Torque Resource Manager 6.0.1 Release Notes

The release notes file contains the following sections:

- New Features
- Differences
- Known Issues
- Resolved Issues

New Features

This section contains a summary of key new features in Torque Resource Manager.

6.0.1

Added ghost_queue Queue Attribute

When pbs_server restarts and recovers a job but cannot find that job's queue, it will attempt to recover the job by creating a ghost_queue for that job.

Added gres_modifier Server Parameter

Gives permission to a list of users to modify the gres resource of their own running jobs.

Untrusted Host Vaidation

Added the ability to trust certain users or groups from hosts without allowing all users from those hosts to submit jobs.

Added Process Adoption Through pbs-track

Added the ability to adopt running processes into a job with pbs_track.

User settable kill_delay Through qsub -K Option

Added a user settable, per-job kill delay, called kill_delay. It is settable via the *new* qsub -K option.

6.0.0

cgroup Support

Torque is enhanced to create one Linux control group (cgroup) per task based on the new NUMA-aware, task-based job submission option (-L) and to create one cgroup for all tasks of a job on the same compute node for the older job-based option (-l). Torque uses cgroups to manage CPU and memory accounting, enforce memory usage limits, set up CPUset management, and bind cores/threads, memory, and accelerators, such as GPUs and MICs, to jobs.

When binding resources that include an accelerator to a task, Torque will make a best-effort attempt to place a task on the cores/threads and memory of the socket/NUMA node to which the accelerator attaches.

Ability to Prevent Nodes Being Dynamically Edited

A new qmgr parameter is available. When 'dont_write_notes_file' is set to true, the nodes file cannot be overwritten for any reason; qmgr commands to edit nodes will be rejected. The default is FALSE.

Execute the Job Starter Script with Elevated Privileges

The '$job_starter_run_privileged' MOM configuration parameter is added and lets you specify whether Torque executes the job starter script with elevated privileges. The default is FALSE.

Differences

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

6.0.1

Submission Syntax Check Added to Prevent Mixing NCPUs and Nodes

qsub guarantees that ncpus and nodes cannot be mixed.

Added Capability to Pass Environment Variables to pbsdsh

Two new options have been added:

-e list - Lets user specify list of environment variables separated by commas. If only a variable name is listed or a variable name is given with no value (ex. name=), its value will be read from the pbsdsh environment if it exists, otherwise it will be empty. If a variable name with a value is specified (ex. name=value) then the specified value will be assigned to the variable name in the tasks' environment.

-E - Include all environment variables from the pbsdsh environment in the tasks' environment.

> ℹ When using -e and -E together, and when common variable names are read (or set in the case of -e), if -e is specified *first* then -E read values will prevail in the tasks' environment. Otherwise, -e specified values will prevail.

Added a Way for allow_node_submit Exceptions

Added a way to exclude compute nodes from allow_node_submit.

pbs_server Enhancement for Very Large Number of Jobs

pbs_server has been enhanced to better handle a very large number of jobs (several hundred thousand or more) by enabling an alternate way for it to store job-related files in the directories $PBS_HOME/server_priv/jobs and $PBS_HOME/server_priv/arrays.

A new boolean server attribute, use_jobs_subdirs, lets an administrator direct the way pbs_server will store its job-related files. When use_jobs_subdirs is unset (or set to false), job and job array files will be stored directly under $PBS_HOME/server_priv/jobs and $PBS_HOME/server_priv/arrays. This is the default behavior and the way the server has stored these files in the past. When use_job_subdirs is set to true, job and job array files will be distributed over 10 subdirectories under their respective parent directories. This method helps to keep a smaller number of files in a given directory.

If an administrator wishes to change the use_jobs_subdirs attribute from its previous value (or when setting it to true when it has not previously been set), it is highly recommended that Torque be drained of all jobs. Failing to take this action may result in the loss of existing jobs.

qmgr Support Added for "loglevel" Attribute

Allows for qmgr to recognize "loglevel" as an equivalent of "log_level". The user can now type in either as a valid attribute.

Revert vmem Calculation Changes

Added the ability to control whether or not vmem is seen as per job or per node when cgroups are enabled.

6.0.0

Default RPM Installation Path Is Changed

The Torque default path for an RPM installation has been changed to match the path used during a tarball (Manual) installation. The default path for both install methods is `/usr/local`.

down_on_error Server Parameter Now Defaults to TRUE

By default, nodes that report an error from their node health check to pbs_ server will be marked down and unavailable to run jobs.

pbs_mom Now Sets Environment Variable for NVIDIA GPUs

A new mom config parameter, $cuda_visible_devices, was added to specify whether pbs_mom sets the CUDA_VISIBLE_DEVICES environment variable when it starts a job. The default is TRUE.

$prologalarm is Always Honored

$prologalarm was ignored on the prologue for a job. Also when the epilogue was run the $prologalarm value was ignored if it was more the 300. Now the $prologalarm value is always honored regardless of how large it is for both prologue and epilogue scripts. The default timeout is still 300 seconds.

Known Issues

This section lists 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 were first reported.

6.0.1

- pbs_mom failed to add job tasks to the devices cgroup on sister nodes of a parallel job. The failure to add a job pid to the devices cgroup results in the job not having restrictions to GPU or MIC devices. All GPU and MIC devices are available to the job. (TRQ-3522)
- Devices subsystem is enabled for cgroups. However for RHEL 6-based systems, the devices subsystem is considered a "Technology Preview". We have tested the devices subsystem and we have it working in our tests. However, any problems with the devices subsystem and Torque may be caused by the early access to this feature.
- When using cgroups, cgroup directories may be left behind for some jobs. Once the jobs are completed, these cgroup directories can be removed using rmdir at the convenience of the sysadmin.

6.0.0

- Running multiple instances of pbsdsh concurrently within a single job is not supported. (TRQ-2851)
- pbsdsh will fail to return under certain conditions (not-passing high-stress tests). (TRQ-3308) *Resolved 6.0.0.1*
- Kernel crashes may occur when using cgroups on CentOS or RHEL prior to 6.6. See https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/6.6_Technical_Notes/kernel.html; especially RHEL6.6 fix BZ#1204626. If cgroups are part of your configuration, Adaptive Computing recommends running a more recent version of CentOS or RHEL. (TRQ-2583)

Resolved Issues

Resolved issues are aggregated and grouped by the release version in which they were resolved. When applicable, each resolved issue has the associated issue number in parentheses.

6.0.1

- Node recovers when behind processing requests. pbs_server now detects when a node is failing too frequently and mark it down temporarily if this happens. Once a node is marked down, it will be marked up again if either two consecutive communications from pbs_server to the node receive successful replies, or after five minutes of staying offline (whichever comes first). A node is considered to be failing too frequently if it has three failures to reply to a server request without have two consecutive successes in between. (TRQ-2517)
- Array subjobs did not have a queued entry in the accounting log. (TRQ-3470)
- Segfault in create_alps_subnode with node_note populated. (TRQ-3445)
- Problems with clearing a node note. Removed length restriction on a node note. (TRQ-3439)
- Jobs that never ran were receiving end records. (TRQ-3432)
- Resources_used.walltime changed to seconds from HH:MM:SS in accounting logs. (TRQ-3385)
- pbs_server timed out connection to pbs_mom. Added load balancing to login nodes when they start to get busy. (TRQ-3367)
- pbs_mom would hang when sending status from a child. Added a timeout for node health check scripts so that they cannot make the mom daemon hang. (TRQ-3364)
- pbs_mom hangs on restart with init script. Ensured that necessary services have been brought up before starting the Torque daemons and that the Torque daemons are shutdown before their required services are shutdown. (TRQ-3345)
- Fixed a memory leak when jobs were being started asynchronously. (TRQ-3326)
- qsub -W stage-in was not working. Fixed failures where the group name showed up in the log as the problem but the user did not belong to the group name given in the error. (TRQ-3312)
- Multiple moms sent invalid destroy_alps_reservation/req_delete_ reservation. Only allows one kill orphaned reservation request per reservation at one time. (TRQ-3299)

- Jobs with square brackets in the name were aborted on restart if they weren't array subjobs. An issue was fixed with jobs getting aborted if they are named with "[]" in the name but aren't Torque array jobs. (TRQ-3214)
- Completed jobs were not getting cleaned up. Fixed various issues when restarting dependency jobs, including them not getting removed even after completion. (TRQ-3175)
- Error condition where the mom's port would be inserted into the .JB file name. (TRQ-3090)
- Logs filled with messages about not sending hierarchy to mom. Failures are only logged the first time it can't send the hierarchy to a mom. (TRQ-3156)
- Down/offline nodes caused TORQUE to not online elastic nodes. pbs_server is now able to bring up new nodes even when there are nodes in the system that are down or offline. (TRQ-3066)
- Torque was not able to release holds on job arrays. Running qrls on an array subjob allows pbs_server to correct slot limit holds for the array it which it belongs. (TRQ-3088)
- Array templates were being reported as jobs. (TRQ-3405)
- Memory calculation issues reported when cgroups enabled and -l vmem|pmem|mem are used. (TRQ-3499)

6.0.0.1

- A hang in pbsdsh occurred if the pbs_mom daemon was started with a -q or -r option. (TRQ-3308)
- Typo found in the error message reported when the swap memory limit could not be set.

6.0.0

- With kill_delay and $exec_with_exec set, a job would be set to a completed state after running qrerun instead of getting set back to queued. (TRQ-2993)
- Array slot limits were not getting decremented when a job is preempted or rerun. (TRQ-3110)
- Jobs were getting stuck in a running state when an asynchronous run failed. (TRQ-3114)
- Interactive jobs not staying on the node from which they were submitted. (TRQ-3122)
- Occasionally a random group name would show up for a user who did not belong in the group. A race condition was fixed by changing to thread safe calls to get group and user ids. (TRQ-3190)

- Threadpool in pbs_mom was not being started. When $thread_unlink_calls is set to true in /var/spool/torque/mom_priv/config, job files were not being deleted at job end in the mom. (TRQ-3232)
- Reporter mom did not correctly handle UNKNOWN role. (TRQ-3245)
- Read timeouts were being retried indefinitely by pbs_server. (TRQ-3306)