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

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

2.2x the transactions per minute of a configuration with two NVMe drives alone

11.3x the transactions per minute of a configuration with 12 SATA SSDs alone

Watch your transactional database performance climb with Intel Optane DC persistent memory

Dell EMC PowerEdge R740xd servers with Intel Optane DC persistent memory handled more transactions per minute than configurations with NAND flash NVMe drives or SATA SSDs

While enterprises offer different goods and services, they're on the same quest to deliver speedy performance to end users and increase profits by making the most out of their resources. Organizations that need exceptional transactional database performance should take note of a new memory technology that could help them meet their goals: Intel® Optane™ DC persistent memory. This new technology looks like memory and offers quick reads and writes, but the 256GB capacity of each DCPMM lets it act like traditional storage and store entire databases on a DCPMM stick.

Through our tests in the Principled Technologies data center, we found that using Intel Optane DC persistent memory in a Dell EMC™ PowerEdge™ R740xd server delivered 2.2 times the Microsoft® SQL Server® 2019 performance of a two-NVMe™ drive configuration and improved performance even more significantly over SATA SSDs—delivering an impressive 11.3 times the transactions per minute. As your data center approaches its peak database performance capabilities, consider adding Intel Optane DC persistent memory—our tests suggest that it might help transactional database performance.

Watch your transactional database performance climb with Intel Optane DC persistent memory October 2019

The science behind the report:

Watch your transactional database performance climb with Intel Optane DC persistent memory

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 Watch your transactional database performance climb with Intel Optane DC persistent memory.

We concluded our hands-on testing on October 24, 2019. During testing, we determined the appropriate hardware and software configurations and applied updates as they became available. The results in this report reflect hardware configurations that we finalized on September 19, 2019 and software configurations that we finalized on October 11, 2019. Unavoidably, these configurations may not represent the latest versions available when this report appears.

Our results

The table below presents our findings in detail.

Configuration	Transactions per second	Transactions per minute
12x Intel® Optane™ DC persistent memory (Intel 8Rx4 PC4-2666V DCPIMM)		
Run 1	38,393	2,303,562
Run 2	38,854	2,331,259
Run 3	37,204	2,232,252
2x NAND flash NVMe™ SSDs (Intel P4610)		
Run 1	16,824	1,009,410
Run 2	17,597	1,055,835
Run 3	17,522	1,051,326
4x NAND flash NVMe SSDs (Intel P4610)		
Run 1	22,547	1,352,814
Run 2	21,641	1,298,440
Run 3	22,834	1,370,020
12x SATA SSDs (Intel D3-S4510)		
Run 1	3,383	202,984
Run 2	3,377	202,609
Run 3	3,488	209,286

System configuration information

The table below presents detailed information on the systems we tested.

Server configuration information	Dell EMC™ PowerEdge™ R740xd
BIOS name and version	Dell 2.3.1
Non-default BIOS settings	Performance Mode
Operating system name and version/build number	VMware® vSphere™ ESXi 6.7.0 Update 3 Build-14320388 (X09)
Date of last OS updates/patches applied	09/19/2019
Power management policy	High Performance
Processor	
Number of processors	2
Vendor and model	Intel Xeon® Platinum 8280M
Core count (per processor)	28
Core frequency (GHz)	2.70
Stepping	5
Memory module(s)	
Total memory in system (GB)	384
Number of memory modules	12
Vendor and model	Micron MTA36ASF4G72PZ-2G9E2VG
Size (GB)	32
Type	PC4-2933
Speed (MHz)	2,933
Speed running in the server (MHz)	2,666
Memory module(s) #2	
Total memory in system (GB)	3,072
Number of memory modules	12
Vendor and model	Intel Optane NMA1XBD256GQS
Size (GB)	256
Type	DCPMM
Speed (MHz)	2,666
Speed running in the server (MHz)	2,666
Storage controller	
Vendor and model	Dell PERC H330 Mini
Cache size (GB)	8
Firmware version	25.5.5.0005

Server configuration information	Dell EMC™ PowerEdge™ R740xd
Local storage #1	
Number of drives	8
Drive vendor and model	Intel P4610
Drive size (GB)	1,600
Drive information (speed, interface, type)	PCIe NVMe SSD
Local storage #2	
Number of drives	12
Drive vendor and model	Intel D3-S4510
Drive size (GB)	1,920
Drive information (speed, interface, type)	6Gb/s SATA SSD
Network adapter #1	
Vendor and model	Intel XXV710
Number and type of ports	2x 25Gb Ethernet
Cooling fans	
Vendor and model	Nidec UltraFlo XN44V-A00
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

Our test bed consisted of two servers, an infrastructure server that held VMware vCenter® and client VMs, and the Dell EMC PowerEdge R740xd under test. We installed 8 NVMe SSDs, 12 SATA SSDs, RAM, and Optane DCPMMs on the Dell EMC PowerEdge R740xd. We tested our database workload against 12x Intel Optane DC persistent memory, 2x NAND flash NVMe SSD, 4x NAND flash NVMe SSD, and 12x SATA SSD configurations. Because we did a single-node test, we were unable to configure hardware-level redundancy for the NVMe drives. As such, we created eight separate datastores, one with each drive, on the system under test (SUT). In most production environments, these drives would either be in a software RAID via something like Windows Storage Spaces or the database would be backed up to a separate system to provide redundancy at the software level. For the sake of this paper, we assumed the latter and left the NVMe drives un-RAIDed. We split the SATA drives into two RAIDs: a four-disk RAID 1 for the database log files, and an eight-disk RAID 10 for the database files. We configured the Intel Optane DCPMMs at the BIOS level in App Direct Mode with interleaving enabled.

Creating the workload client

1. Log into VMware vSphere vCenter.
2. Right-click the client host, and select New Virtual Machine.
3. Select Create a New Virtual Machine, and click Next.
4. Name your VM, select a location, and click Next.
5. Select The Client Host, and click Next.
6. Select Standard, pick the datastore for your VM files, and click Next.
7. Select Compatible with ESXi 6.7 Update 2 and later, and click Next.
8. Select Windows and Microsoft Windows Server 2016 or later (64-bit), and click Next.
9. Set the VM with the following:
    a. 24 vCPUs (2 sockets, 12 cores each).
    b. 32GB RAM.
    c. 100GB New Hard Drive.
    d. Change the New SCSI Controller to VMware Paravirtual.
    e. Select the VM network.
    f. Leave rest at defaults.
10. When finished, click Next.
11. Confirm your settings, and click Finish.

Installing Windows on the client VM

1. Connect the VM's CD/DVD drive, point it to the installation media/ISO file, and start the VM.
2. When prompted to boot from DVD, press any key.
3. When the installation screen appears, leave language, time/currency format, and input method as default, and click Next.
4. When prompted, click Install Now.
5. When prompted, enter the product key.
6. Select Windows Server 2019 Datacenter Edition (Server with a GUI), and click Next.
7. Check I Accept the License Terms, and click Next.
8. When prompted, click Custom: Install Windows only (advanced).
9. Select Drive 0 Unallocated Space, and click Next. Windows will begin and restart automatically.
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 in the previous step.
12. Using Windows update, download and install all updates.
13. Once all updates are installed, disable the Windows Update service to ensure no new updates are applied during testing.

Installing the TPC-C derived workload on the client VM

1. Unzip the workload file to the directory of your choice.
2. Navigate to the windows installer, double-click it, and click Next.
3. Select the installation location, and click Next.
4. To confirm the installation, click Next.
5. When installation finishes, click Close.

Installing Microsoft® SQL Server® Management Studio and the ODBC driver on the client VM

1. Download the latest version of MSSQL Management Studio from https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-2017
2. When done, click Install.
3. When the install finishes, reboot the VM.
4. Download ODBC Driver 13 for SQL from https://www.microsoft.com/en-us/download/details.aspx?id=53339

Creating the SQL VM

1. Log into VMware vSphere vCenter.
2. Right-click the host under test, and select New Virtual Machine.
3. Select Create a New Virtual Machine, and click Next.
4. Name your VM, choose a location, and click Next.
5. Select the R740xd, and click Next.
6. Select Standard, pick the datastore for your VM files, and click Next.
7. Select Compatible with ESXi 6.7 Update 2 and later, and click Next.
8. Select Linux and Red Hat® Enterprise Linux® 8 (64-bit), and click Next.
9. Set the VM with the following:
    a. 48 vCPUs – 2 sockets, 24 cores.
    b. 350GB RAM.
    c. 100GB New Hard Drive.
    d. Leave the New SCSI Controller to VMware Paravirtual.
    e. Select the VM network.
    f. Leave rest at defaults.
10. For the Intel Optane DC persistent memory configuration:
    a. While still on the VM settings screen, click Add New Device.
    b. Select NVDIMM, and set to 1.4TB.
    c. Click Add New Device again, add a second NVDIMM, and set it to 300GB.
11. For the NAND flash NVMe SSD configurations:
    a. While still on the VM settings screen, click Add New Device.
    b. Select New Disk, and set to 1.4TB.
    c. Create two more new disks, and set both to 1.4TB.
    d. Add a final disk, and set to 300GB.
    e. Put each of these new disks on its own NVMe datastore.
12. Select Add New Device one more time, add a new hard Disk, and set it to 300GB. This will store the database backup file and can be stored on the same datastore as the VMd
13. When prompted, click Next.
14. Confirm your settings, and click Finish.

Installing Red Hat Enterprise Linux 8.0 on the SQL VM

1. Open the VM you created in the last section with VMware Remote Console.
2. Click on VMRC→Removable Devices→CD/DVD drive 1→Connect to disk file image (iso) and load the RHEL 8 ISO.
3. Power on the VM, and select Install Red Hat Enterprise Linux 8.0.0 when the ISO boots.
4. Select your language, and click Continue.
5. Select Time & Date, and click the gear button in the top-right corner.
6. Uncheck the default ntp file and add your own, and click OK.
7. When prompted, click Done.
8. Select Software Selection, and choose Minimal Install.
9. When the prompted, click Done.
10. Select Installation Destination, and click the 32GB VMware Virtual disk.
11. Check I will configure partitioning, and click Done.
12. Select Click Here to create them automatically.
13. Double check the partitions are accurate, and the swap file is sized appropriately for your testing. We used the defaults for this test. If necessary, remove the /home partition and resize the root partition to use the entire space available.

14. When prompted, click Done.
15. At the pop-up summary of changes window, click Accept Changes.
16. Select KDUMP and uncheck Enable KDUMP.
17. When prompted, click Done.
18. Select Network and Host Name.
19. Click the switch to turn the Ethernet port to On. If you do not have DHCP, click Configure to set an IP.
20. Set a host name, and click Apply.
21. When prompted, click Done.
22. When prompted, click Begin Installation.
23. When the package download begins, click Root Password, and set the password for the Root user.
24. When prompted, click Reboot.

Configuring Red Hat Enterprise Linux 8.0 for SQL

1. Log in as the root user.
2. Stop and disable the firewall:

```
systemctl stop firewalld
systemctl disable firewalld
```

3. Edit the SELINUX config file to disable SELINUX:

```
vi /etc/selinux/config
```

   a. Type `i`
   b. Change SELINUX=enforcing to SELINUX=disable, and click ESC
   c. Type `:wq`, and click Enter

4. Attach to the subscription repository and run updates

```
subscription-manager register --auto-attach
yum update -y
```

5. Set the tuning profile to MSSQL:

```
subscription-manager repos --enable= rhel-8-for-x86_64-supplementary-rpms
yum install tuned-profiles-mssql
tuned-adm profile mssql
```

6. Edit the 99-sysctl.conf file to disable RHEL-level numa balancing and increase the map count:

```
vi /etc/sysctl.d/99-sysctl.conf
```

   a. Add the following two lines to the end of the file by typing `i` to edit:

```
kernel.numa_balancing=0
vm.max_map_count=262144
```

   a. Hit ESC, write `:wq` and hit Enter.

Preparing the Optane data and log volumes

1. Create the drive directories:

```
mkdir -p /opt/db/{data,logs}
```

2. Check that the NVDIMMs are present:

```
lsblk
```

3. Install the NDCTL packages:

```
yum install ndctl
```

4. Verify the namespace of the Optanes:

```
ndctl list
```

5. Configure the namespaces for fsdax mode:

```
ndctl create-namespace -f -e namespace0.0 --mode=fsdax --map=dev
ndctl create-namespace -f -e namespace1.0 --mode=fsdax --map=dev
```

6. Format the Optane devices

```
mkfs.xfs -m reflink=0 /dev/pmem0
mkfs.xfs -m reflink=0 /dev/pmem1
```

7. Mount the Optane devices

```
mount -o dax,noatime /dev/pmem0 /opt/db/data
mount -o dax,noatime /dev/pmem1 /opt/db/logs
```

8. Change the default IO allocation size to 2MB by running the following command

```
xfs_io -c “extsize 2m” /opt/db/data
xfs_io -c “extsize 2m” /opt/db/logs
```

9. Edit the file system table to add the new drives and mount points

```
vi /etc/fstab
```

   a. Type `i` to enable editing.
   b. Add the following lines to the end of the file:

```
/dev/pmem0 /opt/db/data xfs dax,noatime 0 0
/dev/pmem1 /opt/db/logs xfs dax,noatime 0 0
```

Preparing the NVMe and SATA data, log, and backup volumes

1. Create the drive directories:

```
mkdir -p /opt/db2/{data,logs}
mkdir -p /opt/db3/data
mkdir -p /opt/db4/data
mkdir -p /opt/db/backup
```

2. Create the drive partition for each drive (replace ‘b’ in ‘sdb’ for each appropriate drive letter)

```
fdisk /dev/sdb
```

   a. Type `n` and press Enter.
   b. Press Enter for default.
   c. Press Enter for default.
   d. Press Enter for default.
   e. Press Enter for default.
   f. Type `w` and press Enter.
   g. Type `lsblk` to verify the partitions were created.

3. Format each drive (replace ‘b’ in ‘sdb’ for each appropriate drive letter):

```
mkfs.xfs /dev/sdb1
```

4. Edit the file system table to add the new drives and mount points:

```
vi /etc/fstab
```

   a. Type `i` to enable editing.
   b. Add the following lines to the end of the file:
      i. `/dev/sdb1 /opt/db/backup xfs defaults 0 0`
      ii. `/dev/sdc1 /opt/db2/data xfs defaults 0 0`
   c. Hit ESC, type `:wq` and press Enter.

5. Mount the disks:

```
mount -a
```

Installing Microsoft SQL Server on Linux 2019 preview

1. Install prereqs:

   ```
   yum install python2
   yum install compat-openssl10
   ```

2. Create a python symlink:

   ```
   alternatives --config python
   ```

   a. Enter the number for Python 2, and press Enter.

3. Copy the SQL server RPM file over to the /root file, and install:

   ```
   rpm -Uvh --nodeps mssql-server*rpm
   ```

4. Create the SA password, and choose SQL edition:

   ```
   /opt/mssql/bin/mssql-conf setup
   ```

   a. Type `1` to install the Evaluation edition. Press Enter.
   b. Type `Yes` to accept the license terms. Press Enter.
   c. Type the password you want to assign to the system administrator account, and press Enter.
   d. Confirm the password, and press Enter.

5. Verify that the service is running:

   ```
   systemctl status mssql-server
   ```

Installing SQL Server CLI tools

1. Download the SQL Server CLI tools repository file:

   ```
   curl -o /etc/yum.repos.d/msprod.repo https://packages.microsoft.com/config/rhel/7/prod.repo
   ```

2. Install the tools:

   ```
   yum install -y mssql-tools unixODBC-devel
   ```

   a. Type `YES` to accept the license terms, and press Enter.
   b. Type `YES` again to accept the license terms, and press Enter.

3. Add the path to the tools to your PATH environment for convenience:

   ```
   echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bash_profile
   echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bashrc
   source ~/.bashrc
   ```

4. Check your connection by connecting locally:

   ```
   sqlcmd -S localhost -U SA -P '<YourPassword>'
   ```

Configuring additional SQL settings

1. Give SQL ownership of the drive directories:

   ```
   chown mssql:mssql /opt/db/{data,logs,backup}
   chown mssql:mssql /opt/db2/{data,logs}
   chown mssql:mssql /opt/db3/data
   chown mssql:mssql /opt/db4/data
   ```

2. Change the default data and logs directories:

   ```
   /opt/mssql/bin/mssql-conf set filelocation.defaultdatadir /opt/db/data/
   /opt/mssql/bin/mssql-conf set filelocation.defaultlogdir /opt/db/logs/
   /opt/mssql/bin/mssql-conf set filelocation.defaultbackupdir /opt/db/backup/
   ```

3. Set the SQL memory limit:

```
/opt/mssql/bin/mssql-conf set memory.memorylimitmb 322560
```

4. Disable customer feedback:

```
/opt/mssql/bin/mssql-conf set telemetry.customerfeedback false
```

5. Enable the 3979 traceflag:

```
/opt/mssql/bin/mssql-conf traceflag 3979 on
```

6. Restart SQL Server service:

```
systemctl restart mssql-server.service
```

7. Stop the SQL Server service:

```
systemctl stop mssql-server
```

8. Open the service configuration editor which will create an override for the original service config. This file will be empty.

```
systemctl edit mssql-server
```

9. Add the following lines and save the file:

```
[Service]
Environment="LD_LIBRARY_PATH=/opt/mssql/lib"
```

10. Reload the systemctl daemons:

```
systemctl daemon-reload
```

11. Create symbolic links to OpenSSL 1.0 for SQL Server to use:

```
ln -s /usr/lib64/libssl.so.1.0.2o /opt/mssql/lib/libssl.so
ln -s /usr/lib64/libcrypto.so.1.0.2o /opt/mssql/lib/libcrypto.so
```

12. Start SQL Server:

```
systemctl start mssql-server
```

13. To finish the configuration, install Microsoft SQL Server Management Studio on the Client VM and connect it to the SQL instance using the SA account and password.
14. Expand databases→System databases, and right-click tempdb.
15. Verify that there are 8 tempdb database files.
16. Change the size of each of the 8 database files to 8,192MB. Click OK.
17. Right-click the instance, and select Create New Database.
18. Click Add to create 11 additional database files for 12 total database files.
19. Pre-grow the 12 database files to 8,192 MB and set the autogrowth size to 10GB.
20. Pre-grow the log file to 50GB.
21. Double-check that the files are pointing to the default database and log directories.
22. Set the recovery mode to bulk.
23. Restart the MSSQL service or reboot the database VM.
24. Disconnect and reconnect to the instance in the SQL Management Studio.

Creating the TPC-C database

1. Open the directory where you unzipped the workload on the client system.
2. Navigate to SETUP→scripts→14000.war→database.
3. Open the createdb.sql file in SQL Manager.
4. Edit all filenames to point to the directories we created in previous steps. We built the database directly to the Optane devices and changed all database filenames to '/opt/db/data/<filename>' and the log file to '/opt/db2/logs/MSSQL_log.ldf'
5. Save the file, and click Execute.
6. When the database is finished creating, close the .sql file.

Creating the database tables

1. Open Windows PowerShell and point to the SETUP directory in the workload folder.
2. Run the following command to create the tables for the TPC-C database:

```
./setup.cmd /S:<servername> /U:sa /P:<SQLpassword> /W:14000 /B:objects /D:normal /V:true
```

   a. <servername> should be the IP of your SQL Server VM.
   b. <SQLpassword> should be the password of your SA SQL user.
   c. /W is the number of warehouses. We set 14,000.
   d. /B is the build type. Here we just want to create the tables and nothing else, so we chose Objects.
   e. /D is the type of database. We selected normal.
   f. /V is verbose set to true or false. We selected true.
3. After the build finishes, load the data.

Loading the data and backing up the database

1. Open Windows PowerShell, and point to the SETUP directory in the workload folder.
2. Run the following command to load the data and set the final SQL settings:

```
./setup.cmd /S:<servername> /U:sa /P:<SQLpassword> /W:14000 /B:bulkloadfull /D:normal /V:true
```

   a. The bulkloadfull command tells the workload to load the data and finalize the SQL settings baked into the workload scripts.
3. When the data load is finished, open SQL Server Management Studio, and right-click the new tpcc database.
4. Click Tasks→Backup…
5. Select Add, and type the path to the drive we created previously (/opt/db/backup) and the file name.
6. Select Backup Options, choose Compress Backup from the compression drop-down menu.
7. When prompted, click OK.
8. When the backup is finished, you can start testing.

Configuring the test

For the NVMe drives and Intel Optane DC persistent memory, we ran 100 workload threads to get full storage saturation while fully utilizing the CPU. For the SATA drives, we had to pull back on the thread count to avoid oversaturating the drives. We found that 65 threads provided the best storage performance while fully utilizing the CPU.

1. Double-click the workload shortcut on the desktop.
2. Select New Profile.
3. Select browse, choose TPCC-SQLServer.bdll, and click Next.
4. Click Driver, choose the default template, and click Ok.
5. In the Driver editor screen, do the following:
   a. Name the driver.
   b. Add a path for the Logs (results files).
   c. Set the Connect rate and Start rate to 50.
6. Select Thread Group, choose the TGT-1 template, and click Ok.
7. Select Parameter, set and choose the PST-1 Official Mix template.
8. In the Thread Group editor screen, do the following:
   a. Name the thread group.
   b. Set the Max and Initial thread counts to 100.
   c. Select the driver you just created.
   d. Select the parameter set you just created.
   e. Set W_ID Range Start to 1.
   f. Set W_ID Range End to 14000.
   g. Set W_ID Minimum to 1.
   h. Set W_ID Maximum to 14000.
   i. Set ODBC Driver to SQL Server.
   j. Enter the SQL Server IP.
   k. Enter SA.
   l. Enter the SA user password.
9. When done, click Save.
10. Name the file, and save it to the directory of your choice

Running the test

1. Double-click the workload shortcut on the desktop.
2. Select Load Profile, and load the profile you created and saved in the previous step.
3. When done, click Launch.
4. Select Run when you're ready to start the test. We ran our test for 45 minutes, and reported the last 15 minutes of the run when the system was at a steady state.

Read the report at http://facts.pt/gkuc7nc ▶

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.