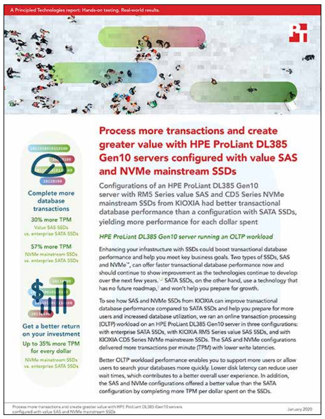
The science behind the report:

Process more transactions and create greater value with HPE ProLiant DL385 Gen10 servers configured with value SAS and NVMe mainstream SSDs



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

Process more transactions and create greater value with HPE ProLiant DL385 Gen10 servers configured with value SAS and NVMe mainstream SSDs

Configurations of an HPE ProLiant DL385 Gen10 server with RM5 Series value SAS and CD5 Series NVMe mainstream SSDs from KIOXIA had better transactional database performance than a configuration with SATA SSDs, yielding more performance for each dollar spent

Complete more database transactions

30% more TPM
Value SAS SSDs vs. enterprise SATA SSDs

57% more TPM
NVMe mainstream SSDs vs. enterprise SATA SSDs

Get a better return on your investment

Up to 35% more TPM for every dollar
NVMe mainstream SSDs vs. enterprise SATA SSDs

HPE ProLiant DL385 Gen10 server running an OLTP workload

Enhancing your infrastructure with SSDs could boost transactional database performance and help you meet key business goals. Two types of SSDs, SAS and NVMe™, can offer faster transactional database performance now and should continue to show improvement as the technologies continue to develop over the next few years.[1,2] SATA SSDs, on the other hand, use a technology that has no future roadmap,[3] and won't help you prepare for growth.

To see how SAS and NVMe SSDs from KIOXIA can improve transactional database performance compared to SATA SSDs and help you prepare for more users and increased database utilization, we ran an online transaction processing (OLTP) workload on an HPE ProLiant DL385 Gen10 server in three configurations: with enterprise SATA SSDs, with KIOXIA RM5 Series value SAS SSDs, and with KIOXIA CD5 Series NVMe mainstream SSDs. The SAS and NVMe configurations delivered more transactions per minute (TPM) with lower write latencies.

Better OLTP workload performance enables you to support more users or allow users to search your databases more quickly. Lower disk latency can reduce user wait times, which contributes to a better overall user experience. In addition, the SAS and NVMe configurations offered a better value than the SATA configuration by completing more TPM per dollar spent on the SSDs.

Process more transactions and create greater value with HPE ProLiant DL385 Gen10 servers configured with value SAS and NVMe mainstream SSDs January 2020

This document describes what we tested, how we tested, and what we concluded. To learn how these facts translate into real-world benefits, read the report Process more transactions and create greater value with HPE ProLiant DL385 Gen10 servers configured with value SAS and NVMe mainstream SSDs.

We completed our hands-on testing on November 16, 2019. During testing, we determined the appropriate hardware and software configurations to represent an OLTP database application environment and applied updates as they became available. The results in this report reflect configurations that we finalized on November 1, 2019 or earlier. Unavoidably, these configurations may not represent the latest versions available when this report appears.

Our results

The table below presents our findings for each configuration of the HPE ProLiant DL385 Gen10 server we tested.

	With enterprise SATA SSDs	With KIOXIA RM5 Series value SAS SSDs	With KIOXIA CD5 Series NVMe mainstream SSDs
Avg. transactions per minute (TPM)	1,389,990	1,813,808	2,187,136
Avg. read latency (ms)	1.440	0.813	0.314
Avg. write latency (ms)	15.496	7.633	6.574
Total hardware cost of the solution (USD)	$29,095.36	$32,533.00	$33,733.00
TPM per dollar (USD)	47.7	55.7	64.8

System configuration information

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

Server configuration information	HPE ProLiant DL385 Gen10
BIOS name and version	A40 v1.46
Operating system name and version/build number	Windows Server 2019 (Build 17763.832)
Date of last OS updates/patches applied	11/11/2019
Power management policy	Default - Virtualization/Max Performance
Processor	
Number of processors	2
Vendor and model	AMD EPYC™ 7601
Core count (per processor)	32
Core frequency (GHz)	2.20
Stepping	ZP-B2
Memory module(s)	
Total memory in system (GB)	128
Number of memory modules	8
Vendor and model	Micron MTA18ASF2G72PDZ-2G6D1SI
Size (GB)	16
Type	DDR4-2666V
Speed (MHz)	2,666
Speed running in the server (MHz)	2,400
Storage controller	
Vendor and model	HPE Smart Array P408i-a SR Gen10
Firmware version	1.99
Driver version	106.100.0.1014
Local storage - SATA	
Number of drives	4
Drive vendor and model	Intel® SSDSC2KB960G8
Drive size (GB)	960
Drive information	SATA SSD
Local storage - SAS	
Number of drives	4
Drive vendor and model	HPE VO000960RWUEV (KIOXIA RM5 Series value SAS)
Drive size (GB)	960
Drive information	SAS SSD

Server configuration information	HPE ProLiant DL385 Gen10
Local storage - NVMe	
Number of drives	4
Drive vendor and model	HPE VO000960KWVBR (KIOXIA CD5 Series NVMe mainstream)
Drive size (GB)	960
Drive information	NVMe SSD
Local storage - offload	
Number of drives	4
Drive vendor and model	HPE VO000960KWVBR (KIOXIA CD5 Series NVMe mainstream)
Drive size (GB)	960
Drive information	NVMe SSD
Local storage - OS / boot	
Number of drives	2
Drive vendor and model	KIOXIA THNSNK256GVN8
Drive size (GB)	256
Drive information	M.2 SSD
Network adapter 1	
Vendor and model	HPE Ethernet 1Gb 4-port 331i Adapter
Number and type of ports	4 x 1GbE
Driver version	214.0.0.0
Network adapter 2	
Vendor and model	Intel Ethernet Converged Network Adapter X520-DA2
Number and type of ports	2 x 10GbE
Driver version	3.12.11.1
Cooling fans	
Vendor and model	HPE 867810-B21
Number of cooling fans	6
Power supplies	
Vendor and model	HPE 865414-B21
Number of power supplies	2
Wattage of each (W)	800

How we tested

Configuring the HPE ProLiant DL385 Gen10

We configured an HPE ProLiant DL385 Gen10 server with two AMD EPYC 7601 processors, 128GB RAM, and three different sets of test drives: four Intel S4510 SATA SSDs, four KIOXIA RM5 Series value SAS SSDs, and four KIOXIA CD5 Series NVMe mainstream SSDs. We used an additional four NVMe SSDs for database backups, tempdb, and logging. We configured the BIOS with default settings and installed Windows Server 2019 Datacenter and SQL Server 2019 Enterprise Edition (Evaluation). For networking, we used the onboard 1GbE for management traffic and a dual-port 10Gb card for client traffic.

For our storage configuration, we used the dual onboard 256GB M.2 SSDs as our OS boot drive. For the SAS and SATA test drives, we configured the four drives of each type in a RAID10 using the HPE Smart Storage Administrator with the default settings. For the four NVMe test drives and the remaining four NVMe drives used for database backups, logging, and tempdb, we configured two volumes using Windows Storage Spaces. We provide steps below for how to configure these volumes in Windows.

For the workload, we used HammerDB 3.2 to generate a TPC-C-like load targeting a single 1.3TB (14,000 scale) database. For each test run, we used a different set of test drives to store the database data files. We performed tests on each drive set three times and reported the median test run for each. Please note, these results are not comparable to official TPC benchmark results.

Below are the steps we followed to configure our SQL server under test, the HammerDB client server, and perform the test.

Installing Microsoft Windows Server 2019

We used the following steps to install and configure our system under test.

1. Insert the Windows Server 2019 installation media into the server.
2. Power on the server, and boot to the installation media.
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 2019 Datacenter Edition (Server with a GUI), and click Next.
7. Check I accept the license terms, and click Next.
8. Click Custom: Install Windows only (advanced).
9. Select the 256GB M.2 SSD drive, and click Next. This starts Windows automatically, and Windows will boot 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 just set up.
12. From Server Manager, disable Windows Firewall.
13. Run Windows Updates.
14. Install the latest firmware and Windows device drivers from HPE.
15. Using the Disk Manager utility, create volumes on the SATA and SAS RAID disks.

Creating the NVMe volumes in Windows

We used the following steps to configure our NVMe SSD volumes. We created one named “NVMe” for our four test drives and one named “Offload” for our database backup, logging and tempdb using four NVMe SSDs.

1. On the server under test, open Server Manager.
2. Once Server Manager loads, click File and Storage Services on the left-hand side.
3. Click Storage Pools.
4. Under the Tasks dropdown menu, click New Storage Pool…
5. Enter a name for the new storage pool, and click Next.
6. On the Physical Disks screen, select the desired drives for the Storage Pool, and click Next.
7. Click Create.
8. Once the storage pool creation finishes, check the box Create a virtual disk when this wizard closes, and click Close.
9. Select the newly created Storage Pool, and click OK.
10. Enter a name for the new virtual disk, and click Next.
11. On the Enclosure Awareness screen, click Next.
12. On the Storage Layout screen, select Mirror, and click Next.
13. On the Provisioning screen, select Fixed, and click Next.
14. On the Size screen, select Maximum size, and click Next.

15. On the Confirmation screen, click Create.
16. Once the virtual disk creation completes, click Close.
17. Ensure that the new virtual disk is selected, and click Next.
18. On the size screen, verify the maximum size is entered, and click Next.
19. On the Drive Letter or Folder screen, select a drive letter for the volume, and click Next.
20. On the File System Settings screen, enter a Volume label, and click Next.
21. On the Confirmation screen, click Create.
22. When the volume creation completes, click Close.
23. Repeat steps 3 through 22 for any additional NVMe volumes.

Installing Microsoft SQL Server 2019

1. Copy the SQL Server 2019 installation ISO to the server's desktop, and mount the ISO.
2. Click Run SETUP.EXE.
3. In the left pane, click Installation.
4. Click New SQL Server stand-alone installation or add features to an existing installation.
5. On the Product Key screen, enter a product key. For our configuration, we selected Evaluation. Click Next.
6. To accept the license terms, click the checkbox, and click Next.
7. Click Use Microsoft Update to check for updates, and click Next.
8. At the Feature Selection screen, select Database Engine Services, Full-Text and Semantic Extractions for Search, Client Tools Connectivity, and Client Tools Backwards Compatibility.
9. Click Next.
10. At the Instance Configuration screen, leave the instance default selection, and click Next.
11. At the Server Configuration screen, leave the default settings, and click Next.
12. At the Database Engine Configuration screen, select the authentication method you prefer. For our testing purposes, we selected Mixed Mode.
13. Enter and confirm a password for the system administrator account.
14. Click Add Current user. This may take several seconds.
15. Click the TempDB tab.
16. Change the data and log directories for TempDB to the NVMe offload volume.
17. Leave the default settings for the other tabs, and click Next.
18. At the Ready to Install screen, click Install.
19. When the install is finished, go to the Installation tab in the Installation Center, and click Install SQL Server Management Tools.
20. In the browser that pops up, click Download SQL Server Management Studio.
21. When the download prompt appears, click Run.
22. When the installer finishes downloading and runs, click Install.
23. When the installation is complete, click Restart.

Creating the database

1. Open SQL Server Management Studio, and connect to the SQL instance.
2. Right-click Databases→New Database.
3. Name the database. We named ours `tpcc`.
4. Click Add to add eleven more database files for a total of twelve.
5. Pre-grow the log file to 50GB.
6. Name the database files, and click OK.
7. Change the file locations to the NVMe database directory and Offload log directory.
8. To create the database, click OK.

Configuring the client system and prepping the workload

For our client system, we configured a virtualized environment and created a Windows Server 2019 VM with 32 vCPUs and 64GB of RAM. We used this VM to drive the HammerDB workload. Below are the steps we used to configure the client after we installed the OS, configured the OS, and installed updates.

Installing HammerDB 3.2

1. Download HammerDB 3.2 from www.hammerdb.com/download.html.
2. Double-click the .EXE file, choose English, and click OK.
3. Click Yes.
4. Click Next.
5. Choose a destination location, and click Next.
6. Click Next.
7. Click Finish.

Populating the database

1. Open HammerDB on the client system, and click Options→Benchmark.
2. Choose Microsoft SQL and TPC-C.
3. Expand Schema Build, and double-click Options.
4. Choose SQL User (SA) connection, and enter the password.
5. Set the number of warehouses to `5000`.
6. Set the number of virtual users to `32`.
7. Click OK.
8. Expand Virtual Users, and double-click Create.
9. Click Run Virtual Users. The build will fail as we did not specify the IP of the SQL server.
10. In the Script Editor tab, scroll to the last line.
11. Edit (local) to be your SQL server IP.
12. Change `5000` warehouses to `14000`.
13. Click Run Virtual Users to start the build.

Backing up the database

1. On the SQL server, Open SQL Server Management Studio.
2. Right-click the TPC-C database, and click Tasks→Back up…
3. Select the Offload volume to store the backup file, and click OK.

Performing the test on Windows

1. On the client system, start HammerDB.
2. Set the database server to SQL Server, and set the workload to TPC-C.
3. Open the Options panel for the Virtual Users by clicking SQL Server→TPC-C→Virtual User→Options.
4. Select 96 Virtual Users, and select 1ms for the User Delay and the Repeat Delay.
5. Select the following:
   - "Show Output"
   - "Log Output to Temp"
   - "Use Unique Log Name"
   - "Log Timestamps"
6. Click OK.
7. Open the Options panel for the Driver: SQL Server→TPC-C→Driver Script→Options.
8. Enter the IP of the target SQL server.
9. Choose SQL Server Authentication connection, and enter the password.
10. Select Timed Driver Script. Select 15 minutes for Rampup and 15 minutes for the Test Duration.
11. Check the box for Use All Warehouses, and click OK.
12. Click Create Virtual Users.
13. Note the name of the log file, and click OK.
14. On the target SQL server, start a custom Perfmon data collector set to capture CPU, RAM, and disk performance counters.
15. Back on the client, click the green arrow to start the HammerDB workload.
16. When the run finishes, stop the data collector set on the target SQL server.
17. Save the HammerDB results text file and data collector output for review.
18. Open SQL Manager, delete the TPC-C database, and restore the database from the SQL backup file, targeting the SATA, SAS, or NVMe volumes for the data directory.
19. Reboot the server.

Read the report at http://facts.pt/zjrz4zd ▶

This project was commissioned by KIOXIA.

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.