

# LSF 作业管理系统使用方法

上海交通大学高性能计算中心

<http://hpc.sjtu.edu.cn>

2013 年 9 月 9 日更新

## 目录

<b>1</b>	<b>通过 LSF 查看集群运行状态</b>	<b>2</b>
1.1	查看 LSF 计算节点列表 <b>bhosts</b> . . . . .	2
1.2	查看 LSF 队列 <b>bqueues</b> . . . . .	2
1.3	查看计算节点负载 <b>lsload</b> . . . . .	3
<b>2</b>	<b>使用 LSF 提交作业<b>bsub</b></b>	<b>3</b>
2.1	手动提交作业 . . . . .	3
2.2	交互式批量提交 . . . . .	4
2.3	编写 LSF 作业控制脚本 . . . . .	4
2.4	一个功能更完整的 LSF 作业控制脚本 . . . . .	5
<b>3</b>	<b>其他作业管理操作</b>	<b>6</b>
3.1	查看作业状态 <b>bjobs</b> . . . . .	6
3.2	终止作业 <b>bkill</b> . . . . .	6
3.3	监视作业输出 <b>bpeek</b> . . . . .	7
3.4	作业历史信息 <b>bhist</b> . . . . .	7
<b>4</b>	<b>参考资料</b>	<b>7</b>

这份文档将指导您通过 LSF 提交和管理作业。文档包含了使用 LSF 提交、查看、删除作业的相关内容。

遵循文档的操作规范和反馈方法，将帮助您顺利完成工作。也欢迎大家对文档内容提出建议，谢谢！

# 1 通过 LSF 查看集群运行状态

## 1.1 查看 LSF 计算节点列表<sub>bhosts</sub>

# bhosts								
HOST_NAME	STATUS	JL/U	MAX	NJOBS	RUN	SSUSP	USUSP	RSV
fat01	ok	-	16	0	0	0	0	0
fat02	ok	-	16	0	0	0	0	0
fat03	ok	-	16	0	0	0	0	0
fat04	ok	-	16	0	0	0	0	0
fat05	ok	-	16	0	0	0	0	0
fat06	ok	-	16	0	0	0	0	0
fat07	ok	-	16	0	0	0	0	0
fat08	ok	-	16	0	0	0	0	0
fat09	ok	-	16	0	0	0	0	0
fat10	ok	-	16	0	0	0	0	0
.....								

## 1.2 查看 LSF 队列<sub>bqueues</sub>

查看所有队列的整体信息：

# bqueues										
QUEUE_NAME	PRIO	STATUS	MAX	JL/U	JL/P	JL/H	NJOBS	PEND	RUN	SUSP
cpu	40	Open:Active	-	-	-	-	2072	0	2072	0
fat	40	Open:Active	-	-	-	-	0	0	0	0
gpu	40	Open:Active	-	-	-	-	288	0	288	0
mic	40	Open:Active	-	-	-	-	0	0	0	0
cpu-fat	40	Open:Active	-	-	-	-	16	0	16	0

查看某个队列的信息：

```
# bqueues fat
QUEUE_NAME      PRIO STATUS      MAX JL/U JL/P JL/H NJOBS  PEND  RUN  SUSP
fat              40  Open:Active    -   -   -   -      0     0    0    0
```

### 1.3 查看计算节点负载lsload

查看整体负载:

```
# lsload
HOST_NAME      status r15s  r1m  r15m  ut   pg  ls   it   tmp  swp  mem
node011        ok    0.0  0.3  0.4  0%   0.0  0 49024 193G 62G 61G
node039        ok    0.0  0.6  0.5  0%   0.0  0 49024 194G 62G 61G
node041        ok    0.0  0.0  0.0  0%   0.0  0 49024 194G 62G 61G
node050        ok    0.0  0.0  0.0  0%   0.0  0 49024 194G 62G 60G
node064        ok    0.0  0.7  0.6  0%   0.0  0 49024 194G 62G 61G
node077        ok    0.0  0.7  0.5  0%   0.0  0 49024 194G 62G 61G
.....
```

查看某个节点的负载:

```
# lsload node001
HOST_NAME      status r15s  r1m  r15m  ut   pg  ls   it   tmp  swp  mem
node001        ok    0.3  0.1  0.1  1%   0.0  0 332 152G 62G 61G
```

## 2 使用 LSF 提交作业bsub

### 2.1 手动提交作业

LSF 使用 bsub 提交作业。bsub 命令的格式是:

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

其中: z提交作业需要的线程数, -q指定作业提交的队列。如果不添加-q选项, 系统将把作业提交到默认的作业队列。INPUTFILE表示程序需要读入的文件

名，`OUTPUTFILE`表示输出文件名，作业提交后的输出到标准输出信息将会保存在这个文件中。

对于串行作业，`COMMAND`可以直接使用您的程序名。例如，将串行程序`mytest`的通过 LSF 提交：

```
bsub -n 1 -q q_default -o mytest.out ./mytest
```

对于 MPI 并行作业，`COMMAND`的格式为`-a mpich_gm mpirun.lsf PROG_NAME`。例如，将并行程序`mytest`，通过 LSF 提交，使用 16 个线程运行这个作业：

```
bsub -n 16 -q q_default -o mytest.out -a mpich_gm mpirun.lsf ./mytest
```

## 2.2 交互式批量提交

使用`bsub`还可以启动一个交互式的 `shell` 环境，一次提交多个运行参数相同的并行作业。例如下面的操作：

```
# bsub
bsub> -n 16
bsub> -q q_default
bsub> -o output.txt
bsub> COMMAND1
bsub> COMMAND2
bsub> COMMAND3
```

等价于：

```
bsub -n 16 -q q_default -o output.txt COMMAND1
bsub -n 16 -q q_default -o output.txt COMMAND2
bsub -n 16 -q q_default -o output.txt COMMAND3
```

## 2.3 编写 LSF 作业控制脚本

```
#BSUB -n 16
#BSUB -q q_default
```

```
#BSUB -o output.txt
-a mpich_gm mpirun.lsf ./mytest
```

bsub还接受来自标准输入的作业说明。因此，我们可以编写 LSF 脚本提交作业。bsub 脚本简单易写，上面一段代码是名为bsub.script的完整例子，将bsub.script通过输入重定向，提交给 LSF：

```
bsub < bsub.script
```

这等价于：

```
bsub -n 16 -q q_default -o output.txt -a mpich_gm mpirun.lsf ./mytest
```

## 2.4 一个功能更完整的 LSF 作业控制脚本

```
#BSUB -J HELLO_MPI
#BSUB -o job.out
#BSUB -e job.err
#BSUB -n 256
source /lustre/utility/intel/composer_xe_2014.3.163/bin/compilervars.sh intel64
source /lustre/utility/intel/mkl/bin/intel64/mklvars_intel64.sh
source /lustre/utility/intel/impi/4.1.1.036/bin64/mpivars.sh
MPIRUN=`which mpirun`
EXE="./mpihello"
CURDIR=$PWD
cd $CURDIR
rm -f nodelist nodes >& /dev/null
touch nodelist
touch nodes
NP=0
for host in `echo $LSB_MCPU_HOSTS |sed -e 's/ /:/g'| sed 's/:n/\nn/g'`
do
echo $host >> nodelist
echo $host | cut -d ":" -f1 >> nodes
nn=`echo $host | cut -d ":" -f2`
NP=`echo $NP+$nn | bc`
done
```

## 3 其他作业管理操作

### 3.1 查看作业状态<sub>bjobs</sub>

检查已提交的作业的运行状态：

```
bjobs
```

以宽格式来显示作业运行状态：

```
bjobs -w
```

显示所有作业：

```
bjobs -a
```

显示正在运行的作业：

```
bjobs -r
```

显示等待运行 (pending) 的作业和等待的原因：

```
bjobs -p
```

显示已经挂起 (suspending) 的作业和挂起的原因：

```
bjobs -s
```

显示JOBID这个作业的所有信息：

```
bjobs -l JOBID
```

### 3.2 终止作业<sub>bkill</sub>

终止不需要的作业：

```
bkill
```

终止JOBID这个作业:

```
bkill JOBID
```

直接将作业JOBID从 LSF 中移除，而不等待该作业的进程在操作系统中终结:

```
bikill JOBID
```

### 3.3 监视作业输出`bpeek`

当作业正在运行时，显示它的标准输出，监视作业运行:

```
bpeek
```

查看JOBID的标准输出:

```
bpeek JOBID
```

### 3.4 作业历史信息`bhist`

显示作业的历史情况:

```
bhist
```

显示JOBID作业的历史情况:

```
bhist JOBID
```

## 4 参考资料