



Programming Environment on Ranger Cluster

Cornell Center for Advanced Computing
October 12, 2009

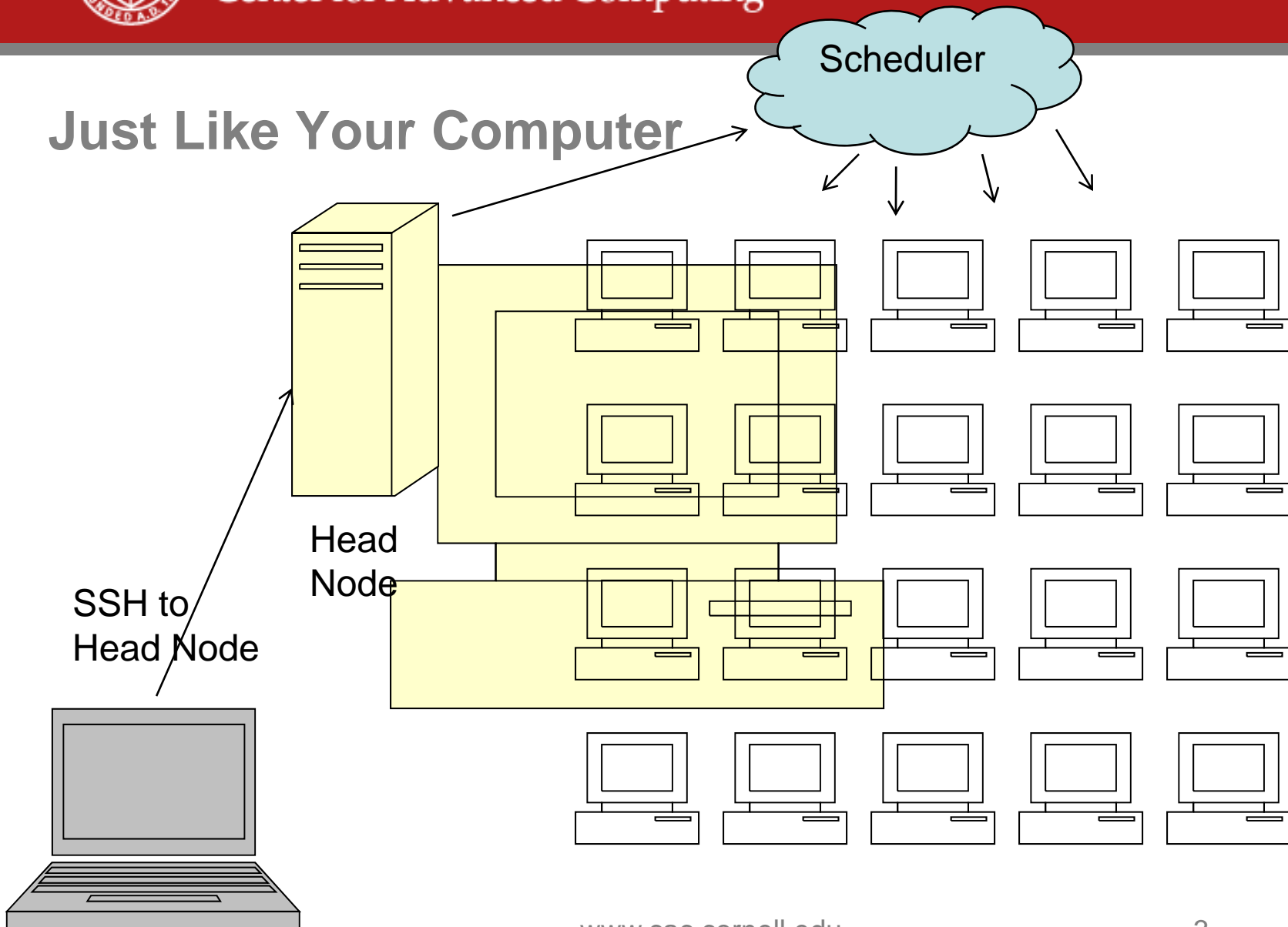


User Guides

- TACC
 - [Ranger](http://services.tacc.utexas.edu/index.php/ranger-user-guide) (http://services.tacc.utexas.edu/index.php/ranger-user-guide)
 - [Spur](http://services.tacc.utexas.edu/index.php/spur-user-guide) (http://services.tacc.utexas.edu/index.php/spur-user-guide)
- CAC
 - [Linux](http://www.cac.cornell.edu/wiki/index.php?title=V4_Linux_Cluster) (http://www.cac.cornell.edu/wiki/index.php?title=V4_Linux_Cluster)
 - [Windows](http://www.cac.cornell.edu/wiki/index.php?title=V4_Windows_Cluster) (http://www.cac.cornell.edu/wiki/index.php?title=V4_Windows_Cluster)
- Tutorials
 - [Beginners Unix](http://info.ee.surrey.ac.uk/Teaching/Unix/) (http://info.ee.surrey.ac.uk/Teaching/Unix/)



Just Like Your Computer





SSH Clients

- Windows: [Putty](#)
- Linux: built-in as “ssh”
- Mac: built-in as “ssh”

Login now to `ranger.tacc.utexas.edu`, e.g.
`ssh train1xx@ranger.tacc.utexas.edu`



Login

- ssh train1xx@ranger.tacc.utexas.edu
- Find your account number at bottom of splash screen.

```
----- Project balances for user train100 -----
| Name          Avail SUs    Expires |
| 20091012DATA    5000  2009-10-20 |
-----
----- Disk quotas for user train100 -----
| Disk          Usage (GB)    Limit    %Used    File Usage    Limit    %Used |
| /share        0.0          6        0.00     47          100000   0.05 |
| /work         0.0         350      0.00     1           2000000  0.00 |
-----
```

- echo \$SHELL
- chsh -l
- man chsh



Login

- `env` (show environment variables – persists)
- `set` (show shell variables – current shell only)

- `pwd`
- `ls -la`
- `cat .login`
- `cat /usr/local/etc/login`
- `cat .login_user` (create then edit this one to personalize)

- `pwd`
- `cp ~train100/day1.tgz .` (copy all files in ~train100 ending in .gz)
- `tar zxvf ~train100/day1.tgz`



Un-TAR Lab Files

- TAR = Tape archive.
- Just concatenates files.
- `tar <switches> <files>`
- z = compress or decompress
- x = extract
- c = create
- v = verbose
- t = list files
- f = next argument is the file to read or write
- ~userid is the home directory of that user
- For example, to create a tar: `tar cvf myfiles.tar dir1 dir2 README`

```
tar zxvf ~train100/day1.tgz
```



Basic file transfer

- SCP (secure copy protocol) is available on any POSIX machine for transferring files.

```
naw47@varushka bin] $ scp ~/oretools_svg.xpi ranger.tacc.utexas.edu:~/oretools.xpi
oretools_svg.xpi          18% 1824KB  1.8MB/s  00:04 ETA
```

- `scp myfile.tar.gz remoteUser@ranger.tacc.utexas.edu:remotePath`
- `scp remoteUser@ranger.tacc.utexas.edu:~/work.gz localPath/work.gz`
- SFTP (secure FTP) is generally available on any POSIX machine and is roughly equivalent to SCP, just with some added UI features. Most notable, it allows browsing:

```
naw47@varushka bin] $ sftp consultrh5
Connecting to consultrh5...
sftp> cd stuff
sftp> lcd ../
sftp> put file
```




PSCP and SFTP Clients

- Windows
 - WinSCP (<http://winscp.net/>)
 - [Putty's](http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html) pscp and psftp
(<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>)

From your local machine, copy a file from Ranger to the local machine.

```
C:\Users\ajd27>pscp train200@ranger.tacc.utexas.edu:README .  
Using keyboard-interactive authentication.  
Password:  
README | 6 kB | 6.6 kB/s | ETA: 00:00:00 | 100%
```



Basic file transfer

- On most Linux systems, scp uses sftp, so you're likely to see something like this:

Command	Filesize	Transfer Speed
scp	5 MB	44 MB/s (10 sec)
sftp	5 MB	44 MB/s
scp	5 GB	44 MB/s (2:00)
sftp	5 GB	44 MB/s (2:00)

- The CW is that sftp is slower than scp and this may be true for your system, but you're likely to see the above situation.



Ranger File Systems

- No local disk storage (booted from 8 GB compact flash)
- User data is stored on 1.7 PB (total) Lustre file systems, provided by 72 Sun x4500 I/O servers and 4 Metadata servers.
- 3 mounted filesystems, all available via Lustre filesystem over IB connection. Each system has different policies and quotas.

Alias	Total Size	Quota (per User)	Retention Policy
\$HOME	~100 TB	6 GB	Backed up nightly; Not purged
\$WORK	~200 TB	350 GB	Not backed up; Not purged
\$SCRATCH	~800 TB	400 TB	Not backed up; Purged periodically



Accessing File Systems

- File systems all have aliases to make them easy to access:
 - cd \$HOME cd
 - cd \$WORK cdw
 - cd \$SCRATCH c ds
- To query quota information about a file system, you can use the lfs quota command:

```
login3%  
login3% lfs quota -u $USER $WORK  
Disk quotas for user tg801871 (uid 801871):  
  Filesystem kbytes  quota  limit  grace  files  quota  limit  grace  
/work/00940/tg801871  
                4      0 367001600      1      0 2000000
```

```
login3% du -sm ~train00  
1316    /share/home/00692/train00
```



Software

- [Software](#) section in User Guide
- [Software](#) list available on Ranger
- The modules utility is used to provide a consistent, uniform method to access software.
- `module help < module_name >`
- `module load < module_name >`



MODULES Command (Ranger-only)

- Affects \$PATH, \$MANPATH, \$LIBPATH
- Load specific versions of libraries/executables
- Works in your batch file
- Define environment variables:
 - TACC_MKL_LIB, TACC_MKL_INC, TACC_GOTOBLAS_LIB

```
----- /opt/apps/intel10_1/modulefiles -----  
acml/4.1.0          hecura/0.1          mvapich2/1.2  
autodock/4.0.1     hmmer/2.3.2        ncl_ncarg/5.0.0  
boost/1.34.1       metis/4.0           nco/3.9.5  
boost/1.37.0       mvapich/1.0         netcdf/3.6.2  
fftw3/3.1.2        mvapich/1.0.1(default) openmpi/1.2.4  
gotoblas/1.26(default) mvapich-devel/1.0  openmpi/1.2.6  
gotoblas/1.30      mvapich-old/1.0.1  openmpi/1.3(default)  
hdf5/1.6.5         mvapich-ud/1.0
```



Try MODULES

- module list
- module avail
- module load intel # look how it responds
- module swap pgi intel # so delete pgi
- module load fftw2
- module del fftw2
- There can be orders to how you load these. Unload MPI, then choose a compiler, then load the MPI version.



Modules Examples

```
login4% module list
```

```
Currently Loaded Modulefiles:
```

- | | | |
|------------------------|---------------------|--------------------|
| 1) TACC-paths | 8) globus/4.0.8 | 15) TERAGRID-BASIC |
| 2) Linux | 9) srb-client/3.4.1 | 16) GLOBUS-4.0 |
| 3) cluster-paths | 10) tg-policy/0.2 | 17) TERAGRID-DEV |
| 4) pgi/7.2-5 | 11) tgproxy/0.9.1 | 18) CTSSV4 |
| 5) mvapich/1.0.1 | 12) tgresid/2.0.3 | 19) cluster |
| 6) binutils-amd/070220 | 13) tgusage/3.0 | 20) TACC |
| 7) gx-map/0.5.3.3 | 14) uberftp/2.4 | |

```
login4% module avail
```

```
----- /opt/apps/pgi7_2/modulefiles -----  
acml/4.1.0          hdf5/1.6.5          mvapich2/1.2  
autodock/4.0.1     hecura/0.1          ncl_ncarg/5.0.0  
fftw3/3.1.2        metis/4.0           nco/3.9.5  
gotoblas/1.26(default) mvapich/1.0.1      netcdf/3.6.2  
gotoblas/1.30      mvapich-old/1.0.1  openmpi/1.3
```




Now Swap Compilers

- If PGI is loaded, load Intel
module swap pgi intel
- Try module avail again and look at what is there.

```
login4% login4% module avail
```

```
----- /opt/apps/intel10_1/modulefiles -----  
acml/4.1.0          hecura/0.1          mvapich2/1.2  
autodock/4.0.1     hmmer/2.3.2         ncl_ncarg/5.0.0  
boost/1.34.1       metis/4.0           nco/3.9.5  
boost/1.37.0       mvapich/1.0         netcdf/3.6.2  
fftw3/3.1.2        mvapich/1.0.1(default) openmpi/1.2.4  
gotoblas/1.26(default) mvapich-devel/1.0  openmpi/1.2.6  
gotoblas/1.30      mvapich-old/1.0.1  openmpi/1.3(default)  
hdf5/1.6.5         mvapich-ud/1.0
```



Two Time Commands

- Used to see how long your program runs and estimate if it's having gross difficulties
- `/usr/bin/time` generally more information

```
login4% time ./hello
Hello world!
0.000u 0.030s 0:00.06 50.0%      0+0k 0+0io 2pf+0w
```

```
login4% /usr/bin/time ./hello
Hello world!
0.00user 0.01system 0:00.03elapsed 32%CPU (0avgtext+0avgdata 0maxresident)k
0inputs+0outputs (0major+213minor)pagefaults 0swaps
```



Submit a Job

- Want to run a batch script:

```
#!/bin/sh
echo Starting job
date
/usr/bin/time ./hello
date
echo Ending job
```

- Have to ask scheduler to do it.

```
qsub -A 20091012DATA job.sge
```

```
#!/bin/sh
#$ -N ht3d-hyb
#$ -cwd
#$ -o $JOB_NAME.o$JOB_ID
#$ -j y
#$ -A C-RANGER
#$ -q development
#$ -pe 4way 16
#$ -V
#$ -l h_rt=00:10:00
echo Starting job
date
/usr/bin/time ./hello
date
echo Ending job
```



How Are the Queues?

- List available queue: `qconf -sql`
- Soft and hard wall clock limits: `qconf -sq <queue name>`
- Queue core limit: `cat /share/sge6.2/default/tacc/sge_esub_control`
 - Try “make cores” in submit directory.
- `showq` or “`showq -u`”
- Delete job: `qdel` or `qdel -f`



Queue Examples

```
login3% qconf -sql  
clean  
development  
large  
long  
normal  
request  
reservation  
serial  
sysdebug  
systest  
vis
```

Slots = number of cores, 16 per node
pe = wayness, how many cores per node
Job is killed if over time limit.

```
login3% qconf -sq development  
qname                development  
qtype                BATCH INTERACTIVE  
pe_list              16way 15way 14way 12way 8way 4way 2way 1way  
slots                16  
tmpdir               /tmp  
shell                /bin/csh  
prolog               /share/sgc/default/pe_scripts/prologWrapper  
epilog               /share/sgc/default/pe_scripts/tacc_epilog_n  
shell_start_mode     unix_behavior  
s_rt                 07:58:00  
h_rt                 08:00:00
```

Why 15way?



Showq is 985 Lines

login3% showq -u

ACTIVE JOBS-----

JOBID	JOBNAME	USERNAME	STATE	CORE	REMAINING	STARTTIME
-------	---------	----------	-------	------	-----------	-----------

=====

378 active jobs : 3629 of 3852 hosts (94.21 %)

WAITING JOBS-----

JOBID	JOBNAME	USERNAME	STATE	CORE	WCLIMIT	QUEUETIME
-------	---------	----------	-------	------	---------	-----------

=====

WAITING JOBS WITH JOB DEPENDENCIES---

JOBID	JOBNAME	USERNAME	STATE	CORE	WCLIMIT	QUEUETIME
-------	---------	----------	-------	------	---------	-----------

=====

UNSCHEDULED JOBS-----

JOBID	JOBNAME	USERNAME	STATE	CORE	WCLIMIT	QUEUETIME
-------	---------	----------	-------	------	---------	-----------

=====

Total jobs: 963 Active Jobs: 378 Waiting Jobs: 469 Dep/Unsched Jobs: 116



States

- Unscheduled – Likely not good
- DepWait – You can ask that one job run after another finishes.
- w(aiting) – Queued, waiting for resources to run.
- r(unning) – As far as SGE is concerned, it's going.
- h(old)
- s(uspended)
- E(rror)
- d(eletion)



Submit a Job Example

- `cd submit` # submit is a subfolder in the lab files
- `make` # Compile the executable "hello". Guess what it does?
- `ls -la` # Take a look at what compiled.
- `./hello` # to run job
- `less job.sge` # examine the script
- `./job.sge` # Run the job by running the script. The node will do this.
- `qsub -A 20091012DATA job.sge` # Submit the job



Running and Output

- `showq -u` # Watch it run.
- `less hello.oXXX` # Look at the output file when it's done.
- Try comparing the environment variables on login with batch.
 - `env | sort > z.txt`
 - `diff z.txt hello.oXXX | less`



Environment Variables in Batch

- > ENVIRONMENT=BATCH
- > HOSTNAME=i182-401.ranger.tacc.utexas.edu
- > JOB_ID=743637
- > JOB_NAME=hello
- > JOB_SCRIPT=/share/sgc/execd_spool//i182-401/job_scripts/743637
- > NHOSTS=1
- > NQUEUES=1
- > NSLOTS=16
- > PE=1way
- > PE_HOSTFILE=/share/sgc/execd_spool//i182-401/active_jobs/743637.1/pe_hostfile
- > QUEUE=development
- > SGE_ACCOUNT=20091012DATA
- > SGE_CWD_PATH=/share/home/0002/train200/submit
- > SGE_O_SHELL=/bin/csh
- > SGE_O_WORKDIR=/share/home/0002/train200/submit
- > SGE_STDERR_PATH=/share/home/0002/train200/submit/hello.o743637
- > SGE_STDOUT_PATH=/share/home/0002/train200/submit/hello.o743637



Parallel Environment

- Each node has 16 cores and is used by one person at a time
- `#$ -pe 1way 16` Run one task on a node with 16 cores
- `#$ -q serial`
- `./hello`

- `#$ -pe 8way 64` Run 8 tasks/node on 4 nodes
- `#$ -q normal`
- `export MY_NSLOTS=31` Launch 31 tasks
- `lbrun ./a.out` Run with mpi wrapper



To Edit A File in VI (short for “visual”)

- “vi filename” will open it or create it if it doesn’t exist.
- Command mode and Insert mode. You start in command mode.
- Command mode. Cursors work here, too.
 - :w Writes a file to disk.
 - :q Quits
 - :q! Quits even if there are changes to a file
 - i Takes you to insert mode
- Insert Mode
 - Cursors, typing characters, and deleting work here.
 - Escape key takes you to command mode.
- Ctrl-c will get you nowhere.



Globus toolkit

- Install the globus client toolkit on your local machine and setup a few environment variables.

```
#GLOBUS Teragrid single sign-on stuff
GLOBUS_LOCATION=$HOME/globus
MYPROXY_SERVER=myproxy.teragrid.org
MYPROXY_SERVER_PORT=7514
export GLOBUS_LOCATION MYPROXY_SERVER MYPROXY_SERVER_PORT
. $GLOBUS_LOCATION/etc/globus-user-env.sh
```

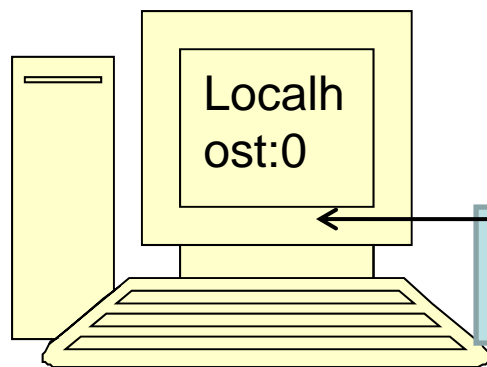
- Acquire a proxy certificate and then you have a temporary certificate which will allow you to ssh/scp/sftp without re-entering a password.

```
naw47@varushka bin]$ myproxy-logon -T -l nwoody
Enter MyProxy pass phrase:
A credential has been received for user nwoody in /tmp/x509up_u16777502.
Trust roots have been installed in /home/gfs01/naw47/.globus/certificates/.
naw47@varushka bin]$ gsiscp ~/file.big ranger.tacc.utexas.edu:~/file.big
file.big 70% 311MB 14.8MB/s 00:08 ETA
```



Again with X-Windows

- Start X-Windows server on local machine.



```
>echo $DISPLAY  
localhost:39.0  
>emacs README&
```

localhost:39.0

```
>jobs  
>kill %1
```



Login with X-Windows

- Start Exceed->Exceed on Windows Startup menu (Already started on Mac and Linux)
- ssh -X on Linux, Mac. For Windows, select in Putty Connection->SSH->X11, and check "X11 Forwarding"
- Type in username and password.
- echo \$DISPLAY
- emacs README& # This runs emacs in the background.
- At the command prompt, type "jobs" to see that you have a backgrounded job.
- Try Emacs for a while, then kill it with
- kill %1



Again with VNC

- VNCServer
 - used to start a VNC (Virtual Network Computing) desktop.
 - a Perl script which simplifies the process of starting an Xvnc server.
 - can be run with no options at all. In this case it will choose the first available display number
- VNCServer copies a bitmap of the X-Windows screen across.
- Can be much less chatty than X-Windows.
- Good for remote graphics.
- VNCServer screen 4 uses TCP/IP port 5904.
- SSH to ranger. Start it. Connect with VNC Client. Kill it.



Connect with VNC

- Start VNC on Ranger
 - First ssh normally.
 - Type “vncserver” and look for screen number, for example. “4”.
- Connect with a client
 - RealVNC or TightVNC on Windows
 - On Linux, vinagre or vncviewer
 - Connect to “ranger.tacc.utexas.edu:4” or your port number
- Be sure to kill it when you are done
 - vncserver –kill 4



VNCServer example

```
login3% vncserver
```

```
New 'login3.ranger.tacc.utexas.edu:1 (train200)' desktop is  
login3.ranger.tacc.utexas.edu:1
```

```
Starting applications specified in /share/home/0002/train200/.vnc/xstartup  
Log file is /share/home/0002/train200/.vnc/login3.ranger.tacc.utexas.edu:1.log
```

```
login3% vncserver -kill :1  
Killing Xvnc process ID 11406
```



Questions?