



PV243

Clustering & Scalability

LAB SESSION

Radoslav Husar

+ special guest Ondrej Chaloupka

Quality Engineering, Red Hat

May 10, 2012

Agenda

4 demos

- Building JGroups chat demo
- Building HA WebApp
 - Verifying failover
- Remote EJB load-balancing
- Clustered Remote Stateful Session Bean demo



Priming Build

```
$ git clone git://github.com/qa/pv243.git
```

```
$ cd pv243/lesson06-clustering
```

```
$ mvn clean install
```

or tag:

```
$ git pull --rebase upstream master
```

```
$ git checkout clustering-priming
```





Chat over JGroups

Chat

- Task

Finish implementation of a simple chat using JGroups API using default UDP stack. Notify when new member joins a chat and display messages from all member and also send messages to all members. No limits on bonus features.

- Checkout `clustering-1` tag to start working





Highly-Available WebApp

WebApp

- Task:

Implement missing pieces in the WebApp to make it highly-available. Let the simple servlet return number of times the servlet has been invoked so that in case of failover it will enable us to verify if the session state is as expected.

- Checkout `clustering - 3` tag to start working



Deploy the App

- Unzip 2 separate installations
- Build war and copy to standalone/deployments
- `./standalone.sh -c standalone-ha.xml`
- `./standalone.sh -c standalone-ha.xml \`
`-Djboss.socket.binding.port-offset=100 \`
`-Djboss.node.name=rhusar2`
- Why using port offset?
- Simulate failover





Load-balancing Remote Stateless Session Beans and Remote Cluster-Aware Stateful Session Beans

Clustered beans

- Task
Create clustered Stateless Session Bean and clustered Stateful Session Bean. Create a remote EJB clients which connect to a JBoss cluster and makes calls to the beans. Watch the load-balancing of SLSB and SFSB failover.
- Checkout `clustering-5` tag to start working



Steps to run the app (1)

- Deploy jar with beans to all servers in your cluster
- Start servers
You need to name servers differently and run one with a port offset.
 - `./bin/standalone.sh -c standalone-ha.xml -Djboss.node.name=server1`
 - `./bin/standalone.sh -c standalone-ha.xml -Djboss.node.name=server2 -Djboss.socket.binding.port-offset=100`



Steps to run the app (2)

- Run the remote client
 - you need to have on classpath:
 - `$JBOSS_HOME/bin/client/jboss-client.jar`
 - interface of the bean which you'll call remotely
 - `cd client-side`
 - `mvn exec:java`
 - Dexec.class=cz.muni.fi.pv243.lesson6.ejb.remote.client.StatelessRemoteClient
 - `mvn exec:java`
 - Dexec.class=cz.muni.fi.pv243.lesson6.ejb.remote.client.StatefulRemoteClient



~FINISHED~
~~Happy Weekend~~

