Removing messages left behind after removing a queue manager from a WebSphere MQ cluster

Question

You remove a WebSphere MQ queue manager from your cluster. Your queue manager that is still a member of the cluster and has messages on the SYSTEM.CLUSTER.TRANSMIT.QUEUE (SCTQ), and those messages were intended for the queue manager that you just removed from the cluster. You need to know how to determine the actions required to disposition these messages.

Answer

If the messages are internal MQ subscription messages, you can just remove those messages as they are no longer needed since the queue manager is no longer in the cluster. You can review technote 1170897 for an example of this situation and how to remove the messages.

If the messages on your SCTQ are actually application data intended for a cluster queue, what you do with them will depend on whether those messages were PUT with the option MQOO_BIND_ON_OPEN.

- If the option MQOO_BIND_ON_OPEN was used the messages will only go to that one queue (which no longer exists on the queue manager you already removed). Those messages will not be rerouted to another queue manager even if you have a queue by the same name on that other queue manager. In this case you can either just remove the messages or you can MQGET them off the queue, and have some application process the data from those messages and send them to a new destination. If you do not have a dead letter queue associated with the queue manager, alter your queue manager to point to a dead letter queue and then put a dummy message targeting the CLUSQ of the deleted queue manager. This may cause the messages to be transferred to your dead letter queue where you can reprocess them.
- If the messages on your SCTQ were PUT with the option MQOO_BIND_NOT_FIXED they can be rerouted to another queue manager in the cluster. You would have to define a new cluster queue by the same name as the queue to which the messages were originally routed. If there is already a CLUSSDR channel (which points to the queue manager with your newly defined cluster queue) on the queue manager with the messages stuck on the SCTQ, then you may need to manually stop and restart the CLUSSDR channel. If there is not already an autodefined CLUSSDR channel, then the easiest way to get one would be to connect to the queue manager with the messages stuck on the SCTQ and put a message to the new cluster queue which you defined (on the other queue manager). This should cause the full repository queue manager to provide the definition and a CLUSSDR channel should be automatically defined. Once the channel is defined, the old messages on the SCTQ should also be rerouted to the new cluster queue you defined on the other queue manager.

-
- Delicious
- Linked In
- Stumbleupon
- Google+

SYSTEM.CLUSTER.TRANSMIT.QUEUE has messages and the queue manager has been removed from the cluster

Technote (troubleshooting)

Problem(Abstract)

You removed your WebSphere MQ queue manager (QM1) from cluster (CLUSTER1) and then removed the cluster. After removing the cluster, you notice that queue manager QM1, still has a few messages that were not transmitted to the target cluster queue manager.

Symptom

You followed the steps in the WebSphere MQ Queue Manager Clusters manual for removing a cluster. A few weeks later, you noticed there are some messages in the SYSTEM.CLUSTER.TRANSMIT.QUEUE on QM1 that are intended to go to CLUSTER1, the one that you had removed.

Cause

The messages in the SYSTEM.CLUSTER.TRANSMIT.QUEUE, which have references to CLUSTER1 are subscription messages that were created when cluster queues were opened on the partial repository queue manager QM1.

Example: When the queue (Queue.Reply) was opened at queue manager QM1 for the first time, subscription messages (for information about the queue) were created, and routed through QM1's SYSTEM.CLUSTER.TRANSMIT.QUEUE to each of your 5 repository queue managers. Apparently QM1 did not have active channel connections to all 5 repository queue managers, and some of these subscription messages were not transmitted to the intended repository queue managers. Since QM1 is no longer a member of any cluster, and CLUSTER1 no longer exists, these messages will never be delivered.

Resolving the problem

The messages in QM1's SYSTEM.CLUSTER.TRANSMIT.QUEUE, will not affect the remaining clusters. They can be safely ignored or removed.

IV25030: WEBSPHERE MQ CLUSTER FAILS, POSSIBLY GENERATING FDC FILES

WITH PROBE IDS RM296000 OR OTHER rrcE_REPOSITORY_ERROR

Error description

- WebSphere MQ cluster stops working. FDCs may include
- but are not limited to (see problem summary)
- error code rrcE_REPOSITORY_ERROR and Probe ID RM296000.
- The repository manager process amqrrmfa ends abruptly.
- Other symptom will include a corrupted cluster name of WMQ
- cluster objects, for example:
- CLUSTER(&#65495;&#65506;&#65521;@@@).
- If amqrfdm output is viewed in binary mode, and the first bytes
- of the CLUSTER name are the hex for EBCDIC characters the
- cluster name is not being properly converted. This issue may be
- related to APAR PM84108 on z/OS platform.
-

Local fix

-

Problem summary

- **
- USERS AFFECTED:
- All users of WebSphere MQ Clusters are potentially affected by
- this error. The issue is significantly more likely to occur
- in clusters with very large numbers of queue managers or
- objects (queues and topics), or where object definitions are
- modified frequently.
-
- Platforms affected:
- All Distributed (iSeries, all Unix and Windows)
- **
- PROBLEM SUMMARY:
- When objects in the cluster repository cache are modified (for
- example, changing an attribute on a cluster queue), the
- details for that object are republished to the cluster.
- Previous records for the object may persist for some time in
- the cluster cache, so that applications currently using them
- (for instance having opened the queue for output) can continue
- processing without interruption.
-
- Periodically, the repository process attempts to 'garbage
- collect' these older records, checking whether they are still
- in use. Where multiple such records exist for a particular

- cluster queue manager object (the record in the cache which
- stores information about the channel definition to reach a
- remote queue manager), and these are held in use for a
- prolonged period, an error in the logic leads to the
- possibility that the storage for parts of these queue manager
- records can be reused (for example overwritten to hold another
- object) while actually still required.
-
- This can lead to a variety of errors depending on the precise
- nature of the reuse, varying from no external symptom to
- entire failure of the cluster repository process. Because
- this affects only the cluster repository cache, message data
- is unlikely to be lost, but corrupted records may lead to MQ
- API calls failing (for example with
- MQRC_CLUSTER_RESOLUTION_ERROR), messages may be DLQ'd, or
- channels may have to stop processing when a message cannot be
- correctly routed.
-

Problem conclusion

- The garbage collection logic in the cluster repository process
- is modified to correctly ensure that all handles on 'old'
- cluster queue manager records are released before freeing
- certain chained areas from the record.
-
- Users should perform the following command on repositories where
- they see incorrect cluster data:
- REFRESH CLUSTER(*) REPOS(YES)