Cluster information

- Euler information: http://sbel.wisc.edu/Resources/
- Remote access to Euler
  - SSH to euler.wacc.wisc.edu
    ```
    $ ssh <username>@euler.wacc.wisc.edu
    ```
  - Use username + password (you should have received an email)
  - From Windows, you can use PuTTY: www.putty.org

- OS: CentOS Linux 7.1.1503 (Core)

- Batch system: SLURM 14.11
  - Submission scripts are regular shell scripts, with a few SLURM-specific comment lines

SLURM

- Simple Linux Utility for Resource Management
- Euler uses SLURM to manage; i.e., queue for execution, your job[s]
- Cheat-sheet: http://slurm.schedmd.com/pdfs/summary.pdf
- Full SLURM documentation: http://slurm.schedmd.com/documentation.html

Example job

Content of sample submission script **`submit_example.sh`**

```
#!/bin/bash
#SBATCH -N 1     # This requests one node
#SBATCH -o job_out # This sends stdout to a file
#SBATCH -e job_err # This sends stderr to a file
cd $SLURM_SUBMIT_DIR # go to job submission directory
./exampleJob
# The following lines rename the job output files
mv job_out jobname.o$SLURM_JOB_ID
mv job_err jobname.e$SLURM_JOB_ID
```

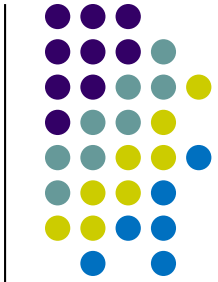
To submit:

```
$ sbatch submit_example.sh
```

To check status of your job:

```
$ squeue (or smap)
```

Rules

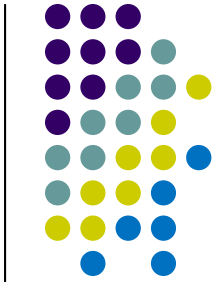


- Unattended jobs will be terminated
  - This means jobs with inactive shells

- GPU jobs should be submitted to the `slurm_shortgpu` queue
  - 8 nodes with GPU reserved
  - Maximum 20 minutes
  - Keeps everybody else's GPU jobs out of the way

- If needed, there are some nodes we can use in interactive mode
  - But the general rule is that we use `slurm`

Job Submission



- Two modes: batch and interactive
- Option 1: Batch Mode
  - Compute task written as shell script, with SLURM-specific comments
- Option 2: Interactive Mode
  - You get access to an interactive shell on a compute node

Job Submission Option 1: Batch Mode

example.sh

(you'll have to create this file)

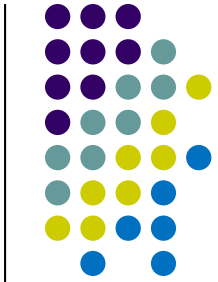
```
#!/bin/bash                          → Shell script
#SBATCH -p queue_name                → Use named queue
#SBATCH --job-name=myJob             → Name of job
#SBATCH -N 1 -n 1 --gres=gpu:1       → Resource selection
#SBATCH -o myJob.o%j                 → Set output file
cd $SLURM_SUBMIT_DIR                 → Set Work Directory
./myJob                              → Run!
```

Submit with:

```
$ sbatch example.sh
```

Output placed in myJob.o[0-9]*

Job Submission Option 2: Interactive

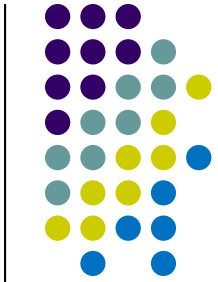


```
$ srun -p <queue_name> -u bash -i

$ ./myJob
```

- Note that for GPU programs, you must use the `<slurm_shortgpu>` queue. It is a special queue reserved for short GPU runs.

Euler: Resource Selection



- Request can follow a flag such as `-N` or `-n`, and/or it can follow a `--gres=…` (Generic RESource) flag.

- Examples
  - One node with one GPU
    ```
    -N 1 --gres=gpu:1
    ```
  - Two nodes with one GPU/node
    ```
    -N 2 --gres=gpu:1
    ```
  - Launch 3 tasks on 2 nodes
    ```
    -N 2 -n 3
    ```

- Note: must request GPUs for GPU jobs

GPU Job Sample Script

Content of sample submission script `gpu_example.sh`:

```
#!/bin/bash
#SBATCH -N 1                # This requests one node
#SBATCH --gres=gpu:1        # This requests one GPU
#SBATCH -p slurm_shortgpu # Use this queue for GPU jobs
#SBATCH -o gpuJob_out       # This sends stdout to a file
#SBATCH -e gpuJob_err       # This sends stderr to a file
cd $SLURM_SUBMIT_DIR        # go to job submission directory
./gpuJob
```

To submit:

```
$ sbatch gpu_example.sh
```

Check status of your job:

```
$ squeue (or smap)
```

OpenMP Job Sample Script

Content of sample submission script `omp_example.sh`:

```
#!/bin/bash
#SBATCH -N 1                # This requests one node
#SBATCH --sockets-per-node=4
#SBATCH --cores-per-socket=16
#SBATCH --threads-per-core=1
#SBATCH -o ompJob_out       # This sends stdout to a file
#SBATCH -e ompJob_err       # This sends stderr to a file
cd $SLURM_SUBMIT_DIR        # go to job submission directory
./ompJob
```

To submit:

```
$ sbatch omp_example.sh
```

Check status of your job:

```
$ squeue (or smap)
```

MPI Job Sample Script

Content of sample submission script `mpi_example.sh`:

```
#!/bin/bash
#SBATCH -N 4                # This requests four nodes
#SBATCH –t 0-0:10.:0        # Specify maximum wall clock time
#SBATCH -o mpiJob_out       # This sends stdout to a file
#SBATCH -e mpiJob_err       # This sends stderr to a file
cd $SLURM_SUBMIT_DIR        # go to job submission directory
mpiexec –np 4 ./gpuJob [args]
```

To submit:

```
$ sbatch mpi_example.sh
```

Check status of your job:

```
$ squeue (or smap)
```