Red Hat JBoss Data Grid 6.2

6.2.1 Release Notes

Known and resolved issues for Red Hat JBoss Data Grid 6.2.1

Edition 1

Last Updated: 2017-11-21

Red Hat JBoss Data Grid 6.2 6.2.1 Release Notes

Known and resolved issues for Red Hat JBoss Data Grid 6.2.1
Edition 1

Gemma Sheldon
Red Hat Engineering Content Services
gsheldon@redhat.com

Legal Notice

Abstract

The Red Hat JBoss Data Grid 6.2 Release Notes list and provide descriptions for a series of bugzilla bugs. The bugs highlight either issues that are known problems for the relevant release or bugs that have now been resolved.

Table of Contents

CHAPTER 1. INTRODUCTION TO RED HAT JBOSS DATA GRID 6.2.1

Welcome to the Red Hat JBoss Data Grid 6.2.1. As you become familiar with the newest version of JBoss Data Grid, these Release Notes provide you with information about resolved issues. Use this document in conjunction with the entire JBoss Data Grid 6.2.1 documentation suite, available at the Red Hat Customer Service Portal's JBoss Data Grid page.

1.1. ABOUT RED HAT JBOSS DATA GRID

Red Hat's JBoss Data Grid is an open source, distributed, in-memory key/value data store built from the Infinispan open source software project. Whether deployed in client/server mode or embedded in a Java Virtual Machine, it is built to be elastic, high performance, highly available and to scale linearly.

JBoss Data Grid is accessible for both Java and Non-Java clients. Using JBoss Data Grid, data is distributed and replicated across a manageable cluster of nodes, optionally written to disk and easily accessible using the REST, Memcached and Hot Rod protocols, or directly in process through a traditional Java Map API.

1.2. OVERVIEW

This document contains information about the resolved issues in Red Hat JBoss Data Grid version 6.2.1. Customers are requested to read this documentation prior to installing this version.

Customers can provide feedback or log bugs through their Support account.

CHAPTER 2. SUPPORTED CONFIGURATIONS

For supported hardware and software configurations, see the Red Hat JBoss Data Grid Supported Configurations reference on the Customer Portal at https://access.redhat.com/site/articles/115883.

CHAPTER 3. COMPONENT VERSIONS

The full list of component versions used in Red Hat JBoss Data Grid is available at the Customer Portal at https://access.redhat.com/site/articles/488833.

JBoss Data Grid 6.2.1 uses the same components as JBoss Data Grid 6.2.

CHAPTER 4. KNOWN ISSUES

The following issues are known to exist in Red Hat JBoss Data Grid 6.2.1 and will be fixed in a subsequent release.

BZ#881080 **- Silence SuspectExceptions**

SuspectExceptions are raised when nodes are shutting down, which is normal and expected, since a suspect node is an unresponsive node. As a result, a SuspectException ERROR is written in the logs. There is currently no fix for this issue. The SuspectExceptions will not be logged in a future version of Red Hat JBoss Data Grid. The SuspectExceptions do not affect data integrity.

BZ#881791 **- Special characters in file path to JDG server are causing problems**

Using special characters in directory paths can cause problems when starting Red Hat JBoss Data Grid server. The JBoss Data Grid server either fails to start or a configuration file for logging capabilities cannot be loaded properly. Special characters include spaces, # (hash sign), ! (exclamation mark), % (percentage sign), $ (dollar sign). This bug causes the JBoss Data Grid server to fail to start properly.Avoiding the use of special characters will stop this problem occurring, allowing JBoss Data Grid server to start properly.

BZ#1012036 **- RELAY2 logs error when site unreachable**

When a site is unreachable, JGroups's RELAY2 logs error for each dropped message. Infinispan has configurable fail policies (ignore/warn/abort), and even with ignore policy the log is filled with errors.

BZ#1013001 **- [ISPN-3375] Map/Reduce jobs with small value sizes fail because of timeouts putting intermediate keys into the cache**

When a Map/Reduce task is executed against a cache with many small values, the task may experience TimeoutExceptions while trying to write the intermediate results to the cache.As a result, when these exceptions occur, the task fails.

There are several things the user can do to try and alleviate this issue:

1. Use larger values in the cache where the Map/Reduce task is executed.
2. Use a collator and/or a combiner on the Map/Reduce task if this is possible.
3. Configure the size of the intermediate chunks written to the cache using the state transfer chunkSize parameter.

BZ#1043434 **- L1 entry added by ST when already invalidated**

In distributed cache mode with L1 enabled and **`onRehash=true`**, when an entry is overwritten during state transfer, the overwrite may be ignored. This node will then report an outdated value of the entry, and after L1 timeout this entry will be completely removed.

As a workaround, use L1 with **`onRehash=false`**.

CHAPTER 5. RESOLVED ISSUES

The following issues have been resolved in Red Hat JBoss Data Grid 6.2.1.

BZ#1043853 **- Concurrent message headers modification causes that message is never sent**

Sending a message through JGroups may fail under some race conditions with ArrayOutOfBoundException. The reliability mechanism will try to resend this message again, but all these subsequent attempts will fail with NullPointerException.

Therefore, the message will be never received on the receiver node and will be held in unacknowledged messages buffer on the sender node. If this message is ordered within JGroups, no more ordered messages from this node will be delivered on the receiver node. Some following messages may be buffered as well, eventually leading to OutOfMemoryException.

It is recommended to use UNICAST3 as this dramatically reduces the chance of this bug to happen (it was observed with UNICAST2).

BZ#1047905 **- REPEATABLE_READ not working for AtomicMap and FineGrainedAtomicMap**

When REPEATABLE_READ isolation mode is enabled, reading values from AtomicMap and FineGrainedAtomicMap is not consistent within the same transaction. If another thread changes the value during the transaction, the new value is reflected immediately in the current thread while the thread should still see the original value.

BZ#1047908 **- WriteSkew not throwing an exception for AtomicMap in REPL and DIST mode**

The writeSkewCheck flag does not work correctly for AtomicMap. When JBoss Data Grid runs in REPEATABLE_READ isolation mode for transactions, the writeSkewCheck flag is enabled, and two concurrent transactions read/write data from the same AtomicMap. The WriteSkewException exception is never thrown when committing either of the transactions.

Consequently, the data stored in the AtomicMap might be changed by another transaction, and current transaction will not register this change during commit.

BZ#1050007 **- Fallback to non-preferred IP stacks if host can't be resolved with the preferred one**

By default, the C++ Hot Rod client resolves server host names using IPv4 addresses. If a server is listening on an IPv6 address, the client will not be able to connect to it. Either specify the server using its IP instead of the name or set the HOTROD_IPSTACK environment variable to IPV4 or IPV6 to force resolution using the preferred stack. The client will connect to the specified server.

BZ#1058887 **- HotRod client keep trying recover connections to a failed cluster for a long time**

If a cluster is no longer reachable for some reason, i.e. network disconnect, the hot-rod client tries to re-establish the lost connections. The client library will retry this by a fixed calculation based on the max numbers of connections from the pool, or 10 multiplied with the number of available servers. This may result in a long delay until the application can continue and react, as it will wait for the read timeout for each try.

This has been fixed by adding a new configuration property infinispan.client.hotrod.max_retries. This property defines the maximum number of retries in case of a recoverable error. A valid value should be greater or equal to 0 (zero). Zero means no retry. Default is 10.

BZ#1059277 **- HotRod client keep trying recover a connection to a cluster member after shutdown**

If a Hot Rod client is connected to a server cluster and one of the servers is shutdown gracefully, the client keeps trying to reconnect to the server. This occurs in replicated mode. There is no failure from the application perspective but there is always a retry if the RoundRobin strategy

returns the unavailable server. This might have impact on performance or even cause a failure if there is a larger cluster and a part is going out of work for some reason.

BZ#1059489 - Preload with async cache store is not efficient

Configuring an asynchronous cache store with 'preload' enabled leads to pre-loading each entry in the store in a loop, each entry being loaded through a store read. This has been fixed. Pre-loading entries from a cache store is now more efficient and does not require loading each entry through a store read.

BZ#1059814 - L1WriteSynchronizer can cause thread to hang

L1 cache in distributed mode with concurrent gets originating from the same node for the same key can result in one being stuck if there is no value found. This can cause massive degradation of performance as the request would never complete.

BZ#1060311 - CACHE_MODE_LOCAL flag only works in primary owner for non-tx caches

The CACHE_MODE_LOCAL flag does not force the EntryWrappingInterceptor to wrap the entry. As a result, the entry will not be stored in the non-transactional cache if the current node is not the primary owner of the entry. This has been fixed for replicated mode. In distributed mode, such an operation stores the entry in L1 cache if it is enabled. If L1 cache is disabled and current node is not the primary owner of the entry, then this is a no-operation.

BZ#1066577 - Pre-signed URLs in the S3_PING protocol do not work with JGroups 3.4.1

The format of pre-signed URLs has changed on Amazon Web Services. All pre-signed URLs must now include the bucket name in the hostname. As a result, Red Hat JBoss Data Grid does not work correctly on Amazon Web Services when pre-signed URLs are used. To fix this, change the URL parsing code in S3_PING JGroups protocol to allow S3_PING to work in AWS. As a result, JBoss Data Grid works correctly on Amazon Web Services when pre-signed URLs are used.

BZ#1073086 - Infinispan directory server module is missing some dependecies

Configuring an indexed cache with 'infinispan' as a directory provider for Lucene will result in an exception at server startup due to missing dependencies. It is not possible to use 'infinispan' as a directory provider for Lucene with Remote Query. Dependencies have been added to the server modules which corrected the functionality.

BZ#1073098 - org.infinispan.query.dsl.Query.getResultSize() gives wrong results when used in remote mode

org.infinispan.query.dsl.Query.getResultSize() was giving wrong results when used in remote mode as a Remote Query.

BZ#1073327 - Stale locks during state transfer in non-tx caches

SingleKeyNonTxInvocationContext.addLockedKey() only sets a isLocked flag, the actual key is set when the entry is wrapped and inserted in the context. If the topology changes between the lock acquisition and the entry wrapping, SingleKeyNonTxInvocationContext.getLockedKeys() will return an empty set and the lock will not be released. Future commands will not try to acquire a lock on this node, so the stale lock is harmless most of the time. However, if there was already a command waiting for the lock, that command will time out (instead of retrying on the new primary owner).

To fix this issue, set the actual key and isLocked flag on SingleKeyNonTxInvocationContext at the same time. Furthermore, prevent from incorrect returning the empty set of keys so that keys can be properly unlocked. As a result, locks are properly released during state transfer and commands do not time out due to locks being held.

BZ#1073331 - Deadlock in RemoteCache getAsync

Future object returned from an async operation on RemoteCache (such as getAsync(key)) allows it to register a listener for the operation completion. When the operation executes very quickly, there is a chance that the client code did not registered the listener yet. In that case, the listener is not called at all. As a result, when the listener is executed (the situation above does not happen), calling get() on the future returned from getAsync() causes a deadlock, the call does not return and the thread executing the listener is blocked forever.

BZ#1073332 - Preloading and Shared options can not be enabled on a JDBCCachestore via configuration files

The configuration schema for library mode of Red Hat JBoss Data Grid does not allow to you enable the "preload" and "shared" attributes of a JDBCCacheStore. As a result, JDBCCacheStore can not be configured to preload data from a relational database on startup or to be used as a shared cache store. This only applies to caches being configured through XML configuration files. The respective functionality is available when JBoss Data Grid is configured through Java API.

The configuration file for library mode is now parsed correctly so that "preload" and "shared" attributes are properly set.

BZ#1073334 - SingleFileStore.FileEntry.compareTo is wrong

The implementation of FileEntry.compareTo() is incorrect in that it never returns 0. As a result, entries are not properly removed from the list of removed/expired entries which is held by the SingleFileStore. Such entries are supposed to be re-used and their space on file system used for newly created entries. Due to this bug, newly created entries are always appended at the end of the file, not freeing up the space occupied by entries removed/expired earlier.

BZ#1073482 - Server should include a jgroups configuration stack for S3 PING

The server configuration files now contain an additional JGroups stack that makes it possible to run Red Hat JBoss Data Grid on Amazon Web Services as a cluster. The stack is called "s3" and includes S3_PING JGroups protocol for discovering new nodes in the cluster.

BZ#1075649 - HRCPP-126: Segfault with cluster and connectionPool with minIdle > 0

When RemoteCacheManager is configured with connectionPool and minIdle attribute greater than zero, and there are multiple instances of JDG server which are clustered, a segmentation fault (Segfault) is raised during runtime. This has been fixed and the segmentation fault no longer occurs.

BZ#1075719 - JDBC store properties are ignore or cause an exception

Setting custom JDBC store properties causes an exception to be thrown. This has been fixed and it is now possible to configure JDBC cache store with custom properties such as databaseType.

BZ#1076042 - (6.2.1) JON server plugin: generic loader cache child creation fails

Configuration of a generic cache loader from JBoss ON fails with exception "org.infinispan.persistence.cluster.ClusterLoader is not a valid cache loader". This is caused by the fact that the configuration parser incorrectly casts every custom cache loader class to an older cache loader API (pre-JBoss Data Grid 6.2 API). This has been fixed and it is now possible to configure a generic cache loader from JBoss ON.

BZ#1076047 - (6.2.1) JON server plugin: generic store cache child creation fails

Configuration of a generic cache store from JBoss ON fails. Generic cache store have the limitation that the ConfigurationBuilder for the store must be in the same Java package as the store itself. This has been fixed and it is now possible to configure a generic cache store from JBoss ON, even if

the ConfigurationBuilder is in a different package.

BZ#1077127 - **Protobuf marshalling on Indexed server caches interferes with keySet and bulk marshalling**

The method keySet() of remote client fails on distributed caches with indexing enabled. This is caused but the fact that a ClassCastException is thrown in the map-reduce task that collects the keys across the cluster. The task expects the cached value to by byte[], which is not the case if protobuf marshalling is used. The value is never used by the task, so avoiding the unnecessary casting also avoids this issue.

APPENDIX A. REVISION HISTORY

Revision 6.2.1-7 **Wed Apr 02 2014** **Misha Husnain Ali**
Removed "new features" from document subtitle.

Revision 6.2.1-6 **Tue Apr 01 2014** **Gemma Sheldon**
Corrected Bugzilla product name in 2.2. Give us Feedback.
Corrected wording inconsistency in Resolved Issues bugs.

Revision 6.2.1-5 **Thu Mar 27 2014** **Gemma Sheldon**
Updated bug list for 6.2.1. Known Issues now included.

Revision 6.2.1-4 **Mon Mar 24 2014** **Gemma Sheldon**
Updated with two bugs.

Revision 6.2.1-3 **Fri Mar 21 2014** **Gemma Sheldon**
These release notes will detail only the bugs that have been resolved in this release.
Updated version number throughout What's New section.

Revision 6.2.1-2 **Wed Mar 12 2014** **Gemma Sheldon**
Updated with 2 Known Issues.

Revision 6.2.1-1 **Mon Mar 10 2014** **Gemma Sheldon**
Removed New Features section for minor release.

Revision 6.2.1-0 **Fri Mar 07 2014** **Gemma Sheldon**
First brew of 6.2.1 Release Notes.