallel stages
 7. David: what about data dependencies
 - a) Intermediate temporary data vs. derived data
 - b) Martin: complex graphs of dependencies in datasets
 - c) Conda forge bot watchers
- B. What goes into a Bakery?



<https://lucid.app/lucidchart/invitations/accept/137757a8-8a5f-46c1-89e5-981078f97320>

1. Tom: maybe we don't need the Bakery API pod
 2. Do we need many prefect servers or just many agents?
 - a) Server is hard to deploy; manage entire federation from one server
 - b) Deploying agent is easy; use tags to manage location of run
 3. How do bakeries provide runtime configuration to recipes (target bucket, cache, etc.)
- C. How should we assign dataset names?
1. Recipe specifies <dataset_name>, multiple unique datasets allowed in each repo
 2. Fully qualified ID:
<recipe_github_org>/<recipe_github_repo>/<dataset_name>
 3. Storage target:
<target_prefix>/<recipe_github_org>/<recipe_github_repo>/<dataset_name>/<run_id>
 4. Do we need org if all recipes are in the same org?
 5. Do we need an identifier for the "run" of the dataset?
 6. Run vs. version?
 7. Conda forge: **build number** + package version
 8. Are "datasets" immutable in pangeo forge?
 9. Store history about how dataset was built
 - a) CF "history" attribute

- 10. How do we monitor changes of dataset over time
 - a) E.g. upstream dataset fixes correction
- V. Tom's NetCDF to Zarr conversion
- VI. Concrete TODOs
 - A. Merge Ryan's PR
 - B. Develop more example recipes
 - C. Refactor recipe class to be more flexible
 - D. Deploy documentation site for Pangeo Forge
 - E. Create Bakery repo, start deploying Bakery somewhere

2020-12-07

Attendance + 60 second update:

- Ryan / LDEO / @rabernat
- Charles Blackmon-Luca / LDEO / @charlesbluca
 - Working on cataloging / PySTAC stuff
- Martin Durant / Anaconda / @martindurant
 - Working on intake pipeline for derived data: declarative text file vs. code
- Naomi Henderson/LDEO/ naomi-henderson@users.noreply.github.com
 - Listening!
- Rich Signell / USGS / @rsignell-usgs
 - Conversion of 4TB of national water model data: still frustrated
 - 200,000 input files...a few bad ones
- Filipe Fernandes / IOOS / @ocefpaf
 - Working on rechunker, helping with package
- Aimee Barciauskas + Sean Harkins / Development Seed
 - <https://github.com/pangeo-data/rechunker/issues/35>
 - GPM IMERG dataset
- Diana Gergel / Rhodium Group/climate impact lab / @dgergel
 - Listening!
- Tom Augspurger / Microsoft / @TomAugspurger
 - No updates
- Anderson / NCAR / @andersy005

Agenda:

- I. GOAL: split this massive projects into pieces we can work on independently

II. Pangeo Forge Website:

- A. <https://pangeo-for.ge/>
 - 1. Catalog
 - 2. Docs about Recipes
 - 3. Status
 - 4. Logo is trademarked -- need to get permission from NumFocus!
- B. <https://github.com/pangeo-forge/pangeo-forge-vue-website>
- C. Vue so we can incorporate stac browser:
<https://github.com/radianteearth/stac-browser/>

III. Overview of refactor

- A. Goal of refactor:
 - 1. More modular
 - 2. Make it easier to write a recipe
 - 3. Make it easier to debug recipes
- B. Presentation of Recipe + Target + Executor framework
- C. StandardSequentialRecipe
- D. Recipe mixins to be implemented
 - 1. OPeNDAP inputs
 - 2. Many chunks per input file
 - 3. Multiple variables in different files
 - 4. Zarr to Zarr
 - 5. Rechunking step

IV. Brainstorming about architecture

- A. Does a recipe specify its target? (Ryan votes no)
 - 1. Target = filesystem + path
 - 2. If not, how does a target get determined
- B. What do we call the thing that executes the recipes in the cloud?
 - 1. "Oven", "Bakery", "Kitchen"? 😊
 - a) Look into chef (automated deployment tool)
 - 2. Standalone service; deploy via helm chart
 - 3. One bucket per Bakery? Or multiple?
 - 4. Knows about secrets (bucket credentials)
- C. What do we call the thing that dispatches the Bakeries?
 - 1. "Dispatcher", "Coordinator", "BakeryManager"?
 - 2. Needs to respond to GitHub events
 - 3. Needs to tell Bakeries what to do
 - 4. How does the BakeryManager choose what bakery to send the recipe to?

V. Brainstorming about Catalogs

- A. Is the bucket the master source of truth?
1. YES? - then we need to regularly crawl all buckets in the pangeo forge universe (or otherwise dispatch services when they are updated)
 2. NO? - then we need some other source of truth, i.e. database
 - 3.

2020-11-23

Attendance + 60 second update:

- Ryan
 - AGU Poster
 - Code refactor
- Joe / CarbonPlan / @jhamman
 - Using terraclimate datasets
- Charles Blackmon-Luca / LDEO / @charlesbluca
 - Working on cataloging, opened an [issue on pangeo-datastore](#)
 - How to make a browser for Pangeo Forge generated catalogs?
 - [STAC Index](#) could be a start
- Aimee Barciauskas / Development Seed / @abarciauskas-bgse
 - Working on Pangeo Forge recipe for GPM IMERG
- Sean
 - Working on prefect / dask stuff
 - Fargate
- Martin Durant / Anaconda / @martindurant
 - Conversation around fsspec-reference-maker etc
 - Link to issue <https://github.com/intake/fsspec-reference-maker>
 - 👉 **seeking comments via issues!**
 - Intake derived datasets?
 - Link to issue / discussion?
- Filipe Fernandes / IOOS / @ocefpaf
 - Watching Pangeo Forge from the side
- Tom Augspurger / Microsoft / @TomAugspurger
 - Conversations at MS
- Rich Signell / USGS / @rsignell-usgs
 - National Water Model: 225,000 hourly netCDF files on S3
 - Rechunker issue with xarray: link?
 -

Agenda

- Refactor
 - Rechunker model
 - Separate Recipe and Executor
 - What is the atomic unit
- Overview of Pangeo Forge repos
- DevSeed folks
 - Aimee: made some minor changes to pangeo forge workflow
 - Sean: NASA work focused on building orchestration pipelines for ETL pipelines
 - Very AWS focused, lots of step functions
 - Making things slightly more platform agnostic
- Theo from Met Office has a similar poster using Dask CloudProvider
- Effort availability
 - Ryan: Managing NSF project
 - Tom: uncertain; don't plan on my effort
 - Joe: booked out through early / mid January; after that, in roadmap
 - Filipe: keep following development; IOOS (17 different federal agencies, 11 regional orgs) supports Pangeo Forge
 -

2020-11-09

Attendance + 60 second update:

- Ryan Abernathey / LDEO / @rabernat
 - SOW for subcontract has been written
<https://docs.google.com/document/d/1zo66onTXExpKgQ6ZeRnpH1MgBhKhoEeQka1eAAq00I/edit>
 - Paper on “Cloud Native Data Repositories” submitted
<https://www.authorea.com/doi/full/10.22541/au.160443768.88917719/v1>
 - Heavy traffic / discussion on CMIP6 pipeline
- Jim Bednar / Anaconda / @jbednar
 - Working on subcontract SOW
- Joe / CarbonPlan / @jhamman
 - Catching up after being out for the last while
- Martin Durant / Anaconda / @martinduran
- Julius Busecke / LDEO / @jbusecke
 - Working on detrending use case
- Charles Blackmon-Luca / LDEO / @charlesbluca
 - Looking at pystac / stac collections

Agenda:

- Subcontracts
- Overview of where we are at on Pangeo Forge tech
 - Brain dump from Tom
 - Infrastructure for evaluating prefect
 - A few example recipes
 - Batch workflow discussion
 - <https://discourse.pangeo.io/t/pangeo-batch-workflows/804>
 - <https://argoproj.github.io/argo/>
 - <https://ploomber.readthedocs.io/en/stable/>
 - <https://github.com/pangeo-forge/staged-recipes/issues>
- Ryan's most recent attempt at a feedstock:
 - <https://github.com/pangeo-forge/noaa-oisst-avhrr-feedstock>
- Need to write a blog post announcing pangeo forge / cloud repo kickoff

2020-10-26

Attendance + 60 second update:

- Ryan Abernathey / LDEO / @rabernat
 - <https://www.overleaf.com/2783152265mqjwbfrwkgq>
- Tom Augspurger / Anaconda / @TomAugspurger
 - Worked on pangeo forge all day on Thursday
 - Pipelines structure is reasonable
 - Don't have to worry about storage / environment
 - Ci for staged-recipes is there
 - Workflow stuff: how do I develop a pipeline?
 - How heavily do we expect people to
- Charles Blackmon-Luca / LDEO / @charlesbluca
 - Responding to reviews on stac-browser PR (Zarr viewer)
 - <https://discourse.pangeo.io/t/google-storage-gs-urls-for-pangeo-datasets-on-gcs/975>
- Discussion:
 - Cataloging
 - We want to include these generated datasets in a (central, distributed?) catalog
 - Recipes will have a recipe/meta.yaml, pangeo-forge will generate the

STAC catalog for each recipe.

- TODO for charles: start thinking about how to add a catalog step to the pipeline: <https://github.com/pangeo-forge/pangeo-forge/issues/25>
- <https://github.com/pangeo-cmip6/sync/issues/1>
 - <https://github.com/treeverse/treeverse-distcp>

2020-10-12

Attendance + 60 second update:

- Ryan Abernathey / LDEO / @rabernat
 - Tried to make a feedstock:
<https://github.com/pangeo-forge/noaa-oisst-avhrr-feedstock>
- Jim Bednar / Anaconda / @jbednar
 - Just lurking; looking for a status update
- Tom Augspurger / Anaconda / @TomAugspurger
- Charles Blackmon-Luca / LDEO / @charlesbluca
 - Working on Zarr vue stuff
 - CMIP6 data on google cloud is getting synced using github action
<https://github.com/pangeo-cmip6/sync/blob/main/.github/workflows/rclone.yml>
- Felipe
 - Sent a of PR
<https://github.com/pangeo-forge/staged-recipes/pull/11>

Agenda

- Sprint!

2020-09-28

Attendance + 60 second update:

- Ryan Abernathey / LDEO / @rabernat
 - Not much for pangeo-forge recently. Back!
- Tom Augspurger / Anaconda / @TomAugspurger
 - Not much since getting the terraclimate recipe up
- Joe
 - Got terraclimate feedstock working on partial dataset, stalled on larger one

- <https://github.com/pangeo-forge/staged-recipes/pull/12>
- Martin Durant / Anaconda/ @martindurant
 - zarr/xarray concurrent read
 - Gcsfs at scale
- Charles Blackmon-Luca / LDEO / @charlesbluca
 - Working on GitHub actions for cron jobs for synchronizing data

Agenda

- What do we need from a subcontract
 - <https://github.com/pangeo-forge/roadmap/issues/6>
 - Automation bits are hard
 - Frontend
 - Prefect stuff?
- What are the blockers?
 - Joe is the only one who has written a full recipe; what are we generalizing?
Need to create more recipes
 - Start from <https://github.com/pangeo-forge/staged-recipes/issues>
 - Then manually create new repos in pangeo-forge org (skip automation)
 - Once we have a few recipes, could start setting up automation for going from staged recipe to feedstock
 - “Test flow” - validate a feedstock’s viability
 - Validate that sources exist
 - Make sure prefect flow is valid / compiles
 - Small dataset: run full flow in test location
 - Large dataset: randomly sample from sources
- Manual QC
- How do people develop recipes? Make transition from hub to recipe as smooth as possible.
 - Start from examples
 - Need tools to debug / develop
- Relationship between recipes + environments / docker images
 - Should we have a pangeo hub for recipe developers?
 - Use a standard base image + optional extra dependencies
 - Pin everything
 - Prefect builds a docker image when you register a flow (based off base docker image)
- What is the best choice of “flow storage” for Pangeo Forge
 - gcs / s3 vs. docker vs. github
 - <https://docs.prefect.io/api/latest/environments/storage.html>

- GitHub workflow exits before flow completes: how to handle flow status

TODOs

- Ryan: will develop a pipeline for some of my data
 - Will need to sync with Tom / Joe
- Ryan: develop SoW for potential subcontract
- Joe: may revisit terraclimate feedstock?
 - <https://github.com/pangeo-forge/terraclimate-feedstock/issues/3>
-

2020-09-14

Attendance + 60 second update:

- Tom Augspurger / Anaconda / @TomAugspurger
 - Main update is on the terraclimate-feedstock
- Joe Hamman / CarbonPlan / @jhamman
 - <https://github.com/pangeo-forge/staged-recipes/pull/12>
- Charles Blackmon-Luca / LDEO / @charlesbluca
 - Working on syncing GCS and S3
 - Thinking with Ryan about large catalogs and databases
 - [Best practices for maintaining a large dynamic STAC catalog?](#)
- Diana / RHG
 - On vacation, not much of an update
 - Still working on CMIP6
- Aimee / Devseed
 - Looking at building a staged recipe for IMERG
- Rich / USGS
 - Cloned the terraclimate feedstock, going to use it on an opendap / netcdf subset dataset

Agenda

- Walk through terraclimate-feedstock example
- Scaling infrastructure (prefect on k8s vs. AWS batch vs. ...)
- CMIP6 sync:
 - <https://github.com/carbonplan/data/blob/master/.github/workflows/rclone.yml>
 - `rclone sync s3:rsignell/nwm/test_week5c jetstream:rsignell/nwm/test_week5c`

--checksum --fast-list --transfers 16

○

2020-08-31

Attendance + 60 second update:

1. Ryan Abernathey / LDEO / @raternat
2. Joe Hamman / CarbonPlan / @jhamman
3. Tom Augspurger / Anaconda / @TomAugspurger
 - a. Tried deploying Prefect on Pangeo
 - b. Got agent working
4. Jeremiah Lowin / Prefect / @jlowin
 - a. ML research, timeseries, financial services
 - b. PMC at Apache Airflow
 - c. Formed prefect
5. Filipe Fernandes / IOOS / @ocefpaf
 - a. Wants to work with Rich on moving some IOOS SECOORA (ROMS) regional forecast model data into cloud
6. Jim Bednar / Anaconda / @jbednar
7. Rich Signell / USGS / @rsignell-usgs
 - a. Wants to work with Filipe on using pangeo-forge & rechunker on IOOS SECOORA model forecast data
 - b. Revisit what met office did with forecast data

Agenda:

1. Quick Pangeo Forge overview of problem (Ryan + Joe)
 - a. Already have dask
2. Prefect discussion
 - a. Prefect is evolving rapidly, lots of refactoring in place
 - b. Prefect exposes high-level API for running the flow
 - c. Prefect is flexible--don't have to write things in a certain way; don't worry!
 - d. Just use Joe's code, but get it to run, we need
 - i. Flow itself
 - ii. "agent"
 - iii. And a "server" for the agent to talk to

- iv. “Executor” execution environment
- e. Hybrid model
 - i. Server needs no knowledge about the execution environment
 - ii. Execution env only receives metadata from the server; only reports back state updates and metadata
 - iii. Server can be centrally located with privileged access
- f. Prefect cloud provides users
 - i. Tenants + Agents
- g. How to do Prefect + Dask?
 - i. Ask Jim Crist 😊
 - ii. Parameterize the executor itself
 - 1. Specify in the flow
 - iii. Use dask kubernetes cluster provisioned by agent
 - iv. How to make sure the dask cluster has access to the dependencies
 - v. Flow is serialized with cloudpickle
- h. Prefect does not manage auto-scaling; rely on kubernetes
- i. Should we use server or cloud?
 - i. Run server
 - ii. Anything related to executing code is in open-source
 - iii. User accounts in cloud
 - iv. Publicly viewable dashboards?
 - 1. Not yet but very soon
 - v. Or could implement your own using graphql API
 - vi. “Cloud hook” - webhook on any state change
- j. Prefect is “putting an API on top of your data application”
- k. Versioning of flows
- l. Tom: will use cloud for now
 - i. “Service account” user
 - ii. Read only user
- 3. Update on CMIP6 meeting with CEDA
 - a. No

TODO:

- Follow up with Jim Crist about Prefect + Dask
- Next steps for Prefect
 - Insert dask executor into flow
 - What environment?
- Squash Zarr appending bug

2020-08-24

Attendance:

8. Ryan Abernathey / LDEO / @rabernat
9. Filipe Fernandes / IOOS / @ocefpaf
10. Jim Bednar / Anaconda / @jbednar
11. Martin Durant / Anaconda / @martindurant
12. Tim Crone / LDEO / @tjcrone
13. Aimee Barciauskas / Development Seed / @abarciauskas-bgse
14. Julius Busecke / Princeton / @jbusecke
15. Naomi Henderson / LDEO / @naomi-henderson
16. Joe Hamman / CarbonPlan / @jhamman
17. Diana Gergel / Rhodium Group / Climate Impact Lab / @dgergel
18. Charles Blackmon-Luca / LDEO / @charlesbluca
19. Rich Signell / USGS / @rsignell-usgs

Intros:

- Ryan
- Filipe here because of conda-forge parallel
- Jim manages viz tools at Anaconda, wants to make it feasible to visualize large cloud-based datasets. Also managing Anaconda's role in EarthCube project
- Martin, works on intake, fsspec, zarr, Dask
- Tim Crone, marine geophysicist and data scientist, connected to OOI and interested in helping other ocean and Earth scientists work with their large datasets more efficiently
- Aimee, data engineer with DevSeed, working on a project with NASA Earth Data in support of cloudification of NASA data, cloud optimized formats.
- Julius working with CMIP6 in the cloud, cmip6 preprocessing
- Naomi led the data management for CMIP6 in the cloud
- Charles working on Pangeo stuff at Lamont, managing Pangeo cloud catalogs, sees opportunities to improve workflow for users
- Joe putting up lots of cloud-optimized data for Carbon Plan
- Diana, climate scientist at Rhodium Group / CIL, long term user of Pangeo, trying to downscale CMIP6 models, contributing to CMIP6 pipeline

- Rich Signell, physical oceanographer at USGS, IOOS, OOI, Pangeo advocate.

Agenda:

- Review of Pangeo Forge EarthCube Proposal (Ryan)
 - <https://www.overleaf.com/read/yxjffxyxwsmw>
- Overview of Where Pangeo Forge Code is at (Joe)
 - <https://github.com/pangeo-forge>
 - Roadmap
 - Brainstorm space
 - Pangeo-forge
 - The “engine” that does the data transformation
 - Analogous to conda-build
 - Not necessarily the automation part
 - staged-recipes / pangeo-smithy
 - Automation bits
 - Not much here yet
 - Review use case submissions
- Update on CMIP6 (Diana / Charles / Naomi)
 - History (Naomi): Started as a collection of jupyter notebooks
 - Search ESGF archive (API)
 - Download datasets to a local server
 - Produce Zarr
 - Upload to GCS
 - Create catalog entry
 - Handle data retractions (who other workflow)
 - Current status (Diana):
 - What can we automate vs. not? Missing data, retractions, etc.
 -
- Open discussion
 - How to contribute?
 - Dataset people (Tim, Rich, Aimee, etc.): Babysit recipe repos, iterate on development
 - Anaconda:
 - Work on cataloging
 - Solve issues in open source projects being used (Intake, Dask, HoloViz, fsspec, plus integrate with Prefect
 - Tom: cloud + dask + some prefect stuff
 - Martin: fsspec, intake, dask, etc.