PBS作业调度系统
管理和使用介绍

目录

集群使用中存在的问题

- 集群结构的松散性
- 节点类型的差别（CPU类型、内存大小、数量等）
- 用户不同类型的作业（串行/并行，各类应用软件）
- 如何对用户可以使用资源进行限制

作业调度系统主要作用

- **单一系统映像**
  - ✓ 解决集群结构松散问题；
  - ✓ 统一用户接口，使用简化；
- **系统资源整合**
  - ✓ 管理异构资源和异构系统；
- **多任务管理**
  - ✓ 统一管理任务，避免冲突；
- **资源访问控制**
  - ✓ 基于策略的资源访问控制；

调度系统是面向集群的操作系统。

作业调度系统主要功能

- **排队**
  收集作业，等待运行
- **调度**
  决定哪个作业什么时间运行在哪里
- **监控**
  监控作业和资源，为调度提供依据

调度系统概述-基本概念

■ 资源（Resource）

- ✓ 作业运行过程中使用的可量化实体都是资源；
- ✓ 包括硬件资源（节点、内存、CPU 、GPU等）和软件资源（ License ）；

■ 集群（Cluster）

- ✓ 包含计算、存储、网络等各种资源实体且彼此联系的资源集合；
- ✓ 在物理上，一般由计算处理、互联通信、I/O 存储、操作系统、编译器、运行环境、开发工具等多个软硬件子系统组成；
- ✓ 节点是集群的基本组成单位，从角色上一般可以划分为管理节点、登陆节点、计算节点、存储节点等。

■ 作业（Job）

- ✓ 物理构成，一组关联的资源分配请求，以及一组关联的处理过程；
- ✓ 交互方式，可以分为交互式作业和非交互式作业；
- ✓ 资源使用，可以分为串行作业和并行作业；

调度系统概述-基本概念（续）

- **队列（queue）**
  - ✓ 带名称的作业容器；
  - ✓ 用户访问控制；
  - ✓ 资源使用限制；
- **作业调度系统（Job Schedule System）**
  - ✓ 负责监控和管理集群中资源和作业的软件系统；
  - ✓ 通常由资源管理器、调度器、任务执行器，以及用户命令和API组成；

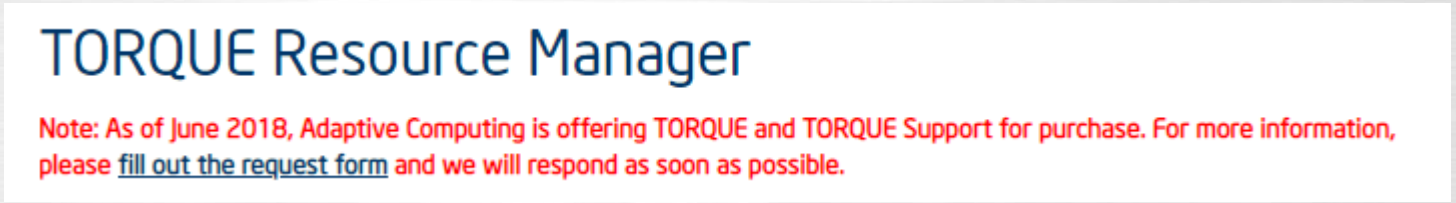
PBS作业调度系统

PBS最初由NASA的Ames研究中心开发，为了提供一个能满足异构计算网络需要的软件包。它力求提供对批处理的初始化和调度执行的控制，允许作业在不同主机间的路由。

PBS作业调度系统

- PBS的开源版本为OpenPBS，目前已经停止开发
- PBS的商业版为PBS Pro，由Altair公司开发和维护
- TORQUE（Tera-scale Open-source Resource and QUEue manager），Torque是Clustering公司接过了OpenPBS，并给与后续支持的一个开源版本。修正了OpenPBS的很多bug，功能和可扩展性都有很大提高

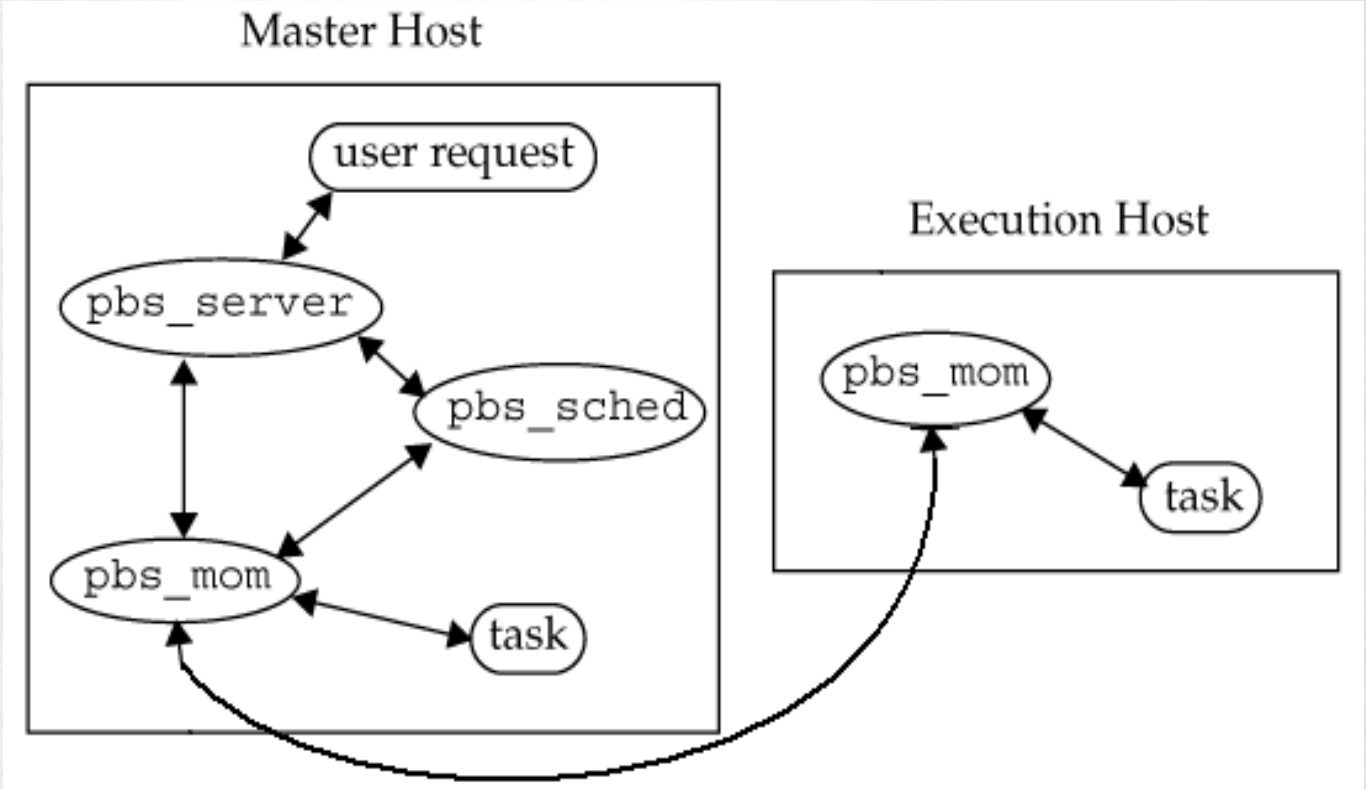
http://www.adaptivecomputing.com/products/torque/



PBS的组成

服务器:pbs_server
调度器:pbs_sched
执行器:pbs_mom
命令行:用户脚本，管理命令等

PBS的结构


Master Host
user request
pbs_server
pbs_sched
pbs_mom
task
Execution Host
pbs_mom
task

PBS的安装（Torque Server端）

- 解压源文件包
  tar zxvf torque-4.X.X.tar.gz

- 编译设置
  cd torque-4.X.X
  ./configure --enable-docs --with-scp --enable-syslog --enable-acct-x --with-pam=yes --prefix=$MANAGER_HOME --with-server-home=$MANAGER_HOME --disable-privports LIBS=-lcrypto >> $GRIDVIEW_NODE_INSTALL_LOG 2>&1

- 编译和安装
  make
  make install

PBS的安装（Torque Client端）

```
./configure --enable-docs --with-scp --enable-syslog --enable-acct-x --with-pam=yes --prefix=$MANAGER_HOME --with-server-home=$MANAGER_HOME --disable-privports LIBS=-lcrypto >> $GRIDVIEW_NODE_INSTALL_LOG 2>&1
```

安装后的目录结构

```
drwxr-xr-x  2 root root 4096 Nov 18 06:11 aux
drwxr-xr-x  3 root root 4096 Nov 15 03:08 bin
drwxrwxrwt  2 root root 4096 Nov 15 03:05 checkpoint
drwxr-xr-x  2 root root 4096 Nov 18 00:00 client_logs
drwxr-xr-x  2 root root 4096 Nov 15 03:05 include
drwxr-xr-x  2 root root 4096 Nov 15 03:05 job_logs
drwxr-xr-x  4 root root 4096 Nov 15 03:05 lib
drwxr-xr-x  2 root root 4096 Nov 18 00:00 mom_logs
drwxr-x--x  4 root root 4096 Nov 15 09:33 mom_priv
-rw-r--r--  1 root root   36 Nov 15 03:05 pbs_environment
drwxr-xr-x  2 root root 4096 Nov 15 03:05 sbin
drwxr-xr-x  2 root root 4096 Nov 15 03:05 sched_logs
drwxr-x---  3 root root 4096 Nov 15 03:05 sched_priv
drwxr-xr-x  2 root root 4096 Nov 18 00:00 server_logs
-rw-r--r--  1 root root    6 Nov 15 03:05 server_name
drwxr-x--- 13 root root 4096 Nov 18 16:03 server_priv
drwxr-xr-x  3 root root 4096 Nov 15 03:05 share
drwxrwxrwt  2 root root 4096 Nov 15 03:05 spool
drwxrwxrwt  2 root root 4096 Nov 15 03:05 undelivered
```

PBS Server配置

- 确认文件$TORQUE_HOME/server_name
- Server配置目录 $TORQUE_HOME/server_priv/
- 计算节点列表及属性： $TORQUE_HOME/server_priv/nodes

```
#节点名 CPU资源 GPU资源  节点特殊属性...
node2 np=12 gpus=1 amd lab1
node3 np=12 gpus=1 amd lab1
node4 np=8  gpus=2 intel lab2
node5 np=8  gpus=2 intel lab2
node6 np=8  gpus=2 intel lab2
```

PBS Server配置

>qstat –Bf可查看Server的配置

> qmgr -c 'p s'可查看Server和队列的所有配置信息

qmgr命令语法格式:
qmgr –c ‘<action> <object-type> <object name> <attributes> = <value>’
Action: active, create, delete, set, unset, list, print, quit
Object: queue server

PBS Server配置

```
[root@gv47 dispatcher]# qstat -Bf
Server: gv47
    server_state = Active
    scheduling = True
    total_jobs = 2
    state_count = Transit:0 Queued:2 Held:0 Waiting:0 Running:0 Exiting:0 Comp
        lete:0
    acl_hosts = gv47,localhost
    acl_users = nanya@*,root@*
    acl_roots = root@*
    managers = root@*
    operators = root@*
    default_queue = FLUENT
    log_events = 255
    mail_from = adm
    query_other_jobs = True
    scheduler_iteration = 600
    node_check_rate = 150
    tcp_timeout = 90
    job_stat_rate = 120
    poll_jobs = True
    log_level = 7
    owner_purge = False
    mom_job_sync = True
    pbs_version = 4.2.7
    keep_completed = 300
    submit_hosts = gv47,gv241,gv174
    log_file_max_size = 512000
    log_file_roll_depth = 3
```

PBS Server配置

属性名称	默认值	含义
job_stat_rate	45 (30 in TORQUE 1.2.0p5 and earlier)	If data is older than this value, the pbs_server daemon will contact the MOMs with stale data to request an update.
keep_completed	300	The amount of time a job will be kept in the queue after it has entered the completed state.
log_events	255	If you want to log only error, system, and job information, use qmgr to set log_events to 11.
log_keep_days	30	Specifies how long (in days) a server or MOM log should be kept.
log_level	7	Specifies the pbs_server logging verbosity. Maximum value is 7.
next_job_number		Specifies the ID number of the next job.
node_check_rate	150	Specifies the minimum duration (in seconds) that a node can be unresponsive to server queries before being marked down by the pbs_server daemon.

更多属性man pbs_server_attributes

设置PBS Server属性

>qmgr -c 'set server default_queue=batch'

```
set server acl_hosts = kmn
set server log_events = 511
set server mail_from = adm
set server node_check_rate = 150
set server tcp_timeout = 6
```

计算节点的配置

$TORQUE_HOME/mom_priv/config

```
$pbsserver gv141
#$ideal_load 100
#$max_load 150
$ideal_load 2.50
$max_load 4.00
$restricted *.gv141
$prologalarm 30
#$usecp *:/home /home
$spool_as_final_name true
$check_poll_time 45
$status_update_time 45
$loglevel 7
$logevent 0x0ff
$log_file_max_size 512000
$log_file_roll_depth 3
$log_keep_days 30
```

属性	含义
logevent	The first example would set the log event mask to 0x1ff (511) which enables logging of all events including debug events. The second example would set the mask to 0x0ff (255) which enables all events except debug events.
ideal_load	Ideal processor load. Represents a low water mark for the load average. A node that is currently busy will consider itself free after falling below ideal_load.
max_load	Maximum processor load. Nodes over this load average are considered busy
status_update_time	Specifies (in seconds) how often the MOM updates its status information to pbs_server.
spool_as_final_name	This makes the job write directly to its output destination instead of a spool directory. This allows users easier access to the file if they want to watch the jobs output as it runs.

PBS的队列设置

- PBS要能正常运行还需要通过qmgr命令在server进行配置，设置一些属性。输入qmgr命令进入配置交互命令。下面是让PBS可以正常运行的基本设置。

创建队列	create queue <queue> set queue <queue> queue_type = execution
打开和启动队列	set queue <queue> enable=true set queue <queue> started=true
设置默认队列	set server default_queue=<queue>

PBS的队列设置（续2）

- 导入server配置文件：
  qmgr < queue.conf
- 备份配置文件：
  qmgr -c 'print server' > queue.conf
  qmgr -c 'print queue middle' 输出队列middle的配置
- 配置文件例子：

```
create queue default
set queue default queue_type = execution
set queue default max_running = 10
set queue default enabled = True
set queue default started = True

set server scheduling = True
set server default_queue = default
set server query_other_jobs = True
```

PBS的队列设置（续3）

中科曙光 Sugon

- 资源限制

resources_default.nodes	该队列默认的使用节点数
resources_default.walltime	该队列默认的墙钟时间，格式：时:分:秒
resources_max.<resource>	该队列最大允许的资源数，资源包括cput, nodes, pmem, walltime等
resources_min.<resource>	该队列最小资源数限制，资源包括cput, nodes, pmem, walltime等
max_running	某队列最多可运行的作业数，如果该项为0或没有该项，表示没有限制
max_queuable	队列中最大的作业驻留数
max_user_run	一个用户最多可以运行的作业数

```
create queue high
set queue high queue_type = execution
set queue high max_running = 20
set queue high enabled = True
set queue high started = True
set queue high resource_default.walltime=24:00:00
set queue high resource_default.nodes=2
set queue high resource_max.nodes=4
```

PBS的队列设置（续4）

- 用户限制

acl_user_enable	是否启用用户访问控制，如果acl_user_enable = True，则在acl_users中列出的用户才能使用该队列
acl_users	格式：<用户名@主机名>，用户名不接受通配符

```
create queue high
set queue high queue_type = execution
set queue high max_running = 20
set queue high enabled = True
set queue high started = True
set queue high resource_default.walltime=24:00:00
set queue high resource_default.nodes=2
set queue high resource_max.nodes=4

set queue high acl_user_enable=true
set queue high acl_users=alice
set queue high acl_users+=bob
set queue high acl_users+=tom
```

PBS的队列设置（续5）

- 队列节点选择

acl_host_enable	如果acl_host_enable = true，则acl_hosts属性中列出的主机才能使用该队列
acl_hosts	该队列可以使用的节点列表

```
create queue high
set queue high queue_type = execution
set queue high max_running = 20
set queue high enabled = True
set queue high started = True
set queue high resource_default.walltime=24:00:00
set queue high resource_default.nodes=2
set queue high resource_max.nodes=4

set queue high acl_host_enable=true
set queue high acl_hosts=node1
set queue high acl_hosts+=node2
set queue high acl_hosts+=node3
set queue high acl_hosts+=node4
```

PBS的队列设置（续6）

keep_completed	Specifies the number of seconds jobs should be held in the Completed state after exiting.
started	Specifies whether jobs in the queue are allowed to execute.
enabled	Specifies whether the queue accepts new job submissions.

更多属性man pbs_queue_attributes

取消某个属性的设置unset queue fast max_running

PBS查看计算节点状态

```
[root@gv47~]$ pbsnodes -a
node2
        state=free
        np=12
        ntype=cluster
        …
node3
        state=job-executive
        np=12
        ntype=cluster
        …
node4
        state=down
        np=12
        ntype=cluster
        …
```

PBS查看计算节点状态（续1）

busy	node is full and will not accept additional work
down	node is failing to report, is detecting local failures with node
free	node is ready to accept additional work
job-exclusive	all available virtual processors are assigned to jobs
offline	node has been instructed by an admin to no longer accept work
reserve	node has been reserved by the server
time-shared	node always allows multiple jobs to run concurrently
unknown	node has not been detected
job-sharing	node has been allocated to run multiple shared jobs and will remain in this state until jobs are complete

PBS查看计算节点状态（续2）

pbsnodes命令的主要参数

-a 列出所有节点及其属性，属性包括“state”和“properties”

-o 将指定节点的状态标记为“offline”。这将帮助管理员暂时停止某些节点的服务

-l 以行的方式列出被标记的节点的状态，如 -l free，-l offline

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

-r 清除指定节点的“offline”状态

PBS查看节点状态（续3）

使用第三方 pestat 命令，可以更简洁直观的查看节点的作业负载

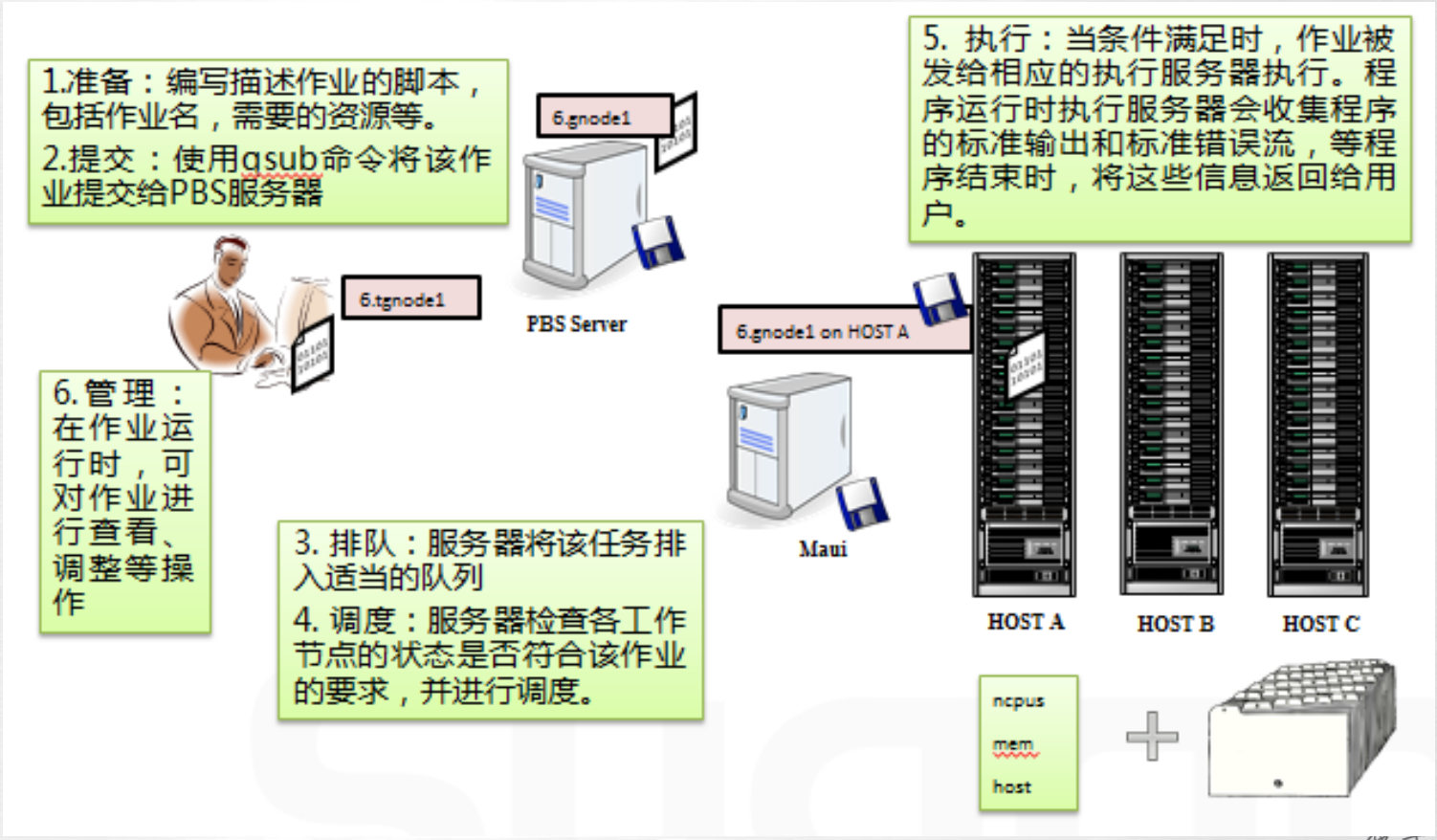
```
[root@gv47 ~]$ pestat
  node     state  load    pmem ncpu   mem   resi usrs tasks  jobids/users
  node21   free  0.00   28070  24  32173    737  0/0    0
  node23   free  0.00  258345  48 262447   4612  0/0    0
  node24   free  0.04  128952  24 133054   2364  0/0    0
  node42   down* 0.00       0   0      0      0  0/0    0
  node43   free  0.00   24025  12  28127    660  0/0    0
  node44   free  0.00   24025  12  28127    661  1/1    0
  node46   free  0.00   24025  12  28127    894  2/1    0
  node47   excl  2.00*  24025  12  24025   1310  2/1   12    2726 NONE*
  node48   down* 0.00       0   0      0      0  0/0    0
  node49   down* 0.00       0   0      0      0  0/0    0
  node50   offl* 0.00   48267  24  52369   1203  0/0    0
  node51   offl* 0.00   48267  24  52369   1202  0/0    0
  node52   offl* 0.00   48267  24  52369   1203  0/0    0
```

PBS管理节点

> qmgr -c "create node node003"
> qmgr -c "delete node node003"

> qmgr -c "set node node001 properties = bigmem"
> qmgr -c "set node node001 properties += dualcore"

作业提交步骤


1.准备：编写描述作业的脚本，包括作业名，需要的资源等。
2.提交：使用qsub命令将该作业提交给PBS服务器
6.gnode1
6.tgnode1
PBS Server
5. 执行：当条件满足时，作业被发给相应的执行服务器执行。程序运行时执行服务器会收集程序的标准输出和标准错误流，等程序结束时，将这些信息返回给用户。
6.gnode1 on HOST A
Maui
HOST A
HOST B
HOST C
6.管理：在作业运行时，可对作业进行查看、调整等操作
3. 排队：服务器将该任务排入适当的队列
4. 调度：服务器检查各工作节点的状态是否符合该作业的要求，并进行调度。
ncpus
mem
host

作业提交-qsub命令

使用方法: qsub <job_attributes/resources> <job_script>

举例: qsub –l nodes=2:ppn=2 test_script.sh

当作业提交后返回作业标识：由作业编号和管理节点名称共同组成

作业批量提交

- 一次可提交一批作业 （Creation of multiple jobs with one qsub command）
- 提交的作业为一组(Reference entire set of jobs as one group)
- 使用环境变量$PBS_ARRAYID可获取其所在的组(Each job can find its place in the array by examining $PBS_ARRAYID)

作业批量提交（Job Array）举例

> qsub -t 0-10 job_script
1098.gnode1
>qstat -t

```
[sghpc@gv47 ~]$ qstat -t
Job ID
-------------------------
1081.gv47
1085.gv47
1086[0].gv47
1086[1].gv47
1086[2].gv47
1086[3].gv47
1086[4].gv47
1086[5].gv47
1086[6].gv47
1086[7].gv47
1086[8].gv47
1086[9].gv47
1086[10].gv47
```

PBS作业脚本

- 本质是一个SHELL脚本
- 注释以“#”开头
- PBS运行参数，以“#PBS”开头
- 可以直接调用SHELL命令和系统命令

```
#PBS -N vasp
#PBS -l nodes=1:ppn=1
#PBS -l walltime=12:00:00
#PBS -q high

cd /home/test/work
./test.exe
```

PBS运行参数

在 PBS 脚本和 qsub 命令行中均有效，qsub命令行参数的优先级更高

运行参数	说　明
-a <作业开始运行的时间>	向PBS系统指定作业运行的开始时间。 作业运行时间格式为：　[[[[CC]YY]MM]DD]hhmm[.SS]
-A <用户名>	使用不同的用户来提交作业，缺省使用当前用户名
-o <标准输出文件的路径> **-e <标准错误输出的路径>**	该参数指定标准错误输出的位置，缺省的情况下，PBS系统把标准输出和标准错误输出放在用户qsub命令提交作业的目录下。 标准错误输出：<作业名>.o<作业号> 标准错误输出：<作业名>.e<作业号> 路径使用如下格式标准：　[<节点名>:]<路径名>
-N <作业名>	指定提交的作业名
-q <目标队列>	指定作业提交的目标队列，其中目标队列可以是目标队列、目标节点名或者是目标节点上的队列。如果目标队列是一个路由队列，那么服务器可能把作业路由到新的队列中。如果该参数没有指定，命令qsub会把作业脚本提交到缺省的队列中。
-l <申请资源列表>	该参数指定作业脚本申请的PBS系统资源列表。 申请资源列表使用如下格式： 　<资源名>[=[<数量>]][,资源名[=[<数量>]]，　.....] 例如作业希望申请在双路节点上申请5个CPU资源的情况， 则可以在脚本中如下： #PBS –l nodes=2:ppn=2+1:ppn=1

PBS运行参数（续）

运行参数	说　明
-d	作业的工作路径（Working directory path to be used for the job）
-W	指定作业的依赖关系（Dependent）
-j	合并作业的输出（标准输出、错误输出）

更多参数 Man qsub

PBS常用的资源需求

变量名	说　明
walltime	作业运行的实际时间的最大值
cput/pcput	作业/单进程占用的CPU时间的最大值
nodes	作业运行需的节点（和cpu）数目
mem/pmem	作业/单个进程可用的物理内存的最大值
vmem/pvmem	作业/单个进程可用的虚拟内存的最大值
gpus	作业需要的GPU卡的数量
software	指定需要的软件资源（如License）
更多资源	Man pbs_resources

作业资源请求参数nodes

中科曙光 Sugon

- 节点请求参数nodes

用法：#PBS -l nodes={<node_count> | <hostname>}[:ppn=<ppn>][:<property>[:<property>]...][+ ...]

举例1：申请2个节点，每个节点起2个进程

#PBS –l nodes=2:ppn=2 job.pbs

调度成功后可能的节点列表如下：

gnode21
gnode21
gnode22
gnode22

作业资源请求参数nodes（续1）

中科曙光 Sugon

- 节点请求参数nodes

举例2：申请2个节点，每个节点起2个进程，另外1个节点启动1个进程

#PBS –l nodes=2:ppn=2+1 job.pbs

调度成功后可能的节点列表如下：

gnode21
gnode21
gnode22
gnode22
gnode23

作业资源请求参数nodes（续2）

中科曙光 Sugon

- 节点请求参数nodes

举例3：申请到节点gnode58上运行4个进程

#PBS –l nodes=gnode58:ppn=4 job.pbs

调度成功后产生的节点列表如下：

gnode58
gnode58
gnode58
gnode58

作业资源请求参数nodes（续3）

中科曙光 Sugon

- 节点请求参数nodes-GPU资源申请

```
# 这是一个申请GPU资源的例子
#PBS -N vasp.Hg
#PBS -j oe
#PBS -l nodes=2:ppn=12:gpus:1
#PBS -q high
```

本例中申请2个计算节点，每个计算节点12个CPU核+1个GPU

指定节点特性

```
# 这是一个并行作业脚本的例子
#PBS -N vasp.Hg
#PBS -j oe
#PBS -l nodes=2:ppn=12:amd
#PBS -q low

echo "This jobs is "$PBS_JOBID@$PBS_QUEUE
NP=`cat $PBS_NODEFILE | wc -l`
cd $PBS_O_WORKDIR
mpirun -np $NP -machinefile $PBS_NODEFILE ./vasp
```

与PBS Server的 $TORQUE_HOME/server_priv/nodes文件中指定的节点特性一致

作业独占请求参数

中科曙光 Sugon

➢独占请求参数

#PBS –W x=NACCESSPOLICY:SINGLEJOB

举例4：申请独占3个节点，其中2个点各启动2个进程，另外一个启动1个进程

示例脚本如下：

```
#PBS –q comm_queue
#PBS –l nodes=2:ppn=1+1
#PBS –W x=NACCESSPOLICY:SINGLEJOB
#PBS –N EXTRA
…
MPIRUN –np 5 –machinefile $PBS_NODEFILE ./myjob
…
```

PBS脚本举例

```
# 这是一个串行作业脚本的例子
#PBS -N test
#PBS -l nodes=1:ppn=1

cd $HOME/test/
./a.out > $HOME/result/a.result
```

```
# 这是一个并行作业脚本的例子
#PBS -N vasp_job
#PBS -l nodes=2:ppn=8
#PBS -q low

echo This jobs is $PBS_JOBID@$PBS_QUEUE
cd $PBS_O_WORKDIR
NP=`cat $PBS_NODEFILE | wc -l`
mpirun -np $NP -machinefile $PBS_NODEFILE ./vasp
```

PBS脚本举例（续1）

- 有时在PBS脚本中，需要对PBS环境变量的内容进行改造

- 比如，$PBS_NODEFILE，该文件内容格式为：

node1
node1
node2
node2

- 对于一般MPI程序，可直接将 $PBS_NODEFILE 作为 MPI 的”-machinefile”参数，如上例所示

- 而一些软件有特殊的节点指定格式，比如ANSYS的命令行参数格式为：

ansys121 -dis -machines node1:2:node2:2 -i test.inp -o test.log

- 这时我们可以对 $PBS_NODEFILE 进行字符处理，得到需要的格式

PBS脚本举例（续2）

```
# 这是一个ANSYS并行作业的例子

#PBS -N ansys_job
#PBS -l nodes=2:ppn=8
#PBS -q low

INPUTFILE=test.inp
OUTPUTFILE=test.log

hosts=`cat $PBS_NODEFILE | uniq -c | awk '{print $2":"$1}' | tr '\n' ':' | sed 's/:$//'`

cd $PBS_O_WORKDIR
ansys121 -dis -machines $hosts -i $INPUTFILE -o $OUTPUTFILE
```

PBS常用的环境变量

变量名	说　明
登陆SHELL继承来的变量	包括$HOME，$LANG，$LOGNAME，$PATH，$MAIL，$SHELL和$TZ。
$PBS_O_HOST	qsub提交的节点名称
$PBS_O_QUEUE	qsub提交的作业的最初队列名称
$PBS_O_WORKDIR	qsub提交的作业的绝对路径
$PBS_JOBID	作业被PBS系统指定的作业号
$PBS_JOBNAME	用户指定的作业名，可以在作业提交的时候用qsub –N <作业名>指定，或者在PBS脚本中加入#PBS –N <作业名>。
$PBS_NODEFILE	PBS系统指定的作业运行的节点名。该变量在并行机和机群中使用。当在PBS脚本中用#PBS –l nodes=2:ppn=2指定程序运行的节点数时，可以使用$PBS_NODEFILE在脚本中引用PBS系统指定的作业运行的节点名。比如： #PBS –l nodes=2:ppn=2 mpirun –np 4 –machinefile $PBS_NODEFILE <程序名>
$PBS_QUEUE	PBS脚本在执行时的队列名

作业监控

- 作业状态

E(退出)	Job is exiting after having run
C(完成)	Job is completed after having run.（作业完成）
H(保留)	Job is held
Q(排队)	Job is Queued
R(运行)	Job is Running
T(移动)	Job is in transition (being moved to a new location)
W(等待)	Job is waiting for its requested execution time to be reached
S(挂起)	Job is suspended

Q->R->E->C

作业监控-qstat命令

qstat –q	查看队列状态
qstat –B	查看管理节点状态
qstat –a,-s,-n,	查看作业状态
qstat –Q	查看队列统计信息
qstat –f *jobid*	查看指定作业详细信息
详细信息	Man qstat

qstat命令举例

- qstat –a
  查看作业列表

```
-bash-3.2$ qstat -a

gnode1:

                                                                  Req'd  Req'd   Elap
Job ID               Username Queue    Jobname          SessID NDS   TSK Memory Time  S Time
-------------------- -------- -------- ---------------- ------ ----- --- ------ ----- - -----
575.gnode1           root     comm_que test.sh            3610     1  --    --  168:0 R 00:10
576.gnode1           ly       multi_co run_tst.c2e       21697     1  --    --  240:0 R 00:02
```

- qstat –s
  可查看注释

```
-bash-3.2$ qstat -s

gnode1:

                                                                  Req'd  Req'd   Elap
Job ID               Username Queue    Jobname          SessID NDS   TSK Memory Time  S Time
-------------------- -------- -------- ---------------- ------ ----- --- ------ ----- - -----
575.gnode1           root     comm_que test.sh            3610     1  --    --  168:0 R 00:10
    --
576.gnode1           ly       multi_co run_tst.c2e       21697     1  --    --  240:0 R 00:02
    --
```

- qstat –n
  可查看节点

```
-bash-3.2$ qstat -n

gnode1:

                                                                  Req'd  Req'd   Elap
Job ID               Username Queue    Jobname          SessID NDS   TSK Memory Time  S Time
-------------------- -------- -------- ---------------- ------ ----- --- ------ ----- - -----
575.gnode1           root     comm_que test.sh            3610     1  --    --  168:0 R 00:10
   gnode21/0
576.gnode1           ly       multi_co run_tst.c2e       21697     1  --    --  240:0 R 00:02
   gnode56/0
```

qstat命令举例（续）

qstat –f <job id>

查看详细作业信息

```
[root@gnode1 ~]# qstat -f 575
Job Id: 575.gnode1
    Job_Name = test.sh
    Job_Owner = root@gLoginNode1
    resources_used.cput = 00:00:00
    resources_used.mem = 2544kb
    resources_used.vmem = 190740kb
    resources_used.walltime = 00:26:04
    job_state = R
    queue = comm_queue
    server = gnode1
    Checkpoint = u
    ctime = Sun Aug 15 15:15:40 2010
    Error_Path = gloginnode1:/root/test.sh.e575
    exec_host = gnode21/0
    Hold_Types = n
    Join_Path = n
    Keep_Files = n
    Mail_Points = a
    mtime = Sun Aug 15 15:15:42 2010
    Output_Path = gloginnode1:/root/test.sh.o575
    Priority = 0
    qtime = Sun Aug 15 15:15:40 2010
    Rerunable = True
    Resource_List.neednodes = 1
    Resource_List.nodect = 1
```

作业监控-qselect命令

qselect -N	指定作业名称（精确查询）
qselect -s	指定作业状态
qselect -u	指定用户名（精确查询）
详细信息	Man qselect

作业管理

- qdel 删除作业
  -m 注释内容
  -p 强制删除（Purge jobs that cannot be properly deleted）
- qalter 修改作业
- qrerun 重新运行作业（管理员执行）
- qsig 发送信号 （管理员执行）

作业管理 qdel

- 使用qdel删除通过qstat查看到的作业
  命令用法: qdel <job id>
  举例: qdel 575.gnode1
- qdel强制删除作业
  命令用法: qdel –p <job id>
  举例: qdel –p 575.gnode1
- qdel删除所有作业
  qdel all

作业管理 qretun

- 执行qrerun命令

命令用法: qrerun <job id>
举例: qrerun 575.gnode1

- 强制重新运行

命令用法: qrerun –f <job id>
举例: qrerun –f 610.gnode1

作业管理 qhold/qrls

- qhold保留排队中的作业

命令用法: qhold <job_id>
举例: qhold 576.gnode1

- qrls释放保留中的作业

命令用法: qrls <job_id>
举例: qrls 576.gnode1

作业管理 qsig

- 挂起运行的作业

  命令用法: qsig -s suspend <job_id>

  举例: qsig -s suspend 576.gnode1

- 恢复保留中的作业

  命令用法: qsig -s resume <job_id>

  举例: qsig -s resume 576.gnode1

When suspended, a job continues to occupy system resources but is not executing and is not charged for walltime. The job will be listed in the "S" state. Manager or operator privilege is required to suspend or resume a job.

交换作业顺序

```
[dawning@node1 ~]$ qstat
Job id            Name             User            Time Use S Queue
----------------  ---------------  ----------------  -------- -  -----
93.node1          test.pbs         zhaocs                 0  R  default
95.node1          vasp.Hg          vasp                   0  E  default
111.node1         structure        amber                  0  Q  default
112.node1         gaussian         gauss                  0  Q  default
```

交换两个作业的排队顺序：

```
qorder 111.node1 112.node1
```

```
[dawning@node1 ~]$ qstat
Job id            Name             User            Time Use S Queue
----------------  ---------------  ----------------  -------- -  -----
93.node1          test.pbs         zhaocs                 0  R  default
95.node1          vasp.Hg          vasp                   0  E  default
112.node1         gaussian         gauss                  0  Q  default
111.node1         structure        amber                  0  Q  default
```

指定作业依赖关系

- PBS脚本中可以指定多个作业之间的依赖关系，比如作业运行前另一个作业必须完成，否则处于排队状态

```
#PBS -N step2
#PBS -l nodes=4:ppn=4
#PBS -q high
#PBS -W depend=after:<JOB_ID>
…
```

- 当指定作业非正常结束，作业才能运行

```
#PBS -N job_rerun
#PBS -l nodes=4:ppn=4
#PBS -q high
#PBS -W depend=afternotok:<JOB_ID>
…
```

PBS的历史作业

$TORQUE_HOME/server_priv/accounting/
format: *YYYYMMDD*

```
[root@gv141 server_priv]# ll
total 56
drwxr-xr-x 2 root root 4096 Nov 18 05:49 accounting
drwxr-x--- 2 root root 4096 Nov 15 03:05 acl_groups
drwxr-x--- 2 root root 4096 Nov 15 03:05 acl_hosts
drwxr-x--- 2 root root 4096 Nov 15 03:05 acl_svr
drwxr-x--- 2 root root 4096 Nov 15 03:05 acl_users
drwxr-x--- 2 root root 4096 Nov 18 06:16 arrays
drwxr-x--- 2 root root 4096 Nov 15 03:05 credentials
drwxr-x--- 2 root root 4096 Nov 15 03:05 disallowed_types
drwxr-x--- 2 root root 4096 Nov 15 03:05 hostlist
drwxr-x--- 2 root root 4096 Nov 18 16:13 jobs
-rw-r--r-- 1 root root   29 Nov 15 03:18 nodes
drwxr-x--- 2 root root 4096 Nov 18 05:58 queues
-rw------- 1 root root 1352 Nov 18 16:03 serverdb
-rw------- 1 root root    5 Nov 15 03:18 server.lock
-rw------- 1 root root    0 Nov 15 03:05 tracking
```

```
[root@gv141 accounting]# ll
total 152
-rw-r--r-- 1 root root  5237 Nov 15 11:38 20181115
-rw-r--r-- 1 root root 72704 Nov 16 19:25 20181116
-rw-r--r-- 1 root root 55151 Nov 17 16:57 20181117
-rw-r--r-- 1 root root  8045 Nov 18 16:08 20181118
```

PBS的历史作业

```
11/15/2018 03:18:34;Q;0.gv141;queue=low
11/15/2018 03:18:35;Q;1.gv141;queue=low
11/15/2018 03:23:46;S;0.gv141;user=test02 group=test02 jobname=test02 queue=low ctime=1542223114 qtime=1542223114 etime=1542223114 work_dir=/tmp/Src
_test/TestAutoInstall/updateGridviewConfig start_count=1 start=1542223426 owner=test02@gv141 exec_host=gv141/0 submit_args=-N test02 -l nodes=gv141
Resource_List.neednodes=gv141 Resource_List.nodect=1 Resource_List.nodes=gv141 Resource_List.walltime=24:00:00
11/15/2018 03:23:46;E;0.gv141;user=test02 group=test02 jobname=test02 queue=low ctime=1542223114 qtime=1542223114 etime=1542223114 work_dir=/tmp/Src
_test/TestAutoInstall/updateGridviewConfig start_count=1 start=1542223426 owner=test02@gv141 exec_host=gv141/0 submit_args=-N test02 -l nodes=gv141
Resource_List.neednodes=gv141 Resource_List.nodect=1 Resource_List.nodes=gv141 Resource_List.walltime=24:00:00 session=0 end=1542223426 Exit_status=
-9 resources_used.cput=00:00:00 resources_used.mem=0kb resources_used.vmem=0kb resources_used.walltime=00:00:00
11/15/2018 03:23:46;S;1.gv141;user=test02 group=test02 jobname=test02 queue=low ctime=1542223115 qtime=1542223115 etime=1542223115 work_dir=/tmp/Src
_test/TestAutoInstall/updateGridviewConfig start_count=1 start=1542223426 owner=test02@gv141 exec_host=gv141/0 submit_args=-N test02 -l nodes=gv141
Resource_List.neednodes=gv141 Resource_List.nodect=1 Resource_List.nodes=gv141 Resource_List.walltime=24:00:00
11/15/2018 03:23:46;E;1.gv141;user=test02 group=test02 jobname=test02 queue=low ctime=1542223115 qtime=1542223115 etime=1542223115 work_dir=/tmp/Src
_test/TestAutoInstall/updateGridviewConfig start_count=1 start=1542223426 owner=test02@gv141 exec_host=gv141/0 submit_args=-N test02 -l nodes=gv141
Resource_List.neednodes=gv141 Resource_List.nodect=1 Resource_List.nodes=gv141 Resource_List.walltime=24:00:00 session=0 end=1542223426 Exit_status=
-9 resources_used.cput=00:00:00 resources_used.mem=0kb resources_used.vmem=0kb resources_used.walltime=00:00:00
```

提交失败的作业不会出现在accounting文件中

PBS的历史作业-作业状态

A	abort（job has been aborted by the server）
C	checkpoint（Job has been checkpointed and held）
D	delete（Job has been deleted）
E	exit（Job has exited (either successfully or unsuccessfully)）
Q	queue（Job has been submitted/queued）
R	rerun（Attempt to rerun the job has been made）
S	start （Attempt to start the job has been made (if the job fails to properly start, it may have multiple job start records)）
T	restart （Attempt to restart the job (from checkpoint) has been made (if the job fails to properly start, it may have multiple job start records)）

PBS的历史作业-作业退出码

值	含义
0	Job execution successful
-1	Job execution failed, before files, no retry
-2	Job execution failed, after files, no retry
-3	Job execution failed, do retry
-4	Job aborted on MOM initialization
-5	Job aborted on MOM init, chkpt, no migrate
-6	Job aborted on MOM init, chkpt, ok migrate
-7	Job restart failed
-8	Exec() of user command failed
-9	Could not create/open stdout stderr files
-10	Job exceeded a memory limit
-11	Job exceeded a walltime limit
-12	Job exceeded a CPU time limit

PBS的历史作业

属性	含义
ctime	Time job was created
etime	Time job became eligible to run
qtime	Time job was queued
start	Time job started to run
end	Time job finished

命令小结

qsub : Submit jobs

qstat : View queues and jobs

qdel: Delete/Cancel batch jobs

qalter : Modify queued batch job

qhold : Hold batch job

qrls : Release batch job holds

qsig: Send a signal to a batch job

qmgr : Mange pbs configuration (queue server, etc.)

pbsnodes : Show and control node status

pestat : show brief status of nodes

qsub常用参数

参数	描述
-d	作业的工作路径（Working directory path to be used for the job）
-N	作业名称（Job Name）
-q	作业使用的队列（Destination queue）
-l	作业需要的资源（Resources required ）
-j	合并作业的输出（标准输出、错误输出）
-W	指定作业的依赖关系（Dependent）
更多参数	Man qsub

qsub -l 用法

该参数指定作业脚本申请的PBS系统资源列表。
申请资源列表使用如下格式：
　　<资源名>[=[<数量>]][,资源名[=[<数量>]]，　.....]
例如作业希望申请在双路节点上申请5个CPU资源的情况，则可以在脚本中如下：
#PBS –l nodes=2:ppn=2+1:ppn=1

#PBS -l，其格式与qsub -l是一样的
ex1: qsub -l nodes=5等价于脚本中的#PBS -l nodes=5,
表示使用集群中任意五个节点来执行该作业
ex2: qsub -l nodes=2:server+14
表示使用集群中的两个server节点与另外十四个节点来执行该作业
ex3: qsub -l nodes=4:ppn=2
表示该作业需要四个节点且每个节点两个处理器
ex4: qsub -l nodes=gvnode1:ppn=4+gvnode2:ppn=2
表示该作业需要gvnode1上的四个处理器与gvnode2上的两个处理器来共同完成

Maui调度器介绍

- PBS带有自己的默认调度策略器（pbs_sched），但是这个最基本的调度策略并不高级。它根据fifo的原则安排作业，对一般的集群管理应该是足够了，但如果你的集群有几百个以上节点，分成若干个队列，那pbs_sched就力不从心了。为此，可以使用一系列第三方的调度策略进行补充。Maui就是被广泛使用的调度策略之一。

- Maui采用积极的调度策略优化资源的利用和减少作业的响应时间。 Maui的资源和负载管理允许高级的参数配置：作业优先级(Job Priority)、调度和分配(Scheduling and Allocation)、公平性和公平共享(Fairness and Fairshare)、回填（Backfill）和预留策略(Reservation Policy)。Maui的QoS机制允许资源和服务的直接传递、策略解除(Policy Exemption)和指定特征的受限访问。

- Maui需要资源管理器和其配合使用。我们可以把Maui想象为PBS中的一个插件。

Maui调度器介绍

重要参数：

USERCFG　用户策略配置

CLASSCFG 队列策略配置

NODEALLOCATIONPOLICY　节点访问策略

RESERVATIONPOLICY　　预约策略

FSPOLICY　公平共享策略

QOSCFG　　服务质量配置（抢占）

Maui调度器介绍

更多信息请查看

http://www.adaptivecomputing.com/products/maui/
http://docs.adaptivecomputing.com/maui/
http://docs.adaptivecomputing.com/maui/pbsintegration.php
http://www.adaptivecomputing.com/support/download-center/maui-cluster-scheduler/

Maui的安装（服务节点上）

- 集群已安装配置好Torque

- 解压源文件包
  tar zxvf maui-3.2.6p17.tar.gz

- 编译设置
  cd maui-3.2.6p17
  ./configure --with-pbs=$MANAGER_HOME --prefix=$SCHEDULER_HOME --with-spooldir=$SCHEDULER_HOME --with-key=2014 LDFLAGS=-lcrypto >> $GRIDVIEW_NODE_INSTALL_LOG 2>&1

  --with-pbs指定Torque安装目录

- 编译和安装
  make
  make install

Maui的安装（服务节点上）（续）

- 编辑启动脚本

cd maui-3.2.6p17
cp etc/maui.d /etc/init.d/
vim /etc/init.d/maui.d
修改其中的” MAUI_PREFIX=/usr/local/maui”
(maui的安装目录，本例为缺省值)

- 停用pbs_sched，启用maui

chkconfig pbs_sched off
chkconfig maui.d on
service pbs_sched stop
service maui.d start

安装后的目录结构

```
drwxr-xr-x 2 root root  4096 Nov 15 03:07 bin
-rw-r--r-- 1 root root  5611 Nov 15 03:07 example.maui.cfg
drwxr-xr-x 2 root root  4096 Nov 15 03:07 include
drwxr-xr-x 2 root root  4096 Nov 15 03:07 lib
drwxr-xr-x 2 root root  4096 Nov 15 03:07 log
-rw-r--r-- 1 root root  2382 Nov 18 05:58 maui.cfg
-rw-r----- 1 root root 38974 Nov 18 16:50 maui.ck
-rw-r----- 1 root root 38974 Nov 18 16:45 maui.ck.1
-rw------- 1 root root     5 Nov 18 05:58 maui.pid
-rw-r--r-- 1 root root    48 Nov 15 03:07 maui-private.cfg
drwxr-xr-x 2 root root  4096 Nov 15 03:07 sbin
drwxrwxrwt 2 root root  4096 Nov 15 03:07 spool
drwxr-xr-x 2 root root  4096 Nov 18 08:05 stats
drwxr-xr-x 2 root root  4096 Nov 15 03:07 tools
drwxr-xr-x 2 root root  4096 Nov 15 03:07 traces
```

Maui的命令

checkjob checknode **showbf** showconfig showgrid **showhold**
showq **showres** **showstart** showstate showstats **runjob** **canceljob** ...

等一系列查询和管理命令，不过只有部分命令是普通用户有权限运行的（比如上面粗体的），其中有些命令与PBS提供的命令功能类似，比如 showq 与 qstat，canceljob 与 qdel 等。

作业诊断（Job Failures）

- 使用checkjob查看作业的详细信息，包括状态、队列、资源请求、实际运行节点、启动时间、结束时间、优先级等

命令用法：

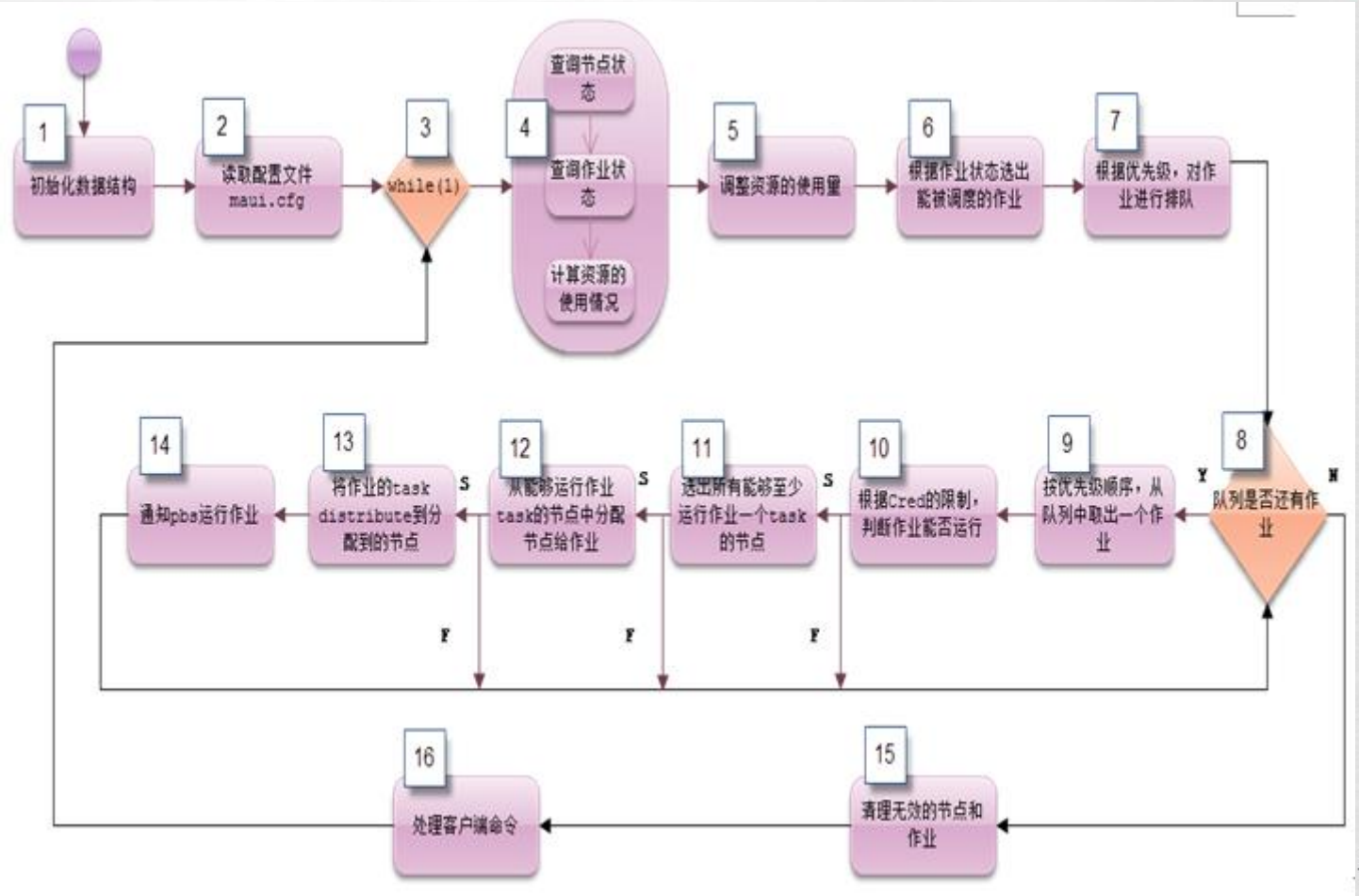
checkjob <job_id>

举例：

checkjob 575

```
[root@gLoginNode1 ~]# checkjob 575


checking job 575

State: Running
Creds:  user:root  group:root  class:comm_queue  qos:DEFAULT
WallTime: 00:37:49 of 7:00:00:00
SubmitTime: Sun Aug 15 15:15:40
  (Time Queued  Total: 00:00:02  Eligible: 00:00:02)

StartTime: Sun Aug 15 15:15:42
Total Tasks: 1

Req[0]  TaskCount: 1  Partition: DEFAULT
Network: [NONE]  Memory >= 1996M  Disk >= 0  Swap >= 0
Opsys: [NONE]  Arch: [NONE]  Features: [NONE]
Dedicated Resources Per Task: PROCS: 1  MEM: 1996M
NodeCount: 1
Allocated Nodes:
[gnode21:1]


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

Reservation '575' (-00:37:53 -> 6:23:22:07  Duration: 7:00:00:00)
PE:  1.00  StartPriority:  1
```

调度器的基本调度流程



调度中的几个概念

回填：指在高优先级的大作业资源无法得到完全满足的情况下，将空闲的资源分配给一些较低优先级的小作业，让它们去运行。当然，小作业不能延迟大作业的运行，即必须在高优先级大作业获得其它需要的资源之前运行完成。WallTime是一个关键的参数。

预约：在某个时间段内为作业预留某些资源。只要该预约存在，就只有该作业可以在指定的时间段访问指定的资源。

抢占：在现有空闲资源无法满足的情况下，一些高优先级的作业可以强制“借用”正在运行的较低优先级的作业所占有的资源，使自己的资源需求得到满足，从而运行起来。待自身运行结束后，再将抢占的资源“送还”给原作业。

常见调度策略

- 优先级策略（Job Priority）

1.CPULOAD
按照节点负载反向排序。负载低的节点优先使用。
2.FIRSTAVAILABLE
按照节点顺序依次选择，名称靠前的节点优先使用。
3.LASTAVAILABLE
与FIRSTAVAILABLE相反。
4.PRIORITY
基于节点静态信息和运行时信息计算节点优先级，按照优先级从高到低选择。
5.MINRESOURCE
选择能满足作业需求的资源最少的一批节点运行作业。
6.MAXBALANCE
选择配置最接近的一批节点运行作业。
7.FASTEST
选择速度最快的一批节点运行作业。

级别，是的资源在不同用户（或用户组、队列、QOS、Account）间均衡的分配。

常见调度策略

优先级策略

调度器提供了一种**多种因素和多级权重组成**的优先级定义机制，如下所述：

1. 作业的优先级是六类组件贡献的优先级的总和；
2. 六类组件，既包括静态信息（如CRED，作业提交之后固定不变），又包括动态信息（如FS，即公平共享）；
3. 每一类组件和子组件均可以分别定义的权重；
4. 每一个子组件贡献的优先级是自身权重与自身取值的乘积；
5. 每一个组件的贡献的优先级是组件权重与子组件优先级之和的乘积。

常见调度策略

优先级策略计算公式：

组件	子组件
CRED（对象凭证）	USER
	GROUP
	ACCOUNT
	QOS
	CLASS
FS（公平共享）	FSUSER
	FSGROUP
	FSACCOUNT
	FSQOS
	FSCLASS
RES（资源请求）	NODE
	PROC
	MEM
	SWAP
	DISK
	PS
	PE
	WALLTIME
SERV（调度服务）	QUEUETIME
	XFACTOR
	BYPASS
TARGET（目标因子）	TARGETQUEUETIME
	TARGETFACTOR
USAGE（资源占用）	CONSUMED
	REMAINING
	PERCENT

以CRED类组件为例，其优先级公式为：

```
CredPriority = CREDWEIGHT * (
   USERWEIGHT * J->U->Priority +
   GROUPWEIGHT * J->G->Priority +
   ACCOUNTWEIGHT * J->A->Priority +
   QOSWEIGHT * J->Q->Priority +
   CLASSWEIGHT * J->C->Priority )
```

以FS类组件为例，其优先级公式为：

```
FSPriority = FSWEIGHT * (
   FSUSERWEIGHT *  (J->U->FSTarget - J->U->FSUsage)  +
   FSGROUPWEIGHT *  (J->G->FSTarget - J->G->FSUsage)  +
   FSCLASSWEIGHT *  (J->C->FSTarget - J->C->FSUsage)  +
   FSQOSWEIGHT *  (J->Q->FSTarget - J->Q->FSUsage)  +
   FSACCOUNTWEIGHT *  (J->A->FSTarget - J->A->FSUsage)
)
```

调度策略设置

优先级策略的设置：

CLASSCFG[<CLASSID>]	list of zero or more space delimited <ATTR>=<VALUE> pairs where <ATTR> is one of the following: **PRIORITY, FSTARGET, QLIST, QDEF, PLIST, PDEF, FLAGS**, or a fairness policy specification.	[NONE]	specifies class specific attributes. See the flag overview for a description of legal flag values. **NOTE:** Only available in Maui 3.0.7 and higher.	`CLASSCFG[batch] MAXJOB=50 QDEF=highprio` (up to 50 jobs submitted to the class `batch` will be allowed to execute simultaneously and will be assigned the QOS `highprio` by default.)

CLASSCFG[high] PRIORITY=11000

USERCFG[<USERID>]	list of zero or more space delimited <ATTR>=<VALUE> pairs where <ATTR> is one of the following: **PRIORITY, FSTARGET, QLIST, QDEF, PLIST, PDEF, FLAGS**, or a fairness policy specification.	[NONE]	specifies user specific attributes. See the flag overview for a description of legal flag values. **NOTE:** Only available in Maui 3.0.7 and higher.	`USERCFG[john] MAXJOB=50 QDEF=highprio` (up to 50 jobs submitted under the user ID `john` will be allowed to execute simultaneously and will be assigned the QOS `highprio` by default.)

USERCFG[liyuan] PRIORITY=500 MAXPROC=1000 MAXJOB=1000

调度策略设置

回填策略的设置：

BACKFILLDEPTH	<INTEGER>	0 (no limit)	specifies the number idle jobs to evaluate for backfill. The backfill algorithm will evaluate the top <X> priority jobs for scheduling. By default, all jobs are evaluated.	`BACKFILLDEPTH 128` (evaluate only the top 128 highest priority idle jobs for consideration for backfill)
BACKFILLMETRIC	one of the following **PROCS**, **PROCSECONDS**, **SECONDS**, **PE**, or **PESECONDS**	**PROCS**	specifies the criteria used by the backfill algorithm to determine the 'best' jobs to backfill. Only applicable when using BESTFIT or GREEDY backfill algorithms	`BACKFILLMETRIC PROCSECONDS`
BACKFILLPOLICY	one of the following: **FIRSTFIT**, **BESTFIT**, **GREEDY**, or **NONE**	**FIRSTFIT**	specifies what backfill algorithm will be used	`BACKFILLPOLICY BESTFIT`

抢占策略的设置：

PREEMPTIONPOLICY	one of the following: **REQUEUE**, **SUSPEND**, **CHECKPOINT**	**REQUEUE**	specifies how preemptible jobs will be preempted (Available in Maui 3.2.2 and higher)	`PREEMPTIONPOLICY CHECKPOINT` (jobs that are to be preempted will be checkpointed and restarted at a later time)

调度策略设置

抢占行为出现必须满足如下的条件：

1.策略配置上允许抢占（PREEMPTIONPOLICY）

目前支持的抢占方法包括：

a)SUSPEND挂起：将“被抢占者”作业的所有进程挂起，释放CPU等资源

b)CANCEL取消：将正在运行的“被抢占者”作业进程杀掉

c)REQUEUE重新入队：将正在运行的“被抢占者”作业进程杀掉，作业重新排队。

d)Checkpoint检查点：为正在运行的“被抢占者”作业生成检查点并杀掉作业进程。等有空闲的资源，“被　抢占者”作业从保存检查点恢复并继续运行。需要注意的是，Checkpoint需要应用支持。

2.抢占作业必须具备“抢占者（PREEMPTOR）”属性

“抢占者”属性，可以设置到队列（QDEF=preemptor），也可以设置到单个作业。

3.被抢占作业必须具备“被抢占者（PREMPTEE）”属性

与抢占者类似，“被抢占者”属性，可以设置到队列（QDEF=preemptee），也可以设置到单个作业。

4.“抢占者”作业的优先级要高于“被抢占者”作业的优先级。

具体见优先级策略。

调度策略设置

抢占策略的设置：

PREEMPTIONPOLICY	one of the following: **REQUEUE, SUSPEND, CHECKPOINT**	**REQUEUE**	specifies how preemptible jobs will be preempted (Available in Maui 3.2.2 and higher)	PREEMPTIONPOLICY CHECKPOINT (jobs that are to be preempted will be checkpointed and restarted at a later time)

Maui的配置

- Maui的配置参数都写在配置文件maui.cfg中，配置参数可以参考官方手册Maui Administrator's Guide

- 主要参数如下：
- vim maui.cfg

```
# 设置Maui服务器主机名
SERVERHOST            server
# 一级权限用户，拥有Maui所有控制权限，包括更改调度策略，更改作业属性
ADMIN1                root
# 二级权限用户，不能更改调度策略，但能更改作业属性
ADMIN2            zhang wang zhao
# 三级权限用户，只有查看权限，ALL表示所有账户
ADMIN3            ALL
```

Maui的配置（续）

定义资源管理器（Resource Manager），指定类型为PBS，以及Torque服务器主机名

```
RMCFG[0] TYPE=PBS HOST=server
RMPOLLINTERVAL    00:00:30
SERVERPORT        42559
SERVERMODE        NORMAL
```

日志设置

```
LOGFILE           maui.log
LOGFILEMAXSIZE    10000000（单位为bytes，也就是10M）
LOGLEVEL          3
```

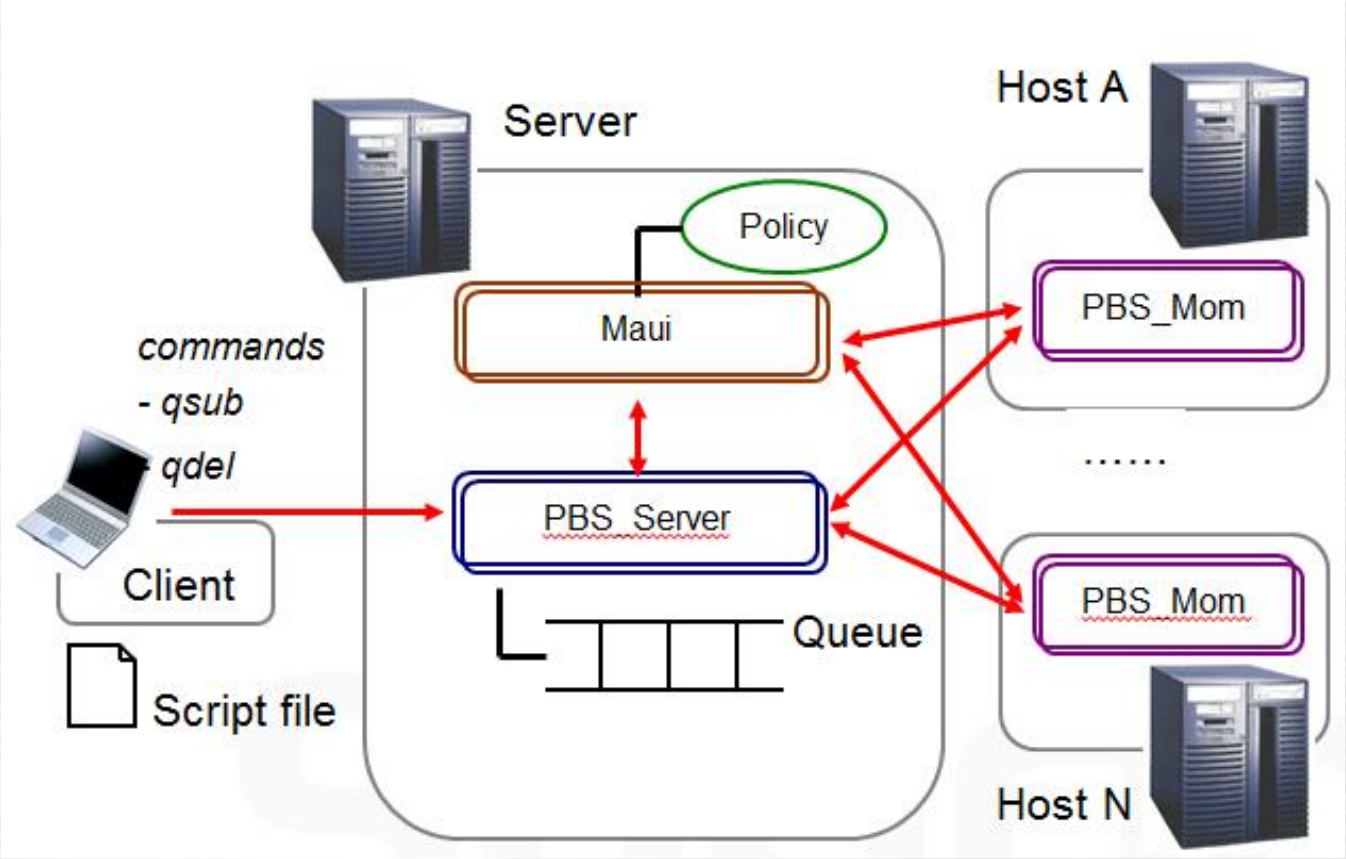
Maui的配置（续2）

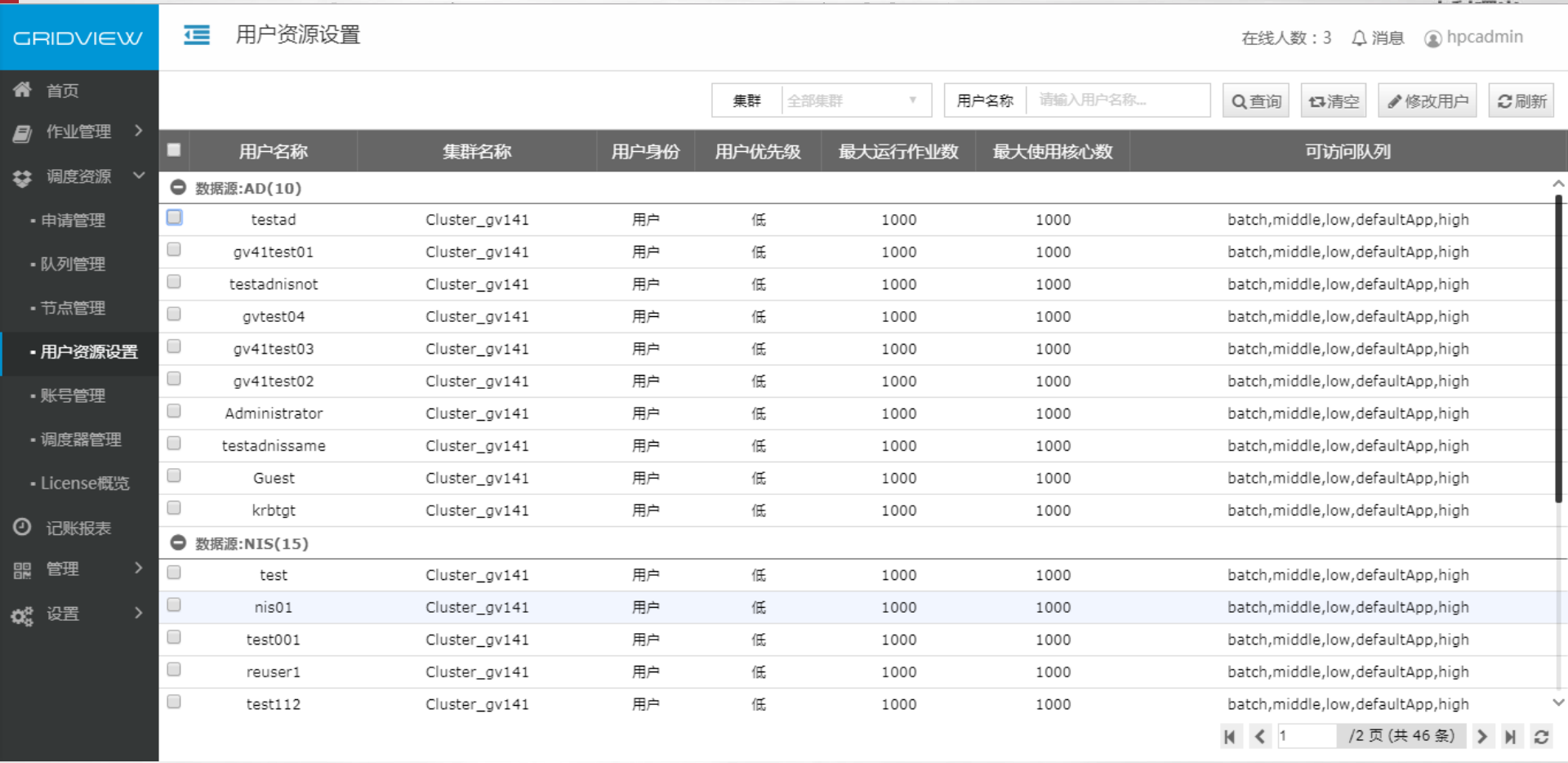
```
#设置Fair share策略
        #FSPOLICY           PSDEDICATED
        #FSDEPTH            7
        #FSINTERVAL         86400
        #FSDECAY            0.80
#设置回填（Backfill）策略
        BACKFILLPOLICY          FIRSTFIT
        RESERVATIONPOLICY   CURRENTHIGHEST
#节点分配策略
        #NODEALLOCATIONPOLICY  MINRESOURCE
        #NODEALLOCATIONPOLICY      CPULOAD
        NODEALLOCATIONPOLICY  FIRSTAVAILABLE
```

Maui的配置（续3）

```
# QOS配置
        QOSCFG[preemptor] QFLAGS=PREEMPTOR
        QOSCFG[preemptee] QFLAGS=PREEMPTEE
        CLASSCFG[high] PRIORITY=11000 QDEF=preemptor
        CLASSCFG[low] PRIORITY=6000 QDEF=preemptee
        CLASSCFG[middle] PRIORITY=9000 QDEF=preempte
#用户优先级设置
        USERCFG[root] MAXPROC=400 MAXNODE=100 MAXJOB=100
        USERCFG[test] PRIORITY=100 MAXPROC=200 MAXJOB=10
        USERCFG[DEFAULT] PRIORITY=100 MAXPROC=100 MAXJOB=4
```

Gridview作业调度组成结构




GRIDVIEW
用户资源设置
在线人数：3
消息
hpcadmin
首页
作业管理
调度资源
申请管理
队列管理
节点管理
用户资源设置
账号管理
调度器管理
License概览
记账报表
管理
设置
集群
全部集群
用户名称
请输入用户名称...
查询
清空
修改用户
刷新
用户名称
集群名称
用户身份
用户优先级
最大运行作业数
最大使用核心数
可访问队列
数据源:AD(10)
testad Cluster_gv141 用户 低 1000 1000 batch,middle,low,defaultApp,high
gv41test01 Cluster_gv141 用户 低 1000 1000 batch,middle,low,defaultApp,high
testadnisnot Cluster_gv141 用户 低 1000 1000 batch,middle,low,defaultApp,high
gvtest04 Cluster_gv141 用户 低 1000 1000 batch,middle,low,defaultApp,high
gv41test03 Cluster_gv141 用户 低 1000 1000 batch,middle,low,defaultApp,high
gv41test02 Cluster_gv141 用户 低 1000 1000 batch,middle,low,defaultApp,high
Administrator Cluster_gv141 用户 低 1000 1000 batch,middle,low,defaultApp,high
testadnissame Cluster_gv141 用户 低 1000 1000 batch,middle,low,defaultApp,high
Guest Cluster_gv141 用户 低 1000 1000 batch,middle,low,defaultApp,high
krbtgt Cluster_gv141 用户 低 1000 1000 batch,middle,low,defaultApp,high
数据源:NIS(15)
test Cluster_gv141 用户 低 1000 1000 batch,middle,low,defaultApp,high
nis01 Cluster_gv141 用户 低 1000 1000 batch,middle,low,defaultApp,high
test001 Cluster_gv141 用户 低 1000 1000 batch,middle,low,defaultApp,high
reuser1 Cluster_gv141 用户 低 1000 1000 batch,middle,low,defaultApp,high
test112 Cluster_gv141 用户 低 1000 1000 batch,middle,low,defaultApp,high
1
/2 页 (共 46 条)

携手成就梦想


GRIDVIEW
队列管理
在线人数：2
消息 8
hpcadmin
首页
作业管理
调度资源
申请管理
队列管理
节点管理
用户资源设置
账号管理
调度器管理
License概览
记账报表
管理
设置
集群
全部集群
添加队列
删除队列
FLUENT
运行作业数：0
队列优先级：中
默认队列：是
batch
运行作业数：0
队列优先级：低
默认队列：否
cfx
运行作业数：0
队列优先级：低
默认队列：否
eeeeee
运行作业数：0
队列优先级：低
默认队列：否
ffff
运行作业数：0
队列优先级：低
默认队列：否
firefox
运行作业数：0
队列优先级：低
默认队列：否
high
运行作业数：0
队列优先级：高
默认队列：否
low
运行作业数：0
队列优先级：低
默认队列：否
middle
运行作业数：0
队列优先级：中
默认队列：否
队列信息
用户最大运行数
请输入用户最大运行作业数(默认无限制)
队列最大运行数
200
*最长运行时间
24:00:00
费率
请输入费率,0到10000之间整数或小数(默认为1)
权限信息
队列优先级
低
Qos级别
[空]
控制信息
可访问节点
请选择队列可用节点(默认全选)
默认队列
否
启用队列
启用
用户访问控制
禁用
可访问用户
请选择队列可访问用户(默认全选)

携手成就梦想


GRIDVIEW
调度器管理
在线人数：2
消息
hpcadmin
首页
作业管理
调度资源
申请管理
队列管理
节点管理
用户资源设置
账号管理
调度器管理
License概览
记账报表
管理
设置
添加调度器
修改调度器
删除调度器
重启调度器
更多配置
调度器名称
调度器类型
IP地址
端口
管理器主目录
调度器配置目录
调度器可用标识
最后修改时间
描述信息
Cluster_gv141
PBS
10.0.35.141
9091
/opt/autogridview/pbs...
/opt/autogridview/pbs/di...
可用
2018-11-15 03:08:18
OK
调度器策略配置
设置回填
回填算法
FirstFit
回填度量
PROCS
回填深度
0
预约深度
1
设置抢占
抢占策略
SuSpend
确定
关闭
/1 页 (共 1 条)

中科曙光
Sugon

谢谢！

IT基础设施及方案的领导者
数据中国百城百行的发起者
中科院产业化联盟的推动者
安全可控信息系统的践行者

携手成就梦想