JoramMQ Entreprise
tools and services for the JORAM user

OW2 Conference
November 28th, 2012

Serge Lacourte – CEO
serge.lacourte@scalagent.com

www.scalagent.com
Plan

JORAM

JoramMQ Entreprise

Benefits for the project actors

- UC: redesign of a WebSphereMQ application with JoramMQ
- prior to the decision
- early design phase
- development phase
- operation phase

Conclusion
An up-to-date MOM

**Open-source & 100% Java MOM**
- portable and light (jars < 1Mo), classical & embedded configurations
- certified JMS 1.1 standalone & J2EE (with JOnAS)
- compliant AMQP 0.9.1, AMQP 1.0 in progress
- published by OW2 under LGPL licence since 2000

**Efficient and reliable**
- 7,000 to 216,000 msgs/s depending on QoS, on this laptop
- hundreds of operational usages all over the world
- supported by ScalAgent DT

**Configurable and extensible**
- configurable deployment architecture, from centralised to peer to peer
- internal OSGi architecture
- open and extensible architecture
An industrial product

A mature open-source project
- 2 to 3 versions a year
  - follows the standards, useful additional functions
  - internal architecture regularly updated
- ready for future needs
  - basically distributed architecture

Chosen in various operational contexts
- France Télécom
  - MOM in the Information System, besides WebSphereMQ
- CNES – CS / Thales
  - reference implementation of the international standard CCSDS/MAL
  - Phoebus product line, monitoring
- French Home Office
  - managing the car certificates between manufacturers, vendors & Home Office
- French Air Force – CSSI
  - military communication system with strong geographical distribution
- RFID middleware
Strong and international users community

- 800 identified users, 62 countries, 5 continents
ScalAgent support offering

Professional support
- with guaranteed response time
- for development and operational phases
- JORAM extensions and expertise

Additional tools and services
- integration to other environments
  - software, configuration & documentation for J2EE, Spring, Talend
- developer's console
  - makes the developer's life easier
- operational console
  - provides meaningful indicators to the operator
- MQPerf (SaaS)
  - helps the project leader managing performance related issues
**Business model**

**JORAM open source**
- JoramMQ runtime based on JORAM latest open source unchanged
- JORAM keeps on evolving (conformance to standards, ...)

**JoramMQ for a fixed price annual subscription**
- for a single project
  - not proportional to a number of CPUs
  - a single operational deployment
- including
  - documentation (reference manual)
  - license for the additional tools (and associated runtime), unlimited in the project
  - unlimited usage of service MQPerf for the project
  - professional support with guaranteed response time
Use case

First step of migration from WebSphere MQ

- WebSphere MQ as a former strategic choice
  - a number of applications with WebSphere MQ servers & clients
  - license fees for each server & client
- refactoring of an application
  - new functions
  - aging environment

Choose JoramMQ Entreprise

- license gains
  - fixed lower cost for the project, not proportional to an unknown number of CPUs
- incremental and non revolutionary
  - Joram – WebSphere MQ bridge keeps the link with the enterprise bus
- France Telecom has done it
- JoramMQ helps
Prior to the decision

JMQ Benefits

can JoramMQ actually do the job?

Paper study

- **Functional capabilities**
  - JMS compliance → pass the TCK from Oracle

- **Performances**
  - published figures are good, can they be trusted?

Prototyping

- efficient, but requires time and expertise
Prior to the decision

MQPerf community

Figures you can trust
- optimal figures for an extended set of usage scenarios
- tests actually performed with a standard JORAM in your environment

Reduce the performance risk
- even before the project begins

<table>
<thead>
<tr>
<th>throughput (msg/s)</th>
<th>persistent</th>
<th>transient</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100 B/msg</td>
<td>10 kB/msg</td>
</tr>
<tr>
<td>queue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>localCF</td>
<td>80 783</td>
<td>12 477</td>
</tr>
<tr>
<td>tcpCF</td>
<td>41 613</td>
<td>6 821</td>
</tr>
<tr>
<td>topic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>localCF</td>
<td>78 894</td>
<td>11 854</td>
</tr>
<tr>
<td>tcpCF</td>
<td>46 204</td>
<td>6 769</td>
</tr>
</tbody>
</table>
Early design phase

how may I best use JoramMQ?

- JMS specifies MOM concepts and a client interface
  - the actual definition of a deployment architecture is not in the scope of the standard
  - each JMS implementation is specific
  - MQSeries implementation comes from long ago, might be best not followed

- Prototyping
  - efficient, but requires time and expertise

- Use and hope for the best
MQPerf standard

Analyze JORAM's behaviour over time

- key internal indicators over time for an extended set of usage scenarios
- tests actually performed on your system, under heavy load conditions

Help you choose your best configuration from start
how can I do a better job more quickly?

JoramMQ developer's console

Monitor and control JoramMQ

- immediate access to middleware specific indicators
  - MOM & JoramMQ specific navigation & presentation
- operations on the running system
**Operation phase**

**JMQ Benefits**

Does JoramMQ become a black box for the operator?

**JoramMQ operator's console**

**Monitor JoramMQ**

- simple yet relevant middleware specific indicators
- integration in a global administration console (Nagios)

---

**Current Network Status**

<table>
<thead>
<tr>
<th>Network Status</th>
<th>Status</th>
<th>Last Check</th>
<th>Duration</th>
<th>Attempts</th>
<th>Status Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>OK</td>
<td>11-22-2011 15:58:08</td>
<td>74% 26,39 Mhz</td>
<td>/4/</td>
<td>CPU - load average: 0.65, 0.50, 0.34</td>
</tr>
<tr>
<td>Disk</td>
<td>OK</td>
<td>11-22-2011 15:58:08</td>
<td>74% 26,39 Mhz</td>
<td>/4/</td>
<td>CPU - load average: 0.65, 0.50, 0.34</td>
</tr>
<tr>
<td>Netstat</td>
<td>OK</td>
<td>11-22-2011 15:58:08</td>
<td>74% 26,39 Mhz</td>
<td>/4/</td>
<td>CPU - load average: 0.65, 0.50, 0.34</td>
</tr>
<tr>
<td>Joram's listener message delivery</td>
<td>OK</td>
<td>11-22-2011 15:58:08</td>
<td>74% 26,39 Mhz</td>
<td>/4/</td>
<td>CPU - load average: 0.65, 0.50, 0.34</td>
</tr>
</tbody>
</table>

**Host Status Details**

<table>
<thead>
<tr>
<th>Host Status Details</th>
<th>Status</th>
<th>Last Check</th>
<th>Duration</th>
<th>Attempts</th>
<th>Status Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host Status Details</td>
<td>OK</td>
<td>11-22-2011 15:58:08</td>
<td>74% 26,39 Mhz</td>
<td>/4/</td>
<td>CPU - load average: 0.65, 0.50, 0.34</td>
</tr>
<tr>
<td>Joram's listener message delivery</td>
<td>OK</td>
<td>11-22-2011 15:58:08</td>
<td>74% 26,39 Mhz</td>
<td>/4/</td>
<td>CPU - load average: 0.65, 0.50, 0.34</td>
</tr>
</tbody>
</table>

**Service Status Details For All Hosts**

<table>
<thead>
<tr>
<th>Service Status Details</th>
<th>Status</th>
<th>Last Check</th>
<th>Duration</th>
<th>Attempts</th>
<th>Status Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Status Details</td>
<td>OK</td>
<td>11-22-2011 15:58:08</td>
<td>74% 26,39 Mhz</td>
<td>/4/</td>
<td>CPU - load average: 0.65, 0.50, 0.34</td>
</tr>
<tr>
<td>Joram's listener message delivery</td>
<td>OK</td>
<td>11-22-2011 15:58:08</td>
<td>74% 26,39 Mhz</td>
<td>/4/</td>
<td>CPU - load average: 0.65, 0.50, 0.34</td>
</tr>
</tbody>
</table>

---

**SCALAGENF**

© ScalAgent Distributed Technologies – 2001-2012
Conclusion

- **JORAM** is a great middleware
  - efficient, reliable, configurable, extensible
  - and open-source

- **JoramMQ** enhances it with tools and services
  - to help the project actors at all phases
    - decision, design, development, operation
  - to provide a professional support with guaranteed response time

  to complete the project more securely and rapidly