

Executive summary

We compared the database performance, hardware cost, and power utilization of two servers in similar common enterprise configurations. On our tests of Microsoft® SQL Server® 2008 database orders per minute (OPM), the Dell™ PowerEdge™ R710 delivered multiple advantages over the HP ProLiant DL380 G6.

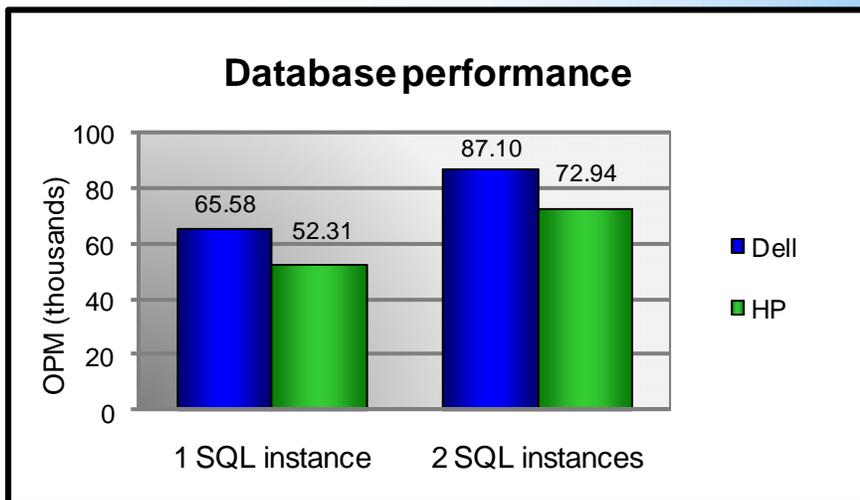


Figure 1: One- and two-database-instance performance results. Higher numbers are better.



Figure 2: One- and two-database-instance normalized price-performance results. Higher numbers are better.



Figure 3: One- and two-database-instance normalized performance/watt results. Higher numbers are better.

KEY FINDINGS

The Dell PowerEdge R710 delivered **25% greater database performance** than the HP ProLiant DL380 G6 with one database instance and **19% greater** with two database instances.

The Dell PowerEdge R710 delivered **32% better price-performance** than the HP ProLiant DL380 G6 with one database instance and **29% better** with two database instances.

The Dell PowerEdge R710 delivered **14% greater performance/watt** than the HP ProLiant DL380 G6 with one database instance and **13% greater** with two database instances.

What we tested

We conducted our testing using DVD Store Version 2 (DS2) to generate realistic e-commerce transactional streams where customers log in, browse, and order products from an online DVD store. DS2 is an open-source application with a back-end database component, a front-end Web application layer, and a driver layer that operates as the middle tier and actually executes the workload. We measure DS2 performance in terms of orders OPM, the workload's main reporting metric. We calculate OPM as the number of overall orders processed during each test, divided by the elapsed seconds multiplied by 60. We report the last OPM score the benchmark reported.

When DS2 is executing, simulated 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 may involve SQL Server 2008 full-text search. The purchase, login, and new customer stored procedures involve execution of database update, insert, and select statements.

Because our goal was to isolate and test database server performance, we did not use the front-end Web application layer. Instead, we ran the driver application on client machines directly via its command-line interface. We used the default DS2 parameters and setup configuration, with the exceptions we note in the DVD Store setup section in the Test methodology section of this report.

For client machines, we used standard Intel® Core™ 2 Duo laptop systems running Microsoft Windows Vista® Professional. Each of these clients ran a single instance of DS2 that spawned 32 threads. To simulate a heavily loaded environment, the load-generating client systems ran with no think time, issuing requests as quickly as the server could handle them. Our goal was to maximize processor utilization for the given server configurations using a demanding workload.

Figure 4 shows the system configuration overview for the two similarly configured Dell and HP servers.

| Servers | Dell PowerEdge R710 | HP ProLiant DL380 G6 |
|------------------|--|---|
| Processors | Dual Intel Xeon® Processor E5540, 2.53 GHz | Dual Intel Xeon Processor E5540, 2.53 GHz |
| Memory | 6 x 8GB PC3-8500R DDR3 | 6 x 8GB PC3-8500R DDR3 |
| Hard disks | 8 x 73GB, 15KRPM, SAS | 8 x 73GB, 15KRPM, SAS |
| Power supplies | 2 Dell A570P-00, 570w | 2 HP HSTNS-PL14, 460w |
| Disk controller | Dell PERC 6/i Integrated RAID Controller, 256MB cache and BBWC | HP 410i Integrated Smart Array Controller, 256MB cache and BBWC |
| Operating system | Windows Server® 2008 R2 Enterprise | Windows Server 2008 R2 Enterprise |
| Database | Microsoft SQL Server 2008 SP1 | Microsoft SQL Server 2008 SP1 |

Figure 4: System configuration overview for the two test servers. See Appendix A for more details.

For each server, we configured its eight internal SAS hard drives as two RAID 1+0 arrays, one composed of two drives and another composed of six drives, and associated each of these arrays to single separate logical volumes. We used the two-drive array for the OS and database logs, and the six-drive array to store data.

The DS2 driver application creates an orders-per-minute performance counter on each client machine. The DVD Store client application visually outputs OPM at 10-second intervals, but we chose to collect this OPM metric via a performance monitor counter on each client machine at 1-second intervals.

We ran this workload on both servers for 30 minutes.

For more details about the DS2 tool, see <http://www.delltechcenter.com/page/DVD+Store>.

What we found

To achieve maximum DS2 results on both servers, we installed SQL Server on each, and ran two separate tests with one- and two-database instances. Note: Each SQL Server instance is capable of running multiple databases. There is no additional software cost for adding these additional instances. We ran one 10GB DS2 database in each instance. We show below the number of OPM each server achieved for the tests. In the two-database-instance test case, the total OPM is the sum of the OPM achieved by each separate instance.

Figures 5 and 6 illustrate the actual results for the Dell PowerEdge R710 server and the HP ProLiant DL380 G6 server, in total OPM, for each of the three one- and two-database-instance tests, respectively. On the median of three runs, the Dell PowerEdge R710 delivered 25% greater database performance than the HP ProLiant DL380 G6 with one database instance and 19% greater with two database instances. Higher OPM numbers are better.

| One-database-instance performance results | | | |
|---|-----------|-----------|-----------|
| | Run 1 | Run 2 | Run 3 |
| | Total OPM | Total OPM | Total OPM |
| Dell PowerEdge R710 server | 65,493 | 65,720 | 65,583 |
| HP ProLiant DL380 G6 server | 52,313 | 52,168 | 53,375 |

Figure 5: One-database-instance performance results for the Dell PowerEdge R710 server and the HP ProLiant DL380 G6 server while running the DVD Store workload. Higher OPM numbers are better.

| Two-database-instance performance results | | | |
|---|-----------|-----------|-----------|
| | Run 1 | Run 2 | Run 3 |
| | Total OPM | Total OPM | Total OPM |
| Dell PowerEdge R710 server | 87,103 | 86,544 | 87,646 |
| HP ProLiant DL380 G6 server | 72,943 | 74,956 | 71,822 |

Figure 6: Two-database-instance performance results for the Dell PowerEdge R710 server and the HP ProLiant DL380 G6 server while running the DVD Store workload. Higher OPM numbers are better.

Figures 7 and 8 show the specific workload power results during the median run of the Dell PowerEdge R710 server and HP ProLiant DL380 G6 server, in watts, for each of the one- and two-database-instance tests, respectively. To calculate the power, we averaged the power we recorded during our workload measurement period, minutes 24 through 29 of each test.

| One-database-instance median power results | |
|--|----------------|
| | Workload power |
| Dell PowerEdge R710 server | 226.2 watts |
| HP ProLiant DL380 G6 server | 206.1 watts |

Figure 7: One-database-instance median power results, in watts, for the Dell PowerEdge R710 server and the HP ProLiant DL380 G6 server while running the DVD Store workload. Results shown are from the runs that produced the OPM scores in Figure 6.

| Two-database-instance median power results | |
|--|----------------|
| | Workload power |
| Dell PowerEdge R710 server | 240.9 watts |
| HP ProLiant DL380 G6 server | 228.1 watts |

Figure 8: Two-database-instance median power results, in watts, for the Dell PowerEdge R710 server and the HP ProLiant DL380 G6 server while running the DVD Store workload. Results shown are from the runs that produced the OPM scores in Figure 7.

The Dell PowerEdge R710 server's retail price is \$11,268, and the HP ProLiant DL380 G6 server's retail price is \$13,224. The Dell PowerEdge R710 server delivered better price-performance than the HP ProLiant DL380 G6 server: 32% better with one database instance and 29% better with two database instances.

Figures 9 and 10 show the median price-performance gain of the Dell PowerEdge R710 server over the HP ProLiant DL380 G6 server, for both one- and two-instance database tests. The Dell PowerEdge R710 delivered

32% better price-performance than the HP ProLiant DL380 G6 with one database instance and 29% better with two database instances. Lower dollars/OPM numbers are better.

| One-database-instance price-performance results | |
|---|-------------|
| | Dollars/OPM |
| Dell PowerEdge R710 server | 0.172 |
| HP ProLiant DL380 G6 server | 0.253 |

Figure 9: One-database-instance price-performance results for the Dell PowerEdge R710 server and the HP ProLiant DL380 G6 server while running the DVD Store workload. Lower dollars/OPM numbers are better.

| Two-database-instance price-performance results | |
|---|-------------|
| | Dollars/OPM |
| Dell PowerEdge R710 server | 0.129 |
| HP ProLiant DL380 G6 server | 0.181 |

Figure 10: Two-database-instance price-performance results for the Dell PowerEdge R710 server and the HP ProLiant DL380 G6 server while running the DVD Store workload. Lower dollars/OPM numbers are better.

Figures 11 and 12 show the median performance/watt results of the Dell PowerEdge R710 server and the HP ProLiant DL380 G6 server, for both one- and two-instance database tests. The Dell PowerEdge R710 delivered 14% greater performance/watt than the HP ProLiant DL380 G6 with one database instance and 13% more with two database instances. Higher OPM/watt numbers are better.

| One-database-instance performance/watt results | |
|--|----------|
| | OPM/watt |
| Dell PowerEdge R710 server | 289.9 |
| HP ProLiant DL380 G6 server | 253.8 |

Figure 11: One-database-instance performance/watt results for the Dell PowerEdge R710 server and the HP ProLiant DL380 G6 server while running the DVD Store workload. Higher OPM/watt numbers are better.

| Two-database-instance performance/watt results | |
|--|----------|
| | OPM/watt |
| Dell PowerEdge R710 server | 361.5 |
| HP ProLiant DL380 G6 server | 319.8 |

Figure 12: Two-database-instance performance/watt results for the Dell PowerEdge R710 server and the HP ProLiant DL380 G6 server while running the DVD Store workload. Higher OPM/watt numbers are better.

How we tested

Adjusting BIOS settings

We used the default BIOS settings on both the Dell PowerEdge R710 server and HP ProLiant DL380 G6 server, except that we disabled hardware pre-fetch and node interleaving.

Setting up and configuring the Dell PowerEdge R710 server

We installed eight internal drives in the Dell PowerEdge R710 server. We configured these drives as two RAID 1+0 arrays, one array with two disks for OS and logs and another array with six disks for data.

We installed Windows Server 2008 R2 and SQL Server 2008 software on the first mirrored internal volume, configured the SQL Server transaction logs on the same volume, and placed the SQL Server database data on the second volume.

Setting up the internal storage for operating system installation and SQL Server transaction logs

1. Enter the RAID controller BIOS by pressing Ctrl+R at the prompt.
2. Highlight Controller 0, and press F2.
3. Select Create New VD.
4. Select the first two drives, select RAID level 1, tab to the OK button, and press Enter. Accept the warning regarding initialization.
5. Select the new virtual drive, press F2, and select Initialization to begin Initialization.

6. Repeat steps 2 through 4 for the second internal volume.
7. Wait for the initialization operations to complete.
8. Press Escape, and choose Save and Exit to return to the boot sequence.

Installing Windows Server 2008 R2

1. Insert the installation DVD for Windows Server 2008 R2 into the DVD drive.
2. Choose the language, time and currency, and keyboard input. Click Next.
3. Click Install Now.
4. Choose Windows Server Enterprise (Full Installation). Click Next.
5. Accept the license terms, and click Next.
6. Click Custom.
7. Click the Disk, and click Drive options (advanced).
8. Click New, Apply, Format, and click Next.
9. After the installation completes, click OK to set the Administrator password.
10. Enter the administrator password twice, and click OK.
11. Click Start, type `change power-saving settings` and press Enter.
12. Click Change plan settings.
13. Change the Turn off the display drop-down menu to Never.
14. Click Save changes, and close the Power Options, Screen Saver Settings, and Personalization windows.

Installing system updates in Windows Server 2008 R2

We installed the following updates on the system using the Microsoft Windows Update feature:

- Security Update for Windows Server 2008 R2 x64 Edition (KB974571)
- Update for Windows Server 2008 R2 x64 Edition (KB974431)
- Security Update for Internet Explorer 8 for Windows Server 2008 R2 x64 Edition (KB974455)
- Security Update for ActiveX killbits for Windows Server 2008 R2 x64 Edition (KB973525)
- Security Update for Windows Server 2008 R2 x64 Edition (KB975467)

Installing SQL Server 2008 on the server

1. Insert the DVD into the physical DVD drive.
2. Click Run SETUP.EXE on the AutoPlay menu.
3. If the application prompts you to install the .NET Framework, click OK.
4. Select the Accept radio button, and click Install.
5. When the installation finishes, click Exit.
6. If the application prompts you, click OK to install a hotfix for Windows (KB942288).
7. Click Restart now to restart the server upon completion.
8. After rebooting, log into the server.
9. Click Start, and click Computer.
10. Double-click the CD/DVD drive.
11. Click Installation.
12. Click New SQL Server stand-alone installation.
13. On the Setup support rules screen, click OK.
14. Specify the Enterprise Evaluation free edition. Click Next.
15. Click the checkbox to accept the license terms. Click Next.
16. Click Install to install the setup support files.
17. You may see a Windows Firewall warning. For now, ignore this, and click Next.
18. On the Feature Selection screen, select Database Engine Services, Full-Text Search, Client Tools Connectivity, Client Tools Backward Compatibility, Management Tools –Basic, and Management Tools – Complete. Click Next.
19. On the Instance configuration screen, leave the default selection of default instance, and click Next.
20. On the Disk space requirements screen, click Next.
21. On the Server configuration screen, choose NT AUTHORITY\SYSTEM for SQL Server Agent, and choose NT AUTHORITY\SYSTEM for SQL Server Database Engine. Click Next.
22. On the Database Engine Configuration screen, select Mixed Mode.
23. Enter a password for the system administrator account.
24. Click Add Current user. This may take several seconds.
25. Click Next.
26. On the Error and usage reporting screen, click Next.
27. On the Installation rules screen, click Next.

28. On the Ready to Install screen, click Next.
29. After installation completes, click Start, All Programs, Microsoft SQL Server 2008, Configuration Tools, SQL Server Configuration Manager.
30. Expand the Network Configuration node, and select the relevant instance.
31. In the right pane, right-click TCP/IP, and choose Enable.
32. In the left pane, click SQL Server Services.
33. In the right pane, right-click the relevant instance, and select Restart.
34. Repeat steps 9 through 33 for the second, third, and fourth SQL Server instances (on each instance install, choose Database Engine Services and Full-Text Search on the feature selection screen, and choose named instance on the instance configuration screen).
35. After all installations complete, open SQL Server Configuration Manager, right-click the SQL Browser service, change the startup type to Automatic, and start the SQL Browser service.
36. Click Start, Administrative Tools, and Local Security Policy.
37. Double-click Perform Volume Maintenance Tasks, and add the System account.
38. Double-click Lock Pages in Memory, and add the System account.
39. Click Start, All Programs, Microsoft SQL Server 2008, and SQL Server Management Studio.
40. In the Object Explorer pane, choose Connect→Database Engine. Choose the default instance.
41. Right-click the instance, and choose properties.
42. Click Processors, and check the Boost SQL Server Priority checkbox. Click OK.
43. Create a SQL Server login for the ds2user (see the DVD Store setup section below for the specific script to use).
44. Repeat steps 39 through 43 on the additional SQL Server instances.
45. Place the DVD Store backup file you prepared in the default location for SQL Server: C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\Backup.

Installing SQL Server 2008 SP1

Service Pack 1 is available for download from

<http://www.microsoft.com/downloads/details.aspx?FamilyID=66ab3dbb-bf3e-4f46-9559-ccc6a4f9dc19&displaylang=en>.

1. Run SQLServer2008SP1-KB968369-x64-ENU.exe.
2. On the Welcome screen, click Next
3. On the License terms screen, accept the license terms, and click Next.
4. On the Features screen, select available features, and click Next.
5. On the Check files in use screen, click Next.
6. On the Update screen, click Next.
7. When the update completes, click Next, and click Close.

Setting up the HP ProLiant DL380 G6 server

We followed a setup process identical to that of the Dell PowerEdge R710. We configured the eight drives as two RAID 1+0 arrays, one array with two disks for OS and logs and another array with six disks for data.

We installed Windows Server 2008 R2 and SQL Server 2008 software on the first mirrored internal volume, configured the SQL Server transaction logs on the same first internal volume, and placed the SQL Server database data on the second volume. We installed a fresh copy of Windows Server 2008 R2 following the same process as for the Dell PowerEdge R710 server.

Installing Windows Server 2008 R2

1. Insert the installation DVD for Windows Server 2008 R2 into the DVD drive.
2. Choose the language, time and currency, and keyboard input. Click Next.
3. Click Install Now.
4. Choose Windows Server Enterprise (Full Installation). Click Next.
5. Accept the license terms, and click Next.
6. Click Custom.
7. Click the Disk, and click Drive options (advanced).
8. Click New, Apply, Format, and click Next.
9. After the installation completes, click OK to set the Administrator password.
10. Enter the administrator password twice, and click OK.
11. Click Start, type `change power-saving settings` and press Enter.
12. Click Change plan settings.

13. Change the Turn off the display drop-down menu to Never.
14. Click Save changes, and close the Power Options, Screen Saver Settings, and Personalization windows.

Installing system updates in Windows Server 2008 R2

We installed the following updates on the system using the Microsoft Windows Update feature:

- Security Update for Windows Server 2008 R2 x64 Edition (KB974571)
- Update for Windows Server 2008 R2 x64 Edition (KB974431)
- Security Update for Internet Explorer 8 for Windows Server 2008 R2 x64 Edition (KB974455)
- Security Update for ActiveX killbits for Windows Server 2008 R2 x64 Edition (KB973525)
- Security Update for Windows Server 2008 R2 x64 Edition (KB975467)

Installing SQL Server 2008 on the server

1. Insert the DVD into the physical DVD drive.
2. Click Run SETUP.EXE on the AutoPlay menu.
3. If the application prompts you to install the .NET Framework, click OK.
4. Select the Accept radio button, and click Install.
5. When the installation finishes, click Exit.
6. If the application prompts you, click OK to install a hotfix for Windows (KB942288).
7. Click Restart now to restart the server upon completion.
8. After rebooting, log into the server.
9. Click Start, and click Computer.
10. Double-click the CD/DVD drive.
11. Click Installation.
12. Click New SQL Server stand-alone installation.
13. On the Setup support rules screen, click OK.
14. Specify the Enterprise Evaluation free edition. Click Next.
15. Click the checkbox to accept the license terms. Click Next.
16. Click Install to install the setup support files.
17. You may see a Windows Firewall warning. For now, ignore this, and click Next.
18. On the Feature Selection screen, select Database Engine Services, Full-Text Search, Client Tools Connectivity, Client Tools Backward Compatibility, Management Tools –Basic, and Management Tools – Complete. Click Next.
19. On the Instance configuration screen, leave the default selection of default instance, and click Next.
20. On the Disk space requirements screen, click Next.
21. On the Server configuration screen, choose NT AUTHORITY\SYSTEM for SQL Server Agent, and choose NT AUTHORITY\SYSTEM for SQL Server Database Engine. Click Next.
22. On the Database Engine Configuration screen, select Mixed Mode.
23. Enter a password for the system administrator account.
24. Click Add Current user. This may take several seconds.
25. Click Next.
26. On the Error and usage reporting screen, click Next.
27. On the Installation rules screen, click Next.
28. On the Ready to Install screen, click Next.
29. After installation completes, click Start, All Programs, Microsoft SQL Server 2008, Configuration Tools, SQL Server Configuration Manager.
30. Expand the Network Configuration node, and select the relevant instance.
31. In the right pane, right-click TCP/IP, and choose Enable.
32. In the left pane, click SQL Server Services.
33. In the right pane, right-click the relevant instance, and select Restart.
34. Repeat steps 9 through 33 for the second, third, and fourth SQL Server instances (on each instance install, choose Database Engine Services and Full-Text Search on the feature selection screen, and choose named instance on the instance configuration screen).
35. After all installations complete, open SQL Server Configuration Manager, right-click the SQL Browser service, change the startup type to Automatic, and start the SQL Browser service.
36. Click Start, Administrative Tools, and Local Security Policy.
37. Double-click Perform Volume Maintenance Tasks, and add the System account.
38. Double-click Lock Pages in Memory, and add the System account.
39. Click Start, All Programs, Microsoft SQL Server 2008, and SQL Server Management Studio.
40. In the Object Explorer pane, choose Connect→Database Engine. Choose the default instance.

41. Right-click the instance, and choose properties.
42. Click Processors, and check the Boost SQL Server Priority checkbox. Click OK.
43. Create a SQL Server login for the ds2user (see the DVD Store setup section below for the specific script to use).
44. Repeat steps 39 through 43 on the additional SQL Server instances.
45. Place the DVD Store backup file you prepared in the default location for SQL Server: C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\Backup.

Installing SQL Server 2008 SP1

Service Pack 1 is available for download from

<http://www.microsoft.com/downloads/details.aspx?FamilyID=66ab3dbb-bf3e-4f46-9559-ccc6a4f9dc19&displaylang=en>.

1. Run SQLServer2008SP1-KB968369-x64-ENU.exe.
2. On the Welcome screen, click Next
3. On the License terms screen, accept the license terms, and click Next.
4. On the Features screen, select available features, and click Next.
5. On the Check files in use screen, click Next.
6. On the Update screen, click Next.
7. When the update completes, click Next, and click Close.

DVD Store setup

Data generation overview

We built the database schema using the scripts in the DS2 distribution package, though we made a few minor modifications. The DS2 stress tool provides options to generate 10MB, 1GB, or 100GB datasets. To get the tool to generate the 10 GB of user data we used in this test, we had to make a few straightforward changes to the source code and to the DVD Store application's scripts. Note: We created our test data on a Linux system to take advantage of the larger RAND MAX.

Editing the ds2_create_orders.c module

The module ds2_create_orders.c defines constants that define the maximum values for the customer ID and the product ID. The constants for the 10GB database size did not exist. We added the constants for this size.

On the command line for the ds2_create_orders.c module, we specified the size. The available options were S (small), M (medium), and L (large). We added the case T for the 10GB 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 10GB database size.

We recompiled the ds2_create_orders.c module on Linux, following the instructions in the header comments. We used the following command line:

```
gcc -o ds2_create_orders ds2_create_orders.c -lm
```

Editing 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. On the command line for the ds2_create_cust.c module, we specified the size. The available options were S (small), M (medium), and L (large). We added the case T for the 10GB 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 10GB database size.

We recompiled the ds2_create_cust.c module on Linux, following the instructions in the header comments. We used the following command line:

```
gcc -o ds2_create_cust ds2_create_cust.c -lm
```

Generating the data for the 10GB database

We used shell scripts to run all four of the executables that generate the data. The distribution did not include shell scripts for the 10GB size. We wrote shell scripts based on the ds2_create_cust_large.sh and ds2_create_orders_large.sh scripts. The ds2_create_prod and ds2_create_inv executables did not ship with associated shell scripts, so we created shell scripts using the instructions in the readme files. We ran the shell scripts in the following order to generate the data for the 10GB database:

1. ds2_create_orders_10gb.sh

2. ds2_create_inv_10gb.sh
3. ds2_create_prod_10gb.sh
4. ds2_create_cust_10gb.sh

We waited until the processes finished before we moved onto the next step.

Creating the database

We modified the database creation SQL Server scripts in the DVD Store distribution package to build the database schema, which includes the file structure, tables, indices, stored procedures, triggers, and so on. We built a master copy of the 10GB database version for SQL Server 2005 and for SQL Server 2008, and then used those master copies to restore our test database to the HP ProLiant DL380 G6 server and Dell PowerEdge R710 server between each test run. We stored the backup file on the C: drive of each server for quick access.

We followed these steps to create the database:

1. We created the database and file structure using database creation scripts in the DS2 download. We made size modifications specific to our 10GB database and the appropriate changes to drive letters.
2. We created database tables, stored procedures, and objects.
3. We set the database recovery model to bulk-logged to prevent excess logging.
4. We loaded the data we generated into the database. For data loading, we used the import wizard in SQL Server Management Studio. Where necessary, we retained options from the original scripts, such as Enable Identity Insert.
5. We created indices, full-text catalogs, primary keys, and foreign keys using the database-creation scripts.
6. We updated statistics on each table according to database-creation scripts, which sample 18 percent of the table data.
7. On each SQL Server instance, we created a ds2user SQL Server login using the following Transact SQL (TSQL) script:

```
USE [master]
GO
CREATE LOGIN [ds2user] WITH PASSWORD=N'',
DEFAULT_DATABASE=[master],
DEFAULT_LANGUAGE=[us_english],
CHECK_EXPIRATION=OFF,
CHECK_POLICY=OFF
GO
```

8. We set the database recovery model back to full.

We made the following changes in the build scripts:

- Because we varied the size of the datasets, we sized the files in our scripts to reflect the database size and the number of files per filegroup. We allowed for approximately 40 percent free space in our database files to ensure that filegrowth activity did not occur during the testing.
- We followed Microsoft's recommendation of having 0.25 to 1 file per filegroup per logical processor. We used two files per filegroup on the HP ProLiant DL380 G6 server, and we used eight files per filegroup on the Dell PowerEdge R710 server.
- We did not use the DBCC PINTABLE command for the CATEGORIES and PRODUCTS tables, both because Microsoft recommends against this practice and because the commands do nothing in SQL Server 2005 and SQL Server 2008.
- In the case of SQL Server 2008, we added the FORCESEEK hint in the BROWSE_BY_ACTOR stored procedure because in initial testing we found that SQL Server 2008 was choosing an index scan instead of the preferred index seek, adding unnecessary overhead. This problem was not evident in SQL Server 2005.
- We created a SQL Server login we called ds2user and mapped a database user to this login. We made each such user a member of the db_owner fixed database role.
- Using the DVD Store scripts as a reference, we created the full-text catalog and index on the PRODUCTS table manually in SQL Server Management Studio.

We then performed a full backup of the database. This backup allowed us to restore the databases to a pristine state relatively quickly between tests.

Editing the workload script - ds2xdriver.cs module

To use the 10GB database we created earlier, we had to change the following constants:

- In the routine Controller(), we changed the string sizes. We added the T option for the 10GB 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.
- We added a command line parameter for the database name:

```
-database_name
```

Editing the workload script - ds2sqlserverfns.cs module

We changed the connection string to increase the number of available connections, to not use the default administrator ("sa") account, and to include a parameter for the database name. We raised the available connections limit from the default of 100 to 200 to allow room for experimentation. We created a user account we called ds2User and used that account.

The ds2connect routine in the ds2sqlserverfns.cs module defines sConnectionString. We used the following string; the changes we made appear in bold:

```
string sConnectionString = "User ID=ds2User;Initial Catalog=+dbname+";Max  
Pool Size=200;Connection Timeout=120;Data Source=" + Controller.target;
```

Recompiling the ds2sqlserverdriver.exe executable

We recompiled the ds2xdriver.cs and ds2sqlserverfns.cs module in Windows by following the instructions in the header comments. Because the DS2 instructions were for compiling from the command line, we used the following steps:

1. Open a command prompt.
2. Use the cd command to change to the directory containing our sources.
3. Run the batch file C:\Program Files\Microsoft Visual Studio 9.0\Common7\Tools\vsvars32.bat. This sets up the environment variables for us.
4. Execute the following command:

```
csc /out:ds2sqlserverdriver.exe ds2xdriver.cs ds2sqlserverfns.cs  
/d:USE_WIN32_TIMER /d:GEN_PERF_CTRS
```

Conducting the test

To perform the test, we used a series of batch files. We stored batch files either under the C:\ServerShare folder on each server or under the C:\ClientShare folder on the clients, and we used the PsExec utility to coordinate the test procedures amongst the client machines, server machines, and controller. We used simple file sharing to copy files from machine to machine, as needed.

The testing procedure consisted of the following steps:

1. Execute batch files to clean up prior outputs on client machines, the server machine, and the controller.
2. Drop the database on each SQL Server instance.
3. Restore the database on each SQL Server instance.
4. Pause briefly to wait for background tasks to complete before the scripts reboot the server.
5. The scripts reboot the client machines and server machine, and wait for a ping response from all test machines.
6. Wait 10 additional minutes for any background tasks to complete.
7. Measure idle power for 2 minutes.
8. Start the counter logs on client machines, and start the data collector set on the server under test.
9. Start the workload connections.

10. Start the workload ramp-up period.
11. Start the workload on the server, and run it for 30 minutes.
12. Stop the workload.
13. Copy all output files to the controller.

We obtained performance test results by running the workload three times under each specific load generation configuration, and by taking the median result as representative of the sample.

Measuring power

To record each server's power consumption during each test, we used an Extech Instruments (www.extech.com) 380803 Power Analyzer/Datalogger. We connected the power cord from the server under test to the Power Analyzer's output load power outlet. We then plugged the power cord from the Power Analyzer's input voltage connection into a power outlet.

We used the Power Analyzer's Data Acquisition Software (version 2.11) to capture all recordings. We installed the software on a separate Intel processor-based PC, which we connected to the Power Analyzer via an RS-232 cable. We captured power consumption at one-second intervals.

To gauge the idle power usage, we recorded the power usage for 2 minutes while each system was running the operating system, but otherwise idle, meaning the system was not running any test workload. To ensure the system was idle, we allowed the system to sit for 10 minutes after booting to the desktop before starting the 2 minutes of idle power capture.

We then recorded the power usage (in watts) for each system during the testing at 1-second intervals. To compute the average power usage, we averaged the power usage during the time the system was producing its peak performance results. We call this time the power measurement interval. See Figures 7 and 8 for the median one- and two-database-instance results of these power measurements.

Appendix A – Test server information

Figure 13 presents detailed information for the test servers we used in this report.

| Servers | Dell PowerEdge R710 | HP ProLiant DL380 G6 |
|---|---|---|
| General dimension information | | |
| Height (inches) | 3.50 | 3.39 |
| Width (inches) | 17.50 | 17.53 |
| Depth (inches) | 27.00 | 25.81 |
| U size in server rack (U) | 2 | 2 |
| Power supplies | | |
| Total number | 2 | 2 |
| Brand and model | Dell A570P-00 | HP HSTNS-PL14 |
| Wattage (W) | 570 | 460 |
| Cooling fans | | |
| Total number | 5 | 6 |
| Dimensions (h x w) | 2.5" x 2.5" | 2 3/8 x 2.5 |
| Voltage (V) | 12 | 12 |
| Amps (A) | 1.60 | 2.45 |
| General processor setup | | |
| Number of processor packages | 2 | 2 |
| Number of cores per processor package | 4 | 4 |
| Number of hardware threads per core | 2 | 2 |
| System power management policy | Balanced | Balanced |
| CPU | | |
| Vendor | Intel | Intel |
| Name | Xeon E5540 | Xeon E5540 |
| Stepping | D0 | D0 |
| Socket type | LGA1366 | LGA1366 |
| Core frequency (GHz) | 2.53 | 2.53 |
| L1 cache | 4 x 32 KB + 32 KB | 4 x 32 KB + 32 KB |
| L2 cache | 4 x 256 KB | 4 x 256 KB |
| L3 cache (MB) | 8 | 8 |
| Platform | | |
| Vendor and model number | Dell PowerEdge R710 | HP ProLiant DL380 G6 |
| Motherboard model number | 0N047H | PADAB0G9VXC1CQ |
| Motherboard revision number | 13 | 13 |
| BIOS name and version | Dell 1.2.6 (07/17/2009) | HP BIOS (10/01/2009) |
| BIOS settings | Hardware Prefetch disabled, Node Interleaving disabled | Hardware Prefetch disabled, Node Interleaving disabled |
| Memory modules | | |
| Total RAM in system (GB) | 48 | 48 |
| Vendor and model number | Samsung M393B1K70BH1 | Samsung M393B1K70BH1 |
| Type | PC3-8500R DDR3 | PC3-8500R DDR3 |
| Speed (MHz) | 1,066 | 1,066 |
| Speed in the system currently running @ (MHz) | 1,066 | 1,066 |

| Servers | Dell PowerEdge R710 | HP ProLiant DL380 G6 |
|---------------------------------------|--|---|
| Timing/latency (tCL-tRCD-iRP-tRASmin) | 7-7-7-20 | 7-7-7-20 |
| Size (GB) | 48 | 48 |
| Number of RAM modules | 6 x 8 GB | 6 x 8 GB |
| Chip organization | Double-sided | Double-sided |
| Hard disk | | |
| Vendor and model number | Seagate ST973451SS | 2 x Fujitsu DH072BAAKN 6 x Seagate DH072BB978 |
| Number of disks in system | 8 | 8 |
| Size (GB) | 73 | 73 |
| Buffer size (MB) | 16 | 16 |
| RPM | 15,000 | 15,000 |
| Type | SAS | SAS |
| Controller | Dell PERC 6/i Integrated RAID Controller, 256MB cache and BBWC | HP 410i Integrated Smart Array Controller, 256MB cache and BBWC |
| Operating system | | |
| Name | Windows Server 2008 R2 Enterprise | Windows Server 2008 R2 Enterprise |
| Build number | 7600 | 7600 |
| File system | NTFS | NTFS |
| Language | English | English |
| Network card/subsystem | | |
| Vendor and model number | Broadcom NetXtreme II gigE 5709C Dual-Port Ethernet | Broadcom NetXtreme II gigE 5709C Dual-Port Ethernet |
| Type | Integrated | Integrated |
| Optical drive | | |
| Vendor and model number | TEAC DV28SV DVD-ROM | LG GDR-D20N DVD-ROM |
| USB ports | | |
| Number | 4 | 4 |
| Type | 2.0 | 2.0 |

Figure 13: Detailed system configuration information for the two test servers.

About Principled Technologies

We provide industry-leading technology assessment and fact-based marketing services. We bring to every assignment extensive experience with and expertise in all aspects of technology testing and analysis, from researching new technologies, to developing new methodologies, to testing with existing and new tools.

When the assessment is complete, we know how to present the results to a broad range of target audiences. We provide our clients with the materials they need, from market-focused data to use in their own collateral to custom sales aids, such as test reports, performance assessments, and white papers. Every document reflects the results of our trusted independent analysis.

We provide customized services that focus on our clients' individual requirements. Whether the technology involves hardware, software, Web sites, or services, we offer the experience, expertise, and tools to help you assess how it will fare against its competition, its performance, whether it's ready to go to market, and its quality and reliability.

Our founders, Mark L. Van Name and Bill Catchings, have worked together in technology assessment for over 20 years. As journalists, they published over a thousand articles on a wide array of technology subjects. They created and led the Ziff-Davis Benchmark Operation, which developed such industry-standard benchmarks as Ziff Davis Media's Winstone and WebBench. They founded and led eTesting Labs, and after the acquisition of that company by Lionbridge Technologies were the head and CTO of VeriTest.



Principled Technologies, Inc.
1007 Slater Rd., 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.