Moab

How can I configure Moab to be aware of my file-system failures?

Issue: How can I configure Moab to be aware of my file-system failures?

Affected Versions: All

Symptom: When a filesystme goes down Moab will continue to start jobs only to have them fail some minuits later.

Solution:

Moab is not aware of a mounted filesystem by default. Historically customers have used GRES to schedule filesystems. In other words, jobs would request a particular filesystem (note this is for a filesystem not disk space) using a generic resource. In many cases this is automatically added by administrators to all jobs. Then, if the filesystem goes down then the administrators configure the gres down to 0 so that any jobs requesting the filesystem won't run.

Using gres to do this is not the best way to accomplish this. While gres as a concept is well understood and supported it is overkill for this situation. The reason it is overkill is because generic resources are a consumed resource and in this particular use case we don't need to track consumption at all. Filesystems are more like scheduler-wide features (as opposed to node features).
For these reasons we have implemented the following:

GRESCFG[<gres>] FEATUREGRES=TRUE

This tells Moab to treat the gres as a scheduler-wide feature, rather than a normal gres. Jobs are still submitted using the same gres syntax.
What follows is a walkthrough on how to use this new feature.
I have 2 filesystem, lustrA and lustrB. I have these configured in my moab.cfg file as follows:

NODECFG[GLOBAL] GRES=lustrA:0
NODECFG[GLOBAL] GRES=lustrB:10000

Currently lustrA is having problems so it is set to 0 and lustrB is available and has 10000 units available (this is set high so that we never hit the limits).
Now I want to use the new feature so I'm going to add 2 more lines:

GRESCFG[lustrA] FEATUREGRES=TRUE
GRESCFG[lustrB] FEATUREGRES=TRUE

This tells Moab to treat these 2 features as the new "FeatureGRes". I can check to make sure Moab read this in correctly using the "mdiag -S -v" command:
$ mdiag -S -v
Moab Server 'Moab' running on ioty:40559 (Mode: NORMAL)
Version: 6.1.7 (revision 1, changeset

551f26a32b9e5b1794aa14eea7bb36b916b1b08c)
.....
Scheduler FeatureGRes: lustrA:off,lustrB:on
.....
Notice that lustrA is off, this is because the count was specified at 0. 0 means off. Any positive number or no number at all is interpreted as "on".
Once Moab has started you can modify whether a featuregres by running the "mschedctl -m" command:
mschedctl -m sched featuregres:lustrA=on

INFO: FeatureGRes 'lustrA' turned on

With that command I just turned on lustrA. I can verify it is turned on using "mdiag -S -v":
mdiag -S -v
.....
Scheduler FeatureGRes: lustrA:on,lustrB:on
.....

I can also turn a feature off using the same command:
$ mschedctl -m sched featuregres:lustrB=off

INFO: FeatureGRes 'lustrB' turned off

mdiag -S -v
.....
Scheduler FeatureGRes: lustrA:on,lustrB:off
.....

If Moab is restarted it will NOT checkpoint the state of these featuregres. It will simply read the moab.cfg file and use that to determine whether the featuregres is "on" or "off".
Jobs are submitted as normal, requesting gres of type lustrA and lustrB. Counts are ignored for featuregres and are just considered on or off.
$ echo sleep 600 | msub -l nodes=1,walltime=600,gres=lustrA

1012
$ checkjob 1012
job 1012

AName: STDIN
State: Running
.....
StartTime: Tue Apr 3 15:33:28
Feature GRes: lustrA
Total Requested Tasks: 1

Req[0] TaskCount: 1 Partition: trq

Notice the "Feature GRes: lustrA" in the output that tells us if it was correctly understood. If I ask for lustrB (which is currently off) I will see the following:
$ echo sleep 600 | msub -l nodes=1,walltime=600,gres=lustrB

1013
$ checkjob -v 1013
job 1013 (RM job '1013.ioty')

AName: STDIN
State: Idle
.....
Feature GRes: lustrB
Total Requested Tasks: 1

Req[0] TaskCount: 1 Partition: ALL
.....

BLOCK MSG: requested feature gres 'lustrB' is off (recorded at last scheduling iteration)

Notice that the job is Idle and there is a message that the job is blocked because "lustrB is off".

A few things to note:
1) If you are running in a grid make sure that the FEATUREGRES=TRUE is set on all members of the grid. Do not mix.
2) You can safely upgrade an existing cluster to use this feature while jobs are running. If you are in a grid you will need to upgrade all clusters at the same time.
3) As mentioned above, FeatureGRes is NOT checkpointed, Moab will use the moab.cfg file each time it starts to determine the state of the FeatureGRes.

Unique solution ID: #1033
Author: Jason Booth
Last update: 2015-06-11 18:12