Basic/default Connection

Connecting to:

Cloud10 / ARC3/4:

1) ssh -Y username@remote-access.leeds.ac.uk
   Password:
   Duo two-factor login for username@leeds.ac.uk
   Passcode or option (1-2): 1
   Success. Logging you in...
2) ssh -Y foe-linux-04
   username@foe-linux-04's password:
3) ssh -Y cloud10 / ssh -Y arc4.leeds.ac.uk

Running ParaView on cloud10:

1) paraview & (no MPI and no job submission, although can also use pvserver if desired)

Running ParaView on ARC4:

0) module add mesa paraview

Parallel:

1) qrsh -cwd -V -l h_rt=00:20:00 -l nodes=1 pvserver --server-port=11111 &
   [1] 191362
   Quite often ARC4 is busy, and will print this message:
   [username@login1.arc4 ~]$ Your "qrsh" request could not be scheduled, try again later.
   At this point, waiting for when it becomes less busy is one option, but can also try the serial approach below (or Advanced Connection).

Serial:

1) qrsh -cwd -V -l h_rt=00:20:00 -l h_vmem=10G pvserver --server-port=11111
   Waiting for client...
   Connection URL: cs://1234567.arc4.leeds.ac.uk:11111
   Accepting connection(s): 1234567.arc4.leeds.ac.uk:11111

Next, MUST set-up ssh-tunnel!

Open new terminal:

1) ssh -L 11111:1234567:11111 login1.arc4.leeds.ac.uk

NOTE: 1234567 is a connection ID given automatically, see Connection URL above (just copy that)

Open ParaView, connect as usual, details here:
https://arcdocs.leeds.ac.uk/software/applications/paraview.html

Connecting to and running ParaView server on:

ARCHER2:

1) ssh username@login.archer2.ac.uk
2) Apply for SHORTQOS (maximum 20 minutes 00:20:00) free
   srun --nodes=1 --exclusive --time=00:20:00 --partition=standard --qos=short --reservation=shortqos --pty /bin/bash
   or if needed for a longer interactive session:
   srun --nodes=1 --exclusive --time=00:50:00 --partition=standard --qos=standard --pty /bin/bash
   run ParaView interactively:

```
username@nid004526:/tmp>
cd /work/e710/e710/username/
module load paraview/5.10.1
srun --oversubscribe -n 32 pvserver --mpi --force-offscreen-rendering
Waiting for client...
Connection URL: cs://nid001023:11111
Accepting connection(s): nid001023:11111
```

Next, MUST set-up ssh-tunnel!

Open new terminal:

1) `ssh -L 11111:nid001023:11111 username@login.archer2.ac.uk`

Open ParaView, connect as usual (details here: https://docs.archer2.ac.uk/data-tools/paraview/)

Advanced Connection

ARC3/4:

For test modules, run:

```
module add test
```

```
then module avail -t paraview
```

Interactive job request:

1) qrsh -l h_rt=1:00:00,h_vmem=16G -pe smp 8
   [1] 191362
   [username@login1.arc4 ~]$ Your "qrsh" request could not be scheduled, try again later.
   Can also apply for a GPU node:
   {GPU NODE:
   qrsh -l h_rt=0:20:00,coproc_k80=1
   }

Running ParaView on ARC4 (WITH MPI)

2) module swap openmpi intelmpi
3) module add mesa/18.3.6 paraview-osmesa/5.3.0

MPI:

4) mpirun pvserver --mesa --server-port=11118 --use-offscreen-rendering
   Waiting for client...
   Connection URL: cs://d11s3b2.arc4.leeds.ac.uk:11111
   Accepting connection(s): d11s3b2.arc4.leeds.ac.uk:11111

Next, MUST set-up ssh-tunnel!

Open new terminal:

5) ssh -L 11111:d10s0b1:11111 login1.arc4.leeds.ac.uk

File Transfer

feng: vglrun /usr/not-backed-up/Paraview/bin/paraview

Copying:

ARC4

PC → ARC4:

rsync -Pv -e"ssh -c aes128-gcm@openssh.com -i path/to/ssh/rsa/file" ./file.py username@arc4.leeds.ac.uk:/destination/

ARC4 → PC

rsync -Pv -e"ssh -c aes128-gcm@openssh.com -i path/to/ssh/rsa/file" username@arc4.leeds.ac.uk:/source .

ARCHER2 → ARC4

Repeatedly poll file transfer:

[username@login1.arc4 ~]$ for i in {1..20}; do ls -lt; date ; sleep 600; done

(source: for i in {1..10}; do echo -n "This is a test in loop $i "; date ; sleep 5; done

1. http://paulsoftech.blogspot.com/2016/02/to-run-command-repeatedly-in-linux.html
2. https://www.tecmint.com/run-repeat-linux-command-every-x-seconds/

scp -3r username@login.archer2.ac.uk:/src username@arc4.leeds.ac.uk:/dst/

source:

1. https://serverfault.com/questions/449705/why-is-it-not-possible-to-use-two-remotes-for-rsync

ARCHER2

ARCHER2 → PC:

scp username@login.archer2.ac.uk:/src .

PC → ARCHER2:

rsync -Pv -e"ssh -c aes128-gcm@openssh.com -i path/to/ssh/rsa/file" ./animation_loader.py username@login.archer2.ac.uk:/work/e710/e710/username/

feng-linux

ARC4 → feng (run from feng terminal)

rsync -Pv -e"ssh -c aes128-gcm@openssh.com -i path/to/ssh/rsa/file" username@arc4.leeds.ac.uk:/src /dest

feng → ARC4 (run from feng terminal)

rsync -Pv -e"ssh -c aes128-gcm@openssh.com -i path/to/ssh/rsa/file" /src/ username@arc4.leeds.ac.uk:/dest

archer2 → feng-linux :

rsync -Pv --verbose -e"ssh -c aes128-gcm@openssh.com -i path/to/ssh/rsa/file" username@login.archer2.ac.uk:/src /dest/

How to use GPU on ARC4 from IT support:

Paraview on ARC4, using GPUs.

Assuming there was a free GPU:

Login to ARC4:

```
module add test paraview-egl
qrsh -V -cwd -l coproc_v100=1,h_rt=1:0:0 -pty y pvserver -p 11111
```

It should output something like:
Waiting for client...
Connection URL: cs://db04gpu3.arc4.leeds.ac.uk:11111
Accepting connection(s): db04gpu3.arc4.leeds.ac.uk:11111

Once that's running, do a separate connection on your local device to make the tunnel to match the host you're allocated. In my case this is:

```
ssh -NL 11111:db04gpu3:11111 arc4.leeds.ac.uk
```

Then run on your local device, paraview to use that tunnel to talk to the remote paraview:

```
paraview --url cs://localhost:11111
```

You know that's connected as you should get "Client connected." appearing on your ARC session.

If you look in Help/About Paraview/Connection Information, it shows that you're connected to a system that's using a Tesla V100 for the rendering with EGL, which all looks good.