Solace Systems®

Solace Web Messaging Benchmarks

This document reports the performance of the Solace Message Router when used for Web Messaging for fanout to Rich Internet Applications (RIAs). It explains the tests executed, the results obtained, the test environment, configuration and tools used in the tests.

Contents

1 Overview

Solace Web Messaging is much more than a Comet streaming server. Solace Web Messaging provides the ability to perform bidirectional, asynchronous communication between a Rich Internet Application (RIA) deployed in a Web Browser, desktop computer, or mobile device and server applications resident in a data center written in various languages such as Java, .NET, C++, or Python. Streaming data from server applications to browser applications over the internet is just one of many uses. For more information on Web Messaging please refer to http://solacesystems.com/webmessaging for general information and http://solacesystems.com/sdp for its application to Single Dealer Platforms.

This document reports the performance of the Solace Message Router when used for Web Messaging for fanout to Rich Internet Applications (RIAs). It explains the tests executed, the results obtained, the measurement methodology, the test environment and the tools used in the tests.

Performance tests executed on any product should have a particular purpose or use case in mind to allow the results to be interpreted as “good or bad” in the context of the problem that a user is trying to solve. Different use cases place different value on performance attributes such as rate, latency, message size, fanout, etc. In some cases optimizing one set can be so much more important from a business point of view than optimizing others.

In this test report, we focus the tests primarily on the use case of a real time financial internet distribution hub as typically employed in Single Dealer Platforms and Equity Brokerage systems to stream quote updates to web-based trading applications implemented using Rich Internet Application (RIA) technology. Future test reports will focus on the needs of other applications such as consumer internet applications.

Among the many requirements of a streaming solution for financial applications, the most important technical characteristics to optimize are:

1. **Latency and latency distribution**.
   If your traffic is going over the Internet with 20+ milliseconds of latency to a desktop or mobile, then why bother optimizing latency at the fanout point? Two main reasons:
   - **Latency costs money** – In applications like FX SDP’s, arbitrage is used by traders to the detriment of the higher latency FX platform. Other than strategic datacenter placement, the Internet latency cannot be controlled by the architects of an SDP. However, the latency internal to their platform can be optimized. Therefore a lower latency platform provides a competitive advantage.
   - **RIAs use is spreading** – RIAs are fast becoming the platform of choice for desktop applications inside the walls of banks and financial institutions because of their richness and ease of deployment. These RIAs have access to internal applications with much less latency than over the internet and therefore the (undesirable) latency at the distribution hub can become a much larger percentage of end-to-end value.

   Therefore, web distribution products should not only have low average latency, but also tight latency distribution.
2. **Number of clients supported at desired performance**.
   Another important criterion is the infrastructure cost of supporting a target number of concurrent users while still providing the desired message rate and latency. Scaling horizontally by adding more servers and licensed software is a viable but expensive approach, so while any reasonable solution must support horizontal scaling, the denser the client support in a given product at the desired performance, the better.

The tests in this report therefore focus on message fanout latency and client scalability as these are key optimization criteria for real time applications such as in financial services platforms.

2 Test Environment and Methodology

2.1 Test Configuration and Tools

The test tool configuration used for the tests in Section 3 is shown in Figure 2-1.

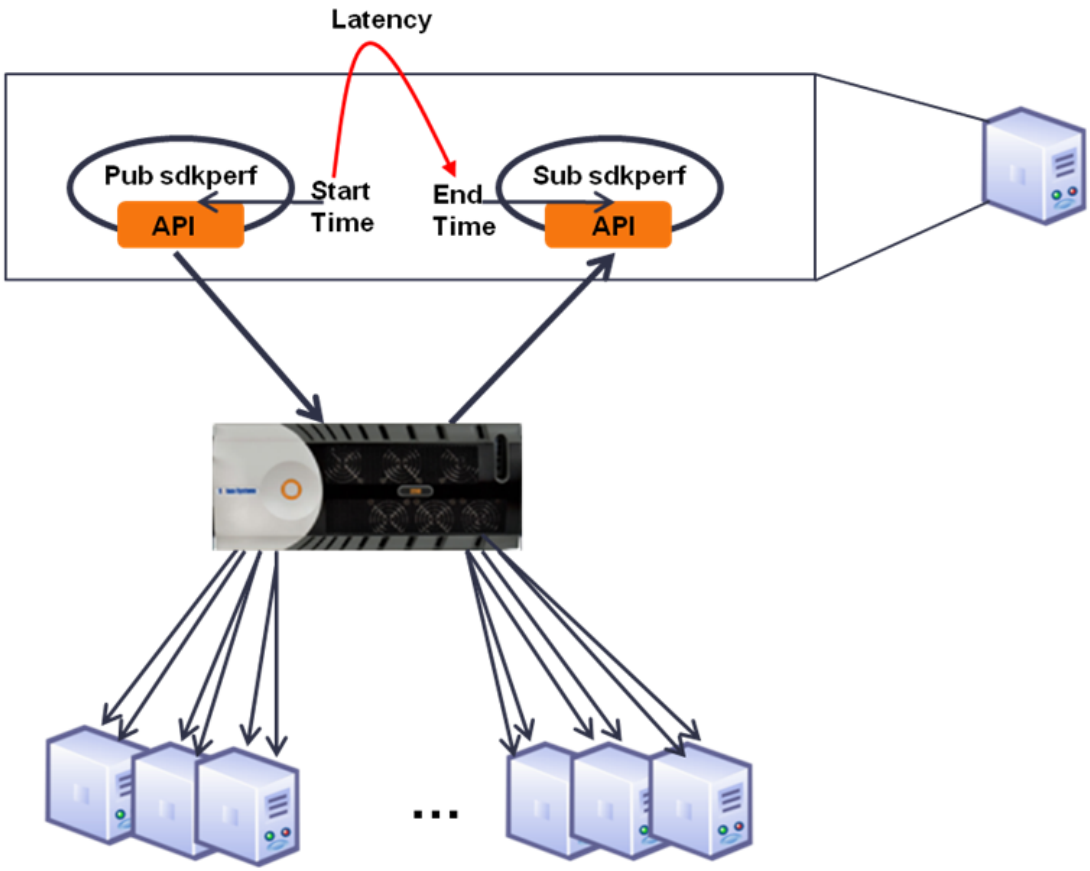

Figure 2-1 Test Configuration

sdkperf is a Solace-produced tool available to our customers that emulates a number of publisher and subscriber applications, creating a connection to the Solace Message Router for each emulated client application and either publishing or receiving messages based on topic subscriptions. It can generate messages of various sizes at configured rates and then measure publisher rate, subscriber receive rate and the latency of each message.

Due to the extremely low latency achieved with the Solace Message Router (microseconds compared to milliseconds with software products) and the difficulty in synchronizing clocks to microsecond accuracy on distinct servers, additional care must be taken to produce valid latency results:

- Latency measurements are performed using a configuration of one publisher and one subscriber sdkperf instance running on the same server, thereby only requiring clock synchronization among cores on the same CPU.
- To measure latency with microsecond accuracy, the sdkperf tool uses the RDTSC instruction (http://en.wikipedia.org/wiki/Time_Stamp_Counter) of the Nehalem processor to embed the value of the time stamp counter into each message it publishes. For processors running in the 2.6GHz range, this easily provides accuracy to one microsecond even across cores of a single CPU.

The value of the time stamp counter (Count-A) is extracted by the publisher sdkperf application and embedded into each message it generates before passing the message to the Solace messaging API. The TSC is then also read by the subscriber sdkperf for each message it receives from the Solace API (Count-B) and the latency for that particular message is obtained as (Count-B – Count-A / ClockFrequency). This latency is from the top of the publisher's API (prior to publish) to the top of the subscriber's API (after receipt) and thus includes all processing by the Solace messaging system.

The Solace Messaging Appliance randomizes the order in which a message is fanned out to a given set of subscribers to avoid the unfair situation where one subscriber is always being first or always last. Therefore, measuring the latency of a single subscriber is representative of the latency experienced by any subscriber.

2.2 Explanation of Result Data

The subscriber sdkperf gathers latency measurements for each message it receives rounding up to one microsecond granularity.

For each test, we provide the following data points computed over the entire stream received by the measurement subscriber sdkperf over test runs of several minutes:

- Average latency
- Median latency
- 99^{th} and 99.9^{th} percentile
- Standard deviation
- Maximum latency

2.3 Hardware Configuration

The Solace Message Router used in the test is a 3260 with a 1Gbps Network Acceleration Blade (NAB) in some tests and a 10Gbps Network Acceleration Blade (NAB) in other tests as indicated.

The servers used to host the publisher and subscriber load generation tools (sdkperf) and the networking connectivity hardware are listed in the following table.

- One server was used to host the sdkperf publishing the messages and the subscriber measuring latency. Two instances of the sdkperf tool were used on this host – one to publish, one to subscribe.
- Twelve servers were used to host the subscribers receiving the fanned out message stream. Each of these housed one instance of sdkperf emulating many clients.

Number of CPUs	2
Number of cores per CPU	4
CPU type	Intel Nehalem X5550 @2.67GHz
Cache	8MB
RAM	12GB
OS	CentOS 2.6.18-92.el5, 64bit
Ethernet Switch	Arista 7124

Figure 2-2 Server and Network Configuration

3 Test Cases and Results

3.1 Client Fanout

3.1.1 High Update Rates

The parameters for this test were chosen to demonstrate the performance that can be achieved for a reasonable size FX platform. The specific parameter values are listed in the following table.

Number of source topics	10,000
Publisher message rate	5,000 msgs/sec (each topic published every 0.5 sec)
Client receive rate	50 msgs/sec
Subscriptions per client	100
Message Size	50 bytes payload, 5 byte topic

The latency results for increasing numbers of client connections are shown in Figure 3-1 when using a single 1GE link from the 3260 Message Router. Note that the latencies are in microseconds, not milliseconds which is the range in which most competitive solutions are measured.

#clients	Total Egress Msg Rate	Avg (µsec)	Median (µsec)	99th (µsec)	99.9th (µsec)	Std Dev (µsec)	Max (µsec)
1,000	50,000	39	40	47	48	6	48
2,000	100,000	47	49	61	62	10	63
3,000	150,000	54	51	76	77	14	78
4,000	200,000	61	58	89	91	15	92
5,000	250,000	71	64	105	106	23	106
6,000	300,000	75	74	118	120	28	120
7,000	350,000	84	84	132	133	41	133
8,000	400,000	77	70	123	124	28	127
9,000	450,000	99	82	160	161	58	161

Figure 3-1 "High" Update Rates using 1GE

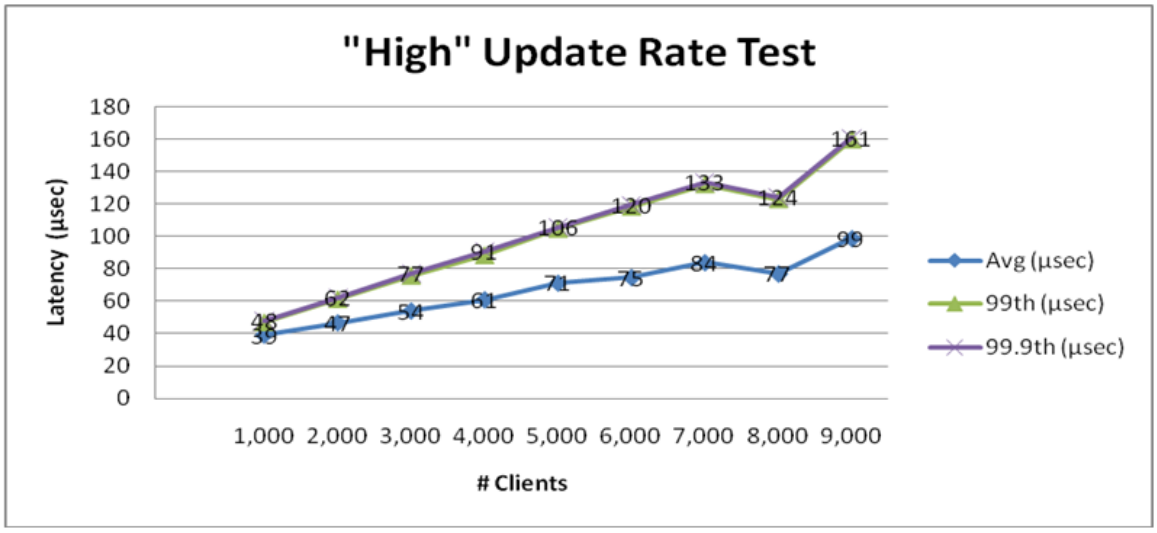

Figure 3-2 Graph "High" Update Rates using 1GE

3.1.2 Very High Update Rates

The parameters for this test were chosen to demonstrate the performance that can be achieved for a high end FX or Single Dealer Platform.

Number of source topics	20,000
Publisher message rate	20,000 msgs/sec (each topic published every 1 sec)
Client receive rate	100 msgs/sec
Subscriptions per client	100
Message Size	50 bytes payload, 5 byte topic

The latency results for increasing numbers of client connections are shown in Figure 3-3 when using a single 1GE link from the 3260 Message Router. Note that the latencies are in microseconds.

#clients	Total Egress Msg Rate	Avg (µsec)	Median (µsec)	99th (µsec)	99.9th (µsec)	Std Dev (µsec)	Max (µsec)
1,000	100,000	33	32	38	39	1	83
2,000	200,000	40	41	48	48	6	101
3,000	300,000	37	37	41	44	2	46
4,000	400,000	48	48	62	63	8	64
5,000	500,000	36	36	39	41	1	44
6,000	600,000	45	45	56	59	6	62
7,000	700,000	56	53	84	85	13	87
8,000	800,000	41	41	49	52	4	81
9,000	900,000	17,523	17,497	25,268	25,843	15,952	28,714

Figure 3-3 "Very High" Update Rates using 1GE

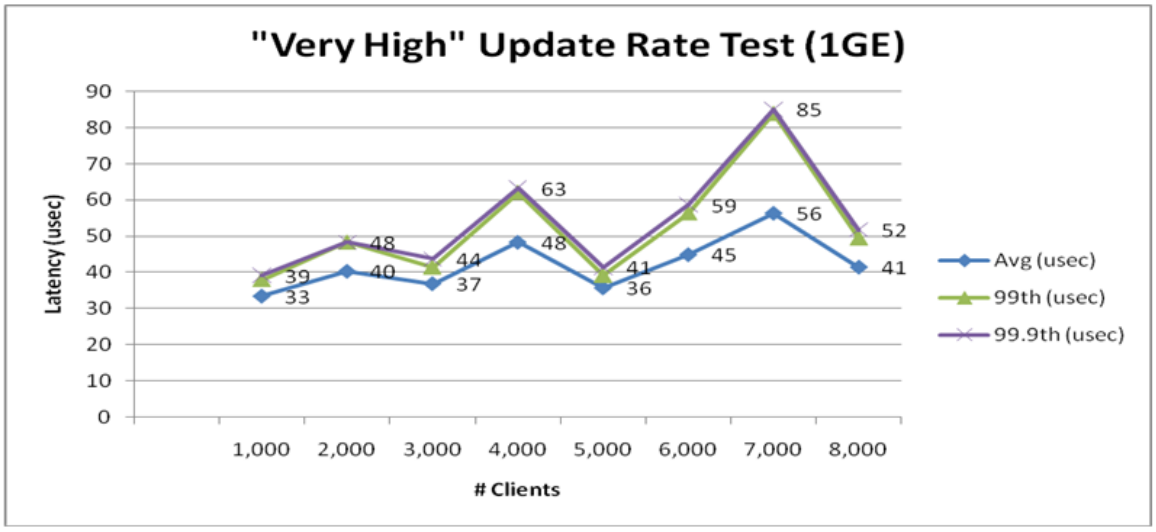

Figure 3-4 Graph "Very High" Update Rates using 1GE

Latency values are very good throughout the test except for at 9,000 connections where they increase considerably. Given that the total bandwidth required when assuming one message per TCP segment approaches the 700Mbps over the single GigE, we expected that the increased latency was due to instantaneous congestion of the 1GE link as the Solace appliance fanned out incoming messages since each inbound message needs to be fanned out to 45 clients. To test this theory, we reran the same test with the only change being to use a single 10GE link from the Solace Message Router. The latency results for this configuration are shown in Figure 3-5.

#clients	Total Egress Msg Rate	Avg (µsec)	Median (µsec)	99th (µsec)	99.9th (µsec)	Std Dev (µsec)	Max (µsec)
1,000	100,000	31	31	32	33	1	44
2,000	200,000	33	35	36	38	1	46
3,000	300,000	33	33	36	37	1	99
4,000	400,000	37	37	41	43	2	48
5,000	500,000	33	35	36	37	1	39
6,000	600,000	37	37	41	43	2	48
7,000	700,000	41	40	51	53	5	75
8,000	800,000	36	36	40	41	2	46
9,000	900,000	37	37	41	44	2	76

Figure 3-5 "Very High" Update Rates using 10GE

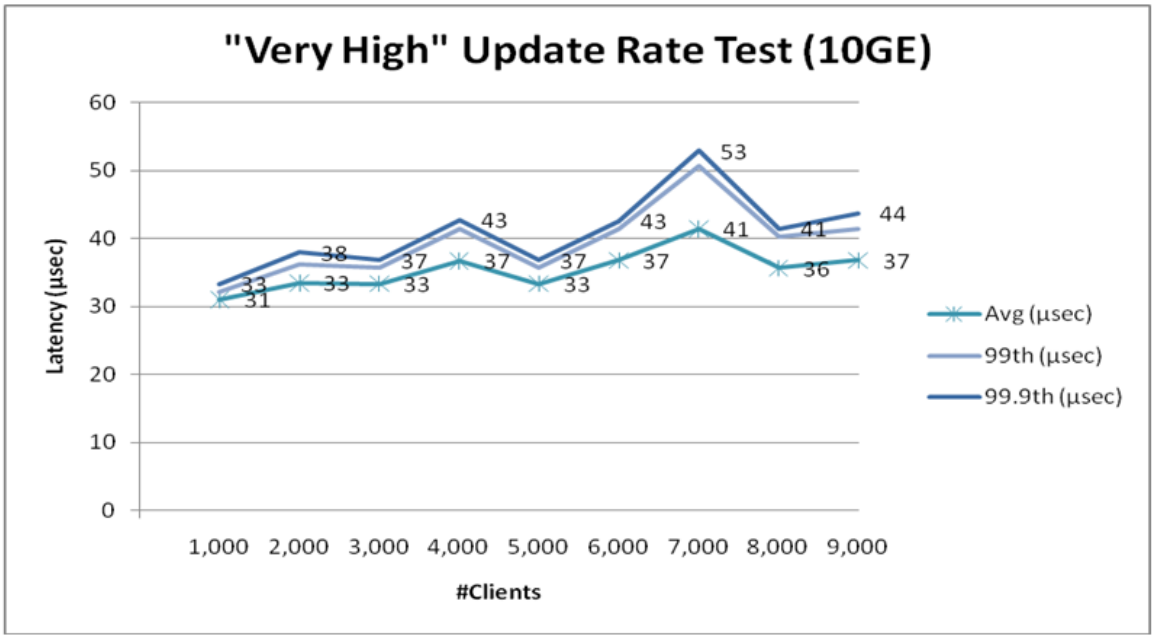

Figure 3-6 Graph "Very High" Update Rates using 10GE

As can be seen, the latency for a fanout of 9,000 clients remains predictably low on average and in the 99.9th percentile tail, confirming that 1GE network congestion was the cause of high latencies at the 9,000 fanout of the previous test.

3.2 Fanout with Input Filtering

In some applications, such as Equities trading platforms, the input message rate to the streaming server can be considerably higher than the 20,000 messages per second of the previous tests and much of this higher message rate needs to be filtered for the subscribing clients and discarded. This test starts with the configuration from Section 3.1.2 using 10GE as the baseline and increases the input message rate to the Solace Message Router to have it receive, lookup and discard messages at an increasingly higher rate while still accepting the original 20,000 messages per second and fanning those out to 9,000 clients at 100 msgs/sec/client to determine the impact on latency for the connected subscribers.

Number of source topics	20,000
Publisher message rate	20,000 msgs/sec (each topic published every 1 sec)
Client receive rate	100 msgs/sec
Subscriptions per client	100
Message Size	50 bytes payload, 5 byte topic

Discard Message Rate	Total Egress Msg Rate	Avg (µsec)	Median (µsec)	99th (µsec)	99.9th (µsec)	Std Dev (µsec)	Max (µsec)
0	900,000	36	37	43	46	4.7	58
500,000	900,000	40	40	53	60	7.9	75
1,000,000	900,000	41	48	55	65	9	89
1,500,000	900,000	41	41	53	56	6.6	65
2,000,000	900,000	40	40	55	63	7.5	74
2,500,000	900,000	42	42	57	75	7.1	175
3,000,000	900,000	43	43	60	65	7.6	104

Figure 3-7 Fanout with Input Filtering using 10GE

The results demonstrate that not only can the Solace Message Router filter/discard incoming messages at 3M msgs/sec, but the impact on latency to subscribing clients is less than 20% on average.

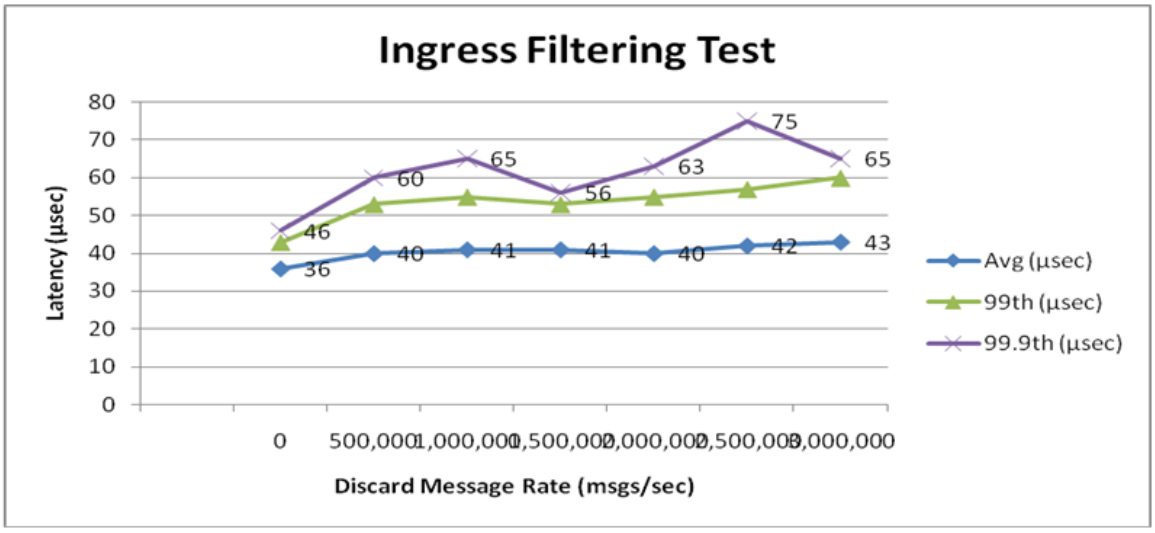

Figure 3-8 Fanout with Input Filtering using 10GE

3.3 Large Messages

The tests in this section test the ability of the Solace Message Router to perform client fanout of much larger messages and measure the impact on latency. All tests use the "Very High Update Rates" configuration, changing only the user payload size of the message.

3.3.1 1KB Messages

This test uses application payloads of 1KB with the "Very High Update Rates" test setup. The specific parameter values are listed in the following table.

Number of source topics	20,000
Publisher message rate	20,000 msgs/sec (each topic published every 1 sec)
Client receive rate	100 msgs/sec
Subscriptions per client	100
Message Size	1,024 bytes payload, 5 byte topic

The latency results for increasing numbers of client connections are shown in Figure 3-5 when using a single 10GE link from the 3260 Message Router. Note that the latencies are in microseconds.

#clients	Total Egress Msg Rate	Egress User BW (Mbps)	Avg (µsec)	Median (µsec)	99th (µsec)	99.9th (µsec)	Std Dev (µsec)	Max (µsec)
1,000	100,000	819	35	35	38	40	1.2	41
2,000	200,000	1,638	37	37	43	44	2.7	57
3,000	300,000	2,458	40	40	48	49	3.9	53
4,000	400,000	3,277	41	38	53	55	7.3	58
5,000	500,000	4,096	44	40	58	60	8.3	61
6,000	600,000	4,915	48	47	63	64	10.5	66
7,000	700,000	5,734	51	50	68	69	12.2	96
8,000	800,000	6,554	50	50	70	71	12.6	145
9,000	900,000	7,373	59	60	76	78	13.4	121

Figure 3-9 1KB Messages using 10GE

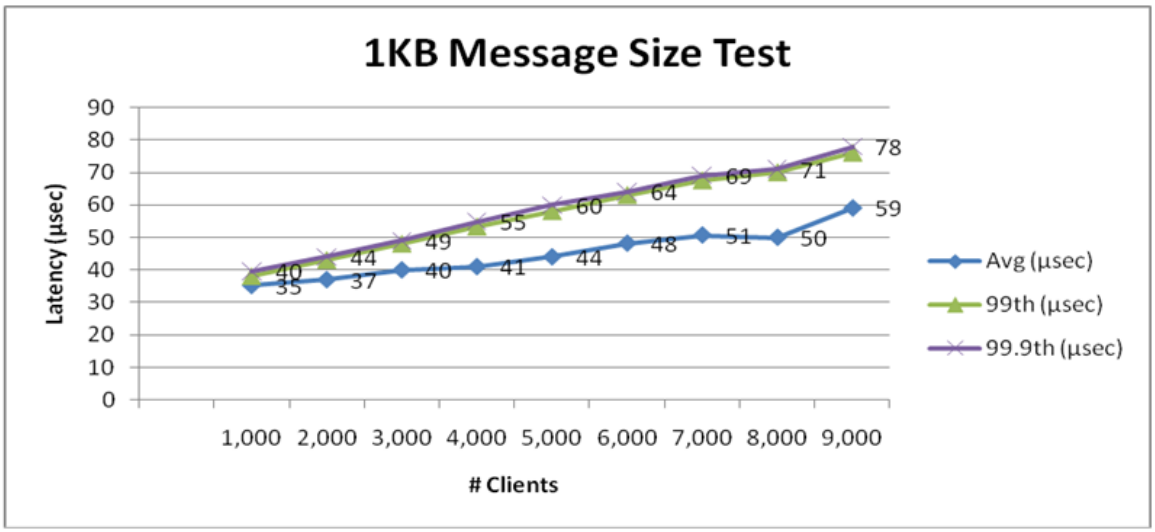

Figure 3-10 Graph 1KB Messages using 10GE

At 9,000 clients, the total bandwidth consumed is very close to the nominal 10GE rate.

3.3.2 2KB Messages

This test uses application payloads of 2KB with the "Very High Update Rates" test setup. The specific parameter values are listed in the following table.

Number of source topics	20,000
Publisher message rate	20,000 msgs/sec (each topic published every 1 sec)
Client receive rate	100 msgs/sec
Subscriptions per client	100
Message Size	2,048 bytes payload, 5 byte topic

The latency results for increasing numbers of client connections are shown in Figure 3-6 when using a single 10GE link from the 3260 Message Router. Note that the latencies are in microseconds (not milliseconds).

#clients	Total Egress Msg Rate	Egress User BW (Mbps)	Avg (µsec)	Median (µsec)	99th (µsec)	99.9th (µsec)	Std Dev (µsec)	Max (µsec)
1,000	100,000	1,638	45	45	52	54	2	56
2,000	200,000	3,277	52	54	59	61	5	65
3,000	300,000	4,915	60	63	69	73	9	75
4,000	400,000	6,554	66	67	77	79	8	85
5,000	500,000	8,192	71	72	89	91	11	179

Figure 3-11 2KB Messages using 10GE

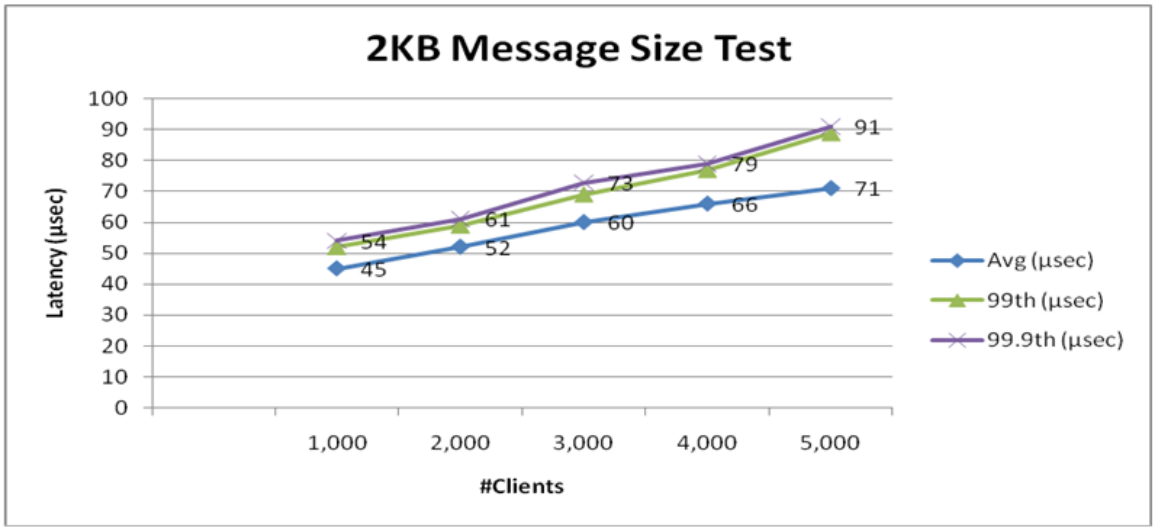

Figure 3-12 Graph 2KB Messages using 10GE

This test demonstrates the ability of the Solace Message Router to fanout messages to the full 10GE line rate for 2KB messages to 5,000 clients with each client consuming 100 msgs/sec.

4 Conclusion

As the test results in this paper have shown, Solace's unique new approach to web messaging offers revolutionary performance. For applications where achieving the lowest possible latency directly affects the user experience and drives competitive advantage, this major leap forward in terms of both speed and throughput has the potential to shift the competitive landscape in favor of those companies who adopt it.

It's important to note that Solace doesn't just offer the lowest latency and highest throughput, but the most stable and predictable performance as well. Solace's purpose-built hardware eliminates the bottlenecks and unpredictability of general purpose operating systems running on commodity servers that weren't built with high-speed I/O in mind. This means that for applications where reliably low latency is as important as raw speed, Solace is still the best solution.

In addition to offering game-changing performance, Solace's solution offers lower complexity and cost than other solutions thanks to the turnkey deployment and operation of Solace's purpose-built hardware. Solace enables dramatic consolidation in the datacenter because Solace handles internet data streaming as part of a powerful platform that handles *all* of the major kinds of message-oriented middleware, each with best of breed functionality and performance.