For safe, long-term data storage, transfer your files to the Lou mass storage system, which allows you to retrieve your stored files quickly and securely whenever you need them.

The Lou mass storage system is an Intel Xeon Gold 6154 "Skylake"-based cluster combined with a Spectra Logic tape storage archive. Lou uses a parallel Data Migration Facility (DMF) system to provide high speed and bandwidth for data transfers between Lou's disks and tapes.

Connecting to Lou

Lou's four hosts, known as Lou front-end systems (LFEs), are designated as lfe[*5-8*].

You can connect automatically to the LFE that has the lowest load by using the "lou" or "lfe" hostnames in your ssh command line: ssh lfe or ssh lou. This behavior is similar to the load balancing process that occurs on the PFEs when you run ssh pfe.

Transferring Files Between /nobackup Filesystems and Lou

The Lustre (/nobackup) filesystems are mounted on the LFEs. To transfer files between your /nobackup filesystem and your Lou home filesystem, use the local file transfer commands shiftc, cp, or mcp. For example:

lfe% cp /nobackup/*your_username*/filename /u/*your_username*

You can also create a tar file that contains the data in one of your /nobackup subdirectories and then store the tar file in the LFE home filesystem. For example:

lfe% cd /nobackup/*your_username*
lfe% tar cf /u/*your_username*/*mydir*.tar *mydir*

Transferring Files from Your Local System to Lou

Recommended: Use the Shift tool (shiftc), which accepts both the alias and specific hostnames: lou, lfe, or lfe[*5-8*].

For more information about transferring files to and from your local system, see Remote File Transfer Commands.

Data Migration Between Disk and Tapes

In addition to 7.6 petabytes of disk space, Lou has 51 LTO-8 tape drives. Each LTO-8 tape holds 12 TB of uncompressed data, for a total storage capacity of approximately 1040 petabytes, or one exabyte (with normal 35% compression). Data migration (from disk to tape) and unmigration (from tape to disk) are managed by the Data Migration Facility (DMF).

Data stored on Lou's home filesystems (on disk) are automatically migrated to tape. Two copies of your data are written to tape media in silos located in separate buildings. When it is necessary to make room for more data, some files that have been written to tape may have their data "released" from disk. This means that the file is still visible on the filesystem, but the data must be retrieved from tape before the file can be used.

When a file completes writing to tape on Lou, its ctime attribute is updated to signal that it is available to be backed up by system software. This should not have an affect on any file transfer applications such as rsync.

If you need to retrieve data that is on tape, be sure to unmigrate the data from tape to your home filesystem on Lou before transferring it to other systems.

TIP: If you use the Shift tool (shiftc) for file transfers, it will automatically ensure that files on Lou are online before the transfer.

If you are not using Shift, you can use the following DMF commands to retrieve your files from tape:

```
$ dmls -al file1 file2 . . .    # show the status of your files.
$ dmget file1 file2 . . . &     # retrieve your file from tape.
```

At this point, you can start your transfer and the files will transfer as they come online.

WARNING: Do not use the /nobackup filesystems for long-term data storage. As the names suggest, these filesystems are not backed up, so any files that are removed cannot be restored. You should store essential data on more permanent storage devices, such as Lou.

For more tips on how to use the Lou storage systems more effectively, see:

- Portable File Names and Sizes
- Dealing with Slow File Retrieval

Post-Processing Your Data

To perform post-processing tasks on Lou data, use the Lou data analysis nodes, which provide dedicated PBS resources for that purpose.

Post-processing is not permitted on the LFEs. To enforce this, a system monitoring tool called query_wms runs on the LFEs and kills any user process that uses more than 1 GB of memory.

Quota Limits On Lou

There are no disk quota limits on your Lou home filesystem. However, there *are* limits on the number of files (inodes):

- 250,000 inode soft limit (14-day grace period)
- 300,000 inode hard limit

To check your quota status and usage on your home filesystem, log into Lou (lfe) and run the quota -ls command as follows:

```
lfe% quota -ls
Disk quotas for user username (uid xxxx):
     Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
/dev/cxvm/sfa2_s2n
               87422M       0       0           59381     250k    300k
```

See Quota Policy on Disk Space and Files for more information about HECC system quotas.

Storing Data in the Home Filesystems for NAS Compute Resources

All data stored in your home filesystem of any NAS compute system are automatically backed up *daily* under script control.

If you lose important data from a home filesystem and need to have it restored, please send a request to the NAS Control Room at support@nas.nasa.gov and provide the following information:

- System name
- Your username
- The directory and/or file name(s) that you wish to restore
- The date the data was last touched

Note: Disk space quota limits are imposed on the home filesystems for NAS compute resources. If your data exceeds your disk space quota limits, reduce your usage by deleting unneeded files, copying important files to Lou, or copying them to storage space at your local site and then removing them from your home filesystem to NAS compute resources.

Storing Data in NAS /nobackup Filesystems

Data in any /nobackup filesystem of any NAS system are *not* backed up by NAS. You are responsible for performing your own backups.

Note: Disk space and inode quota limits are imposed on the Lustre filesystems (/nobackup*X*). If your data exceeds the soft quota limits, reduce your usage by removing some files and/or archiving any important data on Lou or on the storage space at your local site.

In addition, when the overall usage of a /nobackup filesystem is near its capacity, files can be deleted by system administrators. Normally, the few top users are sent emails and asked to clean up their disk space.

Using Shift to Copy Files to Lou

The NAS /nobackup filesystems (for example, /nobackup[*17-19, 27-29*]) are mounted on Lou. As a result, you can log into Lou and copy the files from your /nobackup filesystem to your home directory on Lou.

We recommend using the Shift tool (shiftc), which will automatically utilize the highest performing local file transport. You can also use shiftc if you need to initiate the transfer from a NAS compute front end, such as a PFE. If you need to encrypt the data, even within the NAS enclave, then use the shiftc --encrypt option.

The Data Migration Facility (DMF) keeps disk space on the Lou mass storage systems available to users by moving unused files to storage on tape. Use the DMF commands to list, put, find, and get files on tape, with options for controlling the file transfers.

At the NAS facility, the Lou filesystems support a virtual storage manager feature called the Data Migration Facility (DMF). Its purpose is to allow users to keep an increased volume of data in the files under their home directories by migrating those files that are not currently in use to tape storage, thereby allowing active disk space to be available for active files.

Migrated files are retrieved to active disk when you attempt to read or write to them, and the whole process is substantially transparent, aside from a possible time lag while a file is being retrieved. Using the NAS-developed tool, Shift, you can pre-transfer files from tape to avoid the time lag during a job.

The process of migrating files by backing them up from disk to another medium is implemented using Spectra Logic tape libraries.

DMF Commands

User commands associated with DMF are below, followed by usage examples:

dmls	Directory listing showing file migration status
dmget	Retrieve migrated files to disk
dmput	Cause files to migrate to backup on tape
dmfind	Find files under a directory hierarchy
dmcopy	Copy all or part of offline files
dmattr	List attributes of files

The dmls Command

The dmls command is much like the standard ls command for file or directory listings, with the addition of an option for displaying the DMF status of files. Actually, dmls is derived from the GNU ls command, so it has a few extra "bells and whistles" compared to the standard ls command; for details, see the **dmls man page**.

Example showing the extra status field:

```
% ls -l
total 64
-rw-r-----   1 aeneuman madmag     14713 Mar  1 17:02 file1
-rw-r-----   1 aeneuman madmag     17564 Mar  1 17:02 file2

% dmls -l
total 33
-rw-r-----  1 aeneuman madmag     14713 Mar  1 17:02 (REG) file1
-rw-r-----  1 aeneuman madmag     17564 Mar  1 17:02 (REG) file2
```

DMF has several possible states for files. The first three shown below are the most likely to appear:

REG	Regular. The file exists only online, on active disk.
OFL	Offline. The file's directory entry remains on disk, but its data blocks are located offline only (on tape).
DUL	Dual-state. Identical copies of the file exist online (on disk) and offline (on tape). The online copy will persist if there is no demand for free space in its filesystem. When free space is needed, the online copy of the file is removed, leaving just the offline copy; the file becomes "offline." If you make any change to a dual-state file, the offline copy becomes out of date and invalid, and the file is once again a "regular" file.
MIG	Migrating. The file is in process of migrating from disk to tape.
UNM	Unmigrating. The file has been recalled and is in process of moving back from tape to disk.
NMG	Nonmigratable. The file cannot be migrated.
INV	Invalid. DMF cannot determine the file's state. The most likely reason is that it is in a filesystem that does not use DMF.

The dmget Command

The dmget command explicitly requests an offline file to be retrieved to disk. This command is not strictly required in order to retrieve offline files; any program that tries to use the file will cause it to be retrieved first.

The following example shows that referencing an offline file retrieves it after a pause. Notice that file C was "offlined." Using the command file C caused this file to be retrieved and its status changed from OFL to DUL.

```
% dmls -l
total 1404
-rw-r-----  1 aeneuman madmag      20155 Mar  2 11:24 (REG) A
-rw-r-----  1 aeneuman madmag     201550 Mar  2 11:24 (REG) B
-rw-r-----  1 aeneuman madmag    1209300 Mar  3 11:20 (OFL) C

% file B
[The response is immediate, because the file is already online:]
B:              English text
```

```
% file C
[There will be a short delay before the response, while the file is retrieved from lou:]
C:              English text

% dmls -l
total 1404
-rw-r-----   1 aeneuman madmag     20155 Mar  2 11:24 (REG) A
-rw-r-----   1 aeneuman madmag    201550 Mar  2 11:24 (REG) B
-rw-r-----   1 aeneuman madmag   1209300 Mar  3 11:20 (DUL) C
```

You can retrieve the file explicitly with the command dmget. For example:

```
% dmls -l
total 220
-rw-r-----   1 aeneuman madmag     20155 Mar  2 11:24 (REG) A
-rw-r-----   1 aeneuman madmag    201550 Mar  2 11:24 (REG) B
-rw-r-----   1 aeneuman madmag   1209300 Mar  3 11:20 (OFL) C

% dmget C ; echo Done
[There will be a short delay before the response, while the file is retrieved from lou:]
Done
```

One motivation for retrieving files explicitly with dmget is that you may be able to save time. If you are working with several files that have been migrated at the same time and reside on the same offline tape cartridge, a dmget command that names all the files would be able to retrieve them all in a single tape mount.

By contrast, simply using the files one after another would cause the tape robot to fetch the tape cartridge, retrieve the first file and put away the tape, then go back and fetch the cartridge, retrieve the second file, put away the tape, and so on.

Another good reason for retrieving files explicitly with dmget is that you control when the retrieval takes place. For example, suppose you have a large, multiprocessor application that is going to read a currently migrated data file. You would prefer the retrieval process to take place with just your shell running, not when your main application has spawned several dozen processes on several dozen nodes, with all of them having to sit idle waiting for the file to be retrieved.

See the dmfind example for an efficient method to recall all files in a directory and its subdirectories.

The dmput Command

In filesystems that are under DMF automated space management, large files that have not been used for awhile will be migrated offline automatically. The definitions of "large" and "awhile" are established on each system by its system administrator.

You can invoke migration explicitly with the dmput command. Useful options are:

-r
: Causes DMF to release a file's online disk space immediately, giving it Offline status. Otherwise, the file would be in Dual status until its disk space was needed for other files.

-w
: Causes the dmput command to wait until migration has completed before returning control. Otherwise the dmput command returns immediately with the file in Migrating status.

Here's an example showing immediate return and changing state:

```
% dmls -l
total 1404
-rw-r-----   1 aeneuman madmag     20155 Mar  2 11:24 (REG) A
-rw-r-----   1 aeneuman madmag    201550 Mar  2 11:24 (REG) B
-rw-r-----   1 aeneuman madmag   1209300 Mar  3 12:43 (REG) C

% dmput C ; dmls -l
[The immediate response shows that the file is migrating:]
total 1404
-rw-r-----   1 aeneuman madmag     20155 Mar  2 11:24 (REG) A
-rw-r-----   1 aeneuman madmag    201550 Mar  2 11:24 (REG) B
-rw-r-----   1 aeneuman madmag   1209300 Mar  3 12:43 (MIG) C

[After a delay, the migration to lou is complete:]

% dmls -l
total 1404
-rw-r-----   1 aeneuman madmag     20155 Mar  2 11:24 (REG) A
-rw-r-----   1 aeneuman madmag    201550 Mar  2 11:24 (REG) B
-rw-r-----   1 aeneuman madmag   1209300 Mar  3 12:43 (DUL) C
```

This example shows delayed return, and release of disk space:

```
% dmls -l
total 1404
-rw-r-----   1 aeneuman madmag     20155 Mar  2 11:24 (REG) A
-rw-r-----   1 aeneuman madmag    201550 Mar  2 11:24 (REG) B
-rw-r-----   1 aeneuman madmag   1209300 Mar  3 15:17 (REG) C
```

```
% dmput -r -w C ; dmls -l
[There will be a delay in response, while the file is migrated to lou:]
total 220
-rw-r-----   1 aeneuman madmag      20155 Mar  2 11:24 (REG) A
-rw-r-----   1 aeneuman madmag     201550 Mar  2 11:24 (REG) B
-rw-r-----   1 aeneuman madmag    1209300 Mar  3 15:17 (OFL) C
```

The dmfind Command

The dmfind command is based on the GNU version of the find command and adds the ability to search for files in a given DMF state. This is handy for determining which files are offline and, therefore, candidates for retrieval with dmget.

Example:

```
% dmls -l
total 220
-rw-r-----   1 aeneuman madmag      20155 Mar  2 11:24 (REG) A
-rw-r-----   1 aeneuman madmag     201550 Mar  2 11:24 (REG) B
-rw-r-----   1 aeneuman madmag    1209300 Mar  3 15:17 (OFL) C

% dmfind . -state ofl  -print
./C

% dmfind . -state ofl  -print | dmget

[After a short delay, the retrieval from lou is completed:]

% dmls -l
total 1404
-rw-r-----   1 aeneuman madmag      20155 Mar  2 11:24 (REG) A
-rw-r-----   1 aeneuman madmag     201550 Mar  2 11:24 (REG) B
-rw-r-----   1 aeneuman madmag    1209300 Mar  3 15:17 (DUL) C
```

To efficiently recall all files in a directory named *mydir* and its subdirectories, use the following command:

```
% dmfind mydir -state mig -or -state ofl -print | dmget
```

The dmcopy Command

The dmcopy command copies all or part of an offline file to a destination file, keeping the file offline.

When the offline file is being copied in its entirety, the only arguments needed are the two filenames.

Example:

```
% dmls -l
total 224
-rw-r-----   1 aeneuman madmag      20155 Mar  2 11:24 (REG) A
-rw-r-----   1 aeneuman madmag     201550 Mar  2 11:24 (REG) B
-rw-r-----   1 aeneuman madmag    1209300 Mar  3 15:17 (OFL) C

% dmcopy  C  newC

[After a short delay, the copy is completed:]

% dmls -l
total 1404
-rw-r-----   1 aeneuman madmag      20155 Mar  2 11:24 (REG) A
-rw-r-----   1 aeneuman madmag     201550 Mar  2 11:24 (REG) B
-rw-r-----   1 aeneuman madmag    1209300 Mar  3 15:17 (OFL) C
-rw-------   1 aeneuman madmag    1209300 Mar  4 15:06 (REG) newC
```

The dmcopy command has options that allow copying just part of the offline file, and specifying offset locations in the source and destination files.

-l *length*
: Specifies a data length to copy in bytes. The default is the whole size of the source file.

-o *source-offset*
: Specifies a byte offset in the offline source file where reading is to begin. The default is zero; that is, reading begins at the start of the source file.

-d *destination-offset*
: Specifies a byte offset in the destination file where writing is to begin. Any bytes in the destination file before this offset are zero filled. The default destination offset is whatever the source offset is.

Example: Skip two records of the offline source file and copy the next three records. Records are 2,048 bytes long.

```
% set RECDLEN=2048
% set NRECD=3
% @ LENGTH = $NRECD * $RECDLEN
```

```
% set NSKIP=2
% @ OFFSET = $NSKIP * $RECDLEN

% dmcopy  -l $LENGTH  -o $OFFSET  -d 0  C  newC

[After a short delay, the copy is completed:]

% dmls -l *C
-rw-r-----   1 aeneuman madmag    1209300 Mar  3 15:17 (OFL) C
-rw-------   1 aeneuman madmag       6144 Mar  4 16:10 (REG) newC
```

Note: If the source file is a regular file, without an offline copy in DMF, the destination file is always zero length. The modes on the source file are ignored when creating the destination file. The modes on the destination file are 0666 (readable and writable by everyone), except where blocked by the current umask. Review your output files' permissions and reset them with the command chmod, as needed.

The dmattr Command

The dmattr command prints selected attributes of specified files. This is most useful in shell scripts.

Some options include:

-a *attr1*,*attr2*,...
: Selects a subset of the reportable attributes.

-d *delimiter*
: Specifies a one-character separator between adjacent values. A space is the default.

-l
: Prints the attributes, one per line, with labels.

Examples for a file named "A"

```
% dmattr -l A
      bfid : 0
     emask : 0
   fhandle : 01000000000000188dede3a4b319c5b9000e000000000001000000000b3fd163
     flags : 0
     nregn : 0
     owner : 4771
      path : A
      size : 20155
     space : 20480
     state : REG

% dmattr A
0 0 01000000000000188dede3a4b319c5b9000e000000000001000000000b3fd163 0 0
4771 A 20155 20480 REG

% dmattr  -a owner,state  -d :  A
4771:REG

% foreach file ( * )
? if (`dmattr -a state $file` == OFL) then
? echo $file is offline
? endif
? end

C is offline
```

Portable File Names

Use portable file names. A name is portable if it contains only ASCII letters and digits, `.', `_', and `-'. Do not use spaces or wildcard characters or start with a `-' or `//', or contain `/-'. Avoid deep directory nesting.

If you intend to have tar archives to be read under MSDOS, you should not rely on case distinction for file names, and you might use the GNU doschk program for helping you further diagnose illegal MSDOS names, which are even more limited than Unix like operating system.

Portable File Sizes

Even though Lou's archive filesystem will allow a file size to be greater than several hundred gigabytes, not all operating systems or filesystems can manage this much data in a single file. If you plan to transfer files to an old Mac or PC desktop you may want to verify the maximum filesize it will support. Likely a single file will need to be less than 4 GB before it will transfer successfully.

It is a good practice to confirm whether files are copied correctly to the Lou mass storage system after you transfer them.

TIP: The easiest way to verify the integrity of your file contents is to use the NAS-developed Shift tool to transfer the files. By default, Shift automatically performs a checksum operation on the data at both the source and the destination, as part of the transfer. If corruption is detected, partial file transfers and checksum operations will be performed until the problem is fixed.

If you transfer files using a command other than Shift, such as scp, the simplest and most lightweight method to verify transferred files is to compare the size (disk usage) and number of files of the original with that of the copy. For example, if you use scp to transfer *dir1* from pfe21 to lfe5:

```
pfe21% du -sk dir1
353760  dir1
pfe21% find dir1 | wc -l
51

pfe21% scp -rp dir1 lfe5:

lfe5% du -sk --apparent-size dir1
353684  dir1
lfe5% find dir1 | wc -l
51
```

In the example, the directory sizes nearly match (353760; 353684), and the number of files matches exactly (51).

Note: In most cases the sizes will not match exactly. Using the--apparent-size option of the du command is necessary on the Lou systems because the data may reside on either tape or disk.