TORQUE Resource Manager 5.0.1 release notes

The release notes file contains the following sections:

New Features

The following is a summary of key new features in TORQUE Resource Manager.

5.0.1

Modification of the Output Location

TORQUE now allows for the modification of the output location based on the Mother superior hostname. An environment variable ($HOSTNAME) has been added to the job's environment.

5.0.0

CPU frequency control

TORQUE can now set the CPU frequency on requested nodes for submitted jobs. The request is made with the new cpuclock resource extension. A user can specify a desired clock frequency in megahertz, a Linux power governor policy name, or an ACPI performance state (P-state) number. For more information, see cpuclock in Requesting Resources in the TORQUE RPM Installation Guide.

The qalter -l command has been updated to allow modification of the requested CPU frequency on an already submitted job.

The pass_cpuclock server parameter was added allowing administrators to track, but not grant, the CPU frequency request portion of a job submission. For more information, see pass_cpuclock (Appendix B: Server Parameters) in the TORQUE RPM Installation Guide.

qrerun all command

When you execute the `qrerun all` command, you will be prompted for confirmation. TORQUE will then place all running jobs in a queued state without contacting the MOMs. You should only use this when the entire cluster is down and you cannot contact it.

Node power state control

TORQUE can now set the power state of a node. Depending on the hardware and software capabilities of the node, TORQUE can set the power state to Running, Standby, Suspend, Hibernate, or Shutdown. A new -m option was added to the pbsnodes command to make this work. For more information, see pbsnodes (Appendix A: Commands Overview) in the TORQUE RPM Installation Guide.

The syntax of the command is:

```
pbsnodes -m [running|standby|suspend|hibernate|shutdown] <space delimited list of
nodes to alter>
```

In order to wake a node from a low-power state, Wake-on-LAN must be supported and configured on the node. For more information, see Changing node power states in the TORQUE RPM Installation Guide.

Differences

This section contains differences in previously existing features that require a change in configuration or routine.

5.0.1

qmgr Server Parameter "copy_on_rerun"

A new qmgr option: set server copy_on_rerun=[True|False] is available. When set to True, Torque will copy the OU, ER files over to the user-specified directory when the qrerun command is executed (i.e a job preemption). This setting requires a pbs_server restart for the new value to take in effect. Note that the MOMs and the pbs_server must be updated to this version before setting copy_on_rerun=True will behave as expected.

qmgr Server Parameter "job_exclusive_on_use"

A new qmgr option: job_exclusive_on_use=[True|False] is available. When set to True, pbsnodes will report job-exclusive anytime 1 or more processors are in use. This resolves discrepancies between Moab and TORQUE node reports in cases where Moab is configured with a SINGLEJOB policy.

TORQUE Accounting Improvements

Two new fields were added to the accounting file for completed jobs: total_execution_slots and unique_node_count. total_execution_slots should be 20 for a job that requests nodes=2:ppn=10. unique_node_count should be the number of unique hosts the job occupied.

scan_for_terminated

Improved performance by moving scan_for_terminated to its own thread.

Port Using the Munge API Instead of Forking

TORQUE now uses the Munge API, rather than forking, when configured with the `--enable-munge-auth` option.

pbsdsh -o Option Captures Stdeer

The pbsdsh -o option is modified to add stdeer capturing.

5.0.0

The job_stat_rate parameter has a new default and function

Before this release, pbs_server asked the mother superior of every job for an update on the job every **job_stat_rate** seconds. The mother superior now sends updates on every job with its regular status, so there is no need for pbs_server to regularly poll. Instead, this parameter sets a timeout.

Two threadpools and new default for max_threads

The **max_threads** parameter has a new default: The value of min_threads ((2 * the number of procs listed in /proc/cpuinfo) + 1) * 20. In previous versions, 20 was 10.

Additionally, threadpools are now split. One-fourth of the threads are allocated for background tasks and three-fourths of the threads are allocated for incoming requests from MOMs and through the API (client commands, Moab, and so forth). Additionally, incoming requests no longer build up indefinitely. If a new request comes in, pbs_server evaluates whether it is too busy to service the request. For managers, the request is serviced as long as there are at least two threads available in the threadpool. For non-managers, the request is serviced as long as at least 5% of the threadpool is available. When pbs_server is too busy, it returns the error code PBSE_SERVER_BUSY with the message: "Pbs Server is currently too busy to service this request. Please retry this request."

Job status polling removed from TORQUE

Pbs_server now polls a mom for a job's information only if it hasn't received the information in 5 minutes. Otherwise, the information is communicated with the mom's status information.

TORQUE no longer searches linearly for the nodes in a node list

TORQUE now recognizes when a request to run a job specifies a node list. It directly accesses those nodes instead of searching for them linearly.

The exec_host list has one entry per node

The exec_host list has been condensed to contain one entry per node instead of one entry per execution slot. The node entry contains a string specifying each execution slot index.

TORQUE no longer displays the value of exec_port in qstat.

The qstat -f output for CPUs has been reduced

The output of qstat -f has been condensed to reduce clutter in the CPU section.

Known Issues

The following are known issues in TORQUE Resource Manager. Following each issue description is an associated issue number in parentheses.

Known issues are aggregated and grouped by the release version for which they first occurred or where reported.

5.0.1

- Some limitations exist in the way that pbsdsh can be used. Please note the following situations are not currently supported:
  - Running multiple instances of pbsdsh concurrently within a single job. (TRQ-2851)
  - Launching a large number of processes in succession (causes pbdsdsh to hang). (TRQ-2890)

5.0.0

No known issues.

Resolved issues

The following is a list of some key bugs fixed in TORQUE Resource Manager. Following each issue description is an associated issue number in parentheses.

Resolved issues are aggregated and grouped by the release version in which they were resolved.

5.0.1

- **Torque job can kill processes not owned by the job owner.** Fixed issue around unauthorized termination of processes. (CVE-2014-3684, TRQ-2885)
- **qstat -Q <bad_queue_name> prints queue name twice.** Fixed bug where giving a bad queue name to qstat -Q results in duplicate output. (TRQ-2025)
- **Mail output for qsub -m options was failing to output new lines.** Fixed bug in qsub -m when TORQUE is configured --with-sendmail. Some missing newlines were added. (TRQ-2937)
- **mppnodes hostlist was being re-ordered.** This bug is fixed. (TRQ-2112)
- **Some tasks were incorrectly listed as 0 in qstat -a when requested specific nodes.** This has been resolved. (TRQ-2292)
- **TORQUE accounting problems - jobs without accounting records.** Fixed bug related to accounting records on large systems. (TRQ-2367)
- **qstat wouldn't parse anything after a bad job ID.** Improved qstat behavior in cases where bad job IDs were referenced in the command. (TRQ-2410)
- **Separate headers for multiple jobs IDs provided to qstat.** Fixed output format bug in cases where multiple job IDs are passed into qstat. (TRQ-2411)
- **qsub did not process arguments correctly when a submitfilter is used.** Fixed bug where qsub did not process args correctly when using a submit filter. (TRQ-2646)
- **Parsing bug found when using hostlist ranges in qsub.** This bug is fixed. (TRQ-2652)

- **Build bug reported with MIC libraries.** Fixed build bug related to newer Intel MIC libraries installing in different locations. (TRQ-2653)
- **XML job log error.** Corrected mismatched <Job_Id> XML tags in the job log. (TRQ-2692)
- **TORQUE was not dividing GPUS amongst NUMA nodes.** Fixed problem where GPUs were not split between NUMA nodes. You now need to specify which gpus belong to each node board in the mom.layout file. (TRQ-2730)

  A sample mom.layout file might look like

  ```
  nodes=0 gpu=0
  nodes=1 gpu=1
  ```

  > ⓘ This only works if you use nvml. The nvidia-smi command is not supported.

- **TORQUE was leaving behind error and out files when a job was preempted or requeued.** Fixed bug where OU files were being left in spool when job was preempted or requeued. (TRQ-2732)
- **Reported cput was incorrect.** Fixed bug where reported cput was incorrect. (TRQ-2759)
- **pbsnodes -l offline -n unexpected error.** Fixed unexpected error when running `pbsnodes -l offline -n`. (TRQ-2760)
- **max_user_queuable limit reached, however, there were no jobs in the queue**. Fixed bug where jobs rejected due to max_user_queuable limit reached, yet no jobs in the queue. (TRQ-2795)
- **momctl -q clearmsg didn't seem to clear error messages permanently.** Fixed bug where `momctl -q clearmsg` didn't properly clear error messages. (TRQ-2828)
- **Some bugs found that caused TORQUE core crashes.** These bugs are fixed.
- **TORQUE was crashing.** Fixed crashing.
- **pbs_server segfault after large array deletion.** Fixed segmentation fault. (TRQ-2835)
- **GPU nodes where not passed to sister nodes.** This bug is fixed. (TRQ-2837)
- **pbs_server did not write resource_default units to serverdb file.** This bug is fixed. (TRQ-2852)
- **pbs_mom filling up the logs in a HA environment.** Reduced verbosity in error logging in HA environments. (TRQ-2863)
- **Make trqauthd error messages more meaningful and non-repetitive.** Improved trqauthd error messages to be more meaningful and less redundant. (TRQ-2882)
- **Remote job submissions were being rejected, even when host is in submit_hosts list.** Fixed problem with remote client job submission during ruserok() calls. (TRQ-2918)
- **pbsdsh did not support running multiple instances concurrently.** Improved pbsdsh to better handle simultaneous use of -o and -s options. Also fixed some problems where -o output was sometimes getting truncated. (TRQ-2890)
- **TORQUE was not notifying Moab of completed jobs.** Fixed bug where TORQUE was not honoring KeepCompleted server parameter when job_nanny was set to true. (TRQ-2904)
- **HOST_NAME_SUFFIX was no longer adding suffix to job names.** This bug is fixed. (TRQ-2956)

- **Deadlock when running 'qdel -p' as non-root user.** Fixed deadlock issue. (TRQ-2919)

5.0.0

- **When the queue contained more than 50,000 jobs, TORQUE slowed down substantially.** TORQUE no longer slows down with a large number of jobs in the queue. (TRQ-2345)
- **When pbs_server had a high load, it would get stuck polling.** pbs_server no longer gets stuck polling under these conditions. (TRQ-2620)
- **When a job with a dependent job was deleted with qdel, TORQUE did not clean up the dependent job.** TORQUE now removes the dependent job when you delete its parent. (TRQ-2621)
- **When a lot of jobs were run at once, one job would get stuck in an exiting state.** This error no longer occurs. (TRQ-2622)
- **The afterok dependency did not work as expected.** afterok now works correctly. (TRQ-2626)
- **When running a large amount of jobs, the server would crash during job recycle.** The crash no longer occurs. (TRQ-2628)
- **pbsdsh requires FQDN even if other elements don't.** pbsdsh no longer requires FQDN. (TRQ-2632)
- **A deadlock would occur on job_save failure.** This error no longer occurs. (TRQ-2645)
- **Asynchronous job starts queued in TORQUE but not yet serviced caused Moab to reschedule jobs that would eventually run.** A new job sub-state in TORQUE prevents this from occurring. (TRQ-2715)