Data Management

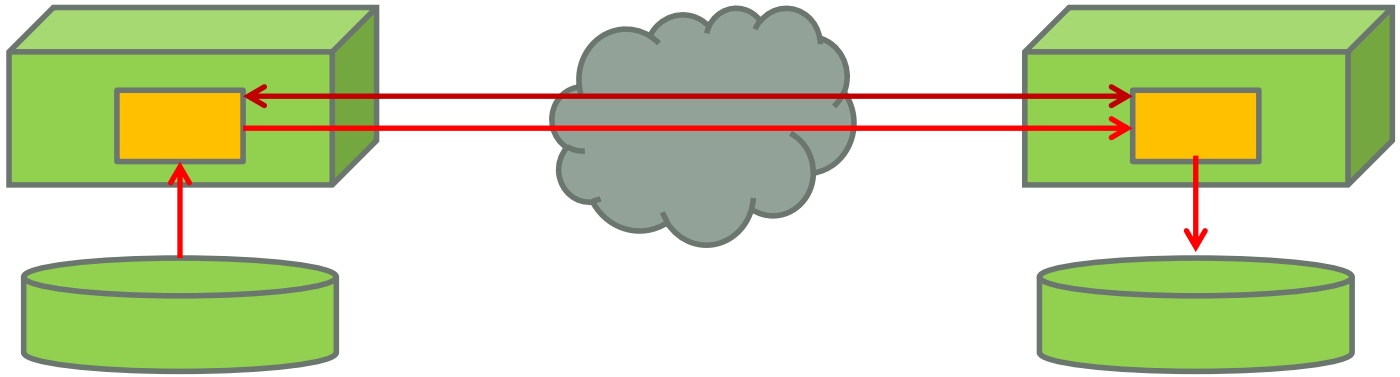
Network transfers

Network data transfers

- Not everyone needs to transfer large amounts of data on and off a HPC service
  - Sometimes data is created and consumed on the same service.
- If you do need to move large amounts of data, what is the best way of doing this?

Basic Architecture

- File transfers require a process on each participating machine
  - Control data names, permissions etc.
  - File data bytes of data.



File system performance

- Can't transfer data faster than file-system transfer rate.
- Unless you have a fast parallel file-system at both ends of the connection this is very likely to be a limiting factor.
- **dd** can give quick estimate of file system performance
- Note read/writes may differ.

```
spb@eslogin006:/work/z01/z01/spb> time dd bs=1M if=/dev/zero of=junk.dat count=4096
4096+0 records in
4096+0 records out
4294967296 bytes (4.3 GB) copied, 12.3631 s, 347 MB/s

real    0m12.835s
user    0m0.000s
sys     0m6.092s
spb@eslogin006:/work/z01/z01/spb> time dd bs=1M if=junk.dat of=/dev/null
4096+0 records in
4096+0 records out
4294967296 bytes (4.3 GB) copied, 1.04441 s, 4.1 GB/s

real    0m1.049s
user    0m0.000s
sys     0m1.040s
```

Disk caches

- Linux uses any otherwise unused RAM as a disk cache
- Repeated access to files in the cache will be served from RAM not disk.
- Perform any benchmarking using large dataset or you might be measuring cache speed not disk speed.
- This also applies to network transfer tests.

ssh based tools

- Common solutions is to build tools on top of ssh.
  - Remote process started via ssh
  - Control and Data sent via ssh connection
- Many tools do this:
  - scp
  - sftp
  - rsync
  - cpio

scp

- A “**cp**” like interface, all arguments passed on command line
  - Progress meter

```
-bash-4.1$ scp random_4G.dat dtn01:junk.dat
random_4G.dat                                 100% 3031MB 137.8MB/s   00:22
-bash-4.1$
```

sftp

- Command prompt interface
  - Allows remote file-system to be listed
  - Multiple operations without re-authenticating
  - Can execute batch files of transfers
  - Progress meter

```
 -bash-4.1$ sftp dtn01
Connecting to dtn01...
sftp> put random_4G.dat junk.dat
Uploading random_4G.dat to /general/z01/z01/spb/junk.dat
random_4G.dat                                 100% 3031MB  89.2MB/s   00:34
sftp>
```

rsync

- Directory synchronisation tool.
- Source or destinations locations in rsync can be on remote hosts.
- Possible metadata problems
  - -bash-4.1$ rsync -av data1 dtn01:data2
  - sending incremental file list
  - data1

  - sent 3178621906 bytes received 31 bytes 147842880.79 bytes/sec
  - total size is 3178233856 speedup is 1.00

Authentication

- SSH based tools can use passwords or “keys”
- Keys have 2 parts
  - Public
    - Install these in .ssh/authorized_keys to allow access to an account
    - Configures the “lock” to accept the key.
  - Private
    - Used from the remote host to gain access
    - Normally encrypted, you need to use a password to decrypt.

Best Practice

- Best practice is NOT to have your private keys on the HPC service
- SSH can forward key requests back through the login chain to your home system
  - -A flag on linux requests forwarding
- Need to run a ssh_agent on the home system
  - Only need to unlock key once at start of session
  - Alternative programs for windows “e.g. pageant”.
- See ARCHER user-guide for more detailed instructions.

Offline ssh access

- Secure use of SSH relies on interactive use.
  - User has to be present to decrypt private keys.
  - Ssh-agent holds decrypted keys in memory on users personal machine to reduce password prompts.
  - Makes it hard to use ssh from batch securely.
- It is possible to remove encryption from a ssh key.
  - However if file is lost it will continue to work as an access key until you delete the entry in **authorized_keys**
  - If you have to use ssh keys from a batch job:
    - Make a new key each time
    - Delete from **all** authorized_keys files once operation is complete.

Pros/Cons

- Pro
  - Works anywhere ssh connections are allowed.
  - Tools generally available on most systems.
  - Connections are encrypted, secure from intercept.
- Con
  - Connections are encrypted, high CPU utilisation, can limit performance.
  - Single socket connection, can limit performance.
  - SSH designed for interactive terminal connections, not always optimal for high data rates.
  - SSH authentication hard to use from batch without compromising security.

Encrypted connections

- Encryption/Decryption adds CPU overhead to the transfer and will limit performance.
  - Impact on performance depends on the speed of the CPUs at each end and the cipher that gets selected.

```
-bash-4.1$ dd if=/dev/zero bs=1M count=1024  | ssh -c 3des-cbc dtn01 dd of=/dev/null
1024+0 records in
1024+0 records out
1073741824 bytes (1.1 GB) copied, 63.7922 s, 16.8 MB/s

-bash-4.1$ dd if=/dev/zero bs=1M count=1024  | ssh -c arcfour dtn01 dd of=/dev/null
1024+0 records in
1024+0 records out
1073741824 bytes (1.1 GB) copied, 7.0445 s, 152 MB/s
```

  - For comparison the same network achieved 676 MB/s with an unencrypted socket.

Parallel SSH connections

- Limit is due to CPU overhead
  - And possibly due to implementation inefficiencies within ssh
- Multiple ssh connections should perform better
  - Provided file-systems can support this
  - Provided network can support this
  - Provided sufficient CPU cores at each end-point

Unencrypted Data connections

- Dedicated data transfer tools tend to use unencrypted sockets to move data traffic
  - Control traffic usually still encrypted
- Most can use multiple socket connections in parallel as this gets better bandwidth in practice:
  - More parallelism in the file-system access.
  - Performance degrades better on congested networks.
  - Works-around some kinds of poor network configuration.
- Needs a range of “non-standard” ports opened in the firewalls.

Firewalls

- We open TCP ports 50000,52000 on the RDF Data-transfer nodes for use by file-transfer tools.
  - May (probably will) require some range open at the remote host as well depending on tool and direction of transfer.
  - Also any institutional/departmental firewalls on the data path.
  - Getting this set-up and working takes time PLAN AHEAD !!
- Security implications
  - Opening firewall ports only allows access to processes that are listening on those ports.
  - Standard file transfer tools only listen as part of a pre-authenticated user session so low risk.
  - Need to check that no system services are using this port range.
  - Need to monitor for misuse by internal users (e.g. file-sharing)
  - Manageable risk for well run HPC system but campus firewall rules have to assume poorly run machines so may default deny.

Network

- Many people assume file transfer is always network limited
  - Most standard network ports are at least 1Gb/s = 125 MB/s
  - Modern servers/data centres: 10Gb/s, 40Gb/s = 1.25GB/s, 5GB/s
  - Janet6 core is 100Gb/s = 12.5 GB/s
  - Janet6 edge 10GB/s = 1.25 GB/s
- However speed is limited by narrowest point.
  - Firewalls may be unable to process traffic at full-speed (especially if they have a large rule-set)
- Network Congestion will reduce this further
  - Though this should vary with time. Consistent poor performance suggests some other problem.

Private networks

- Can set up dedicated private networks to peer sites
  - Avoids network congestion
  - Often fewer routers/firewalls to traverse.
  - Sometimes reliable low performance more useful than high variability.
- Two such networks on ARCHER
  - PRACE 10Gbps
  - JASMIN 2Gbps
- Connected to RDF Data Transfer Nodes
  - Can be tricky to ensure tools use the “right” network

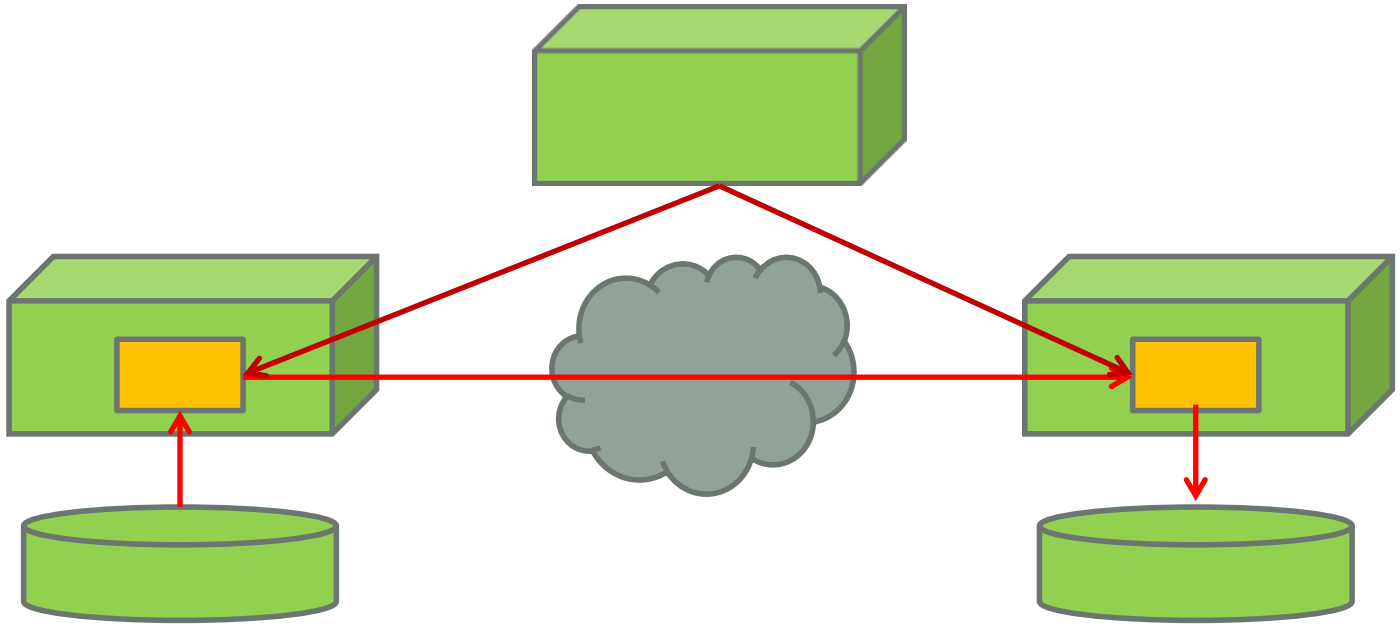
“bb” tools

- File transfer tools developed by the “BaBar” HEP collaboration
  - bbcp
  - bbftp
- Similar to **scp sftp** except that the underlying ssh connection is only used for authentication and control
  - Data moved using parallel unencrypted sockets.

gridFTP

- Very powerful and flexible file transfer mechanism
  - Part of the GLOBUS toolkit.
  - Various clients e.g. **globus-url-copy**
  - Uses parallel unencrypted data sockets (optionally encrypted)
  - Encrypted control path.
- Normally uses GSI certificate based authentication.
  - Short lived proxy certificates safer to embed in batch jobs or portals.
  - Can be configured to be started via ssh instead.
- Supports 3rd party transfers
  - Data transferred directly between 2 remote servers

Third party transfers



Certificate Authentication

- Proxy Certificates allow delegation
  - Temporary credential “signed” using users private key.
  - Have built-in expiry time.
  - Embed file transfer into batch jobs or Web portals like globus-online
- Myproxy service
  - “Drop-box” for certificate proxies
  - Can issue certificates if tied to other login system.
- Many users (and service operators) found infrastructure to issue and validate personal certificates troublesome for casual use.
  - Globus-online can use per-service certificates issued by myproxy (GCS)

gridFTP on the RDF

- RDF Data Transfer Nodes (dtn01 and dtn02) are configured with gridFTP servers
  - Uses personal Grid certificates
  - Register your certificate DN via the SAFE
- Also configured for ssh initiated gridFTP
  - Only needs ssh authentication but remote system still needs gridFTP tools installed.

Using a personal certificate

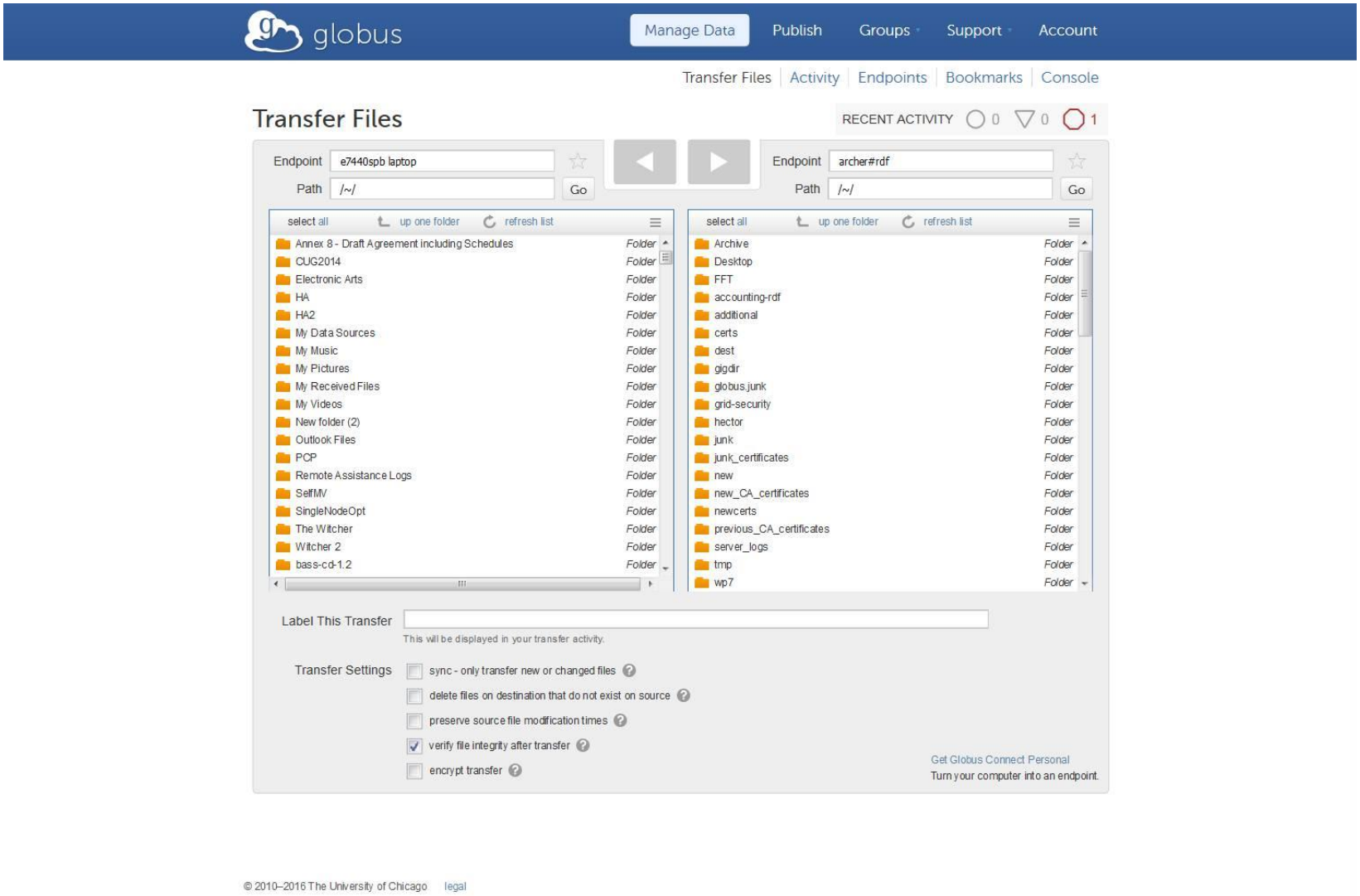
- Add your certificate DN to you account via SAFE
  - Store certificate in **.globus/usercert.pem .globus/userkey.pem**

```
-bash-4.1$ grid-proxy-init
Your identity: /C=UK/O=eScience/OU=Edinburgh/L=NeSC/CN=stephen booth
Enter GRID pass phrase for this identity:
Creating proxy ..................................... Done
Your proxy is valid until: Sat Feb  7 01:43:08 2015

-bash-4.1$ globus-url-copy -vb file:///general/z01/z01/spb/random_4G.dat
gsiftp://dtn02.rdf.ac.uk/general/z01/z01/spb/copy.dat
Source: file:///general/z01/z01/spb/
Dest:   gsiftp://dtn02.rdf.ac.uk/general/z01/z01/spb/
  random_4G.dat  ->  copy.dat

   3129999360 bytes       687.05 MB/sec avg       789.00 MB/sec inst
```

Using ssh authentication

- Can also use ssh based authentication

```
[spbooth@jasmin-xfer1 ~]$ globus-url-copy -vb
                    sshftp://spb@dtn01.rdf.ac.uk/general/z01/z01/spb/random_4G.dat
                                          file:///home/users/spbooth/random_4G.dat
Source: sshftp://spb@dtn01.rdf.ac.uk/general/z01/z01/spb/
Dest:   file:///home/users/spbooth/
  random_4G.dat

   3157262336 bytes        30.72 MB/sec avg        13.50 MB/sec inst
```



archer

epcc

THE UNIVERSITY OF EDINBURGH

Globus online on the RDF

- Public endpoint available on the RDF
  - Search for **archer#rdf**
  - Hosted on dtn03.rdf.ac.uk
  - Currently only mounts /general
- Activate the end-point using your RDF login credentials
  - Your browser will be redirected to dtn03.rdf.ac.uk to provide these
    - Web-server uses a UK eScience certificate so you may get warnings unless you install the A certificates from: http://www.ngs.ac.uk/ukca/certificates/cacerts
  - GO will retrieve a temporary (default 7 days) certificate to access file-system

Command-line access

- Can also access endpoint directly from command-line
  - Useful for off-line access from a script
  - Need to import CA certificates from dtn03

```
-bash-4.1$ myproxy-get-trustroots -s dtn03.rdf.ac.uk
Trust roots have been installed in
/general/z01/z01/spb/.globus/certificates/.
```

- These override the normal CA set so delete the directory when finished.
- Use myproxy-login to create the proxy-certificate

Making the proxy

- -bash-4.1$ myproxy-logon -s dtn03.rdf.ac.uk -l spb
- Enter MyProxy pass phrase:
- A credential has been received for user spb in /tmp/x509up_u5018.
- -bash-4.1$ globus-url-copy -vb -p 4 file:///general/z01/z01/spb/random_4G.dat gsiftp://dtn03.rdf.ac.uk/general/z01/z01/spb/junk.dat
- Source: file:///general/z01/z01/spb/
- Dest: gsiftp://dtn03.rdf.ac.uk/general/z01/z01/spb/
- random_4G.dat -> junk.dat
- 3178233856 bytes 704.88 MB/sec avg 704.88 MB/sec inst

Useful resources

- Netsite http://netsight.ja.net/
  - Public monitoring of janet network status
- Janet High Throughput Networking SIG
  - https://community.ja.net/groups/high-throughput-networking-special-interest-group
  - Quiet recently but useful content

Reusing this material

archer

epcc

THE UNIVERSITY OF EDINBURGH