Intro to SLURM

& CS Clusters

Kevin Miles – Systems Engineer
UVa Computer Science
Talk Overview

CS Computing
- What resources are available
- Where do I find software
- What OS am I using
- Using CUDA

SLURM
- What is SLURM
- Query cluster status
- Control jobs
- Scheduling
- Debugging tips
CS Computing
CS Computing

Resources

- Servers running CentOS Linux
  - Based on Red Hat Enterprise Linux (RHEL)
- Shared “interactive” servers
  - portal.cs.virginia.edu
- Submit jobs to cluster nodes from portal.cs nodes

- Wide variety of available software & programming languages
Software
Software

- Software can be loaded/unloaded using `module` command
- Software stored in network volume
  - mounted to `/sw`
  - filesystem tuned for read access
    - noatime, zfs tweaks, async
  - don’t waste space/bandwidth storing software in your home directory
- Send software requests to cshelpdesk@virginia.edu
Software -- Modules

- Modules control environment variables like
  - PATH
  - LD_LIBRARY_PATH
  - LD_INCLUDE_PATH
  - MANPATH

- Primarily designed for bash

- Support for alternate shells
  - zsh
  - csh
  - tcsh

- Support for scripting languages
  - perl
  - python
  - ruby
Software -- Modules

- If using a non-bash shell you must run init script manually
  - `/etc/profile.d/modules.sh` won’t work
  - Init scripts found in `/sw/centos/Modules/current/init`
- More shells coming soon
  - Modules version 4.3 supports `fish`, `ksh`, `tcl` and `R`

- You can write your own modules!
  - Append path to custom modfiles: `module use -a ~/local/dir`
Hardware
CS Computing

GPGPU

- GPU equipped nodes available inside & outside of Slurm
  - gpusrv01-gpusrv06 available outside of Slurm
- Cuda-toolkit available in multiple versions
- Be sure that you have module loaded
  - module load cuda-toolkit

- cuDNN module also available
CS Computing

GPGPU

- Python3 and Anaconda3 both available
- Including GPU support
- Can create virtualenv’s from either python base
  - inside you can use pip or conda to install your own packages
- Popular packages already available
  - torch
  - pytorch
  - tensorflow-gpu
SLURM
Using Slurm -- Overview

- Viewing cluster status
- Viewing job status
- Submitting jobs
- Debugging
Using Slurm -- Access Model

- Compute servers are non-accessible via ssh
  - login access to server subverts resource allocation
- Interactive session still possible through `srun /bin/bash`
- Need GPUs for testing? Use `gpusrv` servers
Using Slurm -- Access Model

User commands (partial list):
- `scontrol`
- `sinfo`
- `squeue`
- `scancel`
- `sacct`
- `srun`

Controller daemons:
- `slurmctld (primary)`
- `slurmctld (backup)`

Slurmd (optional)

Other clusters

Database

Compute node daemons
- `slurmd`
- `slurmd`
- `slurmd`
Using Slurm -- Access Model

**Slurm Controller**
- slurm-master.cs

**Slurm Clients**
- portal.cs

**Compute Nodes**

**User commands (partial list)**
- `scontrol`
- `sinfo`
- `squeue`
- `scancel`
- `sacct`
- `srun`

**Controller daemons**
- `slurmd`
- `slurmd` (backup)
- `Slurmdbd` (optional)

**Database**

**Other clusters**

**Compute node daemons**
- `slurmd`
- `slurmd`
- `....`
- `slurmd`
Using Slurm -- Commands Overview

- `sinfo` - list information about cluster nodes
- `squeue` - list currently queued jobs
- `srun` - quickly submit single command job
- `sbatch` - submit batch script job
- `scontrol` - admin tool, some commands for non-root users

Check the wiki for command line cheat sheet
Gathering Info
<table>
<thead>
<tr>
<th>PARTITION</th>
<th>AVAIL</th>
<th>TIMELIMIT</th>
<th>NODES</th>
<th>STATE</th>
<th>NODELIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>main*</td>
<td>up</td>
<td>infinite</td>
<td>3</td>
<td>mix</td>
<td>granger[5,8],trillian1</td>
</tr>
<tr>
<td>main*</td>
<td>up</td>
<td>infinite</td>
<td>22</td>
<td>idle</td>
<td>artemis2,granger[2-4,6-7],lynx[08-12],nibbler[1-4],slurm[1-5],trillian[2-3]</td>
</tr>
<tr>
<td>falcon</td>
<td>up</td>
<td>infinite</td>
<td>8</td>
<td>idle</td>
<td>falcon[1-6,8-9]</td>
</tr>
<tr>
<td>intel</td>
<td>up</td>
<td>infinite</td>
<td>2</td>
<td>mix</td>
<td>granger[5,8]</td>
</tr>
<tr>
<td>intel</td>
<td>up</td>
<td>infinite</td>
<td>19</td>
<td>idle</td>
<td>granger[2-4,6-7],lynx[08-12],nibbler[1-4],slurm[1-5]</td>
</tr>
<tr>
<td>amd</td>
<td>up</td>
<td>infinite</td>
<td>1</td>
<td>mix</td>
<td>trillian1</td>
</tr>
<tr>
<td>amd</td>
<td>up</td>
<td>infinite</td>
<td>4</td>
<td>idle</td>
<td>artemis[2,4],trillian[2-3]</td>
</tr>
<tr>
<td>centos</td>
<td>up</td>
<td>infinite</td>
<td>3</td>
<td>mix</td>
<td>granger[5,8],trillian1</td>
</tr>
<tr>
<td>centos</td>
<td>up</td>
<td>infinite</td>
<td>22</td>
<td>idle</td>
<td>artemis2,granger[2-4,6-7],lynx[08-12],nibbler[1-4],slurm[1-5],trillian[2-3]</td>
</tr>
<tr>
<td>share</td>
<td>up</td>
<td>infinite</td>
<td>5</td>
<td>idle</td>
<td>lynx[08-12]</td>
</tr>
<tr>
<td>gpu</td>
<td>up</td>
<td>infinite</td>
<td>1</td>
<td>drain*</td>
<td>ai06</td>
</tr>
<tr>
<td>gpu</td>
<td>up</td>
<td>infinite</td>
<td>2</td>
<td>drain</td>
<td>artemis[6-7]</td>
</tr>
<tr>
<td>gpu</td>
<td>up</td>
<td>infinite</td>
<td>9</td>
<td>mix</td>
<td>ai[01,03-05],lynx[01-02,05-07]</td>
</tr>
<tr>
<td>gpu</td>
<td>up</td>
<td>infinite</td>
<td>1</td>
<td>alloc</td>
<td>lynx03</td>
</tr>
<tr>
<td>gpu</td>
<td>up</td>
<td>infinite</td>
<td>6</td>
<td>idle</td>
<td>artemis[4-5],lynx04,ristretto[01,03-04]</td>
</tr>
</tbody>
</table>
Gathering Info -- squeue

```
ktm5j@portal01 ~ $ squeue

   JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
 1245851      main  bbv_gen    ab4cd  R 10-02:18:38      1 trillian1
 1249230       gpu     bash  ab4cd  R 1-17:11:09      1 ai01
 1249232       gpu     bash  ab4cd  R 1-17:01:12      1 ai03
 1249233       gpu     bash  ab4cd  R 1-16:59:24      1 ai04
 1249234       gpu     bash  ab4cd  R 1-16:54:41      1 lynx05
 1249235       gpu     bash  ab4cd  R 1-16:52:43      1 lynx06
 1249236       gpu     bash  ab4cd  R 1-16:51:34      1 lynx07
 1249237       gpu     bash  ab4cd  R 1-16:49:07      1 ai05
 1249808_28      main PagingGr    wxy7z  R  8:02:39      1 granger8
 1249808_30      main PagingGr    wxy7z  R  8:02:39      1 granger8
 1249808_31      main PagingGr    wxy7z  R  8:02:39      1 granger8
 1249808_96      main PagingGr    wxy7z  R  8:02:39      1 granger5
```
Gathering Info -- partitions

- Slurm partitions separate compute nodes into groups
- GPU enabled nodes are found in the `gpu` partition

```
ktm5j@portal01 ~ $ scontrol show partition | grep PartitionName
PartitionName=main
PartitionName=falcon
PartitionName=intel
PartitionName=amd
PartitionName=centos
PartitionName=share
PartitionName=gpu
```

- `srun` commands: `-p gpu`
- `sbatch` commands: `#SBATCH --partition=gpu`
Using SLURM
Using Slurm -- Modules

- Slurm will execute jobs outside of login shell
- Modules won’t be available unless init called
  - Add to scripts: `source /etc/profile.d/modules.sh`
Using Slurm -- srun

Usage: srun [OPTIONS...] executable [args...]

Common Options

- `-c, --cpus-per-task=<ncpus>`
  Request that `ncpus` be allocated per process.

- `--gpus-per-task=[:]<number>`
  Specify the number of GPUs required for the job on each task to be spawned.

- `--gres=<list>`
  Specifies a comma delimited list of generic consumable resources.

- `-N, --nodes=<minnodes[-maxnodes]>`
  Request that a minimum of `minnodes` nodes be allocated to this job.
Using Slurm -- srun interactive

ktm5j@portal01 ~ $ srun -w granger2 --pty bash -i -l -
ktm5j@granger2 ~ $ hostname
granger2.cs.virginia.edu

- Interactive sessions are useful for debugging
- Please don’t leave open sessions
Using Slurm -- sbatch

```
ktm5j@portal01 ~ $ batch --help
Usage: sbatch [OPTIONS...] executable [args...]
```

Common Options

- `-c, --cpus-per-task=<ncpus>`
  - Request that ncpus be allocated per process.

- `--gpus-per-task=[<type>:]<number>`
  - Specify the number of GPUs required for the job on each task to be spawned

- `--gres=<list>`
  - Specifies a comma delimited list of generic consumable resources.

- `-N, --nodes=<minnodes[-maxnodes]>`
  - Request that a minimum of minnodes nodes be allocated to this job.

- Command arguments can be placed in command line, or in batch file header
- sbatch and srun use the same arguments
SLURM

Using Slurm -- sbatch files

```bash
#!/bin/sh
#SBATCH --time=1
/bin/hostname
```

- Sbatch files are shell scripts
- Can include Slurm arguments in file
  - argument lines begin with `#SBATCH`
- Can also include arguments in command line
Using Slurm -- sbatch example

ktm5j@portal03 ~/slurm $ module load java
ktm5j@portal03 ~/slurm $ ls
batch.sh hello.java
Using Slurm -- sbatch example

```bash
ktm5j@portal03 ~/slurm $ module load java
ktm5j@portal03 ~/slurm $ ls
batch.sh  hello.java
ktm5j@portal03 ~/slurm $ cat hello.java
class Hello {
    public static void main(String args[]) {
        System.out.println("Hello from SLURM!");
    }
}
ktm5j@portal03 ~/slurm $ javac hello.java
```
Using Slurm -- sbatch example

```bash
ktm5j@portal03 ~/slurm $ module load java
ktm5j@portal03 ~/slurm $ ls
batch.sh  hello.java
ktm5j@portal03 ~/slurm $ cat hello.java
class Hello {
    public static void main(String args[]) {
        System.out.println("Hello from SLURM!");
    }
}
ktm5j@portal03 ~/slurm $ javac hello.java
ktm5j@portal03 ~/slurm $ cat batch.sh
#!/bin/bash

#SBATCH --nodelist=ai02
#SBATCH --partition=gpu

source /etc/profile.d/modules.sh

module load java

java Hello
```
Using Slurm -- sbatch example

```
ktm5j@portal03 ~/slurm $ module load java
ktm5j@portal03 ~/slurm $ ls
batch.sh  hello.java
ktm5j@portal03 ~/slurm $ cat hello.java
class Hello {
    public static void main(String args[]) {
        System.out.println("Hello from SLURM!");
    }
}
ktm5j@portal03 ~/slurm $ javac hello.java
ktm5j@portal03 ~/slurm $ cat batch.sh
#!/bin/bash

#SBATCH --nodelist=ai02
#SBATCH --partition=gpu

source /etc/profile.d/modules.sh

module load java

java Hello

ktm5j@portal03 ~/slurm $ sbatch batch.sh
Submitted batch job 1249913
```
Using Slurm -- sbatch example

ktm5j@portal03 ~/slurm $ module load java
ktm5j@portal03 ~/slurm $ ls
batch.sh hello.java
ktm5j@portal03 ~/slurm $ cat hello.java
class Hello {
    public static void main(String args[]) {
        System.out.println("Hello from SLURM!");
    }
}
ktm5j@portal03 ~/slurm $ javac hello.java
ktm5j@portal03 ~/slurm $ cat batch.sh
#!/bin/bash

#SBATCH --nodelist=ai02
#SBATCH --partition=gpu

source /etc/profile.d/modules.sh

module load java

java Hello

ktm5j@portal03 ~/slurm $ sbatch batch.sh
Submitted batch job 1249913
ktm5j@portal03 ~/slurm $ ls
batch.sh Hello.class hello.java slurm-1249913.out
ktm5j@portal03 ~/slurm $ cat slurm-1249913.out
Hello from SLURM!
ktm5j@portal03 ~/slurm $
Using Slurm -- filesystem access

• Job files (sbatch, scripts, executables, input data) must be on shared storage
  - eg. /u, /p or /bigtemp
  - Your job can use local storage for tmp files
Final Notes
Online Resources

- CS Wiki
  - https://www.cs.virginia.edu/wiki/

- Slurm Website
  - https://slurm.schedmd.com
  - https://slurm.schedmd.com/tutorials.html

- Web Search is your friend!

SLURM Problems?

- Submit a Helpdesk Ticket via cshelpdesk@virginia.edu