



# Batch Jobs

Mobeen Ludin

# Compiling on Blue Waters:

```
$ cd Examples/GalaxSee
```

```
$ cp -r ~instr004/BW_Institute/Examples/GalaxSee/run_GalaxSee.pbs
```

```
.
```

Switch default (Cray) compiler suites to GNU using **modules**:

```
$ module swap PrgEnv-cray PrgEnv-gnu
```

❖ Compile your code using **make**:

```
➤ $ ls -al
```

```
➤ $ make NO_X11=1
```

```
➤ $ ls -al (what files were created by make??)
```

# Types of Jobs on Blue Waters

Two type of jobs – **interactive** and **batch**

- ❖ Interactive mode for debug and optimization
- ❖ Batch mode for normal job runs

The job queue is like a valet:

- ❖ You give it brief instructions (pbs script) telling what program to run and how to run it.
- ❖ It will wait until a batch processor is free
  - Runs the program on that processor until it is finished
  - writes out errors and output from your programs
  - Sends you email notifications at start and finish or abort (if you wish)
- ❖ **Interactive jobs:**
  - `$ qsub -I -l nodes=1`
  - `$ qsub -I -l nodes=2:ppn=32:xe -l walltime=00:30:00`

# Submitting Jobs on Blue Waters

## ❖ Submit job:

➤ \$ **qsub run\_GalaxSee.pbs**

## ★ Batch Script:

- Sample scripts are at `/sw/userdoc/samplescripts`
- Specify resources needed
- Provide file names for stdout and stderr
- Define environmental variables
- Load needed modules
- Launch the job via the `aprun` command

# Batch (PBS) Script

```
#PBS -l nodes=2:ppn=32:xe
#PBS -l walltime=00:30:00
#PBS -l mem=2GB
#PBS -N m_GalaxSee
#PBS -e $PBS_JOBID.err
#PBS -o $PBS_JOBID.out
#PBS -m bea
#PBS -M mobeen.ludin@gmail.com
cd $PBS_O_WORKDIR
module swap PrgEnv-cray PrgEnv-gnu
## Export environment variables
export OMP_NUM_THREADS = 32
aprun -n 32 ./GalaxSee.cxx-mpi 10000 500 1000 0
```

# Resources limit (-l)

This directive specifies the number of nodes (**node=2**). You could also specify other properties of the node like the number of processors per node (**ppn=32**), type of nodes (**xe**), amount of memory (**mem=2GB**) .

```
#PBS -l nodes=2:ppn=32:xe
```

```
#PBS -l mem=2GB
```

This directive specifies the maximum walltime(real time, not CPU time) that the job should take. The job will be canceled if it the exceeds limit.

```
#PBS -l walltime=00:30:00
```

# Job ID and output/error

Give a name to your job

```
#PBS -N m_GalaxSee
```

Print out the any runtime error the file named=the\_jobid.err

```
#PBS -e $PBS_JOBID.err
```

Print out the out of the running program into file named=the\_jobid.out

```
#PBS -o $PBS_JOBID.out
```

The jobid is a unique integer id you get once you submit the job.

# Notification

Send an email when the job:

```
#PBS -m bea
```

- ❖ begins

- ❖ ends

- ❖ aborts

Send an email to [mobeen.ludin@gmail.com](mailto:mobeen.ludin@gmail.com) when bea=true:

```
#PBS -M mobeen.ludin@gmail.com
```

By Default the job starts in your home directory. But you can tell PBS to start the job from where ever the script is run:

```
cd $PBS_O_WORKDIR
```



# Status of the job

\$ **qstat**: Shows the status of all jobs (time=cput)

\$ **qstat -a**: View information about queued jobs

\$ **qstat -u userid**: List of jobs for user=userid

\$ **qstat | grep instr**: String matching

\$ **qshow**: Useful options are; -r (running jobs), -i (show idle jobs) and -b (blocked jobs).

\$ **qdel job\_id**: Delete a job with id = job\_id

# Output of qstat

```
$ qstat -a 846147.nid11293
```

```
qstat -a 846147.nid11293
```

```
nid11293: Blue_Waters
```

Job ID	Username	Queue	Jobname	SessID	NDS	TSK	Memory	Time	Req'd S	Time	Elap
846147.nid11293	instr004	normal	m_GalaxSee	--	2	64	--	00:30:00	Q	--	--