Using XA transactions - HA queue managers or GROUPUR?

Jamie Squibb |Dec 14 2016 Updated

A couple of colleagues and I recently received a query that asked for a comparison of the pros and cons of using distributed (XA) transactions with an HA queue manager on a distributed platform, such as the IBM MQ Appliance, versus using a queue-sharing group (QSG) on z/OS with Group Units of Recovery (GROUPUR). Both options are valid. As this was an interesting question, I thought a blog post on this subject might be useful.

**What is a distributed (XA) transaction?**

Transactions are units of work that either complete in their entirety or not at all; they are the basis of many business applications. A distributed transaction is one that updates multiple resources, so coordination is required to ensure it completes successfully or it is universally backed out. A typical example of a distributed transaction in messaging is whereby a message is sent that represents an update that must be made to a database, such as a banking transaction or the placement of a sales order. If the database cannot be updated then it is important that the message is restored to the queue so the update request is not lost and it can be processed again later. To accomplish this a 2-phase commit protocol is typically used whereby the transaction coordinator requests all parties (known as resource managers) first prepare to commit (phase 1). If all parties give a ‘green light’ then the transaction is committed, else it is backed out (phase 2). The XA standard, which is commonly used, is a specification by the Open Group ([http://www.opengroup.org](http://www.opengroup.org/)) that defines a protocol to accomplish this. A transaction that has completed phase 1 but not phase 2 is considered to be in-doubt. The transaction is in-doubt because each resource manager cannot unilaterally decide to commit or back-out without compromising the integrity of the unit of work. The collective response of all parties during phase 1 is required to determine how to resolve the transaction consistently. The transaction coordinator collates these responses then notifies each resource manager of the required resolution.

If a transaction coordinator is disconnected from a resource manager during commit processing then the coordinator is not always able to predict the state of each resource. To allow the transaction to be resolved correctly the coordinator must first query the resource manager upon reconnecting to determine the current state. In the XA specification the xa\_recover request is used for this purpose. If the coordinator unknowingly connects to a different, independent, resource manager it will receive a response stating the transaction is not known. This compromises the integrity of the transaction because the coordinator must infer the resource manager has either committed or backed out its part of the unit of work, when in fact it might still be in-doubt or even resolved in the opposing way.

**What are HA queue managers?**

Queue managers on distributed platforms (Windows, Linux and UNIX) can operate in a high availability (HA) configuration. An HA configuration allows a queue manager to be kept available for applications during a planned or unplanned outage of a single system, by starting it on a secondary system instead. The IBM MQ Appliance includes support for HA queue managers ‘out of the box’ (see <http://www.ibm.com/support/knowledgecenter/en/SS5K6E_1.0.0/com.ibm.mqa.doc/overview/ov00020_.htm>). Two appliances can be connected in a high availability group and data is synchronously replicated between them. Queue managers are automatically started on the secondary appliance when the primary appliance is quiesced or is otherwise offline. On platforms other than the appliance a multi-instance queue manager can be used instead, whereby the data is not replicated but it is stored on a network file server. Alternatively, an HA cluster can be established using PowerHA for AIX (formerly HACMP) or the Microsoft Cluster Service (MSCS). An F5 router, or similar, is often required to route applications to the system where the queue manager is currently active, or applications must use a list of IP addresses and attempt to connect to each system in turn.

**What are queue-sharing groups and GROUPUR?**

Queue managers on z/OS can be configured in a queue-sharing group (QSG). In a QSG queues and channels can be shared by all queue managers in the group using the z/OS Coupling Facility. This capability allows the QSG to present itself as a single highly-available queue manager. The SysPlex Distributor on z/OS can be used to route applications (or queue manager channels), which connect using a single IP address, to any one of the available queue managers in the group. This means they can always access the shared queues and the messages on them, provided that at least one queue manager is active. Queue managers can therefore be quiesced in turn for either general maintenance operations or upgrades without impacting business applications.

As of MQ version 7.0.1 queue-sharing groups support XA transactions, such as those used by WebSphere Application Server (WAS), that are logically owned by the group instead of a single queue manager. This capability is known as Group Units of Recovery (GROUPUR). If connectivity to a queue manager in the QSG is lost when a transaction is in-doubt then the transaction coordinator can reconnect to any member of the group to resolve it. A transaction coordinator that issues xa\_recover is returned a list of all in-doubt XA transactions throughout the QSG that have a group unit of recovery disposition. Similarly, a xa\_commit or xa\_rollback request can be made for any of the returned transactions. For more information see <http://www.ibm.com/support/knowledgecenter/en/SSFKSJ_9.0.0/com.ibm.mq.pro.doc/q004240_.htm>).

**Comparing HA queue managers with GROUPUR**

XA transactions can be safely used with either an HA queue manager or a queue-sharing group with GROUPUR enabled. With HA the queue manager moves from one system to another so applications still connect to the same resource. With GROUPUR applications might be routed to a different queue manager each time they connect, but the queue managers cooperate so they appear as a single resource instead of separate entities.

The main differences between HA queue managers and GROUPUR with respect to support for XA transactions relate to scale and the degree of availability.

HA queue managers have a smaller footprint than a queue-sharing group and they are not limited to a single platform (QSGs are only available on z/OS). However, the scalability of an HA queue manager is limited by the capacity of the system it is running on. There is only one queue manager instance so all applications are routed to the same resource. When using a QSG application connections can be spread across the available set of queue managers. The queue managers in a QSG can run on different LPARs and even different physical hardware, which provides the potential for greater scale and improved performance.

The availability of a QSG is also likely to be higher than an HA queue manager. This is because while an HA queue manager is failing over from one system to another applications are unable to connect until it has restarted. When using a queue-sharing group there is no fail-over of an individual queue manager required, the application only needs to reconnect to be routed to another queue manager in the group. Availability is also greater with a QSG because it is possible for two or more queue managers to always be active to serve applications. With an HA queue manager if the secondary system is unavailable for maintenance there is nowhere else for the queue manager to fail-over to should an error occur.

**Summary**

This blog post has introduced HA queue managers and Group Units of Recovery (GROUPUR) with a queue-sharing group (QSG) and compared them with respect to support for distributed (XA) transactions. Whether an HA queue manager or a queue-sharing group is the preferred solution is likely to depend on the degree of availability and scale that is required by the connecting applications. QSGs provide for greater scale and greater availability during maintenance or in the event of a failure. However, an HA queue manager is likely to be more than sufficient for many use cases.