

JMS Clustering and Failover



Applies to:

SAP NetWeaver AS Java

Summary

This article reveals the details of the Java Message Service (JMS) and Message-Driven Beans (MDBs) clustering and failover concepts and implementation in the SAP NetWeaver AS Java. The theory is illustrated by examples of different clustering setups for both JMS resources and MDBs. We discuss the differences in the clustering configuration and semantics between SAP NetWeaver 2004 and SAP NetWeaver 7.0 (2004s) on one hand and SAP NetWeaver Composition Environment 7.1 on the other.

Authors: Stoyan Vellev, Ekaterina Anachkova

Company: SAP Labs Bulgaria

Created on: 04 April 2008

Authors Bio



Stoyan Vellev is leading the Java Message Service (JMS) development team at SAP Labs Bulgaria. He is in charge of the research and development projects in the JMS area of Java EE.



Ekaterina Anachkova is an information developer at SAP Labs Bulgaria. She writes in the areas of Java Message Service (JMS) and Enterprise JavaBeans (EJB).

Table of Contents

Introduction	3
Overview of the JMS Clustering	3
Overview of the Message-Driven Beans Clustering	5
Message-Driven Bean Clustering in SAP NetWeaver 2004 and SAP NetWeaver 7.0 (2004s).....	6
Message-Driven Beans Clustering in SAP NetWeaver Composition Environment 7.1	6
JMS Failover	7
Message-Driven Bean Failover	7
Summary	8
Related Content.....	8

Introduction

Even if you have heard about the Java Message Service (JMS) Virtual Providers, you still might wonder what their purpose is and how you can benefit from them. This is a comparatively new SAP-specific concept that is related to load balancing, clustering, scalability and failover. Another commonly asked question is what the clustering of message-driven beans (MDBs) is – on which cluster nodes MDBs are activated and how JMS messages are delivered to them.

In this article, we will introduce you to the SAP-specific notion of JMS Virtual Providers and we will review the clustering and failover aspects of JMS and MDBs.

Overview of the JMS Clustering

Clustering in the context of the SAP JMS Provider is the distribution of the JMS resources among the nodes in a cluster and the way messages are transferred between the nodes.

All JMS resources are logically grouped into JMS Virtual Providers. In a cluster, JMS Virtual Providers are distributed among the different cluster nodes. In other words, a JMS Virtual Provider is the minimum set of JMS resources that can be independently hosted by a cluster node.

There are two clustering options for a JMS Virtual Provider:

- The JMS Virtual Provider is instantiated on exactly one node in the cluster and all requests concerning JMS resources of this Virtual Provider are redirected to and processed on the hosting node. In this case the Virtual Provider is called *Global JMS Virtual Provider*. Messages by a producer residing anywhere in the cluster and sent to a destination belonging to a Global Virtual Provider are available to consumers residing on every cluster node.

Here is an example topology of a cluster with six nodes, two of them are hosting Global JMS Virtual Providers (a “blue” one hosted by Server 4 and a “red” one hosted by Server 1). For the “blue” JMS Virtual Provider, there are two message producers (on Server 0 and Server 3) and one message consumer (on Server 2) and for the “red” JMS Virtual Provider there is one message producer (on Server 3) and one message consumer (on Server 5). The arrows indicate the message flow.

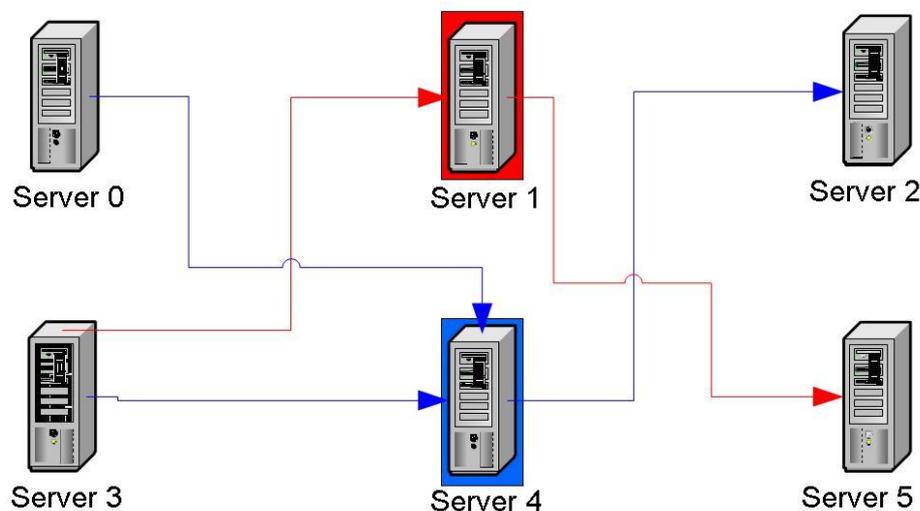


Figure 1. Clustering of Global JMS Virtual Providers

Global JMS Virtual Providers are initially distributed among the nodes during cluster startup. When starting, the nodes compete to become hosts of the Global JMS Virtual Providers. Each node may host zero or more Global JMS Virtual Providers. A cluster node can host either at most one, or unlimited number of Global JMS Virtual Providers, depending on a Boolean configuration property called `hostMultipleVPs`. The default value is `true` and it allows a cluster node to host more than one Global

JMS Virtual Provider while a value **false** limits the number of Global JMS Virtual Providers hosted by a node to one. The default option guarantees that all Global JMS Virtual Providers configured on a system will be hosted (started), regardless of the number of nodes in the cluster.

In case a cluster node hosting some Global Virtual Providers is stopped or crashes, there is a failover mechanism that ensures that the respective Global Virtual Providers will be automatically hosted by one or more of the other nodes in the cluster.

- The JMS Virtual Provider is instantiated on every node in the cluster. Such a Virtual Provider is called *Local JMS Virtual Provider*. Messages by a producer residing on a given node and sent to a destination belonging to a Local Virtual Provider are available only to consumers residing on the same cluster node. In other words, the instances of a local Virtual Provider on the different nodes in the cluster are independent of one another although they share the same name. Messages are delivered (locally) only to the consumers collocated with the message producer (that is residing one and the same node).

Note: Have in mind that Local JMS Virtual Providers and the `hostMultipleVPs` property have been introduced since SAP NetWeaver Composition Environment 7.1.

Here is another example topology of a cluster with six nodes, with two Local JMS Virtual Providers (a “blue” one and a “red” one). Each of the two Virtual Providers is instantiated on (hosted by) each node in the cluster. The messages produced on a node are delivered only to consumers residing on the same node. Hence, Local JMS Virtual Providers do not incur JMS communication between the cluster nodes.

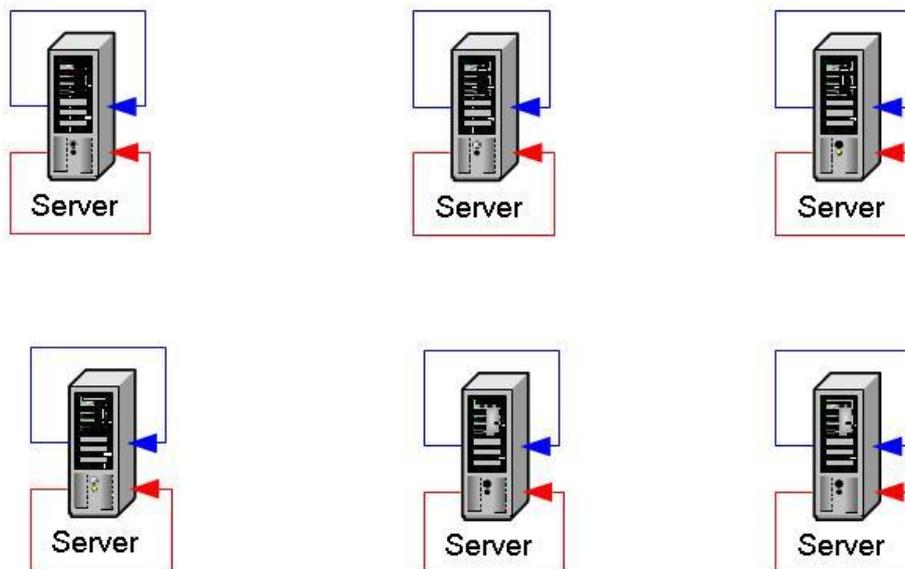


Figure 2. Clustering of Local JMS Virtual Providers

You can set the clustering behavior of a JMS Virtual Provider by its `runningMode` property. To configure the JMS Virtual Provider to be instantiated on all nodes, set the `runningMode` property to `Local` in the `<virtual-provider-properties>` tag in the `jms-resources.xml` deployment descriptor. If you do not specify the `runningMode` property, the default value is `global`.

```
<virtual-provider-properties>
  <name>testRunningMode</name>
  <property>
    <description>runningMode</description>
```

```
<config-property-name>runningMode</config-property-name>  
<config-property-value>local </config-property-value>  
</property>  
</virtual-provider-properties>
```

Overview of the Message-Driven Beans Clustering

Since message-driven beans are widely used in JMS scenarios, message-driven beans' clustering is as important a topic as the JMS clustering.

Clustering in the context of message-driven beans is the distribution of the MDB instances in a cluster and the way messages are distributed to them.

For simplicity, in this article MDBs registered to a Queue are called Queue MDBs and MDBs registered to a Topic are called Topic MDBs.

There are two options regarding MDB clustering – either an instance of the MDB is activated on every node in the cluster or the MDB is activated on exactly one (randomly selected) node.

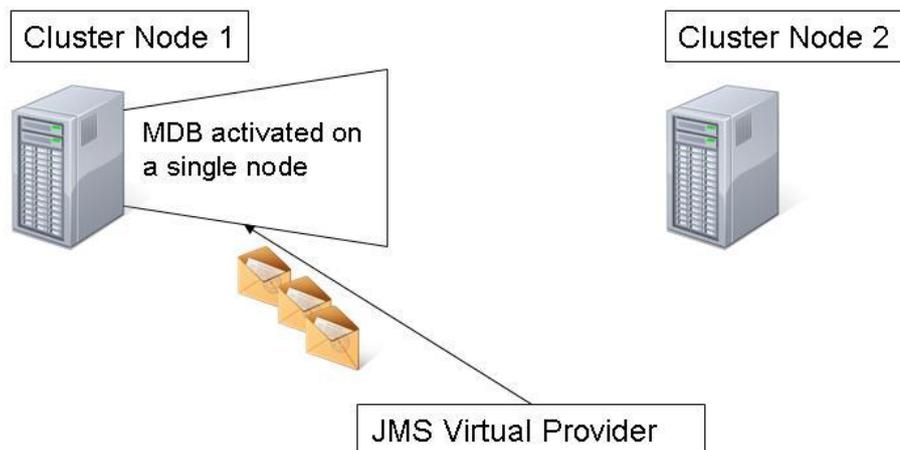


Figure 3. MDB Clustering – Single Cluster Node Activation (*Singleton MDB Activation*)

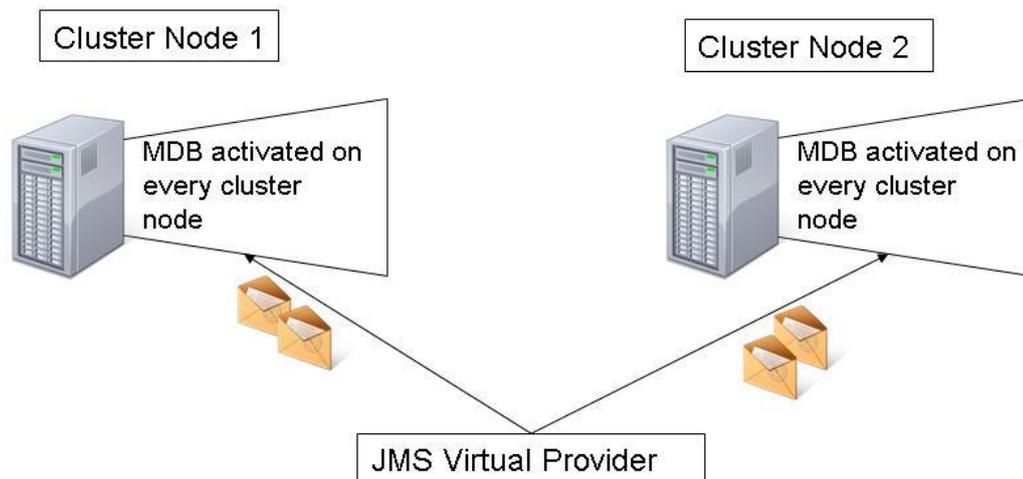


Figure 4. MDB Clustering – Activation on Every Cluster Node (*Homogeneous MDB Activation*)

You can configure the MDB clustering in the `ejb-j2ee-engine.xml`. However, for the different releases of SAP NetWeaver the configuration is different as well.

Message-Driven Bean Clustering in SAP NetWeaver 2004 and SAP NetWeaver 7.0 (2004s)

In SAP NetWeaver 2004 and 7.0 (2004s), Queue MDBs are always activated on all cluster nodes and this behavior is not configurable. Topic MDBs are activated in a single (randomly selected) cluster node by default. If you want to activate a Topic MDB on all cluster nodes, you have to specify the configuration property `topic-on-all-nodes` with value `true` in the `ejb-j2ee-engine.xml`.

```
<message-props>
  <destination-name>myTopic</destination-name>
  <connection-factory-name>myTopicConnectionFactory</connection-factory-name>
  <property>
    <property-name>topic-on-all-nodes</property-name>
    <property-value>true</property-value>
  </property>
</message-props>
```

For Queue MDBs, each JMS message is delivered to exactly one instance, the instance being selected for each message in a round-robin fashion. This is a common way for implementing load balancing of MDB processing logic in the cluster.

For Topic MDBs activated on single cluster node, all JMS messages are delivered to the MDB instances on this single node.

For Topic MDBs activated on all cluster nodes, a copy of each JMS message is delivered to an MDB instance on each node. This behavior can be used by applications to implement cluster communication.

Message-Driven Beans Clustering in SAP NetWeaver Composition Environment 7.1

In SAP NetWeaver Composition Environment 7.1, both Queue and Topic MDBs are activated on all cluster nodes by default. If you want to configure an MDB to be activated on exactly one cluster node, you have to specify the `single-server-node` property with value `true` in the `ejb-j2ee-engine.xml`.

In case of Topic MDB activated on all cluster nodes, a copy of each JMS message is delivered to an MDB instance on each node. If your application needs single processing of the Topic messages, you need to configure it to be activated on a single cluster node.

```
<bean-props>
  <property>
    <property-name>single-server-node</property-name>
    <property-value>true</property-value>
  </property>
</bean-props>
```

Alternatively, you can register the MDB to a destination belonging to a Local JMS Virtual Provider. In this case, even though the MDB is activated on every cluster node, each message (produced anywhere in the cluster) will be processed by exactly one MDB instance – one residing on the same node as the message producer. This second approach is preferable to the first one because you will gain scalability (processing will take place on all nodes where there is a message producer) and improved reliability (you have no single point of failure as with Global JMS Virtual Providers).

In case of Queue MDB activated on all cluster nodes, the different JMS messages are delivered to MDB instances on different cluster nodes in a round-robin fashion.

JMS Failover

Failover in the context of JMS is the process of detecting a system failure (crashing cluster node) leading to unavailability of JMS resources (hosted on the failing node) and migrating these resources to a backup node.

If a node hosting one or more Global JMS Virtual Providers crashes, all other nodes detect the crash of this node and compete to obtain its Virtual Providers and distribute them among themselves. Note that the failover of Global JMS Virtual Providers is automatic and it does not require any manual configuration. However, failover is not transparent to applications. While failover is taking place, the JMS resources of the Virtual Provider(s), which are being transferred to another node, are temporarily unavailable. Applications must handle this case with appropriate reconnection logic.

Local JMS Virtual Providers do not have (and do not need) failover. Messages produced by local producers (that is, residing on the crashing node) are visible only to local consumers (residing on the same node). That is why, the crash of any Local JMS Virtual Providers does not affect in any way applications running on the other nodes of the cluster.

Message-Driven Bean Failover

Failover in the context of MDBs is the process of detecting a system failure (crashing cluster node) leading to unavailability of the JMS destination (hosted on the failing node) to which the MDB is registered and reconnecting the MDB once its destination has been migrated to a backup node and available again.

MDBs registered to destinations belonging to Global JMS Virtual Providers have automatic failover and, upon detecting a failover condition, will try to reconnect. Each MDB will perform a number of reconnection attempts, with some sleep between two consecutive attempts. The two properties that determine the MDB reconnection – maximum number of reconnection attempts (`reconnect-attempts`) and the sleep time between two attempts (`sleep-between-attempts`) - are configurable in `ejb-j2ee-engine.xml`. The default values are 5 reconnection attempts and 1000 milliseconds sleep between two consecutive attempts. Once the reconnection attempts of an MDB have been exhausted, it will never try to reconnect again and it will not receive any messages since. Note that the optimum values of the two reconnection-related configuration properties depend on the hardware setup, the cluster size, the cluster load and all other factors that may influence the time necessary for a successful failover of the Global JMS Virtual Provider.

MDBs registered to destinations belonging to Local JMS Virtual Providers do not have (and do not need) failover because they become unavailable together with the destination they are registered to (and any producers to it).

Summary

In this article we make an overview of the JMS and MDB clustering and failover. In the general case, JMS resources and MDBs are heterogeneous in a cluster, which means that the different cluster nodes cannot be treated as equal and undistinguishable. That is why, it is important to understand the messaging semantics in the different cases. We give examples of how clustering is configured and discuss the version-specific differences between SAP NetWeaver 2004 and SAP NetWeaver 7.0 (2004s) on one hand and SAP NetWeaver Composition Environment 7.1 on the other.

Related Content

- Setting the MDB Properties in the ejb-j2ee-engine.xml:
http://help.sap.com/saphelp_nwce10/helpdata/en/46/518042bc865e3fe10000000a11466f/frameset.htm
- Creating and Configuring JMS Virtual Providers:
http://help.sap.com/saphelp_nwce10/helpdata/en/46/7088a5f11b0e9ce10000000a155369/frameset.htm

Copyright

© 2008 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, System i, System i5, System p, System p5, System x, System z, System z9, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, Informix, i5/OS, POWER, POWER5, POWER5+, OpenPower and PowerPC are trademarks or registered trademarks of IBM Corporation.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided "as is" without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

Any software coding and/or code lines/strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.