Job management with SLURM

For complete SLURM documentation, see https://slurm.schedmd.com/. Here we only show simple examples with system-specific instructions.

HAL Slurm Wrapper Suite (Recommended)

Introduction

The HAL Slurm Wrapper Suite was designed to help users use the HAL system easily and efficiently. The current version is "swsuite-v0.4", which includes

srun (slurm command) swrun : request resources to run interactive jobs.

sbatch (slurm command) swbatch : request resource to submit a batch script to Slurm.

squeue (slurm command) swqueue : check current running jobs and computational resource status.

> The Slurm Wrapper Suite is designed with people new to Slurm in mind and simplifies many aspects of job submission in favor of automation. For advanced use cases, the native Slurm commands are still available for use.

Rule of Thumb

- Minimize the required input options.
- Consistent with the original "slurm" run-script format.
- Submits job to suitable partition based on the number of GPUs needed (number of nodes for CPU partition).

Usage

> **Request Only As Much As You Can Make Use Of**
>
> **Many applications** require some amount of modification to make use of more than one GPUs for computation. **Almost all programs** require nontrivial optimizations to be able to run efficiently on more than one node (partitions gpux8 and larger). Monitor your usage and avoid occupying resources that you cannot make use of.

- **swrun -p <partition_name> -c <cpu_per_gpu> -t <walltime> -r <reservation_name>**
  - <partition_name> **(required)** : cpun1, cpun2, cpun4, cpun8, gpux1, gpux2, gpux3, gpux4, gpux8, gpux12, gpux16.
  - <cpu_per_gpu> (optional) : 16 cpus (default), range from 16 cpus to 40 cpus.
  - <walltime> (optional) : 4 hours (default), range from 1 hour to 24 hours in integer format.
  - <reservation_name> (optional) : reservation name granted to user.
  - **example:** swrun -p gpux4 -c 40 -t 24 (request a full node: 1x node, 4x gpus, 160x cpus, 24x hours)
  - **Using interactive jobs to run long-running scripts is not recommended.** If you are going to walk away from your computer while your script is running, consider submitting a batch job. Unattended interactive sessions can remain idle until they run out of walltime and thus block out resources from other users. **We will issue warnings when we find resource-heavy idle interactive sessions and repeated offenses may result in revocation of access rights.**
- **swbatch <run_script>**
  - <run_script> **(required)** : same as original slurm batch.
  - <job_name> (optional) : job name.
  - <output_file> (optional) : output file name.
  - <error_file> (optional) : error file name.
  - <partition_name> **(required)** : cpun1, cpun2, cpun4, cpun8, gpux1, gpux2, gpux3, gpux4, gpux8, gpux12, gpux16.

- <cpu_per_gpu> (optional) : 16 cpus (default), range from 16 cpus to 40 cpus.
- <walltime> (optional) : 24 hours (default), range from 1 hour to 24 hours in integer format.
- <reservation_name> (optional) : reservation name granted to user.
- **example:** swbatch demo.swb

demo.swb

```
#!/bin/bash
#SBATCH --job-name="demo"
#SBATCH --output="demo.%j.%N.out"
#SBATCH --error="demo.%j.%N.err"
#SBATCH --partition=gpux1
#SBATCH --time=4

srun hostname
```

- **swqueue**
  - **example:** swqueue

New Job Queues (SWSuite only)

Under currently policy, jobs requesting more than 5 nodes will require a reservation. Otherwise, they will be held by the scheduler and will not execute.

Partition Name	Priority	Max Walltime	Nodes Allowed	Min-Max CPUs Per Node Allowed	Min-Max Mem Per Node Allowed	GPU Allowed	Local Scratch	Description
gpux1	normal	24 hrs	1	16-40	19.2-48 GB	1	none	designed to access 1 GPU on 1 node to run sequential and /or parallel jobs.
gpux2	normal	24 hrs	1	32-80	38.4-96 GB	2	none	designed to access 2 GPUs on 1 node to run sequential and/or parallel jobs.
gpux3	normal	24 hrs	1	48-120	57.6-144 GB	3	none	designed to access 3 GPUs on 1 node to run sequential and/or parallel jobs.
gpux4	normal	24 hrs	1	64-160	76.8-192 GB	4	none	designed to access 4 GPUs on 1 node to run sequential and/or parallel jobs.
gpux8	normal	24 hrs	2	64-160	76.8-192 GB	8	none	designed to access 8 GPUs on 2 nodes to run sequential and/or parallel jobs.
gpux12	normal	24 hrs	3	64-160	76.8-192 GB	12	none	designed to access 12 GPUs on 3 nodes to run sequential and/or parallel jobs.
gpux16	normal	24 hrs	4	64-160	76.8-192 GB	16	none	designed to access 16 GPUs on 4 nodes to run sequential and/or parallel jobs.
cpun1	normal	24 hrs	1	96-96	115.2-115.2 GB	0	none	designed to access 96 CPUs on 1 node to run sequential and/or parallel jobs.
cpun2	normal	24 hrs	2	96-96	115.2-115.2 GB	0	none	designed to access 96 CPUs on 2 nodes to run sequential and/or parallel jobs.
cpun4	normal	24 hrs	4	96-96	115.2-115.2 GB	0	none	designed to access 96 CPUs on 4 nodes to run sequential and/or parallel jobs.
cpun8	normal	24 hrs	8	96-96	115.2-115.2 GB	0	none	designed to access 96 CPUs on 8 nodes to run sequential and/or parallel jobs.
cpun16	normal	24 hrs	16	96-96	115.2-115.2 GB	0	none	designed to access 96 CPUs on 16 nodes to run sequential and/or parallel jobs.
cpu_mini	normal	24 hrs	1	8-8	9.6-9.6 GB	0	none	designed to access 8 CPUs on 1 node to run tensorboard jobs.

HAL Wrapper Suite Example Job Scripts

New users should check the example job scripts at **"/opt/samples/runscripts"** and request adequate resources.

Script Name	Job Type	Partition	Walltime	Nodes	CPU	GPU	Memory	Description
run_gpux1_16cpu_24hrs.sh	interactive	gpux1	24 hrs	1	16	1	19.2 GB	submit interactive job, 1x node for 24 hours w/ 12x CPU 1x GPU task in "gpux1" partition.

run_gpux2_32cpu_24hrs.sh	interactive	gpux2	24 hrs	1	32	2	38.4 GB	submit interactive job, 1x node for 24 hours w/ 24x CPU 2x GPU task in "gpux2" partition.
sub_gpux1_16cpu_24hrs.swb	batch	gpux1	24 hrs	1	16	1	19.2 GB	submit batch job, 1x node for 24 hours w/ 12x CPU 1x GPU task in "gpux1" partition.
sub_gpux2_32cpu_24hrs.swb	batch	gpux2	24 hrs	1	32	2	38.4 GB	submit batch job, 1x node for 24 hours w/ 24x CPU 2x GPU task in "gpux2" partition.
sub_gpux4_64cpu_24hrs.swb	batch	gpux4	24 hrs	1	64	4	76.8 GB	submit batch job, 1x node for 24 hours w/ 48x CPU 4x GPU task in "gpux4" partition.
sub_gpux8_128cpu_24hrs.swb	batch	gpux8	24 hrs	2	128	8	153.6 GB	submit batch job, 2x node for 24 hours w/ 96x CPU 8x GPU task in "gpux8" partition.
sub_gpux16_256cpu_24hrs.swb	batch	gpux16	24 hrs	4	256	16	153.6 GB	submit batch job, 4x node for 24 hours w/ 192x CPU 16x GPU task in "gpux16" partition.

Native SLURM style

Available Queues

Name	Priority	Max Walltime	Max Nodes	Min/Max CPUs	Min /Max RAM	Min/Max GPUs	Description
cpu	normal	24 hrs	16	1-96	1.2GB per CPU	0	Designed for CPU-only jobs
gpu	normal	24 hrs	16	1-160	1.2GB per CPU	0-64	Designed for jobs utilizing GPUs
debug	high	4 hrs	1	1-160	1.2GB per CPU	0-4	Designed for single-node, short jobs. Jobs submitted to this queue receive higher priority than other jobs of the same user.

Submit Interactive Job with "srun"

```
srun --partition=debug --pty --nodes=1 \
     --ntasks-per-node=16 --cores-per-socket=4 \
     --threads-per-core=4 --sockets-per-node=1 \
     --mem-per-cpu=1200 --gres=gpu:v100:1 \
     --time 01:30:00 --wait=0 \
     --export=ALL /bin/bash
```

Submit Batch Job

```
sbatch [job_script]
```

Check Job Status

```
squeue               # check all jobs from all users
squeue -u [user_name] # check all jobs belong to user_name
```

Cancel Running Job

```
scancel [job_id] # cancel job with [job_id]
```

PBS style

Some PBS commands are supported by SLURM.

Check Node Status

```
pbsnodes
```

Check Job Status

```
qstat -f [job_number]
```

Check Queue Status

```
qstat
```

Delete Job

```
qdel [job_number]
```

Submit Batch Job

```
$ cat test.pbs
#!/usr/bin/sh
#PBS -N test
#PBS -l nodes=1
#PBS -l walltime=10:00

hostname
$ qsub test.pbs
107
$ cat test.pbs.o107
hal01.hal.ncsa.illinois.edu
```