Moab
Native Resource Manager API examples and explanations

How to configure in the moab.cfg

Here's an example of a native resource manager configuration in the moab.cfg. Note that '//$HOME' resolves to the moab home directory, which is usually '/opt/moab'.

So $HOME/nativerm/queryresource.sh resolves to /opt/moab/nativerm/queryresource.sh

moab.cfg

RMCFG[nativerm] TYPE=NATIVE
RMCFG[nativerm] CLUSTERQUERYURL=exec://$HOME/nativerm/queryresource.sh
RMCFG[nativerm] WORKLOADQUERYURL=exec://$HOME/nativerm/queryworkload.sh
RMCFG[nativerm] JOBSUBMITURL=exec://$HOME/nativerm/submit.pl
RMCFG[nativerm] JOBSTARTURL=exec://$HOME/nativerm/jobstart.sh
RMCFG[nativerm] JOBCANCELURL=exec://$HOME/nativerm/jobcancel.sh

RMCFG[nativerm] JOBMODIFYURL=exec://$HOME/nativerm/jobmodify.sh
RMCFG[nativerm] JOBSUSPENDURL=exec://$HOME/nativerm/jobsuspend.sh
RMCFG[nativerm] RESOURCECREATEURL=exec://$HOME/nativerm/resourcecreate.sh
RMCFG[nativerm] SYSTEMMODIFYURL=exec://$HOME/nativerm/systemmodify.sh
RMCFG[nativerm] JOBRESUMEURL=exec://$HOME/nativerm/jobresume.sh

##

CLUSTERQUERYURL

This might be one of the most important native resource manager APIs since it is how Moab leans about the nodes in the cluster.

moab.cfg

RMCFG[nativerm] CLUSTERQUERYURL=exec://$HOME/nativerm/queryresource.sh

#########################

When Moab executes the CLUSTERQUERYURL script it expects the script to respond with a list of nodes, what state each nodes is in, and other properties about the node. Each line is space delimited and represents a single node. The first token on

each line is the name of the node. Here is a simple example of a line that CLUSTERQUERYURL could return

example output

node-001 CPROC=8 CDISK=200000 CMEMORY=32768 CPULOAD=0.1 CSWAP=65536 NAME=node-001 OS=Rhel7 PARTITION=nativerm STATE=Idle POWER=true

############################

Notice that CPROC means configured processors. This means if the compute node has 8 processors then CPROC=8. CMEMORY stands for configured memory, which should be the amount (in megabytes) of memory (RAM) in the compute node. Likewise CDISK and CSWAP represent disk space and swap space respectively in megabytes.

STATE means the state of the node with respect to whether it can accept new HPC jobs. The most common values for STATE are Idle, Running, Busy, and Down.

Idle: The node is ready to run jobs but currently is not running any.

Running: Node is running some jobs and will accept additional jobs

Busy: The node is running some jobs and will not accept additional jobs

Down: Problems have been detected. Node is incapable of running jobs.

The full list of attributes/value pairs for CLUSTERUERYURL scripts are documented in detail in in O.1.1 Query Resources Data Format of the Moab Workload Manager documention.
See
http://docs.adaptivecomputing.com/9-1-3/MWM/moab.htm#topics/moabWorkloadManager/appendices/wWiki/wikiinterface.html

Parameters Moab passes: None
Example Output:

example script output

node-001 APROC=0 ARCH=x86_64 ARES=alpha:4,bravo:4,charlie:4 CCLASS=[batch:2][batch2:2][batch3:2][batch4:2] CDISK=200000 CMEMORY=2048 CPROC=4 CPULOAD=0.1 CRES=alpha:4,bravo:4,charlie:4 CSWAP=2048 FEATURE=feature1:feature2:feature3 GMETRIC[BWH]=73.07 NAME=node-001 OS=Rhel6 PARTITION=part1 RACK=1 SLOT=1 STATE=Running

node-002 APROC=0 ARCH=x86_64 ARES=alpha:4,bravo:4,charlie:4 CCLASS=[batch:2][batch2:2][batch3:2][batch4:2] CDISK=200000 CMEMORY=2048 CPROC=4 CPULOAD=0.1 CRES=alpha:4,bravo:4,charlie:4 CSWAP=2048 FEATURE=feature1:feature2:feature3 GMETRIC[BWH]=74.09 NAME=node-002 OS=Rhel6 PARTITION=part1 RACK=1 SLOT=2 STATE=Running
node-003 APROC=0 ARCH=x86_64 ARES=alpha:4,bravo:4,charlie:4 CCLASS=[batch:2][batch2:2][batch3:2][batch4:2] CDISK=200000 CMEMORY=2048 CPROC=4 CPULOAD=0.1 CRES=alpha:4,bravo:4,charlie:4 CSWAP=2048 FEATURE=feature1:feature2:feature3 GMETRIC[BWH]=74.47 NAME=node-003 OS=Rhel6 PARTITION=part1 RACK=1 SLOT=3 STATE=Running

##############################

WORKLOADQUERYURL

Moab executes the WORKLOADQUERYURL script to find out about active, idle, and recently completed jobs on the resource manager. To configure it put the following in the moab.cfg

moab.cfg
RMCFG[nativerm]
WORKLOADQUERYURL=exec://$HOME/nativerm/queryworkload.sh
########################

When Moab executes the WORKLOADQUERYURL script it expects the script to respond with a list of jobs, what state each job is in, and other properties about the job. Each line is space delimited and represents a single job. The first token on each line is the ID of the node.

Let's say you submit three jobs like the following:
bash terminal
[hgranger]$ msub -l nodes=5:ppn=3 -l opsys=Rhel6 -l mem=512mb -l walltime=60 ~/myjobscript.sh
Moab.1

[hgranger]$ msub -l nodes=5:ppn=3 -l opsys=Rhel6 -l mem=512mb -l walltime=60 ~/myjobscript.sh
Moab.2

[hgranger]$ msub -l nodes=5:ppn=3 -l opsys=Rhel6 -l mem=512mb -l walltime=60 ~/myjobscript.sh
Moab.3
##################################

Let's say that Moab.1 has already completed, Moab.2 is currently running, and Moab.3 is idle waiting for a chance to run. Moab.4 was cancelled by Moab using the JOBCANCELURL. When Moab calls the WORKLOADQUERYURL script it would expect

output like the following:

example script output
Moab.1 COMMENT="SID=Moab?SJID=Moab.1?SRMJID=Moab.1?"
COMPLETETIME=1584648056 EXEC=/opt/moab/spool/moab.job.cGhrq8
EXITCODE=0 GNAME=hgranger IWD=/home/hgranger JNAME=myjobscript.sh
QUEUETIME=1584647995 STARTTIME=1584647995 STATE=Completed TASKLIST=
node-007,node-007,node-007,node-006,node-006,node-006,node-005,node-005,no
de-005,node-004,node-004,node-004,node-003,node-003,node-003 TASKS=15
TEMPLATE=DEFAULT UNAME=hgranger WCLIMIT=60

Moab.2 COMMENT="SID=Moab?SJID=Moab.2?SRMJID=Moab.2?"
EXEC=/opt/moab/spool/moab.job.xTOzfl GNAME=hgranger IWD=/home/hgranger
JNAME=myjobscript.sh QUEUETIME=1584647996 STARTTIME=1584648056
STATE=Running TASKLIST=node-007,node-007,node-007,node-006,node-006,node-
006,node-005,node-005,node-005,node-004,node-004,node-004,node-003,node-00
3,node-003 TASKS=15 TEMPLATE=DEFAULT UNAME=hgranger WCLIMIT=60

Moab.3 COMMENT="SID=Moab?SJID=Moab.3?SRMJID=Moab.3?"
EXEC=/opt/moab/spool/moab.job.iSEgA7 GNAME=hgranger IWD=/home/hgranger
JNAME=myjobscript.sh QUEUETIME=1584647997 STATE=Idle TASKS=15
TEMPLATE=DEFAULT UNAME=hgranger WCLIMIT=60

Moab.4 COMMENT="SID=Moab?SJID=Moab.4?SRMJID=Moab.4?"
COMPLETETIME=1584729915 EXEC=/opt/moab/spool/moab.job.H9EnjX
EXITCODE=-1 GNAME=hgranger IWD=/home/hgranger JNAME=mysimplejob
QUEUETIME=1584729895 STARTTIME=1 584729896 STATE=Removed
TASKLIST=node-007,node-007,node-007,node-006,node-006,node-006 TASKS=6
TEMPLATE=DEFAULT UNAME=hgranger WCLIMIT=60

##############################

Notice that for completed jobs moab passes the EXITCODE and COMPLETETIME. However these are not passed for active and idle jobs. Also be aware that the resource manager should stop reporting completed jobs after Moab has had a chance to detect that the job has completed. There have been cases where if a resource manager reports jobs that have completed longer than the moab.cfg parameter JOBCPURGETIME (which is 10 minutes) then Moab gets confused and thinks the completed job is actually a new job.

The complete list of attribute value pairs is documented in section O.1.2 Query Workload Data Format of the Moab Workload Manager documentation.
See
http://docs.adaptivecomputing.com/9-1-3/MWM/moab.htm#topics/moabWorkloadManager/appendices/wWiki/wikiinterface.html

Parameters Moab passes: None

Example Output: See above

JOBSUBMITURL

Moab calls the JOBSUBMITURL script to pass a newly submitted job to the resource manager but NOT start it. Moab will pass as parameters to the script the username, the wallclock limit, resource requirements, and the name of a file containing an annotated version of the job script.

To configure it in your moab.cfg

```
### moab.cfg ##############
RMCFG[nativerm] JOBSUBMITURL=exec://$HOME/nativerm/submit.sh
############################
```

Let's say you have a job script that looks like this

```
### bash terminal #########
[hgranger]$ cat myjobscript.sh
#!/bin/bash
echo here is a simple job script
sleep 60
############################
```

And you submit the job to Moab with msub like this

```
### bash terminal #########
[hgranger]$ msub -l nodes=2:ppn=3 -l opsys=Rhel6 -l mem=512mb -N
mysimplejob -l walltime=60 ~/myjobscript.sh
Moab.2
############################
```

Moab will call /opt/moab/nativerm/submit.sh with the following parameters

```
### Parameters Passed to Script #######
UNAME=hgranger NAME=Moab.2 GNAME=hgranger WCLIMIT=60 TASKS=6
JNAME=mysimplejob TEMPLATE=DEFAULT
COMMENT="SID=Moab?SJID=Moab.2?SRMJID=Moab.2?" IWD=/home/hgranger
EXEC=/opt/moab/spool/moab.job.nLbIEX
#####################################
```

At the same time that Moab calls the script Moab will create some new files in /opt/moab/spool.

```
#####################################
```

[root]# cd /opt/moab/spool
[root]# ls -l
-rw------- 1 hgranger hgranger 0 Mar 26 14:56 Moab.2.enWPtfY
-rw------- 1 hgranger hgranger 0 Mar 26 14:56 Moab.2.o8dvFGj
-rw------- 1 root root 1084 Mar 26 14:56 Moab.2.cp
-r-x---- Moab.2.cp-- 1 hgranger hgranger 156 Mar 26 14:56 moab.job.nLbIEX
drwxr-xr-x 2 root root 49 Mar 26 14:53 insight_store
##################################

The hash values at the ends of the file names may change but they will be of the form Moab.<integer>.<hash>, Moab.<integer>.cp, and moab.job.<hash>.

The file that is probably most important is the moab.job.nLbIEX. This will actually be passed to the JOBSUBMITURL as a parameter. It will look like EXEC=/opt/moab/spool/moab.job.nLbIEX Moab will cause this file to appear in in /opt/moab/spool directory. The name will be different for each new job but it seems to be prefixed with moab.job.____ Here's what the contents of that file will look like given the script and the msub above.

bash terminal
[root]# cat /opt/moab/spool/moab.job.nLbIEX
#!/bin/bash
#PBS -l nodes=2:ppn=3
#PBS -l opsys=Rhel6
#PBS -l mem=512mb
#PBS -N mysimplejob
#PBS -l walltime=60
echo here is a simple job script
sleep 60
##############################

The difficult thing about the moab.job.nLbIEX file is that Moab appears to remove it after the JOBSUBMITURL script returns. Again... this is important. It seems that Moab will delete this spool file as soon as the JOBSUBMITURL script returns. So it appears to be up to the JOBSUBMITURL script to copy its contents and do something with it. So if you need to pass the job script to the resource manager getting it out of moab.job.nLbIEX before it disappears is important.

Moab.<integer>.cp or, in this case Moab.2.cp, will be an XML file. Here is an example of what it looks like

##############################
[root]# cd /opt/moab/spool
[root]# cat Moab.2.cp
JOB Moab.2 1585249158 <job AWDuration="61"
CmdFile="/opt/moab/spool/moab.job.OS7S7E" CompletionCode="0"
CompletionTime="1585249095" Ctime="1585248974" DRM="nativerm"
DRMJID="Moab.2" EEDuration="0" EState="Completed"
EffPAL="[nativerm][part1][part2][part3]" Flags="GLOBALQUEUE" GJID="Moab.2"

Group="hgranger" HopCount="0" IWD="/home/hgranger" JobID="Moab.2" JobName="mysimplejob" JobRank="0" LastChargeTime="1585249095" PAL="nativerm,part1,part2,part3" RM="internal" RMXString="SID=Moab;SJID=Moab.2;SRMJID=Moab.2;" ReqAWDuration="60" SID="Moab" SRMJID="Moab.2" StartCount="1" StartPriority="0" StartTime="1585249034" StatMSUtl="0.000" StatPSDed="364.200" StatPSUtl="12.140" State="Completed" SubmissionTime="1585248974" SubmitHost="nativerm" SubmitLanguage="PBS" SuspendDuration="0" TemplateSetList="DEFAULT" TermTime="2140000000" UMask="2" User="hgranger"><req AllocNodeList="node-007:3,node-006:3" AllocPartition="part1" Index="0" NCReqMin="2" RMName="nativerm" ReqMemPerTask="85" ReqNodeAMem=">=0" ReqNodeASwap=">=0" ReqNodeMem=">=0" ReqNodeProc=">=0" ReqNodeSwap=">=0" ReqOpsys="Rhel6" ReqPartition="part1" ReqProcPerTask="1" TCReqMin="6*" TPN="3"/></job>
##############################

The other files that appear in the /opt/moab/spool directory do not seem to be very helpful for the JOBSUBMITURL. Moab.2.o8dvFGj and Moab.2.o8dvFGj never appear to have any data in them. insight_store is only used for Moab's communication to Insight and can be ignored in the JOBSUBMITURL

The JOBSUBMITURL does not appear to need to return any output to Moab.

Parameters Moab passes: Key value pairs like UNAME=hgranger NAME=Moab.2 WCLIMIT=60 ...
Example Output: Nothing. This script does not appear to need to return output to Moab

JOBSTARTURL

Moab calls the JOBSTARTURL script to tell the resource mananger to start the job on a set of nodes.

To configure it in your moab.cfg
moab.cfg
RMCFG[nativerm] JOBSTARTURL=exec://$HOME/nativerm/jobstart.sh
##########################

When Moab wants to start a job it will call this script. Let's say Moab wants to start a job Moab.3 on node-004 and node-005. Let's also say the 2 of the processors on node-004 should be dedicated to the job. Likewise 3 of the processors on node-005 should be dedicated to the job. The user show submitted the job is hgranger. Moab will call /opt/moab/nativerm/jobstart.sh with the following parameters

Parameters Passed to Script
Moab.3 node-005,node-005,node-005,node-004,node-004 hgranger

##

Parameters Moab passes: <jobID> <nodeId><nodId><nodeId>... <username>
Example Output: Nothing. This script does not appear to need to return output to Moab

JOBCANCELURL

Moab calls the JOBCANCELURL script to tell the resource manager to cancel a job.

To configure it in your moab.cfg

moab.cfg
RMCFG[nativerm] JOBCANCELURL=exec://$HOME/nativerm/jobcancel.sh
##########################

When Moab wants to cancel a job it will call this script and pass it as an argument the id of the job. For example let's say Moab wants to cancel the job with the id Moab.1. It will call /opt/moab/nativerm/jobcancel.sh with the following parameter.

Parameters Passed to Script
Moab.1
#####################################

Note what when a job gets cancelled the resource manager code should continue to return the job in the WORKLOADQUERYURL at least until Moab next calls the WORKLOADQUERYURL. The WORKLOADQUERYURL should add an attribute with the completion time of the job. (e.g. COMPLETETIME=1584730152) If the job was running when it got cancelled it is probably a good idea to return a non-zero exit code in the WORKLOADQUERYURL (e.g. EXITCODE=-1) so Moab knows the job did not complete successfully. Also a cancelled job (at least in the native resource manager simulation code) returns STATE=Removed in the WORKLOADQUERYURL rather than STATE=COMPLETED. See the example output in WORKLOADQUERYURL above for more details

Parameters Moab passes: <jobID>
Example Output: Nothing. This script does not appear to need to return output to Moab

Unique solution ID: #1225
Author: Nate Seeley
Last update: 2020-03-30 19:49