Data Movement and Storage
Data Location, Storage, Sharing and Movement

• Four of the seven main challenges of Data Intensive Computing, according to SC06.
• (Other three: viewing, manipulation, interpretation)
• Data growing much faster than Moore's law (abstract)
• Internet: 20 MB/s (less abstract)
  – 1 TB – 14 hours Internet
  – 1 PB – 20 months Internet
The Seriously-Out-of-Date Map
Problem Solved

• TeraGrid network ten times faster.
• What does that fix?
• How do these numbers feel?
  – 1 TB – 14 hours Internet, 1.4 hours TeraGrid
  – 1 PB – 20 months Internet, 2 months TeraGrid
• Factor of 10 is good but we need more complete approaches.
Are You on the Map?

- No NUBB charges.
- Access to 10 Gb connection on campus.
- Access to 10 Gb connection from country.
- Then test it.
  - Network ops help
  - Talk with provider
Secure file transfer - sftp

- sftp <username>@tg-login.ranger.tacc.teragrid.org
- Enter password
- Navigate to appropriate local and remote directories
- Copy file

- Your performance may vary:
  - Getting 31 MB file
    - deneshta (my Mac) - 3.1 MB/s - 10 sec
    - linuxlogin3 (CAC login node) - 0.854 MB/s - 37 sec
Basic file transfer

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

  - `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:

  ```
  naw470@varushka bin]$ sftp consultrh5
  Connecting to consultrh5...
  sftp> cd stuff
  sftp> lcd ../
  sftp> put file
  ```
Basic file transfer

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

<table>
<thead>
<tr>
<th>Command</th>
<th>Filesize</th>
<th>Transfer Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>scp</td>
<td>5 MB</td>
<td>44 MB/s (10 sec)</td>
</tr>
<tr>
<td>sftp</td>
<td>5 MB</td>
<td>44 MB/s</td>
</tr>
<tr>
<td>scp</td>
<td>5 GB</td>
<td>44 MB/s (2:00)</td>
</tr>
<tr>
<td>sftp</td>
<td>5 GB</td>
<td>44 MB/s (2:00)</td>
</tr>
</tbody>
</table>

• 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.
Testing Speeds

- Create 10MB file
  - `dd if=/dev/zero of=$SCRATCH/10mb bs=1024 count=10240`
- sftp that file
  - `sftp trainxxx@tg-login.ranger.teragrid.org`
  - `get /scratch/0000/trainxxx/10mb`
Globus toolkit

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

```bash
# 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.

```bash
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
```
UberFTP

- UberFTP is an interactive GridFTP-enabled client that supports GSI authentication and parallel data channels.
- UberFTP is to globus-url-copy what sftp is to scp
  - GSI authentication means that once you’ve acquired a proxy certificate from the myproxy server, you won’t need to provide a password again.
  - Parallel data channels means the client opens multiple FTP data channels when transferring files, but all are controlled through a single control channel, hopefully increasing the speed.
  - UberFTP and globus-url copy also support third party transfers, which means you can transfer from a remote site to another remote site (provided they all accept the current proxy certificate).
UberFTP example

- Moving a 450 MB file from a workstation on a gigabyte connection to ranger with variable numbers of data channels.

```
[naw470@varushka bin]$ uberftp ranger.tacc.utexas.edu
220 login3.ranger.tacc.utexas.edu GridFTP Server 2.8 (gcc64, 1217607445-63) [Globus Toolkit 4.0.0] ready.
230 User tg801871 logged in.
UberFTP> parallel
Using 1 parallel data channels for extended block transfers
UberFTP> put file.big
file.big: 457651136 bytes in 20.379396 Seconds (21.416MB/s)
UberFTP> parallel 8
Using 8 parallel data channels for extended block transfers
UberFTP> put file.big
file.big: 457651136 bytes in 15.107727 Seconds (28.889MB/s)
UberFTP> parallel 16
Using 16 parallel data channels for extended block transfers
UberFTP> put file.big
file.big: 457651136 bytes in 14.162568 Seconds (30.817MB/s)
UberFTP>
```
GridFTP Optimization in UberFTP

• Lots of network traffic
  – parallel 2
  – tcpbuf 4194304
• Less traffic, large file
  – parallel 1
  – tcpbuf 8388608
• More options
  – Striping
  – Multiple servers, a typical simple approach
  – DMOVER, Phedex represent what can be done.
Practical Approaches To Very Large Data Transfers

- Use short hop to Teragrid site.
- Transfer disks.
- Multiple simultaneous gridftp or even ftp streams.
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.

<table>
<thead>
<tr>
<th>Alias</th>
<th>Total Size</th>
<th>Quota (per User)</th>
<th>Retention Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>$HOME</td>
<td>~100 TB</td>
<td>6 GB</td>
<td>Backed up nightly; Not purged</td>
</tr>
<tr>
<td>$WORK</td>
<td>~200 TB</td>
<td>350 GB</td>
<td>Not backed up; Not purged</td>
</tr>
<tr>
<td>$SCRATCH</td>
<td>~800 TB</td>
<td>400 TB</td>
<td>Not backed up; Purged every 10 days</td>
</tr>
</tbody>
</table>
Accessing File Systems

• File systems all have aliases to make them easy to access:
  – cd $HOME  cd
  – cd $WORK  cdw
  – cd $SCRATCH  cds

• To query quota information about a file system, you can use the lfs quota command:

```
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
```
Lustre

- All Ranger filesystems are Lustre, which is a globally available distributed file system.
- The primary components are the MDS and OSS nodes, OSS contain the data, MDS contains the filename to object map.

Lustre

- The client (you) must talk to both the MDS and OSS servers in order to actually use the Lustre system.
- Actual File I/O goes to the OSS, opening files, directory listings, etc go to the MDS.
- The client doesn’t have to care, the Lustre file system simply appears like any other large volume that would be mounted on a node.
Lustre

• The Lustre filesystem scales with the number of OSS’s available.
• Ranger provides 72 Sun I/O nodes, with an achievable data rate of something like 50GB/s, but this speed is being split by all users of the system.
• Fun comparison:
  – 500 MB file, on my workstation using 2 disks in a striped RAID array.
  – Same file, on Ranger, copying from $HOME to $SCRATCH
  – Lustre scales to multiple nodes reading/writing!

```
Workstation local copy

naw47@varushka ~]$ time cp file.big file2.big
real    0m1.580s
user    0m0.053s
sys     0m1.468s

Ranger Lustre copy

login4% time cp $HOME/file.big $SCRATCH/file.big
0.000u 3.020s  0:03.46  87.2%  0+0k  0+0io 0pf+0w
login4% time cp $HOME/file.big $HOME/file1.big
0.000u 2.220s  0:02.81  79.0%  0+0k  0+0io 0pf+0w
```
Simultaneous Writes

- Poor with most filesystems
Group Test

- Use a large file to test simultaneous access
  `dd if=/dev/zero of=$SCRATCH/1gb bs=1024 count=1024000`
- One person tries
  `time cp $SCRATCH/1gb $SCRATCH/z`
- Then all at once, again.
- And one person deletes
  `time rm $SCRATCH/*`
- And all delete.
Archive

- Over a petabyte. Disk and tape.
- Currently no quota
- Another machine.
- `rcp ${ARCHIVER}:${ARCHIVE}/myfile $WORK`
  `rcp $WORK/* ${ARCHIVER}:${ARCHIVE}`
- Or login to `${ARCHIVER}` and cda to directory to look around.
- May take minutes or hours to reconstitute.
- Don’t go directly from archive to a running job.
BBCP

- Transfer to tape archive `${ARCHIVE}`.
- `scp` much slower. 15 MB/s vs 125 MB/s.
- `login4% bbcp < data > `${ARCHIVER}:`$ARCHIVE`
- Transfers whole directories.
XUFS

- sshfs on steroids, and backwards

[ajd27@v4linuxlogin1 ~]$ xufs/bin/ussh tg123123@ranger.tacc.utexas.edu
Password:
login3% pwd
/share/home/00933/tg459569/xufs-rhome
login3% ls -la
total 15340
drwxr--r-- 23 tg459569 G-80907  4096 Mar 27 15:14 ..
drwxr-xr-x  2 tg459569 G-80907  4096 Mar 27 15:14 Desktop
drwxr-xr-x  2 tg459569 G-80907  4096 Mar 27 15:14 VTune
drwxrwxrwx  2 tg459569 G-80907  4096 Mar 27 15:14 WINDOWS
drwxrwxrwx  2 tg459569 G-80907  4096 Mar 27 15:14 bin
drwxrwxrwx  20 tg459569 G-80907  4096 Mar 27 15:14 dev
XUFS Features

• Metadata as you ls.
• Striped gridftp when fopen().
• Send on close, last close wins.
• Lives in user space on home and remote machines.
• For data and code.
• Offers beta code exciting experience:

*** glibc detected *** malloc(): memory corruption: 0x00000000007858d0 ***
*** glibc detected *** malloc(): memory corruption: 0x0000000000785780 ***
Abort
*** glibc detected *** malloc(): memory corruption: 0x00000000007858d0 ***
*** glibc detected *** malloc(): memory corruption: 0x00000000007858d0 ***
Abort
XUFS Appropriateness

- Similar to GPFS-WAN, sshfs, and many others, but...
- You already have a fair amount of disk space on your home machine.
- You don't want two copies of your code floating around.
- No need for a lightning-fast synchronization when writing.
- Sharing among accounts at TG institution is rare.
- With striped gridftp underneath, there is no loss of efficiency.