LSF Job Manage System HOWTO

The High Performance Computing Center(HPCC) at SJTU<http://hpt.sjtu.edu.cn>

Updated at 2013-10-08

This document will instruct you to submit and manage jobs via LSF, it contains the information about using LSF to submit, check and delete jobs.

Following the document's operation and feedback's methods will help you complete the job successfully. It will be very kind to provide advices, thank you!

# Check the Cluster's Running Status via LSF

## Check the LSF Computing Nodes List bhosts

# bhostsHOST\_NAME STATUS JL/U MAX NJOBS RUN SSUSP USUSP RSVfat01 ok - 16 0 0 0 0 0fat02 ok - 16 0 0 0 0 0fat03 ok - 16 0 0 0 0 0fat04 ok - 16 0 0 0 0 0fat05 ok - 16 0 0 0 0 0fat06 ok - 16 0 0 0 0 0fat07 ok - 16 0 0 0 0 0fat08 ok - 16 0 0 0 0 0fat09 ok - 16 0 0 0 0 0fat10 ok - 16 0 0 0 0 0......

## Check the LSF Queues bqueues

Check the whole queues' overall information:

# bqueuesQUEUE\_NAME PRIO STATUS MAX JL/U JL/P JL/H NJOBS PEND RUN SUSPcpu 40 Open:Active - - - - 2072 0 2072 0fat 40 Open:Active - - - - 0 0 0 0gpu 40 Open:Active - - - - 288 0 288 0mic 40 Open:Active - - - - 0 0 0 0cpu-fat 40 Open:Active - - - - 16 0 16 0

Check for some queue's information:

# bqueues fatQUEUE\_NAME PRIO STATUS MAX JL/U JL/P JL/H NJOBS PEND RUN SUSPfat 40 Open:Active - - - - 0 0 0 0

## Check the load of computing nodes lsload

Check the overall load:

# lsloadHOST\_NAME status r15s r1m r15m ut pg ls it tmp swp memnode011 ok 0.0 0.3 0.4 0% 0.0 0 49024 193G 62G 61Gnode039 ok 0.0 0.6 0.5 0% 0.0 0 49024 194G 62G 61Gnode041 ok 0.0 0.0 0.0 0% 0.0 0 49024 194G 62G 61Gnode050 ok 0.0 0.3 0.4 0% 0.0 0 49024 194G 62G 60Gnode064 ok 0.0 0.7 0.6 0% 0.0 0 49024 194G 62G 61Gnode077 ok 0.0 0.7 0.5 0% 0.0 0 49024 193G 62G 61G......

Check for some node's load:

# lsload node001HOST\_NAME status r15s r1m r15m ut pg ls it tmp swp memnode01 ok 0.3 0.1 0.1 1% 0.0 0 332 152G 62G 61G

# Submit the Jobs via LSF bsub

## Submit Jobs Manualy

LSF uses bsub to submit jobs. The format of bsub is:

bsub -n Z -q QUEUENAME -i INPUTFILE -o OUTPUTFILE COMMAND

Z is the number of threads needed, -q assign the queue. If there is no option -q, the system will submit the jobs to the default queue. INPUTFILE is name of the file read by the program, OUTPUTFILE is the output file's name.

For the serial job, COMMAND can directly be your program's name. Example: submit the serial program mytest via LSF:

bsub -n 1 -q q\_default -o mytest.out ./mytest

For the MPI parallel program, the format of COMMAND is -a mpich\_gm mpirun.lsf PROG\_NAME. Example: submit the parallel program mytest via LSF which uses 16 threads:

bsub -n 16 -q q\_default -o mytest.out -a mpich\_gm mpirun.lsf ./mytest

## Interactive Batch Submit

You can start up an interactive shell environment by using bsub to submit multiple parallel jobs whose running arguments are the same:

# bsubbsub> -n 16bsub> -q q\_defaultbsub> -o output.txtbsub> COMMAND1bsub> COMMAND2bsub> COMMAND3

It is equal to:

bsub -n 16 -q q\_default -o output.txt COMMAND1bsub -n 16 -q q\_default -o output.txt COMMAND2bsub -n 16 -q q\_default -o output.txt COMMAND3

## Write a LSF scripts to submit jobs

#BSUB -n 16#BSUB -q q\_default#BSUB -o output.txt-a mpich\_gm mpirun.lsf ./mytest

bsub also accepts the state of jobs from stdin, that means we can write the LSF script to submit jobs. bsub's script is easy to write, the code above is an example named bsub.script, submit bsub.script to LSF via input redirection:

bsub < bsub.script

It is equal to:

bsub -n 16 -q q\_default -o output.txt -a mpich\_gm mpirun.lsf ./mytest

## A more complicated LSF script

#BSUB -J HELLO\_MPI#BSUB -o job.out#BSUB -e job.err#BSUB -n 256source /lustre/utility/intel/composer\_xe\_2014.3.163/bin/compilervars.sh intel64source /lustre/utility/intel/mkl/bin/intel64/mklvars\_intel64.shsource /lustre/utility/intel/impi/4.1.1.036/bin64/mpivars.shMPIRUN=`which mpirun`EXE="./mpihello"CURDIR=$PWDcd $CURDIRrm -f nodelist nodes >& /dev/nulltouch nodelisttouch nodesNP=0for host in `echo $LSB\_MCPU\_HOSTS |sed -e 's/ /:/g'| sed 's/:n/\nn/g'`doecho $host >> nodelistecho $host | cut -d ":" -f1 >> nodesnn=`echo $host | cut -d ":" -f2`NP=`echo $NP+$nn | bc`done

# Other Job Manage Operations

## Check the jobs' status bjobs

Check the submitted jobs' running status:

bjobs

Display the jobs' running status as wide format:

bjobs -w

Display all the jobs:

bjobs -a

Display the running jobs:

bjobs -r

Display the pending jobs and reasons:

bjobs -p

Display the suspending jobs and reasons:

bjobs -s

Display detailed information of job JOBID:

bjobs -l JOBID

## Kill the Jobs bkill

Kill the jobs unwanted:

bkill

Kill the job JOBID:

bkill JOBID

Remove the job JOBID from LSF instead of waiting its progresses killed by the operating system:

bikill JOBID

## Monitor the Output of Jobs bpeek

Display the stdout and stderr output of a unfinished batch job

bpeek

Display the output of the job with the specified ID JOBID

bpeek JOBID

## Jobs' History Information bhist

display the history of batch jobs

bhist

Display the specified job(s) JOBID only

bhist JOBID