Fixes & future plans for Torque/Moab

Very recently I have attended MoabCon13 in Park City, UT. It is a great conference as you get to talk to developers of Torque and Moab. Having said that, I would like to mention few things we have implemented on our systems after coming back from conference.

Job dependency failures fixed

When a job is dependent on the status of a completed job for it to run, sometimes we’ve seen some of these dependent jobs failing for no reason. Apparently, we need to keep the completed jobs at least for a few seconds before Torque purges them forever. Before I made Torque to purge the jobs immediately after their completion I had it keep them for 12 hours. But sometimes, users submit thousands of jobs with in a couple of minutes and if there is a problem with the script all of them complete in a matter of few seconds. This leads to serious problems such as suffocation of PBS server, leading to a very slow response for the commands qstat, pbstop, etc. This happened many times on Bowery and so I ended up making Torque purge job immediately after completion.

But like I mentioned before this lead to some job failures that are dependent on the status of other jobs. It was hard to debug this problem as not all the dependent jobs failed but just very few here and there. Then I was told by one of the Torque developers we needed to keep the completed jobs at least for a few seconds so that Torque picks up its status and starts/stop the job dependent on this status. Since the version of Torque on our clusters is single threaded, I believe some times it can not get the status of a completed job and do something with the dependent job as it is busy doing something else. I guess this lead to job failures or jobs starting to run no matter what the status of the job it is dependent on as Torque had no idea what happened to the job as it was already gone by the time it wanted to check the status.

The solution is to keep the completed jobs at least for few seconds. Now we keep the completed jobs for 120 seconds after their completion. The researcher who had problems with some of his job dependencies has reported there haven't been any failures since then. It is really important for researchers that dependency works well as jobs depend on each other and this leads to many job failures in a cascading fashion if something goes wrong.

This is the parameter we needed to turn on from pbs_server side:

```
qmgr -c 'keep server_completed = 120'
```

Torque 4 and Moab 7

The version of Torque we have on our clusters is 2.5.11. which is kind of stable. We have plans to get phi cards sometime this year. The only version of Torque that supports phi cards is 4.x.x. Since its coming almost two years back it has had many problems. Finally, now it looks like it is getting into the category of being a stable release. After the conference, listening to the feedback of admins from other institutes I feel like this is the right time for me start testing this version.

Soon, I will start testing it on one of our decommissioned clusters. I think it'll take time to test it thoroughly. No matter how thorough we test it can never be good enough not to face problems when we put it in the production environment. Reasons are simple. First of all, we can never have a test cluster equivalent to production cluster in terms of size and architecture, etc. Still, it does help in many ways.

Regarding Moab, the version we have is 6.1.7. When I put it two years back I changed the database schema to MySQL. We use Torque logs for accounting purpose (usage data). This means we don't use Moab database for anything. The reason I went for it was that it'd be there in case we needed it for something. But we never needed it and at the same time we started seeing problems with Moab. Its memory consumption started spiking leading to master node slowness.

This is bad as master node is the vital element for the cluster as it has Moab and PBS server on it. This means, if something happens to master node, let's say it crashes, users can not submit jobs and check status of already submitted jobs, etc. Even though, installing Moab and PBS server on a separate node is an ideal solution, our situation is that master node has both of them on it.

First we thought there was memory leak in Moab that causes its memory consumption to spike. Finally, we had to keep a cronjob that kills Moab as soon as memory goes up to 4GB. This is not a good solution but the only solution available in stead of moving to another sub release from Moab.

During the Moab conference I came to find out from Moab engineers that Moab had a problem with the database connector and so it starts keeping the data in the memory until it could dump it into the database. Unfortunately because of the bug, it never happened and memory started going up leading to many problems.

My solution to the problem was to turn the database off as we don't use it at all. I have plans to use it in the future. But when I turn it back on I will have newer version of Moab where this bug has been fixed. Not making Moab dump the data into database has fixed the memory problem. Now memory consumption is just like how it should be, 600MB.

By the time we have new nodes on our cluster, sometime this year, I will have tested both Torque and Moab and be ready to put them either on a separate node or on the new master node build. There are many advantages with these versions. Torque 4 is multi threaded and so many users could issue many qstats at the same time without leading to slow response. Everything is seems to be faster with this version. I'm sure there are many more advantages. I'll publish them here when I clearly have an idea on each one of them.

From Moab side, there are many improvements that make it a much more usable piece of software for both admins and users. One good thing is we can query Moab without having to slow it down as from version 7 queries directly goto Mongo database rather than Moab doing it for us. There are many more I will talk soon. I'm sure users will love them as some of them are cool GUI features.
Sreedhar
HPC Support Specialist.