Red Hat Enterprise MRG 2 Technical Notes

Detailed notes on the changes implemented in Red Hat Enterprise MRG 2

Tomáš Čapek
David Ryan
Cheryn Tan
Joshua Wulf

Red Hat Enterprise MRG 2 Technical Notes

Detailed notes on the changes implemented in Red Hat Enterprise MRG 2

Tomáš Čapek
Red Hat, Inc. Engineering Content Services
tcapek@redhat.com

David Ryan
Red Hat, Inc. Engineering Content Services
dryan@redhat.com

Cheryn Tan
Red Hat, Inc. Engineering Content Services
cheryntan@redhat.com

Joshua Wulf
Red Hat, Inc. Engineering Content Services
jwulf@redhat.com

Keywords

Abstract

The Technical Notes book provides descriptions for all security issues, bug fixes, and enhancements released as part of Red Hat Enterprise MRG 2. Read this document before beginning to use the Red Hat Enterprise MRG distributed computing platform.

Table of Contents

Chapter 1. MRG Messaging on Red Hat Enterprise Linux 6

1.1. RHSA-2013:0562 – Moderate: Red Hat Enterprise MRG Messaging 2.3 security, bug fix, and enhancement update

Red Hat Enterprise MRG is a next-generation IT infrastructure incorporating Messaging, Realtime, and Grid functionality. It offers increased performance, reliability, interoperability, and faster computing for enterprise customers.

MRG Messaging is a high-speed reliable messaging distribution for Linux based on AMQP (Advanced Message Queuing Protocol), an open protocol standard for enterprise messaging that is designed to make mission critical messaging widely available as a standard service, and to make enterprise messaging interoperable across platforms, programming languages, and vendors. MRG Messaging includes an AMQP 0-10 messaging broker; AMQP 0-10 client libraries for C++, Java JMS, and Python; as well as persistence libraries and management tools.

Descriptions of the security fixes provided in this advisory can be viewed at https://rhn.redhat.com/errata/RHSA-2013-0562.html. The changes in this advisory for other non-security bugs and enhancements are the same as those for the Red Hat Enterprise Linux 5 MRG Messaging 2.3 Release. Refer to section Section 2.1, “ RHSA-2013:0561 - Moderate: Red Hat Enterprise MRG Messaging 2.3 security, bug fix, and enhancement update ”.

1.2. RHSA-2012:1279 – Moderate: Red Hat Enterprise MRG Messaging 2.2 security, bug fix, and enhancement update

Red Hat Enterprise MRG is a next-generation IT infrastructure incorporating Messaging, Realtime, and Grid functionality. It offers increased performance, reliability, interoperability, and faster computing for enterprise customers.

MRG Messaging is a high-speed reliable messaging distribution for Linux based on AMQP (Advanced Message Queuing Protocol), an open protocol standard for enterprise messaging that is designed to make mission critical messaging widely available as a standard service, and to make enterprise messaging interoperable across platforms, programming languages, and vendors. MRG Messaging includes an AMQP 0-10 messaging broker; AMQP 0-10 client libraries for C++, Java JMS, and Python; as well as persistence libraries and management tools.

Security Fixes

CVE-2012-3467

It was discovered that the Qpid daemon (qpidd) did not require authentication for "catch-up" shadow connections created when a new broker joins a cluster. A malicious client could use this flaw to bypass client authentication.

Bug Fix

BZ#825078

When a broker was configured to require client authentication using the **`ssl-require-client-authentication=yes`** option, the QPID Management tools could not provide authenticated credentials. Consequently, these tools were failing with connection errors while attempting to manage QPID brokers. This update adds the **`--ssl-certificate`** command-line option for the QPID Management tools. This option allows the user to supply a file

containing the correct credentials for the required authentication and the tools can now be used to manage a QPID broker that is configured to require client authentication.

The changes in this advisory for other non-security bugs and enhancements are the same as those for the Red Hat Enterprise Linux 5 MRG Messaging 2.2 Release. Refer to Section 2.2, “ RHSA-2012:1277 – Moderate: Red Hat Enterprise MRG Messaging 2.2 security, bug fix, and enhancement update ”.

All users of the messaging capabilities of Red Hat Enterprise MRG 2.2 are advised to upgrade to these updated packages, which resolve this issue, fix these bugs, and add these enhancements. After installing the updated packages, stop the cluster by either running **`service qpidd stop`** on all nodes, or run **`qpid-cluster --all-stop`** on any one of the cluster nodes. Once stopped, restart the cluster with **`service qpidd start`** on all nodes for the update to take effect.

1.3. RHSA-2012:0528 – Moderate: Red Hat Enterprise MRG Messaging 2.1 security, bug fix and enhancement update

Red Hat Enterprise MRG is a next-generation IT infrastructure incorporating Messaging, Realtime, and Grid functionality. It offers increased performance, reliability, interoperability, and faster computing for enterprise customers.

MRG Messaging is a high-speed reliable messaging distribution for Linux based on AMQP (Advanced Message Queuing Protocol), an open protocol standard for enterprise messaging that is designed to make mission critical messaging widely available as a standard service, and to make enterprise messaging interoperable across platforms, programming languages, and vendors. MRG Messaging includes an AMQP 0-10 messaging broker; AMQP 0-10 client libraries for C++, Java JMS, and Python; as well as persistence libraries and management tools.

Security Fixes

CVE-2011-3620

It was found that Qpid accepted any password or SASL mechanism, provided the remote user knew a valid cluster username. This could give a remote attacker unauthorized access to the cluster, exposing cluster messages and internal Qpid/MRG configurations.

Note

If you are using an ACL, the cluster-username must be allowed to publish to the qpid.cluster-credentials exchange. For example, if your cluster-username is "foo", in your ACL file:

```
acl allow foo@QPID publish exchange name=qpid.cluster-credentials
```

The CVE-2011-3620 fix changes the cluster initialization protocol. As such, the cluster with all new version brokers must be restarted for the changes to take effect. Refer below for details.

Bug Fixes

BZ#748126, BZ#755611

When a failover occurred, some versions of JMS clients sent an invalid sequence of AMQP (Advanced Message Queuing Protocol) controls, duplicating the AMQP 0-10 **`connection-tune-ok`** control. Consequently, the broker left invalid pointers for timers connected with heartbeats, which caused the broker to terminate unexpectedly with a segmentation fault. With this update, invalid sequences of controls are properly caught and sanitized and the broker no longer crashes in the described scenario.

BZ#760112

When an existing durable queue was recovered, management statistics data were not returned. Consequently, it was not possible to monitor the status of any recovered persistent queue. This bug has been fixed and the management statistics are now correctly initialized and can be queried in the same way as with newly created queues.

BZ#735208

Due to broken initialization logic in the store module, when an attempt to use management tools to view the store or journal statistics was made, no statistics were available even when the store was in use. This update fixes the initialization logic and the store module now correctly displays management statistics for the store plug-in.

BZ#738490

Previously, sessions, on which messages had been received from a queue, could delay aspects of that queue's deletion. Consequently, the queue object within the broker was not actually deleted and was still visible from tools but data could no longer be sent to or received from it. With this update, record of messages delivered from the deleted queue and acknowledged by the receiver is no longer kept in the broker after the subscription has been canceled, preventing the queue from being referenced after deletion. Now, deletion of a queue is no longer affected by the existence of sessions, through which messages from that queue have been received and acknowledged.

BZ#730981

Previously, the C++ client incorrectly tried to declare predefined exchanges without setting the passive flag as required by AMQP 0-10 specification for exchanges with reserved names. Consequently, attempts to create a sender (or a receiver) to (or from) a predefined exchange and utilize a create policy in the address in order to establish some node level bindings to it failed. With this update, exchanges with reserved names are treated as special cases and the passive flag is always set when declaring an exchange as part of the create policy. Now, bindings defined in the node section of an address whose named node is a predefined exchange can be established properly.

BZ#726102

Previously, using an address string with the C++ messaging API, which had a map within it with an empty key value, resulted in a parsing exception being returned. With this update, the parser has been updated to correctly handle empty values for key-value pairs in maps and the exception is no longer returned in the described scenario.

BZ#731368

Prior to this update, the example on the usage of the drain command shipped with the C++ client only consumed one message by default rather than all available messages. This update amends the example to work the same way as similar examples for Python and Java clients and

all drain examples now behave consistently in consuming all available messages.

BZ#729441

When adding and removing bindings from the direct exchange using unique keys, for example when creating and deleting lots of uniquely named temporary queues, the bindings were not fully cleared up on removal, causing memory leaks. With this update, the bindings are now properly cleaned up when removed and the memory leaks no longer occur in the described scenario.

BZ#798236

Previously, when browsing the LVQ (Last Value Queue), messages delivered to the browsing session build up indefinitely while the session remained. Consequently, unbounded memory growth and memory leaks occurred. Now, acknowledgments are correctly sent by clients for browsed messages and the broker also handles those acknowledgments to be no longer holding the delivered message against the session. As a result, unbounded memory growth no longer occurs in a browser of an LVQ.

BZ#769828

The store has a flow-to-disk option, in which both transient and durable messages are saved to disk and the message content is freed temporarily from memory. Previously, when several consumers either browsed or consumed flowed-to-disk messages, a race condition could develop and cause unpredictable broker behavior. This update adds a locking mechanism to prevent inconsistent thread states. It is now safe to concurrently access messages with content flowed to disk from multiple threads.

BZ#732369

Previously, the **`qpid-config --help`** returned ambiguous and confusing information regarding the host address option syntax. This update clears up the language in the help message to make it unambiguous, thus fixing this bug.

BZ#733241

When queues were listed using the **`qpid-config queues [queue_name]`** command, the qpid-config utility always returned **`0`**, regardless of whether the queue existed. Similar behavior was observed when listing exchanges. Consequently, scripts that use the return code of qpid-config to detect the presence of a queue or an exchange did not work correctly. This bug has been fixed and qpid-config now provides correct return codes in the described scenario.

BZ#758853

Previously, code that checks the setting of command-line options failed to properly detect options set to the value **`0`**, interpreting it as a missing value. Consequently, these options were ignored by the code, and set to their default value instead. This bug has been fixed and the value **`0`** in the command-line options is now properly recognized.

BZ#759114

In the watchdog module, the path to the watchdog executable was hard-coded. Consequently, it was not possible to use the watchdog feature if the watchdog executable was not in its default location. This update adds the **`--watchdog-exec`** option to the watchdog module to specify path to the qpidd_watchdog executable, thus fixing this bug.

BZ#693845

When a Python client provided a user name and password while authenticating with the broker using the **ANONYMOUS** method, the client incorrectly included the user name into each sent message. Consequently, the messages could not be passed to the broker. With this update, the user name is no longer placed into messages when **ANONYMOUS** authentication is used, thus fixing this bug.

BZ#773700

Previously, when the broker was restarted, QMF (QPID modeling framework) consoles sometimes became unresponsive. Consequently, Python-based QMF consoles failed to detect the broker was up again and were unable to monitor QMF agents. With this update, a timeout has been added to the connection clean-up logic, preventing the hang when a broker is restarted. Now, the console attempts to re-contact the broker periodically when a connection is lost. Once the broker is available, the console connects to it and resumes normal operation.

BZ#704596

Previously, the value of the **toString()** function on the JMS destination of a message being sent defaulted to the BURL syntax. Consequently, the output of **toString()** from a sender and a receiver side printed the JMS destination in different syntax, causing confusion. With this update, the code uses the default syntax of the client when creating the JMS destination from the incoming message and the **toString()** output is now of the same syntax for both a sender and a receiver in the described scenario.

BZ#704608

Previously, Red Hat Enterprise MRG only supported an integer value for the ***JMSDeliveryMode*** message header field (for example **JMSDeliveryMode = 2**). This violated the Java Message System (JMS) specification, which requires ***JMSDeliveryMode*** to be used as a string in a selector (for example **JMSDeliveryMode = 'PERSISTENT'**). This bug has been fixed, and ***JMSDeliveryMode*** can now be used as a string in the selector.

Note

This update breaks backward compatibility. Users must change their code to use strings, not integers as before, to identify JMS delivery modes.

Enhancements

BZ#751845

Support in the broker for listening on SSL-encrypted sockets and plain TCP sockets is implemented by two distinct modules, each listening on its own port. Previously, it was not possible to serve both SSL and non-SSL connections from the same port. A deployment requiring both types needs to advertise two port numbers, which was not always possible or convenient. With this update, the SSL module has been changed to optionally serve plain (non-SSL) connections using the same port. Now, a single port can be advertised that will support both SSL and non-SSL traffic. This is enabled by setting the ***--ssl-port*** and ***--port*** options to the same value.

Note

Note that under this configuration, there is at present no support for IPv6 addresses.

BZ#727182

This update adds support for DTX (distributed transactions) in clusters to the *qpid-cpp* package.

BZ#734115

Support for message grouping with strict sequence consumption across multiple consumers has been added to the *qpid-cpp* package.

BZ#674379

Support for the IPv6 protocol has been added to the **qpid** C++ libraries for MRG Messaging.

BZ#650969

Support for Microsoft Visual Studio 2010 has been added to the *qpid-winsdk* package.

BZ#760636

The QMF Broker method **query** now returns extra detail for a message group queue's internal state.

BZ#705418

Previously, the Session class could only acknowledge a single specified message, or all outstanding messages. With this update, a method has been added to the API to cumulatively acknowledge all messages up to and including a given message in a single method call. This feature saves on round trip times to the broker.

BZ#761186

The JMS client now sets the **TCP_NODELAY** property to **true** by default as it shows an improvement in many general cases. If there is a configuration error, such as an error in connection URL, this property will still be set to **true**.

Note

Note that for high-throughput scenarios, it might be useful to turn **TCP_NODELAY** off as with this setting, packet overhead is reduced and congestion collapse is prevented. For further details, refer to the section 3.2.2. Connection URLs in the programming guide.

All users of the messaging capabilities of Red Hat Enterprise MRG 2.1 are advised to upgrade to these updated packages, which resolve this issue, fix these bugs, and add these enhancements. After installing the updated packages, stop the cluster by either running **service qpidd stop** on all nodes,

or run **qpid-cluster --all-stop** on any one of the cluster nodes. Once stopped, restart the cluster with **service qpidd start** on all nodes for the update to take effect.

1.4. RHSA-2011:1399 – Moderate: Red Hat Enterprise MRG Messaging 2.0 bug fix update

Red Hat Enterprise MRG (Messaging, Realtime, and Grid) is a next-generation IT infrastructure for enterprise computing. MRG offers increased performance, reliability, interoperability, and faster computing for enterprise customers.

MRG Messaging is a high-speed reliable messaging distribution for Linux based on AMQP (Advanced Message Queuing Protocol), an open protocol standard for enterprise messaging that is designed to make mission critical messaging widely available as a standard service, and to make enterprise messaging interoperable across platforms, programming languages, and vendors. MRG Messaging includes an AMQP 0-10 messaging broker; AMQP 0-10 client libraries for C++, Java JMS, and Python; as well as persistence libraries and management tools.

Release Notes

Customers who are using the **qpid-cpp-server-xml** module must be subscribed to the following Red Hat Network channel in order to satisfy the XML dependencies for this release: Red Hat Enterprise Linux Server Optional. Customers who are using the **qpid-cpp-server-cluster** module must be subscribed to the Red Hat Enterprise Linux High Availability channel in order to satisfy all dependencies. If you encounter yum dependency problems, ensure that you are subscribed to all of the required channels as detailed above.

Bug Fixes

BZ#728586

Due to a regression, a memory leak was introduced that prevented the broker from correctly releasing messages. Consequently, the broker's memory footprint grew indefinitely. A patch has been provided to address this issue, and the memory leak no longer occurs in the described scenario.

BZ#734608

After the upgrade of MRG Messaging from version 1.2 to version 1.3, the **qpidd** daemon terminated unexpectedly on some of the cluster nodes. With this update, the broker code has been fixed to handle all cases when a faulty client sends frames before completely opening the connection, thus fixing this bug.

BZ#732063

Previously, specifying an invalid IP address for a destination broker caused a repeatable file descriptor leak. A patch has been provided to address this issue, and the file descriptor leak no longer occurs in the described scenario.

BZ#733543

Prior to this update, when a large message (over 4KB) was sent from the python-qpid client to the broker, the connection became unresponsive and other clients were unable to connect to the broker. This bug has been fixed, and clients no longer hang in the described scenario.

BZ#690107

Under heavy load, the broker generated a large number of timer late/overrun warning messages. Though the messages themselves were usually inoffensive, the time to log them individually caused long (up to several seconds) delays and inflated log files. With this update, logging output for these messages has been restricted to the **`--log- enabled=info`** option, thus preventing this bug.

Users of the messaging capabilities of Red Hat Enterprise MRG 2.0, which is layered on Red Hat Enterprise Linux 6, are advised to upgrade to these updated packages, which fix these bugs.

1.5. RHBA-2011:1340 – Red Hat Enterprise MRG Messaging 2.0 bug fix update

Updated Red Hat Enterprise MRG Messaging packages that fix several bugs are now available for Red Hat Enterprise MRG 2.0 for Red Hat Enterprise Linux 6.

Red Hat Enterprise MRG (Messaging, Realtime and Grid) is a real-time IT infrastructure for enterprise computing. MRG Messaging implements the Advanced Message Queuing Protocol (AMQP) standard, adding persistence options, kernel optimizations, and operating system services.

Release Note

Customers who are using the qpid-cpp-server-xml module must be subscribed to the following Red Hat Network channel in order to satisfy the XML dependencies for this release: RHEL Server Optional.

Customers who are using the qpid-cpp-server-cluster module must be subscribed to the RHEL High Availability channel in order to satisfy all dependencies.

If you encounter yum dependency problems, ensure that you are subscribed to all of the required channels as detailed above.

Bug Fixes

BZ#737177

The system threw a javax.naming.NameNotFoundException if the JNDI file definition in the executed program defined a JNDI file with a syntax error. However, this is not appropriate as the program cannot access the JNDI file and the execution was therefore interrupted. Such internal exceptions are now re-thrown as ConfigurationExceptions with the message "Failed to parse JNDI properties file" and the program is interrupted.

BZ#738352

The broker ignored QMFv2 requests sent from a Java client because the requests did not have the app-id set. The requests are now sent with the app-id and the messages are processed as expected.

BZ#738354

A queue bound to the default exchange dropped messages with a subject. This happened because the routing key was wrongly using the message subject as the queue name. The

underlying code has been changed and such messages are now delivered to the queue.

BZ#738357

When a connection was created with newly specified credentials, the Java client cached the new credentials and overwrote the default credentials provided by the connection URL. As a result, when a new connection was created without credentials, the connection used the cached credentials and the authentication could fail. The credentials specified at a connection creation are no longer cached and the default credentials for connections without credentials are used as expected.

BZ#738358

In a clustered environment, the system threw a SessionException when a session was committed after a failover using the **`session.commit()`** method. Because the client application code did not receive a standard JMSException, it could fail to recover. The system now throws a JMSException under these circumstances and the client recovers as expected.

BZ#738360

The system threw a SessionException if a session was committed after a transaction had exceeded the queue limit (max-queue-size). This caused the client to fail to handle the error. The system now throws a JMSException under these circumstances and client can recover with the JMSException as expected.

BZ#738361

The **`Connection.getJMSXPropertyNames()`** method returned a usable enumeration only when called for the first time. If the user called the **`getJMSXPropertyNames()`** method multiple times, the method returned empty enumerations. This happened because the system overwrote the first created enumeration with the subsequent call of **`getJMSXProperyNames()`**. The underlying code has been changed and **`getJMSXPropertyNames()`** returns the Enumeration value with the property names as expected.

BZ#738362

According to the JMS specification, concurrently executing clients cannot use the same client ID. Previously, it was possible to allow multiple connections to use the same client ID. With this update, it is no longer possible to define such clients. Note that you need to enable the verify client ID feature to prevent the clients from having identical IDs (-Dqpid.verify_client_id=true).

Users of the Realtime capabilities of Red Hat Enterprise MRG 2.0, which is layered on Red Hat Enterprise Linux 6, are advised to upgrade to these updated packages, which fix these bugs.

1.6. RHEA-2011:0892 – Red Hat Enterprise MRG – Messaging 2.0 Release

The changes in this advisory are the same as those for the Red Hat Enterprise Linux 5 MRG Messaging 2.0 Release. Refer to Section 2.6, “ RHEA-2011:0890 – Red Hat Enterprise MRG – Messaging 2.0 Release ”.

Chapter 2. MRG Messaging on Red Hat Enterprise Linux 5

2.1. RHSA-2013:0561 - Moderate: Red Hat Enterprise MRG Messaging 2.3 security, bug fix, and enhancement update

Red Hat Enterprise MRG is a next-generation IT infrastructure incorporating Messaging, Realtime, and Grid functionality. It offers increased performance, reliability, interoperability, and faster computing for enterprise customers.

MRG Messaging is a high-speed reliable messaging distribution for Linux based on AMQP (Advanced Message Queuing Protocol), an open protocol standard for enterprise messaging that is designed to make mission critical messaging widely available as a standard service, and to make enterprise messaging interoperable across platforms, programming languages, and vendors. MRG Messaging includes an AMQP 0-10 messaging broker; AMQP 0-10 client libraries for C++, Java JMS, and Python; as well as persistence libraries and management tools.

Descriptions of the security fixes provided in this advisory can be viewed at https://rhn.redhat.com/errata/RHSA-2013-0561.html. The changes in this advisory for other non-security bug fixes and enhancements are documented below:

python-qpid

BZ#856299

Some message send operations paused for longer than the specificed timeout value. The timeout value was not passed to the sync() call in all situations. In this release the specified message timeout value is honored. All message send operations now return within the timeout value specified.

BZ#850517

A previous code modification did not account for older versions of Python's SSL implementation, causing various qpid tools using SSL to fail on Red Hat Enterprise Linux 5. In this release qpid tools using SSL on Red Hat Enterprise Linux 5 function correctly.

BZ#782806

SSL/TLS support is added to the messaging API. This allows clients using the messaging API to connect to the broker over a connection encrypted using SSL/TLS.

qpid-cpp

BZ#773719

The list of bindings on a queue was insufficiently protected from concurrent access. If multiple clients connected to the broker were adding and deleting bindings on the same queue, the broker might crash. In this release the broker provides adequate protection for this data structure. The broker does not crash with multiple clients manipulating the bindings on the same queue.

BZ#877081

Improper message header record locking could cause a broker crash. When messages were shared among many queues and subsequently rerouted, header processing by one thread

could invalidate header processing by another thread. In this release message header records are effectively locked to prevent concurrent changes. Message header records may now be processed by any number of queues without causing a broker crash.

BZ#734883

Under certain conditions, the store could think the journal was full when it was not. In that case the store could not be recovered and the broker would not start. In this release the logic error is fixed. The store no longer thinks it is full when it is not, and the broker recovers normally.

BZ#737685

A defect in the qpid broker caused messages that were delivered and acquired by a consumer but not yet acknowledged to not be delivered to the alternate exchange when the queue was deleted. Unacknowledged messages could be lost if they were expected to be delivered to the alternate exchange. In this release all unacknowledged messages in a queue are now delivered to the alternate exchange regardless of whether or not they have been acquired by a consumer.

BZ#876664

If a queue creation failed because of a bad property value then artifacts of the failed creation were not cleaned up. If the queue was later created correctly then an error appeared 'two management objects with the same identifier'. In this release when a queue creation fails because of bad property values all artifacts related to the failed queue are deleted. There is no possibility for management objects to have identifier conflicts and there is no error message.

BZ#871774

The algorithm for browsing messages changed, but the message group code was not updated correctly. A message acquired by a client would still remain visible to browsing clients. In this release the message group code is updated to the new algorithm. Acquired group messages are no longer visible to other browsing clients.

BZ#868403

Federated link Id numbers are assigned sequentially from a list of 64k candidates. After 64k link numbers have been assigned then the number sequence starts again at zero and link Id number collisions can occur. Federated links received errors when trying to create sessions using duplicated Id numbers. In this release a list of in-use link Id numbers is maintained and used to avoid duplicate conflicts. Federated links may be created and destroyed indefinitely without receiving errors.

BZ#801605

Federated links could issue cluster events prior to the link connection being fully established (protocol handshake complete). Cluster members received events for unknown federated link connection, which resulted in the members leaving the cluster. In this release federated link IO processing is delayed until after the connection is fully established. Cluster members do not leave the cluster when federated to a non-responsive peer.

BZ#720714

The mechanism for generating Link and Bridge names was incorrect and could generate the same name for multiple distinct Links and Bridges. The broker's internal federation configuration

would become corrupt. This resulted in inconsistencies among clustered brokers, forcing one or more brokers to exit. In this release a new Bridge and Link naming algorithm guarantees unique names for all Bridge and Link objects. Clustered brokers running federation now remain consistent.

BZ#710787

The loss of connection always resulted in at least one attempt to reconnect in the event that a client lost its connection to the broker, even when the reconnect option was disabled. In this release the logic handling the loss of a connection is altered to distinguish it from opening a connection in the first place. The reconnect option is now correctly handled; specifically if set to false there will never be a reconnect attempt made.

BZ#703170

Multiple links can now be configured between federated brokers, allowing for greater message throughput.

BZ#849790

Acl files may be reloaded at any time. However if there is an error in the Acl file then the broker halts. There was no way to load a trial Acl file and see how it behaves; only live Acl rule files could be tested. This release introduces the ability to test an ACL file on an off-line broker. The Acl file may be repeatedly loaded and tested without interrupting service on a mission-critical broker. When the Acl file is finally tested it can be loaded into the live broker. See the Messaging Programming Reference Guide for details.

BZ#849788

Specifying Acl rules that allow named users to create named objects required a large number of specific Acl rules. Administrators had to keep adding users to the Acl file to allow new users to use the broker. In this release user name substitution keywords are added to the Acl file so that a single rule may apply to all users. Keywords are created to substitute for the user name, the domain name, or the user and the domain name together. Keyword substitution is allowed for object names, routing key names, alternate exchange names, and queue names. Refer to the Messaging Installation and Configuration Guide for details.

BZ#784685

Log messages did not show enough detail to audit who created and deleted broker objects. Log messages had no high level filter keys users could specify to help filtering log messages. This release contains a new log category to track creation and deletion of manageable objects in enough detail to track who change the configuration of the broker. Refer to the Messaging Installation and Configuration Guide for details.

BZ#836141

An error in the clustering logic caused cluster nodes to fall out of sync with the master node during federation. In this release the logic is fixed, and cluster nodes remain synchronised. Further details follow: The eldest cluster broker sends commands Creates its federation link and bridge, But leaves behind its juniors, all confused. Not privy to these mysteries, they soon Fall out of sync, and drag the cluster down. Without agreement, how can clusters run? If elder and its juniors disagree, Divide against itself this fragile house, How can it stand? How can the perfect synchrony exist? It can't. And therein lies the rub. The elder broker builds its links and bridge, But tells the youngsters: "Do thou as I do!" They too can build that bridge, maintaining it

Against the day when elder fails. All their Commands are thus maintained in sync. The cluster hums, the traffic leaps and flows. Where once was strife and error, harmony has come. The synchrony will no doubt be disturbed again. But not by this BZ.

BZ#835119

The SSL transport implementation for both C++ broker and C++ client was missing some of the implementation. This meant that even if the broker or client detected heartbeat failure it could not abort the failed connection. So heartbeats did not work over SSL. In this release the missing pieces of the SSL implementation are present for SSL for the broker and client. Heartbeat failure now correctly aborts SSL connections on both broker and client.

BZ#572567

Users were not restricted in the number of queues they may declare. Any user could run the broker out of resources by creating too many queues. This release adds a command line argument "--max-queues-per-user N" that is enforced for all users. When this option is specified each user may have at most N queues concurrently, and broker resources are preserved.

BZ#861838

Dynamic bridges were improperly destroyed after a binding error. This often occured during a broker restart when resources were being recreated sequentially, and bindings could not succeed until all the resources had been recreated. This caused configured bridges to be lost instead of being recovered during a maintenance cycle.In this release dynamic bridges are not deleted after a binding error. After required resources have been restored the dynamic bridge is created properly during a periodic retry.

BZ#799479

Non-privileged users could specify any queue size and stress the broker's resources. This release provides quota limits for message queues, in the form of upper and lower limits for memory usage and for queue message counts. User requests to create queues that are too large or too small, both in-memory and on-disk, are denied.

BZ#754990

The ring policy enforcement logic did not take account of the case where no maximum size was specified, only a maximum count. A maximum size of 0 (i.e. unspecified) would result in a message being needlessly removed. In this release the logic is updated to correctly handle the case where the size is not limited but the count is. The policy is correctly enforced (and the behaviour is the same regardless of the default-queue-limit set).

BZ#866677

A previous release changed the logging of expired message. Messages expired by the queue cleaner are not logged as they were prior to the change. In this release logging is moved to a common place in the source module so all expired messages receive the same logging treatment. The queue name and message properties are included in debug-level log entries.

BZ#802656

This release adds topic key matching to the Acl file routing keys that uses the same logic as the broker's routing key matching. Acl rules can now use the same routing key specification

syntax that is used by the broker's run-time topic key matching logic.

BZ#804752

Client exceptions were not properly cleared when a transport exception is triggered by a broker failure. Client code got stuck in an infinite loop reprocessing old exceptions. In this release when the broker exits unexpectedly, the transport channel is closed and the transport exception is cleared. The client recovers and creates a new connection when the broker returns to service.

BZ#884036

Attempts to cast negative-zero to an unsigned type would fail due to a bug in older GCC libraries. In this release the casting logic is rearranged to handle the "minus" sign separately. Negative-zero now works properly on older and newer GCC libs. Other values are unaffected.

BZ#870058

Running the **`qpidd --config`** command passing an existing directory would hang and not return to the command line. In this release **`qpidd --config`** detects when an existing directory is passed and, if so, returns an error message.

BZ#812376

Bounds checking on fields whose AMQP types restrict their length was not done at a point that the application can be notified. The IO thread caught the problem but at a late stage. Over large values caused the IO thread to spin indefinitely on the error. In this release bounds checking is done when the (potentially) invalid value is set. An error is raised to the application if it sets too large a value.

BZ#813742

The async queue replication code set frame flags incorrectly when there was an empty content frame (i.e. a content frame with content size equal to zero). This was treated as not having a content frame at all so the end-of-frameset flag was incorrectly set on the header frame. Such messages could not be replicated; the receiving broker would reject them as invalid AMQP. In this release the end-of-frameset flag setting takes account of whether there is a content frame, even if it is empty. Such messages are sent as valid AMQP frames and replicate as expected.

BZ#814356

The broker required create permission for user sessions performing a passive declare (i.e. a declare that should not create, but should merely verify existence). This resulted in clients that were not actually trying to create an exchange receiving permission errors due to the lack of create permission. In this release if the declare is passive, the permission required is changed to access rather than create. Clients using but never creating exchanges need not have create permission granted.

BZ#816092

In queue state replication, if duplicate messages were detected then the replication exchange did not increase its counters (**`msgDrop`** and **`byteDrop`**). This caused invalid counters of replication exchange. In this release when message duplicity is detected, relevant counters of the replication exchange are incremented, resulting in correct counters of replication exchange.

BZ#740505

When command-line options contained errors, the **--help** and **--version** options were not processed. With an error in user specified command-line options the presence of the **--help** option caused a message "Use --help to see valid options" but did not display the help screen. In this release the **--help** and **--version** switches are parsed separately so they act despite other errors in the command-line options. When a parse error happens and the user has specified --help then the help usage text is displayed.

BZ#873414

An updated default ACL file is provided at **/etc/qpid/qpidd.acl**. New installations use this file by default. Existing installations that upgrade via rpm can find the file as **/etc/qpid/qpidd.acl.rpmnew**.

BZ#834256

The alternate-exchange option in an x-declare clause within the link-options of an address was being ignored. The alternate exchange property could not be controlled via the link options. In this release the alternate exchange option is checked and used if specified. The alternate-exchange can now be controlled on queues created through the link options in an address.

BZ#678612

Some qpidd log messages were output before the PID when running **qpidd --check**. With **log-to-stdout=yes**, **qpidd --quit** did not work. In this release the unnecessary log messages are removed, and **qpidd --quit** works as expected.

BZ#868881

JournalInactive messages appeared when certain broker scheduling algorithms were triggered and the broker is running at INFO log level. There was nothing a customer could do to stop the scheduling triggers and there was no convenient way to turn the **JournalInactive** messages off. In this release the priority of the JournalInactive log messages is lowered to DEBUG level and the message frequency is lowered from 12/second to 1/second. If the broker is running at INFO log level the **JournalInactive** messages may be selectively re-enabled by using switch: **--log-enable debug+:Timer**.

BZ#783215

When a reroute operation was requested on a queue and the flag was specified to use the alternate exchange of that queue, there was no check whether an alternate exchange has actually been specified for the queue. In such a case the rerouted messages were silently dropped. In this release a check is made to ensure that when requested to use a queue's alternate exchange, that exchange has been specified. If a reroute request is made and the alternate exchange is to be used as the target and the queue did not have an alternate exchange, an error is raised.

BZ#783243

The mechanism for generating Link and Bridge names was incorrect and could generate the same name for multiple distinct Links and Bridges. In that case the broker's internal federation configuration would become corrupt and federation would fail to properly forward messages between brokers. This release fixes the Bridge and Link naming algorithm to guarantee unique

names for all Bridge and Link objects. Now, federation forwards messages as expected.

BZ#846465

In a previous release GSSAPI authentication was removed from the default list of mechanisms. In this release GSSAPI is added to the default list of sasl mechs in **/etc/sasl2/qpidd.conf**. The Installation and Configuration Guide has instructions on configuring GSSAPI authentication with Kerberos.

BZ#849557

The first broker was stopping in the INIT state without having put its URL into the ClusterMap membership list. Since the first broker did not put its URL in the list of cluster members, the second broker cannot register its own URL in the Credentials Exchange of the first. So when the first broker receives its update request, it cannot be authenticated. So the first broker raises an error, and both are stalled. In this release, before the first broker starts waiting for the others to get a quorum, make it call **Cluster::ready()**, which will put its URL on the cluster membership list. It is ready - it is only waiting for the others to come online and get quorum. The second broker can now authenticate to the first, and get its update. The two brokers do not deadlock.

BZ#783482

Queues can now be declared as browse-only.

BZ#783428

Queues could be created with **max_size** and **max_count** values being unreasonably small. ACL parameters to prevent these values from being so small were absent. Users were able to declare queues that will never work since the maximum size and count values are too small. In this release ACL create queue parameters are added to set lower limits on size and message count. These settings prevent the queue **max_size** and **max_count** values from being too small.

BZ#784957

Acl processing logic discarded certain rules in a premature optimization. Some sequences of rules would not work. In this release Acl rule processing keeps all Acl rules and process them in the order given. Allow and Deny rules may be intermixed and give the expected allow or deny result.

BZ#790004

Interfaces with IPv6 addresses were not included in the list of interfaces included when automatically calculating a broker's default urls. Interfaces with IPv6 addresses are now included in the automatically generated list and IPv6 addresses are included in the default broker url.

qpid-java

BZ#785209

The store serializes transactions. Java client is not well optimized for speed and performance. A Java client running against a broker which uses durable and transactional messages was

very slow. In this release the Java client is optimized to improve speed. The store is not changed.

BZ#805881

The JMS client can now send and receive messages encoded as AMQP lists. On the receiver side, List messages are exposed via 3 interfaces. 1. javax.jms.StreamMessage 2. javax.jms.MapMessage 3. org.apache.qpid.jms.ListMessage On the sender side, List messages can be sent in two ways. 1. org.apache.qpid.jms.ListMessage -- by creating it via createListMessage() in org.apache.qpid.jms.Sesson. Ex ListMessage msg = ((org.apache.qpid.jms.Sesson)ssn).createListMessage(); 2. If you set -Dqpid.use_legacy_stream_message=false any stream message you create will be encoded as a list message. Ex StreamMessage msg = jmsSession.createStreamMessage(); From next release onwards we will switch the default, so you don't need to explictly set -Dqpid.use_legacy_stream_message=false We recommend the second option as sticking to the standard JMS interfaces will allow your code to be more portable. This allows the JMS client to send and receive messages encoded as AMQP 0-10 lists. This further enables the use of QMF while using standard JMS interfaces.

BZ#860011

`x-bindings` were not evaluated as defined the address spec. For producers **`x-bindings`** were not created at all. Link bindings were not created/deleted for both producers and consumers as per the address spec. In this release **`x-bindings`** are now evaluated as defined by the addressing spec. **`x-bindings`** can now be created for both producers and consumers for both node and links.

BZ#882243

Messages sent under an XA transaction were replayed on failover. Consequently, transaction atomicity was lost. In this release such messages are no longer replicated, and transaction atomicity guarantees are honoured.

BZ#876193

Incorrect error handling caused applications to not be notified that an exchange of a different type with the same name exists when creating an exchange. In this release an error is thrown when creating an exchange if one already exists with the same name.

BZ#893980

When a producer tried to delete a non existent subscription queue a session error occurred. This session error does not get picked up by the client while it's waiting for completion. The client appears to hang. Eventually the wait times out. In this release a check is added so that a producer does not attempt to delete a non-existent subscription queue.

BZ#824917

A logic error meant that the durable property in addresses was ignored and exchanges created using an address were not marked durable. In this release the durable flag is set based on the durable property specified in the address. The exchange created is marked as durable if the durable property is set to true in the address.

qpid-jca

BZ#888392

The QpidConnectionFactoryProxy only implemented the ConnectionFactory JMS 1.1 API. In this release the proxy implements the Queue/TopicConnectionFactory portions of the API (JMS 1.0).

qpid-qmf

BZ#877553

A rarely used internal data structure was being modified by a thread without being locked to other threads. This would result in the data structure getting corrupted and eventually causing the application to crash. In this release a lock was introduced to allow the data structure to be modified safely. The internal data structure is no longer corrupted.

BZ#845223

In this release the QMF broker events **`clientConnect`**, **`clientDisconnect`**, and **`clientConnectFail`** where extended to include the Client Properties Map. This map is defined by AMQP-0.10 as a map of identifying attributes as provided by the client. QPID clients include their process id, parent process id, and command name in this map. This information allows detailed tracking for auditing and debugging purposes.

BZ#781496

Messages were removed from the Queue when acquired by a client. When querying for the message timestamp, acquired messages were no longer available and the timestamp could not be retrieved. A timestamp of zero was returned. In this release acquired messages remain on the Queue, and are available for querying the timestamp. A valid timestamp is now returned for messages that have been acquired.

qpid-tests

BZ#800912

The qpid-perftest program had a logic error which caused it to wait for the incorrect number of control messages. Running **`qpid-perftest`** with **`--npubs`** < **`--nsubs`** and **`--iterations`** > 0 fails to finish, and appears to hang. In this release the **`qpid-perftest`** program now handles **`--npubs`** < **`--nsubs`** and **`--iterations`** > 0 without appearing to hang, and finishes normally.

qpid-tools

BZ#864933

The command-line tools **`qpid-config`** and **`qpid-stat`** have changes to their command-line options. **`-r`** now lists bindings recursively, and **`-b`** specifies the broker address. For backward compatibility **`-a`** can still be used for the broker address. Refer to the Installation and Configuration Guide for details.

BZ#795324

Client connection data was stored in wrong place in the code of **`qpid-cluster`** tool. As a

result the tool did not display client connection information. In this release the tool displays client connection information.

BZ#740485

Running **qpid-stat** with a remote host address to view info on remote queues displayed the info for the localhost queue. In this release **qpid-stat** connects to the remote host address when a remote host address is specified, and the info for the remote queues is shown.

BZ#770711

qpid-cluster ignored broker security credentials. Consequently the tool was unable to connect to any broker that had authorization turned on. In this release the QMF code pays attention to the credentials, if present, and **qpid-cluster** can connect to brokers with auth turned on.

BZ#895535

Several QPID tools support the '**--ssl-key**' parameter. This parameter is used to specify a file which contains the private key used to sign the certificate that identifies the tool's user with the broker. A coding problem caused the QPID tool to ignore the value supplied to the '**--ssl-key**' parameter. The broker against which the tool was run would not be able to authenticate the user of the tool. This would cause the broker to reject the connection attempt, and the tool's operation would fail. In this release the code is corrected to use the value passed via the '**--ssl-key**' parameter when providing authentication criteria to the broker. Consequently, the tool's user is authenticated by the broker, and the QPID tool's operation is permitted.

BZ#805599

The cluster column was missing in **qpid-stat** output. This release fixes the output to include the broker's cluster. **qpid-stat -g -b <broker>** now shows the cluster name or **<standalone>**.

BZ#786555

Using **qpid-config** to create a queue or exchange with the same name but different options as an existing queue or exchange would succeed, but the options were not changed for the existing queue or exchange. In this release **qpid-config** uses a new library that returns an error when an attempt is made to create an existing queue or exchange. Now an error is returned when attempting to create a queue or exchange name that already exists.

All users of the messaging capabilities of Red Hat Enterprise MRG 2.3 are advised to upgrade to these updated packages, which resolve these issues and add these enhancements.

2.2. RHSA-2012:1277 – Moderate: Red Hat Enterprise MRG Messaging 2.2 security, bug fix, and enhancement update

Red Hat Enterprise MRG is a next-generation IT infrastructure incorporating Messaging, Realtime, and Grid functionality. It offers increased performance, reliability, interoperability, and faster computing for enterprise customers.

MRG Messaging is a high-speed reliable messaging distribution for Linux based on AMQP (Advanced

Message Queuing Protocol), an open protocol standard for enterprise messaging that is designed to make mission critical messaging widely available as a standard service, and to make enterprise messaging interoperable across platforms, programming languages, and vendors. MRG Messaging includes an AMQP 0-10 messaging broker; AMQP 0-10 client libraries for C++, Java JMS, and Python; as well as persistence libraries and management tools.

Security Fixes

CVE-2012-2145

It was discovered that the Apache Qpid daemon (**qpidd**) did not allow the number of connections from clients to be restricted. A malicious client could use this flaw to open an excessive amount of connections, preventing other legitimate clients from establishing a connection to qpidd.

Note

To address CVE-2012-2145, new **qpidd** configuration options were introduced: **max-negotiate-time** defines the time during which initial protocol negotiation must succeed, **connection-limit-per-user** and **connection-limit-per-ip** can be used to limit the number of connections per user and client host IP. Refer to the qpidd manual page for additional details.

CVE-2012-3467

It was discovered that **qpidd** did not require authentication for "catch-up" shadow connections created when a new broker joins a cluster. A malicious client could use this flaw to bypass client authentication.

Bug Fixes

BZ#841196

Previously, when the **setuid()** system function was called after a connection object had been created, and the **select()** function of the **python-qpid** messaging client was interrupted at the same time, an unhandled exception was raised. Consequently, **python-qpid** terminated unexpectedly. With this update, **python-qpid** traps the exception and continues its operation without disruption in the described scenario, thus fixing this bug.

BZ#693444

Previously, the Python and C++ versions of the messaging API were behaving inconsistently with respect to the default reliability of receivers. This inconsistency led to confusion for users involved with both APIs as well as making documentation more complicated. With this update, the Python client has been modified to have the same default reliability for topics, that is for cases where a receiver is established from an exchange. Now, when establishing a receiver from an exchange (topic pattern), the link will be unreliable by default, which is the consistent behavior. If required, the reliability option can be changed within the link options of the address.

BZ#817283

Prior to this update, when the **transport** connection option was used to enable SSL, despite of what the option indicated, SSL was in fact not enabled. This update ensures that **transport**

is properly evaluated and SSL can be enabled using this option as expected.

BZ#834608

Previously, connections made from an existing broker to provide an update to a newly joining broker were not being authenticated. Consequently, brokers were unable to join a cluster using certain authentication types, posing a possible security issue. This bug has been fixed and all connections made to a broker are now fully authenticated and secured.

BZ#729311

Clustered brokers use a special array (also called *local map*) to store pointers to connections. If the broker had the authentication feature turned on and a connection failed to authenticate, the pointer created for it in the local map was never deleted. Consequently, these stale pointers could clog the connection counter up to its maximum, at which point the broker started to refuse new connections, whether properly authenticated or not. This situation posed a risk for a denial of service attack. With this update, pointers to connections that failed to authenticate are properly delted from the local map, thus preventing this bug.

BZ#801465

When both elder brokers of two federated clusters failed, the brokers that remained in the cluster and attempted recovery were not aware that the peer elder broker had failed. Consequently, the federated link could not be re-established as the recovering brokers attempted to reconnect to their peer cluster's elder broker, which no longer existed. With this update, when any broker of a federated cluster fails, the current set of active brokers in that cluster is communicated to all federation peers. Now, the federated link is properly re-established to an available broker in the peer cluster when a failover occurs.

BZ#808090

When a broker in a cluster created the **`x-qpid.cluster-update`** exchange to receive updates from an elder broker, the exchange continued to exist after the update was complete but no reporting tool listed it as active on the broker. Consequently, it was possible to bind to this stale exchange but sending a message through it could not reach the queue. With this update, these exchanges are deleted after an update operation is finished and no hidden exchanges now remain on cluster brokers in the described scenario.

BZ#809357

Previously, code in the **`setTcpNoDelay()`** socket function specific to the Windows operating system tried to set a property before the connection was open. Consequently, an exception was raised and the **qpid-perftest.exe** and **qpid-latency-test.exe** applications terminated. With this update, the request for a property is delayed until the connection socket is open, thus fixing this bug and conforming the Posix implementation.

BZ#840982

When using the **`EXTERNAL`**, **`CRAM_MD5`**, or **`DIGEST-MD5`** authentication mechanism, connections on cluster shadowed connections were created by the **`anonymous`** user but were later deleted by the user name negotiated by the authentication mechanism. Consequently, per-user connection counts were reported incorrectly; counts were added to one user and subtracted from another user for the same connection. With this update, after the user name has been negotiated, the connection count is always applied to that user rather than the

anonymous user and connection counts and quotas now work as expected.

BZ#689408

Previously, during an update, ACL rules were preventing access to the **cluster-update** exchange.Consequently, new brokers could not join the cluster. With this update, catch-up connections, which are authenticated as **cluster-user** are allowed to update state and brokers can now join the cluster as expected.

BZ#826989

When two clusters are connected by a replication queue, brokers in both of these clusters are killed, and replacement brokers are added, one of the clusters gets out of sync over time. This occurred approximately once in 20 attempts when a process killed a broker and a new broker was added. This bug happened because the **delivery count** queue state was not properly replicated from a senior broker to the new one. Consequently, after a period of time, the new broker encountered an error not encountered by the senior broker and the new broker terminated. With this update, the **delivery count** queue state is properly replicated in the described scenario, thus greatly reducing the risk of two brokers getting out of sync.

BZ#844618

When a new broker joined a cluster, the cluster failed to enable the broker's capability to generate queue events upon receiving messages to its queues. Consequently, the broker failed to replicate incoming messages while its senior broker replicated them properly. As a result, the two brokers got out of sync and the new broker eventually terminated. This update ensures that generating queue events is enabled in a new broker during the **CATCHUP** phase of joining the cluster and the desynchronization bug no longer occurs in the described scenario.

BZ#811481

When one or more brokers in the source cluster of a federation fail, the destination broker recovers by failing-over to an alternate broker in the source cluster. However, when the original brokers recovered, the destination broker was not aware that the brokers had returned. Consequently, when another broker in the source cluster failed, the destination broker could not re-establish the federation link. With this update, federated destination brokers monitor the source cluster to keep track of when failed source brokers recover and re-join the source cluster. Now, the destination broker can correctly recover the federation link to a recovered source broker on failover.

BZ#750775

When a session exception was received, the connection was marked as closed, but the underlying TCP or AMQP connection was not in fact closed. Consequently, connection leaks kept occurring until the broker was out of file handles it could use. With this update, the session is closed instead of the connection. The rest of the session and the connection are still usable and the error is still reported by the Connection Listener but session errors no longer cause connection leaks.

BZ#781560

Pending read requests were monitored for completion even though a forced close was in progress. Consequently, on the Windows operating system, closing an AMQP connection over a broken TCP link, such as after a heartbeat timeout, could make the Qpid client to become

unresponsive. With this update, pending read requests are cancelled on close in the Windows Asynch I/O system and the client properly starts a reconnect sequence in the described scenario.

BZ#707682

Due to a bug in the Python management libraries, the **qpid-tool** utility was unable to address the ACL object in the broker. With this update, the **reloadACLFile()** function is called by **qpid-tool** to load ACL policy changes into a running broker and the ACL object can now be accessed as expected.

BZ#841488

Previously, CLI utilities, such as **qpid-stat**, were unable to display multi-byte characters in strings. Consequently, attempts to invoke a CLI utility on a broker, which contained queues or exchanges with multi-byte characters in their names, failed. In such a scenario, an exception was raised and the display of the list of queues or exchanges was stopped. This update fixes the display library used by the CLI utilities to correctly handle the multi-byte characters and **qpid-stat** and its peer utilities now properly display strings that contain multi-byte characters.

Enhancements

BZ#831365

Heartbeats are used to quickly detect loss of connectivity between a client and a broker. In broker-to-broker links (federation), there was no option to use heartbeats. Consequently, loss of connectivity in federation links could go undetected for a very long time, resulting in messages not being delivered. This update adds a new **--link-heartbeat-interval** command-line option, which allow heartbeats to be used at a configurable interval (in seconds) on federation links. Loss of connectivity is now quickly detected in the described scenario.

BZ#609685

This update introduces the new **--max-negotiate-time** qpidd broker option. This option prevents a possible denial of service security concern. It specifies the maximum amount of time a new connection to the broker has in order to do protocol negotiation and authenticate itself. If this procedure does not complete in the specified time, the connection is aborted by the broker.

BZ#683711

Previously, it was not possible to limit the number of simultaneous connections a user could make to a broker. This update introduces new command-line parameters to specify connection limits. A new code to monitor connections enforces the limits. Now, individual users cannot consume all the broker's connection resources and deny service to other users.

BZ#831363

The Qpid Java client has been upgraded to upstream version 0.18, which provides a number of performance enhancements over the previous version.

BZ#819590

This update enhances the **XAResourceImpl** class maintaining a list of sibling resources for the **XAResource** interface in order to satisfy the **DtxStart/DtxEnd** pairings for the AMQP

0.10 protocol. Now, multiple XAResources are supported.

BZ#834416

This update adds support for failover, XA/HA data sources, and the XARecovery interface to the Qpid JCA adapter.

All users of the messaging capabilities of Red Hat Enterprise MRG 2.2 are advised to upgrade to these updated packages, which resolve this issue, fix these bugs, and add these enhancements. After installing the updated packages, stop the cluster by either running **`service qpidd stop`** on all nodes, or run **`qpid-cluster --all-stop`** on any one of the cluster nodes. Once stopped, restart the cluster with **`service qpidd start`** on all nodes for the update to take effect.

2.3. RHSA-2012:0529 – Moderate: Red Hat Enterprise MRG Messaging 2.1 security, bug fix and enhancement update

The changes in this advisory are the same as those for the Red Hat Enterprise Linux 6 MRG Messaging 2.1 Release. Refer to Section 1.3, " RHSA-2012:0528 – Moderate: Red Hat Enterprise MRG Messaging 2.1 security, bug fix and enhancement update ".

2.4. RHBA-2011:1393 – Moderate: Red Hat Enterprise MRG Messaging 2.0 bug fix update

The changes in this advisory are the same as those for the Red Hat Enterprise Linux 6 MRG Messaging 2.0 Release. Refer to Section 1.4, " RHSA-2011:1399 – Moderate: Red Hat Enterprise MRG Messaging 2.0 bug fix update ".

2.5. RHBA-2011-1339 – Red Hat Enterprise MRG Messaging 2.0 bug fix update

Updated Red Hat Enterprise MRG Messaging packages that fix several bugs are now available for Red Hat Enterprise MRG 2.0 for Red Hat Enterprise Linux 5.

Red Hat Enterprise MRG (Messaging, Realtime and Grid) is a real-time IT infrastructure for enterprise computing. MRG Messaging implements the Advanced Message Queuing Protocol (AMQP) standard, adding persistence options, kernel optimizations, and operating system services.

Bug Fixes

BZ#704547

According to the JMS specification, concurrently executing clients cannot use the same client ID. Previously, it was possible to allow multiple connections to use the same client ID. With this update, it is no longer possible to define such clients. Note that you need to enable the verify client ID feature to prevent the clients from having identical IDs (-Dqpid.verify_client_id=true).

BZ#704566

The **`getJMSXPropertyNames()`** method returned a usable enumeration only when called for the first time. If the user called the **`getJMSXPropertyNames()`** method multiple times, the method returned empty enumerations. This happened because the system overwrote the first created enumeration with the subsequent call of **`getJMSXProperyNames()`**. With this update,

the underlying code has been changed and **`getJMSXPropertyNames()`** returns the enumeration with the property names as expected.

BZ#712011

The system threw a SessionException if a session was committed after a transaction had exceeded the queue limit (max-queue-size). This caused the client to fail to handle the error. The system now throws a JMSException under these circumstances and client can recover with the JMSException as expected.

BZ#723750

In a clustered environment, the system threw a SessionException when a session was committed after a failover using the **`session.commit()`** method. Because the client application code did not receive a standard JMSException, it could fail to recover. The system now throws a JMSException under these circumstances and the client recovers as expected.

BZ#726050

When a connection was created with newly specified credentials, the Java client cached the new credentials and overwrote the default credentials provided by the connection URL. As a result, when a new connection was created without credentials, the connection used the cached credentials and the authentication could fail. With this update, the credentials specified at a connection creation are not cached and the default credentials for connections without credentials are used as expected.

BZ#726712

A queue bound to the default exchange dropped messages with a subject. This happened because the routing key was wrongly using the message subject as the queue name. The underlying code has been changed and such messages are now delivered to the queue.

BZ#728484

Previously, the system threw a javax.naming.NameNotFoundException if the JNDI file definition in the executed program defined a JNDI file with a syntax error. However, this is not appropriate as the program cannot access the JNDI file and the execution was therefore interrupted. Such internal exceptions are now re-thrown as ConfigurationExceptions with the message "Failed to parse JNDI properties file" and the program is interrupted.

BZ#732534

The broker ignored QMFv2 requests sent from a Java client because the requests did not have the ***`app-id`*** parameter set. The requests are now sent with ***`app-id`*** and the messages are processed as expected.

Users of the messaging capabilities of Red Hat Enterprise MRG 2.0, which is layered on Red Hat Enterprise Linux 5, are advised to upgrade to these updated packages, which fix these bugs.

2.6. RHEA-2011:0890 – Red Hat Enterprise MRG – Messaging 2.0 Release

Red Hat Enterprise MRG is a next-generation IT infrastructure incorporating Messaging, Realtime, and

Grid functionality. It offers increased performance, reliability, interoperability, and faster computing for enterprise customers.

MRG Messaging is a high-speed reliable messaging distribution for Linux based on AMQP (Advanced Message Queuing Protocol), an open protocol standard for enterprise messaging that is designed to make mission critical messaging widely available as a standard service, and to make enterprise messaging interoperable across platforms, programming languages, and vendors. MRG Messaging includes an AMQP 0-10 messaging broker; AMQP 0-10 client libraries for C++, Java JMS, and Python; as well as persistence libraries and management tools.

Bug Fixes

BZ#453538

Prior to this update, the **MRG Messaging Broker** did not consider the priority of signaled messages when delivering messages. Due to this behavior, applications could not rely on the broker to deliver higher priority messages before lower priority messages. With this update, the **MRG Messaging Broker** can be configured to recognize higher priority messages and adjust delivery accordingly.

BZ#484218

Prior to this update, the *Last Value Queue* (LVQ) used a specific message header by default as the key determining equivalence (**`qpid.LVQ_key`**). Due to this behavior, producers had to be explicitly coded to set that header. This update adds a new configuration option **`qpid.last_value_queue_key`** that allows the choice of LVQ key to be set per-queue. Now, users can choose an application defined header.

BZ#484691

Prior to this update, the *remote direct memory access* (RDMA) protocol transport for Apache Qpid only supported InfiniBand network interfaces. Due to a problem affecting message flow control when using iWarp (*Internet Wide Area RDMA Protocol*) interfaces, the client process was unable to transmit more than 30–40 messages on a single connection. Now, Qpid's use of RDMA supports iWarp network interfaces. Current users of RDMA must upgrade any brokers before upgrading their clients if the upgrade is staged. This upgrade order is necessary as the new brokers can detect both the old and new protocols and switch automatically, but the new clients will only use the new protocol.

BZ#500430

Prior to this update, the **MRG Messaging Broker** could not use SASL (*Simple Authentication and Security Layer*) and/or SSL (*Secure Sockets Layer*) with SASL to provide authentication and security for inter-broker federation links as it could with links between clients and the broker. This update changed the **`qpid-route`** tool and the broker functionality to allow one federation broker to act as an SASL server while the other acts as an SASL client. Now, federated links can be connected with SASL, with the external mechanism of SSL.

BZ#581560

Prior to this update, when using dynamic routes in inter-broker federation, spurious bindings could be left in place when a user binding was deleted. The result of this was that messages might be unnecessarily forwarded from one broker to another. With this update, the spurious bindings are properly removed when a client process removes its bindings.

BZ#611820

Prior to this update, the counters on the broker were not updated for outgoing frames and messages. Due to this behavior, statistics were not accurately reported for outgoing frames and messages, always displaying zero. With this update, the statistics are modified for outgoing frames and messages. Now, reported statistics are accurate.

BZ#614944

Under certain circumstances, the **qpidd** service terminated unexpectedly on startup. With this update, the broker is modified so that no more failures occur.

BZ#615300

Prior to this update, the failover exchange was not listed by management tools. This update fixes this problem and lists the failover exchange by management tools.

BZ#629926

Previous versions of the MRG Messaging Windows software development kit (WinSDK) did not include an example on how to authenticate with a remote broker. An example has now been added.

BZ#632348

Qpid's variant types are internally represented as integer values. Previously, when a user wanted a client application to print messages about variant types in a readable form, they had to write a custom function to convert an integer value to an appropriate string. To address this issue, this update exposes the **qpid::types::getTypeName()** function in the public API, and users are no longer required to write custom conversion functions.

BZ#632625

When a user created a queue with a policy set to **ring**, previous versions of the C++ broker failed to enforce the byte size limit. This update applies an upstream patch that resolves this issue, and the byte size limit for ring queues is now implemented as expected.

BZ#641822

On Windows machines, the **qpid-perftest** utility sometimes measures test intervals as zero seconds. Previously, this could cause the utility to fail to generate statistics instead giving the error message:

```
Controller exception: Bad report: 1.#INF
```

To avoid possible confusion, this update adapts **qpid-perftest** to detect zero second test intervals, and users are now presented with a comprehensive error message in these cases.

BZ#646913

When the hostname in a broker URL contained an underscore (_), the previous version of the Java Message Service (JMS) client incorrectly evaluated it as **localhost**, and failed to establish a connection, giving the following error message:

```
java.net.ConnectException: Connection refused
```

With this update, an upstream patch has been applied to ensure the JMS client evaluates hostnames correctly, and the presence of an underscore in such a hostname no longer prevents it from connecting to a host.

BZ#647858

Previously, the .NET Binding for the C++ Qpid Messaging Client did not provide a function to acknowledge individual messages. This error no longer occurs, and the **`Session::Acknowledge()`** function is now included in the binding layer as expected.

BZ#649003

The **`getPropertyNames()`** method from the **`javax.jms.Message`** interface returns an enumeration of property names in a message. Previously, this included properties with values that are not of a supported type as defined in the JMS specification. Consequently, an attempt to access such a property caused the **`MessageFormatException`** to be raised. With this update, the **`getPropertyNames()`** method has been adapted to return only the names of properties with values of a supported type. As a result, accessing properties returned by this method no longer leads to the aforementioned exception.

BZ#652233

By default, Qpid utilities do not authenticate to a broker when the **`PLAIN`** authentication method is used. Previously, when the **`qpidd`** service was configured to enforce this authentication method, an attempt to run the **`qpid-stat -c`** command failed with a traceback, and the **qpid-tool** utility incorrectly produced empty results. With this update, these utilities now display a proper error message when they fail to establish connection with a broker.

BZ#653167

Prior to this update, the **qpid-config** utility sometimes failed to handle and display certain queues. This happened if the queue attributes, which require integer values (such as **`qpid.max_size`**, or **`qpid.max_count`**), contained string values. With this update, **qpid-config** now displays all queues with their attributes even when the attributes contain invalid values.

BZ#653923

When a connection is established with the **`reconnect`** parameter set to **`true`**, it attempts to retransmit unacknowledged messages whenever it reconnects. Prior to this update, messages transmitted after such reconnection from a C++ client were not marked as **`redelivered`**. With this update, the underlying source code has been changed, and such messages are now marked as **`redelivered`** as expected.

BZ#654020

Previous versions of the MRG Messaging Broker allowed users to declare queues with invalid argument values (for example, negative integers or non-integer values). Such values were ignored by a broker, but querying the queue would still return these invalid values. This update adapts the underlying source code to reject incorrect declarations. As a result, an attempt to declare a queue with invalid argument values now fails with an error.

BZ#654461

Prior to this update, if a queue (or any other broker object) was created and a management operation (such as **purge**) was invoked on the new object shortly after its creation (that is, 10 seconds or less), the operation might fail with unknown object. This update corrects this problem, and allows operations to be reliably invoked on new objects.

BZ#657523

A Spout program (for example, **csharp.example.spout**) sent messages but did not wait for the sending operation to complete before closing the session. As a result, not all of the messages processed by Spout's **sender.Send()** function were received by the session receiver. With this update, the **session.Sync()** function is called before closing the session to assure all messages are received by the session receiver.

BZ#658448

Due to a source file name (**csharp.map.receiver.cs**) being misspelled, it was difficult to find that file using normal search methods. This update fixes the spelling error, and the aforementioned file is shown in directory searches as expected.

BZ#659071

As the **qpid::messaging::Connection:isOpen()** method was previously not marked as const, it could not be invoked on objects referenced through a variable of type const. This update adds a const equivalent to the **qpid::messaging::Connection:isOpen()** function, fixing this issue.

BZ#663013

Prior to this update, the default address created by the Python client was incorrectly marked as *durable*. With this update, the underlying source code has been modified to address this issue, and the Python client no longer marks the default address as *durable*.

BZ#663022

Using the ***assert*** parameter in the address string for object creation (for example: a queue, an exchange) triggers a verification process before messages are sent to an end object, if the created object is of a specific type. Prior to this update, the Python client sent messages to the end objects without checking the object's type. Thus, the Python client was, for example, able to send messages to an exchange even if the ***assert*** parameter was used and set for a different type. With this update, the underlying source code has been modified to address this issue, and ***assert*** types are properly checked.

BZ#664486

The **reroute** management method allows messages to be rerouted to a specific exchange. However, if there were no matching bindings, messages could have been dropped even if that exchange had the ***alternate-exchange*** option defined. This update modifies the broker to properly reroute messages through the alternate exchange, if defined.

BZ#666931

Due to the absence of Boolean variations of the .NET Binding Receiver functions **Get()** and **Fetch()**, access to native functionality of the messaging client (for example, compiling) was not possible. This update adds the missing **Get()** and **Fetch()** functions to the .NET Binding

layer.

BZ#667172

When a connection is established with the **reconnect** parameter set to **true**, it attempts to retransmit unacknowledged messages whenever it is reconnected. Prior to this update, messages transmitted after such reconnection from a JMS (Java Message Service) client were not marked as **redelivered**. With this update, the underlying source code has been changed, and such messages are now marked as **redelivered** as expected.

BZ#667428

Prior to this update, messages could have been enqueued outside of transaction boundaries for transactions that were in progress during a transparent failover. With this update, in case a transparent failover occurs, all transaction sessions are closed and an exception is thrown, and transactions in progress are rolled back.

BZ#667771

Prior to this update, when setting up a durable subscriber with a destination created using an addressing string, a **javax.jms.InvalidDestinationException** exception occurred. The problem has been fixed in this update so the **createDurableSubscription()** method now modifies its destination validation to check if the destination, which was passed in, is an instance of **javax.jms.Topic** and **org.apache.qpid.client.AMQDestination** instead of just checking if the destination is an instance of **org.apache.qpid.client.AMQTopic**; thus the **createDurableSubscription** method is now able to set up a durable subscriber from a destination created using an address string.

BZ#674005

Prior to this update, when setting up a **QueueReceiver** interface with a destination created using an addressing string, a **java.lang.ClassCastException** exception occurred. The problem has been fixed with this update by modifying the **createReceiver()** method to check if the destination, which was passed in, is an instance of **javax.jms.Queue** and **org.apache.qpid.client.AMQDestination** instead of just casting the destination into **org.apache.qpid.client.AMQTopic**; thus the **createReceiver()** method is now able to set up a **QueueReceiver** interface from a destination created using an address string.

BZ#668580

Prior to this update, the messaging binding object access was not properly interlocked. As a consequence, object deletions could have failed with an access violation under certain circumstances. This problem has been fixed by adding interlocks to serialize references to objects being deleted so that object deletions no longer fail due to access violations.

BZ#671369

Prior to this update, the remote direct memory access (RDMA) client terminated unexpectedly with a segmentation fault under certain circumstances if there was no Simple Authentication and Security Layer (SASL) mechanism specified either on a client or server side. The problem has been fixed in this update so that the client now runs as expected even if no SASL mechanism is specified.

BZ#674390

Prior to this update, the encoding of string data into a message buffer did not check if there was enough available space in the buffer for the string. As a consequence, encoding string data that was too large for a given buffer corrupted the buffer memory, and could terminate the broker unexpectedly under certain circumstances. The fix has been provided in this update so that string encoding functionality now verifies the message buffer is large enough to hold the encoded string. If there is not enough room in the buffer to hold the encoded string, an exception is returned to the caller and the buffer is not modified; thus buffer memory corruption no longer occurs and the broker does not terminate unexpectedly.

BZ#674631

In subscriptions using a broker managed credit window, this window was sometimes not moved forward correctly in the presence of rejected messages. As a consequence, sessions that were rejecting messages could have stalled. Also, the sessions did not receive any further messages even if messages were available on the queue they were subscribed to. The problem has been fixed by ensuring that the credit window always moves correctly for rejected messages so that sessions that are rejecting messages now continue to receive messages as expected.

BZ#674678

Prior to this update, closed receivers with an address defined with **`qpid::messaging::Address`** and with the **`delete:always`** option set did not delete queues and did not reroute acquired messages to an alternate exchange. The problem has been fixed in this update so that the closed receivers now properly delete queues and reroute acquired messages as expected.

BZ#675921

Prior to this update, the internal test that showed management objects for inter-broker bridges could have been created inconsistently in a cluster. As a consequence, it was possible for brokers to shut down with an invalid argument error under certain circumstances. The fix for this bug has been provided in this update, correcting the inconsistency so that the bridge management objects are created consistently in a cluster.

BZ#676345

Prior to this update, running the **`qpid-config -a guest/guest@`*`host`*** command against a remote broker resulted in an exception similar to the following under certain circumstances if the **`ANONYMOUS`** method was used for authentication on the remote server:

```
SessionException: ExecutionException(error_code=403, command_id=serial(29),
class_code=0, command_code=0, field_index=0, description=u'unauthorized-
access: authorised user id : anonymous@QPID but user id in message declared
as guest (qpid/broker/SemanticState.cpp:473)', error_info={}, channel=1,
id=serial(2))
```

The problem has been resolved in this update so that running the command no longer results in an exception.

BZ#679212

When rerouting orphaned or rejected messages to a queue's alternate exchange, if there were no matching bindings the broker did not follow the exchange's ***`alternate-exchange`*** option. As

a result, orphaned or rejected messages were dropped instead of following a legitimately defined path. The problem has been fixed in this update by modifying the broker to test whether the rerouting through queue's alternate exchange was successful. If not, the attempt made to reroute to that exchange's alternate exchange is defined so that orphaned or rejected messages are no longer dropped if the exchange they are routed to first has an alternate exchange.

BZ#679216

A queue can be configured with an alternate exchange such that messages routed via the alternate might find their way back to the same queue. Prior to this update, when such a queue was deleted, the messages in the queue might circulate indefinitely within the broker. This update corrects this behavior.

BZ#679723

Prior to this update, the server example C++ application (the **`server.cpp`** file) required a second input argument, that was not used, before entering the third input argument. Only the first and the third input arguments were used by the application. The problem has been fixed in this update so that the application now uses the first and second input arguments as expected.

BZ#680228

Prior to this update, running the topic scalability test caused the broker to become unresponsive when more than 100 subscribers were present in durable mode. This was due to a deadlock in *Berkeley DB* (BDB) database's **`del()`** function. The problem has been fixed in this update so that the broker does not hang anymore with a large number of subscribers.

BZ#680477

When an exchange was declared with a type unknown to the C++ Broker, the incorrect error code 530 — not-allowed was returned. With this update, error handling for the described scenario has been fixed and now the correct error code 404 — not-found is returned.

BZ#680479

When the **`message_cancel()`** call was issued for an unknown subscription destination, the erroneous cancel was ignored and nothing happened. With this update, the broker code was modified to raise an appropriate error as required by the specification, and the 404 — not-found error code is returned in the described scenario.

BZ#681026

When a cluster was run with persistent message store, durable messages and a management agent such as **`qpid-tool`** or **`cumin`**, a broker sometimes terminated with the invalid argument error message. With this update, issues in the store that caused inconsistencies in a cluster have been fixed and brokers no longer terminate in the described scenario.

BZ#681313

The **`rdma`** driver adds a small trailer to outbound buffers. The size of this header was not accounted for when the buffer's size was passed to the codec. If the codec filled all available buffer space, the **`rdma`** driver overwrote the end of the buffer when adding the trailer. As a consequence, the **`qpidd`** server terminated unexpectedly when implementing RDMA (Remote

Direct Memory Access). With this update, a patch has been applied that allows the **RdmaIO** layer to reserve header space in sent buffers, fixing this bug.

BZ#681331

When an exchange was declared with the ***alternate-exchange*** configuration option on one node of a cluster, other nodes in the cluster were not made aware of this configuration, preventing communication with new cluster members. With this update, the code for the cluster update process has been fixed and alternate exchanges now work as expected in a cluster.

BZ#682218

Previously, the broker maintained two lists of exclusive queues, one for a session and one for the connection. The latter list was incorrect. Deletion of queues prior to the session closing did not result in removal of all references to the queue, thus preventing the queue object from being freed. Consequently, exclusive queues that were deleted too soon would not actually free up and would still be reported by management tools. With this update, the broker will only maintain one list of exclusive queues at session scope, and delete any reference in that list when a queue is deleted too soon. Deletion now frees the queue up and the management reports are accurate.

BZ#684182

Due to package reorganization, a direct upgrade of the *debuginfo* package from version 1.3.2 to a later one fails due to file conflicts. As a consequence, installation or upgrade of the *messaging* and *qmf* packages failed with file conflicts if debug symbols were installed and requested to automatically update. With this update, the following workaround is provided: the previous version of the *debuginfo* package needs to be manually uninstalled before attempting to install or upgrade to newer *messaging* and *qmf* packages, with the command below:

```
$ rpm -ev qpid-cpp-mrg-debuginfo
```

This workaround does not introduce any limitations, and is straightforward for users of *debuginfo* packages to execute.

BZ#690261

Previously, when the qpidd server was restarted, if the static keys for brokers were the same, the necessary bindings failed to be created during the broker federation if a route with the same name existed. Consequently, some links were lost after the restart. This bug has been fixed and the interface now requires the remote origin when deleting a route.

BZ#692132

When a broker session was enqueuing a message at the exact moment a receiver was dequeuing it, the broker sometimes terminated unexpectedly. With this update, the **enqueueComplete()** call, responsible for re-queuing messages into a session, has been removed and this bug no longer occurs.

BZ#692134

Previously, in the **qpid-config** utility command line help, descriptions for **--max-queue-size** and **--max-queue-count** command options were incorrect. With this update, the descriptions have been amended.

BZ#693407

Previously, it was possible to define a session name too long for the object defined in the management schema to handle. Due to a missing check in the code, this caused the broker to terminate unexpectedly. With this update, additional exception handling has been added to the **`Timer.cpp`** file and the broker no longer crashes.

BZ#694617

When using the JMS client, if a queue or an exchange object was created with the ***`alternate-exchange`*** option defined, Qpid ignored the alternate exchange setting. This update adds an upstream patch that adapts the underlying code, and such objects are created with the defined ***`alternate-exchange`*** option as expected.

BZ#695263

Prior to this update, the Qpid broker could terminate unexpectedly with a segmentation fault under heavy load. This happened due to a race between the **`SessionState`** thread and the journaling thread. This update adds a lock to the **`SessionState`** thread and the deadlock no longer occurs.

BZ#695353

Prior to this update, the address parser of the Java client failed to parse the destination address if the node durability was defined with string values (**`True`** or **`False`**) and terminated with **`java.lang.ClassCastException`**. With this update, the parser correctly handles both the Boolean and the string versions of the parameters.

BZ#695716

Prior to this update, the C++ broker could terminate unexpectedly under heavy load. This happened due to a deadlock, which occurred when the broker without a store handled concurrent DTX operations on durable queues and durable messages. This update applied an upstream patch, which adds locks to the processes and prevents the deadlock.

BZ#696373

The JMS client did not contain information on prerequisites and basic run instructions for the delivered examples. This update adds a README file with the information.

BZ#696637

Previously, the message queue was inconsistent within nodes of a cluster if the message exchange used the **`ive`** option. This occurred because the message exchange holds the last message and the message cannot be transferred to new members. With this update, the **`ive`** option has been disabled for clusters, thus fixing this bug.

BZ#696655

Prior to this update, a cluster node shut down with an **`inconsistent`** error after an exchange was deleted. This happened because the messages from the deleted exchange were still stored in the exchange cache and the broker attempted to deliver them. With this update, such messages are marked as invalid and are ignored by the broker.

BZ#696698

Prior to this update, if multiple queues were bound to the same routing key and the exchange failed to deliver messages to one of the several defined queues, the delivery to the other queues failed as well. With this update, the exchange delivers messages to every queue, which has not recorded an error since the delivery failure.

BZ#696974

Prior to this update, the **qpid-config** help contained an incorrect description of the **-a** (or **--broker-addr**) command line option and incorrect name of the **--force-if-used** option. With this update, the mistakes have been fixed.

BZ#697913

Prior to this update, the application occasionally became unresponsive when using a synchronous operation. This happened due to a deadlock of two reporting threads during exception reporting. With this update, the underlying code has been modified and the deadlock no longer occurs.

BZ#698721

When sending durable messages, the python spout example sometimes experienced performance issues. This happened because the spout enforced synchronous transmission. However, this solution is not appropriate. With this update, the send options have been changed to use asynchronous and durable transmission.

BZ#701777

If a broker withholds the acknowledgments of a QMF (*Qpid Management Framework*) agent, the agent gets into the flow-stopped mode. Prior to this update, the agent became locked in the mode if it had received a method-call message while in the mode. With this update, the agent's connection is shut down and re-established in such circumstances and the agent continues to work as expected.

Enhancements

BZ#547743

Prior to this update, the Messaging API was unable to dynamically create and delete broker entities such as queues, exchanges and their bindings. Due to this, the application had to use older APIs and workarounds to process these entities. With this update, management methods are exposed to create and delete these entities on the broker via QMF. With QMFv2 this can be achieved by sending a message of a defined format to a specified address.

BZ#569918

With this update, the SSL client options are supported for Qpid messaging.

BZ#585844

Prior to this update, queues marked for automatic deletion were deleted immediately after being released from a session. If a failover occurred the queues were permanently lost. This update introduces a delay between the time a queue is eligible for automatic deletion and the time it is actually deleted. Additionally, if this delay period is longer than the failover time, then the queue

survives the failover and if it is not required it is automatically deleted.

BZ#606357

Prior to this update, the oldest messages were removed first when the queue removed messages to remain within the predefined size. Due to this behavior, in certain circumstances messages that were more valuable would be removed while less valuable messages remained. This update makes such queues configurable to recognize message priority. Now, messages can be removed according to their priority.

BZ#621467

This update adds the option to acknowledge only specific messages. Now, applications can tie acknowledgments to asynchronous confirmations they receive on successful processing of the messages.

BZ#624793

With this update, the *Advanced Message Queuing Protocol* (AMQP) allows the user to inspect exclusive queues that previously could not be browsed.

BZ#632000

When a user subscribed to an LVQ queue, already browsed messages were not replaced, and each subsequent update caused the queue to grow. This prevented subscribers from using the LVQ queues as a read-only cache of the latest values. This update introduces the **`qpid.last_value_queue_key`** configuration options, allowing a user to specify the key used by the application to determine the message equivalence. Now, subscribers always receive latest updates, and updated messages replace older messages with a matching key.

BZ#632396

Previously, when a user attempted to use a **`java.util.UUID`** object as a value of a Map message or in a List that itself is used within a Map message, the Java Message Service (JMS) client failed with an exception. With this update, the JMS client has been adapted to allow the use of the **`java.util.UUID`** objects in Map messages. Also, the JMS client is now able to decode **`java.util.UUID`** from an incoming Map message.

BZ#667463

With this update, MRG now includes a **`tcp_nodelay`** option for the Python API.

BZ#660291

This update adds a flow control mechanism, allowing the broker to measure the current level of data in each queue via the **`high_watermark`** and **`low_watermark`** flags. This flow control mechanism allows the use of credit to prevent a queue overflow event and to provide information to a client about the data levels in a queue.

BZ#660289

Prior to this update, messaging was unable to monitor the growing queue depth aside from polling constantly or waiting until the maximum level was reached to issue a warning. The broker now allows QMF events to be generated when the queue depth reaches a previously configured threshold, providing an early warning for elongated messaging queues.

BZ#657398

Previously, logging level at runtime could not be altered without restarting the broker. A management method now used allows the user to change the level of logging while the program is running, without having to restart the broker. This allows users to get detailed logs during troubleshooting then return to normal logging settings to prevent excessive logs.

BZ#667970

A new feature is added to track statistics about the number of messages transferred over the connection instead of tracking only the numbers of frames and bytes transferred across the connection.

BZ#681166

With this update, the implementation of map encoding into AMQP in Python has been improved; the **`for`** loop to traverse the dictionary was replaced with the **`map()`** function. The map encoding is significantly faster.

BZ#681279

With this update, the number of buffers allocated on a client by the **`TCPConnector`** interface was reduced from 32 to 4, saving considerable amount of memory allocated per connection.

BZ#693895

With this update, the logic that stores callbacks in the **`CyrusSasl`** constructor has been enhanced to make C++ clients behave similarly to other language clients as follows: if there is no username, then do nothing with either **`NAME`** or **`PASSWD`** callbacks; else, if there is a username but no password, then explicitly store an empty **`PASSWD`** callback.

BZ#700822

Producer flow control for clusters is now supported.

All users are advised to upgrade to these updated packages, which fix these bugs and add these enhancements.

Chapter 3. MRG Realtime on Red Hat Enterprise Linux 6

3.1. RHSA-2013:0622 – Important: kernel-rt security and bug fix update

The *kernel-rt* packages contain the Linux kernel, the core of any Linux operating system.

This update provides a build of the kernel-rt package for MRG 2.3, which is layered on Red Hat Enterprise Linux 6.

The security fixes and bug fixes provided in this advisory are documented below:

Security Fixes

CVE-2013-0268, Important

A flaw was found in the way file permission checks for the **/dev/cpu/[x]/msr** files were performed in restricted root environments (for example, when using a capability-based security model). A local user with the ability to write to these files could use this flaw to escalate their privileges to kernel level, for example, by writing to the **SYSENTER_EIP_MSR** register.

BZ#CVE-2013-0871, Important

A race condition was found in the way the Linux kernel's **ptrace** implementation handled **PTRACE_SETREGS** requests when the debuggee was woken due to a SIGKILL signal instead of being stopped. A local, unprivileged user could use this flaw to escalate their privileges.

BZ#CVE-2013-1763, Important

An out-of-bounds access flaw was found in the way **SOCK_DIAG_BY_FAMILY** Netlink messages were processed in the Linux kernel. A local, unprivileged user could use this flaw to escalate their privileges.

BZ#CVE-2012-4542, Moderate

It was found that the default SCSI command filter does not accommodate commands that overlap across device classes. A privileged guest user could potentially use this flaw to write arbitrary data to a LUN that is passed-through as read-only.

BZ#CVE-2013-0290, Moderate

A flaw was found in the way the **__skb_recv_datagram()** function in the Linux kernel processed payload-less socket buffers (skb) when the **MSG_PEEK** option was requested. A local, unprivileged user could use this flaw to cause a denial of service (infinite loop).

The CVE-2012-4542 issue was discovered by Paolo Bonzini of Red Hat.

Bug Fixes

BZ#858396

There was high contention on run-queue lock when load balancing before idling, causing latency spikes on high CPU core count systems. With this update, IPI is used to send notification to cores with pending work, and the cores push the work rather than trying to pull it,

resolving this issue.

BZ#909965

Previously, ACPI lock was converted to an rt_mutex, leading to a traceback when scheduling while atomic. With this update, ACPI lock has been converted back to a raw spinlock.

BZ#912942

Fibre Channel (FC)/iSCSI device state was set to off-line and after a timeout, not set back to running. Such a device would not come back online after a fast_io_fail or timeout. With this update, an explicit check for the device being offline has been added, and the device is set back to running when re-initializing, allowing devices to recover after a failure or timeout.

Users are advised upgrade to these updated *kernel-rt* packages, which contain backported patches to correct these issues. The system must be rebooted for this update to take effect.

3.2. RHSA-2013:0566 – Important: kernel-rt security and bug fix update

The *kernel-rt* packages contain the Linux kernel, the core of any Linux operating system.

This update provides a build of the kernel-rt package for MRG 2.3, which is layered on Red Hat Enterprise Linux 6.

Descriptions of the security fixes provided in this advisory can be viewed at https://rhn.redhat.com/errata/RHSA-2013-0566.html. The bug fixes and enhancements provided in this advisory are documented below:

Upgrade to latest stable Realtime kernel

The MRG Realtime kernel has been rebased to version 3.6, which includes updates to kernel NIC drivers, adds simple wait-queue construct, and uses the SLUB memory allocator by default. (BZ#866596)

Bug Fixes

BZ#707262

```
The MRG Hewlett Packard Smart Array (HPSA) driver has been rebased to the
latest upstream version (3.7) and updated with selected Red Hat Enterprise
Linux 6 fixes and enhancements. The current MRG HPSA driver version is
2.0.2-4-RH1+MRG-RT.
```

BZ#773017

Previously, the MRG Realtime **-devel* packages contained two executable files: **scripts/conmakehash** and **scripts/pnmtologo** that were never changed between kernel variants, and generate the same hash value for all variants. Consequently, debuginfo conflicts occurred on these files. Now, these files have been removed from the **-devel* packages since they are not used to generate kernel modules, so there are no debuginfo conflicts.

BZ#880749

An upstream patch which introduced the rt_mutex construct had exported previous inline functions as **EXPORT_SYMBOL_GPL**, which prevented third-party kernel modules from building. The **EXPORT_SYMBOL_GPL** functions are now changed to **EXPORT_SYMBOL**, so third party modules can build successfully.

BZ#887767

The MRG Realtime kernel, which is newer than the Red Hat Enterprise Linux 6 kernel, has a "filesystem error nag" feature that gets cleared by newer user-space tools so that it is only printed once. This feature is unknown to Red Hat Enterprise Linux 6 filesystem or logging tools, so it never gets cleared. Consequently, the filesystem status is sent to the console and system log each day. The kernel function **print_daily_error_info()** has been patched, and now returns without printing daily error message.

BZ#866600

Support has been added for IEEE 1588 Precision Time Protocol (PTP) to the MRG 2.3 kernel. PTP provides greater time synchronization than is offered by Network Time Protocol (NTP).

Users are advised upgrade to these updated *kernel-rt* packages, which contain backported patches to correct these issues. The system must be rebooted for this update to take effect.

3.3. RHBA-2013:0563 – Red Hat Enterprise MRG Realtime 2.3 bug fix update

Red Hat Enterprise MRG is a next-generation IT infrastructure incorporating Messaging, Realtime, and Grid functionality. It offers increased performance, reliability, interoperability, and faster computing for enterprise customers.

MRG Realtime provides the highest levels of predictability for consistent low-latency response times to meet the needs of time sensitive workloads. MRG Realtime provides new levels of determinism by optimizing lengthy kernel codepaths to ensure that they do not become bottlenecks. This allows for better prioritization of applications, resulting in consistent, predictable response times for high-priority applications.

This update provides bug fixes for the following packages:

libibverbs-rocee

The *libibverbs-rocee* packages provide a library to enable userspace processes to use RDMA

over Converged Ethernet (RoCE) "verbs" acording to the InfiniBand Architecture Specification and the RoCE Protocol Verbs Specification.

libmlx4-rocee

The *libmlx4-rocee* packages provide a device-specific driver for Mellanox ConnectX InfiniBand host channel adapters (HCAs) for the *libibverbs-rocee* library.

rteval

The *rteval* package contains a utility for measuring various aspects of real-time behavior on systems under load. The script unpacks the **hackbench** stress test and benchmark utility, and the kernel source code from the *rteval-loads* package. It then builds hackbench, and goes into a loop, running **hackbench** and compiling a kernel tree. During that loop, the **cyclictest** program is run to measure event response times. After the run time completes, a statistical analysis of the event response times is performed and printed to the screen.

rt-firmware

The *rt-firmware* package contains the contents of the latest linux-firmware tree from the kernel.org project for firmware blobs.

tuna

The *tuna* package provides graphical and command-line interfaces for changing scheduler and interrupt request (IRQ) settings. Changes can be made to CPUs, by thread or at the IRQ level, taking into account the topology of multi-socket and multi-core systems. *Tuna* provides the ability to isolate CPU cores and sockets for use by a specific application or hardware device.

Bug Fixes

BZ#893133

With this update, MRG Realtime provides updates to the latest *rt-firmware* package that includes new **bnx2x** and **cnic** firmware.

BZ#871117

With this update, MRG Realtime provides updates to the latest versions of the *libibverbs-rocee* and *libmlx4-rocee* packages that provide high performance networking.

BZ#601234

The **tuna** graphical user interface now allows changing individual thread scheduling policy inside a thread group. Users can now modify the priority or affinity of individual threads.

BZ#871598

Tuna uses **python-schedutils**, which used to incorrectly raise a SystemError exception instead of a OSError exception when a specified PID did not exist. **python-schedutils** has since been corrected, and **tuna** was also modified to expect the new OSError exception.

Users of the real-time capabilities of Red Hat Enterprise MRG 2.2, which is layered on Red Hat

Enterprise Linux 6, are advised to upgrade to these updated packages, which fix these bugs.

3.4. RHBA-2012:1492 – Red Hat Enterprise MRG Realtime 2.2 bug fix update

Red Hat Enterprise MRG is a next-generation IT infrastructure incorporating Messaging, Realtime, and Grid functionality. It offers increased performance, reliability, interoperability, and faster computing for enterprise customers.

MRG Realtime provides the highest levels of predictability for consistent low-latency response times to meet the needs of time sensitive workloads. MRG Realtime provides new levels of determinism by optimizing lengthy kernel codepaths to ensure that they do not become bottlenecks. This allows for better prioritization of applications, resulting in consistent, predictable response times for high-priority applications.

This update provides bug fixes for the following packages:

rt-setup

The *rt-setup* package configures settings required by the Red Hat Enterprise Linux realtime environment, such as creating the realtime group, adding realtime user privileges to PAM (Pluggable Authentication Models), enabling the configuration of **`kdump`** in real time, and disabling **`irqbalance`** by default.

rtctl

The *rtctl* package contains a set of scripts, which are used to manipulate the scheduling priorities of groups of system threads.

Bug Fixes

BZ#868329, BZ#868442, BZ#868446

MRG Realtime was using the Red Hat Enterprise Linux kernel as the kdump/kexec kernel, but this caused problems due to version differences. With the MRG Realtime 2.x kernels, the **`kexec-tools`** failed to create the initrd image for the Red Hat Enterprise Linux 6 kdump kernel. MRG Realtime now uses the Realtime kernel as the kdump/kexec kernel to eliminate the configuration issues between kernel versions.

BZ#878536

Due to the way the **`rtctl`** configuration file selection works, if there was not a **`rtgroups`** configuration file for a specific kernel or for the general kernel version (for example a 3.2 kernel), the default configuration file was used. Using the default configuration file could send chrt error messages to the console and some threads could have unusual priorities. This update provides a per-kernel **`rtgroups`** configuration file, which eliminates the chrt error messages and defines default **`kthread`** priorities for each kernel.

Users of the real-time capabilities of Red Hat Enterprise MRG 2.2, which is layered on Red Hat Enterprise Linux 6, are advised to upgrade to these updated packages, which fix these bugs.

3.5. RHSA-2012:1491 – Important: kernel-rt security and bug fix update

The *kernel-rt* packages contain the Linux kernel, the core of any Linux operating system.

This update provides a build of the kernel-rt package for MRG 2.2, which is layered on Red Hat Enterprise Linux 6.

Upgrade to latest stable Realtime kernel

Previously, the MRG Realtime kernel missed several upstream changes to the PREEMPT_RT based kernels. MRG Realtime is now rebased to latest stable Realtime kernel, so it can take advantage of latest Realtime design changes. (BZ#864568)

Security Fixes

CVE-2012-3520, Important

A flaw was found in the way Netlink messages without **SCM_CREDENTIALS** (used for authentication) data set were handled. When not explicitly set, the data was sent but with all values set to 0, including the process ID and user ID, causing the Netlink message to appear as if it were sent with root privileges. A local, unprivileged user could use this flaw to send spoofed Netlink messages to an application, potentially resulting in the application performing privileged operations if it relied on **SCM_CREDENTIALS** data for the authentication of Netlink messages.

CVE-2012-4508, Important

A race condition was found in the way asynchronous I/O and **fallocate()** interacted when using the ext4 file system. A local, unprivileged user could use this flaw to expose random data from an extent whose data blocks have not yet been written, and thus contain data from a deleted file.

CVE-2012-2133, Moderate

A use-after-free flaw was found in the Linux kernel's memory management subsystem in the way quota handling for huge pages was performed. A local, unprivileged user could use this flaw to cause a denial of service or escalate their privileges.

CVE-2012-3511, Moderate

A use-after-free flaw was found in the **madvise()** system call implementation in the Linux kernel. A local, unprivileged user could use this flaw to cause a denial of service or escalate their privileges.

CVE-2012-4565, Moderate

A divide-by-zero flaw was found in the TCP Illinois congestion control algorithm implementation in the Linux kernel. If the TCP Illinois congestion control algorithm was in use (the sysctl **net.ipv4.tcp_congestion_control** variable was set to "***illinois***"), a local, unprivileged user could trigger this flaw and cause a denial of service.

CVE-2012-0957, Low

An information leak flaw was found in the **uname()** system call implementation in the Linux kernel. A local, unprivileged user could use this flaw to leak kernel stack memory to user-space by setting the **UNAME26** personality and then calling the **uname()** system call.

CVE-2012-3400, Low

Buffer overflow flaws were found in the **udf_load_logicalvol()** function in the Universal Disk Format (UDF) file system implementation in the Linux kernel. An attacker with physical access to a system could use these flaws to cause a denial of service or escalate their privileges.

CVE-2012-3430, Low

A flaw was found in the way the **msg_namelen** variable in the **rds_recvmsg()** function of the Linux kernel's Reliable Datagram Sockets (RDS) protocol implementation was initialized. A local, unprivileged user could use this flaw to leak kernel stack memory to user-space.

Red Hat would like to thank Pablo Neira Ayuso for reporting CVE-2012-3520; Theodore Ts'o for reporting CVE-2012-4508; Shachar Raindel for reporting CVE-2012-2133; and Kees Cook for reporting CVE-2012-0957. Upstream acknowledges Dmitry Monakhov as the original reporter of CVE-2012-4508. The CVE-2012-4565 issue was discovered by Rodrigo Freire of Red Hat, and the CVE-2012-3430 issue was discovered by the Red Hat InfiniBand team.

Bug Fixes

BZ#843130

The TCP stack for the MRG Realtime kernel was vulnerable to a blind-reset attack. Consequently, an off-path attacker could inject packets into a TCP stream. This update implements the RST and SYN components of RFC 5961, which prevents blind-reset and blind SYN attacks on the MRG Realtime kernel TCP stack.

BZ#856243

Nesting local and remote CPU locks caused a lockdep warning about potentially recursive locking dependencies. Now, only the remote cpu lock is acquired, which provides sufficient protection. The lockdep warning no longer occurs.

BZ#853495

An attempt to load the microcode module on an unsupported AMD CPU would silently fail, without quitting the microcode initialization. Consequently, multiple backtraces, as many as the amount of CPUs on that given system, would be printed to the console and boot logs. This update backports the solution available on newer kernels, which avoids loading the microcode module on unsupported AMD CPU families. The microcode module now loads cleanly without backtraces from the microcode module.

BZ#859226

A subtlety in the relation between preemption depth (**preempt_count**) and the raw spinlocks in the PREEMPT_RT based kernels led to a context where printk buffers were never flushed to the console. Some time after boot, after a certain condition was triggered, the kernel-generated log could not reach the console and system log (**rsyslog**). That specific verification of the **preempt_count** variable was adapted to the reality of the Realtime kernel. Logs generated by

the kernel are now successfully sent to both the console and the system log, according to the system configuration.

Users are advised upgrade to these updated *kernel-rt* packages, which contain backported patches to correct these issues. The system must be rebooted for this update to take effect.

3.6. RHSA-2012:1282 – Moderate: kernel-rt security, bug fix, and enhancement update

The *kernel-rt* packages contain the Linux kernel, the core of any Linux operating system.

This update provides a build of the kernel-rt package for MRG 2.2, which is layered on Red Hat Enterprise Linux 6.

Upgrade to an upstream version

The *kernel-rt* packages have been upgraded to upstream version 3.2, which provides a number of bug fixes and enhancements over the previous version. (BZ#798421)

Security Fix

CVE-2012-4398, Moderate

It was found that a deadlock could occur in the Out of Memory (OOM) killer. A process could trigger this deadlock by consuming a large amount of memory, and then causing request_module() to be called. A local, unprivileged user could use this flaw to cause a denial of service (excessive memory consumption).

Red Hat would like to thank Tetsuo Handa for reporting this issue.

Bug Fixes

BZ#786083

In the MRG 2.1 Realtime kernel, messages from iptables were not being successfully logged into syslog (rsyslog). With this update, this bug has been fixed and the iptables messages are now properly recorded in syslog.

BZ#799385

Previously, the **`spin_trylock()`** function in the migrate timers code disabled preemption, which could cause a deadlock. The code has been changed to allow the lock to sleep in the described scenario and the data is now protected by disabling CPU migration, thus preventing this bug.

BZ#799386

Previously, the futex proxy handler could take the **`pi_lock`** from a task without disabling interrupts. If an interrupt came in while the lock was held, and that interrupt took the same lock, a deadlock occurred. With this update, interrupts has been disabled in the described scenario and the deadlock no longer occurs when **`pi_lock`** is taken.

BZ#799389

Prior to this update, locks for per-CPU data on MRG Realtime kernels could be reactived after a CPU comes online, but with an owner that had not released them. This caused various problems such as blocking the original owner of the lock. Locking in MRG Realtime kernels has been changed to use specific CPU masks and a lock is now released when the CPU is still offline, thus fixing this bug.

BZ#799391

When a task called the **wait_task_interactive()** function, it waited on the task to change its state and could get the wrong result on MRG Realtime kernels. The **spin_locks()** in the MTG Realtime kernel changes the real state of the task but keeps a mirror copy of the state that it resets to when the task unlocks the spin lock. The **wait_task_interactive()** function has been modified to be now aware of the mirror copy used by MTG Realtime kernels as well, thus fixing this bug.

BZ#799399

On MRG Realtime kernels, the **ftrace_dump()** debug function expects a lock, but does not check if interrupts are disabled. If interrupts were disabled, the MRG Realtime kernel could still take the lock. Consequently the **schedule()** function could be called with the kernel in the **atomic** context, indicating a bug had occurred. With this update, the MRG Realtime kernel has been modified to check if interrupts are disabled before giving the lock, thus fixing this bug.

BZ#814689

Previously, the **CONFIG_FIPS** configuration flag was not enabled, causing the **ssh** daemon to report errors on boot. This update enables the **CONFIG_FIPS** flag and the **ssh** daemon now starts up without errors.

BZ#815937

Previously, the firmware files for the RealTek RTL-8169 Gigabit Ethernet network card were moved from the main kernel upstream repository to the linux-firmware Git repository. This update adds the firmware files back to the MRG Realtime kernel and the network cards can now properly load them.

BZ#822298

Previously, the **CONFIG_COMPAT_VDSO** configuration flag caused kernel VDSOs (Virtual Dynamically-linked Shared Objects) to be mapped at predictable addresses. To reduce the security risk, **CONFIG_COMPAT_VDSO** has been disabled, which ensures kernel VDSOs are now mapped to random addresses.

Enhancements

BZ#725799

With this update, the MRG Realtime kernel is able to work as a diskless client.

BZ#798423

With this update, the InfiniBand core and InfiniBand drivers have been updated to more closely

reflect updates available in Red Hat Enterprise Linux 6.3.

BZ#825344

With this update, MRG Realtime kernel provides the *mrg-rt-release* package, which maintains the **/etc/mrg-realtime-release** file. This file contains the MRG release string, which indicates the major, minor, and errata releases of the MRG Realtime kernel.

Users are advised upgrade to these updated *kernel-rt* packages, which contain backported patches to correct these issues. The system must be rebooted for this update to take effect.

3.7. RHEA-2012:1280 – Red Hat Enterprise MRG Realtime 2.2 enhancement update

Red Hat Enterprise MRG (Messaging, Realtime, and Grid) is a next-generation IT infrastructure for enterprise computing. MRG offers increased performance, reliability, interoperability, and faster computing for enterprise customers.

MRG Realtime provides the highest levels of predictability for consistent low-latency response times to meet the needs of time sensitive workloads. MRG Realtime provides new levels of determinism by optimizing lengthy kernel codepaths to ensure that they do not become bottlenecks. This allows for better prioritization of applications, resulting in consistent, predictable response times for high-priority applications.

Enhancements

BZ#798426, BZ#798427

With this update, MRG Realtime introduces the *libibverbs-rocee* and *libmlx4-rocee* packages to provide high performance networking. The *libibverbs-rocee* package provides a library to enable user space processes to use the RoCE (RDMA over Converged Ethernet) "verbs" API according to the InfiniBand Architecture Specification and the RoCE Protocol Verbs Specification. The *libmlx4-rocee* package provides a device-specific driver for Mellanox ConnectX InfiniBand host channel adapters (HCAs) for the **libibverbs-rocee** library.

Users of the Realtime capabilities of Red Hat Enterprise MRG 2.2, which is layered on Red Hat Enterprise Linux 6, are advised to upgrade to these updated packages, which add these enhancements. Note that the system must be rebooted for this update to take effect.

3.8. RHSA-2012:0670 – Important: kernel-rt security and bug fix update

Red Hat Enterprise MRG is a next-generation IT infrastructure incorporating Messaging, Realtime, and Grid functionality. It offers increased performance, reliability, interoperability, and faster computing for enterprise customers.

MRG Realtime provides the highest levels of predictability for consistent low-latency response times to meet the needs of time sensitive workloads. MRG Realtime provides new levels of determinism by optimizing lengthy kernel codepaths to ensure that they do not become bottlenecks. This allows for better prioritization of applications, resulting in consistent, predictable response times for high-priority applications.

Security Fixes

CVE-2012-2123, Important

When a set user ID (setuid) application is executed, certain personality flags for controlling the application's behavior are cleared (that is, a privileged application will not be affected by those flags). It was found that those flags were not cleared if the application was made privileged via file system capabilities. A local, unprivileged user could use this flaw to change the behavior of such applications, allowing them to bypass intended restrictions. Note that for default installations, no application shipped by Red Hat for Red Hat Enterprise MRG is made privileged via file system capabilities.

CVE-2011-4086, Moderate

A flaw was found in the way the Linux kernel's journal_unmap_buffer() function handled buffer head states. On systems that have an ext4 file system with a journal mounted, a local, unprivileged user could use this flaw to cause a denial of service.

Bug Fixes

BZ#808271

The CAP_SYS_ADMIN check was missing from the dmesg_restrict feature. Consequently, an unprivileged and jailed root user could bypass the dmesg_restrict protection. This update adds CAP_SYS_ADMIN to both dmesg_restrict and kptr_restrict, which only allows writing to dmesg_restrict when root has CAP_SYS_ADMIN.

BZ#753230

Previously, the _copy_from_pages() function, which is used to copy data from the temporary buffer to the user-passed buffer, was passed the wrong size parameter when copying data. Consequently, if the user provided a buffer greater than PAGE_SIZE, the getxattr() syscalls were handled incorrectly. This update fixes _copy_from_pages() to use the ACL length, which uses a correctly-sized buffer.

BZ#813892

Some older versions of hardware or their software could not recognize certain commands and would log messages for illegal or unsupported errors the driver could not properly handle. This bug has been fixed and no bogus error messages are now returned in the described scenario.

BZ#818220

Previously, the qla2x00_poll() function did the local_irq_save() call before calling qla24xx_intr_handler(), which had a spinlock. Since spinlocks are sleepable in the real-time kernel, it is not allowed to call them with interrupts disabled. This scenario produced error messages and could cause a system deadlock. With this update, the local_irq_save_nort(flags) function is used to save flags without disabling interrupts, which prevents potential deadlocks and removes the error messages.

Users are advised upgrade to these updated *kernel-rt* packages, which contain backported patches to correct these issues. The system must be rebooted for this update to take effect.

3.9. RHBA-2012:0496 – Red Hat Enterprise MRG Realtime 2.1 kernel bug fix update

The *kernel-rt* packages contain the Linux kernel, the core of any Linux operating system.

This update provides a build of the kernel-rt package for MRG 2.1, which is layered on Red Hat Enterprise Linux 6.

Bug Fixes

BZ#675885

Previously, while working with an open file, the **epoll** kernel code was calling poll operations of another open file without holding the file's lock. Consequently, a deadlock in the code could occur when using the system console. Now, an upstream patch to use the **ep_poll_nested()** function has been provided and the system console now works correctly in the described scenario.

BZ#773746

The **i7core_edac** code did multiple probes for the same hardware component. As a consequence, boot time error message about the code being unable to create a duplicate file in the **/sys/devices/** directory was returned. With this update, duplicate probe code has been removed and the error messages are no longer returned in the described scenario.

BZ#787331

A previous Realtime work queue patch introduced a race conditions in the CPU hotplug code. Consequently, the system could become unresponsive in this code. This update reverts a few past updates to the work queue, which turned out to be unnecessary, and the hangs no longer occur in the described scenario.

BZ#796297

The **CONFIG_CC_OPTIMIZE_FOR_SIZE** configuration parameter, which was previously introduced to MRG Realtime, caused the **gcc** compiler to produce code prone to cache line bouncing. As a result, the kernel performance was decreased. This update turns off this parameter, thus improving cache utilization and performance.

BZ#804119, BZ#756631

Previously, configuration files in MRG Realtime and Red Hat Enterprise Linux 6 were not synchronized regarding modules and built-ins. Consequently, the **mkinitrd** daemon failed to find code that was part of a module in Red Hat Enterprise Linux 6 but part of a built-in in MRG Realtime. Now, the configuration files have been synchronized between these two products and **mkinitrd** correctly finds and sets up kernel components such as **kdump**.

Users of the Realtime capabilities of Red Hat Enterprise MRG 2.1, which is layered on Red Hat Enterprise Linux 6, are advised to upgrade to these updated packages, which fix these bugs. Note that the system must be rebooted for this update to take effect.

3.10. RHBA-2012:0495 – Red Hat Enterprise MRG Realtime 2.1 bug fix update

Red Hat Enterprise MRG (Messaging, Realtime, and Grid) is a next-generation IT infrastructure for enterprise computing. MRG offers increased performance, reliability, interoperability, and faster computing for enterprise customers.

MRG Realtime provides the highest levels of predictability for consistent low-latency response times to meet the needs of time sensitive workloads. MRG Realtime provides new levels of determinism by optimizing lengthy kernel codepaths to ensure that they do not become bottlenecks. This allows for better prioritization of applications, resulting in consistent, predictable response times for high-priority applications.

This update provides bug fixes for the following packages:

ibm-prtm

The *ibm-prtm* package contains a utility and a start-up script for IBM BladeCenter systems. It enables the EDAC (Error Detection and Correction) driver and also turns off the System Management Interrupts (SMI) generation, which improves system response time to events.

rteval

The *rteval* package contains a utility for measuring various aspects of real-time behavior on systems under load. The script unpacks the **hackbench** stress test and benchmark utility, and the kernel source code from the rteval-loads package, builds **hackbench**, and then goes into a loop, running **hackbench** and compiling a kernel tree. During that loop, the **cyclictest** program is run to measure event response times. After the run time completes, a statistical analysis of the event response times is performed and printed to the screen.

rt-setup

The *rt-setup* package configures settings required by the Red Hat Enterprise Linux real-time environment, such as creating the realtime group, adding realtime user privileges to PAM (Pluggable Authentication Models), enabling the configuration of **kdump** in real time, and disabling **irqbalance** by default.

tuna

The *tuna* package provides graphical and command-line interfaces for changing scheduler and interrupt request (IRQ) settings. Changes can be made to CPUs, by thread or at the IRQ level, taking into account the topology of multi-socket and multi-core systems. Tuna provides the ability to isolate CPU cores and sockets for use by a specific application or hardware device.

Bug Fixes

BZ#574166

Signal handlers were commented out during debugging session and not restored for release, which caused the early termination logic to fail. Now, the signal handling logic has been restored and the early termination logic works appropriately.

BZ#773075

The **cyclictest** test always ran with the **--smp** option rather than **--numa** on appropriate

systems because the **numanode** parameter was not correctly passed to the child process. This bug has been fixed and **cyclictest** now uses the correct option when starting measurement threads.

BZ#767605

SMI remediation was not working correctly, resulting in error messages at boot due to an incorrect patch applied to the base **ibm-prtm** script. This update provide an appropriate patch that fixes SMI remediation on IBM non-EFI systems and the error messages are no longer returned.

BZ#545539

Tuna now contains a man page that is correctly displayed with the command **man tuna**.

BZ#698481

Some udev rules added by MRG Realtime were being called at a much higher frequency than anticipated, which could cause inordinately long boot times on systems with more than 15 cores. Now, child process creation has been removed from the Realtime udev rules and the boot times are shorter on such systems.

BZ#791371

Previously, the Realtime bandwidth limiting feature for **SCHED_FIFO/SCHED_RR** threads was disabled. Consequently, various issues occurred in some CPU-intensive applications that depend on this feature. Now, this feature has been enabled and **SCHED_OTHER** threads get 5% of the CPU time on cores that are monopolized by **SCHED_FIFO/SCHED_RR** threads.

Note

Note that when upgrading from a previous installation of *rt-setup,*, **/etc/sysctl.conf** must be manually edited to remove the **kernel.sched_rt_realtime*** lines, which disable bandwidth limiting.

BZ#710503

The *ibm-prtm* package installs a configuration file in the **/etc/modprobe.d/** directory. Prior to this update, the **modprobe** utility displayed the following warning message:

```
WARNING: All config files need .conf: /etc/modprobe.d/ibm-amd, it will be
ignored in a future release.
```

With this update, the configuration file has been renamed to **ibm-amd.config**, thus fixing this bug.

Users of the real-time capabilities of Red Hat Enterprise MRG 2.1, which is layered on Red Hat Enterprise Linux 6, are advised to upgrade to these updated packages, which fix these bugs.

3.11. RHSA-2012:0333 – Important: kernel-rt security and bug fix update

Red Hat Enterprise MRG (Messaging, Realtime, and Grid) is a next-generation IT infrastructure for enterprise computing. MRG offers increased performance, reliability, interoperability, and faster computing for enterprise customers.

MRG Realtime provides the highest levels of predictability for consistent low-latency response times to meet the needs of time sensitive workloads. MRG Realtime provides new levels of determinism by optimizing lengthy kernel codepaths to ensure that they do not become bottlenecks. This allows for better prioritization of applications, resulting in consistent, predictable response times for high-priority applications.

Security Fixes

CVE-2011-4127, Important

SG_IO ioctl SCSI requests on partitions or LVM volumes could be passed to the underlying block device, allowing a privileged user to bypass restrictions and gain read and write access (and be able to issue other SCSI commands) to the entire block device.

CVE-2012-0044, Important

A local, unprivileged user could use an integer overflow flaw in **`drm_mode_dirtyfb_ioctl()`** to cause a denial of service or escalate their privileges.

CVE-2011-2918, Moderate

A local, unprivileged user could use a flaw in the Performance Events implementation to cause a denial of service.

CVE-2012-0038, CVE-2011-4077, Moderate

A local, unprivileged user could use flaws in the XFS file system implementation to cause a denial of service or escalate their privileges by mounting a specially-crafted disk.

CVE-2011-4097, Moderate

A local, unprivileged user could use a flaw in the Out of Memory (OOM) killer to monopolize memory, have their process skipped by the OOM killer, or cause other tasks to be terminated.

CVE-2011-4110, Moderate

A local, unprivileged user could use a flaw in the key management facility to cause a denial of service.

CVE-2011-4131, Moderate

A malicious Network File System version 4 (NFSv4) server could return a crafted reply to a GETACL request, causing a denial of service on the client.

CVE-2011-4132, Moderate

A local attacker could use a flaw in the Journaling Block Device (JBD) to crash the system by mounting a specially-crafted ext3 or ext4 disk.

CVE-2012-0207, Moderate

A flaw in **igmp_heard_query()** could allow an attacker, who is able to send certain IGMP (Internet Group Management Protocol) packets to a target system, to cause a denial of service.

CVE-2012-0810, Moderate

If lock contention during signal sending occurred when in a software interrupt handler that is using the per-CPU debug stack, the task could be scheduled out on the realtime kernel, possibly leading to debug stack corruption. A local, unprivileged user could use this flaw to cause a denial of service.

Red Hat would like to thank Chen Haogang for reporting CVE-2012-0044; Wang Xi for reporting CVE-2012-0038; Shubham Goyal for reporting CVE-2011-4097; Andy Adamson for reporting CVE-2011-4131; and Simon McVittie for reporting CVE-2012-0207.

Bug Fixes

BZ#784733

When a sleeping task, waiting on a futex (fast userspace mutex), tried to get the **spin_lock(hb->lock)** RT-mutex, if the owner of the futex released the lock, the sleeping task was put on a futex proxy lock. Consequently, the sleeping task was blocked on two locks and eventually terminated in the **BUG_ON()** function. With this update, the **WAKEUP_INPROGRESS** pseudo-lock has been added to be used as a proxy lock. This pseudo-lock tells the sleeping task that it is being woken up so that the task no longer tries to get the second lock. Now, the futex code works as expected and sleeping tasks no longer crash in the described scenario.

BZ#786145

When the **CONFIG_CRYPTO_FIPS** configuration option was disabled, some services such as sshd and ipsec, while working properly, returned warning messages regarding this missing option during start up. With this update, **CONFIG_CRYPTO_FIPS** has been enabled and no warning messages are now returned in the described scenario.

BZ#761420

Previously, when a read operation on a loop device failed, the data successfully read from the device was not cleared and could eventually leak. This bug has been fixed and all data are now properly cleared in the described scenario.

BZ#783570

Due to an assembler-sourced object, the **perf** utility (from the *perf-rt* package) for AMD64 and Intel 64 architectures contained an executable stack. This update adds the **.note.GNU-stack** section definition to the **bench/mem-memcpy-x86-64-asm.S** component of **perf**, with all flags disabled, and perf no longer contains an executable stack, thus fixing this bug.

Users are advised upgrade to these updated *kernel-rt* packages, which contain backported patches to correct these issues. The system must be rebooted for this update to take effect.

3.12. RHBA-2012:0044 – kernel-rt bug fix update

Red Hat Enterprise MRG (Messaging, Realtime, and Grid) is a next-generation IT infrastructure for enterprise computing. MRG offers increased performance, reliability, interoperability, and faster computing for enterprise customers.

MRG Realtime provides the highest levels of predictability for consistent low-latency response times to meet the needs of time sensitive workloads. MRG Realtime provides new levels of determinism by optimizing lengthy kernel codepaths to ensure that they do not become bottlenecks. This allows for better prioritization of applications, resulting in consistent, predictable response times for high-priority applications.

Bug Fixes

BZ#725485

The *kernel-rt* package has been upgraded to upstream version 3.0, which provides a number of bug fixes and enhancements over the previous version.

BZ#749575

Some applications use flawed versioning logic that cannot recognize new Linux kernel versions in the format of **3.x.y**. As a workaround to this bug in external applications, the new **uname26** utility has been added to MRG Realtime 2.1. This utility activates the 2.6 personality kernel patch to transform data returned by the **uname(2)** system call to the format of **2.6.40+[minor_release_number]**, and then executes the actual application.

BZ#708407

The **recvmmsg()** and **sendmmsg()** system calls were missing from the code and were previously unavailable. This update restores the code with the system calls.

BZ#663865

Previously, the **/proc/kcore** virtual file could be read beyond the ELF (Executable and Linkable Format) header file information and a malicious root user could access these additional parts of the file. With this update, **kcore** cannot be read beyond its ELF header, thus fixing this bug.

BZ#679263

The **%pK** printk format specifier was not added when printing the data from the **/proc/kallsyms** and **/proc/ modules** interfaces. This could cause kernel address leaks. With this update, **%pK** is properly used when returning data from the interfaces.

BZ#711488

Prior to this update, the *kernel* and *kernel-rt* packages delivered the same set of kernel manual pages. Consequently, file conflicts occurred when both *kernel-doc* and *kernel-rt-doc* were being installed. This update adds the **rt** suffix to the files with *kernel-rt-doc* manual pages and the file conflicts no longer occur.

BZ#725028

Previously, both the Red Hat Enterprise Linux kernel and the Red Hat Enterprise MRG Realtime kernel delivered the **/lib/firmware/WHENCE** file, which caused an installation conflict. With

this update, this file has been moved to a versioned directory in the Realtime kernel, thus fixing this bug.

BZ#717905

The **cred_alloc_blank()** function called the **abort_creds(new)** function with **new->security == NULL** and **new->magic == 0** if the **security_cred_alloc_blank()** function returned an error. As a result, the **BUG()** function was triggered if SELinux was enabled or if the **CONFIG_DEBUG_CREDENTIALS** property was active. With this update, **new->magic** is set before the **security_cred_alloc_blank()** function is called and **cred->security** with the NULL value in **creds_are_invalid()** and **selinux_cred_free()** functions are now handled gracefully.

BZ#679272

Certain kernel static data areas and kernel modules have writable or executable memory areas. Prior to this update, malicious software could overwrite the data and potentially execute code in these areas. With this update, the RO (Read-Only) and NX (No eXecute) bits have been added to the memory areas to prevent such actions.

Users of *kernel-rt* are advised to upgrade to these updated packages, which fix these bugs. The system must be rebooted for this update to take effect.

3.13. RHSA-2011:1253 – Important: kernel-rt security and bug fix update

Red Hat Enterprise MRG (Messaging, Realtime, and Grid) is a next-generation IT infrastructure for enterprise computing. MRG offers increased performance, reliability, interoperability, and faster computing for enterprise customers.

The kernel-rt packages provide the Red Hat Enterprise Linux Realtime kernel along with supporting infrastructure and documentation.

Security Fixes

CVE-2010-4526, CVE-2011-1770, Important

Flaw in the SCTP and DCCP implementations could allow a remote attacker to cause a denial of service.

CVE-2011-1494, CVE-2011-1495, Important

Flaws in Management Module Support for Message Passing Technology (MPT) based controllers could allow a local, unprivileged user to cause a denial of service, an information leak, or escalate their privileges.

CVE-2011-1745, CVE-2011-2022, CVE-2011-1746, Important

Flaws in the **AGPGART** driver and in **agp_allocate_memory()** could allow a local user to cause a denial of service or escalate their privileges.

CVE-2011-2491, CVE-2011-2695, Important

Flaw in the client-side NLM implementation, and in the way the maximum file offset was handled

for ext4 file systems, could allow a local, unprivileged user to cause a denial of service.

CVE-2011-2497, Important

Flaw in the Bluetooth implementation could allow a remote attacker to cause a denial of service or escalate their privileges.

CVE-2011-2517, Important

Flaws in the netlink-based wireless configuration interface could allow a local user, who has the **CAP_NET_ADMIN** capability, to cause a denial of service or escalate their privileges.

CVE-2010-4243, CVE-2011-1593, CVE-2011-1598, CVE-2011-1748, CVE-2011-2213, CVE-2011-1090, CVE-2011-2183, CVE-2011-2484, CVE-2011-2496, Moderate

Flaw in the **execve()**, **next_pidmap()**, **bcm_release()**, **raw_release()**, and **inet_diag_bc_audit()** functions; in the interaction between the Linux kernel's method for allocating NFSv4 ACL data and the method by which it was freed; in the memory merging support (KSM); in the taskstats subsystem; and the way mapping expansions were handled, could allow a local, unprivileged user to cause a denial of service.

CVE-2011-1020, Moderate

The **proc** file system implementation could allow a local, unprivileged user to obtain sensitive information, or possibly cause integrity issues.

CVE-2011-1021, Moderate

Local, privileged user could write arbitrary kernel memory via **/sys/kernel/debug/acpi/custom_method**. This update disables this file. The debugfs file system must be mounted to exploit this issue. It is not mounted by default.

CVE-2011-1478, CVE-2011-1767, CVE-2011-1768, Moderate

Flaws could possibly result in a denial of service when a packet is received.

CVE-2011-1479, Moderate

CVE-2010-4250 fix caused a regression, allowing a local, unprivileged user to cause a denial of service.

CVE-2011-1576, Moderate

napi_reuse_skb() could be called on VLAN packets, allowing attacker on the local network to possibly trigger a denial of service.

CVE-2011-1160, CVE-2011-2492, Low

Information leaks.

CVE-2011-1577, CVE-2011-1776, Low

Flaws in the EFI GUID Partition Table implementation could allow a local attacker to cause a denial of service.

CVE-2011-1585, Low

Local, unprivileged user could mount an existing CIFS share that requires authentication without knowing the password. This fix is a preventative measure: to exploit this flaw, **mount.cifs** must be setuid root, which it is not by default.

CVE-2011-2495, Low

/proc/<*PID*>/io could be read by local, unprivileged users without further restrictions, allowing them to gather confidential information.

Red Hat would like to thank Dan Rosenberg for reporting CVE-2011-1770, CVE-2011-1494, CVE-2011-1495, CVE-2011-2497, and CVE-2011-2213; Vasiliy Kulikov of Openwall for reporting CVE-2011-1745, CVE-2011-2022, CVE-2011-1746, CVE-2011-2484, and CVE-2011-2495; Vasily Averin for reporting CVE-2011-2491; Brad Spengler for reporting CVE-2010-4243; Robert Swiecki for reporting CVE-2011-1593 and CVE-2011-2496; Oliver Hartkopp for reporting CVE-2011-1748; Andrea Righi for reporting CVE-2011-2183; Kees Cook for reporting CVE-2011-1020; Ryan Sweat for reporting CVE-2011-1478 and CVE-2011-1576; Peter Huewe for reporting CVE-2011-1160; Marek Kroemeke and Filip Palian for reporting CVE-2011-2492; and Timo Warns for reporting CVE-2011-1577 and CVE-2011-1776.

Known Issue

BZ#736683

Due to a typographical mistake in the code, the kernel-rt installation attempts to call "tardlink" instead of "hardlink" and the installation returns multiple "tardlink: command not found" errors if the hardlink package is installed. However, installation still completes successfully.

Bug Fixes

BZ#710158

The user-space iptables tools required the CONFIG_NF_CT_ACCT option. This update decouples CONFIG_NF_CT_ACCT from other Netfilter options, disables this option in the configuration files, and therefore the option is no longer available.

BZ#728310

Firmware files for the cxgb3 driver were older than the driver version. This update adds the appropriate firmware files and prevents the warning in the logs.

BZ#728551

Previously, it was necessary to issue the **uname -v** command to acquire information about whether the running kernel is a Red Hat MRG Realtime kernel. This process could become costly and present a source of potential problems. With this update, the system creates a file in the /sys/ directory that is an in-memory file system when the kernel is Realtime and makes it possible to verify if the kernel has Realtime capabilities efficiently.

BZ#730834

If you run the command **echo 0 > /sys/kernel/kexec_crash_size** and kdump was not enabled, a kernel crash occurred and the system printed an oops message. This happened because the called **crash_shrink_memory()** function attempted to release the

crashk_res variable. However, the variable is reserved only if kdump is enabled. With this update, the **crashk_res** variable is released only when kdump is enabled and the problem no longer occurs.

BZ#727699

The push/pull algorithm for migrating Realtime tasks could have resulted in high lock contention and incorrect results. This update improves the algorithm with a faster sequential scan, which results in lower lock contention and better behavior under pressure.

BZ#727549

Non-Realtime kernels set the timer's current base to NULL when switching to a new base. Since this code is pre-emptible on Realtime, this code could loop forever in the **lock_timer_base()** function. This update avoids setting the current base to NULL on Realtime kernels.

BZ#719742

Some older be2net cards and firmware did not recognize certain commands and returned illegal or unsupported errors. Since newer drivers can handle this gracefully, this update removes the error messages.

BZ#681987

Previously the Realtime kernel could not use the crash tool because it had CONFIG_STRICT_DEVMEM set. This update enables the crash kernel module so that the crash tool works on Realtime kernels as expected.

Users are advised to upgrade to these updated kernel-rt packages, which contain backported patches to correct these issues. The system must be rebooted for this update to take effect.

3.14. RHEA-2011:0895 – Red Hat Enterprise MRG – Realtime 2.0 Release

Red Hat Enterprise MRG is a next-generation IT infrastructure incorporating Messaging, Realtime, and Grid functionality. It offers increased performance, reliability, interoperability, and faster computing for enterprise customers.

MRG Realtime provides the highest levels of predictability for consistent low-latency response times to meet the needs of time sensitive workloads. MRG Realtime provides new levels of determinism by optimizing lengthy kernel codepaths to ensure that they do not become bottlenecks. This allows for better prioritization of applications, resulting in consistent, predictable response times for high-priority applications.

The *kernel-rt* packages contain the Linux kernel, the core of any Linux operating system.

This update provides a build of the *kernel-rt* packages for Red Hat Enterprise MRG 2.0, which is layered on Red Hat Enterprise Linux 6.

Enhancements

BZ#666951

Due to changes in the **/sys** virtual file system layout, the Red Hat Enterprise Linux 6 user

space assumes different kernel behavior. To allow various scripts and **udev** for Red Hat Enterprise Linux 6 to work properly with the Red Hat Enterprise MRG Realtime kernel, the **CONFIG_SYSFS_DEPRECATED*** configuration options in the Red Hat Enterprise MRG Realtime kernel have been turned off.

BZ#667440

The startup scripts and services on Red Hat Enterprise Linux 6 assume new kernel behavior, which required several configuration setting changes in the Red Hat Enterprise MRG Realtime kernel. With this update, the **CGROUP** and **KSM** configuration options were turned on, and the **SYSFS_DEPRECATED_V2** option was turned off. This allows the Red Hat Enterprise MRG Realtime kernel to boot properly in the Red Hat Enterprise Linux 6 user space.

BZ#705582

When using the Red Hat Enterprise MRG Realtime kernel, application scaling was limited, because a single spinlock (**idr_lock**) was used to access the POSIX timer structures. This update converts the spinlock to the *read-copy-update* (RCU) protocol, which allows parallel access to the POSIX timer structures and enhances scalability.

Users of the Realtime capabilities of Red Hat Enterprise MRG 2.0, which is layered on Red Hat Enterprise Linux 6, are advised to upgrade to these updated packages, which add these enhancements. Note that the system must be rebooted for this update to take effect.

3.15. RHEA-2011:0894 – Red Hat Enterprise MRG – Realtime 2.0 Release

Red Hat Enterprise MRG is a next-generation IT infrastructure incorporating Messaging, Realtime, and Grid functionality. It offers increased performance, reliability, interoperability, and faster computing for enterprise customers.

MRG Realtime provides the highest levels of predictability for consistent low-latency response times to meet the needs of time sensitive workloads. MRG Realtime provides new levels of determinism by optimizing lengthy kernel codepaths to ensure that they do not become bottlenecks. This allows for better prioritization of applications, resulting in consistent, predictable response times for high-priority applications.

This update provides various enhancements to the following packages:

- The *ibm-prtm* package contains a utility and startup script for IBM BladeCenter systems. It enables an *Error Detection and Correction* (EDAC) driver, and also turns off the *System Management Interrupts* (SMI) generation, which improves system response time to events.
- The *python-linux-procfs* package enables the extraction of information from the **/proc** file system.
- The *python-schedutils* package provides an interface to the scheduler.
- The *rtcheck* package provides an application that tests the running system for Realtime capabilities. This program can be used by Realtime-enabled programs to determine if the system environment is suitable for them to run correctly.
- The *rtctl* package comprises a set of scripts, used to manipulate the scheduling priorities of groups of system threads.
- The *rteval* package contains a utility for measuring various aspects of Realtime behavior on systems under load. The script unpacks the **hackbench** and kernel source code from the *rteval-loads*

package, builds **hackbench**, and then goes into a loop, running **hackbench** and compiling a kernel tree. During that loop, the **cyclictest** program is run to measure event response time. After the run time completes, a statistical analysis of the event response times is done and printed to the screen.

- The *rteval-loads* package provides source code for system loads used by the *rteval* package.
- The *rt-setup* package configures settings required by the Red Hat Enterprise Linux Realtime environment, such as creating the Realtime group, adding Realtime user privileges to PAM (*Pluggable Authentication Models*), enabling the configuration of kdump in Realtime, and disabling **irqbalance** by default.
- The *rt-tests* package includes a set of programs that test and measure various components of Realtime kernel behavior. This package measures timer, signal, and hardware latency then tests the functioning of priority-inheritance mutexes.
- The *tuna* packages include graphical and command line interfaces for changing scheduler and *interrupt request* (IRQ) settings. Changes can be made to CPUs, by thread or at the IRQ level, taking into account the topology of multi-socket and multi-core systems. **Tuna** gives the ability to isolate CPU cores and sockets for use by a specific application or hardware device.
- The *python-numeric* package is a Python module that provides support for numerical operations for **Tuna**.

Enhancements

BZ#704325

This update provides a build of these packages for Red Hat Enterprise MRG 2.0, layered on Red Hat Enterprise Linux 6.

BZ#661896

The description of the *rtcheck* package has been updated to provide a clearer overview of the package.

BZ#666955

On Red Hat Enterprise Linux 6, the *kernel-rt-firmware* package is compliant with the default firmware loading script. To retain compatibility with this system, this update removes both the firmware loading script and the related **udev** rule from the *rt-setup* package.

BZ#676927

Previously, the Linux scheduler bindings provided by the *python-schedutils* package were unable to set the CPU affinity on systems with more than 64 cores. With this update, the package has been modified to dynamically allocate the required space, allowing the CPU affinity to be successfully set on large core systems.

Users of the Realtime capabilities of Red Hat Enterprise MRG 2.0 are advised to upgrade to these updated packages, which add these enhancements. Note that the system must be rebooted for this update to take effect.

Chapter 4. MRG Realtime on Red Hat Enterprise Linux 5

4.1. RHBA-2011:1370 – Red Hat Enterprise MRG Realtime 1.3 bug fix update

Red Hat Enterprise MRG (Messaging, Realtime, and Grid) is a next-generation IT infrastructure for enterprise computing. MRG offers increased performance, reliability, interoperability, and faster computing for enterprise customers.

MRG Grid provides high-throughput computing and enables enterprises to achieve higher peak computing capacity as well as improved infrastructure utilization by leveraging their existing technology to build high performance grids. MRG Grid provides a job-queueing mechanism, scheduling policy, and a priority scheme, as well as resource monitoring and resource management. Users submit their jobs to MRG Grid, where they are placed into a queue. MRG Grid then chooses when and where to run the jobs based upon a policy, carefully monitors their progress, and ultimately informs the user upon completion.

Bug Fixes

BZ#738787

When a socket buffer was reused, the **skb->skb_iif** operation resulted in an invalid value and could cause the kernel to terminate unexpectedly after two or three executions of the **ifup** or **ifdown** commands in a VLAN and bonding network environment. With this update, **skb->skb_iif** is now reset on reuse, thus allowing the **ifup** and **ifdown** commands to work correctly.

BZ#733976

During the **tx/mcc** polling, the **napi_complete()** function was incorrectly called before reaping the **tx** completions. Consequently, the **tx** completion processing was scheduled on another CPU concurrently, which sometimes resulted in a kernel panic. With this update, under load, **napi_complete()** is called after the **tx/mcc** completion processing but before re-enabling interrupts, thus fixing this bug.

BZ#711198

Previously, negative **si_code** values were not allowed. As a consequence, the **glibc aio** kernel API issued EPERM errors and kernel warning messages were returned. Now, only the problematic SI_TKILL code is disallowed, and it allows the **glibc aio** implementation to queue a signal with the SI_ASYNCIO si_code, thus fixing this bug.

BZ#743320

The Red Hat Enterprise MRG Realtime kernel did not provide the tcp_delack_min tunable to reduce network latency as described in the Realtime Tuning Guide. With this update, this tunable is now available in the Realtime kernel for customers to use as recommended in the documentation.

BZ#743639

Prior to this update, the unsupported kernel configuration options, KVM and VIRTIO, were sometimes used accidentally and caused unexplainable non-deterministic behavior. This bug has been fixed, and these options have been explicitly turned off in the Red Hat Enterprise MRG Realtime kernel.

All users of *kernel-rt* are advised to upgrade to these updated packages which fix these bugs.

Chapter 5. MRG Grid on Red Hat Enterprise Linux 6

5.1. RHSA-2013:0565 – Red Hat Enterprise MRG Grid 2.3 security, bug fix and enhancement update

The changes in this advisory are the same as those for the Red Hat Enterprise Linux 5 MRG Grid 2.3 Release. Refer to Section 6.1, " RHSA-2013:0564 – Red Hat Enterprise MRG Grid 2.3 security, bug fix and enhancement update ".

5.2. RHSA-2012:1281 – Moderate: Red Hat Enterprise MRG Grid 2.2 security, bug fix, and enhancement update

The changes in this advisory are the same as those for the Red Hat Enterprise Linux 5 MRG Grid 2.2 Release. Refer to Section 6.2, " RHSA-2012:1278 – Moderate: Red Hat Enterprise MRG Grid 2.2 security, bug fix, and enhancement update ".

5.3. RHSA-2012:0099 – Moderate: MRG Grid security, bug fix and enhancement update

Red Hat Enterprise MRG (Messaging, Realtime, and Grid) is a next-generation IT infrastructure for enterprise computing. MRG offers increased performance, reliability, interoperability, and faster computing for enterprise customers.

MRG Grid provides high-throughput computing and enables enterprises to achieve higher peak computing capacity as well as improved infrastructure utilization by leveraging their existing technology to build high performance grids. MRG Grid provides a job-queueing mechanism, scheduling policy, and a priority scheme, as well as resource monitoring and resource management. Users submit their jobs to MRG Grid, where they are placed into a queue. MRG Grid then chooses when and where to run the jobs based upon a policy, carefully monitors their progress, and ultimately informs the user upon completion.

Security Fix

CVE-2011-4930

Multiple format string flaws were found in Condor. An authenticated Condor service user could use these flaws to prevent other jobs from being scheduled and executed or crash the **condor_schedd** daemon.

Bug Fixes

BZ#765846

Previously, the job classad (classified advertisement) produced by the **cumin** management console was outdated and was still using the ***VMPARAM_Kvm_Disk*** parameter instead of the ***VMPARAM_vm_Disk*** parameter. Consequently, submitting a VM job from within **cumin** appeared to succeed in **cumin** but the actual job failed to start. Now, **cumin** has been fixed to use ***VMPARAM_vm_Disk*** when building the job classad and VM jobs submitted from within **cumin** start as expected.

BZ#751779

The Scheduler object uses a custom view which in turn uses a customized stat set that was

previously not subject to the new display format. As a consequence, the value displayed for up-time on the Scheduler object was using the old format which was not consistent with the **dd:hh:mm** label that is next to the stat name. Now, the rendering logic of the Scheduler view object has been updated and the value displayed for up-time is now consistent with the **dd:hh:mm** format indicated in the stat label.

BZ#782902

When **cumin** fetched more than 25 data entries from the **sesame** messaging management tool to merge them with the data of the **wallaby** configuration service, some of the **sesame** data disappeared for a given refresh interval. This happened when viewing the Inventory page with more than 25 systems that report **sesame** data. With this update, **cumin** gets all **sesame** records before merging them with the **wallaby** data. Now, **sesame** data no longer disappear from the Inventory page in the described scenario.

BZ#782485

Due to an unhandled exception on the Inventory page, when **wallaby** data was expected but not present, the exception dump was returned to the Inventory page. This bug has been fixed, the exception is now handled properly, and the Inventory page now displays without errors.

BZ#771642

Following a host link from the Inventory page opens a summary page for the host. The content of the Configuration tab depends on data encoded in the URL which could become stale. Consequently, selecting the Configuration tab sometimes caused an exception when the host data in the URL was stale. Now, data in the URL is checked for validity before the Configuration tab is generated. Also, the user is notified and redirected back to the Inventory page if stale data is detected, and the exception no longer occurs in the described scenario.

BZ#748735

Following a host link from the Inventory page opens a summary page for the host with the Overview tab selected. A host could be visible in the Inventory page even when no **sesame** data was available for that host, but the Overview tab requires **sesame** data. Consequently, an exception occurred on the Overview tab if the data from **sesame** was missing. Now, missing data from **sesame** is checked when generating the Overview tab, the user is notified with a yellow banner when no data is available, and the exception no longer occurs in the described scenario.

BZ#765713

Previously, the **#000,.1** RGBA value was used for a transparent black background, which appeared gray when the data was plotted. Consequently, the 1-day view of the percent memory chart under the Inventory tab showed a gray artifact for all **x** values that contained some data. With this update, the **#111,.1** RGBA value (transparent white) is used in the described scenario and the graph no longer shows the gray artifact.

BZ#761165

When the **condor_q -long** command was run on a EC2 (Amazon Elastic Compute Cloud) job, the ***EC2AvailabilityZone*** parameter was misspelled. This update fixes the spelling of ***EC2AvailabilityZone*** and this bug no longer occurs.

BZ#761671

When a series of queries was run against the collector while the collector was already under heavy load, the collector blocked waiting on the query operation. With this update, queries have been set to be non-blocking and no longer halt the collector in the described scenario.

BZ#757772

Previously, the scheduler did not properly restore submitter names with accounting group information on restart. Consequently, incorrect submitter names were associated with the wrong accounting group by the **Negotiator** daemon. With this update, additional logic to restore the submitter names with accounting group information, when accounting group is detected, has been added. Now, submitter classads are properly reconstituted on the scheduler restart.

BZ#765902

Previously, no binaries in Condor RPM packages were compiled with the **FORTIFY_SOURCE** flag. As a consequence, stack smashing protections were not enabled by the compiler. With this update, Condor has been recompiled with the **-D_FORTIFY_SOURCE=2** option, thus fixing this bug.

BZ#758212

When the **condor_hold** utility was run with a **%** sign in the hold message when writing an XML-formatted user log, the **schedd** daemon terminated unexpectedly. With this update, the **sprintf()** calls have been fixed to not use direct input strings and **schedd** no longer crashes in the described scenario.

BZ#613931

Addition of accounting groups to the **condor_userprio** utility reporting added redundant values to total sums. Consequently, totals reported by **condor_userprio** were too high. With this update, new logic that prevents accumulation of redundant values from accounting groups has been added to the code and **condor_userprio** no longer includes redundant values to its reports.

BZ#751072

When **condor** tools were run with the **-help** from the command line, they issued incorrect, non-zero return code. This bug has been fixed and **condor** tools now properly return **0** in the described scenario.

BZ#759154

When the **sshd.sh** script failed to execute properly, it returned **0** for all error conditions. This bug has been fixed and **sshd.sh** now displays the correct error codes when handling failure conditions.

BZ#759433

When a job was run in OpenMPI or parallel universe environment, the **condor_chirp** utility failed to write the output file. With this update, **condor_chirp** has been fixed to use absolute paths, rather than relative paths, thus fixing this bug.

BZ#750063

Previously, EC2 jobs executed via the **condor_q -run** command failed to display the **HOST** information. With this update, EC2 instance nodes have been added to the **HOST** string and the **condor_q** utility now displays the **HOST** information correctly in the described scenario.

BZ#752322

Previously, **condor_q** utility failed to display the summary information for suspended jobs. This bug has been fixed and **condor_q** now displays summary information for all job states.

BZ#761588

When the **condor_schedd** utility, managed by Red Hat High Availability, terminated unexpectedly and was restarted, the PID file contained the PID of the previous instance rather than the current one. Moreover, the PID file was not removed when **condor_schedd** was shut down. Now, the startup script has been improved to look for the PID file and remove it before the service starts, if necessary. Also, the stored PID now always corresponds to the currently running instance of **condor_schedd**.

BZ#751834

Prior to this update, setting a region in EC2 Enhanced was not supported. Consequently, using EC2 Enhanced with AMIs in a region other than the user's default region caused jobs fail to start. With this update, a new configuration option (**EC2Region = "<region>"**) has been added and running jobs on AMIs in regions such as **us-east-1** and **eu-west-1** now succeeds.

BZ#753829

Previously, the comparator for an internal collection that tracks active jobs in a submission was insufficient when used with DAG (Direct Acyclic Graph) workflow descriptions. Consequently, the DAG submissions were prematurely destroyed and recreated. As a result, when the QMF (Qpid Management Framework) plug-in of the **schedd** daemon was used for job publishing, DAG submission job state totals were reported incorrectly and did not properly accumulate as the DAG submission progressed through it's node job execution. This bug has been fixed and DAG submission job state totals now increase, decrease and accumulate consistently as viewed by a QMF client.

BZ#754202

When the history file configuration settings were set to very low values, the index file for the current history file could become orphaned. Subsequently, it's inode was quickly reused by the file system and incorrectly maintained in an active list by the **condor_job_server** and **aviary_query_server** utilities. With this update, correct index files are regenerated and the orphaned ones removed from the current set of history backup files *before* the inode reuse could occur. Now, index files are correctly cleaned up after a history file rotation and point to valid archived history backups based on inodes.

BZ#773680

Due to a regression in the **Scheduler::holdJobRaw()** function, the Condor server failed to change the job state of a job that was held and later released. Consequently, the job appeared to be idle while it was in fact running. With this update, the order of calls outside of the QMF and Aviary has been changed so that a released job's state is now correctly reported in the

described scenario.

BZ#760615

Previously, when the **DedicatedResource** option was added through the remote configuration feature, the **STARTD_ATTRS** attribute was overwritten instead of appended to. With this update, **STARTD_ATTRS** properly appends in the default database and adding **DedicatedResource** through remote configuration as described now works as expected.

BZ#760611

Previously, when the **DedicatedResource** option was applied to a group or a node through the remote configuration feature, the affected nodes failed to start the **condor_startd** daemon. With this update, a typo has been fixed in **DedicatedResource** and nodes configured with **DedicatedResource** through remote configuration now start **condor_startd** as expected.

BZ#750264

When the MRG Management Console was configured and then the remote configuration feature was used to configure the Aviary web service, the MRG Grid scheduler stopped reporting data to the Management Console. With this update, plug-ins outside the remote configuration feature are allowed to be configured. Now, installing remote configuration after manually configuring the Management Console does not result in the Console losing data from the MRG Scheduler.

BZ#756402

Previously, the SPQR library and the Wallaby service rejected authentication mechanisms other than **PLAIN**, **ANONYMOUS**, or **GSSAPI**. However, the underlying QMF engine library supported additional mechanisms, which were unavailable to SPQR developers or Wallaby users. With this update, the SPQR library and the Wallaby service have been changed to not reject any valid SASL mechanism. Now, any SASL mechanism that is available to QMF and a configured Qpid broker is available to SPQR and Wallaby.

BZ#756401

The SPQR library, used by the Wallaby service, is designed to automatically select an authentication mechanism based on certain selected options if no authentication mechanism is explicitly specified. However, if the user specified a username and the **ANONYMOUS** mechanism, SPQR would override the user's explicit mechanism selection and instead choose the **PLAIN** authentication. This update removes this confusing behavior and SPQR now prioritizes explicitly-specified mechanisms over the implicitly-selected ones.

Enhancements

BZ#745348

Previously, **condor_status** only allowed output sorting by simple attribute names. Users could not sort their classads using generalized expressions. The construction of the internal sorting expression has been modified so that it can refer to a generalized classad expression. Users can now provide general classad expressions to the ***-sort*** argument to sort the classads.

Users of the grid capabilities of Red Hat Enterprise MRG 2.1, which is layered on Red Hat Enterprise Linux 5, are advised to upgrade to these updated packages, which fix these bugs and add this enhancement.

5.4. RHBA-2012:0046 – Red Hat Enterprise MRG Grid 2.1 bug fix and enhancement update

Red Hat Enterprise MRG (Messaging, Realtime, and Grid) is a next-generation IT infrastructure for enterprise computing. MRG offers increased performance, reliability, interoperability, and faster computing for enterprise customers.

MRG Grid provides high-throughput computing and enables enterprises to achieve higher peak computing capacity as well as improved infrastructure utilization by leveraging their existing technology to build high performance grids. MRG Grid provides a job-queueing mechanism, scheduling policy, and a priority scheme, as well as resource monitoring and resource management. Users submit their jobs to MRG Grid, where they are placed into a queue. MRG Grid then chooses when and where to run the jobs based upon a policy, carefully monitors their progress, and ultimately informs the user upon completion.

Bug Fixes

BZ#741729

Previously, the Cumin web console provided no facility to modify the schema of an existing database on a software upgrade. Consequently, if an upgrade included schema changes, users were required to drop and recreate the database to incorporate the new schema version. This resulted in loss of data. With this update, a new "cumin-admin upgrade-schema" command has been added that allows users to apply schema changes as part of a software upgrade. This command runs necessary scripts to upgrade the schema to the newest version without any loss of data. If there are no schema changes to apply, the user is informed and no action is taken.

BZ#731065

Prior to this update, when a job returned output, an error message, or a log entry containing a file name without full path, the Cumin web console failed to retrieve the file contents under the Output tab for a job. With this update, Cumin attempts to append the working directory from the job ad to the names of output files if the file does not contain a full path. When Cumin submits a job, it prepends the working directory to the output file names before submission. Now, Cumin handles the output file names correctly in most cases. However, if a job is submitted through another mechanism, such as the condor_submit utility, the output files are not given full paths, and the Aviary web service is used by Cumin for data retrieval, Cumin will still not be able to retrieve the output files.

BZ#725038

Previously, messages printed in yellow task status banners in the Cumin web console could potentially contain characters that break XML parsing in a browser during display. If such a message was printed, the browser displayed an error message, no Cumin content was visible, and Cumin had to be restarted to restore the user interface. With this update, code has been added to properly escape special characters in the banner messages before display, thus preventing this bug.

BZ#718093

Previously, the cumin package had a dependency on the dejavu-lgc-sans-fonts package, which is only available through an optional Red Hat Network channel. As a consequence, the Cumin web console could not be installed on Red Hat Enterprise Linux 6 using only the non-optional Red Hat Network channels. With this update, the dependency has been changed to the liberation-sans-fonts package and Cumin now installs correctly in the described scenario.

BZ#733447

Native httplib classes in Python provided support only for client certificate validation; server certificate validation was not supported. Consequently, the Cumin web console could not communicate securely over SSL with Aviary servers. With this update, Cumin has been enhanced to use either the Python SSL module or the M2Crypto SSL toolkit to supply server certificate validation. The cumin-web utility now writes log entries beginning with "AviaryOperations:", which indicates what type of communication is being used for Aviary.

BZ#723848

Previously, string values needed to be explicitly double-quoted by the user when editing a value on the form. When job attributes had been edited via the "Edit Attributes" form, Cumin did not distinguish string values from expression values automatically. Consequently, the value was interpreted by the Condor workload management system as an expression, which sometimes caused errors and prevented the attribute value from being changed. With this update, Cumin uses type information available from Condor to distinguish between strings and expressions and users are no longer required to explicitly quote string values when editing attributes.

BZ#639383

Previously, Cumin based the age of a submission around the creation times of the QMF object that represented the submission in the MRG Messaging space. However, displays such as the Longest Running Grid Submissions table in the default persona were affected by events in the MRG Messaging space, and could therefore be inaccurate. This update ensures that the data generated by condor and integrated into Cumin present the earliest queue date of any job included in a submission, with the result that the Longest Running Grid Submissions display is accurate. In addition, a new column which shows the queue date has been added to the table.

BZ#690297

During signal escalation it was possible for a starter to leak due to disjointed communications. This in turn could cause a claim to be unnecessarily relinquished due to out-of-order processing, such as if a shadow exited before the starter had been cleaned. This update forces the shadow to wait for the final update from the starter, which results in a claim reuse performance increase and increased job throughput during period_evaluation and signal escalation.

BZ#699726

The schedd daemon could spawn more shadows than it could handle given its current default configuration, which would in turn cause shadows to time out when trying to communicate with the schedd. With this update, the MAX_ACCEPTS_PER_CYCLE value has been increased, with the result that zero timeouts occurred under scale testing.

BZ#748339

Previously, when the Windows Java detection issued an exception, no Windows clients were visible via the "condor_status -java" command. Now, the default configuration for the

JAVA_SEPARATOR module has been updated and Windows clients correctly detect Java details in the described scenario.

BZ#734871

Due to a bug in the state transition for overlapping Cron jobs, Cron object states were not marked as idle after forced termination. With this update, the state handling logic has been updated and the state transitions now work as expected.

BZ#734213

Previously, calls to the **`param_without_default()`** function that did not free returned memory resulted in memory leaks in the **`Negotiator`** grid daemon. With this update, the calls have been replaced with new, memory-safe **`param_defined()`** function, and the memory leaks no longer occur in the described scenario.

BZ#580270

If a High Availability scheduler was running and the NFS service was disrupted, then the schedd lock disappeared, which resulted in two schedd processes running on two different nodes, but accessing the same job queue. Red Hat High Availability is now used to manage the job queue in an HA scheduler setup, with the result that a single schedd process runs on the nodes which comprise the HA scheduler setup.

BZ#748825

Load average detection is specific for each kernel major version and only supports kernels of version 1 or 2. Consequently, the load average would appear to be "-1" under a kernel 3.x. This update adds support for load average detection under kernels 3.x, implemented identically as for previous kernel versions, with the /proc/loadavg utility unchanged.

Note

Note that this bug is specific for Red Hat Enterprise Linux 6. Refer to Section 6.4, “ RHBA-2012:0045 – Red Hat Enterprise MRG Grid 2.1 bug fix and enhancement update ” to learn about updates specific for Red Hat Enterprise Linux 5.

BZ#732821

Previously, the **`Negotiator`** daemon performed more **`fsync()`** operations than required for maintaining transactions. Consequently, operations involving large numbers of submitters resulted in a performance hit. With this update, the use of **`fsync()`** calls has been minimized in transactions and performance of **`Negotiator`** on pools with large numbers of submitter records improved significantly.

BZ#712026

When the startd daemon was gracefully stopped with some deferred jobs landed but not yet run, the jobs appeared to be running in the schedd daemon even though no starter was in operation. This update forces the starter to send job exit notification for deferred jobs and the jobs are now properly rescheduled and transitioned to the idle state, thus fixing this bug.

BZ#732797

Code analysis revealed sub-optimal configuration in the dedicated scheduler. Consequence of this was slower than expected performance of the scheduler. Now, the fsync algorithm for the dedicated scheduler has been updated and the performance of the scheduler increased.

BZ#714724

Prior to this update, the **Negotiator** callback functions for the condor_userprio utility were limited to read only the first 63 characters of submitter names from commands. As a consequence, any attempt to invoke a condor_userprio command with a submitter name longer than 63 characters resulted in the name being truncated internally, causing the command to fail. Now, the callback functions have been updated to properly handle names of arbitrary length, thus fixing this bug.

BZ#739203

Previously, when multiple instances of EC2 GAHP (Elastic Compute Cloud Grid ASCII Helper Protocol) clients were started, both instances leaked their file descriptor. This bug has been fixed and the ec2_gahp utility now scales properly.

BZ#675697

The check that determined whether a submitter record could safely be deleted from the **Negotiator** contained a logic error. The logic checked for priority factor equal to default priority factor **DEFAULT_PRIO_FACTOR**, which allowed record deletion if the submitter's factor happened to be set to the default. The deletion-checking logic has been corrected in these updated packages so that any explicitly-set value for priority factor, default or otherwise, is now detected. As a result, submitter records are no longer improperly deleted if the user-set priority factor happens to be equal to the **DEFAULT_PRIO_FACTOR** value.

BZ#738338

When an executable in a job was unable to be executed, the starter shut down immediately without waiting for the exit job hook to complete its operation. With this update, the starter waits 30 seconds by default before giving up on the exit hook and shutting down. A job that fails to execute will now try to make sure a configured exit hook will complete its operation before exiting.

BZ#565501

The condor_status utility used a signed 32-bit integer to represent total disk space and memory available in a group of machines. This integer could have overflowed with a large pool, resulting in negative or nonsensical results when the "condor_status -total" command was run. This update changes the data type to an unsigned 64-bit integer value, which is significantly less likely to overflow.

BZ#703992

Even when job spool directories were disabled, the way that the schedd daemon attempts to clean them had a significant performance impact in certain environments, including those using a shared file system. This update improves the cleanup code so that schedd interacts only minimally in cases where no job spool directories exist, which reduces file system overhead when job spool directories are not enabled.

BZ#715293

When the condor_userprio utility detected failure to locate the **Negotiator** daemon, no error message was returned to inform users. With this update, an informative error message has been added to the failure detection code path and users are now properly informed in the described scenario.

BZ#725746

Due to inaccurate help messages, users were led to believe the condor_router_rm utility can be passed both the user name and the job ID parameters at the same time, which is not possible. This update fixes the help messages to provide accurate information on condor_router_rm usage.

BZ#725758

When the condor_router_q utility was running on job routes whose name contained space characters, condor_router_q returned incorrect output and grouping. With this update, the awk parsing for the script that displays names has been fixed and output for routes whose names contain spaces is now displayed correctly.

BZ#690494

The **-long** and **-format** options to the condor_q command were order-dependent due to a parsing error. This update removes the parsing error, and the order of the aforementioned options no longer results in different behavior.

BZ#712111

When a negative or floating-point number was entered into a **deferral_*** parameter during submission, jobs were placed on hold as a result. With this update, these parameters are checked to contain a positive integer value before submission and now, only jobs with valid **deferral_*** parameters are allowed to run.

BZ#738719

Previously, a statically allocated fault string was incorrectly freed in a WSO2 code path. When a wrong endpoint URL was invoked in the Aviary web service, the WSO2/Axis2C engine used by Aviary failed to load an implementation library and terminated the process unexpectedly. With this update, only a dynamically allocated string is freed in the same code path, the process containing Aviary no longer crashes in the described scenario, and Aviary clients now receive SOAP errors for wrong endpoint invocation.

BZ#694612

Previously, Aviary could only be operated within a secure local network in order to secure communication between Aviary clients and servers. This update includes upstream code enhancements that integrate OpenSSL more fully into the Axis2C SOAP (Simple Object Access Protocol) engine used by the Aviary server implementation. Aviary clients and servers can now exchange x509 certificates to authenticate each side and establish a secure link.

BZ#705016

The Aviary query server could have eventually terminate unexpectedly when the ATTR_SUBMISSION_NAME attribute on a job was modified through the Aviary API after it had been added to the job queue, then removed, followed by an invocation of getJobDetails. This

update corrects the code so that ATTR_SUBMISSION_NAME is prevented from being modified after the job has been submitted, which prevents the possibility of an eventual Aviary query server crash.

BZ#731463

Previously, the getJobDetails implementation was returning integer attributes without evaluation. Consequently, calling the getJobDetails operation returned the CurrentTime attribute as "time()" string literal. This update fixes the getJobDetails implementation to return integer attributes with evaluation and the CurrentTime attribute now correctly appears as evaluated epoch integer.

BZ#733055

When the Aviary getJobSummary operation was invoked to return a job summary including the last_update field on Red Hat Enterprise Linux 64-bit implementation, the last_update field contained an invalid value. With this update, the signature and type conversions in the date encoder implementation have been modified and the last_update field now always contains a correctly formatted value.

BZ#702060

The wso2-wsf-cpp package unnecessarily depended on the wso2-rampart package. This update removes this dependency so that the wso2-rampart package is no longer needed for Aviary functionality.

Note that the wso2-rampart package can be safely removed from the system.

BZ#702489

If a keyword was provided as an attribute name to an Aviary submitJob or setJobAttribute operation, the job submission was either incorrectly parsed or failed. With this update, submitting a job or setting an attribute that includes a keyword results in the operation failing with an error message indicating that a forbidden keyword caused the fault.

BZ#749023

The previous version of the Aviary web service interface generated many per-PID log files from the Axis2/C engine, causing unnecessary proliferation of files in the log directory. Now, Aviary logging setup uses a single log file each for the Aviary schedd plug-in and the query server that is continually appended to across-process restarts, and Aviary Axis2/C log file count is dramatically reduced.

BZ#739205

When attempting to retrieve the contents of a file using the Aviary **`getJobData()`** function, a permission error occurred and the user was unable to finish the operation through the Aviary API even though it was possible to do so via QMF (Qpid Management Framework). With this update, the **`getJobData()`** code has been modified to adjust file permissions earlier, and users can now retrieve files both through QMF and Aviary interfaces.

BZ#699737

The condor_job_server utility maintains index files to history files. Although the history files were properly garbage-collected, the index files were not, which could have resulted in an error when trying to access details of a job that only existed in an index file. This update ensures that

the index files are garbage collected along with the history files, thus preventing this error from occurring.

BZ#733341

In order to use Red Hat High Availability to manage the HAScheduler feature setup, manual modifications to the HAScheduler feature in the base-db were required. With this update, a new feature named "BaseHAScheduler" has been added to be base-db and no manual modifications to features in the base-db are now necessary in the described scenario.

BZ#733481

Previously, the configuration scheme of secure communications for the Aviary web service and the query server using remote configuration required users to add new parameters and features to the database manually. With this update, the SSLEnabledAviaryScheduler and SSLEnabledQueryServer features have been added to the base-db and configuration for secure communication in Aviary, and query server through remote configuration is now easily accomplished.

BZ#724907

The subsystem used by low-latency scheduling in MRG Grid's configuration changed from LOW-LATENCY to CAROD but the configuration in the base-db used by remote configuration still used the old subsystem. Now, the base-db has been updated to use the new subsystem and low-latency scheduling with remote configuration creates configurations with the new subsystem.

BZ#733368

Previously, the Wallaby configuration service shell commands did not provide full functionality when configuring the wallaby utility. Consequently, options not provided by the shell commands had to be configured via the condor_configure_store and condor_configure_pool utilities. This update adds many new wallaby shell commands and the Wallaby database configuration can now be performed through the wallaby shell.

BZ#726108

The remote configuration tools incorrectly split entity names on any commas in the name, and proceeded to treat those names as separate entities. This update corrects the parser so that it allows for names which include commas, with the result that the remote configuration tools now properly identify a name which contains a comma as long as the name is surrounded by quotes escaped with backslash characters, i.e. \".

BZ#703593

Manual pages for the condor_configure_store and condor_configure_pool commands have been added to the condor-wallaby-tools package.

BZ#750315

Previously, the sesame system agent authenticated to the qpidd broker as a guest user by default. In Red Hat Enterprise Linux 6, the guest SASL account is not present out-of-the-box in an installation of the "MRG Messaging" group. Consequently, sesame could not to connect to the broker without additional configuration. With this update, sesame uses anonymous authentication by default. Anonymous authentication is enabled by the qpidd broker by default

on both Red Hat Enterprise Linux 5 and 6. Now, sesame is able to connect to the qpidd broker without additional configuration.

Enhancements

BZ#707771

With this update, two new options, "-grouporder" and "-grouprollup", have been added to the condor_userprio utility to allow accounting group hierarchies to be displayed and group statistics to be rolled up by hierarchy, respectively.

BZ#705365

This update adds several example scripts to the condor-aviary package for job submission, control, attribute set, queries for status, summary, and details. A directed acyclic graph (DAG) submission example has also been added. Users are now provided with a practical set of client examples for the Aviary web service interface written in Python and using the Suds client.

BZ#668039

The **unparse()** function has been added to classads, which enables queries to be constrained using the value of a classad attribute (pre-evaluation).

BZ#679139

The new remote configuration base database packages contain fixes and updates that must be loaded into wallaby in order to take effect. However, loading a database replaces the contents of wallaby, which could have caused user changes to be removed. These updated packages include a wallaby shell command that allows the database to be upgraded so that new versions of the remote configuration base database package can be applied to an existing system without removing a customized configuration.

BZ#635012

First class support has been added for the condor_suspend and condor_continue commands, which allows jobs to be suspended and continued based on userid or jobid, and without releasing the claim. This ability makes it easier to determine the shared resource impact of specific user or job IDs.

BZ#610265

A new attribute, "PreserveRelativeExecutable", has been added to job ads. The value of this attribute defaults to False. When set to True, and given that TransferExecutable is set to False, then an executable with a relative path name is left relative, and no paths are prepended to it. This functionality allows users to easily specify in a job submission that their executable remains relative until it is resolved, which allows executable locations to be resolved by the user's PATH when using a job wrapper.

BZ#733379

With this update, the Wallaby configuration service has been enhanced to use an alternate technique for storing versioned pool configuration. This technique significantly speeds up activation operations on most pools with complex configuration.

BZ#725052

RSA public key file was configured in an EC2 Enhanced route, no error message was given about the invalid public key, the job was routed to EC2 but was not able to run. With this update, an error message is now returned, and the job is not routed to EC2 in the described scenario.

BZ#719052

When a remote configuration tool referred to a store's default group by the name "Internal Default Group", attempts to use this name as a target to act on the store's default group failed. With this update, the remote configuration tools recognize "Internal Default Group" as a special target that corresponds to the store's default group.

BZ#694861

The condor_preen utility now logs information to TOOL_LOG to enable easier problem diagnosis.

Users of the grid capabilities of Red Hat Enterprise MRG 2.1, which is layered on Red Hat Enterprise Linux 6, are advised to upgrade to these updated packages, which fix these bugs and add these enhancements.

5.5. RHSA-2011:1250 – Moderate: Red Hat Enterprise MRG Grid 2.0 security, bug fix and enhancement update

The changes in this advisory are the same as those for the Red Hat Enterprise Linux 5 MRG Messaging 2.0 Release. Refer to Section 6.5, “ RHSA-2011:1249 – Moderate: Red Hat Enterprise MRG Grid 2.0 security, bug fix and enhancement update ”.

5.6. RHEA-2011:0891 – Red Hat Enterprise MRG – Grid 2.0 Release

The changes in this advisory are the same as those for the Red Hat Enterprise Linux 5 MRG Grid 2.0 Release. Refer to Section 6.6, “ RHEA-2011:0889 – Red Hat Enterprise MRG – Grid 2.0 Release ”.

Chapter 6. MRG Grid on Red Hat Enterprise Linux 5

6.1. RHSA-2013:0564 – Red Hat Enterprise MRG Grid 2.3 security, bug fix and enhancement update

Red Hat Enterprise MRG (Messaging, Realtime, and Grid) is a next-generation IT infrastructure for enterprise computing. MRG offers increased performance, reliability, interoperability, and faster computing for enterprise customers.

MRG Grid provides high-throughput computing and enables enterprises to achieve higher peak computing capacity as well as improved infrastructure utilization by leveraging their existing technology to build high performance grids. MRG Grid provides a job-queueing mechanism, scheduling policy, and a priority scheme, as well as resource monitoring and resource management. Users submit their jobs to MRG Grid, where they are placed into a queue. MRG Grid then chooses when and where to run the jobs based upon a policy, carefully monitors their progress, and ultimately informs the user upon completion.

Descriptions of the security fixes provided in this advisory can be viewed at https://rhn.redhat.com/errata/RHSA-2013-0564.html. The changes in this advisory for other non-security bug fixes and enhancements are documented below:

condor

BZ#850392

To allow for accurate matching and reuse of partitionable slots, Condor's default settings for job submission and execute resources have been updated. These enhancements enable more accurate memory tracking of jobs and higher resource utilization.

BZ#862550

Previously, when users tried to suspend local universe jobs, the condor_schedd daemon would crash. This issue has been fixed with the ability to ignore requests and continue scheduler and local universe jobs. As a result, the condor_schedd daemon continues to work normally and reports an error to the user when they try to suspend a local universe job.

BZ#525237

This enhancement includes support for tracking process tree resource usage accounting via Control Groups (or cgroups) in the cpuacct, memory, and blkio controllers, as well as support for limiting memory usage via Control Groups.

BZ#772009

Previously, changing the order of RequestMemory lines inside a requirements expression resulted in a different requirements expression. This fix ensures Condor correctly parses the RequestMemory lines regardless of order, and as a result, the output is the same as the original requirements expression.

BZ#772587

Previously, when running an openmpi job with Condor's openmpiscript, STDERR would show the script as using deprecated parameters. With this fix, the openmpiscript no longer uses the deprecated parameters and STDERR remains empty when the system runs normally.

BZ#782132

Previously, openmpi jobs that were run with Condor openmpiscript on Red Hat Enterprise Linux 6.2 would fail attempting to load shared libraries. This fix passes the --prefix of the install location of the openmpi to mpirun, resulting in jobs that complete without error.

BZ#784105

Previously, when debugging and analyzing the job_queue.log, rollovers would not denote time reference. This fix adds a timestamp to the rollover log name. As a result, users can obtain time and date information from old job_queue.log files.

BZ#825365

Previously, a missing flag reset and inefficient key checking in the accountant initialization and accountant table loops allowed accountant table sanity checking only after negotiator reconfiguration. This was redundant and made sanity checking inefficient. This fix inserts a "once-only" flag reset during initial startup sanity checking, and improves the efficiency of key checks for iterations over the accountant table. As a result, sanity checking of the accountant table is faster, and only takes place on negotiator startup.

BZ#809732

Previously, when a parallel universe job is removed its claim can be reused, causing the job status to become invalid. This fix updates the accuracy of job status reporting and prevents this error.

BZ#803897

Previously, users were unable to configure job preemption policies that are aware of whether a job negotiated via autoregroup, and also what particular group name a job negotiated under. This happened because negotiating job and resource ads did not provide sufficient information to allow group-aware or negotiation-aware preemption policies. This update enhances the negotiator, schedd and startd claiming processes with new fields that include RemoteAutoregroup, RemoteNegotiatingGroup and RemoteGroup, and their Submitter counterparts. As a result, new job and resource attributes are available for use in defining PREEMPTION_REQUIREMENTS expressions that are group-aware and negotiation-aware.

BZ#748053

Previously, group quota limit bookkeeping did not take the possibility of preemption into account, which resulted in job preemption was prevented when group quotas were in effect. This fix extends bookkeeping logic for submitter limits to include possibility of preemption. Job classads in the negotiator were enhanced to allow PREEMPTION_REQUIREMENTS to include accounting group names in the expression. As a result, preemption is now allowed when group quotas are in effect and preemption policies can be configured to allow preemption to respect accounting group boundaries, or ignore them if desired.

BZ#805448

Previously enhanced logic for computing submitter limits did not take the NEGOTIATOR_CONSIDER_PREEMPTION setting into account. Setting NEGOTIATOR_CONSIDER_PREEMPTION to false caused submitter limits to be not as tight as possible, which resulted in inefficiency. This fix updates the logic to take consider-preemption settings into account, resulting in tighter submitter limits when the consider-preemption setting is off.

BZ#740774

Previously, submitting a job with RequestMemory equal to a floating point number caused the job not to run. This fix updates classads to cast the evaluation of RequestMemory to be equal to an integer. As a result, jobs run successfully.

BZ#805581

Previously, log messages for negotiation rejections were misleading and did not properly reflect the negotiation logic in the code. 'Group quota exceeded' was output as a rejection message instead of the correct 'submitter limit exceeded'. This was also output when the submitter limit was allowed to be exceeded. The resulting message was corrected to 'submitter limit exceeded' and logic was updated to not output this message when limits were actually allowed to be exceeded. As a result, the log messages are accurate.

BZ#785289

Previously it was not possible to configure an autoregrouping accounting group policy in the initial implementation of Hierarchical Group Quotas (HGQ). This enhancement changes the semantics of GROUP_AUTOREGROUP and GROUP_AUTOREGROUP_<name> to emulate the behavior of autoregroup prior to the introduction of HGQ. As a result, GROUP_AUTOREGROUP can now be used to configure a legacy-style autoregrouping policy.

BZ#785283

The previous hard-coded ordering of accounting group negotiation prevented users from obtaining the job matchmaking behavior required. This feature allows for configurable group negotiation order, which enables users to use starvation ordering, which allows groups with unfilled quota to negotiate first, as well as enforcing that the root "<none>" group negotiate last, to properly support the autoregroup feature. Additionally, this feature allows a Grid administrator to define custom ordering policies to meet customer-specific needs.

BZ#833095

Previously, local resource allocations were not internally copied to slot-total data structure during dynamic slot instantiation. This caused values for local resource attribute TotalSlotXxx to be reported as zeros on the classad. This fix provides proper data structure copy internally, and as a result, values for TotalSlotXxx are now reported consistently with standard TotalSlotCpus, TotalSlotMemory, and similar.

BZ#783267

When debugging a job while it was running on a target machine, users could not directly attach a debugger to a running executable. This fix adds support to allow users to SSH to the job sandbox directory of the running job. As a result, users can now SSH to their jobs to debug them on the target machine.

BZ#845567

This enhancement adds the PRE_SKIP keyword to the DAG manager. If a PRE script exits with the PRE_SKIP value in a DAGMan job, the node succeeds and the job and the POST script are both skipped.

BZ#755765

Previously, the dprintf configuration would read MAX_<SUBSYS>_LOG as an integer literal to bypass the classad expression limitation of 32-bit integers. Because of this, configuring MAX_<SUBSYS>_LOG with an expression, such as "1024*1024", caused a call to EXCEPT(), which was intercepted by parent condor_master process and made failure diagnosis difficult. This fix replaces the call to EXCEPT() with a call to _condor_dprintf_exit(), which leaves an informative message in the dprintf_failure.<SUBSYS> file in the log directory. As a result, configuration failures for MAX_<SUBSYS>_LOG are now more informative.

BZ#850555

This enhancement adds the -expand option to condor_config_val. When both -dump and -expand options are specified to 'condor_config_val', all configuration variables are expanded before they are printed out.

BZ#850838

This enhancement allows the DAG manager to copy PRIORITY values from the DAG input file to the JobPrio attribute in the job ClassAd. Furthermore, the PRIORITY values are propagated to child nodes and SUBDAGs.

BZ#486480

Previously, when the condor master daemon attempted to send an obituary during a log rollover event of the failed daemon, the obituary was not sent. The logic has been updated so that the master will send an obituary email when a daemon fails during a log rollover.

BZ#766612

Previously, when removing the binary on a node running the condor_schedd daemon in a High Availability configuration with Red Hat High Availability, the condor_schedd would not stop. This is caused when the binary was missing and the service would not failover. This is caused by the init script used to control the schedd, which does not check for the existence of the binary before stopping. As a result, a stop operation will be attempted even if the schedd binary isn't on the system. With this fix, the stop function works correctly.

BZ#850567

This enhancement improves the output of condor_userprio by adding the group_quota and group_priority info to condor_userprio -all.

BZ#767613

Previously, it was difficult to determine the number of active query workers during any one interval due to diagnosing collector performance issues. As a result, it was difficult to determine the number of active query workers during any one interval and to log the number of query workers once a query has been activated. This fix ensures every query will log the number of active query workers.

BZ#788452

Previously, when upgrading a windows condor install from 7.6.3-0.3 to 7.6.5-0.11, Condor's Java detection was broken from the previous install. This fix ensures that during upgrade, the installer will correct the previous version's configuration file. As a result, Condor's Java detection works as expected.

BZ#809551

Previously, users were unable to use a specific key in Condor which has already been generated via ec2-tools. This fix allows the user to specify a key by name.

BZ#502675

Previously, when running a vm-job, the vm-load was not accounted for in the loadavg calculations in the startd. This fix allows the user to obtain the cpu utilization from the vm-job and use it in the loadavg calculation. As a result, the loadavg calculation takes vm-load into account.

BZ#751013

Previously, when continuing a job after it had been suspended, the job would receive SIGCONT twice. This fix updates the logic to ensure that SIGCONT can only be sent once. As a result, the job which continues after it has been suspended will only receive SIGCONT once.

BZ#756096

Previously, defaults caused machines to wake up unnecessarily, and all sleeping machines were woken up after each ROOSTER_INTERVAL. This fix updates the default to wake up machines only if they have been matched within the last half hour. As a result, machines are only woken up if they have a recent match.

BZ#773434

Previously, commands run with the help switch **-h** returned non-zero on exit. This fix allows commands to return zero when the **-h** switch is present.

BZ#846955

Repeated calls to 'service condor restart' resulted in the error: /var/run/condor/condor_master.pid: No such file or directory. This fix updates the script process order so there is no error on restart.

BZ#860308

Previously, statically configuring /etc/hosts with ipv6 entries caused the condor_schedd daemon to crash trying to forward resolve entries. This fix ensures ipv6 addresses are properly handled in /etc/hosts and static configurations. As a result, Condor starts normally.

BZ#864637

Previously, issuing a condor_restart -subsystem condor_had daemon for a HACM node running the negotiator caused the condor_had and negotiator to stop. The had daemon would not restart. This fix ensures the had daemon restarts when this occurs.

BZ#782553

Previously, users could not perform queries against various resources in Condor ec2 jobs. This fix adds support for resource tagging. As a result, during job submission users can now tag their resources and run queries against those tags.

BZ#782552

Previously, performing an idempotent operation more than once yielded the same result as applying it just once. This fix adds support inside of the ec2 protocol when calling RunInstances. As a result, ec2_gahp now supports idempotent RunInstances.

condor-aviary

BZ#807398

Previously, when the Aviary Schedd plug-in and Query Server are deployed in a HA group, Aviary clients experienced stale endpoint references for a longer duration than necessary. Adjustments in the Locator implementation to quickly replace a failed endpoint reference with its new one. As a result, an Aviary client using a Schedd or Query Server endpoint can retrieve the new endpoint faster.

BZ#733515

Previously, Cumin needed a way to locate Aviary SOAP endpoints. Cumin was unable to find Aviary SOAP endpoints that may have been activated with ephemeral ports. WSDL, XSD and code were designed and developed to provide a SOAP interface to locate other SOAP endpoints. As a result, Cumin can retrieve Aviary SOAP endpoints through a well-known point of contact.

BZ#768328

Previously, Aviary XML schema for the various job states: there are no states listed for suspended and transferring_output jobs in the JobStatusType. The Aviary XML schema was updated to add these new job states, and implementation changes were made to reflect these totals in the API as well. As a result, Aviary clients can view the totals for suspended and transferring_output job states from this XML schema type.

BZ#864560

Previously, when a user submitted a job to Aviary using condor_submit with a submission name that is not bounded by quotes, an extra, bogus submission would be recorded when there should be only one. This fix includes an internal change to perform comprehensive checks for the presence of a Submission attribute within the job ad. As a result, a quoted or unquoted Submission attribute in a condor_submit is accurately reflected by Aviary as part of a single distinct submission record.

BZ#856646

Submission IDs were previously retrieved with a BEFORE mode, which led to duplicate submission IDs being returned. The problems that prevented the getSubmissionID implementation from processing the specified time range query parameters have now been fixed, so unique submission IDs are returned to the client.

BZ#768319

Invoking the Aviary query operation getSubmissionSummary on a submission with suspended jobs previously prevented totals for those suspended jobs being listed in the summary. The Aviary XSD has now been updated and implementation code has been added so that all submission totals are now listed in the summary.

BZ#800079

This release of Grid introduces a new capability in the Aviary package for collecting submission ids from a Query Server using a page size and offset. This feature was designed for two purposes: 1) help Aviary clients such as Cumin to manage large submission sets at increasing scale 2) enable an Aviary client to query for new submissions from a point-in-time (i.e., from the latest submission it knows about) In QMF, new submission objects become available in the QMF object space as they are created. Aviary and its use of SOAP requires a pull model to gather information about these new submissions. Also, if Cumin is brought online in an existing Grid deployment but with an empty database for submissions, it may need a way to incrementally load older submissions using a background thread. Once a submission ID is collected, the details of that submission can be retrieved by the Aviary getSubmissionSummary operation. The BEFORE/AFTER mode is optional, so results will instead be returned in lexical order.

BZ#813807

Submitting a job through Aviary or the QMF interface previously caused the Qdate field to show a value of zero (0). Submission and job implementation code in Aviary and QMF have now been adjusted to ensure that any ATTR_Q_DATE variable set that arrives late is properly recorded. This guarantees that the Qdate field shows the correct time value.

BZ#855449

Retrieving submission IDs with a non-exact qdate offset in AFTER mode previously led to submission IDs not being returned even though getSubmissionSummary reports that they do exist for the specified time range. The problems that prevented the getSubmissionID implementation from processing the specified time range query parameters have been fixed, so the correct range of submission IDs are now returned to the client.

BZ#733498

Exposing new suspend/continue capabilities in the Grid job lifecycle management core through a public API was not possible in earlier releases. New WSDL operations have been defined and implemented so users can now remotely suspend and continue jobs using the Aviary scheduler SOAP interface.

BZ#886448

Providing a timeout value from the command line for any of the Aviary sample python scripts previously generated a Backtrace indicating that the timeout value applied to the internal Suds client was not an integer type. The utility function has been modified to cast the input value for a timeout to a valid integer type or to catch an exception and output error message, so supplied timeout values are now correctly parsed and applied.

BZ#732388

Queries to retrieve submissions by a single owner name have previously been unsuccessful because submissions would instead be returned for multiple owners. A remote Aviary client requesting submissions for a single owner is now recognised so the appropriate values are collected and returned. Queries to retrieve submissions by a single owner name are now fulfilled as expected.

BZ#800344

When submitting a job through the Aviary interface and providing an extra attribute (such as "args") that is assumed to be basic and should not be overridden according to the allowOverrides XML attribute, the override prevention used to fail if the case differed from what is defined internally for ATTR_<name>. The code now enables case-insensitive comparisons of the basic attributes that may be overridden, so basic attributes are correctly preserved if allowOverrides is false.

BZ#739219

When the Aviary getJobData operation ran a query for a job output file using the relative filename that may have been defined in a submit file, the file could not be found. The implementation will now try the supplied name first. If that fails, it will also try prepending the IWD value to the first value and retry a stat of the file. This ensures that job output files submitted with relative paths can be implicitly resolved in the getJobData operation.

condor-cluster-resource-agent

BZ#833343

Trying to add a JobServer to a RHHA-controlled schedd that did not exist did not previously raise an error message to indicate a failure. An error message is now printed if the schedd configuration doesn't exist, so the user is notified if the add process fails.

BZ#810982

Aviary locator support for RHHA-managed query servers has been added for this release. Aviary clients need to know which machine the query server is running on, because query servers configured to be managed by RHHA will change machines when the schedd does. The tools for configuring such query servers now provide a service to locate them.

BZ#828983

When used with RHHA, the condor resource agent did not previously verify that a daemon had started during a start operation. The start operation could then report success even though the daemon had not begun to start. To address this, the resource agent now waits 10 seconds to check that the process starts. So when a start operation reports success, the demon is guaranteed to have started.

BZ#833611

When the wallaby cluster-* commands have been used to only perform actions in the configuration store, the user has been prompted for the ricci user password. The user will now be prompted for the ricci user password only if the action will be performed on the cluster configuration, so merely making changes to the store will not require ricci authentication.

condor-plumage

BZ#840076

mongodb will now collect and store its job history and a standalone python client has been added for querying this history. This does not replace the existing history file infrastructure or tools like condor_history. Instead, it moves job history onto a more robust, scalable, and manageable backend data source for enterprise deployments.

BZ#801447

Cumin needs utilization data for reporting, but has been unable to fulfil the basic reporting feature. Records, fields, and indices are now included in the Plumage ODS plug-in data emission to mongodb, so Cumin can retrieve utilization reporting fields from mongodb as needed.

BZ#786815

'From' and 'To' date ranges have not been supported for the userlist, usergrouplist, and resourcelist options on plumage_stats, so a user was unable to limit their queries based on a date range. The 'From' and 'To' date range parameters are now passed to the appropriate functions within plumage_stats, so a user can query user, usergroup, and resource statistics within a specified time span.

BZ#786825

Server hostname and port arguments were not previously parsed correctly, which prompted a TypeError from python when plumage_stats was invoked with a server:port string. The unnecessary 'nargs' keyword is no longer in the parser option for a mongodb host, so plumage_stats now parses optional server arguments correctly.

condor-qmf

BZ#867989

Upstream changes in HTCondor previously modified the names of various condor_schedd daemon ClassAd statistical attributes, which caused statistics for a QMF scheduler object to show 0 values for attributes that should be non-zero. The implementation of the QMF schedd plug-in now implicitly maps from the old attribute names (7.6 series) to those renamed in the 7.8 series, so statistics for a QMF scheduler object show correct values for attributes as appropriate.

BZ#883794

The transition from 7.6 schedd stats to their 7.8 equivalents previously caused the cumulative job total to be incorrect. An incorrect variable assignment in the mapping code has now been fixed so the cumulative job total will be accurate and correct.

BZ#753822

The default configuration of the QMF Job Server used by cumin has been changed. The previous default was for cumin to use an embedded QMF Job Server object managed within a schedd plug-in component. Now, the default configuration is for a standalone daemon (condor_job_server) to be launched and for submission publishing within the plug-in's embedded Job Server to be disabled. Grid users using the previous default configuration should note that details will now be available in the new default for jobs that have been completed or removed from the queue.

condor-vm-gahp

BZ#750818

When running VM Universe jobs on RHEL5, the VM-Gahp previously attempted to update the

utime for the image without the correct permissions. The logic to update utime for the image has been removed from inside of the VM-Gahp because this is the hypervisor's responsibility, so the condor_vm-gahp should no longer trigger a SELinux error.

BZ#782054

Virtual images prepared with VM Universe that had the VNC console enabled would not start because the VNC console was reported as missing. A submission parameter has now been added to enable VNC settings, so images with the VNC console enabled now run correctly under condor.

condor-wallaby-base-db

BZ#800660

The cemote configuration feature was previously unable to configure the Aviary locator feature. The base database for Aviary now has new features that allow this configuration.

BZ#831756

When the ExecuteNode feature did not have ALLOW_NEGOTIATOR set, nodes could not allow matching. The ExecuteNode now includes ALLOW_NEGOTIATOR, so nodes will allow matching.

BZ#831709

Installing only the SharedPort feature on a node used to cause the master daemon to stop. A dependency on Master from the SharedPort Feature has been added so it is no longer possible to install only SharedPort on a node and prevent the master daemon from running.

BZ#746005

Because MRG Grid supports the Plumage feature, the Remote Configuration database now has features to enable Plumage functionality.

BZ#802823

Not all preconfigured features in the remote configuration default database have had names that adequately explain their function. All features now have an annotation with additional information.

BZ#767272

Upgrading wallaby used to change the ownership of database patch files from root to wallaby, which caused a verification problem. To prevent this, patch files are now always owned by wallaby, so upgrading wallaby no longer causes verification issues.

BZ#803359

The UNHIBERNATE parameter in the remote configuration base database has been too loosely defined, causing hibernated machines to wake up unnecessarily. The definition of UNHIBERNATE has now been tightened so not all hibernated machines will be woken up.

BZ#831725

The ALLOW_NEGOTIATOR_* values that were set on features in the Remote Configuration default database did not previously contain IP_ADDRESS. Some pools configured with Remote Configuration would then need additional changes in order to function. ALLOW_NEGOTIATOR_* param values now contain IP_ADDRESS, which has negated the need for additional changes.

condor-wallaby-client

BZ#851222

When multiple wallaby-agents were running against the broker that the configd is talking to, the configd would always choose the first one, which was not necessarily preferred. To address this, the configd now exits with a notification in its log file if it finds more than one wallaby-agent running on the broker.

BZ#815653

The configd previously used command line arguments instead of configuration parameters for broker user/password information. Now, the configd can use the same parameters as other qmf-related daemons (MF_BROKER_USER and QMF_BROKER_PASSWORD_FILE) to specify username and password when connecting to a broker.

BZ#815820

The configd previously used the QMF_BROKER_AUTH_MECHANISM parameter to specify authentication mechanisms when connecting to a broker, whereas other qmf-related daemons used QMF_BROKER_AUTH_MECH. Like other qmf-related daemons, the configd can now use both parameters, with QMF_BROKER_AUTH_MECHANISM taking precedence.

condor-wallaby-tools

BZ#786020

When using the remote configuration tools to remove a configuration feature that shared a parameter requiring user input with another feature, the shared parameter would be removed instead of being kept in the configuration. Logic in the tools has been corrected to detect if a parameter is needed by more than one feature, so the parameter will no longer be deleted along with the feature.

BZ#749569

Previously, a new special group called skeleton group was added by the wallaby service. Remote configuration tools such as condor_configure_pool and condor_configure_store were not able to configure the skeleton group. Additional support for the skeleton group have now been added to the remote configuration tools to rectify the issue.

cumin

BZ#886921

When a purge was attempted on queued messages, the purge failed with the message "Purge: Fauled (Incorrect number of arguments:expected 2, got 1). This was caused by a recent change in the queue.purge method which now required a "filter" argument. The fix now passes an

empty dictionary as the new filter argument for the purge method. The purge method should now complete successfully.

BZ#814386

Cumin previously used QMF methods for remote grid operations by default. This feature enhancement integrates Aviary web servers with Cumin, allowing support for job control, submission and job/submission queries. Also included is support for the discovery of Aviary endpoints through the Aviary locator service. Aviary web services is a default feature on installation. Cumin will only use QMF methods of Aviary is turned off.

BZ#812407

Cumin based reported statistics in the Overview page on Submitter objects which reported statistics on accounting groups. This did not necessarily a direct map of the correct sums across many Submission groups owned by a cumin user ID. This resulted in possibly incorrect values for a particular user. The Grid User Overview has now been changed to use sums from across Submission records where the owner matches the cumin used ID. All reported statistics for running, idle and held jobs should now match the information available on the Submissions tab.

BZ#789351

In order to view fully functional charts with cumin, the Flash Player functionality was needed. Instead of Flash Plager, a new Javascript-based charting solution that requires no special browser plugin has been implemented.

BZ#886924

The Exchange overview page does not display the charts for "Messages received, routed and dropped." An exception occured in the rate chart code that affected all rate charts, causing the issue. The exception has been fixed to pass the argument correctly and all dropped charts should now display correctly on the Exchange overview page.

BZ#846010

Cumin development can become difficult to coordinate between grid and messaging due to UI issues, graphing technologies, etc. To make it easier for grid and messaging to develop cumin for both, cumin-messaging has been created as it's own package. This allows users to choose between grid or messaging based on the package they have installed.

BZ#886937

Previously, attempting to remove a broker link in the Messaging tab resulted in failed errors. The link remove error was caused by a nonstandard object_id that was needed to remove that link. The code to fetch the qmf link object that constructed the link name has been corrected. Clicking "Remove broker link" should now correctly remove the link.

BZ#733516

Changes in Cumin have been implemented to obtain Aviary endpoints from either QMF or a new remote service automatically. This service allows Cumin to discover all of the Aviary endpoints in the pool by querying a service at a single well-known URL. Use of the locator service is turned off by default but maybe enabled in /etc/cumin/cumin.conf. Enabling the locator service relieves the maintenance burden in deployments with multiple Aviary services. For more

information, consult the Management Console Installation Guide.

BZ#799129

This enhancement adds Kerberos authentication in Cumin. This allows a kerberos authentication server to handle all cumin authentication. To enable, add "kerb" to the "auth" config in /etc/cumin/cumin.conf and set the kerberos_realm to the value required by the Kerberos Authentication server's setup in krb5.conf. Cumin will use the python-kerberos library to authenticate users. By default, all kerberos-authenticated users will be treated as non-admins in Cumin. In order to get a kerberos-authenticated admin user, add an "external" user to the cumin datbase and then add the admin role assignment via the cumin-admin utility.

BZ#807838

cumin-reporting has been introduced as a technical preview. cumin-reporting has been added to allow Cumin to use data from the condor-plumage (ODS) database to generate long duration visualizations of grid system behavior. This allows users to visualize system usage over longer periods of time than the 1 day maximum window that Cumin previously provided.

BZ#703859

A new chart has been added to the Grid Overview page that shows resource utilization by accounting groups, identified by the name "Pool Usage by Accounting Group". These charts show job submission//completion statistics and other features. Note that the data in this chart relies on data from the plumage subsystem in condor and will only yield data if the cumin machine has the pymongo package installed (currently only available in el6).

BZ#886942

The Access Control page displayed an incorrect menu, causing the page header to display twice. BrokerAccess Control is derived from ModeSet where classes have summary sections that contain task links. At the same time, Broker AccessControl is also a BrokerFrame modeset. This causes two summary sections to be displayed. To fix this, the ObjectView has been subclassed for Access Control and the header has been suppressed. The Access Control page now shows only one page header.

BZ#881826

The Queue overview page does not display the charts for "Messages enqueued and dequeued." An exception occured in the rate chart code that affected all rate charts, causing the issue. The exception has been fixed to pass the argument correctly and all dropped charts should now display correctly on the Queue overview page.

BZ#887167

The Connections overview page does not display the charts for "Bytes Sent and Received". An exception occured in the rate chart code that affected all rate charts, causing the issue. The exception has been fixed to pass the argument correctly and all dropped charts should now display correctly on the Connections overview page.

BZ#823506

After installing cumin on Red Hat Enterprise Linux 6.3, cumin does not come up after a reboot. In some cases where the postgres database does not fully start, cumin is affected. cumin retries connection to the database if psql returns the error code "psql:FATAL: the database system is

starting up". cumin has now been fixed to check for this error message. This error message will signal a restart of the cumin service. This should allow cumin to startup after a reboot.

BZ#752732

The list of Operating Systems exceeded the boundaries of the Overview page's frame. This is due to the cascading style sheet used on the page content's div box. The float property in the cascading style sheet has been cleared and the page layout of the Overview page should now accommodate the list of Operating Systems regardless of character length.

BZ#782359

Certain QMF method calls returned errors in the log files under normal cumin operation. While these errors are harmless and all method calls have been handled as expected, these may cause users concern when the errors appear on the log file. This fix changes the logging levels on this type of error to "DEBUG" so that the errors will only be seen in development context. These errors will not be visible in a normal user deployment.

BZ#783139

When cumin rendered a page where an object is found to be missing, the user is shown the error message "We can't find the object you requested." but is never taken back to the previous page. This requires the user to manually return to the initial site to try again. This is caused by inefficient handling of deleted submissions and errors in redirection. This fix improves the handling of deleted submissions, errors in the job pages now redirect to job lists or submission lists as appropriate. When errors occur, they now redirect users to the pages that will allow them alternative courses of action instead of a static error page.

BZ#851205

On the Scheduler page, the table and it's contents overran the boundary line of the page, resulting in misaligned formatting. No user functionality was impacted. This was caused by the CSS property table-layout which was set to "auto". To fix the formatting, the table-layout property has been set to "fixed" and column tags have been defined to restrict column widths. The table on the Scheduler page should now never run outside the boundary line.

BZ#799382

When exporting the quotes to CSV format, the cumin page would occasionally hang at the "loading" status instead of showing the actual values. This was caused by the CSV export prematurely triggering the export before all the quote values have been loaded. Cumin code has been fixed so that in CSV mode, cumin will trigger the rendering only when all the quota values are filled in. The exported CSV files should now show the correct values.

BZ#756384

Suspend and Continue buttons were added alongside the Hold, Release and Remove buttons on the QMF and Aviary interfaces in condor. This integrates suspend and continue into cumin. Suspend and Continue task links were added to the list of job control tasks on a job details page. Additionally, the Suspended job count statistic for submissions was added as a column on submissions lists. The "Enqueued" column values were abbreviated to make room, with the full value now displayed when hovering with the mouse.

BZ#887174

When viewing the session for a selected connection on the Connections page, all sessions are displayed for all connections instead of just one specific connection. A filter has been added to display only the sessions for the selected connection.

BZ#799404

When users exported a limits table to CSV format, the exported file included visible HTML markup, because the class that renders limits did not include details about handling export to CSV. Special handling has been added to the class to remove HTML tags during CSV export. Limits now display without markup in the exported file, as intended.

BZ#801047

Previously, the default behaviour of the ***sasl-mech-list*** parameter allowed Cumin to use all available SASL mechanisms to authenticate against a broker. Users were advised to disallow anonymous authentication manually. Since disallowing anonymous authentication is recommended in most use cases, the default behaviour of the ***sasl-mech-list*** has changed.

The new default behaviour for ***sasl-mech-list*** is as follows.

- For broker addresses that specify credentials (username and password) in the URL, ***sasl-mech-list*** defaults to the list of recommended password authentication mechanisms for Cumin (**PLAIN** and **DIGEST-MD5**).
- For broker addresses that do not contain credentials, ***sasl-mech-list*** defaults to **ANONYMOUS**.

The previous default behaviour can be acheived by setting ***sasl-mech-list*** to **AVAILABLE**.

BZ#802704

All wallaby nodes were always shown in the Inventory page, regardless of the filter value. This occurred because the filter was not applied to nodes from the wallaby subsystem. Filters now apply to nodes from the wallaby subsystem, and the Inventory page displays nodes according to the filter set.

BZ#850759

Previously, if a user attempted to submit a form after their session expired or the Cumin server was restarted, the form was deemed invalid, and the user's browser displayed a page with an error message similar to the following:

```
     APPLICATION ERROR

Traceback (most recent call last):
  File "/usr/share/cumin/python/wooly/server.py", line 145, in
service_page_request
    session.check_csrf()
  File "/usr/share/cumin/python/wooly/__init__.py", line 747, in check_csrf
    raise CSRFException("Possible CSRF attempt")
CSRFException: Possible CSRF attempt
```

Cumin has been updated so that users are redirected to the Cumin main page when an expired form is submitted, and prompted to log back in to Cumin when necessary.

BZ#635207

When necessary, Cumin now includes links to sub-shares in the group quota editing utility. This allows users to edit sub-shares more easily. Links to sub-shares are added below the sliders and pie chart in the group quota editing utility. Click on a sub-share link to navigate to the Edit Share form. When editing is complete, the browser returns to the quota table.

BZ#796798

Previously, Cumin installations used the persona named **default** as the default persona. This included both Messaging and Grid user interfaces. Since Cumin is primarily used in Grid deployments, Cumin installations now use the persona named **grid** as the default persona. Users can configure the personas named **messaging** and **default** by editing the **cumin.conf** file.

BZ#871453

The Cumin web server restarted when Cumin failed to handle some errors generated by SSL socket connections. The error handler has been updated to handle these additional errors, and errors generated by SSL sockets no longer cause the web server to restart. Instead, the failed connection is dropped and Cumin continues to operate as expected.

BZ#760567

Quota editing sometimes failed when no group information was present. This resulted in a stack trace being displayed in the browser. Moving to charting that did not require Adobe Flash Player involved adding a mechanism to log errors when this condition occurs, without showing a stack trace to the user.

BZ#853454

The ***StartdIpAddr*** job attribute could previously be edited because of a typographical error in Cumin code. Modifying this parameter has no known negative effects, but negative impacts cannot be ruled out entirely. The code has been modified so that the ***StartdIpAddr*** attribute is treated as read-only, as intended.

BZ#873335

When page update requests were made by a client without valid credentials, the web server did not redirect users correctly. After being prompted to log in, users saw an XML document instead of returning to a valid Cumin page. Cumin now issues a reload directive to the browser when an update request is received from a client without valid credentials.

BZ#848344

The Cumin Aviary Technology Preview requires command arguments for jobs submitted in Cumin using Aviary. Cumin therefore rejects submissions that specify commands without arguments. To work around this issue, expand the submission form by clicking on the **Show more** button, and add the string "**Args =**" to the **Extra attributes** field. This satisfies the restriction in Cumin Aviary without defining attributes.

BZ#846349

During database installation, the PostgreSQL user **cumin** was created as a database superuser. This granted the **cumin** user unnecessary privileges in the PostgreSQL database. The **cumin** user remains the owner of the **cumin** database, but is no longer granted

superuser privileges during installation. Instead, the **cumin** user is granted only the permissions required for Cumin to operate correctly.

BZ#768298

Previously, there was no clear documentation about supported browsers for Cumin. This caused problems for users who were unintentionally using unsupported browsers. A list of supported browsers has been added to the Cumin **About** page to help alleviate this issue.

BZ#801287

Previously, Cumin did not make use of a **pid** file, which made it difficult to determine the current status of the **cumin** service. A **pid** file for Cumin is now created at **/var/run/cumin.pid** when the service is started, and deleted when the service is stopped by **initd**. The existence of **/var/run/cumin.pid** when the **cumin** service is not running is considered evidence of a crash.

BZ#809006

Strings that contained special XML characters were being escaped more than once, which resulted in text containing special XML characters being displayed incorrectly. The **xml_escape** routine has been modified to prevent double-escaping, so strings that contain special XML characters should now display correctly in the browser.

BZ#787138

Notifications displayed by Cumin persisted until dismissed, but gave no indication of when they originally appeared. This reduced the usefulness of the notifications. Each notification message is now displayed alongside a timestamp marking the time of the initial notification display, to make it easier to determine when problems began to occur.

BZ#805029

Slotvis functionality was not optimal, did not scale well, and required Adobe Flash Player. Slotvis has therefore been removed from Cumin, which also removes the requirement for Adobe Flash Player.

BZ#800065

The **cumin** service script did not accurately detect the termination of the Cumin master process. The service assumed termination occurred immediately after it issued a SIGTERM to the master process, and removed the **pid** file. However, since Cumin child processes could take time to shut down gracefully, this was not always true. This meant that, if the **start** and **stop** commands, or the **restart** command, were issued while the process was still running, the **init.d** script erroneously returned **OK**.

The **cumin** service script now checks the status of the Cumin master process after a SIGTERM has been issued. If the process does not shut down within a given period (5 seconds), the service script notes that the process may not have shut down correctly, and does not remove the **pid** file. Existence of a **pid** file without a running service therefore indicates a service crash or a prolonged shutdown. Check **/var/log/cumin/master.log** for shutdown messages to verify.

BZ#750196

Cumin notification banners needed to be dismissed by a user; this meant that they could take up a large amount of screen space if left undismissed. This update adds the ***notification-timeout*** parameter, which specifies a number of seconds after which a notification banner is automatically dismissed. The default value for ***notification-timeout*** is **180** seconds. Set it to ***0*** to enable the previous behaviour, where banners are never dismissed without user interaction.

BZ#891919

The supported version of Firefox has been updated to Firefox 17. The Cumin **About** page has been updated appropriately.

BZ#907866

The Technology Preview of Cumin Aviary forced the **Requirements** field to default to **true** for all jobs submitted with Aviary. This meant that any user-set value for the **Requirements** field was overwritten, and therefore ignored. The construction of arguments during job submission has been corrected, and **Requirements** values set by the user are now passed correctly to condor when a job is submitted using Aviary.

wallaby

BZ#782816

If no authentication mechanism was explicitly specified for the wallaby shell, a spurious warning about an invalid broker mechanism was displayed. Wallaby has been updated so that this warning is displayed only if the explicitly-specified mechanism is invalid.

BZ#796406

Previously, if a group was deleted, and no other changes were made to nodes in the deleted group before they were activated elsewhere, nodes in the deleted group may not have received configuration updates. This resulted in incorrect node configuration when the nodes were later activated. Wallaby now adds members of a deleted group to an internal list of changed nodes, ensuring that their configurations are recalculated the net time the nodes are activated.

BZ#786801

Previous versions of the *wallaby* package included a logrotate recipe that incorrectly named archived logs. Rather than numbering archived logs (for example, **agent_log.1**, **agent_log.2** and so on), the archived logs were named **agent_log.1**, **agent_log.1.1**, **agent_log.1.1.1**, and so on. This updated *wallaby* package corrects the logrotate recipe so that archive files are named as expected.

BZ#802821

Previously, *wallaby* did not expose free-form textual descriptions or comments for features and snapshots. This made it difficult to document and to interpret snapshots. The *wallaby* service now allows users to annotate configuration entities and configuration snapshots with free-form textual descriptions. To annotate an entity, use the **annotation** property and the **setAnnotation** method of each entity type. To annotate snapshots, use the **wallaby make-snapshot** method, or call the **Store#makeSnapshotWithOptions** method with an **annotation** option set in the option map.

BZ#850205

Previous versions of the *wallaby* shell failed to catch certain errors thrown by the parameter handling library. This resulted in incomprehensible error messages being displayed when incorrect command-line arguments were used. The *wallaby* now catches these errors and displays a user-friendly error message.

BZ#807820

Previous versions of the *wallaby* package did not use the assigned UID and GID for the wallaby user. This meant that moving files between machines with *wallaby* installed was unnecessarily complicated. *wallaby* now uses its assigned UID and GID when creating a *wallaby* user if these are available in the operating system, allowing *wallaby* to retain the same UID and GID across systems wherever possible.

wallaby-utils

BZ#802799

Wallaby shell commands that began with **`replace`**, but did not provide a list of entity names, failed with an error. This made it difficult to replace the entity set with an empty set. Replace commands now operate without requiring a list of entity names, which allows entity sets to be replaced with empty sets.

BZ#851217

In some cases more than one *wallaby* agent were started on the same broker. This was not intended, and could lead to unpredictable results if *wallaby* shell commands were run while multiple *wallaby* agents existed on the broker. The *wallaby* shell now ensures that only one *wallaby* agent is running on the specified broker as part of its startup, preventing the aforementioned unpredictable behavior.

BZ#881366

Previously, trying to empty a metadata field with any *wallaby* shell modify command resulted in an error. This meant it was not possible to set metadata fields to empty strings. Modify commands now accept empty values.

BZ#820419

Previously, *wallaby* lacked the ability to display node configuration details that was available in the *condor* utilities. The shell has been updated with a command that allows users to inspect the node configuration as it will be received by the *condor-wallaby-client*.

BZ#801632

This update adds the ability to remove snapshots using the *wallaby* shell.

BZ#864091

Previously, the *wallaby* **`list-users`** command printed users with a role of **`READ`** as if they had a role of **`READ_ONLY`**. This was not consistent with *wallaby* user management commands. User roles displayed with **`list-users`** are now labelled correctly and consistently with the user management commands.

Users of the Grid capabilities of Red Hat Enterprise MRG 2.3 are advised to upgrade to these updated packages, which fix these bugs and add these enhancements.

6.2. RHSA-2012:1278 – Moderate: Red Hat Enterprise MRG Grid 2.2 security, bug fix, and enhancement update

Red Hat Enterprise MRG (Messaging, Realtime, and Grid) is a next-generation IT infrastructure for enterprise computing. MRG offers increased performance, reliability, interoperability, and faster computing for enterprise customers.

MRG Grid provides high-throughput computing and enables enterprises to achieve higher peak computing capacity as well as improved infrastructure utilization by leveraging their existing technology to build high performance grids. MRG Grid provides a job-queueing mechanism, scheduling policy, and a priority scheme, as well as resource monitoring and resource management. Users submit their jobs to MRG Grid, where they are placed into a queue. MRG Grid then chooses when and where to run the jobs based upon a policy, carefully monitors their progress, and ultimately informs the user upon completion.

Security Fixes

CVE-2012-2680

A number of unprotected resources (web pages, export functionality, image viewing) were found in Cumin. An unauthenticated user could bypass intended access restrictions, resulting in information disclosure.

CVE-2012-2681

Cumin could generate weak session keys, potentially allowing remote attackers to predict session keys and obtain unauthorized access to Cumin.

CVE-2012-2683

Multiple cross-site scripting flaws in Cumin could allow remote attackers to inject arbitrary web script on a web page displayed by Cumin.

CVE-2012-2684

An SQL injection flaw in Cumin could allow remote attackers to manipulate the contents of the back-end database via a specially-crafted URL.

CVE-2012-2685

When Cumin handled image requests, clients could request images of arbitrary sizes. This could result in large memory allocations on the Cumin server, leading to an out-of-memory condition.

CVE-2012-2734

Cumin did not protect against Cross-Site Request Forgery attacks. If an attacker could trick a user, who was logged into the Cumin web interface, into visiting a specially-crafted web page, it could lead to unauthorized command execution in the Cumin web interface with the privileges of the logged in user.

CVE-2012-2735

A session fixation flaw was found in Cumin. An authenticated user able to pre-set the Cumin session cookie in a victim's browser could possibly use this flaw to steal the victim's session after they log into Cumin.

CVE-2012-2459

It was found that authenticated users could send a specially-crafted HTTP POST request to Cumin that would cause it to submit a job attribute change to Condor. This could be used to change internal Condor attributes, including the Owner attribute, which could allow Cumin users to elevate their privileges.

CVE-2012-3492

It was discovered that Condor's file system authentication challenge accepted directories with weak permissions (for example, world readable, writable and executable permissions). If a user created a directory with such permissions, a local attacker could rename it, allowing them to execute jobs with the privileges of the victim user.

CVE-2012-3493

It was discovered that Condor exposed private information in the data in the ClassAds format served by condor_startd. An unauthenticated user able to connect to condor_startd's port could request a ClassAd for a running job, provided they could guess or brute-force the PID of the job. This could expose the ClaimId which, if obtained, could be used to control the job as well as start new jobs on the system.

CVE-2012-3491

It was discovered that the ability to abort a job in Condor only required WRITE authorization, instead of a combination of WRITE authorization and job ownership. This could allow an authenticated attacker to bypass intended restrictions and abort any idle job on the system.

Bug Fixes

BZ#784434

Frequent reconfiguration requests sent to the Negotiator daemon during a negotiation cycle could block the negotiation cycle and cause it to fail. With this update, reconfiguration requests received during a negotiation cycle are delayed until after the negotiation cycle has completed. Additionally, Negotiator data structures that are not reentrant with the **`condor_reconfig`** command remain protected.

BZ#784968

Previously, regular updates of concurrency limits required frequent reconfiguration of the Negotiator daemon, which reduced performance of an MRG Grid pool. This update adds a new parameter, **`NEGOTIATOR_READ_CONFIG_BEFORE_CYCLE`**, which forces Negotiator to re-read its configuration before the beginning of the negotiation cycle without a need to invoke a full reconfiguration. Concurrency limits can now be updated regularly without any negative impact on pool performance.

BZ#788617

Previously, the **condor_q -analyze** command could not be used to determine which exceeded concurrency limit prevented a job from being successfully negotiated. This update modifies the Negotiator daemon to provide the name of concurrency limit that causes the match failure. The **condor_q -analyze** command can now be used to determine the specific concurrency limit which causes a job match failure.

BZ#794660

Under certain circumstances, a partitionable slot could split into more dynamic slots than a machine could support. As a consequence, the scheduler could assign the machine jobs that would consume more CPUs and memory than was available. This update adds a new logic that corrects the slot allocation, and a machine can now have assigned only as many jobs as it is able to support.

BZ#799838

Recent changes to make sure that history index files were correctly cleaned up led to a situation where the internal job and submission collections were aggressively purged of active jobs. Consequently, the active jobs submitted through the Aviary or QMF interfaces were not reported by the interface. This update corrects this problem and jobs are now visible as expected during the whole job cycle.

BZ#807738

Previously, the ***DAEMON_LIST*** parameter was marked as **needs_restart** in the remote configuration database. Therefore, if the value of ***DAEMON_LIST*** changed, all daemons of MRG Grid component had to be restarted on a cluster node for the parameter change to take effect, which caused service disruption. With this update, the remote configuration database has been modified and the **needs_restart** flag has been removed from the ***DAEMON_LIST*** parameter. Changes of ***DAEMON_LIST*** no longer result in the restart of all MRG Grid daemons on a node.

BZ#810519

When attempting to remove a job on hold, which had invalid parameters and was never run, the job returned back to the **hold** state and the **condor_rm** command failed. This update corrects the state machine so that the **condor_rm** command now correctly removes a job that has been placed on hold due to invalid input parameters.

BZ#812126

The initial remote configuration database was recently modified to allow plug-in configuration outside of the remote configuration service. However, to simplify configuration, some packages store their configuration files in the directory pointed to by the ***LOCAL_CONFIG_DIR*** variable if the remote configuration was not used. This could cause some daemons to unexpectedly load plug-ins that were not properly configured. This update modifies the initial remote configuration database so that ***.PLUGINS** variables can no longer be defined outside of the remote configuration service. All plug-ins now must be configured using the remote configuration.

BZ#832968

Previously, code used for cleaning up the history index could cause a situation where internal job and submission collections were aggressively purged of active jobs. Consequently, the active jobs and their submissions were not reported by the Aviary or QMF query interface. This bug has been fixed and jobs and submissions for active jobs are now displayed as expected

even while history indexes are being cleaned.

BZ#748507

When the default group configuration used string append operators, Wallaby sometimes generated redundant node configurations. When such configurations were deployed to MRG Grid nodes, the **condor_master** daemon could fail to start. This bug has been fixed, Wallaby no longer generates such configurations, and can also handle archived redundant configurations.

BZ#801543

It is possible to have the **shadow_rec** entry set for a claim while no shadow process is present. During this time, the PID field of **shadow_rec** is 0. In rare scenarios, the **schedd** daemon tried to kill PID 0 and then terminated with the following error message:

```
Send_Signal: sent unsafe pid (0)
```

This update fixes the **Send_Signal()** function to not terminate in the described scenario and **schedd** now cleans up gracefully if there is a **shadow_rec** entry for a claim but no shadow process.

BZ#809799

Due to a regression, when a job with both Pre and Post scripts set was submitted to the DAGMan utility, DAGMan terminated unexpectedly. With this update, the Post script is now always processed even if a DAGMan job fails.

BZ#831235

Previously, Cumin did not validate redirection URLs specified in POST requests, creating a vulnerability in the Open HTTP Redirector.This bug has been fixed and Cumin now validates all redirection URLs. Only redirection to Cumin's own pages is allowed. If an invalid redirection is discovered, a redirection to the Cumin main page for the user is included instead.

BZ#831244

Prior to this update, Cumin allowed unrestricted CR and LF characters in response headers, creating a response header splitting vulnerability. With this update, Cumin scans all response headers for LF characters. Now, Cumin never writes LF characters into any legitimate response header so there is no danger of the scan interfering with legitimate responses. If an illegitimate response header is seen, a simple exception trace page is sent as the response.

BZ#840106

Previously, Cumin used the **random.getrandbits()** function to generate session keys. According to the Python documentation, this function uses the Mersenne Twister algorithm, which can leak state. It was theoretically possible to predict Cumin session keys after gathering enough leaked state information. With this update, Cumin reads from the **/dev/urandom** device to create session keys instead, thus preventing this problem.

BZ#840110

Cumin did not check for a valid session before processing requests for certain auxiliary web pages. Consequently, it was possible to retrieve certain types of data from Cumin or cause

activity on the server without a valid login. With this update, all critical pages now check for a valid session before processing and no unauthorized access to important Cumin pages is allowed.

BZ#840112

In some instances, Cumin did not correctly quote SQL expressions, creating an SQL injection vulnerability. Now, facilities provided in the **pysopg2** module are used to generate properly quoted expressions, thus preventing this problem.

BZ#840118

When requesting chart sizes in the PNG format, extra large graphics files were generated in some cases. Consequently, a denial of service (DoS) attack was possible abusing this feature, as affected Cumin instances and their hosts could eventually become unresponsive or terminate unexpectedly. With this update, if an image larger than two million pixels in size is requested, Cumin creates a log message and returns an image of the default size (360x100 pixels), thus preventing this problem.

BZ#840121

Previously, Cumin had no method to check the validity of POST requests, creating a Cross-site request forgery (CSRF) vulnerability. With thi Cumin injects a randomly-generated unique ID into every rendered form and does not process any POST requests that do not contain the correct ID, thus preventing this problem.

BZ#840123

Previously, Cumin did not reset the session cookie after authentication, creating a session fixation vulnerability. This bug has been fixed, Cumin now always resets the session cookie when the user logs in or logs out, and no session ID is ever preserved outside of a valid login session.

BZ#840133

Cumin did not correctly escape text in error trace displays. Consequently, XML special characters or JavaScript code within the error trace output could cause errors on the page. With this update, all text on the error page is properly escaped during rendering and the error pages are now correctly displayed by browsers, regardless of values that may be present in the stack trace.

BZ#807970

Cumin did not escape single (') and double (") quotation mark characters in strings contained in the page source. Consequently, erroneous operation with strings containing quotation marks could occur. This bug has been fixed and all quotation marks in page source are now properly escaped.

BZ#835595

Previously, the setuid rval variable was not checked, potentially creating a privilege escalation vulnerability. With this update, the setuid rval variable is properly checked and code is no longer vulnerable to privilege escalation via process exhaustion.

BZ#836294

Previously, it was possible for an attacker with a custom CEDAR client to authenticate as a user with a currently running job, creating a security vulnerability. With this update, authentication code has been improved to remove this vulnerability.

BZ#836253

Previously, it was possible for an attacker with a custom CEDAR client to have an unprivileged user to stop running jobs. This update removes unused code that allowed this unauthorized job control, thus preventing this problem.

BZ#836590

The security audit discovered the **`log_except()`** copied an error message of unknown length into the statically sized **`event.message`** buffer using the **`sprintf()`** function, potentially creating a stack squash vulnerability. This update hardens the internal interface, thus preventing this problem.

BZ#837623

Previously, it was possible for an attacker with a custom CEDAR client to have an unprivileged user to control active claims. This update removes unused code that allowed this scenario, thus preventing the problem.

BZ#837890

The security audit discovered a problem in the **`LookupString()`** function, which could potentially lead to a stack smashing vulnerability. This update hardens the internal interface, thus preventing this problem.

BZ#837037

Cumin relied on screen presentation to control which job attributes were writable. Consequently, manually constructed POST requests could allow a user to edit protected attributes by circumventing the screen controls. With this update, Cumin uses information from Condor to validate job attribute changes. It is no longer possible to modify attributes that Condor marks as read-only, even through manually constructed POST requests.

BZ#852321

When using Remote Configuration to configure a High Availablity Scheduler with the **`HAScheduler`** feature, the **`shadow`** daemon could fail to start with the following error message:

```
ERROR "According to /var/lib/condor/spool/spool_version, the SPOOL
directory is written in spool version 0, but I only support versions back
to 1"
```

This update adds parameters to point the **`shadow`** daemon at the same spool location used by the **`HAScheduler`** feature. Now, **`shadow`** runs jobs as usual using the spool pointed to by the node with the **`HAScheduler`** feature installed, thus fixing this bug.

BZ#848212, BZ#835592, BZ#841173, BZ#843476

This update also provides defense in depth patches for Condor.

Enhancements

BZ#769573

Cumin had no mechanism for distinguishing between administrative and general users. As a consequence, the different user roles were not respected and all users could access all displays and use all Cumin functions. With this update, the role enforcement mechanism has been fully implemented but it is disabled by default for backwards compatibility. It can be enabled in the **/etc/cumin/cumin.conf** configuration file. When the role enforcement is enabled, general users can see only the displays under the Grid User tab, and are able to see and manage only their own jobs. By default, all users are assigned to the general user role unless the **cumin-admin** command is used to grant administrative privileges to a user.

BZ#591521

The **startd** daemon did not previously support managing of local machine resources above the standard CPU, memory, disk, and swap resources. With this update, **startd** has been enhanced to allow additional local machine resources to be specified in the cluster configuration. These resources are managed by **startd** and the slot resource accounting mechanism, and can be requested by job submissions.

BZ#721110

The Negotiator daemon previously did not enable to alter concurrency limits for jobs without a need of Negotiator's reconfiguration. This could have a significant impact on Condor pool performance if the daemon reconfiguration was invoked on a frequent basis. This update enhances Negotiator to support named groups for scoping multiple default concurrency limits based on a limit name prefix. Concurrency limits can now be defined with multiple possible default values without invoking frequent Negotiator reconfigurations.

BZ#737979

Previously, Cumin user authentication was limited only to the authentication against the Cumin database. Therefore, sites could not use their existing user LDAP accounts and the accounts had to be recreated in the Cumin database in order to access the Cumin management console. With this update, Cumin has been extended to allow the use of LDAP servers for authentication. If a user is not found in the Cumin database, Cumin attempts to authenticate a user against a specified list of LDAP directories.

BZ#751870

A new Condor Resource Agent has been implemented that allows multiple HA Schedd scheduler daemons, managed by Red Hat High Availability, to run on the same node.

BZ#784051

Previously, the ClassAd log-file system did not report the location of parse errors and it was difficult to locate corruption in the job queue log. With this update, the ClassAd log-file system has been enhanced to provide the number of the record where a parse error was detected. Also, the logic for parse error identification has been improved. Parse errors can now be easily located in ClassAd transaction log files, such as the job queue log.

BZ#785140

Previous implementation of CPU accounting, available with the **condor_status** command, did not handle configurations with multiple partitionable slots on a single machine. Therefore, when attempting to determine CPU utilization for given ClassAd data on such a configuration, the **TotalCpus** attribute provided inaccurate metrics. This update modifies the underlying code and adds new attributes, **TotalSlotCpus**, **TotalSlotMemory**, and **TotalSlotDisk** to the ClassAd slot. The CPU accounting now works correctly on configurations with multiple partitionable slots on a single machine and users can now run simple queries to determine partitionable slot utilization.

BZ#785145

Previously, Red Hat High Availability could only manage a single running HA scheduler instance on a node. This update introduces a new set of tools that allow Red Hat High Availability to manage multiple HA scheduler instances on a node.

BZ#803895

This update adds Deltacloud support to the Condor's Grid Universe so that users can now submit their jobs to Condor, which run against the Deltacloud API.

BZ#806071

This update adds new features and parameters to the remote configuration database so that the database now supports multiple job/query server configurations on a single node managed by Red Hat High Availability.

BZ#820681

Previously, the Cumin web server did not support secure communication over SSL and data from a web browser to the Cumin web server was therefore sent in plain text. This update modifies the code so that the Cumin web server can now be configured to use SSL in the **/etc/cumin/cumin.conf** file. See Management Console Installation Guide for more information.

BZ#806079

This update adds the ***VM_NETWORKING_BRIDGE_INTERFACE*** parameter to the initial Wallaby database to support remote configuration of bridged networking in the virtual machine universe. In addition, the ***VM_SCRIPT*** parameter has been removed from the database and most of the virtual machine universe parameters have been marked as **needs_restart** and added to the **startd** subsystem.

Users of the grid capabilities of Red Hat Enterprise MRG 2.2, which is layered on Red Hat Enterprise Linux 5, are advised to upgrade to these updated packages, which fix these bugs and add these enhancements.

6.3. RHSA-2012:0100 – Moderate: MRG Grid security, bug fix and enhancement update

Red Hat Enterprise MRG (Messaging, Realtime, and Grid) is a next-generation IT infrastructure for enterprise computing. MRG offers increased performance, reliability, interoperability, and faster computing for enterprise customers.

MRG Grid provides high-throughput computing and enables enterprises to achieve higher peak computing capacity as well as improved infrastructure utilization by leveraging their existing technology to build high performance grids. MRG Grid provides a job-queueing mechanism, scheduling policy, and a priority scheme, as well as resource monitoring and resource management. Users submit their jobs to MRG Grid, where they are placed into a queue. MRG Grid then chooses when and where to run the jobs based upon a policy, carefully monitors their progress, and ultimately informs the user upon completion.

Security Fix

CVE-2011-4930

Multiple format string flaws were found in Condor. An authenticated Condor service user could use these flaws to prevent other jobs from being scheduled and executed, crash the **condor_schedd** daemon, or, possibly, execute arbitrary code with the privileges of the **condor** user.

The changes in this advisory are mostly the same as those for the Red Hat Enterprise Linux 6 MRG Grid Release. The following update is specific for Red Hat Enterprise Linux 5:

Bug Fixes

BZ#759200

A bug in the *python-psycopg2-2.0.14* package caused a reference leak when the **cumin-data** utility updated objects in the database. Specifically, the bug manifested when the **cursor.mogrify(operations, params)** or **cursor.execute(operations, params)** functions were called and the **operations** string referenced the same value from the **params** list more than once. Consequently, long-running instances of **cumin** could leak significant amounts of memory. With this update, the **_mogrify()** routine has been fixed and the reference leak no longer occurs when **cumin-data** updates database objects.

Refer to Section Section 5.3, " RHSA-2012:0099 – Moderate: MRG Grid security, bug fix and enhancement update " for the remaining updates in this erratum.

6.4. RHBA-2012:0045 – Red Hat Enterprise MRG Grid 2.1 bug fix and enhancement update

The changes in this advisory are mostly the same as those for the Red Hat Enterprise Linux 6 MRG Grid 2.1 Release. The following updates are specific for Red Hat Enterprise Linux 5:

Bug Fixes

BZ#569561

The MAX_[subsys]_LOG values were read using the **atoi()** function instead of a 64-bit-capable lexical casting function, which rendered it impossible to set a maximum log size equal to, or greater than, 2GB. The lexical casting function **lex_cast()** has been implemented to read MAX_[subsys]_LOG values, with the result that log files of size equal to, or greater than, 2GB, can now be configured.

BZ#748738

MF console application written in Python, after a period of time a noticeable amount of memory

was lost due to a slow memory leak. After a long enough period, the console would use up all available memory and experience problems. This update explicitly calls the clean-up method of the sequence manager for each request once the request had been handled, unneeded objects are now properly released, and the memory leak no longer occurs.

Refer to Section 5.4, " RHBA-2012:0046 – Red Hat Enterprise MRG Grid 2.1 bug fix and enhancement update " for the remaining updates in this erratum.

6.5. RHSA-2011:1249 – Moderate: Red Hat Enterprise MRG Grid 2.0 security, bug fix and enhancement update

Red Hat Enterprise MRG (Messaging, Realtime, and Grid) is a next-generation IT infrastructure for enterprise computing. MRG offers increased performance, reliability, interoperability, and faster computing for enterprise customers.

MRG Grid provides high-throughput computing and enables enterprises to achieve higher peak computing capacity as well as improved infrastructure utilization by leveraging their existing technology to build high performance grids. MRG Grid provides a job-queueing mechanism, scheduling policy, and a priority scheme, as well as resource monitoring and resource management. Users submit their jobs to MRG Grid, where they are placed into a queue. MRG Grid then chooses when and where to run the jobs based upon a policy, carefully monitors their progress, and ultimately informs the user upon completion.

Security Fix

CVE-2011-2925, Moderate

A flaw was discovered in **cumin** where it would log **broker authentication credentials** to the cumin log file. A local user exploiting this flaw could connect to the broker outside of cumin's control and perform certain operations such as scheduling jobs, setting attributes on jobs, as well as holding, releasing or removing jobs. The user could also use this to, depending on the defined ACLs of the broker, manipulate message queues and other privileged operations.

Release Notes

BZ#728285

When MRG Grid run on a node with multiple network interfaces, it tried to estimate the correct interface for its communications with the remaining MRG Grid nodes. As a consequence, the node could have failed to communicate with other parts of MRG Grid correctly if a wrong interface had been chosen. As a workaround to this issue, MRG Grid can be forced to use a specific network interface by setting the ***NETWORK_INTERFACE*** parameter to the IP address of that interface. To find out which interface has been used by MRG Grid when it failed, the **D_HOSTNAME** variable has to be included in logging configuration for a daemon that deals with issues.

N/A

The remote configuration database requires an update to include changes for MRG Grid version 2.0. But the database snapshot provided with MRG only contains a basic configuration, and thus loading the database snapshot would replace the existing pool configuration. To solve this issue, the **upgrade-wallaby-db** tool which upgrades an existing deployment's database has to be used. This tool can be downloaded from the following page: https://access.redhat.com/kb/docs/DOC-58404

BZ#688717

With this update, the **Elastic Compute Cloud Grid ASCII Helper Protocol** (EC2 GAHP) is preferred over **AMAZON GAHP**. The *condor-ec2-enhanced-hooks* package has been updated to detect the correct GAHP for the EC2 Enhanced feature based upon what GAHPs are available on the scheduler. To ensure that jobs are routed to the proper resources, the **'set_gridresource = "amazon"; \'** setting has been removed from all existing **EC2 Enhanced routes** in a MRG Grid's configuration.

Bug Fixes

BZ#723971

When the definition of the ***subsys.parametr*** parameter was missing in MRG Grid's configuration, such as **LL_DAEMON_LOG**, a **LOG** parameter value would have been used instead, which caused a confusing exception to be thrown. This happened because the **LOG** parameter contained the name of a directory which did not correspond to the one in subsys.parameter, which contained an ordinary file name. To fix this issue, a control option, which allows to raise an exception if subsys.parameter does not exist, has been added to the **read_condor_config** script. Subsystem configuration now provides useful error messages.

BZ#720507

Due to a race condition in the **condor_schedd** daemon, the daemon could have accessed stale job **ClassAds** when the matchmaking proceeded too slowly. This would have resulted in the scheduler crashing. To fix this issue, the race condition has been removed from the **condor_schedd** daemon, and the daemon no longer tries to access a stale job **ClassAd**.

BZ#720400

When a cluster ID was rolled over and the ***SCHEDD_CLUSTER_MAXIMUM_VALUE*** parameter was set and the queue log rotated, a consistency verification check ensuring that the stored cluster ID was lower than the maximum cluster ID would have raised an exception causing the scheduler to restart, which resulted in a deleted job queue and failure of the scheduler. To avoid this issue, this consistency check has been disabled when ***SCHEDD_CLUSTER_MAXIMUM_VALUE*** is set. Rolled over cluster IDs no longer cause the scheduler to fail on restart.

BZ#718104

The **ClassAd debug()** function was not implemented with the new style **ClassAd** code, which restrained functionality of all features that previously used it. With this update, the **debug()** function has been re-implemented, and all related tools and logs now contain debugging messages.

BZ#713511

If a *pre-script* returned a failure while executing a DAG (Direct Acyclic Graph) job on an MRG Grid, then the *post-script* failed to run. This could have led to job inconsistencies in the MRG Grid. With this update, the post-script is now always executed regardless of the *pre-script* outcome.

BZ#712973

Due to the **condor_negotiator** daemon attempting to fill the fractional *submitter limit* remainder of a **group quota**, the **schedd** daemon tried to assign every possible job cluster regardless of the fact that it was impossible to achieve. As a consequence, the **negotiator** daemon executed many unproductive negotiation attempts, which ended with rejection due to the exceeded *submitter limit*. To correct this issue, a new verification mechanism has been added in order to halt further negotiations for a particular group when the s*ubmitter limit* is exceeded. Redundant negotiations are no longer executed, and negotiations now properly stop on the first such rejection.

BZ#712529

When compiling the source code, the **CMake** compiler removed the original **condor_router_*** scripts and replaced them with bogus executable files due to certain placeholders. This rendered the **Condor**'s job routing scripts non-functional. With this update, the **CMake** operations has been updated, the correct scripts are now being used during the compilation process, and job routing now works as expected.

BZ#707770

The logic for *deactivating claims* in the dedicated **scheduler** iterated over a *claim data structure* while this structure was being destructively modified. Due to this issue, some of the claims were not properly released. This logic has been corrected and all claims are now properly deactivated.

BZ#707335

When a **condor_reconfig** event occurred in the middle of the negotiation cycle, structure pointers of the **condor_negotiator** daemon that were stored in the stack could have been invalidated, which would have caused a memory read error and the daemon would have terminated unexpectedly. To fix this issue, functionality delaying the **condor_reconfig** events has been added. With this update, any current negotiation cycle now completes first and the **condor_reconfig** event is executed directly after that. The **condor_negotiator** daemon proceeds without the read error and no longer crashes.

BZ#707081

In **Condor**, accounting groups were sorted out according to the group quota usage that they were using and the number of allocated resources. During a negotiation cycle, the least-used group was considered first for claiming, the second-least was considered next, and so on. With the **Hierarchical Accounting Groups** feature implemented, this sorting mechanism had been left out, and it could have happened that certain groups were never used. The original accounting group sorting method has been restored, and this issue no longer occurs.

BZ#707078

Due to the **ignore_schedd_limit** uninitialized stack variable that is called in an inner loop within the **negotiate()** function, the *job submitter limits* could have been ignored when expected to be followed, or the limits could have been followed when expected to be ignored. This obsolete variable has been correctly replaced by the **ignore_submitter_limit** variable, which now guarantees proper behavior regarding *job submitter limits*.

BZ#706512

Previously, a *job submitter* with no idle jobs received a number of slots according to its limit regardless of a need for them. Those unused slots required additional iterations in a *negotiation loop* that could have been avoided if the slots were given to other job submitters. With this update, functionality that pre-filters job submitters with no idle jobs has been added, and thus these freed slots are assigned to job submitters which can use them, making the negotiation phase more effective.

BZ#704597

When the **condor_configd** daemon attempted to retrieve a node object from the configuration store during its periodic verification, an exception was not handled properly, and communication could have been interrupted. When the thread controlling the periodic updates exited, the affected node stopped its periodic verification of the configuration store. With this update, when the exception arises, it is handled properly. All communication threads are now monitored, and they are restarted if any of them are not running. The **condor_config** periodic updates are not interrupted any longer.

BZ#699413

Prior to this update, the **MRG Management Console (Cumin)** filtered objects which were not directly associated with the **condor_collector** daemon that was discovered by **Cumin**, and such objects were not displayed. As a consequence, in hierarchical configurations for collector daemons, certain objects which were published by condor daemons with the **COLLECTOR_HOST** variable set to a subordinate **collector** daemon would have not been visible in **Cumin**. To fix this issue, collector-based filtering has been disabled in **Cumin**. **Cumin** now processes every object that it receives from the **MRG Messaging Broker**, which allows all objects to be visible in the hierarchical **collector** configuration. However, **Cumin** is only able to discover objects from a single Condor pool because deploying multiple Condor pools on the same **Messaging Broker**, or configuring **Cumin** to use multiple **Messaging Brokers**, which serve different Condor pools simultaneously, could lead to unpredictable results.

BZ#719019

Due to support for the **SlotWeight variable**, functionality of the **condor_startd** ***RANK*** expression, which allowed a node to be configured to prefer certain jobs over others, was broken, and ***RANK*** would have preempted only if the user priorities allowed it. In order to fix this issue, the **condor_negotiator** logic has been updated to handle evaluation of ***RANK*** properly. ***RANK*** now preempts as expected.

BZ#718265

When running a low latency job which caused the **condor_starter** daemon to exit prematurely, the **Condor** low latency daemon did not expire the job, and the slot running this job could have not been used for other jobs until the **Condor** low latency daemon was restarted. The issue with a MRG Messaging message expiration has been fixed, and messages now expire as expected.

BZ#716519

Previously, halting conditions for an outer loop of a negotiation cycle were tested at the end of the loop, which, if **pieLeft == 0.0** was true, then the loop could have been executed even though there were no slots to allocate. With this update, this condition is now tested earlier in the loop and additional loop executions are now avoided.

BZ#715973

Due to improper *Boolean value-handling* in the code, submitting the edit attributes page for a job resulted in multiple edit attempts equal to the number of a job's attributes. This has been fixed: Boolean values for the *edit job attributes page* are now handled correctly, and the number of entries in the task indicator now corresponds to the number of edited job attributes.

BZ#715956

The **JobAdsEditor** form had the **update_enable** option set to **true**, which caused page content to refresh in accordance with the page content update interval. If the update interval was set to a low value, job attributes could have been difficult or even impossible to edit. To resolve this issue, periodic page content updates have been disabled for **JobAdsEditor**, and the job editor form now no longer resets.

BZ#712987

When preparing job submissions for a negotiation, jobs were primarily sorted by *submitter priorities*, then secondarily by the *name of the submitter*. This sorting mechanism resulted in *submitter job starvation* because submitters always negotiated in the same order. With this update, the submitter sorting mechanism has been changed so that the submitter is primarily sorted by priority, and secondarily by *starvation ratio*. When negotiating for jobs, all submitters have now equal conditions.

BZ#712975

Previously, the ***GROUP_DYNAMIC_MACH_CONSTRAINT*** expression did not remove startd resource ads which matched the expression from the resource ad list. This issue resulted in resource traversal overhead and decreased performance. To fix this problem, a new ***NEGOTIATOR_STARTD_CONSTRAINT_REMOVE*** parameter has been added. In order to ensure backward compatibility, this parameter is set to **FALSE** by default, but if it is set to **TRUE**, all ads which do not match the ***GROUP_DYNAMIC_MACH_CONSTRAINT*** are removed, and thus redundant resource usage can now be avoided.

BZ#712972

When using the *round-robin* scheduling algorithm, groups containing no submitter could have started a group negotiation, which was obviously inefficient. This was caused by **condor_negotiator** logic that removed submitters with no idle jobs. To resolve this issue, the **condor_negotiator** code has been modified to skip groups without submitters. These groups no longer spend time with group negotiation.

BZ#708944

When a group of long-term running jobs was quickly set on hold and then released, jobs could have accidentally been removed from the job queue. The code has been modified to prevent the issue, and jobs now remain in the queue.

BZ#707576

When running a HTTP session with the **Cumin** MRG Management Console, the session was assumed to always be active. If the session timed out, **Cumin** had no handling mechanism for session redirection requests implemented for the background page update. So, when the session timed out, **Cumin** stopped refreshing the page. With this update, this exception is now properly handled and the background page update code has been modified so that users are

redirected to the login page when their session times out.

BZ#706977

Previously, sanity verification of *group quota surplus* allocation examined an incorrect variable. This problem resulted in incorrect warning messages, which stated that the group quota needs to have a remaining surplus, even though allocation was correct. This sanity verification has been corrected to examine the proper variable, and such redundant warning messages no longer appear.

BZ#703905

Due to unnecessary verification in the **`condor_negotiator`** logic, **Condor** rejected a submitted job in the first phase of the negotiation cycle even though submitters did not reach their limits, which resulted in additional iterations in the next negotiation phase. Redundant verification has been removed and submitters now can complete the negotiation cycle with an expected number of iterations.

BZ#703630

The *full page chart link* was previously mapped so that it was difficult to associate it with the appropriate graph. To fix this issue, the full page chart link has been moved to align with the graph.

BZ#703621

The *y-axis maximal value* was previously determined according to points, which were out of reach of the display window. Consequently, only data with the *low y-axis value* with respect to the scale of the chart were showed in the graph. To fix this issue, the *y-axis maximal value* is now based only on points that are currently in the view, and the graphs are now wholly visible.

BZ#703283

Due to a missing *SQL where-clause* that would have helped to distinguish between idle, running and completed statistics of all users, and the current user, the **`Job Statistic`** table displayed information based on the statistics for all users instead of the current user. With this update, the needed where-clause has been added in order to restrict displayed job statistics to the currently logged-in **Cumin** user. The **`Job Statistic`** table now shows proper statistics for the current grid user.

BZ#703279

When searching data in a **Cumin** table which contained multiple pages, applying a filter to the description field resulted in a seemingly empty result list, and a blank page was displayed if the user was not currently viewing the first page. The user's intervention was required in order to see the actual results. This issue has been fixed: when a new search filter is applied, the first page containing the list of searched results is now displayed.

BZ#703196

When editing attributes in **Condor**, submission of an argument which was not enclosed in double quote characters would have caused this argument to be interpreted as an expression, not a string. When the argument contained non-numeric characters, attempting to set the attribute would have failed even though **Cumin** reported success. As a workaround, arguments must be enclosed in double quotes whenever arguments contain non-numeric values. With this

workaround, all such arguments are now correctly interpreted.

BZ#702440

Stopping **Qpid Management Framework (QMF)** agents while **Cumin** was stopped, or when it was restarted to point to a different **MRG Messaging Broker** than it was prior to the restart, could cause objects from missing agents to be displayed in the **Cumin** when it was restarted. Such objects were deleted by **Cumin** only when missing **Sesame** agents were restarted using the original IDs and then removed while **Cumin** was running. To correct this issue, **Cumin** has been enhanced with a clearing mechanism which ensures that all dynamic data in **Cumin** is deleted at startup, and **Sesame** agents with their associated objects are now rediscovered while **Cumin** is running. **Cumin** now displays only actual objects.

BZ#700863

When using the **condor_configure_pool** command, any attempt to set a feature which contained parameters that had to be modified would have failed if the **--schedd** option or the **--qmfbroker** option was used in the same command. To prevent this issue, the code has been modified to update the QMF group object before any parameters to set scheduler or **QMF Broker** information are applied. Using the **condor_configure_pool** tool to set a feature parameter simultaneously with setting the scheduler or **QMF Broker** information now works correctly.

BZ#699732

Due to a lengthy job submission name, the *submission selection table* as well as the single *submission display page* were distorted and the submission name was displayed running to the right edge of the page. To correct this issue, a displayed *job submission name* on these pages is now truncated to 100 characters and the "..." string is attached to the name, indicating the truncation.

BZ#699643

condor-reuse-slot* user accounts appeared on the welcome screen after **Condor** was installed on the 64-bit version of Windows Server 2008 R2 and Windows 7 operating systems. This issue has been fixed by updating **UserList** registry calls to write entries to the 64-bit registry. Slot user accounts are now hidden on all supported 64-bit versions of the Windows operating system.

BZ#682447

When using the **MRG Management Console**, the *page update timestamp* was updated even though the page update failed with no indication of failure, which led the user to assume that the displayed data was up-to-date. When the last update fails, the user is now informed with timestamp information.

BZ#673273

Under certain circumstances, verifications related to the database server for the **start** command, which are contained in the **/etc/init.d/cumin** script, could have prevented the **status** and **stop** commands from running. To avoid this issue, the script has been modified to run these checks only during start-up of the **Cumin** service. The **status** and **stop** commands now successfully complete even though the database server is stopped.

BZ#637963

Previously in **Cumin**, it was possible to set mutually-exclusive static and dynamic group quotas, which caused an error to be displayed while editing the *quota table*. To prevent this issue, a new column on the page showing statistical limits has been added. Only *dynamic quotas* can now be edited and *static quotas* are now properly displayed in the quotas table.

BZ#609510

The orphaned child process of a **DAGMan** job could have been left in the queue. This happened when the **submission.dag** file was removed while the **DAGMan** process was in the *HOLD* state. With this update, any orphaned **DAGMan** processes are removed from the queue if **submission.dag** becomes unavailable.

BZ#723613

Cumin detected a type mismatch when editing an integer or floating-point attribute value on the **Edit Attributes** page for a job. This scenario was handled incorrectly and consequently, the **Loading...** message was displayed on the page indefinitely. With this update, **Cumin** handles these scenarios correctly and when an invalid value for a floating-point or integer job attribute is entered, the user is now warned that the edit operation failed due to a type mismatch. Operation otherwise proceeds normally.

BZ#720374

Certain utility scripts did not end in a suffix, which is required for executables on Windows. These scripts are no longer included in the **MSI installer**.

BZ#712974

The *limits-editing page* did not follow the convention of hyperlinking the text of the object to be edited, causing confusion. This update ensures that the limit value is hyperlinked instead of the limit name, thus providing consistency with other editing pages.

BZ#710215

The **WantAWS =!= True** condition needs to be specified in the job submission file for **EC2 Enhanced** jobs to ensure that a job is correctly routed to EC2. Previously, with this set up, the job was correctly routed to EC2 but it could not complete. To fix this issue, the **WantAWS = False** condition has been set up in the ClassAd specifying each job, which is provided to an EC2 AMI. Now, **EC2 Enhanced** jobs that include the **WantAWS =!= True** condition in their requirements run and complete as expected.

BZ#707584

The *Inventory page* contained redundant check boxes, which have been removed.

BZ#705437

The **scheduler** could have potentially suffered a memory access error due to a missing check for an empty queue. This check has been implemented, thus eliminating the chance of incurring a memory access error.

BZ#704490

On Windows, the system returned an incorrect **Condor** version. This update modifies the build spec file and the system provides the correct platform data.

BZ#704653

The *Cumin service init script* must be run as the root user. However, the script lacked a check to ensure that it was run by the root user, with the result that running the script as an unprivileged user resulted in misleading error messages. With this update, the script fails immediately (and displays **[FAILED]**) when it is run as an unprivileged user.

BZ#703860

When a column with numeric values was selected as the sort criterion for a table, **Cumin** sorted the rows in ascending order of the numeric values by default. With this update, **Cumin** sorts the tables in descending order if sorting according to a column with numeric values.

BZ#701966

The **condor_configure_store** tool prompts the user whether the configured parameters, which are not in the store, are to be added to the store. Previously, if the user declined, the tool still prompted the user about whether these parameters were to use their default values. With this update, **condor_configure_store** no longer prompts the user for default values in the described scenario, thus fixing this bug.

BZ#701337

This update adds the ***UPDATE_INTERVAL*** parameter to the default database of the remote configuration. The ***UPDATE_INTERVAL*** parameter defines the interval in which the **startd** daemon checks with the **collector** when using remote configuration.

BZ#700595

In previous versions of the **wallaby** service, the *wallaby* package assumed the existence of a **condor** group. However, the *wallaby* package did not depend on the *condor* package, which creates this group, and the *wallaby* installation on a machine without a **condor** group failed. The *wallaby* package now creates a **condor** group if necessary and the installation succeeds on all machines.

BZ#700545

If the user configured **Condor** as an **AviaryScheduler**, the system could fail to activate. This happened because the **AviaryScheduler** did not depend on the **BaseScheduler**. With this update, the **AviaryScheduler** depends on the **BaseScheduler** and the configuration with the **AviaryScheduler** feature is activated successfully.

BZ#700540

The value displayed for **uptime** was in seconds, which was not very readable or usable. This update changes the **uptime** value to be displayed in **DD:HH:MM:SS** format, which is much more readable and useful.

BZ#697093

The *Cumin charts* displayed the values exceeding one million in thousands (for example, "1000k") on the y-axis. This update adjusts the labels to display the values in millions (for

example, "1M").

BZ#697016

In the *limit editing form*, there was no intuitive way of setting the limit to the **unlimited** value. This update adds a button, which changes the text box entry to **Unlimited**. Alternatively, the user can also type the value **unlimited**.

BZ#696697

The user interface allowed negative maximum allowance values to be entered when editing a limit under the **Limits** tab. Although the system accepted the negative value and displayed it in the user interface, internally it treated the value as if it were 0 (zero). This update changes this behavior so that entering a negative maximum allowance value causes **Cumin** to display an error message, and internally the limit is not changed.

BZ#681651

The sample configuration file provided with the *ec2-enhanced-hooks* package had to be renamed and edited in order to fit into the **LOCAL_CONFIG_DIR** directory hierarchy. With this update, this file must no longer be renamed to fit into the **LOCAL_CONFIG_DIR** hierarchy.

BZ#681650

Previously, installing the *condor-ec2-enhanced* package required manual configuration. With this update, the package installs a default configuration file, with the result that manual intervention and user configuration is no longer required.

BZ#681648

The *condor-low-latency* package now provides an example configuration file that can be modified and placed into the directory pointed to by the **LOCAL_CONFIG_DIR** variable. Using **LOCAL_CONFIG_DIR** allows for greater flexibility in configuring external features, and configuration of **Low-Latency** is now easier.

BZ#674598

Cumin's configuration file parser treats a line that begins with whitespace as a continuation of the previous line. In **cumin.conf**, whitespace at the beginning of a line that was not a (valid) continuation of the previous line caused parsing errors which were not communicated to the user, and which could have caused the **cumin** service to restart the application in an endless loop. With this update, any parsing errors are caught by the **init** script, any exception traces are recorded in log files under the **/var/log/cumin/** directory (the exact file depends on whether cumin-data or cumin-web first detected the error), and a parsing error results in the service failing upon startup (with **[FAILED]** printed to alert the user).

BZ#659247

An object that was in the process of being deleted could have still been selected in **Cumin**, and any further operations on such an object resulted in internal errors. In turn, these unhandled exceptions could have caused unpredictable behavior, such as a web form stuck in an infinite loop. This update adds appropriate error-handling around forms which operate on selected objects, with the result that operations on deleted objects are now handled gracefully by **Cumin**, which informs users in the event of an error.

BZ#631804

Running the **condor_router_history** utility with incorrect argument values could have caused a stack dump due to uncaught argument-handling exceptions. Argument parsing now includes proper exception-handling routines so that command usage errors are now handled gracefully, and helpful error messages are printed to the console when an argument is incorrect.

Enhancements

BZ#716466

Prior to this update, an **Amazon Machine Image** (AMI) could not have joined a **Virtual Private Cloud** (VPC). With this update, the necessary parameters allowing this feature have been added to the **Elastic Compute Cloud** (EC2) **Grid ASCII Helper Protocol** (GAHP), and AMIs now can join VPCs.

BZ#709713

With this update, support for the **Amazon Elastic Block Store** (EBS) has been added and **Condor** can now associate EBS volumes with *MRG/EC2 Enhanced instances*.

BZ#719050

Prior to this update, when adding multiple nodes to a group, each node had to be accessed individually because the **condor_configure_store** tool did not allow editing a group's membership directly. The **condor_configure_store** application has been modified to allow this configuration, and this process now requires editing only the respective group.

BZ#700774

Prior to this update, Condor jobs submitted using the **Condor Aviary** module included only a minimal set of attributes necessary to run jobs in most runtime environments defined by Condor. However, this set was not complete for all slot configurations. Consequently, jobs submitted by **Condor Aviary** did not match configuration pattern for dynamic slot provisioning and could not be scheduled unless the user explicitly added all needed attributes. *Aviary submissions* now implicitly include all required attributes for correct matching to dynamic slot provisioning, and such submitted jobs now run without additional user intervention.

BZ#692911

Previously, the **Wallaby** configuration service allowed users to specify parameter values beginning with the **>=** string in order to indicate that these values were to be appended to a comma-separated list of values when the feature was applied. Beginning with version 0.10.5-4, **Wallaby** also supports parameter values that begin with **&&=** and **||=** strings, which indicate lists of values separated by **&&** and **||** delimiters. This makes it possible to declare conjunctions or disjunctions of ClassAd expressions across multiple features.

BZ#652772

Redundant security-related parameters in several features have been removed and security parameter settings have been modified to be more clear. Only the *Master feature* now contains security related parameters. The ***SEC_DEFAULT_AUTHENTICATION_METHODS*** parameter on the

Master feature also includes the **FS** and **NTLM** methods. All security related parameters now have **must_change** and **needs_restart** variables set by default to **false**.

BZ#706108

Previously, the ***ALLOW_ADMINISTRATOR*** parameter in the *Master feature* was set to the **$(ALLOW_ADMINISTRATOR)** value in the default database, rendering it fully dependent on external configuration for remotely configured nodes. In case the external configuration did not include the **$(CONDOR_HOST)** in the ***ALLOW_ADMINISTRATOR*** parameter value, an affected node could have experienced issues while running and administrating jobs in the pool. With this update, the **$(CONDOR_HOST)** is now a part of the ***ALLOW_ADMINISTRATOR*** parameter and remotely configured nodes can be now administered by the central manager(s) without any issues.

BZ#632109

In older versions of the **wallaby inventory** tool, long hostnames were truncated to *25 characters* so that inventory data could fit in an 80-column terminal window. The current version of the **wallaby inventory** tool includes the **-l (--long)** option to ensure that hostnames are not truncated.

All users of the Grid capabilities of Red Hat Enterprise MRG 2.0 are advised to upgrade to these updated packages, which resolve the security issue, fix the bugs and add the enhancements noted in the Red Hat Enterprise MRG 2.0 Technical Notes.

6.6. RHEA-2011:0889 – Red Hat Enterprise MRG – Grid 2.0 Release

Red Hat Enterprise MRG is a next-generation IT infrastructure incorporating Messaging, Realtime, and Grid functionality. It offers increased performance, reliability, interoperability, and faster computing for enterprise customers.

MRG Grid provides high-throughput computing and enables enterprises to achieve higher peak computing capacity as well as improved infrastructure utilization by leveraging their existing technology to build high performance grids. MRG Grid provides a job-queuing mechanism, scheduling policy, and a priority scheme, as well as resource monitoring and resource management. Users submit their jobs to MRG Grid, where they are placed into a queue. MRG Grid then chooses when and where to run the jobs based upon a policy, carefully monitors their progress, and ultimately informs the user upon completion.

Bug Fixes

BZ#693782

The *grid-condor* package has been upgraded to upstream version 7.6, providing a number of bug fixes and enhancements.

BZ#553696

Prior to this update, CD image files could not be transferred. Now, CD images are treated like other disk images and are passed via the **vm_disk** utility. Images transfer and mount as expected.

BZ#563337

Prior to this update, vacate jobs were only sent **SIGTERM** signals via **condor_vacate_job**

when attempting to evict jobs. Due to this behavior, jobs that ignored the **SIGTERM** signal were not evicted. This update adds a signal escalation timer to the job vacate code path. Now, jobs are terminated with the **SIGKILL** signal if the **SIGTERM** signal was ignored.

BZ#580530

Prior to this update, mixed **-constraint** and restriction lists displayed only usage information but no error messages. Due to this, users could become confused. This update corrects the problem. Now mixed **-constraints** and restriction lists report an error message and display the usage information.

BZ#606391

Prior to this update, the **condor_triggerd**, **condor_job_server**, and Qpid Management Framework (QMF) plug-ins did not connect to the AMQP broker (**qpid**) because the broker was configured to restrict access. This update changes the configuration parameters to allow authentication information to be set for the authorization mechanism (**QMF_BROKER_AUTH_MECH** — **PLAIN** or **ANONYMOUS**), for the user to authenticate to the broker (**QMF_BROKER_USERNAME**), the location of the file that contains the broker password in clear text (**QMF_BROKER_PASSWORD_FILE**). Now, the above daemons and plug-ins connect to secured brokers as expected.

BZ#627957

Prior to this update, when a user attempted to use the **condor_configure_pool** utility to add a feature that included another feature without the **must_change** parameter specified, the utility did not prompt the user to supply the missing value. With this update, the **condor_configure_pool** utility uses a new API call to better detect **must_change** parameters. Now, the **condor_configure_pool** prompts the user in case of missing **must_change** parameters.

BZ#631782

Prior to this update, **condor_gridmanager** deleted uninitialized memory when ran as **root** or when passed the **-o** option. This update adds additional checks to avoid the improper deletion. Now, the **condor_gridmanager** behaves as expected.

BZ#634975

Prior to this update, Python 2.4 prevented the resolution of dangling references to QMF objects when references were received before the objects to which they referred. Due to this behavior, certain objects were not visible in the user interface if they contained a dangling reference to another object. This update corrects the programming error so that **Cumin** operates correctly. Now, dangling references are resolved whenever an object previously referenced is received by **Cumin**.

BZ#644302

Previously, the init script for the **cumin** service did not verify that the service is not already running. Under certain circumstances, this allowed multiple instances of **Cumin** to run simultaneously, and could lead to excessive memory consumption. This update adapts the **/etc/rc.d/init.d/cumin** init script to verify that the service is not running before starting a new instance. As a result, only one instance of the **cumin** service can be started.

BZ#644313

Prior to this update, when the **qpidd** broker suddenly disconnected, **Cumin** raised an exception and terminated unexpectedly with an error. This update adapts **Cumin** to handle such an exception. As a result, when the broker is stopped, **Cumin** no longer crashes and the user is presented with an informative error message.

BZ#647500

The **MRG Management Console** displays slot icons grouped by the slot's state and activity. Previously, the state of the slot determined the color of the icon, and the activity determined the icon's shape. To make this distinction more explicit, these slots are now displayed in four groups: **Busy**, **Transitioning**, **Owner**, and **Unclaimed**.

BZ#647758

Previously, when a user supplied an incorrect hostname, the **condor_reschedule**, **condor_vacate**, **condor_off**, **condor_on**, **condor_reconfig**, and **condor_restart** scripts incorrectly returned exit code **0**. With this update, the underlying source code has been adapted to address this issue. Attempts to use an incorrect hostname now cause these scripts to terminate with a non-zero exit code as expected.

BZ#647789

When a related server (such as **condor_master**, **condor_schedd**, or **condor_startd**) stopped unexpectedly, the **condor_reschedule**, **condor_vacate**, **condor_off**, **condor_on**, **condor_reconfig**, and **condor_restart** scripts incorrectly returned exit code **0**. With this update, these scripts have been adapted to return a non-zero exit code when a connection fails.

BZ#669023

When the **condor_q** utility was compiled without the PostgreSQL support, the output of the **condor_q -h** incorrectly listed **-avgqueuetime** as a valid command line option. Consequent to this, an attempt to run **condor_q** with this option failed with an error. This update adapts the utility not to list the **-avgqueuetime** option when it is compiled without the PostgreSQL support.

BZ#671451

Due to recent changes to the **libvirt** library, the previous version of **Condor** was unable to detect the type of an image, and only accepted raw images. This update allows users to specify the image type by using the **vm_disk** option as follows:

```
vm_disk = file_name:device:permissions:image_type
```

As a result, users can now use all image types supported by **KVM**.

BZ#672484

The **MRG Management Console** allows a user to select multiple records to work with as a group. Prior to this update, an automatic page reload caused all previously selected records to be deselected. This no longer occurs, and when a page is automatically reloaded, the selection is preserved as expected.

BZ#672517

Previously, when a user used the **condor_q** utility to analyze a job with a requirements expression containing ClassAd function calls that were evaluated as false, the utility failed. The following error message was written to standard error:

```
error: bad form
```

With this update, the implementation of the ClassAd mechanism has been adapted to prevent this error from occurring, and users are now able to analyze such jobs as expected.

BZ#672583

Prior to this update, the init script for the **Cumin** service sometimes incorrectly reported success even when the service failed to start or encountered an error. Additionally, any errors that occurred before **Cumin**'s logging mechanism was fully operational were only written to standard error. With this update, when an error occurs on service startup, the **/etc/rc.d/init.d/cumin** init script now reports **FAILED**, and detailed information is written to the **$CUMIN_HOME/log/master.log** file.

BZ#673502

Prior to this update, the system returned an unclear error message when the user ran the **wallaby feature-import** command without a file name parameter. The **wallaby feature-import** command now traps the error and displays a clearer error message when run without the parameter.

BZ#673520

Some **wallaby** utility subcommands caused wallaby to exit with an incorrect exit code. With this update, wallaby exits with the correct exit code that reflects the success or failure of the underlying operation.

BZ#673538

The **condor_negotiator** daemon generates statistics on every negotiation cycle. Prior to this update, only the statistics generated for the cycle defined in the ***NEGOTIATOR_UPDATE_INTERVAL*** parameter were propagated to the **condor_collector** daemon. As a consequence, the system sometimes presented outdated data to the user during other cycles. This update adds the ***NEGOTIATOR_UPDATE_AFTER_CYCLE*** parameter. If the parameter is set to **true**, negotiator propagates the statistics to the **condor_collector** daemon after every cycle.

BZ#673592

When the configuration of the **condor_negotiator** daemon was changed or the daemon was restarted, the entries of accounting groups and submitter names were reset to zero. With this update, the entry values are preserved in such circumstances.

BZ#674432

Prior to this update, the **rhubarb** library returned a syntax error when the user created a persisting class with a name identical to an SQL reserved word. The **rhubarb** library now quotes table names in all generated SQL code, and the user can declare persisting classes

with such names.

BZ#674433

Prior to this update, the system returned an error when the user created a persisting class with no declared column. With this update, the necessary initialization steps take place even if no columns are declared and such class is created successfully.

BZ#674630

Previously, help pages for the **condor_configure_pool** and **condor_configure_store** tools listed the username option wrongly as **-u**. This update corrects the error. Now, **--help** contains correctly **-U**.

BZ#675209

Prior to this update, the **triggerd** daemon was not listed as one of the **DaemonCore** daemons by default. The problem has been fixed by adding the **DC_DAEMON_LIST = >= TRIGGERD** value to the **TriggerService** feature.

BZ#675703

Due to case folding of submitted names in the **Negotiator** daemon, multiple submitter name entries were created in **condor_userprio** (one entirely in lower case and one with the correct mix of upper and lower case characters) if the submitter name entry contained upper case letters. Explicit case folding is now removed from **Negotiator** and data maps are updated with a case-insensitive sorting function. As a result, submitter names with upper case letters no longer appear as multiple entries and the accounting group entries now match updated entries by case and full submitted entries are case sensitive.

BZ#675725

Prior to this update, **Condor**'s restart action unconditionally started and ignored errors emitted by **Condor**'s stop action. As a consequence, it was difficult to use **Condor**'s SysV init script properly. The problem has been fixed so that the init script now skips a start attempt and issues an error if the stop action fails during the restart action.

BZ#675935

Prior to this update, the default value for the **CONDOR_HOST** option was set incorrectly in the default database for the **HACentralManager** feature. Due to this problem, users had to explicitly set **CONDOR_HOST** when they enabled the **HACentralManager** feature on a node without the tools asking for a value. With this update, the tools prompt for the value for **CONDOR_HOST** when the **HACentralManager** feature is set. Now, the **HACentralManager** feature asks to set **CONDOR_HOST** when enabled on a node/group.

BZ#675967

Prior to this update, the **schedd** daemon incorrectly set permissions of the job files. As a result, **Condor** failed to transfer back output files for jobs, which were run by unknown users who were allowed to submit jobs. The problem has been resolved in this update so that the **schedd** daemon now sets permissions of the job files to user **nobody**, and **Condor** now correctly transfers the output files back for jobs run by unknown users.

BZ#676411

Prior to this update, if the **condor_trigger_config** command was executed and failed, it incorrectly returned a success return code. The problem has been fixed so that the **condor_trigger_config** command now returns a correct failure return code in case of failure.

BZ#676902

Prior to this update, static resources in **Cumin** used the **If-Modified-Since** request-header field but did not set a cache expiration date. As a result, static content was not transferred for resources which were cached by the browser but there was still a round trip communication to the server when a resource was loaded. This overhead was unnecessary and negatively affected **Cumin**'s performance. The problem has been resolved by adding the **Cache-Control: max-age** header to all static resources served by **Cumin** so that the number of pages per second that can be served by **Cumin** is now significantly improved.

BZ#677398

Prior to this update, the size of **Cumin**'s log file was unlimited. As a consequence, when a log file became too large, it could have affected performance or caused exceptions in the **Cumin** application. The problem has been resolved in this update so that **Cumin** now provides a log file rotation facility, which prevents the performance degradation and alleviates the need for manual log file rotation. The default maximum size for any log file is 10MB. The log file rotation is configurable and is covered in the Management Console Installation Guide.

BZ#677807

Prior to this update, the **rhubarb** library supported serializing arbitrary Ruby objects as members of persistent objects but these fields could have been only assigned to one at a time. This unexpected and undesirable behavior for clients of **rhubarb** other than **Wallaby** has been corrected in this update so that object-valued fields can now be updated via the assignment operator, just like other database-backed fields of persisting objects; and object-valued fields can be updated at an arbitrary number of times.

BZ#678590

Previously, the negotiator loop did not explicitly check to determine when a group reached its quota limit. This could cause unnecessary iterations of the negotiator loop, and eventual termination at each submitter based on individual submitter limits. With this update, an explicit check has been added to the negotiator loop so that the loop will halt as soon as it detects that the group has reached its quota, and the unnecessary iterations no longer occur.

BZ#678621

If a **cumin** application such as **cumin-admin** or **cumin-web** was run as the **root** user, and the **$CUMIN_HOME/log** file was created, the file is locked from being written to by a regular **cumin** user. Subsequently, a **cumin** application terminated if it attempted to write to the file. Now, the file ownership is always changed to match the ownership of the containing directory and the bug no longer occurs.

BZ#679672

Previously, the **Max Allowance** limit input for a submission did not allow a floating-point value, although such value is valid. Subsequently, the **cumin** daemon rejected a limit change to such

a value with the following error message:

```
The Max Allowance field must be an integer
```

This bug has been fixed and both integer or floating-point values are now accepted for **Max Allowance** limits.

BZ#679688

Due to non-standard labeling across the MRG Grid 2.0 user interface, the interface looked inconsistent and user-unfriendly. With this update, the **Add Binding** has been fixed to reflect correct case used in other labels and the user interface now looks more consistent.

BZ#680265

When the **cumin** daemon was started from a window and that window was subsequently closed, program errors were generated if **cumin** wrote output to standard error output or standard output. This bug has been fixed, any extraneous output from the **cumin** daemon are now directed safely to /dev/null or designated log files, and the daemon will continue to run as a service if the console window is closed.

BZ#680434

Previously, upgrading *wallaby* or *condor-wallaby* packages using the remote configuration feature caused the **condor_configd** daemon to fail to communicate with the configuration store and error messages that the store was using an unsupported API version were given. With this update, the API version check has been removed from the **condor_configd** daemon and the API version mismatch bug no longer occurs.

BZ#680437

Previously, the unsupported **/usr/sbin/condor_vm_vmware.pl** file was included in the *condor-vm-gahp* package. With this update, the file has been removed.

BZ#681124

During boot on some systems, the *cumin* database server was not fully functional by the time the **cumin** daemon was started and checked the state of the database. As a consequence, the daemon sometimes reported that the database has not been created, when in fact it has. This bug has been fixed and the **cumin** daemon now detects that the database server process is running, if it cannot make a connection to the server. Then, it will retry the connection for up to 30 seconds before reporting an error.

BZ#688285

Prior to this update, the EC2 Enhanced feature relied on the Amazon GAHP (Grid ASCII Helper Protocol), which has been deprecated, and replaced by the EC2 GAHP. With this update, the **set_gridresource** line has been removed from the example routes, and the translate hook now detects which GAHP to use; EC2 Enhanced works as expected.

BZ#691969

The **condor_startd** daemon could fail to remove dynamic slots that had been in the **OWNER** state after a job, that was executed on those slots, was completed. With this update, the

condor_startd daemon now properly removes dynamic slots after their jobs are completed.

BZ#692635

On Windows, an incorrectly configured ***QMF_BROKER_HOST*** parameter in **Condor**'s configuration would cause the **condor_configd** daemon to be unable to contact the incorrectly configured broker. Additionally, **condor_configd** would not respond to shutdown commands. With this update, the underlying source code has been modified to address this issue, and the **condor_configd** daemon properly shuts down, regardless of whether the ***QMF_BROKER_HOST*** parameter is configured correctly.

BZ#692653

The MRG **Cumin** web application displayed an error page when the last job for a submission was removed. With this update, this issue has been fixed, and the web application now displays the submission list page.

BZ#692741

Because a ***proc*** entry appeared before the ***cluster*** entry in the job queue log, there was no way to update the internal ***SubmissionObject*** once the **ATTR_OWNER** flag was set (from the log) before the submission name. With this update, the internal job object updates its associated ***SubmissionObject*** with an *owner* or *name* value, regardless of order. As a result, submission names and owner data appear correctly in all QMF and Aviary queries.

BZ#694835

Providing invalid arguments, or other invalid scenarios, to the **condor_configure_pool** and **condor_configure_tool** tools returned an error even though the return value indicated a success. With this update, the aforementioned tools properly exit with a non-zero exit code for any error cases.

BZ#695722

Clicking through the path, **Grid → Schedulers → (select a scheduler) → Submissions → (select a submission)**, would result in a page that would correctly display the desired information, but would lack a link back to the selected scheduler frame. With this update, a ***SubmissionFrame*** was added to the ***SchedulerFrame***. In addition, the ***PoolSubmissionSelector*** constructor was changed to allow for passing of the frame name (defaulted to the original value of ***main.grid.submission***) to control the target of the links generated in the ***PoolSubmissionSelector***. As a result, the breadcrumb at the top of the screen now includes a link to the currently selected scheduler when looking at a given submission.

BZ#695800

In the management console, an exception could occur if there was a mismatch in the group names and the dynamic group quota values. With this update, an informative message is displayed indicating which configuration entry is missing.

BZ#699571

Prior to this update, the internal ***uptime*** statistic was reset to **0** before being reported, causing the ***MonitorSelfAge*** parameter to always display the value **0**. With this update, the internal

uptime statistic is no longer reset to **0** before being reported, and the ***MonitorSelfAge*** parameter properly reflects the ***uptime*** statistic.

Enhancements

BZ#584562

Prior to this update, **condor_dagman** calls to **condor_submit** could place a severe and unnecessary load on the collector while each submit looked up the shedd address for the submission. With this update, the **-schedd-daemon-ad-file** and **-schedd-address-file** flags are now added to **condor_submit_dag** to allow targeting a directed acyclic graph (DAG) to a specific schedule daemon (**Schedd**) to bind all its operations to **Schedd**.

BZ#602766

The **condor_triggerd** daemon is now able to detect absent nodes when used in connection with the remote configuration feature. Now, the **condor_triggerd** daemon raises an event for each node that is configured in **wallaby** for which a master qmf object is not detected if absent node detection is enabled.

BZ#610251

Prior to this update, **Condor** allowed declaration of only a single view server. Due to this limit, multiplexing among multiple view servers was not possible. This update adds support of multiple view servers declared on **CONDOR_VIEW_HOST**. Now, **Condor** is able to declare multiple view servers to allow multiplexing among these servers for improved scale.

BZ#610258

Prior to this update, the type of Classified Advertisements (classads) to be forwarded to a **CONDOR_VIEW_HOST** could not be selected. This update adds the parameter **CONDOR_VIEW_CLASSAD_TYPES**. Now, administrators can control the type of classads that are forwarded to **CONDOR_VIEW_HOST**.

BZ#621899

Prior to this update, Condor EC2 jobs could not be bound to an elastic IP. Due to this behavior, a dynamic IP had to be created for each instance. This update uses the **ec2_elastic_ip** parameter to support elastic IP binding for Condor EC2 jobs.

BZ#630544

Prior to this update, the version information for **Cumin** could not be viewed from the web user interface. Due to this lack, users needed to log into the server host and use **rpm** commands to view the installed package information. With this update, the **Cumin** user interface has an **About the console** tab under the **Your Account** page where version information stored in **$CUMIN_HOME/version** displays.

BZ#635197

Prior to this update, the values in the **Max Allowance** column in **Cumin** were incorrectly displayed as integers instead of floating-point numbers. Additionally, very large values were displayed in their explicit form instead of being replaced with the **Unlimited** label. With this

update, the **Max Allowance** values are correctly displayed as floating-point numbers, and values larger than 1,000,000 are now rendered as **Unlimited** as expected.

BZ#642405

The **MRG Management Console** has been updated to pull data displayed in the **Limits**, **Quotas**, and **Job Summaries** tables directly from the broker rather than pulling them from the internal database. Additionally, these tables can now be exported in the comma-separated values (CSV) file format.

BZ#668038

Prior to this update, when runtime reconfiguration was enabled, an authorized user could cause a daemon to terminate unexpectedly by providing a faulty configuration. This happened, because neither the **condor_config_val -set** (or **condor_config_val -rset**) command, nor the daemon being reconfigured would validate the input. With this update, the underlying source code has been adapted to make sure that both the **condor_config_val** utility and the daemons validate the configuration provided, preventing crashes during runtime.

BZ#673178

The Red Hat Enterprise MRG console sorts data in ascending or descending order when the user clicks the relevant column header. However, the GUI (graphical user interface) did not indicate which columns were sorted and in what order. Now, an arrow placed on the selected column header indicates the sorting order and a pop-up tooltip indicates the column sorting order, when the mouse cursor hovers over the column header.

BZ#673180

Prior to this update, the MRG console displayed some tables that receive their data directly from the broker but were unable to search for the desired records. Tables that receive their data directly from the broker now have the ability to search for specific records.

BZ#673183

Some actions in the MRG console, such as displaying the job summary info and the group quotas, retrieve their data directly from the broker. This process can take a few seconds. Prior to this update, the system did not display any feedback to inform the user that the action was pending. The MRG console now displays a message informing the user that data is being loaded while waiting for broker response.

BZ#673187

Cumin displays data in tables that display 100 records per page. When more than 100 records are present in a table, the user could not save all the records to a file. **Cumin** now allows a user to save all records in a table to a comma separated value file.

BZ#673189

Prior to this update, the MRG **Cumin** console presented a list of pools under the **Grid** tab. Generally, only one pool is displayed under the **Grid** tab and a dedicated page to display a list containing one entry is thus unnecessary. The MRG **Cumin** console now does not display the list of pools and if more than one broker is listed in the **brokers=** line of the **Cumin** configuration file, the first broker is used as a default.

BZ#673194

An overview page was added to show the overall health of the grid and provide access to various grid statistics at a glance.

BZ#674161

Condor now supports hibernating machines that are in the idle state.

BZ#674349

Grid now offers a simpler web interface called Aviary, created with the use of Axis2/C and WSO2.

BZ#674659

Condor now includes the **PreJobPrio1**, **PreJobPrio2**, **PostJobPrio1**, and **PostJobPrio2** job ad attributes, which allow jobs to be ordered outside the **JobPrio** attribute.

BZ#674669

Prior to this update, the **LastNegotiationCycleSubmittersShareLimit*N*** **negotiator** **classad** ***stat*** attribute did not account for a submitter reaching the share limits in a ***group-quote*** scenario. The negotiator now includes submitter names in the attribute when any submitter reaches the submitter limit, including group quota limits.

BZ#678025

With this update, the new statistics published in the scheduler's classads provide more detailed information and allow a better assessment of how well the scheduler is performing in the **Condor** pool.

BZ#678029

The *persona* feature has been added to the web chapter of the **cumin** daemon configuration file. The new feature allows users to choose a grid-only or messaging-only views. The customized views serve as alternates to the default view, which incorporates both grid and messaging views.

BZ#678394

The *Power Management* feature has been added to the MRG Grid 2.0 and can be configured manually via the **wallaby** component. Users can configure *Power Management* through the remote configuration feature.

BZ#679553

Previously, **Condor** used AWS EC2's SOAP interfaces for managing virtual instances within EC2. With this update, **Condor** now uses AWS EC2's Query API.

BZ#680260

With this update, the new **export-users** and **import-users** commands have been added to the **cumin-admin** utility. It is now possible to save and restore user data when a *cumin*

database is reinitialized. It is also possible to export user data from one *cumin* database and import it into another *cumin* database.

BZ#680518

Previously, when a reconfigure signal from MRG Grid or a SIGHUP signal from the command line was sent to the **condor_configd** daemon, the daemon terminated unexpectedly. With this update, support to handle SIGHUP signals has been added to the **condor_configd** daemon and the daemon now terminates properly in the described scenario.

BZ#690283

To maintain performance as scale increases, **Cumin** now distributes data processing across multiple instances of **cumin-data**. Responsibility for data processing is partitioned at the level of QMF classes. Without this enhancement, users could notice decreases in **Cumin** performance as scale increases.

BZ#694857

With this update, a new option has been added to the **condor_trigger_config** utility to allow communication through a secured broker. As a result, it is now possible to configure the **triggerd** daemon through a secured broker.

BZ#692169

Certain messaging configuration or installation changes could unexpectedly cause **Cumin** to authenticate to a broker using the *ANONYMOUS* method when password authentication was intended. This was because **Cumin** did not have a mechanism for disallowing the use of the *ANONYMOUS* method. If **Cumin** authenticates using the *ANONYMOUS* method, certain features, such as job submission, will not be available from the user interface. This update adds the ***sasl-mech-list*** configuration parameter to the **[common]** chapter in the **/etc/cumin/cumin.conf** file. This parameter is a space separated list of allowable SASL authentication mechanisms. Names of the available mechanisms are specified in the SASL documentation (for example, *PLAIN* and *ANONYMOUS*). By default, all available mechanisms are allowed. As a result, ***sasl-mech-list*** parameter can be configured to restrict the allowable configuration mechanisms for **Cumin**.

All users are advised to upgrade to these updated packages, which fix these bugs and add these enhancements.

Revision History

Revision 4-2 **Tue Mar 12 2013** **Cheryn Tan**
Added advisory RHSA-2013-0622.

Revision 4-1 **Mon Mar 4 2013** **Cheryn Tan**
Prepared for publishing (MRG 2.3).

Revision 4-0 **Fri Mar 1 2013** **Cheryn Tan**
Added content for MRG Grid 2.3 update (RHSA-2013-0565 and RHSA-2013-0564).

Revision 3-1 **Thu Feb 28 2013** **Joshua Wulf**
Added content for MRG Messaging 2.3 update (RHSA-2013-0561 and RHSA-2013-0562).

Revision 3-0 **Tue Feb 26 2013** **Cheryn Tan**
Added content for MRG Realtime 2.3 update (RHBA-2013-0563 and RHSA-2013-0566).

Revision 2-9 **Wed Nov 28 2012** **Cheryn Tan**
Added content for MRG Realtime 2.2.2 update (RHBA-2012-1492 and RHSA-2012-1491).

Revision 2-8.1 **Tue Nov 13 2012** **Tomáš Čapek**
Fixed a couple of errata URLs.

Revision 2-8 **Mon Sep 17 2012** **Tomáš Čapek**
Content for the MRG 2.2 release added.

Revision 2-6 **Mon Apr 30 2012** **Tomáš Čapek**
Added errata for a major Messaging update.
Sorted errata chronologically, cleaned up duplicate content.

Revision 2-5 **Tue Feb 28 2012** **Tim Hildred**
Updated configuration file for new publication tool.

Revision 2-4 **Mon Feb 6 2012** **Tomáš Čapek**
Added RHSA-2012:0099 and RHSA-2012:0100.

Revision 2-3 **Mon Jan 23 2012** **Tomáš Čapek**
Added RHBA-2012:0046, RHSA-2012:0044, and RHBA-2012:0045.

Revision 2-2 **Thu Dec 08 2011** **Tomáš Čapek**
Added content for RHBA-2011-1399, RHBA-2011-1370, and RHBA-2011-1393.
Added descriptions for MRG 2.1 release.

Revision 2-1 **Fri Oct 07 2011** **Douglas Silas**
Reformat to improved Technical Notes style.
Add 2011:1339 and 2011:1340 Red Hat Enterprise MRG Messaging 2.0 bug fix updates.