PROWESS

Behind the Report: Newer Dell EMC™ PowerEdge™ Servers Significantly Increase Microsoft® SQL Server® Performance

Workload Description

For this testing, Prowess engineers used HammerDB on virtual machines (VMs) distributed across two Dell EMC™ PowerEdge™ C6420 and Dell EMC PowerEdge C6520 server clusters.

Cluster Configuration

Model Name	Dell EMC™ PowerEdge™ C6520	Dell EMC™ PowerEdge™ C6420
CPU	Intel® Xeon® Gold 6330 processor at 2.00 GHz	Intel® Xeon® Gold 6130 processor at 2.10 GHz
Number of CPUs	2	2
Cores/Threads Per CPU	28	16
Total Number of Cores/Threads	56	32
Frequency (Base/SCT/MCT)	2.00 GHz, max 4.00 GHz	2.10 GHz
Storage Controller 01	Dell™ PowerEdge RAID Controller (PERC) H345 PCIe® 3.0 and 12 Gb/s SAS	Dell™ PERC H330 Mini (Mezzanine) PCIe® 3.0 and 12 Gb/s SAS
Disk	KIOXIA KPM6XRUG3T84 3.84 TB	Dell™ 3.84 TB TLC SATA SSD
Number of Disks	2	2
Storage Controller 02	Not applicable (N/A)	Not applicable (N/A)
Disk	Micron® MTFDDAV240TDU 240 GB	N/A
Number of Disks	1	N/A
Installed Memory	512 GB	256 GB
Memory DIMM	32 GB 3,200 MT/s dual-rank ECC DDR4	16 GB 2,933 MT/s dual-rank ECC DDR4
Memory Speed Tested	2,933 MT/s	2,666 MT/s
Number of Memory DIMMs	16	16
PCIe Slots	Up to 4 PCIe® 4.0	Up to 4 PCIe® 3.0
OS	VMware ESXi™	VMware ESXi™
OS Version	7.0.2	7.0.2
OS Kernel	17867351	17867351
BIOS Version	1.1.3	2.10.2

Note: BIOS settings turned to Performance mode on each server in both clusters. Storage disks configured for RAID 1 on each server in both clusters; drivers configured for RAID 1 in BIOS using the default RAID settings.

VMware vCenter® Configuration

1. Hosts were added to VMware vCenter®, and a cluster subsequently created for each of the two server models.
2. Each host was configured in vCenter for Performance mode.
3. Each host was added to a distributed switch in order to manage the networking; both 25 Gb/s network cards were added to the distributed switch for each server.
4. Three VMkernel adapters were configured: two on different VLANs for management and one for VMware vSphere® vMotion®.
5. A single VMFS6 volume was created on each solid-state drive (SSD) R1 mirror set, which was named according to the host IP. These datastores were used to hold the VMs on each host for performance testing.
6. Each host was configured with a single iSCSI virtual adapter to connect to two storage volumes and allow two shared datastores for high availability (HA) and disaster-recovery services (DRS) to be enabled. However, these volumes were not used in testing.

VM Configuration

Model Name	Dell EMC™ PowerEdge™ C6520	Dell EMC™ PowerEdge™ C6420
Operating System (OS)	Windows Server® 2019 Datacenter Eval	Windows Server® 2019 Datacenter Eval
OS Version	1809	1809
OS Kernel	17763.2114	17763.2114
Disk 1—Operating System	50 GB	50 GB
Disk 2—Database Files	80 GB	80 GB
Disk 3—Log Files	20 GB	20 GB
CPU	Intel® Xeon® Gold 6330 processor at 2.00 GHz	Intel® Xeon® Gold 6130 processor at 2.10 GHz
Number of CPUs	4	4
Cores per CPU	2	2
Cores Total	8 vCPUs	8 vCPUs
Memory	32 GB	32 GB
Microsoft® SQL Server® Version	Microsoft® SQL Server® 2019 Enterprise Evaluation (64-bit) 15.0.2080.9	Microsoft® SQL Server® 2019 Enterprise Evaluation (64-bit) 15.0.2080.9
HammerDB	4.2	4.2

HammerDB Workload

The workload for HammerDB took into account the number of virtual CPUs (vCPUs) configured on the VM and the amount of memory assigned to each VM. There were 54 warehouses used in the testing.

HammerDB Settings

TPROC-C Microsoft® SQL Server® Database	TPC-C®
TPROC-C Driver Script	Timed Driver Script
Ramp-up Time, in Minutes	2
Test Duration, in Minutes	20
Use All Warehouses	Checked
Virtual Users	40

VM Creation and Setup

Prowess engineers created a VM to clone and deploy as many clones as possible to each node of the Dell EMC PowerEdge C6420 and Dell EMC PowerEdge C6520 server clusters.

Create a New VM

1. Launch VMware vCenter Server®.
2. Right-click the cluster node.
3. Select **New Virtual Machine**.
4. On the **Select a creation type** page, select **Create a New Virtual Machine**, and then click **Next**.
5. On the **Select a name and folder** page, select **Select a name and folder**, provide a name for the VM (for example, **SQL_BASE**), and leave the default location as is.
6. On the **Select a compute resource** page, select **Select a compute resource**, leave the default location as is, and then click **Next**.
7. On the **Select storage** page, select the storage associated with the previously selected cluster, and then click **Next**.
8. On the **Select compatibility** page, select compatibility, and then click **Next**.
9. On the **Select a guest OS** page, select the following:
   - **Guest OS Family**: **Windows**
   - **Guest OS Version**: **Microsoft Windows Server 2019 (64-bit)**
10. Click **Next**.
11. On the **Customize hardware** page, select the following settings:
    - **CPU**: **8**
    - **Memory**: **32**
    - **Memory.Reservation**: **32768 MB**
    - **Memory.Limit**: **32768 MB**
12. Select **New CD/DVD Drive**, click the drop-down box, select **Datastore ISO store**, locate the Windows Server 2019 ISO file, and then click **OK**.
13. Select **Connect At Power On**.

14. Select **New Hard Disk**, and then reduce the size of the **Virtual Machine Disk (VMDK)** to **50 GB**.
15. Select **ADD NEW DEVICE**, and then select **Hard Disk**.
16. Reduce the size of the **VMDK** to **80 GB**.
17. Select **ADD NEW DEVICE**, and then select **Hard Disk**.
18. Reduce the size of the **VMDK** to **20 GB**.
19. Click the drop-down box for **New Network**, and then select **Browse**.
20. Select your network (example: **VM-LAB-24**).
21. Click **OK**.
22. Click **Next**.
23. On the **Ready to complete** page, click **Finish**.
24. After the VM has been created, click the VM **SQL_BASE**.

Install and Set Up Windows Server 2019 on the VM

1. Click **Launch Remote Console**.
2. Click **Open VMware Remote Console**.
3. Click the green start button.
4. Press any key when the system prompts you to boot to CD/DVD.
5. Click **Next**.
6. Click **Install now**.
7. Click **Windows Server 2019 Datacenter Evaluation (Desktop Experience)**.
8. Click **Next**.
9. Accept the license terms, and then click **Next**.
10. Click **Custom: Install Windows only (Advanced)**.
11. Make sure that **Drive 0** is highlighted, and then click **Next**.
12. Wait for the installation process to finish.
13. Type an administrator password in both password fields.
14. Click **Finish**.
15. Click the **Send Ctrl+Alt+Del to virtual machine** button from the toolbar.
16. Type your administrator password, and then press **Enter**.
17. Click **Local Server** on the left side of the **Server Manager** dashboard.
18. Click **On** next to **IE Enhanced Security Configuration**.

19. Click **Off** for the **Administrators** section.
20. Click **OK**.
21. Click **Disabled** next to **Remote Desktop**.
22. Click **Allow remote connections to this computer**.
23. Click **OK**.
24. Open the Start menu, search for **Disk Management**, and then press **Enter**.
25. Right-click **Disk 1**.
26. Select **Online**.
27. Right-click **Disk 1** again, and then select **Initialize Disk**.
28. Click **OK**.
29. Right-click in the **Unallocated** area next to **Disk 1**.
30. Select **New Simple Volume**.
31. Click **Next**.
32. Click **Next**.
33. Set the drive letter or leave the default setting as is.
34. Click **Next**.
35. Name the **Volume label**: **Data**.
36. Leave the default values for all the other settings.
37. Click **Next**.
38. Click **Finish**.
39. Repeat the preceding steps for **Disk 2**, and make sure to name the **Volume label**: **Logs**.
40. Open the Start menu, type **Update**, and then press **Enter**.
41. Click **Check for updates**.
42. Let Windows Update find and install any needed updates, and then reboot the VM as necessary.

Install and Set Up SQL Server 2019

1. Open Internet Explorer.
2. Navigate to **www.microsoft.com/en-us/evalcenter/evaluate-sql-server-2019**.
3. Click **EXE**.
4. Click **Continue**.
5. Fill out the form with your information.

6. Click **Continue**.
7. Click **Run**.
8. Select **Download Media** as the installation type.
9. Leave the default values for the other settings, and then click **Download**.
10. After the download is completed, click **Open folder**.
11. Click back to the SQL setup tool, and then click **Close**.
12. Click **Yes**.
13. In the **Download** folder, double-click **SQLServer2019-x64-ENU**.
14. Double-click **Setup**.
15. Click **Installation** in the left pane.
16. Click **New SQL Server stand-alone installation or add to an existing installation**.
17. Select **Use Microsoft Update to check for updates (recommended),** and then click **Next**.
18. On the **Install Rules** screen, click **Next**.
19. Make sure **Evaluation** is selected, and then click **Next**.
20. Accept the license terms and privacy statement, and then click **Next**.
21. Select **Database Engine Services**, and then click **Next**.
22. Leave the default values for the other settings, and click **Next**.
23. Accept the default setting for **Service Accounts**, and then click **Next**.
24. Select **Mixed Mode**, and then fill in the password fields.
25. Click **Add Current User**.
26. Click **Data Directories**.
27. Click the ellipsis next to **User database directory**.
28. Click the **Data** drive, and then click **Make New Folder**.
29. Name the new folder, and then click **OK**.
30. Click the ellipsis next to **User database log directory**.
31. Click the **Logs** drive, and then click **Make New Folder**.
32. Name the new folder, and then click **OK**.
33. Click **Next**.
34. Click **Next** on the **Feature Configuration Rules** screen.
35. Click **Install**.

36. After the installation is completed, click **Close**.
37. Click **Install SQL Server Management Tools**.
38. Click **Download SQL Server Management Studio (SMSS)**.
39. Click **Run**.
40. Choose the install location or leave the default location as is, and then click **Install**.
41. Click **Restart**.
42. After the system has rebooted, press the **Windows** key.
43. Search for **Microsoft SQL Server Management Studio**, and then open it.
44. Click **Connect**.
45. Click **New Query** at the top of the screen.
46. Type the following commands in the query window to set the max server memory:

```
sp_configure 'show advanced options', 1;
GO
RECONFIGURE;
GO
sp_configure 'max server memory', 28800;
GO
RECONFIGURE;
GO
```

47. Click **Execute** and wait for the "Success" message.
48. Close the SQL Server Management console.
49. Open the Start menu, and then search for **Group Policy**.
50. Press **Enter** to open the group policy editor.
51. In the **Local Group Policy Editor** console, expand **Computer Configuration**, and then expand **Windows Settings**.
52. Expand **Security Settings**, and then expand **Local Policies**.
53. Select the **User Rights Assignment** folder. The policies are displayed in the main pane.
54. Double-click **Lock Pages in Memory**.
55. On the **Local Security Setting** tab, select **Add User or Group**.
56. On the **Select Users, Service Accounts, or Groups** screen, add an account that has privileges to run sqlservr.exe.
57. Sign out, and then sign back in to implement the change.
58. Open the Start menu, and then search for and open **ODBC Data Sources (64-bit)**.
59. On the **User DSN** tab, click **Add**.
60. Select **ODBC Driver 17 for SQL Server**.

61. Click **Finish**.
62. Specify a name for the connection in the **Name** field, and specify the hostname of the VM in the **Server** field.
63. Click **Next**.
64. Leave the default values for the other settings, and then click **Next**.
65. Click **Next**.
66. Leave the default values for the other settings, and then click **Finish**.
67. Click **Test Data Source**.
68. Wait for the test to complete successfully, and then click **OK**.
69. Click **OK**, and then click **OK** again.
70. Click **OK** to close the **ODBC Data Source Administrator** screen.

Install and Configure HammerDB

1. Open a web browser, and then navigate to **www.hammerdb.com/download.html**.
2. Click **Release 4.2 for Windows 64-bit Installer**, and then click **Run** to download and run the installer.
3. Click **Next**.
4. Click **I accept the agreement**, and then click **Next**.
5. Leave the default **Installation Directory** setting, and then click **Next**.
6. Click **Next** to start the installation process.
7. After the installation is completed, clear the **View Readme File** check box, and then click **Finish**.
8. After HammerDB opens, close it again.
9. Open **File Explorer**, and then navigate to the folder **C:\Program Files\HammerDB-4.2**.
10. Open the **Config** folder.
11. Right-click the file **mssqlserver**, and then click **Edit**.
12. Make sure that the contents of the **mssqlserver** file match the following:

```
<?xml version="1.0" encoding="utf-8"?>
<mssqlserver>
   <connection>
         <mssqls_server>(local)</mssqls_server>
         <mssqls_linux_server>localhost</mssqls_linux_server>
         <mssqls_tcp>false</mssqls_tcp>
         <mssqls_port>1433</mssqls_port>
         <mssqls_azure>false</mssqls_azure>
         <mssqls_authentication>sql</mssqls_authentication>
         <mssqls_linux_authent>sql</mssqls_linux_authent>
```

```
        <mssqls_odbc_driver>ODBC Driver 17 for SQL Server</mssqls_odbc_driver>
        <mssqls_linux_odbc>ODBC Driver 17 for SQL Server</mssqls_linux_odbc>
        <mssqls_uid>sa</mssqls_uid>
        <mssqls_pass>Password2021!</mssqls_pass>
    </connection>
    <tpcc>
        <schema>
            <mssqls_count_ware>1</mssqls_count_ware>
            <mssqls_num_vu>1</mssqls_num_vu>
            <mssqls_dbase>tpcc</mssqls_dbase>
            <mssqls_imdb>false</mssqls_imdb>
            <mssqls_bucket>1</mssqls_bucket>
            <mssqls_durability>SCHEMA_AND_DATA</mssqls_durability>
        </schema>
        <driver>
            <mssqls_total_iterations>1000000</mssqls_total_iterations>
            <mssqls_raiseerror>false</mssqls_raiseerror>
            <mssqls_keyandthink>false</mssqls_keyandthink>
            <mssqls_checkpoint>false</mssqls_checkpoint>
            <mssqls_driver>timed</mssqls_driver>
            <mssqls_rampup>2</mssqls_rampup>
            <mssqls_duration>20</mssqls_duration>
            <mssqls_allwarehouse>true</mssqls_allwarehouse>
            <mssqls_timeprofile>false</mssqls_timeprofile>
            <mssqls_async_scale>false</mssqls_async_scale>
            <mssqls_async_client>10</mssqls_async_client>
            <mssqls_async_verbose>false</mssqls_async_verbose>
            <mssqls_async_delay>1000</mssqls_async_delay>
        </driver>
    </tpcc>
    <tpch>
        <schema>
            <mssqls_scale_fact>1</mssqls_scale_fact>
            <mssqls_maxdop>2</mssqls_maxdop>
            <mssqls_tpch_dbase>tpch</mssqls_tpch_dbase>
            <mssqls_num_tpch_threads>1</mssqls_num_tpch_threads>
            <mssqls_colstore>false</mssqls_colstore>
        </schema>
        <driver>
            <mssqls_total_querysets>1</mssqls_total_querysets>
            <mssqls_raise_query_error>false</mssqls_raise_query_error>
            <mssqls_verbose>false</mssqls_verbose>
            <mssqls_refresh_on>false</mssqls_refresh_on>
            <mssqls_update_sets>1</mssqls_update_sets>
            <mssqls_trickle_refresh>1000</mssqls_trickle_refresh>
        <mssqls_refresh_verbose>false</mssqls_refresh_verbose>
        </driver>
    </tpch>
</mssqlserver>
```

13. Save and close the file.

Set Up Performance Monitor

1. Open the Start menu, and then search for and open **Performance Monitor.**
2. Expand **Data Collector Sets**.
3. Right-click **User-defined**, select **New**, and then select **Data Collector Set**.
4. Enter a name in the **Name** field.
5. Click **Create manually (Advanced)**, and then click **Next**.
6. Select the **Performance counter**, **Event trace data**, and **System configuration information** check boxes, and then click **Next**.
7. Click **Add**.
8. Select each of the following items in the **Available counters** section, and then click **Add**:

```
Processor Information(_Total)\% Processor Time
Memory\Available MBytes
PhysicalDisk(_Total)\Avg. Disk sec/Read
PhysicalDisk(_Total)\Avg. Disk sec/Write
PhysicalDisk(_Total)\Disk Reads/sec
PhysicalDisk(_Total)\Disk Writes/sec
Processor(_Total)\% Processor Time
SQLServer:General Statistics\User Connections
SQLServer:SQL Statistics\Batch Requests/sec
SQLServer:SQL Statistics\SQL Compilations/sec
SQLServer:SQL Statistics\SQL Re-Compilations/sec
System\Processor Queue Length
Process(_Total)\% Privileged Time
Process(_Total)\% User Time
Process(_Total)\IO Read Bytes/sec
Process(_Total)\IO Write Bytes/sec
Processor(_Total)\% Privileged Time
Processor(_Total)\% User Time
Process(_Total)\% Processor Time
Average Disk Sec/Transfer
Average Disk Queue Length
Current Disk Queue Length
%Disk Time
%Idle Time
Buffer Manager: Page Reads/Sec
Page Writes/Sec
Current Disk Queue Length
Avg. Disk Bytes / Transfer
Avg. Disk sec / Transfer
Disk Bytes / sec
Disk Transfers / sec
```

9. Click **OK**.
10. Click **Finish**.

Create the HammerDB Database

1. Open the Start menu, and then search for and open **Microsoft SQL Server Management Studio**.
2. Sign in.
3. Click **New Query.**
4. In the query window, copy or type the following script to create a 5 GB database. This script uses the Data and Logs folders created earlier for storing the database and log files.

```
CREATE DATABASE [tpcc]
CONTAINMENT = NONE
ON PRIMARY
( NAME = N'tpcc', FILENAME = N'E:\Data\tpcc.mdf' , SIZE = 5GB , FILEGROWTH = 10%),
( NAME = N'tpcc1', FILENAME = N'E:\ Data\tpcc1.ndf' , SIZE = 2GB ,
FILEGROWTH = 10%)
LOG ON
( NAME = N'tpcc_log', FILENAME = N'F:\Logs\tpcc_log.ldf' , SIZE = 2048KB ,
FILEGROWTH = 10%),
( NAME = N'tpcc_log1', FILENAME = N'F:\Logs\tpcc_log1.ldf' , SIZE = 1024KB ,
FILEGROWTH = 10%),
( NAME = N'tpcc_log2', FILENAME = N'F:\Logs\tpcc_log2.ldf' , SIZE = 1024KB ,
FILEGROWTH = 10%),
( NAME = N'tpcc_log3', FILENAME = N'F:\Logs\tpcc_log3.ldf' , SIZE = 1024KB ,
FILEGROWTH = 10%)
GO
ALTER DATABASE [tpcc] SET COMPATIBILITY_LEVEL = 150
GO
ALTER DATABASE [tpcc] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [tpcc] SET ANSI_NULLS OFF
GO
ALTER DATABASE [tpcc] SET ANSI_PADDING OFF
GO
ALTER DATABASE [tpcc] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [tpcc] SET ARITHABORT OFF
GO
ALTER DATABASE [tpcc] SET AUTO_CLOSE OFF
GO
ALTER DATABASE [tpcc] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [tpcc] SET AUTO_CREATE_STATISTICS ON(INCREMENTAL = OFF)
GO
ALTER DATABASE [tpcc] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [tpcc] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
ALTER DATABASE [tpcc] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [tpcc] SET CONCAT_NULL_YIELDS_NULL OFF
```

```
GO
ALTER DATABASE [tpcc] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [tpcc] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [tpcc] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [tpcc] SET DISABLE_BROKER
GO
ALTER DATABASE [tpcc] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
ALTER DATABASE [tpcc] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [tpcc] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [tpcc] SET READ_COMMITTED_SNAPSHOT OFF
GO
ALTER DATABASE [tpcc] SET READ_WRITE
GO
ALTER DATABASE [tpcc] SET RECOVERY FULL
GO
ALTER DATABASE [tpcc] SET MULTI_USER
GO
ALTER DATABASE [tpcc] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [tpcc] SET TARGET_RECOVERY_TIME = 60 SECONDS
GO
ALTER DATABASE [tpcc] SET DELAYED_DURABILITY = DISABLED
GO
USE [tpcc]
GO
ALTER DATABASE SCOPED CONFIGURATION SET LEGACY_CARDINALITY_ESTIMATION = Off;
GO
ALTER DATABASE SCOPED CONFIGURATION FOR SECONDARY SET LEGACY_CARDINALITY_ESTIMATION =
Primary;
GO
ALTER DATABASE SCOPED CONFIGURATION SET MAXDOP = 0;
GO
ALTER DATABASE SCOPED CONFIGURATION FOR SECONDARY SET MAXDOP = PRIMARY;
GO
ALTER DATABASE SCOPED CONFIGURATION SET PARAMETER_SNIFFING = On;
GO
ALTER DATABASE SCOPED CONFIGURATION FOR SECONDARY SET PARAMETER_SNIFFING = Primary;
GO
ALTER DATABASE SCOPED CONFIGURATION SET QUERY_OPTIMIZER_HOTFIXES = Off;
GO
ALTER DATABASE SCOPED CONFIGURATION FOR SECONDARY SET QUERY_OPTIMIZER_HOTFIXES =
Primary;
GO
USE [tpcc]
```

```
GO
IF NOT EXISTS (SELECT name FROM sys.filegroups WHERE is_default=1 AND name = N'PRIMARY')
ALTER DATABASE [tpcc] MODIFY FILEGROUP [PRIMARY] DEFAULT
GO
```

5. Click **Execute**, and then wait for the “Success” message.
6. Close SQL Server Management Studio.
7. Open HammerDB by going to the folder **C:\Program Files\HammerDB-4.2** and double-clicking the **hammerdb.bat** file.
8. In the **Benchmark** pane, double-click **SQL Server**.
9. Click **OK**.
10. Click **OK**.
11. Expand **TPROC-C**.
12. Expand **Schema Build**, and then double-click **Options**.
13. Set **Number of Warehouses** to **54**.
14. Set **Virtual Users to build schema** to **40**.
15. Click **OK**.
16. Double-click **Build**.
17. Click **Yes**.
18. Wait for the build-out process to complete, and then close HammerDB.
19. Click the Start menu, and then search for and open **Notepad**.
20. Copy or type the following script into the Notepad window:

```
#!/bin/tclsh
proc runtimer { seconds } {
set x 0
set timerstop 0
while {!$timerstop} {
incr x
after 1000
   if { ![ expr {$x % 60} ] } {
         set y [ expr $x / 60 ]
         puts "Timer: $y minutes elapsed"
   }
update

if {  [ vucomplete ] || $x eq $seconds } { set timerstop 1 }
   }
```

```
return
}
puts "SETTING CONFIGURATION"
dbset db mssqls
diset tpcc mssqls_driver timed
diset tpcc mssqls_rampup 2
diset tpcc mssqls_duration 20
diset tpcc mssqls_allwarehouse true
vuset iterations 1
vuset showoutput 1
vuset logtotemp 1
vuset timestamps 1
vuset unique 1
foreach z {40} {
puts "$z iteration"
loadscript
vuset vu $z
vucreate
vurun
runtimer 1330
vudestroy
after 1340
}
puts "TESTING COMPLETE"
```

21. Click **Text Documents** next to **Save as type**, and then click **All Files**.
22. Name the file, using a .tcl file-type extension (for example, ***test1.tcl***).
23. Save the file to the **C:\Program Files\HammerDB-4.2** folder.
24. Click **File**, and then click **New**.
25. In the new window, copy or type the following script, changing the last line to reflect the .tcl file name you created in step 22:

```
Try{
   Do{
   $mssql = Get-Service -Name MSSQLSERVER -ErrorAction SilentlyContinue
   }
   While($mssql.Status -ne "Running")
}
Catch{
}
logman start "SQL Data Collector"
start-process cmd -WorkingDirectory "C:\Program Files\HammerDB-4.2\" -argumentlist "/C
hammerdbcli.bat auto test1.tcl"
```

26. Click **File**, and then click **Save As**.
27. Click **Text Documents** next to **Save as type**, and then click **All Files**.

28. Name the file, using a .ps1 file-type extension (for example, ***hammerdb.ps1*)**.

29. Save the file to the **C:\Program Files\HammerDB-4.2** folder.

30. Click **File**, and then click **New**.

31. Paste or type the following script, making sure to change the .ps1 file name to the .ps1 file name you created in step 28.

```
@echo off
powershell.exe -executionpolicy Bypass -command ". 'C:\Program Files\HammerDB-4.2\
hammerdb.ps1'"
```

32. Click **File**, and then click **Save As**.

33. Click **Text Documents** next to **Save as type**, and then click **All Files**.

34. Name the file, using a .bat file-type extension (for example, **run.bat)**.

35. Save the file to your desktop.

36. Close Notepad.

37. Click the Start menu, and then search for and open **Task Scheduler**.

38. Click **Create Task**.

39. Name the task (for example, **Run HammerDB**).

40. Click **Run whether user is logged on or not**.

41. Click **Run with the highest privileges**.

42. Click **Triggers**.

43. Click **New**.

44. Click the drop-down box next to **Begin the task**, and then click **At startup**.

45. Click **OK**.

46. Click **Actions**.

47. Click **New**.

48. Click **Browse**.

49. Click the .bat file that you created in step 34.

50. Click **Open**.

51. Click **OK**.

52. Click **OK** to create the task.

53. Click the Start menu and power off the server using the power button.

This next section will discuss creating a VM template and creating multiple VMs from that template.

1. Within vCenter, right-click the VM created at the beginning of this guide (for example, **SQL_BASE**).
2. Hover over **Template**, and then click **Convert to Template**.
3. Click **YES**.

To create the VMs needed for this testing, Prowess used a PowerShell® script to expedite the process. This process is done on a separate computer that is connected to the same network as vCenter.

Prowess used two clusters named 6420 and 6520, with four datastores each. The steps in the following procedure are based on these clusters (named a such) and datastores.

The first step is to create a CSV document.

1. Open Microsoft® Excel® or the application of your choice to create a CSV document. Prowess used Excel, so these instructions will be based on Excel. There are two documents that will be created—one for the 6420 cluster and the other for the 6520 cluster.

Due to memory constraints, the 6420 cluster only had 20 VMs created, whereas the 6520 cluster had 44. Both clusters had HA set up, which means that for each VM created, there has to be room on another server for that VM, so those resources (memory, CPU cycles, and storage) are then set aside. This essentially halves the amount of memory that is available for running the VMs.

2. For **Column A** in **Name**, type **vm-01**, **vm-02**, and so on, up to **vm-20**. For the 6520 document, type **vm-01-C6520**, **vm-02-C6520**, and so on, up to **vm-44-C6520**.
3. For **Column B** in **Datastore**, type the name of the datastore where the VM will be stored, changing each line to a different datastore as needed. Prowess had four datastores, so the document will reflect that.
4. For **Column C**, type **vCPU**, and then fill that column with **8**.
5. For **Column D**, type **Memory**, and then fill that column with **32**.
6. Save the documents to the folder of your choice, and make sure to remember the location of the folder (for example, **C:/newvm_folder)**.
7. Open the Start menu, type **PowerShell**, and then open PowerShell.
8. After PowerShell opens, connect to vCenter by typing **Connect-VIServer** and the name of the server that's hosting vCenter.
9. Press **Enter**.
10. Copy or type the following script, replacing the first line with the location of the file that you saved earlier. This has to be run twice—once for the 6420 cluster and once for the 6520 cluster. The first line will need to point to the correct CSV file, the **$Template** line will need to be the name of the template created earlier, and the **$Cluster** line will need to reflect the cluster that you are creating the VMs on. The example below is for creating the VMs on the 6520 cluster, using the CSV created for the 6520 cluster.

```
$vms = Import-CSV "C:\newvm\new-vm-6520.csv"

foreach ($vm in $vms) {
$Template ="SQL_Base_template"
$Cluster = "C6520"
$Datastore = Get-Datastore -Name $vm.Datastore
$vCPU = $vm.vCPU
$Memory = $vm.Memory
$VMName = $vm.Name+"-$Cluster"
$location = $vm.location

#Where the VM gets built


New-VM -Name $VMName -Template $Template -ResourcePool (Get-Cluster $Cluster | Get-
ResourcePool) -Location $Location -Datastore $Datastore

Start-Sleep -Seconds 10

#Where the vCPU, memory, and network gets set
$NewVM = Get-VM -Name $VMName
$NewVM | Set-VM -MemoryGB $Memory -NumCpu $vCPU -Confirm:$false
$newVM | Get-VMResourceConfiguration | Set-VMResourceConfiguration -MemReservationMB
$NewVMMemoryMB

$newVM | Get-VMResourceConfiguration | Set-VMResourceConfiguration -MemLimitMB $NewVM.
MemoryMB
}
```

11. After that finishes, make sure to run the script again for the other cluster.

Collect Metrics Results from Host Servers

This next section covers the testing process. HammerDB will start on all the VMs automatically at startup. The data collection will consist of the data collected by HammerDB, vCenter, Performance Monitor, and esxtop. This is all done from a computer that is connected to the same network as vCenter.

1. In vCenter, click one of the host servers from the cluster being tested on.
2. Click **Monitor**.
3. Click **Advanced** in the **Performance** section.
4. Click **Chart Options**.
5. Click **Power** under **Chart Metrics**.
6. Choose **Usage**.
7. Click **Save Options As**.
8. Type **Power**, and then click **OK**.
9. Click **Chart Options** again.

10. Click **CPU**, under **Chart Metrics**, and then click **Usage**.
11. Click **Save Options As**.
12. Type **CPU** and then click **OK**.
13. Open a browser window, and then navigate to **www.chiark.greenend.org.uk/~sgtatham/putty/latest.html**.
14. Click the hyperlink next to **64bit x86** to download puTTY.
15. Click **Save**.
16. After the file is downloaded, run it to install puTTY.
17. Click the Start menu, and then search for and open **puTTY**.
18. Type the IP address of the first server; there will need to be a puTTY session for each server.
19. Click **Open**.
20. Type **root**, and then press **Enter**.
21. For the password, use the password that you used to set up ESXi on the servers, and then press **Enter**.
22. Type **esxtop –b –u –d -10 > filename.csv** (the file name can be any name—for example, **C6420_150_run1.csv**). Do not press **Enter** until all servers are set up in puTTY.
23. Repeat this for all the servers in the first cluster. Prowess's setup consisted of four servers per cluster, so there were four puTTY sessions at a time, due to one cluster being tested at a time. In vCenter, click the cluster being tested.
24. Click **VMs** at the top of the screen.
25. Click the first VM, press **Shift**, and then click the last VM. This will select all the VMs being tested.
26. Right-click any VM, hover over **Power**, and then click **Power On**. This will start up all the selected VMs.
27. Go back to the puTTY sessions, and then press **Enter** to start the monitoring process.
28. Let the servers run for 45 minutes.
29. In vCenter, go to the **VMs** page again and select all the VMs.
30. Right-click any VM, hover over **Power**, and then click **Power Off**. This will shut down all the VMs.
31. In all the puTTY sessions, press **Ctrl+C** to stop monitoring.
32. In vCenter, click one of the host servers in the cluster being tested.
33. Click **Monitor**.
34. Click **Advanced** in the **Performance** section.
35. Click the **Export** button on the right side of the screen.
36. Click **To CSV**.

37. Name the file and save it.
38. Click **CPU usage in %** next to **View**.
39. In the drop-down menu that opens, click **Datastore**.
40. Repeat this process of downloading the performance-data CSV file (steps 35–39) for the following views from the **View** drop-down: **Memory**, **Disk**, **Power**, **Storage Adapters**, and **CPU**.
41. Choose those views, and then save the CSV file, making sure that each CSV file has a unique name.
42. Repeat the steps to download the performance data for each view for each server.

Prowess repeated this testing process two more times, for a total of three runs.

Collect Transactions per Minute (TPM) and Metrics Results from VMs

1. On the system used for testing, open the Start menu, and then search for and open **Remote desktop**.
2. Enter the IP address for one of the VMs. The IP address can be obtained by clicking the VM in vCenter. The IP address is shown on the **Summary** page.
3. After signing in to the VM, open File Explorer and navigate to **C:\Users\Administrator\AppData\Local\Temp**.
4. The files you're looking for will have names starting with "hammerdb," followed by a mix of letters and numbers. The date of the file will let you know which test run the file corresponds to.
5. Copy the **hammerdb** files off the server.
6. Open the Start menu, and then search for and open **Performance Monitor**.
7. Expand **Reports**.
8. Expand **User Defined**.
9. Expand **SQL Data collector**.
10. Click the report that corresponds to the first test run; you can determine this from the date and time of the report.
11. Click the **Open data folder** icon at the top of the screen.
12. Copy the Performance Monitor file off the server. The file will be named **systemname_DataCollector01**.
13. Repeat this process for the other runs.