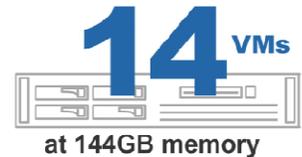


More Kingston server memory means more database virtual machines



OUR FINDINGS

Consolidating database servers with virtualization technology allows companies to reduce energy usage, save data center space and lower management costs. In Principled Technologies' tests in our labs, adding Kingston® memory to a server dramatically increased the number of database virtual machines (VMs) the server could support. In fact, the increase in VMs the server could support increased proportionally with the increase in RAM.

OUR PROCESS

We used the open-source DVD Store benchmark, which provides a workload representative of a real-world database application, to determine how many comparably performing virtual machines a server could run with four different memory configurations. Our server ran VMware® vSphere 4,™ and the VMs ran Microsoft® Windows Server® 2008 R2 Enterprise Edition with Microsoft SQL Server® 2008.



PROJECT OVERVIEW

This report presents the performance benefit one can expect when adding memory to a server. We tested the virtualized database performance of our server with four memory configurations:

- 12GB of RAM (6 x 2 GB)
- 72GB of RAM (18 x 4 GB)
- 144GB of RAM (18 x 8 GB)
- 192GB of RAM (12 x 16 GB)

The results also suggest that, for maximum potential for future virtual machine growth, companies would do well to select the highest capacity memory modules.

WORKLOAD

DVD Store Version 2 (DS2) 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 via the performance monitor counter on each client 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 instance on each virtual machine, each with an 8GB database. We used the default DS2 parameters and setup configuration, with the exceptions we note in the Setting up DVD Store section in the How we tested section of this report.

Each 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 0.5 seconds think time. For more details on the DS2 tool, see <http://www.delltechcenter.com/page/DVD+Store>.

Figure 1 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)	10
Virtual NIC type	VMXNET 3
Number of virtual disks	3 (OS, SQL logs, SQL database)

Figure 1: VM configuration used for testing

MEMORY CONFIGURATION COMPARISON

Figure 2 shows a side-by-side comparison of the memory configurations we tested. We used the same physical server for all configurations. Appendix A presents detailed system information for the test server.

Memory specifications	12GB configuration	72GB configuration	144GB configuration	192GB configuration
Vendor and model number	Kingston KVR1333D3D8R9S/ 2G	Kingston KVR1333D3D4R9S/ 4G	Kingston KVR1333D3D4R9S/ 8G	Kingston KVR1066D3Q4R7S/ 16G
Type	PC3-10600R	PC3-10600R	PC3-10600R	PC3-8500R
Speed (MHz)	1,333	1,333	1,333	1,066
Speed running in system (MHz)	1,333	800	800	800
Size (GB)	2	4	8	16
Total memory in system (GB)	12 (6 x 2 GB)	72 (18 x 4 GB)	144 (18 x 8 GB)	192 (12 x 16 GB)

Figure 2: Details of the four memory configurations tested

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 installations inside the virtual machines. We stored all virtual machines on the NetApp FAS960 filer, which we connected to the server via a QLogic QLE2462 dual-port fibre controller.

WHAT WE FOUND

As Figure 3 shows, the 192GB configuration supported 18 VMs, a 28.5 percent increase over the 144GB configuration, which supported 14 VMs, a 157.1 percent increase over the 72GB configuration, which supported 7 VMs, and a 1,700.0 percent increase over the 12GB configuration, which supported only 1 VM.

As Figure 4 shows, the 192GB configuration achieved a DS2 score of 60,609 OPM, a 23.2 percent increase over the 144GB configuration, which achieved a score of 49,166 OPM, a 134.4 percent increase over the 72GB configuration, which achieved a score of 25,853 OPM, and a 1,520.6 percent increase over the 12GB configuration, which achieved a score of 3,740 OPM.

Figure 5 shows the OPM results for each VM on the four server memory configurations. The results show the OPM recorded by each test client and the total OPM for each configuration.

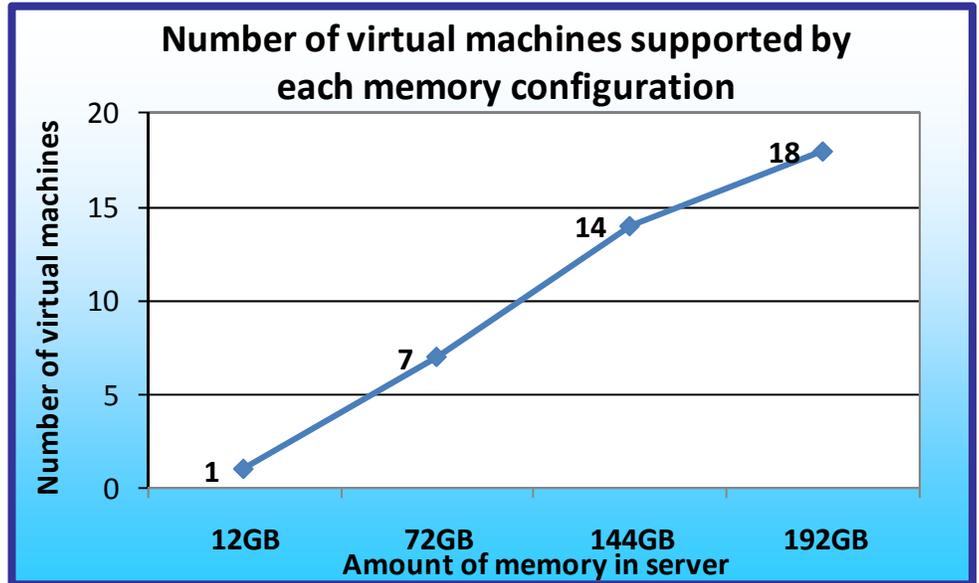


Figure 3: Number of virtual machines for a server with four different memory amounts using DVD Store Version 2. Higher numbers are better.

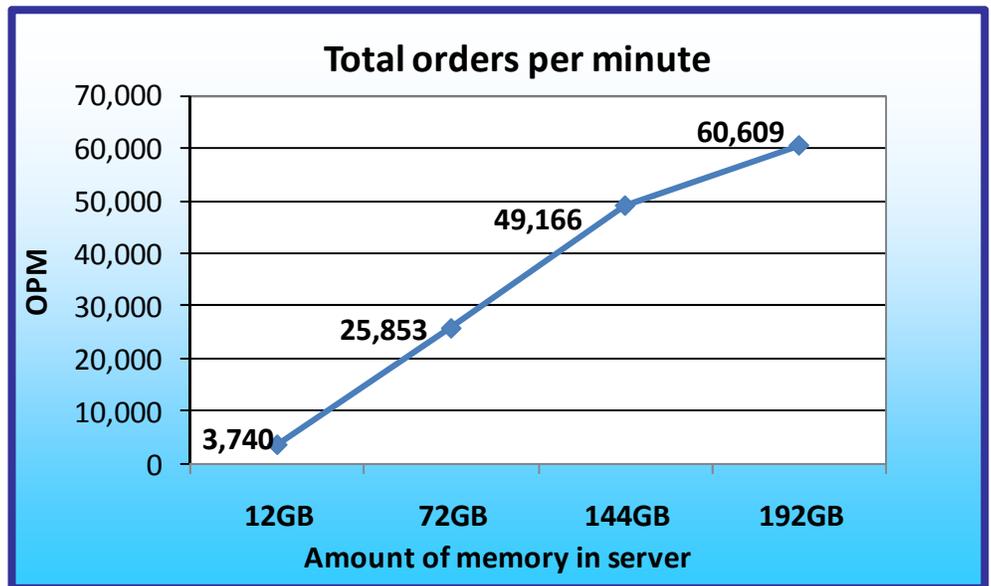


Figure 4: DVD Store Version 2 performance results, in OPM, for a server with four different memory amounts. Higher numbers are better.

For each memory configuration, we used the maximum number of simultaneous virtual machines that could run without over subscribing memory. We performed three runs at the total number of virtual machines to verify the scores met these standards. We report the median of the three runs.

Note that, because we are using 10GB of memory in each VM, it would appear the 192GB configuration could readily support 19 VMs. However, due to memory overhead for the hypervisor and VMs, this configuration would support more than 18 VMs, but only by over-committing memory. ESX would then resort to memory reclamation techniques to manage this demand for additional memory; a situation that would negatively affect our tests.

Client	12GB configuration	72GB configuration	144GB configuration	192GB configuration
Client 1	3,740	3,679	3,552	3,472
Client 2		3,702	3,528	3,458
Client 3		3,691	3,539	3,441
Client 4		3,688	3,531	3,434
Client 5		3,693	3,501	3,420
Client 6		3,706	3,510	3,414
Client 7		3,694	3,506	3,403
Client 8			3,496	3,379
Client 9			3,499	3,338
Client 10			3,498	3,326
Client 11			3,505	3,326
Client 12			3,507	3,320
Client 13			3,502	3,310
Client 14			3,492	3,310
Client 15				3,315
Client 16				3,310
Client 17				3,318
Client 18				3,315
Total OPM	3,740	25,853	49,166	60,609

Figure 5: OPM from the median run of DVD Store Version 2 for each configuration. Higher numbers are better.

We focused our analysis on the overall improvement of the system caused by adding memory. As Figure 5 shows, we did see a modest decrease in OPM per VM performance at the higher VM counts. This is likely due to the changing variables we encountered during different phases of the testing. A combination of

these changing variables could have affected the performance of the database VMs. The factors that affect the OPM performance include the following:

- Change in memory operating speed at higher DIMM counts
- Increase in CPU utilization at higher VM counts
- CPU over-allocation at 17 and 18 VMs
- Effects of NUMA scheduling
- Effects of Hyper-Threading

HOW WE TESTED

Setting up and configuring the NetApp FAS960 filer

We used a NetApp FAS960 SAN for this testing. The FAS960 filer has four DS14 disk shelves with 14 disks in each shelf. We used a QLogic QLE2462 dual-port host bus adapter (HBA) in the server for testing. We connected both ports of the dual port HBA to the FAS960 filer.

We created two RAID groups with 26 disks in each group. We combined the two RAID groups into one aggregate. We created three LUNs on the aggregate, one 1TB LUN for OS, one 1TB LUN for database, and one 600MB LUN for log data.

Configuring the test server

We installed a fresh copy of VMware vSphere 4 (which we also refer to as ESX 4) on the test server.

Installing VMware ESX 4 on the host

1. Insert the ESX DVD, and restart the computer.
2. To start ESX in graphical mode, press Enter.
3. On the Welcome screen, click Next.
4. Click the checkbox to accept the terms of the license agreement, and click Next.
5. Accept the default keyboard layout, and click Next.
6. On the Custom Drivers screen, choose No to install custom drivers, and click Next.
7. You will receive a warning if you chose No. Click Yes to proceed. Drivers will load at this point.
8. Click Next.
9. Enter your serial number now, or choose to enter one later. Click Next.
10. On the Network Configuration screen, choose the NIC that you will use for system tasks and for the client/server test traffic. Click Next.
11. On the next Network Configuration screen, set your IP addressing information, subnet mask, DNS, and hostname. Click Next.
12. On the Setup Type screen, choose standard setup.
13. Select the virtual disk you wish to install ESX on, and click Next.
14. On the Data Loss warning screen, click OK.
15. On the Time Zone Settings screen, choose your appropriate time zone, and click Next.
16. On the Date and Time screen, modify the date and time as you wish, and click Next.

17. Assign a root password for ESX. Optionally, add additional users if you wish, and click Next.
18. On the Summary screen, click Next to install.
19. When the installation is complete, click Next.
20. Click Finish to reboot the system.

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. Click Next.
5. Select the appropriate LUN, and click Next.
6. Click Next.
7. Enter a name for the datastore. 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 virtual disks and SQL Server log virtual disks.

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. We chose the OS Datastore, and click Next.
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 processors. Click Next.
11. Choose 10240MB RAM. Click Next.
12. Click 1 for the number of NICs, and select vmxnet3. Click Next.
13. Leave the default virtual storage controller. Click Next.
14. Choose to create a new virtual disk. Click Next.
15. Make the OS virtual disk size 30 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 25GB for the virtual disk size, and specify the datastore for SQL Server data. Note that we alternated even numbered VMs on one set of LUNs and odd numbered VMs on another.
23. Choose SCSI(1:0) 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 12GB for the virtual disk size, and specify the datastore for SQL Server log virtual disks. Note that we alternated even numbered VMs on one set of LUNs and odd numbered VMs on another.
28. Choose SCSI(2:0) for the device node, and click Next.
29. Click SCSI Controller 1, and choose Change Type.
30. Choose Paravirtual, and click OK.
31. Click SCSI Controller 2, and choose Change Type.
32. Choose Paravirtual, 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 systems 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. Log into the VM after reboot.
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, Windows Update.
2. Click View Advanced Options.
3. Click Never Check for Updates.
4. Click OK.
5. Click Check for Updates.
6. Click Install Now when the application prompts you to install an update for Windows Update.
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 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. Click Run SETUP.EXE on the AutoPlay menu.
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. Click Restart now to restart the server upon completion.
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 data, and choose New Simple Volume. Click Next.
7. Accept the default volume size. Click Next.
8. Choose E: from the drop-down menu for the drive letter. Click Next.
9. Type `sqldata` 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 logs, and choose New Simple Volume. Click Next.
12. Accept the default volume size. Click Next.
13. Choose F: from the drop-down menu for the drive letter. Click Next.
14. Type `sqllog` 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.

Creating the additional VMs on the host blade

After installing and configuring the initial VM, we cloned that VM using a combination of the `vmkfstools` and `vmware-cmd` utilities on the VMware vSphere Server platform. vSphere does offer a GUI wizard to clone out VMs. Because we were cloning out many of the same VMs, we used scripts instead of their GUI. Below is a sample shell script to clone VMs containing three virtual disks, such as our configuration.

```
# cp -reply=yes /sourcepath/VM1.vmx /destpath/VM2.vmx
# sed -i "s/VM1/VM2/g" /destpath/VM2.vmx
# sed -i '/sched.swap.derivedName/d' /destpath/VM2.vmx
```

```
# sed -i '/checkpoint.vmState/d' /destpath/VM2.vmx
# vmkfstools -i /sourcepath/VM1.vmdk /destpath/VM2.vmdk
# vmkfstools -i /sourcepath/VM1_1.vmdk /destpath/VM2_1.vmdk
# vmkfstools -i /sourcepath/VM1_2.vmdk /destpath/VM2_2.vmdk
# vmware-cmd -s register /destpath/VM2.vmx
```

Configuring the additional VMs after cloning

1. Power on the VM, and open the console.
2. In the task bar, right-click the network icon, and choose Open Network and Sharing Center.
3. Click Change Adapter Settings.
4. Right-click the NIC, and choose properties.
5. Select Internet Protocol Version 4 (TCP/IPv4), and click Properties.
6. In the Internet Protocol Version 4 (TCP/IPv4) Properties screen, select the Use the following IP address radio button.
7. Enter a valid static IP address for this new VM, subnet mask, and default gateway.
8. Click OK, and click Close to exit.
9. Click Start, and type `computer name` in the search box.
10. Choose Rename this computer.
11. Click Change Settings to change the computer name.
12. Click Change.
13. Type the new computer name.
14. Click OK, and click OK again.
15. When the application prompts you to do so, reboot the VM.
16. Repeat steps 1 through 15 for each VM.

Installing and configuring the database clients

For the DVD Store scripts, we used a number of clients 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 log on.
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 8 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 8GB 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 G for the 8GB 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 8GB 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 G for the 8GB 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 8GB 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 8GB 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 8GB 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 8GB database:

1. `ds2_create_orders_8gb.sh`
2. `ds2_create_inv_8gb.sh`
3. `ds2_create_prod_8gb.sh`
4. `ds2_create_cust_8gb.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 8GB database version SQL Server 2008, and then used those master copies to restore our test database to each 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 8GB 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 each 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 and 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 8GB database we created earlier, we had to change the following constants:

- In the routine Controller(), we changed the string sizes. We added the G option for the 8GB 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 use 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.
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.

APPENDIX A – SERVER CONFIGURATION INFORMATION

Figure 6 provides detailed configuration information about the test server.

Server	SuperMicro® SuperServer 6026T-NTR+
General processor setup	
Number of processor packages	2
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	Super
Motherboard model number	X8DTN+
Motherboard chipset	Intel 5520
BIOS name and version	Supermicro 1.1a
BIOS settings	Default
Memory modules	
12GB configuration	
Vendor and model number	Kingston KVR1333D3D8R9S/2G
Type	PC3-10600R
Speed (MHz)	1,333
Speed in the system currently running @ (MHz)	1,333
Timing/Latency (tCL-tRCD-iRP-tRASmin)	9-9-9-24
Size (GB)	2
Number of RAM modules	6 x 2 GB
Chip organization	Double-sided
72GB configuration	
Vendor and model number	Kingston KVR1333D3D4R9S/4G
Type	PC3-10600R
Speed (MHz)	1,333
Speed in the system currently running @ (MHz)	800

Server	SuperMicro® SuperServer 6026T-NTR+
Timing/Latency (tCL-tRCD-iRP-tRASmin)	6-6-6-16
Size (GB)	4
Number of RAM modules	18 x 4 GB
Chip organization	Double-sided
144GB configuration	
Vendor and model number	Kingston KVR1333D3D4R9S/8G
Type	PC3-10600R
Speed (MHz)	1,333
Speed in the system currently running @ (MHz)	800
Timing/Latency (tCL-tRCD-iRP-tRASmin)	6-6-6-16
Size (GB)	8
Number of RAM modules	18 x 8 GB
Chip organization	Double-sided
192GB configuration	
Vendor and model number	Kingston KVR1066D3Q4R7S/16G
Type	PC3-8500R
Speed (MHz)	1,066
Speed in the system currently running @ (MHz)	800
Timing/Latency (tCL-tRCD-iRP-tRASmin)	6-6-6-16
Size (GB)	16
Number of RAM modules	12 x 16 GB
Chip organization	Double-sided
Hard disk	
Vendor and model number	Western Digital WD1600AAJS
Number of disks in system	2
Size (GB)	160
Buffer size (MB)	8
RPM	7,200
Type	SATA
Controller	Intel ICH10R SATA 3.0 Gbps Controller
Operating system	
Name	WMware ESX 4.0.0
Build number	208167
File system	VMFS
Language	English
Network card/subsystem	
Vendor and model number	Intel 82576 Dual-Port Gigabit Ethernet Controller
Type	Integrated

Server	SuperMicro® SuperServer 6026T-NTR+
Fibre adapter	
Vendor and model number	QLogic QLE2462 dual port controller
Type	PCI Express
USB ports	
Number	4
Type	USB 2.0

Figure 6: 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.

MKP-195