

OLCF Slurm Migration

Chris Fuson
Aug 28, 2019

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

Slurm Transition Dates

- Transition from Moab/Torque to Slurm
- Impacted Systems:
 - **DTN, Rhea**
- Dates
 - August 13, 2019
 - Slurm partitions available to aid migration
 - 196 Rhea compute nodes
 - 2 DTN scheduled nodes
 - **September 03, 2019**
 - Final migration, MOAB/Torque no longer available
 - MOAB/Torque queue emptied, new submissions rejected

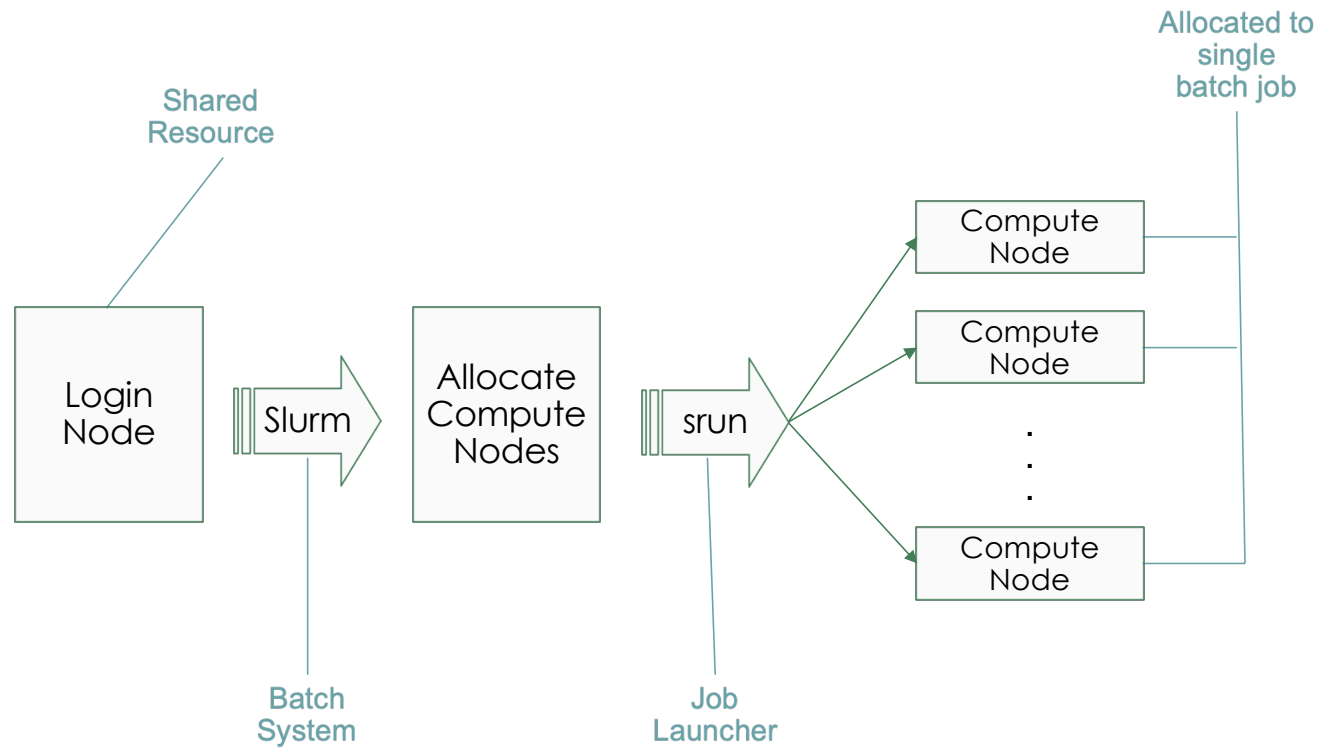
**Begin using Slurm
now and report
issues to
help@olcf.ornl.gov**

Rhea Overview

- Rhea contains 521 compute nodes separated into two partitions

Partition	Node Count	Memory	GPU	CPU
rhea (default)	512	128GB	-	[2x] Intel® Xeon® E5-2650 @ 2.0 GHz – 8 cores, 16 HT (for a total of 16 cores, 32 HT <i>per node</i>)
gpu	9	1TB	[2x] NVIDIA K80	[2x] Intel® Xeon® E5-2695 @ 2.3 GHz – 14 cores, 28 HT (for a total of 28 cores, 56 HT <i>per node</i>)

Rhea Login, Compute Nodes



Slurm Overview

Batch Submission

Task	Moab	LSF	Slurm
Submit batch script	qsub	bsub	sbatch
Submit interactive batch job	qsub -I	bsub -Is \$SHELL	salloc

Job Launch

Task	Moab	LSF	Slurm
Launch parallel job	mpirun / aprun	jsrun	srun

Monitor Queue

Task	Moab	LSF	Slurm
View batch queue	showq	bjobs	squeue

Slurm Example Batch Script

Batch script example

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

```
#SBATCH -t 1:00:00
```

```
#SBATCH -N 2
```

```
#SBATCH -A abc007
```

```
#SBATCH -o example.o%j
```

```
#SBATCH -J example
```

```
srun -n2 -N1 hostname
```

1 hour walltime

2 nodes

ABC007 project

Output file
example.o<jobid>

Job name

Batch submission

```
rhea-login1> sbatch example.slurm  
Submitted batch job 1865  
rhea-login1>
```

Common sbatch Options

Option	Example Usage	Description
-t	#SBATCH -t 1:00:00	Requested Walltime [hours:]minutes:seconds
-N	#SBATCH -N 16	Number of nodes
-A	#SBATCH -A ABC123	Project to which the job should be charged
-J	#SBATCH -J MyJobName	Name of the job. If not specified, will be set to name of submitted script.
-o	#SBATCH -o jobout.%j	File into which job's STDOUT should be directed (%j will be replaced with the job ID number) If not specified will be set to 'slurm-.%j.out'
-e	#SBATCH -e joberr.%j	File into which job's STDERR should be directed
-d	#SBATCH -d afterany:1234	Place dependency on previously submitted jobID 1234
--mail-type	#SBATCH --mail-type=FAIL #SBATCH --mail-type=BEGIN #SBATCH --mail-type=END	Send job report via email once job begins, fails, ends

**More details and flags can be found in the sbatch manpage*

Slurm Interactive Batch Job

- Allows access to compute resources interactively
- Through batch system similar to batch script submission, but returns prompt on launch node
- Run multiple srun with only one queue wait, very useful for testing and debugging
- Syntax
 - salloc
 - Most sbatch flags valid
 - Add batch flags to command line

Presentation examples
use the following to
allocate resources

```
rhea-login1> salloc -A abc007 -N 2 -t 2:00:00 -p gpu
salloc: Granted job allocation 1870
salloc: Waiting for resource configuration
salloc: Nodes rhea-gpu[8-9] are ready for job
rhea-gpu8> srun -n2 -N2 hostname
rhea-gpu8
rhea-gpu9
Rhea-gpu8>
```


Common Slurm Commands

Function	PBS/MOAB	LSF	Slurm
Submit	qsub	bsub	sbatch
Monitor Queue	showq/qstat	bjobs	squeue
Remove Queued Job	qdel	bkill	scancel
Hold Queued Job	qhold	bstop	scontrol hold
Release Held Job	qrls	bresume	scontrol release

Monitoring the Queue

- **squeue**

```
rhea-login1> squeue -l
Mon Aug 26 16:38:24 2019
```

JOBID	PARTITION	NAME	USER	STATE	TIME	TIME_LIMI	NODES	NODELIST(REASON)
1872	batch	job.a	userA	PENDING	0:00	1:00:00	190	(Resources)
1862	batch	sh	userB	RUNNING	1:21:24	2:00:00	1	rhea320
1871	batch	job.s	userC	RUNNING	0:53	1:00:00	19	rhea[323-332,334-343]
1870	gpu	sh	userA	RUNNING	23:46	1:00:00	2	rhea-gpu[8-9]

- **sacct**

```
rhea-login1> sacct -j 1870
```

JobID	JobName	Partition	Account	AllocCPUS	State	ExitCode
1870	sh	gpu	abc123	64	RUNNING	0:0
1870.extern	extern		abc123	64	RUNNING	0:0
1870.0	bash		abc123	1	RUNNING	0:0
1870.1	a.out		abc123	1	COMPLETED	0:0
1870.2	b.out		acb123	2	COMPLETED	0:0

- **jobstat**

```
rhea-login1> jobstat
```

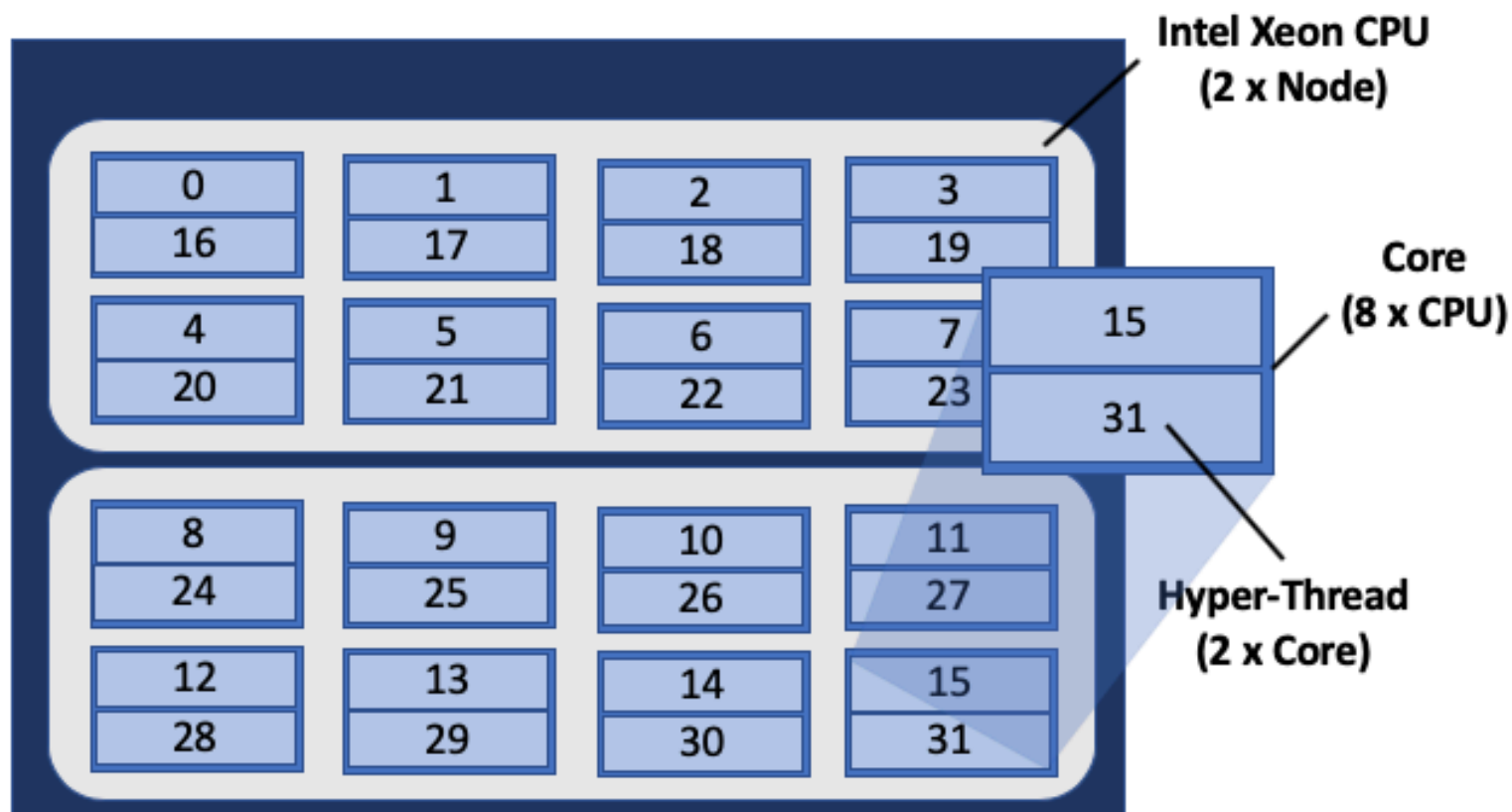
Running jobs-----

ST	JOBID	USER	ACCOUNT	NODES	PARTITION	NAME	TIME_LIMIT	START_TIME	TIME_LEFT
R	1862	userB	abc123	1	batch	sh	2:00:00	2019-08-26T15:17:00	28:39
R	1870	userA	abc123	2	gpu	sh	1:00:00	2019-08-26T16:14:38	26:17

Pending jobs-----

ST	JOBID	USER	ACCOUNT	NODES	PARTITION	NAME	TIME_LIMIT	SUBMIT_TIME	PRIORITY	START_TIME	REASON
----	-------	------	---------	-------	-----------	------	------------	-------------	----------	------------	--------

Rhea Compute Node



srun format and options

- `srun [options] executable [executable args]`

Flag	Example	Description
<code>-n <tasks></code>	<code>srun -n4</code>	Tasks started
<code>-c <ncpus></code>	<code>OMP_NUM_THREADS=4</code> <code>srun -n2 -c4</code>	ncpus per task. Default: 1
<code>--cpu-bind=<type></code>	<code>srun --cpu-bind=threads</code> <code>srun --cpu-bind=cores</code>	Task binding. Physical cores, hardware threads Default: cores
<code>--distribution=<type></code>	<code>srun --distribution=cyclic</code>	Task distribution Default: block
<code>-N <nodes></code>	<code>srun -n2 -N1</code>	Nodes allocated to job

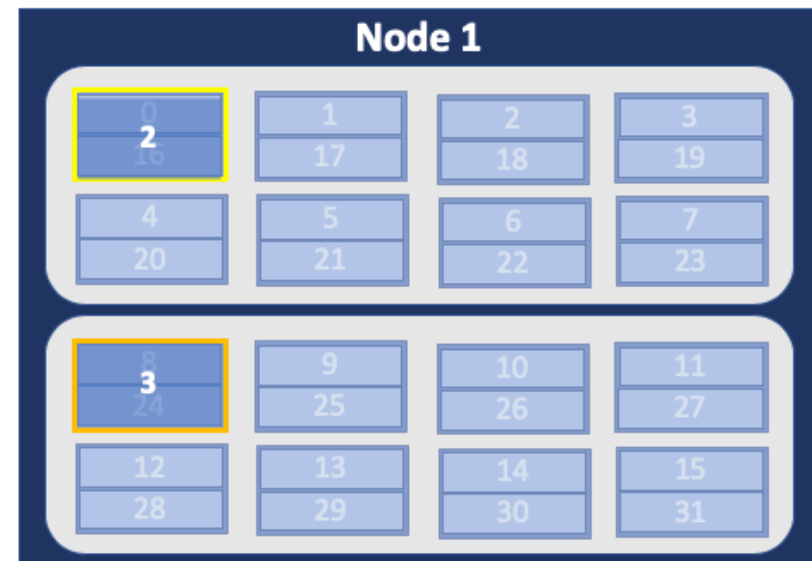
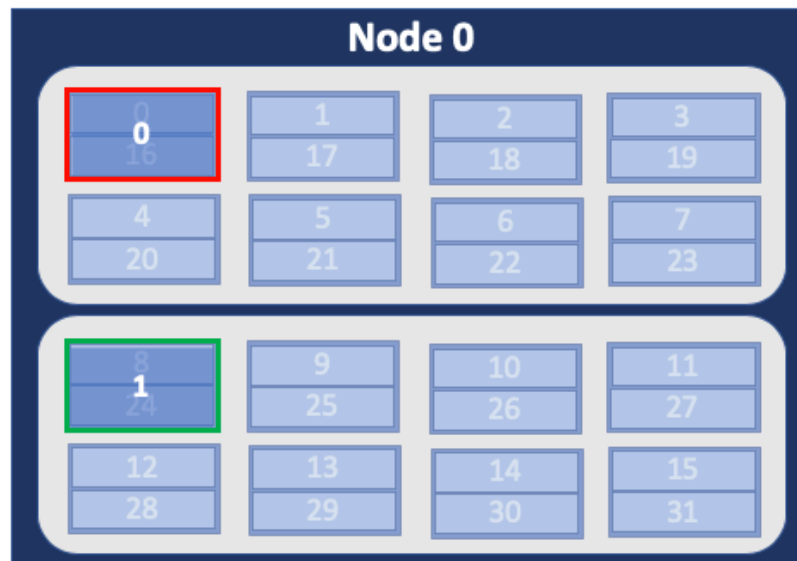
srun example: basic

```
srun -n4 -N2 --cpu-bind=cores ./a.out
```

4
tasks

2
nodes

Bind to
physical cores



srun example: hyper-thread

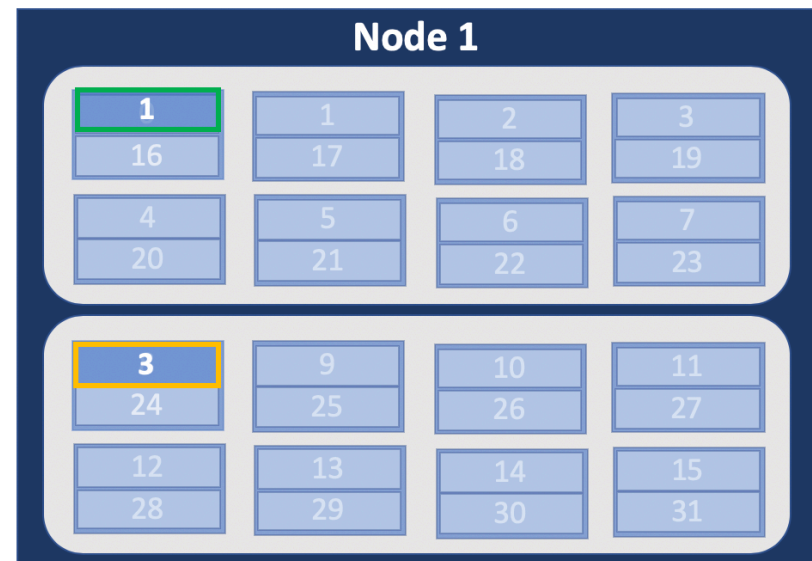
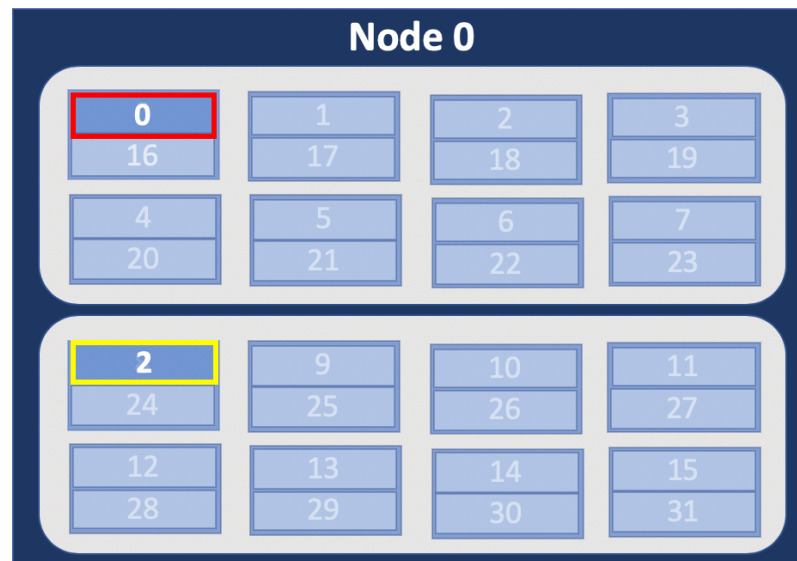
```
srun -n4 -N2 --cpu-bind=threads --distribution=cyclic ./a.out
```

4
tasks

2
nodes

Bind to hyper
thread

Round robin tasks
between nodes



srun example: OpenMP threads

```
srun -n4 -N2 -c2 --cpu-bind=threads ./a.out
```

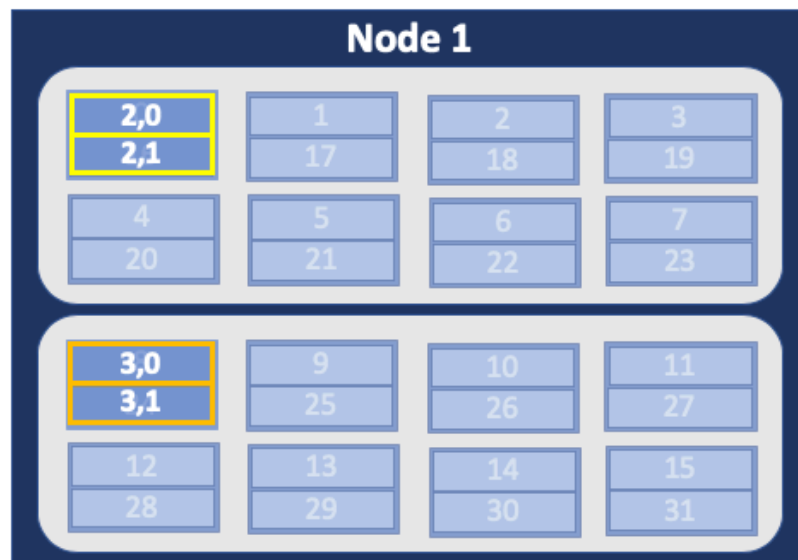
4
tasks

2
nodes

2
threads
per task

Bind to
hyper- thread

OMP_NUM_THREADS = 2



Questions?

- help@olcf.ornl.gov
- www.olcf.ornl.gov/for-users/system-user-guides/rhea/running-jobs/
- <https://slurm.schedmd.com>

