Monitoring

Apache Tomcat

Monitoring Apache Tomcat

OpenOffice/PDF Version

1º Edition , 9 May 2011

© Artica Soluciones Tecnológicas 2011

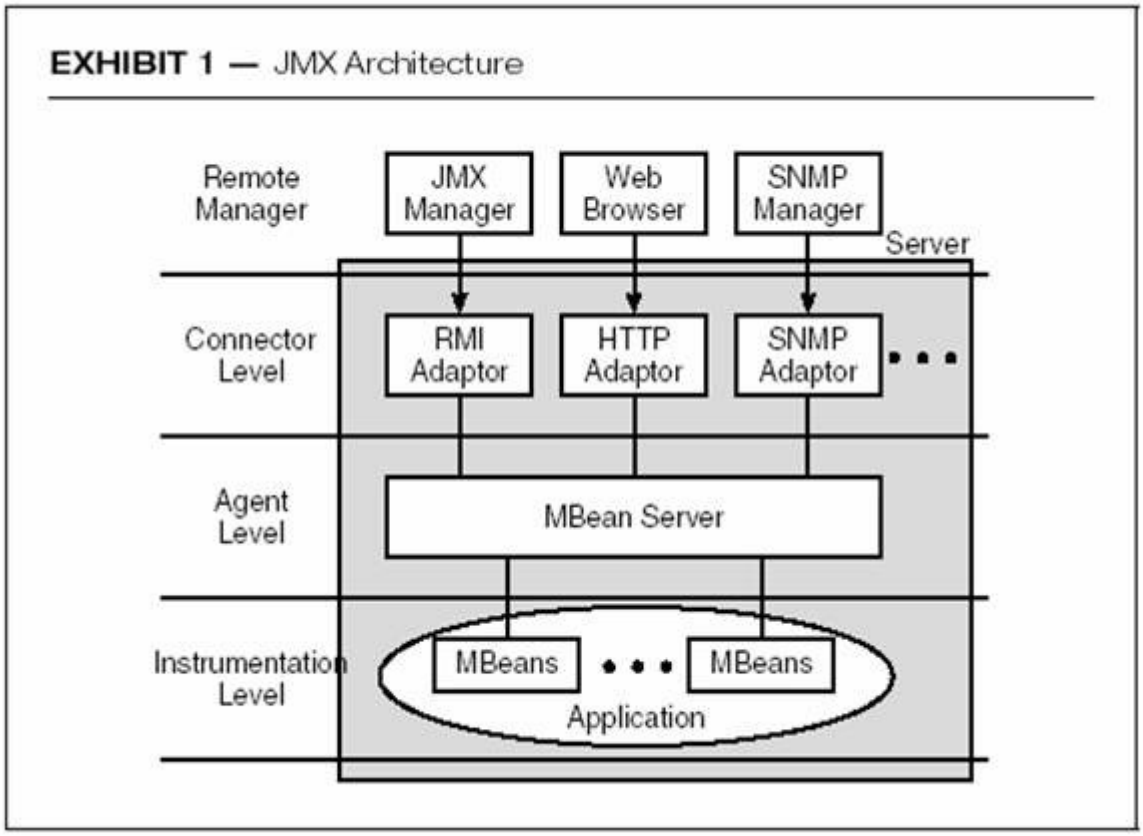
© Tomás Palacios Bote 2011

1 Monitoring Apache Tomcat

1.1. Monitorización mediante plugins y módulos

Once we have installed and configured both Pandora and Tomcat servers, we're going to explain how to obtain info about the server status.

There are different ways to achieve this goal, from plugin agents for XML parsing to SNMP and JMX monitoring:



1.1.1. Plugin Agent (local method)

Let's suppose we have installed Pandora and Tomcat servers on different machines, with all the monitoring systems installed and configured. (Pandora Agent, JMX with jconsole)

We need a script to gather the values we want to monitor from the XML file which contains the Tomcat Server Status.

We've used an Apache Tomcat Turnkey where we can find that file in the following location:

```
http://<Tomcat-IP-Address>/manager/status?XML=true
```

This script is going to store this file in a local temporal file and parse these values in a Pandora-compatible XML format, to create modules dynamically.

The script is the following:

```
pandora_tomcat.pl
```

Once we have located the script and the values we want to monitor, include this file in pandora_agent.conf as a plugin agent. This plugin will create a module for each value, visible in the Pandora server web interface.

By adding new connectors in the Tomcat server, this program will respond by adding new modules dynamically with new parsed values for each new connector.

A Java connector is a programming interface (API) that provides an interface between a Java program and a legacy application such as CICS and R/3. Java Connectors, which are part of the J2EE platform, make legacy applications look like an Enterprise JavaBean (EJB) to the Java program, making them compatible.

This way, each module will be identified by the connector name plus the data name it gathers.

```
/etc/pandora/pandora_agent.conf

module_plugin <plugin-path>
```

```
module_plugin <plugin-address> --user=username –password=userpass http://<TomcatServer-IP-Address>/manager/status?XML=true <plugin-name>
```

jk-127.0.0.1-8009:JK_MAX_...		200		3:06 minutes
jk-127.0.0.1-8009:JK_THRE...		2		3:06 minutes
jk-127.0.0.1-8009:JK_THRE...		4		3:06 minutes
jk-127.0.0.1-8010:JK_MAX_...		200		3:06 minutes
jk-127.0.0.1-8010:JK_THRE...		1		3:06 minutes
jk-127.0.0.1-8010:JK_THRE...		4		3:06 minutes
LastLogin		articast pts		3:06 minutes
Load Average	Average process in CPU (Last m...	0.1		3:07 minutes
memfree		179,272		3:07 minutes
memused	Meminfo postprocess	1,205,020		3:07 minutes
proctotal	Total processes	204		3:07 minutes
Radstat	Informa del tiempo que lleva c...	Reply-Messa		**28 days**
sshDaemon		1		3:06 minutes
TOMCAT_FREE_MEM		52,333,744		3:06 minutes
TOMCAT_MAX_MEM		132,579,328		3:06 minutes
TOMCAT_TOTAL_MEM		65,470,464		3:06 minutes

1.1.2. SNMP Monitoring

You can use one of the multiple SNMP-JMX adaptors in the web to monitor Apache Tomcat.

These are the most important:

- Westhawk SNMP
- Sun JDMK

Each of them allows us to convert JMX notifications into SNMP Traps and extract information about the server status.

It would be just a matter of introducing the OIDs of the generated MIBs by the adaptor with its value and data type in the Pandora server and we could start monitoring it.

Another way to monitor Tomcat and other Java servers is JMX.

1.1.3. JMX Monitoring

JMX (Java Management Extensions) provides the tools to implement distributed, modular, dynamic and web based solutions to manage and monitor, devices, applications and network-driven services, specially designed for Java.

JMX uses an HTTP-adaptor to show all the info it manages via web.

We can take benefit of the HTTP-adaptor data extraction by parsing this data in a Pandora-compatible XML format. When Pandora processes this XML, it will create modules for each data.

However, it's better to use the other two ways for Tomcat monitoring and leave this as an alternative.

There's an easier way to monitor Tomcat server using JMX, if we use the **twiddle** command, a JMX Client which gathers all kind of info about the server MBeans, into a <dataname>=<data> format after its execution. We can gather information about the Mbeans attributes which contains the connector and server status.