# **Running Jobs with SLURM**





### **Overview**

All three clusters, Puma, Ocelote, and ElGato, use SLURM for resource management and job scheduling.

### Additional SLURM Resources and Examples

Link	Description
Official SchedMD User Documentation	Official SchedMD user documentation. Includes detailed information on SLURM directives and commands.
PBS SLURM Rosetta Stone	Table for converting some common PBS job directives to SLURM syntax.
Puma Quick Start	HPC Quick Start guide. If you have never submitted a batch job before, this is a great place to start.
Job Examples	Basic SLURM example scripts. Includes PBS scripts for comparison.
Even More Job Examples!	Growing repository of example SLURM submission scripts
Intro to HPC	A recorded video presentation of our Intro to HPC workshop. Keep your eyes peeled for periodic announcements in the HPC listserv on upcoming live sessions!=

## SLURM and System Commands

Command	Purpose	Example(s)			
Native Slurm Commands					
sbatch	Submits a batch script for execution	sbatch script.slurm			
srun	Run parallel jobs. Can be in place of mpirun/mpiexec. Can be used interactively as well as in batch scripts	srun -n 1mpi=pmi2 a. out			
salloc	Requests a session to work on a compute node interactively	see: Interactive Sessions section below			
squeue	Checks the status of pending and running jobs	squeuejob \$JOBID squeueuser \$NETID			
scancel	Cancel a running or pending job	scancel \$JOBID scancel -u \$NETID			
scontrol hold	Place a hold on a job to prevent it from being executed	scontrol hold \$JOBID			
scontrol release	Releases a hold placed on a job allowing it to be executed	scontrol release \$JOBID			
	System Commands				
va	Displays your group membership, your account usage, and CPU allocation. Short for "view allocation"	va			
interactive	Shortcut for quickly requesting an interactive job. Use "interactivehelp" to get full usage.	interactive -a \$GROUP_NAME			
job-history	Retrieves a running or completed job's history in a user-friendly format	job-history \$JOBID			
seff	Retrieves a completed job's memory and CPU efficiency	seff \$JOBID			
past-jobs	past-jobs Retrieves past jobs run by user. Can be used with option "-d N" to search for jobs run in the past N days.				
job-limits	View your group's job resource limits and current usage.	job-limits \$GROUP			
nodes-busy	Display a visualization of nodes on a cluster and their usage	nodes-busyhelp			
system-busy	Display a text-based summary of a cluster's usage	system-busy			
cluster-busy	Display a visualization of all three cluster's overall usage	cluster-busyhelp			

## **Batch Job Directives**

Command	Purpose
#SBATCH account=group_ name	Specify the account where hours are charged. Don't know your group name? Run the command "va" to see which groups you belong to
#SBATCH partition=part ition_name	Set the job partition. This determines your job's priority and the hours charged. See Job Partition Requests below for additional information
#SBATCH time=DD-HH:MM: SS	Set the job's runtime limit in days, hours, minutes, and seconds
#SBATCH	Allocate N nodes to your job.
	For non-MPI enabled jobs, this should be set to "nodes=1" to ensure access to all requested resources and prevent memory errors.
#SBATCH ntasks=N	ntasks specifies the number of tasks (or processes) the job will run. For MPI jobs, this is the number of MPI processes. Most of the time, you can use ntasks to specify the number of CPUs your job needs. However, in some odd cases you might run into issues. For example, see: Using Matlab
#SBATCH cpus-per-	By default, you will be allocated one CPU/task. This can be increased by including the additional directivecpus-per-task.
task=M	The number of CPUs a job is allocated is cpus/task * ntasks, or M*N
#SBATCH mem=Ngb	Select N gb of memory <b>per node</b> . If "gb" is not included, this value defaults to MB. Directivesmem andmem-per-cpu are mutually exclusive.
#SBATCHmem- per-cpu=Ngb	Select N GB of memory per CPU. Valid values can be found in the Node Types/Example Resource Requests section below. If "gb" is not included, this value defaults to MB.
#SBATCH gres=gpu:N	Optional: Request N GPUs.
#SBATCH constraint=hi_ mem	Optional: Request a high memory node (Ocelote and Puma only).
#SBATCH array=N-M	Submits an array job from indices N to M
#SBATCHjob- name=JobName	Optional: Specify a name for your job. This will not automatically affect the output filename.
<pre>#SBATCH -e output_filenam e.err #SBATCH -o output_filenam e.out</pre>	Optional: Specify output filename(s). If -e is missing, stdout and stderr will be combined.
#SBATCH open- mode=append	Optional: Append your job's output to the specified output filename(s).
#SBATCH mail- type=BEGIN END  FAIL ALL	Optional: Request email notifications. Beware of mail bombing yourself.
#SBATCH mail- user=email@add ress.xyz	Optional: Specify email address. If this is missing, notifications will go to your UArizona email address by default.
#SBATCH exclusive	Optional: Request exclusive access to node.
#SBATCH export=VAR	Optional: Export a comma-delimited list of environment variables to a job.
#SBATCH export=all (default)	Optional: Export your working environment to your job.
#SBATCH export=none	Optional: Do not export working environment to your job.

## **SLURM Environment Variables**

Variable	Purpose	Example Value
\$SLURM_ARRAY_ JOB_ID	Job array's parent ID	399124
\$SLURM_ARRAY_ TASK_COUNT	Total number of subjobs in the array	4
\$SLURM_ARRAY_ TASK_ID	Job index number (unique for each job in the array)	1
\$SLURM_ARRAY_ TASK_MAX	Maximum index for the job array	7
\$SLURM_ARRAY_ TASK_MIN	Minimum index for the job array	1
\$SLURM_ARRAY_ TASK_STEP	Job array's index step size	2
\$SLURM_CLUSTE R_NAME	Which cluster your job is running on	elgato
\$SLURM_CONF	Points to the SLURM configuration file	/var/spool/slurm/d /conf-cache/slurm. conf
\$SLURM_CPUS_O N_NODE	Number of CPUs allocated to target node	3
\$SLURM_GPUS_O N_NODE	Number of GPUs allocated to the target node	1
\$SLURM_GPUS_P ER_NODE	Number of GPUs per node. Only set ifgpus-per-node is specified	1
\$SLURM_JOB_AC COUNT	Account being charged	groupname
\$SLURM_JOB_GP US	The global GPU IDs of the GPUs allocated to the job. Only set in batch and interactive jobs.	0
\$SLURM_JOB_ID	Your SLURM Job ID	399072
\$SLURM_JOB_CP US_PER_NODE	Number of CPUs per node. This can be a list if there is more than one node allocated to the job. The list has the same order as SLURM_JOB_NODELIST	3,1
\$SLURM_JOB_NA ME	The job's name	interactive
\$SLURM_JOB_NO DELIST	The nodes that have been assigned to your job	gpu[73-74]
\$SLURM_JOB_NU M_NODES	The number of nodes allocated to the job	2
\$SLURM_JOB_PA RTITION	The job's partition	standard
\$SLURM_JOB_QOS	The job's QOS/Partition	qos_standard_part
\$SLURM_JOB_US ER	The username of the person who submitted the job	netid
\$SLURM_JOBID	Same as SLURM_JOB_ID, your SLURM Job ID	399072
\$SLURM_MEM_PE R_CPU	The memory/CPU ratio allocated to the job	4096
\$SLURM_NNODES	Same as SLURM_JOB_NUM_NODES - the number of nodes allocated to the job	2
\$SLURM_NODELI ST	Same as SLURM_JOB_NODELIST, The nodes that have been assigned to your job	gpu[73-74]

\$SLURM_NPROCS	The number of tasks allocated to your job	4
\$SLURM_NTASKS	Same as SLURM_NPROCS, the number of tasks allocated to your job	4
\$SLURM_SUBMIT _DIR	The directory where sbatch was used to submit the job	/home/u00/netid
\$SLURM_SUBMIT _HOST	The hostname where sbatch was used to submit the job	wentletrap.hpc. arizona.edu
\$SLURM_TASKS_ PER_NODE	The number of tasks to be initiated on each node. This can be a list if there is more than one node allocated to the job. The list has the same order as SLURM_JOB_NODELIST	3,1
\$SLURM_WORKIN G_CLUSTER	Valid for interactive jobs, will be set with remote sibling cluster's IP address, port and RPC version so that any sruns will know which cluster to communicate with.	elgato:foo:0000:0000: 000

## **SLURM Reason Codes**

Sometimes, if you check a pending job using squeue, there are some messages that show up under Reason indicating why your job may not be running. Some of these codes are non-intuitive so a human-readable translation is provided below:

Reason	Explanation
AssocGrpCpuLim it	This is a per-group limitation on the number of CPUs that can be used simultaneously by all group members. Your job is not running because this limit has been reached. Check your group's limits using "job-limits <group_name>".</group_name>
AssocGrpMemLim it	This is a per-group limitation on the amount of memory that can be used simultaneously by all group members. Your job is not running because this limit has been reached. Check your group's limits using "job-limits <group_name>".</group_name>
AssocGrpCPUMin utesLimit	Either your group is out of CPU hours or your job will exhaust your group's CPU hours.
AssocGrpGRES	This is a per-group limitation on the number of GPUs that can be used simultaneously by all group members. Your job is not running because this limit has been reached. Check your group's limits using "job-limits <group_name>".</group_name>
Dependency	Your job depends on the completion of another job. It will wait in queue until the target job completes.
QOSMaxWallDura tionPerJobLimi t	Your job's time limit exceeds the max allowable and will never run. To see an individual job's limits, run "job-limits <group_name>".</group_name>
Nodes_required _for_job_are_	This very long message simply means your job is waiting in queue until there is enough space for it to run
DOWN, _DRAINED_or_re served_	
or_jobs_in_hig her_priority_	
partitions	
Priority	Your job is waiting in queue until there is enough space for it to run.
QOSMaxCpuPerUs erLimit	This is a per-user limitation on the number of CPUs that you can use simultaneously among all of your jobs. Your job is not running because this limit has been reached. Check your user limits using "job_limits <group_name>".</group_name>
ReqNodeNotAvai l, Reserved for maintenance	Your job's time limit overlaps with an upcoming maintenance window. Run "uptime_remaining" to see when the system will go offline. If you remove and resubmit your job with a shorter walltime that does not overlap with maintenance, it will likely run. Otherwise, it will remain pending until after the maintenance window.
Resources	Your job is waiting in queue until the required resources are available.

### **Job Partition Requests**

Partition	SLURM	Details
standard	#SBATCH account= <pi group=""> #SBATCH partition=standard</pi>	Consumes your group's standard allocation. These jobs cannot be interrupted.
windfall	#SBATCH partition=windfall	Does not consume your group's standard allocation. Jobs may be interrupted and restarted by higher- priority jobs. Theaccount flag needs to be omitted or an error will occur.
high_priority	<pre>#SBATCH account=<pi group=""> #SBATCH partition=high_pri ority #SBATCH qos=user_qos_<pi GROUP&gt;</pi </pi></pre>	Available for groups who have purchased compute resources.
qualified	<pre>#SBATCH account=<pi group=""> #SBATCH partition=standard #SBATCH qos=qual_qos_<pi group=""></pi></pi></pre>	Available for groups that have submitted a special project request.

## **SLURM Output Filename Patterns**

SLURM offers ways to make your job's output filenames customizable through the use of character replacements. A table is provided below as a guide with some examples. Variables may be used or combined as desired. Note: character replacements may also be used with other SBATCH directives such as error filename, input filename, and job name.

Variable	Meaning	Example Slurm Directive(s)	Output
%A	A job array's main job ID	#SBATCHarray=1-2 #SBATCH -o %A.out #SBATCHopen-mode=append	12345.out
%a	A job array's index number	#SBATCHarray=1-2 #SBATCH -o %A_%a.out	12345_1.out 12345_2.out
۶J	Job ID plus stepid	#SBATCH -o %J.out	12345.out
۶j	Job ID	#SBATCH -o %j.out	12345.out
%N	Hostname of the first compute node allocated to the job	#SBATCH -o %N.out	r1u11n1.out
%u	Username	#SBATCH -o %u.out	netid.out
%x	Jobname	#SBATCHjob-name=JobName #SBATCH -o %x.out	JobName.out

## Node Types/Example Resource Requests

#### **Standard Nodes**

Cluster	Max CPUs	Mem/CPU	Max Mem	Sample Request Statement
ElGato	16	4gb	62gb	#SBATCHnodes=1 #SBATCHntasks=16 #SBATCHmem-per-cpu=4gb
Ocelote	28	6gb	168gb	#SBATCHnodes=1 #SBATCHntasks=28 #SBATCHmem-per-cpu=6gb
Puma	94	5gb	470gb	#SBATCHnodes=1 #SBATCHntasks=94 #SBATCHmem-per-cpu=5gb

### **GPU** Nodes

During the quarterly maintenance cycle on April 27, 2022 the ElGato K20s and Ocelote K80s were removed because they are no longer supported by Nvidia.

GPU jobs are requested using the generic resource, or --gres, SLURM directive. In general, the directive to request N GPUs will be of the form: --gres=gpu:N

Cluster	Max CPUs	Mem/CPU	Max Mem	Sample Request Statement
Ocelote	28	8gb	224gb	#SBATCHnodes=1 #SBATCHntasks=28 #SBATCHmem-per-cpu=8gb #SBATCHgres=gpu:1
Puma <sup>1</sup>	94	5gb	470gb	#SBATCHnodes=1 #SBATCHntasks=94 #SBATCHmem-per-cpu=5gb #SBATCHgres=gpu:1
1. Up to four		guested on Dum	o on o oingle Cl	All node with gree growth 2, 2, or 4

1 Up to four GPUs may be requested on Puma on a single GPU node with --gres=gpu:1, 2, 3, or 4

### **High Memory Nodes**

When requesting a high memory node, include both the memory/CPU and constraint directives

Cluster	Max CPUs	Mem/CPU	Max Mem	Sample Request Statement
Ocelote	48	41gb	2015gb	<pre>#SBATCHnodes=1 #SBATCHntasks=48 #SBATCHmem-per-cpu=41gb #SBATCHconstraint=hi_mem</pre>
Puma	94	32gb	3000gb	<pre>#SBATCHnodes=1 #SBATCHntasks=94 #SBATCHmem-per-cpu=32gb #SBATCHconstraint=hi_mem</pre>

### **Interactive Jobs**

When you are on a login node, you can request an interactive session on a compute node. This is useful for checking available modules, testing submission scripts, compiling software, and running programs directly from the command line. We have a built-in shortcut command that will allow you to quickly and easily request a session by simply entering: interactive

When you request a session, the full salloc command being executed will be displayed for verification/copying/editing/pasting purposes. For example:

(ocelote) [netid@junonia ~]\$ interactive Run "interactive -h for help customizing interactive use" Submitting with /usr/local/bin/salloc --job-name=interactive --mem-per-cpu=4GB --nodes=1 --ntasks=1 -time=01:00:00 --account=windfall --partition=windfall salloc: Pending job allocation 531843 salloc: job 531843 queued and waiting for resources salloc: job 531843 has been allocated resources salloc: Granted job allocation 531843 salloc: Waiting for resource configuration salloc: Nodes il6n1 are ready for job [netid@il6n1 ~]\$

Notice in the example above how the command prompt changes once your session starts. When you're on a login node, your prompt will show "junonia" or "wentletrap". Once you're in an interactive session, you'll see the name of the compute node you're connected to.

If no options are supplied to the command interactive, your job will automatically run using the windfall partition for one hour using one CPU. To use the standard partition, include the flag "-a" followed by your group's name. To see all the customization options:

```
(ocelote) [netid@junonia ~]$ interactive -h
Usage: /usr/local/bin/interactive [-x] [-g] [-N nodes] [-m memory per core] [-n ncpus per node] [-Q
optional qos] [-t hh::mm:ss] [-a account to charge]
```

You may also create your own salloc commands using any desired SLURM directives for maximum customization.

### **MPI Jobs**

#### OpenMPI

For openmpi the important variables are set by default, so you do not need to include them in your scripts.

#### Default OpenMPI variables

```
export SBATCH_GET_USER_ENV=1
export OMPI_MCA_btl_openib_cpc_include=rdmacm
export OMPI_MCA_btl_openib_if_include=bnxt_re1
export OMPI_MCA_btl_openib_rroce_enable=1
export OMPI_MCA_btl=vader,self,openib
export OMPI_MCA_oob_tcp_if_include=eth1
```

#### Intel MPI

For Intel MPI, these variables are set for you:

module unload openmpi3 gnu8

If you're using Intel MPI with mpirun and are getting errors, try replacing mpirun -np \$NPROCESSES with:

srun -n \$NPROCESSES --mpi=pmi2

### **Parallel Work**

To make proper use of a supercomputer, you will likely want to use the benefit of many cores. Puma has 94 cores in each node available to Slurm. The exception to that is running hundreds or thousands of jobs using High Throughput Computing.

We have a training course which explains the concepts and terminology of parallel computing with some examples. Introduction to Parallel Computing

This practical course in Parallel Analysis in R is also useful