NCC DOCUMENTATION

System Information

The NVIDIA CUDA Centre GPU system is a shared Dell System with 2 NVidia GPUS. The machine is housed in the ECS server room. Support requests should be sent to iarc@durham.ac.uk

Please note that this system does not come with large persistent storage and files should be transferred in when starting a job and then out after the job. The **file store is not backed up**. Due to reconfigurations taking place in the ECS server room the system is **currently NOT on. UPS**.

System Specs

1x	Dell PowerEdge R730
2x	Intel Xeon E5-2650 v3 2.3GHz,25M Cache,9.60GT/s QPI, Turbo, HT, 10C/20T
2x	NVIDIA K40c
8x	16GB RDIMM
1x	300GB SAS Drive

Running Jobs on NCC.IARC.DURHAM

The NVIDIA CUDA SERVER uses TORQUE and MAUI to schedule jobs on the system. The TORQUE Job Description Language (JDL) is compatible with PBS, Torque, OpenPBS and PBS Pro Batch Queuing and Scheduling Systems. Users familiar with LSF and SGE may find the schema similar. This JDL schema is also used on the UK national resource ARCHER. For the purposes of this wiki the terms TORQUE and PBS will be used interchangeably both referring to the same Batch Queuing System.
To use PBS two major components need to be dealt with, the JDL Schema and the Command Line tools.

ALL ACTIVITY MUST BE THROUGH THE PBS SYSTEM. THIS ENSURES FAIRUSE

Job Description Language

When scheduling a job/simulations/render through PBS a script needs to be written which the job scheduler parses for instructions to itself and then passes the remaining instructions to the command line for execution once the Job starts. This behaviour is one of the major reasons PBS is the preferred job scheduler on the QGG. PBS scripts can be written in any terminal scripting language. As 'BASH' is the default shell environment in the QGG this tutorial will use bash syntax.

Recommended Headers

Below is a sample header section of a PBS Job Script. The comments at the end of the line explain what each statement does.

```
#!/bin/bash                    # This line is required to inform the
                               Linux #command line to parse the
                               script using #the bash shell
#PBS -l nodes=X:ppn=Y          # Instructing PBS to locate and assign
                               #X number of nodes with Y number of
                               #cores in each node.
                               # X,Y are integers. Refer to table for
                               #various combinations
#PBS -q “queuename”            # Governs the run time limit and
                               #resource limit for the job. Refer to
                               table #for various combinations
```

The lines from the script above are required to run any job. The first line creates the user environment for the job. The next two are used to schedule and allocate resources to the job. There are two queue types on the NCC system: *debug* and *batch*.
The debug queue allows jobs running up to 2 hours only. If the system is busy debug jobs will be run through faster. The second queue is the batch queue. This allows for jobs to run for up to 48 hours. These jobs have a lower priority. Shorter run times can be selected by using the command:

```
#PBS -l walltime=HH:MM:SS
```

The ‘-l’ switch requests the total resources required. To book the whole node users should select X=1 and Y=20. If users have smaller jobs and are more interested in throughput then Y can be selected as Y<20. However it is recommended that if running GPU based jobs please select the full node i.e. Y=20.

Other Optional Definitions

There are a few other JDL instructions a user can input to better utilise the resources and to stream line their work.

```
#PBS -e stderr-filename
#PBS -o stdout-filename
#PBS -j oe
```

The previous 3 options allow a user to specify a special file name for the standard error (-e) and for the standard output (-o). These can be set to appear as a more useful name but it is not recommended as on each run it will overwrite the previous logs unless the file name is changed. By default it uses the job number and appends an extension 'err' for the standard error and 'out' for the standard output. The (-j) option allows the user to combine both files into one - which helps reduce the number of files.

```
#PBS -N jobnamehere
```

Assign a name to the job. Helps locate it in the queue or on email notification. Cannot start with a number or contain spaces.

After fully adding the headers normal bash commands can be used to define the steps the scheduler needs to take. (see example script at the bottom of this page).

CLI Tools

Once the job script is prepared it has to be passed on to the job scheduler. There are various command line tools, which allow the user to submit, delete and check the status of jobs and queues.

1. `[username@ncc ~]$ qsub jobscript.jdl`
The `qsub` command submits a JDL file to the PBS system. QSUB takes as an argument the job description language scripts and any headers from the JDL file. (Please see Interactive PBS Jobs for more information).
2. `[username@ncc ~]$ qstat`
The qstat command shows the status of all the queues running on the system. `qstat -h` gives further options to help refine the output.
3. `[username@ncc ~]$ qdel`
To delete a job from the system the qdel command can be used. Note only jobs submitted by you can be deleted using the qdel command. If you feel a job is stuck and not responding to the qdel command please contact IARC.
4. `[username@ncc ~]$ pbs_status`
The pbs_status commands can be used on the head nodes of the clusters to see which nodes are down,available or locked with jobs.

Sample Job Script

This job script below requests the full node from PBS. The job is called HiThere. The job runs in the debug queue. The standard error and output are combined in one file. The program then loads the NVIDIA CUDA libraries changes to the working directory and runs the helloworld program, dumping the output in the myfirstoutput.txt file.

```
#!/bin/bash
#PBS -l nodes=2:ppn=2
#PBS -q parasrt
#PBS -N HiThere
#PBS -j oe

module load nvidia

cd PBS_O_WORKDIR
mpirun -np 4 -machinefile $PBS_NODEFILE hello_world >
myfirstoutput.txt
```