SLURM: Resource Management and Job Scheduling Software

Advanced Computing Center for Research and Education

www.acc.re.vanderbilt.edu
What is SLURM?

* Simple Linux Utility for Resource Management
  * But it’s also a job scheduler!
  * Previously, ACCRE used Torque for resource management and Moab for job scheduling

* Originally developed at Lawrence Livermore National Laboratory, but now maintained and supported by SchedMD

* Open-source, GPL 2.0 licensing
* Excellent support team
* Supports plugins for extending or enhancing functionality
* Increasingly being used at academic research compute centers and national labs
SLURM Features

* Excellent performance
  * Able to process tens of thousands of jobs per hour (scalability) - as of June 2014, six of the top ten supercomputers were using SLURM
  * Multi-threaded
  * High throughput for smaller jobs (accepts up to 1,000 jobs per second)
  * Fault tolerant (backup server can take over transparently)

* Supports Control Groups (cgroups)
  * A cgroup is an OS (i.e. Linux) mechanism for aggregating and partitioning sets of tasks
  * Useful for systems where resources are shared among multiple users
  * Allows memory and CPU requests to be enforced on compute nodes

* Uses a database to store user and account information, as well as job statistics
SLURM Overview

Controller daemons
- slurmd (primary)
- slurmd (backup)

Slurmdbd (optional)

Other clusters

Database

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

Compute node daemons
- slurmd
- slurmd
- slurmd

http://slurm.schedmd.com/overview.html
## SLURM Commands

<table>
<thead>
<tr>
<th>Torque (old way)</th>
<th>SLURM (new way)</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>qsub [job_script]</td>
<td>sbatch [job_script]</td>
<td>Job submission</td>
</tr>
<tr>
<td>qstat or showq</td>
<td>squeue</td>
<td>Job status</td>
</tr>
<tr>
<td>qdel [JOB_ID]</td>
<td>scancel [JOB_ID]</td>
<td>Job deletion</td>
</tr>
<tr>
<td>pbsnodes</td>
<td>scontrol show nodes</td>
<td>Node list</td>
</tr>
<tr>
<td>qhold [JOB_ID]</td>
<td>scontrol hold [JOB_ID]</td>
<td>Job hold</td>
</tr>
<tr>
<td>qrls [JOB_ID]</td>
<td>scontrol release [JOB_ID]</td>
<td>Job release</td>
</tr>
<tr>
<td>qstat -a</td>
<td>sinfo</td>
<td>Cluster status</td>
</tr>
<tr>
<td>qsub -l</td>
<td>salloc -n [CPU_COUNT] --pty /bin/bash</td>
<td>Launch interactive job with CPU_COUNT cores</td>
</tr>
<tr>
<td></td>
<td>srunch [command]</td>
<td>Launch (parallel) job step</td>
</tr>
<tr>
<td></td>
<td>sacct</td>
<td>Displays accounting information for job(s)</td>
</tr>
</tbody>
</table>
Logging In and Example Scripts

To log in to a SLURM gateway:

* ssh VUNETID@slurmlogin.accre.vanderbilt.edu
* Torque gateways: ssh VUNETID@vmplogin.accre.vanderbilt.edu
* Hard-wired log ins to specific gateways are discouraged during the transition period

Download example scripts from ACCRE’s Github repository:

* setpkgs -a git
* git clone https://github.com/accre/SLURM.git
* If you have a Github account, please follow us!
sbatch

* sbatch [options] [script_file]
* Substitute for qsub
* Options tend to be more self-explanatory than Torque syntax
  * --nodes=1; --ntasks=4; --ntasks-per-node=4 ; --mem-per-cpu=1G
* Parses #SBATCH in a job script and/or accepts options from the command line (just like qsub)
* Including options from the command line:
  * sbatch --ntasks=4 script.slurm
  * Option from command line will override value in script
  ```bash
  #!/bin/bash
  #SBATCH --nodes=1
  #SBATCH --ntasks=4
  #SBATCH --mem-per-cpu=1G
  #SBATCH --time=0-00:02:00  # 2 minutes
  #SBATCH --output=my.stdout
  #SBATCH --mail-user=vunetid@vanderbilt.edu
  #SBATCH --mail-type=ALL
  #SBATCH --job-name="just_a_test"
  
  # This will be run once for a single process
  /bin/hostname
  
  # These commands will be run by each process
  # -l flag prepends the task number to each line of output
  srun -l /bin/hostname
  srun -l /bin/pwd
  ```
Torque vs SLURM

Differences:

- Node and CPU core counts are split into separate lines
- By default, SLURM combines stdout and stderr
- Other very minor syntactical differences
Parallel Computing Options

* Several paradigms possible within SLURM:
  * Distributed memory processing (e.g. MPI)
    * Requires explicit MPI code in order to run across multiple CPU cores and/or nodes
    * SLURM allocates 1 CPU core per task (i.e. process) by default
  * Multithreaded (i.e. shared memory) processing
    * Also requires explicit coding in a shared memory API (e.g. Posix Threads, OpenMP)
    * Generally run job on a single node, requesting 1 task and number of threads determined by --cpus-per-task option
  * Embarrassingly parallel processing
    * Multiple input/data files to perform the same analysis on
    * Job arrays allow you to index multiple files for processing within a single batch script
    * Can also generate multiple batch scripts and invoke "sbatch" on each
  * GPU (graphics processing units) processing
    * Requires explicit coding in CUDA (NVIDIA C-like API) or OpenCL
    * User must be in a GPU group
    * Two additional options required in batch script: --partition=gpu and --account=group_gpu
MPI Jobs

# Runs job in parallel across 3*8=24 CPU cores
# For SLURM, we recommend building MPI applications with Intel's MPI library (setpkgs -a intel_cluster_studio_compiler)
# OpenMPI and MPICH2 may require use of "mpiexec" rather than "srun"
# More details: http://slurm.schedmd.com/mpi_guide.html

#!/bin/bash
#SBATCH --mail-user=vunetid@vanderbilt.edu
#SBATCH --mail-type=ALL
#SBATCH --nodes=3
#SBATCH --tasks-per-node=8
#SBATCH --time=7-00:00:00  # 7 days
#SBATCH --mem=3840M
#SBATCH --mem-per-cpu=160M
#SBATCH --output=mpi_job_slurm.log
export I_MPI_PMI_LIBRARY=/usr/scheduler/slurm/lib/libpmi.so
date
echo $SLURM_JOB_NODELIST
time srun -n 24 ./test
date
Multithreaded Jobs

Typically run across multiple CPU cores within a single node
* By default, SLURM allocates 1 CPU core per task (i.e. process)
* Multithreaded programs typically use a single process that forks multiple threads that can then execute in parallel across multiple CPU cores
* Key option is (--cpus-per-task); set this to the number of threads that you would like to execute (max is 12)

```bash
#!/bin/bash
#SBATCH --mail-user=vunetid@vanderbilt.edu
#SBATCH --mail-type=ALL
#SBATCH --nodes=1
#SBATCH --tasks=1
#SBATCH --cpus-per-task=4  # 4 threads of execution
#SBATCH --time=2:00:00    # 2 hours
#SBATCH --mem=4G
#SBATCH --output=multithread.out
#SBATCH --job-name=multithreaded_example

# Launch multithreaded application
./hello
```
Job Arrays

* Submit and manage very similar jobs quickly and easily; only one batch script required

* In this example, three input files (vectorization_0.m, vectorization_1.m, vectorization_2.m) are processed using Matlab
  * Each array index gets its own CPU core

* --array=0-2 requests an array size of 3 (array indices=0,1,2)

* Support for different stride lengths (e.g. --array=0-1 0:2 would give indices of 0,2,4,6,8,10)

```
#!/bin/bash
#SBATCH --mail-user=vunetid@vanderbilt.edu
#SBATCH --mail-type=ALL
#SBATCH --ntasks=1
#SBATCH --time=2:00:00  # 2 hours
#SBATCH --mem=5G
#SBATCH --array=0-2
#SBATCH --output=matlab_array_job_slurm_%A_%a.out
echo "SLURM_JOBID: " $SLURM_JOBID
echo "SLURM_ARRAY_TASK_ID: " $SLURM_ARRAY_TASK_ID
echo "SLURM_ARRAY_JOB_ID: " $SLURM_ARRAY_JOB_ID
setpkgs -a matlab
matlab < vectorization_${SLURM_ARRAY_TASK_ID}.m
```

More details:
http://www.schedmd.com/slurmdocs/job_array.html
GPU Jobs

* Must be in a GPU group and include the `--partition` and `--account` options
* Single job has exclusive rights to allocated node (4 GPUs per node)

```bash
#!/bin/bash
#SBATCH --mail-user=vunetid@vanderbilt.edu
#SBATCH --mail-type=ALL
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --time=2:00:00  # 2 hours
#SBATCH --mem=100M
#SBATCH --output=gpu-job.log
#SBATCH --partition=gpu
#SBATCH --account=accre_gpu  # substitute appropriate group here
setpkgs -a hoomd
pwd
date
hoomd simple-script.py
date
```
#SBATCH Commands

<table>
<thead>
<tr>
<th>Torque (old way)</th>
<th>SLURM (new way)</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-l nodes=[count]</td>
<td>--nodes=[count]</td>
<td>Node count</td>
</tr>
<tr>
<td>-l ppn=[count]</td>
<td>--tasks-per-node=[count]</td>
<td>Processes per node</td>
</tr>
<tr>
<td></td>
<td>--ntasks=[count]</td>
<td>Total process count</td>
</tr>
<tr>
<td></td>
<td>--cpus-per-task=[count]</td>
<td>CPU cores per process</td>
</tr>
<tr>
<td></td>
<td>--nodelist=[nodes]</td>
<td>Job host preference</td>
</tr>
<tr>
<td></td>
<td>--exclude=[nodes]</td>
<td>Job host preference</td>
</tr>
<tr>
<td>-l walltime=[dd:hh:mm:ss]</td>
<td>--time=[min] or --time=[days-hh:mm:ss]</td>
<td>Wall clock limit</td>
</tr>
<tr>
<td>-l mem=[amount]</td>
<td>--mem=[amount][M</td>
<td>G</td>
</tr>
<tr>
<td>-l pmem=[amount]</td>
<td>--mem-per-cpu=[amount][M</td>
<td>G</td>
</tr>
<tr>
<td>-o [file_name]</td>
<td>--output=[file_name]</td>
<td>Standard output file</td>
</tr>
<tr>
<td>-e [file_name]</td>
<td>--error=[file_name]</td>
<td>Standard error file</td>
</tr>
<tr>
<td>-j oe</td>
<td>(default behavior)</td>
<td>Combine stdout and stderr</td>
</tr>
<tr>
<td>-t [array_spec]</td>
<td>--array=[array_spec]</td>
<td>Job arrays</td>
</tr>
<tr>
<td>-M [email_address]</td>
<td>--mail-user=[email_address]</td>
<td>Email address</td>
</tr>
<tr>
<td>-m abe</td>
<td>--mail-type=[BEGIN</td>
<td>END</td>
</tr>
<tr>
<td>-W group_list=[account]</td>
<td>--account=[account]</td>
<td>Account to charge</td>
</tr>
<tr>
<td>-N [name]</td>
<td>--job-name=[name]</td>
<td>Job name</td>
</tr>
<tr>
<td></td>
<td>--constrain=[attribute]</td>
<td>Request node attribute (e.g. “intel”)</td>
</tr>
<tr>
<td></td>
<td>--partition=[name]</td>
<td>Submit job to partition (e.g. “gpu”)</td>
</tr>
</tbody>
</table>
squeue

*squeue [options]*

* Used to view job and job step information for jobs managed by Slurm
* Substitute for qstat or showq
* docs: [http://slurm.schedmd.com/squeue.html](http://slurm.schedmd.com/squeue.html)

*Examples:*

* squeue --user=USER_ID (only show jobs for USER_ID)
* squeue --account=ACCOUNT_ID (only show jobs for ACCOUNT_ID)
* squeue --states=running (only display running jobs)
* squeue --format="%.10i %N" (only show job id and list of nodes)
* squeue --long (report more information for each job)
* squeue --start (report the expected start time of pending jobs)
* squeue --help (show all options)
sacct

*sacct [options]*

* Shows accounting data for all jobs and job steps (queries SLURM database)
* Substitute for tracejob
* docs: [http://slurm.schedmd.com/sacct.html](http://slurm.schedmd.com/sacct.html)

**Examples:**

* sacct --startime 12.04.14 (show jobs starting after midnight of Dec 4, 2014; default start time is the current day)
* sacct --j JOB_ID (show info for specified job)
* sacct --allusers (show jobs for all users; default is current user’s jobs only)
* sacct --accounts=ACCOUNT_ID (show jobs for the specified account)
* sacct --format="JobID,user,account,elapsed,TimeLimit,MaxRSS,MaxVMSize,ncpus"
* sacct --help (show options)
scontrol

scontrol [options] [command]

* View SLURM configuration and state; update job resource request
* Can be used as a substitute for checkjob (see example below)
* docs: http://slurm.schedmd.com/scontrol.html

Examples:
* scontrol show job JOB_ID (display job information; checkjob equivalent)
* scontrol hold JOB_ID (put job on hold, i.e. prevent job from starting)
* scontrol release JOB_ID (release job to run)
* scontrol show nodes (show hardware info for nodes on cluster)
* scontrol update JobID=JOB_ID Timelimit=1-12:00:00 (set/change wall time limit to 1 day, 12 hours)
* scontrol update dependency=JOB_ID (allow job to start after JOB_ID completes; can also accomplish this with sbatch --dependency=JOB_ID)
* scontrol --help (show options)
salloc

salloc [options] <command> [command args]

- Obtain a SLURM job allocation (nodes, CPU cores, etc), execute a command, and then release the allocation when the command is complete
- Can be used to run an interactive job
- docs: [http://slurm.schedmd.com/salloc.html](http://slurm.schedmd.com/salloc.html)

**Examples:**

- `salloc --ntasks 1 --time=1:00:00 srun --pty /bin/bash` (obtain resources for running single task for one hour on compute node)
- `salloc --nodes 2 --ntasks 16 --time=1:00:00 srun --pty /bin/bash` (request 2 nodes and 16 cores for interactive job)
sinfo

sinfo [options]
  * View information about SLURM nodes and partitions
  * Similar to Torque’s pbsnodes command
  * docs: http://slurm.schedmd.com/sinfo.html

Examples:
  * sinfo --Nel (shows nice summary of available vs. used resources)
  * sinfo --Node (displays info in a node-oriented format)
  * sinfo --states=IDLE (display info about idle nodes)
  * sinfo --help (show all options)
sreport

sreport [options] [command]

* Used to generate reports of job usage and cluster utilization for SLURM jobs saved to the SLURM database
* docs: http://slurm.schedmd.com/sreport.html

* Examples:

* sreport cluster utilization (show cluster utilization report)
* sreport user top (show top 10 cluster users based on total CPU time)
* sreport cluster AccountUtilizationByUser start=12-01-2014 (show account usage per user dating back to December 1, 2014)
* sreport job sizesbyaccount PrintJobCount (show number of jobs on a per-group basis)
* sreport —help (show all options)
## SLURM Environment Variables

<table>
<thead>
<tr>
<th>Torque</th>
<th>SLURM</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBS_JOBID</td>
<td>SLURM_JOBID</td>
<td>Job ID</td>
</tr>
<tr>
<td>PBS_O_WORKDIR</td>
<td>SLURM_SUBMIT_DIR</td>
<td>Submit Directory</td>
</tr>
<tr>
<td>PBS_O_HOST</td>
<td>SLURM_SUBMIT_HOST</td>
<td>Submit Host</td>
</tr>
<tr>
<td>PBS_NODEFILE</td>
<td>SLURM_JOB_NODELIST</td>
<td>Node List</td>
</tr>
<tr>
<td>PBS_ARRAYID</td>
<td>SLURM_ARRAY_TASK_ID</td>
<td>Job Array Index</td>
</tr>
<tr>
<td>SLURM_CPUS_PER_NODE</td>
<td></td>
<td>CPU cores per node</td>
</tr>
<tr>
<td>SLURM_NNODES</td>
<td></td>
<td>Node count</td>
</tr>
</tbody>
</table>

Full list: [http://slurm.schedmd.com/sbatch.html#lbAF](http://slurm.schedmd.com/sbatch.html#lbAF)
SLURM distributes wrappers for qsub, qstat, qdel, and other Torque commands that are compatible with many Torque scripts.

- You might not need to convert your job script at all, especially if your script is using fairly standard Torque commands.
- Please test!

However, in the long run we highly encourage users to port their scripts to SLURM syntax.

- As the SLURM code base expands and new functionality is added, it is unlikely that support for these wrappers will be maintained.
- Troubleshooting will be much more difficult if you're using Torque scripts.
- SLURM offers increased flexibility and control over jobs.
Useful Links

* **SLURM homepage:** http://slurm.schedmd.com/
* **ACCRE SLURM documentation:** http://www.accre.vanderbilt.edu/?page_id=2154
* **Documentation from other university research compute centers:**
  * https://rc.fas.harvard.edu/resources/running-jobs/
  * https://rec.uchicago.edu/docs/running-jobs/index.html
  * http://www.tchpc.tcd.ie/node/129
  * http://www.uppmax.uu.se/slurm-user-guide
  * http://www.hpc.ut.ee/user_guides/SLURM
**Dec 1-12**
- Begin user, group, and account conversion
- Contact and meet with heavy users (add to test cluster if desired)
- Begin documentation (slides, website)

**Dec 15-17**
- Hardware arrives
- Add 54 nodes to be managed by SLURM
- Add large users to SLURM database for testing

**Dec 22-Jan 2**
- Heavy testing
- Complete documentation (internal and external)

**Jan 5 (week of)**
- Begin weekly user training
- Begin incrementally moving smaller users and corresponding hardware over to SLURM

**Feb 2 (week of)**
- All users, groups, and accounts converted and on SLURM
- All hardware scheduled by SLURM

**Feb 16**
- Moab shutdown
Getting help from ACCRE

* ACCRE website FAQ and Getting Started pages: [http://www.accre.vanderbilt.edu/support](http://www.accre.vanderbilt.edu/support)
* ACCRE Help Desk: [http://www.accre.vanderbilt.edu/support/contact/submit_RT.php](http://www.accre.vanderbilt.edu/support/contact/submit_RT.php)

* Tickets default to non-rush queue, which is for tickets about an issue which only impacts you and/or which can wait until the next business day.
* Rush tickets are for issues which would impact the cluster as a whole.
* Rush queue pages ACCRE personnel (we normally only have staff on site during normal business hours), so please think before opening up a rush ticket at 3 AM!
* If you open up a rush ticket - day or night - please plan on being available to work with ACCRE personnel to resolve the issue.
* Once the issue (rush or non-rush) is resolved we would appreciate it if you would let us know.
Getting help from ACCRE

Advanced Computing Center for Research & Education

Enabling Researcher-Driven Innovation and Exploration

Helpdesk Request

We ask that you first peruse the FAQ page and the other User Support pages in the menu at left before submitting a help request.

If you have an ACCRE account (cluster or storage) and are experiencing difficulties with any ACCRE Services, or if you just have a comment you would like to pass on, please complete the following form. Your request / comments will be forwarded to ACCRE Staff for review.

If you do not have an ACCRE account and have a general inquiry about our services or would like someone to contact you, please feel free to contact us.

Note: If you don’t receive a confirmation email after you submit this form, we don’t have your ticket. If you have trouble with this form, please contact us immediately using our alternate form.

No confidential, export controlled, or proprietary information is allowed in any ACCRE Request Tracker help tickets.

All tickets are acknowledged on the same day if during business hours or the next day if after hours. If this is an urgent issue that requires waking an ACCRE staff member AND you will be online for the next two hours to work with the ACCRE staff person, please click here.

* (denotes required field)
Please select a category*

Select

Name*

Email Address*

Subject*

Message
Getting help from ACCRE

* accre-forum mailing list.
* ACCRE office hours: open a help desk ticket and mention that you would like to work personally with one of us in the office. The person most able to help you with your specific issue will contact you to schedule a date and time for your visit.