# Torque

PBS是功能最为齐全，历史最悠久，支持最广泛的本地集群调度器之一。 PBS的目前包括openPBS，PBS Pro和Torque三个主要分支。其中OpenPBS是最早的PBS系统，目前已经没有太多后续开发，PBS pro是PBS的商业版本，功能最为丰富。Torque是Clustering公司接过了OpenPBS，并给与后续支持的一个开源版本。

| **Action** | **Slurm** | **Torque/PBS** | **Maui** |
| --- | --- | --- | --- |
| Get information about the job | scontrol show job <jobid> | qstat -f <jobid> | checkjob |
| Display the queue information | squeue | qstat | showq |
| Delete a job | scancel <jobid> | qdel |  |
| Submit a job | srun/sbatch/salloc | qsub |  |
| Display how many processors are currently free |  |  | showbf |
| Display the expected start time for a job | squeue --start |  | showstart <jobid> |
| Display information about available queues/partitions | sinfo/sshare | qstat -Qf |  |

Torque由头节点和计算节点组成，头节点运行pbs\_server，计算节点运行pbs\_mom。用户可在任意主机上提交任务。头节点同时运行pbs\_sched。调度器与pbs\_server通信，采用局部策略分配资源使用和各节点任务。最简单的是FIFO调度，还可以采用Maui或Moab提供的复杂调度。

当用户利用qsub将任务提交给pbs\_server时，pbs\_server通知调度器，调度器向保存在pbs\_server中的节点发送运行任务的指令。Pbs\_server发送列表中的第一个节点新任务，只是由它启动任务。该节点成为Mother Superior，其他节点称为sister moms。

由三个服务组成 pbs\_server, pbs\_mom, pbs\_sched:

pbs\_server 是资源管理系统的服务器

pbs\_mom 是客户端资源监视器，负责监视各客户节点（计算节点）的资源使用情况

pbs\_sched 是任务调度器（torque 自带的这个调度器功能比较简单有限，经常不用它而搭配更强大的 maui)

## 安装

下载：

http://www.adaptivecomputing.com/support/download-center/torque-download/

1. **主节点编译安装**

[root@node0 ~]# yum install readline-devel tk-devel tcl-devel pam-devel libssl-devel libxml2-devel //所有节点

[dwt@node0 software]# tar zxvf torque-4.0.3.tar.gz

[dwt@node0 software]$ cd torque-4.0.3

[dwt@node0 torque-4.0.3]$ ./configure –prefix=/home/dwt/software/torque

[dwt@node0 torque-4.0.3]$ make

[root@node0 torque-4.0.3]$ make install

1. **计算节点安装**

如果源码是放在共享目录（如NFS）下，那么可以直接到计算节点上执行如下命令即可

make install\_mom install\_clients

1. **载入动态链接库**

[root@node0 torque-4.0.3]# echo '/usr/local/lib' > /etc/ld.so.conf.d/torque.conf //所有节点

或[root@node0 lib]# echo '/home/dwt/software/torque/lib' > /etc/ld.so.conf.d/torque.conf

[root@node0 torque-4.0.3]# ldconfig //所有节点

1. **配置pbs\_server**

可执行程序在 /usr/local ，配置文件在 /var/spool/torque，pbs\_server的配置在server\_priv/serverdb文件中，通过qmgr设置。

可通过两种方式配置serverdb：

1、./torque.setup username：调用pbs\_server –t create，同时配置用户作为操作员

2、pbs\_server –t create：只生成最简单的配置文件

每次serverdb修改后要重启pbs\_server。

[root@node0 torque-4.0.3]# ./torque.setup root

//或者通过手动增加所需队列

[root@node0 torque-4.0.3]# pbs\_server -t create

//另外增加一个队列

[root@node0 torque-4.0.3]# qmgr -c "set server scheduling=true"

[root@node0 torque-4.0.3]# qmgr -c "create queue batch queue\_type=execution"

[root@node0 torque-4.0.3]# qmgr -c "set queue batch started=true"

[root@node0 torque-4.0.3]# qmgr -c "set queue batch enabled=true"

[root@node0 torque-4.0.3]# qmgr -c "set queue batch resources\_default.nodes=1"

[root@node0 torque-4.0.3]# qmgr -c "set queue batch resources\_default.walltime=3600"

[root@node0 torque-4.0.3]# qmgr -c "set server default\_queue=batch"

[root@node0 server\_priv]# qmgr -c 'p s' //查看serverdb内容

1. **配置计算节点**

pbs\_server通过nodes文件与pbs\_mom通信

[root@node0 torque-4.0.3]# vi /var/spool/torque/server\_priv/nodes

node0 np=2 //主机名，处理器数

node1 np=2

1. **配置pbs\_mom**

[root@node0 ~]# vi /var/spool/torque/mom\_priv/config //所有节点

$pbsserver node0

$logevent 255

1. **配置服务器**

[root@node0 torque-4.0.3]# vi /var/spool/torque/server\_name

node0

1. **启动**

[root@node0 torque-4.0.3]# cd contrib/init.d

[root@node0 init.d]# cp pbs\_\* /etc/init.d/

[root@node0 init.d]# cp trqauthd /etc/init.d/

如果是安装在其他路径中，则需将所有脚本的默认路径/usr/local/sbin改成安装路径

[root@node0 ~]# qterm -t quick //停止pbs\_server

[root@node0 ~]# service trqauthd start （计算节点）

[root@node0 ~]# service pbs\_server start （主节点）

[root@node0 ~]# service pbs\_mom start （计算节点）

[root@node0 ~]# service pbs\_sched start （主节点）

[root@node0 server\_priv]# qnodes

如果相应计算节点上的 pbs\_mom 服务还没启动，则节点状态为 down，否则为free。

[root@node0 ~]# chkconfig pbs\_server on（主节点）

[root@node0 ~]# chkconfig pbs\_sched on（主节点)

[root@node0 ~]# chkconfig pbs\_mom on （计算节点）

1. **提交任务**

[dwt@node0 torque]$ echo "sleep 30" | qsub

|  |  |
| --- | --- |
| Usage | Description |
| > qsub -l nodes=12 | Request 12 nodes of any type |
| > qsub -l nodes=2:server+14 | Request 2 "server" nodes and 14 other nodes (a total of 16) - this specifies two node\_specs, "2:server" and "14" |
| > qsub -l nodes=server:hippi+10:noserver+3:bigmem:hippi | Request (a) 1 node that is a "server" and has a "hippi" interface, (b) 10 nodes that are not servers, and (c) 3 nodes that have a large amount of memory and have hippi |
| > qsub -l nodes=b2005+b1803+b1813 | Request 3 specific nodes by hostname |
| > qsub -l nodes=4:ppn=2 | Request 2 processors on each of four nodes |
| > qsub -l nodes=1:ppn=4 | Request 4 processors on one node |
| > qsub -l nodes=2:blue:ppn=2+red:ppn=3+b1014 | Request 2 processors on each of two blue nodes, three processors on one red node, and the compute node "b1014" |

1. **查看任务**

[dwt@node0 torque]$ qstat

1. **删除任务**

[dwt@node0 torque]$ qdel -m "hey! Stop abusing the NFS servers" 4807 //-m备注

1. **跟踪任务**

[dwt@node0 torque]$ tracejob 2

## 问题

1. 如果修改了 /var/spool/torque/mom\_priv/config 的设置，要重启 pbs\_mom 服务才有效。

**问题一：**

[dwt@node0 torque]$ qsub pbsjob

qsub: submit error (Unknown queue MSG=requested queue not found)

pbsjob中#PBS –q queuename是否在serverdb中存在相应队列名。

**问题二：**

qsub: submit error (Job exceeds queue resource limits MSG=cannot locate feasible nodes (nodes file is empty or all systems are busy))

原因1：/var/spool/torque/server\_priv/nodes文件再重启pbs\_server时被删除，需重新建立；

原因2：脚本文件中#PBS –l nodes=2:ppn=2 需求资源数大于nodes文件中提供的资源。

**问题三：**

Node为空闲，但提交任务一直在Q状态

搞了半天是4.0.2的bug，torque-4.0.3-snap.201207161713.tar.gz下载补丁包

[root@node0 torque]# qrun 2.node0 //强制其执行

**问题四：**

Cannot connect to default server host

检查server\_priv/nodes是否为空，重启所有服务，计算节点重启pbs\_mom

## 使用手册

### qsub

作业提交

$ qsub -t 0-2 pbsjob //提交批处理

Job-raidx=n //建立通信树的节点数

qsub -l nodes=12 //要求12个任意类型的节点

qsub -l nodes=2:server+14 //要求2个服务器节点和14个其他节点，共16个

qsub -l nodes=server:hippi+10:noserver+3:bigmem:hippi//一个具有hippi标志的server节点，10个非server节点和3个bigmem且hippi的节点

qsub -l nodes=b2005+b1803+b1813 //通过主机名指定

qsub -l nodes=4:ppn=2 //4个节点，每个节点跑4个进程

qsub -l nodes=1:ppn=4 //1个节点跑4个进程

qsub -l nodes=2:blue:ppn=2+red:ppn=3+b1014 //2个blue节点，每个节点跑2个进程，1个red节点跑3进程，1个叫b1014的节点

将nodes=后面的字符串截取出来，以+分隔成小字符串，看每个小字符串的属性，属性间以“：”分隔

qsub -l mem=200mb /home/user/script.sh //可用内存为200mb的节点

qsub -l nodes=node01,mem=200mb /home/user/script.sh //直到node01具有200mb可用内存时，由node01执行

qsub -l other=matlab /home/user/script.sh //具有matlab资源的其他节点上

设置作业抢占：

1、在mom\_priv/config中设置checkpoint\_script:

$pbsserver node06

$logevent 255

$restricted \*.mycluster.org

$checkpoint\_script /opt/moab/tools/mom-checkpoint.sh

2、在/src/include/pbs\_config.h中将MOM\_CHECKPOINT设为1

#define MOM\_CHECKPOINT 1

### pbsnodes

pbsnodes //查看节点状态

pbsnodes -o nodename //让节点offline，不被分配作业

pbsnodes -c nodename //让节点恢复服务

### qmgr

1. **节点**

qmgr -c "create node node003" //新增节点，在server\_priv/nodes中新增node003

qmgr -c "delete node node003" //删除节点

qmgr -c "set node node001 properties = bigmem" //设置node001属性为bigmem

qmgr -c "set node node001 properties += dualcore" //新增node001属性为dualcore

删除为-=

或者直接手动修改TORQUE\_HOME/server\_priv/nodes.

node001 bigmem dualcore

node002 np=4 bigmem matlab

1. **队列**

qmgr -c "create queue batch queue\_type=execution" //新建队列

qmgr -c "set queue batch acl\_groups=staff" //只有属于staff的才可使用batch队列

qmgr -c "set queue batch acl\_group\_enable=true" //只有为true时，只有acl\_groups中指明的组才可使用batch队列

qmgr -c "set queue batch acl\_group\_sloppy=true" //只有为true时，才检查用户的其他组中是否有staff，否则只检查所有组

qmgr -c "set queue batch acl\_hosts=h1+h2+h3" //增加能提交batch队列的主机名

qmgr -c "set queue batch acl\_logic\_enable=true" //只有为true时，acl\_hosts才生效

qmgr -c "set queue batch acl\_logic\_or=true" //为true时，用户或组满足条件就可使用batch队列，为false时，用户和组必须同时满足条件

qmgr -c "set queue batch acl\_users=john" //用户john可使用batch队列

qmgr -c "set queue batch acl\_user\_enable=true" //为true时，只有acl\_user中指明的用户才可使用batch队列

qmgr -c "set queue batch disallowed\_types = interactive" //指明那些任务不能提交，可选值有interactive,batch, rerunable, nonrerunable, fault\_tolerant (as of version 2.4.0 and later), fault\_intolerant (as of version2.4.0 and later), and job\_array (as of version 2.4.1 and later).

qmgr -c "set queue batch enabled=true" //队列是否接受任务提交

### qstat

查看状态

语法：qstat [-f][-W site\_specific] [job\_identifier... | destination...]

qstat -Q [-f][-W site\_specific] [destination...]

qstat -B [-f] [-W site\_specific] [server\_name...]

参数：destination可以为pool，@server，pool@server

作用：查看作业、结点池和批处理服务器的状态。命令格式一可以输出所指定作业ID或者结点池中所有作业的状态，命令格式二可以输出每个结点池的状态信息，命令格式三可以输出服务器的状态。

举例：

1. 显示已经配置的所有结点池状态信息。

qstat -q

2. 显示已经提交的作业状态信息

qstat -a

3. 显示指定作业的所有状态信息

$ qstat -f 23.console

mtime = Sun Apr 28 19:54:48 2002

Output\_Path = console:/home/zhangxq/cpi.o23

Priority = 0

qtime = Sun Apr 28 19:54:48 2002

Rerunable = True

Resource\_List.cput = 00:00:59

Resource\_List.nodect = 2

Resource\_List.nodes = 2:ppn=1

Variable\_List = LJRS\_O\_HOME=/home/zhangxq,LJRS\_O\_LANG=en\_US

……

4. 显示服务器的状态

$qstat –B

Server Max Tot Que Run Hld Wat Trn Ext Status

console 0 0 0 0 0 0 0 0 Active

### qselect

查询作业

语法：qselect [-a [op]date\_time] [-A account\_string] [-h old\_list] [-l resource\_list] [-N name] [-p [op]priority] [-q destination] [-r y|n] [-s states] [-u user\_list]

参数：op表示某一个作业属性值和选项参数值之间的关系。如

.eq. （等于）

.ne. （不等于）

.ge. （大于或等于）

.gt. （大于）

.le. （小于）

其它参数含义见qsub命令。

作用：列出符合选项要求的作业ID。这些作业来自于单个服务器。如果没有任何选项，该命令则列出该用户被授权的服务器上的所有作业。对那些普通用户来说，该命令只显示该用户所提交的作业。

举例：

1. 查询用户所提交的作业

$ qsub -q long cpi.ljrs

28.console

$ qselect

25.console

28.console

2. 查询指定结点池中的作业

$ qstat

Job id Name User TimeUse S Pool

25.console cpi zhangxq 0 Q dque

29.console cpi zhangxq 0 R long

$ qselect -P dque

25.console

3. 查询指定主机上的作业

$ qstat

Job id Name User TimeUse S Pool

25.console cpi zhangxq 0 Q dque

29.console cpi zhangxq 0 R long

$ qselect -P @console

25.console

29.console

4、挂起作业qhold

语法：qhold [-h hold\_list] job\_identifier …

作用：挂起批处理作业。挂起有三种类型：普通用户级‘u’、管理员级（操作员级）‘o’、系统级‘s’，缺省值为不挂起‘n’。用户只能在用户级别挂起自己提交的作业，操作员可以在用户级和操作员级挂起任何作业，系统管理员可以在任何级别上挂起任何作业。

在执行该命令时，如果作业在运行结点池里排队，那么作业将直接被挂起；如果作业处于运行状态，为了中断作业的执行，必须采取其他办法。如果被挂起作业的主机系统支持一致点检查或者重新启动，则挂起正在运行作业将引发以下操作：首先检查作业的一致性，然后释放该作业所占用的资源，最后该作业位于执行结点池中，处于挂起状态；如果被挂起作业的主机系统不支持一致点检查或者重新启动，则仅设置指定的挂起作业类型，而实际上并不能挂起，除非调用qrerun命令重新运行该作业时，该挂起请求才生效。

举例：

1. 使用普通用户身份挂起作业

$ qhold -h u 25.console $ qstat

Job id Name User TimeUse S Pool

25.console cpi zhangxq 0 H dque

2. 以操作员身份挂起作业

$ qhold -h o 25

qhold: Unauthorized Request 25.console

$ su root

$ qhold -h o 25

$ qstat

Job id Name User TimeUse S Pool

25.console cpi zhangxq 0 H dque

5、释放作业qrls

语法：qrls [-h hold\_list] job\_identifier ...

作用：释放被挂起的批处理作业。由于作业的挂起有三种类型：USER、OPERATOR和SYSTEM。所以，要释放不同类型的作业挂起，用户就必须具有相应的权限。缺省为USER级

举例：释放在普通用户和操作员级被挂起的作业。

$ qstat

Job id Name User TimeUse S Pool

25.console cpi zhangxq 0 H dque

$ qrls -h uo 25

$ qstat

Job id Name User TimeUse S Pool

25.console cpi zhangxq 0 Q dque

6、重新运行作业qrerun

语法：qrerun job\_identifier ...

作用：重新运行所指定的作业。

举例：如果作业允许别重新执行（缺省值是可以重新运行）。ROOT用户运行该命令，可以终止本次运行，把该作业放入其原来所在的结点池中，重新运行。

$ qrerun 27

qrerun: Unauthorized Request 27.console

$ su – root

$ qstat

Job id Name User TimeUse S Pool

25.console cpi zhangxq 0 Q dque

27.console cpi zhangxq 0 R dque

$ qrerun 27

$ qstat

Job id Name User TimeUse S Pool

25.console cpi zhangxq 0 Q dque

27.console cpi zhangxq 0 R dque

7、更改作业属性qalter

语法：qalter [-a date\_time] [-A account\_string] [-e path] [-h hold\_list] [-j join] [-k keep] [-l resource\_list] [-m mail\_options] [-M user\_list] [-N name] [-o path] [-p priority] [-r c] [-u user\_list] job\_identifier...

参数：各参数的含义见qsub命令。

作用：更改批处理作业的属性。主要修改所指定作业ID的相关属性（选项表中所列出的属性）。

举例：1. 更改批处理作业的运行时间属性。

$ qalter -a 0309251000 23.console

2. 更改给用户发送邮件的时间为作业中止和运行结束时

$ qalter –m ae 23.console

8、删除作业qdel

语法：qdel [-W delay|force] job\_identifier ...

参数：-W 当指定delay 时，表示在删除作业前需要等待的时间（秒）

当指定force时，强制删除该作业。

作用：删除批处理作业。按照命令行中所指定的作业ID的顺序来删除作业。

举例：$ qstat

Job id Name User TimeUse S Pool

22.console cpi zhangxq 0 Q long

$ qdel 22

9、给作业发送消息qmsg

语法：qmsg [-E] [-O] message\_string job\_identifier ...

参数：-E 将消息串写入错误输出文件

-O将消息串写入输出文件

作用：给批处理作业发送消息。该命令通过给作业的所有者（批处理服务器）发送消息，从而把消息写入作业的输出文件，也就是说，该命令并不是直接把消息写入作业的输出文件。

举例：给正在运行的作业发送消息，该消息被写入所指定作业的错误输出文件中。

$ qstat

Job id Name User TimeUse S Pool

25.console cpi zhangxq 0 Q dque

26.console cpi zhangxq 0 R dque

$ qmsg “The job is running” 26

$ more cpi.e26

The job is running

10、在结点池之间移动作业qmove

语法：qmove destination job\_identifier ...

参数：destination可以为结点池名

作用：把批处理作业移到其他结点池中去运行。即把作业从所在的结点池中删除，并放在其他结点池中。

举例：

$ qstat

Job id Name User TimeUse S Pool

22.console cpi zhangxq 0 Q dque

$ qmove long 22

$ qstat

Job id Name User TimeUse S Pool

22.console cpi zhangxq 0 Q long

11、给作业发送信号qsig

语法：qsig [-s signal] job\_identifier ...

参数：参数signal可以为信号名称，如SIGKILL,KILL,SIGNULL或者无符号整数，如9、0。

作用：给正在运行的批处理作业发送信号。如果不指定-S选项，则发送“SIGTERM”信号。如果有下列原因之一，如用户无权给作业发送信号、作业未处于运行状态和该信号请求对作业所运行的系统无效，则拒绝执行该命令请求。

12、查看和管理结点ljrsnodes

语法：ljrsnodes [- {c|o|r}] [nodename …] ljrsnodes - {a|l }]

参数：-a 列出所有结点及其属性，属性包括“state”和“properties”

-c 清除结点列表中的“offline”或“down”状态设置，使结点可以被分配给作业。

-l 以行的方式列出被标记的结点的状态 -o 将指定结点的状态标记为“offline”。这将帮助管理员暂时停止某些结点的服务。

-r 清除指定结点的“offline”状态

作用：该命令可以标记结点的状态为“offline”、“down”或“free”,也可以帮助用户查看结点信息。除了”-a” 和“-l”选项外，使用其他选项需要拥有操作员和管理员权限。

举例：1、查看所有结点信息

ljrsnodes -a

2、将结点标记为“offline”状态

ljrsnodes –o c0108

3、清除结点的“offline”状态设置

ljrsnodes –c c0108 或 ljrsnodes –r c0108

13、查看所有计算结点的状态shownodes

语法：shownodes [-h]

作用：按照结点池分类显示各结点池中结点的状态，结点状态以不同颜色显示。并在结点名后以数字代表可用CPU数和总共CPU数。例如c0101-2/2,表示c0101结点可用2CPU总共2CPU，c0102-1/2表示C0102可用1CPU总共2CPU。

举例：$shownodes –h

# LSF

Platform LSF是Platform Suite的一个组件。Platform Suite for SAS包括三个组件：Process Manager，LSF，Grid Management Services(简称GMS)。

LSF Suite管理负载的工具集，包括：

1. LSF Base：作为其他组件的基础，必须安装，包含了网络服务器，LSF API，lsmake，负载平衡工具。
2. LSF Batch：提供批处理作业工具集，保证资源优化分配。
3. LSF JobScheduler：分布式作业调度，将异构服务器整合成虚拟主机。
4. LSF MultiCluster：多集群间平衡负载，保证集群整体性。
5. LSF Analyzer：数据分析的图形工具，分析LSF Batch和LSF JobScheduler对资源的使用情况。
6. LSF Parallel：管理并行作业执行。
7. LSF Make：分布式并行Make。

LSF Suite分为LSF Enterprise Edition和LSF Standard Edtion两类。

所有组件都在同一包中，可指定安装组件。默认安装LSF Base和LSF Batch。LSF配置文件和LSF批处理工作目录必须安装在共享目录中（最好是NFS共享的home目录）。

## 安装

准备工作：/usr/share/lsf目录NFS共享，节点间能无密码的rsh互访

[root@node0 lsf]# useradd lsfadmin

[root@node0 lsf]# vi /etc/hosts //LSF根据该处查找是否存在对应主机名

192.168.0.62 node0

192.168.0.63 node1

//修改完/etc/hosts文件，必须重启系统，主机名才生效

[root@node0 software]# tar zxvf lsf8.0.1\_lsfinstall.tar.Z

将lsf8.0.1\_linux2.6-glibc2.3-x86\_64.tar.Z和license.dat放到与lsf8.0.1\_lsfinstall.tar.Z位于同一目录

[root@node0 software]# cd lsf8.0.1\_lsfinstall

[root@node0 lsf8.0.1\_lsfinstall]# vi install.config

LSF\_TOP="/usr/share/lsf" LSF安装路径

L LSF\_ADMINS="lsfadmin" 管理员帐号

LSF\_CLUSTER\_NAME="cluster1" 集群名称

LSF\_MASTER\_LIST="node0" Master服务器的主机名列表

其他都保留默认值

LSF\_LICENSE：license.dat所在目录，默认为当前目录的上层目录

ENABLE\_EGO：在LSF中使用Platform EGO，默认为N

ENABLE\_HPC\_INST：是否启用HPC支持，我们暂时不打开，选择N

[root@node0 lsf8.0.1\_lsfinstall]# vi ./slave.config //不用修改

[root@node0 lsf8.0.1\_lsfinstall]# ./lsfinstall -f install.config

Have you purchased LSF? (y/n) [n] 输入y回车

Do you accept the terms and conditions of the END USER SOFTWARE LICENSE AGREEMENT ? (y/n) [n] y 输入y回车

1) linux2.6-glibc2.3-x86\_64

Press 1 or Enter to install this host type: 找到当前目录下的该包，按1并回车确认

完成安装

[root@node0 ~]# cd /usr/share/lsf/8.0/install/

[root@node0 install]# ./hostsetup --top="/usr/share/lsf" --boot="y"

[lsfadmin@node0 install]# cd /usr/share/lsf/conf

[lsfadmin@node0 install]# . profile.lsf //配置LSF环境变量

[lsfadmin @node0 ~]# vi ~/.bash\_profile //添加到环境变量中

LSF\_TOP=/usr/share/lsf

source $LSF\_TOP/conf/profile.lsf

[lsfadmin@node0 ~]# lsfstartup

### 问题

**问题1： rsh not find**

which rsh //检查rsh是否安装

yum install rsh\*

**问题2：Start up LIM on <hostname> ...... hostname: Connection refused**

rsh服务没有启动，或者当前用户无法无密码的互相rsh，官网提供改用ssh实现的方式，本文仍实现节点间无密码rsh互访

**问题3：LSF\_STARTUP\_PATH not defined in /etc/lsf.sudoers**

由于没有/etc/lsf.sudoers文件，只有root能够启动和终止LSF，LSF管理员不行。参看启动配置章节。

**问题4：Host does not have a software license**

License文件有错或路径不对，原因可能如下：

1. License文件在Windows下编辑过导致结束符为^M；
2. FEATURE必须为1行，没有回车，DEMO license以FEATURE开始，DEMO结束；permanent license，以FEATURE开始，Platform结束；
3. DEMO license过期；
4. License server daemon（lmgrd）没有运行；

[lsfadmin@node0 conf]$ lmgrd -c /usr/share/lsf/conf/license.dat -l /usr/share/lsf/log/license.log

1. 安装了新license，但原来没许可的lim仍运行

[root@node0 software]# ps -ef | grep lim

[root@node0 software]# service lsf stop

1. FEATURE与PRODUCTS不符；

For example, if your license has the following FEATURE lines:

FEATURE lsf\_base lsf\_ld 6.000 1-jun-0000 10 DCF7C3D92A5471A12345 "Platform"

FEATURE lsf\_manager lsf\_ld 6.000 1-jun-0000 10 4CF7D37944B023A12345 "Platform"

the PRODUCTS line in LSF\_CONFDIR/lsf.cluster.cluster\_name must contain:

PRODUCTS=LSF\_Base LSF\_Manager

**问题5：license manager: can't initialize: No SERVER lines in license file.**

参看3.1.2节

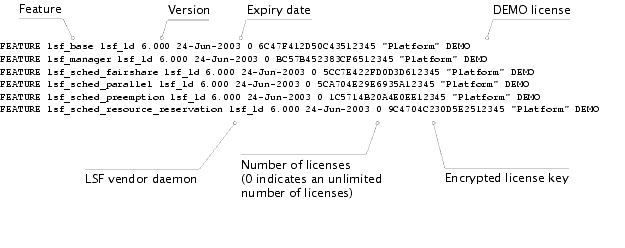
**问题6：Start up LIM on <node1> ...... Permission denied.**

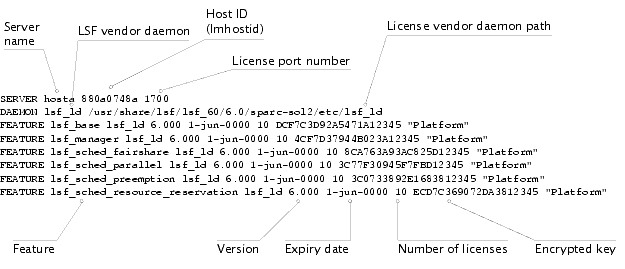
必须保证执行lsfstartup的用户能够无密码的在节点间rsh，因此要用lsfadmin，而不是root执行，因为root不支持redhat的无密码rsh。

### LSF License

LSF License分两种：Demo License和Permanent License。

LSF\_CONFDIR/lsf.conf的LSF\_LICENSE\_FILE指定License.dat的存放路径。





### 配置License

首先，用lmhostid 获得硬件的FLEXlm ID，[将ID发送license@platform.com](mailto:将ID发送license@platform.com)，获得license;

[lsfadmin@node0 ~]$ cd /usr/share/lsf/conf

[lsfadmin@node0 conf]$ lmhostid

lmhostid - Copyright (c) 1989-2006 Macrovision Europe Ltd. and/or Macrovision Corporation. All Rights Reserved.

The FLEXlm host ID of this machine is "000c29559c2d"

[lsfadmin@node0 conf]$ which lsf\_ld //FLEXlm license server必须要能找到该文件

/usr/share/lsf/8.0/linux2.6-glibc2.3-x86\_64/etc/lsf\_ld

[lsfadmin@node0 conf]$ vi license.dat

//增加以下2行，node0 为Server名，000c29559c2d为FLEXlm ID，1700为FLEXlmServer使用的TCP端口号，如果被占用，可改成其他端口

SERVER node0 000c29559c2d 1700

DAEMON lsf\_ld /usr/share/lsf/8.0/linux2.6-glibc2.3-x86\_64/etc/lsf\_ld

//以LSF 管理员登录Server（非root），运行lmgrd

[lsfadmin@node0 conf]$ lmgrd -c /usr/share/lsf/conf/license.dat -l /usr/share/lsf/log/license.log

[lsfadmin@node0 conf]# lsfrestart //由于root不能无密码rsh，因此采用lsfadmin普通用户，需保证各节点都useradd lsfadmin passwd lsfadmin lsfadmin能无密码rsh互访

Starting up all LIMs ...

Do you really want to start up LIM on all hosts ? [y/n]y

Start up LIM on <node0> ...... **done //此处不能为error**

Start up LIM on <node1> ...... **done**

Waiting for Master LIM to start up ..........................................//此次第一次启动会很久，第二次就快很多了

License.dat的FEATURE行与LSF\_CONFDIR/lsf.cluster.cluster\_name的PORDUCTS行相对应。例如：

License.dat的FEATURE如下：

FEATURE lsf\_base lsf\_ld 6.000 1-jun-0000 10 DCF7C3D92A5471A12345 "Platform"

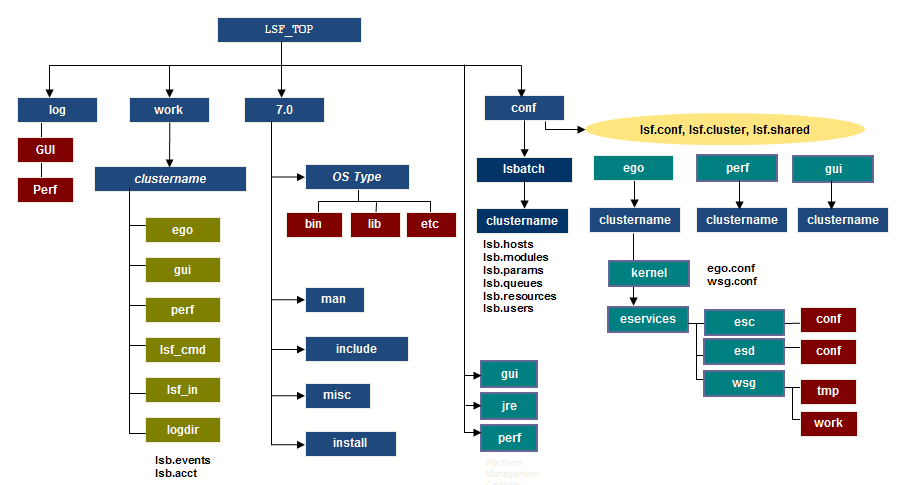
FEATURE lsf\_manager lsf\_ld 6.000 1-jun-0000 10 4CF7D37944B023A12345 "Platform"

则LSF\_CONFDIR/lsf.cluster.cluster\_name必须为：

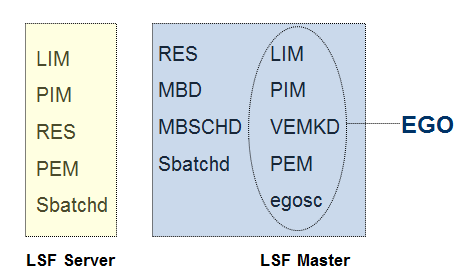
PRODUCTS=LSF\_Base LSF\_Manager

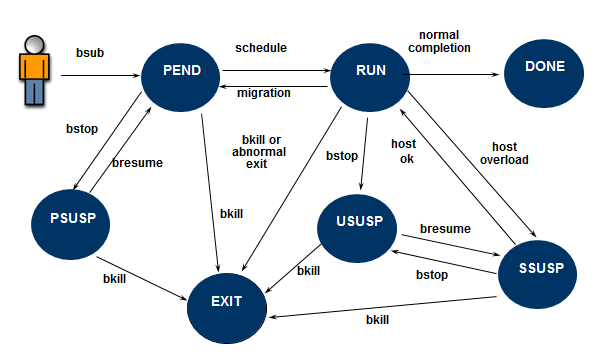
## 使用

LSF目录：



完整的LSF必须在主从节点启动如下deamon：





### lsfstartup

所有主机都启动LSF

### lsadmin/badmin

让新的License.dat生效用lsadmin和badmin。

lsadmin 控制lim和res；badmin控制mbatch和sbatchd。

**启动LSF： 启动deamon：**

% lsadmin limstartup LIM,PIM,VEMKD,PEM

% lsadmin resstartup RES

% badmin hstartup SBD[MBD,MBSCHD]

**启动远程LSF：**

% lsadmin limstartup host1 [host2…hostn]

% lsadmin resstartup host1 [host2…hostn]

% badmin hstartup host1 [host2…hostn]

**启动所有lsf.cluster文件中的主机：**

% lsadmin limstartup all

% lsadmin resstartup all

% badmin hstartup all

**停止LSF：**

# badmin hshutdown all

# lsadmin resshutdown all

# lsadmin limshutdown all

**重新配置LSF：**

To reload modified LSF configuration files and restart lim:

# lsadmin reconfig

To reload modified LSF Batch configuration files:

# badmin reconfig

To reload modified LSF Batch configuration files and restart mbatchd:

# badmin mbdrestart

**检查LSF配置：**

#lsadmin ckconfig –v

#badmin ckconfig -v

### lsid

检查LSF环境是否配置正确。

[lsfadmin@node0 conf]$ lsid

My cluster name is cluster1

My master name is node0

如果报Cannot open lsf.conf file，则LSF\_ENVDIR环境变量配置错误，检查profile.lsf。

### lsload

查看当前负载。同lshosts，lsinfo。

### lmreread

修改License.dat文件后用lmreread重新读入License.dat

# lmreread -c /usr/share/lsf/conf/license.dat

### lmstat

Lmstat检测License.dat状态，

# lmstat -a -c /usr/share/lsf/conf/license.dat

### bhosts

检查LSF Batch是否正常。

### bqueues

显示不同队列的状态和配置参数。如果队列能够接受和分发作业，则为Open:Active。

其显示内容保存在LSB\_CONFDIR/cluster\_name/configdir/lsb.queues中。

% bqueues -l normal

同bparams，bhist。

### bsub

提交批处理作业。

bsub sleep 60

bsub -q short sleep 60 //提交作业到名为short的队列中

bsub -q short -o /home/lsfadmin/output.%J -e /home/lsfadmin/errors.%J ls -l //-o结果输出到，-e错误输出到，%J是作业ID号

bsub -I ls //交互式作业

### bjobs

显示作业状态。

### bstop/bresume/bkill

bstop：暂停作业；bresume：重启作业；bkill：删除作业

举例：

[lsfadmin@node0 ~]$ bsub sleep 1000

Job <7> is submitted to default queue <normal>.

[lsfadmin@node0 ~]$ bjobs

JOBID USER STAT QUEUE FROM\_HOST EXEC\_HOST JOB\_NAME SUBMIT\_TIME

7 lsfadmi PEND normal node0 node1 sleep 1000 Oct 31 23:14

[lsfadmin@node0 ~]$ bstop 7

Job <7> is being stopped

[lsfadmin@node0 ~]$ bjobs

JOBID USER STAT QUEUE FROM\_HOST EXEC\_HOST JOB\_NAME SUBMIT\_TIME

7 lsfadmi **USUSP** normal node0 node1 sleep 1000 Oct 31 23:14

[lsfadmin@node0 ~]$ bresume 7

Job <7> is being resumed

[lsfadmin@node0 ~]$ bjobs

JOBID USER STAT QUEUE FROM\_HOST EXEC\_HOST JOB\_NAME SUBMIT\_TIME

7 lsfadmi RUN normal node0 node1 sleep 1000 Oct 31 23:14

[lsfadmin@node0 ~]$ bkill 7

Job <7> is being terminated

### lsrun/lsgrun

执行交互作业。

[lsfadmin@node0 ~]$ lsrun ls -l /usr/share/lsf/conf

[lsfadmin@node0 ~]$ lsrun -v -m node1 hostname //在node1上执行hostname命令

<<Execute hostname on remote host node1>>

node1

lsgrun与lsrun类似，但可以指定多个远程主机。

[lsfadmin@node0 ~]$ lsgrun -v -m "node0 node1" hostname

### 增加主机

1. 安装该主机类型的LSF二进制文件；by root

[root@node0 ~]# cd /usr/share/lsf/8.0/install

[root@node0 ~]# vi install.config //增加新主机

[root@node0 ~]#./lsfinstall -f install.config

1. 增加主机信息到lsf.cluster.cluster\_name文件中；by lsfadmin

[lsfadmin@node0 ~]$ cd /usr/share/lsf/conf/

[lsfadmin@node0 conf]$ vi lsf.cluster.cluster1

//在Host部分增加新主机信息，如果为!则自动检测，server字段如果为1则是LSF SERVER，0则LSF Client-only。

Begin Host

HOSTNAME model type server r1m mem swp RESOURCES

node0 ! ! 1 3.5 () () (mg)

node1 ! ! 1 3.5 () () (mg)

End Host

[lsfadmin@node0 conf]$ lsadmin reconfig

[lsfadmin@node0 conf]$ badmin reconfig

1. 配置新主机。 By root

[root@node0 ~]# cd /usr/share/lsf/8.0/install

[root@node0 ~]#./hostsetup --top="/usr/share/lsf/ " --boot="y"

[root@node0 ~]# lsadmin limstartup

[root@node0 ~]# lsadmin resstartup

[root@node0 ~]# badmin hstartup

[root@node0 ~]# bhosts //查看是否成功

[root@node0 ~]# lshosts

### 删除主机

删除Master主机前，先修改lsf.cluster.cluster\_name文件，重设一个默认Master主机，再删除原先的Master主机。

[root@node0 ~]# badmin hclose

[root@node0 ~]# lsfshutdown

[root@node0 ~]# vi /usr/share/lsf/conf/lsf.cluster.cluster1

删除Host部分中相应主机信息

删除所有与该主机相关的信息，如：

* LSF\_CONFDIR/lsf.shared
* LSB\_CONFDIR/cluster\_name/configdir/lsb.hosts
* LSB\_CONFDIR/cluster\_name/configdir/lsb.queues

登录集群中的其他主机，重新配置lim

[root@node1 ~]# lsadmin reconfig

[root@node1 ~]# badmin reconfig

### 增加队列

[lsfadmin@node0 ~]$ cd /usr/share/lsf/conf/lsbatch/cluster1/configdir

[lsfadmin@node0 configdir]$ vi lsb.queues

任选一队列拷贝过来，修改QUEUE\_NAME和其他参数

[lsfadmin@node0 configdir]$ badmin ckconfig

### 删除队列

[lsfadmin@node0 ~]$ bqueues

查看队列,如果待删除队列中有其他作业,则将作业转到其他队列,再删除队列；否则作业会移到lost\_and\_found队列中。

[lsfadmin@node0 ~]$ badmin qclose night //night队列停止接收作业

[lsfadmin@node0 ~]$ bjobs –u all – q night //查看night队列的所有作业

[lsfadmin@node0 ~]$ bswitch –q night idle 0 //将night队列的所有作业移到idle中，0指所有

[lsfadmin@node0 ~]$ vi /usr/share/lsf/conf/lsbatch/cluster1/configdir/ lsb.queues //删除night

[lsfadmin@node0 configdir]$ badmin ckconfig

### 启动配置

为让LSF管理员能够启动和终止LSF，需配置/etc/lsf.sudoers。否则只有root能启动和终止LSF。

以root用户登录，所有节点增加/etc/lsf.sudoers文件

[root@node0 ~]# vi /etc/lsf.sudoers

LSF\_STARTUP\_USERS="lsfadmin dwt"

LSF\_STARTUP\_PATH=/usr/share/lsf/8.0/linux2.6-glibc2.3-x86\_64/etc/

[root@node0 ~]# chmod 600 /etc/lsf.sudoers

[root@node0 ~]# lsfrestart

### 动态资源

ELIM：External Load Information Manager

[lsfadmin@node0 ~]$ vi /usr/share/lsf/conf/lsf.shared

//定义2个动态资源，license1，license2

Begin Resource

RESOURCENAME TYPE INTERVAL INCREASING RELEASE DESCRIPTION # Keywords

license1 Numeric 30 N Y (license1 resource)

license2 Numeric 30 N Y (license2 resource)

End Resource

Type：资源类型，可为Numberic,Boolean,String。

INTERVAL：ELIM更新共享资源的周期。

INCREASING：N指许可资源减少。

RELEASE：Y指当作业挂起时，释放许可资源。

[lsfadmin@node0 ~]$ vi /usr/share/lsf/conf/lsf.cluster.cluster1

//配置license1，license2与主机之间的映射，all指所有主机共享

Begin ResourceMap

RESOURCENAME LOCATION

license1 [all]

license1 [all]

End ResourceMap

ELIM输出可用资源的格式：

number\_indexes [index\_name index\_value] ...

2 license1 3 license2 2

ELIM必须是可执行文件，且位于LSF\_SERVERDIR目录下。Lim每次启动时会调用elim，获得输出。

# HPC

在IBM System x系列的硬件环境中,IBM Platform Computing的解决方案具备良好定义的文件部署模型，易于构建动态的云基础设施，以及大型的HPC应用部署、集群管理和网格应用。

IBM Platform Computing主要包括五大产品，分别为IBM Platform LSF、IBM Platform HPC、IBM Platform Symphony、IBM Platform MPI和IBM Platform Cluster Manager。其中Platform LSF和Platform HPC并称为高性能计算平台。

IBM Platform HPC提供完整的集群管理功能，从集群部署管理到工作负载管理监控，且易于安装和使用，便于集群实现快速高效的部署，包括传统HPC应用程序和其他应用。

与以往需要安装整合多重软件包不同，IBM Platform HPC提供了统一的管理功能集，可简化应用程序集成过程，使得最终用户可以专注于开发应用程序而不是集群的管理。它具有一个统一的Web门户网站，管理员和用户均可通过其来访问和管理集群中的资源。

<http://server.zdnet.com.cn/server/2013/0315/2148773.shtml>

ANSYS Fluent MSC NASTRAN BLAST LS-DYNA HOMEGROWN APP UNIFIED WEB - BASED INTERFACE
CLUSTER MANAGEMENT MPI LIBRARY OS MULTI-BOOT WORKLOAD MANAGMENT GPU　SCHEDULING APP INTEGRATION 

## 安装

主节点最小配置：

2.5GB内存；80GB硬盘，/depot分区最少12GB，/var分区最少40GB；DVD光驱； x86硬件；配置好网关，DNS IP。

### 准备工作

[root@sps-pm conf]# authconfig --enablemd5 --update //采用MD5身份验证，然后用passwd命令修改所有用户的密码

[root@sps-pm conf]# rpm -e python-sqlalchemy --nodeps //卸载python-sql

[root@sps-pm software]# rpm -e openais-devel --nodeps //卸载openais-devel

[root@sps-pm software]# ll -d /home //home目录可写

[root@sps-pm conf]# /etc/init.d/NetworkManager stop

[root@ sps-pm conf]# chkconfig NetworkManager off

[root@sps-pm conf]# vi /etc/selinux/config

SELINUX=disabled

[root@sps-pm ~]# vi /etc/yum.repos.d/rhel-source.repo //设置Redhat ISO yum

1. **准备HPC**

如果用ISO安装：

[root@sps-pm software]# mount -o loop /home/lsfadmin/software/hpc-3.1-7211.rhel.iso /mnt

[root@sps-pm network-scripts]# vi /etc/fstab

/home/lsfadmin/software/hpc-3.1-7211.rhel.iso /mnt iso9660 loop,ro,auto 0 0

[root@sps-pm network-scripts]# umount /mnt

[root@sps-pm network-scripts]# mount -a

[root@sps-pm network-scripts]# df -h

1. **检测系统**

用ISO安装：

[root@sps-pm ~]# /mnt/pcm-installer //不能在挂载目录mnt下运行pcm-installer

用DVD安装，DVD自动挂载到/media/HPC-3.1目录：

[root@sps-pm ~]# /media/HPC-3.1/pcm-installer

检测硬件和软件配置，将缺失的包安装上，直至显示OK

### 检查问题

**问题1: Problem with PCM installation because of error: Failed dependencies:createrepo is needed by kusu-setup-2.2-1.x86\_64**

[root@localhost software]# rpm -ivh deltarpm-3.5-0.5.20090913git.el6.x86\_64.rpm

[root@localhost software]# rpm -ivh python-deltarpm-3.5-0.5.20090913git.el6.x86\_64.rpm

[root@localhost software]# rpm -ivh createrepo-0.9.8-5.el6.noarch.rpm

[root@sps-pm ~]# yum install createrepo

**问题2: Not enough statically configured network interfaces.**

**At least 2 statically configured network interface required.**

HPC要2个网络接口（配上静态IP），eth0配私网IP ，eth1配公网IP。



[root@sps-pm ~]# vi /etc/sysconfig/network-scripts/ifcfg-eth0

DEVICE="eth0"

BOOTPROTO=none

NM\_CONTROLLED="yes"

ONBOOT=yes //要将ONBOOT设为yes，再关闭NetworkManager，重启network服务

TYPE="Ethernet"

UUID="48e23432-c18a-43af-9d4d-459678a90339"

IPADDR=192.168.0.62

PREFIX=24

GATEWAY=192.168.0.254

DNS1=124.16.136.254

DNS2=159.226.5.56

DEFROUTE=yes

IPV4\_FAILURE\_FATAL=yes

IPV6INIT=no

NAME=eth0

HWADDR=40:61:86:86:AB:7E

LAST\_CONNECT=1348816183

[root@sps-pm ~]# cp /etc/sysconfig/network-scripts/ifcfg-eth0 /etc/sysconfig/network-scripts/ifcfg-eth1

[root@sps-pm ~]# vi /etc/sysconfig/network-scripts/ifcfg-eth1

DEVICE="eth1"

BOOTPROTO=none

NM\_CONTROLLED="yes"

ONBOOT=yes

TYPE="Ethernet"

UUID="48e23432-c18a-43af-9d4d-459678a90339"

IPADDR=192.168.100.1

PREFIX=24

DEFROUTE=yes

IPV4\_FAILURE\_FATAL=yes

IPV6INIT=no

NAME=eth1

HWADDR=00:05:5D:E8:59:75

LAST\_CONNECT=1348816183

[root@sps-pm ~]# service NetworkManager stop

[root@sps-pm ~]# service network restart

[root@sps-pm ~]#ifconfig //检测到eth0，eth1两个接口

**问题3：the current hostname is invalid.**

**Please enter a valid FQDN hostname (e.g. headnode.noa):**

FQDN(fully qualified domain name, for example: master.domain.com)

[root@headnode ~]# vi /etc/hosts

192.168.0.63 headnode.nao

192.168.100.1 public. headnode.nao

[root@headnode ~]# reboot

**问题4： Invalid password encryption found for users: 'lsfadmin', 'root', 'wangxq'.**

**Passwords must be re-generated using the current encryption algorithm setting. Reset the passwords for these accounts using the 'passwd' command**

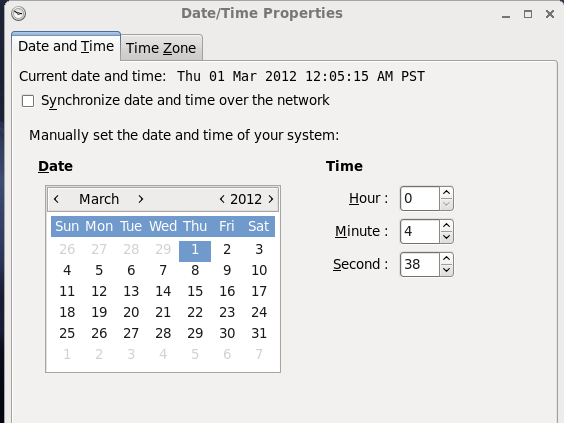
[root@headnode ~]# authconfig --enablemd5 –update

[root@headnode ~]# passwd //所有用户重新设置密码

**问题5：license过期**

1. 通过X界面修改系统时间

System->Administration->Date and Time，去掉Synchronize date and time over the network选项，然后设置时间，会同时修改系统和硬件时间



1. 虚拟机可在启动前修改宿主机的时间，实现修改系统时间；实体机如果联网会自动NTP更新系统时间，重设时间后拔网线，重启；
2. 通过命令行修改系统时间

[root@headnode ~]# date -s “20120301”

[root@headnode ~]# hwclock –systohc //同步系统时间到硬件时间systohc：SYStem clock to Hardware Clock

[root@headnode ~]# clock –w //写入CMOS

[root@headnode ~]# hwclock –r //查看硬件时间

1. 开机后立刻设置时间date –s ‘20120301’，然后再登录localhost:8080网站，就可以逃过license检测。

**问题6：重新安装LSF**

将/etc/hpc\_full\_install文件删除

### 主节点配置

Pcm-installer检测都为OK后，配置Master node：

Select the interface to be used for provisioning [1]: //回车eth0，内网

Select the interface to be used for public [1]: //回车eth1，外网

Specify private cluster domain [private.dns.zone]: //回车，集群域名为private.dns.zone

Do you want to setup HPC HA now? (Y/N) [N]: //回车，不启动HA

Select the mount point to be used [1]: //回车，Kusu放置/depot的挂载点

Select the media to install the Operating System from

1) DVD drive

2) ISO image or mount point

[1] >>2 //如果光盘放入DVD中，则回车，自动检测并拷贝；如果已将iso拷贝到系统中，则选2，回车

Provide the path to the ISO image or mount point: /home/lsfadmin/software/rhel-server-6.3-x86\_64-dvd.iso //提供iso的绝对路径

Choose one of the following actions:

1) List installed kits

2) Delete installed kits

3) Add extra kits

4) Continue

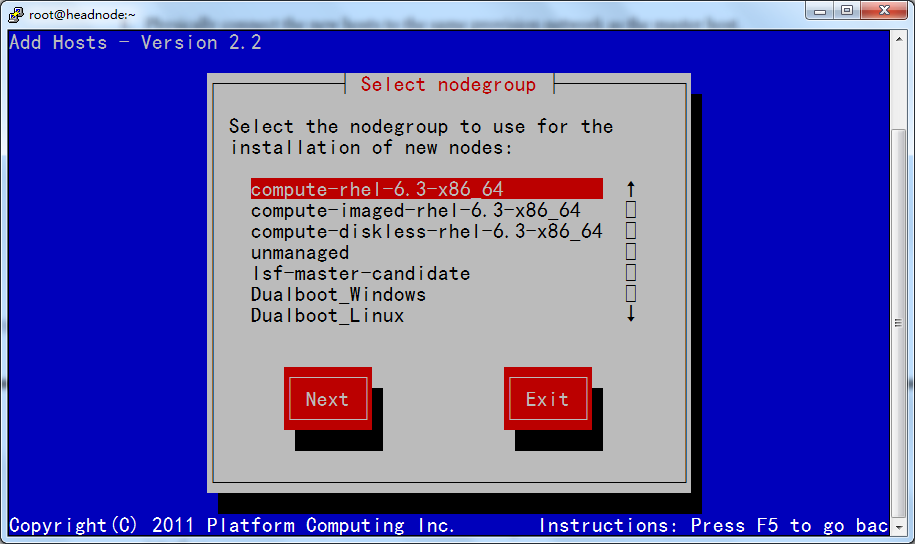
[4] >> //如果DVD iso只有1个文件，回车

Specify the full path to the Platform HPC license file. /home/lsfadmin/software/license.dat //回车

[root@headnode ~]# source /opt/kusu/bin/pcmenv.sh

### 计算节点配置

[root@headnode ~]# kusu-addhost



选择配置模板compute-rhel-6.3-x86\_64，Next

选择监听新主机的网络，eth0，Next

一旦kusu-addhost监听网络eth0，就可启动计算节点。计算节点启动时，需保证HPC安装DVD已从光驱中弹出，否则安装程序会运行找到的media。计算节点必须以PXE方式启动，如果不是，F2进入BIOS修改启动顺序。

如果配置正确，计算节点启动时会下载initrd(initial ram disk)，安装HPC。Kusu-addhost会报告检测到的新主机和安装过程。

输入localhost:8080登陆HPC网页

### 配置网页

1. **安装Flash显示Performance图表**
2. **安装Java JRE显示Java Applet**

下载jre-7u17-linux-x64.rpm

[root@headnode ~]# rpm –ivh jre-7u17-linux-x64.rpm

[root@headnode ~]# cd /usr/lib64/mozilla/plugins

[root@headnode plugins]# ln –s /usr/java/jre1.7.0\_17/lib/amd64/libnpjp2.so

Firefox输入about:plugins查看是否安装java jre成功，Java(TM) Plug-in 1.7.0\_17

#### VNC配置

<http://www.tightvnc.com/download/1.3.10/tightvnc-1.3.10_javabin.tar.gz>

[root@headnode ~]# mkdir tightvnc

[root@headnode ~]# tar -zxvf tightvnc-1.3.10\_javabin.tar.gz –C tightvnc

[root@headnode ~]# keytool -genkey -alias vnc -keystore platform.jks -keyalg rsa -storetype PKCS12 -validity 3650

生成platform.jks文件，将该文件放到与VncViewer.jar同目录中，进入该目录

[root@headnode ~]# jarsigner -verbose -keystore platform.jks -storetype PKCS12 -signedjar sVncViewer.jar VncViewer.jar vnc

生成sVncViewer.jar文件

jarsigner在jdk的bin目录下，redhat自带OpenJDK缺少jarsigner包，需卸载后重新安装

#### 安装JDK

[root@headnode ~]# java –version

[root@headnode ~]# yum remove java

[root@headnode ~]# rpm –ivh jdk-7u17-linux-x64.rpm

[root@headnode ~]# vi ~/.bash\_profile

export JAVA\_HOME=/usr/java/jdk1.7.0\_17

export PATH=$PATH:$HOME/bin:$PATH

export CLASSPATH=$JAVA\_HOME/lib:$CLASSPATH