

## OLTP performance test comparing Dell PowerVault MD3000i iSCSI SAN Array and HP StorageWorks 2000i Modular Smart Array

### Executive summary

Dell Inc. (Dell) commissioned Principled Technologies (PT) to compare performance for online transaction processing (OLTP) of two entry-level storage area network (SAN) solutions, which we tested with two shelves of disks and list in alphabetical order:

- Dell PowerVault MD3000i SAN Array with Dell PowerVault MD1000 expansion enclosure (Dell PowerVault iSCSI SAN)
- HP StorageWorks 2000i Modular Smart Array with HP StorageWorks 2000sa Modular Smart Array expansion enclosure (HP StorageWorks iSCSI SAN)

We tested both SAN solutions with Internet Small Computer Systems Interface (iSCSI) protocol. The storage arrays each included two full shelves of disks, with 15 disks per shelf on the Dell PowerVault iSCSI SAN and 12 disks per shelf on the HP StorageWorks iSCSI SAN.

For this comparison, we selected the DVD Store Version 2 (DS2) test tool. DS2 is an open-source simulation of an online e-commerce DVD store. We ran DS2 from two servers with two quad-core Intel Xeon E5405 processors at 2.0 GHz and 16 GB of RAM. We ran a single instance of DS2 and tested with two 250GB database loaded on the SAN solutions under test. The main DS2 metric is orders per minute, or OPM. We also used Windows Performance Monitor to measure disk transfers per second throughput. We report OPM and throughput results from a 120-second period of steady activity and heavy load during the DS2 test run.

In these OLTP tests, the Dell PowerVault iSCSI SAN delivered higher and better performance than the HP StorageWorks iSCSI SAN.

### KEY FINDINGS

- The Dell PowerVault iSCSI SAN delivered higher performance on our OLTP workload tests. It delivered 11 percent higher OPM per disk: 847.5 compared to the HP StorageWorks iSCSI SAN's 761.9 OPM per disk. (See Figure 1.)
- The Dell PowerVault iSCSI SAN delivered 48 percent greater throughput: 1.60 MB per second per disk compared to the HP StorageWorks iSCSI SAN's 1.08 MB per second per disk. (See Figure 2.)

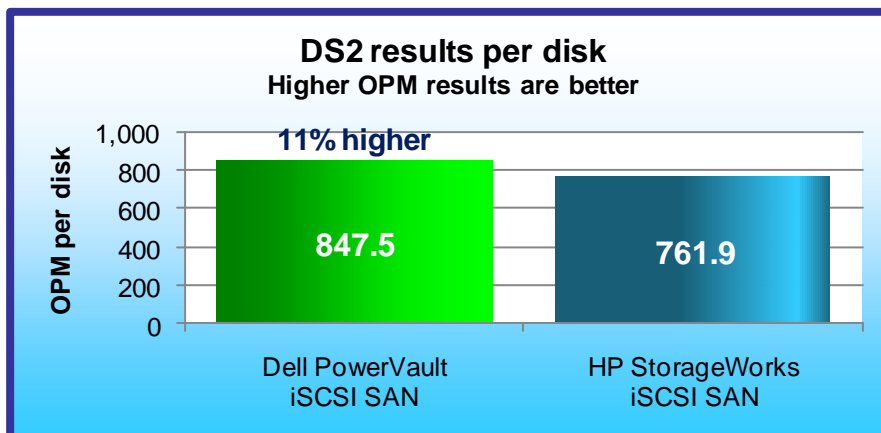


Figure 1: DS2 performance results in OPM per disk for the two SAN solutions. Results are the OPM divided by the number of disks: 30 for the Dell PowerVault iSCSI SAN and 24 for the HP StorageWorks iSCSI SAN. Higher numbers are better.

To provide comparable results for the different disk counts of the storage solutions, we divided the median OPM results of three DS2 runs for each solution by the number of disks in the solution. We show these results in Figure 1. The disk count includes all disks in both shelves.

The Dell PowerVault iSCSI SAN provided 11 percent better DS2 performance than the HP StorageWorks iSCSI SAN: 847.5 OPM per disk for the Dell PowerVault iSCSI SAN vs. 761.9 OPM per disk for the HP StorageWorks iSCSI SAN.

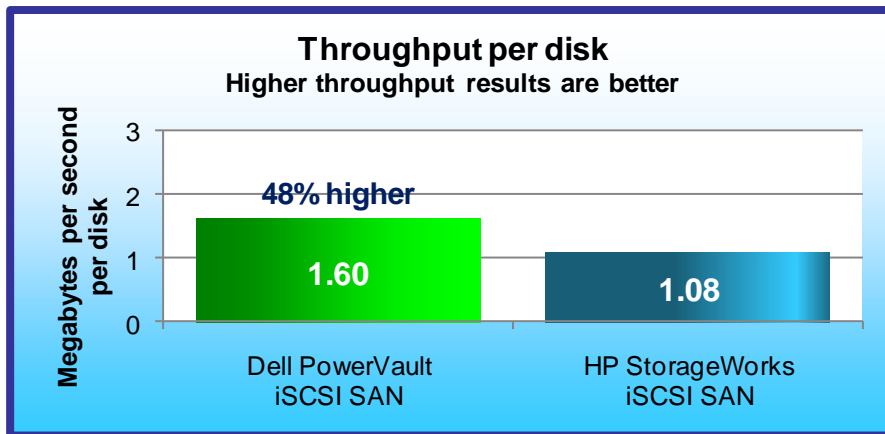


Figure 2 shows the average throughput in average disk transfers (in megabytes per second per disk) that the Windows Performance Monitor logged during the same DS2 runs that produced the results in Figure 1. The Dell PowerVault iSCSI SAN delivered 48 percent higher throughput, at 1.60 MB per second per disk. The HP StorageWorks iSCSI SAN delivered 1.08 MB per second per disk.

**Figure 2: Average disk transfers (in megabytes per second per disk) for the storage arrays during the DS2 test. Results are megabytes per second for the storage array divided by the number of disks; 30 disks for the Dell PowerVault iSCSI SAN and 24 for the HP StorageWorks iSCSI SAN. Higher numbers are better.**

## Test results

We report results for two performance counters: OPM and Disk transfers MB/sec. For both, we present the average results for 120 seconds of steady activity and heavy load during a DS2 test run; higher numbers are better for both performance measures. Per-disk results divide the OPM and Disk transfers MB per second results by the number of disks: 30 for the Dell PowerVault iSCSI SAN and 24 for the HP StorageWorks iSCSI SAN.

We ran the test three times on each SAN and report results from the run that produced the median of the three OPM results for the SAN.

Figure 3 provides test results for the two storage solutions.

	Dell PowerVault iSCSI SAN	HP StorageWorks iSCSI SAN
Number disks tested	30	24
OPM	25,426	18,287
OPM per disk	847.5	761.9
Disk transfers MB per second	48.0	25.9
Disk transfers MB per second per disk	1.60	1.08

**Figure 3: Test results for the two storage solutions...**

## Workload

We conducted our testing using DVD Store Version 2, 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.

Because our goal was to isolate and test database server storage, we did not use the Web application layer. Instead, we ran the driver application directly via its command-line interface.

DS2 models an online 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 full-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.

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

Each server ran a single instance of DS2, which spawned 32 threads. This simulated a heavily loaded environment; the load-generating system ran with no think time, blasting requests as quickly as the servers could handle them.

The DS2 driver application creates an orders-per-minute performance counter on the system. We used the Reliability and Performance Monitor on the servers to create a data collector set to collect statistics once every second. Our experiments showed that this function did not affect the performance of the test. We used these results to verify that neither the processor nor the memory was a bottleneck in the test.

## Test environment

We configured a test bed for each storage array. (See Figure 4.)

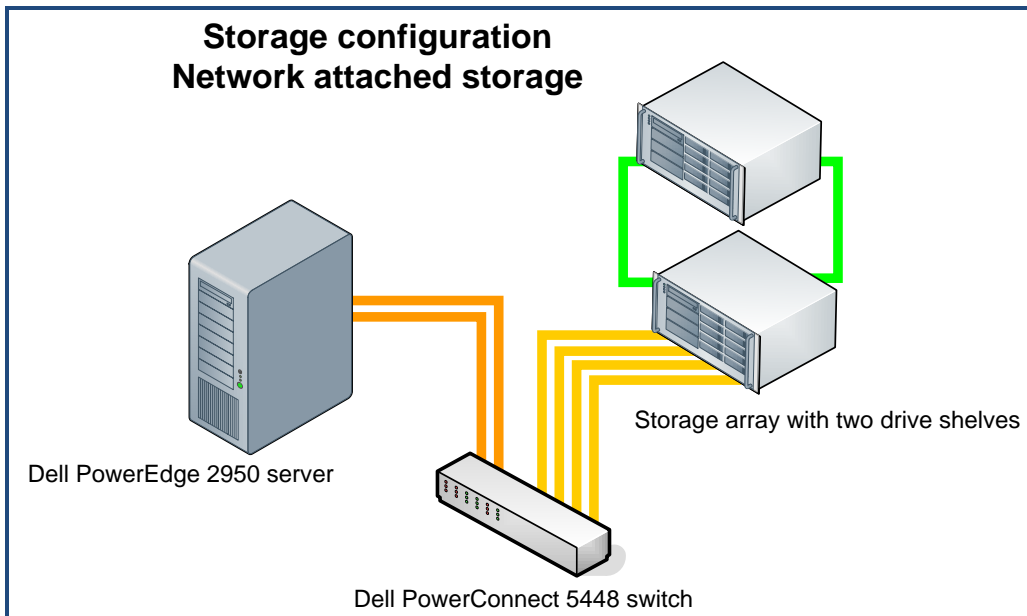


Figure 4: Storage test bed configuration.

Each test bed included the following components:

- Two Dell PowerEdge 2950 servers (see Figure 5). We attached each server to the switch via two discrete NICs. We ran DB2 from the servers.
- One Dell PowerConnect 5448 switch
- SAN array under test:
  - Dell PowerVault MD3000i SAN Array with Dell PowerVault MD1000 expansion enclosure
  - HP StorageWorks 2000i Modular Smart Array with HP StorageWorks 2000sa Modular Smart Array expansion shelf

Component	Details
Server	Dell PowerEdge 2950 server
Processors	Two quad-core Intel Xeon E5405 processors at 2.0 GHz
Memory	16 GB of FB-DDR2 PC5300 memory
Internal disk	One 80GB, 7,200 RPM Western Digital WD800AAJS-18SATA 3.0Gb drive One 73GB, 15,000 RPM Seagate Cheetah 15K.5 ST373455SS 3.0Gb drive
NIC	Two Intel PRO/1000PT NICs
OS	Windows Server 2003 R2 Enterprise x64 Edition SP2; SQL Server 2008
Test software	DVD Store Version 2 (DS2)

Figure 5: Configuration for each of the two Dell PowerEdge 2950 servers.

Appendix B provides more detailed information on the test environment.

### Storage arrays under test

The storage arrays we tested each included two full shelves of disks. We configured the storage arrays with disks of similar speed and size. Disks were 146 GB and 15K RPM. We tested both arrays with iSCSI connections. PT tested both storage arrays using all disks to hold the database files and an internal drive to hold the log files.

Figure 6 identifies the systems we tested, their disk counts, the amount of useable storage, and the RAID technology they use.

Appendix C provides additional details about the tested storage arrays. Appendix D provides details about the storage layout.



	Dell PowerVault iSCSI SAN	HP StorageWorks iSCSI SAN
		
Arrays	Dell PowerVault MD3000i SAN Array with Dell PowerVault MD1000 expansion enclosure	HP StorageWorks 2000i Modular Smart Array with HP StorageWorks 2000sa Modular Smart Array expansion shelf
Disks	30 146GB 15K RPM SAS disks total for the two shelves	24 146GB 15K RPM SAS disks total for the two shelves
Formatted storage capacity	2,042 GB	1,640 GB
Connection	iSCSI	iSCSI
RAID technology	RAID 10	RAID 10

Figure 6: The SANs we tested.

## Test procedures

We followed the procedures in this section to set up and conduct the tests.

### Setting up the storage disks

#### *Dell PowerVault iSCSI SAN*

1. Run the PowerVault storage manager.
2. Select the Configure tab.
3. Click Create Virtual Disk.
4. Select Unconfigured capacity.
5. Click Next.
6. Select Manual.
7. Click Next.
8. Select RAID 10 as the RAID level.
9. Select 7 drives from each enclosure.
10. Click Add.
11. Click Calculate Capacity.
12. Click Next.
13. Type 475 as the capacity of the first virtual disk.
14. Type SQL-DB1 as the name.
15. Click Next.
16. Select Map Now and map the drive to test server #1.
17. Click Finish.
18. Click Create Virtual Disk.
19. Select Free Capacity.
20. Expand the menu next to Disk Group 1.
21. Click Free Capacity.
22. Click Next.
23. Type 475 as the capacity for the virtual disk.
24. Type SQL-DB2 as the name.
25. Click Next.
26. Select Map Now and map the drive to test server #2.
27. Click Finish.
28. Click Create Virtual Disk.
29. Select Unconfigured capacity.
30. Click Next.
31. Select Manual.
32. Click Next.
33. Select RAID 10 as the RAID level.
34. Select 8 drives from each enclosure.
35. Click Add.
36. Click Calculate Capacity.
37. Click Next.
38. Type 540 as the capacity of the first virtual disk.
39. Type SQL-DB3 as the name.
40. Click Next.
41. Select Map Now and map the drive to test server #1.
42. Click Finish.
43. Click Create Virtual Disk.
44. Select Free Capacity.
45. Expand the menu next to Disk Group 2.
46. Click Free Capacity.
47. Click Next.
48. Type 540 as the capacity for the virtual disk.
49. Type SQL-DB4 as the name.

50. Click Next.
51. Select Map Now and map the drive to test server #2.
52. Click Finish.
53. Perform the rest of the steps on both test servers:
54. Run Microsoft iSCSI Initiator:
  - a. Select the Discovery tab.
  - b. Click Add.
  - c. Enter the IP Address for the array.
  - d. Select the Targets tab.
  - e. Click the Refresh button.
  - f. Select the first Inactive Target listed.
  - g. Click Log On.
  - h. Check the Automatically restore this connection when the system boots checkbox.
  - i. Check the Enable multi-path check box.
  - j. Click OK.
  - k. Repeat steps f through j for the second Inactive Target listed.
55. Open a command prompt.
56. Type `cd c:\windows\system32.`
57. Type `diskpart.exe.`
58. Type `List Disk` to determine the name of the RAID array.
59. Type `Select Disk #` where # is the disk number.
60. Type `Create partition primary align=64.`
61. Repeat steps 59 and 60 for both drives.
62. Type `Exit.`
63. In Windows, click Start, right-click My Computer, and select Manage.
64. Click Disk Management.
65. Right-click each of the new partitions, and select Format.
66. Name the disks `SQL-DB1` and `SQL-DB2.`
67. Mount the database volumes:
  - a. Right-click the 475GB drive, and select Change Drive Letter and paths.
  - b. Click Add.
  - c. Select drive E as the drive letter.
  - d. Click OK.
  - e. Click OK.
  - f. Right-click the 540GB drive, and select Change Drive Letter and paths.
  - g. Click Add.
  - h. Select drive F as the drive letter.
  - i. Click OK.
  - j. Click OK.

#### *HP StorageWorks iSCSI SAN*

1. Log into the management interface.
2. Click Manage.
3. Click Virtual Disk Config.
4. Click Create a Vdisk.
5. Select Manual Virtual Disk Creation.
6. Type `vDisk1` as the virtual disk name.
7. Select RAID 10 as the RAID type.
8. Click Create New Virtual Disk.
9. Select 6 drives on one enclosure and 6 drives on the other enclosure.
10. Select No for the dedicated spare option.
11. Click Continue.
12. Set the number of volumes to 2.
13. Click OK.

14. Type 419,840 for the size of both volumes, and name them SQL-DB1 and SQL-DB2.
15. Click Create Virtual Disk.
16. Click Create a Vdisk.
17. Select Manual Virtual Disk Creation.
18. Type vDisk2 as the virtual disk name.
19. Select RAID 10 as the RAID type.
20. Click Create New Virtual Disk.
21. Select 6 drives on one enclosure and 6 drives on the other enclosure.
22. Select No for the dedicated spare option.
23. Click Continue.
24. Set the number of volumes to 2.
25. Click OK.
26. Type 419,840 for the size of both volumes, and name them SQL-DB3 and SQL-DB4.
27. Click Create Virtual Disk.
28. Run Microsoft iSCSI Initiator on both test servers:
  - a. Select the Discovery tab.
  - b. Click Add.
  - c. Enter the IP Address for the array.
  - d. Select the Targets tab.
  - e. Click the Refresh button.
  - f. Select the first Inactive Target listed.
  - g. Click Log On.
  - h. Check the Automatically restore this connection when the system boots checkbox.
  - i. Check the Enable multi-path check box.
  - j. Click OK.
  - k. Repeat steps f through j once for the second Inactive Target.
29. Open a command prompt.
30. Type `cd c:\windows\system32.`
31. Type `diskpart.exe.`
32. Type `List Disk` to determine the name of the RAID array.
33. Type `Select Disk #` where # is the disk number.
34. Type `Create partition primary align=64.`
35. Type `Exit.`
36. Repeat steps 29 through 35 for the first and third drives on test server 1 and the second and fourth drives on test server 2.
37. In Windows, click Start, right-click My Computer, and select Manage.
38. Click Disk Management.
39. Right-click the two new partitions, and select Format.
40. Name the disks SQL-DB1 and SQL-DB2.
41. Mount the database volumes:
  - a. Right-click the first formatted 410GB drive, and select Change Drive Letter and paths.
  - b. Click Add.
  - c. Select drive E as the drive letter.
  - d. Click OK.
  - e. Click OK.
  - f. Right-click the second formatted 410GB drive, and select Change Drive Letter and paths.
  - g. Click Add.
  - h. Select drive F as the drive letter.
  - i. Click OK.
  - j. Click OK.
42. Repeat steps 37 through 41 on both test servers.

### **Installing Windows Server 2003 R2 Enterprise x64 Edition Service Pack 2 on the server**

We began our testing by installing a fresh copy of Windows Server 2003 R2 Enterprise x64 Edition Service Pack 2 on the server. We followed this process for each installation:

1. Assign a computer name of `Server`.
2. For the licensing mode, use the default setting of five concurrent connections.
3. Enter a password for the administration logon.
4. Select Eastern Time Zone.
5. Use typical settings for the Network installation.
6. Assign a name of `Testbed` for the workgroup.

Note: We used the default BIOS settings on the server.

### **Installing and setting up SQL Server 2008 on the test servers**

1. Enter the SQL Server 2008 installation CD.
2. Click OK.
3. Accept the license agreement.
4. Click Install.
5. Click Exit.
6. Click Next.
7. Select I Agree.
8. Click Next.
9. Click Continue.
10. Click Finish.
11. Click OK.
12. Restart the Server.
13. Click My Computer.
14. Double-click the CD Drive.
15. Click Installation.
16. Click New SQL Server Stand-Alone Installation or Add Features to an Existing Installation.
17. Click OK.
18. Select Specify a Free Edition.
19. Click Next.
20. Accept the license terms.
21. Click Next.
22. Click Install.
23. Click Next.
24. Select the following options:
  - Database Engine Services
  - Full-Text Search
  - Client Tools Connectivity
  - Client Tools Backwards Compatibility
  - Management Tools Basic
  - Management Tools Complete
25. Click Next.
26. Click Next.
27. Click Next.
28. Change the SQL Server Agent Account Name to `NT AUTHORITY\SYSTEM`.
29. Change the SQL Server Database Account Name to `NT AUTHORITY\SYSTEM`.
30. Click Next.
31. Select Mixed Mode.
32. Enter a password for the administrator "sa" account.
33. Confirm your password.
34. Click Add Current User.
35. Click Next.
36. Click Next.



37. Click Next.
38. Click Install.
39. Click Next.
40. Click Close.
41. Right-click My Computer, and select Manage.
42. Click Local Users and Groups.
43. Click Users.
44. Right-click, and select New User.
45. Type `SQLFTUser` for the username.
46. Type `password` for the password, and confirm password.
47. Uncheck the User Must Change Password at Next Logon Checkbox.
48. Check the Password Never Expires box.
49. Click Create.
50. Repeat steps 44 through 49 for the username `SQLLocalUser`.
51. Run the SQL Configuration Manager.
52. Click SQL Server Services.
53. Right-click SQL Server, and select Properties.
54. Select Log On as This Account.
55. Type `SQLLocalUser` for the account name.
56. Type `password` for the password, and confirm password.
57. Click OK.
58. Click Yes.
59. Right-click SQL Full-text Filter, and select Properties.
60. Select the Service tab.
61. Change the Start Mode to automatic.
62. Click Apply.
63. Select Log On as This Account.
64. Type `SQLFTUser` for the account name.
65. Type `password` for the password, and confirm password.
66. Click Apply.
67. Click Start.
68. Click OK.

### Installing and setting up DVD Store on the test servers

1. Copy the `ds2` folder containing the DVD Store executables to `C:\ds2`.
2. Double-click the folder.
3. Create a file named `DS2_1.bat` that contains the following DVD Store run command line:  

```
c:\ds2run\ds2sqlserverdriver --target=localhost --ramp_rate=10 --run_time=20
--n_threads=32 --db_size_str=W --think_time=0 --database_name=DS2_1
```
4. Run `DS2_1.bat` for 10 seconds to create the DVD Store testing counters.
5. Click Start→Run.
6. Type `perfmon`, and press Enter.
7. Expand Performance Logs and Alerts.
8. Click Counter Logs.
9. Right-click the right pane, and select New Log Settings.
10. Type `DVD Store` as the name.
11. Click OK.
12. Click Add Counters.
13. Add the following counters: `Memory\Available Mbytes`, `Physical Disk(0 C:)\% Idle Time`, `Physical Disk(0 C:)\Disk Read Bytes/sec`, `Physical Disk(0 C:)\Disk Reads/sec`, `Physical Disk(0 C:)\Disk Write Bytes/sec`, `Physical Disk(0 C:)\Disk Writes/sec`, `Physical Disk(1 E:)\% Idle Time`, `Physical Disk(1 E:)\Disk Read Bytes/sec`, `Physical Disk(1 E:)\Disk Reads/sec`, `Physical Disk(1 E:)\Disk Write Bytes/sec`, `Physical Disk(1 E:)\Disk Writes/sec`, `Physical Disk(1 E:)\Disk`

```
Transfers/sec, Processor(0-23)\% Processor Time, SQLServer:Buffer  
Manager\Buffer cache hit ratio, SQLServer:Buffer Manager\Checkpoint  
pages/sec, Test\MaxRT, Test\OPM
```

14. Click Close.
15. Change the Sample Interval to 1.
16. Select the Log Files tab.
17. Change the Log file type to Text File (Comma delimited).
18. Select the Schedule tab.
19. Change the Start log to start manually.
20. Click OK.
21. Click Yes.
22. Open My Computer.
23. Double-click Drive E.
24. Create a new folder named `SQLData`.
25. Right-click that folder, and select Properties.
26. Select the Security tab.
27. Click Add.
28. In the Object Names box, type `SQLLocalUser`.
29. Click OK.
30. For full access, check the box beside Allow.

### Generating the datasets

We built the database schema using the scripts included 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 250GB 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.

#### *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 250GB 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 W for the 250GB 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 250GB 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 W for the 250GB 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 250GB 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 20GB 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 250GB 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 250GB database:

- `ds2_create_orders_20gb.sh`

- ds2\_create\_inv\_20gb.sh
- ds2\_create\_prod\_20gb.sh
- ds2\_create\_cust\_20gb.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 included 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 250GB database version for SQL Server 2008, and then used that master copy to restore our test database to the test server between each test run.

We followed these steps to create the database:

1. We created the database and file structure using database creation scripts provided with DS2. We made size modifications specific to our 250GB 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 generated data. 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. We created ds2user SQL Server login and user for testing.
8. We set the database recovery model back to full.

We made the following several 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 core and we used eight files per filegroup on our 24-core 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 2008.
- In SQL 2008, we added the FORCESEEK query hint to the BROWSE\_BY\_ACTOR stored procedure, to force SQL Server 2008 to use an index seek, instead of an index scan, in its query execution plan. We made this change because our initial tests showed that SQL Server was using a highly inefficient index scan. Therefore, we created the SQL Server 2008 BROWSE\_BY\_ACTOR procedure as follows:

```
CREATE PROCEDURE BROWSE_BY_ACTOR
(
    @batch_size_in INT,
    @actor_in VARCHAR(50)
)
AS

SET ROWCOUNT @batch_size_in
SELECT * FROM PRODUCTS

--added to force index seek
```

```

WITH (FORCESEEK)

WHERE CONTAINS(ACTOR, @actor_in)
SET ROWCOUNT 0
GO

```

- We created a SQL Server login called `ds2user` and a database user mapped 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 server to a pristine state relatively quickly between tests.

### Editing the DVD Store scripts

We had to make a few minor modifications to the DVD Store test scripts. We detail these modifications below.

#### *Editing the ds2xdriver.cs module*

To use the 250GB 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.
- We added a command line parameter for the database name: `-database_name`

#### *Editing the 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 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 on 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

```

### Creating a script to delete and recreate the DB2 database

We created the following script, `DS2_Drop_and_Restore.sql`, to delete and recreate the DB2 database between each run:

```
USE [master]
GO
```

```
/****** Object: Database [test] Script Date: 10/16/2008 12:27:59 *****/
IF EXISTS (SELECT name FROM sys.databases WHERE name = N'DS2_1')
DROP DATABASE [DS2_1]
GO
```

```
RESTORE DATABASE [DS2_1] FROM DISK = N'B:\SQLBackup\250GB_DVDStore.bak' WITH FILE
= 1,
MOVE N'primary' TO N'E:\SQLData\DS2_1.mdf', MOVE N'cust1' TO
N'E:\SQLData\DS2_1_1.ndf',
MOVE N'cust2' TO N'F:\SQLData\DS2_1_2.ndf', MOVE N'cust3' TO
N'E:\SQLData\DS2_1_3.ndf',
MOVE N'cust4' TO N'F:\SQLData\DS2_1_4.ndf', MOVE N'cust5' TO
N'E:\SQLData\DS2_1_5.ndf',
MOVE N'cust6' TO N'F:\SQLData\DS2_1_6.ndf', MOVE N'cust7' TO
N'E:\SQLData\DS2_1_7.ndf',
MOVE N'cust8' TO N'F:\SQLData\DS2_1_8.ndf', MOVE N'ind1' TO
N'E:\SQLData\DS2_1_9.ndf',
MOVE N'ind2' TO N'F:\SQLData\DS2_1_10.ndf', MOVE N'ind3' TO
N'E:\SQLData\DS2_1_11.ndf',
MOVE N'ind4' TO N'F:\SQLData\DS2_1_12.ndf', MOVE N'ind5' TO
N'E:\SQLData\DS2_1_13.ndf',
MOVE N'ind6' TO N'F:\SQLData\DS2_1_14.ndf', MOVE N'ind7' TO
N'E:\SQLData\DS2_1_15.ndf',
MOVE N'ind8' TO N'F:\SQLData\DS2_1_16.ndf', MOVE N'ds_misc1' TO
N'E:\SQLData\DS2_1_17.ndf',
MOVE N'ds_misc2' TO N'F:\SQLData\DS2_1_18.ndf', MOVE N'ds_misc3' TO
N'E:\SQLData\DS2_1_19.ndf',
MOVE N'ds_misc4' TO N'F:\SQLData\DS2_1_20.ndf', MOVE N'ds_misc5' TO
N'E:\SQLData\DS2_1_21.ndf',
MOVE N'ds_misc6' TO N'F:\SQLData\DS2_1_22.ndf', MOVE N'ds_misc7' TO
N'E:\SQLData\DS2_1_23.ndf',
MOVE N'ds_misc8' TO N'F:\SQLData\DS2_1_24.ndf', MOVE N'orders1' TO
N'E:\SQLData\DS2_1_25.ndf',
MOVE N'orders2' TO N'F:\SQLData\DS2_1_26.ndf', MOVE N'orders3' TO
N'E:\SQLData\DS2_1_27.ndf',
MOVE N'orders4' TO N'F:\SQLData\DS2_1_28.ndf', MOVE N'orders5' TO
N'E:\SQLData\DS2_1_29.ndf',
MOVE N'orders6' TO N'F:\SQLData\DS2_1_30.ndf', MOVE N'orders7' TO
N'E:\SQLData\DS2_1_31.ndf',
MOVE N'orders8' TO N'F:\SQLData\DS2_1_32.ndf', NOUNLOAD, STATS = 10
GO
```

```
USE [DS2_1]
GO
CREATE USER [ds2user] FOR LOGIN [ds2user]
GO
USE [DS2_1]
GO
EXEC sp_addrolemember N'db_owner', N'ds2user'
GO
```

## Running the DS2 test

Before starting each DS2 test, we deleted and recreated the DB2 database. We rebooted the servers and allowed them to sit idle for at least 10 minutes to ensure that they finished with all aspects of the boot process. We then ran the DS2 tests three times. This section describes this procedure:

1. Run the “DS2\_Drop\_and \_Restore.sql” script on the test servers to restore the database to its original state.
2. Click Connect.
3. Press F5.
4. Restart the servers once the database finishes restoring.
5. Wait 10 minutes.
6. Wait 1 minute and 50 seconds.
7. Begin the perfmon DVD Store Counter on the test servers.
8. Wait 10 seconds.
9. Run DS2\_1.bat on the test servers to begin the DVD Store benchmark.
10. Stop perfmon on the test servers once DVD Store completes its test.
11. Repeat steps 1 through 10 two more times, for a total of three runs.

## Reporting results

Windows Performance Monitor data collector sets on the servers collected the OPM statistics once every second during the tests. We averaged the OPM values for the period from 480 seconds to 600 seconds into the test for each of the three test runs and summed those values from the two servers. This time is a period of steady activity and heavy load, and it suffers from neither ramp-up nor ramp-down effects. We report the median of the three results as the OPM result for the configuration. We calculated the average Disk transfers MB per second values from the data collector sets on the two servers for the same period and report the sum of those results from the same run that produced the median OPM result.

## Appendix A: Test results

Figure 7 gives the test results from each server and totals for the two servers for the three test runs, shows the median of those totals, the per-disk average of that median, the throughput results from the same run that provided the median OPM total.

	Dell PowerVault iSCSI SAN	HP StorageWorks iSCSI SAN
Number disks tested	30	24
<b><i>OPM results from three test runs</i></b>		
Run 1 Server 1	9,161.15	9,366.33
Run 1 Server 2	15,986.29	8,920.19
<b>Run 1 total</b>	25,147.44	18,286.52
Run 2 Server 1	10,291.17	9,059.00
Run 2 Server 2	15,134.68	8,714.64
<b>Run 2 total</b>	25,425.85	17,773.64
Run 3 Server 1	10,238.07	9,088.83
Run 3 Server 2	15,503.56	9,579.14
<b>Run 3 total</b>	25,741.63	18,667.97
<b><i>Median OPM results</i></b>		
OPMs (higher is better)	25,426	18,287
OPMs per disk (higher is better)	847.5	761.9
<b>Throughput</b>		
Average disk transfers MB/sec	48.0	25.9
Disk transfers MB/sec per disk (higher is better)	1.60	1.08

Figure 7: Test results.

## Appendix B: Test environment

We used two SQL Server 2008 database servers to generate the workload and create demand on the storage. Figure 8 provides detailed configuration information for these servers.

Server	Dell PowerEdge 2950
<b>General</b>	
Number of processor packages	2
Number of cores per processor package	4
Number of hardware threads per core	1
System Power Management Policy	Always On
<b>CPU</b>	
Vendor	Intel
Name	Xeon E5405
Stepping	6
Socket type	Socket 771-LGA
Core frequency (GHz)	2.0
Front-side bus frequency (MHz)	1,333
L1 cache	32 KB + 32 KB (per core)
L2 cache	12 MB (2 x 6 MB shared)
<b>Platform</b>	
Vendor	Dell Inc.
Motherboard model number	0M332H
Motherboard chipset	Intel 5000X
Motherboard revision number	12
Motherboard serial number	JD6HWG1
BIOS name and version	Dell 2.3.1 (4/29/2008)
BIOS settings	Default
Chipset INF driver	Intel 7.3.0.1010
<b>Memory module(s)</b>	
Vendor and model number	Samsung M395T5750EZ4-CE66
Type	FB-DDR2 PC5300
Speed (MHz)	667
Speed in the system currently running @ (MHz)	667
Timing/Latency (tCL-tRCD-iRP-tRASmin)	5-5-5-15
Size (MB)	16,384
Number of RAM modules (MB)	8 x 2,048
Chip organization	Double-sided
Channel	Dual
<b>Hard disk</b>	
Number of disks in system	2
Vendor and model number	Western Digital WD800AAJS-18



<b>Server</b>	<b>Dell PowerEdge 2950</b>
Size (GB)	80
Buffer size (MB)	8
RPM	7,200
Type	SATA 3.0 Gbps
Vendor and model number	Seagate Cheetah 15K.5 ST373455SS
Size (GB)	73
Buffer size (MB)	8
RPM	15,000
Type	SAS 3.0 Gbps
Controller	Intel 6321ESB
Controller driver	Intel 7.3.0.1010 (11/18/2005)
<b>Operating system</b>	
Name	Windows Server 2003 R2 Enterprise x64 Edition
Build number	3790
Service Pack	SP2
Microsoft Windows update date	8/26/2008
File system	NTFS
Language	English
Microsoft DirectX version	9.0c
<b>Graphics</b>	
Vendor and model number	ATI ES1000
Chipset	ATI ES1000
BIOS version	BK-ATI VER008.005.028.000
Type	Integrated
Memory size (MB)	16
Resolution	1,024 x 768 x 32-bit
Driver	ATI 8.24.3.0 (4/5/2006)
<b>Network card/subsystem</b>	
Vendor and model number	Intel PRO/1000 PT Server Adapter
Type	Discrete
Driver	9.7.34.0 (2/1/2007)
<b>Optical drive</b>	
Vendor and model number	TEAC CD-224E-N
Type	CD/DVD-ROM
Interface	SATA
Dual/single layer	Dual
<b>USB ports</b>	
Number of ports	4
Type of ports (USB 1.1, USB 2.0)	2.0

**Figure 8: Detailed system configuration information for the test server.**

## Dell PowerConnect 5448 switch

We used one Dell PowerConnect 5448 switch. (See Figure 9.)

Component	Details
Switch	One Dell PowerConnect 5448 switch
Frame size	Jumbo frames enabled on all connected devices. Size set to 9,014 bytes
Flow control	Flow control set to Generate & Respond on all network interface cards
iSCSI optimization	iSCSI optimization disabled
Receive descriptors	Set to 256 on all network interface cards

Figure 9: Dell PowerConnect 5448 switch configuration.

## Individual test beds

The remainder of this appendix shows diagrams of the components in each of the test beds and indicates their connections.

### Dell PowerVault iSCSI SAN

The Dell PowerVault MD3000i has two controllers with two 1Gbps iSCSI ports per controller, one 100Mbps Ethernet management port per controller, and one SAS 4x port per controller. We connected the Dell MD1000 expansion shelf to the controllers in the Dell PowerVault MD3000i with one SAS 4x cable per controller. We connected the iSCSI switch to two iSCSI ports per controller and one management port per controller. Figure 10 shows the wiring diagram for the Dell PowerVault MD3000i.

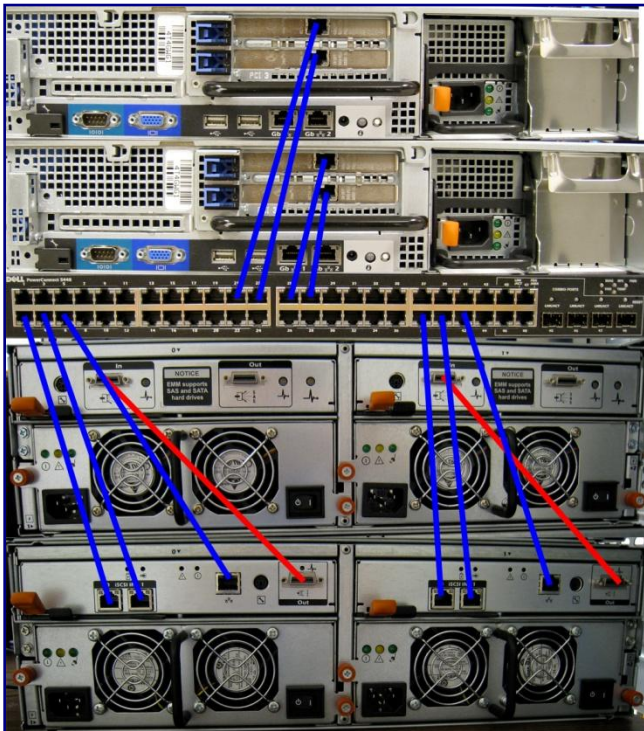


Figure 10: Dell PowerVault iSCSI SAN wiring diagram for the servers, switch, and storage. Blue lines represent iSCSI connections between the servers and the switch and the switch and the array. Red lines represent SAS 4x connections to different parts of the array.

### HP StorageWorks iSCSI SAN

The HP StorageWorks iSCSI SAN has two controllers with two 1 Gbps iSCSI ports per controller, one 100Mbps Ethernet management ports per controller, and one SAS 4x port per controller. We used one SAS 4x cable per controller to connect the HP StorageWorks 2000i Modular Smart Array to the controllers in the HP StorageWorks 2000sa Modular Smart Array expansion shelf. We connected two iSCSI ports per controller and one management port per controller to the iSCSI switch. Figure 11 shows the wiring diagram for the HP StorageWorks iSCSI SAN.

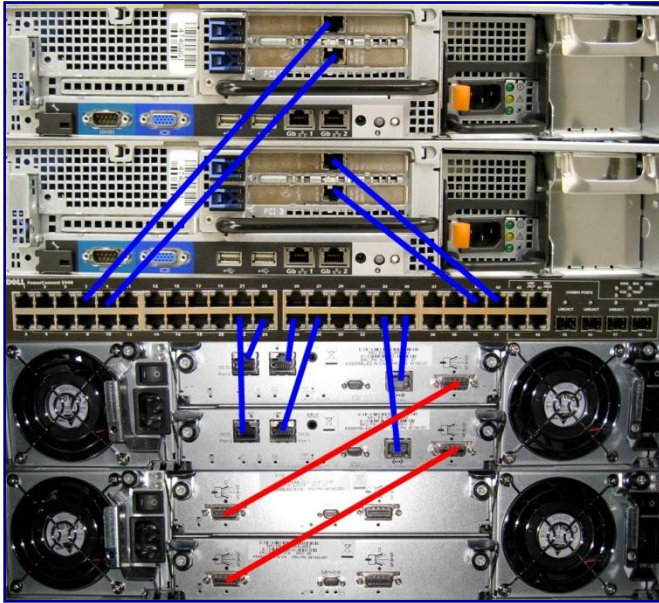


Figure 11: HP StorageWorks iSCSI SAN wiring diagram. Blue lines represent iSCSI connections between the servers and the switch and the switch and the array. Red lines represent SAS 4x connections to different parts of the array.

## Appendix C: Storage configuration

Figure 12 describes the storage disks.

System	Dell PowerVault iSCSI SAN	HP StorageWorks iSCSI SAN
Storage connectivity (Fibre Channel, SAS, SATA, iSCSI)	iSCSI	iSCSI
Storage model and OS/firmware revision	PowerVault MD3000i Firmware 06.70.17.60	HP Modular Storage Array 2000i Firmware J210P03
Storage cache	512 MB	2 GB (1GB per controller)
Number of storage controllers	2	2
Number of storage ports	4 (2 per controller)	4 (2 per controller)
Maximum bandwidth of storage connectivity to host	1 Gb x 4	1 Gb x 4
Switch type/model/firmware revision	Dell PowerConnect 5448 v1.0.0.31	Dell PowerConnect 5448 v1.0.0.31
NIC model and firmware	Intel PRO1000PT NIC with Jumbo Frames enabled	Intel PRO1000PT NIC with Jumbo Frames enabled
Number of NICs/host	2	2
Total number of disks tested in solution	30	24

Figure 12: Primary storage hardware.

Figure 13 shows the primary storage software.

System	Dell PowerVault iSCSI SAN	HP StorageWorks iSCSI SAN
NIC driver	9.7.34.0	9.7.34.0
Multi-pathing	Microsoft iSCSI Software Initiator Version 2.07 integrated MPIO MD Storage Manager 02.71.G6.07	Microsoft iSCSI Software Initiator Version 2.07 integrated MPIO
Host OS	Windows Server 2003 R2 Enterprise x64 Edition SP2	Windows Server 2003 R2 Enterprise x64 Edition SP2
ESE.dll file version	08.01.0240.005	08.01.0240.005

Figure 13: Primary storage software.

Figure 14 shows the primary storage disk configuration.

System	Dell PowerVault iSCSI SAN	HP StorageWorks iSCSI SAN
Disk type, speed, and firmware revision	15x Seagate SAS 15k, ST3146356SS, HS09; 15x Hitachi SAS 15k, HUS153014VLS300, A4C2	24x Seagate SAS 15k, ST3146855SS
Raw capacity per disk (GB)	146	146
Number of physical disks in test	30	24
Total raw storage capacity (GB)	4,380	3,504
RAID level	RAID 10	RAID 10
Total formatted capacity (GB)	2,042	1,640

Figure 14: Primary storage disk configuration.

## Appendix D: Storage layout

Figures 15 and 16 show the disk layout for each of the storage solutions we tested.

Dell PowerVault iSCSI SAN				
<b>Enclosure 1</b>				
Drive 0	Drive 3	Drive 6	Drive 9	Drive 12
Drive 1	Drive 4	Drive 7	Drive 10	Drive 13
Drive 2	Drive 5	Drive 8	Drive 11	Drive 14
<b>Enclosure 2</b>				
Drive 0	Drive 3	Drive 6	Drive 9	Drive 12
Drive 1	Drive 4	Drive 7	Drive 10	Drive 13
Drive 2	Drive 5	Drive 8	Drive 11	Drive 14
<b>Disk Group 1 (Controller 0)</b>		<b>Disk Group 2 (Controller 1)</b>		
SQL-DB1 (Server #1)		SQL-DB1 (Server #2)		
SQL-DB2 (Server #2)		SQL-DB2 (Server #1)		

Figure 15: Dell PowerVault iSCSI SAN disk layout.

HP StorageWorks iSCSI SAN			
<b>Enclosure 1</b>			
Drive 0	Drive 3	Drive 6	Drive 9
Drive 1	Drive 4	Drive 7	Drive 10
Drive 2	Drive 5	Drive 8	Drive 11
<b>Enclosure 2</b>			
Drive 0	Drive 3	Drive 6	Drive 9
Drive 1	Drive 4	Drive 7	Drive 10
Drive 2	Drive 5	Drive 8	Drive 11
<b>vDisk 1 (Controller A)</b>		<b>vDisk 2 (Controller B)</b>	
SQL-DB1 (Server #1)		SQL-DB1 (Server #2)	
SQL-DB2 (Server #2)		SQL-DB2 (Server #1)	

Figure 16: HP StorageWorks iSCSI SAN disk layout.

## 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 Road, Suite 250  
Durham, NC 27703  
[www.principledtechnologies.com](http://www.principledtechnologies.com)  
[info@principledtechnologies.com](mailto:info@principledtechnologies.com)

Principled Technologies is a registered trademark of Principled Technologies, Inc.  
Intel, Xeon, and Pentium are registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.\*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.