Quick Reference Guide
Converting SGE Clusters to Platform LSF

Contents:

1 SGE to LSF Guide

This document is to aid both cluster administrators and users in migrating from Sun Grid Engine (SGE) clusters to Platform LSF. It shows what the equivalent SGE feature/functionality is in Platform LSF on topics such as Batch Job Submission, Job Monitoring, Job Control, Job States, Environment Variables, Limits, Cluster Information, Cluster Configuration and Queues. It also highlights functionality that is not present in SGE as well as differences.

1.1 Job Submission

Job Submission Differences

SGE's job submission command is qsub which is roughly equivalent to LSF's bsub. SGE has limited interactive batch job submission and requires qrsh. LSF has multiple options for interactive batch job submission including pseudo-terminal and shell-mode support built-into the product.

Both support submission scripting and can interpret special comments in scripts, so it's not difficult to convert submission scripts;

```
#SGE Directives
#$ -e error
$ -h u,
process.sh data

#!/bin/sh
# LSF Directives
#BSUB –q priority
#BSUB –R "type==any"
#BSUB –M 1024
```

SGE's qsub command line is treated as the name of the script, whereas LSF bsub considers the command line as contents of a script.

In LSF, bsub < myscript

Switch conversions are listed below, as well switches that do not readily exist in SGE

SGE	LSF	Notes
qsub	bsub	LSF has more flexibility with job submission and more switches to give the user additional control over the job
-l h_core=bytes[,..] -l s_core=bytes[,..]	-C core_limit	Set core file size limit, in lsf.conf set LSF_UNIT_FOR_LIMITS so this can be in units of KB, MB, GB, TB, PB or EB
-l h_data=bytes[,..] -l s_data=bytes[,..]	-D data_limit	Set data segment size limit, in lsf.conf set LSF_UNIT_FOR_LIMITS so this can be in units of KB, MB, GB, TB, PB or EB
-l h_stack=bytes[,..] -l s_stack=bytes[,..]	-S stack_limit	Set stack size limit, in lsf.conf set LSF_UNIT_FOR_LIMITS so this can be in units of KB, MB, GB, TB, PB or EB

-l h_vmem=bytes[,..] -l s_vmem=bytes[,..]	-M mem_limit	Set memory size limit, in lsf.conf set `LSF_UNIT_FOR_LIMITS` so this can be in units of KB, MB, GB, TB, PB or EB
-l h_fsize=blocks[,..]	-F file_limit	Set file size limit, in lsf.conf set `LSF_UNIT_FOR_LIMITS` so this can be in units of KB, MB, GB, TB, PB or EB
-m [**b**|e|a|s|n]	-B	Send email at the beginning of job
-m [b|**e**|a|s|n]	-N	Send email at the end of job
-M name@host	-u email_address	Use this email address for job notifications
-q “queue1,queue2	-q “queue1 ..”	Specify batch queue for job
-P project_name	-P project_name	Specify batch project for job
-e [[hostname]:path	-e[o] error_file	Defines stderr output for the job. SGE always overwrites, LSF can control append or overwrite the stderr file using the o switch
-o [[hostname]:path	-o[o] output_file	Defines stdout output for the job. SGE always overwrites, LSF can control append or overwrite the stderr file using the o switch
-r y|n	-r	Specify if the batch job is rerunnable or not
-h	-H	Holds the job in a pending state
-t n[-m]:s -N name	-J “job_name[index_list]%job_slot_limit]	Specifies Job Array Name, Size of Job Array and Job Slot Limit. LSF supports up to 2 billion job array elements.
-hold-jid-ad -hold-jid	-w “expression”	Specify job or job array dependencies
-help	-h, -V	Help on the command and prints version
-p priority	-sp priority	Specify the job’s priority, Valid values for LSF job priority are any integers between 1 and MAX_USER_PRIORITY (configured in `lsb.params`, displayed by `bparams -l`).
-c ckpt_selector	-k “chkpntdir[chkpnt_period] [method=method_name]	Specify checkpointing, checkpoint period and checkpoint mechanism for job.

-a [[CC]YY]MMDDhhmm,[ss]	-b begin_time	Specify start time of batch job
-dl[[CC]YY]MMDDhhmm,[ss	-t term_time	Specify termination time of batch job
-l h_cpu=sec[,..] -l s_cpu=sec[,..]	-c cpu_time[/host_name]/host_model]	Specify cpu time limit for job. LSF supports normalization against a hostname or hostmodel.
-l h_rt=sec[,..] -l s_rt=sec[,..]	-W cpu_time[/host_name]/host_model]	Specify wall clock time limit for job. LSF supports normalization against a hostname or hostmodel, as well as absolute wall clock limit via lsb.params ABS_RUNLIMIT=Y
-l …	-R “res_req” –R “res_req”	Specify resource requirements such as memory or license for the job. LSF supports compound resource requirements as well. SGE needs to specify limits and attributes for the job
Qrsub, qrstat, qrdel	-U reservation_ID	Specify Advanced Reservation for job.
-s [/bin/sh]/bin/csh..] sge_conf qconf –sconf/mconf	-L login_shell	Specify a login shell for the job. SGE starts (and sources) all shells specified in ‘login_shells’
-n num_proc=#[,…]	-n min_proc[,max_proc]	Specify the minimum (and maximum) number of processors for the job
	-i input file| -is input_file -Zs	Specify input file for job, enable spooling. SGE accepts input file by default and always spools (no flexibility)
-notify	-wt ‘[hour,]’minute’	Specify the amount of time before a job control action occurs that a job warning action is to be taken. In SGE, you must define the grace period in seconds in queue configuration

LSF bsub functionality that has no exact match in SGE

	-p process_limit	Specify process limit for job. You have to use ulimit –p in your SGE job script
qrsh	-I | -Ip | Is -K	Interactive batch job with pseudo terminal and shell mode support and blocking bsub (submit a batch job and wait for its completion to return control to the shell). SGE has to use qrsh which is a poor substitute and has limitations
qsub –x exclusive1	-x	Exclusive job scheduling is built into bsub, but SGE forces you to define an exclusive queue that consumes all slots on a class of compute nodes.
	-E “pre_exec script”	Specify pre-execution script for job. In SGE, you must define this in queue and host properties
	-app application_profile	Specify application profile for job. SGE has no concept of an application profile

	-f "local file op [remote_file]	Specify file(s) to be transferred to/from execution host. You need to manually script this in SGE
	-jsdl filename | - jsdl_strict file_name	Job Submission Description Language/XML submission support. "Job Submission Description Language (JSDL) Specification" at http://www.gridforum.org/documents/GFD.56.pdf You have to use DRMAA in SGE
-l hostname=host[,…[	-m "hostname[@clusterName] + pref_level | host_group[[!]]+[pref)level]].."	Specify hostname(s)/cluster and host preference level of eligible batch hosts as well as host groups and mandatory first execution host. SGE allows for basic host selection from the job submission. More granular configuration must be defined in the queue definition

1.2 Job States

In SGE, you need to run qstat – explain [[code]]. LSF gives information on the batch job state and details using bjobs –l

SGE	LSF	Notes
R,t,running,R,T, running,re-ruin	RUN	Job is in a running state
qacct –j	DONE	Job is done successfully, LSF MBD holds finished job data in memory up to lsf.conf CLEAN_PERIOD (default 1hr). Other you needs to use bacct/bhist to access lsb.acct/lsb.events
qacct –j	EXIT	Job has exited, LSF MBD holds finished job data in memory up to lsf.conf CLEAN_PERIOD (default 1hr). Other you needs to use bacct/bhist to access lsb.acct/lsb.events. bacct can track the detailed reason why the job was terminated
Qw,pending, hqw, pending user and system hold, hRwq, pending user or system hold, requeue	PEND	Job is in a pending state
s, ts	USUSP	Job is suspended by user
S,tS T,tT	SSUSP	Job is suspended by the system
	PSUSP	Job is suspend by the user while pending
Rs,Rts, RSD, RtS, RT, RtT		Job is suspended with re-run

1.3 Job Monitoring

SGE	LSF	Notes
qacct	bacct/bhist	LSF can query the lsb.acct for finished job data stats as well as bhist for a job's history. SGE's qacct is limited, and one needs to use the ARCO for historical accounting
qstat	bjobs	Displays information about pending, running and suspended jobs
	bjgroup	LSF can display information about job groups. SGE has no direct equivalent.

	bpeek	LSF can view the current output of the batch job. SGE needs a shared directory to provide stdout/stderr as by default it puts stdout/stderr in files names $REQUEST.[e\|o].$JOB_ID[.$TASK_ID] in the working directory of the job

1.4 Job Controls

In SGE, job control commands qdel, qhold, qalter, qrls, qmod for which LSF equivalents are the following

SGE	LSF	Notes
qalter, qmod	bmod	Modify a running or pending job's options
qalter –p priority jobid	bbot	In LSF bbot places the job's priority to the bottom of list of pending jobs. In SGE, sets the job priority to the bottom of the queue
qalter –p priority jobid	btop	In LSF, btop places the job's priority to the top of the list of pending jobs. In SGE, set's the priority to 1024 to force the job to top
qalter –q queue_name jobid[,job_id..]	bswitch	Changes the queue of the pending job
qhold job_id, qmod –s job_id	bstop	Suspends the job
qrls job_id, qmode –us jobid	bresume	Resumes the suspended job
qmod –sj job_id qalter –c –s jobid [qdel jobid] or [qmod –usj jobid]	bchkpnt	Checkpoints the batch job (assuming checkpoint/restart configuration was enabled)
qhold jobid, qresub jobid, qdel jobid	brequeue	In LSF, cleanly requeues existing job id, even a job that has finished. In SGE, you cannot requeue a completed job nor can you reuse the same job id

2 Cluster Configuration & Cluster Information

2.1 Batch Cluster Configuration

In SGE, cluster configuration is modified via SGE utilities. In LSF, you have the flexibility to modify all of the configuration files.

SGE	LSF	Notes
qconf [-sconf]| -mconf]	install.config	LSF has a configurable script to automated installation.
qconf	lsf.shared	In LSF, the resources are defined in lsf.shared.
	lsf.cluster.[clusterName]	In LSF, the hosts are defined in lsf.cluster among other cluster definitions. In SGE, set's the priority to 1024 to force the job to top
qconf –sql, qconf –aq, qconf –mq queuename	lsb.queues	In LSF, the batch queue definitions are configured in lsb.queues. Queues can be added, modified and/or deleted. LSF queues are cluster wide, whereas, historically, in SGE queues were attached to one of more compute nodes. Most recent versions of SGE now support clusterwide queues.
qconf	lsb.params	In LSF, the batch parameters are defined in lsb.params
qconf	lsb.users	In LSF, the batch user definitions are configured in lsb.users
qconf	lsb.resources	LSF supports the concept of generalized limits for resource allocation and usage limits. SGE's host and queue configurations are changed via command line.
	lsb.applications	**LSF supports application encapsulation, where one can define LSF policy against an application to help simplify queue configuration. SGE has no such functionality.**