

Queuing system and batch jobs

Shyam Gopalakrishnan
18th February 2021

What have you learnt so far?

- Computer architecture
- Logging in
 - Finding and loading modules
- Running interactive jobs
 -

Batch jobs

Batch jobs are resource provisions that run applications on nodes away from the user and **do not require supervision or interaction**. Batch jobs are commonly used for **applications that run for long periods of time or require little to no user input**.

Batch jobs are created from **a job script which provide resource requirements and commands for the job**.

Batch jobs



RECIPE _____ Y _____
From the kitchen of _____
Ingredients 

Directions 

Notes

A large, light gray arrow pointing from the left towards the right, indicating a flow or transition.

An illustration of a woman with dark hair in a ponytail, wearing a red shirt and a headset with a microphone. She is sitting at a desk with a laptop. Surrounding her are various icons: a clock, gears, speech bubbles, and a circular arrow with "24/7" inside, suggesting a 24/7 service.

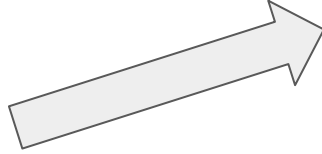
A large, light gray arrow pointing downwards, indicating a flow or transition.

A close-up photograph of a stack of lasagna on a white plate. The lasagna has layers of pasta, meat, and melted cheese. A sprig of green parsley is in the bottom left corner.

Batch jobs



Resource requirements and commands

[illegible]

Batch system + Compute nodes



Output of commands, ready for eat

Interactive jobs vs. batch jobs

Interactive job

- One at a time *
- Need to keep track of the job
- Cannot do other things *



Batch job

- Many at a time
- Scheduler can keep track of jobs
- Continue with other things



How do I run batch jobs? - the qsub command

... a job script which provide resource requirements and commands for the job ...

Running a batch job is easy:

```
qsub myScript.sh
```

A template script file

```
#!/bin/bash

### Account information
#PBS -W group_list=ku_fa -A ku_fa
### Job name
#PBS -N jobName
### Error stream
#PBS -e myError.err
### Output stream
#PBS -o myOut.log
### Resource requirements
#PBS -l nodes=1:ppn=40,mem=50gb,walltime=1:00:00
### Working directory
#PBS -d workingDirectory
### Mail on failure
#PBS -m n

### Load modules
module load xxx/version
module load yyy/version

### Run your jobs
Job1
Job2
...
```


A template script file

```
#!/bin/bash
```

Shebang line

```
### Account information
#PBS -W group_list=ku_fa -A ku_fa
### Job name
#PBS -N jobName
### Error stream
#PBS -e myError.err
### Output stream
#PBS -o myOut.log
### Resource requirements
#PBS -l nodes=1:ppn=40,mem=50gb,walltime=1:00:00
### Working directory
#PBS -d workingDirectory
### Mail on failure
#PBS -m n
```

Batch system
specifications -
resources, etc.

```
### Load modules
module load xxx/version
module load yyy/version
```

Module loads

```
### Run your jobs
Job1
Job2
...
```

Commands!

Batch system specifications: Let us break it down

Account information

#PBS -W group_list=ku_fa -A ku_fa

Job name

#PBS -N jobName

Error stream

#PBS -e myError.err

Output stream

#PBS -o myOut.log

Resource requirements

#PBS -l nodes=1:ppn=40,mem=50gb,walltime=1:00:00

Working directory

#PBS -d workingDirectory

Mail on failure

#PBS -m n

**Which account should be
charged for the job?
Required!**

Batch system specifications: Let us break it down

```
### Account information
#PBS -W group_list=ku_fa -A ku_fa
### Job name
#PBS -N jobName
### Error stream
#PBS -e myError.err
### Output stream
#PBS -o myOut.log
### Resource requirements
#PBS -l nodes=1:ppn=40,mem=50gb,walltime=1:00:00
### Working directory
#PBS -d workingDirectory
### Mail on failure
#PBS -m n
```

Name of the job.
Optional - by default the
name of the script

Batch system specifications: Let us break it down

```
### Account information
#PBS -W group_list=ku_fa -A ku_fa
### Job name
#PBS -N jobName
### Error stream
#PBS -e myError.err
### Output stream
#PBS -o myOut.log
### Resource requirements
#PBS -l nodes=1:ppn=40,mem=50gb,walltime=1:00:00
### Working directory
#PBS -d workingDirectory
### Mail on failure
#PBS -m n
```

Error and output files:
Optional- if not specified,
derived from job name.
Error: *jobname.ejob_id*
Output: *jobname.ojob_id*

Batch system specifications: Let us break it down

```
### Account information
#PBS -W group_list=ku_fa -A ku_fa
### Job name
#PBS -N jobName
### Error stream
#PBS -e myError.err
### Output stream
#PBS -o myOut.log
### Resource requirements
#PBS -l nodes=1:ppn=40,mem=50gb,walltime=1:00:00
### Working directory
#PBS -d workingDirectory
### Mail on failure
#PBS -m n
```

**Resource requirement -
Multiple different resources
specified on the same line.
Required!**

Batch system specifications: Let us break it down

```
### Account information
#PBS -W group_list=ku_fa -A ku_fa
### Job name
#PBS -N jobName
### Error stream
#PBS -e myError.err
### Output stream
#PBS -o myOut.log
### Resource requirements
#PBS -l nodes=1:ppn=40,mem=50gb,walltime=1:00:00
### Working directory
#PBS -d workingDirectory
### Mail on failure
#PBS -m n
```

Three resources have to be specified for a job to run on C2:

- 1. Number of nodes and cores per node:**
nodes=1:ppn=40
1 node and 40 cores.
- 2. Amount of memory:**
mem=50gb
50 GB of RAM
- 3. Time allotment for the job:**
walltime=1:00:00
1 hour (dd:hh:mm:ss)

Batch system specifications: Let us break it down

```
### Account information
#PBS -W group_list=ku_fa -A ku_fa
### Job name
#PBS -N jobName
### Error stream
#PBS -e myError.err
### Output stream
#PBS -o myOut.log
### Resource requirements
#PBS -l nodes=1:ppn=40,mem=50gb,walltime=1:00:00
### Working directory
#PBS -d workingDirectory
### Mail on failure
#PBS -m n
```

**Working directory where the
job should be run.
Optional - by default your
home directory.**

Batch system specifications: Let us break it down

```
### Account information
#PBS -W group_list=ku_fa -A ku_fa
### Job name
#PBS -N jobName
### Error stream
#PBS -e myError.err
### Output stream
#PBS -o myOut.log
### Resource requirements
#PBS -l nodes=1:ppn=40,mem=50gb,walltime=1:00:00
### Working directory
#PBS -d workingDirectory
### Mail on failure
#PBS -m n
```

**Get email on failure to
complete job.
Optional.**

More batch system options

#PBS -j oe

Merge the output and error stream into one stream - the output stream

#PBS -k oe

Merge the output and error stream into one stream - the output stream - with **continuously updated streams in your home directory.**

#PBS -V

Transfer environmental variables to compute node - not recommended for non-experts

Example script - we will do this in the exercise

```
#!/bin/bash
#PBS -W group_list=ku_fa -A ku_fa
#PBS -N canid1Map
#PBS -e canid1.err
#PBS -o canid1.log
#PBS -l nodes=1:ppn=40,mem=50gb,walltime=1:00:00
#PBS -d /home/projects/C2_test

### Load modules
module load bwa/0.7.15
module load samtools/1.9
module load htslib/1.9

### Run your jobs
bwa mem -t 40 refgenome/canFam31.fasta Canid1.R1.fastq.gz > Canid1.sam
samtools view -b Canid1.sam > Canid1.bam
samtools index Canid1.bam
```

slido

What order do the jobs get launched in?

 Start presenting to display the poll results on this slide.

How do I monitor my jobs?

Checkjob *job_id*

```
shygop@g-12-l0002 ~/projects/C2_test
```

```
$ checkjob 30518436
```

```
job 30518436
```

```
AName: canid1Map
```

```
State: Completed
```

```
Completion Code: 0   Time: Wed Feb 17 15:19:23
```

```
Creds: user:shygop group:ku-cbd account:ku-cbd class:batch
```

```
WallTime: 00:01:10 of 1:00:00
```

```
SubmitTime: Wed Feb 17 15:18:05
```

```
(Time Queued Total: 00:00:08 Eligible: 00:00:08)
```

```
TemplateSets: DEFAULT
```

```
NodeMatchPolicy: EXACTNODE
```

```
Total Requested Tasks: 40
```

```
Req[0] TaskCount: 40 Partition: pbs
```

```
Dedicated Resources Per Task: PROCS: 1 MEM: 1280M
```

```
GMetric[energy_used] Current: 0.00 Min: 0.00 Max: 0.00 Avg: 0.00 Total: 0.00
```

```
NodeSet=FIRSTOF:FEATURE:[NONE]
```

```
Allocated Nodes:
```

```
[g-01-c0019:40]
```

```
Applied Nodeset: g-01-sw101
```

```
SystemID: Moab
```

```
SystemJID: 30518436
```

```
IWD: /home/projects/ku-cbd/people/shygop/C2_test
```

```
StartCount: 1
```

```
Execution Partition: pbs
```

```
Flags: RESTARTABLE
```

```
Attr: checkpoint
```

```
StartPriority: 124999
```

```
IterationJobRank: 0
```

```
shygop@g-12-l0002 ~/projects/C2_test
```

```
$
```

How do I monitor my jobs?

qstat

```
shygop@g-12-10002 ~/projects/C2_test
```

```
$ qstat
```

```
moab.eth.cla:
```

Job ID	Username	Queue	Jobname	SessID	NDS	TSK	Req'd Memory	Req'd Time	S	Elap Time
30518436	shygop	batch	canid1Map	34062	1	40	50gb	01:00:00	C	--
30518447	shygop	batch	canid1Map	35300	1	40	50gb	01:00:00	C	--

```
shygop@g-12-10002 ~/projects/C2_test
```

```
$
```

Exercise time:
Let us try and run a job

Other ways to launch jobs: In line PBS options

```
#!/bin/bash
### Load modules
module load bwa/0.7.15
module load samtools/1.9
module load htslib/1.9

### Run your jobs
bwa mem -t 40 refgenome/canFam31.fasta Canid1.R1.fastq.gz > Canid1.sam
samtools view -b Canid1.sam > Canid1.bam
samtools index Canid1.bam
```

```
qsub -W group_list=ku_fa -A ku_fa -N canid1Map -e canid1.err -o canid1.log -l
nodes=1:ppn=40,mem=50gb,walltime=1:00:00 -d /home/projects/C2_test
mapCanid2.sh
```

Other ways to launch jobs: In line script and PBS options

```
echo “#! /bin/bash
### Load modules
module load bwa/0.7.15
module load samtools/1.9
module load htslib/1.9

### Run your jobs
bwa mem -t 40 refgenome/canFam31.fasta Canid1.R1.fastq.gz > Canid1.sam
samtools view -b Canid1.sam > Canid1.bam
samtools index Canid1.bam
” | qsub -W group_list=ku_fa -A ku_fa -N canid1Map -e canid1.err -o
canid1.log -l nodes=1:ppn=40,mem=50gb,walltime=1:00:00 -d
/home/projects/C2_test mapCanid2.sh
```