



Hewlett Packard
Enterprise

HPE Reference Architecture for Microsoft SQL Server 2019 Standard Edition on an HPE ProLiant DL380 Gen10 with HPE Persistent Memory

Enterprise Analytics at best price/performance value proposition

Contents

Executive summary.....	3
Solution overview.....	4
HPE ProLiant DL380 Gen10 server.....	4
HPE Persistent Memory.....	4
Red Hat Enterprise Linux (RHEL 8) & Direct Access (DAX).....	5
Microsoft Windows Server 2019.....	5
Microsoft SQL Server 2019 Standard Edition.....	5
Solution components.....	7
Hardware.....	7
Software.....	8
Best practices and configuration guidance for the solution.....	8
BIOS Tuning/Configuration.....	8
HPE Persistent Memory BIOS Configuration.....	9
Capacity and sizing.....	9
Workload description.....	10
Test Configurations.....	10
Analysis and recommendations.....	11
Summary.....	14
Implementing a proof-of-concept.....	14
Appendix A: Bill of materials.....	15
Appendix B: HPE Persistent Memory configuration.....	16
Appendix C: Preparing HPE Persistent Memory Devices for use with SQL Server 2019.....	17
Appendix D: Individual query performance comparison between NVMe SSD and Persistent Memory configurations.....	20
Version History.....	24
Resources and additional links.....	25



Executive summary

Today's businesses require the fastest access to data to put it to work more quickly for better business outcomes. Businesses need solutions designed and built around the processes and applications that are at the core. With speeds closer to memory and the non-volatility of storage, HPE Persistent Memory, available in 128, 256, and 512 GB capacities and featuring Intel® Optane™ DC Persistent Memory, can transform critical database analytics workload. As byte addressable storage media, HPE Persistent Memory unlocks new levels of performance using Microsoft SQL Server 2019 that now brings in new features and enhancements with native support for persistent memory. Hewlett Packard Enterprise collaborated with Microsoft to devise and support a database solution involving these hardware and software technologies to deliver a high-performance, low-cost database solution. Key highlights of the solution are:

- Gain up to a 16.4x improvement in decision support system (DSS) performance when upgrading from SQL Server 2014 Standard Edition running on an HPE ProLiant Gen8 Server to SQL Server 2019 Standard Edition running on an HPE ProLiant Gen10 server with HPE Persistent Memory.
- Realize a performance improvement up to 3.6x with HPE Persistent Memory in analytic workloads over current generation NVMe technology with SQL Server 2019 Standard Edition deployments.
- Leverage the In-Memory Database architectures within SQL Server 2019 to enhance application performance with Direct Access (DAX) to HPE Persistent Memory.
- Achieve significant savings through reduced power and cooling requirements when upgrading from traditional low latency storage options to HPE Persistent Memory.

This solution is ideal for customers who are looking to:

- Upgrade from older versions of Microsoft SQL Server 2014 to SQL Server 2019.
- Refresh their hardware for Microsoft SQL Server database deployments.
- Improve the performance of their database deployments without significant increase in solution hardware cost.

While this solution is applicable for both Windows and Linux platforms, this document is focused on deployments running Linux.

Target audience: This Hewlett Packard Enterprise Reference Architecture is designed for Solution Architects and IT professionals who use, manage, or administer databases that require scalability, reliability, and performance. Specifically, this information is intended for those who evaluate, recommend, or design new IT high-performance architectures.

Document purpose: The purpose of this document is to describe a Reference Architecture, highlighting recognizable benefits to technical audiences.



Solution overview

This solution is running on Microsoft SQL Server 2019 Standard Edition on an HPE ProLiant DL380 Gen10 server equipped with HPE Persistent Memory running Red Hat® Enterprise Linux® 8. This high-capacity persistent memory product gives customers the flexibility to place either parts or a complete database in persistent memory. An overview of each critical hardware/software component of the solution is discussed in this section.

HPE ProLiant DL380 Gen10 server

The HPE ProLiant DL380 Gen10 server delivers the latest in security, performance, and expandability. It supports the second generation Intel® Xeon® Processor Scalable Family supporting HPE 2933 MT/s DDR4 SmartMemory. The HPE ProLiant DL380 Gen10 server has an adaptable chassis, including new HPE modular drive bay configuration options with up to 30 SFF, up to 19 LFF, or up to 20 NVMe drive options along with support for up to 3 double wide GPU options. Along with an embedded 4x1GbE, there is a choice of HPE FlexibleLOM or PCIe standup adapters which offer a choice of networking bandwidth (1GbE to 40GbE) and fabric allowing customers to adapt and grow to changing business needs. The HPE ProLiant DL380 Gen10 server comes with a complete set of HPE Technology Services, delivering confidence, reducing risk, and helping customers realize agility and stability.



Figure 1. HPE ProLiant DL380 Gen10 server

HPE Persistent Memory

HPE Persistent Memory modules are designed for use with second generation Intel Xeon Scalable processors, and are available in 128 GB, 256 GB and 512 GB capacities. HPE Persistent Memory modules use the standard DIMM form factor. These modules are installed alongside of regular DDR4 DRAM DIMMs in a server memory slot. HPE Persistent Memory supports three different operating modes, Memory mode, App Direct mode, and Mixed mode to ensure flexibility to accommodate any workload. Memory mode works as volatile memory. To an operating system, HPE Persistent memory is indistinguishable from DRAM. By taking advantage of HPE Persistent Memory's high capacity in Memory mode, it is like a memory-extender while maintaining DRAM-like performance. In App Direct mode, HPE Persistent Memory exercises the persistence capability together with its higher capacity while still achieving near DRAM performance. In Mixed mode, customers can choose a combination of HPE Persistent Memory in App Direct mode or Memory mode to best fit their business needs.

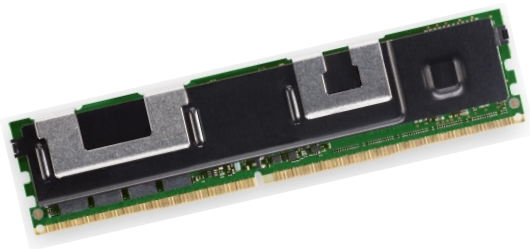


Figure 2. HPE Persistent Memory DIMM



The amount of persistent memory that can be installed in the HPE ProLiant DL380 Gen10 depends upon the choice of capacity of HPE Persistent Memory modules and the DIMM population scheme used. A maximum of 6 HPE Persistent Memory modules and 6 DRAM DIMMs can be installed per processor.

Total capacity of memory that can be installed with a processor is indicated by a suffix in the processor product SKU description. No memory suffix indicates that the processor can support up to 1 TB of memory. An “M” suffix indicates that the processor can support up to 2TB of memory. An “L” suffix can support up to 4.5 TB of memory per processor. It is important to note that the maximum memory capacity supported by a processor is inclusive of both installed HPE Persistent Memory and the regular volatile memory.

HPE Persistent Memory modules are installed in specific configurations based on the workload capacity and performance requirements. A total of 1, 2, 4, or 6 HPE Persistent Memory modules can be populated per processor. The best throughput can be achieved by using all channels in the configuration. For more information, refer to the population rules in the “Server memory and persistent memory population rules for HPE Gen10 servers with Intel Xeon Scalable processors” technical whitepaper at: <https://h20195.www2.hp.com/v2/Getdocument.aspx?docname=a00017079enw>.

HPE Persistent Memory operates at 1.2 volts and have a rated power consumption between 10 watts and 18 watts based on the operating profile and BIOS settings. This is significantly lower compared to NVMe devices that operate at 12 volts and typically consume around 25 watts of power. NVMe devices also require additional cooling to meet ambient temperature levels, which consumes additional power. For example, an HPE ProLiant DL380 Gen10 server requires a high performance fan kit when NVMe devices are installed.

Red Hat Enterprise Linux (RHEL 8) & Direct Access (DAX)

RHEL 8 provides support for a wide range of new and innovative technologies, which includes support for Persistent Memory.

Filesystem DAX enables applications to memory map data on byte-addressable storage such as persistent memory in to an application’s address space. RHEL 8 currently provides Filesystem DAX (FS DAX) support in Technology Preview mode only. However, RHEL 8 officially supports the FS DAX feature for SAP HANA and Microsoft SQL Server deployments given the extensive amount of testing and validation effort that was performed by HPE and Microsoft in collaboration with Red Hat. An official support statement from Red Hat can be found at <https://access.redhat.com/articles/4070821>

Microsoft Windows Server 2019

While not tested for this Reference Architecture, similar performance is expected with Windows Server® 2019.

Microsoft SQL Server 2019 Standard Edition

SQL Server 2019 Standard Edition, as with all versions since SQL Server 2016 SP1, have a common programmability surface area to ensure consistent development experience that scales across editions and cloud. Some limits are placed on scale and high availability in every edition except Enterprise edition which has unlimited scale and a full range of high availability options. While the primary scale limitation on SQL Server 2014 Standard Edition was a maximum of 16 CPU cores per SQL Server instance, this limitation has been increased to 24 cores per instance in later releases. All releases of Standard edition restrict the buffer pool size to 128 GB, including SQL Server 2019. An interesting aspect of Standard Edition of SQL Server 2019 Standard Edition is the ability to use features such as Hybrid Buffer Pool that leverages persistent memory as both a storage device and memory without the persistent memory capacity contributing to the buffer pool limit. The persistent memory capacity is not counted as part of the 128 GB limit calculated solely on the buffer pool usage in DRAM.

There are certain workloads that can achieve performance benefits, even on Standard edition, by incorporating In-Memory Database architectural awareness into their database design strategy. The following sections outline the new features introduced since SQL Server 2014 Standard Edition that have been used to enhance the performance of the workloads discussed in this document. The new persistent memory features were introduced in SQL Server 2019.

Enlightenment

Enlightenment is a SQL Server 2019 feature supported on Linux platforms. It enables database IO operations to bypass the traditional kernel IO subsystem resulting in breakthrough IO acceleration. Access to data and transaction logs is through load/store operations by memory mapping the files residing on persistent memory. SQL Server automatically enables the Enlightenment feature when it discovers files residing on persistent memory over a Direct Access (DAX) capable file system.



Hybrid Buffer Pool (HBP)

Similar to Enlightenment, Hybrid Buffer Pool (HBP) capitalizes on the memory mapping capability of data files. With this feature enabled, read consistent data pages are directly accessed on memory-mapped data files. Pages are only copied into the regular buffer pool when they need to be modified. This significantly reduces the amount of buffer pool space residing on regular volatile memory.

Performance gains achieved by this feature are completely workload dependent. If most of the accessed pages need to be duplicated in the buffer pool, overall performance gains may be limited. For the warehouse/TPC-H like workload discussed in this document, queries are mostly read-only and executed against a data set that does not involve many inserts or updates. These types of workloads see a significant reduction in buffer pool volatile memory requirements while experiencing enhanced performance. More information on the SQL Server 2019 Hybrid Buffer Pool feature can be found at <https://docs.microsoft.com/en-us/sql/relational-databases/in-memory-database>.

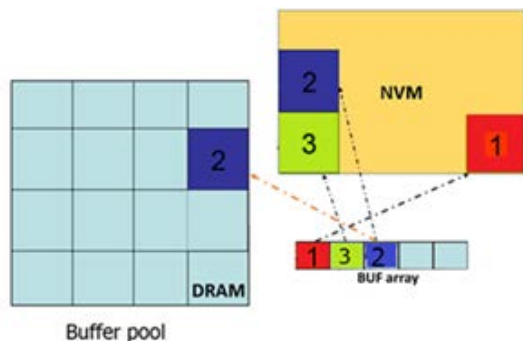


Figure 3. Hybrid Buffer Pool example

Figure 3 shows both the traditional buffer pool in volatile memory and HBP in non-volatile memory (NVM / persistent memory). Pages 1 and 3 are accessed in a read consistent state. Such accesses do not result in duplication of the pages into the buffer pool residing in volatile memory. In the case of data page 2, the page is marked for modification resulting in the page being duplicated in the buffer pool.

Clustered-Columnstore Index (CCI)

For the mostly read-only warehouse/TPC-H-like workloads discussed in this document, CCI greatly enhances performance. With CCI implemented on a table, data from a table's column is tightly packed into a single page. This boosts the performance of analytic queries while achieving best in class data compression. For a terabyte (TiB) of text data stored in tables (using a TPC-H like schema) with CCI indices, a SQL Server data file disk footprint of approximately 495 gigabytes (GiB) can be achieved. CCIs are supported with Standard edition licenses in SQL Server 2016 and later release. With SQL Server 2019 Standard Edition, a query's maximum degree of parallelism (DOP) is 2. The performance of analytic queries with CCIs and DOP=2 significantly outperform the same queries running against tables with non-CCI indices and the maximum allowed DOP. More about CCIs can be found on Microsoft's official page at: <https://docs.microsoft.com/en-us/sql/relational-databases/indexes/columnstore-indexes-described?view=sql-server-2014&viewFallbackFrom=sql-server-linux-ver15>.

This feature was introduced to the Standard edition with SQL Server 2016. It was introduced to the Enterprise edition with SQL 2014.

A significant portion of the performance gains demonstrated in this document with SQL Server 2019 Standard Edition versus SQL Server 2014 Standard Edition can be attributed to the CCI feature.

Partitioning Schemes

This feature allows tables and indices to be partitioned to maximize performance with parallel operations. The warehouse/TPC-H-like schema used in the evaluation workload was partitioned based on a date scheme placed into week-based partitions. Support for partitioning scheme was introduced for Standard edition with SQL Server 2016.

Note

In SQL Server 2019, batch mode query processing on rowstore has been introduced. This feature unlocks the advantages of batch mode execution in cases where there is no columnstore participating in the query. This is primarily targeted at analytics queries which are characterized as scanning many rows, and doing significant aggregations, sorts, and group-by operations across these rows. Batch mode has not been enabled for queries which involve rowstore in this Reference Architecture.



Solution components

Linux support for SQL Server was launched with SQL Server 2017. For the solution discussed in this document, SQL Server 2019 Standard Edition was used on an HPE ProLiant Gen10 server running Red Hat Enterprise Linux 8. The baseline performance tests were done on SQL Server 2014 Standard Edition running Windows Server 2016 on an HPE ProLiant Gen8 Server. The following sub-sections discuss the hardware components, software components and the workload semantics used in each of the configurations.

Hardware

Table 1. Details of the hardware configurations tested

	HPE ProLiant DL380 Gen10 with HPE Persistent Memory	HPE ProLiant DL380 Gen10 with NVMe Drives	HPE ProLiant DL380p Gen8 with SAS SSDs
Processor	Two 12-core Intel Xeon Gold 6246 CPU @ 3.30 GHz	Two 12-core Intel Xeon Gold 6246 CPU @ 3.30 GHz	Two 8-core Intel Xeon E5-2687W v2 @ 3.40 GHz
Hyper-Threading	Disabled for columnstore database schemas Enabled for rowstore database schemas	Disabled for columnstore database schemas Enabled for rowstore database schemas	(columnstore schemas not available) Enabled for rowstore database schemas
Memory	192 GB of Volatile Memory (12 x HPE 16GB DDR4 2933 MT/s SmartMemory R-DIMMs)	192 GB of Volatile Memory (12 x HPE 16GB DDR4 2933 MT/s SmartMemory R-DIMMs)	192 GB of Volatile Memory (12 x HPE 16GB PC3-14900R 1866 MT/s SmartMemory R-DIMMs)
Ethernet	HPE Ethernet 10/25Gb 2-port 631FLR-SFP28 Adapter	HPE Ethernet 10/25Gb 2-port 631FLR-SFP28 Adapter	Embedded 1GbE Network Adapter
OS Drives	2 x HPE 300 GB SAS 12G Enterprise 10K SFF	2 x HPE 300 GB SAS 12G Enterprise 10K SFF	2 x HPE 400 GB 12G SAS SSD Mainstream Endurance SFF
Database Storage	8 x HPE 128 GB Persistent Memory modules for the columnstore configuration. --and-- 8 x HPE 256 GB Persistent Memory modules for the rowstore configuration	8 x HPE 1.92 TB NVMe x4 Lanes Read Intensive SFF	8 x HPE 400 GB 12G SAS SSD Mainstream Endurance SFF
Backup Drives	6 x HPE 600 GB SAS 12G Enterprise 15K SFF	6 x HPE 600 GB SAS 12G Enterprise 15K SFF	6 x HPE 600 GB 12G SAS 15K rpm SFF
Storage Controllers	1 x HPE Smart Array P408i-a SR Gen10	1 x HPE Smart Array P408i-a SR Gen10	3 x HPE Smart Array P420i Ctrlr FIO Reman Kit

Disclaimer

Products sold prior to the separation of Hewlett-Packard Company into Hewlett Packard Enterprise Company and HP Inc. on November 1, 2015 may have a product name and model number that differ from current models.



Software

Table 2 lists the details of test environment.

Table 2. Details of test environment

	HPE ProLiant DL380 Gen10 with HPE Persistent Memory	HPE ProLiant DL380 Gen10 with NVMe SSDs	HPE ProLiant DL380p Gen8 with SAS SSDs
OS	Red Hat Enterprise Linux 8	Red Hat Enterprise Linux 8	Windows Server 2016
Database Version	SQL Server 2019 Standard Edition	SQL Server 2019 Standard Edition	SQL Server 2014 Standard Edition
Data file and TempDB storage	6 Persistent Memory devices	6 NVMe SSDs	6 SAS SSDs
Transaction log storage	2 PMEM devices with software RAID1 using mdadm	2 NVMe SSDs with software RAID1 using mdadm	2 SAS SSDs with RAID1 using an HPE Smart Array Controller.
Database backup storage	2 600 GB HDDs with RAID0 using an HPE Smart Array Controller	2 600 GB HDDs with RAID0 using an HPE Smart Array Controller	2 600 GB HDDs with RAID0 using an HPE Smart Array Controller

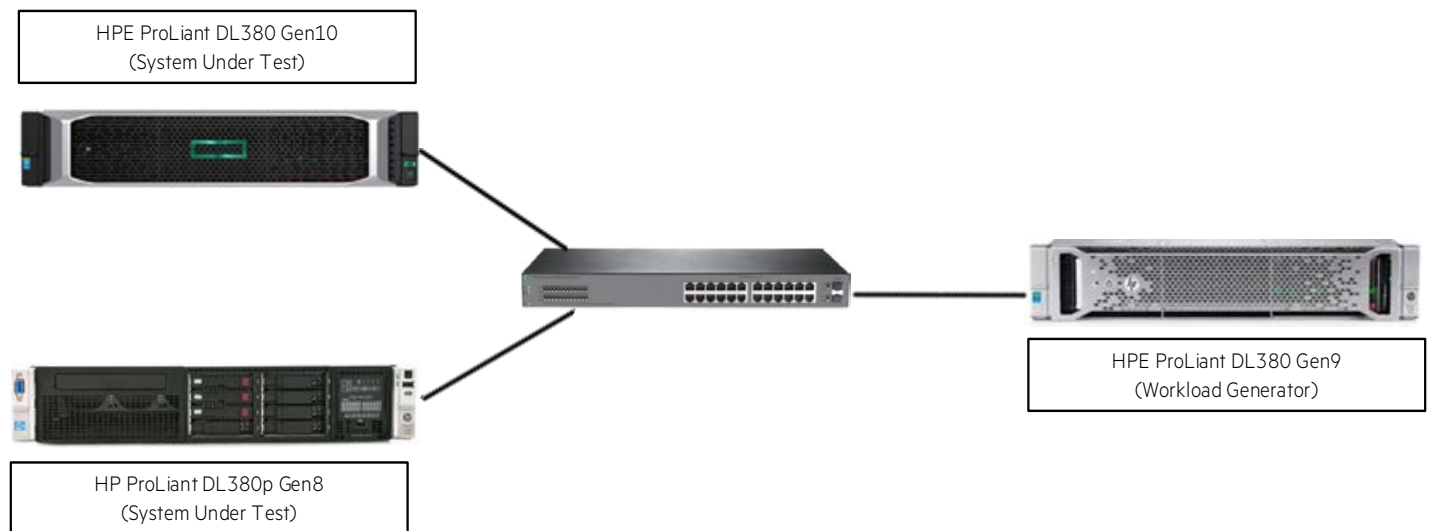


Figure 4. A high-level overview of test environment

Best practices and configuration guidance for the solution

BIOS Tuning/Configuration

Workload Profiles, one of the HPE Intelligent System Tuning (IST) features, allows tuning of the resources in an HPE ProLiant Gen10 server by choosing a preconfigured workload profile. The server will automatically configure the BIOS settings to match the selected workload. The High Performance Compute (HPC) workload profile designed for sustained maximum utilization rates for extended periods was selected in order to achieve the best performance for the warehouse workload.



The following BIOS options were used:

- WorkloadProfile= HighPerformanceCompute(HPC)
- LlcPrefetch=Enabled
- XptPrefetcher=Enabled
- ThermalConfig=EnhancedCooling

On the HPE ProLiant Gen8 Server used for the SQL Server 2014 Standard Edition baseline comparisons BIOS options were manually set to closely mimic the HPC workload profile in HPE ProLiant Gen10 servers.

HPE Persistent Memory BIOS Configuration

This section provides a high-level overview of key HPE Persistent Memory BIOS settings. Refer to Appendix B for a detailed sequence configuring the BIOS options. Appendix C describes how to prepare the Persistent Memory devices for use with SQL Server.

- VolatileMemoryCapacity=0%
- PersistentMemoryInterleaving=Disabled
- PerformanceSetting=Bandwidth-Optimized
- SnoopyModeForAppDirect=Enabled

An exhaustive reference to the HPE Persistent Memory BIOS configuration can be found in the “HPE Persistent Memory User Guide for HPE ProLiant Gen10 servers and HPE Synergy” at: https://support.hpe.com/hpsc/doc/public/display?docId=emr_na-a00074717en_us&docLocale=en_US.

HPE Persistent Memory can be configured for use in Memory mode or App Direct mode. For the solution in this document, HPE Persistent Memory was used in App Direct mode. By setting “VolatileMemoryCapacity” to zero in the BIOS HPE Persistent Memory configuration options, all available HPE Persistent Memory capacity is enabled for App Direct mode usage.

Interleaving enables striping of data across all HPE Persistent Memory Modules (HPE PMM) within a socket while presenting a single persistent memory device special file (DSF) to the operating system. A general performance recommendation is to enable interleaving, however, for the workload discussed in this paper, it is best to disable interleaving. The user has better control on the placement of data files for performance and availability when each of the HPE Persistent Memory modules are presented to the operating system as individual DSFs. Microsoft SQL Server ensures that the content of a database is uniformly distributed across all available data files when sized appropriately.

Capacity and sizing

With the solution proposed in this document, HPE Persistent Memory is configured to run in App Direct mode. For a database scale factor of 1.0 TB, eight of 128 GB modules for the columnstore configuration and eight of 256 GB modules for the rowstore configuration are utilized. These modules are populated alongside of 12 DDR4 DIMMs. All databases data files, transaction log files, and TempDB files are hosted on HPE Persistent Memory using the XFS filesystem. However, users can choose to place only parts of the database in persistent memory.

Database Storage Capacity Requirement

The total amount of HPE Persistent Memory capacity required for a database is dependent on how the database schema is designed. With a database schema utilizing Clustered Columnstore indices built on tables, there is a significant amount of reduction in required storage capacity. For a database scale at 1.0 TB, the data file storage space requirement is less than 50%. The total storage capacity required by the database was satisfied by using eight of the 128 GB HPE Persistent Memory modules. Similarly, the database schema with rowstore indices, the storage capacity requirement was close to 125% of the actual database scale. For this configuration, eight of the 256 GB capacity HPE Persistent Memory DIMMs were used.

TempDB Database Storage Capacity Requirement

Analytic queries heavily depend on TempDB in their execution. It is strongly recommended that data files for the TempDB be hosted on persistent memory for best performance. Space requirement for a specific workload is again based on the schema. Columnstore configurations tend to use a lower amount of TempDB space while rowstore configuration have larger requirements. For the tests in this document, TempDB



space required with a columnstore schema was in the range of 4-5% of the scale factor while the schema with rowstore indices had a requirement of about 20% of the database scale.

Transaction Log Storage Capacity Requirement

For the tests in this document, transaction log space required was in the range of 7-9% of the database scale factor. To address availability requirements two different HPE Persistent Memory modules from two different processors were configured for software RAID 1. The Linux `mdadm` command was used to configure RAID 1.

Processor sizing for databases with columnstore indices

With SQL Server Standard Edition, databases with columnstore indices have a per query maximum degree of parallelism (DOP) of 2. Identify the maximum number of queries that will ever execute in parallel, multiple the result by 2, and divide by the number of processors that will be installed. This calculation will provide the number of cores per processor needed for maximum performance. Databases with rowstore indices do not have the DOP=2 limit and can take advantage of all cores in a processor.

Workload description

The primary workload tested represents a data warehouse environment with a TPC-H-like schema. The target database was at 1 TB scale. All the 22 queries are executed independently in various modes to demonstrate the performance traits of the proposed solution. The following sections provide a brief description of each of these modes.

Individual queries

In this mode, individual analytic queries are executed and their elapsed times recorded for analysis against each configuration. A query that is run when the database's buffer pool is completely empty is known as a cold query. The buffer pool can be emptied by either dropping all clean buffers and the procedure cache or by restarting the SQL Server instance. A query that finds some or all of the needed data cached in the buffer pool is known as a warm query.

The performance improvement provided by adopting faster storage devices can be measured by comparing an individual query's cold and warm run performance. The performance improvement of a cold query run is the absolute best possible improvement for that query. A cold query run will always obtain data from the storage device. Conversely, the performance improvement of a warm query run is the bare minimum performance gain for that query. With a warm query, some or all of the data is already cached in the buffer pool so obtaining the data from the storage device may not be necessary. Since the content of the buffer pool is completely dedicated to the same exact query's data set, the absolute minimum performance improvement for that storage device can be derived.

Query streams

Comparing the performance of individual queries demonstrates how each query would take advantage of the new storage tier provide by HPE Persistent Memory. However, this does not shed light on the same query's performance when batched together with other queries. A query stream is the execution of a sequence of all 22 queries one after the other. This comparison brings the evaluation one-step closer to a typical production environment where various types of analytic queries are run as a batch.

Parallel query streams

This workload is more representative of typical production environments with multiple users running queries against the database in parallel. In the comparisons below, seven parallel query streams of 22 queries each are used. Queries are ordered differently in each query stream. The elapsed time between the start of first parallel query stream execution and end of last parallel query stream execution is compared.

By comparing the results from parallel query streams, more realistic performance expectation HPE Persistent Memory and all the software enhancements enabling its use can be built.

Test Configurations

The primary intent of this document is to provide performance guidance for existing SQL Server Standard Edition deployments that have reached or are reaching their end-of-life support or are targeted for a hardware/software refresh. For this specific reason, gathering performance baselines against databases running SQL Server 2014 Standard Edition on Windows Server was chosen.

Standard edition deployments were evaluated as they allow the use of persistent memory without any memory size limitation that helps in realizing the best price/performance metric. As discussed earlier, SQL Server 2019 Standard edition supports columnstore indices (albeit with limitations) resulting in even better performance for analytic queries. This gives rise to a dilemma for users to consider the continued use of



rowstore indices or to switch to columnstore indices. Extensive testing shows the performance improvement for both rowstore and columnstore indices.

Not all performance gains realized when upgrading to an HPE ProLiant DL380 Gen10 server with HPE Persistent Memory can be attributed to persistent memory. To build a clear understanding of performance gains delivered by persistent memory, comparisons were made against NVMe SSD storage devices that are considered the most performant in the solid state device category. The following configurations are compared:

- SQL Server 2014 Standard Edition with rowstore indices on SAS SSD storage
- SQL Server 2019 Standard Edition with rowstore indices on NVMe SSD storage
- SQL Server 2019 Standard Edition with rowstore indices on Persistent Memory
- SQL Server 2019 Standard Edition with columnstore indices on NVMe SSD storage
- SQL Server 2019 Standard Edition with columnstore indices on Persistent Memory

The SQL Server 2014 baseline data was obtained with an HPE ProLiant DL380p Gen8 Server that was available in the timeframe SQL Server 2014 was released. The SQL Server 2019 data was obtained with the HPE ProLiant DL380 Gen10 server.

Analysis and recommendations

Comparing the performance of all five configurations across parallel query streams, a single query stream and individual queries, configurations with HPE Persistent Memory clearly outperform peer configurations using NVMe SSD and SAS SSD storage devices. While the performance delta between the older generation database and hardware is overwhelming, as expected, HPE Persistent Memory configurations demonstrated multifold performance over the current generation server with NVMe SSD storage.

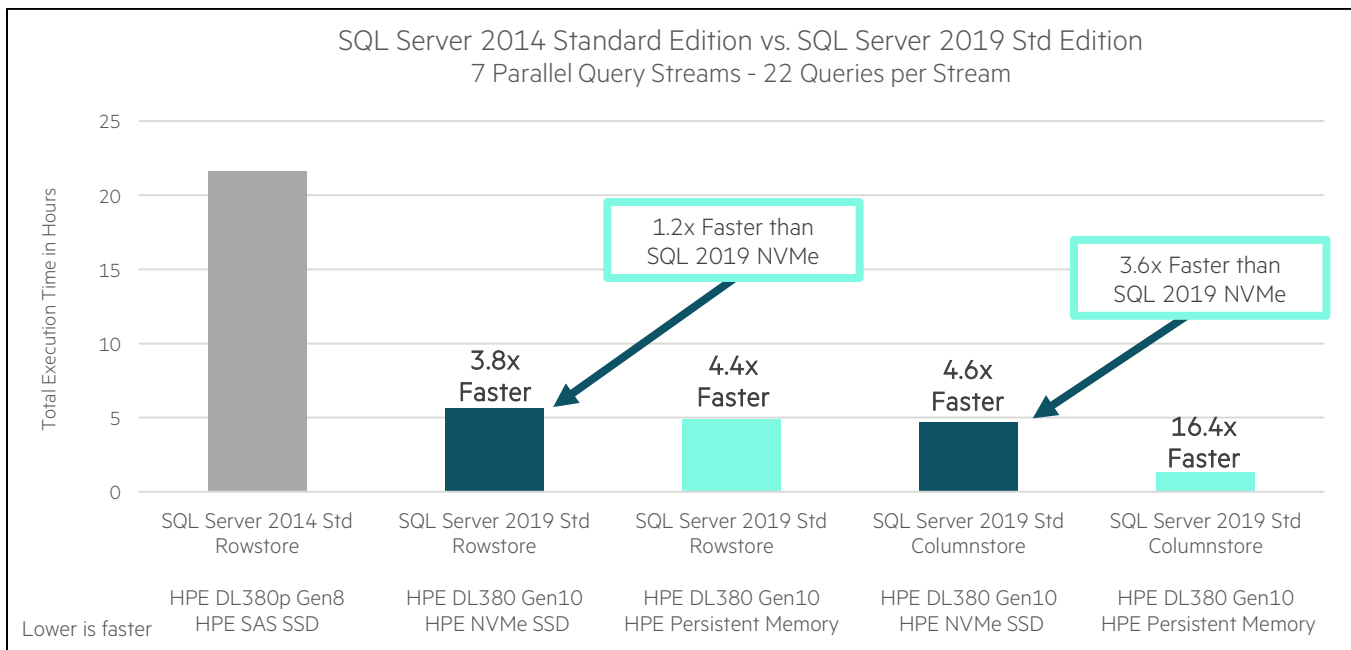


Figure 5. Comparison of elapsed execution time of seven parallel query streams for all configurations



SQL Server 2019 columnstore configuration using HPE Persistent Memory completed its execution 16.4x faster than the SQL Server 2014 baseline. A similar columnstore configuration using NVMe SSD storage completed the workload run 4.6x faster. Categorically comparing the performance of rowstore configurations with columnstore configurations, the latter can be seen performing noticeably better even though the maximum degree of parallelism is constrained to a value of two (DOP=2).

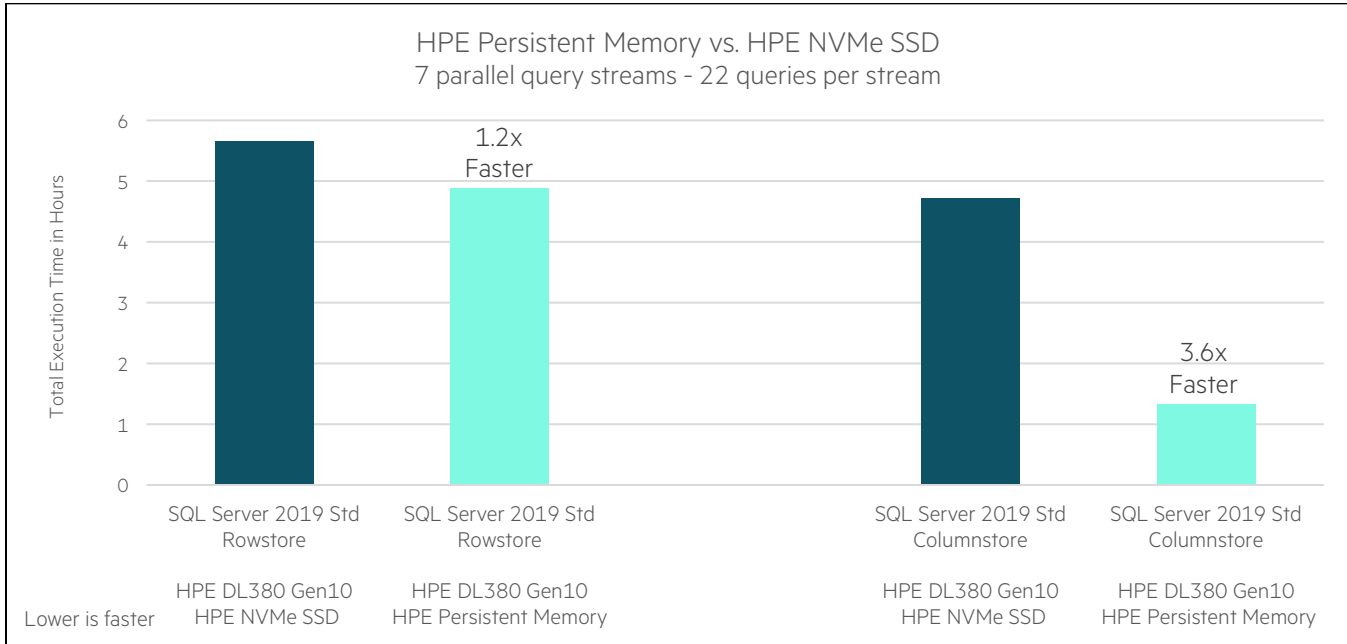


Figure 6. Comparison of elapsed execution time of seven parallel query streams for HPE NVMe SSD and HPE Persistent Memory

Drawing a comparison for the same workload with parallel query streams between NVMe SSD and Persistent Memory configurations, the Persistent Memory configuration using columnstore indices outperforms the NVMe SSD configuration by 3.6 times.

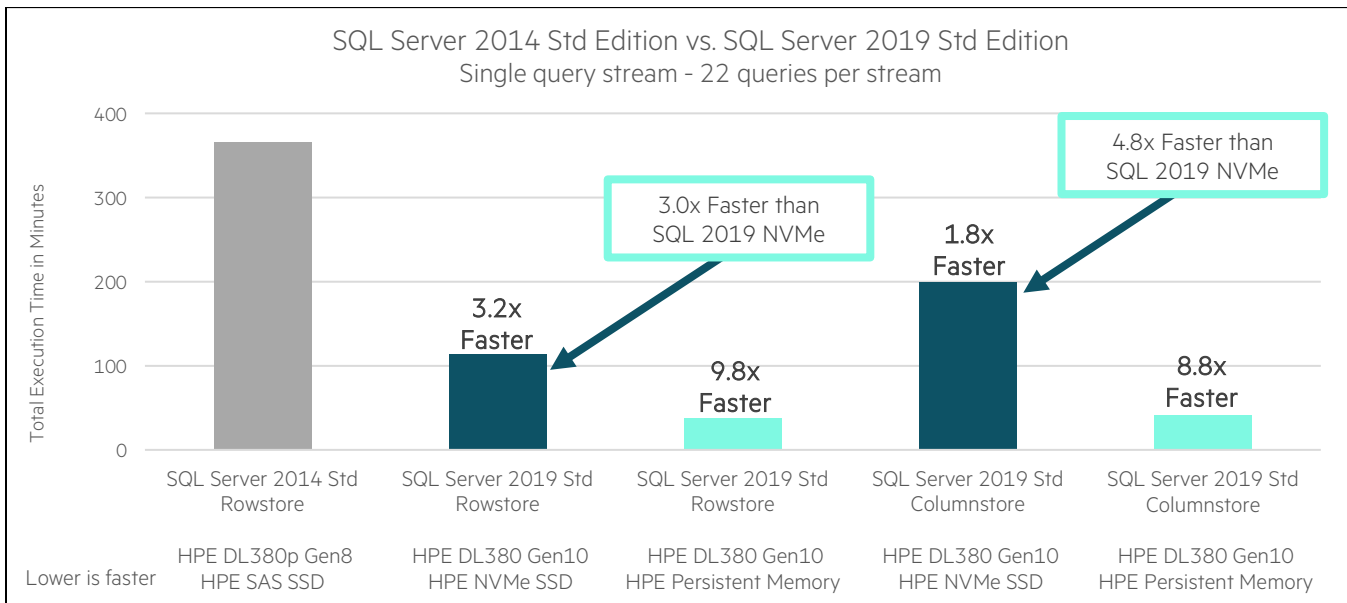


Figure 7. Comparison of elapsed execution time of a single query stream for all configurations



Switching to a single query stream, data showed that there is no noticeable performance delta between a cold query stream and a warm query stream due to the small buffer pool size. The persistent memory configuration with columnstore indices is 8.8 times faster than the baseline configuration. In comparison, workloads like the seven parallel query streams have higher IO throughput requirements and benefit more from persistent memory than a single query stream.

Rowstore configurations outperformed columnstore configurations when running single query streams because rowstore schemas are not limited by DOP=2.

The last performance comparison is with individual cold and warm queries. Performance expectations remain the same as discussed for other workload modes. While this is a significant amount of data, users familiar with TPC-H-like schemas and the constituent queries can build a better understanding of which queries stand to benefit more from HPE Persistent Memory.

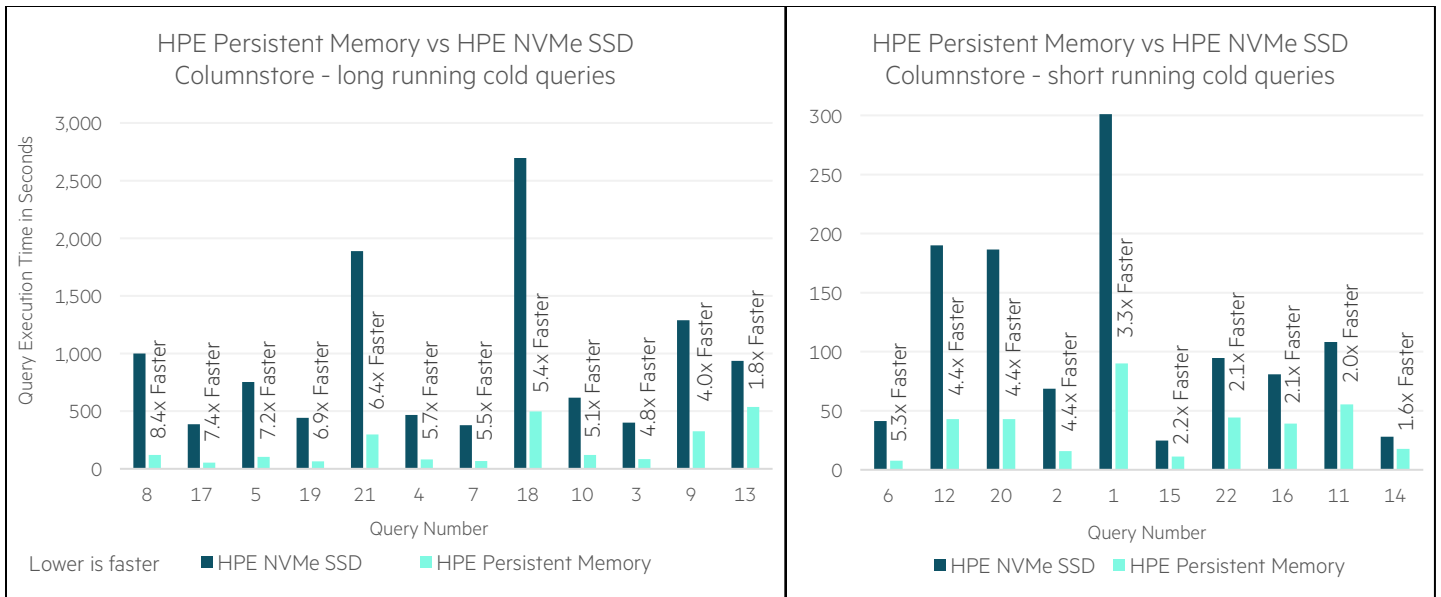


Figure 8. Elapsed times of Individual cold query runs from HPE Persistent Memory and HPE NVMe SSD configurations



As noted earlier, cold query runs enjoy the absolute best improvement in performance with HPE Persistent Memory while warm query runs realize the minimum performance gains (which is still significant in many cases). The following figure shows HPE Persistent Memory vs. HPE NVMe SSD performance comparisons of individual warm queries against a columnstore database.

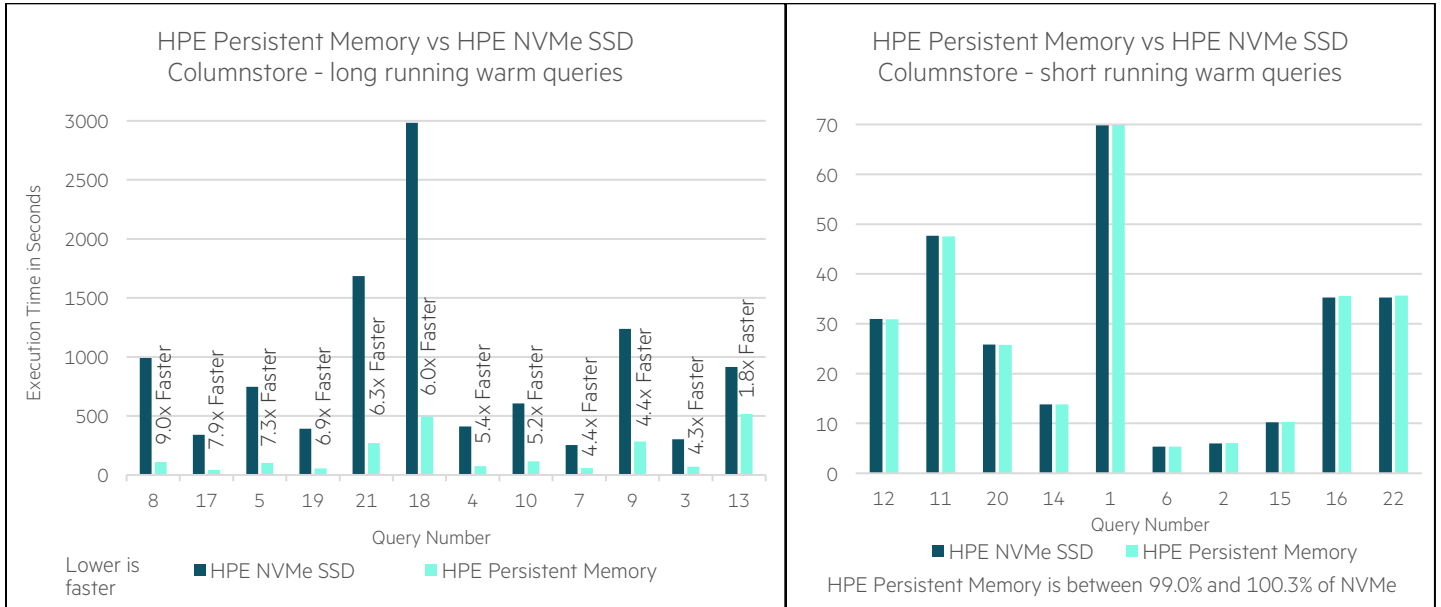


Figure 9. Elapsed times of Individual warm query runs from HPE Persistent Memory and HPE NVMe SSD configurations

When running warm queries, short running queries perform similar between HPE Persistent Memory and HPE NVMe SSD configurations. The limitation of a 128 GB buffer pool is adequate for some of these short running queries. Warm runs of these queries make effective use the buffer pool in NVMe SSD configurations. With limited data pages being accessed, the total access time between the standard buffer pool and the HBP is negligible.

Please refer to Appendix D for more charts comparing performance of individual query runs with both row and columnstore.

Summary

HPE Persistent Memory as a fast storage tier on HPE ProLiant Gen10 servers supporting second-generation Intel Xeon Scalable processors has a compelling value proposition for accelerating analytic workloads and other database workloads in general. Combined with Microsoft’s SQL Server 2019 release that enables an In-Memory Database design strategy for Persistent Memory, customers can realize significant improvements in performance.

This Reference Architecture describes solution testing performed in September and October 2019.

Implementing a proof-of-concept

As a matter of best practice for all deployments, Hewlett Packard Enterprise recommends implementing a proof-of-concept using a test environment that matches as closely as possible the planned production environment. In this way, appropriate performance and scalability characterizations can be obtained. For help with a proof-of-concept, contact a Hewlett Packard Enterprise Services representative (hpe.com/us/en/services/consulting.html) or your Hewlett Packard Enterprise partner.



Appendix A: Bill of materials

Note

Part numbers are at time of publication/testing and subject to change. The bill of materials does not include complete support options or other rack and power requirements. If you have questions regarding ordering, please consult with your HPE Reseller or HPE Sales Representative for more details. hpe.com/us/en/services/consulting.html.

Table 1a. Bill of materials for two HPE ProLiant DL380 Gen10 servers; one with NVMe storage and another with Persistent Memory

Quantity	Part number	Description
HPE ProLiant DL380 Gen10 server hardware		
1	868703-B21	HPE DL380 Gen10 8SFF CTO Server (Configure-to-order)
1	P15758-B21	HPE DL380 Gen10 Intel Xeon-Gold 6246 (3.3GHz/12-core/165W) Processor Kit
1	P15758-L21	HPE DL380 Gen10 Intel Xeon-Gold 6246 (3.3GHz/12-core/165W) FIO Processor Kit
12	P00922-B21	HPE 16 GB (1x16GB) Dual Rank x8 DDR4-2933 CAS-21-21-21 Registered Smart Memory Kit
1	817709-B21	HPE Ethernet 10/25Gb 2-port 631FLR-SFP28 Adapter
1	804331-B21	HPE Smart Array P408i-a SR Gen10 (8 Internal Lanes/2GB Cache) 12G SAS Modular Controller
2	872475-B21	HPE 300 GB SAS 12G Enterprise 10K SFF (2.5in) SC 3yr Wty Digitally Signed Firmware HDD
6	870757-B21	HPE 600 GB SAS 12G Enterprise 15K SFF (2.5in) SC 3yr Wty Digitally Signed Firmware HDD
Database Storage Option #1: HPE NVMe Storage for Column or Rowstore database		
8	P10214-B21	HPE 1.92 TB NVMe x4 Lanes Read Intensive SFF (2.5in) SCN 3yr Wty Digitally Signed Firmware SSD
Database Storage Option #2 HPE Persistent Memory for Columnstore database		
8	835804-B21	HPE 128 GB 2666 Persistent Memory Kit featuring Intel Optane DC
Database Storage Option #3: HPE Persistent Memory for Rowstore database		
8	835807-B21	HPE 256 GB 2666 Persistent Memory Kit featuring Intel Optane DC

Table 1b. Bill of materials (sample for the HPE ProLiant DL380p Gen8 server)

Quantity	Part number	Description
HPE ProLiant DL380p Gen8 server hardware		
1	653200-B21	HPE ProLiant DL380p Gen8 8 SFF Configure-to-order Server (Configure-to-order)
2	E2Q80AA	HPE DL380p Gen8 Xeon E5-2687W v2 8C 3.40 25MB 1866 CPU
12	708641-B21	HPE 16 GB (1x16GB) Dual Rank x4 PC3-14900R (DDR3-1866) Registered CAS-13 Memory Kit
1	684208-B21	HPE Ethernet 1 Gb 4-port 331FLR FIO Adapter
3	692276R-B21	HPE Smart Array P420i Ctrlr FIO Reman Kit
10	741142-B21	HPE 400 GB 12G SAS Mainstream Endurance SFF 2.5-in SC Enterprise Mainstream 3yr Wty Solid State Drive
6	759212-B21	HPE 600 GB 12G SAS 15K rpm SFF (2.5-inch) SC Enterprise 3yr Warranty Hard Drive

Disclaimer

Products sold prior to the separation of Hewlett-Packard Company into Hewlett Packard Enterprise Company and HP Inc. on November 1, 2015 may have a product name and model number that differ from current models.



Appendix B: HPE Persistent Memory configuration

Most of the Persistent Memory configuration is typically handled at the firmware level using a dedicated configuration section in the BIOS options. If a systemd based Linux variant exists on the server, issue the following command to reboot it to the BIOS/System options page:

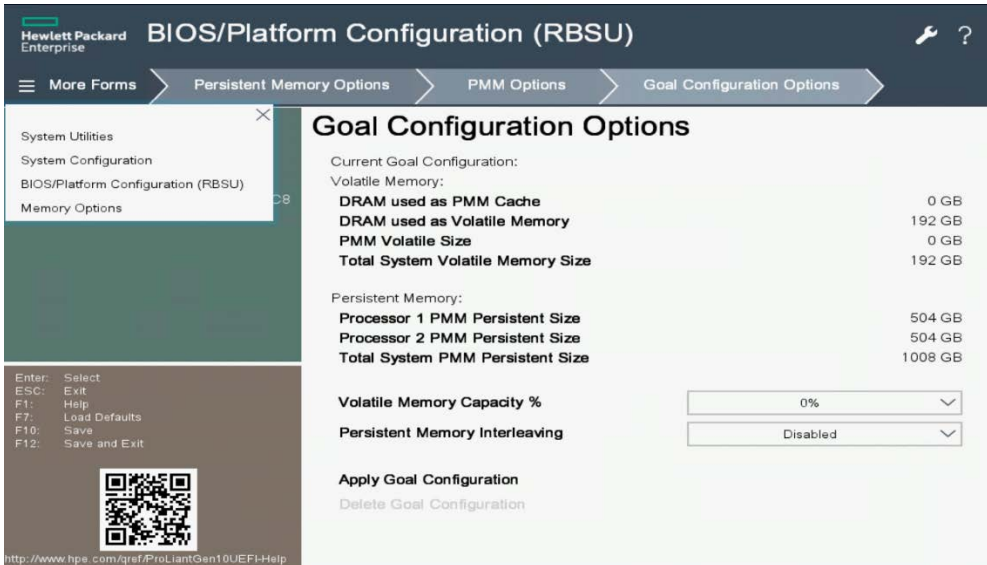
```
# systemctl reboot -firmware-setup
```

For any other operating system or with a server with no OS, reset the server and switch to the BIOS/System options by pressing the “F9” key while at POST (Power On Self-Test) screen.

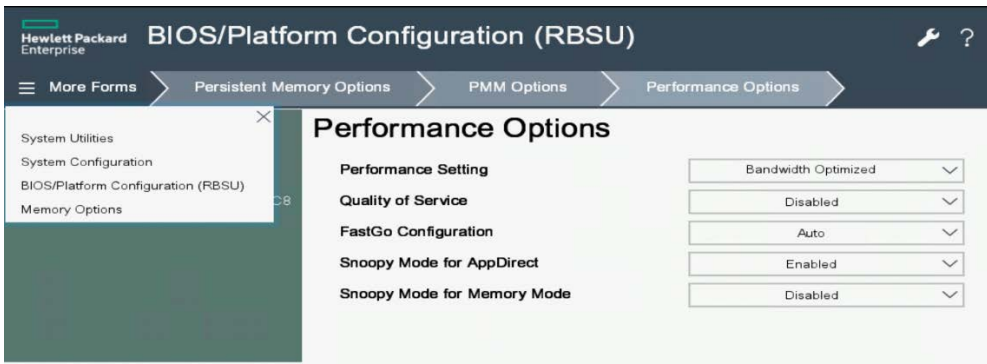
1. Once at the System Configuration menu, navigate to Persistent Memory configuration page.



2. Set Goal Configuration as follows and apply the changes.



3. Set Persistent Memory performance options.



Appendix C: Preparing HPE Persistent Memory Devices for use with SQL Server 2019

Once the HPE Persistent Memory BIOS configuration is set and the server is booted to the operating system, the preparation sequence can begin. By default, namespaces for the Persistent Memory devices are not set and there are no `/dev/pmem*` DSFs created.

1. Check if any namespaces already exist using the `ndctl` command.

```
# ndctl list
```

2. If any stale namespace exist, delete them using following `ndctl` command syntax.

```
# ndctl disable-namespace -n <namespace as listed in above command>
```

```
# ndctl destroy-namespace -n <namespace as listed in above command>
```

3. Create a namespace per persistent memory region listed by the `"ndctl list"` command (multiple namespaces can be created if desired using `-size flat` of `ndctl` command). To list the available Persistent Memory regions, run the following command.

```
# ndctl list --regions -v
```

A sample output looks like:

```
# ndctl list --regions -v
[
 {
   "dev": "region1",
   "size": 135291469824,
   "available_size": 135291469824,
   "max_available_extent": 135291469824,
   "type": "pmem",
   "numa_node": 0,
   "iset_id": -7530904164068980190,
   "persistence_domain": "memory_controller"
 },
 ...
```

4. Create a namespace per region as follows.

```
# ndctl create-namespace --region=region<region number as listed in above command> --mode=fsdax --map=mem --sector-size=4096 --verbose --name=<a name for the namespace>
```

A sample command and its output would look like:

```
# ndctl create-namespace --region=region0 --mode=fsdax --map=mem --sector-size=4096 --verbose --name=proc0_dimm1
{
  "dev": "namespace0.0",
  "mode": "fsdax",
```



```

"map": "mem",
"size": "126.00 GiB (135.29 GB)",
"uuid": "564c77b5-8dc0-4951-bbbb-ba57cd84f206",
"raw_uuid": "688de788-1f3d-4815-bb8b-871eaa9abc47",
"sector_size": 4096,
"blockdev": "pmem0",
"name": "proc0_dimm1",
"numa_node": 0
}

```

- After creating namespaces for each of the persistent memory regions, the namespaces and their configuration can be listed using the `ndctl` command.

```

# ndctl list -v
[
{
  "dev": "namespace1.0",
  "mode": "fsdax",
  "map": "mem",
  "size": 135289372672,
  "uuid": "239efbaa-d2dd-495f-8e84-507408b014c9",
  "raw_uuid": "fb93e60b-e8c4-4fd8-a17e-5ee3dd87a18d",
  "sector_size": 4096,
  "blockdev": "pmem1",
  "name": "proc0_dimm2",
  "numa_node": 0
},
...

```

- After successfully creating the namespaces, the DSFs can now be listed as follows.

```
# ls /dev/pmem*
```

Sample output is as follows:

```
# ls /dev/pmem*
```



```
/dev/pmem0 /dev/pmem2 /dev/pmem3.1 /dev/pmem5 /dev/pmem7  
/dev/pmem1 /dev/pmem3 /dev/pmem4 /dev/pmem6 /dev/pmem7.1
```

7. To create a filesystem on the devices and use them for SQL Server consumption, follow instructions from, <https://docs.microsoft.com/en-us/sql/linux/sql-server-linux-configure-pmem?view=sqlallproducts-allversions>, and https://pmem.io/2018/05/15/using_persistent_memory_devices_with_the_linux_device_mapper.html.



Appendix D: Individual query performance comparison between NVMe SSD and Persistent Memory configurations

This section has the detailed performance comparison charts for individual queries between persistent memory and NVMe SSD configurations. The following chart compares the cold query performance running longer than 350 seconds with the columnstore schema.

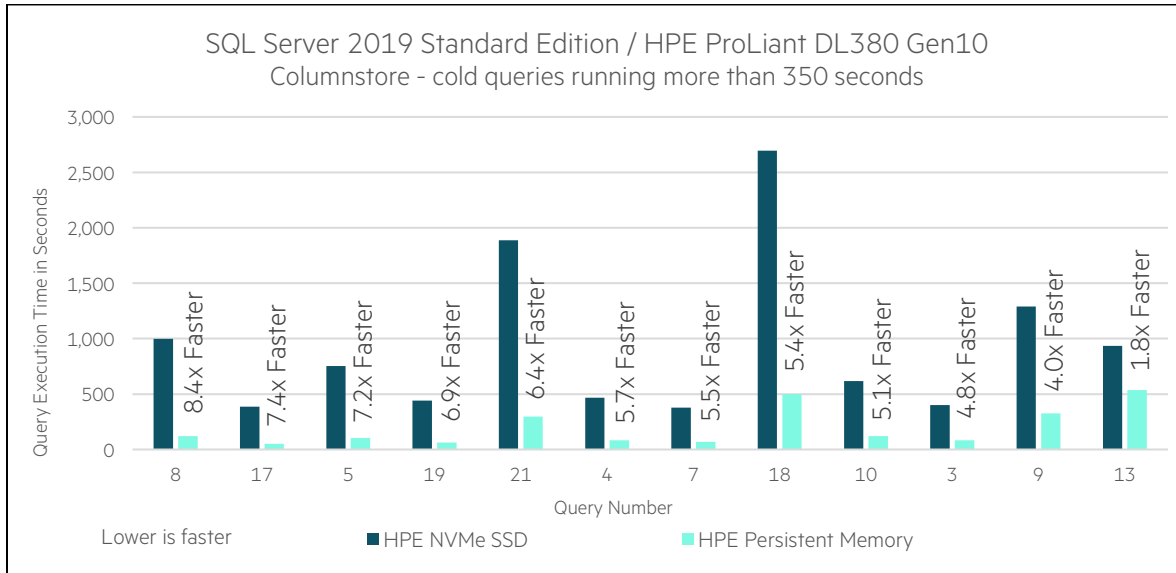


Figure 10a. Elapsed times of long running queries when run cold – database with columnstore indices

The following chart compares the cold query performance running less than 350 seconds with the columnstore schema.

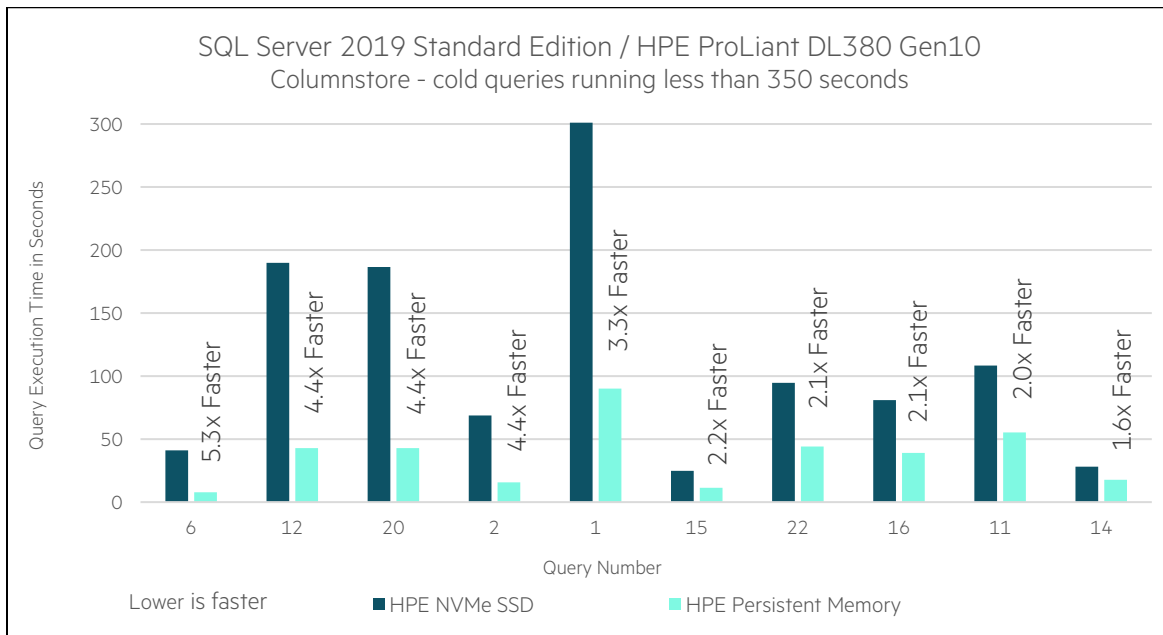


Figure 10b. Elapsed times of short running queries when run cold– database with columnstore indices



The following chart compares the warm query performance running longer than 300 seconds with the columnstore schema.

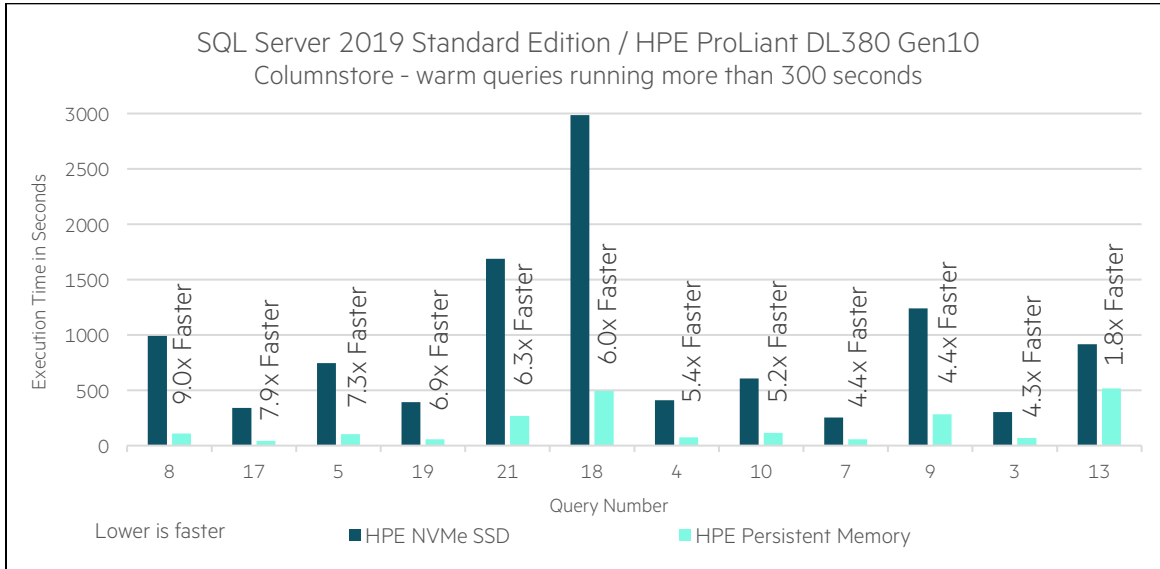


Figure 11a. Elapsed times of long running queries when run warm- database with columnstore indices

The following chart compares the warm query performance running less than 100 seconds for the columnstore schema.

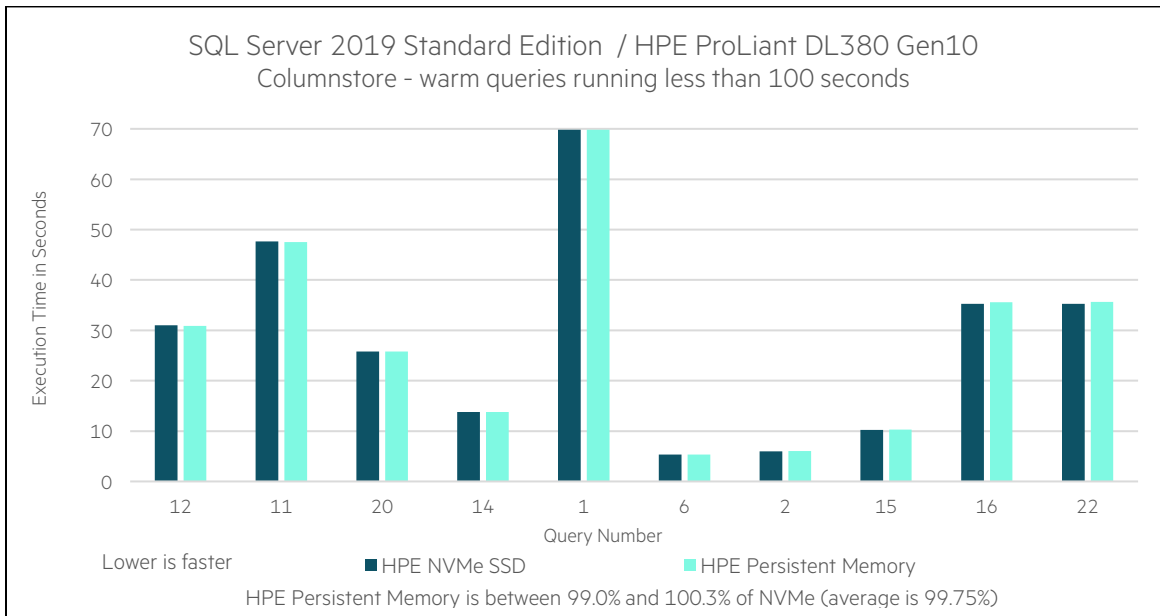


Figure 11b. Elapsed times of short running queries when run warm- database with columnstore indices



The following chart compares the cold query performance running longer than 250 seconds with the rowstore schema.

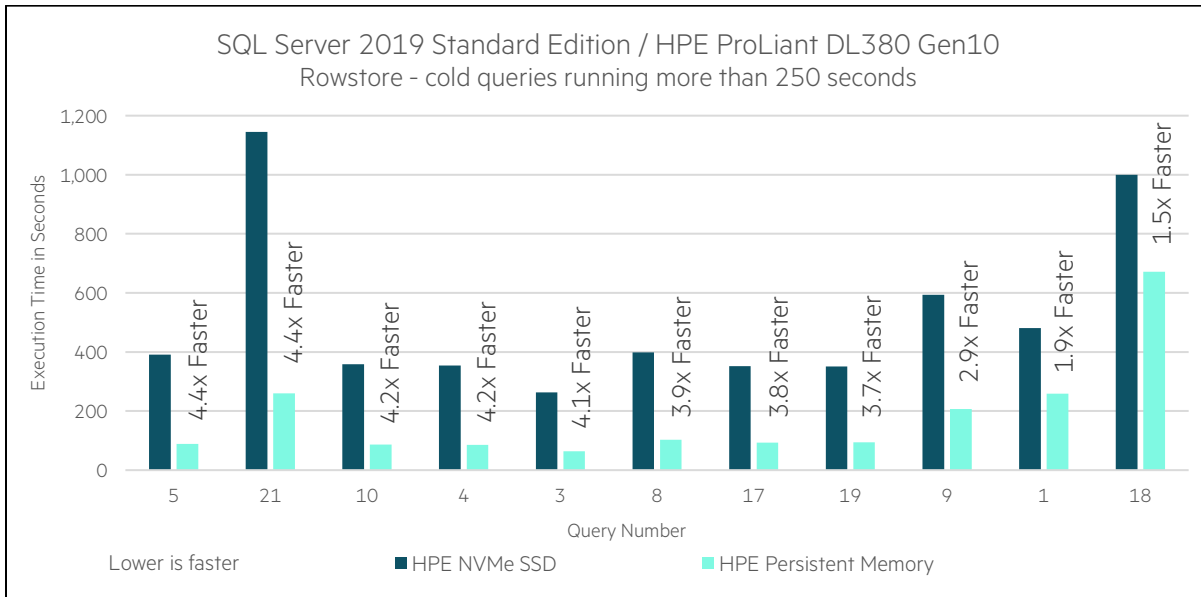


Figure 12a. Elapsed times of long running queries when run cold – Database with rowstore indices

The following chart compares the cold query performance running less than 260 seconds with the rowstore schema

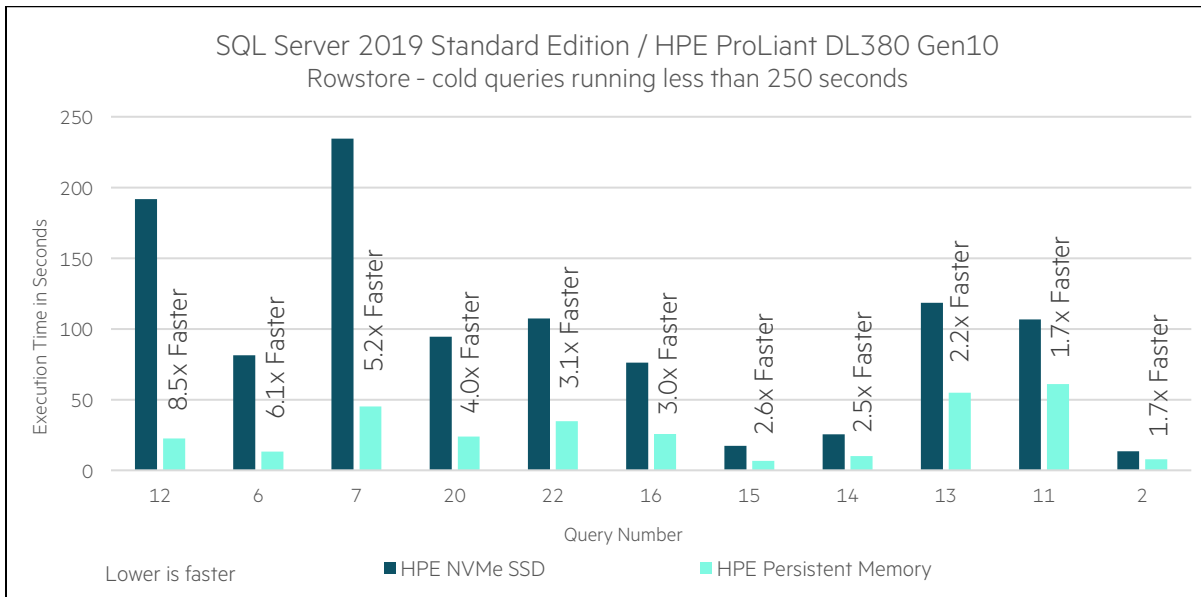


Figure 12b. Elapsed times of short running queries when run cold – Database with rowstore indices



The following chart compares the warm query performance running longer than 300 seconds with the rowstore schema.

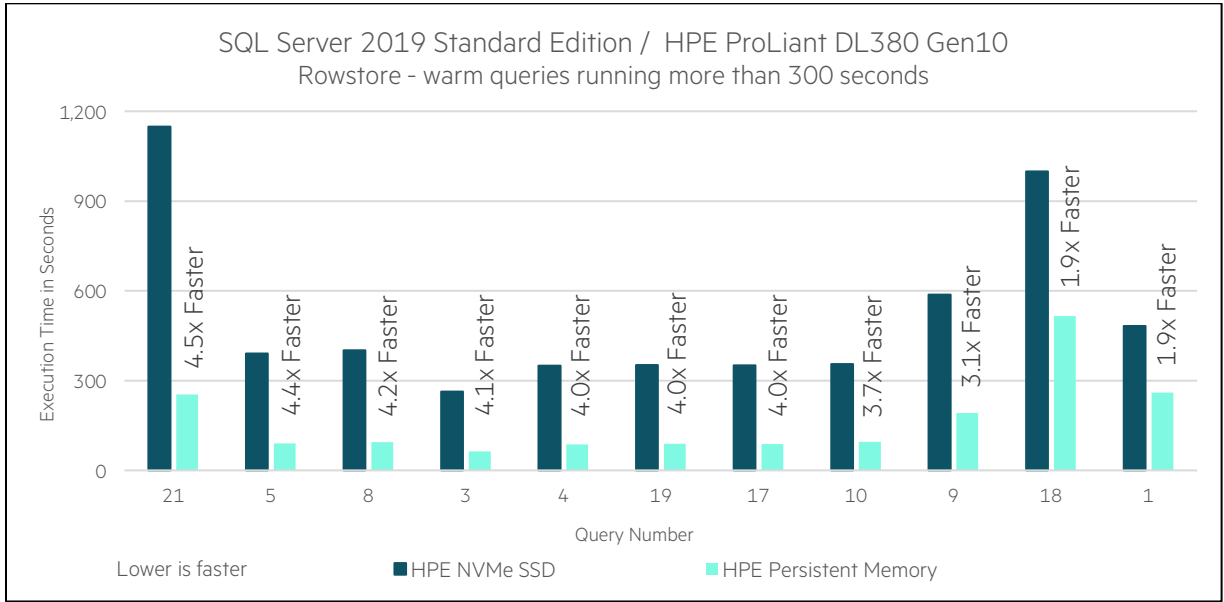


Figure 13a. Elapsed times of long running queries when run warm – Database with rowstore indices

The following chart compares the warm query performance running less than 300 seconds with the rowstore schema.

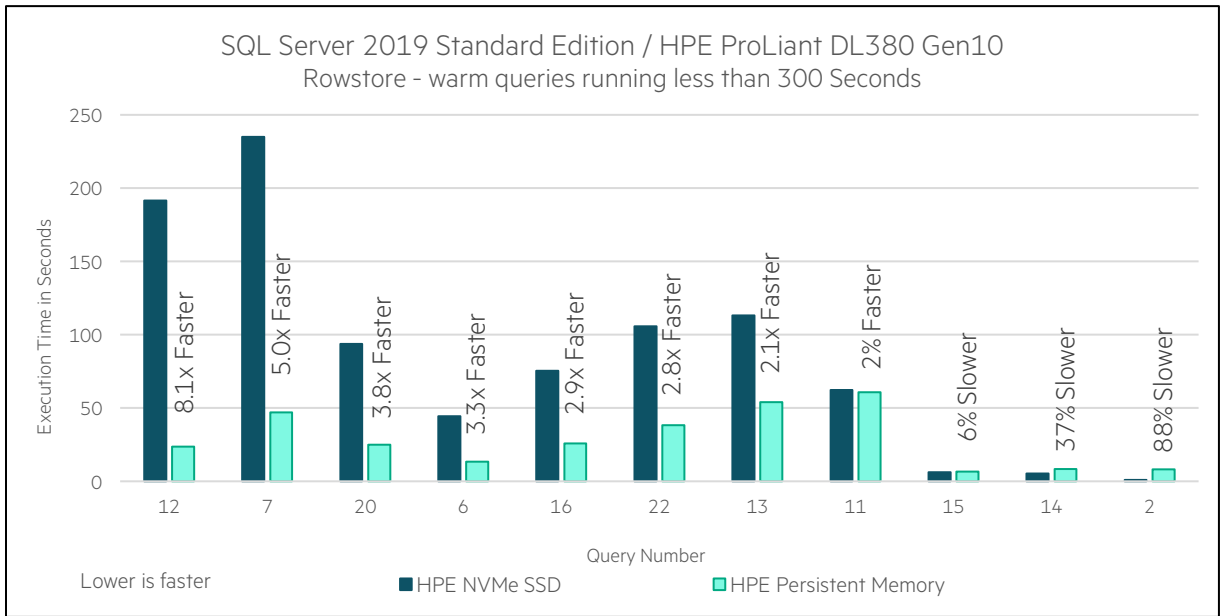


Figure 13b. Elapsed times of short running queries when run warm – Database with rowstore indices



Version History

Project: HPE Reference Architecture for Microsoft SQL Server 2019 Standard Edition on an HPE ProLiant DL380 Gen10 with HPE Persistent Memory

Status: Published

Document version	Date	Description of change
1.0	11/02/2019	Initial publication



Resources and additional links

HPE Persistent Memory, hpe.com/info/persistentmemory

HPE Persistent Memory User Guide for HPE ProLiant Gen10 servers and HPE Synergy, https://support.hpe.com/hpsc/doc/public/display?docId=a00074717en_us

Server memory and persistent memory population rules for HPE Gen10 servers with Intel Xeon Scalable processors, <https://h20195.www2.hpe.com/v2/GetDocument.aspx?docname=a00017079enw>

HPE Reference Architectures, hpe.com/info/ra

HPE Servers, hpe.com/servers

HPE Storage, hpe.com/storage

HPE Networking, hpe.com/networking

HPE Technology Consulting Services, hpe.com/us/en/services/consulting.html

To help us improve our documents, please provide feedback at hpe.com/contact/feedback.

Learn more at

<https://www.hpe.com/us/en/servers/persistent-memory.html>

© Copyright 2019 Hewlett Packard Enterprise Development LP. The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

Red Hat is a registered trademark of Red Hat, Inc. in the United States and other countries. Linux is the registered trademark of Linus Torvalds in the U.S. and other countries. Intel, Xeon, and Optane are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Microsoft, Windows, and Windows Server are registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All third-party trademarks are the property of their respective owners.

a5000414enw, version 1.0, November 2019

