Batch Systems

Running calculations on HPC resources

Outline

- What is a batch system?
- How do I interact with the batch system
  - Job submission scripts
  - Interactive jobs
- Common batch systems
- Converting between different batch systems

Batch Systems

What are they and why are they used?

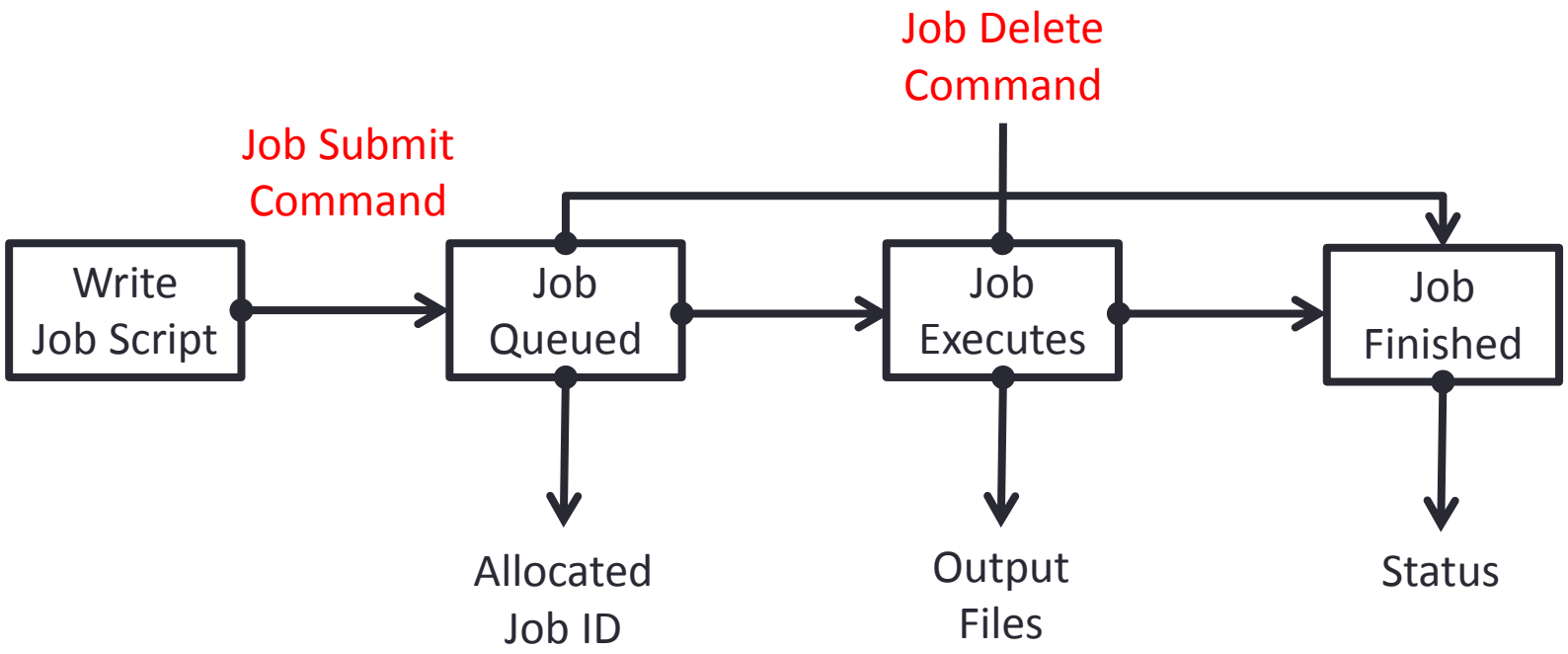
What is a batch system?

- A batch system controls access to the resources on a machine
- Used to ensure all users get a fair share of resources
  - As machine is usually oversubscribed
- Allows user to setup computational *job*, place it into batch queue and then log off machine
  - Job will be processed when there is space and time
  - Do not need to be continually logged-in for simulations to run
- Usually assumed that jobs are non-interactive
  - It runs for a time and produces results without intervention from the user
  - (Unlike interactive programs on a laptop.)

Reservation and Execution

- When you submit a job to a batch system you specify the resources you require:
  - Number of cores, job time,
- The batch system *reserves* a block of resources for you to use
- You can then use that block as you want, for example:
  - For a single job that spans all cores and full time
  - For multiple shorter jobs in sequence
  - For multiple smaller jobs running in parallel

Batch system flow



Running calculations

Interacting with the batch system

Batch and interactive jobs

- Most resources allow both batch and interactive jobs to be run through the batch system
- Batch jobs are non-interactive.
  - They run without user intervention and you collect the results at the end
  - Write a *job submission script* to run your job
- Interactive jobs allow you to use the resources interactively
  - For debugging/profiling
  - For visualisation and data analysis
- How you run these types of jobs differs with batch system and site

Job submission scripts

- Contain:
  - Batch system options
  - Commands to run
- Example (PBS on ARCHER)

```
#!/bin/bash –login
#PBS -N Weather1
#PBS -l select=171
#PBS -l walltime=1:00:00
cd $PBS_O_WORKDIR
aprun –n 4096 ./weathersim
```

how many nodes

how long

which directory

Program name

Parallel job launcher

#processes
(<= 24* #nodes)

epcc

THE UNIVERSITY OF EDINBURGH

Example: Sun Grid Engine

```
#!/bin/bash
#$ -V
#$ -l h_rt=:10:
#$ -cwd
#$ -pe mpi 4

mpiexec -n $NSLOTS ./myprogram
```

export local environment variables to batch job

how long

which directory

how many processors

Parallel job launcher

#processes inherited from #processors

Program name

Common batch systems

Batch systems

- PBS, Torque
- Grid Engine
- SLURM
- LSF – IBM Systems
- LoadLeveller – IBM Systems

Common concepts

- Queues
  - Portions of machine and time constraints
  - Generally small numbers of defined queues
- Generally specify:
  - Executable name
  - Account name
  - Maximum run time
  - Number of CPUs
  - Output file names/directories

HPCx (phase3) batch system LPAR allocation status at: 2008-10-20 14:24:30

	11	12	13	14	15	16	17	18
f401	(cm)	(inter)						
f402	(serial)	(inter)						
f403	.....	uclmbw	dlrojo	uclmbw	uclmbw	uclmbw	uclmbw	uclmbw
f404	uclmbw	pvs	pvs	uclmbw	pvs	pvs	dlrojo	dlrojo
f405	tal06	dlrojo	ucjela	cillin	dlrojo	tal06	dlrojo	ucjela
f406	rashed	rashed	jcatto	tal06	jcatto	rashed	tal06	tal06
f407	cdomene	cloenarz	wojcik	cdomene	cillin	cillin	cloenarz	jcatto
f408	tal06	rashed	cillin	jcatto	dlrojo	cdomene	dlrojo	tal06
f409	cdomene	ucjela	tal06	ucjela	.....	swr04ojb	shosking	emmaria
f410	swr05vas	.....	.....	swr04ojb	shosking	.....	.....	.....
f411	ndd21	vboppana	cloenarz	uqshe7	uqshe7	hpx0sjw1	hpx00061	uqshe7
f412	meli	uqshe7	hpx0sjw1	cdomene	rashed	uqshe7	hpx0sjw1	uqshe7
f413	ndd21	hpx00061	cloenarz	cdomene	hpx00061	hpx0sjw1	rashed	uqshe7
f414	cdomene	ndd21	vboppana	hpx00061	jw344	cdomene	uqshe7	ndd21
f415	jony	jony	jony	jony	jony	jony	jony	jony
f416	jony	jony	jony	jony	jony	jony	jony	jony
f417	jony	jony	jony	jony	jony	jony	jony	jony
f418	jony	jony	jony	jony	jony	jony	jony	jony
f419	jony	jony	jony	jony	jony	jony	jony	jony
f420	jony	jony	jony	jony	jony	jony	jony	jony
f421	jony	jony	jony	jony	jony	jony	jony	jony
f422	jony	jony	jony	jony	jony	jony	jony	jony

	Total	Alloc	Idle
Batch parallel (capability):	64	64	0
Batch parallel (capacity (S)):	16	15	1
Batch parallel (capacity (L)):	36	36	0
Batch parallel (capacity (vL)):	32	32	0
Batch parallel (development):	12	6	6
Batch parallel (test):	0	0	0
Batch parallel (course):	0	0	0
Interactive shared parallel:	2	0	2
Batch serial:	1	1	0
Unavailable:	0	0	0
Total:	163	154	9

Control programs

- Monitor, submit, and delete programs
- E.g. PBS on ARCHER
  - qsub
  - qdel
  - qstat

Migrating

Changing your scripts from one batch system to another

Conversion

- Usually need to change the batch system options
- Sometimes need to change the commands in the script
  - Particularly to different paths
  - Usually the order (logic) of the commands remains the same
- There are some utilities that can help
  - Bolt – from EPCC, generates job submission scripts for a variety of batch systems/HPC resources: https://github.com/aturner-epcc/bolt

Best practice

- Run short tests using interactive jobs if possible
- Once you are happy the setup works write a short test job script and run it
- Finally, produce scripts for full production runs
- Remember you have the full functionality of the Linux command line available in scripts
  - This allows for sophisticated scripts if you need them
  - Can automate a lot of tedious data analysis and transformation
  - …be careful to test when moving, copying deleting important data – it is very easy to lose the results of a large simulation due to a typo (or unforeseen error) in a script