Reuters Market Data System

RMDS 6.3

Performance Test Results on IBM BladeCenter HS22

DISCLAIMER: The test results and recommendations contained in this report are made available for informational purposes only. By issuing this report, IBM does not guarantee similar performance results. All information contained herein is provided on an “AS-IS” BASIS WITHOUT WARRANTY OF ANY KIND. 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. IBM is pleased to provide this report for our customers. We appreciate that each organization has unique requirements, and therefore may find this information insufficient for its needs. Customers wishing to obtain custom analysis for their systems are encouraged to contact their local IBM representative.

Issue 1.0
Date of Issue: 06 22 2009

Contents

1 General Information

1.1 Objective

The objective of this document is to report the performance test results for RMDS 6.0, for a particular hardware and software platform. The test procedures are described in *Reuters RMDS 6.0 Performance Test Procedures and Results* document.

The goal of these tests is to measure throughput and latency through RMDS 6.0 infrastructure components, specifically the Point-to-Pont Server (P2PS) and Source Distributor. The tests are grouped into two categories:

- Update throughput using RSSL/RWF data (see 3.1)
- End-to-end RSSL/RWF latency using embedded timestamp (see 3.2)

1.1.1 Results Summary

- **End-to-end latency test:** 1.30 million updates per second with a mean latency on less than 1 millisecond
- **Source Distributor throughput:** 1.70 million updates per second
- **P2PS (no fan-out) throughput:** 1.70 million updates per second
- **P2PS (producer 50/50) throughput:** 4.02 million updates per second

1.2 Testing Methodology

For throughput testing, the ***sink_driven_src*** utility was used to generate update traffic, and the ***rmdstestclient*** utility was used to consume the updates. Level 1 data was used, with a Marketfeed (MF) update size of 140 bytes, and an equivalent Reuters Wire Format (RWF) update size of 74 bytes. Tests with no fan-out of updates used a 100,000 item watchlist. The infrastructure is tuned for maximum throughput, and the update rate was increased until the CPU limit was reached with no errors reported. Where needed, and as noted, multiple Source Distributors or multiple P2PSs were used to create the load necessary to measure the component under test.

The embedded timestamp approach was used to calculate end-to-end latency for Level 1 (Quotes and Trades) data. RMDS 6.0 end-to-end update latency is measured by using ***sink_driven_src*** as the publisher and ***rmdstestclient*** as the subscriber.

In the embedded timestamp approach, the publisher embeds timestamps into selected updates which the subscriber uses for latency calculations. In this scenario, the publisher and subscriber must be running on the same node for accurate timestamps.

1.3 Software Versions

1.3.1 RMDS

src_dist ver. mdh6.3.2
p2ps ver. p2ps6.3.4
rrcp as included in p2ps6.3.4

1.3.2 RMDS Test Tool

sink_driven_src (from MDH load above)
rmdstestclient (from P2PS load above)

1.3.3 Operating Systems

- Red Hat (RHEL 5.3 64 bit), Linux kernel 2.6.18-128.el5
- Intel NIC driver version 2.0.24.4.

1.4 Hardware

The performance tests were performed on a single IBM BladeCenter-H with the following components:

1.4.1 Compute nodes

5 IBM Blade Server HS22 (7870) blades.
Each has:
2 QC Intel Xeon X5570 processors (2.93 GHz);
6 4GB 1333 MHz DDR III SDRAM;
143GB HDD;
2 integrated Broadcom 1GbE controllers;
1 dual port Intel 10GbE XF SR Server Adapter;

1.4.2 Blade chassis

1 BC-H (8852) which contains
1 Advanced Management Module;
2 Power Modules;
1 Cisco 3012 1GbE Switch Module (I/O Bay 1);
2 Blade Network Technology (BNT) 6-port 10GbE Switch Modules (I/O Bays 7 & 9);
5 HS22 (as mentioned above) in Blade Bays 1 to 5.

1.4.3 Network

Each blade is on 3 networks:
1 1Gigabit Ethernet (GbE) network solely for management purposes and
2 10 Gigabit Ethernet (10GbE) networks for low latency and high throughput RMDS communications.

Port 1 of the built-in dual-port1GbE is connected to Cisco 1GbE switch module.
Port 1 of the dual-port Intel 10GbE card is connected to BNT 10GbE switch module.
Port 2 of the dual-port Intel 10GbE card is connected to another BNT 10GbE switch module.

2 Preparation for Performance Test

2.1 Network

All the performance tests were run where the machines were connected to a private network via 10 Gbps switches. All the network cards and switch ports were set to Auto Negotiate.

2.2 Hardware

All RMDS components were run on the same class of machine.

2.3 Operating System Configuration

Earlier tests have shown that the value chosen for ticks per second (tps) on the test application machine has a significant impact on latency measurement. Accordingly, a tps value of 1000 was used in these tests.

2.3.1 TCP and UDP Buffers

Any settings changed from the defaults are noted below:

Step	Procedure		
1	**OS**	**Enter the following lines in system file noted**	**System File**
	Linux	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	

2.4 RMDS Configuration

The configuration template r***mds.cnf.template*** was customized for the tests.

Config File	Description	Path
rmds.cnf.template	Configuration file	***./config***

2.5 Miscellaneous Notes

Any other significant deviations from the standard test procedures, or clarifications, are noted below (such as number/type of machines used, CPU binding policy, etc.):

Test	Deviation	Comments
All	CPU Binding	Linux *irqbalance* was disabled and all interrupts were handled by CPU 0. Linux *taskset* command was used to bind RMDS processes. The *rrcpd* daemons were bound to CPUs 1 to 4, the *src_dist* processes were bound to CPUs 5 to 7, and so were the p2ps processes (running on a separate blade).The s*ink_driven_src* process was bound to CPU 1. *rmdstestclient* was bound to CPU 2.

3 Detailed Results

3.1 RSSL/RWF Update Throughput

- All the throughput numbers quoted here are for Level 1 data.
- The data file used in these tests has 1 update, with an update (data, not including header) size of 74 bytes in RWF.
- All of the tests with no fan-out used 100,000 item watchlist.
- In most of the throughput tests the individual processes were bound to particular CPU(s).
- ***sink_driven_src*** and ***rmdstestclient*** were used as the publisher and consumer of data.
- In some Source Distributor tests, two P2PSs were used to create sufficient load.

3.1.1 Standalone Source Distributor

Configuration Option	Transport	Max Throughput	Comments
Cache Disabled	RRCP	1.70 million updates per second	One P2PS and one src_Dist used

3.1.2 P2PS/LAN

Configuration Option	Mounts : Commonality	Transport	Max Throughput	Comments
Cache Disabled	No fan-out	RRCP	1.70 Million updates per second	One P2PS and one src_Dist used
Cache Disabled	100 mounts; Producer 50/50	RRCP	4.02 Million updates per second	One P2PS and one src_Dist used

3.2 End-to-End RSSL/RWF Latency

Latency is defined as the time for a data item to propagate through one or more RMDS components. “End to end” 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 both the latency contribution from the API and the core infrastructure components.

NOTES:

- Caching was disabled in both the Source Distributor and the P2PS during these tests.
- Optimized binaries of the RMDS infrastructure components were used.
- NTP was disabled on the tools node, as any drifts in time will affect the reported latency.
- Tests were run with 100,000 item watchlist and RWF data update size of 74 bytes [Data file (***sample.xml***) was used]. The update size is equivalent to a 140-byte IDN update.
- Latency tests were run at each update rate for at least 5 minutes, up to the maximum sustainable update rate for a given setup.
- Decode of data was turned on in these tests.

3.2.1 RRCP Backbone Results

Update Rate [74-byte RWF messages]	Mean Latency (microsec)	Std Deviation (microsec)	Maximum Latency (microsec)	Minimum Latency (microsec)	Number of Latency Points
1000	92.91	1.86	105	89	3060
5000	101.19	4.13	219	96	3020
10000	111.65	9.10	232	105	2930
20000	140.53	21.96	296	120	3020
30000	169.70	31.92	389	142	2960
40000	175.71	24.32	402	142	2970
50000	187.09	50.38	406	143	3000
60000	227.42	60.35	676	147	3160
70000	246.68	65.06	594	153	2940
80000	264.31	58.68	1173	165	2990
90000	274.07	58.71	873	189	2930
100000	284.46	65.01	1172	190	3150
150000	361.03	118.68	1112	153	2850
200000	403.18	142.09	1819	154	3180
250000	446.97	127.02	1056	155	3080
300000	482.59	136.89	1221	156	3330
350000	504.61	137.24	1482	157	2820
400000	516.49	151.75	1632	149	3050
450000	535.85	153.50	1720	147	2980
500000	548.00	169.36	1987	147	3130
550000	572.91	177.99	2205	145	3130
600000	596.15	195.85	2341	147	2970
650000	611.71	219.19	2448	148	3140
700000	634.77	218.34	2645	148	3150
750000	662.69	257.43	2842	149	2760
800000	682.97	270.48	2955	149	3390
850000	699.97	300.74	3260	148	2930
900000	722.18	333.09	3329	147	3000
950000	736.17	337.66	3418	150	3240
1000000	759.65	352.75	3349	148	4370
1100000	813.73	423.05	3902	147	2760
1150000	809.86	384.55	3716	149	2900
1200000	850.85	425.99	3849	149	3110
1250000	882.40	473.50	3927	152	3310
1300000	961.23	567.18	4439	150	3180
1350000	1019.43	670.38	4799	152	3200
1400000	1037.34	620.34	4790	154	3360
1500000	1237.06	874.99	5662	151	6040