LSF Primer

Job Scheduling

On an HPC cluster, the scheduler manages which jobs run where and when. On our clusters, you control your jobs using a job scheduling system called IBM Spectrum Scale LSF that allocates and manages compute resources for you. By accurately requesting the resources you need, you can have your jobs execute as quickly as possible on available nodes which can process them.

Job resource control enforcement in LSF with cgroups

LSF makes use of Linux control groups (cgroups) to limit the CPU cores, number of GPUs and memory that a job can use. The goal is to isolate the jobs from each other and prevent them from consuming all the resources on a machine. All LSF job processes are controlled by the Linux cgroup system. Jobs can only access the GPUs which have been assigned to them. If the job processes on a host use more memory than requested, the job will be terminated by the Linux cgroup memory sub-system.

LSF Job Submission

The bsub command submits jobs to LSF. It consists of two parts: resource requests and job steps. A resource request consists of the number of CPU cores, amount of memory per CPU core, wall time (expected duration), GPUs etc. You can find complete lists of the available resources on the cluster pages lilac and juno. Job steps specify tasks which must be done, software which must be run, etc. Upon submission, the job will advance in the queue until the requested resources are available. At this point LSF will allocate the requested CPU cores, GPUs, and memory, and execute it.

You can submit your jobs in one of two ways. For testing and small jobs you may want to run a job interactively. This way you can directly interact with the compute node(s) in real time. However, the preferred way for multiple jobs or long-running jobs, is to put all of your resource requests and executable into the command line or combining them into a bsub script and submitting that to the job scheduler.

Interactive Job Examples

Interactive batch jobs provide interactive access to compute resources, they can be helpful for debugging.

Here is an example command to create an interactive bash shell on a compute node with a wall time of 2 hours:

```
bsub -W 2:00 -Is
/bin/bash
```

Here is an example command for an interactive job with X11 forwarding with a wall time of 30 minutes:

```
bsub -W 0:30 -Is -XF
/bin/bash
```

Command Line Submission Example

Users can submit jobs directly on the command line.This submission requests 2 CPUs, 8GB of memory and 10 hours wall time. Note that the memory requirement (-R rusage[mem=4]) is in GB (gigabytes) and is PER CORE (-n) rather than per job. A total of 8GB of memory will be allocated for this example job.

```
bsub -n2 -R rusage[mem=4] -W 4:00
myjob.sh
```

Job <17536228> is submitted to default queue <cpuqueue>.

When you submit a job, LSF will return a jobID for it. The bjobs command will show you all of your running and pending jobs. You can get information about a specific running or pending job using the jobID.

```
bjobs -l <jobID>
```

This can give you useful information about the resources the job used or maybe why it failed. You can also use bhist to get information about jobs which ran in the past.

```
bhist -l <jobID>
```

Script Submission Example

Typically a user submits a job submission script which contains resource requests and job steps to LSF. A submission script is usually a shell (bash) script.There are default values for all batch parameters for each of the clusters, but it is a good idea to always specify the number of threads, GPUs (if needed), memory per thread, and expected wall time for batch jobs. To minimize time spent waiting in the queue, specify the smallest wall time that will safely allow your jobs to complete.

Here is an executable lsf batch script called myRjob.lsf that requests 1 CPU, 4G RAM and 10 hours wall time. It uses modules to load R and executes the program.

```
#!/bin/bash
#BSUB -J myRjobname
#BSUB -W 10:00
#BSUB -n 1
#BSUB -R rusage[mem=4]
#BSUB -e <some directory>/%J.err
#BSUB -o <some directory>/%J.out
module load R
cd ~
# execute program
R CMD BATCH runme.R
```

Submit the batch script with the bsub command:

```
bsub < myRjob.lsf
```

More information

For more information on the commands to submit and manage jobs, please see the following pages:

LSF bsub flags

LSF Commands

Juno Cluster Guide

Lilac Cluster Guide