Can I use my LSF scripts?

SLURM Workload Manager is a queue management system which replaces the commercial LSF scheduler as the job manager on CSUC.

You can use the ***rosetta stone_ext-link*** from SchedMD or take a look below, there is a quick reference table comparing commands between them:

Commands:

LSF	SLURM	Description
bsub < example.lsf	sbatch example.slm	Submits a job to he queue system.
	sbatch --test-only example.slm	Test and find out when your job is estimated to run use (this does not submit the job).
bkill <*job_id*>	scancel <*job_id*>	Kills the job with the specified ID.
bjobs	squeue	List user's active jobs
bqueues	sinfo	Show the partitions (queues) information and the nodes status associated to them
bacct	sacct	Display accounting data
interactiu	srun --pty /bin/bash	Obtain a job allocation and execute and application (interactive jobs).

Job Environmental Variables:

LSF	SLURM	Description
$LSB_JOBID	$SLURM_JOB_ID	Job ID
$LSB_SUBCWD	$SLURM_SUBMIT_DIR	Submission directory
$LSB_SUB_HOST	$SLURM_SUBMIT_HOST	Submission host
$LSB_HOSTS	$SLURM_JOB_NODELIST	Allocated calculation nodes
$LSB_DJOB_NUMPROC	$SLURM_NTASKS	Number of processors allocated
	$SLURM_JOB_PARTITION	Queue

Job submitting parameters:

LSF	SLURM	Description
#BSUB	#SBATCH	Scheduler directive.

-J <*job_name*>	-J <*job_name*> --job-name=<*job_name*>	Name of the job that will appear when querying jobs.
-o <*output_file*>	-o <*output_file*> --output=<*output_file*>	Defines the name of the file where stdout is redirected.
-e <*error_file*>	-e <*error_file*> --error=<*error_file*>	Defines the name of the file where stderr is redirected.
-q <*queue_name*>	-p <*queue_name*> --partition=<*queue_name*>	Submits the job to the specified queue (partition).
-u <*email*>	--mail-user=*email*	When job finishes, send a mail notification.
-M 100	--mem=<*size[units]*>	Total memory required. Units can be: *K*, *B* or *G*.
	--mem-per-cpu=<*size[units]*>	Memory requirement per core. Units can be: *K*, *B* or *G*.
-n 4	-n <*num*> --ntasks <*num*>	Number of tasks (processors). It is generally used to define the number of MPI tasks.
	-c <*num*> --cpus-per-task=<*num*>	Asks a number of processors per task. Generally used to define the number of threads OMP.
-R "span[ptile=2]"	--tasks-per-node=<*num*>	Processes per node.
-gpu num=1	--gres=gpu:pascal:<*num_gpus*>	Allocates the indicate number of GPUs (1 or 2).