

From Pirineus to Pirineus II

16-07-18

Changes in queues management

SLURM Workload Manager is a queue management system which replaces the commercial LSF scheduler as the job manager on CSUC. Below there is a quick reference table comparing commands between them:

Commands:

LSF	SLURM	Description
<code>bsub < example.lsf</code>	<code>sbatch example.slm</code>	Submit a job to the queue system
	<code>sbatch --test-only example.slm</code>	Test and find out when your job is estimated to run use (this does not submit the job)
<code>bkill 1843</code>	<code>scancel 1843</code>	Kills the job with the specified ID
<code>bjobs</code>	<code>squeue</code>	List user's active jobs
<code>bqueues</code>	<code>sinfo</code>	Show the partitions (queues) information and the nodes status associated to them
<code>bacct</code>	<code>sacct</code>	Display accounting data
<code>interactiu</code>	<code>srn --pty <application></code>	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_SUBMIR_HOST	Submission host
\$LSB_HOSTS	\$SLURM_JOB_NODELIST	Allocated calculation nodes
\$LSB_DJOB_NUMPROC	\$SLURM_NTASKS	Number of processors allocated
	\$SLURM_JOB_PARTITION	Queue

More documentation can be found at [SLURM website](#):

Job submissions

In the CSUC Knowledge Base, there is a job script generator that makes easier the transition from LSB to SLURM. It will generate the necessary file to submit your job. You can find it at <https://confluence.csuc.cat/display/HPCKB/Job+Script+Generator>.

It has the following structure:

The screenshot shows the Job Script Generator interface. On the left, there are configuration options for job submission, including job name, input file name, email notifications, machine selection, queue, and resource allocation. A red box labeled '2' highlights the 'Generate' button. On the right, the 'Generated Script' section displays a pre-generated SLURM script. A red box labeled '3' highlights the 'Enable text editor' button. Below the script, there are three buttons: 'Download' (labeled '4a'), 'Copy' (labeled '4b'), and 'Reset'.

- 1- Fill the formulary with the specifications for your job.
- 2- Click generate button.
- 3- Modify the script, if needed, enabling the text edition before download / copy.
- 4-
 - a. Download the **.slm* file and transfer it to your working path.
 - b. Copy the script text and paste it to your open file in the terminal

As a summary, the most common parameters for job submitting, and their equivalences from LSB to SLURM are:

LSF	SLURM	Description
#BSUB	#SBATCH	Scheduler directive
-J <job_name>	-J <job_name>	Name of the job that will appear when querying jobs
-o <output_file_name>	-o <output_file_name>	Name of the output file (redirection from STDOUT)
-e <error_file_name>	-e <error_file_name>	Name of the error file (redirection from STDERR)
-q <queue_name>	-p <queue_name>	Submits the job to the specified queue
-u <email>	--mail-user=<email>	When job finish, send a mail notification
-M 100	--mem=800M --mem-per-cpu=3G	Total memory required in MB Memory requirement per processor in GB
-n 4	-n 4	Number of tasks (processors)
-R "span[ptile=2]"	--tasks-per-node=2	Processes per node
-gpu num=1	--gres=gpu:1	Allocate 1 GPU per allocated node

Modules changes

Modules have suffered some structure changes and are defined as follows:

- All library modules are under *libs/*
 - E.g.: fftw, lapack, mkl, etc.
- All compilers and scripting languages are under *tools/*
 - E.g.: python, gcc or intel compilers, etc.
- All the HPC Applications compiled by the CSUC team are under *apps/*
 - E.g.: Gaussian, Gromacs, ADF, etc.
- Now it's possible to load multiple modules at once using the predefined *toolchains/*
 - E.g.: combination of Intel compiler + MKL + Open MPI

All the available modules can be listed executing the command *module av*. Examples:

Old version	New version	Description
module load gaussian/g16b1	module load app/gaussian/g16b01	Load the Gaussian v.16b01 module
module load intel/comp_xe_2015	module load tools/intel/comp_xe_2018	Load the Intel® compiler
module load intel/mkl_11.2	module load libs/mkl/2018	Load the Intel® Math Kernel Library module
module load intel/comp_xe_2015 intel/mkl_11.2 openmpi/3.0.1-intel	module load toolchains/intel_mkl_ompi	Load the Intel® compiler, the Intel® Math Kernel Library and the Intel® compiled Open MPI library