

ClusteringHA

Definitions

There are two very different reasons to cluster:

- Reliability/Fault Tolerance
 - Cluster members replicate state.
 - If one member fails, clients can fail-over to another.
- Scalability/throughput/load balancing:
 - Distribute large work load across multiple brokers for higher throughput.

Its important not to confuse the two goals. Note that a reliable cluster will be LESS scalable and performant than even a single broker - replication is extra work on top of normal processing. There is also:

- Federation (a set of distributed exchanges and queues, seperately managed and wired together)

It's not clear where to draw the line between federation and clustering for scalability.

Reliability clustering is orthogonal to scalability clustering/federation, which means they can be combined. Just replace the individual brokers in your federation or scalability cluster with reliable broker clusters and you have a reliable and scalable system.

Requirements/Use Cases

- [Reliability Requirements](#)
- [ClusteringAndFederation](#)

Design notes

- [Cluster Design Note](#) - Cluster for reliability. A reliable broker cluster can participate as a single broker in federation or throughput clusters.
- [Persistent Cluster Restart Design Note](#) - re-starting a cluster with persistent members.
- [Cluster Failover Modes](#) - how a cluster and its clients deal with failures.
- [AMQP breakdown for clustering](#) - Analysis of AMQP 0-10 commands and their effect on replicated state in a cluster.
- [Federation Design Note](#) - Discussion of what has been done to date in C++
- [Java Federation Design Proposal](#) - Discussion of what could be implemented in the Java Broker.

Related reading

- AMQP specification, chapter 3 "Sessions" and session class documentation in chapter 9. <http://jira.amqp.org/confluence/download/attachments/720900/amqp.0-10.pdf?version=1>
- openais.org docs on Closed Process Group (CPG), cpg man pages in openais install.