PV243

Clustering & Scalability

LAB SESSION

Radoslav Husár
Senior Software Engineer
Red Hat

April 24, 2017

Agenda

4 demos

- Trying a JGroups chat demo
- Building HA WebApp
  - Verifying failover
- Building a monitoring Servlet using a CommandDispatcher

Optional

- *Remote EJB load-balancing*
- *Clustered Remote Stateful Session Bean demo*

Priming Build

$ git clone https://github.com/qa/pv243-a4m36jee-2016-clustering-seminar.git

$ git checkout clustering-00

$ cd chat-jgroups

$ mvn clean install

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 “room” and display messages from all member. Send messages to all members. You can add bonus features but keep the message format the same.

- Checkout clustering-00 tag to start working

Demo

$ cd chat-jgroups

$ mvn clean install

$ sudo ip route add 224.0.0.0/4 dev lo table local

$ mvn exec:java -Djava.net.preferIPv4Stack=true
-Duser.name=`whoami`

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. Add ?readonly=true and ?invalidate=true options.

- Checkout clustering-01 tag to start working

Deploy the App

- `Build + 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=node2
- Why use port-offset?
- How to simulate failover?

Command Dispatcher

Cluster Monitoring Servlet using Command Dispatcher

- Task:

  Implement missing pieces in the WebApp to make it display memory usage of the cluster nodes. Use CommandDispatcher to retrieve memory heap usage from all cluster nodes and display it in a simple page generated by a servlet.

- Checkout clustering-02 tag to start working

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

Clustered beans

- Task
  Implement clustered Stateless Session Bean and clustered Stateful Session Bean. Implement a remote EJB clients which connects to WildFly cluster and invokes operations on the beans. Demonstrate the load-balancing of SLSB and SFSB failover.
- Checkout clustering-03 tag to start working

Steps to run the app (1)

- Deploy jar containing the beans to all servers in the 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=node1
  - ./bin/standalone.sh -c standalone-ha.xml -Djboss.node.name=node2 -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.mainClass=cz.muni.fi.pv243.seminar.clustering.ejb.remote.client.StatelessRemoteClient
  - mvn exec:java -Dexec.mainClass=cz.muni.fi.pv243.seminar.clustering.ejb.remote.client.StatefulRemoteClient

Thank you!