

Speed up your research:

How to get 40 computers to do your work for you

Bingbing Yuan

June 19, 2008

- **barra: 4 GB RAM**
- **LSF (Load Sharing Facility) Cluster**
 - 36 machines (+ 42 lab specific machines)
 - **34: 4 GB RAM per machine**
 - **2: 8 GB RAM per machine**

User Jobs submitted from barra

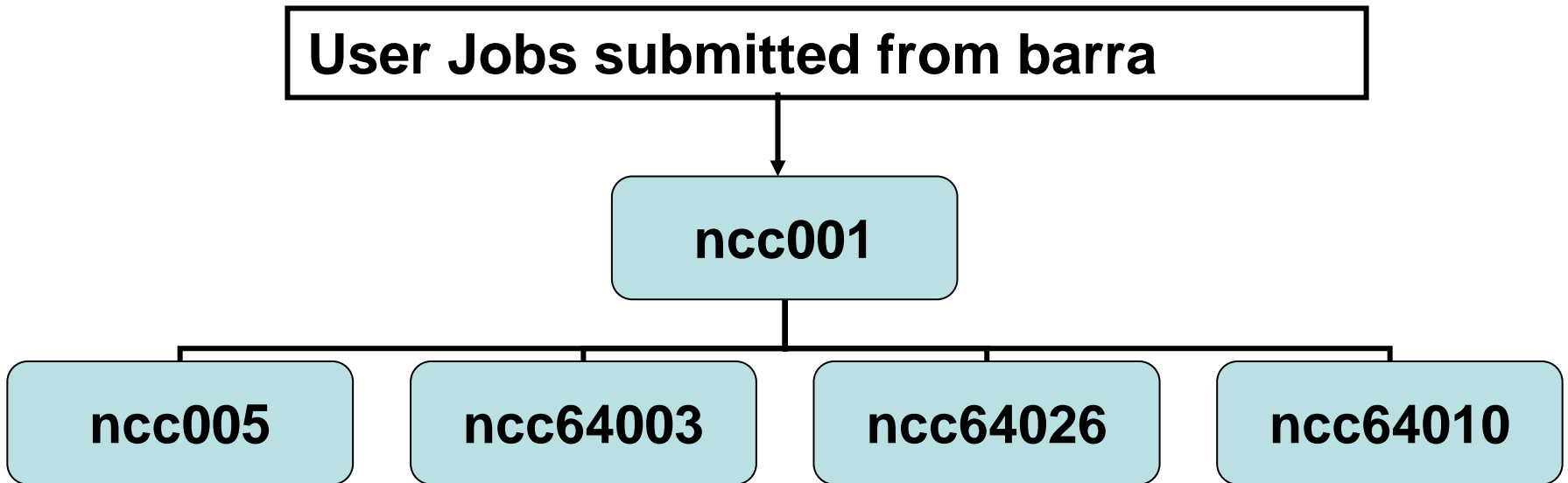
ncc001

ncc005

ncc64003

ncc64026

ncc64010



bsub – submit jobs

- `bsub myscript`
- Send notification to specified email
 - `bsub -u address@yahoo.com myscript`
- Send error and standard output to files
 - `bsub -e error_file -o std_file myscript`
 - `bsub -e error_file -o std_file "myscript >result"`
- Send job with specific queue
 - `bsub -q sq32hp myscript`
- Send job to a host
 - `bsub -m ncc64022 myscript`

Check the job status

- **bjobs**: pending, running and suspending jobs

```
byuan@barr:~$ bjobs -u bellott
```

JOBID	USER	STAT	QUEUE	FROM_HOST	EXEC_HOST	JOB_NAME	SUBMIT_TIME
123856	bellott	RUN	lq64lp	barra.wi.mi	ncc64015.wi	*allcam.m8	Jun 18 09:16
123857	bellott	RUN	lq64lp	barra.wi.mi	ncc64026.wi	*allcam.m8	Jun 18 09:16
123859	bellott	RUN	normal	barra.wi.mi	ncc64006.wi	*allcam.m8	Jun 18 09:17
123855	bellott	PEND	normal	barra.wi.mi		*08_b.temp	Jun 18 09:08
123860	bellott	PEND	normal	barra.wi.mi		*8_b2.temp	Jun 18 09:33
123861	bellott	PEND	normal	barra.wi.mi		*8_b3.temp	Jun 18 09:33
123862	bellott	PEND	normal	barra.wi.mi		*8_dn.temp	Jun 18 09:33

bjobs

Show all the running jobs

```
byuan@barra: % bjobs -u all -r
```

JOBID	USER	STAT	QUEUE	FROM_HOST	EXEC_HOST	JOB_NAME	SUBMIT_TIME
120636	jhughes	UNKWN	lq64lp	barra.wi.mi	ncc64025.wi	*cut.fa.m8	Jun 11 08:44
123856	bellott	RUN	lq64lp	barra.wi.mi	ncc64015.wi	*allcam.m8	Jun 18 09:16
123913	byuan	RUN	lq64lp	barra.wi.mi	ncc64004.wi	*1.output1	Jun 18 13:36
123917	byuan	RUN	lq64lp	barra.wi.mi	ncc64023.wi	*2.output2	Jun 18 13:36
123918	byuan	RUN	lq64lp	barra.wi.mi	ncc64013.wi	*3.output3	Jun 18 13:36
123919	byuan	RUN	lq64lp	barra.wi.mi	ncc64014.wi	*4.output4	Jun 18 13:36
123924	byuan	RUN	lq64lp	barra.wi.mi	ncc64012.wi	*9.output9	Jun 18 13:36
123925	hguo	RUN	bartel	barra.wi.mi	ncc64bartel	*Btrial.py	Jun 18 13:38
123859	bellott	RUN	normal	barra.wi.mi	ncc64006.wi	*allcam.m8	Jun 18 09:17
123912	gurdzie	RUN	normal	barra.wi.mi	ncc64008.wi	* gfclient	Jun 18 13:36

bjobs

- also show finished jobs: -a

```
byuan@barra:/nfs/BarC/Smed_chip_v2/filter[193]% bjobs -a -u nspies
```

JOBID	USER	STAT	QUEUE	FROM_HOST	EXEC_HOST	JOB_NAME	SUBMIT_TIME
124059	nspies	RUN	bartel-bla	barra.wi.mi	bartelblade	*sities.py	Jun 18 14:40
124155	nspies	RUN	bartel-bla	barra.wi.mi	bartelblade	*p.out.txt	Jun 18 16:47
124074	nspies	EXIT	bartel-bla	barra.wi.mi	-	*sities.py	Jun 18 15:55
124087	nspies	DONE	bartel-bla	barra.wi.mi	bartelblade	*p.out.txt	Jun 18 15:55
124137	nspies	EXIT	bartel-bla	barra.wi.mi	bartelblade	*p.out.txt	Jun 18 16:32
124138	nspies	EXIT	bartel-bla	barra.wi.mi	bartelblade	*p.out.txt	Jun 18 16:33
124139	nspies	EXIT	bartel-bla	barra.wi.mi	-	*p.out.txt	Jun 18 16:34
124140	nspies	EXIT	bartel-bla	barra.wi.mi	bartelblade	*p.out.txt	Jun 18 16:34
124141	nspies	EXIT	bartel-bla	barra.wi.mi	bartelblade	*p.out.txt	Jun 18 16:35
124142	nspies	EXIT	bartel-bla	barra.wi.mi	bartelblade	*p.out.txt	Jun 18 16:42

bkill – kill jobs

- bkill JOBID
 - bkill 124047
- Kill all jobs
 - bkill 0
- kill all jobs running as ‘normal’ queue
 - bkill -q normal 0

- **bpeek** – peek at the stdout and stderr output of unfinished job

- **bpeek** JOBID

- **bpeek 124047**

- **bstop** - suspends unfinished jobs

- **bstop 124047**

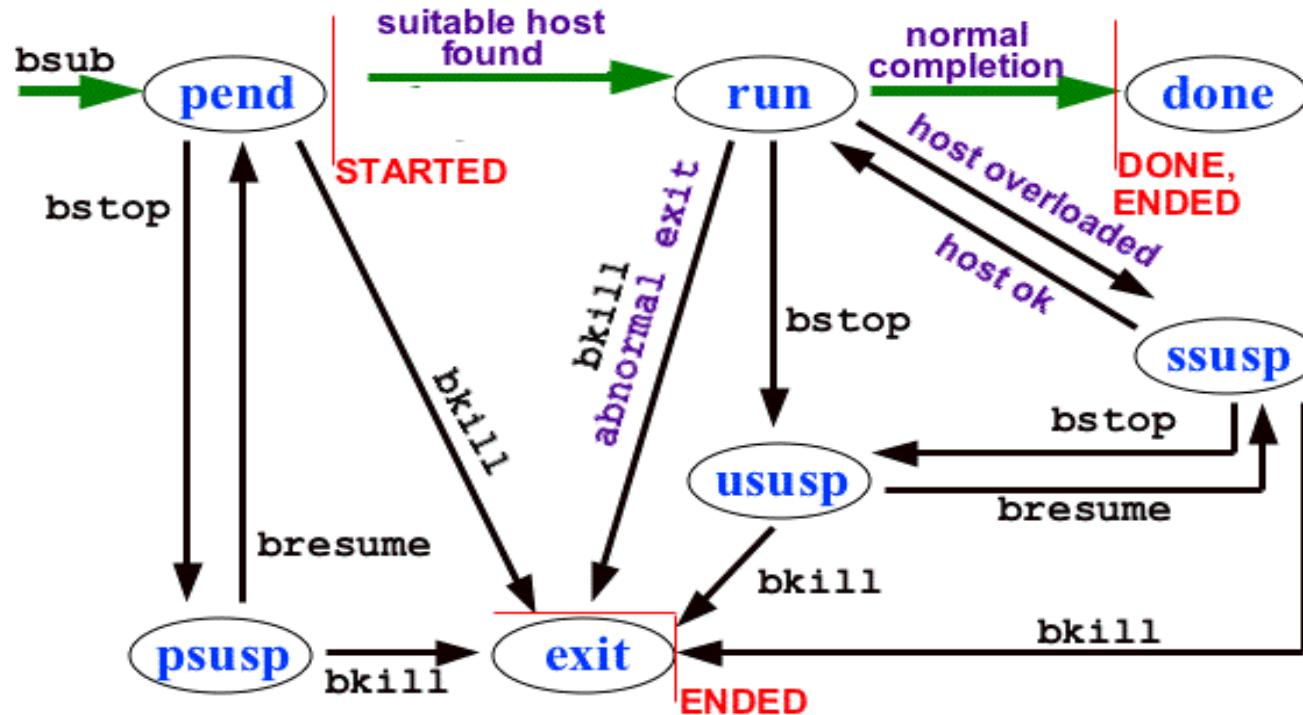
```
JOBID  USER  STAT  QUEUE      FROM_HOST  EXEC_HOST  JOB_NAME  SUBMIT_TIME
124047  byuan  USUSP lq64lp     barra.wi.mi ncc64014.wi *1.output1 Jun 18 14:35
```

- **bresume** - resumes suspended jobs

- **bresume 124047**

```
JOBID  USER  STAT  QUEUE      FROM_HOST  EXEC_HOST  JOB_NAME  SUBMIT_TIME
124047  byuan  RUN   lq64lp     barra.wi.mi ncc64014.wi *1.output1 Jun 18 14:35
```


LSF Job States and Events



LSF commands are shown in **black**;
Job states are shown in **blue**;
Job events are shown in **red**;
host resource information is shown in **purple**.

LSF selects which job to run next based on:

- Resources requirements of the applications
 - **queue**
 - job requirement
- Current load conditions
- How important you are

bqueues -- queue

```
byuan@barra: % bqueues -u byuan
```

QUEUE NAME	PRIO	STATUS	MAX	JL/U	JL/P	JL/H	NJOBS	PEND	RUN	SUSP
sq32hp	50	Open:Active	-	-	-	-	0	0	0	0
sq64hp	50	Open:Active	-	-	-	-	0	0	0	0
lq32hp	50	Open:Active	-	-	-	-	0	0	0	0
lq64hp	50	Open:Active	-	-	-	-	0	0	0	0
sq32mp	40	Open:Active	-	-	-	-	0	0	0	0
lq32mp	40	Open:Active	-	-	-	-	0	0	0	0
lq64lp	30	Open:Active	-	-	-	-	4	0	4	0
sq32lp	20	Open:Active	-	-	-	-	0	0	0	0
young	20	Open:Active	-	-	-	-	0	0	0	0
gf_serv	20	Open:Active	-	-	-	-	0	0	0	0
qiqo	20	Open:Active	-	-	-	-	0	0	0	0
solexa_test	20	Open:Active	-	-	-	-	0	0	0	0
normal	10	Open:Active	-	-	-	-	6	4	2	0

Total
JobSlots

JobSlots
Pending

JobSlots
Running

JobSlots
suspended

queues in the cluster

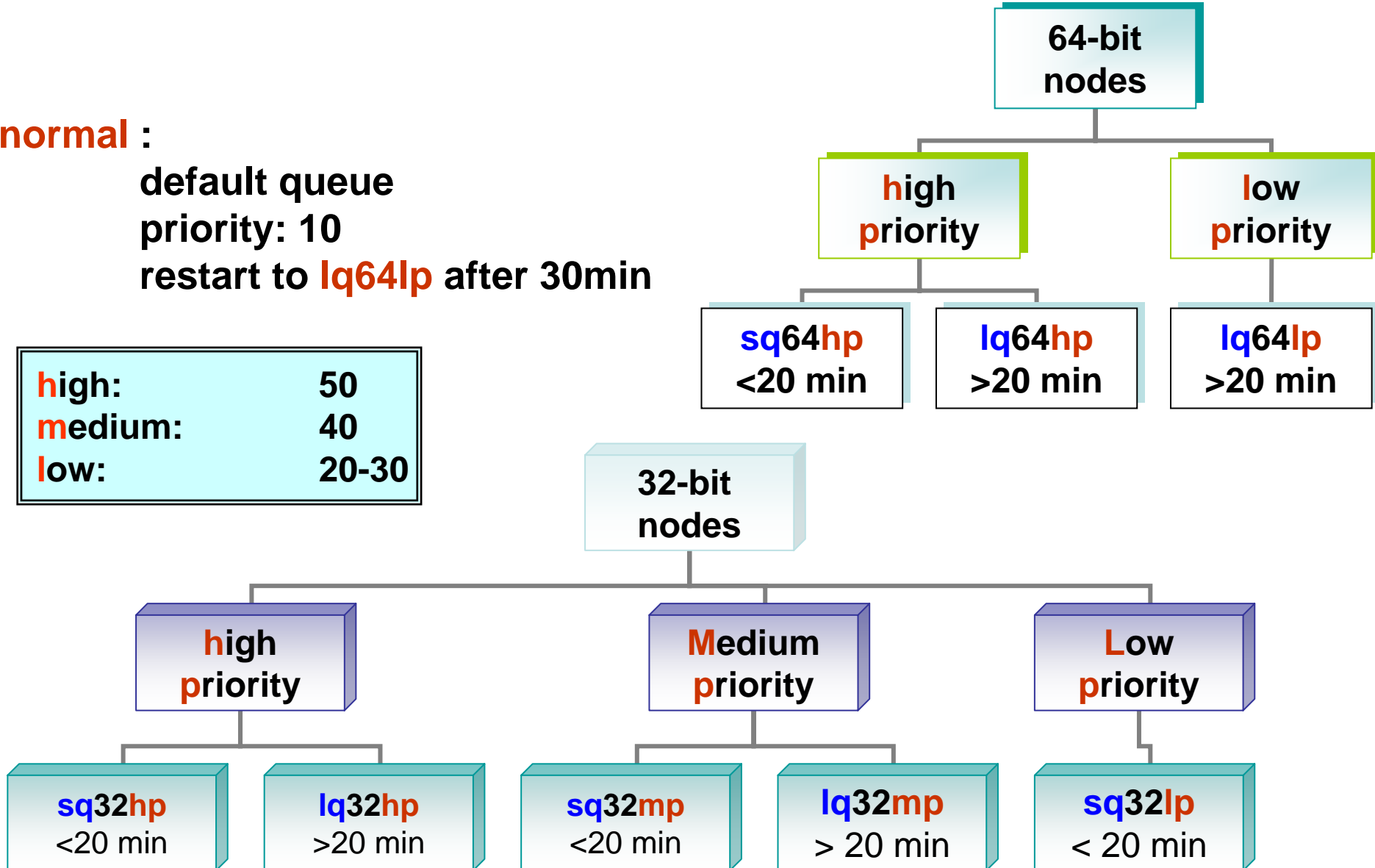
normal :

default queue

priority: 10

restart to **lq64lp** after 30min

high:	50
medium:	40
low:	20-30



Only available on 32-bit nodes

- RepeatMasker
 - Mask repetitive DNA
- EMBOSS applications
 - A suite for sequence analysis
 - <http://iona.wi.mit.edu/bio/tools/emboss/>

LSF selects which job to run next based on:

- Resources requirements of the applications

- queue: `bsub -q sq32hp myscript` ✓

- **job requirement**

- Current load conditions
- How important you are

Standard out from previous job

```
.....  
Successfully completed.  
  
Resource usage summary:  
  
CPU time      :      56.52 sec.  
Max Memory    :      735 MB  
Max Swap      :      752 MB  
  
Max Processes :          3  
Max Threads   :          3  
.....
```

Request 1G of memory

– **bsub -R "rusage[mem=1000]" myscript**

Ishosts – static resource information for the machines

CPU factor

number of processors

```
byuan@barra:~[19] lshosts
HOST_NAME      type      model    cpuf  ncpus  maxmem  maxswp  server  RESOURCES
ncc001.wi.m    LINUX86  Opteron8 60.0   2    3869M  4099M   Yes    ()
ncc002.wi.m    UNKNOWN  UNKNOWN_  1.0   -      -      -      Yes    ()
ncc003.wi.m    LINUX86  Opteron8 60.0   2    3869M  4099M   Yes    ()
ncc004.wi.m    LINUX86  Opteron8 60.0   2    3869M  4099M   Yes    ()
ncc005.wi.m    LINUX86  Opteron8 60.0   2    3869M  4099M   Yes    ()
ncc006.wi.m    LINUX86  Opteron8 60.0   2    3869M  4099M   Yes    ()
ncc64001.wi    UNKNOWN  UNKNOWN_  1.0   -      -      -      Yes    ()
ncc64002.wi    UNKNOWN  UNKNOWN_  1.0   -      -      -      Yes    ()
ncc64003.wi    LINUX86  Opteron8 60.0   2    8001M  4102M   Yes    ()
ncc64004.wi    LINUX86  Opteron8 60.0   2    8001M  4102M   Yes    ()
ncc64005.wi    LINUX86  Opteron8 60.0   2    3899M  4099M   Yes    ()
ncc64006.wi    LINUX86  Opteron8 60.0   2    3961M  4099M   Yes    ()
```


LSF selects which job to run next based on:

- Resources requirements of the applications
 - queue: `bsub -q sq32hp myscript` ✓
 - job requirement ✓
- **Current load conditions**
- How important you are

Isload- current dynamic load activity

byuan@barra:~[47]% lsload

HOST_NAME	status	rl5s	rlm	rl5m	ut	pg	ls	it	tmp	swp	mem
...											
ncc64bartel008.	ok	0.0	0.0	0.0	0%	3.7	0	77888	69G	4092M	3904M
ncc64021.wi.mit	ok	0.0	0.0	0.0	0%	3.8	0	80960	62G	4094M	7936M
ncc64003.wi.mit	ok	0.0	0.0	0.0	0%	3.9	0	80960	69G	4100M	7932M
ncc64bartel010.	ok	0.0	0.0	0.0	0%	4.3	0	516	69G	4100M	3904M
ncc64024.wi.mit	ok	0.0	0.0	0.0	0%	1.7	0	29456	61G	4096M	3322M
ncc64017.wi.mit	ok	0.0	0.0	0.0	0%	2.0	0	80960	56G	4096M	7848M
bartelblade010.	ok	0.3	0.0	0.0	0%	0.1	0	31936	59G	2862M	7976M
ncc64015.wi.mit	ok	1.0	1.0	1.0	25%	3.4	0	80960	12G	4090M	1297M
bartelblade011.	ok	1.8	1.0	1.0	25%	0.7	0	31936	59G	2862M	6660M
...											

Annotations:

- Accept job ? (points to status)
- Load index (points to rl5s, rlm, rl5m)
- CPU utilization (points to ut)
- free space in /tmp (points to tmp)
- available swap space (points to swp)
- available RAM (points to mem)

bhosts – static and dynamic resources

Accept job?

Max jobSlots per host

Max jobSlots per user

JobSlots started

```
byuan@barra:~[59]# bhosts
HOST_NAME      STATUS      JL/U      MAX      NJOBS      RUN      SSUSP      USUSP      RSV
bartelblade001.wi. ok          3         8         0         0         0         0         0
bartelblade002.wi. ok          3         8         0         0         0         0         0
bartelblade003.wi. ok          3         8         0         0         0         0         0
bartelblade004.wi. ok          3         8         0         0         0         0         0
bartelblade005.wi. ok          3         8         0         0         0         0         0
bartelblade006.wi. ok          3         8         0         0         0         0         0
bartelblade007.wi. ok          3         8         0         0         0         0         0
bartelblade008.wi. ok          3         8         0         0         0         0         0
bartelblade009.wi. ok          3         8         0         0         0         0         0
bartelblade010.wi. ok          3         8         0         0         0         0         0
bartelblade011.wi. ok          3         8         1         1         0         0         0
ncc001.wi.mit.edu closed      0         0         0         0         0         0         0
ncc002.wi.mit.edu unavail    2         5         0         0         0         0         0
ncc003.wi.mit.edu ok         2         5         0         0         0         0         0
```

LSF selects which job to run next based on:

- Resources requirements of the applications
 - queue: `bsub -q sq32hp myscript` ✓
 - job requirement ✓
- Current load conditions ✓
- **How important you are**

User priority

- **bqueues -l normal**

...

SHARE_INFO_FOR: normal/

USER/GROUP	SHARES	PRIORITY	STARTED	RESERVED	CPU_TIME	RUN_TIME
skm	1	0.333	0	0	0.0	0
thiruvil	1	0.333	0	0	0.0	0
byuan	1	0.333	0	0	0.0	0
dweinber	1	0.333	0	0	0.0	0
gurdziel	1	0.333	0	0	0.0	0
gbell	1	0.333	0	0	0.0	0
bellott	1	0.059	1	0	1258.2	55691
jhughes	1	0.007	1	0	0.0	662391

...

Commands we have learned

- bsub
- bjobs
- bpeek
- bstop
- bresume
- bkill
- bqueues
- lshosts
- lsload
- bhosts

References

- **Platform LSF Reference:**
 - Descriptions of all commands
- **Running Jobs with Platform LSF**
 - Introduction to basic concepts of LSF software to run and monitor jobs

http://iona.wi.mit.edu/bio/bioinfo/docs/LSF_help.html