CLUSTER® RESOURCES INC.

QoS Enforcement

QoS Enforcement

- QoS Features
- Creating a QoS
- Configuring a QoS
- Enforcing a QoS

http://www.clusterresources.com/products/mwm/docs/7.3qos.shtml

QoS Features

- Usage Limits and Overrides
- Service Access Thresholds
- QoS Specific Charging
- User Reservations
- Specified Job Deadline Enforcement
- Service Level Enforcement

http://clusterresources.com/moabdocs/7.3qos.shtml

Differentiate Available Levels of Service

- QoS's – what is the job worth to the user
- Control resource access
- Control responsiveness
- Control costs
- Control service access

http://clusterresources.com/moabdocs/7.3qos.shtml

Creating a QoS

- QoS is created by referencing it in the moab.cfg
- QoS is named by an arbitrary string

Format:

```
QOSCFG[<QOSNAME>]   <ATTRIBUTE>
```

```
# moab.cfg
QOSCFG[hiprio]   PRIORITY=10000
```

http://www.clusterresources.com/products/mwm/docs/a.fparameters.shtml#qoscfg

Configuring a QoS

- Preemption
  - QOSCFG[preempt]   FLAGS=PREEMPTOR
- Charging
  - QOSCFG[hi] DEDRESCOST=4.0
  - QOSCFG[lo] DEDRESCOST=1.0
- Thresholds
  - QOSCFG[hi] PREEMPTQTTHRESHOLD=30:00
- Special Flags
  - QOSCFG[dl] FLAGS=DEADLINE

Enforcing a QoS

- QLIST, QDEF, and MEMBERULIST
- Examples:
  - USERCFG[DEFAULT] QLIST=low,med
  - USERCFG[jon] QLIST=hi,med,low
  - QOSCFG[hi] MEMBERULIST=jon,jill,jack
  - ACCOUNTCFG[projA] QDEF=hi
  - CLASSCFG[batch] QLIST=hi,med,low

Special QoS Features

- QoS Based Preemption
- Service Access Thresholds
- Override limits
- User Reservations
- Job Deadlines
- Service Deadlines

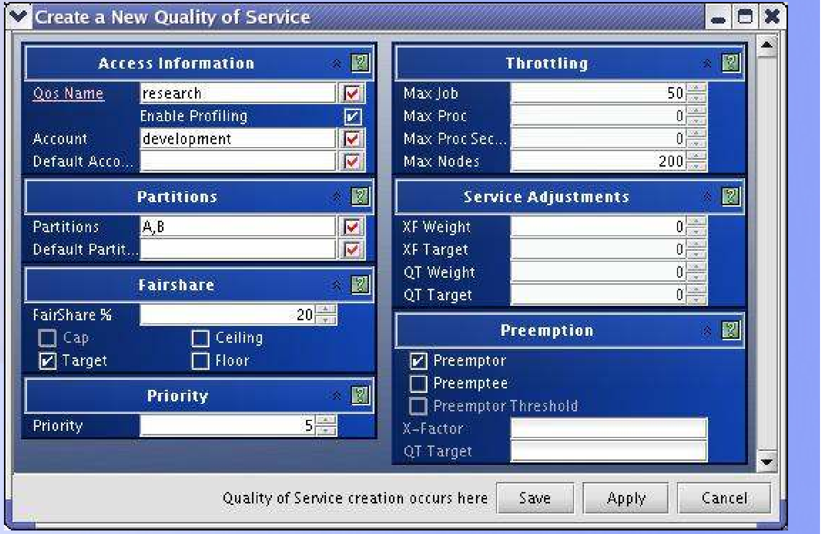
QOS Based Preemption

- Preemption **only** occurs when the following 3 conditions are satisfied:
- The *preemptor* job has the **PREEMPTOR** attribute set
- The *preemptee* job has the **PREEMPTEE** attribute set
- The *preemptor* job has a higher priority than the *preemptee* job

```
PREEMPTPOLICY REQUEUE
# enable qos priority to make preemptors higher
priority than preemptees

QOSWEIGHT 1
QOSCFG[high] QFLAGS=PREEMPTOR PRIORITY=1000
QOSCFG[med]
QOSCFG[low] QFLAGS=PREEMPTEE

 # associate class 'special' with QOS high
CLASSCFG[special] QDEF=high&
```



Create a New Quality of Service

Access Information

Qos Name: research ☑
Enable Profiling ☑
Account: development ☑
Default Acco...: ☑

Partitions

Partitions: A,B ☑
Default Partit...: ☑

Fairshare

FairShare %: 20
□ Cap □ Ceiling
☑ Target □ Floor

Priority

Priority: 5

Throttling

Max Job: 50
Max Proc: 0
Max Proc Sec...: 0
Max Nodes: 200

Service Adjustments

XF Weight: 0
XF Target: 0
QT Weight: 0
QT Target: 0

Preemption

☑ Preemptor
□ Preemptee
□ Preemptor Threshold
X-Factor:
QT Target:

Quality of Service creation occurs here

Preemption based Backfill

- The PREEMPT backfill policy allows the scheduler to start backfill jobs even if required walltime is not available.
- If the job runs too long and interferes with another job which was guaranteed a particular timeslot, the backfill job is preempted and the priority job is allowed to run.
- When another potential timeslot becomes available, the preempted backfill job will again be optimistically executed.

Service Access Thresholds

Access to resources can be conditionally based on current metrics of an idle job.

- **PREEMPTQTTHRESHOLD** a job with this QoS becomes a *preemptor* if the specified queuetime threshold is reached
- **RSVQTTHRESHOLD** a job with this QoS becomes can create a job reservation to guarantee resource access if the specified queuetime threshold is reached
- **ACLQTTHRESHOLD** a job with this QoS can access reservations with a corresponding QoS ACL only if the specified queuetime threshold is reached
- **TRIGGERQTTHRESHOLD** if a job with this QoS fails to run before this threshold is reached, any failure triggers associated with this QoS will fire

Override Limits

- USERCFG[DEFAULT] MAXJOB=4
- QOSCFG[hi] OMAXJOB=24
- QOSCFG[hi] OMAXPROC=50
- ACCOUNTCFG[projA] MAXPROC=25

Personal User Reservations

- Allow end users to create their own reservations – within limits
- Limit size of individual reservations
  - RMAXDURATION, RMAXPROC, RMAXPS, RMAXCOUNT
- Limit total size of all individual reservations
  - RMAXTOTALDURATION, RMAXTOTALPROC, RMAXTOTALPS
- *OR:*
- Personal Reservation Sandbox
  - SRCFG[prsv] FLAGS=ALLOWPRSV

Personal Reservation Examples

- Reservation Limits
  - QOSCFG[prsv] QFLAGS=ENABLEUSERRSV
  - QOSCFG[prsv] RMAXPROC=5 RMAXDURATION=2:00:00
- Personal Reservation Sandbox
  - QOSCFG[prsv] QFLAGS=ENABLEUSERRSV
  - SRCFG[prsv] HOSTLIST=r:n[01-20] FLAGS=ALLOWPRSV
  - SRCFG[prsv] PERIOD=DAY DAYS=SAT,SUN

Specified Job Deadlines

- Deadline specifies time by which job must complete
- Deadline can be absolute or relative
- Deadline support enabled utilizing QoS
- Specified on a per job basis
- Moab reserves resources to guarantee deadline enforcement
- If deadline cannot be immediately guaranteed, a configurable action can be taken
  - Retry, Ignore, Cancel

http://clusterresources.com/moabdocs/11.9jobdeadlines.shtml

Specified Job Deadlines

```
#moab.cfg

QOSCFG[high]      QFLAGS=DEADLINE
USERCFG[DEFAULT]  QDEF=high
```

```
> qsub –l deadline=2:00:00,nodes=1,walltime=01:00:00 myjob.sh
```

- a one hour job requesting QoS special will have a completion time deadline set to 3 hours after the job's submission time.

http://clusterresources.com/moabdocs/11.9jobdeadlines.shtml

Service-level Deadlines

- QoS's may be set up with both the DEADLINE flag and a response time target. For job's requesting these qualities of service, Moab will identify and set job deadlines so as to satisfy the corresponding response time targets.
- For example, consider the configuration below which sets a queue time response target of 1 hour:

```
#moab.cfg

QOSCFG[special] QFLAGS=DEADLINE QTTARGET=1:00:00
ACCTCFG[ajax]   QLIST=special,fast,base
```

- Given this configuration, a 2 hour job requesting QoS special will have a completion time deadline set to 3 hours after the job's submission time.

http://clusterresources.com/moabdocs/11.9jobdeadlines.shtml