JMS Performance Comparison

Performance Comparison for Publish Subscribe Messaging
## Contents

Executive Summary.............................................................................................................................................. 3

1. Test Methodology............................................................................................................................................ 3
   1.1 Test Conditions .............................................................................................................................................. 3
   1.2 Test Scenario’s ............................................................................................................................................... 4
   1.3 Test Duration ............................................................................................................................................... 5
   1.4 Environment Setup ....................................................................................................................................... 5
   1.5 Measurement ............................................................................................................................................... 5
   1.6 Topology .................................................................................................................................................... 5

2. Performance Results.......................................................................................................................................... 6
   2.1 Topic Scalability ........................................................................................................................................... 6
   2.2 Server Scalability ........................................................................................................................................ 7
   2.3 Persistent Publisher, Durable Subscribers ................................................................................................. 8
   2.4 Non-Persistent Publisher, Non-Durable Subscribers .................................................................................. 9

3. System Configuration..................................................................................................................................... 10
   3.1 Hardware Configuration ............................................................................................................................... 10
   3.2 Software Configuration ............................................................................................................................... 10

About Fiorano Software.................................................................................................................................... 11
Executive Summary

This paper presents a performance analysis of publish/subscribe messaging throughput of FioranoMQ® 10.0.0, Tibco EMS v8.0, ActiveMQ 5.9.0, HornetQ 2.4.0, OpenMQ5.0.0, RabbitMQ 3.2.0, and IBM WebSphere MQ 7.5. This analysis provides a head-to-head comparison of these products designed to illustrate the products’ relative performance characteristics for several messaging scenarios.

The test scenarios represent stress level conditions for real world applications. The tests examine performance under load, where a single message broker is required to support many publishers and subscribers.

The testing tool used for these performance tests is highly configurable and can be used to test any JMS broker. Also, this tool allows running and measurement of a wide range of test definitions.

Do note that the different configurations or performance tuning of any JMS broker may potentially yield throughput gains (or losses) for any of these tests. Changes to the test definitions will produce different throughput rates and this should be considered when attempting to map these results to expected performance of any particular JMS application.

All the JMS brokers were configured with out-of-the-box default values and no performance specific product tuning was carried out for any of them. It’s observed from the detailed results that the relative performance of the message brokers varies under various conditions. While performance analysis should always be conducted for a particular messaging environment, the results of these tests suggest that FioranoMQ will deliver messages more efficiently in demanding messaging environments in today’s real-time enterprises.

1. Test Methodology

All the tests described in this section were carried out using a highly configurable testing tool. This tool allows running and measurement of a wide range of test definitions.

This section begins with a brief description of test conditions which are created to test the JMS server. This is followed by a section that describes the tests and their results. The final section provides a brief description of the hardware and software configurations.

1.1 Test Conditions

All the tests were conducted under the following conditions:

- Each client runs on a separate JMS connection.
- All test results are recorded after the client connections have been established and publishers/subscribers and other objects had been created.
- All tests were run multiple times to assure repeatability.
• Performance was measured under maximum load by publishing as many messages as possible using default settings of the server.
• During the test, no other applications were running and using resources on the system under test.
• Dups_ok was used by all consumers.
• All servers were tested in the default mode - which meant running IBM MQ, Tibco EMS in "Evaluation" (non-HA) mode, ActiveMQ 5.9 (default configuration mode), FioranoMQ and others in normal production ready (non-HA) mode.

1.2 Test Scenario's

The tests were conducted for the most popular messaging models employed using Topics in JMS.

Non-Persistent Publishers & Non-Durable Subscribers

This model is typically used by applications which are exchanging high volume of messages and have a requirement of minimum latency.

Persistent Publishers & Durable Subscribers

This model is typically employed by applications which need maximum level of redundancy and need once and only once guarantee of message delivery irrespective of the client or server failure.

The following tests were conducted based on typical customer use-cases:

a. **Topic Scalability Tests**: These tests observe the performance characteristics of JMS server with varying # of Pub/Sub clients on a fixed number of topics. The results illustrate the scalability of JMS server as more clients (all working on same JMS Topic) are employed.

b. **Server Scalability Tests**: These tests observe the performance characteristics of JMS server with varying # of Topics with fixed # of Pub/Sub clients per topic. The results illustrate the scalability of JMS server as more clients (each working on independent JMS Topics) are employed.

c. **Persistent Producer, Multiple Durable Consumers**: These tests observe the performance characteristics of JMS server when a single persistent publisher is used to publish messages to multiple durable subscribers.

d. **Non-Persistent Producer, Multiple Non-Durable Consumers**: These tests observe the performance characteristics of JMS server when a single non-persistent publisher is used to publish messages to multiple non-durable subscribers.

In order to generate the highest amount of message load, no processing time is introduced at either side of the client message exchanges. Allowing publishers to send messages as fast as possible in this manner enables these tests to expose the maximum message throughput rates. The test message size was chosen to reflect use cases observed in typical customer proof of concept scenarios.
1.3 Test Duration

All test scenarios were executed for a total of five minutes. Each test execution comprised of five, sixty-second intervals. The first two and last intervals were considered **ramp-up** and **ramp-down** intervals, respectively.

Ramp-up intervals are times during which the systems are increasing their message handling capacities, typically via resource allocation in response to the newly introduced client load.

Ramp-down intervals are times in which the systems are decreasing their capacity in response to decreased client loads that result from test completion. The remaining five intervals were considered **measurement** intervals during which steady-state performance was achieved.

Steady-state is the condition in which message rates exhibit negligible change.

1.4 Environment Setup

All client connections, publishers and subscribers were established before any testing ramp-up periods were started.

Each product’s message store, log files, queues, and topics were deleted and recreated therefore the broker stopped and restarted between each test.

1.5 Measurement

Performance data was collected during the five-minute measurement intervals only. No data was collected during ramp-up and ramp-down intervals. Tests were run twice, and measurements were averaged to obtain final results.

1.6 Topology

The topology contains two machines: One for running the clients and the other for running the server. The system configurations are detailed later in this document. These systems having 1GB NIC cards were interconnected using a 1 GBPS peer to peer connection.
2. Performance Results

2.1 Topic Scalability

<table>
<thead>
<tr>
<th>P/S/T</th>
<th>Message Type</th>
<th>Subscriber Type</th>
<th>Message Size (bytes)</th>
<th>Subscription Rate (messages / sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Fiorano MQ 10</td>
</tr>
<tr>
<td>1/1/1</td>
<td>Non-Persistent</td>
<td>Non-Durable</td>
<td>1024</td>
<td>67876</td>
</tr>
<tr>
<td>10/10/1</td>
<td>Non-Persistent</td>
<td>Non-Durable</td>
<td>1024</td>
<td>99726</td>
</tr>
<tr>
<td>25/25/1</td>
<td>Non-Persistent</td>
<td>Non-Durable</td>
<td>1024</td>
<td>103094</td>
</tr>
<tr>
<td>50/50/1</td>
<td>Non-Persistent</td>
<td>Non-Durable</td>
<td>1024</td>
<td>103660</td>
</tr>
</tbody>
</table>

Topic Scalability

![Graph showing performance results for different systems and message sizes]
### 2.2 Server Scalability

<table>
<thead>
<tr>
<th>P/S/T, Message Type, Subscriber Type</th>
<th>Message Size (bytes)</th>
<th>Subscription Rate (messages / sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/1/1, Non-Persistent, Non-Durable</td>
<td>1024</td>
<td>Fiorano MQ 10: 67876, Rabbit MQ 3.2.0: 14298, ActiveMQ: 39137, IBM MQ: 3176, OpenMQ: 23476, Tibco EMS: 31905, HornetQ: 52581</td>
</tr>
<tr>
<td>10/10/10, Non-Persistent, Non-Durable</td>
<td>1024</td>
<td>Fiorano MQ 10: 76298, Rabbit MQ 3.2.0: 37148, ActiveMQ: 39298, IBM MQ: 12165, OpenMQ: 26721, Tibco EMS: 67035, HornetQ: 49459</td>
</tr>
<tr>
<td>25/25/25, Non-Persistent, Non-Durable</td>
<td>1024</td>
<td>Fiorano MQ 10: 83270, Rabbit MQ 3.2.0: 35528, ActiveMQ: 38062, IBM MQ: 21740, OpenMQ: 26923, Tibco EMS: 64825, HornetQ: 45805</td>
</tr>
<tr>
<td>50/50/50, Non-Persistent, Non-Durable</td>
<td>1024</td>
<td>Fiorano MQ 10: 84330, Rabbit MQ 3.2.0: 35780, ActiveMQ: 39179, IBM MQ: 22827, OpenMQ: 24911, Tibco EMS: 79165, HornetQ: 43033</td>
</tr>
</tbody>
</table>

**Server Scalability**

![Server Scalability Chart](chart.png)

*Legend:*
- **FioranoMQ**
- **Rabbit MQ**
- **ActiveMQ**
- **IBM MQ**
- **Open MQ**
- **Tibco EMS**
- **HornetQ**
### 2.3 Persistent Publisher, Durable Subscribers

<table>
<thead>
<tr>
<th>P/S/T</th>
<th>Message Type</th>
<th>Subscriber Type</th>
<th>Message Size (bytes)</th>
<th>Subscription Rate (messages / sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/1/1</td>
<td>Persistent</td>
<td>Durable</td>
<td>1024</td>
<td>Fiorano MQ 10: 3818 Rabbit MQ 3.2.0: 9152 ActiveMQ 5.9.0: 1192 IBM MQ 7.5.0_2: 1097 Open MQ 5.0.0: 3375 Tibco EMS 8.0: 4195 HornetQ 2.4.0: 1767</td>
</tr>
<tr>
<td>1/10/1</td>
<td>Persistent</td>
<td>Durable</td>
<td>1024</td>
<td>Fiorano MQ 10: 33894 Rabbit MQ 3.2.0: 9831 ActiveMQ 5.9.0: 8206 IBM MQ 7.5.0_2: 6025 Open MQ 5.0.0: 6005 Tibco EMS 8.0: 16785 HornetQ 2.4.0: 14571</td>
</tr>
<tr>
<td>1/25/1</td>
<td>Persistent</td>
<td>Durable</td>
<td>1024</td>
<td>Fiorano MQ 10: 47448 Rabbit MQ 3.2.0: 9774 ActiveMQ 5.9.0: 13383 IBM MQ 7.5.0_2: 8341 Open MQ 5.0.0: 9482 Tibco EMS 8.0: 28003 HornetQ 2.4.0: 26941</td>
</tr>
<tr>
<td>1/50/1</td>
<td>Persistent</td>
<td>Durable</td>
<td>1024</td>
<td>Fiorano MQ 10: 50519 Rabbit MQ 3.2.0: 9734 ActiveMQ 5.9.0: 7416 IBM MQ 7.5.0_2: 5609 Open MQ 5.0.0: 9071 Tibco EMS 8.0: 33804 HornetQ 2.4.0: 40180</td>
</tr>
</tbody>
</table>

**Persistent Publisher, Durable Subscriber**

![Graph showing subscription rate for different publishers and subscribers](image-url)
2.4 Non-Persistent Publisher, Non-Durable Subscribers

<table>
<thead>
<tr>
<th>P/S/T</th>
<th>Message Type</th>
<th>Subscriber Type</th>
<th>Message Size (bytes)</th>
<th>Fiorano MQ 10</th>
<th>Rabbit MQ 3.2.0</th>
<th>Active MQ 5.9.0</th>
<th>IBM MQ 7.5.0_2</th>
<th>Open MQ 5.0.0</th>
<th>Tibco EMS 8.0</th>
<th>HornetQ 2.4.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/1/1</td>
<td>Non Persistent</td>
<td>Non Durable</td>
<td>1024</td>
<td>67876</td>
<td>14298</td>
<td>39137</td>
<td>3176</td>
<td>23476</td>
<td>31905</td>
<td>52581</td>
</tr>
<tr>
<td>1/10/1</td>
<td>Non Persistent</td>
<td>Non Durable</td>
<td>1024</td>
<td>101096</td>
<td>13676</td>
<td>52868</td>
<td>20059</td>
<td>23307</td>
<td>93083</td>
<td>68741</td>
</tr>
<tr>
<td>1/25/1</td>
<td>Non Persistent</td>
<td>Non Durable</td>
<td>1024</td>
<td>104637</td>
<td>13578</td>
<td>52148</td>
<td>29941</td>
<td>6103</td>
<td>101425</td>
<td>66163</td>
</tr>
<tr>
<td>1/50/1</td>
<td>Non Persistent</td>
<td>Non Durable</td>
<td>1024</td>
<td>101853</td>
<td>13491</td>
<td>49046</td>
<td>35905</td>
<td>9071</td>
<td>102432</td>
<td>71002</td>
</tr>
</tbody>
</table>

Non Persistent Publisher, Non Durable Subscriber
3. System Configuration

3.1 Hardware Configuration

Server System
- Linux CentOS 2.6.18-92.el5 (x64)
- 2 Quad Core Intel(R) Xeon(R) CPU 5405 @ 2.00GHz
- 64 bit 16 GB RAM

Client System
- Linux CentOS 2.6.18-92.el5 (x64)
- Quad Core Intel(R) Xeon(R) CPU 5405 @ 2.00GHz
- 64 bit 16 GB RAM

Network Settings
- Client and Server were on the same network
- Network Speed: 1GBPS

3.2 Software Configuration

- Java Runtime Environment, Standard Edition (build 1.7.0_45-b18)
- FioranoMQ v 10.0.0
- RabbitMQ 3.2.0
- Tibco EMS v 8.0 (In persistent tests, the TIBCO topics were set to failsafe to ensure persistence to disk)
- ActiveMQ v 5.9.0
- HornetQ 2.4.0
- OpenMQ 5.0.0
- IBM WebSphere 7.5
ABOUT FIORANO SOFTWARE

Founded in 1995, Silicon Valley based Fiorano is a USA (California) Corporation, a trusted provider of Digital Business Backplane and enterprise integration middleware, high performance messaging and peer-to-peer distributed systems. Fiorano powers real time, digital enterprises with bimodal integration and API Management strategy that leverages the best of systematic (centralized, high-control) and adaptive (federated, high-speed) approaches to deliver solutions across cloud, on-premise and hybrid environments. Fiorano operates through its worldwide offices and a global network of technology partners and value-added resellers.

Global leaders including AT&T Wireless, Boeing, British Telecom, Federal Bank, L’Oréal, McKesson, NASA, POSCO, Rabobank, Royal Bank of Scotland, Schlumberger, US Coast Guard and Vodafone have deployed Fiorano to drive innovation through open, standards-based, event-driven real-time solutions yielding unprecedented productivity.

To find out more about how Fiorano can help you meet your enterprise integration objectives, visit www.fiorano.com or e-mail sales@fiorano.com