New Configuration Parameters for JBMESSAGING-1842

1. DB connection retry parameters:

RetryOnConnectionFailure

This is a boolean type parameter. It indicates whether to retry on DB connection failures. Default is false.

MaxRetry

This is an Integer type parameter. It specifies maximal retry times on DataSource failures, default is 25.
If you want retry forever, set it to -1.

RetryInterval

This is an Integer type parameter. It specifies the retry interval (in milliseconds) between two consecutive retries,
default 1000 (1 sec).

These three parameters apply to three mbeans in JBM, i.e. the JDBCPersistenceManager, JMSUserManager and MessagingPostOffice.
They are configured separately.

2. New Failover Model control parameters

With JBMESSAGING-1842 a new type of failover behavior was introduced. This new behavior allows for proper handling of the case
where a node is falsely reported to have left the cluster while it still alive. This is done by the cluster observing a
cluster timestamp table to determine whether a node is really dead or not.

Two parameters were added to MessagingPostOffice mbean:

KeepOldFailoverModel

This is a boolean type parameter indicating whether to enable the new failover mode. Default is true (disable
new failover behavior).

NodeStateRefreshInterval

This long type parameter tells a node how long it should be between two consecutive timestamp update. A node needs
constantly updates its timestamp to tell the cluster that it is alive. And the cluster uses this parameter to decide

when it treats a node as being dead. Default is 30000 (30 seconds).

3. Additional Failure Handling Enhancement

With new failover mode, if a node is falsely reported to have left a cluster and at the same time this node also loses its DB connection, the node won't be able to update its timestamps during this failure time and yet keeps alive. The cluster has no way to tell this node's real state but to think it is dead and will failover for it -- resulting in potential issues described in JIRA 1842.

To avoid this, a new MessagingClusterHealthMBean mbean is introduced. This mbean is responsible for monitoring the node's state and stops/starts a node under such a failure situation just mentioned. When a node is 'shunned' from the cluster and also lose its DB connection, this mbean will shut down the node immediately, waiting for it to be failed over, observing the JGroups status and DB status, and restarting the node if DB connection is restored and JGroups is back to normal.

To enable this feature, you need to deploy this mbean and add a 'MessagingClusterHealthMBean' attribute in your ServerPeer mbean, like

```
<depends optional-attribute-name="MessagingClusterHealthMBean">jboss.messaging:service=MessagingClusterHealthMBean</depends>
```