

Submitting Jobs to the Supercomputer

Shelley Knuth

shelley.knuth@colorado.edu

www.rc.colorado.edu

Slides:

https://github.com/ResearchComputing/Basics_Supercomputing

General Info

Outline

- Submit specific jobs to the supercomputer!
 - Batch job
 - Running external scripts
 - Interactive job

RC Access

- For this tutorial, we will be using accounts on RC resources
- In a terminal or Git Bash window, type the following:

```
ssh <username>@tutorial-login.rc.colorado.edu
```

Password:

Clone the repo

- Once you are logged in, run the following command (all on one line):

```
Git clone  
https://github.com/ResearchComputing/Basics_Superco  
mputing.git
```

Maintenance!

- RC resources need maintenance
- Instead of Summit, running jobs on Crestone
- Made appropriate comments on slides
- Have two sets of scripts on Github – Crestone and Summit

Useful Slurm Commands - sbatch

- **sbatch**: submit a batch script to slurm
- You can use a bunch of flag options in a batch script or on the command line
- Useful to put in script so have for future use

- Example:

```
sbatch test.sh
```

OR

```
sbatch --partition=shas test.sh
```

<http://slurm.schedmd.com/sbatch.html>

SBATCH Options

<http://slurm.schedmd.com/sbatch.html>

`#SBATCH <options>` `sbatch <options>`

- Allocation: `--account=<account_no>`
 - Partition: `--partition=<partition_name>`
 - Sending emails: `--mail-type=<type>`
 - Email address: `--mail-user=<user>`
 - Number of nodes: `--nodes=<nodes>`
 - Quality of service: `--qos=<qos>`
 - Reservation: `--reservation=<name>`
 - Wall time: `--time=<wall time>`
 - Job Name: `--job-name=<jobname>`
- FYI: You do NOT actually type `<>` above – this designates something specific you as a user must enter about your job

Working on RC Resources

- After today, if you are submitting jobs from the login nodes, you'll need to run the following command to load the slurm module:

```
module load slurm/summit
```

- Maintenance today means you will run

```
module load slurm
```

- After you run this command you can run sbatch to submit jobs

Practice Examples

Submit Your First Job!

- Submit a slurm job with the following instructions:
 1. The job should run the Unix “hostname” command
 2. The job will be submitted from a bash script named `hostname_summit.sh`
 3. The job will run on 1 node
 4. We will request 1 minute wall time
 5. Run from the crestone (debug) QOS
 6. Run on the crc-serial (shas) partition
 7. Use the basics17 reservation
 - This is only for this workshop

Hostname_summit.sh

```
#!/bin/bash
#SBATCH --nodes=1                # Number of requested nodes
#SBATCH --time=0:01:00          # Max wall time
#SBATCH --qos=crestone           # Specify crestone QOS
#SBATCH --partition=crc-serial   # Specify crestone nodes
#SBATCH --output=hostname_%j.out # Rename standard output file
#SBATCH --reservation=basics17   # Reservation name

# purge all existing modules
module purge

hostname
```

Running the script

- Load up the slurm module

```
module load slurm  
(module load slurm/summit)
```

- Submit the job:

```
sbatch hostname_summit.sh
```

- Check output

Another slurm command

- **queue**
 - View information about jobs located in the slurm scheduling queue
- **OPTIONS:**
 - User: `-u <user_list>`
 - Queues: `-q <qos_list>` or `--qos=<qos_list>`
- **EXAMPLE:**
`queue --qos=debug`

<http://slurm.schedmd.com/queue.html>

Your turn

- Submit a slurm job with the following instructions:
 1. The job should run first the whoami command, then the Unix “sleep” command for 30 seconds, then the hostname command
 - Syntax for these Unix commands are below:

whoami

sleep 30

hostname

<http://slurm.schedmd.com/squeue.html>

Your turn

<http://slurm.schedmd.com/squeue.html>

- Submit a slurm job with the following instructions:
 1. The job will be submitted from a bash script named sleep.sh
 2. The job will run on 1 node
 3. Request a 1 minute wall time
 4. Run from the crestone (debug) QOS
 5. Run on the crc-serial (shas) partition
 6. Name your job sleep
 7. Email yourself the results at the end of the job run
 - Hint: Requires two SBATCH options to do this – see link at top of this slide
 8. Use the basics17 reservation

Sleep.sh

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --time=0:01:00
#SBATCH --qos=crestone
#SBATCH --partition=crc-serial
#SBATCH --output=sleep_%j.out
#SBATCH --job-name=sleep
#SBATCH --mail-type=end
###SBATCH --mail-user=<user>@colorado.edu
#SBATCH --reservation=basics17

# Number of requested nodes
# Max walltime
# Specify crestone QOS
# Specify crc-serial nodes
# Rename standard output file
# Job submission name
# Email you when the job ends
# Email address to send to
# Reservation name

# purge all existing modules
module purge

whoami
sleep 30
hostname
```

Running an external script

- Let's run a Matlab program
- We will run the batch script `matlab.sh`
- This script calls and runs `matlab_tic.m`

Running the script

- Submit the job:

```
sbatch matlab.sh
```

- Check output

Matlab.sh

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --time=0:02:00
#SBATCH --qos=crestone
#SBATCH --partition=crc-serial
#SBATCH --output=matlab_%j.out
#SBATCH --reservation=basics17

# Number of requested nodes
# Max walltime
# Specify debug QOS
# Specify Summit haswell nodes
# Output file name
# Reservation name

# purge all existing modules
module purge

# Load Matlab module
module load matlab

# Run matlab without a GUI
matlab -nodisplay -nodesktop -r "clear; matlab_tic;"
```

Your turn

- Submit a slurm job with the following instructions:
1. Create an R program called `R_program.R` that creates a vector called “planets” and then list the planets in the vector
 - Syntax: `planets -> planets <- c("Mercury", "Venus", "Earth", "Mars", "Jupiter", "Saturn", "Uranus", "Neptune", "Pluto")`
 2. Print off the vector
 - Syntax: `planets`
 3. Create a bash script called `R_code.sh` that runs the R script
 - Syntax: `Rscript R_code.sh`
 4. The job will run on 1 node
 5. We will request a 1 minute wall time
 6. Run from the crestone (debug) QOS
 7. Run on the crc-serial (shas) partition
 8. The output will be put in a file called `R_code_%j.out`
 9. Use the basics17 reservation
 10. Don't forget to load the R module!

Solution – R_code.sh

```
#!/bin/bash
#SBATCH --nodes=1                # Number of requested nodes
#SBATCH --time=0:01:00          # Max walltime
#SBATCH --qos=crestone           # Specify crestone QOS
#SBATCH --partition=crc-serial   # Specify crestone nodes
#SBATCH --output=R_code_%j.out   # Output file name
###SBATCH --reservation=basics17 # Reservation name

# purge all existing modules
module purge

# Load the R module
module load R/3.3.0

# Run R Script
Rscript R_program.R
```

Solution – R_program.R

```
#Simple R code example by Shelley Knuth  
(shelley.knuth@colorado.edu)
```

```
# Create vector
```

```
planets <- c("Mercury", "Venus", "Earth", "Mars", "Jupiter",  
"Saturn", "Uranus", "Neptune", "Pluto")
```

```
# Print off vector
```

```
planets
```

Interactive jobs!

- Sometimes we want our job to run in the background
- Sometimes we want to work in program in real time
- For example, Matlab
- Let's run an interactive Matlab job

Interactive job

- To do this, we are going to log out and log back in
 - Only necessary for demo
 - Need to add something to the sign in process

- For Mac Users:

```
ssh -X username@tutorial-login.rc.colorado.edu
```

- For Windows Users, must set up X-forwarding through your SSH client program
- Also must have an X-server package on your laptop
 - Xming for Windows or XQuartz for Mac

Interactive job

- To work with Matlab interactively, we're going to request some time from the supercomputer
- When the resources become available then we will start up Matlab
- Commands to run:

```
module load slurm  
(module load slurm/summit in the future)  
sinteractive --qos=crestone  
              --reservation=basics17
```

- Once we receive a prompt, then:

```
module load matlab  
matlab
```

- Once we finish we must exit!

Questions?

- Email rc-help@colorado.edu
- Link to survey on this topic:
<http://tinyurl.com/curc-survey16>

Speaker: Shelley Knuth

Title: Submitting Jobs to the Supercomputer July 2017
BSW

- Slides:
https://github.com/ResearchComputing/Basics_Supercomputing