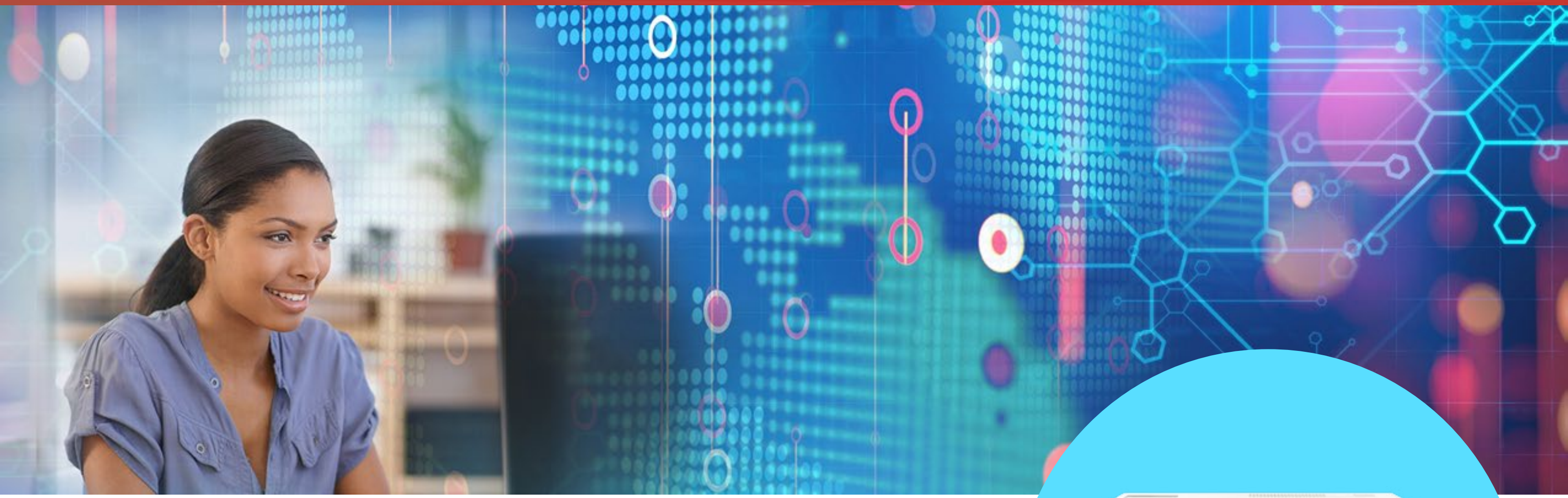


Dell EMC ScaleIO Ready Node solution

Support more database users while minimizing datacenter sprawl

A Dell EMC software-defined storage solution featuring ScaleIO Ready Nodes outperformed an HPE solution with flash storage

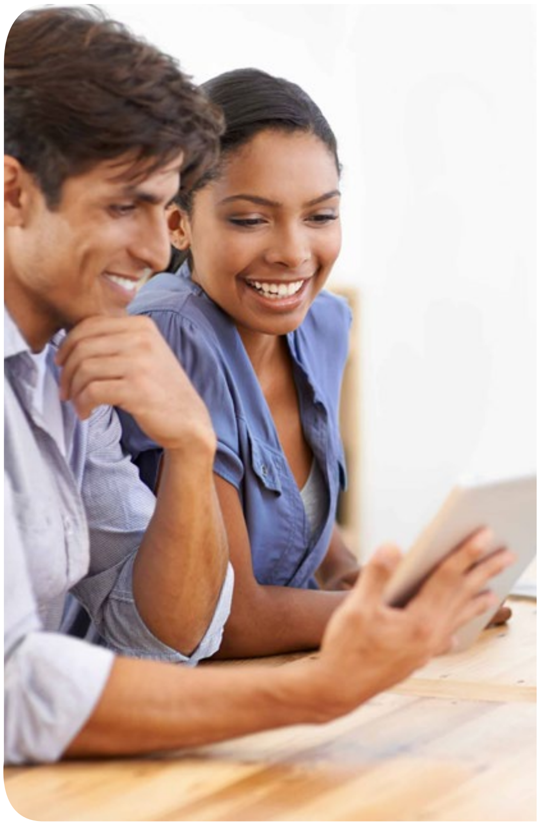
Storage architectures that have dedicated workloads and SANs cannot always provide the scale, elasticity, and flexibility to meet the demands of today's high-performing applications. These complex and siloed environments can be plagued by inefficient and rigid planning, high costs, and longer time to market. A Dell EMC™ software-defined storage solution featuring ScaleIO® Ready Nodes departs from these traditional architectures and can deliver strong performance for multiple database workloads in a space-efficient package.

Compared to a flash storage-based HPE solution of ProLiant DL360 Gen9 server nodes and 3PAR 8450 storage, the Dell EMC software-defined storage solution, featuring ScaleIO, processed more database transactions and scaled a mixed-workload environment without sacrificing performance. The Dell EMC ScaleIO solution also recovered workloads and applications from a simulated hardware failure more quickly than the HPE solution did. This reliability and scalability ensures that your database users can stay active and have a good experience, even with unexpected changes in the datacenter.

Process more data
Up to 1.8X more database operations

Double the work
by scaling out resources

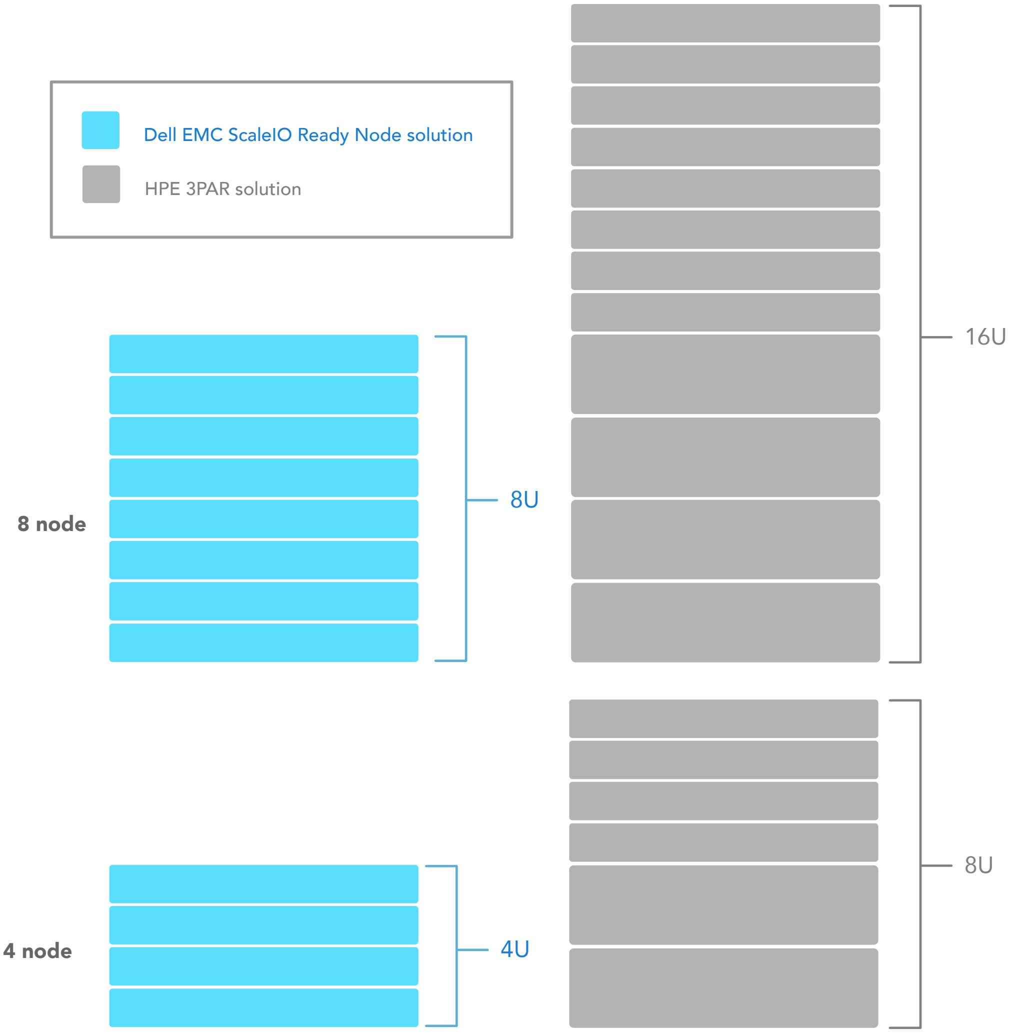
Recover quickly
with effective fault-tolerant features



The goal of our performance testing was to stress the solutions with two workload scenarios before and after scaling. We ran four-node configurations of both solutions in an OLTP-only scenario and in an OLTP and data mart scenario. After scaling out, we ran eight-node configurations in both scenarios.

The graphic below shows the rack space advantage for compute and storage hardware for the Dell EMC ScaleIO solution in both configurations compared to the HPE solution. It's important to note that we did not need to add a separate storage array to the Dell EMC ScaleIO solution as we did for the HPE solution. In addition, we turned on data compression features of the HPE solution during testing to determine performance using common features. In terms of networking, only the HPE solution required fiber channel switches. ScaleIO software-defined storage runs on 10Gbps Ethernet with the ability to use existing Ethernet network infrastructure as opposed to requiring expensive fibre channel infrastructure.

By implementing a Dell EMC ScaleIO Ready Node solution, your admins have less hardware to manage, and your organization can see potential OpEx savings as a result of consuming less rack space. Software-defined ScaleIO Ready Nodes can increase the efficiency of your storage solution by addressing the challenges of managing multiple storage silos.



Our test tool: DVD Store 2

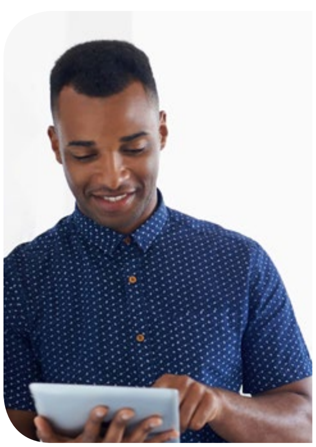
We used DVD Store 2 Version 2.1 (DS2) to create a real-world database workload. DS2 simulates an online store, mimicking the way thousands of users would shop. It reports these simulated user actions by showing how many OPM the system can handle. The more user-initiated orders a server can fulfill, the better its performance. DS2 also performs other actions, such as adding new customers.

To learn more about DS2, visit en.community.dell.com/techcenter/extras/w/wiki/dvd-store. To download DS2, visit https://github.com/dvdstore.

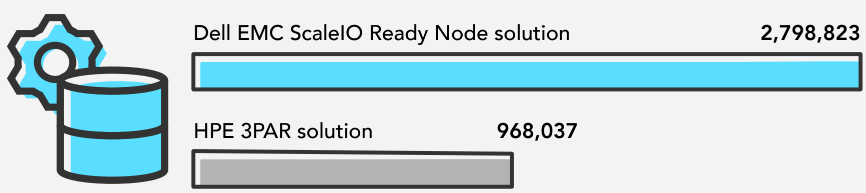
Do more work with your transactional databases

Processing database transactions to meet user demand requires powerful compute and vast storage resources. The more transactions your system can support, the more revenue your online business can generate while ensuring employees can access valuable data simultaneously.

The Dell EMC ScaleIO solution handled up to 1.8 times more database orders per minute (OPM) than the HPE solution: 2,798,823 compared to 968,037. These outputs came from the two eight-node configurations running the OLTP and data mart workloads. Even in the four-node configuration we tested, the Dell EMC ScaleIO solution handled more OPM than the HPE (48 percent more). Processing more customer requests can translate to a better experience for your end users. For more information on how we tested and for detailed results, see Appendix B.



189% more orders per minute with the eight-node Dell EMC solution



What's a data mart?

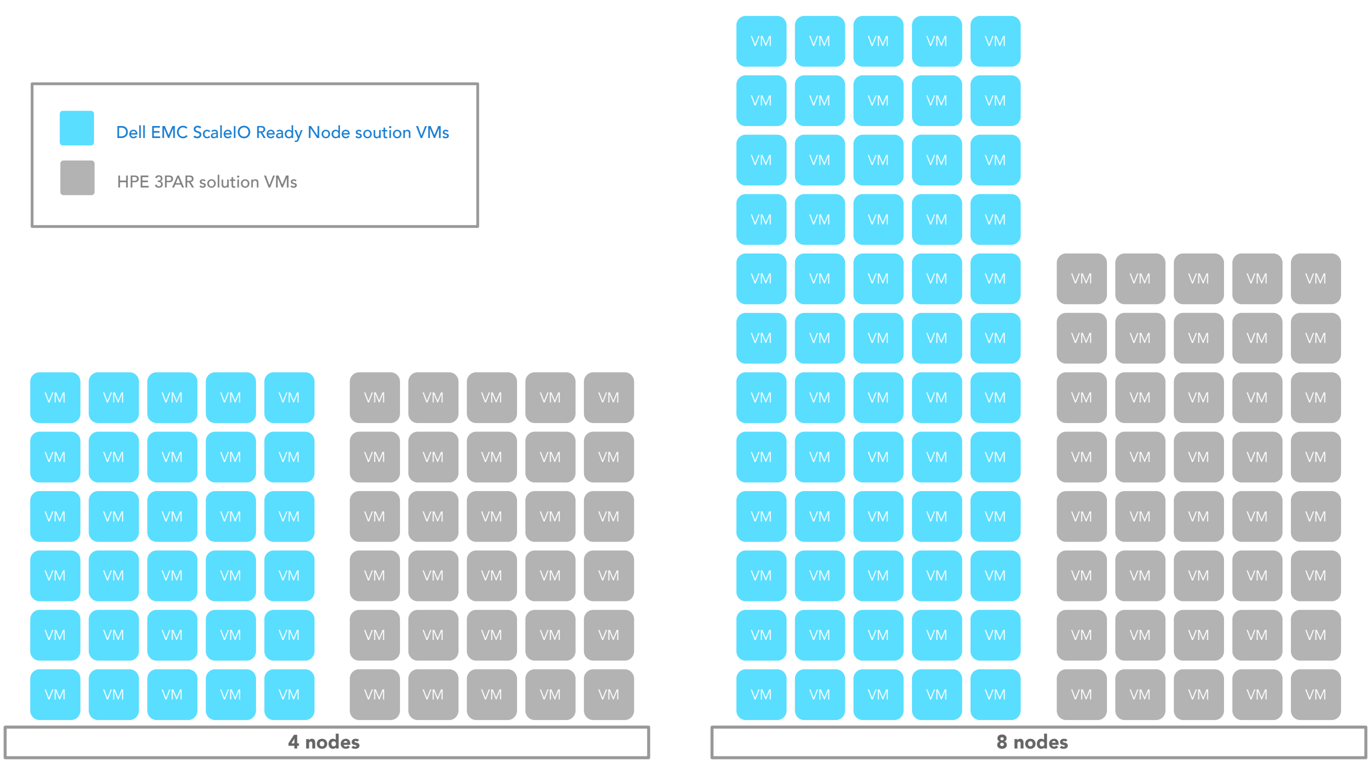
Your enterprise business collects data from many different departments. Whether it's sales, marketing, or research and development, you will often want to bundle data from these disparate sources and load it into a single location for analysis and reporting. The data mart is a convenient place to store different departments' information for processing and analysis.

Get more from your workloads when scaling out resources

By abstracting, pooling, and automating storage with ScaleIO, you can eliminate multiple silos of storage arrays and consolidate the capacity and workloads into a simplified software-defined infrastructure. Compute and storage resources scale independently or together, providing your organization with the flexibility to add or remove these resources on the fly. Storage additions and removals can be done in small or large increments because the ScaleIO Ready Node solution automatically re-balances and optimizes resources with the goal of increasing performance and enhancing resiliency.

Both four-node configurations of the solutions supported 30 VMs. To see how the two solutions handled additional database work from more VMs, we increased the number of compute nodes in both configurations from four to eight. We added two 2U storage arrays to the HPE solution, and the eight-node configuration successfully ran the OLTP and data mart scenario at 40 VMs. When we ran more than 40 VMs however, the eight-node HPE configuration dropped threads and failed to run the workloads properly. In comparison, the eight-node Dell EMC configuration doubled its VM count (30 to 60).

In eight-node configurations, both solutions processed more transactions than the four-node configurations due to the additional resources. When it ran the OLTP-only scenario and the OLTP and data mart scenario, the eight-node Dell EMC ScaleIO solution handled more than twice the OPM that it supported in the four-node configuration. In comparison, the eight-node HPE solution processed just over 12 percent more OPM than it did with four nodes, offering significantly lower scaling in performance than the Dell EMC ScaleIO solution. By taking advantage of ScaleIO software, your organization can minimize the risk of adding excessive hardware components and running into datacenter sprawl. More powerful hardware allows IT to consolidate several workloads without sacrificing database user experience. For more information on how we tested and for detailed results, see Appendix B.



Keep up with customer and employee activity

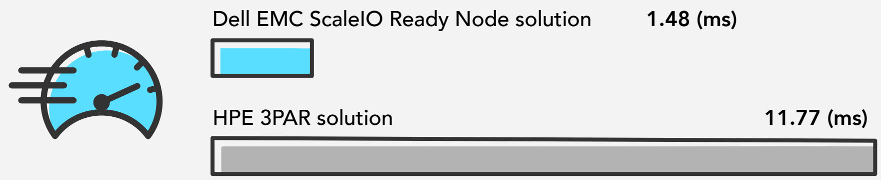
Today's users have a low threshold for waiting. If pages and sites won't load quickly for an ecommerce application, they're likely to take their business elsewhere. If updates to an organization's internal transactional database application take too long, employees might get frustrated and the organization could lose productivity.

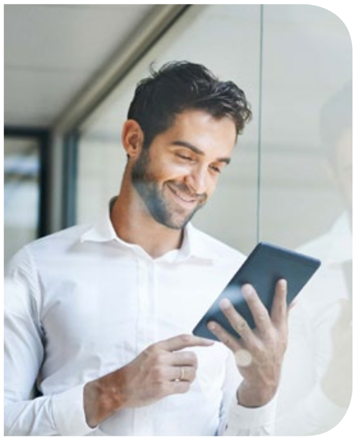
The versatile Dell EMC ScaleIO Ready Node

The Dell EMC ScaleIO Ready Node brings software-defined storage to the core datacenter. Dell EMC pre-validates, optimizes, and configures x86 Dell EMC PowerEdge™ servers to run ScaleIO, offering many configuration options.[1] Organizations can deploy ScaleIO Ready Nodes as a hyper-converged architecture, a traditional architecture with storage and compute layers (like legacy external storage systems), or a solution that mixes the two approaches. You can learn more about ScaleIO Ready Nodes here.

The eight-node Dell EMC ScaleIO solution delivered faster response times, beating the eight-node HPE solution by up to 87 percent when the solutions ran OLTP and data mart workloads. Those fractions of a second are crucial when processing multiple and concurrent transactions, even if users may not notice a difference in a single transaction.

87% lower latency with the eight-node Dell EMC solution





Reduce the risk of downtime

Data protection in your datacenter is a necessity. There's often no way to know when an event can cause hardware to fail and go offline, putting your applications and data at risk. Fault-tolerant hardware can recover from a failure, but it doesn't protect against downtime or guarantee that applications will return to pre-failure performance. ScaleIO Ready Nodes offer fault-tolerant features to help protect data while mitigating downtime.

We ran both eight-node configurations through four scenarios to see how well they can protect database application performance: node disconnection from the network, node failure, node power loss, and drive failure. In all four scenarios, the Dell EMC ScaleIO solution maintained near-optimal database performance and experienced no ScaleIO-related storage downtime. During the ScaleIO Ready Node failure test, the hypervisor rebooted the subset of VMs from the failed node onto one of the other seven hosts in the cluster, resulting in a workload interruption lasting less than one minute. In comparison, the HPE solution experienced unexpected workload downtime in two scenarios. When we simulated a 3PAR controller failure in the HPE solution, workloads delivered 28 percent fewer OPM. The following table shows if the solutions returned to near-optimal database performance. Your organization deserves an enterprise-grade solution that delivers reliable fault tolerance to keep workloads moving at the speed of business. See Appendix B for more information on how we tested and detailed test results.

	Dell EMC ScaleIO Ready Node	HPE 3PAR
Network cable unplugged	✓	✓ (!)
ScaleIO Ready Node failure/ 3PAR controller failure	✓	✗ (!)
Power cable unplugged	✓	✓
Drive failure	✓	✓

(!) HPE 3PAR storage-related service interruptions

Conclusion

A space-efficient hardware solution that effectively handles your growing database demands can pay off for your business. In our datacenter, a Dell EMC ScaleIO Ready Node solution handled more OPM, scaled performance linearly to match the scaling out of resources, responded more quickly, required less rack space to scale out, and maintained database performance in more failover scenarios compared to an HPE solution with flash storage. These advantages offered by ScaleIO can help your organization support more database application users, shorten wait times, and protect data.

To find out more about Dell EMC ScaleIO Ready Nodes, click here.

1 "SCALEIO READY NODE: Hardware Specification Sheet," September 7, 2017, https://www.emc.com/collateral/specification-sheet/h15406-scaleio-ready-node-ss.pdf?domainUrlForCanonical=https%3A%2F%2Fwww.emc.com.

On June 1, 2017, we finalized the hardware and software configurations we tested. Updates for current and recently released hardware and software appear often, so unavoidably these configurations may not represent the latest versions available when this report appears. For older systems, we chose configurations representative of typical purchases of those systems. We concluded hands-on testing on August 16, 2017.

Appendix A: System configuration information

Server configuration information	Dell EMC PowerEdge R630	HPE ProLiant DL360 Gen9
BIOS name and version	Dell 2.2.5	HPE P89 2.40
Non-default BIOS settings	ScaleIO Ready Node settings	Performance mode
Operating system name and version/build number	VMware ESXi 6.5.0 build 5310538	VMware ESXi™ 6.5.0 build 5310538
Date of last OS updates/patches applied	06/01/17	06/01/17
Power management policy	ScaleIO Ready Node setting	Performance mode
Processor		
Number of processors	2	2
Vendor and model	Intel® Xeon® E5-2680 v4	Intel Xeon E5-2680 v4
Core count (per processor)	14	14
Core frequency (GHz)	2.40	2.40
Stepping	M0	M0
Memory module(s)		
Total memory in system (GB)	512	512
Number of memory modules	16	16
Vendor and model	Micron MTA36ASF4G72PZ-2G3D1QG	Hynix HMA84GR7AFR4N-UH
Size (GB)	32	32
Type	PC4-19200	PC4-19200
Speed (MHz)	2,400	2,400
Speed running in the server (MHz)	2,400	2,400
Storage controller		
Vendor and model	Dell HBA330 Mini (Embedded)	HPE Smart Array P440ar
Cache size (GB)	N/A	2 GB
Firmware version	13.17.03.00	5.04
Local storage		
Number of drives	10	2

Server configuration information	Dell EMC PowerEdge R630	HPE ProLiant DL360 Gen9
Drive vendor and model	Dell MZILS1T9HEJH0D3	HPE 862119-001
Drive size	1.92 TB	300 GB
Drive information (speed, interface, type)	12Gb SAS, SSD	10K, 6Gb SAS, HDD
Network adapter		
Vendor and model	Intel Ethernet Server Adapter X520-DA2	HP StoreFabric SN1100E
Number and type of ports	2 x 10GbE SFP+	2 x 16Gb Fibre Channel
Second type of network adapter		
Vendor and model	Dell C63DV Intel X520/I350 Quad Port Network Daughter Card	N/A
Number and type of ports	2 x 10GbE SFP+, 2 x 1GbE	N/A
Power supplies		
Vendor and model	Dell E750E-S1	HPE 723600-201
Number of power supplies	2	2
Wattage of each (W)	750	800

Storage configuration information

Storage configuration information	HPE 3PAR StoreServ 8400 storage
OS version	3.31.215
Number of storage controllers	2
Number of storage shelves	4 (1 x 3PAR StoreServ 8400 + 3 x 3PAR StoreServ 8000)
Number of drives per shelf	20
Drive vendor and model number	HP DOPE1920S5xnNMRI
Drive size (TB)	1.92
Drive information (speed, interface, type)	6Gb SAS, SSD

Network switch configuration information

This provides detailed configuration information for the Dell EMC Connectrix® DS-6620B switch we used to connect the HPE ProLiant DL360 Gen9 servers to the HPE 3PAR storage and for the two Dell EMC Networking S4048-ON switches we used to connect the ScaleIO Ready Nodes.

Network switch configuration information	Dell EMC Connectrix DS-6620B switch	Dell EMC Networking S4048-ON switch
Number and type of ports	48 x SFP+ 32Gb, 4 x QSFP 32Gb	48 x SFP+ 10GbE, 6 x QSFP+ 40GbE
Number and type of ports used in test	20 x SFP+ 32Gb	16 x SFP+ 10GbE, 2 x QSFP+ 40GbE

Appendix B: An overview of how we tested with test results

For the four- and eight-node configurations, we set up the server nodes in a VMware vSphere® 6.5-based high availability (HA) cluster. In addition, we set the server nodes to performance mode. We configured the storage in each case to use commonly enabled settings and features for each of the solutions, including compression on the 3PAR storage volumes. In each configuration, we distributed VMs evenly among all nodes in the cluster. All performance testing results are the median of three runs.

Four-node configurations

Dell EMC ScaleIO solution

- Servers: four Dell EMC PowerEdge R630 ScaleIO Ready Nodes
- Storage: ScaleIO software-defined storage on the server nodes with 40 1.92TB SSDs (10 drives per node)

HPE solution

- Servers: four HPE ProLiant DL360 Gen9 servers
- Storage: an HPE 3PAR StoreServ 8400 two-node storage array with one HPE 3PAR StoreServ 8000 expansion array with 40 1.92TB SSDs (20 drives per array)

First, we ran a 30-VM OLTP-only workload on both four-node configurations for 45 minutes. The Dell EMC ScaleIO solution delivered up to 34 percent greater OLTP database performance and up to 32 percent lower latency.

Four nodes, OLTP-only workload	Dell EMC ScaleIO solution	HPE solution
Avg total IOPS	100,899	77,812
Avg total reads/s	45,701	36,128
Avg total writes/s	55,136	41,614
Avg latency (ms)	2.42	3.43
Avg read latency (ms)	1.22	0.98
Avg write latency (ms)	3.36	4.93
Avg total OPM (all VMs)	1,272,737	948,624

Then, we ran the same 30-VM OLTP workload for three hours on both four-node configurations and added a data mart workload that ran during approximately the middle hour. The Dell EMC ScaleIO solution delivered up to 48 percent greater OLTP database performance while also running the data mart workload and up to 76 percent lower latency.

Four nodes, OLTP + data mart workload	Dell EMC ScaleIO solution	HPE solution
Avg total IOPS	95,130	66,606
Avg total reads/s	39,811	29,115
Avg total writes/s	55,259	37,422
Avg latency (ms)	2.33	8.05
Avg read latency (ms)	1.14	1.14
Avg write latency (ms)	3.24	13.56
Avg total OPM (all VMs)	1,270,780	857,592

Four nodes, OLTP + data mart workload	Dell EMC ScaleIO solution	HPE solution
Data mart time (hr:mins)	1:14	1:00
VMs supported	30	30

Eight-node configurations

Dell EMC ScaleIO solution

- Servers: eight Dell EMC PowerEdge R630 ScaleIO Ready Nodes
- Storage: ScaleIO software-defined storage on the server nodes with 80 1.92TB SSDs total (10 drives per node)

HPE solution

- Servers: eight HPE ProLiant DL360 Gen9 servers
- Storage: an HPE 3PAR StoreServ 8400 two-node storage array with three HPE 3PAR StoreServ 8000 expansion arrays with 80 1.92TB SSDs total (20 drives per array)

After scaling out the configurations from four to eight nodes, we scaled from 30 to 60 VMs and re-ran the OLTP workload on both solutions for 45 minutes. The Dell EMC ScaleIO solution delivered up to 163 percent greater OLTP database performance and up to 85 percent lower latency. We also observed linear scaling in the Dell EMC ScaleIO solution when scaling out from four to eight nodes in this workload, with OPM performance doubling.

Eight nodes, OLTP-only workload	Dell EMC ScaleIO solution	HPE solution
Avg total IOPS	197,629	85,713
Avg total reads/s	94,004	41,529
Avg total writes/s	103,493	44,018
Avg latency (ms)	1.73	11.79
Avg read latency (ms)	0.98	5.83
Avg write latency (ms)	2.49	17.63
Avg total OPM (all VMs)	2,824,608	1,072,666

Then, we determined how many database VMs each solution could support while running the OLTP and the data mart workloads. We ran the OLTP workload for three hours on both eight-node configurations and added a data mart workload that ran during approximately the middle hour. The Dell EMC ScaleIO solution supported all 60 VMs when running the data mart workload, but the HPE solution supported only 40 VMs when running the data mart workload. We believe the Dell EMC ScaleIO solution scaled well due to its hyperconverged design, scaling compute and storage evenly with each added node. Conversely, we believe the HPE solution experienced limited scaling due to the storage processors in the HPE 3PAR controllers being fully utilized.

Additionally, the Dell EMC ScaleIO solution delivered up to 189 percent greater OLTP database performance while also running the data mart workload and up to 89 percent lower latency. Like with the OLTP only workload, the performance of the Dell EMC ScaleIO solution scaled linearly in the mixed workload tests, with performance doubling when scaling out from four to eight nodes. Finally, the Dell EMC ScaleIO solution completed the data mart workload 30 minutes faster than the HPE solution.

Eight nodes, OLTP + data mart workload	Dell EMC ScaleIO solution	HPE solution
Avg total IOPS	190,427	74,168

Eight nodes, OLTP + data mart workload	Dell EMC ScaleIO solution	HPE solution
Avg total reads/s	84,672	34,492
Avg total writes/s	105,598	39,546
Avg latency (ms)	1.48	11.77
Avg read latency (ms)	0.87	8.05
Avg write latency (ms)	2.15	15.25
Avg total OPM (all VMs)	2,798,823	968,037
Data mart time (hr:mins)	1:18	1:48
VMs supported	60	40

Fault tolerance testing configurations

For our fault tolerance testing scenarios, we used the 60-VM OLTP-only workload as our representative workload. Approximately halfway through each of the 45-minute test runs, we simulated the following single component failures:

- Unplugged network cable:
  - Dell EMC ScaleIO solution – We unplugged a single 10GbE SFP+ cable from one of the server nodes.
  - HPE solution – We unplugged a single Fibre Channel cable from one of the 3PAR controller nodes.
- Node/controller failure:
  - Dell EMC ScaleIO solution – We disconnected one of the server ScaleIO ready nodes.
  - HPE solution – We disconnected one of the 3PAR controller nodes from the 3PAR StoreServ 8400 array.
- Unplugged power cable:
  - Dell EMC ScaleIO solution – We unplugged a single power cable from one of the server nodes.
  - HPE solution – We unplugged a single power cable from one of the 3PAR controller nodes.
- Drive failure:
  - Dell EMC ScaleIO solution – We removed a single drive from one of the server nodes.
  - HPE solution – We removed a single drive from one of the 3PAR storage trays.

The table below shows a summary of the results of each of the scenarios for each solution.

Fault tolerance tests	Dell EMC ScaleIO solution	HPE solution
Network cable unplugged	No downtime and no effect on performance.	Temporary downtime on a subset of VMs, with varying numbers of VMs affected each time. No performance effect once multi-pathing had re-routed storage connectivity through the redundant controller.
ScaleIO ready node failure / 3PAR controller failure	Approximately a 30-second service interruption while the VMs hosted on the downed node failed over and rebooted on new hosts. Less than <10% drop in performance.	Temporary downtime on a subset of VMs, with varying numbers of VMs affected each time. Performance dropped by 28%.
Power cable unplugged	No downtime and no effect on performance.	No downtime and no effect on performance.

Fault tolerance tests	Dell EMC ScaleIO solution	HPE solution
Drive failure	No downtime and no effect on performance.	No downtime and no effect on performance.

The following table compares the OLTP workload results and downtime of the eight-node Dell EMC ScaleIO solution without failure and in the Node/controller failure scenario. This was the only scenario that resulted in any interruption or change in performance for the Dell EMC ScaleIO solution, with a roughly seven percent change in OPM performance of the Dell EMC ScaleIO solution after a single node failure.

Dell EMC ScaleIO solution	Baseline (no failures)	After the ScaleIO Ready Node failure
OPM	2,824,608	2,633,698
Downtime of 7 VMs (min:sec)	N/A	31s

The following table compares the OLTP workload results of the eight-node HPE solution without failure and in the two scenarios that resulted in database downtime or performance decreases. During each of the two scenarios, the simulated failure caused dropped threads and forced databases temporarily offline. In the Node/controller failure scenario, performance dropped by 28 per

HPE solution	Baseline (no failures)	After unplugging a network cable	After the 3PAR Controller failure
OPM	1,072,666	1,088,866	770,683
Number of VMs with dropped threads	N/A	25	44

Appendix C: How we tested details

Configuring the Dell EMC Networking S4048-ON switches

We configured the two switches we used with our ScaleIO Ready Nodes as follows via SSH.

```
enable
restore factory-defaults stack-unit all clear-all
Proceed to delete startup-config [confirm yes/no]yes

delete startup-config
Proceed to delete startup-config [confirm yes/no]yes
reload
System configuration has been modified. Save? [yes/no]no
Proceed with reload [confirm yes/no]yes

management route 0.0.0.0/0 10.213.0.1
ip ssh server enable

username admin privilege 15 password 0 Password1
enable password level 15 0 Password1

configure
interface ManagementEthernet 1/1
 ip address 10.213.11.1/16
 no shutdown
exit

interface range TenGigabitEthernet 1/1-1/48
mtu 9216
 portmode hybrid
 switchport
 spanning-tree rstp edge-port
 no shutdown
exit

interface range fortyGigE 1/49-1/50
 no ip address
 mtu 9216
 no shutdown
exit

interface Port-channel 100
 description “VLTi - interconnect link”
channel-member fortyGigE 1/49,1/50
 no shutdown
exit

interface Vlan 1
untagged TenGigabitEthernet 1/1-1/48
untagged Port-channel 100
management route 0.0.0.0/0 10.213.0.1
ip ssh server enable
no shutdown
exit
exit

Copy running-config startup-config┬á
Do write
Exit
```

Installing VMware ESXi 6.5 on the host servers

We completed this installation process on all eight hosts for the HPE and Dell EMC ScaleIO solutions.

1. Attach the installation media.
2. Boot the server.
3. At the VMware Installer screen, press Enter.
4. At the EULA screen, to Accept and Continue, press F11.
5. Under Storage Devices, select the appropriate storage device, and press Enter.
6. As the keyboard layout, select US, and press Enter.
7. Enter the root password twice, and press Enter.
8. To start installation, press F11.
9. After the server reboots, press F2, and enter root credentials
10. Select Configure Management Network, and press Enter.
11. Select the appropriate network adapter, and select OK.
12. Select IPv4 settings, and enter the desired IP address, subnet mask, and gateway for the server.
13. Select OK, and restart the management network.
14. Repeat steps 1 through 13 on the rest of the servers.

Deploying the VMware vCenter Server 6.5 appliance

We deployed a vCenter instance, hosted on a separate management server for each of the two solutions.

1. Attach the installation media.
2. Navigate to the appropriate folder, and install the vcsa installation client.
3. Open the link to vcsa-setup.html found in the main folder.
4. Click Install.
5. Check the box to accept the EULA, and click Next.
6. Enter the FQDN or IP of the target server, root, the password, and click Next.
7. Click Yes.
8. Enter a name for the appliance, an OS password, confirm the OS password, and click Next.
9. Select Install vCenter Server with embedded Platform Services Controller, and click Next.
10. Select Create a new SSO domain.
11. Enter a vCenter SSO password, and confirm the password.
12. Enter an SSO Domain name, an SSO Site name, and click Next.
13. Select the appliance size, and click Next.
14. Select a datastore, check to enable Thin Disk Mode, and click Next.
15. Select Use an embedded database (PostgreSQL), and click Next.
16. Enter a network address, system name, subnet mask, gateway, and DNS server(s).
17. Configure time synch by entering NTP server(s), enable SSH, and click Next.
18. Click Finish.

Creating a cluster and adding the hosts to VMware vCenter

We completed the following steps on each vCenter deployment to add the eight corresponding hosts to each VMware cluster.

1. Once logged into the vCenter, 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. Enter vSAN as the name for the new cluster.
7. Click OK.
8. Once the cluster has been created, right-click the cluster, and click Add Host.

9. Enter the IP address for the first server, and click Next.
10. Enter the root credentials for the server, and click Next.
11. To accept the server's certificate, click Yes.
12. Review the server details, and click Next.
13. Assign the desired license, and click Next.
14. Disable Lockdown mode, and click Next.
15. Click Finish.
16. Repeat steps 10 through 16 for the remaining servers.

Configuring ScaleIO storage

Dell EMC configured the ScaleIO Ready Nodes in both the four-node and eight-node VMware-based ScaleIO deployments using the direct path option. From there, we used the ScaleIO GUI to deploy volumes as follows.

1. Double-click the icon to open the ScaleIO GUI application.
2. Enter the IP address of the management server and click Connect
3. Enter the appropriate credentials and click Login
4. Click the Frontend button, and select Volumes.
5. Right-click pool1 and select add volume.
6. Enter a volume name
7. Check the Create multiple volumes box
8. Enter the appropriate number of copies. For our testing, we created 30 volumes at a time.
9. Enter the starting number for the copied volumes.
10. Click OK.

Mapping the ScaleIO volumes and creating datastores in ESXi

1. While in the Volumes section of the ScaleIO GUI, expand pool1.
2. Find the first unmapped volume, right click on the volume and select Map Volumes.
3. Select all hosts except for the Data Mart host and click Map Volumes.
4. Click Close.
5. Log into vCenter, navigate to Hosts and Clusters, and select any of the hosts in the cluster.
6. Click the Datastores tab.
7. Click New Datastore.
8. Name the new datastore according to the volume name in the ScaleIO GUI (e.g. OSVolume1), select the corresponding LUN and click Next.
9. Select VMFS 6 and click Next.
10. Leave all other settings at default and click Next.
11. Click Finish.
12. Repeat steps 1-12 until all ScaleIO volumes have been mapped and respective datastores created in vCenter.

Creating volumes in HP 3PAR

We created a single RAID 1 CPG using all available drives. We created the LUNs necessary for the environment. For each VM, we created a Virtual Volume Set that contained all the LUNs necessary to each VM.

Creating the storage pool

1. Log into the HPE 3PAR SSMC GUI.
2. At the top, click 3PAR StoreServ, and click Common Provisioning Groups under Block Persona.
3. In the left-hand column, under Common Provisioning Groups, click Create CPG.
4. Under the General heading, enter a name, and verify the correct system is selected in the System drop-down menu.
5. Under the Allocation Settings heading, verify Device type is set to SSD 100K.
6. Verify the RAID type is set to RAID 1.
7. Change Availability to Cage (Default)
8. Click Create.

Creating the Virtual Volumes

1. At the top, click 3PAR StoreServ, and click Virtual Volumes under Block Persona.
2. In the left-hand column, under Virtual Volumes, click Create virtual volume.
3. Under the General heading, enter a name for the VVol.
4. Verify the correct system is selected in the System drop-down menu.
5. Under Provisioning, select Thin Provisioned.
6. Under Compression, select Yes.
7. Uncer CPG, select the RAID 1 SSD group you created.
8. Enter the size of the VVol in the Size field.
9. Click Edit additional settings
10. Under number of volumes, enter the number of identical volumes to be created. We created 30 volumes at a time.
11. Click Create.
12. Repeat steps 1 through 11 as necessary to create all VVols for the environment.

Adding the ESXi hosts

Complete setup for Installing VMware vCenter 6.5 before proceeding with the storage setup.

1. At the top, click 3PAR StoreServ, and click Hosts under Block Persona.
2. In the left-hand column, under Hosts, click Create host.
3. Under the General heading, enter a name for the host in the Name field.
4. Verify the correct system is selected in the System drop-down menu.
5. Set the Host OS to VMware (ESXi)
6. Under the Paths heading, click on Add FC.
7. In the Add FC window, select the WWNs associated with the first host, and click Add.
8. Click Create.
9. Repeat steps 1 through 9 to add the additional hosts.

Creating the Host Sets

1. At the top, click 3PAR StoreServ, and click Host Sets under Block Persona.
2. In the left-hand column, under Host Sets, click Create host set.
3. Under the General heading, enter a name for the host set in the Name field.
4. Under the Host Set Members heading, click on Add hosts.
5. Select all hosts, and click Add.
6. Click Create.

Mapping the HP 3PAR volumes and creating datastores in ESXi

1. At the top of the GUI, click 3PAR StoreServ and then Virtual Volumes.
2. Find the first unexported virtual volume and select it.
3. Click Actions and then Export on the right.
4. Under Export To, click Add.
5. Change the radio button selection to Host Set.
6. Select the appropriate host set and click Add.
7. Click Export and wait for the operation to complete.
8. Log into vCenter, navigate to Hosts and Clusters, and select any of hosts in the cluster.
9. Click the Datastores tab.
10. Click New Datastore.
11. Name the new datastore according to the volume name in the 3PAR GUI (e.g. OSVolume1), select the corresponding LUN and click Next.
12. Select VMFS 6 and click Next.
13. Leave all other settings at default and click Next.
14. Click Finish.
15. Repeat steps 1-14 until all 3PAR virtual volumes have been mapped and respective datastores created in vCenter for each.

Creating the first VM in ESXi

1. In VMware vCenter, navigate to Virtual Machines.
2. To create a new VM, click the icon.
3. Leave Create a new virtual machine selected, and click Next.
4. Enter a name for the virtual machine, and click Next.
5. Place the VM on the desired host with available CPUs, and click Next.
6. Select the appropriate datastore to host the VM, and click Next.
7. In the Customize Hardware section, use the following settings:
   - Set the vCPU count to 4.
   - Set the Memory to 16GB and check the Reserve all guest memory (All locked) checkbox.
   - Add 1 x 80GB VMDK for OS and set to thick provision eager zeroed. Note: You will create the additional VMDKs post-clone.
   - Create three additional VMware Paravirtual SCSI controllers.
   - Attach the OS ISO to the CD/DVD drive.
8. Click Next.
9. Click Finish.

Installing Microsoft® Windows Server® 2016 Datacenter Edition on the first VM

1. Boot the VM to the installation media.
2. Press any key when prompted to boot from DVD.
3. When the installation screen appears, leave language, time/currency format, and input method as default, and click Next.
4. Click Install now.
5. When the installation prompts you, enter the product key.
6. Check I accept the license terms, and click Next.
7. Click Custom: Install Windows only (advanced).
8. Select Windows Server 2016 Datacenter Edition (Desktop Experience), and click Next.
9. Select Drive 0 Unallocated Space, and click Next, at which point Windows begins automatically, and restarts automatically after completing.
10. When the Settings page appears, fill in the Password and Reenter Password fields with the same password.
11. Log in with the password you set up previously.

Installing VMware Tools on the first VM

1. Right-click the VM in vCenter and click Install VMware Tools to mount the appropriate image to the VM's virtual CD-ROM drive.
2. Ensure the VM is powered on and log in as an administrator.
3. Navigate to the virtual CD-ROM drive in the VM and double-click setup.exe to begin the wizard.
4. Follow the wizard and select the Typical installation option.
5. When the VMware Tools installation has completed, restart the VM.

Installing SQL Server® 2016 on the first VM

1. Prior to installing, add the .NET Framework 3.5 feature to the server.
2. Attach the installation media ISO for SQL Server 2016 to the VM.
3. Click Run SETUP.EXE. If Autoplay does not begin the installation, navigate to the SQL Server 2016 DVD, and double-click it.
4. In the left pane, click Installation.
5. Click New SQL Server stand-alone installation or add features to an existing installation.
6. Specify Evaluation as the edition you are installing, and click Next.
7. To accept the license terms, click the checkbox, and click Next.
8. Click Use Microsoft Update to check for updates, and click Next.
9. At the Feature Selection screen, select Database Engine Services, Full-Text and Semantic Extractions for Search, Client Tools Connectivity, and Client Tools Backwards Compatibility.
10. Click Next.
11. At the Instance configuration screen, leave the default selection of default instance, and click Next.
12. At the Server Configuration screen, accept defaults, and click Next.

13. At the Database Engine Configuration screen, select the authentication method you prefer. For our testing purposes, we selected Mixed Mode.
14. Enter and confirm a password for the system administrator account.
15. Click Add Current user. This may take several seconds.
16. Click Next.
17. At the Ready to Install screen, click Install.
18. Close the installation window.
19. In the SQL Server Installation Center, click on Install SQL Server Management Tools.
20. Click Download SQL Server Management Studio.
21. Click Run.
22. When the Microsoft SQL Server Management Studio screen appears, click Install.
23. When the installation completes, click Close.
24. Close the installation window.
25. Shut down the VM, and create clones for the remaining number of VMs onto corresponding OS LUNs on the storage, as well as one additional base image to be transferred to the data mart server.

Creating the additional VMDKs in ESXi

Once the VMs were cloned, we created VMDKs in each of the datastores for database data and log files to correspond to each VM as follows:

- 3 x 60GB VMDKs for database files, with each VMDK assigned to its own Paravirtual SCSI controller.
- 1 x 60GB VMDK for database logs, assigned to the default SCSI controller to be shared with the OS VMDK

We set all VMDKs to thick provision eager zeroed.

Configuring the data mart workload server

We configured a Dell EMC PowerEdge R930 server with VMware ESXi 6.5. We then moved a clone of the Windows Server 2016 and SQL Server 2016 base VM onto this server. We configured the VM as follows:

- 64 vCPUs
- 128GB RAM
- 3 dedicated 2.0TB PCIe flash storage devices for local storage of the data mart files.
- For the HPE solution, we created 8 x 500GB data volumes and 1 x 400GB log virtual volumes and presented the storage to the VM through a dedicated dual-port 16Gb Fibre Channel card.
- For the Dell EMC ScaleIO solution, we configured the data mart server as a ScaleIO Data Client (SDC), then created and mapped 8 x 500GB data volumes and 1 x 400GB log volumes on the ScaleIO ready node hosts to the data mart server.

Configuring the bulk load data mart

We used HammerDB to generate TPCH-compliant source data at scale factor 3000, for a total of 3.31TB of raw data. The generated data exists in the form of pipe-delimited text files, which we placed on NVMe PCIe SSDs for fast reads. We split the six largest tables into 32 separate files for parallel loading. Each chunk had its own table in SQL Server, for a total of 6x32 one-to-one streams. We used batch scripting and SQLCMD to start 32 simultaneous SQL scripts. Each script contained BULK INSERT statements to load the corresponding chunk for each table. For example, the 17th SQL script loaded ORDERS_17.txt into table ORDERS_17, and upon finishing, began loading LINEITEM_17.txt into table LINEITEM_17, and so on through each table.

Generating the data

1. Download HammerDB v2.21 and run hammerdb.bat.
2. Click Options→Benchmark.
3. Select the radio buttons for MSSQL Server and TPC-H, and click OK.
4. In the left pane, expand SQL Server→TPC-H→Datagen, and double-click Options.
5. Select the radio button for 3000, enter a target location for the TPC-H data, and select 32 for the number of virtual users. Click OK.
6. Double-click Generate and click Yes.

Creating the target database

We used the following SQL script to create the target database (some lines have been removed and replaced with an ellipsis for clarity):

```
IF EXISTS (SELECT name FROM master.dbo.sysdatabases WHERE name = 'tpch3000')
    DROP DATABASE tpch3000
GO
CREATE DATABASE tpch3000
ON PRIMARY
(   NAME        = tpch3000_root,
    FILENAME    = 'F:\tpch\tpch_root.mdf',
    SIZE        = 100MB,
    FILEGROWTH  = 100MB),
FILEGROUP   DATA_FG_MISC
(   NAME           = tpch3000_data_ms,
    FILENAME    = 'F:\tpch\tpch_data_ms.mdf',
    SIZE        = 500MB,
    FILEGROWTH  = 100MB),
FILEGROUP DATA_FG_01 (NAME = tpch3000_data_01, FILENAME = 'F:\tpch\tpch_data_01.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB),
FILEGROUP DATA_FG_02 (NAME = tpch3000_data_02, FILENAME = 'G:\tpch\tpch_data_02.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB),
FILEGROUP DATA_FG_03 (NAME = tpch3000_data_03, FILENAME = 'H:\tpch\tpch_data_03.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB),
FILEGROUP DATA_FG_04 (NAME = tpch3000_data_04, FILENAME = 'I:\tpch\tpch_data_04.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB),
FILEGROUP DATA_FG_05 (NAME = tpch3000_data_05, FILENAME = 'J:\tpch\tpch_data_05.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB),
FILEGROUP DATA_FG_06 (NAME = tpch3000_data_06, FILENAME = 'K:\tpch\tpch_data_06.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB),
FILEGROUP DATA_FG_07 (NAME = tpch3000_data_07, FILENAME = 'L:\tpch\tpch_data_07.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB),
FILEGROUP DATA_FG_08 (NAME = tpch3000_data_08, FILENAME = 'M:\tpch\tpch_data_08.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB),
FILEGROUP DATA_FG_09 (NAME = tpch3000_data_09, FILENAME = 'F:\tpch\tpch_data_09.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB),
FILEGROUP DATA_FG_10 (NAME = tpch3000_data_10, FILENAME = 'G:\tpch\tpch_data_10.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB),
FILEGROUP DATA_FG_11 (NAME = tpch3000_data_11, FILENAME = 'H:\tpch\tpch_data_11.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB),
FILEGROUP DATA_FG_12 (NAME = tpch3000_data_12, FILENAME = 'I:\tpch\tpch_data_12.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB),
FILEGROUP DATA_FG_13 (NAME = tpch3000_data_13, FILENAME = 'J:\tpch\tpch_data_13.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB),
FILEGROUP DATA_FG_14 (NAME = tpch3000_data_14, FILENAME = 'K:\tpch\tpch_data_14.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB),
FILEGROUP DATA_FG_15 (NAME = tpch3000_data_15, FILENAME = 'L:\tpch\tpch_data_15.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB),
FILEGROUP DATA_FG_16 (NAME = tpch3000_data_16, FILENAME = 'M:\tpch\tpch_data_16.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB),
FILEGROUP DATA_FG_17 (NAME = tpch3000_data_17, FILENAME = 'F:\tpch\tpch_data_17.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB),
FILEGROUP DATA_FG_18 (NAME = tpch3000_data_18, FILENAME = 'G:\tpch\tpch_data_18.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB),
FILEGROUP DATA_FG_19 (NAME = tpch3000_data_19, FILENAME = 'H:\tpch\tpch_data_19.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB),
FILEGROUP DATA_FG_20 (NAME = tpch3000_data_20, FILENAME = 'I:\tpch\tpch_data_20.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB),
FILEGROUP DATA_FG_21 (NAME = tpch3000_data_21, FILENAME = 'J:\tpch\tpch_data_21.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB),
FILEGROUP DATA_FG_22 (NAME = tpch3000_data_22, FILENAME = 'K:\tpch\tpch_data_22.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB),
FILEGROUP DATA_FG_23 (NAME = tpch3000_data_23, FILENAME = 'L:\tpch\tpch_data_23.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB),
FILEGROUP DATA_FG_24 (NAME = tpch3000_data_24, FILENAME = 'M:\tpch\tpch_data_24.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB),
FILEGROUP DATA_FG_25 (NAME = tpch3000_data_25, FILENAME = 'F:\tpch\tpch_data_25.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB),
```

```
FILEGROUP DATA_FG_26 (NAME = tpch3000_data_26, FILENAME = 'G:\tpch\tpch_data_26.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB),
FILEGROUP DATA_FG_27 (NAME = tpch3000_data_27, FILENAME = 'H:\tpch\tpch_data_27.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB),
FILEGROUP DATA_FG_28 (NAME = tpch3000_data_28, FILENAME = 'I:\tpch\tpch_data_28.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB),
FILEGROUP DATA_FG_29 (NAME = tpch3000_data_29, FILENAME = 'J:\tpch\tpch_data_29.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB),
FILEGROUP DATA_FG_30 (NAME = tpch3000_data_30, FILENAME = 'K:\tpch\tpch_data_30.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB),
FILEGROUP DATA_FG_31 (NAME = tpch3000_data_31, FILENAME = 'L:\tpch\tpch_data_31.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB),
FILEGROUP DATA_FG_32 (NAME = tpch3000_data_32, FILENAME = 'M:\tpch\tpch_data_32.mdf', SIZE =
112640MB, FILEGROWTH  = 100MB)
LOG ON
(   NAME        = tpch3000_log,
    FILENAME    = 'N:\LOG\tpch3000\tpch3000_log.ldf',
    SIZE        = 360000MB,
    FILEGROWTH  = 100MB)
GO
/*set db options*/
ALTER DATABASE tpch3000 SET RECOVERY SIMPLE
ALTER DATABASE tpch3000 SET AUTO_CREATE_STATISTICS OFF
ALTER DATABASE tpch3000 SET AUTO_UPDATE_STATISTICS OFF
ALTER DATABASE tpch3000 SET PAGE_VERIFY NONE
USE tpch3000
GO
create table CUSTOMER_1 ([c_custkey] [bigint] NOT NULL,[c_mktsegment] [char](10) NULL,[c_nationkey]
[int] NULL,[c_name] [varchar](25) NULL,[c_address] [varchar](40) NULL,[c_phone] [char](15) NULL,[c_
acctbal] [money] NULL,[c_comment] [varchar](118) NULL) on DATA_FG_01
create table CUSTOMER_2 ([c_custkey] [bigint] NOT NULL,[c_mktsegment] [char](10) NULL,[c_nationkey]
[int] NULL,[c_name] [varchar](25) NULL,[c_address] [varchar](40) NULL,[c_phone] [char](15) NULL,[c_
acctbal] [money] NULL,[c_comment] [varchar](118) NULL) on DATA_FG_02
…
create table CUSTOMER_32 ([c_custkey] [bigint] NOT NULL,[c_mktsegment] [char](10) NULL,[c_nationkey]
[int] NULL,[c_name] [varchar](25) NULL,[c_address] [varchar](40) NULL,[c_phone] [char](15) NULL,[c_
acctbal] [money] NULL,[c_comment] [varchar](118) NULL) on DATA_FG_32

create table LINEITEM_1 ([l_shipdate] [date] NULL,[l_orderkey] [bigint] NOT NULL,[l_discount] [money]
NOT NULL,[l_extendedprice] [money] NOT NULL,[l_suppkey] [int] NOT NULL,[l_quantity] [bigint] NOT
NULL,[l_returnflag] [char](1) NULL,[l_partkey] [bigint] NOT NULL,[l_linestatus] [char](1) NULL,[l_
tax] [money] NOT NULL,[l_commitdate] [date] NULL,[l_receiptdate] [date] NULL,[l_shipmode] [char]
(10) NULL,[l_linenumber] [bigint] NOT NULL,[l_shipinstruct] [char](25) NULL,[l_comment] [varchar](44)
NULL) on DATA_FG_01
create table LINEITEM_2 ([l_shipdate] [date] NULL,[l_orderkey] [bigint] NOT NULL,[l_discount] [money]
NOT NULL,[l_extendedprice] [money] NOT NULL,[l_suppkey] [int] NOT NULL,[l_quantity] [bigint] NOT
NULL,[l_returnflag] [char](1) NULL,[l_partkey] [bigint] NOT NULL,[l_linestatus] [char](1) NULL,[l_
tax] [money] NOT NULL,[l_commitdate] [date] NULL,[l_receiptdate] [date] NULL,[l_shipmode] [char]
(10) NULL,[l_linenumber] [bigint] NOT NULL,[l_shipinstruct] [char](25) NULL,[l_comment] [varchar](44)
NULL) on DATA_FG_02
…
create table LINEITEM_32 ([l_shipdate] [date] NULL,[l_orderkey] [bigint] NOT NULL,[l_discount]
[money] NOT NULL,[l_extendedprice] [money] NOT NULL,[l_suppkey] [int] NOT NULL,[l_quantity] [bigint]
NOT NULL,[l_returnflag] [char](1) NULL,[l_partkey] [bigint] NOT NULL,[l_linestatus] [char](1) NULL,[l_
tax] [money] NOT NULL,[l_commitdate] [date] NULL,[l_receiptdate] [date] NULL,[l_shipmode] [char]
(10) NULL,[l_linenumber] [bigint] NOT NULL,[l_shipinstruct] [char](25) NULL,[l_comment] [varchar](44)
NULL) on DATA_FG_32

create table ORDERS_1 ([o_orderdate] [date] NULL,[o_orderkey] [bigint] NOT NULL,[o_custkey] [bigint]
NOT NULL,[o_orderpriority] [char](15) NULL,[o_shippriority] [int] NULL,[o_clerk] [char](15) NULL,[o_
orderstatus] [char](1) NULL,[o_totalprice] [money] NULL,[o_comment] [varchar](79) NULL) on DATA_FG_01
create table ORDERS_2 ([o_orderdate] [date] NULL,[o_orderkey] [bigint] NOT NULL,[o_custkey] [bigint]
NOT NULL,[o_orderpriority] [char](15) NULL,[o_shippriority] [int] NULL,[o_clerk] [char](15) NULL,[o_
orderstatus] [char](1) NULL,[o_totalprice] [money] NULL,[o_comment] [varchar](79) NULL) on DATA_FG_02
…
create table ORDERS_32 ([o_orderdate] [date] NULL,[o_orderkey] [bigint] NOT NULL,[o_custkey] [bigint]
NOT NULL,[o_orderpriority] [char](15) NULL,[o_shippriority] [int] NULL,[o_clerk] [char](15) NULL,[o_
```

```
orderstatus] [char](1) NULL,[o_totalprice] [money] NULL,[o_comment] [varchar](79) NULL) on DATA_FG_32

create table PART_1 ([p_partkey] [bigint] NOT NULL,[p_type] [varchar](25) NULL,[p_size] [int]
NULL,[p_brand] [char](10) NULL,[p_name] [varchar](55) NULL,[p_container] [char](10) NULL,[p_mfgr]
[char](25) NULL,[p_retailprice] [money] NULL,[p_comment] [varchar](23) NULL) on DATA_FG_01
create table PART_2 ([p_partkey] [bigint] NOT NULL,[p_type] [varchar](25) NULL,[p_size] [int]
NULL,[p_brand] [char](10) NULL,[p_name] [varchar](55) NULL,[p_container] [char](10) NULL,[p_mfgr]
[char](25) NULL,[p_retailprice] [money] NULL,[p_comment] [varchar](23) NULL) on DATA_FG_02
…
create table PART_32 ([p_partkey] [bigint] NOT NULL,[p_type] [varchar](25) NULL,[p_size] [int]
NULL,[p_brand] [char](10) NULL,[p_name] [varchar](55) NULL,[p_container] [char](10) NULL,[p_mfgr]
[char](25) NULL,[p_retailprice] [money] NULL,[p_comment] [varchar](23) NULL) on DATA_FG_32

create table PARTSUPP_1 ([ps_partkey] [bigint] NOT NULL,[ps_suppkey] [int] NOT NULL,[ps_supplycost]
[money] NOT NULL,[ps_availqty] [int] NULL,[ps_comment] [varchar](199) NULL) on DATA_FG_01
create table PARTSUPP_2 ([ps_partkey] [bigint] NOT NULL,[ps_suppkey] [int] NOT NULL,[ps_supplycost]
[money] NOT NULL,[ps_availqty] [int] NULL,[ps_comment] [varchar](199) NULL) on DATA_FG_02
…
create table PARTSUPP_32 ([ps_partkey] [bigint] NOT NULL,[ps_suppkey] [int] NOT NULL,[ps_supplycost]
[money] NOT NULL,[ps_availqty] [int] NULL,[ps_comment] [varchar](199) NULL) on DATA_FG_32

create table SUPPLIER_1 ([s_suppkey] [int] NOT NULL,[s_nationkey] [int] NULL,[s_comment] [varchar]
(102) NULL,[s_name] [char](25) NULL,[s_address] [varchar](40) NULL,[s_phone] [char](15) NULL,[s_
acctbal] [money] NULL) on DATA_FG_01
create table SUPPLIER_2 ([s_suppkey] [int] NOT NULL,[s_nationkey] [int] NULL,[s_comment] [varchar]
(102) NULL,[s_name] [char](25) NULL,[s_address] [varchar](40) NULL,[s_phone] [char](15) NULL,[s_
acctbal] [money] NULL) on DATA_FG_02
…
create table SUPPLIER_32 ([s_suppkey] [int] NOT NULL,[s_nationkey] [int] NULL,[s_comment] [varchar]
(102) NULL,[s_name] [char](25) NULL,[s_address] [varchar](40) NULL,[s_phone] [char](15) NULL,[s_
acctbal] [money] NULL) on DATA_FG_32

Inserting the data into Microsoft SQL Server
We used 32 individual SQL scripts to create a BULK INSERT process on each filegroup. The first script
is shown here as an example:

bulk insert tpch3000..CUSTOMER_1 from 'O:\CUSTOMER_1.tbl' with
(TABLOCK,DATAFILETYPE='char',CODEPAGE='raw',FieldTerminator='|',BATCHSIZE=14062500)
bulk insert tpch3000..LINEITEM_1 from 'O:\LINEITEM_1.tbl' with
(TABLOCK,DATAFILETYPE='char',CODEPAGE='raw',FieldTerminator='|',BATCHSIZE=562500000)
bulk insert tpch3000..ORDERS_1 from 'O:\ORDERS_1.tbl' with
(TABLOCK,DATAFILETYPE='char',CODEPAGE='raw',FieldTerminator='|',BATCHSIZE=140625000)
bulk insert tpch3000..PART_1 from 'O:\PART_1.tbl' with
(TABLOCK,DATAFILETYPE='char',CODEPAGE='raw',FieldTerminator='|',BATCHSIZE=18750000)
bulk insert tpch3000..PARTSUPP_1 from 'O:\PARTSUPP_1.tbl' with
(TABLOCK,DATAFILETYPE='char',CODEPAGE='raw',FieldTerminator='|',BATCHSIZE=75000000)
bulk insert tpch3000..SUPPLIER_1 from 'O:\SUPPLIER_1.tbl' with
(TABLOCK,DATAFILETYPE='char',CODEPAGE='raw',FieldTerminator='|',BATCHSIZE=937500)
```

Starting the SQL BULK INSERT scripts

We used Windows CMD and SQLCMD to start the 32 BULK INSERT scripts with CPU affinity:

```
start /node 0 /affinity 1 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\Administrator\
Documents\gen_1.sql
start /node 0 /affinity 2 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\Administrator\
Documents\gen_2.sql
start /node 0 /affinity 4 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\Administrator\
Documents\gen_3.sql
start /node 0 /affinity 8 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\Administrator\
Documents\gen_4.sql
start /node 0 /affinity 10 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\Administrator\
Documents\gen_5.sql
start /node 0 /affinity 20 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\Administrator\
Documents\gen_6.sql
start /node 0 /affinity 40 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\Administrator\
Documents\gen_7.sql
```

```
start /node 0 /affinity 80 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\Administrator\
Documents\gen_8.sql
start /node 0 /affinity 100 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\Administrator\
Documents\gen_9.sql
start /node 0 /affinity 200 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\Administrator\
Documents\gen_10.sql
start /node 0 /affinity 400 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\Administrator\
Documents\gen_11.sql
start /node 0 /affinity 800 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\Administrator\
Documents\gen_12.sql
start /node 0 /affinity 1000 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\
Administrator\Documents\gen_13.sql
start /node 0 /affinity 2000 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\
Administrator\Documents\gen_14.sql
start /node 0 /affinity 4000 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\
Administrator\Documents\gen_15.sql
start /node 0 /affinity 8000 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\
Administrator\Documents\gen_16.sql
start /node 1 /affinity 1 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\Administrator\
Documents\gen_17.sql
start /node 1 /affinity 2 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\Administrator\
Documents\gen_18.sql
start /node 1 /affinity 4 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\Administrator\
Documents\gen_19.sql
start /node 1 /affinity 8 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\Administrator\
Documents\gen_20.sql
start /node 1 /affinity 10 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\Administrator\
Documents\gen_21.sql
start /node 1 /affinity 20 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\Administrator\
Documents\gen_22.sql
start /node 1 /affinity 40 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\Administrator\
Documents\gen_23.sql
start /node 1 /affinity 80 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\Administrator\
Documents\gen_24.sql
start /node 1 /affinity 100 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\Administrator\
Documents\gen_25.sql
start /node 1 /affinity 200 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\Administrator\
Documents\gen_26.sql
start /node 1 /affinity 400 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\Administrator\
Documents\gen_27.sql
start /node 1 /affinity 800 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\Administrator\
Documents\gen_28.sql
start /node 1 /affinity 1000 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\
Administrator\Documents\gen_29.sql
start /node 1 /affinity 2000 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\
Administrator\Documents\gen_30.sql
start /node 1 /affinity 4000 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\
Administrator\Documents\gen_31.sql
start /node 1 /affinity 8000 sqlcmd -S localhost -d tpch3000 -U sa -P ****** -i C:\Users\
Administrator\Documents\gen_32.sql
```

DVD Store 2 Benchmark Configuration

Data generation overview

We generated the data using the Install.pl script included with DVD Store version 2.1 (DS2), providing the parameters for our 40GB database size and the Microsoft SQL Server 2016 platform. We ran the Install.pl script on a utility system running Linux®. The Install.pl script also generated the database schema.

After processing the data generation, we transferred the data files and schema creation files to a Windows-based system running SQL Server 2014. We built the 40GB database in SQL Server 2014, and then performed a full backup, storing the backup file on the C: drive for quick access. We used that backup file to restore the database to the SQL Server 2016-based VMs between test runs.

The only modification we made to the schema creation scripts were the specified file sizes for our database. We explicitly set the file sizes higher than necessary to ensure that no file-growth activity would affect the outputs of the test. Besides this file size modification, the database schema was created and loaded according to the DVD Store documentation. Specifically, we followed the steps below:

1. We generated the data and created the database and file structure using database creation scripts in the DS2 download. We made size modifications specific to our 40GB database and the appropriate changes to drive letters.
2. We transferred the files from our Linux data generation system to a Windows system running SQL Server.

3. We created database tables, stored procedures, and objects using the provided DVD Store scripts.
4. We set the database recovery model to bulk-logged to prevent excess logging.
5. 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.
6. We created indices, full-text catalogs, primary keys, and foreign keys using the database-creation scripts.
7. We updated statistics on each table according to database-creation scripts, which sample 18 percent of the table data.
8. On the SQL Server instance, we created a ds2user SQL Server login using the following Transact-SQL (T-SQL) script:

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

9. We set the database recovery model back to full.
10. We created the necessary full text index using SQL Server Management Studio.
11. We created a database user and mapped this user to the SQL Server login.
12. 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.

Logical name	Filegroup	Initial size (MB)
Database files		
primary	PRIMARY	4
cust1	DS_CUST_FG	6,000
cust2	DS_CUST_FG	6,000
cust3	DS_CUST_FG	6,000
cust4	DS_CUST_FG	6,000
ind1	DS_IND_FG	3,205
ind2	DS_IND_FG	3,205
ind3	DS_IND_FG	3,205
ind4	DS_IND_FG	3,205
ds_misc	DS_MISC_FG	200
orders1	DS_ORDERS	3,000
orders2	DS_ORDERS	3,000
orders3	DS_ORDERS	3,000
orders4	DS_ORDERS	3,000
Log files		
ds_log	Not applicable	24,781

Configuring the database workload client

For our testing, we used a virtual client for the Microsoft SQL Server client. To create this client, we installed Windows Server 2012, assigned a static IP address, and installed .NET 3.5.

Running the DVD Store tests

We created a series of batch files, SQL scripts, and shell scripts to automate the complete test cycle. DVD Store outputs an orders-per-minute metric, which is a running average calculated through the test. In this report, we report the last OPM reported by each client/target pair. Each complete test cycle consisted of general steps:

1. Clean up prior outputs from the target system and the client driver system.
2. Drop the databases from the target.
3. Restore the databases on the target.
4. Shut down the target.
5. Reboot the host and client system.
6. Wait for a ping response from the server under test and the client system.
7. Let the test server idle for 10 minutes.
8. Start the DVD Store driver on the client.

We used the following DVD Store parameters for testing:

```
ds2sqlserverdriver.exe --target=<target_IP> --ramp_rate=10 --run_time=180 --n_threads=32 --db_
size=40GB --think_time=0.1 --detailed_view=Y --warmup_time=15 --report_rate=1 --csv_output=<drive
path>
```

This project was commissioned by Dell EMC.