Welcome!

Virtual tutorial starts at 15:00 BST

archer

epcc

THE UNIVERSITY OF EDINBURGH

Data Transfer to UK-RDF

Archiving and Copying

ARCHER Filesystems

`/home`: backed-up, NFS, available on login and service nodes.

`/work`: **not backed-up**, Lustre, available on login, service and compute nodes.

RDF: backed-up only for disaster proofing **(accidental deletion recovery not supported)**, GPFS, available on login nodes (and serial nodes).

Accessing the RDF

Directly mounted on ARCHER at:

```
/epsrc
/nerc
/general
```

depending on your funding body.

RDF additionally has its own Data Transfer Nodes (DTNs): **dtn01.rdf.ac.uk**, **dtn02.rdf.ac.uk**. Should be used when transferring between the RDF and a remote machine.

Archiving – Motivation

More efficient use of the filesystem – single file requires fewer metadata operations to move/copy/access.

Can dramatically improve performance, especially with a large number of small files.

Example, 23GB of data = ~13000 32KB-5MB files:

```
$> time cp -r mydata /general/z01/z01/dsloanm/

real    59m47.096s
user    0m0.148s
sys     0m37.358s
```

Archiving – Motivation

Same files in an archive:

```
$> time cp mydata.tar /general/z01/z01/dsloanm/

real    3m3.698s
user    0m0.008s
sys     0m33.958s
```

Some initial overhead required for archive creation but time saved on subsequent accesses.

Serial queues should be used for any long running tasks.

Archiving – Utilities

Common archiving utilities on ARCHER:

- tar
- cpio
- zip

Some technical differences but choice mostly personal preference.

Generally recommend forgoing compression to speed up process but there is a compression/transfer time trade-off.

Archiving – tar creation

Ubiquitous “tape archive” format.

Common options:

-c create a new archive
-v verbosely list files processed
-W verify the archive after writing
-l confirm all file hard links are included in the archive
-f use an archive file

Example command:

```
tar -cvWlf mydata.tar mydata
```

Archiving – tar extraction and verification

-x extract from an archive

```
tar -xf mydata.tar
```

-d “diff” archive file against a set of data

```
$> tar -df mydata.tar mydata
mydata/damaged_file: Mod time differs
mydata/damaged_file: Size differs
```

Note: tar archives do not store file checksums

Original data must be present during verification.

Archiving – cpio creation

Archiving utility provided by most Linux distributions.

Common options:

-o create a new archive (copy-out mode)

-v verbose

-H use the given archive format (crc recommended)

No recursive flag – combine with “find” for directories

Example command:

```
find mydata/ | cpio -ovH crc > mydata.cpio
```

Archiving – cpio extraction and verification

-i extract from archive (copy-in mode)

-d create directories as necessary

```
cpio -id < mydata.cpio
```

--only-verify-crc verifies file checksums (skips extraction)

```
$> cpio -i --only-verify-crc < mydata.cpio
cpio: mydata/file: checksum error (0x1cd3cee8,
should be 0x1cd3cf8f)
204801 blocks
```

Archiving – zip creation

Widely used and supported by most major systems, including current versions of Windows.

Common options:

-r recursively archive files and directories

-0-9 compression level (-0 recommended on ARCHER)

Example command:

```
zip -0r mydata.zip mydata
```

Note: zip files **do not preserve hard links** (data is copied).

Archiving – zip extraction and verification

Uses a separate utility for extraction.

```
unzip mydata.zip
```

-t test archive (zip file stores CRC values by default)

```
$> unzip -t mydata.zip
Archive: mydata.zip
      testing: mydata/  OK
      testing: mydata/file    OK
No errors detected in compressed data of mydata.zip.
```

Copying – Utilities

Local copy from ARCHER

- cp
- rsync

Via SSH

- scp
- rsync

For very large transfers

- Grid-FTP
- bbcp

Copying – Local Copy

```
cp –r source /epsrc/gid/gid/destination
```

Copying to the mounted RDF filesystem exactly the same as a normal copy between directories.

```
rsync –r source /epsrc/gid/gid/destination
```

Pro: rsync will not attempt to transfer files that already exist.

Con: this “mirroring” requires a large number of metadata operations, slowing performance.

Recommend rsync over cp when resynchronising a previously copied directory containing large files.

Copying – SSH

For remote transfers DTNs should be used.

```
scp –r source user@dtn01.rdf.ac.uk:[destination]
```
Analogue of standard cp.

```
rsync –r –e ssh source user@dtn01.rdf.ac.uk:[destination]
```
Same utility for both local and remote transfers.

All traffic encrypted – secure but performance penalty.
Different ciphers can be used to improve speed.
Algorithm “arcfour” usually fastest but least secure:

```
scp –c arcfour …
rsync –e “ssh –c arcfour” …
```

Copying – Large Transfers – GridFTP

Very large transfers require specialised tools. These use multiple unencrypted socket connections for performance.

GridFTP transfers use the “globus-url-copy” utility.

Common options:

-r recursive

-vb display progress and transfer rate (-verbose-perf)

```
globus-url-copy -vb file:///full/path/to/local.file
sshftp://user@dtn01.rdf.ac.uk/full/path/to/destination/
```

Can alternatively authenticate with a personal grid-certificate to allow use of other protocols (gsiftp). See documentation on how to register via the SAFE.

Copying – Large Transfers – bbcp

Uses SSH authentication then transfers data using parallel unencrypted streams.
Needs to be installed at both source and destination (available as a module on ARCHER and the RDF).

Common options:

-s #	number of streams to use (default -s 4)
-P #	print progress every # seconds
-T <cmd>	command to launch bbcp on remote site
-z	reverse connection (useful for avoiding firewall problems)

```
bbcp –s 2 –T “ssh user@dtn01.rdf.ac.uk module load bbcp;
bbcp” source.file user@dtn01.rdf.ac.uk:destination.file
```

To transfer from the current host to the RDF – assuming relevant ports are open.

Summary

RDF mounted directly on ARCHER login nodes. DTNs available for remote transfers

Archiving improves performance. Be aware of metadata operation bottleneck.

More than one way to copy data. Additional security-performance trade off.

For advice contact: support@archer.ac.uk

Links and References

- Data Management Guide:
  - http://www.archer.ac.uk/documentation/data-management/
- UK-RDF Guide:
  - http://www.archer.ac.uk/documentation/rdf-guide/
- User Guide – ARCHER Filesystems:
  - http://www.archer.ac.uk/documentation/user-guide/resource_management.php#sec-3.3
- GridFTP tools:
  - http://toolkit.globus.org/toolkit/downloads/

Reusing this material

archer

epcc

THE UNIVERSITY OF EDINBURGH