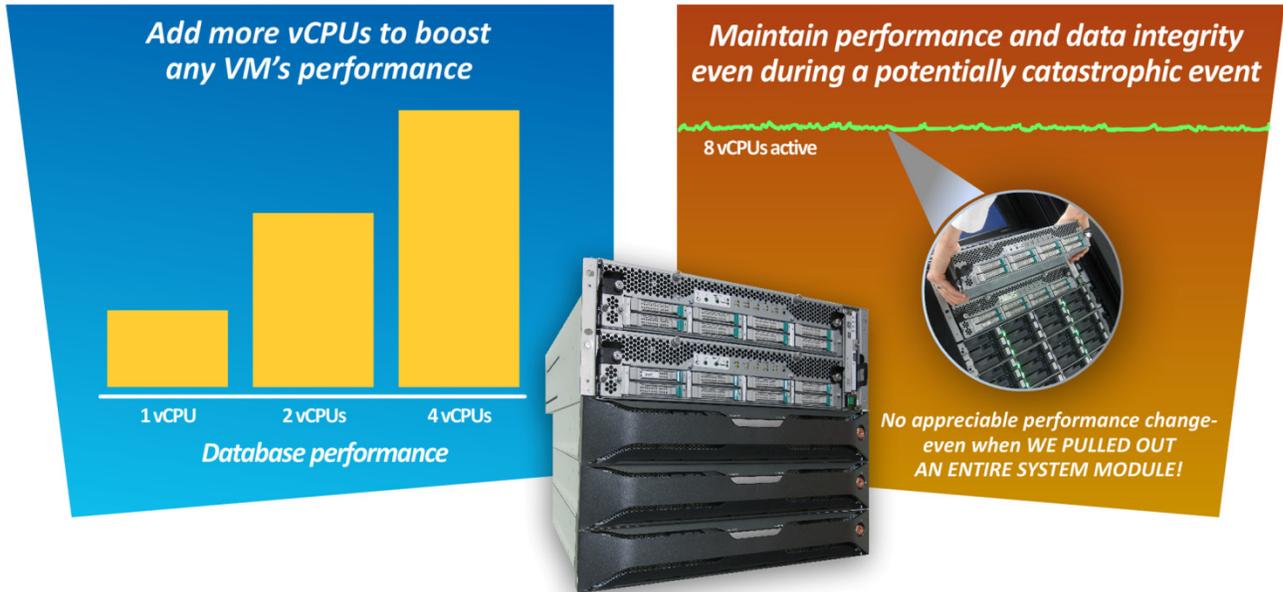


FAULT TOLERANT SERVER: PERFORMANCE SCALABILITY AND DATA LOSS PREVENTION

A Fault-Tolerant Server solution that lets you...



OUR FINDINGS

Both performance and ultra-high availability are crucial in many virtualized environments. In Principled Technologies' tests in our labs, the Stratus ftServer 6300/NEC Express5800/R320a solution provided out-of-the-box, scalable database performance and fault tolerance to all guest virtual machines (VMs) by supporting multiple virtual CPUs (vCPUs) per VM. It also continued to operate—without data loss and with nearly unchanged database performance on guest VMs—in the face of a potentially catastrophic event more extreme than is likely to ever occur in a real system: we caused one of the lock-stepped CPU/IO enclosures to fail by removing it from the chassis. *Note: The NEC/Stratus fault tolerance solution is hardware-based and thus all guest VMs are immediately fault tolerant. We performed these tests without the use of either the VMware Fault Tolerance or VMware High Availability products.*

OUR PROCESS

We measured database performance in a single VM with different numbers of vCPUs to see how well the Fault Tolerant Server's performance could scale within a single VM. We also pushed the server to a very high work level (8 vCPUs in one VM) and removed an entire CPU/IO module. We used Microsoft SQL Server 2008 and the open-source DVD Store Version 2 benchmark (DS2) as our example database workload. DS2 takes orders and performs other functions representative of a real-world database application.



PROJECT OVERVIEW

Stratus and NEC have worked together for the past 10 years to bring Fault Tolerance to the Intel® Server Architecture (<http://www.stratus.com/news/2005/20051124.htm>). Both firms have commissioned this test to assess the Fault-Tolerant Server’s performance scalability within a VMware VM and its data loss prevention in the face of a potentially catastrophic event.

Our first goal was to gauge the type of scaling the NEC/Stratus solution achieves in a single VM as one adds virtual CPUs (vCPUs). This scaling is important for database applications that require additional processing power beyond the amount a single vCPU can provide. We ran the DVD Store Version 2 benchmark inside a VM, using VMware ESX 4.0 as the hypervisor. We ran Microsoft® Windows Server® 2008 R2 with Microsoft SQL Server® inside the VM. (For more information on DS2, see the Workload section.) To determine how performance scaled as we added cores, we tested with one, two, and four virtual processors assigned to the VM with a single database.

Our second goal was to gauge the performance level and data integrity of a database application on the Fault Tolerant Server in the face of a potentially catastrophic event. In our example test, we used a single heavily loaded 8vCPU VM – as many vCPUs as the server had physical processors cores. (Note, the Fault Tolerant Server has redundant sets of components, including processors, so there are 16 physical processor cores, but with redundancy, only eight of them are available.) We began the database workload, and then we actually pulled one of the lock-stepped CPU/I/O enclosures from the chassis. An enclosure includes both a CPU and I/O element, so removing an enclosure removes half the system’s processor, memory, and I/O capabilities, simulating an event far more catastrophic than one that is likely to occur in a real-world environment.

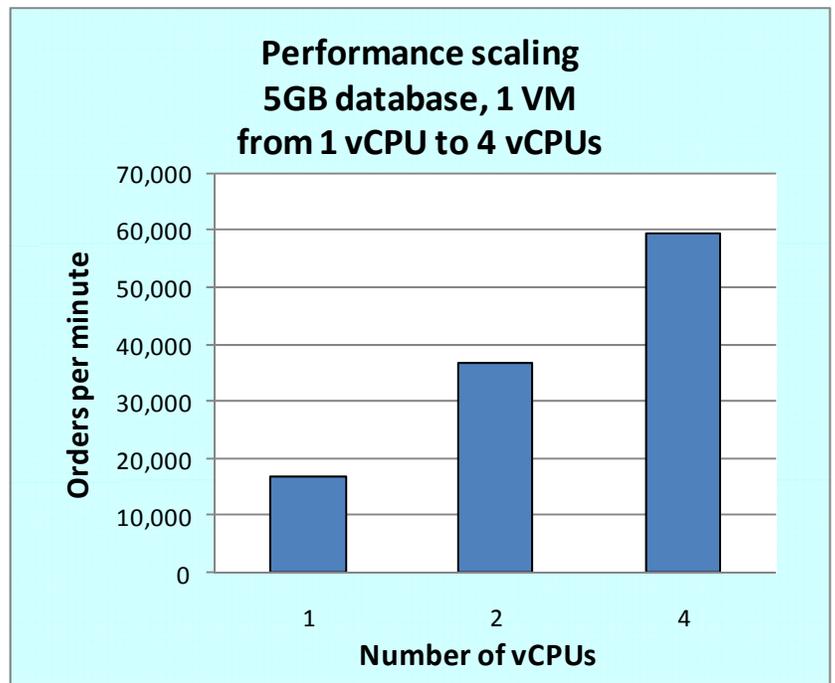


Figure 1: Number of DVD Store Version 2 orders per minute as the number of vCPUs increased from 1 to 4. Higher numbers are better.

WHAT WE FOUND

Scaling

As Figure 1 shows, database performance scaled linearly as the number of vCPUs increased, and memory remained flat at 48GB. This shows that adding virtual processors increased system performance inside the VM with our workload. Figure 2 shows the number of orders per minute and the performance increase, relative to the single-vCPU configuration, for the three configurations we tested. On this and several of our preliminary test runs, the performance scaling from one to two vCPUs was slightly better than linear. This behavior may be due to SQL Server optimizations for multiprocessor architectures.

Number of vCPUs	OPM (higher numbers are better)	OPM increase relative to 1 vCPU (higher numbers are better)
1	16,771	0.0%
2	36,833	119.6%
4	59,409	254.2%

Figure 2: Performance and percentage increase from the median run of DVD Store Version 2 for the three configurations we tested.

Continued performance and no data loss during a potentially catastrophic event

As we mention in the Project overview, we induced a potentially catastrophic event beyond that which any real-world server environment is likely to experience: We actually pulled a lock-stepped CPU/IO enclosure from the Fault Tolerant Server. We noted the performance before we removed the enclosure. We then observed a slight drop in performance that lasted 12 seconds. After this 12-second period, performance stabilized and began to increase gradually. After 26 seconds, performance returned to the pre-event level. As Figure 3 shows, the lowest performance during this period was 76,010 OPM, which is only 0.16 percent lower than the pre-event level of 76,129 OPM. The server lost no data at all.

Potentially catastrophic event, 5GB database, 1 VM, 8 vCPU	
OPM the server achieved before the potentially catastrophic event occurred	76,129
Lowest OPM the server achieved after the potentially catastrophic event occurred	76,010
Percentage performance drop due to the potentially catastrophic event	0.16%
Seconds from the potentially catastrophic event to the time performance drop ended and performance stabilized	12
Seconds from the time performance stabilized to the time performance returned to pre-event level	26

Figure 3: DVD Store Version 2 performance results for a single VM with 8 vCPU during a potentially catastrophic event. Higher numbers of orders per minute are better.

WORKLOAD

DVD Store Version 2 is an accessible, open-source, flexible framework with which we can model a typical online transaction processing (OLTP) database. DS2, which simulates an online e-commerce DVD store, has database components and Web server components, and includes driver programs that put heavy loads on the server. We opted to use the included driver program to stress the database layer directly, and bypassed the Web interface.

When DS2 is executing, simulated customers log in; browse movies by actor, title, or category; and purchase movies. Each DS2 order consists of a customer login, a number of searches for movies, and a purchase. Each search is by title, actor, or category. The title and actor searches use full-text search. The other customer actions, including adding new customers, exercise a wide range of database functions.

The main DS2 metric is orders per minute (OPM), which the driver program calculates and reports via the Windows Performance Monitor utility on the client machines. While the DVD Store client application outputs OPM at 10-second intervals visually, we chose to collect this OPM metric by extracting the data from the application to a log file at 1-second intervals. We ran this workload on the virtual machines for 30 minutes and we report the last OPM score the benchmark reported.

We ran one Microsoft SQL Server 2008 Service Pack 1 instance on a single virtual machine with a 5GB database. We used the default DS2 parameters and setup configuration, with the exceptions we note in the DVD Store setup section in the How we tested section of this report.

The client machine ran a single instance of DS2, with 32 threads, which simulate users executing actions within the application. This simulated a heavily loaded environment; the load-generating client VMs

ran with no think time. For more details on the DS2 tool, see <http://www.delltechcenter.com/page/DVD+Store>.

Figure 4 shows the settings we used on the test server's virtual machines. Refer to the How we tested section below for detailed steps of how we set up the VMs.

VM specifications	VM settings
vCPU	1
Memory (GB)	48
Virtual NIC type	E1000
Number of virtual disks	3 (OS, SQL logs, SQL database)

Figure 4: VM configuration we used for testing.

We configured the two internal hard drives as a RAID 1 and installed vSphere on them. We installed Microsoft Windows Server 2008 R2 and SQL Server 2008 SP1 installations inside a single virtual machine. We stored the virtual machine on the NEC D3-10 SAN, which we connected to the server via two Emulex® LightPulse® LPe1150 single-port host bus adapters (HBAs). Appendix A presents detailed system information for the test server.

HOW WE TESTED

Setting up and configuring the NEC D3-10

We used a NEC D3-10 Fibre channel SAN storage array for this testing. The NEC D3-10 has 3 disk-shelves with 12 disks in each shelf. We used two Emulex LightPulse LPe1150 single-port HBAs in the server for testing. We connected both ports of the two single-port HBAs to the NEC D3-10 SAN.

We created three storage pools, one 4-disk pool for the VM operating systems, and two 16-disk pools for the SQL Server database and logs. We configured all pools using a RAID 10 configuration. We created three logical disks, one on each storage pool. We used one SQL Server database for testing, but distributed it across the two database logical disks. We put the SQL Server logs on the first database logical disks.

Configuring the test server

We installed a fresh copy of VMware vSphere 4 on the test server.

Installing VMware ESX 4 on the host¹

1. Ensure that the top enclosure (element '10') has the primary LED light on.
2. Power on the chassis.

¹ Source: <http://stratadoc.stratus.com/vmware/3.0.0/r004e-00/wwhelp/wwhimpl/js/html/wwhelp.htm?context=r004e-00&file=ch1r004e-00.html>

3. Insert the ftSSS DVD.
4. On the ftSys for vSphere welcome screen, click Next.
5. Accept the Stratus License agreement, and click Next.
6. To configure Internal Disk mirror menu, select at least one pair of disks to mirror both in respective slot '0' (you will be warned that all data on these disks will be lost). Click Next.
7. When the application prompts you, remove the ftSSS DVD, and insert the VMware vSphere 4 DVD.
8. To start ESX in graphical mode, press Enter.
9. On the Welcome screen, click Next.
10. Click the checkbox to accept the terms of the license agreement, and click Next.
11. Accept the default keyboard layout, and click Next.
12. On the Custom Drivers screen, choose No to install custom drivers, and click Next.
13. You will receive a warning if you chose No. Click Yes to proceed and begin loading drivers.
14. Click Next.
15. Enter your serial number now, or choose to enter one later.
16. Ensure only the left embedded Ethernet port (labeled port 1) in upper CPU-I/O enclosure has an active network connection. Click Next.
17. On the Network Configuration screen, choose the NIC that is marked with a green check mark. Click Next.
18. On the next Network Configuration screen, set your IP addressing information, subnet mask, DNS, and hostname. Click Next.
19. On the Setup Type screen, choose standard setup.
20. On the ESX Storage Device page, select the first device (mpm0). Click Next.
21. Select the virtual disk you wish to install ESX on, and click Next.
22. On the Data Loss warning screen, click OK.
23. On the Time Zone Settings screen, choose your appropriate time zone, and click Next.
24. On the Date and Time screen, modify the date and time as you wish, and click Next.
25. Assign a root password for ESX. Optionally, add additional users if you wish, and click Next.
26. On the Summary screen, click Next to install.
27. When the installation is complete, click Next.
28. Click Finish to reboot the system.

Setting up Stratus ftServer software ²

1. Insert the ftSSS DVD in the drive, and log in as root.
2. At the command prompt, type

```
# /opt/ft/sbin/ft-install
```
3. The installation program prompts you to answer configuration questions and accept EULAs. It then runs system checks and advises you of your system's current configuration. Unless you change the default system settings, you can press Enter at most prompts to accept the default selections.
4. When the application prompts you, specify a password for the ftadmin user.
5. When the application prompts you, eject the ftSSS DVD, and remove it from the DVD drive.
6. Press Enter at the final prompt to reboot your system.

² Source: <http://stratadoc.stratus.com/vmware/3.0.0/r004e-00/wwhelp/wwhimpl/js/html/wwhelp.htm?context=r004e-00&file=ch1r004e-00.html>

Enabling secure shell (ssh) access for the root user

1. Log in as root to the ESX console.
2. Type the following command to change to the appropriate directory:
`# cd /etc/ssh`
3. Edit the `sshd_config` file using `vi`. Use the following command:
`# vi sshd_config`
4. Press the down arrow key to move the cursor to the `PermitRootLogin` line, and move the cursor to the word `no`. Press the `i` key to insert text.
5. Delete the word `no`, and type the word `yes`.
6. Press the Escape key to return to command mode.
7. Type the following command to save and exit the file:
`:wq`
8. Type the following command to restart the ssh service:
`# service sshd restart`

Installing vSphere client on a management workstation

1. Configure your management workstation to be on the same subnet as your ESX server. In Internet Explorer, navigate to the IP address of your ESX Server.
2. Accept certificate warnings, and click Continue.
3. Click vSphere to download the vSphere client.
4. Choose Run to install.
5. Choose your language, and click OK.
6. On the Welcome screen, click Next.
7. Accept the license terms, and click Next.
8. Enter user and organization details, and click Next.
9. Optionally, click the checkbox to install the update utility. Click Next.
10. On the Destination Folder screen, click Next.
11. Click Install.
12. Click OK.

Adding the external LUNs as datastores to vSphere storage

1. In vSphere, click the Configuration tab.
2. In the Hardware pane, click Storage.
3. Click Add Storage...
4. Choose Disk/LUN, and click Next.
5. Select the appropriate LUN, and click Next.
6. Click Next.
7. Enter a name for the datastore, and click Next.
8. Click Next to accept the defaults for block size.
9. Click Finish to complete the addition of the LUN.
10. Repeat steps 3 through 9 on the remaining two LUNs that you will use for SQL Server data-A and SQL Server data-B virtual disks. (The SQL Server logs will be on SQL Server data-A virtual disk.)

Creating the virtual machine with vSphere

1. On the vSphere workstation, click Start→All Programs→VMware→VMware vSphere client.
2. Enter the IP address or hostname, user name, and password. Click Login.
3. Click the Virtual Machines tab.
4. Right-click, and choose New Virtual Machine.
5. Choose Custom, and click Next.
6. Assign a name to the virtual machine. Click Next.
7. Select a datastore for the virtual machine files, and click Next. (We chose the OS datastore.)
8. Choose Virtual Machine Version 7. Click Next.
9. Choose Microsoft Windows, then Microsoft Windows Server 2008 R2 (64-bit). Click Next.
10. Choose one virtual processor, and click Next.
11. Choose all available RAM, and click Next.
12. Click 1 for the number of NICs, and select E1000. Click Next.
13. Leave the default virtual storage controller. Click Next.
14. Choose to create a new virtual disk, and click Next.
15. Make the OS virtual disk size 100 GB, specify the OS datastore on the external storage, and click Next.
16. Keep the default virtual device node, and click Next.
17. Click Finish.
18. Right-click the VM, and choose Edit Settings.
19. On the Hardware tab, click Add...
20. Click Hard Disk, and click Next.
21. Click Create a new virtual disk for SQL Server data, and click Next.
22. Specify 200GB for the virtual disk size, and specify the datastore for SQL Server-A data. Note that we installed SQL Server logs on this virtual disk as well.
23. Choose SCSI(2:0) for the device node, and click Next.
24. On the Hardware tab, click Add...
25. Click Hard Disk, and click Next.
26. Click Create a new virtual disk for the SQL Server log, and click Next.
27. Specify 200GB for the virtual disk size, and specify the datastore for SQL Server-B data.
28. Choose SCSI(3:0) for the device node, and click Next.
29. Click SCSI Controller 1, and choose Change Type.
30. Choose LSI Logic SAS, and click OK.
31. Click SCSI Controller 2, and choose Change Type.
32. Choose LSI Logic SAS, and click OK.
33. Click Finish, and click OK.

Installing the guest operating system

1. Insert the installation DVD for Windows Server 2008 R2 x64 into the DVD drive.
2. In vSphere, right-click the virtual machine, and choose CD/DVD drive.
3. Click the Host Device radio button, and choose the appropriate drive.
4. Click OK.
5. Right-click the machine, and choose Power→Power On.
6. Right-click the machine, and choose Open console.
7. Choose the language, time and currency, and keyboard input. Click Next.

8. Click Install Now.
9. Choose Windows Server Enterprise (Full Installation). Click Next.
10. Accept the license terms, and click Next.
11. Click Custom.
12. Click the Disk, and click Drive options (advanced).
13. Click New, Apply, Format, and click Next.
14. Let the installation process continue. The VM will reboot several times.
15. After the installation completes, click OK to set the Administrator password.
16. Enter the administrator password twice, and click OK.
17. When the operating system finishes booting, choose VM, Guest, Install/Upgrade VMware Tools.
18. On the Install VMware Tools screen, click OK.
19. On the AutoPlay menu, click Run setup.EXE.
20. On the Welcome screen, click Next.
21. On the Setup Type screen, choose Typical, and click Next.
22. Click Install.
23. On the various Windows Security screens, click Install each time.
24. Click Finish to complete the installation.
25. Click Yes to reboot the VM.
26. After rebooting, log into the VM.
27. Click Start → Control Panel, and double-click System.
28. Click Change Settings.
29. Click Change.
30. Enter the new computer name, and click OK.
31. Click OK to restart, click Close, and click Restart Now.

Installing Windows updates

1. Click Start, and then select Windows Update on the Start menu.
2. Click View Advanced Options.
3. Click Never Check for Updates.
4. Click OK.
5. Click Check for Updates.
6. When the application prompts you to install an update for Windows Update, click Install Now.
7. Click View available updates to ensure you install the necessary updates.
8. Click Install.
9. Agree to the license terms, and click Finish.
10. Reboot the VM when the application requests you do so.
11. After reboot, check Windows Update again for further updates. Click Start, Windows Update.
12. Click Check for updates.
13. If Windows lists no important updates, close the updates window.
14. Power off the VM.

Installing SQL Server 2008 on the VM

1. Insert the SQL Server 2008 DVD into the physical DVD drive.
2. Right-click the VM, and choose Power → Power On.
3. Right-click the VM, and choose Open Console.

4. Click the Connect CD/DVD button. Choose CD/DVD Drive 1, Connect to host device...
5. On the AutoPlay menu, click Run SETUP.EXE.
6. If the application prompts you to install the .NET Framework, click OK.
7. Select the Accept radio button, and click Install.
8. When the installation finishes, click Exit.
9. If any warnings appear about compatibility, accept them, as installing SP1 in the next section should solve them.
10. If the application prompts you, click OK to install a hotfix for Windows (KB942288).
11. Upon completion, click Restart now to resart the server.
12. After rebooting, log into the server.
13. Click Start, and click Computer.
14. Double-click the CD/DVD drive.
15. Click Installation.
16. Click New SQL Server stand-alone installation.
17. On the Setup support rules screen, click OK.
18. Specify the Enterprise Evaluation free edition, and click Next.
19. Click the checkbox to accept the license terms, and click Next.
20. Click Install to install the setup support files.
21. You may see a Windows Firewall warning. For now, ignore this, and click Next.
22. 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.
23. On the Instance configuration screen, leave the default selection of default instance, and click Next.
24. On the Disk space requirements screen, click Next.
25. 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.
26. On the Database Engine Configuration screen, select Mixed Mode.
27. Enter a password for the system administrator (sa) account.
28. Click Add Current user. Note: This may take several seconds.
29. Click Next.
30. On the Error and usage-reporting screen, click Next.
31. On the Installation rules screen, click Next.
32. On the Ready to Install screen, click Next.
33. After installation completes, click Start→All Programs→Microsoft SQL Server 2008→Configuration Tools→SQL Server Configuration Manager.
34. Expand the Network Configuration node, and select the relevant instance.
35. In the right pane, right-click TCP/IP, and choose Enable.
36. In the left pane, click SQL Server Services.
37. In the right pane, right-click the SQL Server service, and select Restart.
38. Create a SQL Server login for the ds2user (see the DVD Store setup section below for the specific script to use).
39. 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.

Configuring additional drives in the VM in Disk Management

1. Power on the VM, and open the console.
2. Click Start, Administrative Tools, Computer Management.
3. Click Disk Management.
4. Right-click the uninitialized disks (if any), and choose Initialize Disk.
5. Choose MBR.
6. Right-click the volume meant for SQL Server-A data and SQL Server logs, and choose New Simple Volume. Click Next.
7. Accept the default volume size, and click Next.
8. Choose D: from the drop-down menu for the drive letter. Click Next.
9. Type `sqldata-A` for the volume name. Click Next.
10. Choose NTFS, 64K allocation unit size, and check the Perform a quick format checkbox. Click Next.
11. Right-click the volume meant for SQL Server-B data, and choose New Simple Volume. Click Next.
12. Accept the default volume size. Click Next.
13. Choose E: from the drop-down menu for the drive letter. Click Next.
14. Type `sqldata-B` for the volume name. Click Next.
15. Choose NTFS, 64K allocation unit size, and check the Perform a quick format checkbox. Click Next.
16. Click Finish.

Installing and configuring the database clients

For the DVD Store scripts, we used a client to simulate a number of users putting a load on the server. For our clients, we created a folder we called C:\ClientShare to store workload scripts, and shared this folder for access from our controller machine. We installed the .NET 3.5 framework on each client VM, as the DVD Store test executable requires at least .NET 2.0. We created a performance counter log on each client machine to track the number of OPM each virtual database server returned. We followed this process for each installation:

1. Install Microsoft Windows Server 2003 R2 Enterprise x86 Edition Service Pack 2 on the client.
2. Assign a computer name of `Clientx` for the database client, where `x` is the client number.

3. For the licensing mode, use the default setting of five concurrent connections.
4. Enter a password for the administrator logon.
5. Select Eastern Time Zone.
6. Use typical settings for the Network installation.
7. Type `Workgroup` for the workgroup.
8. Install Windows Updates, .NET 3.5 framework, and copy the DVD Store client executable into the `C:\ClientShare` folder.

Setting up DVD Store

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 5GB 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 5GB 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 V for the 5GB 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 5GB 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 V for the 5GB 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 5GB 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 5GB 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 5GB 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 5GB database:

1. ds2_create_orders_5gb.sh
2. ds2_create_inv_5gb.sh
3. ds2_create_prod_5gb.sh
4. ds2_create_cust_5gb.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 more. We built a master copy of the 5GB database version SQL Server 2008, and then used those master copies to restore our test database to the VM between each test run. We stored the backup file on the C: drive of each VM 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 5GB 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.
9. 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 VM.
 - 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.
 - 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. 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.
10. 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 5GB database we created earlier, we had to change the following constants:

- In the routine Controller(), we changed the string sizes. We added the V option for the 5GB 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
```

Testing procedure

To perform the test, we used a series of batch files. We stored batch files respective to each system on that system, either under the `C:\ServerShare` folder or the `C:\ClientShare` folder, and we used the PsExec, Plink, and WinSCP utilities to coordinate the test procedures amongst the client machine, server machine, and controller. We used simple file sharing, or secure FTP, to copy files from machine to machine as we needed.

The testing procedure consisted of the following steps:

1. Execute batch files and shell scripts to clean up prior outputs on client virtual machines, the server physical machine, and the controller.
2. Drop the database on each server VM.
3. Restore the database on each server VM.
4. Pause for 5 minutes to wait for background tasks to complete before server reboot.
5. Reboot the physical machines, and wait for a ping response from the physical machine, as well as from all VMs involved in testing.

6. Wait 10 additional minutes for any background tasks to complete.
7. Start ESXtop on the server under test in batch mode.
8. Start the workload connections.
9. Start the workload ramp-up period.
10. Start the workload.
11. Stop the workload.
12. Copy all output files to the controller.

Potentially catastrophic event testing procedure

To perform the potentially catastrophic event test, we followed the same steps as in the Testing procedure section above, with one additional step. Twenty minutes after we started the workload, we removed the primary enclosure from the Fault Tolerant Server chassis, and let the test complete. We then looked at the SQL Server OPM counter to see how the OPM performance suffered, and looked at the ESX Server percent ready time to see how long it took for the system to recover from the potentially catastrophic event.

APPENDIX A – SERVER CONFIGURATION INFORMATION

Figure 5 provides the complete detailed configuration information about the test server. Because the Fault Tolerant Server is a redundant server, the CPU, memory, and hard disk are redundant. This means that of the four processor packages, only two are available to the OS; the other two are for redundancy. The same is true for memory and hard drives.

Server	Stratus ftServer 6300/NEC Express5800/R320a
Power Supplies	
Total number	2
Vendor and model number	Astec DS850-3
Wattage of each (W)	850W
Cooling fans	
Total number	8
Vendor and model number	San Ace 80 9G0812P1G09
Dimensions (h x w) of each	3-1/4" x 1-1/2"
General processor setup	
Number of processor packages	8
Number of cores per processor package	4
Number of hardware threads per core	2
CPU	
Vendor	Intel®
Name	Xeon® X5570
Stepping	D0
Socket type	LGA 1366
Core frequency (GHz)	2.93
Bus frequency	6.4 GT/s
L1 cache	32 KB + 32 KB (per core)
L2 cache	256 KB (per core)
L3 cache (MB)	8
Thermal design power (TDP, in watts)	95
Platform	
Vendor and model number	ft Server 6300
Motherboard model number	P6300-S2
BIOS name and version	4.1:82
BIOS settings	Defaults
Memory	
Total RAM in system (GB)	96
Vendor and model number	Samsung m393b1k70BH1-CF8
Type	PC3-8500R

Server	Stratus ftServer 6300/NEC Express5800/R320a
Speed (MHz)	1,066
Speed running in the system (MHz)	1,066
Size (GB)	8
Number of RAM modules	12 x 8 GB
Chip organization	Double-sided
Rank	Dual
Hard disk	
Vendor and model number	Seagate Savvio ST973452SS
Number of disks in system	2
Size (GB)	73
Buffer size (MB)	16
RPM	15 K
Type	SAS
Controller	LSI 1068E
Raid configuration	RAID 10
Operating system	
Name	VMware ESX 4.0.0
Build number	208167
File system	VMFS
Language	English
Network card/subsystem	
Vendor and model number	Intel 82576 GB network adapter
Type	Integrated
Fiber adapter	
Vendor and model number	Emulex LPe1150 4GB HBA
Type	PCI Express
USB ports	
Number	1-front, 3 rear
Type	USB 2.0

Figure 5: Configuration information for the test server.

ABOUT PRINCIPLED TECHNOLOGIES



Principled Technologies, Inc.
1007 Slater Road, Suite 300
Durham, NC, 27703
www.principledtechnologies.com

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 our clients assess how it will fare against its competition, its performance, its market readiness, 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 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.
