JMS Performance Comparison

Performance Comparison for Publish Subscribe Messaging

Prepared by

Krissoft Solutions

October 2004

Disclaimer

Krissoft Solutions has made reasonable efforts to ensure the accuracy and validity of its testing. However, Krissoft specifically disclaims any warranty, expressed or implied, relating to the test results and analysis, their accuracy, completeness or quality, including any implied warranty of fitness for any particular purpose. All persons or entities relying on the results of any testing do so at their own risk, and agree that Krissoft shall have no liability whatsoever from any claim of loss or damage on account of any alleged error or defect in any testing procedure or result. In no event shall Krissoft be liable for indirect, special, incidental, exemplary or consequential damages in connection with its testing, even if advised of the possibility of such damages.

About Krissoft Solutions

Krissoft Solutions is a leading provider of technology solutions to global enterprises. Krissoft works with premium clients for providing efficient and cost-effective innovative extension solutions, emphasizing the maximum usage of existing infrastructure such as hardware and software, thereby minimizing the requirement of new investment. Krissoft Solutions helps its customers to analyze, understand and leverage new economy trends by incorporating an effective balance of people, technology and service. For more information visit www.krissoft.com.

Krissoft Solutions
327, White House,
Sector - 21, Nerul,
Navi Mumbai - 400 706, INDIA
Email: info@krissoft.com

Contents

Executive Summary

This paper presents a performance analysis of the publish/subscribe messaging throughput of FioranoMQ 7.5, SonicMQ 6.0, Tibco EMS 4.0 and IBM WebSphereMQ 5.3. The 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. All tests and their corresponding results were performed and analyzed by Krissoft, a leading technology consultancy and testing company (www.krissoft.com).

The testing tool used for these performance tests is highly configurable, and can be used to test any JMS broker. In addition, this tool allows for the running and measurement of a wide range of test definitions. It must be noted that 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 will be 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.

Test Methodology

All the tests described in this chapter were carried out using a highly configurable testing tool. This tool allows for the running and measurement of a wide range of test definitions. This chapter begins with a brief description of the test conditions, created to test the JMS servers. 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.

Test Conditions

All the tests were conducted under the following conditions:

- Each client was run on a separate JMS connection.
- All test results were recorded after client connections had 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 on both servers.
- During the test, no other applications were running and using resources on the system under test.
- Dups_ok was used by all consumers (For persistent message and durable subscription tests, the databases of both products were emptied and cleaned out before each test).
- All servers were tested with default parameters - which meant running SonicMQ, Tibco EMS and WebSphereMQ in "Evaluation" (non-HA) mode and FioranoMQ in normal production ready (non-HA) mode.
- Unlike some of our competitors who tend to misguide customers by publishing results using different JVMs and JVM arguments for showing their own product in better light, these results use EXACTLY the same JVM and JVM args for both the competing products.

Test Scenarios

The tests were conducted for the two 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 client or server failure.

For each of the above models, the tests are further sub-categorized to check the scalability of JMS server w.r.t following two parameters:

a. **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 depict the scalability of JMS server as more clients (each working on independent JMS Topics) are employed.

b. **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 depict the scalability of JMS server as more clients (all working on same JMS Topic) are employed.

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 the 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.

Test Duration

All test scenarios were executed for a total of thirty-three minutes. Each test execution comprised of thirty-three 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. Similarly, during ramp-down intervals, the systems are decreasing their capacity in response to decreased client loads that result from test completion. The remaining thirty intervals were considered “measurement” intervals during which steady-state performance was achieved. Steady-state is the condition in which message rates exhibit negligible change.

Environment Setup

All client connections, publishers and subscribers were established before any testing ramp-up periods were begun. Each product’s message store, log files, queues, and topics were deleted and recreated, and the broker stopped and restarted between each test.

Measurement

Performance data was collected during the thirty-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.

Topology

The topology consisted of 2 machines: 1 for running all the clients and the other for running the server. The system configurations are detailed later in this document. These systems were interconnected on an isolated network using a single network switch.

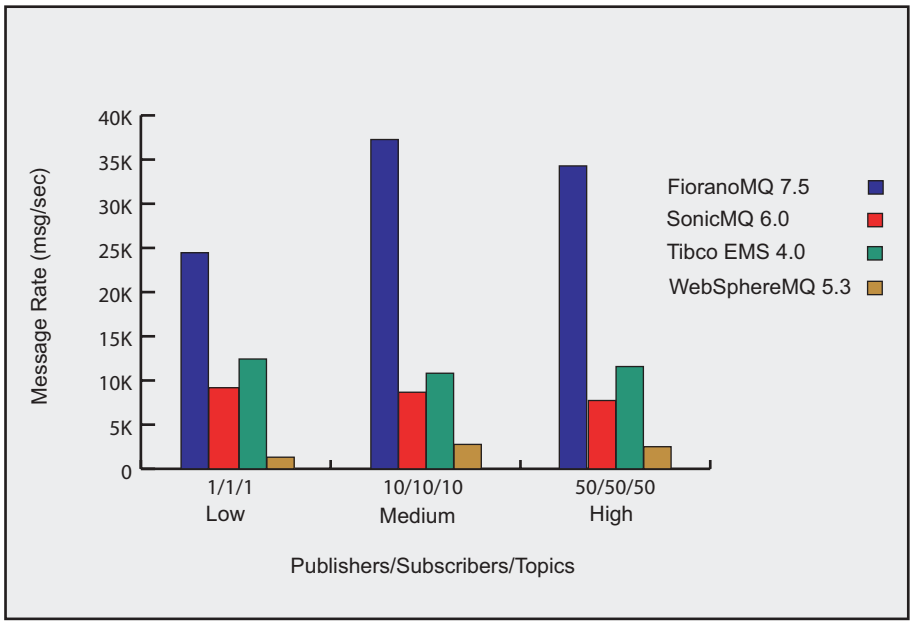
The following section contains a tabular data and charts detailing the receive message rates for each test.

Performance Results

Non-persistent Publisher Non-durable Subscriber

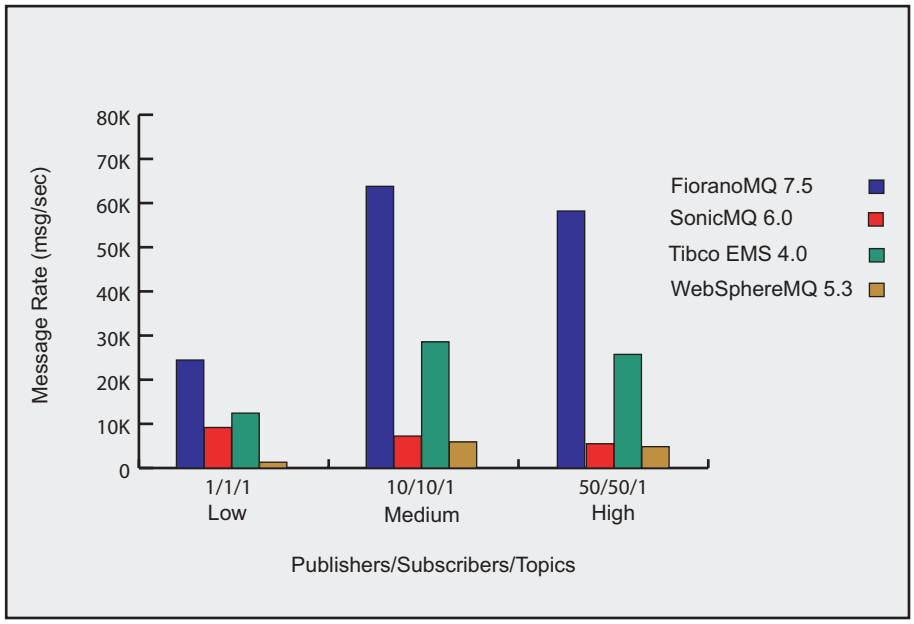
Server Scalability

P/S/T	Message Type	Subscriber Type	Message Size	FioranoMQ 7.5 Subscriber Rate (msg/sec)	SonicMQ 6.0 Subscriber Rate (msg/sec)	TibcoEMS 4.0 Subscriber Rate (msg/sec)	WebSphereMQ 5.3 Subscriber Rate (msg/sec)
1/1/1	Non-persistent	Non-durable	1024	24457.83	9179.50	12429.00	1317.15
10/10/10	Non-persistent	Non-durable	1024	37268.86	8669.26	10819.60	2764.36
50/50/50	Non-persistent	Non-durable	1024	34283.52	7733.20	11576.65	2503.17



Topic Scalability

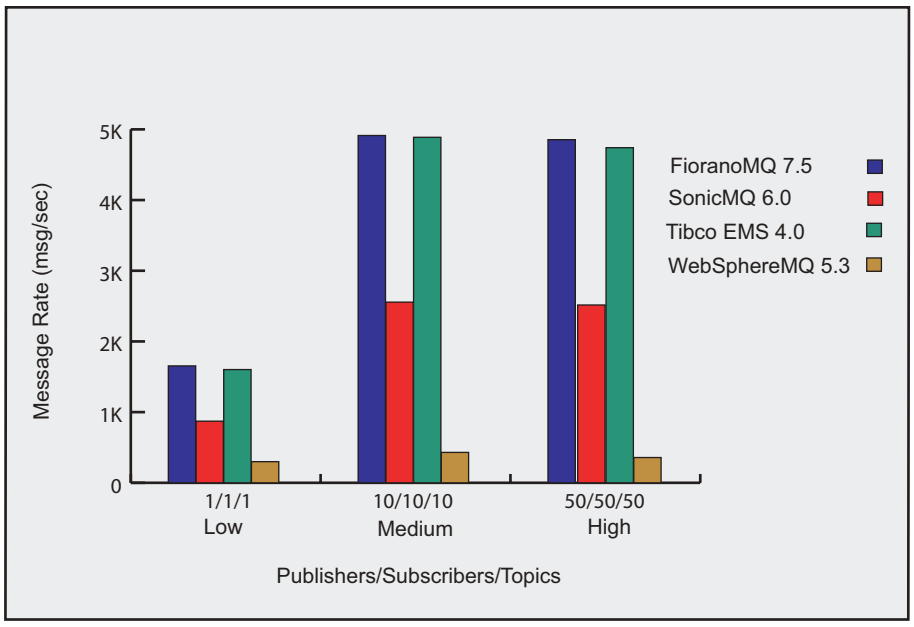
P/S/T	Message Type	Subscriber Type	Message Size	FioranoMQ 7.5 Subscriber Rate (msg/sec)	SonicMQ 6.0 Subscriber Rate (msg/sec)	Tibco EMS 4.0 Subscriber Rate (msg/sec)	WebSphereMQ 5.3 Subscriber Rate (msg/sec)
1/1/1	Non-persistent	Non-durable	1024	24457.83	9179.50	12429.00	1317.15
10/10/1	Non-persistent	Non-durable	1024	63791.00	7237.00	28570.38	5909.82
50/50/1	Non-persistent	Non-durable	1024	58201.25	5486.00	25732.3	4838.27



Persistent Publisher Durable Subscriber

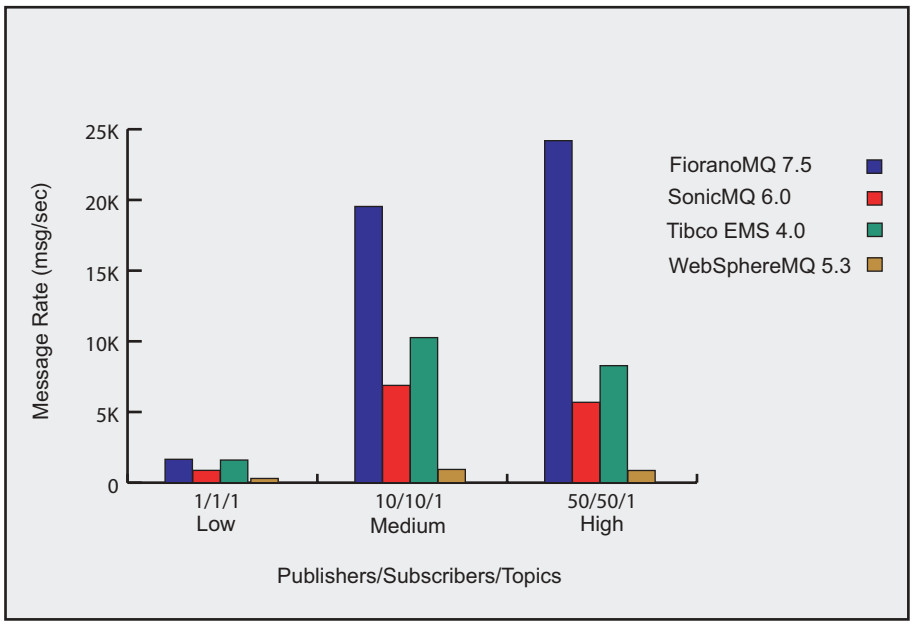
Server Scalability

P/S/T	Message Type	Subscriber Type	Message Size	FioranoMQ 7.5 Subscriber Rate (msg/sec)	SonicMQ 6.0 Subscriber Rate (msg/sec)	Tibco EMS 4.0 Subscriber Rate (msg/sec)	WebSphereMQ 5.3 Subscriber Rate (msg/sec)
1/1/1	Persistent	Durable	1024	1654.09	871.82	1601.96	299.23
10/10/10	Persistent	Durable	1024	4913.06	2555.89	4887.69	430.15
50/50/50	Persistent	Durable	1024	4854.3	2515.47	4740.54	357.52



Topic Scalability

P/S/T	Message Type	Subscriber Type	Message Size	FioranoMQ 7.5 Subscriber Rate (msg/sec)	SonicMQ 6.0 Subscriber Rate (msg/sec)	Tibco EMS 4.0 Subscriber Rate (msg/sec)	WebSphereMQ 5.3 Subscriber Rate (msg/sec)
1/1/1	Persistent	Durable	1024	1654.09	871.82	1601.96	299.23
10/10/1	Persistent	Durable	1024	19538.00	6882.00	10261.60	937.26
50/50/1	Persistent	Durable	1024	24192.00	5686.00	8278.30	863.25



Conclusion

In the various test scenarios it is observed that the message throughput of FioranoMQ is outstandingly more when compared to SonicMQ, Tibco EMS and IBM WebSphereMQ.

In the Server Scalabiliy tests for NP/ND messaging, FioranoMQ outperforms SonicMQ by 2 to 4 times, Tibco EMS by 2 to 3.5 times, and IBM WebSphereMQ by a massive 13.5 to 18.5 times. Likewise, in the Topic Scalability tests for NP/ND messaging, FioranoMQ outperforms SonicMQ by 2 to 10 times, Tibco EMS by upto 2 times, and IBM WebSphereMQ by 10 to 18 times.

In the Server Scalability tests for P/D messaging, FioranoMQ outperforms SonicMQ by upto 2 times, and IBM WebSphereMQ by 5 to 13 times. Likewise, in the Topic Scalability tests for P/D messaging, FioranoMQ outperforms SonicMQ by 2 to 4 times, Tibco EMS by 2 to 3 times, and IBM WebSphereMQ drastically by 5 to 28 times.

Overall analysis shows that FioranoMQ outperforms SonicMQ, Tibco EMS and IBM WebSphereMQ in all the scenarios.

System Configuration

Hardware Configuration

Client System

- Windows 2000
- Xeon Dual CPU 2.8GHz each
- 2 GB RAM
- No of client machines: 1

Server System

- Windows 2000
- Xeon Dual CPU 1.8GHz each
- 1 GB RAM
- No of server machines: 1

Network Settings

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

Software Configuration

Client System

- Java HotSpot ™ Client VM (Build 1.4.0_01_b03)
- Argument: Xmx256m - Xms256m

Server System

- Java HotSpot ™ Client VM (Build 1.4.0_01_b03)
- Argument: Xmx512m - Xms512m

JMS Brokers

- **FioranoMQ** Version 7.5 Build No 3567
- **SonicMQ** Version 6.0 Build No 457
- **Tibco EMS** Version 4.0
- **IBM WebSphereMQ** Version 5.3