Difference between Full Repositories and Partial Repositories – Cluster Subscriptions

SeanCawood
Published on 17/10/2019

One of the key parts to the operation of MQ Clustering, are the internal Cluster subscriptions. The term 'cluster subscription' refers to the internal cluster record within the cluster cache which deals with a specific Queue Managers interest in a clustered object (cluster objects are cluster Queues, Queue Managers and Topics). They deal with how a queue manager receives updates for cluster objects. Many clustering related APAR's mention these subscriptions, and in the past have caused confusion with Publish/Subscribe messaging, however there is no relation between Cluster subscription records and Publish/Subscribe messaging. This blog discusses the operation of these records with aim to resolve this confusion that can occur, and describes the difference in operation between their use in partial and full repositories.

Background Reading

This blog article discusses a specific area to how clusters operate, to allow better understanding of the applicability of APARs that mention these 'cluster subscriptions' in their details (such as the Error Description and Problem Summary fields). As such some background knowledge and experience of MQ Clusters is needed. More details on this, including some of the terms used (such as full and partial repositories, and the cluster cache) are described in the knowledge center. The knowledge center has a topic and subtopics describing components of a MQ cluster.

Full Repositories vs Partial Repositories

Full repository Queue Managers host a complete set of information about every object in the cluster, whereas other Queue Managers in a cluster are known as Partial Repository Queue Managers as they only need to hold a subset of the objects in the cluster. This subset of information consists of only the objects which have been required by applications connected to the Partial Repository Queue Manager. When an application attempts to use an object from a Partial Repository, such as by issuing an MQOPEN for a clustered Queue, an expression of interest is recorded and used to request the applicable clustered objects as well as future updates to these objects. The record of this expression of interest is the cluster subscription.

What is a cluster subscription?

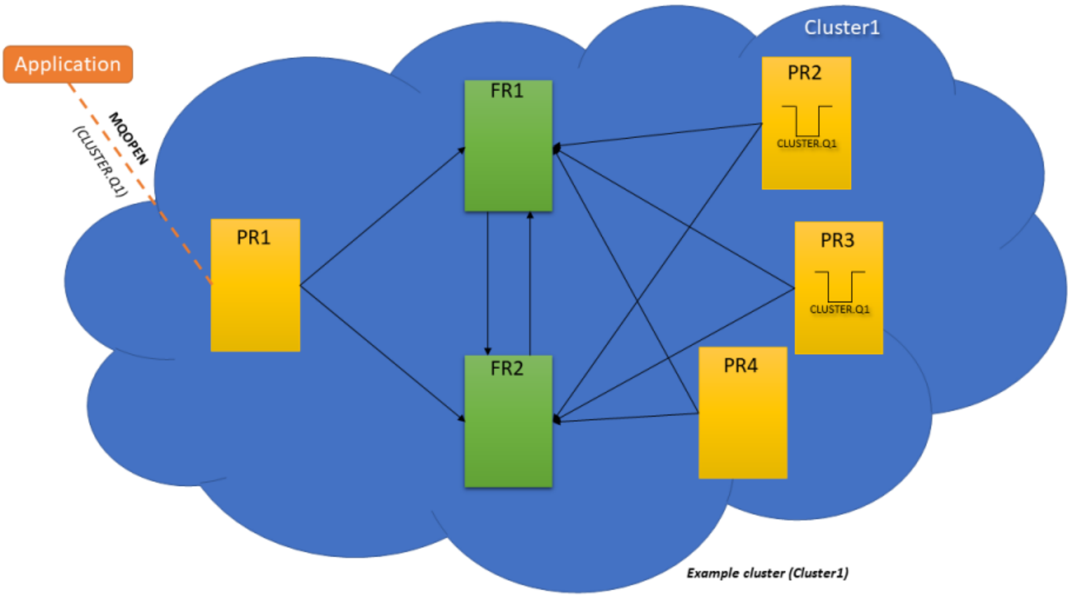
A cluster subscription record is a cluster cache record created by a Partial Repository detailing that it has interest in objects within the cluster for the given name. This expression of interest is the record which controls which object updates are sent to a Partial Repository Queue Manager, and as such defines the subset of remote cluster objects, within the cluster, are in the partial repository cluster cache.

How they work

Subscriptions are created whenever a cluster partial repository wishes to use an object by name. The expression of interest is the name of the object wished to be used. This means there will be subscriptions made for each object name (for each object type), irrespective of the number of objects of that name defined in the cluster. Subscriptions are created for each cluster and the full repositories (up to 2 per cluster, published to up to 2 full repositories). As such in a normal cluster, with the recommended 2 full repositories, 2 subscriptions are created for each named object by a partial repository, with each subscription being stored on a different full repository. If only one full repository exists, only one subscription is created; but with 3 full repositories in a cluster, only two subscriptions are created, with each only existing on one of the full repositories, resulting in one full repository not holding this expression of interest.

Example

For example, in the following cluster (with 2 full repository and 4 partial repository queue managers):



Example cluster (Cluster1)

With a clustered queue ‘CLUSTER.Q1’ defined on PR2 and PR3. When an application connected to PR1 attempts to open the queue for output (MQOPEN with MQOO_OUTPUT). During the MQOPEN process, PR1 expresses interest in ‘CLUSTER.Q1’, this creates two subscriptions in the cache on PR1, each explicitly destined for a specific full repository in the cluster. Each subscription is then flowed to the specific full repository, to query the cluster for objects of that name. When the query is received by the full repository, it is stored and used to push future object changes to the queue manager that ‘owns’ the subscriptions, and the full repositories cache is queried for object records of the queried type and name. Any matching cluster objects are then sent back to the requesting partial repository.

Updates

While these subscriptions are owned and maintained by the partial repository, the majority of the work done with these are by the full repository queue managers. Apart from responding to the initial interest by querying the full repository cache and returning the cluster object records to the partial repository queue manager, they also deal with providing updates to the partial repository for these named objects. These will be:

- When an existing cluster object (with the name as in the subscription) is altered or deleted, these changes are flowed to the full repository. The full repository checks for applicable subscriptions, and forwards these updates to the appropriate partial repository queue managers (the partial which owns the interest/subscription).
- When a new object is added into the cluster. This insert of a new clustered object is handled by the full repository, which checks against all the subscriptions and forwards these updates to the appropriate partial repository queue manager. This results in the partial repository adding this new object record to its local cache, to allow it to be used (as described by the cluster workload management algorithm knowledge center topic).

Lifetime

The expression of interest/subscription is created by the partial repository queue manager when an application wishes to use a named object, as such the subscription is owned by this queue manager and it is its responsible to ensure this interest is maintained (also known as re-subscribing). This is done by keeping track of when objects with the name in the subscription are used, and during specific periods re-subscribing, this updates the subscription in the local cache, as well as the full repository. These subscriptions last until they expire. These will expire when the named object has not been used for a period of time (approximately 30 days after being marked as used). In normal use of the object, the partial repository with the interest will re-subscribe, to update its interest in the named object. When the subscription expires, they expire from both the partial repository queue manager which 'owns' the interest, and from the full repositories within the cluster.

Full Repository Queue Managers

A full repository, by definition, has knowledge of all objects and queue managers in the cluster, it does not require a subscription to receive updates. As such a full repository does not usually have subscriptions for cluster objects it uses. This can change when a queue manager becomes promoted to a full repository or demoted to a partial repository. However, when a queue manager is a member of multiple clusters, where it is only a full repository for a subset of these, it is expected to have subscriptions for all objects. These subscriptions however will only list the clusters the queue manager is a partial repository for two up to two of the full repositories for each cluster.

Impact of multiple clusters

Subscriptions are created for each named object that should be queried. They specify the object name and object type. Subscriptions are created for every cluster the queue manager is a partial repository in. This includes subscriptions being extended to other clusters if the queue manager is added to further clusters as a partial repository.