Red Hat JBoss Data Grid 6

6.0.0 Release Notes

Release Notes for Red Hat JBoss Data Grid 6

Edition 1

Red Hat JBoss Data Grid 6 6.0.0 Release Notes

Release Notes for Red Hat JBoss Data Grid 6
Edition 1

Misha Husnain Ali
Red Hat Engineering Content Services
mhusnain@redhat.com

Legal Notice

Abstract

The JBoss Data Grid 6.0 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. ABOUT JBOSS DATA GRID 6

Red Hat's JBoss Data Grid is an open source, distributed, in-memory key/value datastore built from the Infinispan open source software project. Whether deployed in client/server mode or embedded in a Java VM, it is built to be elastic, performant, 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.

CHAPTER 2. KNOWN ISSUES

The following issues are known to exist in JBoss Data Grid 6 and will be fixed in a subsequent release.

BZ#745865 **- ServerWorker thread naming problem**

If a server is used for both Hot Rod and Memcached, it is not possible to distinguish between the worker threads for each protocol because they are all named "MemcachedServerWorker". This does not affect the functionality of the server.

This behavior persists in JBoss Data Grid 6.

BZ#760895 **- Reopened: Error detecting crashed member during shutdown of EDG 6.0.0.Beta**

Occasionally, when shutting down nodes in a cluster, the following message is reported: "ERROR [org.infinispan.server.hotrod.HotRodServer] ISPN006002: Error detecting crashed member: java.lang.IllegalStateException: Cache '___hotRodTopologyCache' is in 'STOPPING' state and this is an invocation not belonging to an on-going transaction, so it does not accept new invocations. Either restart it or recreate the cache container."

This is due to the fact that a node has detected another node's shutdown and is attempting to update the topology cache while itself is also shutting down. The message is harmless, and it will be removed in a future release

This behavior persists in JBoss Data Grid 6.

BZ#807674 **- JDBC Cache Stores using a JTA Data Source do not participate in cache transactions**

In JBoss Data Grid's library mode, JDBC cache stores can be configured to use a JTA-aware datasource. However operations performed on a cache backed by such a store during a JTA transaction, will be persisted to the store outside of the transaction's scope. This issue is not applicable to JBoss Data Grid's Remote Client-Server mode because all cache operations are non-transactional.

This behavior persists in JBoss Data Grid 6.

BZ#807741 **- Reopened: Invalid magic number**

The issue is a race condition on the Hot Rod server, which can lead to topologies being sent erroneously as a result of addition of a new node to the cluster. When the issue appears, clients will start seeing "Invalid magic number" error messages as a result of unexpected data within the stream.

When this problem is encountered, the recommended approach is to restart the client. If the client is not restarted, on some occasions, the client may recover after the unexpected data is consumed but this is not guaranteed. If the client recovers with a restart, the view topology displayed does not display one of the nodes added, resulting in uneven request distribution.

This behavior persists in JBoss Data Grid 6.

BZ#808623 **- Some entries not available during view change**

In rare circumstances, when a node leaves the cluster, instead of going directly to a new cluster view that displays all nodes save the note that has departed, the cluster splits into two partitions which then merge after a short amount of time. During this time, some nodes do not have access to all the data that previously existed in the cache. After the merge, all nodes regain access to all the data, but changes made during the split may be lost or be visible only to a part of the cluster.

Normally, when the view changes because a node joins or leaves, the cache data is rebalanced on the new cluster members. However, if the number of nodes that leaves the cluster in quick succession equals or is greater than the value of numOwners, keys for the departed nodes are lost. This occurs during a network split as well - regardless of the reasons for the partitions forming, at least one partition will not have all the data (assuming cluster size is greater than numOwners).

While there are multiple partitions, each one can make changes to the data independently, so a remote client will see inconsistencies in the data. When merging, JBoss Data Grid does not attempt to resolve these inconsistencies, so different nodes may hold different values even after the merge.

This behavior persists in JBoss Data Grid 6.

BZ#810155 - Default configuration may not be optimal for a 2-node cluster

The default JBoss Data Grid JGroups configuration (using UNICAST2 and RSVP) does not lead to optimal performance in a two node cluster scenario. Two node clusters show improved performance with UNICAST and without RSVP.

This behavior persists in JBoss Data Grid 6.

BZ#818863 - Reopened: Tests for UNORDERED and LRU eviction strategies fail on IBM JDK

The LinkedHashMap implementation in IBM's JDK sometimes behaves erratically when extended (as is done by the eviction strategy code). This wrong behaviour is exposed by the JDG Test Suite. The recommendation is, if using eviction, to use other JDKs (Oracle or OpenJDK) which are not affected by this issue.

This behavior persists in JBoss Data Grid 6.

BZ#822815 - NPE during JGroups Channel Service startup

Occasionally, when starting a JDG server, the JGroups subsystem would not start because of a NullPointerException during service installation, leaving the server in an unusable state. This situation does not affect data integrity within the cluster, and simply killing the server and restarting it solves the problem.

This behavior persists in JBoss Data Grid 6.

BZ#829813 - Command line bind address being ignored for HTTP and Infinispan servers

The JBoss Data Grid server by default binds its listening ports to a loopback address (`127.0.0.1`). The `-b` switch can be used to modify the address on which to bind the public interface. However, since the JBoss Data Grid endpoints are bound to the management interface for security reasons, the `-b` switch does not affect them. The user should modify the `standalone.xml` configuration file to place the endpoints on the public interface:

```
<socket-binding name="hotrod" interface="public" port="11222"/>
```

After the above modification, the `-b` switch will determine the network address on which the hot rod port will be bound.

This behavior persists in JBoss Data Grid 6.

CHAPTER 3. RESOLVED ISSUES

The following issues have been resolved in JBoss Data Grid 6.

BZ#745923 - NPE in CacheLoaderInterceptor

The startInterceptor method was called before it was initialized. When four nodes started, **NullPointerException** errors displayed. This is fixed so that the startInterceptor method is now called after it is initialized. As a result, when four nodes are now started, they operate as expected with no errors.

BZ#758178 - Jdbc cache stores do not work with ManagedConnectionFactory in a transactional context

Previously, when the JDBC Cache Store was configured without specifying **connnectionFactoryClass**, the **ManagedConnectionFactory** was selected by default. As a result, **ManagedConnectionFactory** could not connect to the database. This behavior is now fixed and a connection to the database is established as expected when no **connectionFactoryClass** is specified for the JDBC Cache Store.

BZ#765759 - StateTransferInProgressException during cluster startup

Previously, when state transfer was started with different relays on different nodes, the lock was denied and a **StateTransferInProgressException** occurred to prevent a deadlock. Despite the timeout not expiring, a "??:??:??,??? ERROR [org.infinispan.interceptors.InvocationContextInterceptor] (undefined) ISPN000136: Execution error: org.infinispan.statetransfer.StateTransferInProgressException: Timed out waiting for the state transfer lock, state transfer in progress for view ?" error appeared on the server. This is fixed and the **StateTransferInProgressException** no longer displays when a cluster starts up.

BZ#786202 - State transfer taking too long on node join

With JBoss Data Grid cluster of size 4 nodes or higher, if a node crashes and then is brought up again, the subsequent state transfer takes a very long time to conclude. This occurs despite a mild data load (5% of heap and client load). The data load concludes within a minute while the state transfer unexpectedly requires more time than this.

This problem has only occurred once and has not been reproduced in tests since. It is a low risk performance problem.

BZ#791206 - Include appropriate quickstarts / examples for each edition

Quickstarts and examples for JBoss Data Grid 6 are now available within the **jboss-datagrid-quickstarts-1.0.0.zip** file.

BZ#801296 - Storing a byte array via Memcached client fails on Windows

A byte array's storage was occasionally misinterpreted by the server as an invalid command. This was caused by the memcached server not always consuming the CR/LF delimiter which marks the end of a client request. The server then attempted to decode the delimiter as the header of the following request and incorrectly reported this as a bad message. This is now fixed and running the server operates as expected.

BZ#806855 - SuspectedException blocks cluster formation

When a new node joined an existing JBoss Data Grid cluster, in some instances the new node splits into a separate network partition before the state transfer concludes and logged a SuspectException. As a result, the existing nodes do not receive a new JGroups view for ten minutes, after which the

state transfer fails. The cluster did not form with all three members as expected.

This problem occurs rarely, due to two problems. First, the SuspectException on the new node does not allow a new cluster with just the single new node to form. Secondly, the two existing nodes in the cluster did not install a new JGroups cluster view for ten minutes, during which time state transfer remains blocked.

BZ#808422 - **Missing XSD files inf docs/schema directory**

The latest version of some of the configuration schema files was missing from the **docs/schema** directory. This did not affect the functionality of the server. The relevant configuration schema files are now available in the **docs/schema** directory, namely **jboss-as-infinispan_1_3.xsd**, **jboss-as-jgroups_1_1.xsd**, **jboss-as-config_1_3.xsd** and **jboss-as-threads_1_1.xsd**.

BZ#809060 - **Getting CNFE: org.infinispan.loaders.jdbc.connectionfactory.ManagedConnectionFactory when using jdbc cache store**

As a result of a race condition between the server module and the Infinispan subsystem, a server configured with JDBC cache store may occasionally fail to start. The server either started as expected or failed to start.

This occurred when the Memcached server attempted to obtain the memcachedCache before the Infinispan subsystem had a chance to start it (even in EAGER mode). As the server module was forcing its own classloader as TCCL, the cache could not find the necessary classes needed by the cache loaders.

This behavior is now fixed and a server configured with the JDBC cache store no longer experiences and unexpected starting failures.

BZ#809631 - **Uneven request balancing after node restore**

Previously, after a node crashed and rejoined the cluster, it did not receive client load at the same level as the other nodes. The Hot Rod server is now fixed so that the view identifier is not updated until the topology cache contains the addresses of all nodes.

APPENDIX A. REVISION HISTORY

Revision 1.0-3.400   **2013-10-31**   **Rüdiger Landmann**
   Rebuild with publican 4.0.0

Revision 1.0-3   **Mon Aug 05 2013**   **Misha Husnain Ali**
   Published with new product name.

Revision 1.0-2   **Fri July 14 2012**   **Misha Husnain Ali**
   Last bug addition.

Revision 1.0-1   **Fri July 14 2012**   **Misha Husnain Ali**
   Approved for release.