

Moab HPC Suite - Enterprise Edition 8.0.1 release notes

The release notes file contains the following sections:

- [New Features on page 1](#)
- [Differences on page 7](#)
- [Installation and Upgrade Information on page 11](#)
- [Known Issues on page 13](#)
- [Resolved issues on page 14](#)

New Features

The following is a summary of key new features in Moab HPC Suite - Enterprise Edition.

- [Moab Workload Manager on page 1](#)
- [Moab Web Services on page 4](#)
- [Moab Accounting Manager on page 5](#)
- [TORQUE Resource Manager on page 6](#)

Moab Workload Manager

8.0.1

Added Support for "flags=allprocs"

You can now request one or more nodes and Moab will allocate all the processors on every node that is assigned to the job. For example, ``qsub -l nodes=1,flags=allprocs``.

Energy-Consumption-by-Job Accounting

The Moab HPC Suite - Enterprise Edition has the ability to report, record, and charge a cost for the electrical energy consumed by a job. Currently, Moab 8.0.x supports this capability only for Cray XC systems running CLE 5.2 or later. See Appendix L in the Moab HPC Suite - Enterprise Edition Administrator Guide for more information on configuring power management and tracking energy consumption.

8.0.0

Data staging



The old method of data staging has been deprecated in Moab Workload Manager 8.0 and will be removed from the product in a future release.

Moab data staging has undergone a major redesign in 8.0. Consequently, Moab includes several new features that improve the data staging experience. These features include the following:

- Moab data staging system jobs, separately schedulable from the user job, that increase system performance and utilization by not reserving compute nodes during input or output data-staging unless a compute node's local file system is involved.
- New `msub` options that are required to submit data staging jobs. At least one of `--stagein`, `--stageinfile`, `--stageout`, and/or `--stageoutfile` is required to stage data for your job.
 - `--stagein` and `--stageout` specify individual files and/or directories to stage in or out, respectively.
 - `--stageinfile` and `--stageoutfile` specify the path to an individual file that contain the paths of files and/or directories to stage in or out, respectively.
- `--stageinsize` and `--stageoutsize` options that give Moab an estimate of the size of the files or directories to stage in or out, respectively, so it can more accurately schedule the associated data staging jobs. For more information, see `msub` in Scheduler Commands in the Moab Workload Manager Administrator Guide for more information.
- A customizable reference data staging job submit filter that sums the size of all input files and passes the sum to Moab. It uses the `--stageinsize` option so you do not have to estimate an input data size and Moab can more accurately estimate wallclock time when scheduling the input data staging system job. See Configuring data staging in the Moab Workload Manager Administrator Guide for more information.
- Customizable reference scripts that use `scp` or `rsync` to stage data (`/opt/moab/tools/data-staging/ds_move_scp` and `/opt/moab/tools/data-staging/ds_move_rsync`, respectively). The scripts work in an out-of-the-box environment, but you should modify one or both to work with your unique environment. You can use the reference scripts as a guide to create your own script that supports data staging with a different Linux file transfer utility or commercial file transfer utility, such as one from Aspera.
- The ability to set generic metrics on partitions (for more information, see Per-Partition Settings in the Moab Administrator Guide).
 - The new `DATASTAGINGBANDWIDTH_MBITS_PER_SEC` metric, required for data staging, specifies the transfer rate of the partition in megabits per second. Moab uses it and the `--stageinsize` and/or `--stageoutsize` `msub` options submitted with the job to estimate the wallclock time for better scheduling of the input and output data staging system jobs.
 - The new metric can be dynamically updated by a resource manager or configured by an administrator.

- When you run `checkjob -v` on a data staging job, Moab returns the source and destination of data staging files and their total size. See `checkjob` in Scheduler Command in the Moab Administrator Guide for more information.
- Advanced configuration options allow you to rename the default template, support multiple file transfer script utilities in a grid on a per-partition basis, configure a notification email to be sent after a data staging job completes, add a non-default template via `msub`, and use `msub` to return all job IDs in the workflow at submission time. This is useful when you dynamically generate user workflows using scripts that must create dependencies on data-staging system jobs (when input or output data-staging has completed, for example), not the user job. For more information, see Configuring data staging with advanced options in the Moab Administrator Guide.

The features associated with data staging in previous Moab releases have been deprecated in this release and will be removed in the next. For more information about data staging in Moab Workload Manager, see About data staging in the Moab Administrator Guide.

CPU frequency control

Moab can now ask the resource manager to change the CPU frequency on allocated nodes for submitted jobs. The request is made with the new CPUCLOCK resource manager 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 the Resource Manager Extensions in the Moab Administrator Guide.

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

CPUCLOCK has also been added as a job template extension. The job template extension overrides the job script CPUCLOCK extension and the job submission CPUCLOCK option.

Improved Performance and Scalability

Moab 8.0 includes new multi-threaded scheduling routines and is now compiled with optimizations enabled (`-O`). These improvements should increase scalability, efficiency, and performance for the majority of sites. The size of the thread pool can be throttled using the `THREADPOOLSIZE` parameter.

Finer-grained Logging Timestamps

The timestamps in the Moab logs now include milliseconds, which can be helpful with higher `LOGLEVEL` settings.

Deleting checkpoint file doesn't reset IDs

Moab now persists counters for job and reservation IDs to an external file (`<moab home>/counters`), in addition to persisting the counters to checkpoints. The new external counter file is created during installation. On startup, Moab reads the ID counters from either or both sources and uses the maximum of the two values that it sees as the resumption point for new IDs. This means that you can safely delete a checkpoint file without causing your IDs to be reset.

If both the external counter file and the checkpoint file are missing, Moab refuses to start. This is a precaution to protect the integrity of a relational database or external processes that rely on the IDs. If you need to override this behavior, you can do so by manually creating a new external counter file that starts IDs at an arbitrary number.

CANCELFAILEDDEPENDENCYJOBS scheduler flag

The new **CANCELFAILEDDEPENDENCYJOBS** scheduler flag automatically cancels dependency jobs that will never run because of an unmet requirement. For more information, see Job Dependencies in the Moab Administrator Guide.

Reduction in command processing time

If your system's scheduling cycle regularly takes longer than the **CLIENTTIMEOUT** value, you can configure Moab to fork a copy of itself that will respond to certain information-only client commands (`checkjob`, `showbf`, and `showstart`). This enables you to run intense diagnostic commands while Moab is in the middle of its scheduling process. For more information, see Reduce command processing time in the Moab Administrator Guide.

New --workflowjobids option for msub

A new `--workflowjobids` option for the `msub` command returns all the job IDs, including data-staging system job IDs, in a data-staging workflow at submission time (Moab creates an internal workflow for jobs submitted with data-staging options). For more information, see Configuring data staging with advanced options in the Moab Administrator Guide.

NAMI interface improvements

The Native Accounting Manager Interface now implements all charge policies and chargeable properties implemented by the MAM Accounting Manager Interface.

New support diagnostic script

The `support.diag.pl` script used for gathering information for Adaptive support representatives has been deprecated with the 8.0.1 release. Use the new `support-diag.py` script instead. For more information, see Diagnostic Scripts in the Moab Administrator Guide.

Moab Web Services

8.0.1

No new features.

8.0.0

PAM authentication support

Support for PAM (pluggable authentication module) has been added to MWS. PAM treats the user as if it is local to the machine doing the authenticating, and it uses whatever the user is authenticating with, whether it be LDAP or NIS. For information about configuring PAM with MWS, see PAM (pluggable authentication module) configuration using `mws-config.groovy` in the Moab Web Services Administrator Guide.



There is a security risk when authenticating local users through your PAM configuration. This behavior is highly discouraged and is not supported by Adaptive Computing.

OAuth authentication support

Support for OAuth has been added to MWS. OAuth allows trusted client applications to securely delegate authentication to MWS. Once MWS has authenticated a user by verifying the username and password in LDAP, PAM, or NIS, MWS returns an access token to the client. The client then presents this access token to MWS to access resources. For information about configuring PAM with MWS, see OAuth configuration using `mws-config.groovy` in the Moab Web Services Administrator Guide.

Node power management plugin

The MWS power management plugin acts as a power resource manager; it monitors the power states of the nodes in the Node Configuration File using the TORQUE `pbsnodes` command and a query script. The plugin also allows you to customize the power state of each node when it should be "turned off." For more information, see the documentation included with the plugin in Moab Web Services.

Fairshare policy added to MWS Policies object

The Moab "fairshare" policy has been added to the Policies object in MWS. With this addition, you can get fairshare policy information and make fairshare policy modifications through the MWS API. For more information, see Policies in the Moab Web Services Administrator Guide.

Modifiable attributes added to the MWS job resource

You can now use MWS to modify an idle job's system priority, the nodes that the job requests, and the resources per task required for the job. For more information, see Modify job attributes in the Moab Web Services Administrator Guide.

View and modify single credentials

You can now query a single credential in MWS. Additionally, you can modify a credential's fields and use the `change-mode` parameter to modify the credential's list fields. For more information, see Credentials in the Moab Web Services Administrator Guide.

View and modify scheduler priorities

You can now query and modify the scheduler priorities through the Priority resource in MWS. For more information, see Priority in the Moab Web Services Administrator Guide.

Moab Accounting Manager

8.0.1

No new features.

8.0.0

Accounting mode

A new **accounting.mode** parameter has been introduced which modifies the way in which accounting actions (e.g. charges, lien, quotes, refunds, etc.) are processed. The accounting mode may be selected at initial product configuration or changed later. The following are valid values for the accounting mode.

- `strict-allocation` - Use this mode if you wish to strictly enforce allocation limits. Under this mode, jobs can be prevented from running if the end-users do not have sufficient funds. This is the default.
- `fast-allocation` - Use this mode if you wish to debit allocations, but need higher throughput by eliminating the lien and quote of strict-allocation mode.
- `notional-charging` - Use this mode if you wish to calculate and record charges for workload usage, but not keep track of fund balances or allocation limits.
- `usage-tracking` - Use this mode if you wish to record workload usage details, but not to calculate a charge nor keep track of fund balances or allocation limits.

Commands reference

The Moab Accounting Manager Administrator Guide now contains Appendix A: Commands Reference that provides the equivalent of a man page for each command.

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 Administrator 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 Administrator 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 Administrator 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 Administrator Guide.

Differences

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

- [Moab Workload Manager on page 7](#)
- [Moab Web Services on page 9](#)
- [Moab Accounting Manager on page 9](#)
- [TORQUE Resource Manager on page 10](#)

Moab Workload Manager

8.0.1

MAXPROCPERNODE Can Also Be Configured Per Node

Added capability to configure MAXPROCPERNODE per class and per node. Before, MAXPROCPERNODE could be configured per class but it applied to all nodes. Now you can configure something like the following:

```
CLASSCFG[cpu] MAXPROCPERNODE [n1, n2, n3]=20 MAXPROCPERNODE [n4, n5, n6]=10
```

8.0.0

Green policy scripts might need updated values

If you do string comparisons in your green scripts, you might need to convert the incoming strings from Moab to all uppercase or lowercase.

NAMI scripts and interface changes

The Native Accounting Manager Interface scripts and interface now use the same SSS XML sent over the wire to MAM for the MAM Accounting Manager Interface type. The default Native AM configuration uses the following supplied NAMI scripts:

```

AMCFG [mam] TYPE=NATIVE
AMCFG [mam] CREATEURL=exec://$TOOLSDIR/mam/usage.quote.mam.pl
AMCFG [mam] STARTURL=exec://$TOOLSDIR/mam/usage.reserve.mam.pl
AMCFG [mam] UPDATEURL=exec://$TOOLSDIR/mam/usage.charge.mam.pl
AMCFG [mam] PAUSEURL=exec://$TOOLSDIR/mam/usage.charge.mam.pl
AMCFG [mam] RESUMEURL=exec://$TOOLSDIR/mam/usage.reserve.mam.pl
AMCFG [mam] ENDURL=exec://$TOOLSDIR/mam/usage.charge.mam.pl
AMCFG [mam] DELETEURL=exec://$TOOLSDIR/mam/lien.delete.mam.pl

```

These new scripts utilize the MAM Perl API and logging; they no longer use the `nami.cfg` or `nami.log` files. If you have created custom NAMI scripts, you will need to adjust them to parse the SSS XML. See Charging and Allocation Management in the Moab Workload Manager Administrator Guide for examples of the new SSS syntax.

In previous releases, the **UPDATEURL** NAMI script fulfilled the dual function of charging for the previous interval and obtaining a lien for the next interval. In this release, the **UPDATEURL** script only charges for the previous interval and the **RESUMEURL** script obtains a lien for the next interval (following both **PAUSEURL** and **UPDATEURL**).

mshow -a policy flag deprecated

The `mshow -a policy` flag (`mshow -a --flags=policy`) has been deprecated and may be removed in a future release.

RPM installations of MAM based on PostgreSQL database

Because the Moab Suite utilizes a PostgreSQL database in this release, new RPM installations of Moab Accounting Manager are based on a PostgreSQL database instead of MySQL.

Moab versions have been condensed to one build

There is now only one Moab build to download, install, and configure. In past releases, customers have had to choose between Moab builds that were compiled against ODBC and/or TORQUE libraries. Moab is still compatible with ODBC and TORQUE libraries, however there is now only one package to download. Use of these libraries is enabled/disabled with the existing configuration parameters in `moab.cfg` (e.g., `USEDATABASE ODBC`).

CLASSCFG MAXPROCPERNODE no longer solely global

You can now specify individual nodes with the **CLASSCFG MAXPROCPERNODE** attribute, which limits the number of processors a class can have on a node.

```
CLASSCFG [batch] MAXPROCPERNODE [n1, n2, n3]=2
```

The batch class is limited to 2 processors on nodes n1, n2, and n3.

Complicated workflows contained in a single VC

Jobs that are created from a workflow template are now associated with a single VC (Virtual Container). Previously, for complex workflows, multiple VCs were created that each contained portions of the workflow jobs.

Moab Web Services

8.0.1

Node Power States

Moab only supports node power states of "On" and "Off". The API has been updated accordingly. See Power Management Plugin in the Moab Web Services Administrator Guide for details on supporting a larger set of node power states.

Reporting Plugins Are No Longer Turned On By Default

The reporting plugins (i.e., VMUtilizationReport and NodeUtilizationReport) are no longer turned on by default during installation. They are still available for use for those wishing to enable them.

Tenancy Plugin Removed

The MWS tenancy plugin was removed in 8.0.

8.0.0

API version 1 removed

API version 1 was deprecated in 7.5.0 and has been removed from Moab Web Services in this release. It is highly recommended that you use the most recent API version (version 3). See Requesting specific API versions in the Moab Web Services Administrator Guide for more information.

Moab Accounting Manager

8.0.1

No known differences.

8.0.0

The low-level fund balance action has been removed

The low-level fund balance action has been removed since equivalent functionality exists by using the allocation query action. The `--total`, `--available` and `--ignore-liens` options have been removed from the `gbalance` command.

Common logging format

Logging has been enhanced to follow the new formatting guidelines for the Adaptive Computing Standard Log Format. This allows added predictability for common log parsing across Moab suite components.

Consumption Rate passed via attribute

Charging routines look for a `consumptionRate` attribute on each datum property individually instead of looking for a datum property called `ConsumptionRate`. This way, consumption rates can be attributed to the specific property to which it applies instead of indiscriminantly applying to all preadd-rate properties.

SQLite database no longer supported

The SQLite embedded database is no longer an option as of this release. It lacked sufficient functionality to allow object customizations and upgrades.

PostgreSQL 9.3 as default database

PostgreSQL 9.3 is now offered as the new default database. Customers that are already using MySQL or earlier versions of PostgreSQL may continue to run on these databases but new customers are encouraged to use PostgreSQL 9.3.

TORQUE Resource Manager

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 $((2 * \text{the number of procs listed in /proc/cpuinfo}) + 1) * 20$. 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.

Installation and Upgrade Information



When installing or upgrading, it is *strongly* recommended that administrators configure Moab with mauth authentication with a complex key value. See Mauth Authentication (Appendix E: Security) in the Moab Workload Manager Administrator Guide for more information.



Moab 8.0.x supports TORQUE 4.2.8 and later. However, TORQUE 5.0.x requires Moab 8.0.x.

i Moab has been tested with the latest versions of Postgres 9.3, MySQL 5.6, and Oracle 12c.

Installing Moab HPC Suite – Enterprise Edition 8.0.1

See the Moab HPC Suite Installation and Configuration Guide for requirements and instructions.

Installing TORQUE 5.0.x on RHEL 5

TORQUE requires the download of boost version 1.36 or later header files in order to build on most RHEL 5 installations. TORQUE 5.0.x needs the `unordered_map` from the boost library. This did not become part of the boost standard until boost version 1.36. Standard RHEL 5 installations only have boost version 1.33.1. TORQUE 5.0.x has added a configure option named `--with-boost-path`. This option allows the user to be able to download a newer version of boost without the need to install boost.

You can download a new boost library from www.boost.org. Extract the tarball to a directory of your choosing. When you are configuring TORQUE you then add the `--with-boost-path=<boost_path>` to the configuration line. The `boost_path` is the location where you unzipped the boost library. Within the unzipped files is a directory named `boost`. This directory contains the `.hpp` files with the needed boost classes. Once the `--with-boost-path` option has been added to the configuration line you can continue to install TORQUE as per the installation guide.

i This procedure is not required on RHEL 6 and later versions.

Upgrading to Moab HPC Suite – Enterprise Edition 8.0.1

See the Moab HPC Suite Installation and Configuration Guide for requirements and instructions.

Moab Accounting Manager Database Requirements

Moab HPC Suite 8.0.x is only compatible with a Moab Accounting Manager database schema version of 8.0. You can determine your current database schema version by running `goldsh System Query`. If your current database schema version is less than 8.0, you will need to migrate your database to 8.0. See [Upgrading Moab Accounting Manager](#) in the Moab HPC Suite Installation and Configuration Guide for instructions on migrating the dataase schemea.

Moab Database Schema Updates

The Moab database schema has been updated for 8.0.

Table name	New additions	Changes
Nodes	<ul style="list-style-type: none"> CPUclock (VARCHAR (64)) 	<ul style="list-style-type: none"> Column name "Partition" changed to "PartitionName". The word "partition" became a reserved word as of MySQL 5.6

Table name	New additions	Changes
Requests	<ul style="list-style-type: none"> • CPUClock (VARCHAR (64)) 	n/a
NodeStats	<ul style="list-style-type: none"> • CPUClock integer • CPUMaxClock integer • CPUMinClock integer 	n/a
Reservations	n/a	<ul style="list-style-type: none"> • Column name "Partition" changed to "PartitionName". The word "partition" became a reserved word as of MySQL 5.6

Migrating Your Database to Newer Versions of Moab in the Moab Workload Manager Administrator Guide for instructions on migrating your database schema.

Known Issues

The following are known issues in Moab HPC Suite - Enterprise Edition. 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.

8.0.1/5.0.1

- The node collection in the moab database in MongoDB has an index on the attributes field. This field can grow too large to index. (MOAB-7506)
 - For existing installations, the following commands on the MongoDB server will fix the problem:

```
$ mongo moab -u moab user -p secret2
> db.node.dropIndex({"attributes":1})
```

The username and password for your database are most likely different from the above example. Check with your database administrator.

- For new installations using this and future releases, the index is no longer created and does not need to be dropped.
- Jobs submitted with invalid credentials are put in a held state, instead of rejected, until the administrator can respond. The checkjob command gives administrators further information regarding why the job is held. Blindly assuming that all held jobs should in fact be running RIGHT

NOW is not only unsafe, but circumvents intentional Moab policies and workflow. An administrator should exercise care when resolving held jobs. (MOAB-7478)

- 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 pbsdsh to hang). (TRQ-2890)
- If the Moab **JOBNODEMATCHPOLICY** is set to **EXACTNODE** and `requirements.tasksPerNode` is used in a job submission to MWS, then Moab will double the resources requested. To avoid this problem, use `requirements.resourcesPerTask.processors.dedicated` instead. (MOAB-7424)
- **BACKFILLPOLICY BESTFIT** does not support multi-req jobs. Only **FIRSTFIT** supports multi-req jobs. (MOAB-6824)

8.0.0/5.0.0

- Connecting Moab to MySQL via ODBC can cause Moab to slow down. You can work around this issue by creating the tables with the old storage engine (ENGINE = MyISAM). Also note that other factors will also affect DB performance, such as the type of DB connection (remote vs. local), the DB server's hardware specifications, Moab's configuration for how much to write to the DB (see **REALTIMEDBOBJECTS**), etc (MOAB-6316).
- ~~Sites that used RPMs to install Moab HPC Suite will not be able to upgrade to 8.0.0 RPMs until 7.2.9 is released. The issue exists in all pre-8.0 RPMs and cannot be worked around without uninstalling the suite. Thank you for your patience as we work to resolve these upgrade issues. Until the issues are addressed, we suggest a fresh installation (either manual or RPM) on a separate machine (MOAB-7192).~~ This issue has been resolved when upgrading from Moab HPC Suite 7.2.9.

Resolved issues

The following is a list of some key bugs fixed in Moab HPC Suite – Enterprise Edition. 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.

8.0.1/5.0.1

- **Submitting remote job scripts (that is, including the script as part of the JSON payload) is not currently supported.** Job scripts can now be included in the POST body when submitting jobs via MWS. The new field is called `commandScript`. (WS-2112)
- **A job could charge the wrong allocation if it had no lien and was out of funds but another lien was present.** This issue is fixed. (MAM-267)
- **Incorrect proportions could be charged to expired allocations.** This issue is fixed. (MAM-262)

- **Some systems can get "Insecure \$ENV{ENV} while running with -T switch" when using the gauth security promotion mechanism.** This issue is fixed. (MAM-276)
- **Split charges could occur or liens could be denied because charges, liens, and quotes were factoring in inactive liens** This issue is fixed. (MAM-274)
- **Migration scripts had PATH problem when using promotion method gauth.** Fixed problem when using promotion method. (MAM)
- **Role permission updates were incorrectly applied to user permissions.** This issue is fixed. (WS-2340)
- **Moab was not decrementing GRES correctly within a Cray environment.** Fixed a GRES bug found in Cray environments. (MOAB-6577)
- **multireq jobs take hours to start.** Fix bug where multi-req jobs were slow to start in certain cases. (MOAB-6824)
- **When using .moab.key and mauth, having the string "actor=" in an environment variable in the job script causes mauth to fail.** Fixed bug in environment variable handling in mauth. (MOAB-6908)
- **Jobs were not rejected even though they violated fairshare policy.** This bug is fixed. (MOAB-7077)
- **support-diag.py does not processes backtrace information correctly.** This bug is fixed. (INT-808)
- **Script fails when more than one moab service is running on the same system.** Updated support-diag.py to better handle cases where more than one moab service is running simultaneously. (INT-811)
- **Moab changes task count on jobs on restart.** Fixed bug around incorrect task counts on multi-req jobs across a Moab restart in grid environment. (MOAB-7104)
- **No quote in MAM when NODEACCESSPOLICY is SINGLEJOB.** Fixed bug around incorrect accounting quote (MAM) when NODEACCESSPOLICY was set to SINGLEJOB. (MOAB-7213)
- **Extra zero charges on some jobs.** Added a fix to prevent an extraneous zero charge for a canceled idle job after it has been requeued. (MOAB-7637)
- **Reservation consumption rate charging was not working correctly.** Fixed bug related to accounting consumption rate. (MOAB-7172)
- **Jobs purged within Moab could leave a disconnected reserve usage record in MAM.** Fixed issue to prevent the extraneous lien delete ensuring a combined usage record. (MOAB-7353)
- **Moab was still blocking on *checkjob* and *showres*.** Added *showres* to the list of commands available when UIMANAGEMENTPOLICY is set to FORK. (MOAB-7233)
- **ALERT log message did not end in a new line.** Added a missing new line character to a log message. (MOAB-7329)
- **Reservation was not being created when some resources are available.** Fixed an issue with standing reservations that occasionally weren't created when they should be. (MOAB-7384)

- **Moab XML submission not working.** Fixed some API bugs preventing job submission via XML (Moab Wire Protocol). (MOAB-7485)
- **Some bugs found that caused Moab 8.0 crashes.** Fixed bugs that were causing crashes.
- **Issue found with sqlite 8.0.0 integration.** Fixed an issue where some of the sqlite integration scripts were not being packaged properly.
- **Moab was unable to modify the hostlist of a idle job correctly.** Fixed bug with modifying hostlists on idle jobs. (MOAB-7490)
- **nallocpolicy=cpload was being ignored.** Fixed bug where requested nallocpolicy was inadvertently getting overridden. (MOAB-7443)
- **UIMANAGEMENTPOLICY caused Moab to hang.** Fixed a bug where UIMANAGEMENTPOLICY caused Moab to hang. (MOAB-7578)
- **Job dependencies were not consistently working as expected.** Fixed bug related to job dependencies getting out of sync with TORQUE in some cases. (MOAB-7472)
- **Performance bug found in ReportJob.groovy.** Fixed scalability issue with large sample sets (e.g., node utilization). (WS-2273)
- **Services stuck in Deploying state.** Fixed bug where some services were getting stuck in Deployed state. (WS-2293)
- **Slow performance reported on large LDAP systems.** Fixed bug with LDAP integration to improve performance on large LDAP systems. (WS-2252)
- **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)

8.0.0/5.0.0

- **Moab attempted to power off drained nodes.** Moab now skips drained nodes when it powers off nodes. (MOAB-6036)
- **Moab Authentication Bypass issue.** This has been fixed. (CVE-2014-5300, MOAB-7100, MOAB-7524)
- **In environments where a license RM was configured and created a global node, Moab would try to power the global node on and off.** Now Moab does not allow the global node to be powered down as part of a pool of eligible nodes. (MOAB-6608)
- **The mdiag -n command did not return the correct amount of available memory on the node.** Running mdiag -n once again returns the correct amount of available memory. (DOC-6708)
- **Moab did not honor features configured in the DEFAULTNODEFEATURES parameter.** The DEFAULTNODEFEATURES parameters once again works as expected.
- **When Moab would rerun a job after the job migrated to another partition, it attempted the rerun on the wrong partition.** Moab reruns migrated jobs in the correct partition. (MOAB-6734)
- **When using a JOBMIGRATEPOLICY of IMMEDIATE, Moab did not call the CREATEURL Native Accounting Manager Interface script.** Moab now calls the CREATEURL NAMI script. (MOAB-6867)
- **Running msub and qsub in an interactive environment produced different results.** Running msub -l no longer hangs in an interactive environment. (MOAB-7165)
- **Moab ignored requirements set using the "hostlist" parameter.** Moab now honors hostlist requirements. (MOAB-6475)
- **The MAM GUI did not display as expected.** The MAM GUI was fixed to display the right blue pagination arrow properly. (GOLD-238)
- **Certain commands did not recognize accounts that contained a '+'. Plus signs (and other non-alpha characters) now work correctly in account names.** (GOLD-249)
- **MAM liens were not clearing.** The charge now removes liens if the lien amount is zero. (GOLD-250)
- **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)

Resolved issues

- **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)