

Slurm Script

Basic Slurm script commands

Slurm script command	Description
<code>#!/bin/bash</code>	Sets the shell that the job will be executed on the compute node
<code>#SBATCH --ntasks=1</code> <code>#SBATCH --n1</code>	Requests for 1 processors on task, usually 1 cpu as 1 cpu per task is default.
<code>#SBATCH --time=0-05:00</code> <code>#SBATCH -t 0-05:00</code>	Sets the maximum runtime of 5 hours for your job
<code>#SBATCH --mail-user= <email></code>	Sets the email address for sending notifications about your job state.
<code>#SBATCH --mail-type=BEGIN</code> <code>#SBATCH --mail-type=END</code> <code>#SBATCH --mail-type=FAIL</code> <code>#SBATCH --mail-type=REQUEUE</code> <code>#SBATCH --mail-type=ALL</code>	Sets the scheduling system to send you email when the job enters the following states: BEGIN,END,FAIL,REQUEUE,ALL
<code>#SBATCH --job-name=my-named-job</code>	Sets the Jobs name

Slurm script commands

Slurm script command	Description
#SBATCH --ntasks=X	Requests for X tasks. When cpus-per-task=1 (and this is the default) this requests X cores. When not otherwise constraint these CPUs may be running on any node
#SBATCH --nodes=X	Request that a minimum of X nodes be allocated to this job
#SBATCH --nodes=X-Y	Request that a minimum of X nodes and a maximum of Y nodes be allocated to this job
#SBATCH --cpus-per-task=X	Request that a minimum of X CPUs per task be allocated to this job
#SBATCH --tasks-per-node=X	Requests minimum of X task be allocated per node

Slurm script commands

Slurm script commands	Description of effects
<code>#SBATCH --ntasks=1</code> <code>#SBATCH --cpus-per-task=1</code>	Requests 1 CPU (Serial) <code>cpus-per-task</code> is set to 1 by default and may be omitted.
<code>#SBATCH --cpus-per-task=X</code> <code>#SBATCH --ntasks=1</code> <code>#SBATCH --nodes=1</code>	Requests for X CPUs in 1 task on 1 node (OpenMP) Both <code>ntasks</code> and <code>nodes</code> are set to 1 by default and may be omitted
<code>#SBATCH --ntasks=X</code> <code>#SBATCH --tasks-per-node=X</code> <code>#SBATCH --cpus-per-task=1</code>	Requests for X CPUs and tasks on 1 node (OpenMP) <code>cpus-per-task</code> is set to 1 by default and may be omitted.
<code>#SBATCH --ntasks=X</code> <code>#SBATCH --nodes=1</code> <code>#SBATCH --cpus-per-task=1</code>	Requests for X CPUs and tasks on 1 node (OpenMP) <code>cpus-per-task</code> is set to 1 by default and may be omitted.

Slurm script commands

Slurm script commands	Description of effects
<code>#SBATCH --ntasks=X</code> <code>#SBATCH --cpus-per-task=1</code>	Requests X CPUs and tasks (MPI) <code>cpus-per-task</code> is set to 1 by default and may be omitted.
<code>#SBATCH --ntasks=X</code> <code>#SBATCH --ntasks-per-node=Y</code> <code>#SBATCH --cpus-per-task=1</code>	Requests for X CPUs and tasks with Y CPUs and tasks per node (MPI) <code>cpus-per-task</code> is set to 1 by default and may be omitted.
<code>#SBATCH --ntasks=X</code> <code>#SBATCH --nodes=1</code> <code>#SBATCH --cpus-per-task=1</code>	Requests for X CPUs and tasks on the same node, <code>cpus-per-task</code> is set to 1 by default and may be omitted.
<code>#SBATCH --ntasks=X</code> <code>#SBATCH --nodes=1</code> <code>#SBATCH --cpus-per-task=1</code>	Requests for X CPUs and tasks on the 1 node <code>cpus-per-task</code> is set to 1 by default and may be omitted.

Slurm script commands

PBS script command	Description
#SBATCH --mem=4000	Requests 4000 MB of memory in total
#SBATCH --mem-per-cpu=4000	Requests 4000 MB of memory per cpu
#SBATCH --licenses=sas:2	Requests 2 SAS licenses
#SBATCH --gres=gpu:1	Requests that your job get 1 GPU allocated per node
#SBATCH --exclusive	Requests that your job run only on nodes with no other running jobs
#SBATCH --dependency=after:job_id1	Requests that the the job start after job (jobid1) has started
#SBATCH --dependency=afterany:job_id1, job_i2	Requests that the the job start after ether job (jobid1) or job (jobud2) has finished
#SBATCH --dependency=afterok:job_id1	Requests that the the job start after job (jobid1) has finished successfully

SLURM Environment Variables

Environment Variable	Description
SLURM_JOB_NAME	User specified job name
SLURM_JOB_ID	Unique slurm job id
SLURM_NNODES	Number of nodes allocated to the job
SLURM_NTASKS	Number of tasks allocated to the job
SLURM_ARRAY_TASK_ID	Array index for this job
SLURM_ARRAY_TASK_MAX	Total number of array indexes for this job
SLURM_MEM_PER_CPU	Memory allocated per CPU
SLURM_JOB_NODELIST	List of nodes on which resources are allocated to Job
SLURM_JOB_CPUS_PER_NODE	Number of CPUs allocated per Node
SLURM_JOB_PARTITION	List of Partition(s) that the job is in.
SLURM_JOB_ACCOUNT	Account under which this job is run.

Slurm script commands

PBS script command	Description
#SBATCH --mem=4000	Requests 4000 MB of memory in total
#SBATCH --mem-per-cpu=4000	Requests 4000 MB of memory per cpu
#SBATCH --licenses=sas:2	Requests 2 SAS licenses
#SBATCH --gres=gpu:1	Requests that your job get 1 GPU allocated per node
#SBATCH --exclusive	Requests that your job run only on nodes with no other running jobs
#SBATCH --dependency=after:job_id1	Requests that the the job start after job (jobid1) has started
#SBATCH --dependency=afterany:job_id1, job_i2	Requests that the the job start after ether job (jobid1) or job (jobud2) has finished
#SBATCH --dependency=afterok:job_id1	Requests that the the job start after job (jobid1) has finished successfully

Slurm script commands

PBS script command	Description
#SBATCH --account=acc_name	To submit a job to a specific accounting group such as RAC/RAS allocation or different role
#SBATCH --tmp=200G	Asks for 200Gb of temporary disk space
#SBATCH --constraint=blue	To ask for a node feature or constraint set by cluster admin. Here we are looking for “blue” nodes.
#SBATCH --partition=partition_name	To ask for the job to run in a specific partition or queue by name, (unlike Moab there can be more than 1 partition per Job)
--prolog=<executable>	Run by srun only, runs the executable before the step
--epilog=<executable>	Run by srun only, runs the executable after the step finishes

SLURM Environment Variables

Environment Variable	Description
SLURM_JOB_NAME	User specified job name
SLURM_JOB_ID	Unique slurm job id
SLURM_NNODES	Number of nodes allocated to the job
SLURM_NTASKS	Number of tasks allocated to the job
SLURM_ARRAY_TASK_ID	Array index for this job
SLURM_ARRAY_TASK_MAX	Total number of array indexes for this job
SLURM_MEM_PER_CPU	Memory allocated per CPU
SLURM_JOB_NODELIST	List of nodes on which resources are allocated to Job
SLURM_JOB_CPUS_PER_NODE	Number of CPUs allocated per Node
SLURM_JOB_PARTITION	List of Partition(s) that the job is in.
SLURM_JOB_ACCOUNT	Account under which this job is run.

Getting information on your Job

Command	What its used for
<code>squeue -u <username></code>	List all current jobs for a user
<code>squeue -u <username> -t PENDING</code>	List all pending jobs for a user
<code>squeue -u <username> -t RUNNING</code>	List all running jobs for a user
<code>Squeue -p <partitionname></code>	List all the jobs in a partition
<code>scontrol show job <jobid></code>	List information on Job
<code>scontrol show jobid -dd <jobid></code>	List detailed information on Job
<code>Squeue -o "%.18i %.30P %.8j %.8u %.2t %.8p %.10M %.6D %R "</code>	Formatted output of squeue: we added priority and made the partition field bigger (30 characters)

Controlling jobs

Command	What its used for
<code>scancel <jobid></code>	Cancel job
<code>scancel -u <username></code>	Cancel all the jobs for a user
<code>scancel -t PENDING -u <username></code>	Cancel all the pending jobs for a user:
<code>Scancel -name JobName</code>	Cancel one or more jobs by name
<code>scontrol hold <jobid></code>	Hold a job, prevent it form starting
<code>scontrol resume <jobid></code>	Release a job hold, allowing the job to try to start
<code>scontrol requeue <jobid></code>	Requeue a running, suspended or finished job into pending state
<code>scontrol requeuehold<jobid></code>	First requeue the job than put a hold on it.
<code>queue -u <username> -ho %A -t R</code>	List running jobs by user
<code>queue --start</code>	Show expected start time of jobs. (This can change)

Getting information on your Job

Command	What its used for
<code>sstat -- format=AveCPU,MaxRSS,MaxV MSize,JobID -j <jobid></code>	List info resource used by your completed job : average cpu time, Max memory, Max virtual memory, JobId
<code>sacct -u <username> -- format=JobID,JobName,AveCPU ,MaxRSS,MaxVMSize,JobID,Elap sed</code>	List resources used by all jobs of a user
<code>sprio</code>	List job priority information

Getting information on you and your group

Command	What its used for
sacctmgr list Users USERS=<username>	List user and their default account (accounting group)
sacctmgr show user <username> withassoc	List user and their default account (accounting group) and shows more extensive information
sshare	Shows usage info for user usage and priority.
sshare -l	Shows even more info for user usage and priority.

Getting information on your Cluster

Command	What its used for
<code>sinfo --states=idle</code>	Show idle node on cluster
<code>sinfo -R</code>	Show down, drained and draining nodes and their reason
<code>sinfo --Node --long</code>	Show detailed node info.
<code>scontrol show reservation</code>	Shows reservations on the cluster
<code>partition-stats</code>	Compute Canada script to show jobs and nodes by partition

```
scontrol create reservation  
user=root starttime=now  
    duration=infinite  
flags=maint  
nodes=<nodeid>
```