Introduction to SLURM on the High Performance Cluster at the Center for Computational Research

Cynthia Cornelius
Center for Computational Research
University at Buffalo, SUNY
701 Ellicott St
Buffalo, NY 14203
Phone: 716-881-8959
Office: B1-109
cdc at ccr.buffalo.edu

June 2013
What is SLURM?

- SLURM is an acronym for Simple Linux Utility for Resource Management.
- SLURM is a workload manager that provides a framework for job queues, allocation of compute nodes, and the start and execution of jobs.
- SLURM replaces the PBS resource manager and MAUI scheduler on the CCR clusters.
Why SLURM?

- SLURM provides superior scalability and performance.
- It can manage and allocate the compute nodes for large clusters.
- SLURM can accept up to 1,000 jobs per second.
- SLURM is a comprehensive resource manager.
  - Individual node resources, such as GPUs or number of threads on a core, can be scheduled with SLURM.
What has changed?

Outline of topics:

- New compute nodes.
- Front-end (login) machine.
- Home directory path.
- Queue names.
- Scheduler commands.
- Syntax for specifying resource requests.
- Task launching.
- Node sharing policy.
- Interactive job submission.
- Job monitoring.
CCR Cluster Compute Nodes

- New! 32 16-core nodes with 128GB and Mellanox IB
- New! 12-core Visualization node with 256GB, 2 Nvidia Fermi GPUs and remote visualization clients
- 372 12-core nodes with 48GB and Qlogic IB
- 256 8-core nodes with 24GB and Mellanox IB
- 16 32-core nodes with 256GB and Qlogic IB
- 2 32-core nodes with 512GB and Qlogic IB
- 32 12-core nodes with 2 Nvidia Fermi GPUs, 48GB and Mellanox IB
Front-end and Home directory

- The new front-end machine is `rush.ccr.buffalo.edu`
  - 32-core node with 256GB of memory.
- Home directories are `/user/username` on the front-end and compute nodes.
  - The `/user/username/u2` directory still exists under `/user/username`. **No data have been lost.**
- Users who were using the `/user/username/u2` directory should copy the `.bashrc`, `.bash_profile` and the `.ssh` directory to `/user/username`. 
SLURM Partitions

- All general and faculty clusters are partitions in SLURM.
- The **general-compute** partition corresponds to the u2 cluster default queue.
- The **debug** partition corresponds to the u2 debug queue.
- GPU nodes are requested resources rather than residing in a separate partition.
- Faculty clusters have separate access controlled partitions.
SLURM Commands

- `squeue` – shows the status of jobs.
- `sbatch` – submits a script job.
- `salloc` – submits an interactive job.
- `srun` – runs a command across nodes.
- `scancel` – cancels a running or pending job.
- `sinfo` – provides information on partitions and nodes.
- `sview` – graphical interface to view job, node and partition information.
squeue example

squeue -u cdc

<table>
<thead>
<tr>
<th>JOBID</th>
<th>PARTITION</th>
<th>NAME</th>
<th>USER</th>
<th>ST</th>
<th>TIME</th>
<th>NODES</th>
<th>NODELIST(REASON)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4832</td>
<td>general-c</td>
<td>hello_te</td>
<td>cdc</td>
<td>R</td>
<td>0:20</td>
<td>2</td>
<td>f16n[10-11]</td>
</tr>
</tbody>
</table>

- **Job status:**
  - **R** – job is running.
  - **PD** – job is waiting for resource.
    - Reasons are usually (Resources) or (Priority).
  - Others commons reasons are **CA** (cancelled) and **CD** (completed).
sinfo example

sinfo -p general-compute

PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
general-compute* up 3-00:00:00 264 idle d07n07s[01-02],d07n08s[01-02], ...

- Node states:
  - alloc – all cores are in use.
  - mix – some cores are available.
  - idle – node is free. All cores are available.
  - down - node is down.
  - drained – node is offline.
sinfo example

More detailed sinfo query:

```bash
sinfo --exact --partition=general-compute --format="%15P %5a %10A %.4c %6m %6G %16f %t %N" | more
```

<table>
<thead>
<tr>
<th>PARTITION</th>
<th>AVAIL</th>
<th>NODES(A/I)</th>
<th>CPUS</th>
<th>MEMORY</th>
<th>GRES</th>
<th>FEATURES</th>
<th>STATE</th>
<th>NODELIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>general-compute* up</td>
<td>0/0</td>
<td>12</td>
<td>48000</td>
<td>(null)</td>
<td>IB,CPU-E5645</td>
<td>CPU</td>
<td>up</td>
<td>k13n17</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>k13n18s[01-02],k13n19s[01-02],k13n23s[01-02],k13n24s[01-02],</td>
</tr>
<tr>
<td>general-compute* up</td>
<td>0/151</td>
<td>12</td>
<td>48000</td>
<td>(null)</td>
<td>IB,CPU-E5645</td>
<td>CPU</td>
<td>idle</td>
<td>k13n17</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>k13n18s[01-02],k13n19s[01-02],k13n23s[01-02],k13n24s[01-02],</td>
</tr>
</tbody>
</table>
```
sview example
sbatch example

sbatch slurmHelloWorld

#!/bin/sh
#SBATCH --partition=general-compute
#SBATCH --time=00:15:00
#SBATCH --nodes=2        [in PBS -Inodes=2:ppn=16]
#SBATCH --ntasks-per-node=16
#SBATCH --job-name="hello_test"
#SBATCH --output test.out
#SBATCH --mail-user=usename@buffalo.edu
#SBATCH --mail-type=END
...

UB CENTER FOR COMPUTATIONAL RESEARCH
University at Buffalo The State University of New York
Commonly used SLURM variables

- `$SLURM_JOB_ID`
- `SLURM_JOB_NODELIST`
  - Node list in SLURM format; for example f16n[04,06].
- `$SLURM_NNODES`
  - Number of nodes
- `$SLURMTMPDIR`
  - `/scratch/jobid` ($PBSTMPDIR)
- `$SLURM_SUBMIT_DIR`
  - Directory from which the job was submitted ($PBS_O_WORKDIR).

**NOTE!** Jobs start in the `$SLURM_SUBMIT_DIR`. 
Task Launching

- The number of cores/processes must be computed.
  - The $PBS_NODEFILE does not exist, however it can be created if necessary.
- Intel-MPI mpirun and mpiexec are SLURM aware. **Note:** mpirun does not launch properly if nodes are undersubscribed.
  - The $PBS_NODEFILE is not necessary.
- srun will execute a command across nodes.
  - It can be used to generate the number of processors and a PBS-like nodelist, as well as launch MPI computations.
Node Sharing

- Compute nodes are shared among different jobs and users.
- In most cases, tasks will be limited to the number of cores and memory specified.
- The integration of CPUSets and SLURM makes this possible.
  - CPUSet is a Linux kernel level mechanism that can be used to control the processor memory utilization.
- The --exclusive flag will request the nodes as dedicated. The nodes will not be shared.
Interactive Job

- The `salloc` command requests the nodes.
- Once the nodes have been allocated to the job, then the user can login to the compute node.
  - The user is **not** logged into the compute node when the job starts.
- Only `ssh` can be used to login to the nodes assigned to a job.
- The job allocation can persist through logout.
Example of an Interactive Job

```
[cdc@rush:~]$ salloc --partition=general-compute --nodes=1 --time=01:00:00 --exclusive
salloc: Granted job allocation 54124
[cdc@rush:~]$ export | grep SLURM
declare -x SLURM_JOBID="54124"
declare -x SLURM_JOB_CPUS_PER_NODE="8"
declare -x SLURM_JOB_ID="54124"
declare -x SLURM_JOB_NODELIST="d07n35s01"
declare -x SLURM_JOB_NUM_NODES="1"
declare -x SLURM_NNODES="1"
declare -x SLURM_NODELIST="d07n35s01"
...
[cdc@rush:~]$ exit
exit
salloc: Relinquishing job allocation 54124
salloc: Job allocation 54124 has been revoked.
[cdc@rush:~]$`
```
Example of an Interactive Job

```bash
[cdc@rush ~]$ salloc --partition=general-compute -nodes=1 --time=01:00:00 --exclusive &
[1] 14269
[cdc@rush ~]$ salloc: Granted job allocation 4716
[cdc@rush ~]$  
```

**Note!**

Placing the `salloc` in the background allows the allocation to persist.

The user is **not** logged into the compute node when the job starts.
Job monitoring

- The **NEW slurmjobvis** is a graphical display of the activity on the node. CPU, memory, network, as well as GPU utilization are displayed.
- This an improved version of ccrjobvis.
- User can login using ssh to the compute nodes in the job.
More Information and Help

- **CCR SLURM** web page
- **Compute Cluster** web page
- **Remote Visualization** web page
- More sample SLURM scripts can be found in the /util/slurm-scripts directory on rush.
- Users can get assistance with the transition to SLURM by sending a request to **ccr-help@ccr.buffalo.edu**.
- Drop-in hours for assistance:
  - North Campus: Monday and Tuesday 9am-11am in Bell 107
  - COE: Friday 9am-11am in Visualization Room