Persistent Cluster Restart Design Note

Persistent cluster, user perspective.

A persistent cluster is one where all members have a persistent store. A cluster must have all transient or all persistent members, mixed clusters are not allowed.

cluster-size option

`cluster-size N` Wait for at least N initial members before completing cluster initialization and serving clients.

Use this option in a persistent cluster so all brokers in a persistent cluster can exchange the status of their persistent store and do consistency checks before serving clients.

Clean and dirty shut-down.

Each store is an independent replica of the cluster's state. If a broker crashes while there are other brokers running, its store is marked "dirty" because it will be out-of-date with regard to the rest of the cluster.

If the broker is re-started to re-join the a running cluster it will discard the dirty store and get an update from an active cluster member to re-synchronize its state.

If the entire cluster is shut down by an administrator using the `qpid-cluster -k` command, then all brokers will shut down at exactly the same point with the same state in their stores. In this case the stores are marked "clean".

If the cluster is reduced to a single broker, and that broker is shut down, its store is marked clean since it is the the only broker and therefore has the authoritative store.

When the cluster is restarted, brokers with clean stores will recover from their store, brokers with dirty stores will get an update from a clean broker.

Consistency checks

Two UUIDs are saved with each broker's store: cluster-id and shutdown-id. These are used during startup to detact a mistaken attempt to use mis-matched stores.

The cluster-id identifies the persistent cluster. It remains the same if the cluster is shut down and restarted. It ensures no accidental mixing of stores belonging to different clusters.

The shutdown-id identifies a particular clean shut-down event. It ensures that all clean stores were shut down at the same point.

If there is any mis-match in these IDs, all members of the cluster will log a message and exit.

Manual recovery

In the unlikely event that all brokers in a cluster crash so close together that its impossible to determine which was the last one to shut down, all there stores will be dirty.
In this case manual intervention is required to identify which store to recover from.

TODO: describe manual intervention: two parts. First identify which is the best store to start from. Second mark the store as clean by writing a UUID to the shudown ID in the data directory.

Design details

Persistent restart scenarios:

- first run of persistent cluster, all members have empty stores.
- persistent member crashes is re started - re-joins running cluster
- automatic restart after orderly shutdown of persistent cluster
- manual recovery after total cluster failure of persistent cluster

Other requirements:

- cluster initialization: wait for N initial members before going active.
- enforce consistency of broker options that need to be identical across cluster

Persistent cluster

Store statess on broker start-up:

- empty: not used before.
- clean: has state, was shut down by admin. Has intial and shutdown-ids
- dirty: has state, not shut down by admin. Has cluster-id.

cluster-id is stored on the first run of a persistent cluster. Used to ensure members are part of the same cluster.

shutdown-id is stored at administrative shut-down of the cluster. Used to ensure clean stores are from the same shut-down event.

Initialization

1. Wait for N initial members
2. Verify options are consistent for all members or abort.
3. Verify valid store states or abort (see below)
4. Members with empty/dirty stores get update from clean member.

All empty is a valid store state: all members record the same cluster-id and go active.

If any are non empty then

- at least one store must be clean
- all clean stores must have same shutdown-id.
- all clean and dirty stores must have same cluster-id.

All clean members restore from stores. All empty members set the cluster-id from the cluster. All dirty/empty members get an update from a clean member.

Joining

If the new member has a non-empty store, the cluster-id must match the cluster. The new member gets an update from the cluster.

Manual Recovery

TDB: how to identify the best store?