

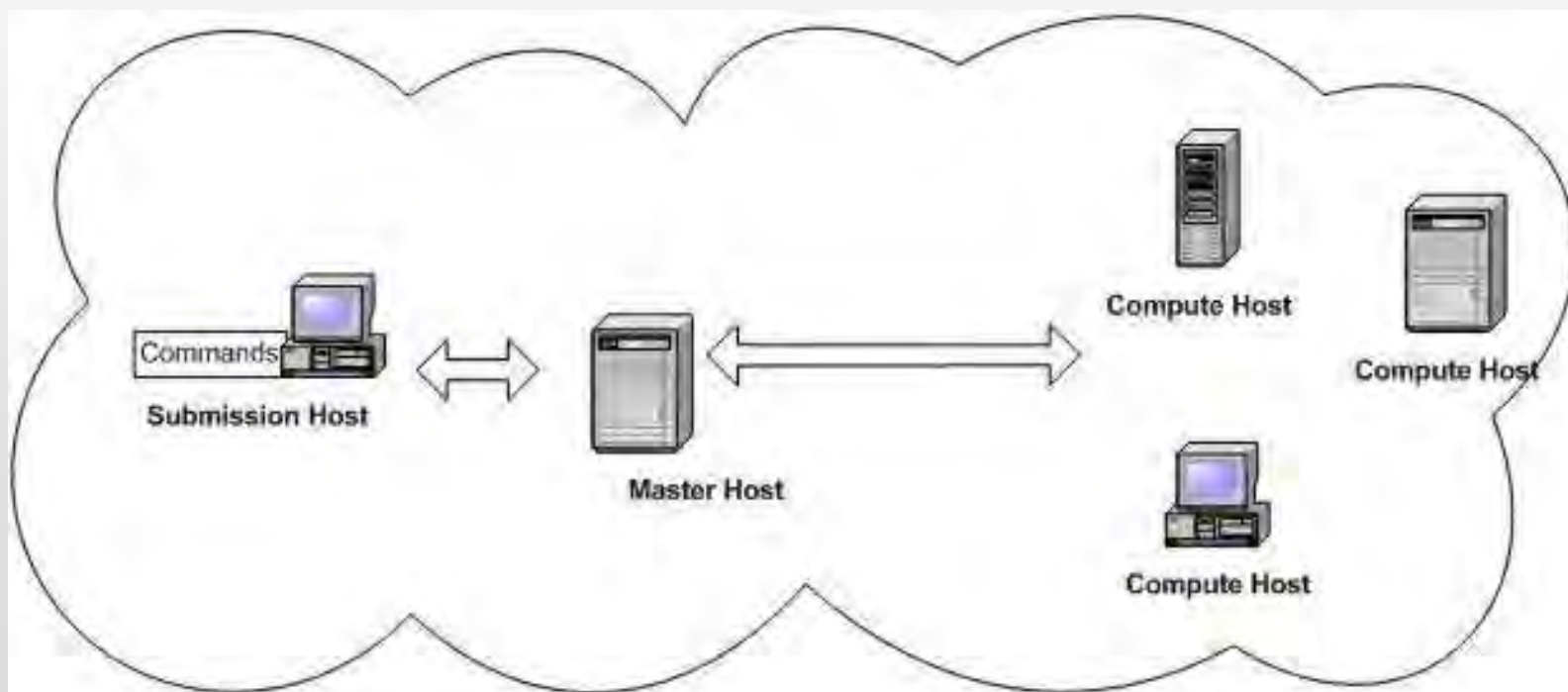
# LSF基本命令使用培训

# LSF基本命令使用培训

- 基本概念
  - 集群
  - 节点
  - 用户
  - 作业
  - 队列
- 基本命令
- 常用配置文件（管理员使用）

## 集群的基本概念-集群

一群或一组计算机，通过运行LSF来共同工作，从而组成一个整体，不但能把各个节点机的计算能力集合在一起，而且彼此间能够分担负载、分享资源。一个集群，就是把分散的计算资源整合，向外提供一个统一的系统映像。



# 集群的基本概念-节点

## • 管理节点(manager hosts)

- 负责集群的管理
- 一个master节点，多个slave节点
- 管理节点可担当计算节点
- 管理节点可担当登陆节点

## • 登陆节点 (submission hosts,client hosts)

- 用户通过登陆节点登陆集群，使用计算资源

## • 计算节点(compute hosts,server hosts)

- 集群的计算资源，负责具体计算工作)
- 计算节点可担当登陆节点

## 集群的基本概念-计算节点的cpu、thread、core、socket

```
[root@node10 ~]# lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                40
On-line CPU(s) list:   0-39
Thread(s) per core:    2
Core(s) per socket:    10
Socket(s):              2
```

以core而不以thread作为最基本的计算单元使用。

# 集群的基本概念-用户

## • 集群管理员

- 首席管理员 首席管理员 是必须存在的，是在安装期间定义的。首席集群管理员有权执行针对整个集群有效的操作，比如更改配置文件，重新设置集群，以及控制所有用户提交的作业。
- 一般的管理员 一般集群管理员是可选的，在安装过程中或其后所设置的。一般集群管理员能够执行针对集群内所有作业和队列的管理操作，但没有权限改变集群配置文件。

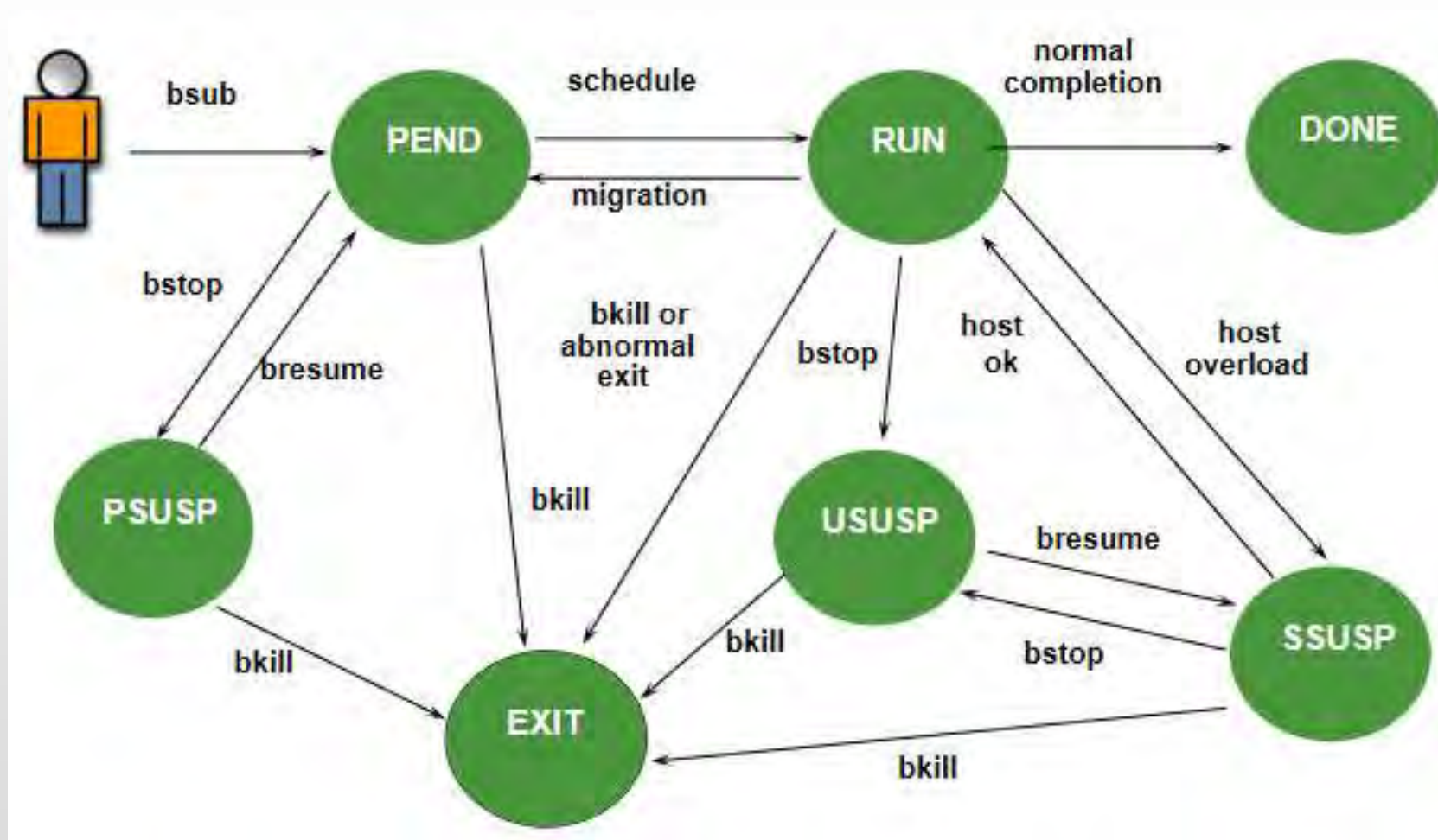
## • 一般用户

- 集群资源的申请使用者
- 用户组：在Isf中可以把具有相似资源使用需求的用户定义成一组，这样可以便于配置用户和资源之间分配策略

# 集群的基本概念-作业

- 作业是集群中最重要的一个概念，用户通过提交作业申请集群计算资源，完成计算任务
- 作业状态
  - PEND（待定）-- 作业在队列中等待调度与分派
  - RUN（运行）-- 作业已经被分派到节点机上，正在运行
  - DONE（完成）-- 作业已正常结束，正常退出值为0
  - EXIT（退出）-- 作业已经结束，但退出值不是0
  - PSUSP（待定挂起）-- 作业在待定时被挂起
  - USUSP（用户挂起）-- 作业被用户指定挂起
  - SSUSP（系统挂起）-- 作业被LSF系统挂起
  - POST\_DONE（后处理完成）-- 作业的后处理动作，成功完成
  - POST\_ERR（后处理出错）-- 作业的后处理动作，出错
  - UNKWN（不确定）-- mbatchd与执行节点机上的sbatchd失去联系从而使作业状态不确定
  - ZOMBI（僵尸）-- 作业成为僵尸是因为系统失去了与作业执行节点机的联系

## 集群的基本概念-作业状态





# 集群的基本概念-队列

- 用户提交的作业首先被放到特定的队列中
- 队列相当于一系列的待发作业，待发作业在队列中按照预定的顺序排列并等待着使用资源的机会
- 队列实施不同的作业调度和控制策略。所有提交到同一队列中的作业，拥有相同的作业调度与控制策略
- 队列一些基本属性：
  - 名称：这是队列唯一的标识
  - 优先级：队列中的作业使用资源的优先级越高
  - 队列限制：可以限制节点，作业数目，用户，组，处理器等
  - 调度策略：先到先服务策略（FCFS）、公平策略、抢占策略、排它策略
  - 调度窗口：队列中的作业使用集群资源的时间窗口

# 登录方法

使用方法：通过ssh工具登录：202.114.252.112。

使用自己的账户名和密码。

修改个人密码：

使用ssh工具登录到202.114.252.112上面，运行以下指令：

```
#yppasswd userxxx
```

输入旧密码

输入两次新密码

实际执行过程中userxxx使用想要修改的用户名代替。

# LSF基本管理命令

- **集群管理命令**（普通用户，无需关注）
- **作业管理命令**
- **队列管理命令**
- **节点管理命令**
- **用户管理命令**

# 作业管理命令

- **bsub** – submit jobs
- **bjobs** – display jobs information(recently)
- **bhist** – display jobs information((overrides -d, -p, -r, and -s)
- **bbot** – moves a pending job to the bottom of the queue
- **btop** – moves a pending job to the top of the queue
- **bkill** – sends a signal to kill jobs
- **bmod** – modifies job submission options of a job
- **bpeek** – displays the stdout and stderr of an unfinished job
- **bstop** – suspend unfinished jobs
- **brresume** – resumes one or more suspended jobs
- **bswitch** – switches unfinished jobs to another queue

# 作业管理命令

- `bsub [options] command [cmdargs]`
- `bjobs [-a][-J jobname][-u usergroup|-u all][...] jobID`
- `bhist [-a][-J jobname][-u usergroup|-u all][...] jobID`
- `bbot/btop [jobID | "jobID[index_list]"] [position]`
- `bkill [-J jobname] [-m] [-u ] [-q] [-s signalvalue]`
- `bmod [bsub_options] jobID`
- `bpeek [-f] jobID`
- `bstop/bresume jobID`
- `bswitch destination_queue jobID`

## 作业管理命令-提交作业 (bsub)

- `bsub [options] command [cmdargs]`
- `bsub [commonly used options]`
  - n # - number of CPUs required for the job
  - o *filename* – redirect stdout, stderr to the specified *output file*
  - e *filename* – redirect stderr to the specified error file
  - i *filename* – use the specified file as standard input for the job
  - q *qname* – submits the job to the specified queue
  - m *hname* – select host(s) or host group.
  - J *jobname* – assigns the specified name to the job
- `#source /apps/intel/impi/2018.4.274/bin64/mpivars.sh`
- `#bsub -J testMPI -q normal -n 10 -R span[ptile=1] -o output.txt -e error.txt mpirun hostname`

# 作业管理命令-查看作业信息 (bjobs)

## bjobs Can display parallel jobs

- a Display information about jobs in all states (including finished jobs)
- A Display summarized information about job arrays
- d Display information about jobs that finished recently
- l|-w Display information in long or wide format
- p Display information about pending jobs
- r Display information about running jobs
- g *job\_group* Display information about jobs in specified group
- J *job\_name* Display information about specified job or array
- m *host\_list* Display information about jobs on specified hosts or groups
- P *project\_name* Display information about jobs in specified project
- q *queue\_name* Display information about jobs in specified queue
- u *user\_name* Display information about jobs for specified users/groups

# 作业管理命令-查看作业历史信息 (bhist)

## bhist

- a Display information about all jobs (overrides -d, -p, -r, and -s)
- b|-l|-w Display information in brief, long, or wide format
- d Display information about finished jobs
- p Display information about pending jobs
- s Display information about suspended jobs
- t Display job events chronologically
- C|-D|-S|-T *start\_time,end\_time* Display information about completed, dispatched, submitted, or all jobs in specified time window
- P *project* Display information about jobs belonging to specified project
- q *queue* Display information about jobs submitted to specified queue
- u *username|all* Display information about jobs submitted by user



## 作业管理命令-examples- (bsub,bjobs)

```
[lsfadmin@r71 tmp3]$ bsub -J myjob1 -q normal sleep 10000
Job <1564> is submitted to queue <normal>.
[lsfadmin@r71 tmp3]$ bsub -J myjob2 -q normal sleep 10000
Job <1565> is submitted to queue <normal>.
[lsfadmin@r71 tmp3]$ bsub -J myjob3 -q normal sleep 10000
Job <1566> is submitted to queue <normal>.
[lsfadmin@r71 tmp3]$ bsub -J myjob4 -q normal sleep 10000
Job <1567> is submitted to queue <normal>.
[lsfadmin@r71 tmp3]$ bsub -J myjob5 -q normal sleep 10000
Job <1568> is submitted to queue <normal>.
[lsfadmin@r71 tmp3]$ bsub -J myjob6 -q normal sleep 10000
Job <1569> is submitted to queue <normal>.
[lsfadmin@r71 tmp3]$
[lsfadmin@r71 tmp3]$ bjobs
```

JOBID	USER	STAT	QUEUE	FROM_HOST	EXEC_HOST	JOB_NAME	SUBMIT_TIME
1564	lsfadmi	RUN	normal	r71	r73	myjob1	Aug 7 14:19
1565	lsfadmi	RUN	normal	r71	r72	myjob2	Aug 7 14:19
1566	lsfadmi	PEND	normal	r71		myjob3	Aug 7 14:19
1567	lsfadmi	PEND	normal	r71		myjob4	Aug 7 14:19
1568	lsfadmi	PEND	normal	r71		myjob5	Aug 7 14:19
1569	lsfadmi	PEND	normal	r71		myjob6	Aug 7 14:19

## 作业管理命令-examples- (bjobs bhist)

```
[lsfadmin@r71 tmp3]$ bjobs
JOBID   USER      STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
1564    lsfadmi  RUN   short      r71         r73         myjob1    Aug  7 14:19
1569    lsfadmi  RUN   normal    r71         r72         myjob6    Aug  7 14:19
1568    lsfadmi  PEND  short      r71         r72         myjob5    Aug  7 14:19
1567    lsfadmi  PEND  normal    r71         r72         myjob4    Aug  7 14:19
1566    lsfadmi  PEND  normal    r71         r72         myjob3    Aug  7 14:19
```

```
[lsfadmin@r71 tmp3]$
```

```
[lsfadmin@r71 tmp3]$ bhist
```

Summary of time in seconds spent in various states:

JOBID	USER	JOB_NAME	PEND	PSUSP	RUN	USUSP	SSUSP	UNKWN	TOTAL
1564	lsfadmi	myjob1	1	0	1828	0	0	0	1829
1566	lsfadmi	myjob3	1822	0	0	0	0	0	1822
1567	lsfadmi	myjob4	1818	0	0	0	0	0	1818
1568	lsfadmi	myjob5	1814	0	0	0	0	0	1814
1569	lsfadmi	myjob6	1461	0	349	0	0	0	1810

## 作业管理命令-examples- (bjobs -l jobid)

```
[lsfadmin@r71 tmp3]$ bjobs -l 1565
```

```
Job <1565>, Job Name <myjob2>, User <lsfadmin>, Project <default>, Status <RUN>  
      , Queue <normal>, Command <sleep 10000>, Share group charged </lsfadmin>
```

```
Wed Aug 7 14:19:28: Submitted from host <r71>, CWD <${HOME}/tmp3>;
```

```
Wed Aug 7 14:19:29: Started 1 Task(s) on Host(s) <r72>, Allocated 1 Slot(s) on  
      Host(s) <r72>, Execution Home </apps/users/lsfadmin>, Execution CWD </apps/users/lsfadmin/tmp3>;
```

```
Wed Aug 7 14:25:22: Resource usage collected.
```

```
MEM: 2 Mbytes; SWAP: 0 Mbytes; NTHREAD: 4  
PGID: 89868; PIDs: 89868 89870 89872
```

### MEMORY USAGE:

```
MAX MEM: 2 Mbytes; AVG MEM: 2 Mbytes
```

### SCHEDULING PARAMETERS:

	r15s	rlm	r15m	ut	pg	io	ls	it	tmp	swp	mem
loadSched	-	-	-	-	-	-	-	-	-	-	-
loadStop	-	-	-	-	-	-	-	-	-	-	-

### RESOURCE REQUIREMENT DETAILS:

```
Combined: select[type == local] order[r15s:pg]
```

```
Effective: select[type == local] order[r15s:pg]
```

## 作业管理命令-examples-(btop,bbot)

```
[lsfadmin@r71 tmp3]$ btop 1569
Job <1569> has been moved to position 1 from top.
[lsfadmin@r71 tmp3]$ bjobs
```

JOBID	USER	STAT	QUEUE	FROM_HOST	EXEC_HOST	JOB_NAME	SUBMIT_TIME
1564	lsfadmi	RUN	normal	r71	r73	myjob1	Aug 7 14:19
1565	lsfadmi	RUN	normal	r71	r72	myjob2	Aug 7 14:19
1569	lsfadmi	PEND	normal	r71		myjob6	Aug 7 14:19
1566	lsfadmi	PEND	normal	r71		myjob3	Aug 7 14:19
1567	lsfadmi	PEND	normal	r71		myjob4	Aug 7 14:19
1568	lsfadmi	PEND	normal	r71		myjob5	Aug 7 14:19

```
[lsfadmin@r71 tmp3]$ bbot 1566
Job <1566> has been moved to position 1 from bottom.
[lsfadmin@r71 tmp3]$ bjobs
```

JOBID	USER	STAT	QUEUE	FROM_HOST	EXEC_HOST	JOB_NAME	SUBMIT_TIME
1564	lsfadmi	RUN	normal	r71	r73	myjob1	Aug 7 14:19
1565	lsfadmi	RUN	normal	r71	r72	myjob2	Aug 7 14:19
1569	lsfadmi	PEND	normal	r71		myjob6	Aug 7 14:19
1567	lsfadmi	PEND	normal	r71		myjob4	Aug 7 14:19
1568	lsfadmi	PEND	normal	r71		myjob5	Aug 7 14:19
1566	lsfadmi	PEND	normal	r71		myjob3	Aug 7 14:19

## 作业管理命令-examples-(bstop,bresume)

```
[lsfadmin@r71 tmp3]$ bstop 1565
```

```
Job <1565> is being stopped
```

```
[lsfadmin@r71 tmp3]$ bjobs
```

JOBID	USER	STAT	QUEUE	FROM_HOST	EXEC_HOST	JOB_NAME	SUBMIT_TIME
1564	lsfadmi	RUN	normal	r71	r73	myjob1	Aug 7 14:19
1565	lsfadmi	USUSP	normal	r71	r72	myjob2	Aug 7 14:19
1569	lsfadmi	PEND	normal	r71		myjob6	Aug 7 14:19
1567	lsfadmi	PEND	normal	r71		myjob4	Aug 7 14:19
1568	lsfadmi	PEND	normal	r71		myjob5	Aug 7 14:19
1566	lsfadmi	PEND	normal	r71		myjob3	Aug 7 14:19

```
[lsfadmin@r71 tmp3]$ bresume 1565
```

```
Job <1565> is being resumed
```

```
[lsfadmin@r71 tmp3]$ bjobs
```

JOBID	USER	STAT	QUEUE	FROM_HOST	EXEC_HOST	JOB_NAME	SUBMIT_TIME
1564	lsfadmi	RUN	normal	r71	r73	myjob1	Aug 7 14:19
1565	lsfadmi	RUN	normal	r71	r72	myjob2	Aug 7 14:19
1569	lsfadmi	PEND	normal	r71		myjob6	Aug 7 14:19
1567	lsfadmi	PEND	normal	r71		myjob4	Aug 7 14:19
1568	lsfadmi	PEND	normal	r71		myjob5	Aug 7 14:19
1566	lsfadmi	PEND	normal	r71		myjob3	Aug 7 14:19

## 作业管理命令-examples-(bkill,bswitch)

```
[lsfadmin@r71 tmp3]$ bkill 1565
```

```
Job <1565> is being terminated
```

```
[lsfadmin@r71 tmp3]$ bjobs
```

JOBID	USER	STAT	QUEUE	FROM_HOST	EXEC_HOST	JOB_NAME	SUBMIT_TIME
1564	lsfadmi	RUN	normal	r71	r73	myjob1	Aug 7 14:19
1569	lsfadmi	RUN	normal	r71	r72	myjob6	Aug 7 14:19
1567	lsfadmi	PEND	normal	r71		myjob4	Aug 7 14:19
1568	lsfadmi	PEND	normal	r71		myjob5	Aug 7 14:19
1566	lsfadmi	PEND	normal	r71		myjob3	Aug 7 14:19

```
[lsfadmin@r71 tmp3]$ bswitch short 1568
```

```
Job <1568> is switched to queue <short>
```

```
[lsfadmin@r71 tmp3]$ bjobs
```

JOBID	USER	STAT	QUEUE	FROM_HOST	EXEC_HOST	JOB_NAME	SUBMIT_TIME
1564	lsfadmi	RUN	normal	r71	r73	myjob1	Aug 7 14:19
1569	lsfadmi	RUN	normal	r71	r72	myjob6	Aug 7 14:19
1568	lsfadmi	PEND	short	r71		myjob5	Aug 7 14:19
1567	lsfadmi	PEND	normal	r71		myjob4	Aug 7 14:19
1566	lsfadmi	PEND	normal	r71		myjob3	Aug 7 14:19

## 节点管理命令-查看节点静态信息 (lshosts)

```
[root@r73 ~]# lshosts
HOST_NAME      type      model    cpuf  ncpus  maxmem  maxswp  server  RESOURCES
r71            X86_64   PC6000  116.1    1    3.9G   1.9G    Yes  (mg)
r72            X86_64   PC6000  116.1    1    1.9G   1.9G    Yes  ()
r73            X86_64   PC6000  116.1    -     -     -     Yes  ()
```

HOST\_NAME: 节点名.

type: 节点类别 (芯片架构) .

model: 节点型号.

cpuf: 相对cpu性能因子, cpu速度越快, 相对cpu性能因子越高.

ncpus:节点cpu核数.

maxmem:用户可用的最大物理内存.

maxswp:交换空间.

server:机器是否为计算节点.

RESOURCES:该节点具有系统定义的资源名称.

## 节点管理命令-查看节点负载信息 (lsload)

```
[root@r73 ~]# lsload
HOST_NAME      status  r15s  r1m  r15m  ut    pg  ls    it    tmp    swp    mem
r72            ok     0.6   0.0  0.0   0%    0.0  0    303   24G   1.9G  1.2G
r71            ok     0.8   0.1  0.1   4%    0.0  2     0    22G   1.9G  2.1G
r73            unavail
```

HOSTNAME:节点

status:节点的状态

r15s:15秒运行队列的长度

r1m:1分钟运行队列的长度

r15m:15 分钟的运行队列的长度

ur:CPU的使用率

pg: (内存的) 页调度比率

ls:登陆用户的个数(ls)

it:交互式工作模式下的空闲时间

swp:可用的交换空间的大小

mem:可用的内存大小

tmp:可用的临时空间大小



## 节点管理命令-查看节点负载信息 (lsload-status)

状态	描述
OK	节点可用，能够接收与运行批处理作业或远程任务。
-OK	LIM 进程运行着，但是 RES 进程联系不到。
Busy	不影响批处理作业，只适用于远程作业的调度，例如用 <code>lssrun</code> 命令提交的作业。Busy 表示负载指标已经超出所设的阈值，其阈值是在 <code>lsf.cluster.cluster_name</code> 文件中设置的，可通过 <code>lshosts -l</code> 命令来显示。这个超出设定的负载指标由星号 (*) 标示。
LockW	不影响批处理作业，只适用于远程作业的安排，例如用 <code>lssrun</code> 命令提交的作业。节点是由运行窗口锁定的，运行窗口是在在 <code>lsf.cluster.cluster_name</code> 文件中设置的，可通过 <code>lshosts -l</code> 命令来显示。
LockU	既不会接收新的批处理作业也不接收远程作业。这是由 LSF 管理员或是 root 用户，通过运行 <code>lsadmin limlock</code> 命令来指定的，或有一个排他型的作业运行在这个节点上，比如由 <code>bssub -x</code> 提交的作业。使用 <code>lsadmin limunlock</code> 命令来解开当前节点 LIM 的锁定。
Unavail	节点宕机，或是 LIM 进程联系不到。

## 节点管理命令-查看运行状态 (bhosts)

```
[root@r73 ~]# bhosts
```

HOST_NAME	STATUS	JL/U	MAX	NJOBS	RUN	SSUSP	USUSP	RSV
r71	ok	-	1	0	0	0	0	0
r72	ok	-	1	0	0	0	0	0
r73	unavail	-	1	1	1	0	0	0

HOST\_NAME:节点名.

STATUS:节点状态.

JL/U:每用户在该节点上的最大作业数, (-) 表示没有限制.

MAX:该节点最大作业数.

NJOBS:现在分派到该节点的作业数.

RUN:现在该节点运行的作业数.

SSUSP:现在该节点被系统挂起的作业数

USUSP:现在该节点被用户挂起的作业数.

RSV: 现在该节点被保留的 job slots.

## 节点管理命令-查看运行状态 (bhosts-status)

状态	描述
OK	节点能够接收与运行新的批处理作业。
Unavail	节点处于宕机状态，或是找不到 LIM 和 sbatchd 进程。
Unreach	LIM 进程运行着，但是联系不到 sbatchd 进程。
Closed	节点不接收新的作业，使用 <code>bhosts -l</code> 命令来检查其原因。

## 节点管理命令-关闭/打开节点 (badmin hclose/hopen)

```
[root@r73 ~]# bhosts
HOST_NAME      STATUS      JL/U      MAX  NJOBS      RUN  SSUSP  USUSP  RSV
r71            ok          -         1    0          0    0      0      0
r72            ok          -         1    0          0    0      0      0
r73            unavail    -         1    1          1    0      0      0
[root@r73 ~]# badmin hclose r72
Close <r72> ..... done
[root@r73 ~]# bhosts
HOST_NAME      STATUS      JL/U      MAX  NJOBS      RUN  SSUSP  USUSP  RSV
r71            ok          -         1    0          0    0      0      0
r72            closed     -         1    0          0    0      0      0
r73            unavail    -         1    1          1    0      0      0
[root@r73 ~]#
[root@r73 ~]# badmin hopen r72
Open <r72> ..... done
[root@r73 ~]# bhosts
HOST_NAME      STATUS      JL/U      MAX  NJOBS      RUN  SSUSP  USUSP  RSV
r71            ok          -         1    0          0    0      0      0
r72            ok          -         1    0          0    0      0      0
r73            unavail    -         1    1          1    0      0      0
```

## 队列管理命令-查看队列信息 (bqueues)

```
[root@r73 ~]# bqueues
QUEUE_NAME      PRIO STATUS      MAX JL/U JL/P JL/H NJOBS  PEND  RUN  SUSP
owners          43  Open:Active   -   -   -   -     0     0    0    0
priority        43  Open:Active   -   -   -   -     0     0    0    0
night           40  Open:Inact    -   -   -   -     0     0    0    0
ansys           40  Open:Active   -   -   -   -     0     0    0    0
short           35  Open:Active   -   -   -   -     1     0    1    0
normal          30  Open:Active   -   -   -   -     0     0    0    0
interactive     30  Open:Active   -   -   -   -     0     0    0    0
idle            20  Open:Active   -   -   -   -     0     0    0    0
```

### 队列状态 (status) 说明

状态	描述
----	----

---

Open: Active	接收并派发作业, 正常处理中
--------------	----------------

---

Open: Inactive	只接收作业但不派发作业, 正在收集作业
----------------	---------------------

---

Closed: Active	只派发作业但不接收新作业, 正在排空队列
----------------	----------------------

---

Closed: Inactive	既不接收新作业, 也不派发作业, 所有活动都停止
------------------	--------------------------

## 队列管理命令-查看队列详细信息 (bqueues -l)

```
[root@r73 ~]# bqueues -l normal

QUEUE: normal
  -- For normal low priority jobs, running only if hosts are lightly loaded.  This is the default queue.

PARAMETERS/STATISTICS
PRIO NICE STATUS          MAX JL/U JL/P JL/H NJOBS  PEND  RUN  SSUSP USUSP  RSV
 30   0 Open:Active        -   -   -   -    0    0    0    0    0    0
Interval for a host to accept two jobs is 0 seconds

SCHEDULING PARAMETERS
          rl5s   rlm  rl5m  ut      pg   io  ls   it   tmp  swp  mem
loadSched -     -    -    -      -   -   -   -   -   -   -   -
loadStop  -     -    -    -      -   -   -   -   -   -   -   -

SCHEDULING POLICIES: FAIRSHARE NO_INTERACTIVE
USER_SHARES: [default, 1]

SHARE_INFO_FOR: normal/
  USER/GROUP  SHARES  PRIORITY  STARTED  RESERVED  CPU_TIME  RUN_TIME  ADJUST
znwrsmadmin   1       0.333     0         0         0.0       0         0.000
rsmadmin      1       0.333     0         0         0.0       0         0.000
znw           1       0.333     0         0         0.0       0         0.000
lsfadmin      1       0.333     0         0         2.8       0         0.000

USERS: all
HOSTS: all
```

## 队列管理命令-关闭/开启队列 (badadmin qclose/qopen)

```
[root@r73 ~]# bqueues
QUEUE_NAME      PRIO STATUS      MAX JL/U JL/P JL/H NJOBS  PEND  RUN  SUSP
owners          43 Open:Active   -  -  -  -    0    0    0    0
priority        43 Open:Active   -  -  -  -    0    0    0    0
night           40 Closed:Inact -  -  -  -    0    0    0    0
ansys           40 Closed:Active -  -  -  -    0    0    0    0
short           35 Closed:Active -  -  -  -    1    0    1    0
normal          30 Open:Active   -  -  -  -    0    0    0    0
interactive     30 Open:Active   -  -  -  -    0    0    0    0
idle            20 Open:Active   -  -  -  -    0    0    0    0
[root@r73 ~]#
[root@r73 ~]# badadmin qclose idle
Queue <idle> is closed
[root@r73 ~]# badadmin qopen night
Queue <night> is opened
[root@r73 ~]#
[root@r73 ~]# bqueues
QUEUE_NAME      PRIO STATUS      MAX JL/U JL/P JL/H NJOBS  PEND  RUN  SUSP
owners          43 Open:Active   -  -  -  -    0    0    0    0
priority        43 Open:Active   -  -  -  -    0    0    0    0
night           40 Open:Inact   -  -  -  -    0    0    0    0
ansys           40 Closed:Active -  -  -  -    0    0    0    0
short           35 Closed:Active -  -  -  -    1    0    1    0
normal          30 Open:Active   -  -  -  -    0    0    0    0
interactive     30 Open:Active   -  -  -  -    0    0    0    0
idle            20 Closed:Active -  -  -  -    0    0    0    0
[root@r73 ~]#
```

## 队列管理命令-阻止/激活队列 (badmin qinact/qact)

```
[root@r73 ~]# bqueues
QUEUE_NAME      PRIO STATUS      MAX JL/U JL/P JL/H NJOBS  PEND  RUN  SUSP
owners          43  Open:Active   -   -   -   -     0     0    0    0
priority        43  Open:Active   -   -   -   -     0     0    0    0
night           40  Open:Inact    -   -   -   -     0     0    0    0
ansys           40  Open:Active   -   -   -   -     0     0    0    0
short           35  Open:Inact    -   -   -   -     1     0    1    0
normal          30  Open:Active   -   -   -   -     0     0    0    0
interactive     30  Open:Active   -   -   -   -     0     0    0    0
idle            20  Closed:Active -   -   -   -     0     0    0    0
[root@r73 ~]#
[root@r73 ~]# badmin qinact ansys
Queue <ansys> is inactivated
[root@r73 ~]# badmin qact short
Queue <short> is activated
[root@r73 ~]#
[root@r73 ~]# bqueues
QUEUE_NAME      PRIO STATUS      MAX JL/U JL/P JL/H NJOBS  PEND  RUN  SUSP
owners          43  Open:Active   -   -   -   -     0     0    0    0
priority        43  Open:Active   -   -   -   -     0     0    0    0
night           40  Open:Inact    -   -   -   -     0     0    0    0
ansys           40  Open:Inact    -   -   -   -     0     0    0    0
short           35  Open:Active   -   -   -   -     1     0    1    0
normal          30  Open:Active   -   -   -   -     0     0    0    0
interactive     30  Open:Active   -   -   -   -     0     0    0    0
idle            20  Closed:Active -   -   -   -     0     0    0    0
```



## 用户管理命令-显示用户(组)信息 (busers)

```
[root@r73 ~]# busers all
USER/GROUP          JL/P    MAX  NJOBS  PEND    RUN  SSUSP  USUSP  RSV
default             -       -    -      -      -    -      -      -
lsfadmin            -       -    0      0      0    0      0      0
rsmadmin            -       -    0      0      0    0      0      0
```

busers 命令能够显示用户与用户组的信息。

busers 命令能够显示：

- ◆ 在一个处理器上，用户或用户组最多可以执行的作业数目；
- ◆ 在集群上，用户或用户组最多可以使用的作业槽的数目；
- ◆ 在系统上，用户或用户组最多可以有多少个挂起的作业；
- ◆ 所有这个用户提交的作业所需要的作业槽的总数量；
- ◆ 处于PEND, RUN, SSUSP 和 USUSP 状态的作业，所需要的作业槽的数量

## 常用配置文件

`$LSF_TOP/conf/lsf.conf`

`$LSF_TOP/conf/lsf.cluster`

`$LSF_TOP/conf/lsf.share`

`$LSF_TOP/conf/lSBATCH/cluster71/configdir/lb.queue`

`$LSF_TOP/conf/lSBATCH/cluster71/configdir/lb.hosts`

`$LSF_TOP/conf/lSBATCH/cluster71/configdir/lb.users`

`$LSF_TOP/conf/lSBATCH/cluster71/configdir/lb.params`

## 常用配置文件-lsf.conf-常用配置

```
...  
LSF_LOG_MASK=LOG_WARNING #LOG_WARNING, LOG_DEBUG等  
LSF_MASTER_LIST="master master1 master2"  
LSF_UNIT_FOR_LIMITS=MB #KB,MB,GB,TB  
LSF_USER_DOMAIN=domainnam  
LSF_RSH="ssh"  
...
```

# 常用配置文件-Isf.cluster-常用配置

.....

## #add 集群管理员

```
Begin ClusterAdmins
```

```
Administrators = Isfadmin user1 user2
```

```
End ClusterAdmins
```

## #增加新的节点

```
Begin Host
```

```
HOSTNAME model type server RESOURCES
```

```
r72 ! ! 1 ()
```

```
r73 ! ! 1 ()
```

```
r71 ! ! 1 (mg)
```

```
End Host
```

.....

# 常用配置文件-lsb.queues-常用配置

.....

## #增加新的队列

Begin Queue

QUEUE\_NAME = idle

PRIORITY = 20

#RUN\_WINDOW

FAIRSHARE = USER\_SHARES [ [default, 1] ]

#USERS

#HOSTS

End Queue

.....

## 常用配置文件-Isb.hosts-常用配置

- .....

Begin HostGroup

```
GROUP_NAME    GROUP_MEMBER    #GROUP_ADMIN # Key words
```

```
hgroup1      (hostA hostD )  #()          # Define a host group
```

```
hgroup2      (hostB hostE )  #()          # Define a host group
```

```
hgroup3      (hostC hostF )  #()          # Define a host group
```

End HostGroup

.....

## 常用配置文件-lsb.users -常用配置

• ...

Begin UserGroup

GROUP_NAME	GROUP_MEMBER	USER_SHARES	#GROUP_ADMIN
ugroup1	(user1 user2 user3 user4)	([user1, 4] [others, 10])	#(user1 user2[full])
ugroup2	(all)	([user3, 10] [others, 15])	#(user3[usershares])

End UserGroup

.....

Begin User

USER_NAME	MAX_PEND_JOBS	
#user1	800	# user1 has pend threshold of 800
#ugroup1	1000	# collectively ugroup1 has threshold of 1000
#default	100	# default, any user/group has threshold of 100

End User