



DOPSY
group

Distributed Systems Lab
Wiesbaden University of Applied Sciences

ARM-Instrumentation Framework - Quickstart Guide -

Jens Bloemer

Version: May 18, 2007

Contents

1	Introduction	3
2	Prerequisites	4
2.1	Required software	4
3	Installing the Petstore sample application	5
3.1	MySQL preparation for Petstore	5
3.2	Building Petstore	6
3.3	Deployment	7
4	Instrumentation	8
4.1	ARM libraries	8
4.2	JBoss instrumentation build	9
4.3	Measure certain methods in Petstore	9
4.4	Tomcat instrumentation build	10
4.5	Binding the ARM instrumentation	11
4.6	Deploying the instrumented Petstore	12

1 Introduction

The JBoss ARM-Instrumentation Framework of Distributed Systems Lab is a very effective way of measuring the performance of applications or parts thereof, since there is no need to change an application's source code and the performance instrumentation can be activated or deactivated during runtime.

This guide is to show a brief example of installing the framework. Though there are many possibilities, Java Petstore is used as a sample application.

The sample platform is Linux, though the ARM-Instrumentation Framework can be applied on a Windows platform as well.

2 Prerequisites

2.1 Required software

Please download the following software packages:

- JBoss Application Server 4.0.5 from <http://labs.jboss.com/jbossas/download>
- Apache Ant in its current release from <http://ant.apache.org>
- Java JDK 1.5 from http://java.sun.com/javase/downloads/index_jdk5.jsp
- MySQL 5.0 Community Server from <http://mysql.org/downloads/mysql/5.0.html>
- The Tomcat/JBoss instrumentation framework from
<http://wwwvs.informatik.fh-wiesbaden.de/oss/jboss-arm/jb-tomcat-fw.tgz>
- The Petstore sample application from
<http://wwwvs.informatik.fh-wiesbaden.de/oss/jboss-arm/petstore.tgz>

Right after downloading you should install the JDK and build Ant, JBoss and MySQL according to the respective documentations.

3 Installing the Petstore sample application

In order to demonstrate the use of the ARM Instrumentation Framework, this guide uses Java Petstore as sample application.

3.1 MySQL preparation for Petstore

At first JBoss is configured for the use of a MySQL database. Therefore a privileged user `jboss` is needed with his own database. Start the MySQL daemon and the MySQL console:

```
$ mysqld_safe &
$ mysql -u root
mysql> create database jboss;
mysql> create user jboss identified by 'password';
mysql> grant all on jboss.* to 'jboss'@'localhost'
    -> identified by 'password';
mysql> \q
```

Of course, 'password' should be replaced with the literal password of your choice.

For the ease of use set your directory environment as follows:

```
$ export JBOSS_HOME=/your/jboss/home
$ export PETSTORE_HOME=/your/path/to/petstore-ejb3
$ export ARM_HOME=/your/arm/home
$ export INSTRUMENTATION_HOME=/your/instrumentation/home
$ export JBOSS_INSTR_HOME=${INSTRUMENTATION_HOME}/JBoss-ejb3-ARM-Instrumentation
$ export TOMCAT_INSTR_HOME=${INSTRUMENTATION_HOME}/Tomcat-ARM-Instrumentation
```

Delete the file `hsqldb-ds.xml` from `$JBOSS_HOME/server/all/deploy` and replace it with the file `mysql-ds.xml` from `$JBOSS_INSTR_HOME/conf/jboss`. Open `mysql-ds.xml` and configure connection-url, user-name and password:

```
<connection-url>jdbc:mysql://yourhost:3306/jboss</connection-url>
<user-name>jboss</user-name>
<password>password</password>
```

Delete the files `hsqldb-jdbc2-service.xml` and `hsqldb-jdbc-state-service.xml` from `$JBOSS_HOME/server/all/deploy-hasingleton/jms`. Replace them with the files `mysql-jdbc2-service.xml` and `mysql-jdbc-state-service.xml` from `$JBOSS_INSTR_HOME/conf/jboss`.

Download the JDBC driver for MySQL from <http://dev.mysql.com/downloads/connector/j/3.0.html> and copy `mysql-connector-java-3.0.x-ga-bin.jar` to `$JBOSS_HOME/server/all/lib`.

Import the Petstore SQL tables into the JBoss database:

```
$ mysql -ujboss -p jboss < $PETSTORE_HOME/sql/*.sql
```

Finally configure the mail service for JBoss by editing the file `mail-service.xml` from `$JBOSS_INSTR_HOME/conf/jboss`. Change the following attributes according to your available POP3 and SMTP servers:

```
[...]  
<attribute name="User">your-username</attribute>  
<attribute name="Password">your-password</attribute>  
[...]  
<configuration>  
    [...]  
    <property name="mail.pop3.host" value="your-pop3-host"/>  
    <property name="mail.smtp.host" value="your-smtp-host"/>  
    [...]  
</configuration>  
[...]
```

You may want to change the attribute `mail.from` as well.

3.2 Building Petstore

Edit the file `$PETSTORE_HOME/build.properties`.

Configure `jboss.home` and `jboss.deploy.dir` according to your local environment, where `jboss.deploy.dir` equals `$JBOSS_HOME/server/all/deploy`.

Change to the petstore home directory, execute the following commands:

```
$ ant cleanall  
$ ant build_all
```

3.3 Deployment

Start JBoss with its standard configuration:

```
$ $JBOSS_HOME/bin/run.sh -c all
```

Copy the file `petstore.ear` from `$PETSTORE_HOME/build/lib` to `$JBOSS_HOME/server/all/deploy`.

If everything went right, `petstore` now is available under `http://localhost:8080/petstore`.

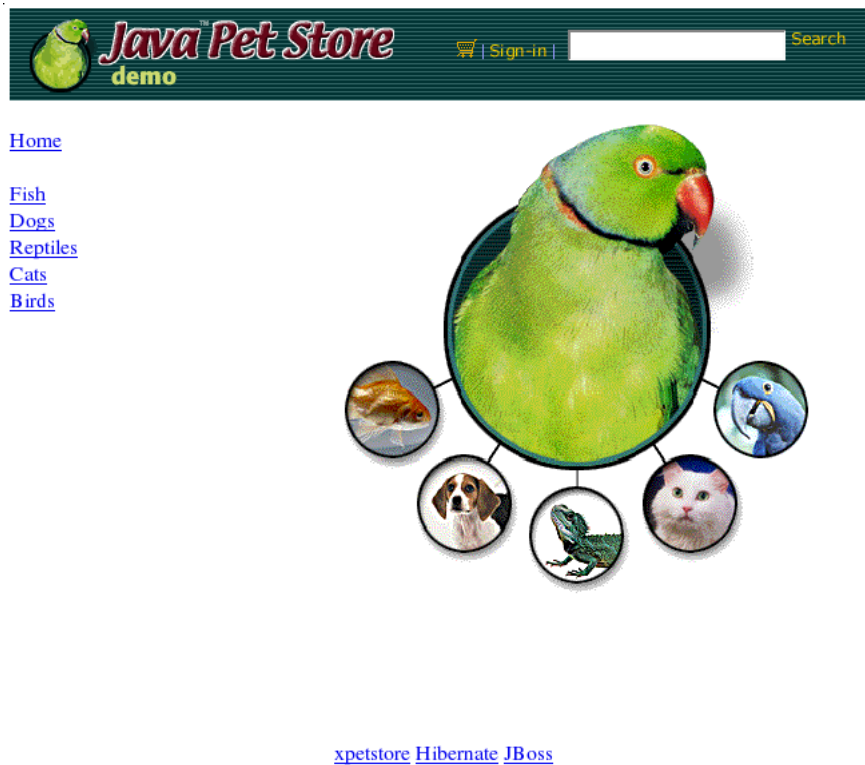


Figure 3.1: The Petstore Sample Application

Please shutdown the JBoss Application Server to avoid complications within the further configuration.

4 Instrumentation

4.1 ARM libraries

Within this guide, the Opengroup ARM 4.0 SDK was chosen for ARM instrumentation. Of course, other ARM Distributions can be used as well, though the build properties, interceptor constants and instrumentation meta data for each instrumentation have to be configured properly.

The following steps are required for setting up the Opengroup ARM 4.0 SDK. Other implementations may need different configuration.

Please download the Opengroup ARM 4.0 SDK from <http://www.opengroup.org/tech/management/arm> and build the ARM libraries according to the ARM documentation.

Copy the ARM libraries `arm40.jar` and `arm40sdk.jar` from `$ARM_HOME/java` to `$JBOSS_HOME/server/all/lib`.

Create a directory `$ARM_HOME/shared` and add it to your `LD_LIBRARY_PATH`:

```
$ export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ARM_HOME/shared
```

The following shared libraries have to be copied into this directory:

- `libarm4.so` from `$ARM_HOME/libarm4`
- `libarm4sdk.so` from `$ARM_HOME/c/arm4sdk`
- `libarm4jni.so` from `$ARM_HOME/java/jni`

4.2 JBoss instrumentation build

The ARM-Instrumentation uses additional interceptors to measure application response times. Therefore, the JBoss ARM interceptor classes have to be defined and interceptor stacks have to be added to the JBoss configuration. The file `ejb3-interceptors-aop.xml` from `$JBOSS_INSTR_HOME/conf` contains all necessary changes. Please copy this file to `$JBOSS_HOME-server/all/deploy` and overwrite the current version or copy and paste the marked ARM interceptor entries and the additional interceptor stack definitions.

Change to the directory `$JBOSS_INSTR_HOME/build` and edit the file `build.properties`. Replace the values for the following properties according to your environment:

```
jboss.home=/your/jboss/homedir
arm.home=/your/arm/homedir
arm.interfaces.lib=/your/arm/arm.jar
arm.vendor.lib=/your/arm/yourarmlibs.jar
```

Finally, build the instrumentation by executing the following commands in the same dir:

```
$ ant cleanall
$ ant packaging
```

4.3 Measure certain methods in Petstore

In most cases you don't want to measure all invocations to every EJB deployed, but only certain method calls that are of special interest. For these purposes, edit the file `ejb3-interceptors-aop.xml` from `$JBOSS_INSTR_HOME/conf`.

Determine whether you want to instrument a stateful or a stateless bean and search for the following tag:

```
<domain name="ARM [Stateful|Stateless] Bean" [...]>
```

The standard configuration is:

```
<bind pointcut="execution(public * *->*(..))">
```

This means that all methods in all EJBs of this type are instrumented. To change this, just replace the wildcards (*) with a method of your choice, for example:

```
<bind pointcut="execution(public void de.[...].CartLocal->addItem(..))">
```

Of course, you can use wildcards to spread an instrumentation over a group of methods or EJBs.

4.4 Tomcat instrumentation build

JBoss uses an embedded Tomcat server for displaying JSP's and hosting servlets. For servlet instrumentation edit the file `build.properties` from `$TOMCAT_INSTR_HOME/build` and change the following properties:

```
jboss.dir=/your/jboss/homedir
jboss.profile=all
arm.home=/your/arm/homedir
arm.interfaces.lib=/your/arm/arm.jar
arm.vendor.lib=/your/arm/yourarmlibs.jar
```

The JBoss profile `all` is just a default value. Of course you may enter the JBoss profile name of your choice. Please do not uncomment the `tomcat.dir` property, since we are using the build-in Tomcat functionality of JBoss.

The file `instrumentation-ejb.xml` from `$TOMCAT_INSTR_BUILD/meta-data/instrumentation` holds some configuration parameters for the instrumentation.

Deploy the instrumentation by executing the following commands in the same directory:

```
$ ant cleanall
$ ant installEJB3embedded
```

In case you like to uninstall the Tomcat instrumentation, enter `uninstallEJB3embedded` instead.

For the settings to take effect please restart the JBoss server.

4.5 Binding the ARM instrumentation

Petstore has to be configured for using the ARM instrumentation. Therefore you have to bind the ARM interceptors to the Petstore Enterprise Java Beans (or certain method calls) you would like to measure. Edit the file `jboss.xml` from `$PETSTORE_HOME/src/java/META-INF`. For an example only the two beans `Cart` and `Petstore` are going to be bound to our ARM instrumentation. There are two different kinds of ARM measurement interceptors: Those for *stateful* session beans and those *stateless* session beans. Since there has to be saved information in the Petstore shopping cart, `Cart` is a *stateful* bean. The Petstore bean itself does not contain any individual information states so in this case you have to configure the interceptor for a *stateless* bean.

Please Insert the following XML code between the `jboss` tags:

```
<enterprise-beans>
<session>
  <ejb-name>Cart</ejb-name>
  <remote-binding>
    <interceptor-stack>
      ARMClusteredStatefulSessionClientInterceptors
    </interceptor-stack>
  </remote-binding>
  <aop-domain-name>ARM Stateful Bean</aop-domain-name>
</session>
<session>
  <ejb-name>Petstore</ejb-name>
  <remote-binding>
    <interceptor-stack>
      ARMClusteredStatelessSessionClientInterceptors
    </interceptor-stack>
  </remote-binding>
  <aop-domain-name>ARM Stateless Bean</aop-domain-name>
</session>
</enterprise-beans>
```

Change to the petstore home directory, execute the following commands:

```
$ ant cleanall
$ ant build_all
```

4.6 Deploying the instrumented Petstore

Copy the file `petstore.ear` from `$PETSTORE_HOME/build/lib` to `$JBOSS_HOME/server/all/deploy`, overwriting the previous, non-instrumented version.

If everything went right, your JBoss startup log should show some ARM entries when initiating any servlet actions in the Petstore web interface under

`http://localhost:8080/petstore:`

```
arm_start_transaction()
time=15:50:02 05/03/07
application instance handle=8
transaction id=10000000 00000000 00000000 00000000
parent correlator: none
flags=NONE
application name=Petstore
transaction name=ContextRequestTransaction (Petstore)
transaction instance handle=7
arm_stop_transaction()
time=15:50:02 05/03/07
transaction instance handle=7
transaction status=GOOD
flags=NONE
application name=Petstore
transaction name=ContextRequestTransaction (Petstore)
```

Figure 4.1: Logged ARM transactions

Congratulations, you have instrumented your first sample application.