LSF @ SLAC

Using the SLAC LSF Batch Cluster

Neal Adams
SLAC National Accelerator Laboratory
neal@slac.stanford.edu

What is LSF?

- Load Sharing Facility (LSF) product by Platform Computing Corporation.
- Allows queuing and scheduling of batch jobs.
- Provides scheduling of jobs based on load conditions and resource requirements specified by the user.

What is a batch job?

- "A unit of work run in the LSF system."
- A batch job can be a script, command or program.

  Example: bsub *hostname*

Why batch over interactive?

- Running jobs in the LSF batch system does not tie up shared interactive resources.
- No contention with other user's jobs.
- The user does not have to look for a machine with the appropriate resources. LSF does it for you.
- Many available job slots!
  - Approximately 2100 multi-core LSF servers.

LSF Servers

- LSF commands for querying and job submission can only be performed from licensed LSF hosts.

- Public interactive servers licensed for LSF.
  Linux: noric and iris (RHEL3, RHEL4, RHEL5 (test))
  Solaris: tersk and flora (Solaris 10)

- We do not allow interactive logins on most of our batch servers.
  - Some exceptions for group specific servers (SDC, KIPAC, SIMES, etc)

Refer to:

http://www.slac.stanford.edu/comp/unix/public-machines.html
for a description of SLAC's public access machines and batch servers.

Interactive Servers

Load balanced interactive server pools accessible via ssh.

Pool Name	Intended Use
iris	Linux light interactive work
noric	Linux compute intensive interactive work
flora	Solaris light interactive work
tersk	Solaris compute intensive work

You may and should ssh into any of these via their pool names.

For example: ssh noric

Interactive Servers

Linux load balanced interactive servers accessible by kernel.

Pool Name	Intended Use
rhel3-32	Interactive compute intensive work.
rhel4-32	Interactive compute intensive work
rhel4-64	Interactive compute intensive work
rhel5-64-test	Interactive compute intensive work (RHEL5 testing)

You may and should ssh into any of these via their pool names.

For example: ssh rhel4-64

What is a batch queue?

- A cluster-wide container for jobs. All jobs wait in queues until they are scheduled and dispatched to hosts.
- Each queue can use all server hosts in the cluster, or a configured subset of the server hosts.
- Each of our "general" queues are differentiated by their CPU and RUN (wall clock) time limits and use a subset of our LSF server hosts.

General Queues

- The following "general" queues are accessible to all SLAC users.

  express
  short
  medium
  long
  xlong
  xxl
  idle

Batch Queues

- Approximately 2700 cores/jobs slots available to the general queues.
- Special group queues for running batch jobs on servers purchased for the exclusive use of these groups (FGST, Babar, KIPAC, SIMES etc).
- Administrative queues for use by the LSF administrators.
- Preemptable queues. (idle) Jobs preempted by higher priority jobs are suspended until a job slot becomes available.
- Parallel processing queues for Myrinet clusters.

Useful LSF Commands

bsub	submit a batch job to LSF
bjobs	display batch job information
bkill	kill batch job
bmod	modify job submission options
bqueues	display batch queue information
busers	displays information about batch users
lshosts	display LSF host information

For more details use: *man <command_name>*.

Using bsub

- To submit batch jobs to the SLAC LSF cluster use the *bsub* command.

  bsub [*bsub options*] *command* [*arguments*]

  For example:

  bsub -o *outputfilename* date -u

Using bsub

Example of a simple bsub:

```
iris01 sf/neal> bsub hostname
Job <235254> is submitted to default queue <short>.

iris01 sf/neal> bjobs
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
235254  neal    PEND  short      iris01                  hostname   Mar  4 19:17

iris01 sf/neal> bjobs
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
235254  neal    RUN   short      iris01      yili0146    hostname   Mar  4 19:17

iris01 sf/neal> bjobs 235254
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
235254  neal    DONE  short      iris01      yili0146    hostname   Mar  4 19:17
```

Using bsub

Output from my simple batch job:

```
Job <hostname> was submitted from host <iris01> by user <neal>.
Job was executed on host(s) <yili0146>, in queue <short>, as user <neal>.
</u/sf/neal> was used as the home directory.
</u/sf/neal> was used as the working directory.
Started at Sun Mar  4 19:21:19 2007
Results reported at Sun Mar  4 19:21:57 2007
Your job looked like:

------------------------------------------------------------
# LSBATCH: User input
hostname
------------------------------------------------------------

Successfully completed.
Resource usage summary:
CPU time   :      0.22 sec.
Max Memory :         3 MB
Max Swap   :        11 MB
Max Processes  :         3
Max Threads    :         3

The output (if any) follows:
yili0146
```

Using bsub

Default behavior using bsub at SLAC.

- Job will be submitted to the default *short* job queue.
- Output will be returned via email.
- Job will be scheduled on a host of the same OS type.

  SUN5
  LINUX
  MACOSX
  WINDOWS

A few useful bsub options.

- Submit with a output file specification: bsub -o

  Example: bsub –o ~/myjunk/date.output date

- Submit with a CPU limit (normalized): bsub -c

  Example: bsub -c 24:00 date

- Submit with a RUN limit (wallclock): bsub -W

  Example: bsub -W 24:00 date

- Submit with a jobname: bsub -J "job_name"

  Example: bsub -J "Date_job" date

Useful LSF Commands

- bqueues

```
87 iris01 neal/bin> bqueues
QUEUE_NAME      PRIO STATUS          MAX JL/U JL/P JL/H NJOBS  PEND   RUN  SUSP
…
short           185  Open:Active       -    -    -    -     0     0     0     0
medium          180  Open:Active       -    -    -    -   153   102    51     0
long            175  Open:Active       -    -    -    -   897   757   140     0
xlong           170  Open:Active       -    -    1    2  1636  1359   277     0
genmpiq         168  Open:Active       -    -    -    -     0     0     0     0
xxl             165  Open:Active     160   64    -    1    56     1    55     0
…
```

- busers

```
85 iris01 neal/bin> busers
USER/GROUP          JL/P    MAX  NJOBS   PEND    RUN  SSUSP  USUSP    RSV
neal                   -      -      0      0      0      0      0      0

79 sprocket sf/neal> busers moritzb
USER/GROUP          JL/P    MAX  NJOBS   PEND    RUN  SSUSP  USUSP    RSV
moritzb                -      -    384      0    384      0      0      0
```

Useful LSF Commands

- lshosts

```
50 sprocket sf/neal> lshosts
HOST_NAME      type    model  cpuf ncpus maxmem maxswp server RESOURCES
farmboss1     LINUX AMD_2400   6.7     4 15976M 16386M    Yes (linux linux64 rhel40 master)
farmboss2     LINUX AMD_2400   6.7     4 15976M 16386M    Yes (linux linux64 rhel40  master)
farmnfs        SUN5   UF_900   2.8     2  4096M  7209M    Yes (solaris sol9 master)
farmhand       SUN5  UT1_440   1.0     1   256M  2220M    Yes (bcs solaris sol9)
sunlics1       SUN5  UT1_440   1.0     2  2048M  5731M    Yes (lics solaris sol10)
sunlics2       SUN5  UT1 440   1.0     2  2048M  3673M    Yes (lics solaris sol10)
sunlics3       SUN5  UT1_440   1.0     2  2048M  5726M    Yes (lics solaris sol10)
sprocket      LINUX   PC_200   0.5     1  2009M  4094M    Yes (linux linux64 rhel40 dungheap)
adam         MACOSX  G5_2000   4.8     -      -      -    Yes (macosx ppc_darwin)
[…]

51 sprocket sf/neal> lshosts cob0001
HOST_NAME      type    model  cpuf ncpus maxmem maxswp server RESOURCES
cob0001       LINUX AMD_2000   7.7     4  3959M 16386M    Yes (bs linux rhel30 cob)
```

Batch Job Scheduling Policy

- By default LSF is configured for FCFS scheduling.
- SLAC uses fairshare scheduling in the general queues.
- Fairshare controls how resources are shared between competing users or user groups.
- Job priorities are dynamic and change based upon your usage in the queues over the last few days. (Usage values decay over a period of hours.)

What is an LSF "resource"?

- LSF uses built-in and configured resources to track resource availability and usage.
- Jobs are scheduled according to the resources available on individual hosts.
- LSF monitors resource usage of running jobs.
- Users may specify resource requirements for particular jobs.

Good Practice

- Specify output files for batch job output. (bsub with -o or -oo options). Make sure the file path exists and that you have the appropriate permissions.
- Use /nfs for NFS file path names. Do not use the automounter /a path.
- Everything required by the batch job (incl. binary) needs to be visible from the batch nodes. (i.e. in AFS or NFS).
- Before submitting 100s of jobs to LSF, please try submitting a smaller number to ensure that you get the expected results.
- LSF can handle tens of thousands of jobs. However we would prefer that not all of them are yours.
- Scripting batch commands is OK but please be nice! Running commands such as bjobs every second is unnecessary and can cause excessive load on the LSF master.

Good Practice

- Have your jobs use local /scratch space on the LSF servers for job files and output files!
- Having many jobs doing many reads and writes to either AFS or NFS file systems can degrade the performance of their servers.
- Work locally (/scratch) then move the data.

Using /scratch

- Using local /scratch space is more efficient for constant writing and/or reading than doing so via NFS or AFS (i.e. over the network).
- Most of our batch server machines have local /scratch file systems that can be used as temporary space for your batch job input and output files.
- Create a wrapper for your batch program that does the following.
  - Create a directory in /scratch using the batch job ID ($LSB_JOBID).
  - Copy any required input files to your /scratch directory.
  - Write your program output to the newly created directory.
  - When the program/script/command finishes copy the output file to a more permanent location.
  - Remove your job directory.

Using /scratch

Sample shell script using /scratch.

```
#!/bin/tcsh -f
# Shell script for demonstrating use of batch server local
# /scratch space. NVA 3/2008
# Make a directory for my job in /scratch using batch job ID
# variable set by LSF and mktemp for randomness.

set JOBFILEDIR=`mktemp -d /scratch/$LSB_JOBID.XXXXXX`

# Create environmental variables for the job output file.
set OUTPUTFILE=$JOBFILEDIR/lsfhosts.out

# Copy my input file to the local job file directory.
cp -p /u/sf/neal/lsf.hosts $JOBFILEDIR/lsf.hosts

# Run commands and redirect output to my output file.
echo "The approx. number of licensed CPUs in our LSF cluster" > $OUTPUTFILE
/u/sf/neal/bin/addup 5 $JOBFILEDIR/lsf.hosts >> $OUTPUTFILE
echo "This is a test for job $LSB_JOBID" >> $OUTPUTFILE
echo "The JOBFILEDIR is $JOBFILEDIR" >> $OUTPUTFILE

# Copy my output file from batch server to the output file I specified using "bsub -o" ($LSB_OUTPUTFILE).
cp -p $OUTPUTFILE $LSB_OUTPUTFILE

# Clean up after myself!
rm -R $JOBFILEDIR
```

Using /scratch

Running the sample shell script in LSF.

```
41 iris03 sf/neal> bsub -R scratch -o ~neal/tmp/scr_test.out ~neal/bin/scr_test
Job <698922> is submitted to default queue <short>.

48 iris03 sf/neal> bjobs 698922
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
698922  neal    DONE  short      iris03      boer0008    */scr_test Jul 24 19:33

53 iris03 sf/neal> ls -l ~neal/tmp/scr test.out
-rw-r--r--    1 neal     sf           1013 Jul 24 19:33
    /u/sf/neal/tmp/scr_test.out

54 iris03 sf/neal> cat ~neal/tmp/scr_test.out
The approx. number of licensed CPUs in our LSF cluster
5788
This is a test for job 698922
```

Batch Job Exit Codes

- Job exit codes 1-128 are from whatever the user is running while those exceeding 128 are the signal values modulo 128.

  Example:

  A job exit code of 137 would indicate that the job was sent SIGKILL (137-128=9) or kill signal 9.

  A job exit code of 152 would indicate that the job was sent SIGXCPU (152-128=24) or kill signal 24.

- To determine the signal name and number use *man.*

  Linux: `man 7 signal`

  Solaris: `man -s3head signal`

Is LSF having problems?

```
batch system daemon not responding ... still trying
batch system daemon not responding ... still trying
batch system daemon not responding ... still trying
```

- SLAC's LSF cluster can be very busy at times causing the LSF master to respond slowly to your command requests (bsub, bjobs, etc).

  This does not effect jobs already running or pending in the LSF cluster.
  It only affects LSF's ability to talk to you. The commands will eventually complete.

- If you see these messages Monday through Thursday between 19:35 and 19:55 (7:35-7:55PM) we automatically run an LSF reconfiguration during those times.

- Scheduled outage or reconfiguration of the LSF cluster (usually announced in comp-out).

- If you experience this for very long periods (> 30 minutes) please do not hesitate to notify us by emailing unix-admin@slac.stanford.edu. This can indicate a problem with LSF.

LSF Documentation

- SLAC specific LSF documentation.

  http://www.slac.stanford.edu/comp/unix
  Click on "High Performance"

- Platform LSF documentation.

  http://www.slac.stanford.edu/comp/unix/package/lsf/currdoc/html/index.html
  http://www.slac.stanford.edu/comp/unix/package/lsf/currdoc/pdf/manuals/

Problem Reporting

Send email to:

unix-admin@slac.stanford.edu

Questions ?