

Oracle database performance on Sun and Dell servers running Solaris

Executive summary

Dell Inc. (Dell) commissioned Principled Technologies (PT) to measure the Oracle database performance of the following two servers running Sun's Solaris operating system:

- Dell PowerEdge R900 running Oracle 10g R2 Enterprise Edition on Solaris 10
- Sun Fire V440 running Oracle 9i Enterprise Edition on Solaris 9

Our goal was to compare a typical existing installation with a typical new purchase configuration, so we could analyze what customers might experience if they were staying with Solaris and Oracle but migrating from RISC to x86. Our goal was not to achieve maximum performance but rather to focus on the performance typical of such a real-world scenario. We selected a common system configuration for an existing Sun™ environment—a 3-year-old Sun Fire v440 with 32 GB of RAM and all of its disk drive bays full (four 73GB 10K drives)—and compared it to a moderate configuration of a new six-core Xeon 7400 series-based Dell PowerEdge R900 server with 64 GB of RAM and all of its disk drive bays full (eight 73GB 15K drives).

We used the DVD Store™ Version 2 (DS2) test tool, which is available from <http://www.delltechcenter.com/page/DVD+Store> at no charge. DS2 is an open-source simulation of an online e-commerce DVD store. Virtual customers log in; browse movies by actor, title, or category; and purchase movies. The workload also creates new customers. Browsing movies involves select operations, some of which use Oracle Text search, and some of which do not. The purchase, login, and new customer procedures involve updates and inserts, as well as selects. The workload's main reporting metric is orders per minute, or OPM. Unless we state otherwise, all results in this report are in orders per minute.

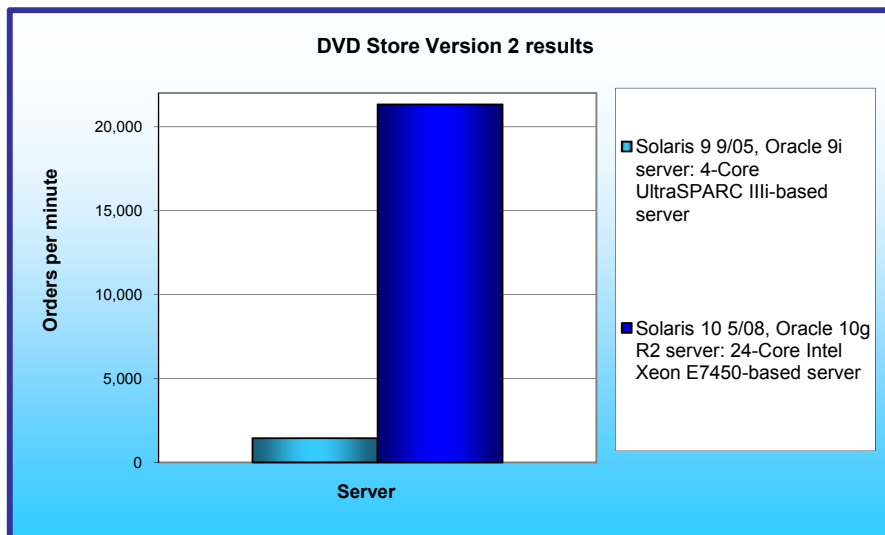


Figure 1: DS2 results of the test configurations. Higher numbers are better.

KEY FINDING

The Dell PowerEdge R900 running Oracle 10g R2 Enterprise Edition on Solaris 10 achieved 13 times greater performance than the Sun Fire V440 running Oracle 9i Enterprise Edition on Solaris 9 (see Figure 1).

Figure 1 shows the DVD Store performance of each configuration in orders per minute (OPMs). Each result is the median peak score of three runs of the benchmark. A higher number of OPMs indicates the server can handle a greater load. The Dell PowerEdge R900 running Oracle 10g R2 Enterprise Edition on Solaris 10 achieved a score of 21,330, which is 13 times greater than the performance of the Sun Fire V440 running Oracle 9i Enterprise Edition on Solaris 9, which achieved a score of 1,447.

Figure 2 details the results, each of which is the median peak score of three runs of the benchmark. Higher scores are better.

Server	OPMs
Dell PowerEdge R900 running Oracle 10g R2 Enterprise Edition on Solaris 10	21,330
Sun Fire V440 running Oracle 9i Enterprise Edition on Solaris 9	1,447

Figure 2: DS2 results for the two systems under test. Higher numbers are better.

Test methodology

Because our goal was to isolate the performance of the servers, we used the directly attached drives for storage. Had we used external storage, the performance of the external storage subsystem could have changed the focus of the test results from the servers to the storage. (See Appendix A for full configuration information on each server.) The Sun Fire V440 had four internal drives, which we configured as a two-disk RAID 1 for the Solaris 9 operating system, a drive for the Oracle system tablespaces, and a drive for the DS2 data. The Dell PowerEdge R900 had eight internal drives, which we configured as a two-disk RAID 1 for the Solaris operating system and Oracle software, and three two-disk RAID 1s for database storage, managed by Oracle's Automatic Storage Management (ASM).

On the Sun Fire V440, we installed Solaris 9 9/05 Operating System for SPARC-based systems (64-bit) (Solaris 9) and Oracle 9i Enterprise Edition Release 9.2.0.1.0 for Solaris on SPARC™-based systems (64-bit) (Oracle 9i). We gave the Sun Fire V440 Oracle's recommended amount of swap space, which is the amount of physical memory in the system (32,768 MB). We set the kernel parameters as follows per Oracle's guidelines (http://download.oracle.com/docs/html/A96167_01/pre.htm#CHDCEHBG). After installation, we set the database buffer cache size to 9 GB and the pga_aggregate_target to 4 GB, and kept the database in NOARCHIVELOG mode, the default.

```
set semsys:seminfo_semmani=100
set semsys:seminfo_semmns=1024
set semsys:seminfo_semmnsl=256
set shmsys:shminfo_shmmax=4294967295
set shmsys:shminfo_shmmin=1
set shmsys:shminfo_shmmani=100
set shmsys:shminfo_shmseg=10
```

On the Dell PowerEdge R900, we installed Solaris 10 5/08 Operating System for x86-based systems (64-bit) (Solaris 10) and Oracle 10g R2 Enterprise Edition Release 10.2.0.1.0 for Solaris on x86_64-based systems (Oracle 10g R2). We set only one kernel parameter: we raised the maximum shared memory segment size to 32 GB. We also had to edit the /etc/system file to deal with a known issue the 5/08 release of Solaris has on systems with more than 16 cores. For details, see *Migrating from Solaris 9 Enterprise on Sun Fire V440 with Oracle 9i Enterprise Edition to Solaris 10 Enterprise on Dell R900 with Oracle 10g R2 Enterprise Edition* at http://principledtechnologies.com/Clients/Reports/Dell/Solarisv440_to_Solaris_R900_migration_guide.pdf.

DS2 execution

The DS2 stress tool provides options to generate 10MB, 1GB, or 100GB datasets. To get the tool to generate the 20 GB of user data we used in this test, we had to make a few straightforward changes to the source code. We detail these changes and explain how to generate the data in Appendix B.

We also had to modify the DS2 client so that it would be aware of the 20GB database size. We detail these changes and explain how to generate the data in Appendix C.

When building the DS2 database schema on the Sun Fire V440, we followed Dell's provided build scripts available for download via the DS2 Web site. We made modifications to pathnames as appropriate on our system, as well as file sizes to account for the 20GB database. We sized the physical files to be approximately one-third

the size of the default large (100GB) scripts Dell provided, as we were only using 20GB database size instead of 100GB. This allowed ample unallocated space to ensure that file growth during testing was not an issue. We loaded data using shell scripts Dell provided.

For details about how we ported the data from the Sun Fire V440 to the Dell PowerEdge R900, see *Migrating from Solaris 9 Enterprise on Sun Fire V440 with Oracle 9i Enterprise Edition to Solaris 10 Enterprise on Dell R900 with Oracle 10g R2 Enterprise Edition* at http://principledtechnologies.com/Clients/Reports/Dell/Solarisv440_to_Solaris_R900_migration_guide.pdf.

On the PowerEdge R900, we encountered very high levels of latch contention on one of the DS2 SQL statements, a phenomenon that can occur on systems with very high throughput levels. We addressed this issue by setting the Oracle `_SPIN_COUNT` parameter to 500,000. This change let the test run without the latch contention artificially lowering the system's performance.

We used a desktop-class system, which we connected to our network via a gigabit switch, to generate the workload for our tests. The system contained the DS2 driver application and executed a workload against a single database on the server. DS2 spawned 20 threads and ran without think time. This command line lists the exact parameters we used:

```
C:\ds2_oracle\ds2oracledriver --target=r900oracle --ramp_rate=10 --run_time=60  
--n_threads=20 --db_size_str=W --think_time=0
```

The DS2 driver application creates an orders-per-minute (OPM) performance counter on all systems. We created a data collector set to collect the OPM statistics once every second.

Appendix A – Test system configuration information

This appendix provides detailed configuration information about the test servers.

Server	Dell PowerEdge R900	Sun Fire V440
General processor setup		
Number of processor packages	4	4
Number of cores per processor package	6	1
Number of hardware threads per core	1	1
CPU		
Vendor	Intel	Sun
Name	Xeon 7450	UltraSPARC IIIi
Socket type	PGA604	959
Core frequency (GHz)	2.4GHz	1.59 GHz
Front-side bus frequency (MHz)	1066 MHz	JBus running at up to 200 MHz
L1 cache	32 KB instruction 32 KB data	64 KB 4-way Data cache 32 KB 4-way Instruction cache
L2 cache	9 MB shared L2 cache	1 MB on-chip, 4-way set associative
L3 cache	12MB shared L3 cache	N/A
Platform		
Vendor and model number	Dell PowerEdge R900	Sun Fire V440
Motherboard model number	CN-0947H-13740-87N-000P	540-5418-06
Motherboard revision number	X00	50
BIOS name and version	Dell BIOS 1.1.9	OBP 4.22.33 2007/06/10 12:42 Sun Fire V440v Netra 440
BIOS settings	default	default
Memory module(s)		
Vendor and model number	Samsung M395T5750EZ4-CE65	Micron MT36VDDF25672G- 335D2
Type	PC2-5300F	PC2700R DDR 333
Speed (MHz)	667 MHz	333 MHz
Speed in the system currently running @ (MHz)	667 MHz	333 MHz
Size	2GB	2 GB
Number of RAM modules	32	16
Chip organization	Double sided	Double sided
Hard disk		
Vendor and model number	Seagate ST973451SS	Fujitsu MAP3735NC
Number of disks in system	8	4
Size	73 GB	73.5 GB
Buffer size	16 MB	8 MB
RPM	15,000	10,000
Type	SAS / Serial Attached SCSI	Ultra320 SCSI
Controller	PERC 6/i	LSI 1030
Driver	mega_sas (00.00.01.23)	SCSI Disk Driver 1.435
Operating system		
Name	Solaris 10 5/08 Operating System for x86-based systems (64-bit)	Solaris 9 9/05 Operating System for SPARC-based systems (64- bit)

Server	Dell PowerEdge R900	Sun Fire V440
Build number	Solaris 10 5/08 s10x_u5wos_10 x86	Solaris 9 9/05 HW s9s_u9wos_06b SPARC
File system	UFS	UFS
Kernel	SunOS 5.10 Generic_127128-11	SunOS 5.9 Generic_118558-34
Language	English	English
Database software		
Name	Oracle 10g R2 Enterprise Edition Release 10.2.0.1.0 for Solaris on x86_64-based systems	Oracle 9i Enterprise Edition Release 9.2.0.1.0 for Solaris on SPARC-based systems (64-bit)
Build number	10.2.0.1.0	9.2.0.1.0
Graphics		
Vendor and model number	ATI ES 1000	None
Resolution	1280 x 1024	N/A (Used command line interface)
Network card/subsystem		
Vendor and model number	Broadcom BCM5708	Broadcom BCM5421
Type	4 x integrated 4 x PCI card	2 x Integrated
Driver	4.4.3	CE Ethernet Driver v1.158
Optical drive		
Vendor and model number	Toshiba Samsung TS-L333A	Toshiba Samsung SD-C2612
USB ports		
Number	4	4
Type	USB 2.0	USB 1.0
Power Supplies		
Total number	2	2
Wattage of each	1570	680
Cooling Fans		
Total Number	4 x Nidec UT094 4 x Nidec V34809-35DELF	2 x NMB BG9003-B044-00S 1 x NMB 4715kl-04W-B59
Dimensions	Nidec UT094: 88 mm(h) x 80 mm(w) Nidec V34809-35DELF: 120 mm(h) x 140 mm(w)	NMB BG9003-B044-00S: 33 mm (h) x 97 mm (w) NMB 4715kl-04W-B59: 125 mm (h) x 120 mm (w)
Voltage	Nidec UT094: 12 Nidec V34809-35DELF: 12	NMB BG9003-B044-00S: 12 NMB 4715kl-04W-B59: 12
Amps	Nidec UT094: 1.76 Nidec V34809-35DELF: 3.3	NMB BG9003-B044-00S: 1.34 NMB 4715kl-04W-B59: 1.30

Figure 3: Detailed system configuration information for the two test configurations.

Appendix B – Changes we made to the data generation code and how we generated the test data

We built the CSV files to import into Oracle according to the instructions and recommendations in the DS2 readme file.

Code changes to the ds2_create_orders.c module

The module ds2_create_orders.c defines constants that set the bounds of the customer ID and the product ID. The constants for the 20GB database size did not exist. We added those.

On the command line for the ds2_create_orders module, we specified the size. The available options were S (small), M (medium), and L (large). We added the case W for the 20GB database. In the switch statement that sets the values for the variables max_cust_id and max_prod_id, we added cases that assigned them the proper values for the 20GB database size.

We recompiled the ds2_create_orders.c module following the instructions in the header comments. We used the following command line: `gcc -o ds2_create_orders ds2_create_orders.c -lm`

Code changes to the ds2_create_cust.c module

We had to make the same changes to the ds2_create_cust.c module that we made to the ds2_create_orders.c module. The module ds2_create_cust.c defines constants that set the bounds of the product ID. The constants for the 20GB database size did not exist. We added those constants.

On the command line for the ds2_create_cust module, we specified the size. The available options were S (small), M (medium), and L (large). We added the case W for the 20GB database. In the switch statement that sets the values for the variables max_cust_id and max_prod_id, we added cases that assigned them the proper values for the 20GB database size.

We recompiled the ds2_create_cust.c module following the instructions in the header comments. We used the following command line: `gcc -o ds2_create_cust ds2_create_cust.c -lm`

The scripts we used to generate the data

We used scripts to run all four executables. The distribution did not include scripts for the 20GB size. We wrote scripts using the ds2_create_cust_large.sh and ds2_create_orders_large.sh scripts as starting points. The ds2_create_prod and ds2_create_inv executables did not ship with associated scripts, so we created scripts using the instructions in the readme files.

We found two dependencies:

- In the file ds2_create_prod_readme.txt, it says:
In ../orders, after creating orderlines files, run ds2_create_inv to total up sales by product and create inventory load file:
`ds2_create_inv n_prods > ../prod/inv.csv`

Then in this directory:
`ds2_create_prod n_prods > prod.csv`
- The parameter you give to ds2_create_inv.sh must match the number you gave to ds2_create_orders.sh. The inv program uses the product id as an array index. You must have runds2_create_orders.sh first for the lookup to work.

Therefore, we had to run the scripts in a specific order. We ran them in the following sequence to create the 20GB database:

- ds2_create_orders_20gb.sh

- ds2_create_inv_20gb.sh
- ds2_create_prod_20gb.sh
- ds2_create_cust_20gb.sh

The scripts launch all their tasks in the background. Because there are order dependencies, we had to check and make sure each process had completed before we moved to the next step. We used the command `ps -A | grep ds2`. Because the scripts create only processes with the “ds2” prefix in the process name, this command displayed the list of currently running processes that the last script we launched had created.

Appendix C – Code changes to the DS2 client

Changes to the ds2xdriver.cs module

To use the 20GB database we created earlier, we had to change the following constants:

- In the routine Controller(), we changed the string “sizes”. We added the W option for the 20GB database size. DS2 uses the sizes string to interpret the db_size_str option.
- In the class Controller, we changed the arrays MAX_CUSTOMER and MAX_PRODUCT. To each, we added values specifying the bounds for the customer and product IDs. The Controller() routine uses these arrays.

Building the ds2oracledriver.exe executable

We recompiled the ds2xdriver.cs and ds2oraclefns.cs module by following the instructions in the header comments. Because the instructions were for compiling from the command line, we used the following steps:

1. Opened a command prompt.
2. Used the `cd` command to change to the directory containing our sources.
3. Ran the batch file `C:\Program Files\Microsoft Visual Studio 9.0\Common7\Tools\vsvars32.bat`. This set up the environment variables.
4. Executed the following command: `csc /out:ds2oracledriver.exe ds2xdriver.cs ds2oraclefns.cs /d:USE_WIN32_TIMER /d:GEN_PERF_CTRS`



Principled Technologies, Inc.
1007 Slater Road, Suite 250
Durham, NC 27703
www.principledtechnologies.com
info@principledtechnologies.com

Principled Technologies is a registered trademark of Principled Technologies, Inc.
All other product names are the trademarks of their respective owners.

Disclaimer of Warranties; Limitation of Liability:

PRINCIPLED TECHNOLOGIES, INC. HAS MADE REASONABLE EFFORTS TO ENSURE THE ACCURACY AND VALIDITY OF ITS TESTING, HOWEVER, PRINCIPLED TECHNOLOGIES, INC. 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 PRINCIPLED TECHNOLOGIES, INC., ITS EMPLOYEES AND ITS SUBCONTRACTORS 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 PRINCIPLED TECHNOLOGIES, INC. BE LIABLE FOR INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH ITS TESTING, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL PRINCIPLED TECHNOLOGIES, INC.'S LIABILITY, INCLUDING FOR DIRECT DAMAGES, EXCEED THE AMOUNTS PAID IN CONNECTION WITH PRINCIPLED TECHNOLOGIES, INC.'S TESTING. CUSTOMER'S SOLE AND EXCLUSIVE REMEDIES ARE AS SET FORTH HEREIN.