资源管理软件 TORQUE 与作业调度软件 Maui 的安装、设置及使用

李会民（hmli@ustc.edu.cn）

中国科学技术大学网络信息中心

2008 年 1 月

目录

1 资源管理软件 TORQUE 的安装与设置

TORQUE 和 Maui 可以从 http://www.clusterresources.com 上下载。以下仅是粗略配置，详细配置请参考相关手册：

- TORQUE：http://www.clusterresources.com/torquedocs21/
- Maui：http://www.clusterresources.com/products/maui/docs/mauiusers.shtml

1.1 服务节点安装 TORQUE

这里假设服务节点的机子名为 kd50，其中一个计算节点的名字为 node0101。

root@kd50# **tar zxvf torque-2.2.1.tar.gz**

root@kd50# **cd torque-2.2.1**

root@kd50# **./configure –prefix=/opt/torque-2.2.1 –with-rcp=rcp**

上面 –with-rcp=rcp 设置为利用 rsh 协议在节点间传输文件，也可设置为 –with-rcp=scp 以利用 scp 协议进行传输。利用 rcp 或者 scp 传输需要配置节点间无须密码访问，具体请参看相关文档。

root@kd50# **make**

root@kd50# **make install**

1.2 服务节点初始化并设置 TORQUE

将 TORQUE 的可执行文件所在的目录放入系统的路径中，修改 /etc/profile：

```
TORQUE=/opt/torque-2.2.1
MAUI=/opt/maui-3.2.6p20
if [ "`id -u`" -eq 0 ]; then
  PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:"
  PATH=$PATH:$TORQUE/bin:$TORQUE/sbin:$MAUI/bin:$MAUI/sbin
else
  PATH="/usr/local/bin:/usr/bin:/bin:/usr/games:$TORQUE/bin:$MAUI/bin"
  PATH=$PATH:$TORQUE/bin:$MAUI/bin
fi
```

上面将同时设置 Maui 的路径，如在这里已经设置了，并且 Maui 安装路径为上面的话，后面就无需再设置 Maui 的路径。

修改后使设置的环境变量生效：

source /etc/profile

将 root **设置为** TORQUE **的管理帐户：**

root@kd50# **./torque_setup root**

在 /var/spool/torque/server_priv/nodes **中添加计算节点的机器名，类似：**

```
kd50
node0101
```

如果服务节点不参与计算的话，需要将服务节点的机器名去掉。如果 node0101 **上有两个处理单元，就设置为** node0101 np=2。

如果 /var/spool/torque **下的目录** spool **和** undelivered **的权限不是** drwxrwxrwt **的话，需要** chmod 1777 spool undelivered。

创建作业队列：

root@kd50# **pbs_server -t create**

root@kd50# **qmgr**

输入下面 Qmgr: **后的内容，将设置一个默认队列** dque**：**

```
Qmgr: create queue dque queue_type=execution
Qmgr: set server default_queue=dque
Qmgr: set queue dque started=true
Qmgr: set queue dque enabled=true
Qmgr: set server scheduling=true
```

可以通过下面的代码来检查 pbs_server **是否正常运行，若** pbs_server **没有运行，则首先运行该程序，然后执行下面的代码：**

```
# shutdown server
  qterm -t quick
# start server
  pbs_server
# verify all queues are properly configured
  qstat -q
# view additional server configuration
  qmgr -c 'p s'
# verify all nodes are correctly reporting
  pbsnodes -a
# submit a basic job
  echo "sleep_30" | qsub
```

```
# verify jobs display
  qstat
```

1.3 计算节点上安装 TORQUE

先在服务节点上的编译 TORQUE **的目录下执行下面命令生成所需要的包：**

root@kd50# **make packages**

该命令执行之后一共产生五个包，分别为：

- torque-package-clients-linux-i686.sh
- torque-package-devel-linux-i686.sh
- torque-package-doc-linux-i686.sh
- torque-package-mom-linux-i686.sh
- torque-package-server-linux-i686.sh

然后将这些包传送给机群中的所有计算节点并在各计算节点上执行安装，比如：

root@node0101# **./torque-package-clients-linux-i686.sh –install**

1.4 计算节点配置 TORQUE

/var/spool/torque **是** TORQUE **的配置目录，只要在该目录下创建一个文件** server_name**，其内容是服务节点的机器名。**

对于 NFS **文件共享系统来说，还必须告诉** TORQUE **这种共享的用户目录，编辑** /var/spool/torque/mom_priv/config**，其内容类似：**

```
$pbsserver      kd50        # note: hostname running pbs_server
$logevent       255         # bitmap of which events to log
$usecp kd50:/home /home
```

其中 $pbsserver **后指定服务节点的主机名，**$usecp **后面的表示的共享** home**。**

在 /etc/profile 中设置环境变量：

```
TORQUE=/opt/torque-2.2.1
if [ "`id -u`" -eq 0 ]; then
  PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:"
  PATH=$PATH:$TORQUE/bin:$TORQUE/sbin
else
  PATH="/usr/local/bin:/usr/bin:/bin:/usr/games"
  PATH=$PATH:$TORQUE/bin
fi
```

source /etc/profile 后，可以直接运行pbs_mom 启动计算节点的守护进程。

2 安装与配置作业调度软件：Maui

TORQUE 自带的作业调度进程 pbs_sched 非常简单，建议不要启动此服务，而是在服务节点上安装 Maui 来进行作业调度，注意：在计算节点上无须安装 Maui。

2.1 服务节点上安装 Maui

root@kd50# **tar zxvf maui-3.2.6p20-snap.1182974819.tar.gz**

root@kd50# **cd maui-3.2.6p20**

root@kd50# **./configure –prefix=/opt/maui-3.2.6p20 –with-pbs=/opt/torque-2.2.1**

root@kd50# **make**

root@kd50# **make install**

2.2 服务节点上配置 Maui

修改 /usr/local/maui/maui.cfg，主要为下面几项：

```
SERVERHOST          kd50
# primary admin must be first in list
ADMIN1              root

# Resource Manager Definition

RMCFG[KD50] TYPE=PBS@RMNMHOST@
```

```
RMTYPE[0] PBS
```

在 /etc/profile **中设置环境变量：**

```
TORQUE=/opt/torque-2.2.1
MAUI=/opt/maui-3.2.6p20
if [ "`id -u`" -eq 0 ]; then
  PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:"
  PATH=$PATH:$TORQUE/bin:$TORQUE/sbin:$MAUI/bin:$MAUI/sbin
else
  PATH="/usr/local/bin:/usr/bin:/bin:/usr/games"
  PATH=$PATH:$TORQUE/bin:$MAUI/bin
fi
```

source /etc/profile **后启动** Maui：

root@kd50# **maui**

注意不要在服务节点上启动 pbs_sched。

3 作业运行

系统利用 TORQUE **和** Maui **进行资源和作业管理，所有需要运行的作业无论是用于程序调试还是业务计算均必须通过** qsub **命令提交，提交后可以利用** TORQUE **和** Maui **的相关命令查询作业状态等。为了利用** qsub **提交作业，用户需针对此作业创建提交脚本，在脚本里面设定需要运行的作业参数等。在此分别给出串行和并行的简单脚本，用户可以修改此脚本以适用于自己的作业，如需要更加高级的功能请参考** TORQUE **手册。**

3.1 串行作业

对于串行程序，用户可编写命名为 serial_job.pbs**（此脚本名可以按照用户喜好命名）的串行作业脚本，其内容如下：**

```
#!/bin/sh
#PBS -N job_name
#PBS -o job.log
#PBS -e job.err
#PBS -q dque
cd yourworkdir
echo Running on hosts `hostname`
echo Time is `date`
echo Directory is $PWD
echo This job runs on the following nodes:
cat $PBS_NODEFILE
echo This job has allocated 1 node
./yourprog
```

注意[1]**，**TORQUE **建立在** PBS **作业管理系统之上，**PBS **的参数需在作业提交脚本中利用** #PBS **设置。上述脚本利用** qsub **命令提交后，表示进入** yourworkdir **目录后，提交到** dque **队列，其作业名为** job_name**，标准输出和错误输出将分别存在此目录下的** job.log **和** job.err **文件中。上述脚本中以** #PBS **开头的几行的** -N、-o、-e、-q **参数后分别设置的是这个作业的名字** job_name**、标准输出定向到的文件名** job.log**、标准错误输出定向到的文件名** job.err**、作业使用的队列名** dque**。**

作业脚本编写完成后，可以按照下面命令提交作业：

user@kd50:~ /work$ **qsub ser_job.pbs**

如果成功，将有类似下面的输出：

```
37.kd50
```

其中 37.kd50 **表示的是作业号，由两部分组成，**37 **表示的是作业序号，**kd50 **表示的是作业管理系统的主机名，也就是登录节点名，之后可以用此作业号来查询作业及终止此作业等。**

[1]**此脚本中** `hostaname` **等中的是键盘左上角的反引号** `**，不是右侧的** '

3.2 并行作业

与串行作业类似，对于并行作业，则需要编写类似下面脚本：

```
#!/bin/sh
#PBS -N job_name
#PBS -o job.log
#PBS -e job.err
#PBS -q dque
#PBS -l nodes=4
cd yourworkdir
echo Time is `date`
echo Directory is $PWD
echo This job runs on the following nodes:
cat $PBS_NODEFILE
NPROCS=`wc -l<$PBS_NODEFILE`
echo This job has allocated $NPROCS nodes
mpiexec -machinefile $PBS_NODEFILE -np $NPROCS ./yourprog
```

与串行程序的脚本相比，主要不同之处在于在#PBS **开头的** -l **参数后设置：**nodes=**所需要的进程数，另外请注意需采用** mpiexec **的命令格式提交并行可执行程序。**

与串行作业类似，可使用下面方式提交：

user@kd50:~ /work$ **qsub par_job.pbs**

3.3 常用作业管理命令

与作业相关 TORQUE **和** Maui **常用的用户命令主要有：**

- canceljob**：取消已存在的作业**
- checkjob**：显示作业状态、资源需求、环境、限制、信任、历史、已分配资源和资源利用等**
- nqs2pbs**：将** nqs **作业脚本转换为** pbs **作业脚本**
- pbsnodes**：显示节点信息**
- printjob**：显示指定作业脚本中的作业信息**
- qdel**：取消指定的作业**

- qhold：**挂起一个作业**
- qmove：**将一个作业从一个队列移到另一个队列中**
- qnodes：pbsnodes **的别名，显示节点信息**
- qorder：**交换两个作业的排队顺序**
- qrls：**将被挂起的作业送入准备运行的队列中**
- qselect：**显示符合条件的作业的作业号**
- qstat：**显示队列、服务节点和作业的信息**
- qsub：**提交作业**
- showbf：**显示有特殊资源需求的资源的可用性**
- showq：**显示已激活和空闲的作业的优先级细节**
- showstart：**显示空闲作业的估计开始时间**
- tracejob：**追踪作业信息**

具体请参考 TORQUE **和** Maui **用户手册。**

3.3.1 查看队列中的作业状态：qstat

利用 qstat **可以查看作业的运行状态：**

user@kd50:~ /work$ **qstat**

输入上面命令后，将给出类似下面的输出：

Job id	Name	User	Time Use	S	Queue
48.kd50	job_name4	user	0	E	dque
49.kd50	job_name1	user	00:00:00	R	dque
50.kd50	job_name2	user	0	H	dque
51.kd50	job_name3	user	0	Q	dque

上面几列的含义分别为：作业号、作业名、用户名、使用的时间、状态、队列名，其中状态中的 E、Q、H **和** R **分别表示作业处于退出、挂起、排队和运行中。**

3.3.2 挂起作业：qhold

qhold 命令可以挂起作业，被挂起的作业将不被执行，这样可以让其余作业优先得到资源运行，被挂起的作业在用 qstat 命令查询时显示的状态标志为 H，下面命令将挂起作业号为 50.kd50 的作业：

user@kd50:~ /work$ **qhold 50.kd50**

3.3.3 取消挂起：qrls

被挂起的作业可以利用 qrls 来取消挂起，重新进入等待运行状态：

user@kd50:~ /work$ **qrls 50.kd50**

3.3.4 终止作业：qdel 和 canceljob

用户如果想终止一个作业，可以利用 qdel 或 canceljob 来取消：

user@kd50:~ $ **qdel 50.kd50**

user@kd50:~ $ **canceljob 51.kd50**

3.3.5 查看作业状态：checkjob

利用 checkjob **可以查看作业的状态：**

user@kd50:~ $ **checkjob 51.kd50**

```
checking job 51

State: Hold
Creds:  user:user  group:user  class:dque  qos:DEFAULT
WallTime: 00:00:00 of 99:23:59:59
SubmitTime: Sun Dec  2 19:22:19
  (Time Queued  Total: 00:46:13  Eligible: 00:24:40)

Total Tasks: 4

Req[0]  TaskCount: 4  Partition: ALL
Network: [NONE]  Memory >= 0  Disk >= 0  Swap >= 0
Opsys: [NONE]  Arch: [NONE]  Features: [NONE]

IWD: [NONE]  Executable:  [NONE]
Bypass: 0  StartCount: 0
PartitionMask: [ALL]
Flags:       RESTARTABLE

PE:  4.00  StartPriority:  24
cannot select job 51 for partition DEFAULT (non-idle state 'Hold')
```

从上面的 State: Hold **可以看出作业已被挂起。**

user@kd50:~ $ **checkjob 49.kd50**

```
checking job 49
State: Running
Creds:  user:user  group:user  class:dque  qos:DEFAULT
WallTime: 1:07:14 of 99:23:59:59
SubmitTime: Sun Dec  2 19:02:10
  (Time Queued  Total: 00:00:01  Eligible: 00:00:01)
StartTime: Sun Dec  2 19:02:11
Total Tasks: 4
Req[0]  TaskCount: 4  Partition: DEFAULT
Network: [NONE]  Memory >= 0  Disk >= 0  Swap >= 0
Opsys: [NONE]  Arch: [NONE]  Features: [NONE]
```

```
NodeCount: 4
Allocated Nodes:
[node04:1][node03:1][node02:1][node01:1]

IWD: [NONE]  Executable:  [NONE]
Bypass: 0  StartCount: 1
PartitionMask: [ALL]
Flags:       RESTARTABLE

Reservation '49' (-1:06:52 -> 99:22:53:07  Duration: 99:23:59:59)
PE:  4.00  StartPriority:  1
```

从上面的 State: Running **可以看出作业处于运行中，并且可以看到占用的资源状态。**

3.3.6 交换两个作业的排队顺序：qorder

利用 qorder **可以交换两个作业的排队顺序：**

user@kd50:~ $ **qstat**

Job id	Name	User	Time Use	S	Queue
52.kd50	job_name1	user	0	H	dque
53.kd50	job_name2	user	0	Q	dque
54.kd50	job_name3	user	0	Q	dque

user@kd50:~ $ **qorder 53.kd50 54.kd50**

user@kd50:~ $ **qstat**

Job id	Name	User	Time Use	S	Queue
52.kd50	job_name1	user	0	H	dque
54.kd50	job_name3	user	0	Q	dque
53.kd50	job_name2	user	0	Q	dque

可见 qorder 53.kd50 54.kd50 **后，作业** 53.kd50 **和** 54.kd50 **的排队顺序相互对换了，这样作业** 54.kd50 **将优先于** 53.kd50 **运行。**

3.3.7 选择符合特定条件的作业的作业号：qselect

qselect **可以用来显示符合一定条件的作业的作业号，比如选择被挂起的作业，可以用下面的命令：**

user@kd50:~ $ **qselect -s H**

```
52.kd50
```

3.3.8 显示队列中作业的信息：showq

user@kd50:~ $ **showq**

```
ACTIVE JOBS--------------------
JOBNAME  USERNAME      STATE  PROC   REMAINING            STARTTIME

52             user  Running     4  99:22:44:09  Sun Dec  2 21:04:37
     1 Active Job     4 of    4 Processors Active (100.00%)

IDLE JOBS----------------------
JOBNAME  USERNAME      STATE  PROC     WCLIMIT            QUEUETIME

54             user       Idle     4  99:23:59:59  Sun Dec  2 21:04:45 1
Idle Job

BLOCKED JOBS----------------
JOBNAME  USERNAME      STATE  PROC     WCLIMIT            QUEUETIME

53             user       Hold     4  99:23:59:59  Sun Dec  2 21:04:37
Total Jobs: 3   Active Jobs: 1   Idle Jobs: 1   Blocked Jobs: 1
```

3.3.9 显示节点信息：pbsnodes 和 qnodes

利用 pbsnodes **和** qnodes **（实际两者是同一个命令的两个名字）可以显示系统各个节点的信息，比如空闲（**free**）、当机（**down**）、离线（**offline**）。例如：显示所有空闲的节点：**

user@kd50:~ $ **pbsnodes -l free**

其输出为：

```
node0101              free
node0102              free
node0104              free
```