PBS SLURM Rosetta Stone

Between 2020 and 2021, all HPC clusters were transitioned from using the PBS job scheduler to SLURM. In general, SLURM can translate and execute scripts written for PBS. This means that if you submit a PBS script written for Ocelote or ElGato on Puma (with the necessary resource request modifications), your script will likely run. However, there are a few caveats that should be noted:

- You will need to submit your job with the new SLURM command, e.g. `sbatch` instead of `qsub`
- There may be some PBS directives that do not directly translate to SLURM which cannot be interpreted
- The environment variables specific to PBS and SLURM are different. If your job relies on these, you will need to update them. Common examples are `PBS_O_WORKDIR` and `PBS_ARRAY_INDEX`

Refer to the following list of common PBS commands, directives, and environment variables and their SLURM counterparts. For a PDF version, click here.

PBS	SLURM	Purpose
Job Management		
`qsub <options>`	`sbatch <options>`	Batch submission of jobs to run without user input
`qsub -I <options>`	`salloc <options>`	Request an interactive job
`N/A`	`srun <options>`	Submit a job for realtime execution. Can also be used to submit an interactive session
`qstat`	`squeue`	Show all jobs
`qstat <jobid>`	`squeue --job <jobid>`	Check status of a specific job
`qstat -u <netid>`	`squeue -u <netid>`	Check status of jobs specific to user
`tracejob <jobid>`	`sacct -j <jobid>`	Check history of a completed job
`qdel <jobid>`	`scancel <jobid>`	Delete a specific job
`qdel -u <netid>`	`scancel -u <netid>`	Delete all user jobs
`qstat -Q`	`sinfo`	View information about nodes and queues
`qhold <jobid>`	`scontrol hold <jobid>`	Places a hold on a job to prevent it from being executed
`qrls <jobid>`	`scontrol release <jobid>`	Releases a hold placed on a job allowing it to be executed
Job Directives		
`#PBS -W group_list=group_name`	`#SBATCH --account=group_name`	Specify group name where hours are charged
`#PBS -q standard`	`#SBATCH --partition=standard`	Set job queue
`#PBS -l walltime=HH:MM:SS`	`#SBATCH --time HH:MM:SS`	Set job walltime
`#PBS -l select=<N>`	`#SBATCH --nodes=<N>`	Select N nodes
`#PBS -l ncpus=<N>`	`#SBATCH --ntasks=<N>` `#SBATCH --cpus-per-task=<M>`	PBS: Select N cpus SLURM: Each task is assume to require one cpu. Optionally, you may include cpus-per-task if more are required. Requests NxM cpus Note: Puma has 94 cpus available on each node
`#PBS -l mem=<N>gb`	`#SBATCH --mem=<N>gb`	Select N gb of memory per node
`#PBS -l pcmem=<N>gb`	`#SBATCH --mem-per-cpu=<N>gb`	Select N gb of memory per cpu Note: Puma defaults to 5GB per cpu
`#PBS J N-M`	`#SBATCH --array=N-M`	Array job submissions where N and M are integers
`#PBS -l np100s=1`	`#SBATCH --gres=gpu:1`	Optional: Request a GPU
`#PBS -N JobName`	`#SBATCH --job-name=JobName`	Optional: Set job name
`#PBS -j oe`	`(default)`	Optional: Combine stdout and error

(default)	#SBATCH -e <job_name>-%j.err #SBATCH -o <job_name>-%j.out	Optional: Separate stdout and stderr (SLURM: %j is a stand-in for $SLURM_JOB_ID)
#PBS -o filename	#SBATCH -o filename	Optional: Standard output filename
#PBS -e filename	#SBATCH -e filename	Optional: Error filename
N/A	#SBATCH --open-mode=append	Optional: Combine all output into single file. Note: If this is selected, each job run will append to that filename, including preexisting files with that name
#PBS -v var=<value>	#SBATCH --export=var	Optional: Export single environment variable var to job
#PBS -V	#SBATCH --export=all (default)	Optional: Export all environment variables to job
(default)	#SBATCH --export=none	Optional: Do not export working environment to job
#PBS -m be	#SBATCH --mail-type=BEGIN|END|FAIL|ALL	Optional: Request email notifications Beware of mail bombing yourself
#PBS -M <netid>@email.arizona.edu	#SBATCH --mail-user=<netid>@email.arizona.edu	Optional: email address used for notifications
#PBS -l place=excl	#SBATCH --exclusive	Optional: Request exclusive access to node
Environment Variables		
$PBS_O_WORKDIR	$SLURM_SUBMIT_DIR	Job submission directory
$PBS_JOBID	$SLURM_JOB_ID	Job ID
$PBS_JOBNAME	$SLURM_JOB_NAME	Job name
$PBS_ARRAY_INDEX	$SLURM_ARRAY_TASK_ID	Index to differentiate tasks in an array
$PBS_O_HOST	$SLURM_SUBMIT_HOST	Hostname where job was submitted
$PBS_NODEFILE	$SLURM_JOB_NODELIST	List of nodes allocated to current job
Terminology		
Queue	Partition	
Group List	Association	
PI	Account	