Data Transfer

COMPLECS Team

https://bit.ly/COMPLECS
https://github.com/sdsc-complecs

SDSC
SAN DIEGO SUPERCOMPUTER CENTER

UC San Diego

About COMPLECS

COMPLECS (COMPrehensive Learning for end-users to Effectively utilize CyberinfraStructure) is a new training program offered by SDSC that covers the most important ***non-programming*** concepts and skills that you need to effectively use supercomputers. Topics include parallel computing concepts, Linux tools and bash scripting, security, batch and interactive computing, how to get help, and data management.

COMPLECS is supported by NSF CISE/OAC-2320934

COMPLECS:
Data Management Series

Data Transfer

How to get the data you need for your research to and from high-performance computing systems.

Data Storage and File Systems

How to use the data storage and file systems you'll find mounted on high-performance computing systems.

Parallel I/O

How to make effective use of parallel filesystems.

Data Management is a HUGE topic

Image Credit: Harvard Biomedical Data Management

Data management is a field of study in its own right. In general, we will only be able to address a limited set of the most essential data management topics as they relate to high-performance computing as part of the COMPLECS training program. If your research involves complex data management issues, then we highly encourage you to seek out the advice of experts at your institution and/or refer to additional resources that may be available online.

Data Transfer

Learning Goals:

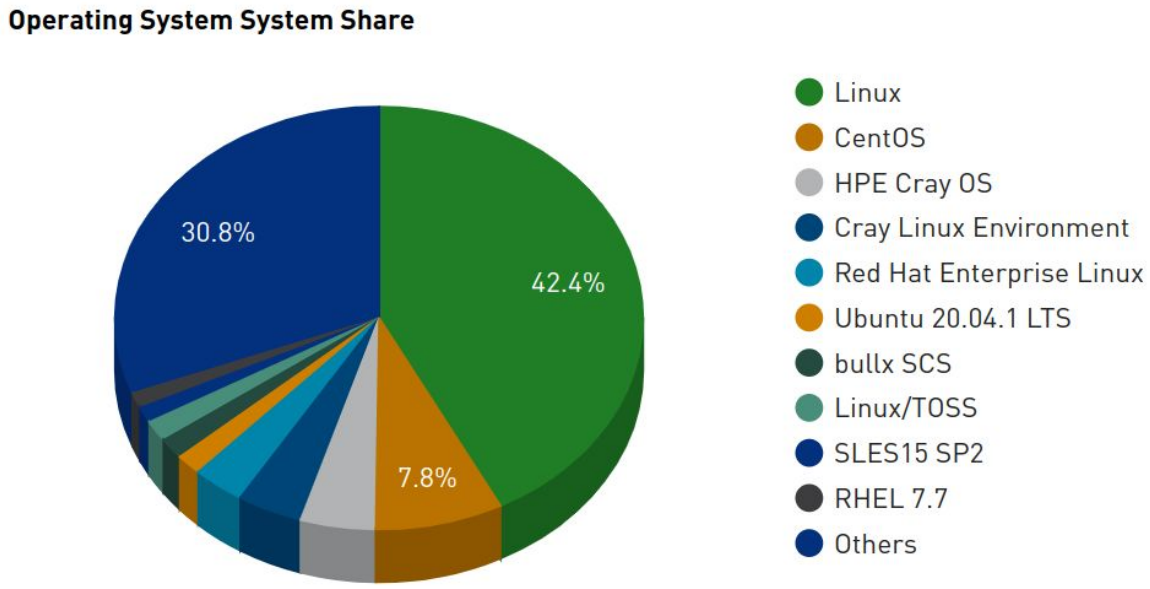
- Download data from the Internet
- Check the integrity of your data
- How to use data compression
- About data storage and file systems
- Key data transfer tools
- Data distribution networks and federations

Prerequisite Knowledge:

- Intermediate Linux and Shell Scripting

HPC runs on Linux

High-performance computing (HPC) and advanced cyberinfrastructure (CI) run on Linux. If you don't believe us, then look no further than the latest statistics from the TOP500 --- a list of the most powerful supercomputers in the world.



Therefore, throughout the COMPLECS training program, we aim to highlight the use of standard command-line tools and applications that are available for Unix-like operating systems such as Linux and macOS.

Recommendation for Windows users: Install the Windows Subsystem for Linux on your personal computer.

What is Data Transfer?

When we talk about *data transfer* in the context of scientific and high-performance computing, we typically use it as a synonym for *file transfer*. i.e., the process of transmitting a computer file from one computer system to another over a network communication channel, which is mediated by a file transfer protocol.

A *data transfer* may also refer to an individual instance of a *file transfer*.

Key components and/or characteristics in any *data transfer*:

- Source — Where is the data located?
- Format — How is the data organized?
- Size — How much data is there?
- Destination — Where is the data going?
- Method — How should we transfer the data?

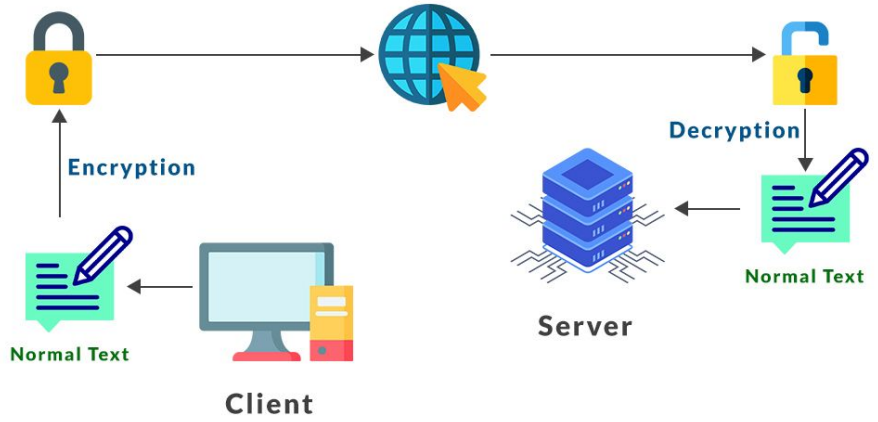


Table of Contents

wget - non-interactive network downloader

wget is a simple command-line tool for downloading files from the Web. It supports HTTP/S and FTP protocols as well as file retrieval through proxy servers. It has been designed for robustness over slow or unstable network connections. If a download fails due to a network problem, you can setup the download to keep retrying, resuming the download from where you left off until the whole file has been retrieved.

Download a file

```
$ wget https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz
```

Download multiple files

```
$ wget -i cifar-100.txt
```

```
$ cat cifar-100.txt
https://www.cs.toronto.edu/~kriz/cifar-100-python.tar.gz
https://www.cs.toronto.edu/~kriz/cifar-100-matlab.tar.gz
https://www.cs.toronto.edu/~kriz/cifar-100-binary.tar.gz
```

Resume a download

```
$ wget -c https://www.image-net.org/data/ILSVRC/2012/ILSVRC2012_img_train.tar
```

curl - transfer a URL

curl is a sophisticated command-line tool (and software library) for transferring data from or to a server using various network protocols. By default, curl invokes an HTTP/S GET request on a URL, displaying the retrieved data to standard output.

Download a file

```
$ curl -O https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz
```

Download multiple files

```
$ curl -K cifar-100.txt
```

```
$ cat cifar-100.txt
-O
url = "https://www.cs.toronto.edu/~kriz/cifar-100-python.tar.gz"
-O
url = "https://www.cs.toronto.edu/~kriz/cifar-100-matlab.tar.gz"
…
```

Resume a download

```
$ curl -C - -O https://www.image-net.org/data/ILSVRC/2012/ILSVRC2012_img_train.tar
```

Download multiple files in parallel

```
$ curl -Z -O 'https://www.cs.toronto.edu/~kriz/cifar-100-{python,matlab,binary}.tar.gz'
```

aria2c - ultra fast download utility

aria2c is a command-line tool for downloading files. It supports both HTTP/S and S/FTP as well as BitTorrent. More interestingly, it can download a file from multiple sources over different protocols in parallel and tries to utilize your maximum download bandwidth.

Download a file

```
$ aria2c https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz
```

Download multiple files

```
$ aria2c -i cifar-100.txt
```

```
$ cat cifar-100.txt
https://www.cs.toronto.edu/~kriz/cifar-100-python.tar.gz
https://www.cs.toronto.edu/~kriz/cifar-100-matlab.tar.gz
https://www.cs.toronto.edu/~kriz/cifar-100-binary.tar.gz
```

Resume a download

```
$ aria2c https://www.image-net.org/data/ILSVRC/2012/ILSVRC2012_img_train.tar
```

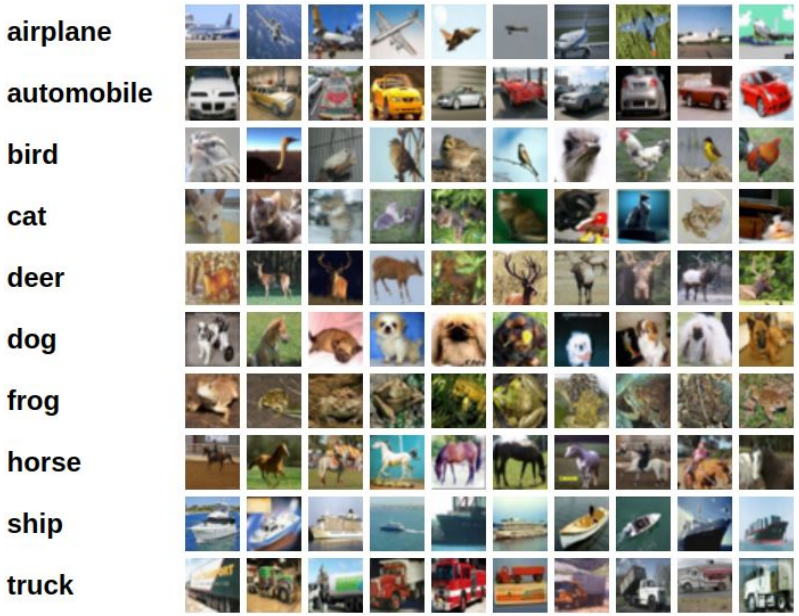
Download a file in parallel with multiple connections per host

```
$ aria2c -x2 https://www.image-net.org/data/ILSVRC/2012/ILSVRC2012_img_train.tar
```

Table of Contents

Data Integrity

Data integrity is related to the preservation of data consistency over time. It is the opposite of data corruption. For example, upon downloading a file from a given source, data integrity techniques may be used to ensure the copy of the file you have obtained from the source is the same as when it was originally created and/or recorded at the source.

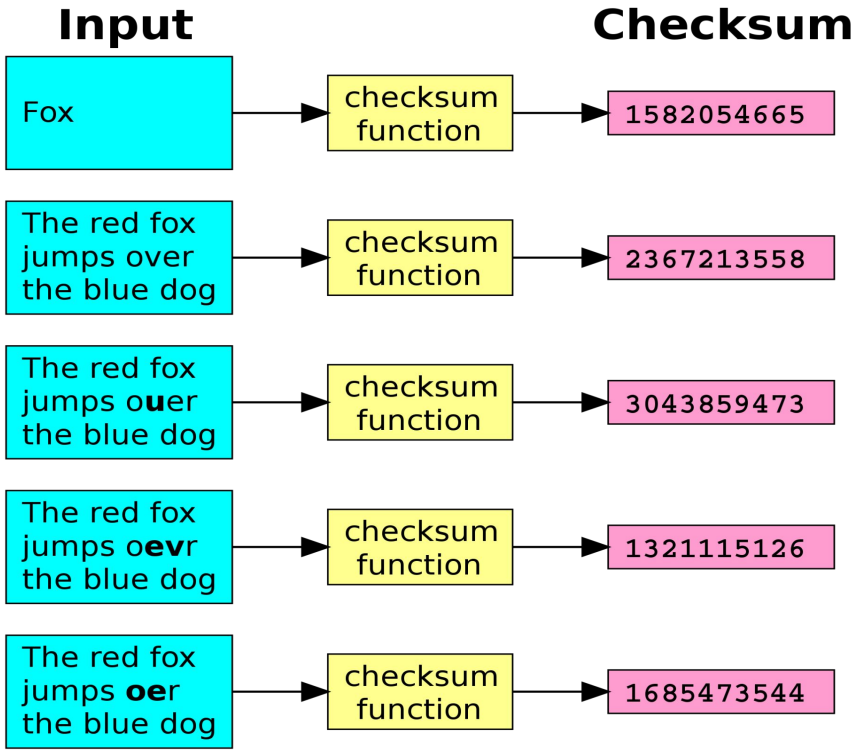


So how do you check the integrity of a file you've downloaded from the Internet and ensure it is has not been corrupted?

https://www.cs.toronto.edu/~kriz/cifar.html

Checksum

A **checksum** is a small block of data — almost like to a unique fingerprint or ID – derived from another, larger block of data, which may then be used to detect any errors introduced into the larger data block either during data transfer and/or long-term storage of the data.



Checksums are used to verify data integrity.

Checksum algorithms (or functions) are typically implemented using hash functions, which are functions that can map input data of arbitrary size to a fixed-sized output value known as a *hash value*, *digest*, or simply, *hashes*.

Checksums should be easy to compute and minimize duplicate output values for different inputs — a problem known as a hash collision.

md5sum - compute & check MD5 message digest

md5sum is a command-line tool that calculates and verifies a 128-bit MD5 hash value of a file. While you will commonly find MD5 hashes still in use, please note that the underlying MD5 algorithm is no longer deemed to be secure.

Version	Size	md5sum
CIFAR-10 python version	163 MB	c58f30108f718f92721af3b95e74349a
CIFAR-10 Matlab version	175 MB	70270af85842c9e89bb428ec9976c926
CIFAR-10 binary version (suitable for C programs)	162 MB	c32a1d4ab5d03f1284b67883e8d87530

Compute the MD5 hash of a file

```
$ md5sum cifar-10-python.tar.gz
```

Compute the MD5 hashes of multiple files and output them to a file

```
$ md5sum cifar-10-python.tar.gz cifar-10-matlab.tar.gz cifar-10-binary.tar.gz > cifar-10.md5
```

Check and verify an MD5 hash file

```
$ md5sum -c cifar-10.md5
```

* macOS has an equivalent command-line tool named md5.

sha256sum - compute & check SHA256 message digest

sha256sum is a command-line tool that calculates and verifies a 256-bit SHA256 hash value of a file. It is the recommended checksum utility you should use over md5sum.

Compute the SHA256 hash of a file

```
$ sha256sum cifar-100-python.tar.gz
```

Compute the SHA256 hashes of multiple files and output them to a file

```
$ sha256sum cifar-100-python.tar.gz \
            cifar-100-matlab.tar.gz \
            cifar-100-binary.tar.gz > cifar-100.sha256
```

Check and verify a SHA256 hash file

```
$ sha256sum -c cifar-100.sha256
```

```
$ cat cifar-100.sha256
85cd44d02ba6437773c5bbd22e183051d648de2e7d6b014e1ef29b855ba677a7  cifar-100-python.tar.gz
5fb8d16a77f63e3e417308497f2dcd58fca65b85b8c765ee02ec4e01c9f1d0d6  cifar-100-matlab.tar.gz
58a81ae192c23a4be8b1804d68e518ed807d710a4eb253b1f2a199162a40d8ec  cifar-100-binary.tar.gz
```

* macOS has an equivalent command-line tool named sha256.

Table of Contents

Data Compression

Data compression is the process of encoding information using fewer bits than used in its original representation, reducing the resources required to store or transmit data. Different compression techniques are classified either as *lossless* or *lossy*.

Lossless compression techniques reduce bits by identifying and eliminating statistically redundant information. This allows the original data to be reconstructed from the compressed data with no information loss. Examples include Lempel-Ziv (LZ).

Lossy compression techniques reduce bits by removing unnecessary or less important information. This allows lossy methods to achieve higher compression ratios. Examples include discrete cosine transform (DCT).

gzip - compress or expand files

gzip is a file format and command-line tool for file compression and decompression. It utilizes a combination of Lempel-Ziv (LZ77) and Huffman coding. Whenever possible, each file is replaced by one with the extension ***.gz**, while keeping the same ownership modes, access and modification times.

Decompress a file

```
$ gzip -d cifar-10-python.tar.gz      $ gunzip cifar-10-python.tar.gz
```

Compress a file

```
$ gzip cifar-10-python.tar
```

Decompress multiple files and keep the originals

```
$ gzip -dk cifar-10-matlab.tar.gz cifar-10-binary.tar.gz
```

Compress files with different speeds (or levels). The default is 6.

```
$ gzip -1 cifar-10-matlab.tar         $ gzip -9 cifar-10-binary.tar
$ gzip --fast cifar-10-matlab.tar     $ gzip --best cifar-10-binary.tar
```

tar - archiving utility

`tar` is a command-line tool for bundling a collection of many files into a single archive file, which helps organize and make large datasets more manageable for other tasks like data compression, storage, and transmission. It also supports a number of different compression algorithms. ***.tar** files are commonly referred to as a *tarball*.

List the contents of an archive

```
$ tar -tf cifar-10-python.tar.gz  $ tar --list --file cifar-10-python.tar.gz
```

Extract all of the files from an archive

```
$ tar -xf cifar-10-python.tar.gz  $ tar --extract --file cifar-10-python.tar.gz
```

Create a new archive file

```
$ tar -cf cifar-10-batches-py.tar cifar-10-batches-py/
```

Create a new compressed archive file with gzip

```
$ tar -czf cifar-10-batches-py.tar.gz cifar-10-batches-py/
```

Create a new compressed archive file with bzip2

```
$ tar -cjf cifar-10-batches-py.tar.bz2 cifar-10-batches-py/
```

bzip2 - block-sorting file compressor

bzip2

bzip2 is another file format and command-line tool for file compression and decompression. It utilizes the Burrows-Wheeler block sorting text compression algorithm and Huffman coding, which generally achieves better compression ratios than more conventional LZ77/LZ78-based compressors like gzip for most types of files.

Decompress a file

```
$ bzip2 -d cifar-10-batches-py.tar.bz2
```

```
$ bunzip2 cifar-10-batches-py.tar.bz2
```

Compress a file

```
$ bzip2 cifar-10-batches-py.tar
```

Decompress a file and keep the original

```
$ bzip2 -dk cifar-10-batches-py.tar.bz2
```

Check the integrity of the compressed file

```
$ bzip2 -tv cifar-10-batches-py.tar.bz2
```

pigz - compress or expand files in parallel

pigz is parallel implementation of gzip for modern multi-processor, multi-core machines. Note, however, decompression can't be parallelized. As a result, pigz uses a single thread for decompression, but will create three other threads for reading, writing, and check calculation, which can speed up decompression in some cases.

Decompress a file

```
$ pigz -d cifar-10-python.tar.gz  $ unpigz cifar-10-python.tar.gz
```

Compress a file

```
$ pigz cifar-10-python.tar
```

Decompress multiple files and keep the originals

```
$ pigz -dk cifar-10-matlab.tar.gz cifar-10-binary.tar.gz
```

Compress a file with different speeds (or levels) using a specific number of threads

```
$ pigz -1 -p2 cifar-10-matlab.tar        $ pigz -9 -p4 cifar-10-binary.tar
```

lbzip2 - parallel bzip2 utility

lbzip2 is a parallel bzip2-compatible compression utility. Unlike pigz, it can both compress and decompress files in parallel.

Decompress a file

```
$ lbzip2 -d cifar-10-batches-py.tar.bz2
```

```
$ lbunzip2 cifar-10-batches-py.tar.bz2
```

Compress a file

```
$ lbzip2 cifar-10-batches-py.tar
```

Decompress a file and keep the original

```
$ lbzip2 -dk cifar-10-batches-py.tar.bz2
```

Create an archive file and compress it using a specific number of threads

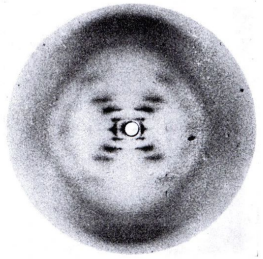
```
$ tar -I 'lbzip2 -n4' -cf cifar-10-batches-py.tar.bz2 cifar-10-batches-py/
```

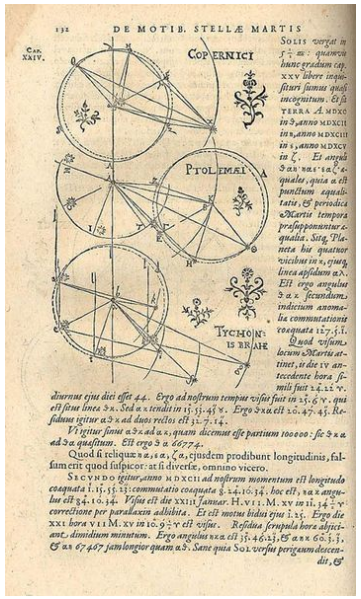
* The last major release was in 2014.

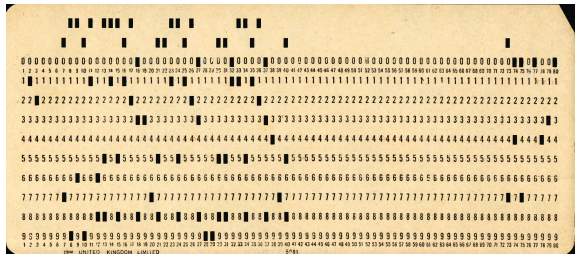
Table of Contents

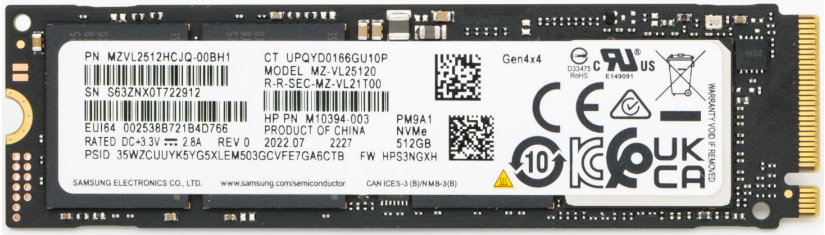
Data Storage

Data storage is the process of recording information (or data) to a *storage medium* such that it is both preserved and retrievable.



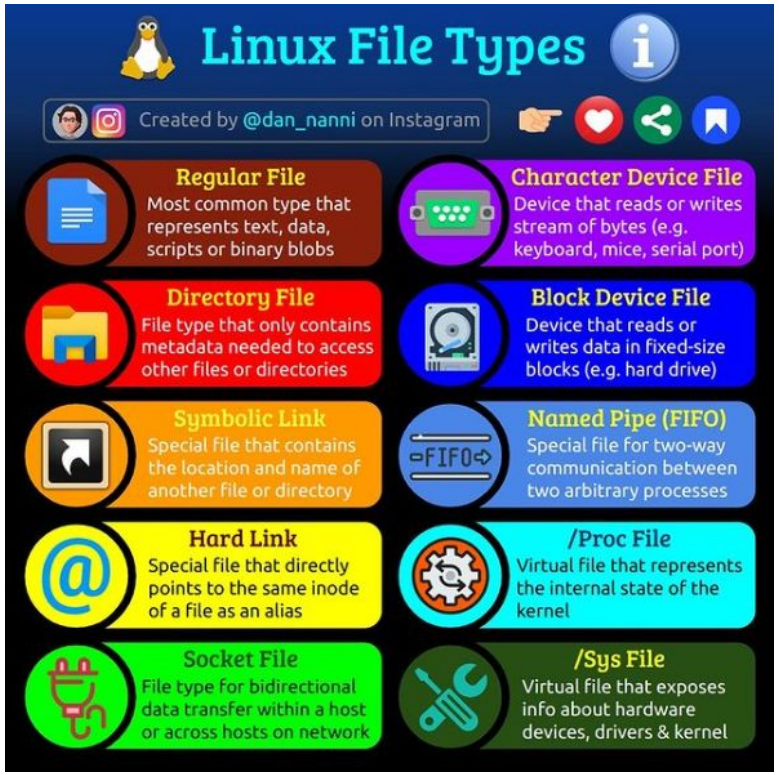






Computer File

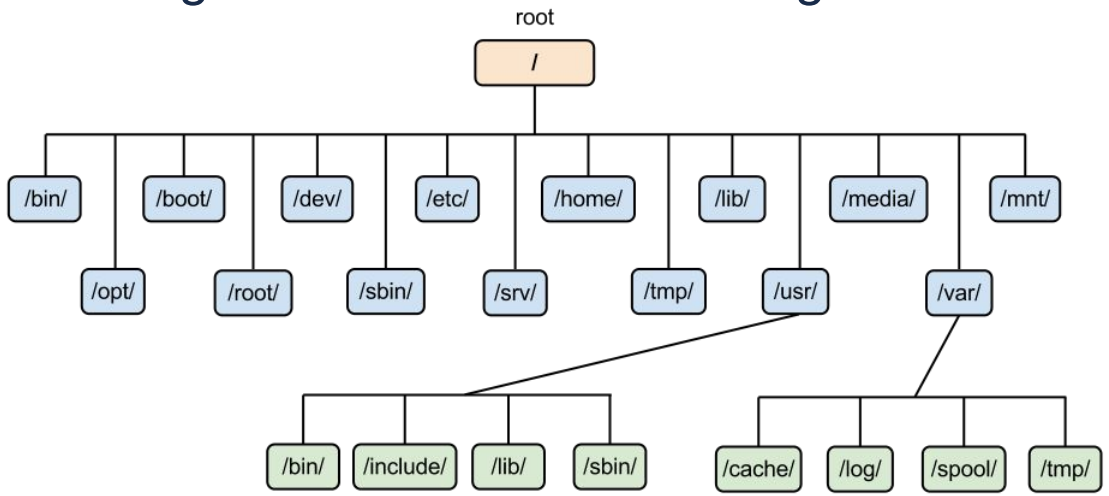
A **`computer file`** is a system resource for recording data on a storage device. Files have different specialized formats to provide encodings for the particular type of data or information that they are designed to store.



Everything is a file!

File System

A **file system** provides an operating system with the ability to create, organize, and manage access to files on storage devices.




```
Filesystem                                              Type      Size  Used Avail Use% Mounted on
devtmpfs                                                devtmpfs   63G     0   63G   0% /dev
tmpfs                                                   tmpfs      63G  9.5M   63G   1% /run
/dev/sda2                                               ext4       32G   22G  8.2G  73% /
none                                                    tmpfs      63G  149M   63G   1% /dev/shm
tmpfs                                                   tmpfs      63G     0   63G   0% /sys/fs/cgroup
/dev/sda4                                               ext4       32G  5.2G   25G  18% /tmp
/dev/sda1                                               vfat      100M     0  100M   0% /boot/efi
/dev/sdb1                                               ext4      879G   44K  834G   1% /scratch
10.22.100.113:/pool3/home                               nfs       199T   20T  180T  10% /expanse/nfs/home3
10.22.100.111:/pool1/home                               nfs       214T   19T  196T   9% /expanse/nfs/home1
10.22.100.114:/pool4/home                               nfs       211T   19T  192T  10% /expanse/nfs/home4
10.22.100.112:/pool2/home                               nfs       211T   23T  188T  11% /expanse/nfs/home2
10.21.0.21:6789,10.21.11.7:6789,10.21.11.8:6789:/       ceph      1.6T  1.1T  520G  68% /cm/shared
10.22.101.123@o2ib:10.22.101.124@o2ib:/expanse/projects lustre     11P  7.9P  3.0P  73% /expanse/lustre/projects
10.22.101.123@o2ib:10.22.101.124@o2ib:/expanse/scratch  lustre     11P  7.9P  3.0P  73% /expanse/lustre/scratch
192.168.43.5:6789,192.168.43.6:6789:/                   ceph      2.0P  148T  1.9P   8% /expanse/ceph
…
```

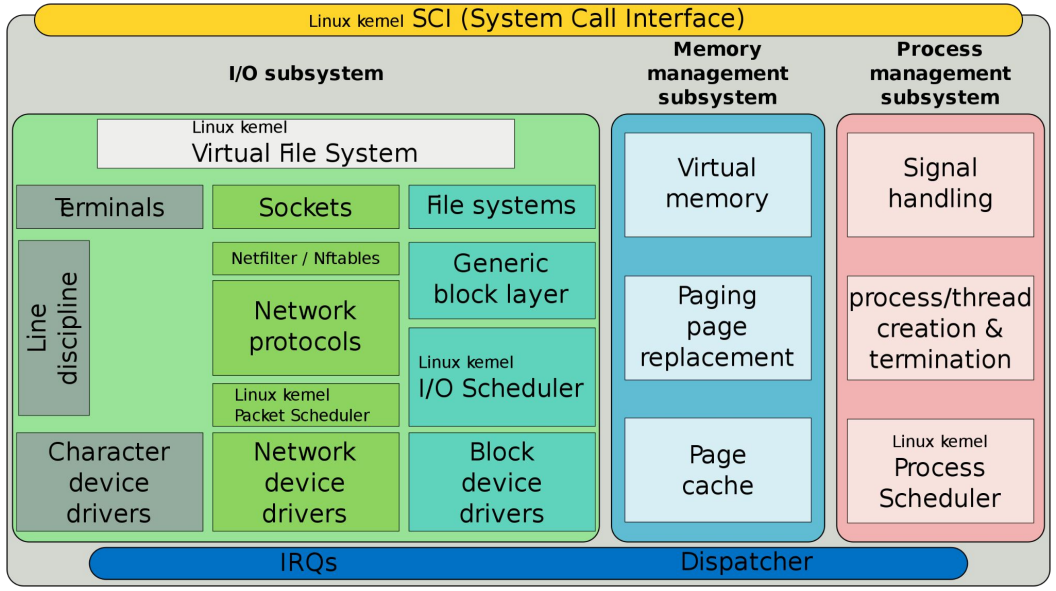
A *local* file system provides the ability to interact with files on a storage device attached to the same computer.

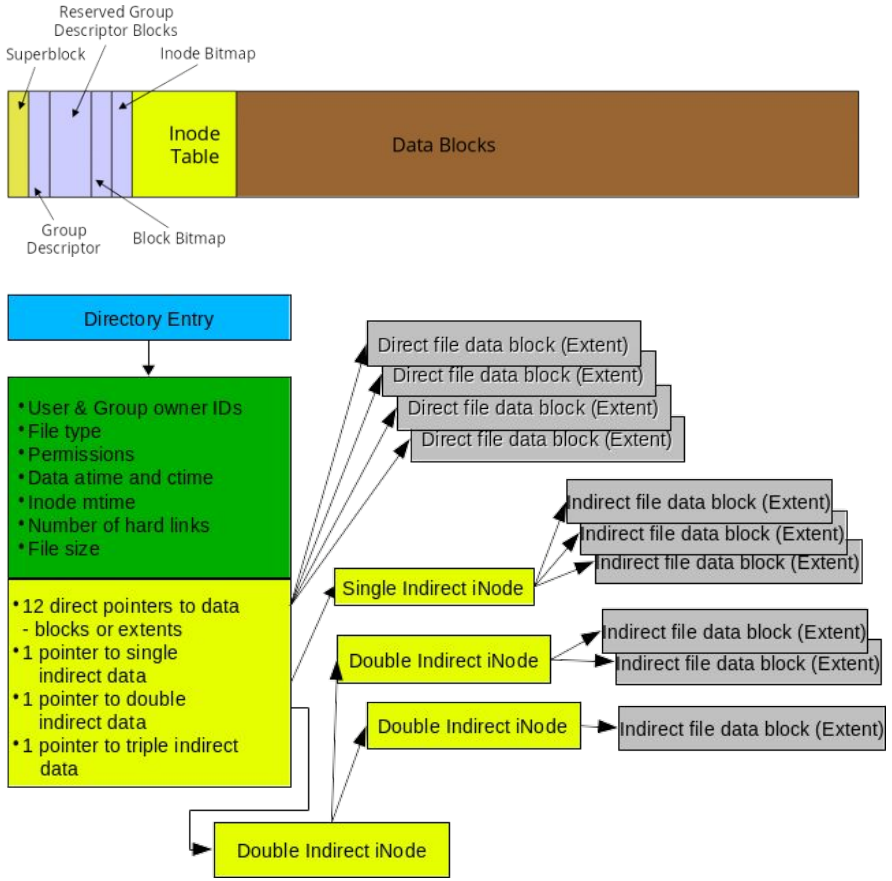
A *distributed* file system provides a service to store and share files across multiple storage devices over a network protocol. In general, however, they allow files to be accessed using the same types of interfaces and semantics as a local file system.

Every file system has its own advantages and limitations, which will determine how and where you will want to store your data.

ext4

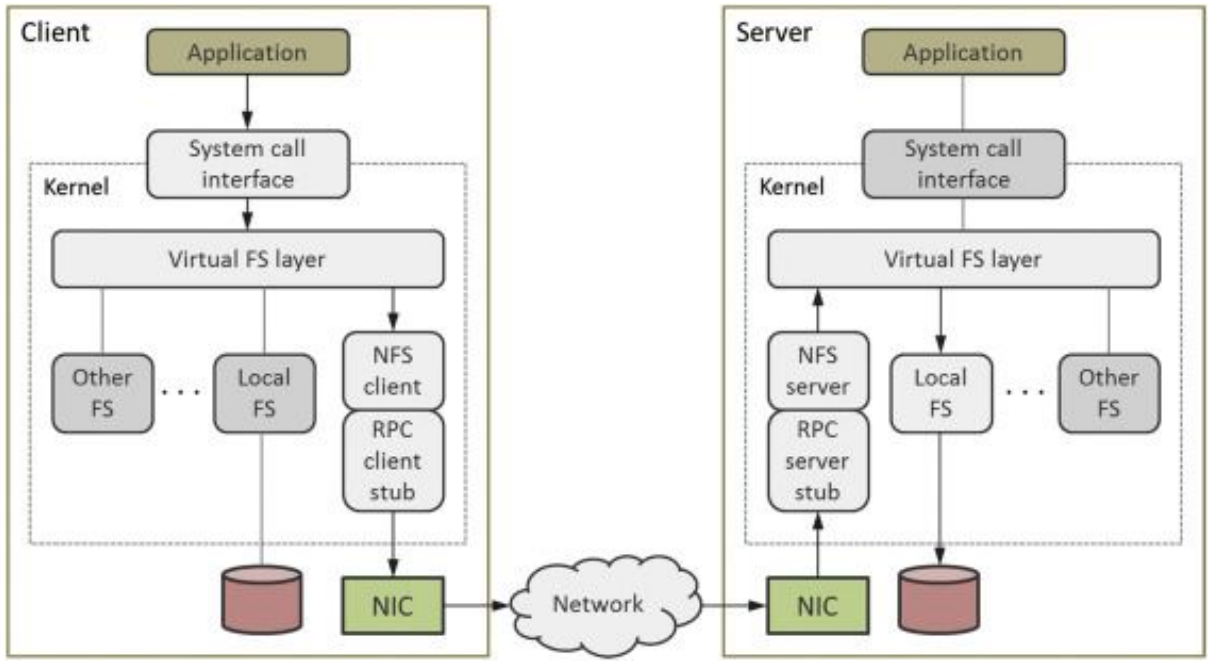
ex4 is the default local filesystem for most Linux operating system distributions. It is a journaling file system that can support large storage volumes — up to 1 exbibyte (EiB) — and large single file sizes — up to 16 tebibytes (TiB).





Network File System (NFS)

The **Network File System (NFS)** is one of the most widely used distributed file system protocols in use on high-performance computing systems. It allows a user on a client computer to access files stored on a remote server over a network.



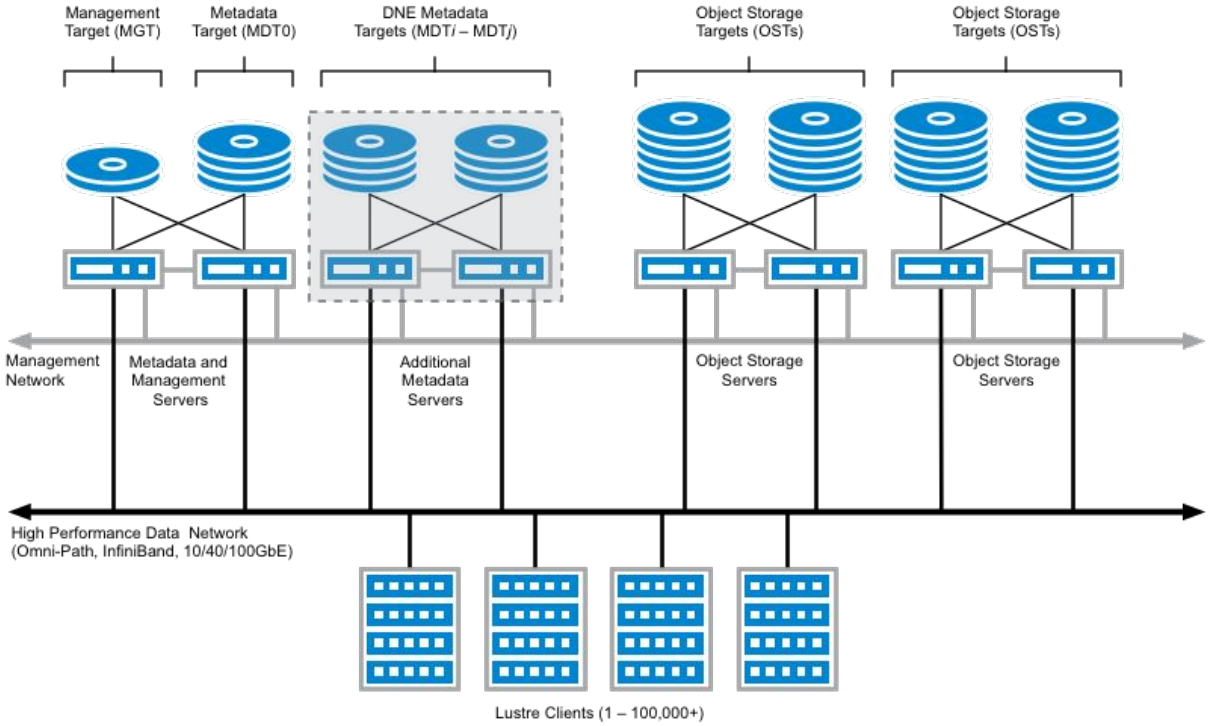
NFS is often used to store and mount your $HOME directory system-wide across all nodes of an HPC system.

In this use case, the total amount of storage is typically quite limited and intended only for software installation, batch job scripts, and essential input/output files.

An NFS will usually not be where you transfer large amounts of data

Lustre

Lustre is a parallel, distributed file system commonly used in high-performance computing. it is highly-scalable and can support up to hundreds of thousands of clients, hundreds of petabytes (PB) of storage across hundreds of storage servers, and tens of terabytes per second (TB/s) of aggregate I/O throughput.



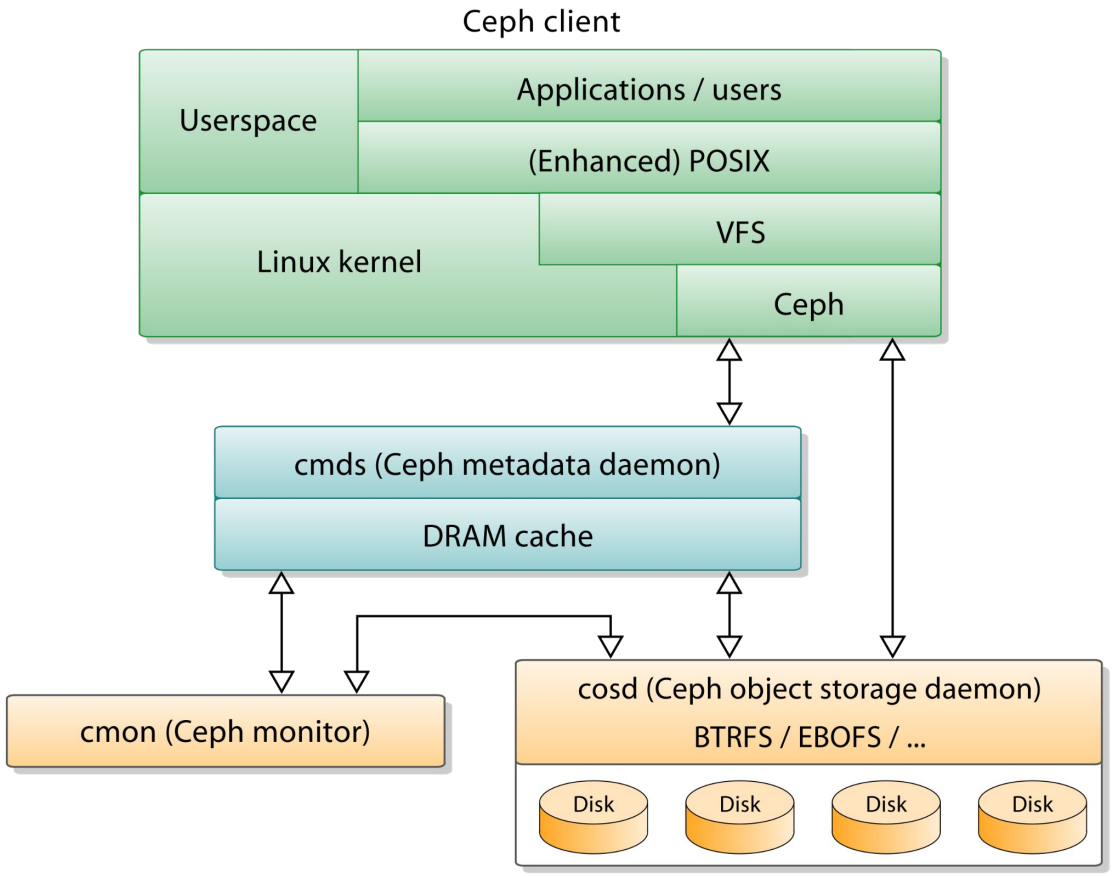
Lustre is typically used as a high-performance scratch filesystem where you can read and write large amounts of data (in parallel; e.g., MPI I/O).

Note, however, Lustre was not designed for frequent, large bursts of metadata operations. **e.g. reading or writing too many small files too quickly or too often can cause problems filesystem-wide**

Lustre may be a common target source and/or destination file system for your data transfers.

Ceph

Ceph is a distributed storage system that provides block, file, and object-based storage.



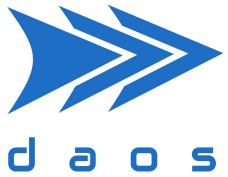
Ceph is typically used as longer-term, large-scale project storage on an HPC system.

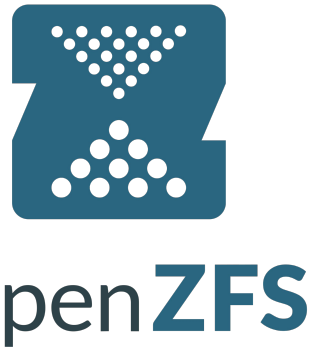
Ceph excels at data replication with fault tolerance and high-availability, which in turn helps provide strong data durability.

Ceph may be a more common target source and/or destination storage system for your data transfers in the future.

Other Data Storage and File Systems

- BeeGFS
- DAOS
- IBM Spectrum Scale (GPFS)
- OrangeFS
- Qumulo
- VAST
- WEKA
- ZFS





https://io500.org

Table of Contents

scp - secure file copy

scp is a command-line tool to securely transfer (a copy of) files between a local computer system and remote system or two remote systems (or hosts). It relies on the Secure Shell (SSH) protocol and is included by default in almost every Unix-like operating system.

Upload a file from your local system to a remote host and rename the file

```
$ scp cifar-10-python.tar.gz mkandes@oasis-dm-interactive.sdsc.edu:~/cifar-10.tar.gz
```

Download a file from a remote host to your local system

```
$ scp mkandes@oasis-dm-interactive.sdsc.edu:~/cifar-10.tar.gz ./
```

Upload a directory from your local system to a remote host

```
$ scp -r cifar-10-batches-py/ mkandes@login.expanse.sdsc.edu:~/
```

Transfer a directory from one remote host to another

```
$ scp -r login.expanse.sdsc.edu:~/cifar-10-batches-py/ bridges2.psc.edu:~/
```

Upload a directory from your local system to a remote host with *compression enabled*

```
$ scp -C -r cifar-10-batches-py/ login.expanse.sdsc.edu:~/cifar-10/
```

sftp - secure file transfer

sftp is an interactive command-line (client) tool that utilizes the SSH File Transfer Protocol (SFTP) to provide secure file access, management, and transfers between a local system and a remote server.

Open a connection to the remote system

```
$ sftp mkandes@login.expanse.sdsc.edu
Connected to login.expanse.sdsc.edu.
sftp>
```

List the contents of the pwd directory on the remote system

```
sftp> ls
```

Change directories on the remote system

```
sftp> cd cifar-10-batches-py/
```

Download a file from the remote system

```
sftp> get data_batch_1
```

Upload a file to the remote system

```
sftp> put cifar-10.txt
```

Close the connection to the remote system

```
sftp> exit
```

rsync

rsync is a command-line utility for transferring and *synchronizing* files and directories between two computer storage systems over a network. By default, it determines which files differ between the two by checking the modification time and size of the files on both ends.

Synchronize the contents of a directory with another on the same system

```
$ rsync -r cifar-10-batches-py/ cifar-10-batches-py_recursive_copy
```

Synchronize the contents of a directory in 'archive mode', which preserves almost all file properties

```
$ rsync -a cifar-10-batches-py/ cifar-10-batches-py_archive_copy
```

Synchronize the contents of a local directory with one on a remote system

```
$ rsync -a cifar-10-batches-py/ mkandes@login.expanse.sdsc.edu:~/cifar-10-batches-py
```

Synchronize the contents of a remote directory with a local version

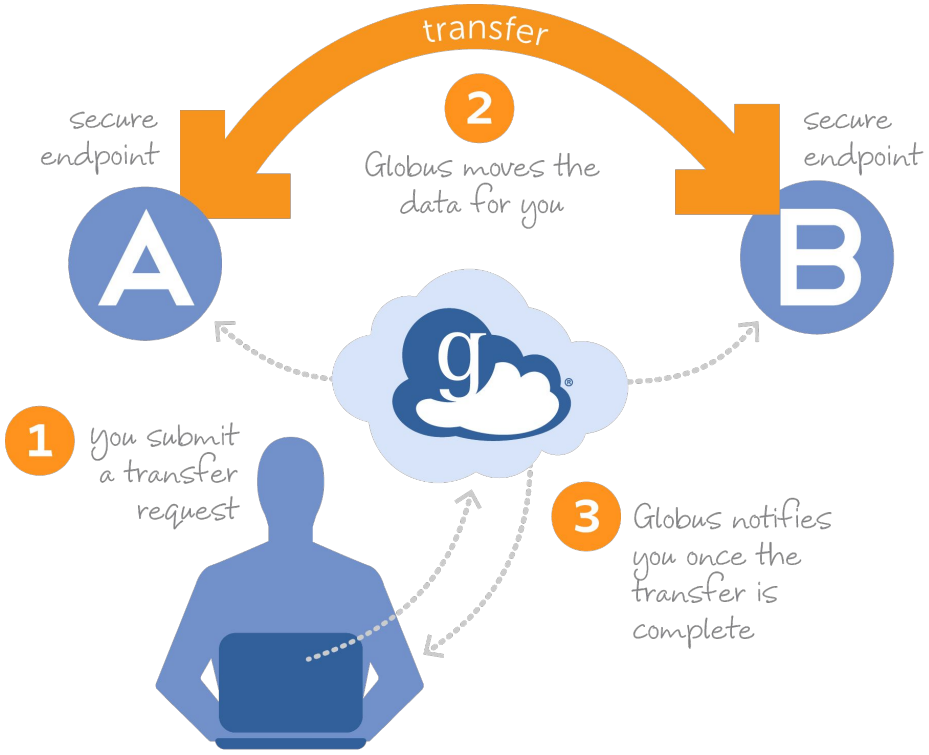
```
$ rsync -a mkandes@login.expanse.sdsc.edu:~/cifar-10-batches-py ~/cifar-10-batches-py
```

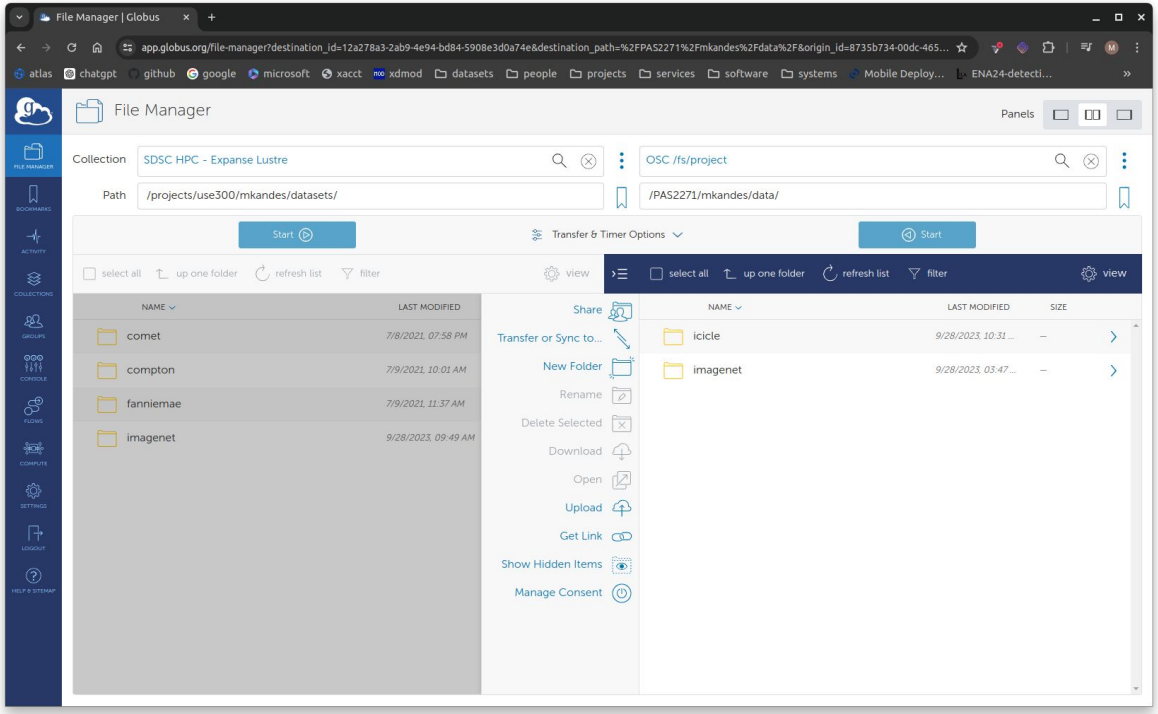
Compress data before it is sent to the destination system

```
$ rsync -az cifar-10-batches-py/ mkandes@login.expanse.sdsc.edu:~/cifar-10-batches-py
```

Globus (GridFTP)

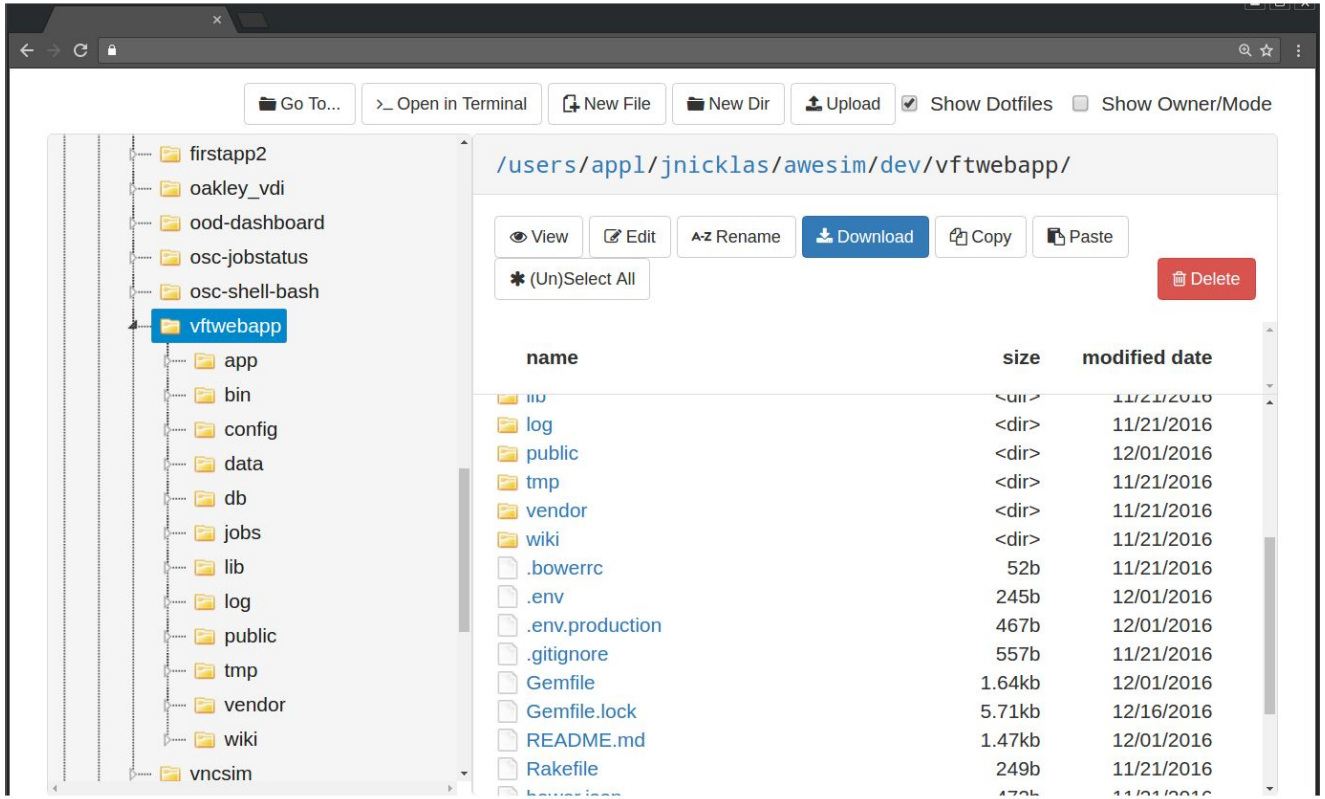
Globus is a non-profit organization run by the University of Chicago that develops and operates cyberinfrastructure products and services for research. It's most popular offering is a **GridFTP**-based data transfer service, which provides a fast, secure, and reliable way to transfer large amounts of data — up to petabytes (PBs).





Open OnDemand

Open OnDemand is a web portal for accessing supercomputers. It provides a Files App that allows a user to remotely interact with the files stored on the HPC systems file systems. This includes the ability to upload and download files via your web browser.



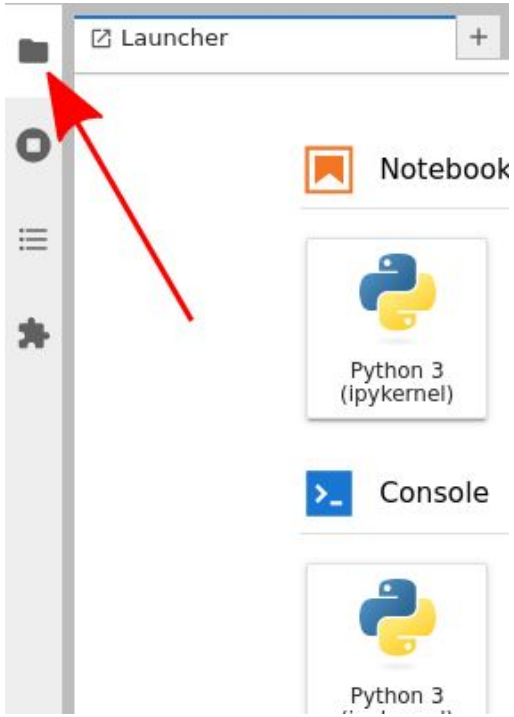
OPEN OnDemand

portal.expanse.sdsc.edu

JupyterLab

JupyterLab is the next-generation user interface for Project Jupyter, which provides a web-based interactive computing environment.

Like Open OnDemand, JupyterLab includes an integrated file browser.



Files can be uploaded to the current folder accessible from the file browser by dragging and dropping the them, or by clicking the “Upload Files” button at the top of the file browser.

Any file accessible from the current folder can be downloaded by right-clicking the file in the browser and selecting the “Download” option from the context menu.

rclone – syncs your files to cloud storage

RCLONE

`rclone` is a command-line program to manage files on cloud storage like Amazon's S3 object storage service. Note, however, it supports more than 70 different storage backends with a number of advance capabilities.

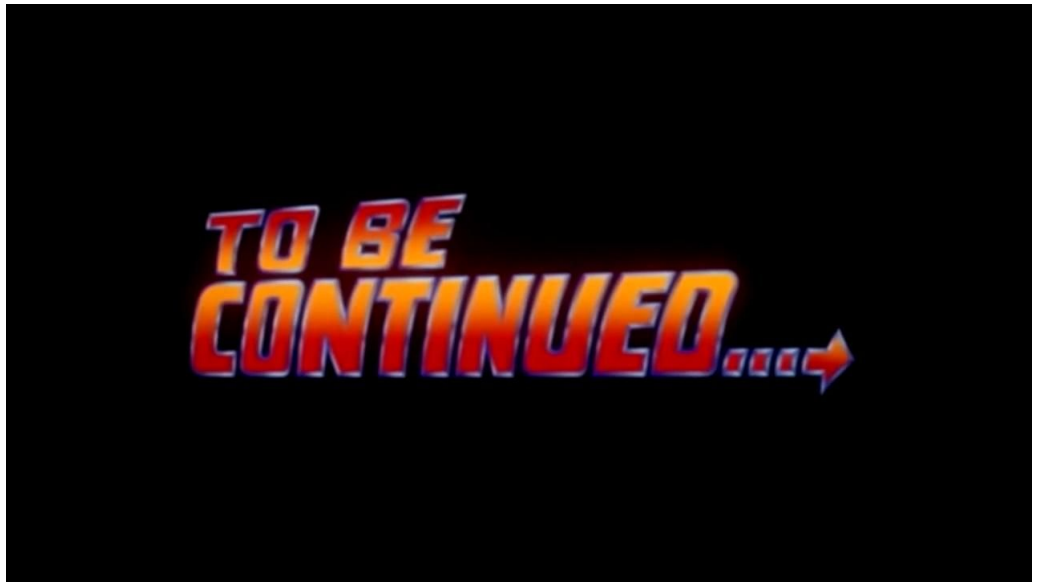


Table of Contents

Open Science Data Federation (OSDF)

The Open Science Data Federation (OSDF) is an Open Science Grid (OSG) service designed to support the sharing of files staged in autonomous data “origins” to provide efficient access to those files from anywhere in the world via a global namespace and network of data storage caches.

Open Storage Network (OSN)

The Open Storage Network (OSN) is distributed network of storage systems (known as pods) that aim to help make data storage and transfer at scale simpler for scientific research.

National Science Data Fabric (NSDF)

The National Science Data Fabric (NSDF) is a platform agnostic testbed for integrated data delivery and access to shared storage, networking, and computing for scientific discovery.

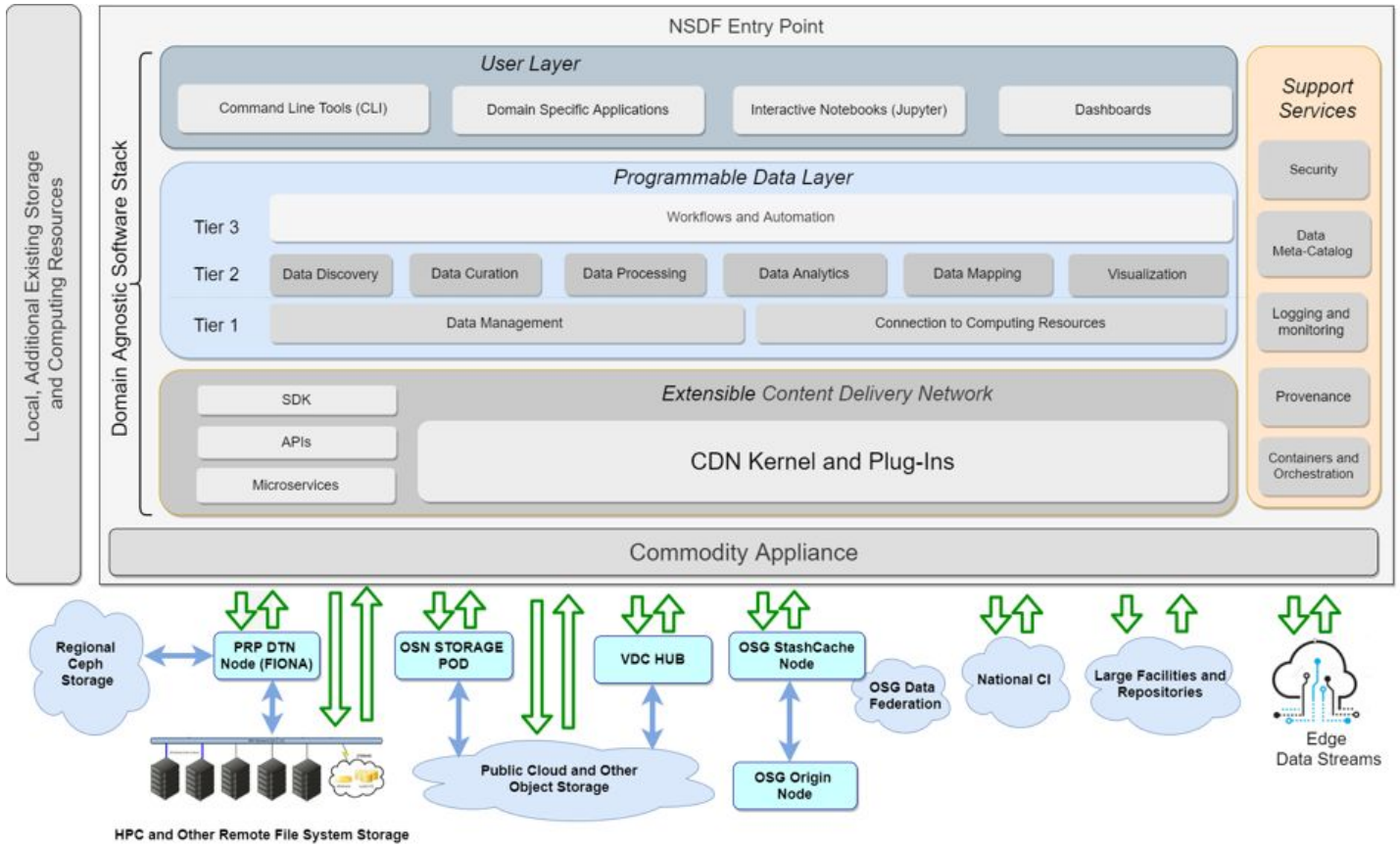


Table of Contents

Additional Information

- How To Use Wget to Download Files and Interact with REST APIs
- How to Download Files with cURL
- How To Use Rsync to Sync Local and Remote Directories

Questions & Answers

HPC System Architecture: Conceptual Model

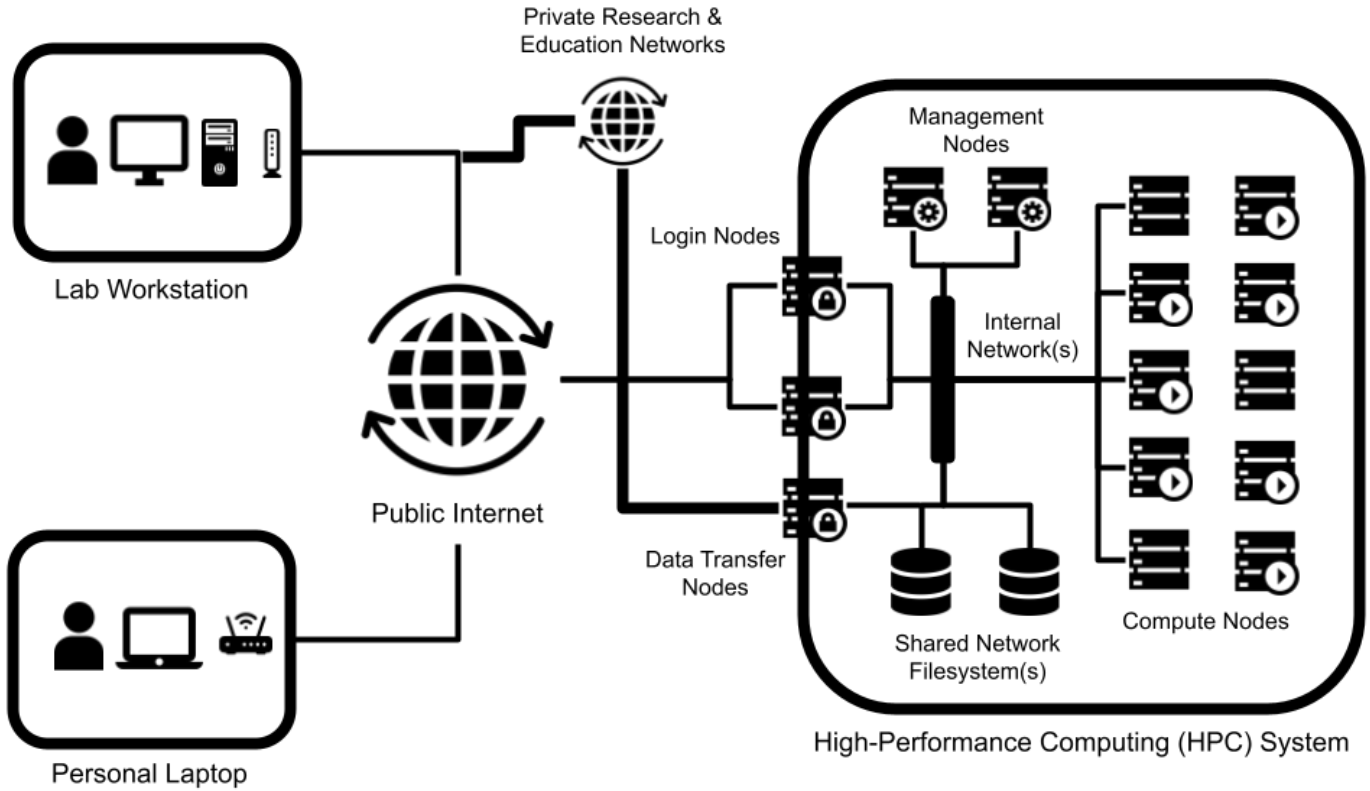
Login node(s): Provide remote access to an HPC system; use only for simple tasks such as editing files, limited data transfers to and from the system, and batch job submission

Compute nodes: Run computational workloads: simulations, data analysis and visualization

Internal Network(s): Provide high-bandwidth, low-latency communication between computes nodes ; access to shared (parallel) filesystems; system management

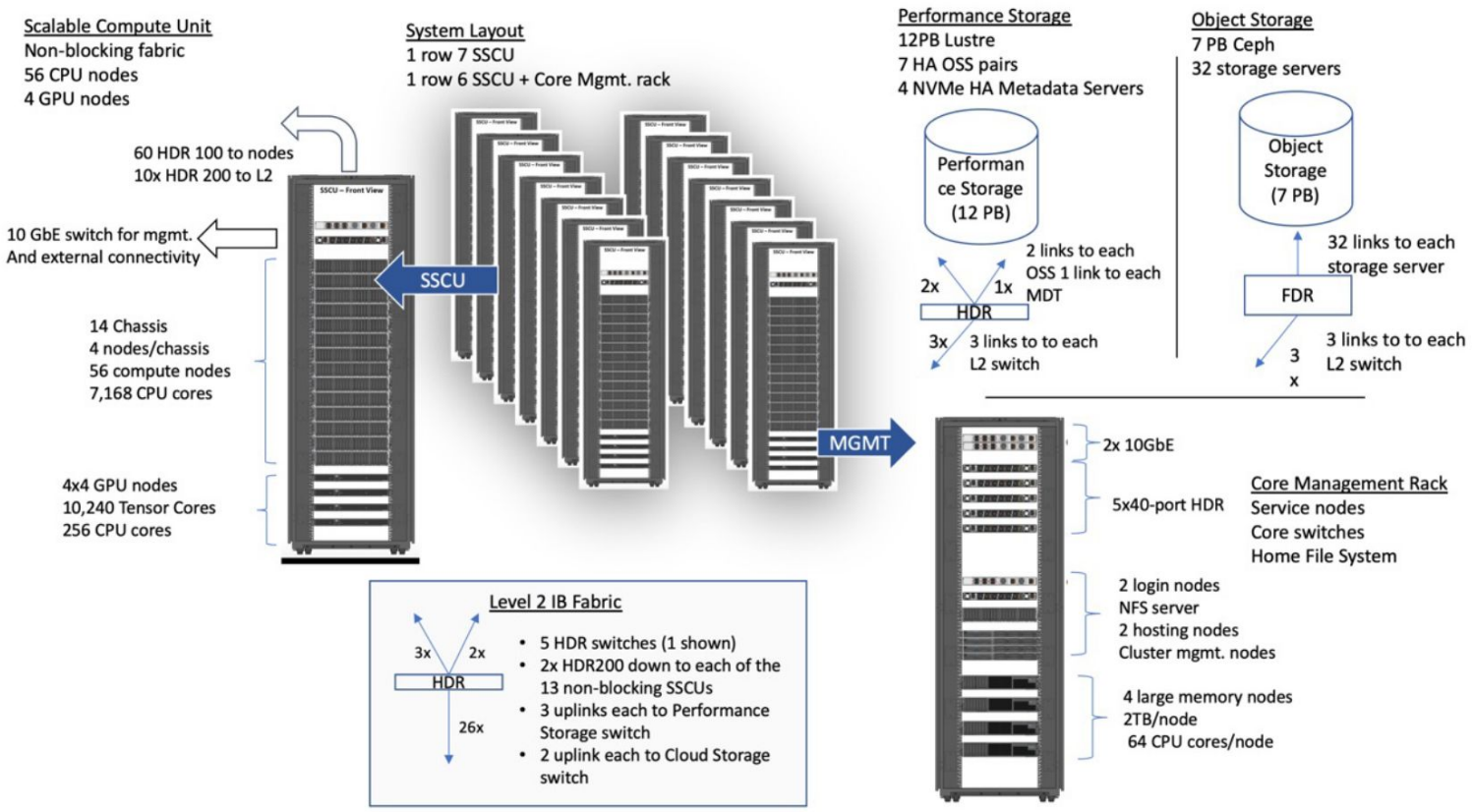
Shared Network Filesystem(s): Provide input/output (I/O) access to data storage systems from any compute node

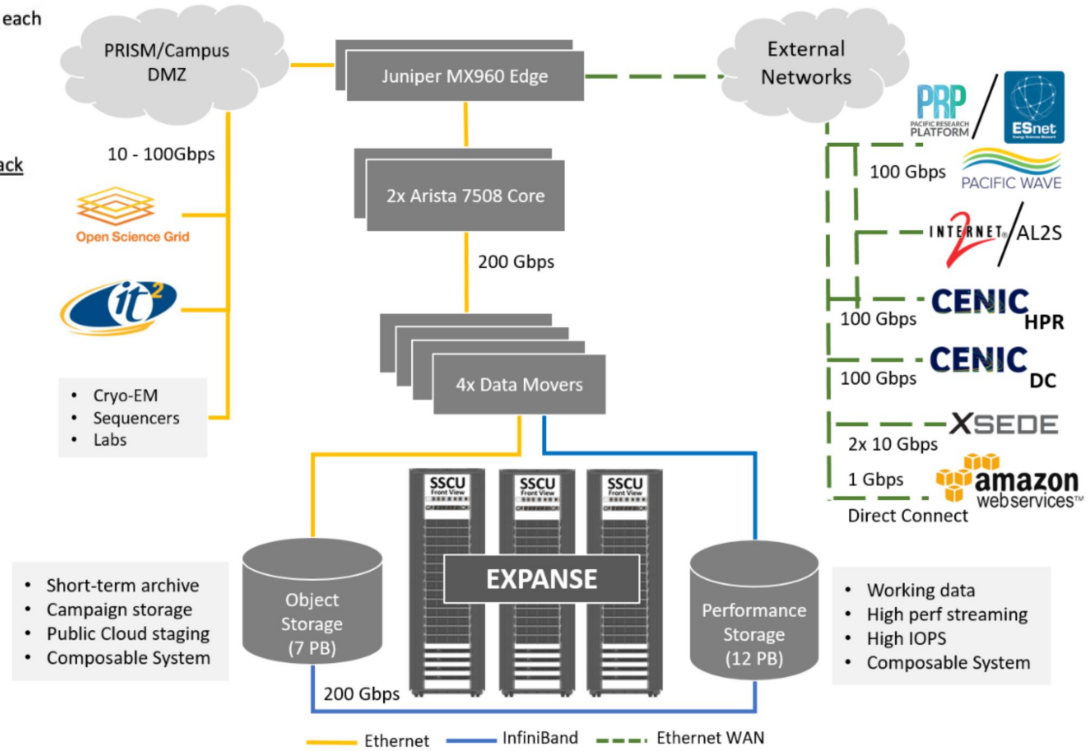
Data Transfer Node(s) : Deployed and configured specifically for transferring data over networks, usually between HPC data storage systems



Management node(s): Run core system services such as cluster management software, system monitoring software, *batch job scheduler*, etc

HPC System Architecture: Expanse @ SDSC





https://access-ci.org

https://expanse.sdsc.edu

https://www.youtube.com/watch?v=uNZyg6X_t3s

HPC System Architecture: Conceptual Model