CONVERSION FROM SGE TO SLURM

A. CONVERSION TABLE

COMMAND	SGE	SLURM	COMMENTS
BASIC COMMAND			
Interactive login	qlogin	interactive	
Cluster status	qhost	sinfo	
Job submission	qsub <job_file>	sbatch <job_file>	
Job deletion	qdel <job_ID>	scancel <job_ID>	
Job status	qstat or show	squeue	
Job status by job	qstat -j <job_ID>	squeue -j <job_ID>	
Job status by user	qstat -u <username>	squeue -u <username>	
Job hold	qhold <job_ID>	scontrol hold <job_ID>	
Queue list	qconf -sql	scontrol show partition	
Node list	qhost	scontrol show nodes	
Monitor all jobs	qacct	sacct	
Monitor a job	qacct -j <job_ID>	sacct -j <job_ID>	
Cluster status	qhost -q	sinfo	
ENVIRONMENT VARIABLES			
Job name	$JOBNAME	$SLURM_JOB_NAME	
Job ID	$JOBID	$SLURM_JOB_ID	
Submission directory	$SGE_O_WORKDIR	$SLURM_SUBMIT_DIR	
Submission host	$SGE_O_HOST	$SLURM_SUBMIT_HOST	
Node list	$PE_HOSTFILE	$SLURM_JOB_NODELIST	
Job user	$USER	$SLURM_JOB_USER	
Job array index	$SGE_TASK_ID	$SLURM_ARRAY_TASK_ID	
First array task	$SGE_TASK_FIRST	$SLURM_ARRAY_TASK_MIN	
Last array task	$SGE_TASK_LAST	$SLURM_ARRAY_TASK_MAX	
Queue name	$QUEUE	$SLURM_JOB_PARTITION	
Number of allocated processors	$NSLOTS	$SLURM_NTASKS	

JOB SUBMISSION			
Script directive	#$	#SBATCH	
Assign environment settings	-V <variable[=value],...>	Default in SLURM, no need to specify.	
Set working directory	-wd <directory_path>	--chdir=<directory_path> or -D <directory_path>	
Assign job name	-N <job_name>	--job-name=<job_name> or -J <job_name>	
Output file	-o <output_file>	-o <output_file> or --output=<output_file>	
Error file	-e <output_file>	-e <error_file> or --error=<error_file>	
Queue	-q <queue name>	--partition=<queue> or -p <queue>	
Job/Task allocation	-pe <parallel environment>	--nodes=<nodes> or -N<nodes> --ntasks=<cores> or -n<cores> --ntask-per-core=<tasks_per_core> --ntask-per-node=<tasks_per_node> -- ntask-per-socket=<tasks_per_socket> > --cpus-per-task=<cpu_per_task>	
Memory allocation	-l h_vmem=<float>G (1)	--mem=<float>G (2) --mem-per-cpu=<float>G (3)	1) -l h_vmem assigns memory per core 2) --mem assigns memory per node 3) –mem-per-cpu assigns memory per core
Time limit	-l h_rt=<HH:MM:SS>	--time=<HH:MM:SS> or -t <HH:MM:SS>	
Project name	-P <project name>	--account=<project_name> or -A <project_name>	In our current cluster, all jobs are run under “default” as

			<project_name> by default.
Array range and increment	-t <start_num>-<end_num>:<increment>	--array=<start_num>-<end_num>:<increment> or -a <start_num>-<end_num>:<increment>	
Notification emails		--mail-user=<username@ttu.edu> (1) --mail-type=<BEGIN | END | FAIL | REQUE | ALL> (2)	1) Job notifications will be emailed to the provided email address <username@ttu.edu>. 2) Job notifications will be emailed at a certain event specified in the command options: BEGIN: when the job starts to execute. END: when the job completes. FAIL: if and when the job fails. REQUEUE: if and when the job is requeued. ALL: for all of the above cases.
Job dependency	-hold_jid <job_ID | job_name>	--dependency=after:job_id[:job_id...] (1) --dependency=afterany:job_id[:job_id...] (2) --dependency=afternotok:job_id[:job_id...] (3) --dependency=afterok:job_id[:job_id...] (4) --dependency=aftercorr:job_id (5) --dependency=singleton (6)	1) Job begins after the specified jobs have begun execution. 2) Job begins after the specified jobs have terminated. 3) Job begins after the specified jobs have terminated in some failed state. 4) Job begins after the specified jobs have

			successfully executed. 5) A task of an job array can begin after the corresponding task ID in the specified job has completed successfully. 6) Job begins after any previously launched jobs sharing the same job name and user have terminated
Begin Time	-a <YYMMDDhhmm>	--begin=<YYYY-MM-DD[THH:MM:[SS]]>	
Generic Resources Scheduling		--gpus=<count> (1) --gpus-per-node=<count> (2)	1) The total number of GPUs to be allocated generic resources 2) The number of GPUs per node to be allocated generic resources

B. JOB SUBMISSION LAYOUT CONVERSION FORMAT FROM CURRENT SGE JOBS TO SLURM JOBS

PARAMETERS IN SGE (1)	PARAMETERS IN SLURM (2)	COMMENTS
#$	#SBATCH	1) Script directive for SGE 2) Script directive for SLURM
-V	Default in SLURM, no need to specify.	1) Instructs the scheduler to use the current environment settings in the batch job
-cwd	--chdir=./ or -D ./	Instructs the scheduler to use the current directory where the job is submitted as the job's working directory.
-S /bin/bash	#!/bin/bash	1) Instructs the scheduler to use the /bin/bash as the shell for the batch session.

		2) This command should be provided as the first line of the scipt
-N <job_name>	--job-name=<job name> or -J <job name>	Sets the name for the job
-o $JOB_NAME.o$JOB_ID	--output=%x.o%j or -o %x.o%j	Indicates the name of the standard output file in the format of <your_job_name>.o<your_job_ID>
-e $JOB_NAME.e$JOB_ID	--error=%x.e%j or -e %x.e%j	Indicates the name of the standard error file in the format of <your_job_name>.e<your_job_ID>
-q <queue name>	--partition=<queue> or -p <queue>	Instructs the scheduler to use the queue defined by <queue>.
-pe <parallel environment>	--nodes=<nodes> or -N <nodes> --ntasks=<cores> or --n <cores>	1) Instructs the scheduler to use the parallel environment defined by <parallel environment> 2) Specifies the number of cores and nodes to use for job submission
-l h_vmem = <float>G (1)	--mem-per-cpu=<float>G (2)	Instructs the scheduler to reserve <float> gigabytes of virtual memory per **core**
-l h_rt = HH:MM:SS	--time=<HH:MM:SS> or -t <HH:MM:SS>	Instructs the scheduler to set the maximum job time to HH:MM:SS
-P <project_name>	--account=<project_name> or -A <project_name>	Instructs the scheduler to use the project/account defined by <project_name>. In our current cluster, all jobs are run under “default” as <project_name> by default.
-t <start_num>-<end_num>:<increment>	--array=<start_num>-<end_num>:<increment> or -a <start_num>-<end_num>:<increment>	Specifies the first task number, the last task number, and step increment in an array job
	--gpus=<count>	Specify the number of gpus used for generic resources

C. EXAMPLE JOB SCRIPTS

C1) Standard job

Quanah

```
#!/bin/bash
#SBATCH -J Job_Test
#SBATCH -o %x.o%j
#SBATCH -e %x.e%j
#SBATCH -p quanah
#SBATCH -N 1
#SBATCH --ntaks-per-node=36
#SBATCH --mem-per-cpu=5370MB
#SBATCH -t 48:00:00
#SBATCH -A default


for i in {1..100}; do
echo $i >> Numbers.txt
done
```

Nocona

```
#!/bin/bash
#SBATCH --job-name=Job_Test
#SBATCH --output=%x.o%j
#SBATCH --error=%x.e%j
#SBATCH --partition=nocona
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=128
#SBATCH --mem-per-cpu=4027MB
#SBATCH --time=48:00:00
#SBATCH --account=default

for i in {1..100}; do
echo $i >> Numbers.txt
done
```

Matador

```
#!/bin/bash
#SBATCH -J Job_Test
#SBATCH -o %x.o%j
#SBATCH -e %x.e%j
#SBATCH -p matador
#SBATCH -N 1
#SBATCH --ntasks-per-node=40
#SBATCH --mem-per-cpu=9639MB
#SBATCH -t 48:00:00
#SBATCH -A default
#SBATCH --gpus-per-node=2

for i in {1..100}; do
echo $i >> Numbers.txt
done
```

Toreador

```
#!/bin/bash
#SBATCH --job-name=Job_Test
#SBATCH --output=%x.o%j
#SBATCH --error=%x.e%j
#SBATCH --partition=toreador
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=16
#SBATCH --mem-per-cpu=12064MB
#SBATCH --time=48:00:00
#SBATCH --account=default
#SBATCH --gpus-per-node=3

for i in {1..100}; do
echo $i >> Numbers.txt
done
```

C2) Array job

Quanah

```
#!/bin/bash
#SBATCH --job-name=ArrayJob
#SBATCH --partition nocona
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=36
#SBATCH --time=48:00:00
#SBATCH --mem-per-cpu=5370MB
#SBATCH --array=1-37:6

echo "$SLURM_ARRAY_TASK_ID"
```

Nocona

```
#!/bin/bash
#SBATCH --job-name=ArrayJob
#SBATCH --partition nocona
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=128
#SBATCH --time=48:00:00
#SBATCH --mem-per-cpu=4027MB
#SBATCH --array=1-37:6

echo "$SLURM_ARRAY_TASK_ID"
```

Matador	Toreador
#!/bin/bash #SBATCH -J ArrayJob #SBATCH -p matador #SBATCH -N 1 #SBATCH --ntasks-per-node=40 #SBATCH -t 48:00:00 #SBATCH --mem-per-cpu=9639MB #SBATCH -a 1-37:6 #SBATCH --gpus-per-node=2 echo "$SLURM_ARRAY_TASK_ID"	#!/bin/bash #SBATCH --job-name=ArrayJob #SBATCH --partition toreador #SBATCH --nodes=1 #SBATCH --ntasks-per-node=16 #SBATCH --time=48:00:00 #SBATCH --mem-per-cpu=12064MB #SBATCH --array=1-37:6 #SBATCH --gpus-per-node=3 echo "$SLURM_ARRAY_TASK_ID"

C3) MPI job

Quanah	Nocona
#!/bin/bash #SBATCH --job-name=MPI_Test_Job #SBATCH --output=%x.o%j #SBATCH --error=%x.e%j #SBATCH --partition quanah #SBATCH --nodes=1 #SBATCH --ntasks-per-node=36 #SBATCH --time=48:00:00 #SBATCH --mem-per-cpu=5370MB module load intel impi mpirun --machinefile machinefile.%j -np $SLURM_NTASKS ./mpi_hello_world	#!/bin/bash #SBATCH --job-name=MPI_Test_Job #SBATCH --output=%x.o%j #SBATCH --error=%x.e%j #SBATCH --partition nocona #SBATCH --nodes=1 #SBATCH --ntasks-per-node=128 #SBATCH --time=48:00:00 #SBATCH --mem-per-cpu=4027MB module load intel impi mpirun --machinefile machinefile.%j -np $SLURM_NTASKS ./mpi_hello_world
Matador	**Toreador**
#!/bin/bash #SBATCH -J MPI_Test_Job #SBATCH -o %x.o%j #SBATCH -e %x.e%j #SBATCH -p matador #SBATCH -N 1 #SBATCH --ntasks-per-node=40 #SBATCH -t 48:00:00 #SBATCH --mem-per-cpu=9639MB #SBATCH -gpus-per-node=2 module load intel impi mpirun --machinefile machinefile.%j -np $SLURM_NTASKS ./mpi_hello_world	#!/bin/bash #SBATCH -J MPI_Test_Job #SBATCH -o %x.o%j #SBATCH -e %x.e%j #SBATCH -p matador #SBATCH -N 1 #SBATCH --ntasks-per-node=16 #SBATCH -t 48:00:00 #SBATCH --mem-per-cpu=12064MB #SBATCH --gpus-per-node=3 module load intel impi mpirun --machinefile machinefile.%j -np $SLURM_NTASKS ./mpi_hello_world