Submitting Jobs to the Supercomputer

Shelley Knuth
shelley.knuth@colorado.edu

www.rc.colorado.edu Questions? #RC_BasicSC

Link to survey on this topic: http://tinyurl.com/rcpresurvey

Slides:
https://github.com/ResearchComputing/Final_Tutorials/tree/master/Basics_Supercomputing

Outline

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

Janus/Summit Access

- To access RC’s computing, for this tutorial:

```
ssh tutorial-login.rc.colorado.edu –l user00XX
```

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.sh
3. The job will run on 1 node
4. We will request 1 minute wall time
5. We will name the job hostname
6. The output will be put in a file called hostname.out

Contents of Batch Script

Bash Script hostname.sh:

```
#!/bin/bash
#SBATCH -N 1                      # Number of requested nodes
#SBATCH --time=0:01:00            # Max walltime
#SBATCH --job-name=hostname       # Job submission name
#SBATCH --output=hostname.out     # Output file name
#SBATCH --reservation=scbasics    # Reservation

hostname
```

Running the script

- Load up the slurm module

```
ml slurm
```

- Submit the job:

```
sbatch hostname.sh
```

- Check output:

```
nano hostname.out
```

Another slurm command

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

- EXAMPLE:

```
squeue -q janus-debug
```

http://slurm.schedmd.com/squeue.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 90 seconds, then the hostname command
   - Syntax for sleep: `sleep 90`
   - `hostname` runs as before. `Whoami` runs same as hostname
2. The job will be submitted from a bash script named sleep.sh
3. The job will run on 1 node
4. We will request a 1 minute wall time
5. The output will be put in a file called sleep.out
6. Email yourself the results at the end of the job run
   - Hint: Requires two SBATCH options to do this

Hmmm

- Why didn’t that work?

- Sleep 90 causes the script to wait 90 seconds before running
- We only asked for a wall time of 1 minute
- So we ran out of time requested before the job ran!

- Let’s fix it!

Solution – contents of script

```
#!/bin/bash
#SBATCH -N 1
#SBATCH --time=0:01:00
#SBATCH --job-name=sleep
#SBATCH --output=sleep.out
#SBATCH --reservation=scbasics    # Reservation
#SBATCH --mail-type=end
#SBATCH --mail-user=ralphie@colorado.edu

whoami
sleep 30
hostname
```

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

- NEVER EVER EVER EVER EVER EVER
- EVER EVER EVER EVER EVER EVER EVER EVER
- Run matlab from the command line without running an interactive job first
- BAD USER!

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

```
ssh tutorial-login.rc.colorado.edu –X –l
user00XX
```

- This enables x-windows on our laptops

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:

  <what command do we need to run first?>

  ```
  sinteractive --reservation=scbasics
  ```

- Once we receive a prompt, then:

  ```
  ml matlab
  matlab
  ```

- Once we finish we must exit!

Your turn – Interactive Job

1. Run an interactive job that opens up TotalView
   - Make sure you run the job using the reservation name `scbasics`
2. Don’t forget to exit!

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

- This is NOT an interactive job

Running the script

- Load up the Matlab module

```
ml matlab
```

- Submit the job:

```
sbatch matlab.sh
```

- Check output:

```
nano Matlab.out
```

Contents of Batch Script

Bash Script matlab.sh:

```
#!/bin/bash
#SBATCH -N 1                      # Number of requested nodes
#SBATCH --time=0:02:00            # Max walltime
#SBATCH --job-name=Matlab         # Job submission name
#SBATCH --output=Matlab.out       # Output file name
#SBATCH --reservation=scbasics    # Reservation name



# Run matlab without a GUI
matlab -nojvm -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. The output will be put in a file called R.out
7. Name your job R_code
8. Email yourself the results at the end of the job run
   - Hint: Requires two SBATCH options to do this

Solution – R_code.sh

```
#!/bin/bash
#SBATCH -N 1                          # Number of requested nodes
#SBATCH --time=0:01:00                # Max walltime
#SBATCH --job-name=R_code             # Job submission name
#SBATCH --output=R_code.out           # Output file name
#SBATCH --reservation=scbasics        # Reservation - will only
work during workshop



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
```

Questions?

- Email rc-help@colorado.edu
- Twitter: CUBoulderRC

- Link to survey on this topic:

http://tinyurl.com/curc-survey16

- Slides: https://github.com/ResearchComputing/Final_Tutorials/tree/master/Basics_Supercomputing