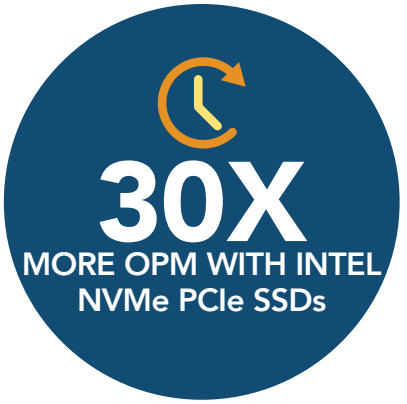
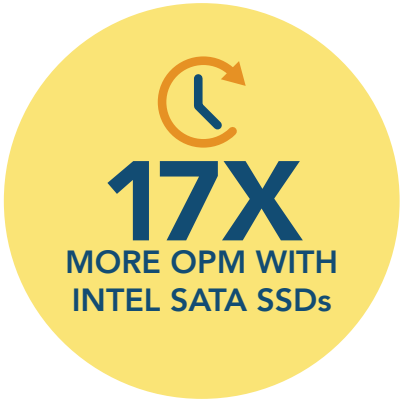
Get more from your new server by choosing faster drives

New Intel SSDs with NVMe delivered more database orders per minute for storage-intensive applications





Maybe your users have begun to grumble about wait times. Or worse, your customers are abandoning your website. Whether the database applications that drive your business are in the cloud or in your datacenter, performance matters. When the servers running these applications can no longer meet demand, it's past time to upgrade.

Storage plays a significant role in how much work a server can accomplish. Updated processors won't be of any help if they spend most of the time waiting on slow storage. Eliminating a storage bottleneck with speedy new solid state drives from Intel can enable your new server to handle more database transactions in a given amount of time than your old server with hard disk drives.

To see what improvement you can expect, we set up three database server configurations: a 4-year-old server with HDDs and a new server powered by Intel® Xeon® processors E5-2699 v4 with either six Intel S3520 SATA SSDs or three Intel P3520 NVMe™ PCIe SSDs. On each, we set up a Microsoft® Hyper-V® environment with virtual machines running SQL Server® 2016 and ran a database workload that measures orders per minute, or OPM.

With Intel SATA SSDs, the new server processed 17 times more OPM than the older server, and with the Intel NVMe PCIe SSDs, the new server handled 30 times more more OPM while using only half the number of drives. This could make either of them great additions to your new server. But which solution is the best choice for your datacenter? We'll take a holistic look at both options in this report.

Reach for the sky

Many companies use the cloud to sell products, maintain records, and carry out other business activities. Behind the scenes, the cloud is actually a group of physical servers —which can reside on-premises—that require storage to power the applications that customers use. When these hardware resources are robust and up to date, they deliver a good experience to users. When they are outdated and slow, performance falters. If you're in the market for new servers to support your in-house cloud applications, you can maximize their power by including Intel solid state storage.

Time to upgrade

Joel handles IT for a company that offers training in business skills. They use a cloud-based registration system that currently runs on a group of servers they acquired roughly 4 years ago. SSDs were prohibitively expensive at the time, so they outfitted their servers with traditional hard drives.

This hardware served them well, but as the business has grown, keeping up with demand has became a challenge. More and more visitors to their website leave before registering for a training course and Joel is convinced slow response times are at least partly to blame.

He got the green light to upgrade, so he's doing his research. The new servers will have Intel Xeon processors E5-2699 v4 and—yes!—SSDs. Joel has narrowed the choice to two options: Intel S3520 SATA SSDs and Intel P3520 NVMe PCIe SSDs.

He's leaning toward the NVMe drives. SATA drives are a definite step up from HDDs, but NVMe SSDs are even more powerful. They cost a bit more, but Joel suspects that this option would provide better value to his company by letting them use fewer drives or even fewer servers.

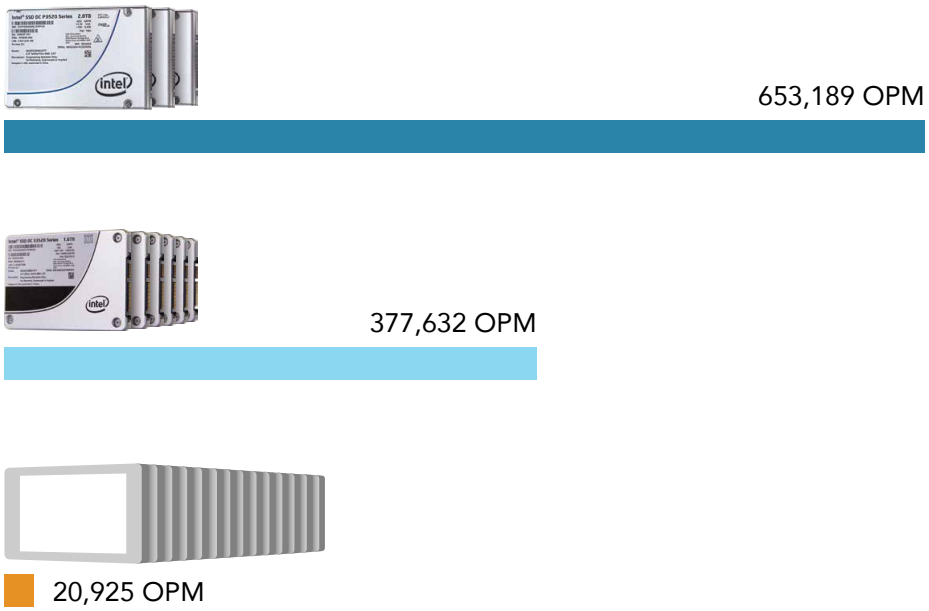
If you're in the market for new servers to support your in-house cloud applications, you can maximize their power by including Intel solid state storage.

Putting Joel's theory to the test

To compare the database performance improvements the two upgrade options could offer, we began with a four-year-old server powered by Intel Xeon processors E5-2680 with hard drives. We created eight database VMs and ran a workload that simulates an online ordering system.

Then, we set up a new server powered by Intel Xeon processors E5-2699 v4 server—first with six Intel S3520 SATA SSDs and then with three Intel P3520 NVMe PCIe SSDs. On both storage configurations we created 22 VMs and ran the same workload we ran on the older server. As the chart below shows, both SSD configurations of the new server handily outperformed the old server with HDD storage, but the NVMe SSDs did nearly twice the total work of the SATA drives. Put another way, the new server with SATA storage processed 17 times more database orders than the older server did and the new server with NVMe storage processed 30 times more orders.

Total database orders per minute across all VMs



Which SSD processed more orders per dollar?

Joel thought that the Intel P3520 NVMe PCIe SSDs would be more cost-effective than the Intel S3520 SATA SSDs despite their slightly higher price. To find out, we calculated the price of the three NVMe drives and the six SATA SSDs that the new server would use for storing data and logs. We didn't include the price of the server itself or the drives that would hold the operating system because those costs would remain constant regardless of drive type.

Next, we took the total number of OPM each server configuration achieved and divided it by the price of the SSDs used for storage. As the following table shows, the NVMe drives delivered an OPM-per-dollar figure of 221.3, which is 2.6 times the 85.2 OPM per dollar the SATA drives delivered. This means that the NVMe drives would give Joel's company an excellent return on their investment.

Solid state drives used for storing data and logs	Price (USD as of October 2016)	Performance (Orders per minute)	OPM/dollar
Intel S3520 SATA SSDs (six @ $739 each)	$4,434	377,632	85.2
Intel P3520 NVMe PCIe SSDs (three @ $984 each)	$2,952	653,189	221.3
Performance/dollar advantage with Intel P3520 NVMe PCIe SSDs			**2.6X**

Greater performance could mean fewer servers

In addition to providing users and customers with a better experience, the performance boost from using SSDs in your new servers could let your company support a given database workload with a smaller number of servers. This means spending less on hardware, of course, and can also lead to savings on software licensing, power and cooling, administrative management time, and even the amount of datacenter space required.

Conclusion

Now that SSDs have become more affordable, including them in your new servers makes great business sense. In our testing, both Intel S3520 SATA SSDs and Intel P3520 NVMe PCIe SSDs helped the new server we tested deliver many times more database work than a four-year-old server with hard drives—17 times and 30 times respectively.

While both Intel SSD options are strong, we saw greater performance per dollar from three NVMe SSDs than from six SATA SSDs. This confirms Joel's belief that Intel P3520 NVMe PCIe SSDs would bring greater value to his company.

On September 9, 2016, 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 September 12, 2016.

Appendix A – About the hardware and software we tested

About the new Intel Xeon E5 v4 processor family

The latest Intel Xeon processor E5-2600 v4 family contain up to 22 cores at that run at 2.2 GHz – a 20 percent increase compared to previous generations. The Intel Xeon processor E5-2600 v4 family offers features to enhance virtualization, orchestration, and security, including cache monitoring and allocation technologies, memory bandwidth monitoring, and enhanced key security.

To learn more about the Intel Xeon E5 processor product family, read the product brief at http://www.intel.com/content/dam/www/public/us/en/documents/product-briefs/xeon-e5-brief.pdf

About Microsoft SQL Server 2016

SQL Server 2016 delivers performance and security features built-in for mission-critical transactional systems and data warehouses, along with an integrated business intelligence and advanced analytics solution for building data analytics applications. High-performance features are key to ensuring your applications can deliver a smooth transactional experience and also potentially support demanding real-time operational analytics.

You can experience the full features of SQL Server 2016 through the free developer edition. Visit the SQL Server 2016 webpage to learn more about new features and download the SQL Server 2016 e-book.

About DVD Store Version 2.1

To create our real-world ecommerce workload, we used the DVD Store Version 2.1 benchmarking tool. DS2 models an online DVD store, where customers log in, search for movies, and make purchases. DS2 reports these actions in orders per minute that the system could handle, to show what kind of performance you could expect for your customers. The DS2 workload also performs other actions, such as adding new customers, to exercise the wide range of database functions you would need to run your ecommerce environment.

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

Appendix B – Test systems

Server configuration information	Server with Intel Xeon processors E5-2680 and HDDs	Server with Intel Xeon processors E5-2699 v4 and SATA SSDs	Server with Intel Xeon processors E5-2699 v4 and NVMe PCIe SSDs
BIOS name and version	Intel SE5C610.86B.01.01.0016.033120161139	Intel SE5C600.86B.02.06.0002.101320150901	Intel SE5C610.86B.01.01.0016.033120161139
Non-default BIOS settings	Virtualization enabled	Virtualization enabled	Virtualization enabled
Operating system name and version/build number	Microsoft® Windows Server® 2016 Datacenter build 14300.1045	Microsoft® Windows Server® 2016 Datacenter build 14300.1045	Microsoft® Windows Server® 2016 Datacenter build 14300.1045
Date of last OS updates/patches applied	8/7/2016	8/8/2016	8/7/2016
Power management policy	Performance	Performance	Performance
Processor			
Number of processors	2	2	2
Vendor and model	Intel Xeon E5-2680	Intel Xeon E5-2699 v4	Intel Xeon E5-2699 v4
Core count (per processor)	8	22	22
Core frequency (GHz)	2.70	2.20	2.20
Stepping	7	1	1
Memory module(s)			
Total memory in system (GB)	64	256	256
Number of memory modules	8	8	8
Vendor and model	Kingston® 9965516-048.A00LF	Micron® 36ASF4G72PZ-2G3A1	Micron 36ASF4G72PZ-2G3A1
Size (GB)	8	32	32
Type	PC3-10600	PC4-2400	PC4-2400
Speed (MHz)	1,333	2,400	2,400

Server configuration information	Server with Intel Xeon processors E5-2680 and HDDs	Server with Intel Xeon processors E5-2699 v4 and SATA SSDs	Server with Intel Xeon processors E5-2699 v4 and NVMe PCIe SSDs
Speed running in the server (MHz)	1,333	2,400	2,400
Storage controller			
Vendor and model	LSI MegaRAID SAS 9265-8i	LSI® MegaRAID® SAS 9265-8i	Intel Embedded Server RAID Technology 2
Cache size	1GB	1GB	N/A
Firmware version	3.460.95-6434	3.460.95-6434	N/A
Driver version	6.710.11.00	6.710.11.00	17.01.2016.0216
Local storage			
Disk #1 (OS)			
Number of drives	2	2	2
Drive vendor and model	Seagate® ST9250610NS	Intel SSDSC2BB240G6	Intel SSDSC2BB016T6
Drive size (GB)	250	240	1,600
Drive information v(speed, interface, type)	7.2K, 6 Gb/s SATA, HDD	6 Gb/s SATA, SSD	6 Gb/s SATA, SSD
Disk #2 (data and logs)			
Number of drives	14	6	3
Drive vendor and model	Seagate ST1200MM0017	Intel SSDSC2BB016T6	Intel SSDPEDMX02
Drive size (GB)	1,200	1,600	2,000
Drive information (speed, interface, type)	10K, 6Gb/s SAS, HDD	6 Gb/s SATA, SSD	NVMe SSD
Network adapter			
Vendor and model	Intel I350 Gigabit	Intel I350 Gigabit	Intel I350 Gigabit
Number and type of ports	4 x 1GbE	2 x 1GbE	2 x 1GbE
Driver version	12.14.7.0	12.14.7.0	12.14.7.0

Server configuration information	Server with Intel Xeon processors E5-2680 and HDDs	Server with Intel Xeon processors E5-2699 v4 and SATA SSDs	Server with Intel Xeon processors E5-2699 v4 and NVMe PCIe SSDs
Cooling fans			
Vendor and model	Nidec UltraFlo V60E12BS1B5-07A014	Nidec® UltraFlo® V60E12BS1B5-07A016	Nidec UltraFlo V60E12BS1B5-07A016
Number of cooling fans	6	6	6
Power supplies			
Vendor and model	Delta DPS-750XB	Delta® DPS-750XB	Delta DPS-750XB
Number of power supplies	2	2	2
Wattage of each (W)	750	750	750

Appendix C – How we tested

Installing Windows Server 2016 Tech Preview 5 Datacenter Edition

We installed the host operating system on direct-attached SAS drives placed into a RAID1 pair. After completing installation, we disabled the firewall and IPsec, enabled Remote Desktop, and ran Windows updates.

1. Connect the installation media to the server. We used a USB boot drive.
2. Boot the server to the Windows Server installer.
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. Select Windows Server 2016 Datacenter Edition (with Desktop), and click Next.
7. Check I accept the license terms, and click Next.
8. Click Custom: Install Windows only (advanced).
9. Select Drive 0 Unallocated Space, and click Next. This starts Windows automatically, and Windows will restart 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.

Configuring the HDDs and SATA SSDs

We used the Intel hardware RAID adapter to create virtual disks for the HDD and SATA SSD configs. In both cases, we created RAID10 virtual disks. On the HDDs, we created a virtual disk using 10 drives for the OS and data virtual hard drives (VHDs), and a second virtual disk using 4 drives for the log VHDs. On the SATA SSDs, we created a single virtual disk to hold all VHDs, and turned off all caching.

Configuring the NVMe SSDs

We used Microsoft Storage Spaces to create a software RAID 0 virtual disk using all 3 NVMe SSDs. We placed all VHDs on this virtual disk.

Creating a volume using Microsoft Storage Spaces

1. Log into the NVMe-equipped host.
2. In Server Manager, on the left side, click File and Storage Services→Storage Pools.
3. In the Storage Pools pane, click the Tasks drop-down menu, and click New Storage Pool.
4. Click Next.
5. Give the Storage Pool a name, and select the pool of NVMe drives.
6. Click Next.
7. Select all NVMe drives, and click Next.
8. Click Create.
9. Once the operation completes, close the Storage Pool creation wizard.
10. In the Virtual Disks pane, click the Tasks drop-down menu, and click New Virtual Disk.
11. Click Next.
12. Select the Storage Pool created with the previous wizard, and click Next.
13. Give the virtual disk a name, and click Next.
14. Click Next.
15. Select Simple, and click Next.
16. Click Next.
17. Click Next.
18. Select the Maximum Size radio button, and click Next.
19. Click Create.
20. When the operation completes, close the Virtual Disk creation wizard.

21. In the left pane, click Disks.
22. In the Volume pane, click the Tasks drop-down menu, and click New Volume.
23. Click Next.
24. Select the Virtual Disk created in the previous wizard, and click Next.
25. Select the Maximum Size radio button, and click Next.
26. Give the volume a drive letter, and click Next.
27. Click Next.
28. Click Create.
29. Exit the wizard after the task completes.

Creating the workload virtual machines

We created eight VMs on the legacy HDD-equipped server and 22 VMs on the SATA/NVMe SSD-equipped server with Windows Server 2016 Technical Preview 5 Datacenter Edition and SQL Server 2016 to run our test workload. We created three VHDs for each VM: a 60GB VHD for the operating system, a 60GB VHD for SQL data, and a 30GB VHD for SQL logs. On the HDD-configured host, we placed the OS and Data VHDs on one RAID group and the Logs VHDs on another. On the SATA and NVMe SSD-configured hosts, we placed all VHDs in a single large RAID group.

Creating the base VM

1. In Hyper-V Manager, click New→Virtual Machine.
2. Click Next.
3. Give the VM a name.
4. Check the Store the virtual machine in a different location checkbox, and enter the OS/Data virtual disk drive letter.
5. Click Next.
6. Select the Generation 2 checkbox, and click Next.
7. Set the Startup memory to 8,192 MB, and click Next.
8. Select the testing virtual switch from the drop-down menu, and click Next.
9. For the first virtual hard disk, enter the name OS, set the location to be the OS/Data virtual disk, and set the size to be 60GB. Click Next.
10. Click Finish.
11. Right-click the newly created VM, and click Settings.
12. Click Processor and increase the number of virtual CPUs to 8.
13. Click Add Hardware, select SCSI Controller, and click Add.
14. Select Hard Drive, and click Add.
15. Click New.
16. Click Next.
17. Select the Fixed Size radio button, and click Next.
18. Name the VHD Data, and set the location to be the OS/Data virtual disk. Click Next.
19. Set the size of the VHD to be 60GB, and click Finish.
20. Repeat steps 13 through 19 to add a new SCSI Controller and a new 30GB VHD placed on the Logs virtual disk.
21. Select the OS SCSI Controller, select DVD Drive, and click Add.
22. Select the Image File radio button, and click Browse.
23. Browse to the location of the Windows Server 2016 Technical Preview 5 installation media.
24. Click Apply.
25. Click OK.
26. Right-click the VM, and click Start to power on the VM.
27. Right-click the VM, and click Connect to connect to the VM console.
28. Follow the instructions in the above Windows Server installation section to install Windows Server 2016 Technical Preview 5 onto the VM.

Installing .NET Framework 3.5

1. In the VM, open Server Manager.
2. Click Manage→Add Roles and Features.
3. Select Role-based or feature-based installation, and click Next.
4. Under Server pool, select the local server, and click Next. Click Next.

5. Under Features, select .NET Framework 3.5 Features, and click Next.
6. Click Install.
7. Upon completion, click Close.

Installing SQL Server 2016

1. Attach the installation media ISO for SQL Server 2016 to the VM.
2. Click Run SETUP.EXE. If Autoplay does not begin the installation, navigate to the SQL Server 2016 DVD, and double-click it.
3. In the left pane, click Installation.
4. Click New SQL Server stand-alone installation or add features to an existing installation.
5. To accept the license terms, click the checkbox, and click Next.
6. Click Use Microsoft Update to check for updates, and click Next.
7. To install the setup support files, click Install.
8. If there are no failures displayed, click Next.
9. At the Setup Role screen, choose SQL Server Feature Installation, and click Next.
10. At the Feature Selection screen, select Database Engine Services, Full-Text and Semantic Extractions for Search, Client Tools Connectivity, Client Tools Backwards Compatibility, Management Tools – Basic, and Management Tools – Complete.
11. Click Next.
12. At the Instance configuration screen, leave the default selection of default instance, 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 Error and usage reporting screen, click Next.
18. At the Installation Configuration Rules screen, check that there are no failures or relevant warnings, and click Next.
19. At the Ready to Install screen, click Install.
20. Close the installation window.

Enabling Lock pages in memory

1. Inside the VM, click Start, type `gpedit.msc`, and press Enter.
2. In the Local Group Policy Editor, expand Computer Configuration→Windows Settings→Security Settings→Local Policies.
3. Select User Rights Assignment, and in the right pane, scroll down to Lock pages in memory.
4. Right-click Lock pages in memory, and click Properties.
5. Click Add User or Group, and add your SQL user account.
6. Click OK.

Configuring the DVD Store 2 benchmark

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 to build the database. We then increased the compatability level to SQL Server 2016, performed a full backup, and stored the backup file on the C: drive for quick access. We used that backup file to restore the server 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. Apart from this file size modification, we created and loaded the database schema according to the DVD Store documentation. Specifically, we followed the steps below:

1. Generate the data and create the database and file structure using database creation scripts in the DS2 download. Make size modifications specific to 40GB database and the appropriate changes to drive letters.
2. Transfer the files from the Linux data generation system to a Windows system running SQL Server 2014.
3. Create database tables, stored procedures, and objects using the provided DVD Store scripts.
4. Set the database recovery model to bulk-logged to prevent excess logging.
5. Load the data generated into the database. For data loading, use the import wizard in SQL Server Management Studio. Where necessary, retain options from the original scripts, such as Enable Identity Insert.
6. Create indices, full-text catalogs, primary keys, and foreign keys using the database-creation scripts.
7. Update statistics on each table according to database-creation scripts, which sample 18 percent of the table data.
8. On the SQL Server instance, create 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. Set the database recovery model back to full.
10. Create the necessary full text index using SQL Server Management Studio.
11. Create a database user and map this user to the SQL Server login.
12. Perform a full backup of the database. Note: This backup allowed us to restore the databases to a pristine state relatively quickly between tests.

The table below shows our initial file size modifications.

Logical name	Filegroup	Initial size (MB)
Database files		
primary	PRIMARY	5
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	3,206
ind3	DS_IND_FG	3,205
ind4	DS_IND_FG	3,204
ds_misc1	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 2008 R2, 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 that each client/target pair reported.

1. Each complete test cycle consisted of general steps:
2. Clean up prior outputs from the target system and the client driver system.
3. Drop the database from the target.
4. Restore the database on the target.
5. Shut down the target.
6. Reboot the host and client system.
7. Wait for a ping response from the server under test and the client system.
8. Let the test server idle for 10 minutes.
9. 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=30 --n_threads=32 --db_size=40GB
--think_time=0 --detailed_view=Y --warmup_time=15 --report_rate=1 --csv_output=<drive path>
```

This project was commissioned by Intel.