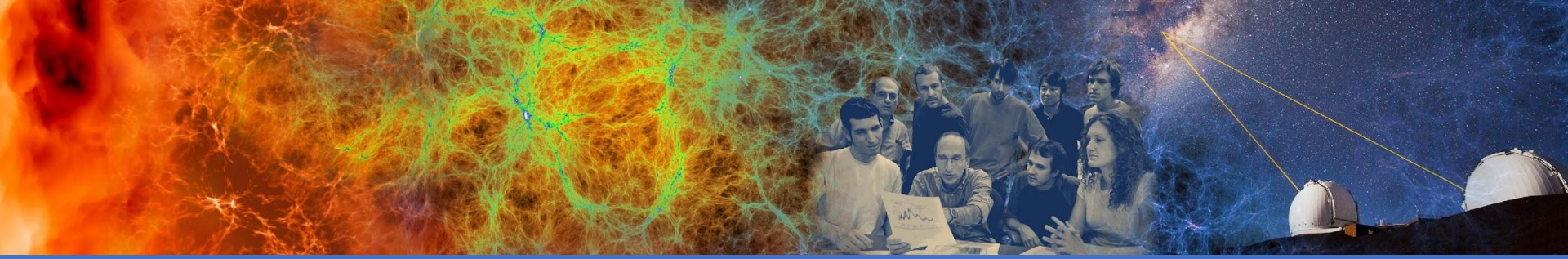
Data Storage and Sharing Best Practices

New User Training
February 16, 2024

Lisa Gerhardt
Data, AI, And Analytics Group



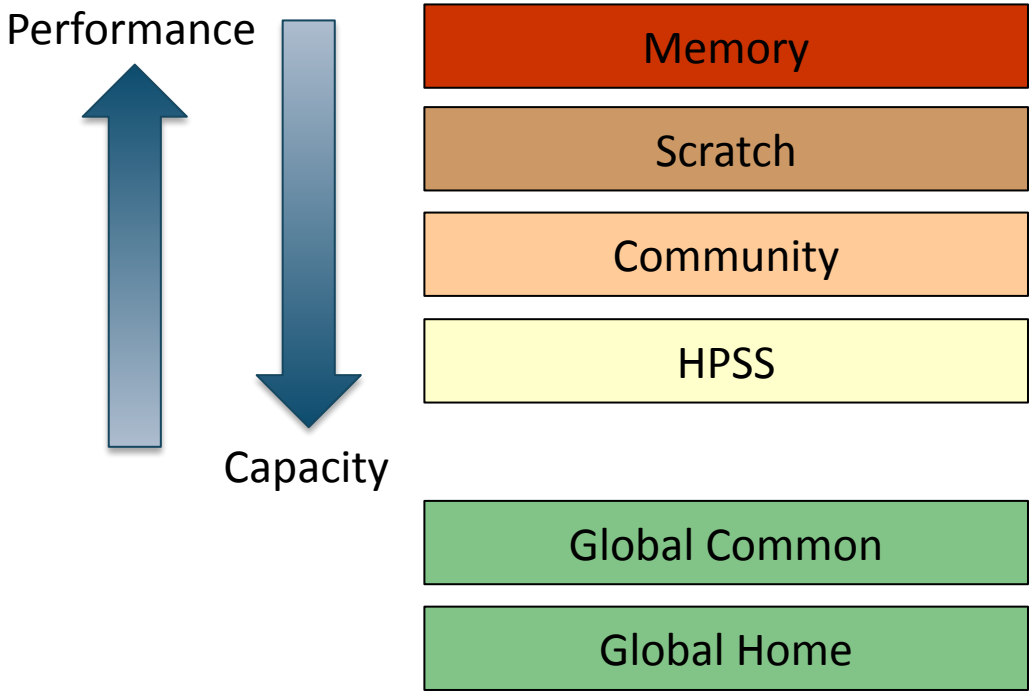
Data Storage Best Practices

NERSC Data Storage Policy

- NERSC provides its users with the means to store, manage, and share their research data products
- NERSC resources are intended for users with *active* allocations. It is strongly recommended that if you no longer have an allocation at NERSC, you transfer your data somewhere that you have access.
- PIs can request the modification, deletion, or transfer to another NERSC file system of data associated with their NERSC award
- Files are protected only using UNIX file permissions based on Iris user and group IDs. It is the user's responsibility to ensure that file permissions and umasks are set to match their needs
- Users have ultimate responsibility for managing and backing up their data

https://docs.nersc.gov/policies/data-policy/policy/

Simplified NERSC File Systems



36 PB SSD Perlmutter Scratch

Lustre 6 TB/s, temporary (purge)

114 PB HDD Community

Spectrum Scale (GPFS)

150 GB/s, permanent

315 PB Tape HPSS Archive

Long Term

20 TB SSD Global Common Software

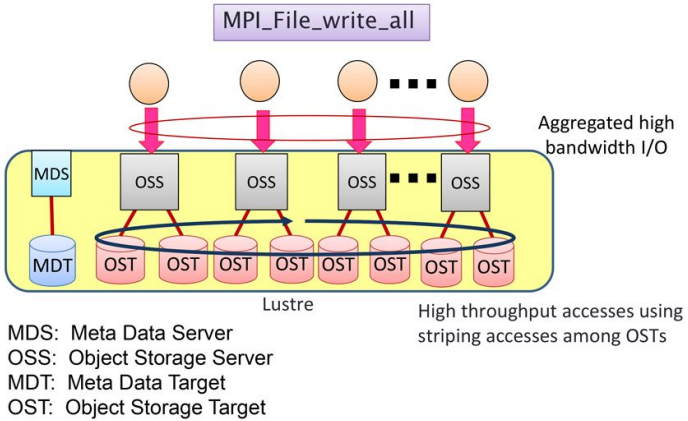
Spectrum Scale, Permanent

Faster compiling / Source Code

https://docs.nersc.gov/filesystems/

Perlmutter Scratch

- Lustre, one of the most successful/mature HPC FS. **This is where to store data being actively read or written by jobs on computes**
- Directories are user-readable and writable by default
- Purged! Back up any important data
- Quotas are 20TB (soft) and 30TB (hard). After you exceed the hard quota, you will not be able to write any more data to the file system



Using MPI-IO on Lustre[1]

1. MPI-IO on Lustre: https://www.sys.r-ccs.riken.jp/ResearchTopics/fio/mpiio/

“Scratch”: Optimize Performance with Striping

Scratch Striping Recommendations

- By default data on 1 OST, ideal for small files and file-per-process IO
- Single shared-file I/O should be striped according to its size
- Helper scripts

  `stripe_small`, `stripe_medium`, `stripe_large`

- Manually query with

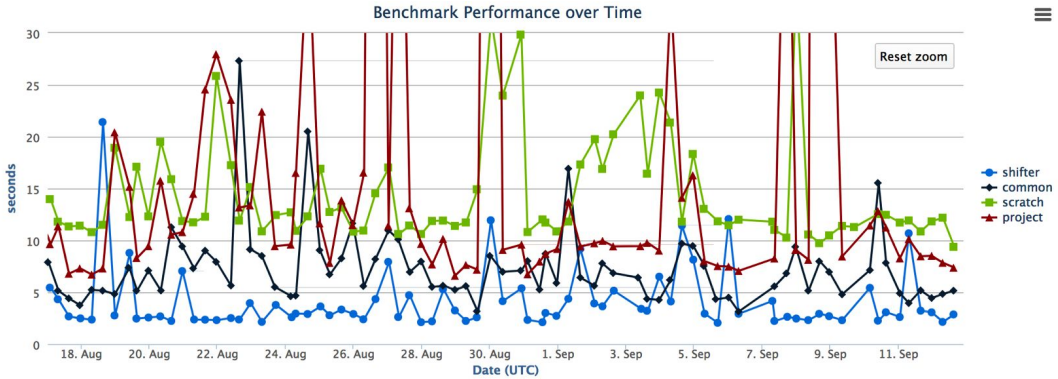
  `lfs getstripe <path>`

- Set striping on a directory
  - New files will automatically pick it up
  - Copy files in to inherit the striping

	Single Shared-File I/O	File per Process
File size (GB)	command	
< 1	keep default striping	keep default striping
1 - 10	`stripe_small`	keep default striping
10 - 100	`stripe_medium`	keep default striping
> 100	`stripe_large`	keep default striping
> 1000	`stripe_large`	`stripe_large`

Global Common: Software Filesystem

- For: software stacks - Why? Library load performance, and enhanced caching



- Group writable directories similar to community, but with a smaller quota, `/global/common/software/<projectname>`
  - Write from login node; read-only on compute node
- Smaller block size for faster compiles than CFS

Community File System

- For: large datasets that you need for a longer period
- Set up for sharing with group read permissions by default
- Not for intensive I/O - use Scratch instead
- Use the “dvs_ro” mount if you’re reading from CFS during jobs
- Data is never purged. Snapshots. Usage managed by quotas
- Projects can split their space allocations between multiple directories and give **separate** working groups **separate** quotas
  - Environment variable $CFS points to /global/cfs/cdirs

https://docs.nersc.gov/filesystems/community/

HPSS

- For: data from your finished paper, raw data you might need in case of emergency, really hard to generate data
- HPSS is tape!
  - Data first hits a spinning disk cache and gets migrated to tapes, cache is sized for several days of retention
  - Files can end up spread all over, so use `htar` to aggregate into bundles of 100 GB - 2 TB
  - Archive the way you intend to retrieve the data
  - `hsi` and `htar` give the best performance within NERSC
- Quotas are controlled in Iris. If you're a member of multiple projects you can adjust the percentage you want charged to each

Home Directories

- For: source files, scripts for **testing**, notes
- 40G quota
- Not intended for intensive I/O (e.g. application I/O) - use Scratch instead
- Backed up monthly by HPSS
- Snapshots are also available e.g. my homedir is at `/global/homes/.snapshots/2022-06-14/e/elvis`

General Advice for I/O

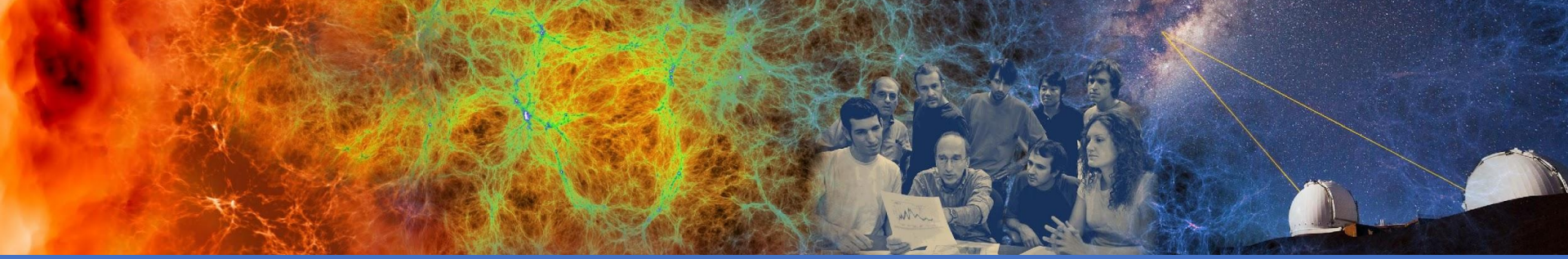
- I/O from batch jobs should go to Perlmutter's scratch file system (/pscratch, $SCRATCH)
  - Input data
  - Configuration files
  - Output data
- Software for batch jobs should go in a container or to Global Common (/global/common/software/<your_project_name>)
  - Conda environments
  - Anything you install with config / make / cmake etc.
- Don't generate a million small files, especially not in one directory
- Aggregating reads and writes into bigger pieces is generally better

DVS at NERSC

- DVS is an I/O forwarder that's been used on NERSC systems for many years
- Uses a set of 24 nodes to forward I/O and offer high performance
- Two type of mounts: read-only and read-write
- Read-only serves files with all 24 I/O nodes (each one pushes a little piece), and has large cache
- Read-write picks ONE I/O server to handle a file that's determined at file creation, very small cache
- **CFS, Global Common, and Homes are mounted via DVS on the compute nodes so using them at scale requires special consideration**
  - Keep in mind that scale means "multiple concurrent jobs", so 100 single node jobs that all start at once count for this too!

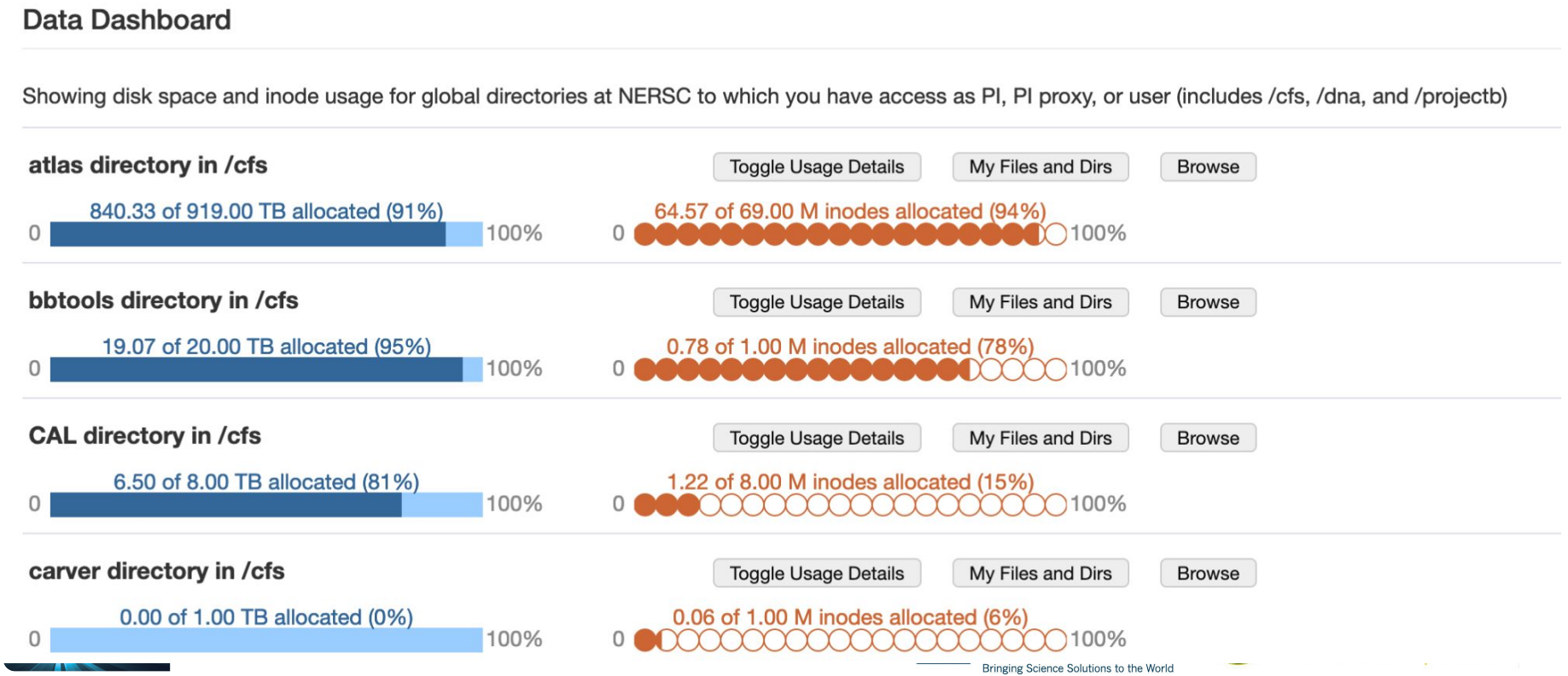
Best Practices for DVS

- Conda environments should be in a container or global common
  - By default they install to your home dir, which causes **A LOT** of problems at scale
  - Also, if you load a conda environment at login, **ALL** of the very large number of library paths are dragged along to your slurm job. Consider whether you want this or not
  - Python automatically adds your current working directory to the library load path
- Best choice for large scale I/O is always scratch!
- If your data is too large and you need to read it off of CFS, use “/dvs_ro” instead of “/global”
    - “/global/cfs/cdirs/myproject/mega_important_config” -> “/dvs_ro/cfs/cdirs/myproject/mega_important_config”
- Avoid ACLs on files over DVS. These keep the system from using any caching and slows things down

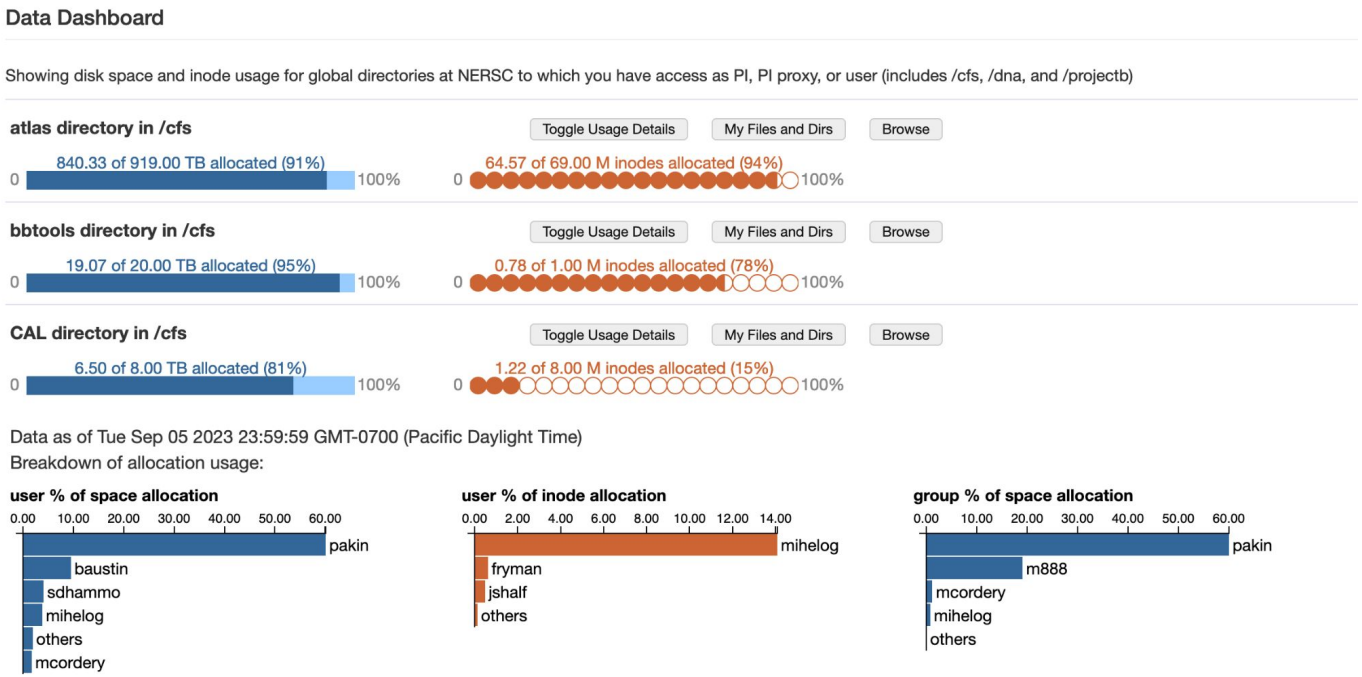


Data Management Tools

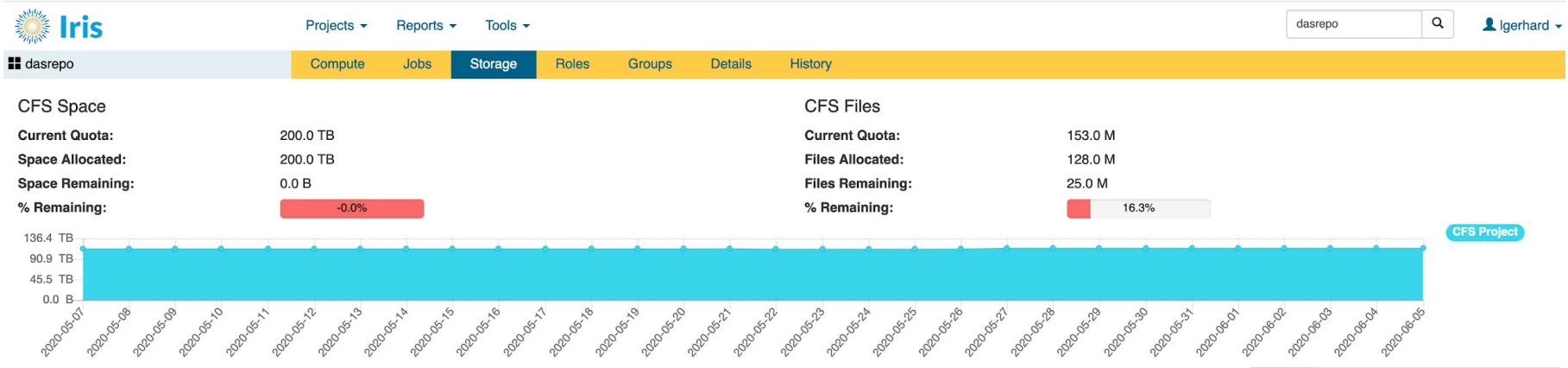
Data Dashboard in my.nersc.gov



Data Dashboard: Usage Reports



Adjusting Quotas in IRIS

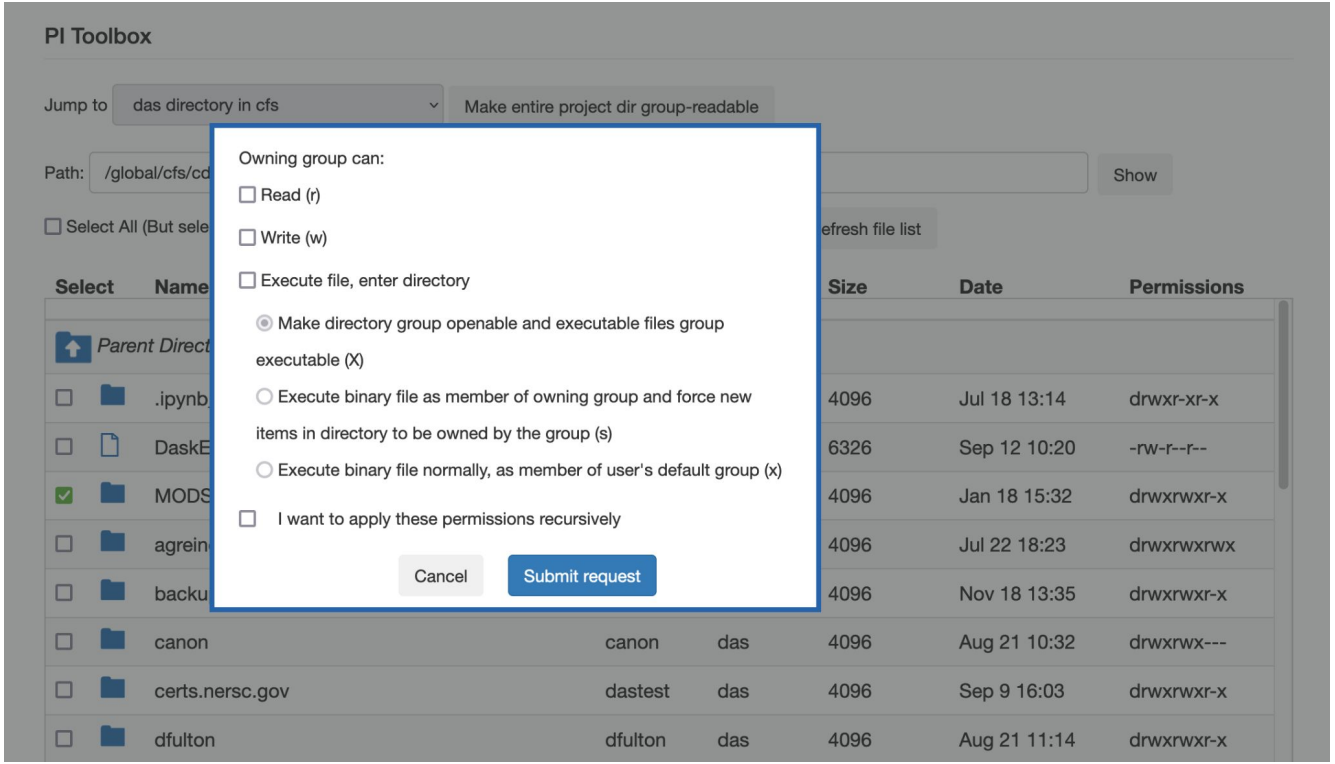


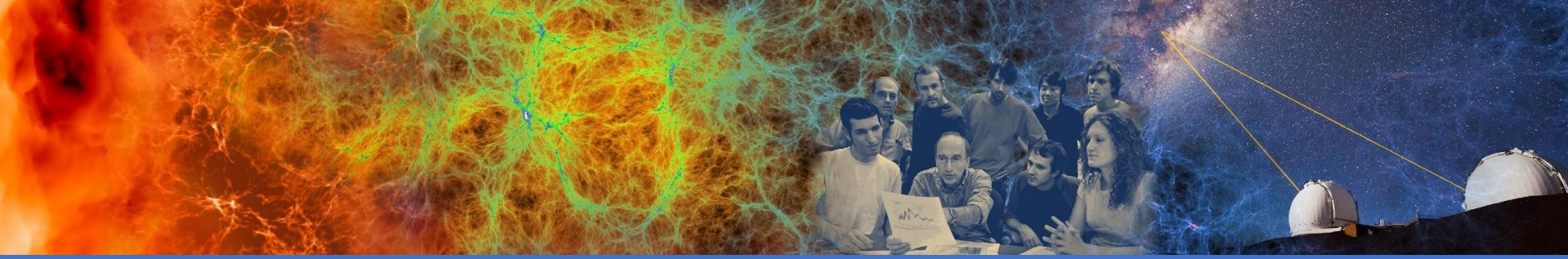
CFS Directory Usage

+ New | Edit | Rename | Activate

Directory	File System	Owner	Group	Active	Storage Used	Byte Limit	% Storage Used	Files Used	File Limit	% Files Used	Updated On
dasrepo	gpfs	Prabhat, Mr	dasrepo	✔	60.8 TB	90.0 TB	67.5%	27 M	100.0 M	26.6%	2020-06-05
ProjectDisCo	gpfs	Gerhardt, Lisa	projectd	✔	44.4 TB	91.0 TB	48.8%	134 K	20.0 M	0.7%	2020-06-05
mantissa	gpfs	Prabhat, Mr	mantissa	✔	5.2 TB	10.0 TB	51.8%	641 K	1.0 M	64.1%	2020-06-05
das	gpfs	Prabhat, Mr	das	✔	4.0 TB	5.0 TB	80.2%	1 M	2.0 M	59.7%	2020-06-05
ClimateNet	gpfs	Gerhardt, Lisa	climaten	✔	888.3 GB	1.0 TB	86.7%	108 K	1.0 M	10.8%	2020-06-05
datamap	gpfs	Gerhardt, Lisa	datamap	✔	111.4 GB	1.0 TB	10.9%	910 K	2.0 M	45.5%	2020-06-05
gbclimat	gpfs	Pseudo User, g…	gbclimat	✔	0.0 B	1.0 TB	0.0%	1	1.0 M	0.0%	2020-06-05
dastest	gpfs	Pseudo User, d…	dastest	✔	0.0 B	1.0 TB	0.0%	1	1.0 M	0.0%	2020-06-05

PI Toolbox: my.nersc.gov/pitools/





Data Sharing Best Practices

Sharing Inside of NERSC

- **Community File System: CFS**
  - Every project has at least one directory that has permissions set up to be group writable and readable
  - PI Toolbox (my.nersc.gov/pitools/) can manage permissions
- **HPSS Project Directories**
  - Directories in HPSS with group writable and readable permissions
- **Collaboration Accounts**
  - Account tied to a group instead of an individual user. Access is controlled by the project's PI. Useful for managing shared datasets, running shared workloads
- **Scratch**
  - User who desire to share data on scratch can do it by adjusting Linux permissions
    - Only share read access. If you want to allow writes, we recommend using a collaboration account instead
    - `chgrp -R <project_name> $SCRATCH; chmod g+rX $SCRATCH` (read only)
- **give / take**
  - Mechanism to give single files to any other NERSC user

Sharing with External Collaborators

- **Public HTML access**
  - Project specific area can be created:
    - `/global/cfs/cdirs/<yourproject>/www`
  - These are available for public access under the URL:
    - `https://portal.nersc.gov/project/<yourproject>/`
- **Science Gateways** (docs.nersc.gov/services/science-gateways/)
  - Web portals allow you to interface with your data and computation at NERSC
  - For more sophisticated web applications: **Spin** (docs.nersc.gov/services/spin/)
- **Globus Sharing** (docs.nersc.gov/services/globus/#globus-sharing)
  - Projects can set up read-only endpoints for sharing data with certain Globus users
  - Excellent way to share large volumes of data, can be incorporated into web pages

NERSC's Dedicated Data Transfer Nodes

- **Data Transfer Nodes** (DTNs, https://docs.nersc.gov/systems/dtn/)
  - Dedicated servers for moving data at NERSC (dtnXX.nersc.gov)
  - Servers include high-bandwidth network interfaces & are tuned for efficient data transfers
    - Monitored bandwidth capacity between NERSC & other major facilities such as ORNL, ANL, BNL, SLAC…
  - Direct access to Community, HPSS Archive, and Cori Scratch
- Use NERSC DTNs to move large volumes of data in and out of NERSC or between NERSC systems
- User Perlmutter Login nodes for data transfers to Perlmutter Scratch

General Tips for Transferring Data: Globus

The **recommended** tool for moving data in, out & within NERSC

- Reliable & easy-to-use web-based service:
  - Automatic retries
  - Email notification of success or failure
- Accessible to all NERSC users
- NERSC-managed endpoints on DTNs for optimized data transfers
- Web based GUI for drag-and-drop transfers
- NERSC Globus scripts for command line transfers
- REST/API for scripted interactions with service
- Globus Connect Personal for setting up endpoints on your laptop

https://docs.nersc.gov/services/globus/

Other Tips for Transferring Data

- Use Globus Online for **large transfers**
  - Also for internal NERSC transfers e.g. between CFS and scratch
- Transfers involving HPSS need special care (more later)
- **scp** is fine for **smaller, one-time transfers** (<100MB)
  - But note that Globus is also fine for small transfers
- **Don’t use DTN nodes for non-data transfer purposes**
  - Use system login nodes for more general routine tasks

Performance Considerations

- Performance is often **limited by the remote endpoint**
  - Not tuned for WAN transfers or have limited network link
  - These can lower performance <100 MB/sec.
- File system contention may be an issue
  - Try the transfer at a different time or on a different FS.
- **Don't use your $HOME directory for I/O!**
  - Instead use CFS, $SCRATCH …
- If you think you are not getting the transfer rates you expect, let us know: help.nersc.gov

Transferring with NERSC HPSS

- HPSS tape archive is recommended for **archiving large amounts** of data for **long periods of time**
  - See: https://docs.nersc.gov/filesystems/archive/
- Use interactive DTNs or xfer queue to transfer to / from HPSS
  - HSI for individual files and conditional access
  - HTAR for aggregation & optimization of storage/archival of large numbers of files. Aim for bundle sizes of 200GB - 2 TB
- NERSC Globus Command line tools for external Globus transfers
  - Will automatically sort files in tape order
  - However the Globus web interface does not directly support aggregation with HTAR or tape-ordering
  - Preferred use: small number of large files

https://docs.nersc.gov/services/globus/#command-line-globus-transfers-at-nersc

Conclusion

- NERSC has multiple file systems to fulfill different performance and capacity needs
- Many different ways to share and transfer data
- Further reading: https://docs.nersc.gov/filesystems/

Thank You and Welcome to NERSC!