Gridengine Basics

James Owers

27 June 2016

Contents

1 Who is this for **1**

1.1 Useful references 1

2 Getting started **2**

3 Basic SGE commands **2**

3.1 Interactive session on a node (just like `ssh`ing) 2

3.2 Submit a script to the queue 3

3.3 View status of your subitted jobs 3

3.4 Deleting Jobs 3

3.5 Viewing Node Status 3

4 Example: Running an IPython Notebook and accessing it from outside DICE **4**

5 Current issues **4**

6 Tips **4**

TL;DR: To use `James` or `Charles` servers as if you were `ssh`ing into them as before, just `ssh renown` then `qlogin`.

1 Who is this for

People who use the `James` or `Charles` servers. Until now we have `ssh`'d into the servers but now `ssh` access has been removed from all but a few. Now in place is Son of a Grid Engine (`SGE`) to control access to servers. This guide shows you how to continue much like before and how to use basic `SGE` commands.

Son of a Grid Engine is an open source version of Univa Grid Engine (née Oracle Grid Engine (née Sun Grid Engine))

1.1 Useful references

- SGE project site
- SGE documentation
- `man qsub` from within `renown`
- MIT SGE introduction

2 Getting started

Log in to the gridengine machine `renown`

```
## If not on a dice machine
kinit s0816700
aklog
ssh -K s0816700@staff.ssh.inf.ed.ac.uk
# ssh -K s0816700@student.ssh.inf.ed.ac.uk
ssh renown
```

You will be in your home directory, in my case, `/home/s0816700`. We can see that lots of space has been added to the `/home/` directory (`/mnt/cdt_gridengine_home`)

```
df -h
```

```
Filesystem                                     Size  Used Avail Use% Mounted on
/dev/vda1                                       24G  5.1G   18G  23% /
devtmpfs                                       2.0G     0  2.0G   0% /dev
tmpfs                                          2.0G     0  2.0G   0% /dev/shm
tmpfs                                          2.0G  9.4M  2.0G   1% /run
tmpfs                                          2.0G     0  2.0G   0% /sys/fs/cgroup
/etc/glusterfs/gv0.vol                         147G   84G   57G  60% /disk/glusterfs/gv0
charles11.inf.ed.ac.uk:/cdt-gridengine-common  385G  264M  365G   1% /mnt/cdt_gridengine_common
anne.inf.ed.ac.uk:/cdt-gridengine-home         2.7T  432G  2.2T  17% /mnt/cdt_gridengine_home
/dev/vda4                                      7.6G   65M  7.1G   1% /var/cache/afs
AFS                                            2.0T     0  2.0T   0% /afs
tmpfs                                          396M     0  396M   0% /run/user/656624
tmpfs                                          396M     0  396M   0% /run/user/28328
tmpfs                                          396M     0  396M   0% /run/user/1559549
tmpfs                                          396M     0  396M   0% /run/user/1421660
```

Not covered here: how to run parallel jobs and writing to the distributed file system 'Gluster'. For information on running parallel jobs using SGE, see the latter half of the MIT SGE introduction.

3 Basic SGE commands

3.1 Interactive session on a node (just like sshing)

```
qlogin
```

Useful options:

- specify a specific node

  ```
  qlogin -l h=charles14
  ```

- specify resource must have a GPU

```
qlogin -l gpu=1
```

3.2 Submit a script to the queue

```
qsub myscript.sh
```

OUTPUT two files containing the stdout and sterr [script-name].o[jobnr] and [script-name].e[jobnr], and whatever files or directories your script creates

3.3 View status of your subitted jobs

```
qstat
```

OUTPUT

```
job-ID  prior   name       user         state submit/start at     queue                          slots
--------------------------------------------------------------------------------------------------------
     15 0.55500 long_sleep s0816700     r     06/03/2016 22:54:38 all.q@charles11.inf.ed.ac.uk        1

state = *qw*/**r** for *queued and waiting*/**running**
```

3.4 Deleting Jobs

```
qdel
```

3.5 Viewing Node Status

qhost

OUTPUT

```
HOSTNAME                ARCH         NCPU NSOC NCOR NTHR  LOAD  MEMTOT  MEMUSE  SWAPTO  SWAPUS
----------------------------------------------------------------------------------------------
global                  -               -    -    -    -     -       -       -       -       -
anne                    lx-amd64       64    4   64   64  0.02  995.6G    8.5G   31.2G     0.0
charles01               lx-amd64       32    2   16   32  1.01   62.7G    8.3G   31.3G     0.0
charles02               lx-amd64       32    2   16   32  0.27   62.7G    3.7G   31.3G     0.0
charles03               lx-amd64       32    2   16   32  0.01   62.7G    3.2G   31.3G     0.0
charles04               lx-amd64       32    2   16   32  0.04   62.7G    2.5G   31.3G     0.0
charles05               lx-amd64       32    2   16   32 13.61   62.7G    6.0G   31.3G     0.0
charles06               -               -    -    -    -     -       -       -       -       -
charles07               -               -    -    -    -     -       -       -       -       -
charles08               -               -    -    -    -     -       -       -       -       -
charles09               -               -    -    -    -     -       -       -       -       -
charles10               -               -    -    -    -     -       -       -       -       -
charles11               lx-amd64       24    2   12   24  0.01   62.8G    2.6G   31.4G     0.0
charles12               lx-amd64       24    2   12   24  0.01   62.8G    2.4G   31.4G     0.0
charles13               lx-amd64       24    2   12   24  0.01   62.8G    2.6G   31.4G     0.0
charles14               lx-amd64       24    2   12   24  0.01   62.8G    2.6G   31.4G     0.0
```

4 Example: Running an IPython Notebook and accessing it from outside DICE

1. Setup python virtual environment with IPython Notebook installed
    - Tip: install it in your home directory on DICE
2. `qlogin` to your server of choice
3. Check GPU use with `nvidia-smi`
4. activate your python virtual environment (you'll need to `kinit` & `aklog` if this is located on your DICE home as recommended)

    ```
    source /afs/inf.ed.ac.uk/user/s08/s0816700/venv/nolearn/bin/activate
    ```

5. create a password hash using python

    ```
    from IPython.lib import passwd
    passwd()
    exit
    ```

6. start the IPython Notebook

    ```
    longjob -28day -c 'ipython notebook  --ip="*" --NotebookApp.password=sha1:0880f873e98f:9ddab235858
    ```

7. access the notebook
    - From within forum simply browse to `http://charles13:1337` (replacing charles13 with where you were)
    - Outside the forum either:
        - first kinit & aklog then, ssh port forward charles13:1337 back to your computer: `ssh -K -L 8889:charles13:1337 s0816700@staff.ss.inf.ed.ac.uk` then go to `http://localhost:8889`
        - or connect via VPN and navigate to `...` TODO: fill this in!

WARNING: If anyone finds/hacks your password…they have access to your filesystem

AWESOME WIN: this longjob process allows continual access to your filesystem after the original afs ticket expires

5 Current issues

- Automatic resource allocation doesn't appear to take into account GPU use…
- …working with Charles and Iain Rae on that
- IPython Notebook solution isn't very secure
- If you are running a script containing `longjob` using `qsub`, how is your kerberos ticket handled

6 Tips

- when logged in to `renown` type `q` then double tap `tab` to get a list of the commands for use!
- `nvidia-smi` lets you check the GPU use on a server - if the command doesn't work then the server you are on doesn't have a GPU; try to login to another server; you can specify a specific server with `qlogin -l h=charles14`