Notes on CUDA practicals on ARC HTC cluster

Mike Giles

1 The ARC HTC cluster

The layout of the ARC HTC (high throughput computing) cluster is shown in Figure 1.

The login node which sits on the university network is `htc-login.arc.ox.ac.uk`. This can be accessed from machines on the university network through the SSH command:

```
ssh -X username@htc-login.arc.ox.ac.uk
```

The `-X` option provides X-window forwarding which is convenient when using windows-based editors such as `emacs`.

The nodes `htc-g045` to `htc-g049` each have 8 V100 GPUs.

2 ARC User Guide

The ARC User Guide is available at https://arc-user-guide.readthedocs.io/en/latest/

3 CUDA

To use CUDA on ARC HTC you need to enter the command:

```
module load CUDA
```

Amongst other things, on the compute nodes this will set your `PATH` and other environment variables so that you can use the NVIDIA compiler `nvcc` and link in the various libraries, and access the relevant header files.

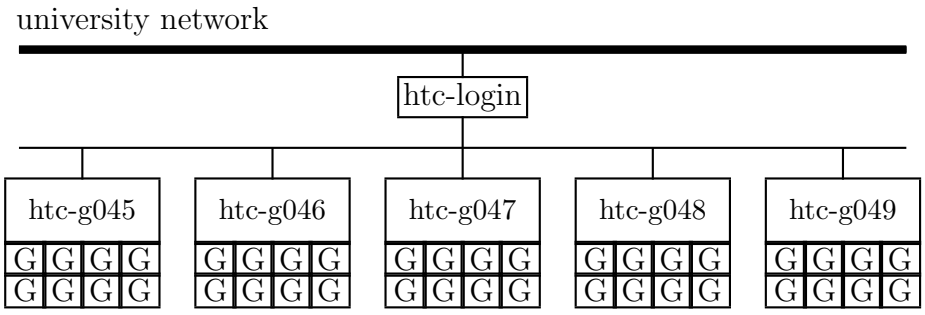

Figure 1: ARC HTC cluster with login node and GPU compute nodes

4 Editing and file transfer

`emacs`, `vim` and `nano` are all available on the ARC HTC login node – see next page for info on using `vim` and `nano`.

The alternative is to edit files on your laptop and move them to, and from, ARC using `scp` commands such as:

```
scp local_file username@htc-login.arc.ox.ac.uk:remote_file
scp username@htc-login.arc.ox.ac.uk:remote_file local_file
```

See the next page for SCP applications for Windows users.

5 Doing the practicals

A tar file for all of the practicals can be obtained by using the command:

```
wget https://people.maths.ox.ac.uk/gilesm/cuda/practicals.tar.gz
```

and then untarred using the command:

```
tar zxvf practicals.tar.gz
```

For each practical you need to compile the code using the supplied Makefile and run it. Both parts of this have to be done on one of the GPU compute nodes using a batch script with a command of the form:

```
sbatch job
```

where `job` is the name of a job submission script which is provided for the first few practicals.

You should see the slurm batch management system return something like the following:

```
sbatch:  GPU gres requested, checking settings/requirements...
Submitted batch job 3051862
```

Once the job has executed there will be output from slurm, which in the case of batch job `3051862` will be in the file `slurm-3051862.out`

More information on job submission scripts is given in the ARC User Guide and an example is included on the last page.

6 Links for help on Linux commands and editors

Intro to Linux commands for Windows users

https://kinsta.com/blog/linux-commands/

https://www.geeksforgeeks.org/linux-vs-windows-commands/

https://www.hostinger.co.uk/tutorials/linux-commands

https://blog.robertelder.org/intro-to-linux-command-line-for-windows-users/

https://phoenixnap.com/kb/linux-commands

nano editor

https://www.nano-editor.org/

https://www.nano-editor.org/dist/latest/cheatsheet.html

https://help.ubuntu.com/community/Nano

vim editor

https://opensource.com/article/19/3/getting-started-vim

https://opensource.com/sites/default/files/gated-content/cheat_sheet_vim_final_v2_0.pdf

7 Other tools for Windows users

Some Windows users like MobaXTerm which combines a remote terminal, file copying and X11 server in one (free) product.

WinSCP is another popular alternative for copying files to/from remote servers.

8 An example slurm batch script

```
#!/bin/bash
# set the number of nodes and processes per node
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --partition=short

# set max wallclock time (hours:minutes:seconds)
#SBATCH --time=00:10:00
#SBATCH --gres=gpu:1

# set name of job
#SBATCH --job-name=prac1

# use our reservation
#SBATCH --reservation=cuda2025

# initialise module system then load CUDA module
module purge
module load CUDA

# get rid of old executables and make new ones
make clean
make

# run the code prac1a in the current working directory
./prac1a
```