



The science behind the report:

Get important data insights faster by adding Intel Optane persistent memory to Dell EMC PowerEdge R740 servers

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 [Get important data insights faster by adding Intel Optane persistent memory to Dell EMC PowerEdge R740 servers.](#)

We concluded our hands-on testing on May 28, 2020. During testing, we determined the appropriate hardware and software configurations and applied updates as they became available. The results in this report reflect configurations that we finalized on May 18, 2020 or earlier. Unavoidably, these configurations may not represent the latest versions available when this report appears.

Our results

Table 1: TPC-H-like workload results for the Dell EMC PowerEdge R740 with and without Intel Optane PMem. Source: Principled Technologies

	Without PMem	With PMem	PMem config % less time
	Median run	Median run	
Single query stream (minutes)	59.90	56.43	6.14%
Longest of seven query streams (minutes)	116.35	100.67	15.58%
Average of seven query streams (minutes)	110.45	93.84	17.69%

System configuration information

Table 2: Server configuration information. Source: Principled Technologies.

Server configuration information	Dell EMC™ PowerEdge™ R740
BIOS name and version	Dell 2.5.4
Non-default BIOS settings	Performance Mode & LLC Prefetcher enabled
Operating system name and version/build number	Windows Server 2019 Standard Version 1809
Date of last OS updates/patches applied	5/18/2020
Power management policy	Performance
Processor	
Number of processors	2
Vendor and model	Intel® Xeon® Gold 5215L
Core count (per processor)	10
Core frequency (GHz)	2.5
Stepping	6
Memory module(s) type 1	
Total memory in system (GB)	192
Number of memory modules	12
Vendor and model	Hynix HMA82GR7AFR8N-VK
Size (GB)	16
Type	DDR-4 PC4-21300 Dual-Rank
Speed (MHz)	2,666
Speed running in the server (MHz)	2,666
Memory module(s) type 2 (PMem)	
Total memory in system (GB)	512
Number of memory modules	4
Vendor and model	Intel Optane™ Persistent Memory NMA1XXD128GPS
Size (GB)	128
Type	Persistent memory DDR-4 Single-Rank
Speed (MHz)	2,666
Speed running in the server (MHz)	2,666
Storage controller	
Vendor and model	Dell PERC H730p
Cache size (MB)	2,048
Firmware version	25.5.6.0009
Driver version	6.604.6.0

Server configuration information		Dell EMC™ PowerEdge™ R740
Local storage type 1		
Number of drives		4
Drive vendor and model		Intel S4510
Drive size (TB)		1.92
Drive information (speed, interface, type)		6Gbps SATA SSD
Local storage type 2		
Number of drives		2
Drive vendor and model		Intel S4610
Drive size (GB)		480
Drive information (speed, interface, type)		6Gbps SATA SSD
Local storage type 3		
Number of drives		2
Drive vendor and model		Intel SSDSCKJB120G7R
Drive size (GB)		120
Drive information (speed, interface, type)		6Gbps SATA M.2 SSD
Network adapter type 1		
Vendor and model		Intel Ethernet Converged Network Adapter X710-T4
Number and type of ports		4x 10GbE RJ45
Driver version		1.11.101.0
Cooling fans		
Vendor and model		NMB 06038DA-12S-EWH
Number of cooling fans		6
Power supplies		
Vendor and model		Dell D1100E-S0
Number of power supplies		2
Wattage of each (W)		1,100

How we tested

Testing overview

We installed Microsoft Windows Server 2019 Standard and Microsoft SQL Server 2019 Standard on a Dell EMC PowerEdge R740 server. We created a 1,000-scale TPC-H-like database and ran single-user and seven-user tests against the database. Each simulated user in the test workload ran a serialized string of 22 read-only queries against the database. With the multi-user test, each simulated user ran through the queries in a different order so the same queries weren't running at the same time. In the first configuration (without PMem), we stored the database on traditional SSD storage that we configured in a RAID 10 for the database files and RAID 1 for the log files.

We then added four Intel Optane persistent memory modules to create the second configuration (with PMem). We moved the database under test to the PMem volumes, enabled the SQL Server Hybrid Buffer Pool feature, and ran the tests again to show the performance advantages from adding Intel Optane persistent memory.

We collected the time it took for both configurations to complete the single user tests and multi-user tests. For reporting purposes for the multi-user test, we chose to report the times for the user that took the longest to finish and the average query stream completion times.

Installing and configuring Windows on the R740

Before we installed Windows, we made sure the BIOS was the most recent version and updated all firmware and drivers. We set the BIOS to performance and enabled the LLC Prefetcher cache.

Installing Windows Server 2019 Standard

1. Navigate to the iDRAC console, and open the remote console.
2. Attach the Windows Server 2019 Standard ISO, and start (or restart) the server.
3. When prompted to boot from DVD, press any key.
4. When the installation screen appears, leave language, time/currency format, and input method as default, and click Next.
5. Click Install now.
6. When prompted, enter the product key.
7. Select Windows Server 2019 Standard Edition (Server with a GUI), and click Next.
8. Check I accept the license terms, and click Next.
9. Click Custom: Install Windows only (advanced).
10. Select the Unallocated Space on the OS drive (we used the M.2 BOSS cards), and click Next. This will start the installation automatically, and Windows will restart automatically after completing the install.
11. When the Settings page appears, fill in the Password and Reenter Password fields with the password of your choice.
12. Log in with the password you set.

Configuring Windows Server 2019 Standard

1. In Server Manager, set the time zone, enable RDP if desired, and disable the firewall.
2. Run Windows updates until you have no new updates to install.
3. Go to Disk Manager, and make sure the additional drives are online, formatted, and presented to the OS.
4. Go to Settings→System→Power & Sleep→Additional Power Settings, and set Preferred plans to High performance.
5. Go to Control Panel→System and Security→System→Advanced System Settings.
6. In the Performance box, click Settings→Adjust for best performance.
7. Go to the Advance tab, and choose Adjust for best performance of Programs.
8. In the Virtual memory box in the Advanced tab, click Change....
9. For all volumes in the system, change the paging file size to Custom size, and set the Initial and Maximum sizes to 4096 MB, and click OK.
10. To close the Performance Options window, click OK, and on the popup warning about rebooting the system, click OK.
11. To close the System Properties window, click OK.
12. Reboot the server.

Installing and configuring SQL Server 2019 Standard

Installing SQL Server 2019 Standard

1. Download or copy the ISO to the server, and unzip it.
2. Double-click the Setup application.
3. Click Installation→New SQL Server Standalone installation or add features to an existing installation.
4. Enter the product key, and click Next.
5. Check the I accept the license terms and Privacy Statement box, and click Next.
6. Check the Use Microsoft Update to check for updates (recommended) box, and click Next.
7. On the Install Rules page, click Next.
8. Check the boxes for the following features, and click Next:
 - a. Database Engine Services
 - b. Full-Text and Semantic Extractions for Search
 - c. Client Tools Connectivity
 - d. Client Tools Backwards Compatibility
9. Leave the Default instance, and click Next.
10. Leave the default Service Accounts, and click Next.
11. On the Server Configuration tab, choose Mixed Mode, and enter and confirm a Password for the SQL Server system administrator (sa) account.
12. Click Add Current User to Specify the SQL Server administrators.
13. On the Data Directories tab, change the Data root directory to the database volume.
14. Choose the log volume for the User database log directory.
15. If you plan to have a locally-stored backup, then change the Backup directory to point to that drive.
16. On the TempDB tab, set the tempdb database and log files to 1024MB with 1024MB autogrowth.
17. Click Next.
18. Once you've passed the rule check, click Next.
19. Click Install.
20. Once the install finishes, navigate to the SQL Server Installation Center, and click Install SQL Server Management Tools.
21. Download the SSMS file, and install with defaults.
22. Reboot the server when prompted.
23. To ensure there aren't any new updates for SQL (make sure Windows Updates are set to get updates for other Microsoft products), run Windows Update one more time.
24. To disable Windows Update service, click the Start button, type `services` to open the Services list, and disable the Windows Update service.

Configuring SQL Server 2019 Standard

1. Open SQL Server Management Studio.
2. Select SQL Server Authentication, type `sa` and your password as your login and password respectively, and click connect.
3. To show advanced options, type the following command:

```
exec sp_configure 'show advanced options', '1'  
go  
reconfigure with override
```

4. To set the maximum memory for SQL to 90% of the available RAM, or 172GB, type the following SQL query:

```
exec sp_configure 'max server memory', '176128'  
go  
reconfigure with override
```

5. To set the maximum degree of parallelism (MAXDOP) to 8, type the following SQL query:

```
exec sp_configure 'max degree of parallelism', '8'  
go  
reconfigure with override
```

6. To set the maximum worker threads to 3,000, type the following SQL query:

```
exec sp_configure 'max worker threads', '3000'  
go  
reconfigure with override
```

Locking pages in memory

1. Click Start, and type `Local Security Policy`. When the program appears in the search, open the program.
2. Expand Local Policies, and click User Rights Assignment.
3. In the right-hand pane, scroll down and double-click Lock pages in memory.
4. Click Add User or Group, type `NT Service\MSSQLSERVER`, and click OK.
5. To close the Properties window, click OK, and close the Local Security Policy window.

Setting up the workload test

To create the 1,000-scale database, we made a new database in SQL Server Management Studio. In addition to the default Primary SQL Server filegroup, we created a secondary data filegroup in the new database. We created 24 files in this filegroup and sized them all at approximately 20.9 GB.

When testing on the traditional storage drives, we stored the data files on the SATA RAID and the log files on the NVMe RAID. We resized the single transaction log file to 20.9GB as well. When we tested with PMem, we split the data files between the two PMem volumes and kept the log files on the NVMe RAID. For all testing, we kept TempDB database files on the SATA RAID and their log files on the NVMe RAID.

We created the specific tables required for a TPC-H-like database according to TPC and HammerDB documentation: Part, Supplier, PartSupp, Customer, Nation, Lineitem, Region, and Orders. We then created a partition function using a date data type, with weekly values as follows:

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

We then applied this partition function, using a partition scheme, to the secondary filegroup containing the 24 data files.

The HammerDB tool and dbgen defined the remainder of the schema definition. We created the test data using the dbgen tool and loaded the data into the prepared database. We used clustered columnstore indexing where possible and applied the partition scheme specifically to the ShipDate and OrderDate columns, in the Lineitem and Orders table, respectively. For our specific schema DDL scripts, contact us at info@principledtechnologies.com.

Creating the empty database

1. Open SQL Server Management Studio.
2. Right-click Databases, and choose New Database...
3. Name the database.
4. Click the Filegroups tab, and click Add Filegroup.
5. Name the new filegroup, and set it to Default.
6. To create 24 additional data files, in the general tab, click Add.
7. Name each file, and set the initial size for each new file to 20.9GB. Make sure all new files are on the new filegroup you created.
8. If necessary, change all data and log file paths to point to the data and log drives on the system.
9. On the options tab, set the Recovery Model to simple.
10. To start the build, click OK. This may take a while.

Generating and loading data into the database

In this process, we used dbgen to create the flat files for the 1,000-scale database that we then loaded into the empty database. To ensure there was enough room to simultaneously hold the flat files and load the database, we conducted these steps on a separate infrastructure server with SQL Server. We then created a backup file and used it to restore the finished database to the server under test.

1. Download the dbgen tool from GitHub here: <https://github.com/electrum/tpch-dbgen>.
2. Unzip the file to a volume with roughly 1.5TB capacity.
3. Download and install the Community edition of VisualStudio.
4. Navigate to the dbgen folder, and build the tpch.vcproj in Visual Studio.
5. Navigate to the debug folder in the dbgen folder, and in the folder, hold shift and right-click. Click the Open PowerShell Window.
6. To test to ensure dbgen.exe works, in PowerShell, type `./dbgen.exe -h`. If the help menu appears, then dbgen.exe works. If you get an error, copy the dbgen.exe file to the dbgen folder, open PowerShell, and run the command again.
7. Once dbgen.exe works properly, type the following command to create the 1,000-scale files:

```
./dbgen.exe -s 1000
```

8. Once the files finish building (and this will take some time), load them into the database.
9. To load the files into the tables, open Microsoft SQL Management Studio, and run bulk insert commands.
10. Once the data finishes loading, create the database indexes, and finalize the database as described above.

Backing up the database

We didn't perform any write tasks in the workload tests, but we ensured we had a backup of a clean database that we could restore from should we need it. We stored this backup file on a separate infrastructure server. To create a database backup file, follow these steps:

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

Configuring for persistent memory tests

We set the PMem to App Direct Mode and created two interleaved regions. We then used Windows PowerShell commands to format the two regions into two file system direct access volumes with 2MB allocation size. This was in accordance to SQL Server best practices for Hybrid Buffer Pool. Using the database restore file, we alternated the database files between the PMem volumes so that half lived on one and half on the other. We then enabled Hybrid Buffer Pool, a SQL tool that utilizes byte-addressable direct-access volumes to improve performance. We then repeated the tests on the persistent memory configuration that we ran on the traditional storage. The logs and tempdb remained on the original RAIDs, where they were located for the non-PMem testing.

Creating the persistent memory volumes

1. Install the PMem according to the Dell instructions found here: https://www.dell.com/support/manuals/us/en/04/poweredge-r640/dcpmm_ug_pub/dcpmm-recommended-topologies?guid=guid-85f17354-b7af-4c04-ac6a-f5468b4b4ab7&lang=en-us.
2. Boot the server, and log into Windows.
3. Install the IPMCTL tool found here: <https://downloadcenter.intel.com/download/29380/DCPM-Software-for-Intel-Optane-DC-Persistent-Memory-for-Windows-Server-2019?v=t>.
4. To create interleaved, app-direct PMem regions using all the available space, type the following commands:

```
ipmctl create -f -socket 0x0 -goal MemoryMode=0 PersistentMemoryType=AppDirect
ipmctl create -f -socket 0x1 -goal MemoryMode=0 PersistentMemoryType=AppDirect
```
5. Reboot the server.
6. Log into Windows, and open PowerShell.
7. To create two disks from the regions you created previously, type the following command (this creates disks for both regions, so you need to run it only once):

```
Get-PmemUnusedRegion | New-PmemDisk
```
8. Type the following command, and note the DiskNumber of each PMem disk:

```
Get-PmemDisk
```
9. To initialize the disks, type the following command for each disk using the DiskNumber you recorded in the previous step:

```
Initialize-Disk -Number <insert DiskNumber> -PartitionStyle GPT
```
10. To create a new partition on each disk and assign a drive letter, type the following command for each disk using the DiskNumbers you recorded in step 8:

```
New-Partition -DiskNumber <insert DiskNumber> -UseMaximumSize -DriveLetter <insert drive letter>
```
11. To format the disk with NTFS, enable direct access mode, set the allocation unit size, and name the volume, type the following command on both disks using the Drive Letters you set in the previous step:

```
Format-Volume -DriveLetter <insert letter> -FileSystem NTFS -IsDAX $true -AllocationUnitSize 2097152
-NewFileSystemLabel PMEMDAX-<insert number> -Confirm:$false
```

Configuring Hybrid Buffer Pool

We enabled Hybrid Buffer Pool for the tests where the PMem stored the database. To enable Hybrid Buffer Pool, use the following steps:

1. Open SQL Server Management Studio.
2. To open a blank page, click New Query.
3. Type the following, and click Execute:

```
ALTER SERVER CONFIGURATION SET MEMORY_OPTIMIZED HYBRID_BUFFER_POOL = ON;
```
4. If you need to confirm the Hybrid Buffer Pool status, click New Query, and execute the following commands:

```
SELECT * FROM sys.configurations WHERE name = 'hybrid_buffer_pool';
SELECT name, is_memory_optimized_enabled FROM sys.databases;
```
5. To disable Hybrid Buffer Pool on the TPC-H-like database, click New Query, and execute the following command:

```
ALTER DATABASE <databaseName> SET MEMORY_OPTIMIZED = OFF;
```


Running the TPC-H-like workload

We ran the OLAP TPC-H-like workload using the HammerDB benchmark. As instructed in the documentation, we ran the HammerDB test directly on the server under test because the overhead on the hardware was negligible. Thus, we did not need clients for this test.

Installing HammerDB on the PowerEdge R740

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

Running the test

1. Open HammerDB.
2. Select Options→Benchmark.
3. Choose MSSQL Server and TPC-H.
4. Expand SQL Server→TPC-H→Schema Build.
5. Double-click Options, change the driver to ODBC Driver 17 for SQL Server, set the scale to 1000, set MAXDOP to 8, and check the box for Clustered Columnstore. Click OK.
6. Expand Driver Script, double-click Options, and click OK to load.
7. Expand Virtual User, and double-click Options.
8. Choose 1 or 7 users depending on whether you are running a single- or multi-user test.
9. Check the boxes for Show Output, Log Output to Temp, and Use Unique Log Name.
10. Click OK.
11. To load the driver script, double-click Load.
12. Double-click Create users.
13. To capture performance metrics on the system, start Performance monitor set to record CPU, Memory, and drive usage information.
14. To begin the test run, click Start.
15. When the test run finishes, stop Perfmon, and save the HammerDB results file and Perfmon output.
16. Complete the test two more times at each user count, and record the median run.

Read the report at <http://facts.pt/r2ui89n> ▶

This project was commissioned by Dell EMC.



Facts matter.®

Principled Technologies is a registered trademark of Principled Technologies, Inc.
All other product names are the trademarks of their respective owners.

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.