Maxwell HPC

Accelerating Machine Learning
- Introductory Guide

Email: aiden.durrant@abdn.ac.uk

Written Guide & Examples: github.com/AidenDurrant/abdn-hpc

What is Maxwell?

Maxwell is a Linux supercomputing cluster housed in the Edward Wright Datacentre.

It is designed for running high computational workloads for simulation, modelling and training.

- NOT a development environment.
- NOT designed to host/interface model inference.

It is funded by University of Aberdeen and National Decommissioning Center and operated by external provider OCF.

1495 UNIVERSITY OF ABERDEEN

Resources Available

CPU (compute) Nodes:

- 11 x Compute Nodes

High Memory Nodes:

- 11 x High Memory Nodes
- 1 x Very High Memory Node (3TB)

GPU Nodes:

- 2 x 2080ti nodes (2 x 12GB RTX 2080ti per node)
- 4 x A100 nodes (3 x 80GB A100 per node)

1 Petabyte of tiered storage

- Each user 1TB

Wide array of HPC optimised software.

- Availability of environment managers

Graphical Interface

- Galaxy Server

1495 UNIVERSITY OF ABERDEEN

SLURM Workload Manager

Slurm is an open source, fault-tolerant, and highly scalable cluster management and job scheduling system for large and small Linux clusters.

Slurms Purpose:

1. Allocation of resources to users for some period of time.
2. Provides a framework for starting, executing, and monitoring work.
3. Manages a queue of pending work.

The short of it:

You interface with the home node, which in-turn communicates with compute nodes to execute your code.

1495 UNIVERSITY OF ABERDEEN

Accessing Maxwell

- Contact Digital Research to gain access to Maxwell
    - Note this is only accessible for PGR, Staff, and under any project where funding is allocated.
- For UG, and PGT the MacLeod server is available.
    - Follow the below link, and fill out the form in the quick links named “Access MacLeod”.

https://www.abdn.ac.uk/research/digital-research/hpc-for-research-1097.php

1495 UNIVERSITY OF ABERDEEN

VPN, SSH & SFTP

- The HPC services can be accessed via SSH when connected to the university network (not eduroam).
- Connecting to the university network can be achieved via the f5 VPN if not using a managed device.
  - https://www.abdn.ac.uk/staffnet/working-here/it-services/remote-access.php

Maxwell login nodes:

- Maxlogin1.abdn.ac.uk : port 22
- Maxlogin2.abdn.ac.uk : port 22

MacLeod login nodes:

- Macleod1.abdn.ac.uk : port 22
- Macleod2.abdn.ac.uk : port 22

1495 UNIVERSITY OF ABERDEEN

Directory Structure

Two key directories:

- home
- sharedscratch

<u>Home:</u>

Personal 50GB resilient, backed-up.

/uoa/home/<username>/

<u>Sharedscratch:</u>

1TB scratch space for working storage.

/uoa/scratch/users/<username>/

Storage	HPC Backup and Restore Policy
Home Space	Data is backed up as follows: ● Daily backups kept for 14 days< ● Weekly backups kept for 1 week Files can be restored as follows: ● To the path that folder/files came, or ● To a different specified HPC file path, or ● To a specified shared drive Restoration request process: ● Contact servicedesk@abdn.ac.uk
Shared scratch	Not backed up

1495 UNIVERSITY OF ABERDEEN

Data Management

- Transferring of data is best done via SFTP
  - WinSCP on managed devices.
- All code, data, and outputs should be stored in sharedscratch.
  - home should be use only for small quantities of critical data.
- It is recommended to use version control software for code management. (git - github/gitlab)

Maxwell storage is not optimised for speed rather for redundancy.

- Preprocess prior if possible.
- Store data in more efficient formats (i.e. HDF5)
- Avoid multiple jobs accessing same data.

ALWAYS CHECK WHAT YOU ARE WRITING TO DISK BEFORE RUNNING!

Software

Maxwell is pre-installed with a number of HPC optimized software.

- A list of available software can be found at: https://www.abdn.ac.uk/it/documents-uni-only/Maxwell-Galaxy-Software.pdf
- To utilize software when running a job simply load it via:

```
module load <name_of_software>
```

- You can load multiple software packages, and unload via:

```
module unload <name_of_software>
```

- To request any additions to the software list contact the digital research team.

Creating Environments (1/2)

In may cases it can be easier to install your own packages in a virtual environment

For data science workflows that employ Python or R, Anaconda is available.

- Create personal environments.
- Install any python packages and libraries via conda.
- Manage different environments for different projects.

Creating Environments (2/2)

1. module load miniconda3
2. conda create -n <your_env_name> python=3.11

Note: For first time installs you will be prompted to configure your shell, run the command `conda init bash` and then exit / close the terminal, ssh back into maxwell and load the miniconda3 module to continue.

3. conda activate <your_env_name>
4. conda install pytorch torchvision torchaudio cudatoolkit=11.7 -c pytorch

Note: The current installed version of cuda is 11.7 on maxwell.

1495 UNIVERSITY OF ABERDEEN

Requesting Resources (1/3)

Unlike your standard desktop computer, you instead must submit a "job" for execution to the SLURM scheduler.

Some useful commands are as follows:

- `sinfo` - Show summary information
- `squeue` - Show job queue
- `squeue -u <username>` - Show job queue for a specific user
- `scontrol show partition a100_full` - Show gpu partition "a100_full" details
- `scancel <job_ID>` - Cancel a job

1495 UNIVERSITY OF ABERDEEN

Requesting Resources (2/3)

To run a job (run your program) on Maxwell you must submit a script to the SLURM scheduler.

The SLURM script must contain 3 things:

1. Define the resource requirements for the job.
2. Activate the environment we created earlier.
3. Specify the script we wish to run.

```
#!/bin/bash
#SBATCH --nodes=1 # number of nodes
#SBATCH --cpus-per-task=12 # number of cores
#SBATCH --mem=32G # memory pool for all cores

#SBATCH --ntasks-per-node=1 # one job per node
#SBATCH --gres=gpu:2 # two gpus per node
#SBATCH --partition=a100_mig

#SBATCH -o slurm.%j.out # STDOUT
#SBATCH -e slurm.%j.err # STDERR

#SBATCH --mail-type=ALL
#SBATCH --mail-user=<username>@abdn.ac.uk

module load anaconda3
source activate test

srun python example_script.py --epochs=25 --save /home/<username>/sharedscratch/
```

Requesting Resources (3/3)

- The partition specifies which compute nodes you will use, e.g. hmem, cgpu, a100, etc.
- To submit the previous job script named `run_example.sh` to the Slurm scheduler we use the `sbatch` command:

  `sbatch run_example.sh`

- The job may run immediately or may take upto a few days to run depending on the number of existing jobs, you will receive an email notification if you use the SLURM command `email`.
- To check the status of queued and running jobs, use the following:

  `squeue -u <username>`

- If you wish to cancel a job simply use:

  `scancel <job_ID>`

Other SLURM Settings

SLURM arrays:

A slurm array batch job is similar to just running a ‘for’ loop over the sbatch script.

- All runs have the same jobid with a predictable id as a suffix.
- #SBATCH --array=1-10

```
srun python addone.py $SLURM_ARRAY_TASK_ID
```

#SBATCH Commands:

A full list can be found here: https://slurm.schedmd.com/sbatch.html

Monitoring Jobs

Monitoring jobs can most simply be achieved by printing the logs specified in the run script.

```
#SBATCH -o slurm.%j.out
#SBATCH -e slurm.%j.err
```

This produces two logs per submitted job:

- STDOUT
- STDERR

Many other web/cloud monitoring tools exist (ML focused):

- Weights and Biases - https://wandb.ai
- Neptune.ai - https://neptune.ai

Wall Clock Time

The duration for which the nodes remain allocated to you.

- It is important to note that all jobs have a maximum walltime of 24 hours.

```
#SBATCH -t 24:00:00
```

If your job requires computation that is longer than 24 hours, you must handle it within the code/script.

- Checkpointing
  - Save and Resume
- Detection of resume checkpoint
- Auto-Requeue SLURM

```
timeout 23h ./the_program
if [[ $? == 124 ]]; then
  scontrol requeue $SLURM_JOB_ID
fi
```

Distributed Computation

Using more than 1 GPU

- When requesting more than 1 GPU ensure your code is configured to utilize the additional GPU's
- By default PyTorch will only use 1 GPU.
- To use multiple GPUs on one node implement DataParallel.

https://pytorch.org/docs/stable/generated/torch.nn.DataParallel.html

Using more than 1 Node (*Advanced*)

- When requesting more than 1 node ensure your code is written with distributed computation.
- By default PyTorch will run separate processes on each node.
- To use multiple nodes, ensure your code communicates across nodes.

https://pytorch.org/tutorials/intermediate/ddp_tutorial.html

Additional Resources

- SLURM: https://slurm.schedmd.com/quickstart.html
- Interactive Allocation: https://engaging-web.mit.edu/eofe-wiki/slurm/srun/
- Jupyter Notebooks: https://nero-docs.stanford.edu/jupyter-slurm.html
- ChatGPT for SLURM
- University Information: https://www.abdn.ac.uk/research/digital-research/hpc-for-research-1097.php
- Written Guide: https://github.com/AidenDurrant/abdn-hpc

1495 UNIVERSITY OF ABERDEEN