

# Torque Resource Manager 6.0.3 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.

### 6.0.3

#### *Torque Setup Script Improvements (TRQ-3866)*

The `torque.setup` script now permits specification of a config path (default is the path used when configuring Torque). New options have also been added to specify a prefix path (path to executables is in `PREFIXPATH/bin` and `PREFIXPATH/sbin`) and a new option has been added to log errors to `syslog`.

#### *pbs\_mom Switch to Force the Server to Accept a Hardware Update*

`pbs_mom -f` forces the server to accept an update of the hardware on the node. You should use this switch the first time you start the MOM after an upgrade.

#### *Added default\_gpu\_mode Server Parameter*

The `default_gpu_mode` server parameter sets the default value GPU mode. This should facilitate using CUDA 8.

#### *Simplified Torque RPM Installation*

The RPM installation no longer writes the server name into `pbs_mom`'s config file, eliminating the need to configure the MOM config file in the default case.

#### *Simplified Installation Process*

"make install" and the package installer now install service/unit files, set up the path to the Torque libraries, and establish a shell path to binaries for `sh` and `csh` shells.

#### *Resource Plug-In*

There is now an API for creating resource plug-ins that report custom `varattrs`, generic resources, generic metrics, and features. Additionally, jobs can report custom resources through the plug-in.

#### *-L Submission Recorded in the Accounting Log*

Jobs submitted with the `-L` request syntax will now have the `-L` submission recorded in the accounting log.

#### *Change hostname for a Torque Job to Match the Corresponding Moab Job ID*

When jobs are submitted using `qsub -J`, the job's hostname is set to the job ID of the Moab job.

### *qstat -f Reports Slot Limit For Arrays*

A slot limits for an array job is now shown in "qstat -f" output under the job\_array\_request attribute. A hold type due to a slot limit for an array job is now shown under the Hold\_Types attribute ("I" hold type means the hold is due to a slot limit).

### *MOM Parameter to Overwrite Output Files*

The \$force\_overwrite MOM parameter, if set to true, forces the output files to be overwritten each time a job is started.

### *Option to Turn Off trqauthd Logging*

Added a -n option to trqauthd that disables trqauthd from logging anything.

## 6.0.2

### *Recover Array Subjobs*

The new server parameter "ghost\_array\_recover" is added. pbs\_server will now recover array subjobs even when the array (.AR file) couldn't be recovered. This parameter is set to TRUE by default.

### *Cray-Enabled Torque May Also Be Configured with cgroups*

Support is added for Cray-enabled Torque configured with cgroups.

- On the login node, each job will have all of the CPUs and all of the memory controllers in its cgroup.

### *Epilogue Script Runs, Even if Output Files Cannot Be Appended*

The epilogue script will now run when spool\_as\_file is configured, even if error and output files are not available and cannot be appended.

### *Job Script Path is an Argument to Prologue and Epilogue Scripts*

A new positional parameter contains the full path of a job's job script to the job's prologue and epilogue scripts when the Torque pbs\_mom "mother superior" launches the scripts. The prologue script is the new 8th positional parameter. The epilogue script is the new 11th positional parameter.

### *User May Request No Emails be Sent, Even on Failure*

The qsub -m p option makes it so no emails are sent for the job, even on failure.

### *Torque Commands Appear in User's Login Shell Path*

RPMs created via build-torque now create /etc/profile.d/torque.sh (and torque.csh) files so that Torque commands appear in a user's login shell path.

### *Option to Email User on All Non-zero Exit Codes*

The `qsub -m f` option sends an email if a job has a non-zero exit code. The `qsub -m f` option can be used with the `a`, `b`, and `e` options, but not with `n` or `p`.

### *Added an Option for When to Send Mail Notifications*

Added a `-m f` option to `qsub` to send mail when a job terminates with a non-zero exit code.

### *Option to Disable Reading of RUR Information*

Added `$cray_check_rur` configure option to disable reading of Resource Utilization Reporting (RUR) energy usage for Cray login nodes. If set to `false`, login MOMs will not look at the energy resource information used for each job. Disabling this may improve performance.

### *New "email\_batch\_seconds" Server Parameter*

The new server parameter "email\_batch\_seconds" lets you control at what frequency a batch of emails are sent to each user.

#### 6.0.1

### *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.

### *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 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 Process Adoption Through pbs-track*

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

## 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.

**i** All six default roles are present regardless of whether your configuration includes Nitro or Remote Visualization.

### 6.0.3

#### *pbs\_server Does Not Shut Down When trqauthd is Stopped*

The *trqauthd* service is no longer a requirement of *pbs\_server*, so *pbs\_server* no longer shuts down when *trqauthd* is stopped.

#### *TorqueConfigure Script Does Not Enable the GUI Component*

When the *devel-tk* and *devel-tcl* libraries have been installed, the Torque configuration script does not enable the GUI component. To build the GUI component, specify `--enable-gui` when building from source.

#### *\$CUDA\_VISIBLE\_DEVICES is not set by default*

`$CUDA_VISIBLE_DEVICES` is not set by default if you're using *cgroups*. Also, this makes its contents compatible with varying CUDA versions.

#### *Default Value for keep\_completed Parameter Changed*

The default value for the Torque `keep_completed` parameter has been changed to 300 seconds. The `keep_completed` parameter specifies the number of seconds jobs are kept in the job execution queue after they are completed. Keeping completed jobs in the queue enables Torque to report on the status of the jobs.

### 6.0.2

#### *NUMA-Aware cgroup Creation by Per Task or Per Job*

A new Torque server parameter "`cgroup_per_task`" is available to let you specify whether *cgroups* are created per task or per job. The default is `FALSE`, meaning jobs submitted with the `-L` syntax will have *one* *cgroup* created per host; this behavior is similar to the pre-6.0 *cpuset* implementation.

The `qsub/msub -L` syntax is also modified to let you specify whether the *cgroup* is per task or per job at the job submission time.

**i** Some MPI implementations are not compatible with using one *cgroup* per task.

### *legacy\_vmem Server Parameter Affects Behavior of the -l vmem Option*

legacy\_vmem is a new server parameter that affects the behavior of the -l vmem option. When set to true, the vmem request will be the amount of memory requested for each node of the job. When it is unset or false, vmem will be the amount of memory for the entire job and will be divided accordingly.

### *Queue Support for Both resources\_default.\* and req\_information\_default.\* Settings*

When queues have both resources\_default.\* and req\_information\_default.\* set then they are applied according to their resource request type. resources\_default.\* settings are applied to jobs that do not explicitly use the -L syntax, while req\_information\_default.\* settings are applied only to jobs that explicitly use the -L resource syntax.

### *Prohibited Mode Not Allowed for User Jobs*

Setting the compute mode of an NVIDIA GPU to prohibited makes it so the GPU cannot be used at all. In previous versions of Torque users were allowed to set a GPU to prohibited mode. But then it could not set the mode to anything else since the GPU was now prohibited. This change went into effect for version 5.1.3, 6.0.2, and later.

### *Support for Single Job Dependencies and Array Dependencies at the Same Time*

Jobs can depend on single job dependencies and array dependencies at the same time.

### *Reduced the Number of Logging Statements*

Reduced the number of logging statements when a node isn't up and therefore can't receive the mom hierarchy.

### *Added tcp\_incoming\_timeout Server Parameter*

tcp\_incoming\_timeout specifies the number of seconds before incoming connections timeout. tcp\_timeout now specifies the timeout for outgoing connections or connections initiated by pbs\_server. tcp\_incoming\_timeout functions exactly the same as tcp\_timeout, but governs incoming connections while tcp\_timeout governs only outgoing connections (or connections initiated by pbs\_server).

## 6.0.1

### *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.

### *Submission Syntax Check Added to Prevent Mixing NCPUs and Nodes*

qsub guarantees that ncpus and nodes cannot be mixed.

### *Added a Way for allow\_node\_submit Exceptions*

Added a way to exclude compute nodes from allow\_node\_submit.

### *Added Capability to Pass Environment Variables to pbsdsh*

Added capability to pass environment variables to tasks created using 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.

### *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.

### *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.

## 6.0.0

### *\$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.

### *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.

### *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.

### *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.

## 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.

### 6.0.2

No known issues.

### 6.0.2

- Torque won't compile when the tk-devel and tcl-devel packages are installed on your host. (TRQ-3723)

As a work around, disable building of the gui component by using `--disable-gui` when executing `configure`.

- `qsub -X` may incorrectly look for `xauth` in `/usr/X11R6/bin/` instead of `/usr/bin/`. (TRQ-3489)

As a workaround, you can set `XAUTHPATH /usr/bin/xauth` in `TORQUE_HOME/torque.cfg` on client machines, and `$xauthpath /usr/bin/xauth` in `TORQUE_HOME/mom_priv/config` on the compute nodes. Alternatively, you may be able to work around the issue by simply creating a symlink from `/usr/X11R6/bin/xauth` to `/usr/bin/xauth` on `pbs_mom` hosts.

### 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) *Resolved 6.0.2*
- 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. *Resolved 6.0.2*

### 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](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.

### 6.0.3

- GPU models k40s and k80s were not detected correctly when using GPUs and cgroups. (TRQ-3861)
- pbs\_server segfaulted on systems with large number of open file descriptors. (TRQ-3847)
- pbs\_server segfaulted due to a change in the pthread library. (TRQ-3841)
- qsub loaded erroneous jobs after a server shutdown, causing pbs\_server to crash. (TRQ-3839)
- Deleting a job without permission caused the dependencies to be removed. (TRQ-3833)
- Torque sometimes calculated available disk space incorrectly. (TRQ-3832)
- Queues applied conflicting defaults, such as when a job requests nodes and the queue has a procs default value. (TRQ-3830)
- Torque crashed due to a job with failed dependencies. (TRQ-3810)
- pbs\_server crashed when using the mail throttling feature. (TRQ-3801)
- Systemd unit files do not correctly start/stop Torque daemons. A new -F (don't fork) switch has been added to pbs\_server, pbs\_mom, and trqauthd. (TRQ-3795)
- stdout was not properly delivered on NFS file systems when Torque was configured with --disable-spool. (TRQ-3792)
- Torque was not allocating resources correctly for cgroup jobs. (TRQ-3790)
- pbs\_server was not starting when an array dependency was cleared before pbs\_server started. (TRQ-3774)
- Torque had communication issues when IPv6 was configured but not used. (TRQ-3773)
- Stdout/stderr files were not handled correctly when there were spaces in the path. (TRQ-3767)
- Torque was not waiting for a compute node to do a copy out process. (TRQ-3762)
- Fixed node locking issues that were causing pbs\_server to abort when built with newer versions of the threading library. (TRQ-3755)

- Server build was failing with --disable-spool configure option. (TRQ-3751)
- Requesting nodes with both mem and vmem limits caused cgroup memory limits to be set to an incorrect value. (TRQ-3749)
- Forced requeue command (qrerun -f) was not obeying kill\_delay settings. (TRQ-3737)
- Job dependencies were not released when job\_suffix\_alias was set to something other than the hostname. (TRQ-3735)
- When adding a dynamic node, the server was initially putting the node in free state, but not reporting this state to the server. (TRQ-3733)
- The memory cgroup was not being set correctly for jobs requesting pmem and procs. (TRQ-3728)
- Jobs were not being removed from node\_usage files after ending. (TRQ-3727)
- pbs\_server aborted under ghost array recovery when job array files were missing. (TRQ-3719)
- pbs\_mom crashed when started on the reporter node without \$apbasil\_protocol set in the config. (TRQ-3715)
- momctl was not displaying the \$varattr script or the correct time and date. (TRQ-3712) (TRQ-3617)
- Error messages logged when jobs failed to copy files were misleading. (TRQ-3711)
- Jobs' environments and stdout paths were invalid when submitted from a deleted working directory. (TRQ-3609)
- qsig was not working correctly when display\_job\_server\_suffix = false. (TRQ-3102)
- Allocating GPUs did not correctly set shared mode when requested. (TRQ-3213)
- pbs\_mom init.d script sometimes timed out during startup. (TRQ-2911)
- qsub -v was not processing environment variables with no value correctly. (TRQ-2699)
- qstat -f was not displaying init\_work\_dir. (TRQ-2459)
- pbs\_demux had an unnecessary library dependency. (TRQ-1996)

#### 6.0.2

- Torque was not allocating enough memory controllers to satisfy memory requests. (TRQ-3681)
- pbs\_server was not being properly shut down when in HA mode. (TRQ-3670)

- pbs\_server was not detecting and updating total\_threads when a node's hyperthreading was enabled. (TRQ-3662)
- pbs\_server was not properly restarted when running "service pbs\_server restart" during installation. (TRQ-3657)
- Memory and swap limits were not set in cgroup. For information on how memory and swap options are used, see [1.1 -L NUMA Resource Request](#). (TRQ-3656)
- Jobs submitted with -l option with exclusive access to the node were not receiving all CPUs and memory controllers in the cgroup. (TRQ-3649)
- Corrected logging to only log that a signal is sent to a process when it is actually issued. (TRQ-3638)
- pmem was not getting set correctly. With cgroups enabled, pmem is the amount of resident memory allocated per process where are process is given by the value of ppn. For example: `qsub -l nodes=1:ppn=2,pmem=250mb` will allocate a total of 500 MB on the node where the job is run, 250 MB per ppn. (TRQ-3628)
- NUMA -L syntax defaulted to override user-specified parameters. (TRQ-3623)
- qstat -x returned nothing (instead of an empty XML document) when there are not jobs queued. (TRQ-3622)
- Jobs in which a task required more than one socket could not be started using NUMA -L syntax. (TRQ-3618)
- \$usecp parameter was ignored when specifying which directories should be staged. (TRQ-3613)
- Server deadlocked when job\_save() failed. (TRQ-3605)
- Tasks' memory usage was sometimes not reported. (TRQ-3601)
- Crash/infinite loop when loading certain node usage files. (TRQ-3576)
- Interactive jobs skipped submit filter directives if the first line was not #PBS. (TRQ-3585)
- Issue reported with alps login nodes. Updated cpusets for alps login nodes so that all of the cpus are in the job's cpuset. (TRQ-3568)
- Torque crashed intermittently when using the -L syntax. (TRQ-3566)
- Torque returned non-specific network failure messages to Moab. (TRQ-3539)
- Completed jobs were still reported in pbsnodes. (TRQ-3525)
- A deadlock occurred when handling job dependencies. (TRQ-3519)

- cgroup directories were not removed when jobs were completed. (TRQ-3515)
- drmaa unable to link with Torque. (TRQ-3511)
- Epilogue not showing up in momctl -d3 output. (TRQ-3495)
- Job dependencies were not being cleared with High Availability server. (TRQ-3477)
- A shell escape in pbs\_mom's config file when specifying GRES did not show up in pbsnodes or Moab. (TRQ-3393)
- libtorque.so was not being created. (TRQ-3374)
- qrls gave no response and logged no problem when a failure occurred due to a slot limit restriction. (TRQ-3328)
- Problems building RPMs on Red Hat 6/CentOS 6 systems. (TRQ-3283)
- Jobs started even if mother superior could not resolve the hostname for a sister node. (TRQ-3134)
- Several log messages were unclear. (TRQ-2860)
- Job holds were not updated when the slot limit was changed for a job array. (TRQ-2360)

#### 6.0.1

- 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)
- 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)
- 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)
- 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)
- Error condition where the mom's port would be inserted into the .JB file name. (TRQ-3090)
- 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)
- Completed jobs were not getting cleaned up. Fixed various issues when restarting dependency jobs, including them not getting removed even after completion. (TRQ-3175)
- 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)

#### 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

- 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)
- Read timeouts were being retried indefinitely by pbs\_server. (TRQ-3306)
- Reporter mom did not correctly handle UNKNOWN role. (TRQ-3245)
- 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)
- Interactive jobs not staying on the node from which they were submitted. (TRQ-3122)
- Jobs were getting stuck in a running state when an asynchronous run failed. (TRQ-3114)
- Array slot limits were not getting decremented when a job is preempted or rerun. (TRQ-3110)
- 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)