Reliability Requirements

Reliability Requirements

Fail-over (session state)

A cluster member informs its clients of backup candidates for each session. It can update the list periodically.

After an unexpected disconnect the client can connect to one of the candidates and resume its session transparently. All session state is preserved including:

- Open references
- Active consumers
- Commands-in-flight
- Open transactions (question: Is there any value in fail-over that aborts TX and/or DTX transactions?)

Sessions *do not* survive

- multiple failures that include the current node and all back-up nodes for that session.
- shutdown/restart of the cluster.

Cluster Restart (durable resources)

The AMQP entities that survive a restart are those defined by AMQP to survive broker restart. AMQP defines *durable* exchanges and queues and *persistent* messages. Some further definitions:

- *durable* message: persistent messages on a durable queues.
- *durable* enque: act of enqueuing a persistent message on a durable queue.
- *durable* binding: binding between durable exchange and durable queue.

The following are preserved if the entire cluster shuts down/crashes and is re-started:

- *Durable* wiring: durable exchanges, queues and bindings.
- *Durable* messages
- *Prepared* DTX transactions

The following do not survive a restart:

- Session state
- Non-durable wiring
- TX transactions are aborted.
- Unprepared DTX transactions are aborted.
- Non-durable effects of prepared DTX transactions are lost.

Restarting DTX Transactions

On restart, prepared DTX transactions may commit or rollback. In either case the outcome is *as if* the transaction had comitted or rolled back just *before* the restart: All durable transaction effects survive the restart, all non-durable effects are lost.

In particular

- On ***commit*****:** *non durable* messages enqueued in the transaction are *lost*, as if they had been enqueued before the restart and were lost in the restart.
- On ***rollback:*** *non durable* messages dequeued in the transaction are *lost*, as if they had been put back on the queue before restart and then lost in the restart.