DISTRIBUTED SYSTEMS

Jms and Jmx

Master M1 MOSIG, Grenoble Universities

GRÄBNER Laurent
TCHOUGOURIAN Tigran

10/04/2011

Supervisor :
BOUCHENAK Sara

The assignment is about building an asynchronous-communication application with JMS.

Table of contents

1. Architecture of the application

The architecture of the application is quite simple. The OverlayCreator generate the structure of the servers and run one virtual node by server. Each virtual node used to transfer information to the top layer, has a unique topic where its children publish. Every 15 seconds, a node access, thanks to JMX, some information about the server. Once the local data are recovered, the node collects data from its children, aggregates them and sends them to the upper layer thanks to JMS. Because the system is asynchronous, two threads are running every time. The collecting thread is a simple listener which receives JMS messages and updates data. In the other hand the publishing thread sends information about the server every 15 seconds

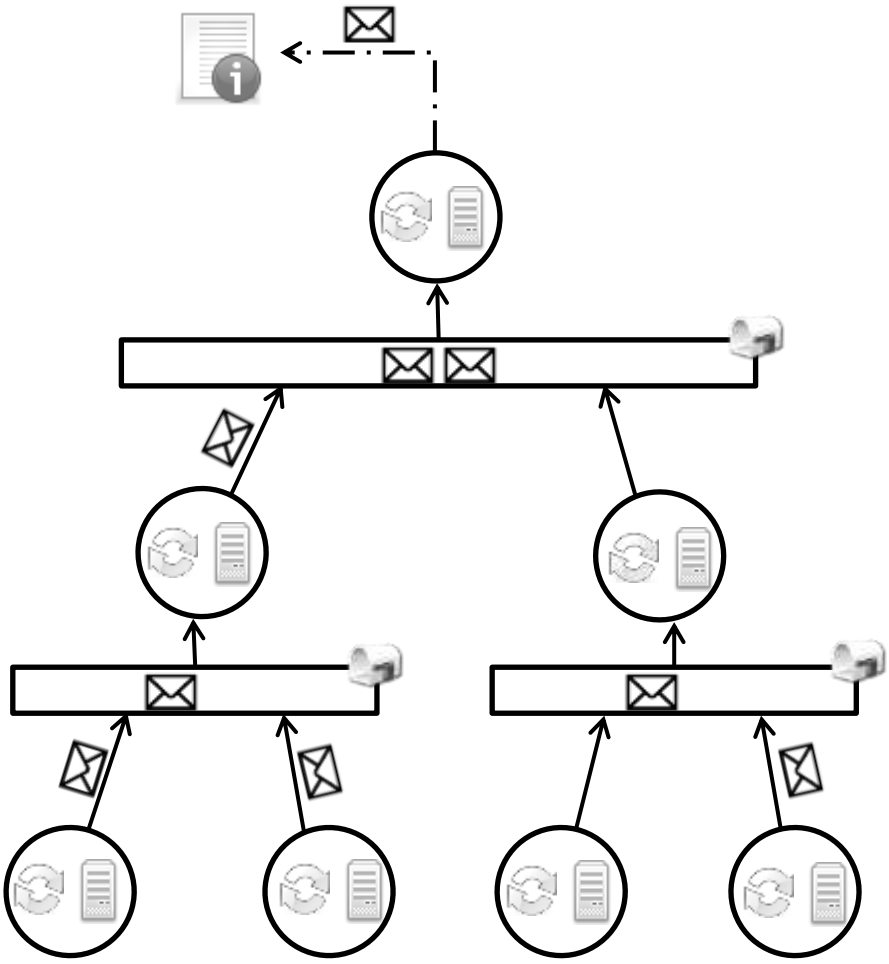

Figure 1-1: Global architecture

The Figure 1-1 shows an instance of the overlay. Each node runs on a server and collect information every 15seconds. Send a message to the upper lever. The messages are always up to date in the mail box. Only the higher node (master node) saves the finale message in a log file.

Legend	
	A node running on a server

	JMX collects information about the server state.
	A message waiting to be collected in the topic of the upper node.
	Shows the topic where a node publishes.
	A message sends to a topic.

2. Overlay

The nodes have to be started single-handedly from one program that has knowledge on the structure of the tree of nodes. The structure of tree of nodes has to be known beforehand and has therefore to be created virtually in advance. For this purpose a class "VirtualNode", which represents the basic attributes of a node, and a main class "OverlayCreation", which builds a tree and executes the node processes, are provided.

2.1. Virtual Node

The virtual Node is a help class containing the basic attributes a node has, such as the ID, the Parent and the Childs.

2.2. Overlay Creation

This main class parses the arguments given by the user, such as the number of nodes, the arity or the depth of the tree. The information provided by the user is then used to calculate the missing properties of the tree and eventually a virtual tree is built. After the creation of a virtual tree, the main function proceeds to traverse the tree and starts each time a node process, which corresponds to the node it is traversing. After the execution of a process, the process is stored in a list of processes, so the main class "OverlayCreation" has the possibility to end all these processes upon exiting.

3. Node

The node is the virtual representation of a server. It runs on each server from the Overlaycreator. The nodes are communicated between them with JMS whereas all the monitoring is made with JMX.

3.1. JMS

To be able to communicate between nodes:

- Each node with a parent starts publishing to its parent topic.
  - Every 30sec the node publishes data to the topic.
- Each node with children, create a unique topic
  - A listener (MemoryListener here) is set to receive asynchronously data from children.

3.2. JMX

To be able to monitor the server status, JMX is used. Every 30 sec JMX provide information about the memory of the server. To avoid the node to waste CPU time, the thread managing publishing information sleeps when not needed.

4. Handbook

Step 1: Execute openjms server, startup.bat (windows) or startup.sh (linux).

Step 2: Launch overlaycreator.jar

```
>java –jar Overlaycreator –N 7 –d 4
>java –jar Overlaycreator –N 7 –a 3
```

The log file is created in the java JDK folder (on windows: C:\Program Files\Java\jdk1.6.0_21). To modify this folder you have to modify the property pattern in the logging.properties file (C:\Program Files\Java\jdk1.6.0_21\jre\lib)