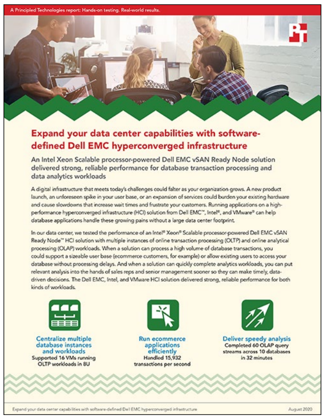

A Principled Technologies report: Hands-on testing. Real-world results.

Expand your data center capabilities with software-defined Dell EMC hyperconverged infrastructure

An Intel Xeon Scalable processor-powered Dell EMC vSAN Ready Node solution delivered strong, reliable performance for database transaction processing and data analytics workloads

A digital infrastructure that meets today's challenges could falter as your organization grows. A new product launch, an unforeseen spike in your user base, or an expansion of services could burden your existing hardware and cause slowdowns that increase wait times and frustrate your customers. Running applications on a high-performance hyperconverged infrastructure (HCI) solution from Dell EMC™, Intel®, and VMware® can help database applications handle these growing pains without a large data center footprint.

In our data center, we tested the performance of an Intel® Xeon® Scalable processor-powered Dell EMC vSAN Ready Node™ HCI solution with multiple instances of online transaction processing (OLTP) and online analytical processing (OLAP) workloads. When a solution can process a high volume of database transactions, you could support a sizeable user base (ecommerce customers, for example) or allow existing users to access your database without processing delays. And when a solution can quickly complete analytics workloads, you can put relevant analysis into the hands of sales reps and senior management sooner so they can make timely, data-driven decisions. The Dell EMC, Intel, and VMware HCI solution delivered strong, reliable performance for both kinds of workloads.

Centralize multiple database instances and workloads
Supported 16 VMs running OLTP workloads in 8U

Run ecommerce applications efficiently
Handled 15,932 transactions per second

Deliver speedy analysis
Completed 60 OLAP query streams across 10 databases in 32 minutes

Expand your data center capabilities with software-defined Dell EMC hyperconverged infrastructure August 2020

The science behind the report:

Expand your data center capabilities with software-defined Dell EMC hyperconverged infrastructure

This document describes what we tested, how we tested, and what we found. To learn how these facts translate into real-world benefits, read the report Expand your data center capabilities with software-defined Dell EMC hyperconverged infrastructure.

We concluded our hands-on testing on June 11, 2020. During testing, we determined the appropriate hardware and software configurations and applied updates as they became available. The results in this report reflect configurations that we finalized on May 1, 2020 or earlier. Unavoidably, these configurations may not represent the latest versions available when this report appears.

Our results

Table 1: Results of our OLTP (TPC-E-like) testing (16 VMs) on the Dell EMC vSAN Ready Node solution with Dell EMC PowerEdge R740 servers. Source: Principled Technologies.

OLTP testing (16 VMs)	
	Median run
Total TPS	15,932.69
Average TPS	995.79
% CPU utilization	66.06
Avg read latency (ms)	0.426
Avg write latency (ms)	7.552

Table 2: Results of our OLAP (TPC-H-like) testing (10 VMs) on the Dell EMC vSAN Ready Node solution with Dell EMC™ PowerEdge™ R740 servers. Source: Principled Technologies.

OLAP testing (10 VMs)		
Median run (minutes)	Single-stream test	Multi-stream test (Six streams)
Average VM time to complete	16.37	31.29
Longest VM time to complete	17.05	32.68
% CPU utilization	20.64	53.03
Avg. read latency (ms)	7.16	24.42

System configuration information

Table 3: Detailed configuration information for the servers under test. Source: Principled Technologies.

Server configuration information	Dell EMC PowerEdge R740
BIOS name and version	Dell 2.5.4
Non-default BIOS settings	Performance mode & LLC Prefetcher enabled
Operating system name and version/build number	VMware ESXi™ 7.0 Standard (Build 15843807)
Date of last OS updates/patches applied	05/01/2020
Power management policy	Performance
Processor	
Number of processors	2
Vendor and model	Intel® Xeon® Gold 6240L
Core count (per processor)	18
Core frequency (GHz)	2.6
Stepping	7
Memory module(s)	
Total memory in system (GB)	384
Number of memory modules	12
Vendor and model	Micron® MTA36ASF4G72PZ-2G9E2
Size (GB)	32
Type	DDR-4 PC4-23400 Dual-Rank
Speed (MHz)	2,933
Speed running in the server (MHz)	2,933
Storage controller	
Vendor and model	Dell HBA330 Adapter
Cache size (MB)	0
Firmware version	16.17.00.05
Driver version	N/A
Local storage (type A)	
Number of drives	4
Drive vendor and model	Intel S4510
Drive size (GB)	1.92
Drive information (speed, interface, type)	6Gbps SATA SSD

Server configuration information	Dell EMC PowerEdge R740
Local storage (type B)	
Number of drives	1
Drive vendor and model	Intel S4610
Drive size (GB)	960
Drive information (speed, interface, type)	6Gbps SATA SSD
Local storage (type C)	
Number of drives	2
Drive vendor and model	Dell 0R7YTT
Drive size (GB)	16
Drive information (speed, interface, type)	Class 10 SD card
Network adapter (type 1)	
Vendor and model	Broadcom® Gigabit Ethernet BCM5720
Number and type of ports	4x 1GbE RJ45
Driver version	17.2.1.0
Network adapter (type 2)	
Vendor and model	Intel Ethernet Network Adapter XXV710-DA2
Number and type of ports	2x 25GbE SFP28
Driver version	i40en
Cooling fans	
Vendor and model	Nidec UltraFlo V60E12BS10M3
Number of cooling fans	6
Power supplies	
Vendor and model	Dell D1100E-S0
Number of power supplies	2
Wattage of each (W)	1,100

How we tested

Testing overview

We created a four-node vSAN cluster of PowerEdge R740 servers with a 25Gb switch for vSAN traffic. For the OLTP testing, we created 16 VMs, each with 8 vCPUs, 64GB of RAM, and a 30-scale TPC-E-like database. Using a separate server as client host with a 25Gb connection to the SUTs, we leveraged Benchmark Factory TPC-E-like workload transactions against each VM's database. Using 20 threads per client, we ran a 30-minute warmup period and a one-hour test period and report both the transactions per second per VM and the total across the cluster.

For the TPC-H-like data warehouse testing, we created 10 VMs each with 16 vCPUs, 120GB RAM, and 300-scale databases. We created the databases using the dbgen tool and used the HammerDB TPC-H-like workload to run a stream of 22 queries against each VM. We ran both a single-stream and multi-stream test, both of which are read-only queries. We report the amount of time it took the single stream to finish as well as the time of the longest stream in the multi-stream test.

Installing ESXi 7.0 on the servers

Changing System Profile to Performance

1. Boot the server.
2. At the POST menu, select F2 for System Setup.
3. Select System BIOS.
4. Select System Profile Settings.
5. Click the drop-down menu for System Profile, and select Performance.
6. Click Back.
7. Click Finish.
8. Click Finish, and reboot the server.

Installing ESXi vSphere 7.0

1. Attach the installation media.
2. Boot the server.
3. At the POST menu, select F11 for Boot Manager.
4. Select One-shot UEFI Boot Menu.
5. Select the installation media.
6. At the VMware Installer screen, press Enter.
7. At the EULA screen, press F11 to Accept and Continue.
8. Under Storage Devices, select the appropriate virtual disk, and press Enter.
9. Select US as the keyboard layout, and press Enter.
10. Enter the root password twice, and press Enter.
11. To start the installation, press F11.
12. After the server reboots, press F2, and enter root credentials.
13. Select Configure Management Network, and press Enter.
14. Select the appropriate network adapter, and click OK.
15. Select IPv4 settings, and enter the desired IP address, subnet mask, and gateway for the server.
16. Select OK, and restart the management network.
17. For the remaining servers, complete steps 1 through 16.

Deploying VMware vCenter Server 7.0

1. On a Windows machine or VM, locate the VMware-VCSA installer image.
2. Mount the image, navigate to the vcsa-ui-installer folder, and double-click win32.
3. Double-click installer.exe
4. Click Install.
5. Click Next.
6. Accept the terms of the license agreement, and click Next.
7. Leave the default vCenter Server with an Embedded Platform Services Controller selected, and click Next.
8. Enter the FQDN or IP address of the host onto which the vCenter Server Appliance will be deployed.
9. Provide the servers credentials, and click Next.
10. At the Configure Network Settings page, configure the network settings for your environment, and click Next.
11. Review your settings, and click Finish.
12. When the deployment completes, click Next.
13. At the Introduction page, click Next.
14. At the Appliance configuration page, select the time synchronization mode and SSH access settings, and click Next. We used a local NTP server and enabled SSH.
15. Select Create a new SSO domain.
16. Provide a password, and confirm it.
17. Provide an SSO Domain name and SSO Site name, and click Next.
18. At the CEIP page, click Next.
19. At the Ready to complete page, click Finish.
20. When the installation completes, click Close.
21. Using the vSphere web client, log into the vCenter server using the credentials previously provided.

Creating a cluster in VMware vCenter and adding hosts

Creating the vSAN cluster

1. Log into vCenter, and navigate to Hosts and Clusters.
2. Select the primary site management vCenter.
3. Right-click the vCenter object, and select New Datacenter…
4. Enter a name for the new datacenter, and click OK.
5. Right-click the new datacenter, and click New Cluster…
6. For the name for the new cluster, enter `vSAN`
7. Click OK.
8. To create an Infra cluster, complete steps 1 through 7.

Adding the hosts

1. Once the cluster has been created, right-click the cluster, and click Add Host.
2. Enter the FQDN or IP address of the first host, and click Next.
3. Enter the root credentials for the server, and click Next.
4. To accept the server's certificate, click Yes.
5. Review the server details, and click Next.
6. Assign the desired license, and click Next.
7. Disable Lockdown mode, and click Next.
8. Click Finish.
9. For the remaining vSAN hosts and the client host, complete steps 1 through 8.

Configuring the vSAN cluster

Configuring the vSAN networking

1. Log into the vCenter web client, and navigate to Hosts and Clusters.
2. Expand the primary site management vCenter, and select the client host.
3. Click the Configure tab.
4. Select VMkernel adapters, and click Add Networking.
5. Select Virtual Machine Port Group for a Standard Switch, and click Next.
6. Select New standard switch, change MTU (Bytes) to 9000, and click Next.
7. Click Add adapters, and select your desired physical adapter. We used a 25GbE connection.
8. Choose a network label, and click Next. We labeled our client traffic network `Priv_Net`
9. Click Finish.
10. In the Hosts and Clusters pane, expand the vSAN cluster.
11. Select the first host, and navigate to the Configure tab.
12. Under Networking, select VMkernel adapters, and click Add Networking.
13. Select VMkernel Network Adapter, and click Next.
14. Select New standard switch, change MTU (Bytes) to 9000, and click Next.
15. Click Add adapters, and select your desired physical adapter. We used a 25GbE connection.
16. For the Network label, type `vSAN` input your VLAN ID (if using), and select vSAN from the Available services below. Click Next.
17. Choose your network settings, and click Next.
18. Click Finish.
19. Click Add Networking.
20. Select VMkernel Network Adapter, and click Next.
21. Choose Select an existing standard switch, browse to the vSwitch created in step 14, click OK, and click Next.
22. For the Network label, type `vMotion` input your VLAN ID (if using), and select vMotion from the Available services below. Click Next.
23. Choose your network settings, and click Next.
24. Click Finish.
25. Click Add Networking.
26. Select Virtual Machine Port Group for a Standard Switch, and click Next.
27. Choose Select an existing standard switch, browse to the vSwitch created in step 14, click OK, and click Next.
28. Choose a network label, and click Next. The label should match the client traffic network label created in step 8. We used the label `Priv_Net`
29. Click Finish.
30. For the remaining hosts in the vSAN cluster, complete steps 11 through 29.

Turning on the vSAN service

1. Log into the vCenter web client, and navigate to Hosts and Clusters.
2. Select the vSAN cluster, and click Configure.
3. Under vSAN, click Services.
4. Click Turn on vSAN.

Creating and configuring a baseline Windows SQL VM

For both workloads, we tested Microsoft Windows Server 2019 Standard with Microsoft SQL Standard databases. We created a base VM with Windows, then altered the settings to the necessary vCPUs, memory, and drive sizes before installing SQL Server and cloning the VMs for each workload. This section outlines the steps to create the base VM, install and configure Windows, and install and configure SQL Server. See each workload section for the specific VM configurations and SQL Server adjustments.

Creating the VM

1. Launch the VMware vCenter HTML5 client.
2. Right-click the infrastructure host, and choose New Virtual Machine….
3. Choose Create a new virtual machine, and click Next.
4. Name the virtual machine, and click Next.
5. Choose the correct host, and click Next.
6. Choose the correct datastore, and click Next.
7. Choose ESXi 7.0 and later, and click Next.
8. Choose Windows for the Guest OS Family, and Microsoft Windows Server 2019 (64-bit) for Guest OS Version. Click Next.
9. Set the New Hard Disk to 40GB (expand to 85GB for TPC-E VMs), choose the proper network from the drop-down menu, and attach the CD drive. The rest of the settings will be modified based on the workload specifics in the sections below. Click Next.
10. Double-check your settings on the summary page, and click Finish.

Installing and configuring Windows Server 2019 Standard

Follow these steps to install Windows and configure it for performance testing:

1. In the VMware vCenter HTML5 client, click the VM you created, and click Launch Remote Console.
2. Attach the Windows Server 2019 Standard ISO to the VM, and turn the VM on.
3. When prompted to boot from DVD, press any key.
4. When the installation screen appears, leave language, time/currency format, and input method as default, and click Next.
5. Click Install now.
6. When prompted, enter the product key.
7. Select Windows Server 2019 Standard Edition (Server with a GUI), and click Next.
8. Check I accept the license terms, and click Next.
9. Click Custom: Install Windows only (advanced).
10. Select Drive 0 Unallocated Space, and click Next. This will start the installation automatically, and Windows will restart automatically after completing it.
11. When the Settings page appears, fill in the Password and Reenter Password fields with the password of your choice.
12. Log in with the password you set.
13. In the VMware Remote Console, attach VMware Tools to the VM.
14. In Windows, VMware Tools should show up in the CD drive. Double-click to start the install.
15. Choose Typical as the install type, and reboot the VM when the install is finished.
16. Log into Windows, and in Server Manager, set the time zone, enable RDP if desired, and disable the firewall.
17. Run Windows updates until you have no new updates to install.
18. Navigate to Disk Manager and ensure the additional drives are online, formatted, and presented to the OS.
19. Navigate to Settings→System→Power & Sleep→Additional Power Settings and set Preferred plans to High performance.
20. Navigate to Control Panel→System and Security→System→Advanced System Settings.
21. In the Performance box, click Settings→Adjust for best performance.
22. Go to the Advanced tab, and in the Virtual memory box, click Change…
23. For all volumes in the system, change the paging file size to Custom size, set the Initial and Maximum sizes to 4096 MB, and click OK.
24. To close the Performance Options window, click OK.
25. To close the System Properties window, click OK.
26. Click Start, and type `Local Security Policy`. Open the program when it pops up in the search.
27. Expand Local Policies, and click User Rights Assignment.
28. In the right-hand pane, scroll down, and double-click on Lock pages in memory.
29. Click Add User or Group, type `NT Service\MSSQLSERVER`, and click OK.
30. Click OK to close the Properties window, and close the Local Security Policy window.

Installing SQL Server 2019 Standard

1. Mount the ISO to the VM.
2. Double-click the Setup application.
3. Click Installation→New SQL Server Standalone installation or add features to an existing installation.
4. Enter the product key, and click Next.
5. Check the I accept the license terms and Privacy Statement box, and click Next.
6. Check the Use Microsoft Update to check for updates (recommended) box, and click Next.
7. On the Install Rules page, click Next.
8. Check the boxes for the following features, and click Next:
    a. Database Engine Services
    b. Full-Test and Semantic Extractions for Search
    c. Client Tools Connectivity
    d. Client Tools Backwards Compatibility
9. Leave the Default instance, and click Next.
10. Leave the default Service Accounts, and click Next.
11. On the Server Configuration tab, choose Mixed Mode, and enter and confirm a Password for the SQL Server system administrator (sa) account.
12. Click Add Current User to Specify the SQL Server administrators.
13. On the Data Directories tab, change the Data root directory to the database volume.
14. Choose the log volume for the User database log directory.
15. If you plan to have a locally stored backup, change the Backup directory to point to that drive.
16. Click Next.
17. Once you've passed the rule check, click Next.
18. Click Install.
19. When the install is finished, go back to the SQL Server Installation Center, and click Install SQL Server Management Tools.
20. Download the SSMS file, and install with defaults.
21. When prompted, reboot the server.

OLTP database testing with TPC-E-like workload

For OLTP database tests, we used Benchmark Factory to create and test a 30-scale TPC-E-like database on each VM. In this study, we were able to support four VMs per node for a total of 16 VMs across the cluster. We set up an infrastructure server to hold 16 Windows VMs with Benchmark Factory installed to drive the workload on each VM. The VMs under test had the following hardware configuration: 8 vCPUs, 64GB RAM, 85GB OS drive (we increased the OS drive to fit the backup file on it), 500GB database drive, and 50GB log drive. Once we configured and created the database on one VM, we cloned the additional 15.

Installing Benchmark Factory on the Windows client

We deployed a Windows Server 2019 Enterprise VM on an ESXi infrastructure host managed by our vCenter. We then installed Benchmark factory as follows.

1. Download the free Benchmark Factory trial software here: https://www.quest.com/products/benchmark-factory/.
2. Double-click the install file, and on the Welcome screen, click Next.
3. Choose I accept the terms in the license agreement, and click Next.
4. Leave the default destination folder, and click Next.
5. Click Install.

Configuring SQL Server for testing and creating the TPC-E-like database on the SQL VM

1. Open SQL Server Management Studio.
2. Select SQL Server Authentication, type `sa` and your password as your login and password respectively, and click Connect.
3. Set the maximum memory for SQL to 90% of the available RAM, or 57GB, using the following SQL query:

```
exec sp_configure 'max server memory', '58368'
```

4. Set the maximum degree of parallelism (MAXDOP) to 1 to disable parallelization to avoid CPU scheduling bottlenecks with the following SQL query:

```
exec sp_configure 'max degree of parallelism', '1'
```

5. Set the maximum worker threads to 3,000 with the following SQL query:

```
exec sp_configure 'max worker threads', '3000'
```

6. Right-click Databases→New Database….
7. Name the database `tpce`
8. Click Filegroups→Add Filegroup, create a TPCE filegroup, and set it as Default.
9. Click General, and add 16 Database files. Set the Initial Size of each file to 25,600MB.
10. Set the location of each database file to the data volume, and the location of the log file to the log drive.
11. Click OK.

Populating the 30-scale database on the SQL VM

1. On the Benchmark Factory client, double-click the Benchmark Factory shortcut to open the program.
2. Select whether to participate in the Product Improvement Program or not, and click OK.
3. When the program first opens, a New Job Wizard window appears. To create a connection profile to the VM under test, click Create connection profile.
4. From the drop-down menu, choose Microsoft SQL Server as the database type.
5. In the Server name field, type the name or IP address of the SQL VM.
6. Choose SQL Server Authentication, and enter `sa` and the SQL admin password.
7. Choose the tpce database you created, name the Connection, and click Create Connection.
8. In the New Job Wizard, under the Database Under Test section, to ensure that the connection is properly set up, test the connection. Optionally, the Benchmark Factory wizard can also detect the environment information.
9. Click Add Workload.
10. From the Test type drop-down menu, select Create/Delete Benchmark Objects Test.
11. In the Select the benchmark type drop-down menu, choose, TPC-E, and choose Create/Load objects if objects don't exist (no backup sets created).
12. Click Add Test→Test Options.
13. Set the Benchmark Scale to 30.
14. On the Options tab, choose Distribute Index creation (one index per virtual user), and Distribute load using Benchmark Factory Agents.
15. Set the number of users to perform creation/load to 100.
16. Click Job Setup, and leave defaults.
17. Click Agent Setup, and choose the local machine agent.
18. Click Run Job. This build could take several hours.

Backing up the database

1. When the database build has finished, open SQL Server Management Studio on the SQL VM.
2. Right-click the tpce database, and click Tasks→Back up….
3. Click Backup Options, and choose Compress backup in the Set back up compression drop-down menu.
4. Choose a location on the C: drive, name the backup file, and click OK.

Cloning the SQL and client VMs

Once you have installed SQL and Windows, configured all the settings, and created the database, you can clone your VMs.

1. Open the vCenter console, and ensure that the VM is powered off.
2. Right-click the SQL VM→Clone→Clone to Virtual Machine….
3. Name the VM, select your vSAN Datacenter, and click Next.
4. Choose the host for your VM, and click Next.
5. Choose the data store, and select vSAN Default Storage Policy from the VM Storage Policy drop-down menu. Click Next.
6. Click Next.
7. Click Finish.
8. Repeat steps 2 through 7 until you have 16 SQL VMs total.
9. Ensure that the gold client VM is powered off.
10. Repeat steps 2 through 4.
11. Leave the default VM Storage Policy, Keep existing VM storage policies, and click Next.
12. Click Next.
13. Click Finish.
14. Repeat steps 10 through 13 until you have 16 client VMs total.

Creating connections between clients and SQL VMs

1. After cloning the clients, power up all of the client VMs and all of the SQL VMs.
2. Launch the web console for the first client VM, and sign in.
3. Open Benchmark Factory.
4. Click Edit Connections.
5. Double-click the connection to the gold SQL VM.
6. Change the IP and Connection name to that of the corresponding SQL VM (e.g., SQL01 for Client01, SQL02 for Client02, etc.).
7. Click Test Connection to ensure that the connection is valid, and click OK.
8. For the remaining client VMs, complete steps 2 through 7.

Creating the TPC-E-like job

1. On the first client VM, open Benchmark Factory.
2. Click New Job.
3. Select the corresponding SQL VM connection as the Database Under Test.
4. Click Workload.
5. Select Industry Standard Benchmark Test from the drop-down menu, and select TPC-E (Improved TPCC).
6. Click Add Test.
7. In the left pane of the New Job Wizard, select Create Objects for TPC-E.
8. Click the Test Options tab, and set the Benchmark Scale to 30.
9. Click the Options Tab, and select Create/Load objects if objects don't exist (no backup sets created).
10. In the left pane, click TPC-E Transaction Mix.
11. Click the Test Options tab, and set each Transaction to No Delay for Latency.
12. Click the User Load tab, Delete the Selected User Load(s), and add a Single User Load of 20.
13. Click the Timing tab, set the Pre-Sampling time to 30 minutes, and set the Sampling time to 1 hour.
14. Click Agent, and ensure the local machine is selected.
15. Click Save/Close.
16. For the remaining client VMs, complete steps 1 through 15.

Running the TPC-E test

1. On the first client VM, open Benchmark Factory.
2. Under the TPC-E Job in the left pane, double-click TPC-E Transaction Mix.
3. Click the Job Setup tab.
4. Click the Schedule tab.
5. Check the Enable Scheduling checkbox.
6. Input the time at which you would like the test to start.
7. Repeat steps 1 through 6 for the remaining client VMs.
8. Start ESXtop counters and VM perfmon counters, if using.

Each complete test cycle consisted of the following general steps for each VM:

1. Clean up prior outputs from the target system.
2. Drop the database from the target.
3. Restore the database on the target.
4. Pregrow the database log file to 15GB.
5. Reboot the target VM.
6. Let the target VM idle for 10 minutes.
7. Start performance counters.
8. Start the Benchmark Factory TPC-E workload on the corresponding client.

OLAP testing with the TPC-H-like workload

We ran the OLAP TPC-H-like workload using the HammerDB benchmark. As instructed in the documentation, we ran the HammerDB test directly on the server under test as the overhead on the hardware is negligible. Thus, no clients are needed for this test. For OLAP database tests, we used dbgen to create a 300-scale TPC-H-like database, then put a copy on each VM. In this study, we were able to support a total of 10 VMs across the cluster. The VMs under test had the following hardware configuration: 16 vCPUs, 120GB RAM, 40GB OS drive, 450GB database drive, and 75GB log drive. Once we configured and created the database on one VM, we cloned out the additional 9.

Creating the database and its tables

To create the 300-scale database we made a new database in SQL Server Management Studio. In addition to the default Primary SQL Server filegroup, we created a secondary data filegroup in the new database. We created 10 files in this filegroup, all sized at approximately 36GB. We resized the single transaction log file to 36GB, as well. When testing, we stored the data files one virtual disk and the logs on a second. TempDB database files were stored the same way with the data files on the data drive and the log files on the log drive. We created the specific tables required for a TPC-H-like database according to TPC and HammerDB documentation: Part, Supplier, PartSupp, Customer, Nation, Lineitem, Region, and Orders. We then created a partition function using a date data type, with weekly values as follows:

```
CREATE PARTITION FUNCTION [DATEPARTITION](date) AS RANGE LEFT FOR VALUES (N'1992-01-07T00:00:00.000',
N'1992-01-14T00:00:00.000' [, ……n]).
```

We then applied this partition function using a partition scheme, to the secondary filegroup containing the 10 data files we describe above. The remainder of the schema definition was as defined by dbgen and the HammerDB tool. We created the test data using the dbgen tool and loaded it into the prepared database. We used clustered columnstore indexing where possible and applied the partition scheme specifically to the ShipDate and OrderDate columns, in the LineItem and Orders table, respectively. For our specific schema DDL scripts, contact us at info@principledtechnologies.com.

1. Open SQL Server Management Studio.
2. Right-click Databases, and choose New Database…
3. Name the database.
4. Click the Filegroups tab, and click Add Filegroup
5. Name the new filegroup, and set it to Default.
6. In the general tab, click Add to create 10 additional data files.
7. Name each file and set the initial size for each new file to 36GB. Make sure all new files are on the new Filegroup you created.
8. If necessary, change all data and log file paths to point to the data and log drives on the system.
9. On the Options tab, set the Recovery Model to simple.
10. To start the build, Click OK. This may take a while.

Generating data, loading it into the database, and creating indexes

We used the dbgen tool to create the flat files for the 300-scale database and loaded them into the empty database. To ensure enough room to simultaneously hold the flat files and load the database, we conducted these steps on a separate infrastructure server with SQL Server installed. We then created a backup file and used it to restore the finished database to the server under test.

1. Download the dbgen tool from GitHub here: https://github.com/electrum/tpch-dbgen.
2. Unzip the file to a volume with roughly 400GB capacity.
3. Download and install the Community edition of VisualStudio.
4. Navigate to the dbgen folder, and build the tpch.vcproj in Visual Studio.
5. In the dbgen folder, navigate to the debug folder and press shift + right-click in the folder. Click Open PowerShell Window.
6. In PowerShell test, ensure dbgen.exe works by typing `./dbgen.exe -h`. If the help menu pops up, continue. If you get an error, try copying the dbgen.exe file to the dbgen folder, opening PowerShell there, and running the command again.
7. Once you have dbgen.exe working properly, create the 300 scale files with the following command:

   ```
   ./dbgen.exe -s 300
   ```

8. Once the files have finished building (and this will take some time), you can load them into the database.
9. Open Microsoft SQL Management Studio, and run bulk insert commands to load the files into the tables.
10. Create the database Indexes as described above.
11. Once the index creation has finished, right-click the TPC-H database, and click Properties.
12. On the Options tab, set the Recovery model to Full, and click OK.
13. Right-click the TPC-H database again, and navigate to Tasks→Shrink→Database.
14. Check the box for Reorganize files before releasing unused space, and set the Maximum free space to 0%
15. Click Ok.
16. Once the shrink task is done, the database is ready to back up.

Backing up the database

While we didn't perform any write tasks in the workload test, we always have a clean database backup that we can restore from should we need it. We stored this backup file on a separate infrastructure server. Follow these steps to create a database backup file:

1. Open Microsoft SQL Management Studio, and right-click the TPC-H database.
2. Navigate to Tasks→Backup.
3. On the General tab, click Add…to set the name of your backup file and the location to store the file.
4. On the Backup Options tab, in the Set backup compression drop-down menu, select Compress backup.
5. Click OK.

Configuring SQL Server for running the TPC-H-like workload on the SQL VM

1. Open SQL Server Management Studio.
2. Select SQL Server Authentication, type sa and your password as your login and password respectively, and click Connect.
3. Set the maximum memory for SQL to 90% of the available RAM, or 108GB, using the following SQL query:

   ```
   exec sp_configure 'max server memory', '110592'
   ```

4. Set the maximum degree of parallelism (MAXDOP) to 8 in accordance with TPC-H best practices with the following SQL query:

   ```
   exec sp_configure 'max degree of parallelism', '8'
   ```

5. Set the maximum worker threads to 3,000 with the following SQL query:

   ```
   exec sp_configure 'max worker threads', '3000'
   ```

Restoring the database on the VM

1. Copy the backup file to the gold VM.
2. Open SQL Server Management Studio.
3. Select SQL Server Authentication, type sa and your password as your login and password respectively, and click Connect.
4. Right click Databases, and select Restore Database…
5. Select Device.
6. Click Add, and select the backup file. Click OK, and click OK again.
7. In the left pane, click Files, and ensure that the files are pointing to the correct data and log directories.
8. After reviewing, click OK to restore the database.

Installing HammerDB on the VM

1. Download HammerDB from here: https://hammerdb.com/download.html.
2. Double-click the .exe file, choose English, and click OK.
3. Click Yes.
4. Click Next.
5. Chose a destination location, and click Next.
6. Click Next.
7. Click Finish.

Cloning the SQL VMs

Once you have installed SQL and Windows, configured all the settings, and created the database, you can clone your VMs.

1. Open the vCenter console, and ensure that the VM is powered off.
2. Right-click the SQL VM→Clone→Clone to Virtual Machine…
3. Name the VM, select your vSAN Datacenter, and click Next.
4. Choose the host for your VM, and click Next.
5. Choose the data store, and from the VM Storage Policy drop-down menu, select vSAN Default Storage Policy. Click Next.
6. Click Next.
7. Click Finish.
8. Repeat steps 1 through 7 until you have 10 SQL VMs total.
9. Ensure that the gold client VM is powered off.

Running the test

1. Log into the first SQL VM.
2. Open HammerDB.
3. Select Options→Benchmark.
4. Choose MSSQL Server and TPC-H.
5. Click Mode.
6. Select Master Mode, and click OK. Make a note of the Master ID and Master Hostname.
7. Log into the next SQL VM.
8. Open HammerDB.
9. Click Mode.
10. Select Slave Mode, and input the Master ID and Master Hostname given previously. Click OK.
11. Repeat steps 7-10 for the remaining SQL VMs.
12. Log back into the Master SQL VM.
13. Expand SQL Server→TPC-H→Schema Build.
14. Double-click Options, change the driver to ODBC Driver 17 for SQL Server, set the scale to 300, set MAXDOP to 8, and check the box for Clustered Columnstore. Click OK.
15. Expand Driver Script, and double-click Options. Click OK to load.
16. On the top pane, click Open Edit Menu.
17. Click Master Distribution.
18. Expand Virtual User, and double-click Options.
19. Choose 1 or 4 users depending on whether you are running a single- or multi-user test.
20. Check the boxes for Show Output, Log Output to Temp, and Use Unique Log Name.
21. Click OK.
22. Double-click Create users.
23. To capture performance metrics on the system, start Performance monitor set to record CPU, Memory, and drive usage information.
24. To begin the run, click Start.
25. When the run finishes, stop Perfmon and save the HammerDB results file and Perfmon output.
26. Reboot the VMs.
27. Complete the test two more times for a total of three runs at each user count, and record the median run.

Read the report at http://facts.pt/0n1766h ▶

This project was commissioned by Dell Technologies.

Principled Technologies®

Facts matter.®

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.