

Data Movement and Storage – Lab

To get started on labs, untar the lab files into your own directory with:

```
cd && tar xzf ~train100/labs.tgz
```

Resources

Ranch User Guide – <http://services.tacc.utexas.edu/index.php/ranch-user-guide>

GridFTP tools – <https://www.teragrid.org/web/user-support/gridftp>

Windows Secure Copy client – <http://tartarus.org/~simon/putty-snapshots/x86/pscp.exe>

To Complete This Section

- Exercise: Retrieve Files with Secure Copy
- Striping Test

Filesystems on Ranger

Filesystem	Total Size	Per User Quota	Shortcut	Retention Policy
\$HOME	~100 TB	6 GB	cd	Nightly backup
\$WORK	~200 TB	350 GB	cdw	No backup
\$SCRATCH	~800 TB	400 TB	cds	Purged every 10 days

Usage is listed in the login splash screen. You can check usage with the du command.

```
$ du -sm $HOME
1316 /share/home/00692/train00
```

Exercise: Retrieve Files with Secure Copy

From Linux or Mac

1. Open a terminal window with a right-click on the desktop. You should now have two windows open, one that is on the local machine and one that is logged into Ranger.
2. On Ranger, create a 100 MB file with “cd && dd if=/dev/zero of=big bs=100M count=1”.
3. Use “ls -la” to find out how large the file is.
4. On your local machine, copy that file from Ranger with scp. Put your username where the example shows train2xx. The single period at the end represents the current directory on the local drive as the place to store the retrieved file.
[local]\$ scp train2xx@ranger.tacc.utexas.edu:big .
About how long did that take?
5. Now copy the file back and use the time command to measure how long it takes.
[local]\$ scp big train2xx@ranger.tacc.utexas.edu:big

From Windows

1. You already have one Putty window open and logged into Ranger. Now open a local command line by going to the Start Menu and typing “cmd” into the search box. In what directory is this command prompt? Type “dir” to see the contents of this directory.
2. Right-click on this link, <http://tartarus.org/~simon/putty-snapshots/x86/pscp.exe>, to download Putty. Select “save as...” and save it in the same directory as the command prompt location.
3. On Ranger, create a 100 MB file with “cd && dd if=/dev/zero of=big bs=100M count=1”.
4. Use “ls -la” to find out how large the file is.
5. Run pscp at the command prompt to copy the file to the local machine. Replace train2xx with your username on Ranger.
C:\users\lab> pscp train2xx@ranger.tacc.utexas.edu :big .
6. Copy the same file back to to Ranger.
C:\users\lab> pscp big train2xx@ranger.tacc.utexas.edu:big

Striping Test

The lfs setstripe command has four arguments:

```
lfs setstripe <file|dir> <bytes per OST> <index of first stripe> <number of OSTs>
```

1. File or directory for which to set the stripe.
2. The number of bytes on each OST, specified with k, m, or g for KB, MB or GB.
3. OST index of first stripe (-1 for filesystem default)
4. Number of OSTs to stripe over.

So you would, for instance, call “lfs setstripe bigfile 4M -1 2” to stripe across two OSTs.

1. The lfs command will tell you quotas and striping for filesystems. Get the quota for \$WORK with
lfs quota \$WORK
To see striping, try creating a small file and then using lfs to get its stripe information.
ls>file.txt
lfs getstripe file.txt
The listing at the end of the results show which OSTs have parts of the file.
2. You can set striping on a file or directory with the set stripe command. First set it for a file.
lfs setstripe stripy.txt
ls -la>stripy.txt
lfs getstripe stripy.txt
Now try the same thing for a directory. First create a directory, then set its striping, then make a file within that directory.
mkdir s
cd s
lfs setstripe . 4M -1 6
ls -la>file.txt
lfs getstripe file.txt
You should see the file striped across six OSTs.

3. Now we run an MPI program that writes in parallel to a single file and test how the speed depends on striping. The MPI-IO program comes from <http://beige.ucs.indiana.edu/I590/node86.html>. First, compile the code.

```
cd mpiio  
make
```
4. Then examine ranger.sh. It performs the same striping commands you have just tried. This is what the script does:
 - a. Create a working directory on \$SCRATCH.
 - b. Copy the mpiio writing and reading programs into that directory and cd there.
 - c. Run the writing and reading programs with default striping, taking timings in the process.
 - d. Do the same for 8-way striping.
 - e. Do the same for 2-way striping.
 - f. Delete the working directory.
5. You can change the BLOCKS variable to change the size of the MPIIO file. Don't forget to set the account to the correct account for this class.
6. Then submit ranger.sh with qsub.
7. Extra while waiting for the scheduler: What is the default stripe for \$HOME, \$WORK, and \$SCRATCH? Does this choice make sense?