Trinity 2020

UNIVERSITY OF
OXFORD

arc
advanced
research
computing

Job Submission and understanding the scheduler

Introduction to Job Management Systems

- AKA: Resource Management System, Workload Manager and Batching System
- Purpose: utilise many computing resources, maximise throughput, assign hardware resources to users' jobs.
- Functionalities: queuing, scheduling and resource management.
- User JMS: request resources by submitting jobs, which would be sequential or parallel.
- Flavours: SLURM, Torque, LSF, Loadleveler, PBSPro, SGE and more.

Terminology for reference

Glossary

- core = unit that does the work (sometimes use CPU as a synonym)
- processor = collection of cores in a single package all sharing the same memory
- node = a collection of processors all sharing the same memory
- interconnect = the network in a machine the joins together the separate nodes

Each node has its own memory and cannot directly "see" another node's memory

Terminology for reference

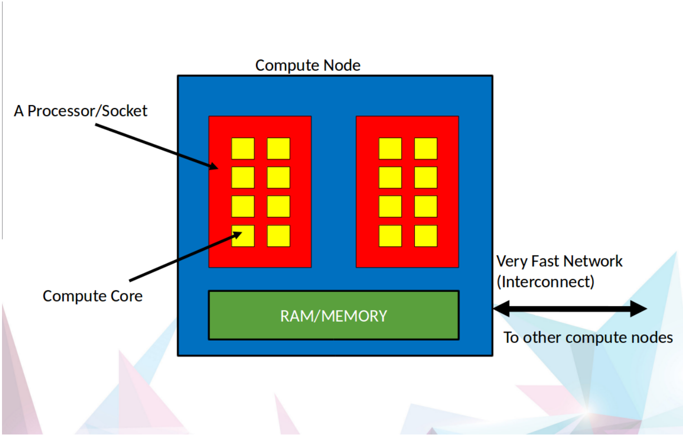
Distinction between processor, process and thread

processor a physical piece of hardware

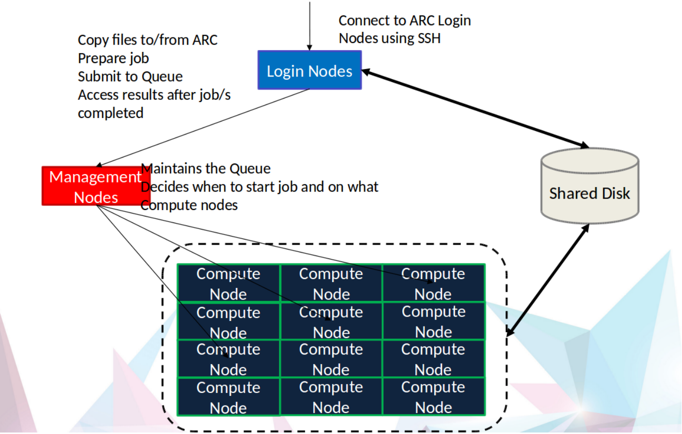
process an instance of a running program (software)

threads a process can perform multiple computations, i.e., program flows, concurrently

Single Compute Node



Cluster of Compute Nodes



Connect to ARC Login Nodes using SSH

Login Nodes

Copy files to/from ARC
Prepare job
Submit to Queue
Access results after job/s completed

Management Nodes

Maintains the Queue
Decides when to start job and on what Compute nodes

Shared Disk

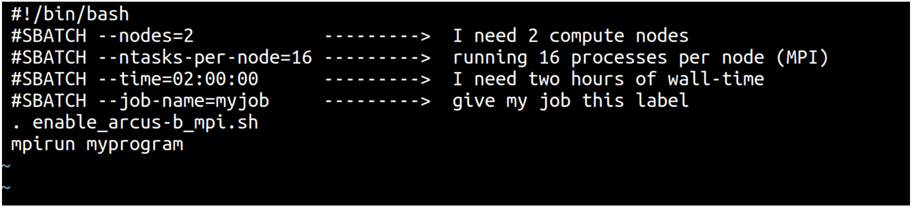
Slurm

- Simple Linux Utility for Resource Management
- Manages the queue When jobs start, what order and when
- Manages the compute node Schedules work on compute nodes that are free
- Support for “accelerator cards” such as Nvidia GPGPU

Prepare Job for submission on arcus-b

Write Shell script (simple text file) with instructions to SLURM

- SLURM instructions or directives request resources
- Shell commands say what to do in job
- Example (MPI or Message Passing Interface job)



```
#!/bin/bash
#SBATCH --nodes=2            ---------->  I need 2 compute nodes
#SBATCH --ntasks-per-node=16 ---------->  running 16 processes per node (MPI)
#SBATCH --time=02:00:00      ---------->  I need two hours of wall-time
#SBATCH --job-name=myjob     ---------->  give my job this label
. enable_arcus-b_mpi.sh
mpirun myprogram
~
~
```

Run Job, Get Results

- sbatch: Submit job (text file) to queue
- squeue: Monitor the queue
- scancel: Cancel the job (made a mistake?)
- Output from job will appear where you specify (shared file system)

sbatch

- Basic syntax: sbatch script.sh
- Requeueable jobs: sbatch [–requeue —–no-requeue]
- Job dependencies: sbatch -d afterok:¡jobid¿
- Job arrays: sbatch -a 1-20
- Requesting GPUs: sbatch –gres=gpu:1

squeue

- Basic syntax:squeue
- Single user: squeue -u bob
- Single job: squeue -j jobid
- More info: squeue -l
- Array elements: squeue -r

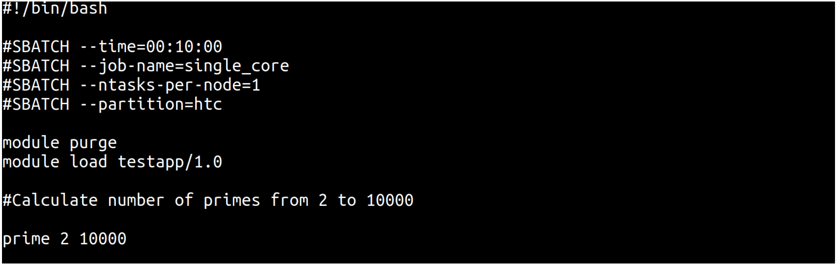
Other Slurm commands

- salloc: allocate resources in real time
- srun: used to submit a job for execution in real time
- sinfo: reports the state of partitions and nodes managed by SLURM
- scancel
- sacct: report job accounting information for active or completed jobs

Prepare Job for submission for arcus-htc

Write Shell script (simple text file) with instructions to SLURM

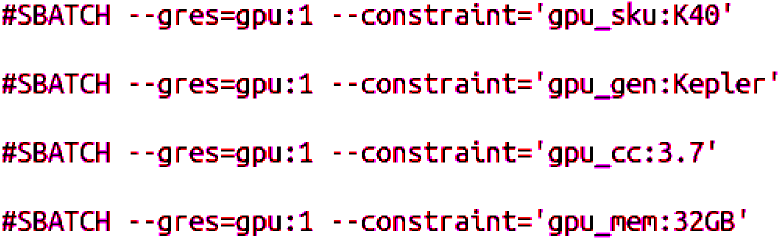
- SLURM instructions or directives request resources
- Shell commands say what to do in job



```
#!/bin/bash

#SBATCH --time=00:10:00
#SBATCH --job-name=single_core
#SBATCH --ntasks-per-node=1
#SBATCH --partition=htc

module purge
module load testapp/1.0

#Calculate number of primes from 2 to 10000

prime 2 10000
```

GPU nodes on arcus-htc

The most basic way you can access a GPU is by requesting a GPU device using the –gres option in your submission script:



```
#SBATCH --gres=gpu:1 --constraint='gpu_sku:K40'

#SBATCH --gres=gpu:1 --constraint='gpu_gen:Kepler'

#SBATCH --gres=gpu:1 --constraint='gpu_cc:3.7'

#SBATCH --gres=gpu:1 --constraint='gpu_mem:32GB'
```

Hilary 2020- Job Submission

UNIVERSITY OF
OXFORD

arc
advanced
research
computing

Connect to ARC systems

Job Submission Exercises - Connecting to arcus-b cluster For this exercise you need to login to one of arcus-b's login nodes. The SSH protocol is used for all remote user connections to our systems. Windows users cane use well-known SSH clients "Putty" or "Mobaxterm". Linux users can use the Linux terminal and run OpenSSH client.

- Open a terminal and from the prompt enter your username and password given to you by the instructor.
- ssh teaching01@arcus-b.arc.ox.ac.uk

Copy a Gromacs example

As an example we are using the software package Gromacs. We need to copy an input file for this package.

- mkdir examples
- cd examples
- cp /home/ouit0578/teaching-examples/ion_channel.tpr .
- ls -l
- you should see : -rw-r—— 1 teaching01 teaching 5368424 Jan 19 10:48 ion_channel.tpr

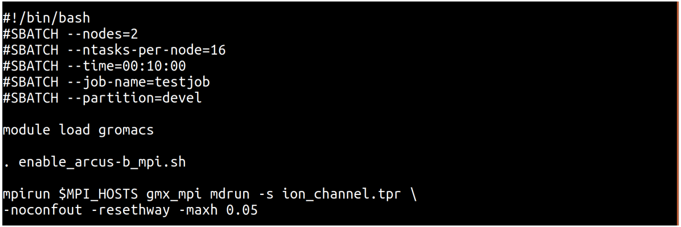
SLURM submission script

Any SLURM submission script is always in the form:

- SLURM directives section, using #SBATCH
- Commands section

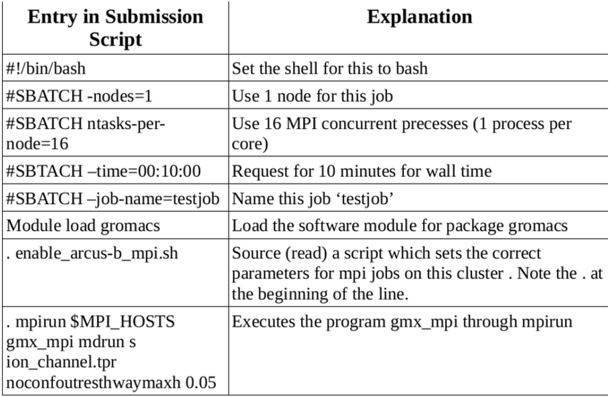
Writing a SLURM submission script for Gromacs

In the "examples" directory, do the following: Using the editor nano, create a file named job1.run. This is for arcus-b cluster Type : nano job1.run and add the following lines in the editor window



```
#!/bin/bash
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=16
#SBATCH --time=00:10:00
#SBATCH --job-name=testjob
#SBATCH --partition=devel

module load gromacs

. enable_arcus-b_mpi.sh

mpirun $MPI_HOSTS gmx_mpi mdrun -s ion_channel.tpr \
-noconfout -resethway -maxh 0.05
```

Explaining the lines in the submission script



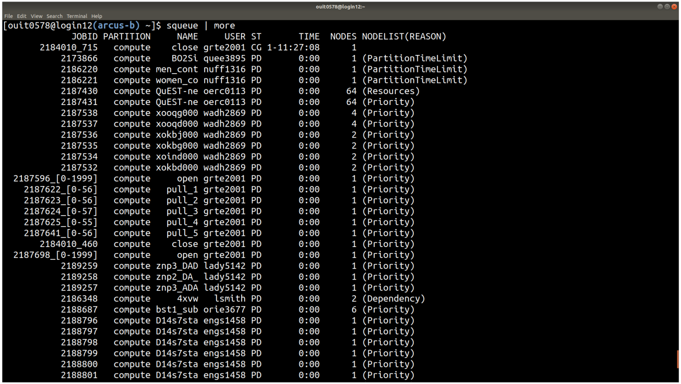
Entry in Submission Script	Explanation
#!/bin/bash	Set the shell for this to bash
#SBATCH -nodes=1	Use 1 node for this job
#SBATCH ntasks-per-node=16	Use 16 MPI concurrent precesses (1 process per core)
#SBTACH –time=00:10:00	Request for 10 minutes for wall time
#SBATCH –job-name=testjob	Name this job ‘testjob’
Module load gromacs	Load the software module for package gromacs
. enable_arcus-b_mpi.sh	Source (read) a script which sets the correct parameters for mpi jobs on this cluster . Note the . at the beginning of the line.
. mpirun $MPI_HOSTS gmx_mpi mdrun s ion_channel.tpr noconfoutresthwaymaxh 0.05	Executes the program gmx_mpi through mpirun

Submit your job for execution

- sbatch: Submit job job1.run SLURM will respond with an output that looks like this: teaching01@login12(arcus-b) : submitted batch job 49017
- squeue: Monitor the queue squeue -u your userid
- scancel: Cancel the job (made a mistake?)
- ls -l to see the output from the run .

squeue

Use the command squeue to see jobs currently running on one of the ARC clusters



```
[ouit0578@login12(arcus-b) ~]$ squeue | more
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
       2184010_715   compute    close grte2001 CG 1-11:27:08      1 
           2173866   compute    BO2Si quee3895 PD       0:00      1 (PartitionTimeLimit)
           2186220   compute men_cont nuff1316 PD       0:00      1 (PartitionTimeLimit)
           2186221   compute women_co nuff1316 PD       0:00      1 (PartitionTimeLimit)
           2187430   compute QuEST-ne oerc0113 PD       0:00     64 (Resources)
           2187431   compute QuEST-ne oerc0113 PD       0:00     64 (Priority)
           2187538   compute xooqg000 wadh2869 PD       0:00      4 (Priority)
           2187537   compute xooqd000 wadh2869 PD       0:00      4 (Priority)
           2187536   compute xokbj000 wadh2869 PD       0:00      2 (Priority)
           2187535   compute xokbg000 wadh2869 PD       0:00      2 (Priority)
           2187534   compute xoind000 wadh2869 PD       0:00      2 (Priority)
           2187532   compute xokbd000 wadh2869 PD       0:00      2 (Priority)
  2187596_[0-1999]   compute     open grte2001 PD       0:00      1 (Priority)
    2187622_[0-56]   compute   pull_1 grte2001 PD       0:00      1 (Priority)
    2187623_[0-56]   compute   pull_2 grte2001 PD       0:00      1 (Priority)
    2187624_[0-57]   compute   pull_3 grte2001 PD       0:00      1 (Priority)
    2187625_[0-55]   compute   pull_4 grte2001 PD       0:00      1 (Priority)
    2187641_[0-56]   compute   pull_5 grte2001 PD       0:00      1 (Priority)
       2184010_460   compute    close grte2001 PD       0:00      1 (Priority)
  2187698_[0-1999]   compute     open grte2001 PD       0:00      1 (Priority)
           2189259   compute znp3_DAD lady5142 PD       0:00      1 (Priority)
           2189258   compute znp2_DA_ lady5142 PD       0:00      1 (Priority)
           2189257   compute znp3_ADA lady5142 PD       0:00      1 (Priority)
           2186348   compute     4xvw   lsmith PD       0:00      2 (Dependency)
           2188687   compute bst1_sub orie3677 PD       0:00      6 (Priority)
           2188796   compute D14s7sta engs1458 PD       0:00      1 (Priority)
           2188797   compute D14s7sta engs1458 PD       0:00      1 (Priority)
           2188798   compute D14s7sta engs1458 PD       0:00      1 (Priority)
           2188799   compute D14s7sta engs1458 PD       0:00      1 (Priority)
           2188800   compute D14s7sta engs1458 PD       0:00      1 (Priority)
           2188801   compute D14s7sta engs1458 PD       0:00      1 (Priority)
```

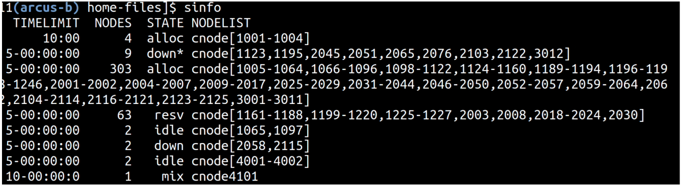
Information about your job in the queue and current state of the cluster

The command squeue reports the state of jobs or job steps. It has a large number of options for sorting, filtering.

- Single user: squeue -u yourusername
- Single job: squeue -j jobid
- More info: squeue -l
- scontrol show JobID=yourjobid

Information about state of the cluster

The command sinfo reports the state of partitions and nodes managed by Slurm.



```
11(arcus-b) home-files]$ sinfo
 TIMELIMIT  NODES  STATE NODELIST
     10:00      4  alloc cnode[1001-1004]
 5-00:00:00      9  down* cnode[1123,1195,2045,2051,2065,2076,2103,2122,3012]
 5-00:00:00    303  alloc cnode[1005-1064,1066-1096,1098-1122,1124-1160,1189-1194,1196-119
3-1246,2001-2002,2004-2007,2009-2017,2025-2029,2031-2044,2046-2050,2052-2057,2059-2064,206
2,2104-2114,2116-2121,2123-2125,3001-3011]
 5-00:00:00     63   resv cnode[1161-1188,1199-1220,1225-1227,2003,2008,2018-2024,2030]
 5-00:00:00      2   idle cnode[1065,1097]
 5-00:00:00      2   down cnode[2058,2115]
 5-00:00:00      2   idle cnode[4001-4002]
 10-00:00:0      1    mix cnode4101
```

Other Slurm commands

- salloc: allocate resources in real time
- srun: used to submit a job for execution in real time
- sinfo: reports the state of partitions and nodes managed by SLURM
- scancel JobID
- sacct: report job accounting information for active or completed jobs asst -j JobID
- scontrol show JobID : While a job is running scontrolwill give information about Start Time, EndTime,nodes

Define job dependencies

SLURM has numerous directives, the most useful can be found in the help section on of the ARC website. Users normally submit large numbers of jobs and some times they need to request for jobs which depend on each other

- Use your previous job submission script to create a new job script: cp job1.run job2.run
- Use nano to edit job1.run , nano job1.run and replace the final lines (star ng with mpirun ...) with: sleep 120 and save the file
- sbatch job1.run
- sbatch –dependency=afterok:49021 job2.run (where 49021 is the JobId of the first job) is very useful to those who 'feed' the input of their jobs using the output from previously executed jobs.

Submit Job arrays

At times, users want to submit many identical jobs at once and Job arrays can be used to do this.

- Use sbatch to define an array of 4 jobs
- sbatch –array=1-4 job2.run

Now we will have 4 different jobs performing the gromacs analysis! If you examine the output you will notice that the performance numbers might be slightly different across nodes.

Deleting Jobs

For several reasons you may want to delete a job while it is waiting in queue or during execution.

- sbatch job1.run Submitted batch job 49027
- scancel 49027 (where 49027 is the JobID)

Checking credit balance

Users are bound to a credit allocation, usually shared with other users of the same project. You can check the number of credits at any point using the command mybalance. The command shows the existing number of credits and the number of credits reserved from jobs for all users sharing the same project.

- mybalance
- Result: Please wait: Calculating balance ... You are a member on the following project(s): system,system-priority,system-basic and your current balance is: 1077842827 credits (299400 hours) Detailed account balance: Id Name Amount Reserved Balance CreditLimit Available —- ——————— ———— ———— ———— ————— ———— 51 system 897848728 0 897848728 0 897848728 5723 system-priority 89994288 0 89994288 0 89994288

Connect to ARC systems

Job Submission Exercises- Connecting to arcus-htc

same as above you need ssh however you need to login via oscgate :

- ssh teaching01@oscgate.arc.ox.ac.uk
- ssh arcus-htc