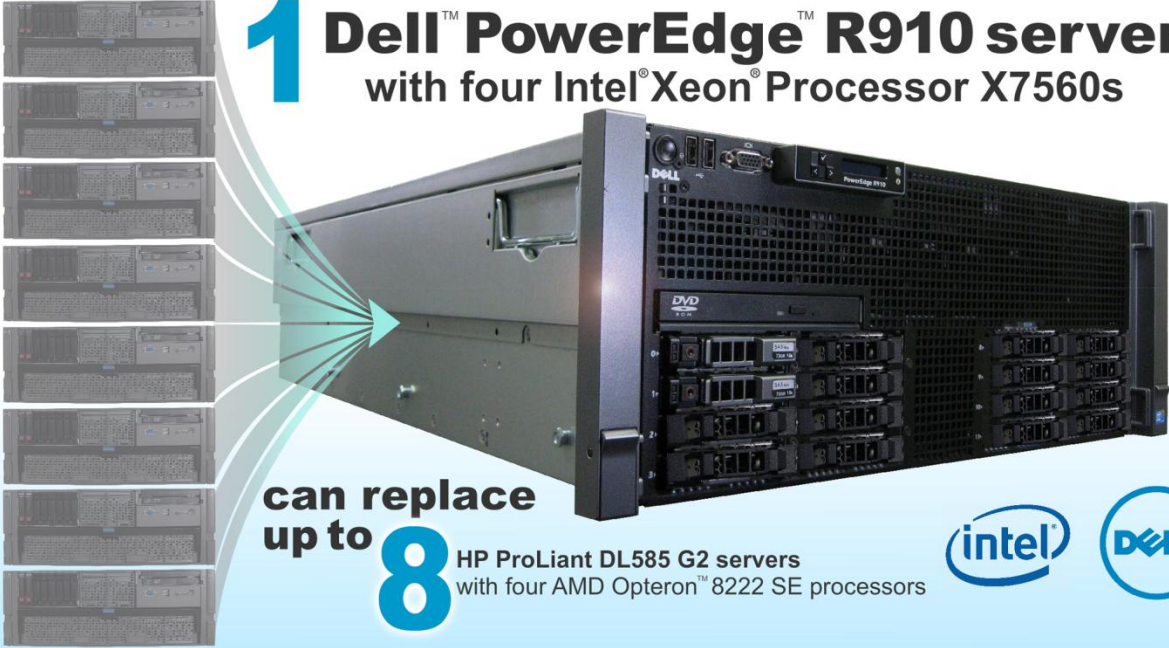



DATABASE SERVERS: CONSOLIDATION VIA VIRTUALIZATION



1 Dell™ PowerEdge™ R910 server
with four Intel® Xeon® Processor X7560s

can replace
up to **8** HP ProLiant DL585 G2 servers
with four AMD Opteron™ 8222 SE processors



OUR FINDINGS

Consolidating database servers via virtualization enables you to reduce energy usage, save precious data center space, and lower management costs¹. In Principled Technologies' tests in our labs, as the above graphic illustrates, a Dell PowerEdge R910 with four new Intel Xeon Processor X7560s will let you consolidate eight HP ProLiant DL585 G2 servers each with four AMD Opteron 8222 SE processors. Such consolidation would deliver significant savings in all three of these cost areas.

OUR PROCESS

We used the open-source DVD Store benchmark to provide a workload representative of many real-world database applications. With it, we measured the non-virtualized performance of a typical 5-year old legacy dual socket server. We then created multiple VMware® vSphere™ 4 (vSphere) virtual machines (VM) each running Microsoft® Windows Server® 2008 R2 Enterprise Edition with Microsoft SQL Server® 2008 R2 on the test servers. We next measured how many of those legacy VMs the servers under test could host with each VM maintaining the same (or greater) DVD Store performance as the legacy server.

¹ Source: Based on DVD Store benchmark testing conducted by Principled Technologies®, Inc., and reported in "DATABASE SERVERS: CONSOLIDATION VIA VIRTUALIZATION," an August 2010 report commissioned by Dell Inc.



PROJECT OVERVIEW

We tested the following servers:

- Dell PowerEdge R910 with Intel Xeon Processor X7560s
- HP ProLiant DL585 G2 with AMD Opteron 8222 SE processors

For testing, all servers ran Microsoft Windows Server 2008 R2 Enterprise Edition with Microsoft SQL Server 2008 R2 inside a VMware vSphere 4 virtual machine.

To determine the consolidation achievable through virtualization, we first measured the average orders-per-minute (OPM) results from our typical 5-year old legacy dual socket server, which delivered 9,000 OPM. We then determined that the HP ProLiant DL585 G2 with four AMD Opteron 8222 SE processors was able to consolidate only 4 legacy servers through virtualization, while the Dell PowerEdge R910 with four Intel Xeon Processor X7560s was able to consolidate 32 legacy servers, and was thus able to consolidate 8 of the HP ProLiant DL585 G2 servers with AMD Opteron 8222 SE processors.

WORKLOAD

We used DVD Store Version 2 (DS2), an open-source application that models an online DVD store, to build the workload. DS2 has a back-end database component, a front-end Web application layer, and a driver layer that executes the workload. Because our goal was to test database server performance, we did not use the front-end Web client component of DS2, but ran the included driver program directly via the client machine's command-line interface. We used the default setup configuration for DS2, with exceptions we note in the DVD Store setup section in the How We Tested Section of this report.

DS2 reports results in orders per minute (OPM). A DS2 order consists of a customer login, a search for movies by title, actor, or category, and a purchase. The workload also performs other actions, such as adding new customers, to exercise a wide range of database functions.

We configured the HP ProLiant DL585 G2 VMs with a 10GB DS2 database. The new Dell PowerEdge R910 server contained a 10GB DS2 database on each VM it ran.

Each client machine ran a single instance of DS2, with 32 threads, to simulate a heavily loaded environment. The client machines ran with no think time, processing requests as quickly as they were able. The workload ran for 30 minutes on each server, with the VMs on each server running the workload simultaneously.

While the DVD Store client application outputs OPM at 10-second intervals visually, we chose to collect OPM via a performance monitor counter on each client VM at 1-second intervals. We scored a VM as acceptable if its OPM results were at the same level as or better than the OPM results of the older servers.

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

SYSTEM COMPARISON

Figure 1 shows a side-by-side comparison of the key hardware differences between the servers.

Hardware specifications	Dell PowerEdge R910 with Intel Xeon Processor X7560s	HP ProLiant DL585 G2 with AMD Opteron 8222 SE processors
CPU	Intel Xeon X7560	AMD Opteron 8222 SE
CPU speed (GHz)	2.27	3.00
Number of processor packages	4	4
Number of cores per processor package	8	2
Number of hardware threads per core	2	1
Memory type	PC3-8500 DD3	PC2-5300P DDR2
Total memory (GB)	512	32

Figure 1: Key system configuration information for the two test servers.

Figure 2 shows the settings we used on the VMs for the Dell PowerEdge R910 and the HP ProLiant DL 585 G2 for testing. To determine the amount of memory per VM, we found the maximum number of VMs the server could run without over-subscribing the server's processor threads. We configured each VM with 2 vCPUs, or two threads. The HP ProLiant DL585 G2 has 4 dual-core processors, or eight threads, so it ran four VMs without over-subscribing processors. We configured each VM on the HP ProLiant DL585 G2 with 8GB of memory to completely use up its 32 GB of system memory. The Dell PowerEdge R910 has 4 eight-core processors with Hyper-threading, for a total of 64 threads. We configured 32 VMs with 15 GB on the Dell PowerEdge R910 in order to use nearly all of the 512 GB system memory and not over-subscribe the logical processors. Refer to the How We Tested section below for detailed steps of how we set up the VMs.

VM specifications	Dell PowerEdge R910 with Intel Xeon Processor X7560s	HP ProLiant DL585 G2 with AMD Opteron 8222 SE processors
vCPU	2	2
Memory (GB)	15	8
Virtual NIC type	vmxnet3	vmxnet3
Number of virtual disks	3	3

Figure 2: VM configuration information for the Dell PowerEdge R910 with Intel Xeon Processor X7560s and the HP ProLiant DL585 G2 server with AMD Opteron 8222 SE processors.

WHAT WE FOUND

As Figure 3 shows, the Dell PowerEdge R910 with the Intel Xeon Processor X7560s provides a performance consolidation factor of eight over the HP ProLiant DL585 G2 with AMD Opteron 8222 SE processors².

To determine the maximum number of simultaneous virtual machines the two servers could handle, we added VMs on each server until one or more VM had OPM results drop below the legacy server's median OPM results. We then subtracted one VM to identify the total number of VMs the server could handle. We performed three runs at the total number of VMs to verify the scores met these standards.

Figure 4 shows the total OPM results for the two servers. The Dell PowerEdge R910 with Intel Xeon Processor X7560s achieved 324,990 total OPM, while the HP ProLiant DL585 G2 with AMD Opteron 8222 SE processors achieved only 53,286 total OPM³. Figure 5 shows the OPM results, by VM, for the two servers. The

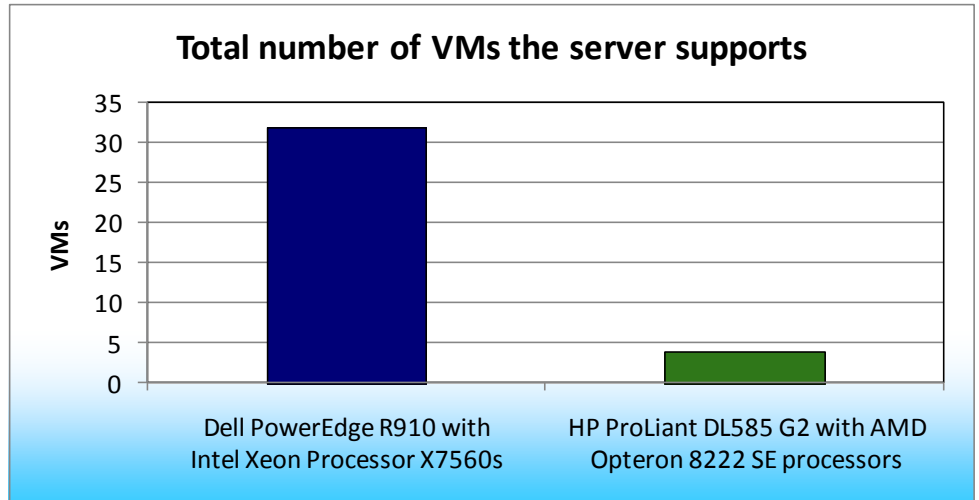


Figure 3: SQL Server performance consolidation results for the Intel Xeon Processor X7560-based Dell PowerEdge R910 and the AMD Opteron 8222 SE processor-based HP ProLiant DL585 G2. Higher numbers are better.

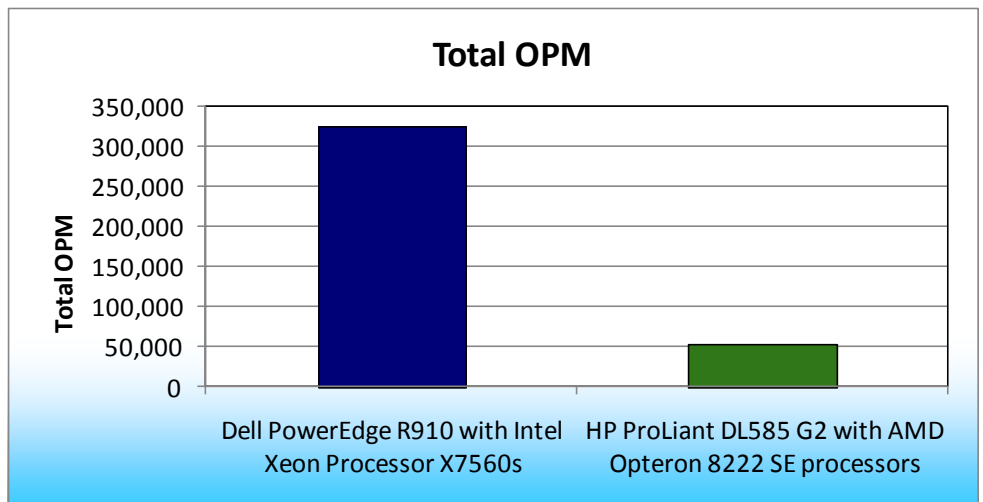


Figure 4: Total OPM results for all VMs on the Intel Xeon Processor X7560-based Dell PowerEdge R910 and the AMD Opteron 8222 SE-based HP ProLiant DL585 G2.

² Source: Based on DVD Store benchmark testing conducted by Principled Technologies, Inc., and reported in "DATABASE SERVERS: CONSOLIDATION VIA VIRTUALIZATION," an August 2010 report commissioned by Dell Inc.

³ Ibid.

Dell PowerEdge R910 with Intel Xeon Processor X7560s results also show that 32 VMs achieved greater total OPM than 4 VMs on the HP ProLiant DL585 G2 server with AMD Opteron 8222 SE processors⁴. Thus, one Dell PowerEdge R910 server with Intel Xeon Processor X7560s can replace eight HP ProLiant DL585 G2 servers with AMD Opteron 8222 SE processors while providing equal or greater performance.

OPM	Dell PowerEdge R910 with Intel Xeon Processor X7560s	HP ProLiant DL585 G2 with AMD Opteron 8222 SE processors
Virtual machine 1	11,081	14,113
Virtual machine 2	9,562	14,344
Virtual machine 3	9,875	10,911
Virtual machine 4	10,214	13,918
Virtual machine 5	10,073	
Virtual machine 6	9,812	
Virtual machine 7	10,390	
Virtual machine 8	10,874	
Virtual machine 9	9,148	
Virtual machine 10	9,674	
Virtual machine 11	10,300	
Virtual machine 12	9,964	
Virtual machine 13	9,806	
Virtual machine 14	9,755	
Virtual machine 15	9,973	
Virtual machine 16	10,049	
Virtual machine 17	10,628	
Virtual machine 18	9,760	
Virtual machine 19	10,068	
Virtual machine 20	10,040	
Virtual machine 21	10,300	
Virtual machine 22	10,010	
Virtual machine 23	10,667	
Virtual machine 24	9,704	
Virtual machine 25	10,362	
Virtual machine 26	10,092	
Virtual machine 27	10,563	
Virtual machine 28	10,343	
Virtual machine 29	10,210	

⁴ Source: Based on DVD Store benchmark testing conducted by Principled Technologies, Inc., and reported in "DATABASE SERVERS: CONSOLIDATION VIA VIRTUALIZATION," an August 2010 report commissioned by Dell Inc.

Virtual machine 30	10,279	
Virtual machine 31	10,407	
Virtual machine 32	11,007	
Total OPM	324,990	53,286

Figure 5: DVD Store total OPM results and results for the 32 VMs on the Dell PowerEdge R910 with Intel Xeon Processor X7560s and the 4 VMs on the HP ProLiant DL585 G2 with AMD Opteron 8222 SE processors.

HOW WE TESTED

Setting up and configuring the Dell PowerEdge R910 storage

We used an EMC® CX3-40 Fibre Channel SAN for testing. The SAN had two storage processors (SP), SP-A and SP-B. We installed a QLogic® QLE2462 dual-port host bus adapter (HBA) in the Dell PowerEdge R910 rack server. We cabled one HBA port to each SP (A and B) on the SAN to balance the load between SPs. We used seven enclosures of disks on the SAN; with one exception where one enclosure had 13 disks, each enclosure had 15 disks. We designated RAID 0 for all RAID group configurations.

We then created two RAID groups of 11 disks, and created one logical unit number (LUN) on each RAID group for SQL logs. We assigned one log LUN to each SP. To balance processing load and disk transfers, we assigned Log-LUN-1 to SP-B and Log-LUN-2 to SP-A. We created eight RAID groups with 10 disks in each. We then created one 512GB LUN in each RAID group, for a total of eight LUNs dedicated to virtual machines to hold the SQL Server 2008 database data and the operating system’s virtual hard drives (VHDs). We used the EMC MetaLUN feature and assigned four LUNs to MetaLUN-1 and four LUNs to MetaLUN-2. To balance processing load and disk transfers, we assigned MetaLUN-1 to SP-A and MetaLUN-2 to SP-B. We evenly distributed the VMs database data across the two MetaLUNs.

Setting up and configuring the HP ProLiant DL585 G2 storage

We used an HP StorageWorks 4000 Enterprise Virtual Array (EVA) for testing. The array had two HSV200 controllers and two M5314 disk drive enclosures. Each M5314 enclosure had 14 hard disks. The hard disks were 146GB 10,000 RPM disks. We cabled the two controllers to the disk drive enclosures as outlined by HP’s documentation. We installed an HP FC2242 StorageWorks PCI-E 4Gb dual-port host bus adapter (HBA) in the HP ProLiant HP DL585 G2 Server. We connected both HBA ports and the HP StorageWorks 4000 EVA storage to a Brocade 200E fibre switch.

We configured the HP StorageWorks 4000 EVA disks in two RAID 0 groups. We then created two logical unit number (LUN) on the RAID groups for the virtual hard drives (VHDs) that would hold the operating systems for the virtual machines and the SQL Server 2008 log data. We created two LUNs for SQL Server

database data. We put even number VMs on one LUN and odd number of VMs on the other. For example, VM 1 and 3 were on one LUN and VM 2 and 4 were on another LUN. We did this to evenly distribute the load on the storage.

Setting up the servers

We installed a fresh copy of VMware vSphere 4 on all servers.

Installing VMware vSphere 4 on the host

1. Insert the vSphere DVD, and restart the computer.
2. To start vSphere 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 vSphere 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 vSphere. 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 vSphere 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. Type the word `yes` and delete the word `no`.
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 the vSphere client on a management workstation

1. Configure your management workstation to be on the same subnet as your vSphere server. In Internet Explorer®, navigate to the IP address of your vSphere server.
2. Accept certificate warnings, and 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. 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. 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 two virtual processors. Click Next.
11. Choose 15 GB RAM on Dell PowerEdge R910 and 8 GB RAM on HP ProLiant DL585 G2. 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. See the Setting up and configuring the storage section for more details.
23. Choose SCSI(1:0) for the device node, and click Next.
24. Click Finish.
25. On the Hardware tab, click Add...
26. Click Hard Disk, and click Next.
27. Click Create a new virtual disk for the SQL Server log, and click Next.
28. Specify 20GB 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. See the storage section for more details.
29. Choose SCSI(2:0) for the device node, and click Next.
30. Click Finish.
31. Click SCSI Controller 1, and choose Change Type.
32. Choose Paravirtual, and click OK.
33. Click SCSI Controller 2, and choose Change Type.
34. Choose Paravirtual, and click OK.
35. Click Finish, and click OK.

Installing the guest operating system

1. Insert the installation DVD for Windows Server 2008 R2 (64-bit) into the DVD drive.
2. Select the VM in the left hand window pane.
3. Power on the VM.
4. Click the CD/DVD icon, and select Connect to ISO image on local disk.
5. Choose your ISO image for the OS, and click Open.
6. To load the OS disc, right-click the VM name in the left pane, and choose Guest → Send Ctrl+Alt+Delete.
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, right click VM in left-hand pane, and select Guest→Install/Upgrade VMware Tools from the drop-down menu.
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. Click Finish to complete the installation.
24. Click Yes to reboot the VM.
25. Log into the VM after reboot.
26. Click Start, Control Panel, and double-click System.
27. Click Change Settings.
28. Click Change.
29. Enter the new computer name, and click OK.
30. Click OK to restart, click Close, and click Restart Now.

Installing Windows updates

1. Click Start→Windows Update.
2. Click the Let me choose my settings link.
3. Select Never check for updates from the drop-down menu.
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 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. On the Microsoft SQL Server 2008 R2 Setup pop-up window asking you to enable the .NET Framework Core role, click OK.
7. Click Installation.
8. Click New installation or add features to an existing installation.
9. On the Setup support rules screen, click OK.
10. Specify the Evaluation free edition, and click Next.

11. Click the checkbox to accept the license terms, and click Next.
12. Click Install to install the setup support files.
13. When the install is finished, click Next.
14. On the Setup Role screen, choose SQL Server Feature Installation, and click Next.
15. 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.
16. On the Installation Rules screen, click Next.
17. On the Instance configuration screen, leave the default selection of default instance, and click Next.
18. On the Disk space requirements screen, click Next.
19. 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.
20. On the Database Engine Configuration screen, select Mixed Mode.
21. Enter a password for the system administrator (sa) account.
22. Click Add Current user. This may take several seconds.
23. Click Next.
24. On the Error and usage-reporting screen, click Next.
25. On the Installation rules screen, click Next.
26. On the Ready to Install screen, click Install.
27. When the install finishes, click Close.
28. Click Start→All Programs→Microsoft SQL Server 2008 R2→Configuration Tools→SQL Server Configuration Manager.
29. Expand the Network Configuration node, and select the relevant instance.
30. In the right pane, right-click TCP/IP, and choose Enable.
31. In the left pane, click SQL Server Services.
32. In the right pane, right-click the SQL Server service, and select Restart.
33. Create a SQL Server login for the ds2user (see the DVD Store setup section below for the specific script to use).
34. 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.

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.
10. Choose NTFS, 64K allocation unit size, and check the Perform a quick format checkbox. Click Next.
11. Click Finish.
12. Right-click the volume meant for SQL Server logs, and choose New Simple Volume. Click Next.

13. Accept the default volume size. Click Next.
14. Choose F: from the drop-down menu for the drive letter. Click Next.
15. Type `sqllog` for the volume name. Click Next. (20GB)
16. Choose NTFS, 64K allocation unit size, and check the Perform a quick format checkbox. Click Next.
17. 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 VMs. Because we were cloning 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 -f /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 the Change button.
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 virtual clients to simulate a number of users putting a load on the server. We configured a Dell PowerEdge R910 with vSphere 4.0 as the host OS. We created a VM running Microsoft Windows Server 2003 R2 Enterprise Edition for our virtual clients. We configured each virtual client with one vCPU and 4GB of memory. Each virtual client connected to a single VM. To ensure valid results, we used a physical client running the same OS and memory configuration. This single client ran the executable for the first server VM on both systems.

On each client we created a folder 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 orders per minute (OPM) each virtual database server returned. We followed this process for each installation:

1. Install Microsoft Windows Server 2003 R2 Enterprise Edition with 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 10 GB of user data we used in this test, we had to make a few straightforward changes to the source code and to the DVD Store application's scripts. Note: We created our test data on a Linux system to take advantage of the larger RAND MAX.

Editing the `ds2_create_orders.c` module

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

On the command line for the `ds2_create_orders.c` module, we specified the size. The available options were S (small), M (medium), and L (large). We added the case T for the 10GB database. In the switch statement that sets the values for the variables `max_cust_id` and `max_prod_id`, we added cases that assigned them the proper values for the 10GB database size.

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

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

Editing the `ds2_create_cust.c` module

We had to make the same changes to the `ds2_create_cust.c` module that we made to the `ds2_create_orders.c` module. On the command line for the `ds2_create_cust.c` module, we specified the size. The available options were S (small), M (medium), and L (large). We added the case T for the 10GB database. In the switch statement that sets the values for the variables `max_cust_id` and `max_prod_id`, we added cases that assigned them the proper values for the 10GB database size.

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

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

Generating the data for the 10GB database

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

1. `ds2_create_orders_10gb.sh`
2. `ds2_create_inv_10gb.sh`
3. `ds2_create_prod_10gb.sh`
4. `ds2_create_cust_10gb.sh`

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

Creating the database

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

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

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

We made the following changes in the build scripts:

- Because we varied the size of the datasets, we sized the files in our scripts to reflect the database size and the number of files per filegroup. We allowed for approximately 40 percent free space in our database files to ensure that filegrowth activity did not occur during the testing.
- We followed Microsoft's recommendation of having 0.25 to 1 file per filegroup per logical processor. We used two files per filegroup on 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.
- In our pre-test experimentation, we found that SQL Server 2008 would occasionally choose an inefficient query plan in the DVD Store workload. To flush the plan cache during these cases, we used the DBCC FREEPROCACHE command.

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

Editing the workload script - ds2xdriver.cs module

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

- In the routine Controller(), we changed the string sizes. We added the T option for the 10GB database size. DS2 uses the sizes string to interpret the db_size_str option.
- In the class Controller, we changed the arrays MAX_CUSTOMER and MAX_PRODUCT. To each, we added values specifying the bounds for the customer and product IDs. The Controller() routine uses these arrays.
- We added a command-line parameter for the database name:

```
-database_name
```

Editing the workload script - ds2sqlserverfns.cs module

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

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

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

Recompiling the ds2sqlserverdriver.exe executable

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

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

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

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 20 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. Run the DBCC FREEPROCACHE command if necessary (see the Creating the database section).
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 servers.

System	Dell PowerEdge R910 with Intel Xeon Processor X7560s	HP ProLiant DL585 G2 with AMD Opteron 8222 SE processors
General		
Number of processor packages	4	4
Number of cores per processor	8	2
Number of hardware threads per core	2	1
CPU		
Vendor	Intel	AMD
Model	Xeon X7560	Opteron 8222 SE
Stepping	D0	3
Socket type	LGA1567	Socket F (1207)
Core frequency (GHz)	2.27	3.00
Bus frequency	6.4 GT/s	2,000 MHz Hyper Transport
L1 cache	32 KB + 32 KB (per core)	64 KB + 64 KB (per core)
L2 cache	256 KB (per core)	2 MB (2 x 1 MB)
L3 cache	24 MB	N/A
Platform		
Vendor and model number	Dell PowerEdge R910	HP ProLiant DL585 G2
Motherboard model number	0P658H	PB729AE9QUDO49
Motherboard chipset	Intel 7500	NVIDIA nForce Pro 2050
BIOS name and version	Dell 1.1.7	HP BIOS A07 (12/14/2009)
BIOS settings	Default	Default
Memory module(s)		
Total RAM in system (GB)	512	32
Vendor and model number	Samsung M393B1K73BHD-CF8	Hynix HYMP525P72BP4-Y5
Type	PC3-8500 DDR3	PC2-5300P DDR2
Speed (MHz)	1,066	667
Speed running in the system (MHz)	1,066	667
Size (GB)	8	2
Number of RAM module(s)	32	16
Chip organization	Double-sided	Double-sided
Rank	Quad	Dual
Hard disk		
Vendor and model number	Seagate ST973451SS	HP DH072ABAA6
Number of disks in system	2	2
Size (GB)	73	72
Buffer size (MB)	16	16

System	Dell PowerEdge R910 with Intel Xeon Processor X7560s	HP ProLiant DL585 G2 with AMD Opteron 8222 SE processors
RPM	15,000	15,000
Type	SAS 3Gb/s	SAS 3Gb/s
Disk Controller		
Vendor and model	Dell PERC H700	HP Smart Array P400
RAID configuration	RAID 1	RAID 1
Operating system (host)		
Name	VMware ESX 4.1	VMware ESX 4.1
Build number	261974	261974
Service pack	Update 2	Update 2
File system	VMFS	VMFS
Kernel	2.6.18-164.ESX	2.6.18-164.ESX
Language	English	English
Operating system (virtualized)		
Name	Windows Server 2008 R2 Enterprise Edition	Windows Server 2008 R2 Enterprise Edition
Build number	7600	7600
Service pack	N/A	N/A
File system	NTFS	NTFS
Language	English	English
Graphics		
Vendor and model number	Matrox G200eW	NVIDIA NForce
Graphics memory (MB)	8	32
Ethernet		
Vendor and model number	Broadcom NetXtreme II Gigabit Quad Port	Two embedded NC371i Multifunction Gigabit Network Adapter
Type	PCI Express	Integrated

Figure 6: System configuration information for the two test servers.

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.
