



Reuters Market Data System

RMDS 6.0

Performance Test Results on Sun SPARC M4000 with (1GbE) Ethernet Infrastructure

DISCLAIMER: The test results and recommendations contained in this report are made available for informational purposes only. By issuing this report, Sun Microsystems, Inc. does not guarantee similar performance results. This report is provided for your internal use only and may not be redistributed or published in any form without the prior written consent of Sun Microsystems, Inc. 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. Sun Microsystems, Inc. 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 Sun Microsystems, Inc. representative.

Issue 1.0

Date of Issue: June 22, 2007

Amjad.Khan@sun.com

Contents

1 General Information	3
1.1 Objective.....	3
Results Summary	3
1.2 Testing Methodology.....	3
1.3 Software Versions.....	3
1.3.1 RMDS.....	3
1.3.2 RMDS Test Tool.....	4
1.3.3 Operating Systems.....	4
1.4 Hardware.....	4
2 Preparation for Performance Test.....	5
2.1 Network.....	5
2.2 Hardware.....	5
2.3 Operating System Configuration.....	5
2.3.1 Solaris Kernel Parameters.....	5
2.3.2 TCP and UDP Buffers.....	5
2.3.3 Intel NIC (e1000g) Driver Settings.....	5
2.4 RMDS Configuration.....	6
2.5 Miscellaneous Notes.....	7
3 Detailed Results.....	8
3.1 RSSL/RWF Update Throughput.....	8
3.1.1 Standalone Source Distributor.....	8
3.1.2 P2PS/LAN.....	10
3.2 Update Throughput via P2PS/POP.....	12
3.2.1 RSSL/RWF.....	12

© Sun Microsystems, Inc. 2007. All Rights Reserved.

Sun Microsystems, Inc., by publishing this document, does not guarantee that any information contained herein is and will remain accurate or that use of the information will ensure correct and faultless operation of the relevant service or equipment. Sun Microsystems, Inc., its agents and employees, shall not be held liable to or through any user for any loss or damage whatsoever resulting from reliance on the information contained herein.

This document contains information proprietary to Sun Microsystems, Inc. and may not be reproduced, disclosed, or used in whole or part without the express written permission of Sun Microsystems, Inc.

Any Software, including but not limited to, the code, screen, structure, sequence, and organization thereof, and Documentation are protected by national copyright laws and international treaty provisions. This manual is subject to U.S. and other national export regulations.

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-Point Server (P2PS) and Source Distributor. The tests are grouped into two categories:

- Update throughput using RSSL/RWF data (see 3.1)
- Update throughput via P2PS/POP (see 3.2)

1.1.1 Results Summary

RMDS Components Configuration	RMDS Throughput & End-to-End Latency Using Ethernet Infrastructure
<i>P2PS-LAN, RRCP, Producer 50/50 Fanout (Cache Disabled)</i>	1,121,100 updates/sec
<i>P2PS-LAN, RVD, Producer 50/50 Fanout (Cache Disabled)</i>	1,116,050 updates/sec
<i>P2PS-POP, Producer 50/50 Fanout (Cache Enabled)</i>	1,090,800 updates/sec
<i>Src_dist, RRCP (Cache Disabled)</i>	485,000 updates/sec
<i>Src_dist, RVD (Cache Disabled)</i>	442,000 updates/sec
<i>P2PS-LAN, RRCP No Fanout (Cache Disabled)</i>	462,000 updates/sec
<i>P2PS-LAN, RVD No Fanout (Cache Disabled)</i>	395,000 updates/sec
<i>P2PS-POP, No Fanout (Cache Enabled)</i>	326,000 updates/sec

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 Reuters Wire Format (RWF) update size of 74 bytes. Tests with no fanout 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.

1.3 Software Versions

1.3.1 RMDS

src_dist ver. mdh6.0.2.L2
p2ps ver. p2ps6.0.2.L2
rrcp as included in p2ps6.0.2.L2
rvd 7.4.19

1.3.2 RMDS Test Tool

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

1.3.3 Operating Systems

- **Solaris 10 8/07** (update 4)
- **Patch Level 120011-11**

1.4 Hardware

- **Sun SPARC M4000**- 2 X 2.15 GHz dual-core SPARC64-VI processors (total 4 cores)
- **Cisco Catalyst 4948** 1 Gbps Network Switch

2 Preparation for Performance Test

2.1 Network

All the performance tests were run where the machines were connected to a private network via 1 Gbps switch (Cisco Catalyst 4948). All the network cards and switch ports were set to Auto Negotiate. For testing purposes the e1000g0 and e1000g1 interfaces were used only.

2.2 Hardware

All RMDS components were run on the same class of machine. Some test support systems in the test bed had 3.0GHz AMD Opteron processors. On all of the system the Intel (e1000g) interfaces were used for MDH backbone and TCP network traffic.

2.3 Operating System Configuration

The Solaris 10 11/06 (update 4) was installed on all systems. The e1000g network driver was used for all testing. The network parameters set via ndd commands may be set from the command line or via a Service Management Facility (SMF) manifest and method file (site/ndd) available from: <http://opensolaris.org/os/community/smf/manifests/>

For Solaris 10 update 3 and beyond, the udp_smallest_anon_port parameter should be set from the network-anon manifest and method files that are also available from the above [URL](#).

2.3.1 Solaris Kernel Parameters

Any settings changed from the defaults are noted below:

Step	Procedure		
1	OS	Enter the following lines in system file noted	System File
	Solaris	set hires_tick=1 (Only on the sink_driven_src and rmdstestclient systems)	/etc/system
		set lgrp_mem_pset_aware=1 set lgrp_mem_default_policy=3	

2.3.2 TCP and UDP Buffers

Any settings changed from the defaults are noted below:

Step	Procedure		
2	OS	Enter the following lines in system file noted	System File
	Solaris	/usr/sbin/ndd -set /dev/tcp tcp_max_buf 8388608	/lib/svc/met hod/net-init
		/usr/sbin/ndd -set /dev/tcp tcp_recv_hiwat 2097152	
		/usr/sbin/ndd -set /dev/tcp tcp_xmit_hiwat 2097152	
		/usr/sbin/ndd -set /dev/udp udp_max_buf 8388608	
		/usr/sbin/ndd -set /dev/udp udp_xmit_hiwat 1048576	
/usr/sbin/ndd -set /dev/udp udp_recv_hiwat 1048576			

2.3.3 Intel NIC (e1000g) Driver Settings

Any settings changed from the defaults are noted below:

Step	Procedure		
3	OS	Enter the following lines in the file noted	System File
	Solaris e1000g	/usr/sbin/ndd -set /dev/e1000g0 tx_interrupt_enable 0 /usr/sbin/ndd -set /dev/e1000g1 tx_interrupt_enable 0	/lib/svc/met hod/net-init

2.4 RMDS Configuration

The configuration templates—*rmds.cnf.template* and *pop.cnf.template*—were customized for the tests.

Config File	Description	Path
<i>rmds.cnf.template</i>	Configuration file	<i>\$RMDS_SW/config</i> <i>on mdh and p2ps/LAN systems</i>
<i>pop.cnf.template</i>	Configuration file to do P2PS/POP test	<i>\$RMDS_SW/config</i> <i>on p2ps/POP system only</i>

Any non-required changes (i.e., IP addresses, hostnames, etc). are noted below:

Config File	Parameter	Value
<i>rmds.cnf</i>	*RRCP*udpRecvBufSize	8192
<i>rmds.cnf</i>	*RRCP*udpSendBufSize	8192
<i>rmds.cnf</i>	*RDFD*readBias	60 (add this entry)
<i>rmds.cnf</i>	*p2ps*maxOutputBuffers	1000
<i>rmds.cnf</i>	*p2ps*guaranteedOutputBuffers	800

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
src_dist	CPU binding and interrupts	<p>Except where noted, src_dist was bound to CPU 3 and the respective transport daemon (rrcpd or rvd) was bound to CPU 0. Interrupts were disabled on CPU 3 & 0 (no-intr using psradm).</p> <p>Commands Used for process bindinging:</p> <pre>% psrset -c 3 % psrset -f 1 % psrset -b 1 `pgrep src_dist`/1</pre> <p>In case of RVD Daemon:</p> <pre>% pbind -b 0 `pgrep rvd`/3</pre>
p2ps	CPU binding and interrupts	<p>Except where noted, for p2ps tests interrupts were disabled on CPU 2 & 3 (no-intr using psradm), the primary p2ps thread was bound to CPU 3 (thread 4 with rrcpd, thread 7 with rvd), and all other p2ps threads were bound to CPU 2. All other threads of the transport daemon process were left running on CPU 0 and CPU 1.</p> <p>Commands Used for process bindinging:</p> <pre>% psrset -c 2 3 % psrset -f 1 % psrset -e 1 ./p2ps</pre> <p>In case of RRCP Daemon:</p> <pre>% pbind -b 3 `pgrep p2ps`/4 % pbind -b 2 `pgrep p2ps`/1-3,5</pre>

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	485,000	<p>Created a processor set out of CPU 3. Interrupts were disabled on CPU 3. Thread 1 of src_dist was bound to a processor set created out of CPU 3. All other src_dist threads and rrcpd daemon threads were left running on the remaining CPUs.</p> <p>FX scheduling class was assigned to src_dist and rrcpd daemon processes.</p> <p>NOTE: Two systems were used for p2ps in the test bed. Single processor CPU utilization for thread 1 of src_dist was 96%.</p>
Cache Enabled	RRCP	256,000	<p>Created a processor set out of CPU 3. Interrupts were disabled on CPU 3. Thread 1 of src_dist was bound to a processor set created out of CPU 3. All other src_dist threads and rrcpd daemon threads were left running on the remaining CPUs.</p> <p>FX scheduling class was assigned to src_dist and rrcpd daemon processes.</p>

Cache Disabled	Rendezvous	442,000	<p>Created a processor set out of CPU 3. Interrupts were disabled on CPU 0 & 3. Thread 1 of src_dist was bound to a processor set created out of CPU 3. The thread 3 of rvd was bind to CPU 0. All other src_dist threads and rvd daemon threads were left running on the remaining CPUs.</p> <p>FX scheduling class was assigned to src_dist and rvd daemon processes.</p> <p>NOTE: Two systems were used for p2ps in the test bed. Single processor CPU utilization for thread 1 of src_dist was 89%. RVD runs out of the cpu first than src_dist.</p>
Cache Enabled	Rendezvous	247,000	<p>Created a processor set out of CPU 3. Interrupts were disabled on CPU 3. Thread 1 of src_dist was bound to a processor set created out of CPU 3. All other src_dist threads and rvd daemon threads were left running on the remaining CPUs.</p> <p>FX scheduling class was assigned to src_dist and rvd daemon processes.</p>

3.1.2 P2PS/LAN

Configuration Option	Mounts : Commonality	Transport	Max Throughput	Comments
Cache Disabled	No fan-out	RRCP	462,000	<p>Created a processor set out of CPU 2 & 3. Interrupts were disabled on CPU 2 & 3. Thread 4 of p2ps was bound to CPU 3 and all other p2ps threads were bound to CPU 2. All the rrcpd daemon threads were left running on the remaining CPUs.</p> <p>FX scheduling class was assigned to p2ps and rrcpd processes.</p>
Cache Enabled	No fan-out	RRCP	246,000	<p>Created a processor set out of CPU 2 & 3. Interrupts were disabled on CPU 2 & 3. Thread 4 of p2ps was bound to CPU 3 and all other p2ps threads were bound to CPU 2. All the rrcpd daemon threads were left running on the remaining CPUs.</p> <p>FX scheduling class was assigned to p2ps and rrcpd processes.</p>
Cache Disabled	100 mounts; Producer 50/50	RRCP	22,200 input 1,121,100 output	<p>Created a processor set out of CPU 2 & 3. Interrupts were disabled on CPU 2 & 3. Thread 4 of p2ps was bound to CPU 3 and all other p2ps threads were bound to CPU 2. All the rrcpd daemon threads were left running on the remaining CPUs.</p> <p>FX scheduling class was assigned to p2ps and rrcpd processes.</p>

Cache Disabled	No fan-out	Rendezvous	395,000	<p>Created a processor set out of CPU 2 & 3. Interrupts were disabled on CPU 2 & 3. Thread 7 of p2ps was bound to CPU 3 and all other p2ps threads were bound to CPU 2. Thread 3 of rvd was bound to CPU 0 and all the other rvd daemon threads were left running on the remaining CPUs.</p> <p>FX scheduling class was assigned to p2ps and rvd daemon processes.</p>
Cache Enabled	No fan-out	Rendezvous	218,000	<p>Created a processor set out of CPU 2 & 3. Interrupts were disabled on CPU 2 & 3. Thread 7 of p2ps was bound to CPU 3 and all other p2ps threads were bound to CPU 2. All the other rvd daemon threads were left running on the remaining CPUs.</p> <p>FX scheduling class was assigned to p2ps and rvd daemon processes.</p>
Cache Disabled	100 mounts; Producer 50/50	Rendezvous	22,100 input 1,116,050 output	<p>Created a processor set out of CPU 2 & 3. Interrupts were disabled on CPU 2 & 3. Thread 7 of p2ps was bound to CPU 3 and all other p2ps threads were bound to CPU 2. All the other rvd daemon threads were left running on the remaining CPUs.</p> <p>FX scheduling class was assigned to p2ps and rvd daemon processes.</p> <p>The throughput number was obtained with single processor CPU utilization of 100% for p2ps.</p>

3.2 Update Throughput via P2PS/POP

These tests were performed using a P2PS/POP with data provided by an upstream P2PS/LAN, with an RRCP transport.

3.2.1 RSSL/RWF

Configuration Option	Mounts : Commonality	Max Throughput	Comments
Cache Enabled	No fan-out	326,000	<p>Interrupts were disabled on CPU 2 & 3. Thread 4 of p2ps was bound to CPU 3 and all other p2ps threads were bound to CPU 2.</p> <p>FX scheduling class was assigned to p2ps process.</p> <p>The throughput number was obtained with single processor CPU utilization of 98% for p2ps/POP.</p>
Cache Enabled	100 mounts; Producer 50/50	21,600 input 1,090,800 output	<p>Interrupts were disabled on CPU 2 & 3. Thread 4 of p2ps was bound to CPU 3 and all other p2ps threads were bound to CPU 2.</p> <p>FX scheduling class was assigned to p2ps process.</p> <p>The throughput number was obtained with single processor CPU utilization of 99% for p2ps/POP thread 4.</p>