



SQL Server performance comparison: Server consolidation with virtualization on Cisco UCS B200 and B250 platforms for M1 and M2 generations

Executive summary

Cisco Systems®, Inc. (Cisco) commissioned Principled Technologies (PT) to compare the relative Microsoft® SQL Server® 2008 performance of the Intel® Xeon® Processor X5680 and the Intel Xeon Processor X5570 on the following two blade servers using the VMware vSphere 4 (vSphere):

- Cisco UCS B250 Extended Memory Blade Server (Cisco UCS B250 M2 and Cisco UCS B250 M1)
- Cisco UCS B200 Blade Server (Cisco UCS B200 M2 and Cisco UCS B200 M1)

Figure 1 shows the results for these systems.

In a Cisco UCS B250 M1 Extended Memory Blade Server with vSphere, the Intel Xeon Processor X5570, and 384 GB of RAM handled seven more virtual machines (VMs) than did the Cisco UCS B200 M1 Blade Server with vSphere, the Intel Xeon Processor X5570, and 96 GB of RAM, for a total of 53.85 percent more VMs.

In a Cisco UCS B250 M2 Extended Memory Blade Server with vSphere, the Intel Xeon Processor X5680, and 384 GB of RAM handled 15 more virtual machines (VMs) than did the Cisco UCS B200 M2 Blade Server with vSphere, the Intel Xeon Processor X5680, and 96 GB of RAM, for a total of 115.38 percent more VMs.

In a Cisco UCS B250 M2 Extended Memory Blade Server

KEY FINDINGS

- The Cisco UCS B250 M1 Extended Memory Blade Server with the Intel Xeon Processor X5570 and 384 GB of RAM with VMware vSphere handled 53.85% more VMs and handled seven more virtual machines running SQL Server 2008 than did the Cisco UCS B200 M1 Blade Server with the Intel Xeon Processor X5570 and 96 GB of RAM. (See Figure 1.)
- The Cisco UCS B250 M2 Extended Memory Blade Server with the Intel Xeon Processor X5680 and 384 GB of RAM with VMware vSphere handled 115.38% more VMs and handled 15 more virtual machines running SQL Server 2008 than did the Cisco UCS B200 M2 Blade Server with the Intel Xeon Processor X5680 and 96 GB of RAM. (See Figure 1.)
- The Cisco UCS B250 M2 Extended Memory Blade Server with the Intel Xeon Processor X5680 with VMware vSphere handled 40.00% more VMs and handled eight more virtual machines running SQL Server 2008 than did the Cisco UCS B250 M1 Extended Memory Blade Server with the Intel Xeon Processor X5570. (See Figure 1.)

Microsoft SQL Server 2008 workload results Number of virtual servers supported

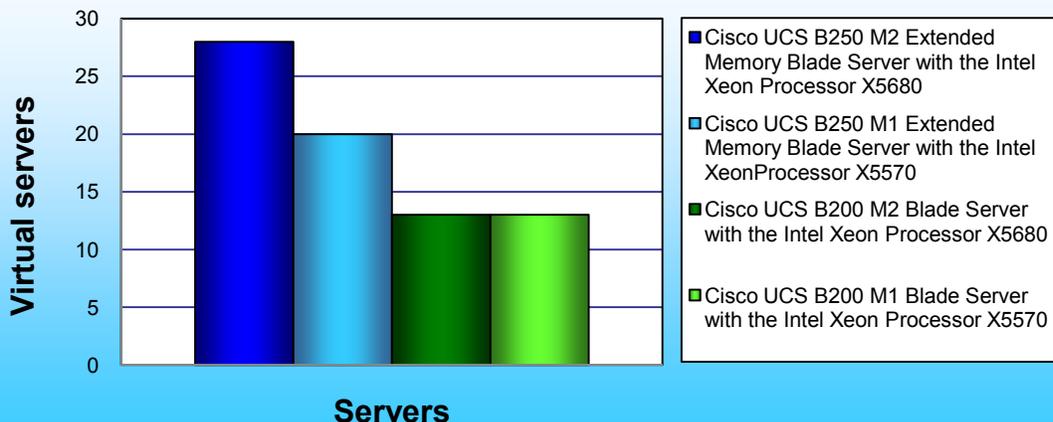


Figure 1: Blade server comparison results in peak number of VMs. Higher numbers are better.

with vSphere, the Intel Xeon Processor X5680, and 384 GB of RAM handled eight more virtual machines (VMs) than did the Cisco UCS B250 M1 Blade Server with vSphere, the Intel Xeon Processor X5570, and 384 GB of RAM, for a total of 40.00 percent more VMs.

We define the peak number of VMs per server as the maximum number of concurrent VMs under load where the average score across the VMs had the performance output in orders per minute (OPM) as the baseline. We used a baseline throughput of 5,135 OPM. We based the baseline throughput on the capacities of a 3-year-old legacy server. See the Workload section for details. The average throughput across the total number of VMs on each test server had to score at least 5,135 OPM. We ensured the maximum number of concurrent VMs on each system by increasing VMs until either the average score fell below the baseline or the total throughput for the server dropped. We then decreased the number of concurrent VMs by one for the final results. We explain in more detail the method of getting this baseline score in the Workload section below.

In our testing, the Cisco UCS B250 M2 with Intel Xeon Processor X5680 and 384 GB of RAM ran 28 concurrent VMs with an average OPM result over the baseline result, while the Cisco UCS B250 M1 with Intel Xeon Processor X5570 and 384 GB of RAM could only run 20 concurrent VMs before we saw a drop in the total OPM. Both the Cisco UCS B200 M2 Blade Server with vSphere and the Intel Xeon Processor X5680 with 96 GB of RAM and Cisco UCS B200 M1 Blade Server with vSphere and the Intel Xeon Processor X5570 with 96 GB of RAM ran 13 VMs with an average OPM result over the baseline.

As Figure 2 shows, in a Cisco UCS B250 M2 Extended Memory Blade Server with vSphere, the Intel Xeon Processor X5680, and 384 GB of RAM delivered a total of 144,306 OPM across all 28 of its VMs versus the UCS B250 M1 Extended Memory Blade Server with vSphere, the Intel Xeon Processor X5570, and 384 GB of RAM's total of 104,236 OPM across its 20 VMs, for a total of 38.44 percent more throughput. We tested at the 5, 10, 15, 20, 25, 27, and 28 VM data points for the Cisco UCS B250 M2. The graph shows interpolated values for the VM counts between those points.

As Figure 2 also shows, in a Cisco UCS B200 M2 Blade Server with vSphere, the Intel Xeon Processor X5680, and 96 GB of RAM delivered a total of 75,170 OPM across all 13 of its VMs versus the Cisco UCS B200 M1 Blade Server with vSphere, the Intel Xeon Processor X5570, and 96 GB of RAM's total of 72,132 OPM across its 13 VMs, for a total of 4.21 percent more throughput. We tested at the 5, 10, 12, and 13 VM data points for the Cisco UCS B200 M2. The graph shows interpolated values for the VM counts between those points.

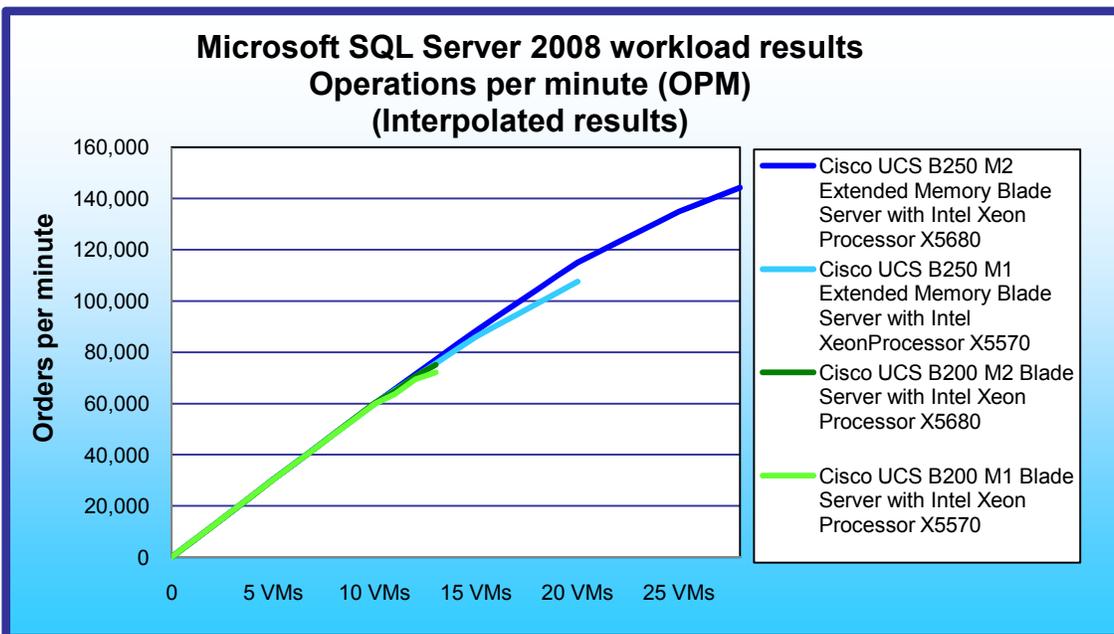


Figure 2: Blade server comparison results in operations per minute. Higher numbers are better.

Workload

To build the workload, we used DVD Store Version 2 (DS2). DS2 is an open-source simulation of an online e-commerce DVD store, where customers log in, browse, and order products. DS2 has database components and Web server components, and includes driver programs that put heavy loads on these components. We used the included driver program to stress the database component. Each blade server under test ran multiple Microsoft SQL Server 2008 workloads, one per VM. One copy of the DS2 database was on each VM, with a 10GB database per VM. By combining multiple virtual machines, all running a CPU, memory, and disk-intensive workload, the test placed a heavy load on both the server and the storage. Thus, it measured the performance of the blade server, especially as it relates to an active online transaction processing (OLTP) environment. The main DS2 metric is OPM, which the driver program calculates and reports via the Microsoft Windows Performance Monitor utility on the client machines. We used the output from the driver program to record OPM, and we report the last OPM score the benchmark reported.

We found the number of OPM on a 3-year-old legacy HP server running a typical workload with under 20 percent processor utilization and used this for our reference score of 5,135 OPM. The legacy HP server was a dual-socket server with two Intel Xeon 3.6GHz processors, 4 GB of RAM and two onboard NICs using one tray of external storage. We used the reference score as our minimum acceptable average score for all four test servers.

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.

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

Each client machine ran a single instance of DS2, with 32 threads. This simulated a heavily loaded environment; the load-generating client VMs ran with 0.3 seconds think time, blasting requests as quickly as the blade servers could handle them.

The DS2 driver application creates an OPM performance counter on the client VM. While the DVD Store client application outputs OPM at 10-second intervals visually, we chose to collect the OPM metric via a performance monitor counter on each client VM at 1-second intervals.

We ran this workload on all virtual machines simultaneously for 30 minutes. We considered a blade server to have delivered acceptable performance if the total OPM for the server did not decrease and if the average OPM result across all clients at the end of the test was above the baseline of 5,135 OPM.

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

Figure 3 shows the settings we used on VMs on all four test servers. Refer to the Test methodology section below for detailed steps of how we set up the VMs. In this test, it is generally the case that the more RAM the blade server can hold, the more VMs the blade server can run. As such, the Cisco UCS B250 M2 Extended Memory Blade Server with vSphere, the Intel Xeon Processor X5680 and the Cisco UCS B250 M1 Extended Memory Blade Server with vSphere, the Intel Xeon Processor X5570 were able to run more VMs with their 384 GB of RAM than the Cisco UCS B200 M2 Blade Server with vSphere, the Intel Xeon Processor X5680 and Cisco UCS B200 M1 Blade Server with vSphere, the Intel Xeon Processor X5570 with their 96 GB of RAM.

VM specifications	VM settings
vCPU	1
Memory (GB)	8
Virtual NIC type	VMXNET 3
Number of virtual disks	3 (OS, SQL Logs, SQL database)

Figure 3: VM configuration information we used for testing.

Test results

Figure 4 shows the peak number of virtual machines each test server could handle during testing before dipping below the acceptable threshold and the minimum client score from the median run.

Figure 4 also shows the OPM for each virtual system and the total OPM for each test server. Note that, for both Cisco blade servers, the Intel Xeon Processor X5680 produced more OPM than the older Intel Xeon Processor X5570. This was true even when the performance was constrained by memory.

To determine the maximum number of simultaneous virtual machines each test server could handle, we added virtual machines until the average OPM score dropped below the 5,135 threshold or the total throughput decreased. We then subtracted one VM to identify the total number of VMs the blade servers could handle. We performed three runs at the total number to verify the scores met these standards.

	Cisco UCS B250 M2 Extended Memory Blade Server with Intel Xeon Processor X5680	Cisco UCS B250 M1 Extended Memory Blade Server with Intel Xeon Processor X5570	Cisco UCS B200 M2 Blade Server with Intel Xeon Processor X5680	Cisco UCS B200 M1 Blade Server with Intel Xeon Processor X5570
Virtual machine 1	5,152	5,204	5,771	5,409
Virtual machine 2	5,168	5,225	5,875	5,288
Virtual machine 3	5,147	5,219	5,730	5,422
Virtual machine 4	5,162	5,245	5,786	5,718
Virtual machine 5	5,157	5,192	5,849	5,520
Virtual machine 6	5,119	5,181	5,713	5,665
Virtual machine 7	5,140	5,176	5,926	5,479
Virtual machine 8	5,159	5,243	5,885	5,342
Virtual machine 9	5,155	5,184	5,732	5,807
Virtual machine 10	5,197	5,244	5,737	5,610
Virtual machine 11	5,152	5,187	5,717	5,543
Virtual machine 12	5,181	5,240	5,772	5,598
Virtual machine 13	5,144	5,183	5,677	5,731
Virtual machine 14	5,174	5,240		
Virtual machine 15	5,181	5,198		
Virtual machine 16	5,180	5,264		
Virtual machine 17	5,119	5,194		
Virtual machine 18	5,174	5,201		
Virtual machine 19	5,159	5,185		
Virtual machine 20	5,176	5,231		
Virtual machine 21	5,158			
Virtual machine 22	5,194			
Virtual machine 23	5,093			
Virtual machine 24	5,171			
Virtual machine 25	5,147			

	Cisco UCS B250 M2 Extended Memory Blade Server with Intel Xeon Processor X5680	Cisco UCS B250 M1 Extended Memory Blade Server with Intel Xeon Processor X5570	Cisco UCS B200 M2 Blade Server with Intel Xeon Processor X5680	Cisco UCS B200 M1 Blade Server with Intel Xeon Processor X5570
Virtual machine 26	5,169			
Virtual machine 27	5,142			
Virtual machine 28	5,036			
Total	144,306	104,236	75,170	72,132

Figure 4: OPM from the median run for each virtual machine for the blade servers. Higher numbers are better.

Figure 5 shows the read, write, and total input/output operations per second (IOPS) for the median run of the four test servers. Higher numbers are better, as they show that the system is handling more requests.

IOPS	Cisco UCS B250 M2 Extended Memory Blade Server with Intel Xeon Processor X5680	Cisco UCS B250 M1 Extended Memory Blade Server with Intel Xeon Processor X5570	Cisco UCS B200 M2 Blade Server with Intel Xeon Processor X5680	Cisco UCS B200 M1 Blade Server with Intel Xeon Processor X5570
Read	3,534	2,794	1,287	1,507
Write	7,814	6,769	4,523	4,381
Total	11,348	9,563	5,810	5,888

Figure 5: IOPS from the median run of the blade servers. Higher numbers are better.

Test methodology

Setting up and configuring the EMC storage

We used an EMC CX4-240 Fibre Channel SAN for testing. The SAN had two storage processors, SP A and SP B. We used the integrated dual-port host bus adapter (HBA) in the Cisco UCS B250 M1 Extended Memory Blade Server. We cabled one HBA port to each SP (A and B) on the SAN to balance the load between SPs. We used five enclosures of disks on the SAN; with one exception, each enclosure had 15 disks. We designated RAID 10 for the virtual machine OS LUN and RAID 10 for all other RAID group configurations.

We created one RAID group on the enclosure that had only 12 disks. We then created one logical unit number (LUN) on this RAID group for the virtual hard drives (VHDs) that would hold the operating systems for the virtual machines. We created eight RAID groups for SQL Server database data. Each of these eight RAID groups was composed of six disks. We then created one 256GB LUN in each RAID group, for a total of eight LUNs dedicated to virtual machines to hold SQL Server database data. 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.

We also created two more RAID groups for SQL Server Log data. Each of these RAID groups was composed of four disks. We then created one 536GB LUN in each RAID group, for a total of two LUNs dedicated to virtual machines to hold SQL Server log data. We assigned one log d 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.

Figure 6 shows the configuration that we used to connect the EMC CLARiiON CX4-240 to both Cisco blade servers for testing.

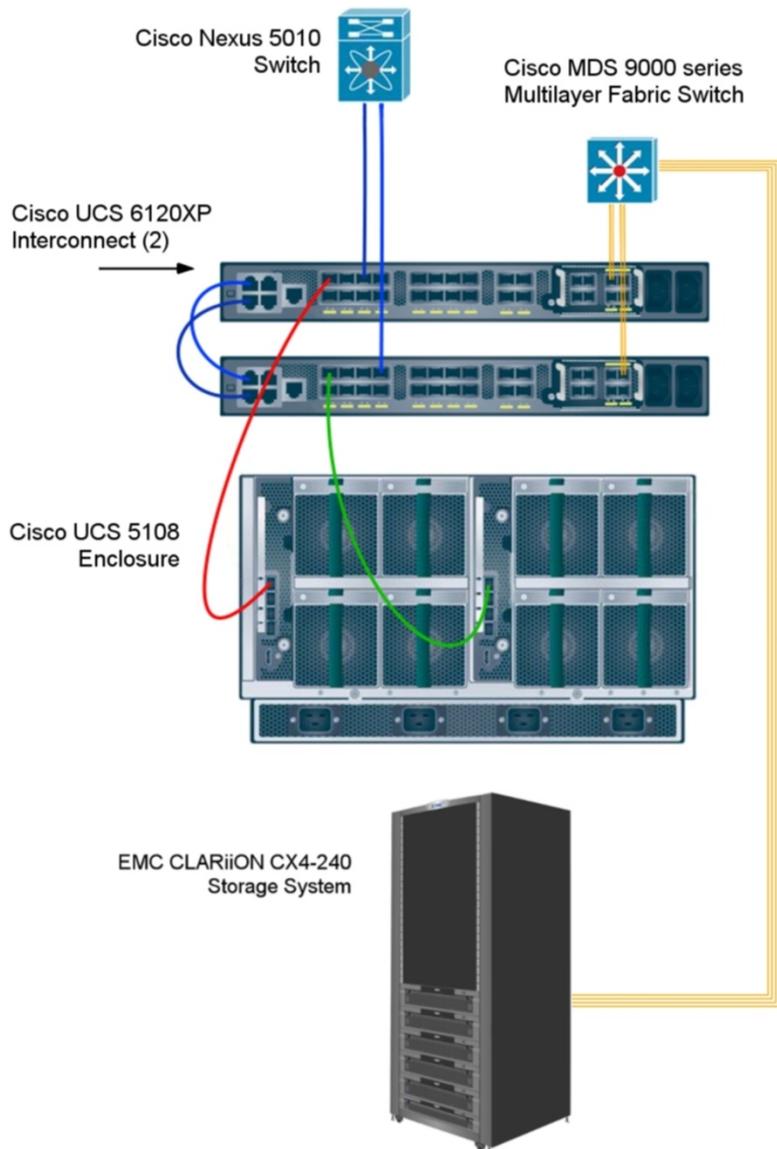


Figure 6: Our test bed configuration.

Setting up the blade servers

We installed a fresh copy of VMware vSphere 4 on both Cisco blade 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 directory 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 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 (64-bit). Click Next.
10. Choose one virtual processor. Click Next.
11. Choose 8192MB 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. See the storage section for more details.
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 22GB 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.
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 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. 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. Click Next.
19. Click the checkbox to accept the license terms. 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. 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 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 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 orders per minute (OPM) each virtual database server returns. 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 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.

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 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. 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 – Blade server configuration information

Figure 7 provides detailed configuration information about the blade servers.

Servers	Cisco UCS B200 M2 Blade Server and Cisco UCS B200 M1 Blade Server	Cisco UCS B250 M2 Extended Memory Blade Server and Cisco UCS B250 M1 Extended Memory Blade Server
Enclosure		
Blade enclosure	Cisco UCS 5108	Cisco UCS 5108
Power supplies		
Total number	4	4
Wattage of each (W)	2,500	2,500
Cooling fans		
Total number	8	8
Dimensions (h x w) of each	3 5/8" x 5 1/2"	3 5/8" x 5 1/2"
General processor setup		
Number of processor packages	2	2
Number of cores per processor package	4	4
Number of hardware threads per core	2	2
Intel Xeon Processor X5680 (Cisco UCS B200 M2 Blade Server and Cisco UCS B250 M2 Extended Memory Blade Server)		
Vendor	Intel	Intel
Name	Xeon X5680	Xeon X5680
Stepping	C0	C0
Socket type	LGA 1366	LGA 1366
Core frequency (GHz)	3.33	3.33
Bus frequency	6.4 GT/s (QPI)	6.4 GT/s (QPI)
L1 cache (KB)	32+32 per core	32+32 per core
L2 cache (KB)	6 x 256 KB	6 x 256 KB
L3 cache (MB)	12	12
Thermal design power (TDP, in watts)	130	130
Intel Xeon Processor X5570 (Cisco UCS B200 M2 Blade Server and Cisco UCS B250 M1 Extended Memory Blade Server)		
Vendor	Intel	Intel
Name	Xeon X5570	Xeon X5570
Stepping	D0	D0
Socket type	LGA 1366	LGA 1366
Core frequency (GHz)	2.93	2.93
Bus frequency	6.4 GT/s (QPI)	6.4 GT/s (QPI)
L1 cache (KB)	128 KB	128 KB
L2 cache (KB)	4 x 256 KB	4 x 256 KB
L3 cache (MB)	8	8
Thermal design power (TDP, in watts)	95	95
Platform		
Vendor and model number	Cisco B200 M1 Blade Server	Cisco B250 M1 Extended Memory Blade Server

Servers	Cisco UCS B200 M2 Blade Server and Cisco UCS B200 M1 Blade Server	Cisco UCS B250 M2 Extended Memory Blade Server and Cisco UCS B250 M1 Extended Memory Blade Server
Motherboard model number	N20-B6625-1	N20-B6625-2
Motherboard chipset	Intel 5520	Intel 5520
BIOS name and version	Cisco S5500.1.2.0.4.012820101602	Cisco S5500.1.2.0.3.020320102313
BIOS settings	Default	Default
Memory modules		
Total RAM in system (GB)	96	384 (8GB x 48)
Number of types of memory modules	1	1
Speed in the system currently running @ (MHz)	1,333	1,333
Timing/Latency (tCL-tRCD-iRP-tRASmin)	9-9-9-27	9-9-9-27
Vendor and model number	Samsung M393B5170EH1 – CH9	Samsung M393B5170EH1 – CH9
Type	DDR3 PC3-10600R	DDR3 PC3-10600R
Speed (MHz)	1,333	1,333
Size (GB)	8	8
Number of RAM modules	12	48
Chip organization	Double-sided	Double-sided
Hard disk		
Vendor and model number	Seagate ST9146802SS	Seagate ST9300603SS
Number of disks in system	2	2
Size (GB)	146	300
Buffer size (MB)	16	16
RPM	10,000	10,000
Type	SAS	SAS
Controller	LSI Logic SAS1064E	LSI Logic SAS1064E
Operating system		
Name	vSphere 4.0.0 Update 1	ESX 4.0.0 Update 1
Build number	208167	208167
File system	VMFS	VMFS
Kernel	2.6.18-128.ESX	2.6.18-128.ESX
Language	English	English
Network card/subsystem		
Vendor and model number	Intel 82958BE 10 Gigabit AF Dual Port	Intel 82958BE 10 Gigabit AF Dual Port
Type	Integrated	Integrated
Fibre adapter		
Vendor and Model number	QLogic QLA2432	QLogic QLA2432
Type	PCI-Express	PCI-Express
USB ports		
Number	0	0
Type	N/A	N/A

Figure 7: Detailed configuration information for the blade servers.

Appendix B – OPM, CPU and memory usage

Figure 8 provides detailed information about the CPU and memory usage for each configuration. Note that this table includes data for the VM count that failed the requirements of the test.

Cisco UCS B250 M2 Extended Memory Blade Server with the Intel Xeon Processor X5680 and 384 GB of RAM			
Number of VMs	OPM	CPU utilization	Memory utilization
5	30,327	18.07%	11.24%
10	60,010	37.69%	21.50%
15	88,482	61.15%	31.77%
20	115,088	82.07%	42.02%
25	134,987	96.64%	52.27%
27	141,164	98.04%	56.38%
28	144,306	97.37%	58.44%
29	145,815	98.34%	60.49%
Cisco UCS B250 M1 Extended Memory Blade Server with the Intel Xeon Processor X5570 and 384 GB of RAM			
Number of VMs	OPM	CPU utilization	Memory utilization
5	30,312	27.73%	11.24%
10	59,927	58.83%	21.51%
15	86,059	91.46%	31.76%
20	107,586	99.17%	42.04%
21	105,555	99.22%	44.08%
Cisco UCS B200 M2 Blade Server with the Intel Xeon Processor X5680 and 96 GB of RAM			
Number of VMs	OPM	CPU utilization	Memory utilization
5	30,184	18.44%	42.82%
10	60,141	36.21%	83.84%
12	70,677	49.72%	95.13%
13	75,170	56.99%	95.58%
14	65,435	42.92%	97.60%
Cisco UCS B200 M1 Blade Server with the Intel Xeon Processor X5570 and 96 GB of RAM			
Number of VMs	OPM	CPU utilization	Memory utilization
5	30,288	27.16%	42.79%
10	59,806	57.28%	83.81%
12	69,432	78.14%	93.02%
13	72,132	88.01%	96.43%
14	55,872	58.18%	97.71%

Figure 8: OPM, CPU utilization and memory utilization for the blade servers.

About Principled Technologies

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

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

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

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



Principled Technologies, Inc.
1007 Slater Rd., Suite 250
Durham, NC 27703
www.principledtechnologies.com
info@principledtechnologies.com

Principled Technologies is a registered trademark of Principled Technologies, Inc.
All other product names are the trademarks of their respective owners.

Disclaimer of Warranties; Limitation of Liability:

PRINCIPLED TECHNOLOGIES, INC. HAS MADE REASONABLE EFFORTS TO ENSURE THE ACCURACY AND VALIDITY OF ITS TESTING, HOWEVER, PRINCIPLED TECHNOLOGIES, INC. SPECIFICALLY DISCLAIMS ANY WARRANTY, EXPRESSED OR IMPLIED, RELATING TO THE TEST RESULTS AND ANALYSIS, THEIR ACCURACY, COMPLETENESS OR QUALITY, INCLUDING ANY IMPLIED WARRANTY OF FITNESS FOR ANY PARTICULAR PURPOSE. ALL PERSONS OR ENTITIES RELYING ON THE RESULTS OF ANY TESTING DO SO AT THEIR OWN RISK, AND AGREE THAT PRINCIPLED TECHNOLOGIES, INC., ITS EMPLOYEES AND ITS SUBCONTRACTORS SHALL HAVE NO LIABILITY WHATSOEVER FROM ANY CLAIM OF LOSS OR DAMAGE ON ACCOUNT OF ANY ALLEGED ERROR OR DEFECT IN ANY TESTING PROCEDURE OR RESULT.

IN NO EVENT SHALL PRINCIPLED TECHNOLOGIES, INC. BE LIABLE FOR INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH ITS TESTING, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL PRINCIPLED TECHNOLOGIES, INC.'S LIABILITY, INCLUDING FOR DIRECT DAMAGES, EXCEED THE AMOUNTS PAID IN CONNECTION WITH PRINCIPLED TECHNOLOGIES, INC.'S TESTING. CUSTOMER'S SOLE AND EXCLUSIVE REMEDIES ARE AS SET FORTH HEREIN.