Overview

Abstractions

ActiveCluster provides two different abstractions which can be used independently or seamlessly together.

Node list abstraction

ActiveCluster maintains a list of Node objects, each Node representing a member of the Cluster.
A node can optionally have some local state which is replicated to other members of the cluster.

You can look at the node list whenever you like and listen to nodes joining, leaving and updating their state.

Messaging abstraction

In addition the Cluster (the group of nodes) and each Node in the cluster have a Destination. The other part of the ActiveCluster API is a simplified facade to messaging - way simpler and more abstract than JMS - which allows you to send application level messages to the cluster as a whole or to individual members of the cluster.

Partitioning

Its very easy to implement ActiveCluster on top of any messaging or networking technology. The default implementation out of the box uses JMS. This means we can take advantage of

- destination wildcards
- JMS selectors
- content based routing
  To partition one logical cluster into many different parallel partitions. Different members in the cluster can see a different slice of the cluster based on what they want to see/do.

e.g. some nodes in the cluster may not respond to application level messages - they may wish to just view the cluster to monitor the full cluster behaviour. Other nodes may wish to be participants but only see their immediate buddy-group.

We can create sophisticated slicing & dicing of the cluster, participants, state replication and application message flow using standard JMS techniques

For a more detailed look at the ActiveCluster, check out the javadocs