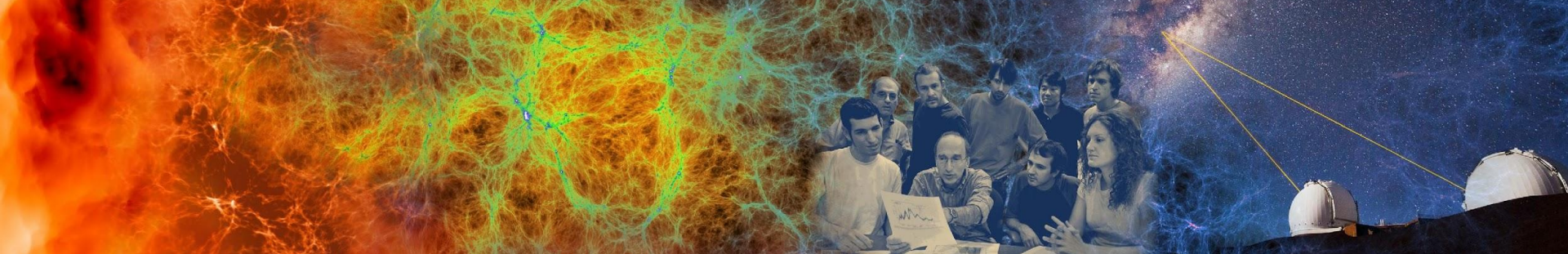


# Data Storage and Sharing Best Practices



New User Training  
February 16, 2024

Lisa Gerhardt  
Data, AI, And Analytics Group



# File Systems Overview



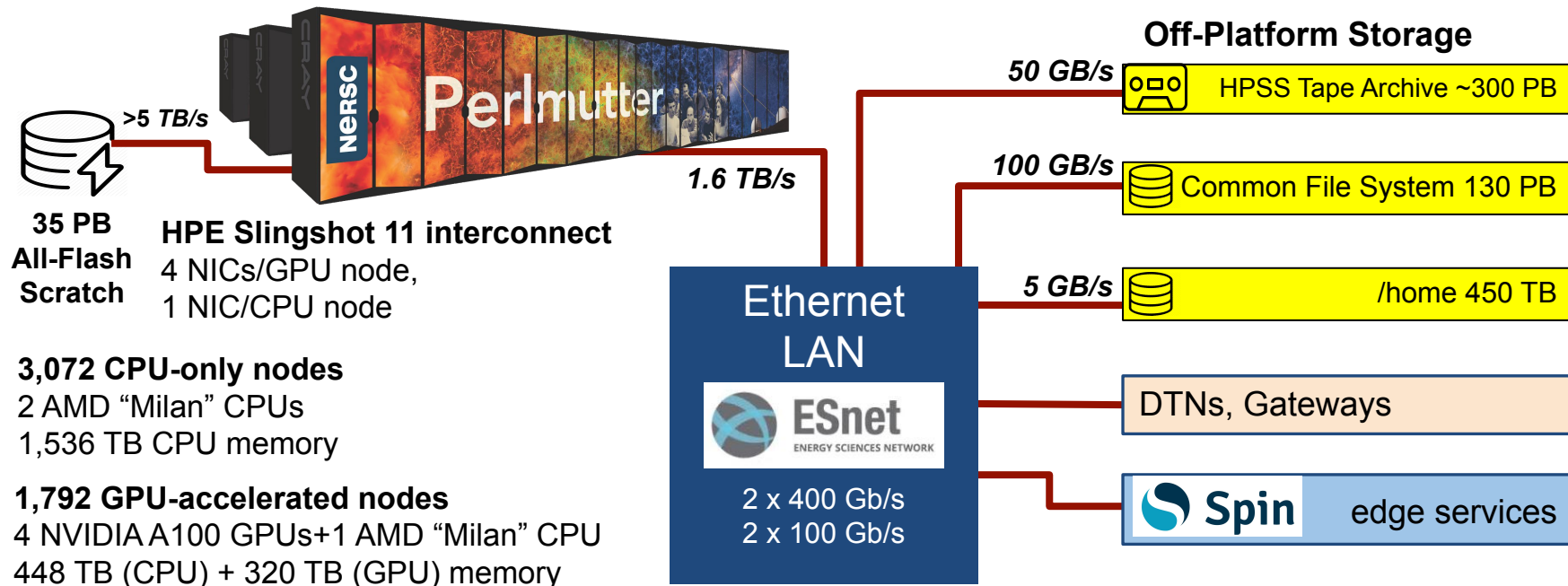
BERKELEY LAB



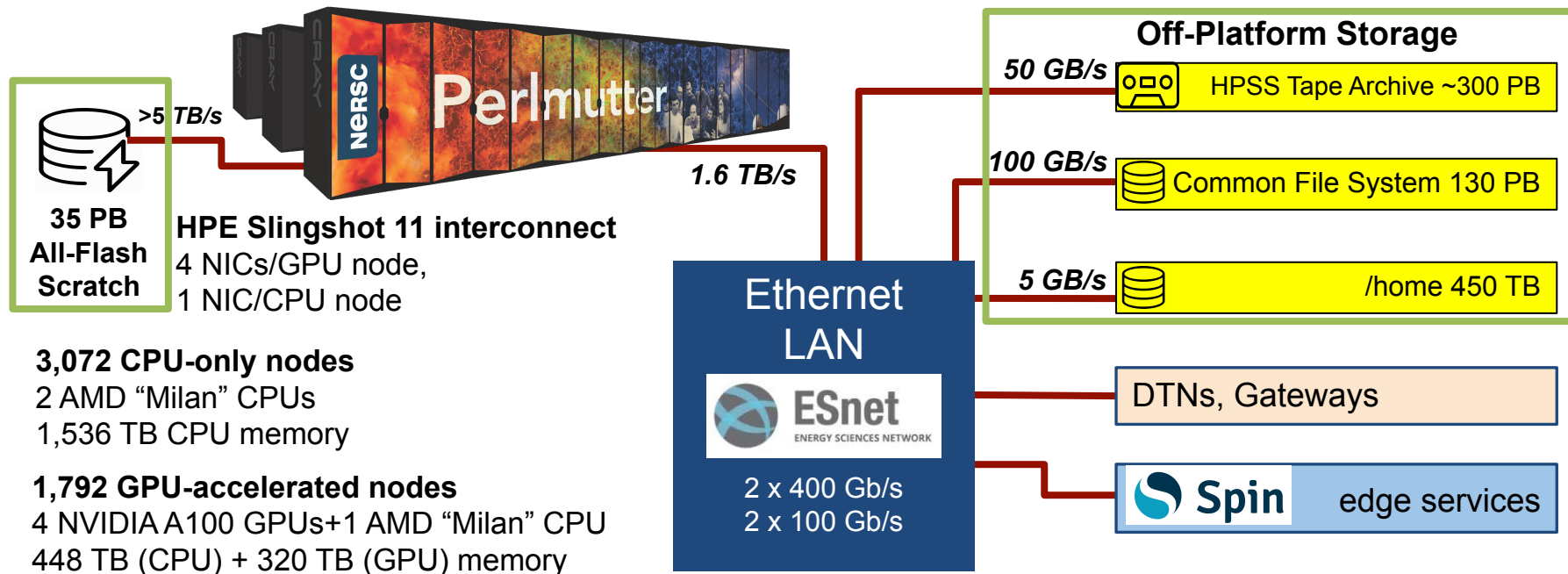
U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science

# The System is a Sum of Many Parts!



# The System is a Sum of Many Parts!

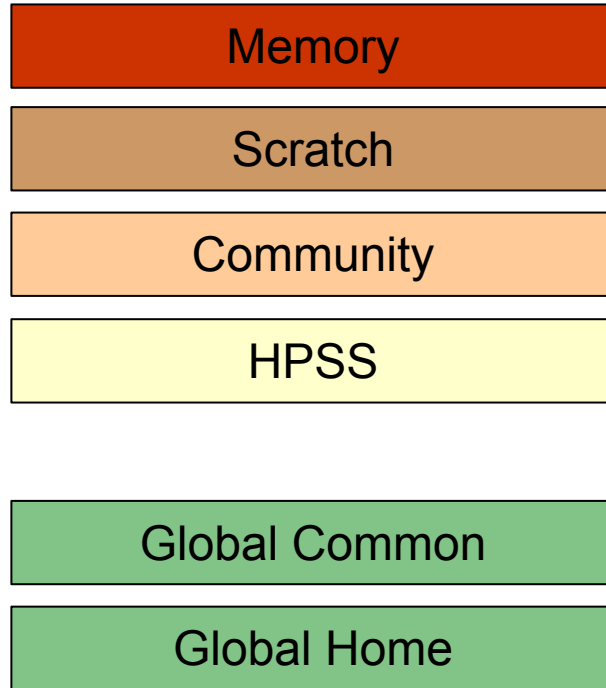


# Simplified NERSC File Systems

Performance



Capacity



## **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/>



# NERSC Data Storage Policy

- NERSC provides its users with the means to store, manage, and share their research data products

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



# 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.

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



# 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

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





# 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

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



# 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/>



# Perlmutter File Systems

## Global Home

- Permanent, relatively small storage
- NOT tuned to perform well for parallel jobs
- Snapshot backups
- **Perfect for storing data such as source codes, shell scripts**
- `cd $HOME`

# Perlmutter File Systems

## Global Home

- Permanent, relatively small storage
- NOT tuned to perform well for parallel jobs
- Snapshot backups
- **Perfect for storing data such as source codes, shell scripts**
- `cd $HOME`

## Global Community File System (CFS)

- Permanent, larger storage
- Medium performance for parallel jobs
- Snapshot backups
- **Perfect for sharing data within research group**
- `cd $CFS`

# Perlmutter File Systems

## Global Home

- Permanent, relatively small storage
- NOT tuned to perform well for parallel jobs
- Snapshot backups
- **Perfect for storing data such as source codes, shell scripts**
- `cd $HOME`

## Global Community File System (CFS)

- Permanent, larger storage
- Medium performance for parallel jobs
- Snapshot backups
- **Perfect for sharing data within research group**
- `cd $CFS`

## Local Scratch

- Large, temporary storage
- Optimized for read/write operations, NOT storage
- Not backed up
- Purge policy (8 weeks)
- **Perfect for staging data and performing computations**
- `cd $SCRATCH`

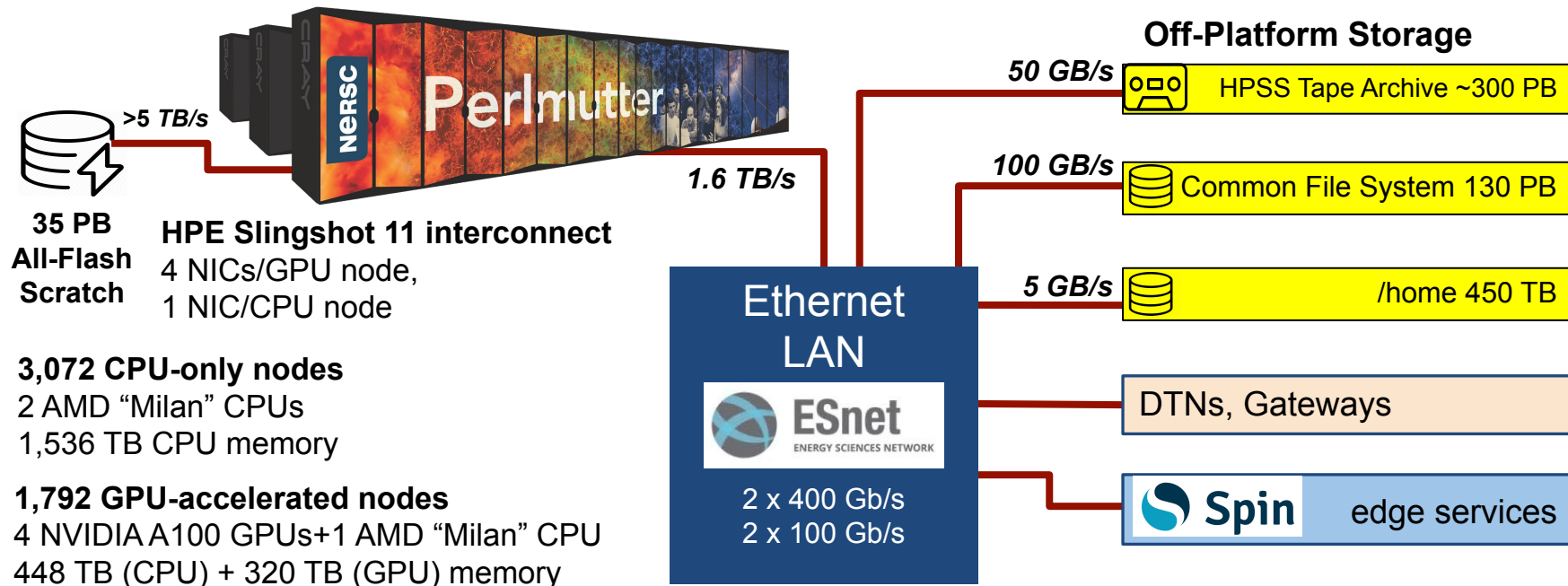
# Long-Term Storage System

## HPSS

- High-Performance Storage System
- Archival storage of infrequently accessed data
- Hierarchical storage:
  - Data first ingested onto high-performance disk arrays
  - Migrated to large enterprise tape subsystem for long-term retention



# The System is a Sum of Many Parts!



# Perlmutter Scratch (>5TB bandwidth)

- **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

**How to:** move your run scripts into scratch and have your application set up to read/write data within scratch (e.g. file paths to locations within scratch)

# Perlmutter Scratch (>5TB bandwidth)

- **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

**How to:** move your run scripts into scratch and have your application set up to read/write data within scratch (e.g. file paths to locations within scratch)

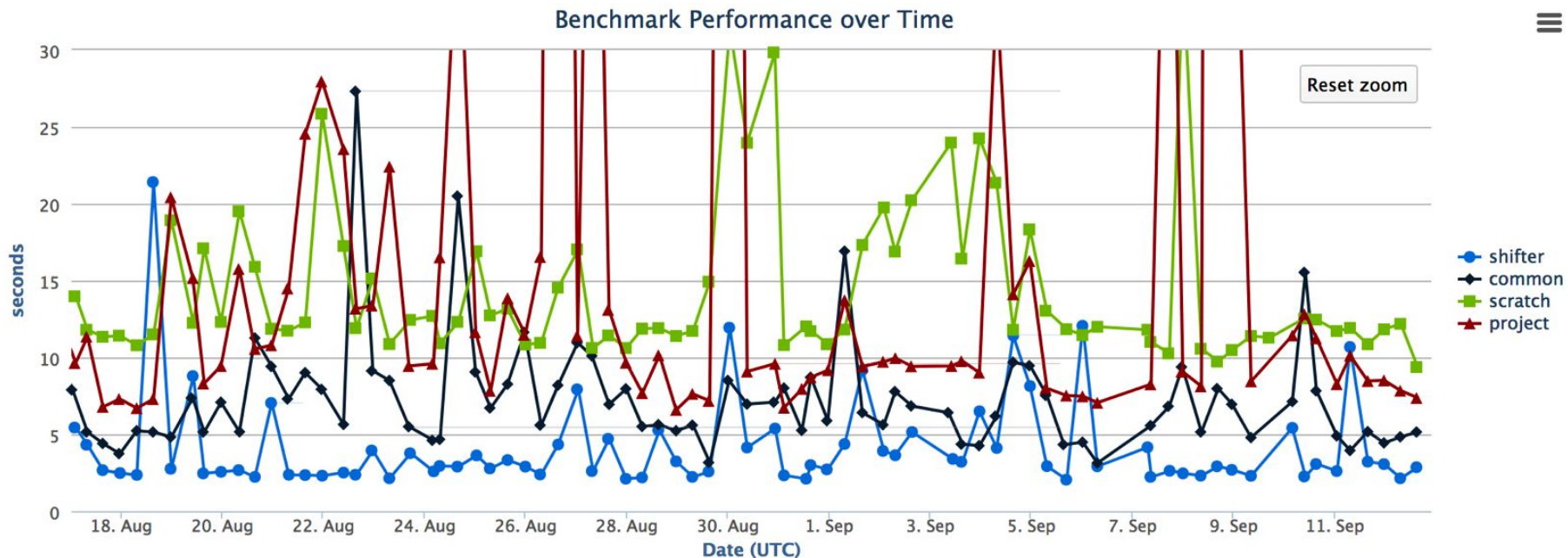
Advanced users may want to optimize data striping:

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

<https://docs.nersc.gov/performance/io/lustre/>

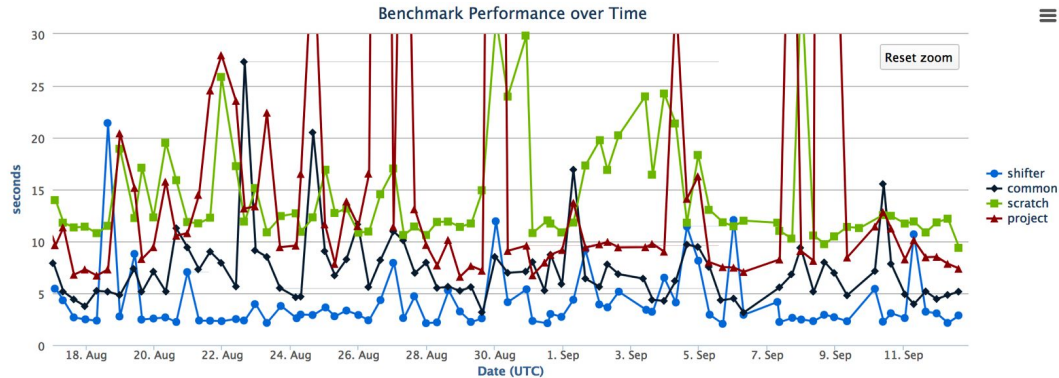
# Global Common: Software Filesystem

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



# 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

## How to:

```
cd $CFS/<projectNumber>
```

```
mkdir <name_of_choice>
```

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



# Community File System

- Use the “dvs\_ro” (/dvs\_ro/cfs) mount if you’re reading from CFS during jobs
- Data is never purged
- Projects can split their space allocations between multiple directories and give **separate** working groups **separate** quotas

## How to:

```
cd $CFS/<projectNumber>
```

```
mkdir <name_of_choice>
```

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

# Community File System

- Use the “dvs\_ro” (/dvs\_ro/cfs) mount if you’re reading from CFS during jobs
- Data is never purged
- Projects can split their space allocations between multiple directories and give **separate** working groups **separate** quotas

## How to:

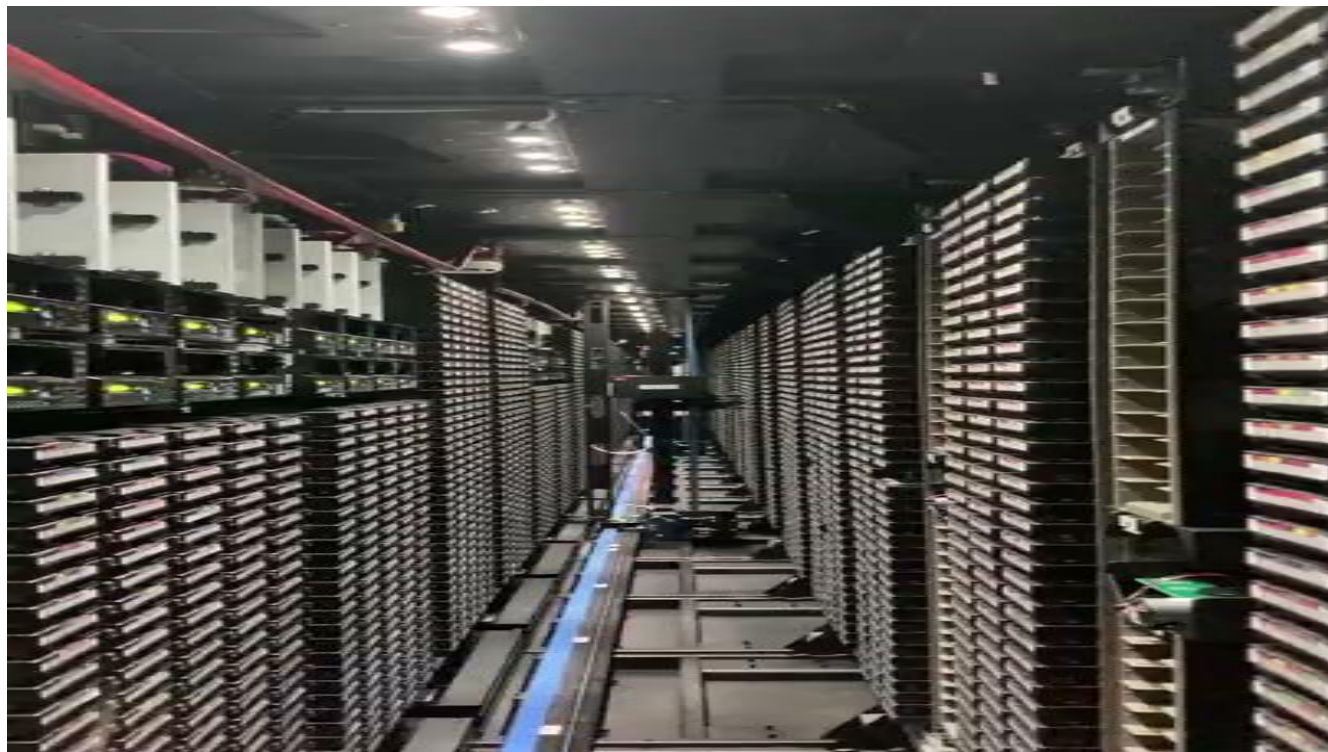
```
cd
$CFS/<projectNumber>
mkdir <name_of_choice>
```

## Snapshots:

```
cd $CFS/<projectNumber>
cd .snapshots
cp .snapshots/<date>/<file> $CFS/<projectNumber>/dir
```

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

# HPSS



<https://docs.nersc.gov/filesystems/archive/>

LAB  
the World



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science

# 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`

**How to:**  
`cd $HOME`

**Snapshots:**

`cd $HOME`

`cd .snapshots`

`cp .snapshots/<date>/<file> <where-you-want-the-file>`

# 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



# 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.

# 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

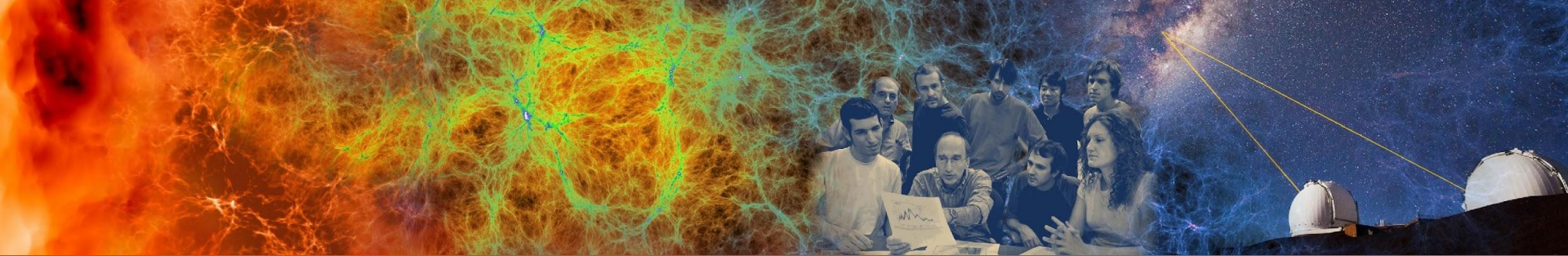
# Best Practices for DVS

- DVS is an I/O forwarder
- Uses a set of 24 nodes to forward I/O and offer high performance

# Best Practices for DVS

- DVS is an I/O forwarder
- Uses a set of 24 nodes to forward I/O and offer high performance

- 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



BERKELEY LAB



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science

# Adjusting Quotas in IRIS


Projects ▾ Reports ▾ Tools ▾
dasrepo
🔍
👤 Igerhard ▾

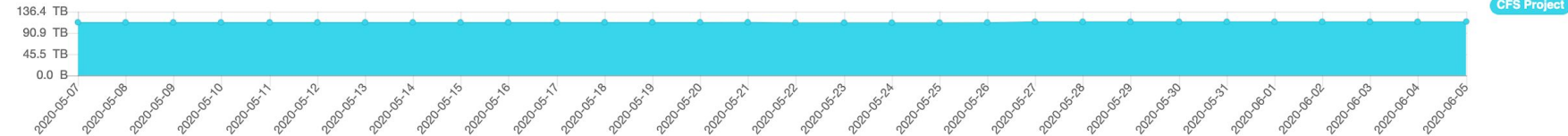
dasrepo
Compute
Jobs
Storage
Roles
Groups
Details
History

## CFS Space

**Current Quota:** 200.0 TB  
**Space Allocated:** 200.0 TB  
**Space Remaining:** 0.0 B  
**% Remaining:** -0.0%

## CFS Files

**Current Quota:** 153.0 M  
**Files Allocated:** 128.0 M  
**Files Remaining:** 25.0 M  
**% Remaining:** 16.3%



## 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	<span style="background-color: #ffc107; padding: 2px;">67.5%</span>	27 M	100.0 M	<span style="background-color: #28a745; padding: 2px;">26.6%</span>	2020-06-05
ProjectDisCo	gpfs	👤 Gerhardt, Lisa	projectd	✓	44.4 TB	91.0 TB	<span style="background-color: #28a745; padding: 2px;">48.8%</span>	134 K	20.0 M	<span style="background-color: #28a745; padding: 2px;">0.7%</span>	2020-06-05
mantissa	gpfs	👤 Prabhat, Mr	mantissa	✓	5.2 TB	10.0 TB	<span style="background-color: #ffc107; padding: 2px;">51.8%</span>	641 K	1.0 M	<span style="background-color: #ffc107; padding: 2px;">64.1%</span>	2020-06-05
das	gpfs	👤 Prabhat, Mr	das	✓	4.0 TB	5.0 TB	<span style="background-color: #dc3545; padding: 2px;">80.2%</span>	1 M	2.0 M	<span style="background-color: #ffc107; padding: 2px;">59.7%</span>	2020-06-05
ClimateNet	gpfs	👤 Gerhardt, Lisa	climaten	✓	888.3 GB	1.0 TB	<span style="background-color: #dc3545; padding: 2px;">86.7%</span>	108 K	1.0 M	<span style="background-color: #28a745; padding: 2px;">10.8%</span>	2020-06-05
datamap	gpfs	👤 Gerhardt, Lisa	datamap	✓	111.4 GB	1.0 TB	<span style="background-color: #28a745; padding: 2px;">10.9%</span>	910 K	2.0 M	<span style="background-color: #28a745; padding: 2px;">45.5%</span>	2020-06-05
gbclimat	gpfs	👤 Pseudo User, g...	gbclimat	✓	0.0 B	1.0 TB	<span style="background-color: #28a745; padding: 2px;">0.0%</span>	1	1.0 M	<span style="background-color: #28a745; padding: 2px;">0.0%</span>	2020-06-05
dastest	gpfs	👤 Pseudo User, d...	dastest	✓	0.0 B	1.0 TB	<span style="background-color: #28a745; padding: 2px;">0.0%</span>	1	1.0 M	<span style="background-color: #28a745; padding: 2px;">0.0%</span>	2020-06-05



# PI Toolbox: [my.nersc.gov/pitools/](https://my.nersc.gov/pitools/)

The screenshot shows the PI Toolbox interface with a modal dialog open. The dialog is titled "Owning group can:" and contains the following options:

- Read (r)
- Write (w)
- Execute file, enter directory
- Make directory group openable and executable files group executable (X)
- Execute binary file as member of owning group and force new items in directory to be owned by the group (s)
- Execute binary file normally, as member of user's default group (x)
- I want to apply these permissions recursively

Buttons for "Cancel" and "Submit request" are at the bottom of the dialog.

The background interface shows a file list with columns: Select, Name, Size, Date, and Permissions. The file list includes entries like .ipynb, DaskE, MODS, agrein, backu, canon, certs.nersc.gov, and dfulton.

Select	Name	Size	Date	Permissions
<input type="checkbox"/>	Parent Direct			
<input type="checkbox"/>	.ipynb	4096	Jul 18 13:14	drwxr-xr-x
<input type="checkbox"/>	DaskE	6326	Sep 12 10:20	-rw-r--r--
<input checked="" type="checkbox"/>	MODS	4096	Jan 18 15:32	drwxrwxr-x
<input type="checkbox"/>	agrein	4096	Jul 22 18:23	drwxrwxrwx
<input type="checkbox"/>	backu	4096	Nov 18 13:35	drwxrwxr-x
<input type="checkbox"/>	canon	4096	Aug 21 10:32	drwxrwx---
<input type="checkbox"/>	certs.nersc.gov	4096	Sep 9 16:03	drwxrwxr-x
<input type="checkbox"/>	dfulton	4096	Aug 21 11:14	drwxrwxr-x

# Data Dashboard in my.nersc.gov

## Data Dashboard

Showing disk space and inode usage for global directories at NERSC to which you have access as PI, PI proxy, or user (includes /cfs, /dna, and /projectb)

### atlas directory in /cfs



[Toggle Usage Details](#)

[My Files and Dirs](#)

[Browse](#)



### bbtools directory in /cfs



[Toggle Usage Details](#)

[My Files and Dirs](#)

[Browse](#)



### CAL directory in /cfs



[Toggle Usage Details](#)

[My Files and Dirs](#)

[Browse](#)



### carver directory in /cfs



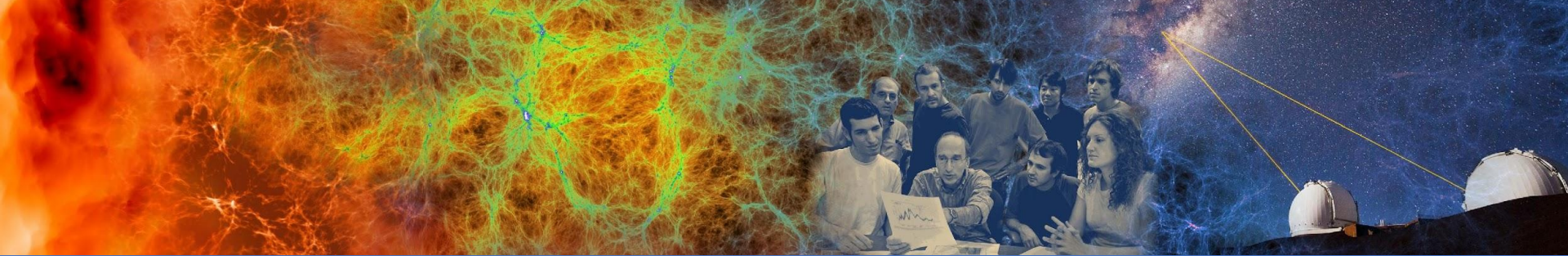
[Toggle Usage Details](#)

[My Files and Dirs](#)

[Browse](#)







# Data Sharing Best Practices



BERKELEY LAB



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science

# 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/](https://my.nersc.gov/pitools/)) can manage permissions
- **HPSS Project Directories**
  - Directories in HPSS with group writable and readable permissions

# 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/](https://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

# 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/](http://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)`

# 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/](http://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 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/](http://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 or project's PI. Useful for

- **Scratch**

- User who desire to share files
  - Only share read and write permissions with collaboration accounts
  - `chgrp -R <project>`

- **give / take**

- Mechanism to give single files to any other NERSC user

**How to:**

```
give -u <receiving_username> <file or directory>
```

```
take -u <sending_username> <filename>
```

# 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>/](https://portal.nersc.gov/project/<yourproject>)
- **Science Gateways** ([docs.nersc.gov/services/science-gateways/](https://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/](https://docs.nersc.gov/services/spin/))
- **Globus Sharing** ([docs.nersc.gov/services/globus/#globus-sharing](https://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

## How to (for small-ish files):

(logged onto the system you want to move the files to)

```
scp <username>@dtn0[1-4].nersc.gov:<path/to/file> <local_path>
```

```
scp lgupta@dtn01.nersc.gov:$HOME/script.sh .
```



# 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
- 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/>

# General Tips for Transferring Data

The **recommended** tool for moving data is

- Reliable & easy-to-use web-based interface
  - Automatic retries
  - Email notification of successful transfers
- Accessible to all NERSC users
- NERSC-managed endpoints
- Web based GUI for drag and drop
- NERSC Globus scripts for command line
- REST/API for scripted interactions
- Globus Connect Personal for

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



## Examples

Transfer files from [NERSC's Community file system](#) to [NERSC's Perlmutter Scratch file system](#)

### Tip

This can be used to stage data on Perlmutter scratch before using it in a running job. See the script `stage_data.script` (included in the globus-tools module) for an example of how to do this.

First, generate a list of files and directories you wish to transfer. If a directory is included in this list, its contents will be recursively transferred to the target directory.

```
global/cfs/cdirs/<myrepo>/<my_dataset_directory> > transfer.txt
global/cfs/cdirs/<myrepo>/<my_other_dataset_directory>/data01.dat >> transfer.txt
```

Then invoke the transfer script

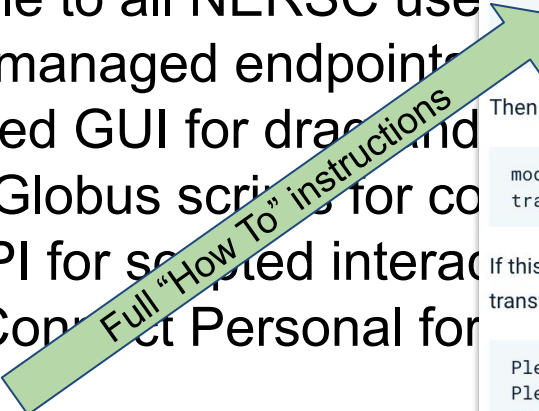
```
module load globus-tools
transfer_files.py -s dtn -t perlmutter -d /pscratch/sd/<letter>/<your_username>/in
```

If this is the first time running the script, you'll see the next two lines, followed by the standard transfer information:

```
Please go to this URL and login: https://auth.globus.org/v2/oauth2/authorize?client=
Please enter the code you get after login here: <snipped>
Transfer ID is b'XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX' label is <username>_<datestamp>
```

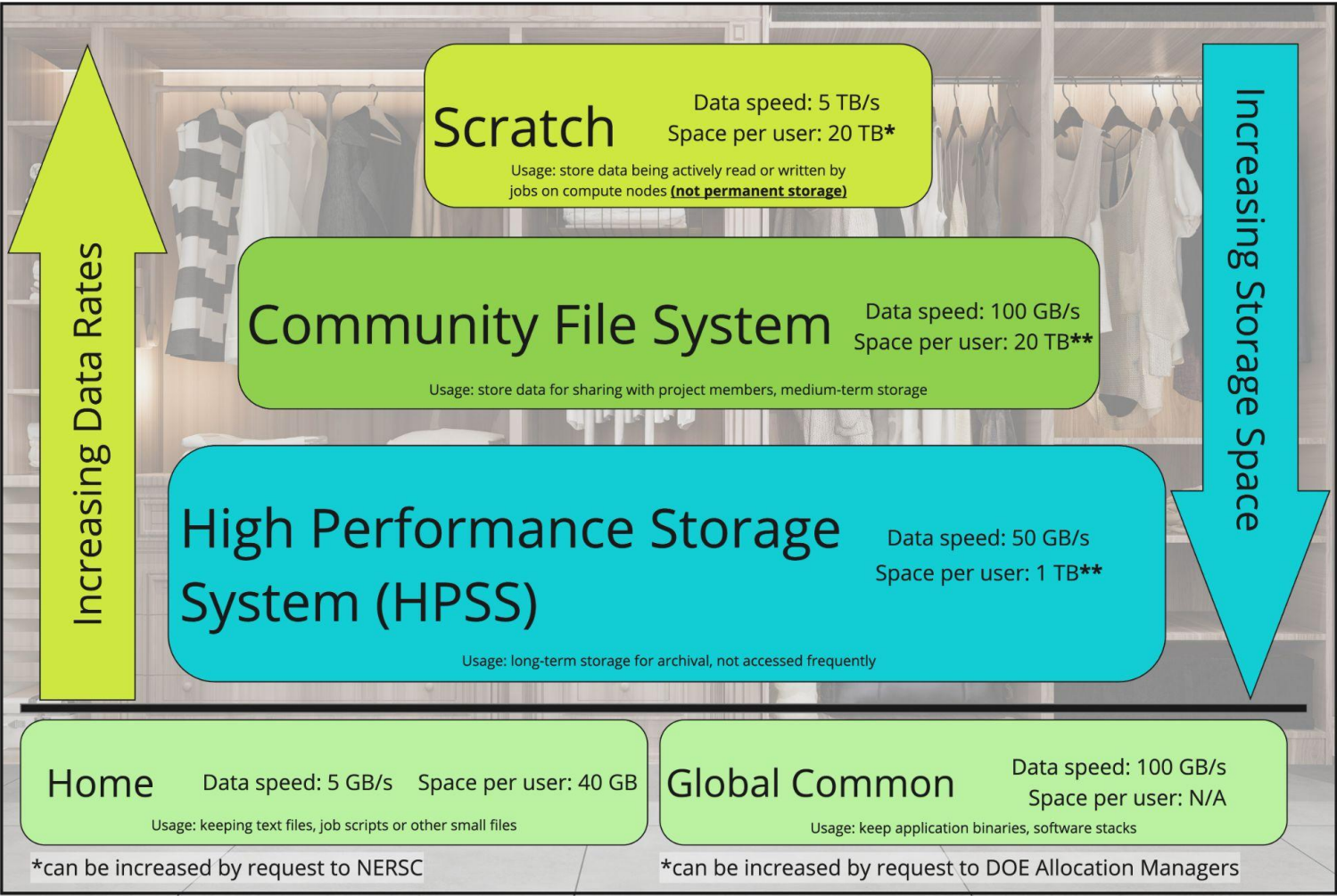
You can check on the status of the transfer with the `check_transfer.py` script

```
nersc> module load globus-tools
nersc> check_transfer.py -i XXXXXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
```



# 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](https://help.nersc.gov)



Increasing Data Rates

Increasing Storage Space

**Scratch** Data speed: 5 TB/s  
Space per user: 20 TB\*  
Usage: store data being actively read or written by jobs on compute nodes (**not permanent storage**)

**Community File System** Data speed: 100 GB/s  
Space per user: 20 TB\*\*  
Usage: store data for sharing with project members, medium-term storage

**High Performance Storage System (HPSS)** Data speed: 50 GB/s  
Space per user: 1 TB\*\*  
Usage: long-term storage for archival, not accessed frequently

**Home** Data speed: 5 GB/s Space per user: 40 GB  
Usage: keeping text files, job scripts or other small files

**Global Common** Data speed: 100 GB/s  
Space per user: N/A  
Usage: keep application binaries, software stacks

\*can be increased by request to NERSC

\*can be increased by request to DOE Allocation Managers



Thank You and  
Welcome to  
NERSC!

