For announcements on ESS-DIVE activities (i.e. webinars, publications, new feature announcements)...

Follow ESS-DIVE on Twitter! @ESSDIVE

Join ESS-DIVE's Community Mailing List!
https://groups.google.com/a/lbl.gov/g/ESS-DIVE-Community

ESS Large Data session notes

Example Large Dataset : UAS dataset https://doi.org/10.5440/1778212
COGS example https://github.com/TESTgroup-BNL/exploringCOGs

A: What if your data is already at NERSC?
Q: Yes, you could just copy the data into our NERSC resources but we want to caution against not using ESS-DIVE for data storage so easily

A: Do I understand this correctly? If we have data files in a hierarchy, we can upload them using the Globus option to preserve the hierarchy instead of uploading it as a zip file using the API?
Q: uploading via Globus will preserve the hierarchy but it means you are storing data in the Tier 2 and only the metadata in the primary storage.

Is it possible/viable to have a copy of the hierarchical data files in Globus as well as published as a zip on ESS-DIVE?

- Shawn and Terri are willing to be guinea pigs for this process

A: Of course I realize model data will likely be netCDF and some might want HDF5, as an interim I was thinking that for basic RS perhaps geotiff/COG is a good interim solution for retrieving data like a THREDDS/nc API would work but clearly in the future with model output THREDDS may be better
Q:

A: Do you see this being linked ESGF or other repos that can host large data files?

Q: We did a prototype of this last year to think about this more. We do think that there will be data thatr belongs in other repositories so we will need a refined external linking...

A: Do you have an example of an ESS-DIVE record linked to Tier 2 so we can see how it all connects?
Q: We have a demo but we won't have something worth showing until mid-summer

A:
Q:

COMMENTS

Ben Sulaman: Honestly I feel like being able to preserve a folder hierarchy outweighs almost any other added value factor in a data repository.

Shawn: right - esp if the apache interface is there since that still allows for remote extraction via other tools or at least on-demand download of individual files

Kyle Redilla: Yeah, +1 for ability to query subsets of large data via API

Shawn:
An example of something that works really well for sharing large data and preserving hierarchy?
To me one of the most useful repos I have found given the immense functionality, R/Python API for data retrieval, wikis, and other tools is OSF.org
https://osf.io/sn6em/

you can create really nice all-encompassing archives that are openly accessible. I could see a benefit to making data similarly easy to organize into a folder hierarchy within a DIVE portal to serve up data in a similar way; for the most part its already on track for this

Ben: Maybe this is a me problem but I've had pretty bad luck using Globus in the past. I ran into weird permission errors on one end or the other, and most recently I ended up falling back on scp or rsync because getting globus to work didn't seem worth the time investment when there was a simpler option.

told him that we would be helping to manage permissions

Thanks, that sounds good. I think my main goal would be flexibility in case some tools work better than others for some users

I feel like almost every data set I've ever used, big or small, is easier to navigate with a hierarchical structure than a flat list of files

indeed. Even better if you can query a package and get back the directory tree in a manner that is parsable on the user end

Terri: FLMD is zipped into the data file folder which does not make it easily accessible. Additionally, FLMD takes file hierarchy into account-- should we have FLMD outside the zipped folder and within the zipped folder? You don't want to lose the FLMD when you open the zipped folder if the file isn't included.

maybe a directory tree inside a zip might be simpler (and easier to generate)?

David +1 that hierarchical data files should be prioritized

Shawn: I think to me these all relate to a formal data download support backend. 1) hierarchical data storage; 2) API requests to allow a user to query what's in a data package and their individual file URLs; 3) (stretch goal) allow for API to facilitate server-side processing/extraction to return only what you need from files.

What about managing licenses across external data linking?

- What about providing an easy access citation from the external source?