

MCM

How can I troubleshoot reports in MCM, and from where does MCM get its data?

Question: How can I troubleshoot reports in MCM, and where does MCM get its data?

Affected Versions: All

Issue: When running reports, it can be helpful to know where MCM gathers its data, and know how to validate the information MCM displays.

Solution:

Be advised that Adaptive has deprecated Moab Cluster Manager, aka MCM. Support for MCM ended as of the release of the Adaptive HPC Suite version 9.1.1. Adaptive Computing will no longer do code fixes to any parts of the HPC Suite in order to fix problems involving the use of MCM. We recommend upgrading to the new reporting framework. If you find that the new framework lack specific capabilities enjoyed in MCM, please let us know, and we'll be more than happy to work with you to create the reports that will extract the data you need, and present it a useful way.

MCM is a tool to interface with Moab. It has many features for configuring and managing policies credentials, and reporting. This article will discuss how to troubleshoot how MCM interfaces with Moab.

We will look at:

- mcredctl XML output
- Verifying that the information being returned correctly displays in the graphs.
- Note: currently some issues exist with the Moab statistics; at a minimum, we recommend independent sanity checks to verify accuracy of the data.

mcredctl example:

MCM sends the mcredctl command to moab to query statistical data. To obtain the specific command and XML, follow these steps.

- `tail -f $LOCATION_OF_MCM/logs/mcm.log`
- Open MCM and execute the desired action. In this case, it was a report on utilization.
- After you execute the needed command look at the output of mcm.log for the associated mcredctl. In my case it was

```
mcredctl -q profile group --format=xml --timeout=00:10:00 -o  
time:1301641200,1304229599
```

MCM

- Run the command on the system to verify that Moab returns correct data.

```
ssh eval1@pukan -C "mcredctl -q profile group --format=xml --timeout=00:10:00 -o time:1301641200,1304229599" | tidy -i -xml > /tmp/tidy_out.txt
```

Note: the "tidy" command format the output of the query, making it more human-readable.

At this point, you can use the resources link to verify the output is correct.

- tidy_out.txt

```
<group ID="RedRock">
<Profile AppBacklog="0*287" AppFailureRate="0*287"
AppLoad="0*287" AppLoad2="0*287" AppResponseTime="0*287"
AppThroughput="0*287" Count="608"
Duration="3100*94,3100*169,3600*17,3600*7"
GMetric.BWH="0*263,42461.65,40116.16,35698.33,35754.41,35729.23,36034.52,3
6011.50,52160.56,50265.99,43280.39,35728.47,36088.33,42667.03,43688.43,428
43.75,42844.62,42801.89,42871.95,43246.84,42965.74,42067.57,35653.51,32991.
43,18488.61"
GMetric.cpu="0*287" GMetric.io="0*287" GMetric.mem="0*287"
GMetric.pwatts="0*287" GMetric.temp="0*287"
GMetric.watts="0*287"
IC="0*263,115,117,116*3,117*2,118,117*2,116,117,116,118,116*3,116,117,116*3,
117,100"
IDSU="0*287" MBP="0*287"
MQT="0*270,2191396,2182164,0,2195034,0*11,2214589,0"
MXF="0*264,1.00,0*5,2030.07,607.16,0,204.24,0*9,1.00,0,18.09,0"
MaxCount="1000" QOSCredits="0*287" SpecDuration="3600"
StartTime="1301641200,1302670800"
TANC="0*263,10465,10543,10324*3,10413*2,14362,15541,13221,10348,10413,11
244,11446,11252*3,11252,11349,11252,11148,10324,9585,5700"
TAPC="0*263,10465,10543,10324*3,10413*2,14532,15541,13221,10348,10413,11
244,11446,11252*3,11252,11349,11252,11148,10324,9585,5700"
TBP="0*287" TDSA="0*287" TDSU="0*287"
TET="0*264,16001.00,0*5,4564.00,3601.00,0,10801.00,0*9,32001.00,0,259202.00,
0"
TJA="0*264,1.00,0*5,3.00,1.00,0,1.00,0*9,1.00,0,2.00,0"
TJC="0*264,1,0*5,3,1,0,1,0*9,1,0,2,0" TMSA="0*287" TMSD="0*287"
TMSU="0*287" TNC="0*287"
TNJA="0*264,32002.00,0*5,39396.00,72020.00,0,259224.00,0*9,256008.00,0,4147
232.00,0"
TNJC="0*264,2,0*5,34,20,0,24,0*9,8,0,32,0" TNL="0*287"
TNM="0*287"
TNXF="0*264,2.00,0*5,60906.17,12143.13,0,4901.85,0*9,8.00,0,577.99,0"
TPC="34560*94,34560*186,34560*7"
TPSD="0*263,325597.09,326572.23,320047.56,320054.68,320053.79,320407.12,3
20497.01,444006.56,478849.69,407832.82,319184.63,321648.67,348574.67,3498
```

MCM

```
90.64,348824.61*2,348823.64,348822.67,351825.79,348920.64,345329.57,319168
.24,296078.83,176702.85"
TPSE="0*264,32002.00,0*5,39396.00,72020.00,0,259224.00,0*9,256008.00,0,4147
232.00,0"
TPSR="0*264,32000.00,0*5,39360.00,72000.00,0,259200.00,0*9,256000.00,0,414
7200.00,0"
TPSU="0*263,325597.09,326572.23,320047.56,320054.68,320053.79,320407.12,3
20497.01,438731.51,478849.69,407832.82,319184.63,321648.67,348574.67,3498
90.64,348824.61*2,348823.64,348822.67,351825.79,348920.64,345329.57,319168
.24,296078.83,176702.85"
TQJC="0*287" TQM="0*264,1,0*5,3,1,0,1,0*9,1,0,2,0" TQPH="0*287"
TQT="0*270,2191427,2182164,0,2195034,0*11,4375880,0"
TRT="0*264,16000.00,0*5,4560.00,3600.00,0,10800.00,0*9,32000.00,0,259200.00
,0"
TStartJC="0*270,5,0*4,1,0*11"
TStartPC="0*264,1,0*5,81,1,0,1,0,8,0*7,1,0,2,0"
TStartQT="0*270,8315354,0*16"
TStartXF="0*270,3597.00,0*4,1.00,0*11"
TSubmitJC="0*270,5,0*4,1,0*11"
TSubmitPH="0*270,616.00,0*4,8.00,0*11"
TXF="0*264,1.00,0*5,2032.10,607.16,0,204.24,0*9,1.00,0,35.76,0">
</Profile>
</group>
<group ID="root">
```

Explanation of what the sections mean, using TPSD as an example:

```
TPSD="0*263,325597.09,326572.23,320047.56,320054.68,320053.79,320407.12,3
20497.01,444006.56,478849.69,407832.82,319184.63,321648.67,348574.67,3498
90.64,348824.61*2,348823.64,348822.67,351825.79,348920.64,
345329.57,319168.24,296078.83,176702.85"
```

TPSD - Total proc-seconds (proc * sec) dedicated by this credential in the profiling interval

The "0*263" means that nothing happened in the first 363 profile iterations. The 325597.09 is utilizations in seconds. In short: if one iteration is 30 minutes or 1800 seconds, then a total of 1800 could be used. If this was 2 nodes & 1 processor each, then it would effectively be doubled to 3600 seconds. Again, this article was intended as a brief introduction to troubleshooting MCM queries and XML output.

Unique solution ID: #1034

Author: Jason Booth

Last update: 2017-05-18 03:08