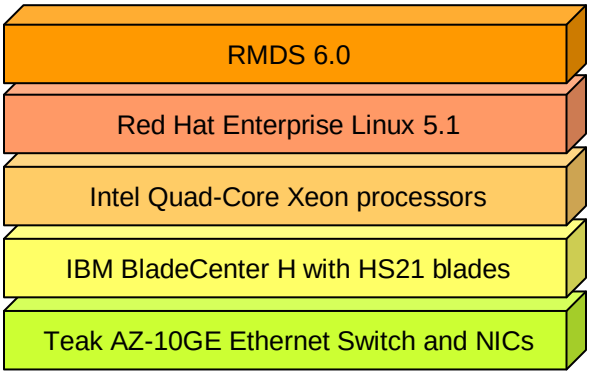
Teak Technologies AZ-10GE with RMDS on Intel®-Based IBM BladeCenter® Servers

Issue 1.0, 11 Mar 08

Technology Stack Under Test



Test methodology: Custom*

Key Results

- ➔ Peak throughput of 9.6 gigabits/second with AZ-10GE algorithms enabled, compared to 7.6 gigabits/second without AZ-10GE (in an RMDS configuration optimized for throughput).
- ➔ 1 ms or less of mean infrastructure latency at rates up to 450,000 updates per second (in an RMDS configuration optimized for latency and with AZ-10GE algorithms disabled).
- ➔ At every level of throughput, mean latencies of this 10GE-based RMDS stack are significantly better than those of any 1GigE-based RMDS stacks that STAC has tested.
- ➔ These results were obtained without the use of TOE, RDMA, iWarp or other client-side acceleration.

* *The test methodology for this STAC Report is not based on STAC Benchmark specifications, but it is designed to allow comparison to many previous STAC Reports. The relevant STAC Benchmark specifications are currently under development by the STAC Benchmark Council. For more information, see www.STACresearch.com/council.*

STAC SECURITIES TECHNOLOGY ANALYSIS CENTER

Disclaimer

The Securities Technology Analysis Center, LLC (STAC®) prepared this report at the request of Teak Technologies. It is provided for your internal use only and may not be redistributed, retransmitted, or published in any form without the prior written consent of STAC. All trademarks in this document belong to their respective owners.

The test results contained in this report are made available for informational purposes only. STAC does not guarantee similar performance results. All information contained herein is provided on an "AS-IS" BASIS WITHOUT WARRANTY OF ANY KIND. STAC has made commercially reasonable efforts to adhere to Reuters' published test procedures and otherwise ensure the accuracy of the contents of this document, but the document may contain errors. STAC explicitly disclaims any liability whatsoever for any errors or otherwise.

The evaluations described in this document were conducted under controlled laboratory conditions. Obtaining repeatable, measurable performance results requires a controlled environment with specific hardware, software, network, and configuration in an isolated system. Adjusting any single element may yield different results. Additionally, test results at the component level may not be indicative of system level performance, or vice versa. Each organization has unique requirements and therefore may find this information insufficient for its needs.

Customers interested in a custom analysis for their environment are encouraged to contact STAC.

Contents

Summary

Market data technologists are paying a great deal of attention to new networking technologies as a potential means of dealing with traffic that is ballooning and latency tolerances that are collapsing.

Teak Technologies offers AZ-10GE switches and NICs designed to work with the IBM Blade Center platform that promise to increase capacity and speed for a variety of workloads. One of the distinguishing features that Teak claims for its products is what it calls "Acceleration Zone Ethernet" a hardware-based collection of algorithms that detect, diagnose and rectify networking artifacts that hinder application performance. Teak asked STAC to run a few tests that would measure the performance of its products in a market data environment.

Together we established two goals for these tests:

1) Find the maximum throughput of the Teak network with and without AZ-10GE algorithms enabled in an RMDS stack that was optimized for throughput and configured to generate massive amounts of traffic.

2) Using the basic functionality of the Teak system (AZ-10GE algorithms disabled), measure the latency versus throughput for a Reuters Market Data System (RMDS) stack that was optimized for low latency.

To summarize, we found:

- Peak throughput of 9.6 gigabits/second using Teak AZ-10GE congestion control, compared to 7.6 gigabits/second without AZ-10GE (in an RMDS configuration optimized for throughput).
- 1 millisecond or less of mean infrastructure latency at message rates up to 450,000 updates per second (in an RMDS configuration optimized for latency).
- At every level of throughput, mean latencies of this 10GigE-based RMDS stack are significantly better than those of any 1GigE-based RMDS stacks that STAC has tested.
- These results were obtained without the use of TOE, RDMA, iWarp or other client-side acceleration.

1. Background

Real-time financial market data traffic is increasing rapidly around the world. Update-rate increases of 2 to 6 times in a single year are no longer uncommon. The largest single cause of this traffic growth is automated trading, which not only drives up transaction volumes but also increases the ratio of quotes and cancellations to actual trades. While North American venues still produce the most traffic, many observers expect the Markets in Financial Instruments Directive (MiFID) to trigger a sharp increase in European traffic as the number of trade-reporting venues proliferates. On top of this, large sell-side institutions often generate enormous amounts of real-time data internally, which they pump onto their internal market data system. The traffic from internal content sometimes exceeds that of information coming in from external sources.

An unfortunate consequence of higher volumes is a well-established tradeoff between throughput and latency. This matters because data latency has a huge impact on the overall speed with which a trading firm can execute a transaction in response to new information. In some markets, firms can profit from as little as one millisecond of advantage over competitors, which drives them to find sub-millisecond optimizations of the systems fueling their trades. The latency obsession has resulted from the spread of automated trading to nearly every geography and asset class, and the resulting imperative to exploit—or defend against—new latency arbitrage opportunities.

This combination of forces keeps market data technologists on the lookout for new technologies that can shift the performance tradeoffs in the right direction. One layer of the technology stack that receives ongoing scrutiny is the network (switches, interface cards, drivers, etc.). Market data systems are extremely network-I/O intensive. Most clients with heavy market data demands currently use 1-gigabit Ethernet (GigE). But requirements for some of the most intense automated trading systems are beginning to exceed a gigabit. This is also true of market data distribution beyond automated trading, particularly in cases where end users are located separately from the infrastructure. In those cases, the bandwidth required to stream data to end users is increasing quickly.

One of the new networking technologies receiving a great deal of attention is 10-gigabit Ethernet (10GigE). With 10 times the bandwidth of GigE, 10GigE offers considerable headroom for growth. However, Teak claims that ordinary 10 GigE solutions can suffer from congestion and other networking artifacts in real-world deployments, particularly those characterized by bursty traffic. This can significantly decrease available bandwidth and end-to-end latency, thereby making application performance erratic and unpredictable.

Teak Technologies has developed an AZ-10GE switching solution for the IBM BladeCenter H. The company claims that this technology, which they call "Acceleration Zone Ethernet", overcomes networking related artifacts and accelerates application performance, even when approaching the theoretical 10Gb/sec limit in bursty environments. According to Teak, the Teak switch employs technology to detect, diagnose, and rectify networking artifacts at very high speeds, and communicates with special NICs and converged adapters on compute and storage end-points. Upon detection, the switch and NICs employ high-speed digital control mechanisms to self-regulate rates at which traffic is exchanged in the network. The company says that its technology is self-adapting in that it requires no changes in configuration and adapts well to changes in network topology, add/moves, and traffic mix/volume. In theory, this should cut

down on re-transmissions, and lead to improved performance. We put this theory to the test by simulating the distribution of data from a market data system to hundreds of users. We also examined the latency of the Teak networking solution in a market data system configuration designed for low-latency distribution.

In both cases, we ran the Reuters Market Data System (RMDS), using Reuters' standard benchmarking procedures. We chose these procedures in order to enable easy comparison to other tests we have run with RMDS and because the emerging industry-standard STAC Benchmarks for market data middleware (STAC-M2) are still in development.

2. Description of Tests

2.1 Methodology

2.1.1 Test Setup

Throughput-optimized configuration

In order to generate the high message load for the throughput test, we used RMDS in a "stacked" topology, running multiple instances of certain processes on a single server. Figure 2-1 shows a conceptual view of the test harness. We ran it on 14 IBM HS-21 blades, each with two Quad-Core Intel® Xeon® 5355 2.66 GHz "Clovertown" processors. We used two blades to generate the message traffic, four to run the 16 source distributors, four to run the 16 Point-to-Point servers (P2PS) and four to run the message consumers. With this much horsepower, we were able to push the update rates to the highest levels the network could handle.

We installed the NICs on each blade, and installed the switch in the BCH chassis. The Teak AZ-10GE NICs were partitioned by creating virtual interfaces in order to support the stacked configuration. Virtual interfaces eth2:0, eth2:1 and eth2:2 were each created on a different subnet so the multicast traffic on each network could be isolated.

All traffic between the RMDS components (src_dist/p2ps) and the publishers and consumers was routed through a single switch port by partitioning the switch into two bridges and patching the two bridges through external switch ports using XFP modules and a fiber cable. This was done in order to simulate the separation between servers and clients in a real world scenario, where the server farms and end users are usually in different network segments bridged via the network core. It was also done to push all the traffic through a single switch port.

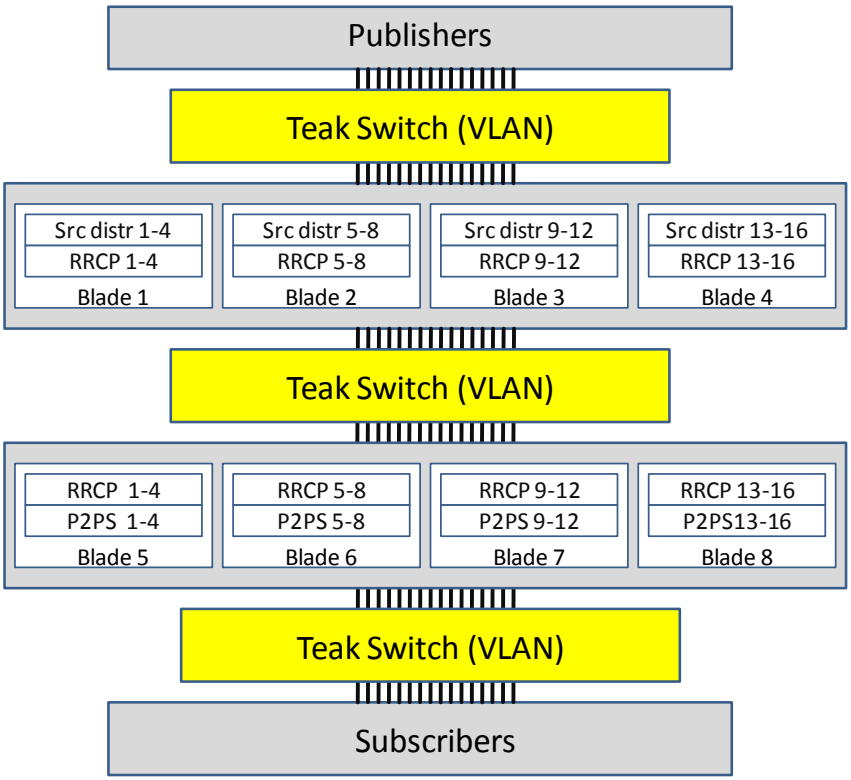


Figure 2-1: Throughput Test Setup

Latency-optimized configuration
For the latency-minimized configuration, we reconfigured RMDS to use a traditional topology (a single instance of each process per server) and ran this test on a different set of blades. Since this topology didn't call for as many cores as the throughput test, we chose hardware that would make the results easier to compare to prior STAC reports. We used IBM HS21 blades configured with two Dual-Core Intel Xeon 5160 "Woodcrest" processors running at 3.00 GHz. On the blades running the P2PS and source distributor, we configured the NICs with a virtual interface in order to separate the RRCP multicast traffic from the TCP traffic.

2.1.2 Procedures

The tests followed the procedures set forth by Reuters for hardware vendors, and used the test data supplied by Reuters.

Throughput-optimized configuration
For the throughput tests, we used the sink_driven_src utility to generate update traffic, and the rmdstestclient utility to consume the updates. Level 1 data was used, with a Reuters Wire Format (RWF) update size of 74 bytes (payload, not including header). Tests with fan-out of updates used a 200-item watchlist. RMDS was tuned for maximum throughput, and the update rate was increased until errors

were reported by the P2PS or RMDS client application. As noted above, multiple Source Distributors and multiple P2PSs were used to create the load necessary to measure the component under test.

The Linux sar utility was used to determine throughput rates. We ran 'sar -n DEV' on each of the 4 Point-to-Point Servers, and summed the outputs to get the total bytes per second (which we have presented as bits per second in the results). We noted that the Teak switch also reports throughput rate, and by eyeball inspection, we found it to be well correlated with the sar output. However, we didn't attempt to validate its accuracy.

Latency-optimized configuration
For the latency test, we again used sink_driven_src as the publisher and rmdstestclient as the subscriber, again with 74-byte RWF payloads. This time, we tuned the RMDS for low latency, and turned off the AZ-10GE feature of the Teak switch. We took measurements for five minutes at each message rate.

The Reuters procedure we followed uses an embedded timestamp approach to calculate end-to-end latency for Level 1 (Quotes and Trades) data. In this approach, the publisher embeds timestamps into selected updates, which the subscriber uses for latency calculations. We ran the publisher and subscriber on the same node for accurate timestamps. NTP was disabled on the tools node. Decode of data was turned on in these tests.

2.1.3 Time synchronization

All timestamps for the latency-optimized test were recorded on the same server blades.

2.1.4 Limitations

The Reuters test methodology uses sampling to determine latency statistics for an interval, which reduces the accuracy of distribution-related statistics such as max and standard deviation.

2.2 System Specifications

2.2.1 Servers

The IBM BladeCenter used in these tests was:

Vendor Model	IBM BladeCenter H Chassis
Blade Bays	14 (14 blades in chassis, all utilized in this test environment)
Rack Units	9U
Power Supplies	2

For the throughput test, each of the servers in the test harness had the following specifications:

Vendor Model	IBM eServer BladeCenter HS21
Processors	2
Processor type	Quad-Core Intel Xeon 5355 @ 2.66 GHz
Cache	8MB Integrated L2 Cache split between 4 cores
Bus speed	1.333 MHz

Memory	8 GB (4x2048 MB) DDR DIMMS
Disk	146 GB SAS
AZ-10GE cards	Teak NICs, Teak Driver Version 071227
NIC note	The MTU used for this test was 4000.
BIOS	BCE125-RK

For the latency test, each of the servers in the test harness had the following specifications:

Vendor Model	IBM eServer BladeCenter HS21
Processors	2
Processor type	Dual-Core Intel Xeon 5160 @ 3.00 GHz
Cache	4MB Integrated L2 Cache split between 2 cores
Bus speed	1.333 MHz
Memory	4 GB (2x2048 MB) DDR DIMMS
Disk	73 GB SAS
AZ-10GE cards	Teak NICs, Teak Driver Version 071227
NIC note	The MTU used for this test was 4000.
BIOS	BCE 1.08

2.2.2 Networking

Switch	Teak L2 AZ-10GE Switch Module for IBM BladeCenter H, I3000 BCH-S-20P4-10
Switch note	Two Bridges were setup for this test, one VLAN per Bridge

2.2.3 Network Interface Configuration

Any settings changed from the defaults are noted below

The following values were set on each Ethernet interface (eth0, eth1, … <ethX>) used for RMDS traffic:	Command
Txqueuelen	ifconfig <ethX> txqueuelen 5000

2.2.4 Operating System

Version	Red Hat Enterprise Linux 5.1 32-bit Kernel 2.6.18-36.el5
OS services	The following services were stopped: acpid anacron canna apmd arptables_jf atd autofs cpuspeed cron crond cups cups-config-daemon gpm haldaemon hpoj ip6tables iptables irqbalance isdn messagebus netfs nfs nfslock nscd pcmcia portmap postfix rhnsd rpcgssd rpcidmapd sendmail xfs xinetd iiim vsftpd snmpd

2.2.5 TCP and UDP Buffers

Any settings changed from the defaults are noted below:

	Values were those specified by the Reuters guidelines. The following lines were entered into the System File (/etc/sysctl.conf):	System File
Typical RMDS Setup	net.core.wmem_max = 8388608	*/etc/sysctl.conf*
	net.core.wmem_default = 8388608	
	net.core.rmem_max = 8388608	
	net.core.rmem_default = 8388608	
	net.ipv4.tcp_rmem = 4096 8388608 16777216	
	net.ipv4.tcp_wmem = 4096 8388608 16777216	
	net.ipv4.tcp_mem = 4096 8388608 16777216	
	net.ipv4.ip_local_port_range = 34800 65535	

2.2.6 Cache Enabled/Disabled in Source Distributor

Cache	**Change the lines in *rmds.cnf* to:**
Disabled	*<*serviceName*>*cacheLocation : srcApp

2.2.7 Cache Enabled/Disabled in P2PS

Cache	**Change the lines in *rmds.cnf***
Disabled	*p2ps*enableCache : False

2.2.8 RRCP Port Conflict Avoidance

OS	**Enter the following lines in system file noted**	**System File**
Linux	net.ipv4.ip_local_port_range = 34800 65535	***/etc/sysctl.conf***

2.2.9 Application Software

RMDS Binaries	src_dist ver. mdh6.0.2.L2 p2ps ver. p2ps6.0.2.L2 rrcp as included in p2ps6.0.2.L2
RMDS Test Tools	sink_driven_src (from infra_tools.0.0.2.L3) rmdstestclient (from infra_tools.0.0.2.L3)

RMDS Configuration	Change the lines in *rmds.cnf* to:
Common to all tests	*p2ps*sslMsgPacking : True
	*p2ps*rsslMsgPacking : True
	*p2ps*tcpSendBufSize : 64240
	*p2ps*hashTableSize = 200000

	*usePointToPointData = False
	*RRCP*maxPktPoolSize : 80000
	*RRCP*pktPoolLimitHigh : 70000
	*RRCP*pktPoolLimitLow : 60000
	*RRCP*userQLimit : 32768
	*RRCP*udpRecvBufSize : 4096
	*RRCP*udpSendBufSize : 4096
Throughput-optimized configuration	*p2ps*timedWrites : True
	*p2ps*flushInterval : 20
	*p2ps*tcpNoDelay : False
	*<*serviceName*>*rrmpFlushInterval : 20
	*p2ps*guaranteedOutputBuffers : 800
	*p2ps*maxOutputBuffers : 5000
	*p2ps*poolSize : 32000
Latency-optimized configuration	*p2ps*timedWrites : False
	*p2ps*flushInterval : 0
	*p2ps*tcpNoDelay : True
	*<*serviceName*>*rrmpFlushInterval : 0
	*p2ps*guaranteedOutputBuffers : 200
	*p2ps*maxOutputBuffers : 400
	*p2ps*poolSize : 16000
	*src_dist*route*numIpcInputBuffers : 10
	*src_dist*route*numIpcOutputBuffers : 100
	*src_dist*server*ipc*transmissionBus*guaranteedOutputBuffers : 200
	*src_dist*server*ipc*transmissionBus*numInputBuffers : 3
	*src_dist*server*ipc*transmissionBus*poolSize : 1600

3. Results

3.1 Throughput-optimized configuration

We ran the throughput tests with and without Teak's AZ-10GE technology and algorithms, and found a significant difference. With AZ-10GE disabled, we were unable to utilize about a quarter of the theoretical bandwidth of 10Gbit/second. But with AZ-10GE enabled, we nearly reached the theoretical limit.

Mode	Throughput rate (Gbits/sec)
Without AZ-10GE enabled	7.7
With AZ-10GE enabled	9.6

Table 1

3.2 Latency-optimized configuration

"End to end" RMDS latency is defined as the delta between the time an update is posted by the publisher application to its API and the time the same update is received by the consuming application from its API, i.e. it includes the latency contribution from both the API and the core infrastructure components. Table 2 records the latency statistics for the Teak-based RMDS configuration.

Update Rate [74-byte RWF messages/sec]	Mean Latency (milliseconds)	Std Deviation (millseconds)	Maximum Latency (milliseconds)	Minimum Latency (milliseconds)	Number of Latency Points
1,000	0.146	0.005	0.181	0.136	3000
5,000	0.205	0.007	0.232	0.188	3000
10,000	0.278	0.007	0.327	0.260	3000
20,000	0.304	0.024	0.453	0.264	3000
30,000	0.315	0.040	0.757	0.258	3000
40,000	0.357	0.058	0.864	0.258	3000
50,000	0.380	0.073	1.068	0.260	3000
60,000	0.401	0.086	1.129	0.260	3000
70,000	0.427	0.097	1.345	0.261	3000
80,000	0.446	0.110	1.501	0.261	3000
90,000	0.459	0.123	1.668	0.259	3000
100,000	0.473	0.130	1.860	0.261	3000
150,000	0.532	0.127	1.363	0.263	3000
200,000	0.597	0.175	1.747	0.257	3000
250,000	0.654	0.207	2.111	0.261	3000
300,000	0.735	0.257	2.770	0.263	3000
350,000	0.815	0.323	3.309	0.263	3000
400,000	0.844	0.343	3.760	0.262	3000
450,000	0.947	0.465	4.223	0.263	3000
500,000	1.066	0.576	5.007	0.267	3000
550,000	1.167	0.757	5.634	0.273	3000
600,000	1.406	0.990	6.140	0.270	3000
650,000	1.783	1.309	7.112	0.294	3000

Table 2

The mean latencies are presented in Figure 3-1 below, while the standard deviations are presented in Figure 3-2. At all update rates, the mean latency of the RMDS stack using Teak AZ-10GE and RHEL 5.1 was significantly better than that of the 1GigE-based RMDS stacks we have tested using earlier versions of RHEL on the same processors. Standard deviation (jitter) was comparable to the standard deviations observed on GigE-based systems using RHEL 4 and SLES 9.

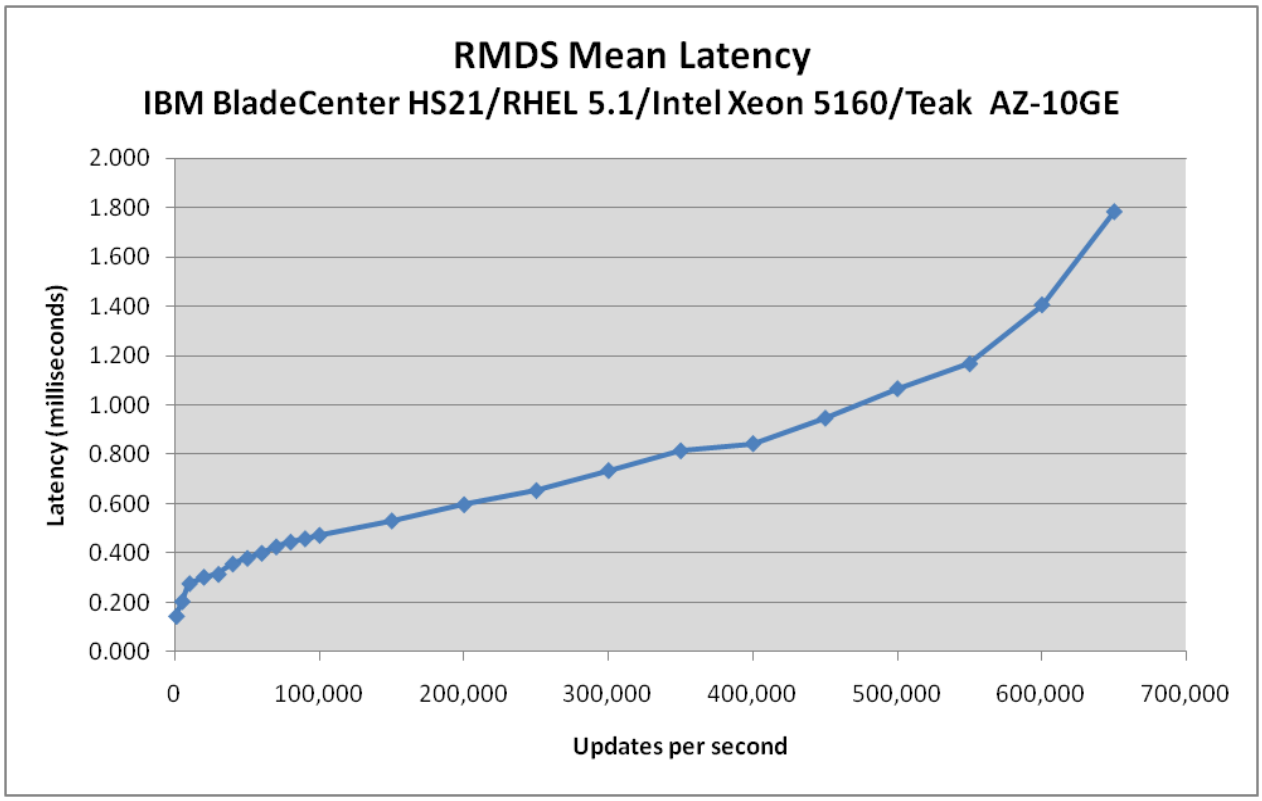


Figure 3-1: Mean Latency

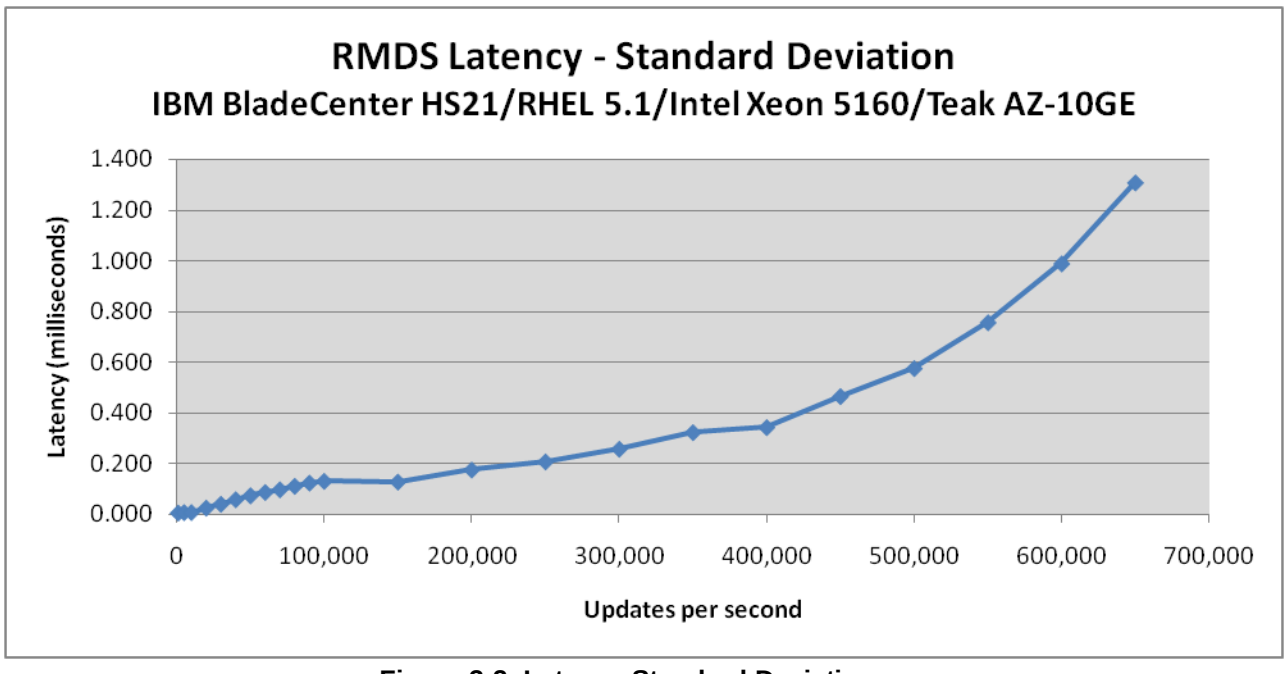


Figure 3-2: Latency Standard Deviation

About STAC

The Securities Technology Analysis Center, or STAC, conducts private and public hands-on research into the latest technology stacks for capital markets firms and their vendors. STAC provides optimization expertise, advanced tools, and simulated trading environments in STAC Labs. Public STAC Reports, available for free at www.STACresearch.com, document the capability of specific software and hardware to handle key trading workloads such as real-time market data, analytics, and order execution.

STAC also facilitates the STAC Benchmark Council, an organization of leading trading firms and vendors that specify standard ways to measure the performance of trading solutions (see **www.STACresearch.com/council**).

To be notified when new STAC Reports like this one are issued, or to learn more about STAC, see our web site at **www.STACresearch.com**.