If you are moving large files, use the shiftc command instead of cp or scp. An online NAS service can help diagnose your remote network connection issues, and our network experts can work with your specific file transfer problems.

For fastest file transfer between NAS /nobackup and Lou, log into Lou and use shiftc, cxfscp, or mcp. A simple cp or tar will also work, but at slower speeds.

Moving large amounts of data efficiently to or from NAS across the network can be challenging. Often, minor system, software, or network configuration changes can increase network performance an order of magnitude or more.

If you are experiencing slow transfer rates, try these quick tips:

- NAS /nobackup are mounted on Lou, enabling disk-to-disk copying, which should give the highest transfer rates. You can use the shiftc, cp, or mcp commands to copy files or even make tar files directly from NAS /nobackup to your Lou home directory.
- If using the scp command, make sure you are using OpenSSH version 5 or later. Older versions of SSH have a hard limit on transfer rates and are not designed for WAN transfers. You can check your version of SSH by running the command ssh -V.
- For large files that are a gigabyte or larger, we recommend using shiftc. This application allows for transferring simultaneous streams of data and, when used without the --secure option, doesn't have the overhead associated with encrypting all the data (authentication is still encrypted).
- Another reliable option for large file transfers is through the Shift transfer tool, which includes options specific to the NAS environment, such as checking to see whether files residing on Lou are also on tape.

One-on-One Help

If you would like further assistance, contact the NAS Control Room at support@nas.nasa.gov, and a network expert will work with you or your local administrator one-on-one to identify methods for increasing your transfer rates.

To learn about other network-related support areas see End-to-End Networking Services.

On Lou, commands that should finish quickly may occasionally take a long time. This problem is usually due to slow retrieval of files from tape to disk.

When you run the ls command on Lou, the output shows all your Lou files on disk. However, most of the files are actually written to tape using the Data Migration Facility (DMF).

One reason for slow file retrieval is that for some multiple file transfers—for example, if you do an scp transfer with a list of files—Linux feeds each file to DMF one at a time, and DMF does not deal well with retrieving one file at a time from a long list of files. This means that the tape(s) containing the files is constantly being loaded and unloaded, which is very slow (and is bad for the tape and tape drives). As the list of files gets longer (through the use of "*" or moving a "tree" of files), the problem grows to where it can take hours to transfer a set of files that would only take a few minutes if they were on disk. This can be particularly problematic when several people do these types of file transfers at the same time.

The methods described below can help you avoid these problems.

Note: For more information about the commands in this section, see Data Migration Facility (DMF) Commands.

Optimizing File Retrieval

You can fetch files to disk as a group by running the dmget command before running your file transfer. dmget reads the tape once and gets all the requested files in a single pass.

Run dmget on the same list of files you are about to transfer. Then, after the dmget operation completes, you can transfer the files using scp/ftp/cp as you had originally intended. Or, you can put dmget in the background and run your transfer while dmget is working. If any files are already on disk, dmget sees this and doesn't try to get them from tape.

DMF also provides the dmfind command, which enables you to walk a file tree to find offline files to give to dmget.

Note: Be sure you are in the correct directory before running dmfind. Use the pwd command to determine your current directory.

Please check to make sure too much data isn't brought back online at once, either by using du with the --apparent-size option or by using /usr/local/bin/dmfdu. For example:

```
lou% /usr/local/bin/dmfdu filename
   filename
         13 MB regular          340 files
       1114 MB dual-state       1920 files
      74633 MB offline          2833 files
         13 MB small            340 files
      75761 MB total            5093 files
```

File transfer rates vary depending on the load on the system and how many users are transferring files at the same time. Typically, scp transfers between Lou and NAS compute systems on the /nobackup file system run between 30-120 MB/s for files larger than 100 MB, using the 10-gigabit network interface.

Example 1:

```
lou%  dmget *.data &
lou%  scp -qp *.data  myhost.jpl.nasa.gov:/home/user/dir_name
```

Example 2:

```
lou%  dmfind  /u/username/FY2000 -state OFL -print | dmget &
lou%  scp -rqp /u/username/FY2000 hostname:/nobackup/username/dir_name
```

You can see the state of a file by running dmls -l instead of ls -l.

Maximum Amount of Data to Retrieve Online

The online disk space for Lou is considerably smaller than its tape storage capacity, and it is impossible to retrieve all files to online storage at the same time. Using the Shift tool for file transfers automatically ensures that files on Lou are retrieved in batches and released afterwards so there is no need to manually split up the transfer. If you do not use Shift, however, then you should confirm whether there is enough disk space before you retrieve a large amount of data.

The df command shows the amount of free space in a filesystem. The Lou script dmfdu reports how much total (online and offline) data exists in a directory. To use dmfdu, simply cd into the directory you want to check, and execute the script.

If you would like to know the total amount of data under your home directory on Lou, simply cd to your home directory and use dmfdu *, which will show use for each file or directory.

Lou's archive filesystems are between 254 TB and 1.3 PB in size, but the available space typically floats between 10% to 30%. In Example 3, 29% of space is unused.

It is best to retrieve no more than 10 TB at a time. As shown in Example 3, it is best to release the space (dmput -r) after using the retrieved files (scp, edit, compile, etc), then retrieve the next group of files, use them, and release the space again, and so on.

Example 3:

To retrieve one directory's data from tape, copy the data to a remote host, release the data blocks, and then retrieve more data from tape:

```
lou% df -lh .
Filesystem            Size  Used Avail Use% Mounted on
/dev/cxvm/la1         1.3P  1.1P 225T  83%  /lou/la1


lou% dmfdu project1 project2
project1
         2 MB regular          214 files
        13 MB dual-state         1 files
   2229603 MB offline          101 files
         2 MB small            214 files
   2229606 MB total            315 files

project2
         7 MB regular          245 files
      4661 MB dual-state        32 files
   2218999 MB offline           59 files
         7 MB small            245 files
   2223668 MB total            336 files


lou% cd project1

lou% dmfind . -state OFL -print | dmget &

lou% scp -rp /u/username/project1 remote_host:/nobackup/username

##(Verify that the data has successfully transferred)

lou% dmfind . -state DUL -print | dmput -rw

lou% df -lh .

lou% cd ../project2

lou% dmfind . -state OFL -print | dmget &

lou% scp -rpq /u/username/project2 remote_host:/nobackp/username

lou% dmfind . -state DUL -print  | dmput -rw
```

You can maximize your wide-area network bulk data transfer performance by tuning the TCP settings on your local host. This article shows some common configuration tasks for enabling high-performance data transfers on your system.

Note that making changes to your system should only be done by a lead system administrator or someone who is authorized to make changes.

Linux

1. Edit the file sysctl.conf located under the /etc directory, and add the following lines:

   ```
   net.core.wmem_max = 4194304
   net.core.rmem_max = 4194304
   ```

2. Then have them loaded by running `sysctl -p`

Windows

We recommend using a tool like Dr. TCP.

1. Set the "Tcp Receive Window" to at least 4000000
2. Turn on "Window Scaling," "Selective Acks,"and "Time Stamping"

Other options for tuning Windows XP TCP are the SG TCP Optimizer or using Windows Registry Editor to edit the registry, but the latter is only recommended for Windows users who are already familiar with registry parameters.

Mac OS 10.4

Note that these changes require root access.

In order to allow the Mac operating system to retain the parameters after a reboot, edit the following variables in /etc/sysctl.conf:

1. Set maximum TCP window sizes to 4 megabytes

   ```
   net.inet.tcp.sendspace= 4194304
   net.inet.tcp.recvspace= 4194304
   ```

2. Set maximum Socket Buffer sizes to 4 megabytes

   ```
   kern.ipc.maxsockbuf= 4194304
   ```

Mac OS 10.5 and Later

Use the `sysctl` command for the following variable:

```
sysctl -w net.inet.tcp.win_scale_factor=8
```

If you follow these steps and are still getting less than your expected throughput, please contact the NAS network group at support@nas.nasa.gov (attn: Networks). We will work with you on tuning your system to optimize file transfers.

You can also try the additional steps outlined in the related articles listed below.

This document describes additional TCP settings that can be tuned on high-performance Linux systems. This is intended for 10-Gigabit hosts, but can also be applied to 1-Gigabit hosts. The following steps should be taken in addition to the steps outlined in TCP Performance Tuning for WAN transfers.

Configure the following /etc/sysctl.conf settings for faster TCP

1. Set maximum TCP window sizes to 12 megabytes:

   net.core.rmem_max = 11960320
   net.core.wmem_max = 11960320

2. Set minimum, default, and maximum TCP buffer limits:

   net.ipv4.tcp_rmem = 4096 524288 11960320
   net.ipv4.tcp_wmem = 4096 524288 11960320

3. Set maximum network input buffer queue length:

   net.core.netdev_max_backlog = 30000

4. Disable caching of TCP congestion state (Linux Kernel version 2.6 *only*). Fixes a bug in some Linux stacks:

   net.ipv4.tcp_no_metrics_save = 1

5. Use the BIC TCP congestion control algorithm instead of the TCP Reno algorithm (Linux Kernel versions 2.6.8 to 2.6.18):

   net.ipv4.tcp_congestion_control = bic

6. Use the CUBIC TCP congestion control algorithm instead of the TCP Reno algorithm (Linux Kernel versions 2.6.18 and newer):

   net.ipv4.tcp_congestion_control = cubic

7. Set the following to 1 (should default to 1 on most systems):

   net.ipv4.tcp_window_scaling =1
   net.ipv4.tcp_timestamps = 1
   net.ipv4.tcp_sack = 1

A reboot will be needed for changes to /etc/sysctl.conf to take effect, or you can attempt to reload sysctl settings (as root) with `sysctl -p`.

For additional information visit the Energy Science Network website.

If you have a 10-Gb system or if you follow these steps and are still getting less than your expected throughput, please contact NAS Control Room staff at support@nas.nasa.gov, and we will work with you on tuning your system to optimize file transfers.

Some users prefer to streamline the storage of files (created during a job run) to Lou, within a PBS job. Because direct access to the Lou storage nodes from the NAS compute nodes has been disabled, all file transfers to Lou within a PBS job must first go through one of the appropriate front-end systems (for example, PFEs for bro_ele, sky_ele, cas_ait, rom_ait).

Here is an example of what you can add to your PBS script to accomplish this:

1. ssh to a PFE (for example, pfe*21*) and create a directory on lou where the files are to be copied.

   ssh -q pfe*21* "ssh -q lou mkdir -p $SAVDIR"

   Here, $SAVDIR is assumed to have been defined earlier in the PBS script. Note the use of -q for quiet-mode, and double quotes so that shell variables are expanded prior to the ssh command being issued.

2. Use scp via a PFE to transfer the files.

   ssh -q pfe*21* "scp -q $RUNDIR/* lou:$SAVDIR"

   Here, $RUNDIR is assumed to have been defined earlier in the PBS script.

The following quick and easy techniques may improve your performance rates when transferring files remotely to or from NAS.

This can increase your transfer rates by 5x, compared to older methods such as 3des.

- Transfer files from the /nobackup filesystem, which is often faster than the locally mounted disks.
- If you are using scp and your data is compressible, try adding the -C option to enable file compression, which can sometimes double your performance rates:

  % scp -C *filename user@remote_host.com*:

- For SCP transfers, use a low-process-overhead cipher such as aes128-gcm@openssh.com or arcfour:

  % scp -c -aes128-gcm@openssh.com *filename user@remote_host.com*:

- If you are transferring files from Lou, make sure they are online, rather than on the tape archive, before you perform the transfer operation.

  Note: If you use the shiftc command to transfer your files, it will automatically bring any files that are on the tape archive online before it transfers them. If you are not using shiftc, use the following DMF commands to determine the location of your files and bring them online if necessary:

  % dmls -al *filename* # show the status of your file.
  % dmget *filename* # retrieve your file from tape prior to transferring.

  For a full list of DMF commands, see DMF commands.

- If you are transferring many small files, try using the tar command to compress them into a single file prior to transfer. Copying one large file is faster than transferring many small files.
- For files larger than a gigabyte, we recommended using the Shift Transfer Tool, which can achieve much faster rates than single-stream applications such as scp or rsync.

To improve your performance by modifying your system, see TCP Performance Tuning for WAN Transfers.

If you continue experiencing slow transfers and want to work with a network engineer to help improve file transfers, please contact the NAS Control Room at support@nas.nasa.gov.

To address security issues with the scp command, we are in the process of adding checks to ensure that the files returned by a remote server match the files requested by the user. In some cases, the checks implemented in scp to address this issue may cause requested files to be rejected with the following error message:

```
protocol error: filename does not match request
```

This error can occur when you use quotation marks to escape special characters (such as a space) on the remote server, or when you use wildcard characters. The safest way to avoid this issue is to use the sftp command instead of scp to retrieve files. This avoids complications due to interpretation of the requested file names by the shell on the remote server.

For file retrieval, the syntax and command-line options for sftp are very similar to those for scp. For example, to retrieve files matching test*.c from a remote server to the directory *somedir*, use the following command line:

```
?$ sftp somehost:'test*.c' somedir/
```

Alternatively, you can add the -T option to the scp command line to disable checking of the file names returned by the remote server. Following the example above, the scp command line would be:

```
$ scp -T somehost:'test*.c' somedir/
```

If you need further help, please contact the NAS Control Room at (800) 331-8737 or (650) 604-4444.