

# Fiorano<sup>®</sup>

## JMS Performance Comparison

Performance Comparison for Publish Subscribe Messaging

Feb 2008

Copyright © 2006-2008

Fiorano Software Technologies Pvt. Ltd., All Rights Reserved

# Contents

---

<b>Contents .....</b>	<b>2</b>
1 Executive Summary .....	3
2. Test Methodology .....	3
2.1 Test Conditions .....	3
2.2 Test Scenario's.....	4
2.3 Test Duration.....	5
2.4 Environment Setup .....	5
2.5 Measurement.....	5
2.6 Topology .....	5
3. Performance Results .....	6
3.1 Server Scalability .....	6
3.2 Topic Scalability .....	7
3.3 Persistent Publisher, Durable Subscribers.....	8
3.4 Non-Persistent Publisher, Non-Durable Subscribers .....	9
4. System Configuration.....	10
4.1 Hardware Configuration.....	10
4.2 Software Configuration.....	10

## 1 Executive Summary

This paper presents a performance analysis of publish/subscribe messaging throughput of FioranoMQ 2008, Sonic MQ 7.0, Tibco EMS v4.4.0 and ActiveMQ 4.1.0, Jboss Messaging 1.4 SP1 and Sun JAVA MQ 4.1. 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 methodology and driving program were the ones developed by Sonic Software, Inc. and are available at

[http://www.sonicsoftware.com/products/sonicmq/performance\\_benchmarking/index.ssp](http://www.sonicsoftware.com/products/sonicmq/performance_benchmarking/index.ssp)

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.

## 2. 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.

### 2.1 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 SonicMQ, Tibco EMS in "Evaluation" (non-HA) mode, ActiveMQ 4.1.0 (default configuration mode) and FioranoMQ in normal production ready (non-HA) mode.

## 2.2 Test Scenario's

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

### I. 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.

### II. 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:

1. **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.
2. **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.
3. **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.
4. **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.

## 2.3 Test Duration

All test scenarios were executed for a total of eight minutes. Each test execution comprised of eight, 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.

## 2.4 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.

## 2.5 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.

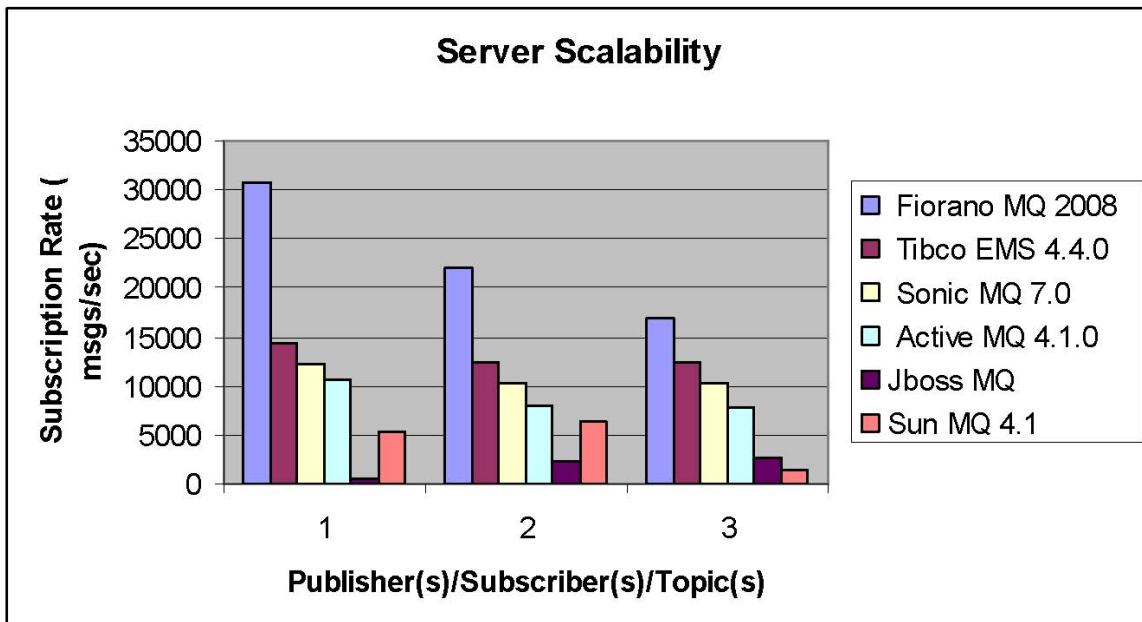
## 2.6 Topology

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.

### 3. Performance Results

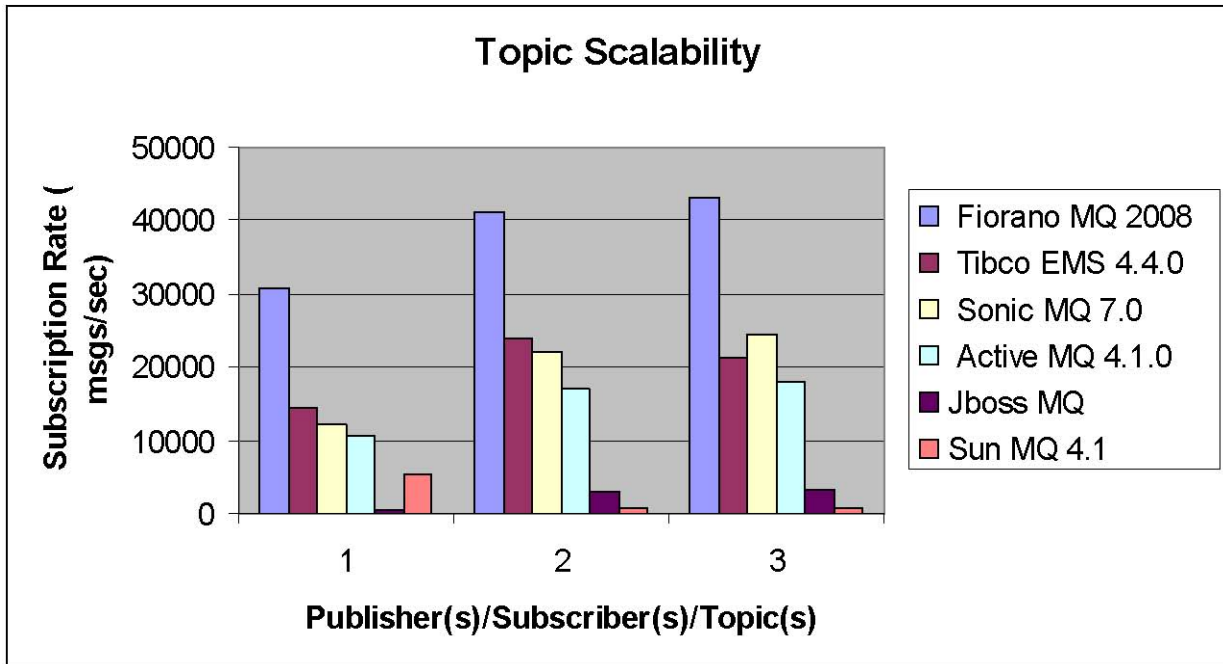
#### 3.1 Server Scalability

P/S/T	Message Type	Subscriber Type	Message	Subscription Rate (messages / sec)					
				FioranoMQ 2008	Tibco EMS 4.4.0	Sonic MQ 7.0	Active MQ 4.1.0	Jboss 1.4	Sun MQ 4.1
1/1/1	Non-Persistent	Non-Durable	1024	30655	14341	12246	10742	454	5340
10/10/10	Non-Persistent	Non-Durable	1024	22033	12472	10261	7938	2326	6353
25/25/25	Non-Persistent	Non-Durable	1024	16943	12444	10322	7761	2612	1359
50/50/50	Non-Persistent	Non-Durable	1024	14823	10278	7239	6021	1921	912



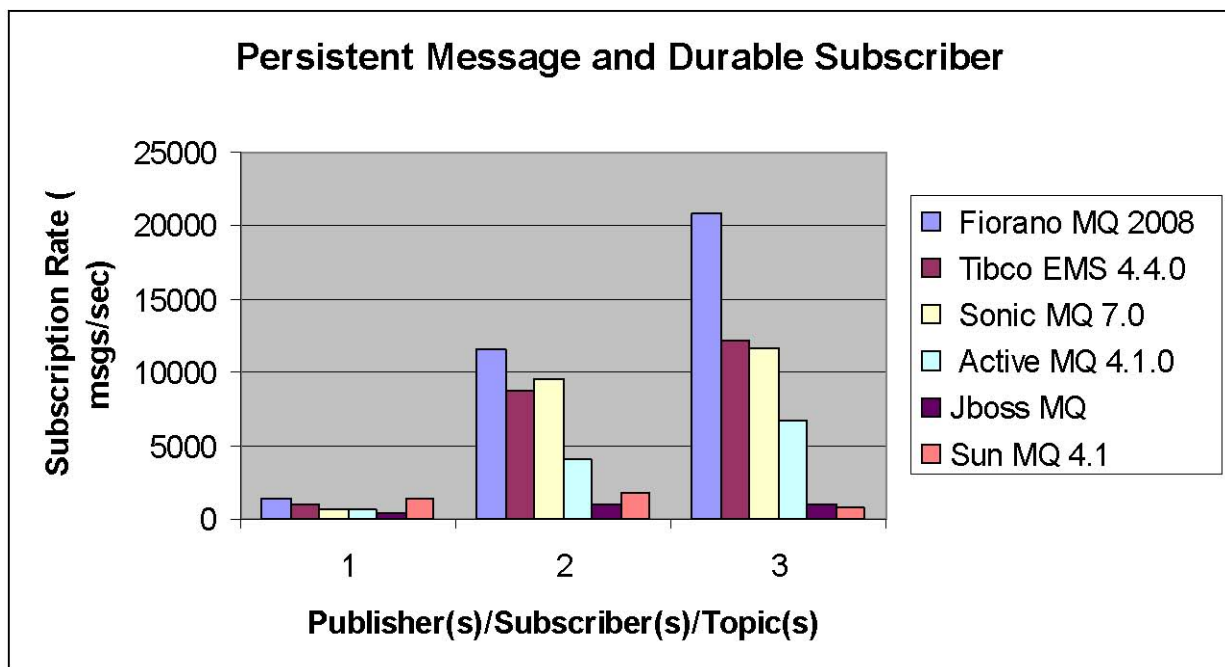
### 3.2 Topic Scalability

P/S/T	Message Type	Subscriber Type	Message	Subscription Rate (messages / sec)					
				FioranoMQ 2008	Tibco EMS 4.4.0	Sonic MQ 7.0	Active MQ 4.1.0	Jboss 1.4	Sun MQ 4.1
1/1/1	Non-Persistent	Non-Durable	1024	30655	14341	12246	10742	454	5340
10/10/1	Non-Persistent	Non-Durable	1024	41081	23809	22177	17011	2970	636
25/25/1	Non-Persistent	Non-Durable	1024	43184	21230	24331	17922	3362	818
50/50/1	Non-Persistent	Non-Durable	1024	38723	17281	19212	14038	2129	621



### 3.3 Persistent Publisher, Durable Subscribers

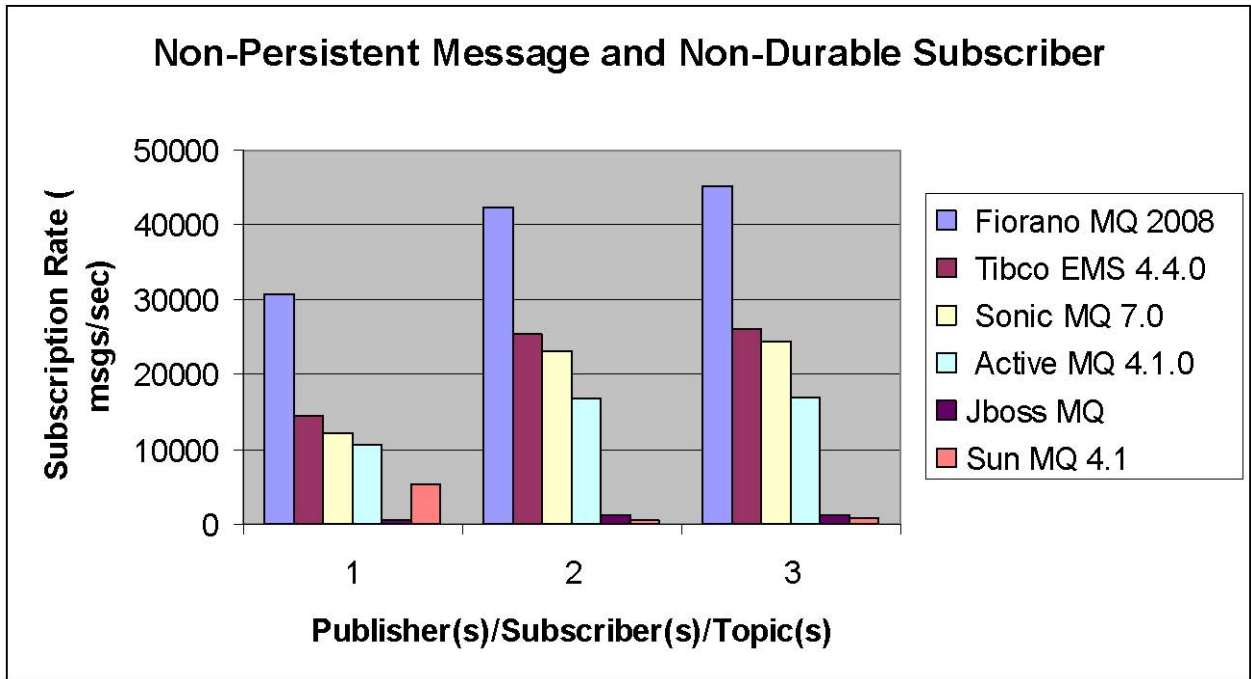
P/S/T	Message Type	Subscriber Type	Message	Subscription Rate (messages / sec)					
				FioranoMQ 2008	Tibco EMS 4.4.0	Sonic MQ 7.0	Active MQ 4.1.0	Jboss 1.4	Sun MQ 4.1
1/1/1	Persistent	Durable	1024	1353	985	690	596	431	1373
1/10/1	Persistent	Durable	1024	11596	8708	9470	4103	990	1778
1/25/1	Persistent	Durable	1024	20820	12215	11671	6695	1007	748
1/50/1	Persistent	Durable	1024	18133	10424	9121	3912	831	541





### 3.4 Non-Persistent Publisher, Non-Durable Subscribers

P/S/T	Message Type	Subscriber Type	Message	Subscription Rate (messages / sec)					
				FioranoMQ 2008	Tibco EMS 4.4.0	Sonic MQ 7.0	Active MQ 4.1.0	Jboss 1.4	Sun MQ 4.1
1/1/1	Non-Persistent	Non-Durable	1024	30655	14341	12246	10742	454	5340
1/10/1	Non-Persistent	Non-Durable	1024	42471	25329	23103	16717	1278	579
1/25/1	Non-Persistent	Non-Durable	1024	45101	26219	24348	17057	1196	643
1/50/1	Non-Persistent	Non-Durable	1024	42921	22128	19223	14231	933	493



## 4 System Configuration

### 4.1 Hardware Configuration

#### Server System

Windows 2000

Four CPU Intel Xeon - 2 GHz each

4 GB RAM

#### Client System

Windows 2000

Single CPU Pentium 4 - 3GHz

2 GB RAM

No of Client machines: 1

#### Network Settings

Client and Server were on the same network.

Network Speed: 1GBPS.

### 4.2 Software Configuration

Java (TM) 2 Runtime Environment, Standard Edition (build 1.5.0\_05-b05)

Fiorano MQ 2008

Sonic MQ v7.0

Tibco EMS v4.4.0 (In persistent tests, the TIBCO topics were set to failsafe to ensure persistence to disk.)

ActiveMQ 4.1.0

JBOSS 1.4

Sun JAVA MQ 4.1