Virtual Tutorial July 15th 2020

Kevin Stratford

http://www.cirrus.ac.uk/

Engineering and Physical Sciences Research Council

Contents

- Access mechanisms
  - Responsive / Regular
  - Technical assessments
- Using Cirrus
  - Logging in / modules / user environment
  - Submitting jobs via SLURM
- Using the new GPU resource
  - Hardware
  - Storage/network

Levels

- Tier One
  - National Service /Archer
  - To fulfill largest parallel compute requirements
- Tier Two
  - Cirrus / …
  - Intermediate / specialist requirements
- Tier Three
  - Institutional level facilities
  - Generally smaller

Mechanisms for access

- Standard responsive mode grants
  - Research proposal in EPSRC remit
  - CPU time as part of requested resources
- “Access to High Performance Computing”
  - Twice a year (next close is 4th September)
  - CPU time only (12 month calendar time)
- Other
  - Instant access (pump priming)
  - Commercial access

Technical assessments

- All modes of access
  - Require a technical assessment
  - Deadline before final closing date (7th August)
- We check what is proposed/required
  - Is technically possible, legally possible, …
  - … and reasonable for Cirrus
- Requested resources
  - Do the requested resources make sense?
  - Necessary and sufficient answer the scientific question?

Cirrus

“Phase I”

- First in use late 2016
- SGI/HPE cluster
- 282 Intel Xeon (Broadwell) nodes
- Per node: 2x18 cores and 256 GB
- Infiniband (FDR) network
- `/lustre` file system (total c. 400 TB)

“Phase II”

- Originally scheduled for Q2 2020
- Upgrade retains older components
  - 282 “standard” nodes still available
  - Also 2 “older” NVIDIA V100 nodes (Intel skylake)
- Upgraded operating system
  - New software available
  - Some older versions “removed”

Logging in to Cirrus

- Generate ssh public/private key pair

```
$ ls –la ~/.ssh
-rw-------  1 kevin  staff   3434 Jun 16 16:25 id_rsa_cirrus.ac.uk
-rw-r--r--  1 kevin  staff    738 Jun 16 16:25 id_rsa_cirrus.ac.uk.pub
```

  - Upload public key to SAFE
  - Request a new machine password

Using ssh

- Config file

```
$ cat ~/.ssh/config
...
Host cirrus* cirrus*.epcc.ed.ac.uk login.cirrus.ac.uk
     IdentityFile ~/.ssh/id_rsa_cirrus.ac.uk
     User kevin
     ControlMaster auto
     ControlPath ~/.ssh/sockets/%r@%h-%p
     ControlPersist 600
     ForwardX11 yes
```

- Problems

```
$ ssh –vvv login.cirrus.ac.uk
```

User environment

- Home directory

```
/lustre/home/project/userid
```

- For example,

```
/lustre/home/z04/kevin
```

- Subject to quota

```
$ lfs quota –hg z04 /lustre
Disk quotas for grp z04 (gid 37733):
Filesystem    used   quota   limit   grace   files  quota
/lustre/    10.03T      0k   12.6T       -  30298523      0
```

Moving data/code

- Use secure copy

```
$ scp source userid@login.cirrus.ac.uk:~/dest
```

- Use revision control

```
$ git clone https://github.com/my-repo.git
```

- Use rsync

```
$ rsync source userid@login.cirrus.ac.uk:~/dest
```

Note on default file permissions

```
$ touch my-file
$ ls -l my-file
-rw-r--r-- 1 kevin z04 0 Jul 14 19:51 my-file
```

- May wish to consider

```
$ umask 077
```

Cirrus

Module system

```
$ module avail
altair-hwsolvers/13.0.213    intel-cmkl-19
altair-hwsolvers/14.0.210    intel-compilers-18
...
gdal/2.1.2-intel             openfoam/v1912
...

$ module avail gcc
gcc/6.2.0  gcc/6.3.0(default)  gcc/8.2.0
```

Module load/unload

```
$ module list
Currently Loaded Modulefiles:
1) git/2.21.0(default)
2) epcc/utils

$ module load gcc
$ module list
[kevin@cirrus-login1]$ module list
 1) git/2.21.0(default)   3) gcc/6.3.0(default)
 2) epcc/utils
```

Queue system

- PBS in Phase I has been replaced by SLURM
  - Partitions (broadly, physical hardware)

```
$ sinfo
PARTITION    AVAIL  TIMELIMIT  NODES  STATE NODELIST
standard        up 4-00:00:00      1  down* r1i3n27
standard        up 4-00:00:00      4   resv r1i4n[8,17,26,35]
standard        up 4-00:00:00      1    mix r1i0n22
standard        up 4-00:00:00    249  alloc ...
standard        up 4-00:00:00     25   idle ...
gpu-skylake     up       10:00      2   idle r2i3n[0-1]
gpu-cascade     up       10:00     34   idle ...
```

Quality of Service (QoS)

- Limits for actual jobs
  - Quality of Service

```
$ sacctmgr show qos format=name,...
      Name     MaxWall     MaxTRESPU MaxJobsPU MaxSubmitPU
---------- ----------- ------------- --------- -----------
  standard  4-00:00:00                      20         500
      long 14-00:00:00                       5          20
commercial  4-00:00:00                      50         100
highprior+  4-00:00:00                      10          20
       gpu  4-00:00:00   gres/gpu=16        10          50
```

SLURM: what's in the queue

squeue

```
$ squeue

  JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
  27911  standard    job30    user1 PD       0:00      1 (QOSMaxJobsPerUserLimit)
  27910  standard    job20    user1  R       1:49      1 r1i0n22
...
28618_0  standard   array1    user2  R   16:58:27      1 r1i2n11
28618_1  standard   array2    user2  R   16:58:27      1 r1i2n12
...
  28897  standard     mpi1    user3  R    8:29:08      4 r1i1n[25-28]
...
```

squeue -u userid

SLURM: submitting jobs

```
sbatch

$ sbatch [options] my-script.sh
Submitted batch job 28687

$ scancel 28687
```

SLURM submission script

```
#!/bin/bash

#SBATCH --nodes=1
#SBATCH --time=00:10:00
#SBATCH --account=z04

#SBATCH --partition=standard
#SBATCH --qos=standard

... do something ...
```

Serial job

```
#!/bin/bash

#SBATCH --job-name=serial
#SBATCH --time=00:20:00
...

#SBATCH --ntasks=1

...
```

Parallel job

```
#!/bin/bash

#SBATCH --nodes=4
#SBATCH --tasks-per-node=36
#SBATCH --cpus-per-task=1
#SBATCH --exclusive

...
srun ./mpi-code.x
```

Array job

```
#!/bin/bash

#SBATCH --array=1-100
#SBATCH --ntasks=1

# Use ${SLURM_ARRAY_TASK_ID}
```

Interactive job

```
[cirrus-login1]$ srun --exclusive --nodes=1 ...\
                      --pty /usr/bin/bash --login
[r1i0n14]$

...

[r1i0n14]$ exit
[cirrus-login1]$
```

Phase II

Phase II

- An expansion of existing Cirrus
  - Adds new GPU compute nodes
  - Adds fast non-volatile storage capability
- Aims
  - Support AI/machine learning workloads
  - Explore heterogeneous architectures

Hardware

- 36 “cascade” GPU nodes
  - Host 2x20 core Intel Xeon (Cascadelake)
  - Host memory 384 GB per node
  - Each node 4 x NVIDIA V100 SXM2 GPUs (16 GB per GPU)

```
$ module avail nvidia
nvidia/compilers-20.5   nvidia/cuda-10.2(default)
nvidia/cuda-10.1        nvidia/mathlibs-10.2
```

GPU job

```
#!/bin/bash

#SBATCH --partition=gpu-cascade
#SBATCH --qos=gpu

#SBATCH --gres=gpu:4

...
```

Using containers

```
#!/bin/bash

...
#SBATCH --ntasks=1
...

module load singularity

srun –cpu-bind=cores singularity run <file.sif>
```

Other hardware additions

- Non-volatile storage (NV memory)
  - HPE XFS storage layer
  - 256 GB usable
  - Exact configuration TBC
- EDR infiniband
  - Limited to new GPU nodes
  - To be integrated to NV storage/rest of infiniband
  - Exact details TBC

Acknowledgment

Please acknowledge Cirrus in your work:

“This work used the Cirrus UK National Tier-2 HPC Service at EPCC (http://www.cirrus.ac.uk/) funded by the University of Edinburgh and EPSRC (EP/P020267/1).”

References

- Access
  - https://www.cirrus.ac.uk/access/
  - https://epsrc.ukri.org/funding/calls/access-to-high-performance-computing/
- User Doucmentation
  - https://cirrus.readthedocs.io/
- Support
  - E-mail: `support@cirrus.ac.uk`