



Hewlett Packard
Enterprise

HPE Reference Architecture for Microsoft SQL Server 2017 on Red Hat Enterprise Linux with HPE ProLiant DL560 Gen10

Contents

Executive summary.....	3
Introduction.....	3
Microsoft SQL Server 2017.....	3
HPE ProLiant DL560 Gen10 server.....	5
HPE Service Guard for Linux (SGLX).....	7
Solution overview.....	7
Solution components.....	9
Hardware.....	9
Software.....	10
Application software.....	10
Best practices and configuration guidance for the solution.....	10
HPE ProLiant DL560 Gen10 server.....	10
SQL Server Configuration guidance.....	10
Red Hat 7.3 Enterprise.....	11
Capacity and sizing.....	11
Workload description and test methodology.....	11
SQL OLTP workload test.....	13
SQL backup tests.....	15
SQL index rebuild tests.....	16
Analysis and recommendations.....	17
Summary.....	18
Appendix A: Bill of materials.....	18
Appendix B: Configuration adjustments.....	20
SQL Server 2017 Installation and Configuration.....	20
Appendix C: Monitoring batch requests per second in Linux.....	21
Resources and additional links.....	22

Executive summary

The demands of database implementations continue to escalate. Faster transaction processing speeds, scalable capacity, high availability and increased flexibility are required to meet the needs of today's business. This Reference Architecture is intended to provide customers with the expected performance implications associated with deploying Microsoft® SQL Server 2017 on Linux® with HPE ProLiant DL560 Gen10 server. This will help customers plan to provide the appropriate level of performance and continue to meet SLAs that may be a requirement for the business enterprises.

This Reference Architecture is ideal for:

- Customers who are looking to improve cost/transaction on general purpose and Entry/Mid-level SQL Server database deployments.
- Customers running older versions of SQL Server (e.g. SQL 2005) on older infrastructure (x86, Itanium) and seeking better transaction performance.
- Oracle customers who wish to stay on Linux and migrate to a lower cost solution.
- Customers who are looking to standardize their OS on Linux, coexisting with other database solutions.

This Reference Architecture (RA) will introduce Microsoft SQL Server 2017 running on a Linux OS using the latest HPE ProLiant DL560 Gen10 server with localized SSD storage. This RA will demonstrate an optimized solution for a small to mid-range multi-database scale-up scenario. Performance testing on this RA configuration demonstrated that the solution scales from 25K batch request per second with a 2-socket configuration up to 53K batch requests per second with a 4-socket configuration, a 114% increase in performance in the same hardware footprint. This RA also show the performance characterization of SQL backups and index rebuilds with SQL Server on Linux. The white paper highlights and analyzes database performance improvements using specific SQL commands and switches to improve throughput and shorten the maintenance window of these important operational tasks.

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

This white paper describes testing completed in October 2017.

Document purpose: The purpose of this document is to describe a Reference Architecture, highlighting benefits and key implementation details to technical audiences.

Introduction

To meet the growing and evolving demands of your business, you need scalability, industry-leading performance, reliability, and security, Microsoft's latest SQL Server 2017 coupled with the new HPE ProLiant DL560 Gen10 server can provide your business a great solution for your database needs. Below are the key features and benefits of SQL Server 2017 and the HPE ProLiant DL560 Gen10 server.

Microsoft SQL Server 2017

SQL Server 2017 brings the best features of the Microsoft relational database engine to the open source Linux operating system. Not a rewrite or a port, SQL Server 2017 on Linux is the same Microsoft relational database management system now available on more operating systems which is made possible by the SQL Server Platform Abstraction Layer (SQLPAL).

SQLPAL is responsible for abstraction of the calls and semantics of the underlying operating system and its libraries from the software itself, which enables the same subset of Windows® libraries used by SQL Server to run on Linux. All resources in the process are managed by SQLPAL. In this new architecture, Win32 and NT APIs are based on SQLPAL so every memory allocation and thread would be controlled by SQLPAL. This approach has delivered the same reliability and performance for which SQL Server is recognized to Linux ecosystem¹.

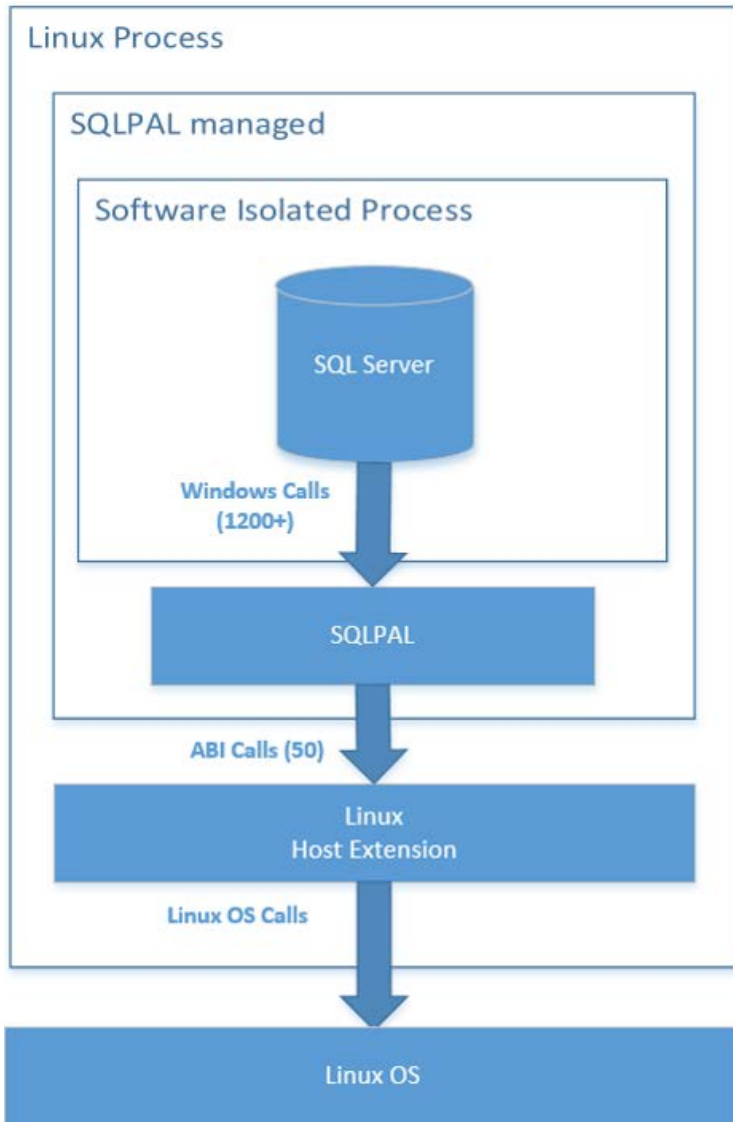


Figure 1. SQL Server 2017. SQLPAL overview

Industry-leading performance features available on Linux

- Features like Real-Time Operational Analytics, in-memory OLTP, In-memory columnstore, compression, and partitioning are available on Linux.
- SQL Server 2017 includes many new Database Engine features and performance improvements, including resumable online index rebuild, automatic database tuning, graph database capability, and adaptive query processing to mention a few. To learn more about all features, capabilities, and services available on SQL Server 2017, see Resources and additional links for Microsoft SQL Server on Linux.

¹Scott Konersmann and Tobias Ternstrom, "SQL Server on Linux: How? Introduction", at http://info.microsoft.com/rs/157-GQE-382/images/EN-US-CNTNT-Whitepaper-DBMC-SQLServerOnLinuxQuickStart_Updated.pdf, (December, 16, 2016)

Flexibility for high availability and disaster recovery on Linux

- Read-Scale Availability Groups provide load balancing of read-only workload replicas in mixed-OS environments but not high availability. This architecture does not require a cluster manager.
- Always On Failover Cluster Instance (FCI) provides instance-level high availability, allowing a SQL Server instance to remain available during planned or unplanned outages caused by hardware and software failures, or system maintenance.
- Always On Availability Groups provide database-level high availability, disaster recovery, and read-scale balancing.

Security

- Security features such as Auditing, Transparent Data Encryption, Row-Level Security, Always Encrypted, and Dynamic Data Masking, provide server-side security that greatly simplify prevent unauthorized access, without the need to modify existing client applications.
- Transparent Data Encryption protects data at rest at file level, while Always Encrypted secures data in motion and at rest.

Cost

- SQL Server has a significantly lower total cost of ownership than similar enterprise data solutions. A key factor in reducing TCO is the inclusion of features with SQL Server which comparable database solutions are either available at additional cost or simply not available. With all the availability, performance, and security features mentioned above, SQL Server can provide a fraction of the cost of the competition.

Multi-OS support

- With all these benefits and capabilities available on SQL Server 2017, organizations can now choose their deployments environment.
- SQL Server 2017 is now available on more operating systems- Red Hat® Enterprise Linux®, SUSE Linux Enterprise Server, and Ubuntu and more cloud and container platforms like Docker.

Linux native-user experience with support for existing SQL Server tools

Microsoft has focused in delivering a Linux-native user experience for SQL Server by having a package-based installation, and by having the ability to have scripted installation with environment variables and configuration files is exactly what Linux administrator wants to have for an easy installation and configuration experience with SQL Server. And what better compliments SQL Server 2017 than the HPE ProLiant DL560 Gen10 server.

HPE ProLiant DL560 Gen10 server

The HPE ProLiant DL560 Gen10 server is a high-density, 4P server with high-performance, scalability, and reliability, in a 2U chassis. Supporting the latest Intel® Xeon® Scalable processors, the new HPE ProLiant DL560 Gen10 server offers greater processing power, up to 6TB of DDR4 memory, IO of up to eight PCIe 3.0 slots, plus the intelligence and simplicity of automated management with HPE OneView and HPE iLO 5.

Figure 2 shows the HPE ProLiant DL560 Gen10 server with the new front-side iLO Service port to access the server directly. See the HPE ProLiant DL560 Gen10 server QuickSpecs for more details at <https://www.hpe.com/h20195/v2/gethtml.aspx?docname=a00008181enw>



Figure 2. HPE ProLiant DL560 Gen10 server with 24 local SSDs

Highlights of the new HPE ProLiant DL560 Gen10 server

Scalable 2P to 4P performance in a dense 2U form factor

- The HPE ProLiant DL560 Gen10 server starts with 2P computing that supports the latest Intel Xeon Scalable processors which are available in two configurations; Gold (up to 22 CPU cores), or Platinum (up to 28 CPU cores). At a later time, upgrade to a 4P computing which provide up to 68% more processor performance and 27% more cores than the previous generation².
- The HPE ProLiant DL560 Gen10 server supports up to 48 DIMM slots which support up to 6TB for 2666 MT/s DDR4 SmartMemory. HPE DDR4 SmartMemory improves workload performance and power efficiency while reducing data loss and downtime with enhanced error handling.
- Supports up to 24 slots for NVDIMMs, providing up to 384GB of persistent memory.
- Choice of HPE FlexibleLOM adapters offers a range of networking bandwidth (1GbE to 25GbE) and fabric offers adaptable, scalable networking to change with business needs.
- HPE Innovation with Intelligent System Tuning enhances workload performance using customized profiles to tune internal resources; with improved throughput for all workloads including latency-sensitive workloads such as high frequency trading with jitter smoothing. To find out more about Intelligent System Tuning (IST), read the Workload-based Performance and Tuning Guide for Gen10 servers and HPE Synergy at https://support.hpe.com/hpsc/doc/public/display?docId=a00016408en_us.

Flexible new generation expandability and reliability for multiple workloads

- The HPE ProLiant DL560 Gen10 server has a flexible processor tray allowing to scale up from two to four processors only when you need, saving on upfront costs and the flexible drive cage design supports up to 24 SFF SAS/SATA with a maximum of 12 NVMe drives.
- Supports up to eight PCIe 3.0 expansion slots for other Smart Array controllers, graphical processing units (GPUs), networking cards, and other IO devices offering increased I/O bandwidth and expandability.
- Up to four, HPE 800W or 1600W Flex Slot Power Supplies enable higher power redundant configurations and flexible voltage ranges. The slots provide the capability to trade-off between 2+2 power supplies or use as extra PCIe slots.

Secure and reliable

- HPE offers industry standard servers with HPE designed custom BMC (HPE iLO5) silicon chip that is physically altered to anchor most server essential firmware including HPE iLO, UEFI, IE, ME, and SPLD. This anchor into the HPE silicon chip creates a true Silicon Root of Trust that protects the server from firmware attacks, as well as the ability to recover the essential server firmware automatically.
- Millions of lines of firmware code run before server operating system boots and with Runtime Firmware Validation, enabled by HPE iLO Advanced Premium Security Edition, the server firmware is checked every 24 hours verifying validity and credibility of essential system firmware.
- Secure Recovery allows server firmware to roll back to the last known good state or factory settings after detection of compromised code.
- Additional security options are available with, Trusted Platform Module (TPM), to prevent unauthorized access to the server and securely stores artifacts used to authenticate the server platforms while the Intrusion Detection Kit logs and alerts when the server hood is removed.

Agile infrastructure management for accelerating IT service delivery

- With the HPE ProLiant DL560 Gen10 server, HPE OneView provides infrastructure management for automation simplicity across servers, storage and networking.
- A suite of embedded and downloadable tools is available for server lifecycle management including Unified Extensible Firmware Interface (UEFI), Intelligent Provisioning; HPE iLO 5 to monitor and manage; HPE iLO Amplifier Pack, Smart Update Manager (SUM), and Service Pack for ProLiant (SPP).

² HPE measurements. Up to 68% performance increase of Intel Xeon Platinum versus previous generation E5-4600 v4 average performance based on generational gains on HPE servers comparing 4-socket Intel Xeon Platinum 8180 to E5-4699 v4 Family processors. Any difference in system hardware, software design, or configuration may affect actual performance. May 2017

HPE Service Guard for Linux (SGLX)

SQL Server 2017 supports the following high availability and disaster recovery deployment models:

- Failover Cluster Instance (FCI). Based on shared storage architecture, an FCI contains two or more cluster nodes with only one node active at a time and secondary node(s) are available but passive, ready to an active role during failover.
- Always on Availability Groups (AG). In this architecture, primary and standby databases are configured with SQL Server Always on Availability Groups. The databases can be located on the same premises or in geographically dispersed data centers and data can be replicated by SQL Server.

In order to achieve high availability and disaster recovery, both of these architectures would require a cluster manager to perform various critical tasks like monitoring the health of databases, server, network, storages, virtualization layer, virtual machine guests, OS and take automatic actions to minimize application downtimes. HPE Serviceguard for Linux (SGLX) introduces the support of SQL Server to provide mission critical class of robustness to SQL Server deployments and ensures robust monitoring, reliable actions, and protection against data loss. This market-leading high availability and disaster recovery clustering solution protects your applications from a multitude of infrastructure and application faults across physical or virtual environments over any distance. It reduces the impact of unplanned downtime without compromising data integrity and performance, and helps you achieve near zero planned downtime for maintenance.

When SGLX detects a failure in case of Failover Cluster Instance deployments, it follows shutdown procedures, and recovers the database by restarting SQL Server on the adoptive cluster node. In case of (AG) deployments, SGLX will also monitor and administer the replication between primary and standby databases. In case of failures SGLX will perform automatic role management to recover from failures by promotion of the standby database instance to primary. With Availability Groups support, SGLX provides database level protection with much faster recovery times. See Resources and additional links for more information about HPE Serviceguard for Linux.

Solution overview

Customers constantly deal with the element of choice concerning their data. On premises, public cloud, or private cloud? What language to use to write applications? What platforms to run the apps from? Microsoft's latest SQL Server 2017 now provides customers the option to choose which language to use to write their applications, plus which platform makes more sense for their business. Whether it be Windows, MacOS, or Linux, SQL Server 2017 brings industry-leading performance, real-time intelligence, least vulnerability, and low TCO to the platform of choice.

Many businesses can take advantage of multiple platforms, meaning data and applications can easily migrate across Windows and Linux by using the recommended method of SQL Server backup and restore. In addition, Always On Availability Group configured across the platforms provides high availability and fail over to avoid costly downtime.

Migration from other database servers are also supported for SQL on Linux. Business can migrate databases from other databases systems, such as Microsoft Access, DB2, MySQL, Oracle, and Sybase to SQL Server 2017 on Linux. Migration automation from these databases can be accomplished through SQL Server Management Assistant (SSMA).

HPE ProLiant DL560 Gen10 server is a highly-dense scale-up server for compute-intensive workload. A great platform for a large memory footprint in a 2U form factor, for in-memory application such as SQL. Customers can start with a 2-socket configuration then later add CPUs and memory to scale up as business needs grow without adding another hardware footprint.

This Reference Architecture explores online transaction processing (OLTP) use cases with SQL Server 2017 on Red Hat Linux Enterprise with HPE ProLiant DL560 Gen10 server and compares database performances between a 2-socket and a scale-up 4-socket configuration. The configuration of the HPE ProLiant DL560 Gen10 server in this RA consists of four RAID sets on local SSD drives that will be used for the Linux O/S and SQL Server database. While Microsoft SQL Server 2017 runs on Linux platform, it either can be managed by a SQL Server Management Studio v17.3 (SSMS) running on a Windows Server or the new mssql-tools package that is available for Linux platform. A Windows Server that hosts SSMS is also used as a remote SSH client to access the RHEL machine for configuration and performance metrics purposes. Figure 3 shows a diagram of the components involved.

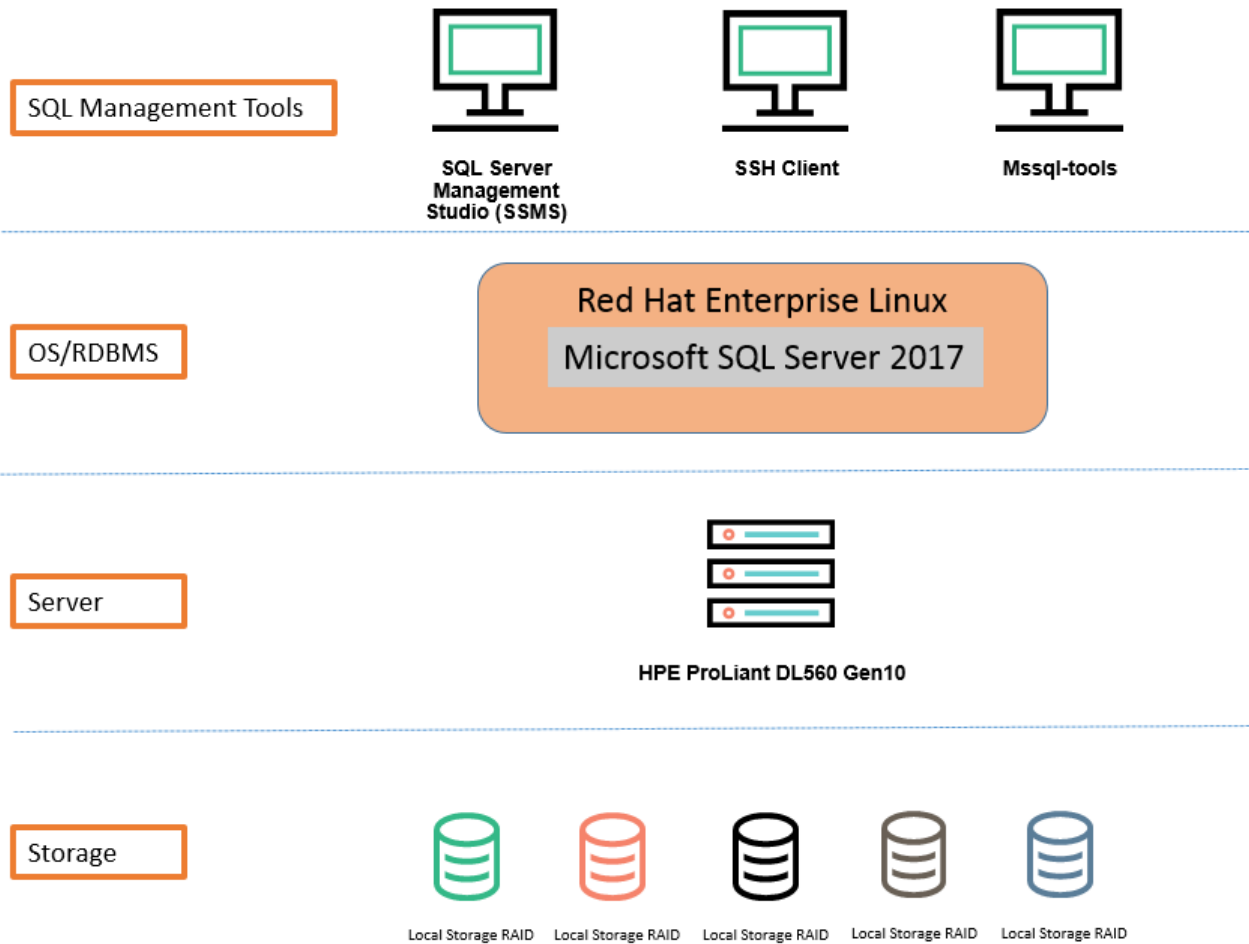


Figure 3. Solution diagram showing SQL Server 2017 on Linux using HPE ProLiant DL560 Gen10 server

Solution components

Hardware

HPE ProLiant DL560 Gen10 server configuration

Table 1 shows the server hardware configuration for the 2-socket and 4-socket hardware model that was used for testing. Based on availability, the 28-core Platinum series Intel Xeon processors were used in this testing; however, other processor series at lower core-count can be used to reduce up-front hardware and software licensing cost. The management network configuration was based on 10Gb Ethernet. Local storage consisted of 400GB/800GB/1.6TB SSDs which provided the media for RHEL and SQL Server. With the scale-up 4-socket configuration, two CPUs, and corresponding RAM were added to double the CPU cores and physical memory. Four SSDs were added to increase the SQL database size from 1TB to 2TB.

Note

Based on availability, 800GB 6G SATA and 1.6TB 12G SAS Mixed-Use SSDs were used for our testing. To achieve higher performance, use Write-Intensive 12G SAS SSDs or NVMe drives for your solution. See Appendix A: Bill of materials for a sample bill of material (BOM) with the latest Gen10 storage equivalent. For the latest list of Gen10-supported storage, review the HPE ProLiant DL560 Gen10 QuickSpecs at: <https://www.hpe.com/h20195/v2/GetDocument.aspx?docname=a00008181enw>.

Table 1. Server Components. Two hardware configuration used for testing: 2-socket and a 4-socket server configuration

	HPE ProLiant DL560 Gen10 (2-socket)	HPE ProLiant DL560 Gen10 (4-socket)
Processor	Two 28-core Intel® Xeon® Platinum 8176 processors at 2.10GHz	Four 28-core Intel® Xeon® Platinum 8176 processors at 2.10GHz
Memory	768GB memory (12 x 64GB HPE DDR4 SmartMemory LRDIMMs)	1.5TB memory (24 x 64GB HPE DDR4 SmartMemory LRDIMMs)
Ethernet	HPE Ethernet 10Gb 2-port 560FLR-SFP+ Adapter	HPE Ethernet 10Gb 2-port 560FLR-SFP+ Adapter
Local Drive	2 x 400GB 12G SAS MU 2.5in SSD	2 x 400GB 12G SAS MU 2.5in SSD
Local Drive	6 x 800GB 6GB SATA WI 2.5in SSD	10 x 800GB 6GB SATA WI 2.5in SSD
Local Drive	4 x 1.6TB 12G SAS MU 2.5 in SSD	4 x 1.6TB 12G SAS MU 2.5in SSD

Table 2 shows how the local disks are provisioned and mounted on RHEL. /DATA01, /DATA02, /LOGS, and /BACKUP will be used for SQL databases and backups.

Table 2. Storage Provisioning. Two hardware configuration used for testing: 2-socket and a 4-socket server configuration

Local Storage Devices	RAID Configuration	Device mapping	OS Mount Point	Purpose
2 x 400GB 12G SAS MU 2.5in SSD	RAID1	/dev/sda	RHEL default mount points	RHEL O/S related partitions
4 x 1.6TB 12G SAS MU 2.5in SSD	RAID10	/dev/sdb	/LOGS	SQL default log repository
4 x 800GB 6GB SATA 2.5in SSD	RAID10	/dev/sdc	/DATA01	SQL default data repository
4 x 800GB 6GB SATA 2.5in SSD (4-socket testing only)	RAID10	/dev/sdd	/DATA02	SQL data repository
2 x 800GB 6GB SATA 2.5in SSD	RAID1	/dev/sde	/BACKUP	SQL default backup repository

HPE ProLiant DL560 Gen10 BIOS configuration

New to HPE ProLiant Gen10 servers, Workload Profiles are configuration options to deploy BIOS settings based on the workload customer intends to run on the server. Workload Profiles are a configuration option to deploy BIOS settings to accommodate the intended application of the server. HPE ProLiant Gen10 servers provide 12 unique workload profiles, plus a Custom option which will disable Workload Profiles. Use this option to set specific BIOS options for your deployment manually. When you select Custom, all the settings for the previously selected profile are carried forward. You can edit all or some of the options. Most important, Custom is not a profile and settings that you specify are not saved as a template.

The workload profile chosen for this solution was Transactional Application Processing. This profile is intended to be used for business processing environments, such as online transaction processing (OLTP) applications that require a database back-end. For example, workloads typically comprised of a high number of user-based, transactional applications running on a single server with cohosted database component. The profile balances the requirement of managing both peak frequency and throughput. For more details on Transactional Application Processing profile and its dependencies, see the Resources and additional links at the end of this white paper.

Software

- Red Hat Linux 7.3 Enterprise – build 3.10.0-514.el7.x86_64 or above
- Microsoft SQL Server 2017 RTM – 14.0.1000.169

Application software

The following management and support software was used for the solution:

- Microsoft SQL Server Management Studio (SSMS) 17.3; 17.0+ is required from a Windows Server® machine
- mssql-conf was used to manage and configure Microsoft SQL Server 2017 within Linux
- SSH Terminal such as PuTTY from a Windows Server machine

Best practices and configuration guidance for the solution

HPE ProLiant DL560 Gen10 server

Initial setup for the HPE ProLiant DL560 Gen10 server consisted of various BIOS and SQL settings. The following BIOS settings were modified from default:

- Workload Profile – Transactional Application Processing
- Hyper-Threading – Disabled
- Virtualization Technology – Disabled
- Intel VT-d – Disabled
- SR-IOV – Disabled

SQL Server Configuration guidance

- T834 – Use large pages
- disable stats collection
- Max server memory = 80% of 1.5TB
- Min server memory =80% of 1.5TB
- Network packet size = 4096
- Lightweight pooling = 0
- Max worker threads = 0
- Max degree of parallelism = 1
- SQL PAL was configured to use 93% of available memory through BASH environment variable \$PAL_MEMORY_SIZE

- Soft-NUMA disabled
- Multiple tempdb data files created after installation
- Resource Governor enabled for CPU affinity

Red Hat 7.3 Enterprise

- Transparent_hugepage - enabled for use with T834
- Tuned-adm profile – throughput-performance

Capacity and sizing

Workload description and test methodology

An OLTP database and its associated workload engines that simulated a brokerage firm involving stock trades, market orders and reporting provided the workload. Test configuration comprised of 4 databases at 500GB scale factor and each of these databases' data spread across 8 data files sized at 62.6GB each. Using HPE Smart Storage Administration utility, we carved independent LUNs for use with each database.

As shown on Table 3 and 4, each database had their own and separate OS mount points for data files and transaction log. Data files were hosted on a LUN that performed better with random reads and random mix of reads and writes, whereas the transaction log is hosted on a LUN that performed better with sequential writes.

Table 3. 2-socket server configuration. TPCE Database file locations for OLTP TPCE databases

Database Name	Database Size	Data files location	Log file location
TPCE_01	500GB	/data01/db_01	/logs/db_01
TPCE_02	500GB	/data01/db_02	/logs/db_02

Table 4. 4-socket server configuration. TPCE Database file locations for OLTP TPCE databases

Database Name	Database Size	Data files location	Log file location
TPCE_01	500GB	/data01/db_01	/logs/db_01
TPCE_02	500GB	/data01/db_02	/logs/db_02
TPCE_03	500GB	/data02/db_03	/logs/db_03
TPCE_04	500GB	/data02/db_04	/logs/db_04

SQL Server Resource Governor was enabled and configured such that each database had its CPU/Memory affinity isolated at CPU socket level. In other words, each database had its affinity set to use resources from a dedicated NUMA node on the server. For the most part the default resource pool settings were used to create the resource pools, except the AFFINITY NUMANODE value was uniquely set for each database. For the OLTP workload groups, in order to have consistent and reproducible results across multiple workload runs, we set MAXDOP value to 1. Unlike DSS/Warehouse queries, OLTP transactions do not yield a performance benefit with increased parallelism.

The following tables show how the Resource Governor was configured to funnel workload sessions with unique logins to use the corresponding workload group, which in turn, used the corresponding resource pool. With 2P, only two users were used for access TPCE_01 and TPCE_02 database workloads. A script was created for ease of configuration. The same script was used for the 2P and 4P testing. For sample scripts to configure the Resource Governor, see Appendix C: Monitoring batch requests per second in Linux, for more details.

Table 5. 2-socket Resource Governor Classifier Group logins with corresponding workload and resource group

SUSER_NAME	Workload Group	Resource Group
DB01User	gDB01QueryStream	pDB01Queries
DB02User	gDB02QueryStream	pDB02Queries

Table 6. 2-socket Resource Governor Resource Pool Affinity and Workload Group MAXDOP values

Resource Pools	Affinity NUMANODE	Workload groups	Degree of Parallelism
pDB01Queries	0	gDB01QueryStream	1
pDB02Queries	1	gDB02QueryStream	1

Table 7. 4-socket Resource Governor Classifier Group logins with corresponding workload and resource group

SUSER_NAME	Workload Group	Resource Group
DB01User	gDB01QueryStream	pDB01Queries
DB02User	gDB02QueryStream	pDB02Queries
DB03User	gDB03QueryStream	pDB03Queries
DB04User	gDB04QueryStream	pDB04Queries

Table 8. 4-socket Resource Governor Resource Pool Affinity and Workload Group MAXDOP values

Resource Pools	Affinity NUMANODE	Workload groups	Degree of Parallelism
pDB01Queries	0	gDB01QueryStream	1
pDB02Queries	1	gDB02QueryStream	1
pDB03Queries	2	gDB03QueryStream	1
pDB04Queries	3	gDB04QueryStream	1

With a physical memory of 768GB on 2-socket and 1.5TB on 4-socket configuration, the SQL Max Memory was set at the default 80% of physical memory for our workload. Once the SQL databases were laid-out optimally on the local storage and the SQL Max Memory was set, our OLTP SQL environment was ready for SQL OLTP workload performance testing.

Each test started with freshly restored OLTP and tempDB databases in order to have a common start point. After total CPU utilization reached at about 80% and batch requests have been steady for at least 30 minutes, measurements were taken in five-minute intervals for 30 minutes. Metrics were collected using various Linux tools such as SAR and vmstat to capture data IO and log bandwidth, and CPU utilization. Batch request measurements were collected by running a select statement query using sys.dm_os_performance_counters at 5-minute intervals. A script was written to take the two measurements to calculate the batch request per second in a 5-minute timeframe for the period of time when the workload was running. See Appendix C: Monitoring batch requests per second in Linux for a sample script.

SQL OLTP workload test

The workload performance test started with a 2P/2DB configuration. A baseline test with a workload drive point of 100 users per database was used to achieve an overall system CPU utilization at about 80%. The baseline consisted of the following BIOS and SQL Server settings:

- Workload Profile – Transactional Application Processing
- Hyper-Threading disabled
- No hardware NUMA / CPU affinity testing
- SQL Server – Soft-NUMA disabled
- Trace Flag T834
- Resource Governor disabled

Once the baseline testing was complete, the next workload test was with CPU affinity with Resource Governor enabled. Figure 4 depicts the 2P/2DB performance testing. SQL Server with CPU affinity databases via the Resource Governor resulted in a 9.06% increase in batch requests per second than our baseline test.

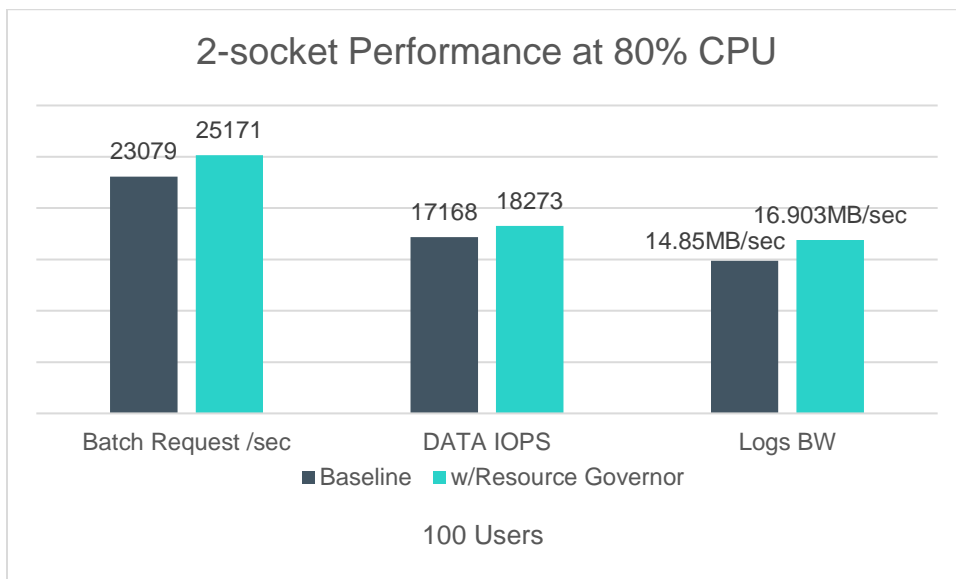


Figure 4. 2-socket performance comparison between Baseline and with Resource Governor

After both tests were complete then we moved to the 4-socket testing with added matching CPUs and physical memory on the two empty sockets, then restored the 3rd and 4th database from backup. The 4-socket testing mimicked the same baseline and Resource Governor tests as with the 2-socket configuration. Figure 5 depicts the 4P/4DB performance testing, result showed a 6.76% increase in batch requests per second with using Resource Governor for workload CPU affinity than the baseline.

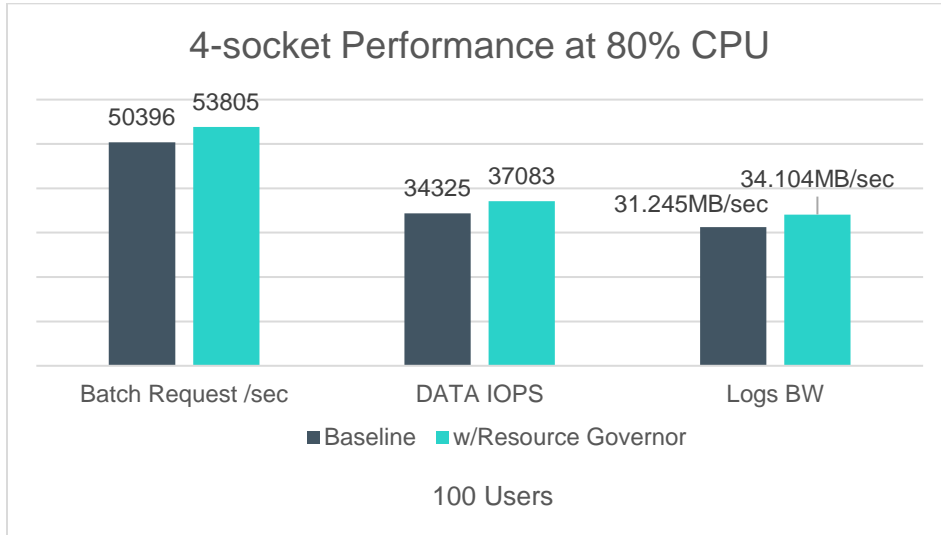


Figure 5. 4-socket performance comparison between Baseline and with Resource Governor

One great benefit of the HPE ProLiant DL560 Gen10 server is the ability to improve server performance and capacity by scaling up computing power and storage at a lesser cost than purchasing a whole new hardware footprint. Figure 6 illustrates 114% performance improvement from 2P to 4P configuration with doubled database capacity at only 40% cost increase.

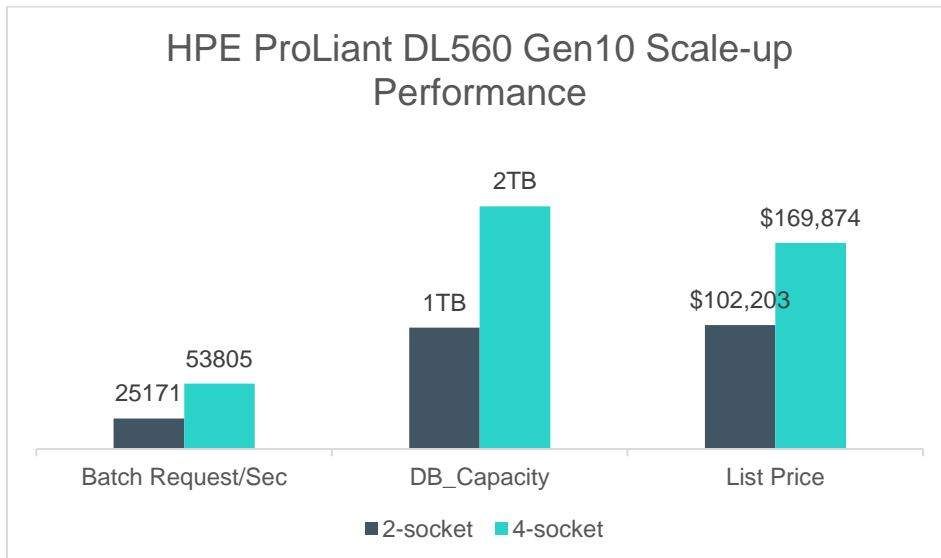


Figure 6. 2-socket to 4-socket scale-up performance, capacity, and cost comparison. Pricing calculated with 2P/4P BOM found in Appendix A: Bill of materials³

³ Pricing is at list price and does not include any discounts.

SQL backup tests

In addition to evaluating our primary database transactional workload and identifying optimum system settings for performance with SQL on Linux, the following backup tests are intended to show the system's ability handle a secondary surge workload with no or minimal interference to the performance of the primary workload. Performing backup in Linux environment can either be done in T-SQL using SSMS or SQLCMD.

SSMS GUI-invoked backups do not reveal certain backup command options that are only available using command line scripting. Running backups in Transact-SQL unlocks backup switch options that can allow businesses to perform offline backups at faster speeds, or online backups at a reduced SQL performance during the regular overnight maintenance periods.

Our backup test plan was based on a 500GB OLTP database; we ran backups in two sets with three different scenarios.

Table 9. Backup Test Plan Scenarios with / without compression switch options

Backup Test Plan Scenario	Compression	Blocksize	Maxtransfersize	Buffercount
Database size=500GB				
No Load - Backup with default settings	No	-	-	-
No Load - Backup with default compression settings	yes	-	-	-
No Load - Backup with compression and additional switch settings	yes	65536	4194304	300
With Load - Backup with default compression setting	no	-	-	-
With Load - Backup with default compression setting	yes	-	-	-
With Load - Backup with compression and additional switch settings	yes	65536	4194304	300

Table 10 shows the no load test with 2-socket and 4-socket configuration respectively. Depending on 2 or 4-socket configuration, our test resulted in one key point:

- The rate of speed of the backup was 3.2 times faster when using compression with additional options which reduced the duration of backup from 16 minutes to just under 5 minutes for a physical database size of 500GB.

Table 10. 2 & 4-sockets with no concurrent load – Compression / No compression backup test results

Backup performed with no active OLTP workload	# of sockets	MB/sec	% Difference	Duration (sec)	% Difference
Database size=500GB					
Backup with default settings (baseline)	2	406MB	--	924	--
Backup with default compression settings	2	319MB	-21%	1176	27%
Backup with compression and additional switch settings	2	1306MB	222%	287	-69%
Backup with default settings (baseline)	4	412MB	--	910	--
Backup with default compression settings	4	281MB	-32%	1336	47%
Backup with compression and additional switch settings	4	1282MB	211%	293	-68%

In Tables 11 and 12 with a workload running, depending on 2 or 4-socket configuration, our test resulted in these key points:

- The rate of speed of the backup increased up to 3.7 times faster from when using compression with additional options which reduced the time spent on backup from under 22 minutes to just under 6 minutes for a physical database size of 500GB
- Most important, performance on the running workload decreased by 25-42% for less than 6 minutes during the backup

Table 11. 2 & 4 socket backup testing with concurrent workload – Workload Performance

Backup performed during active OLTP workload	# of sockets	BRPS before backup	BRPS during backup	% Diff
Database size=500GB				
Backup with default settings (baseline)	2	24359	24078	-1%
Backup with default compression settings	2	23842	23088	-3%
Backup with compression and additional switch settings	2	24043	14026	-42%
Backup with default settings (baseline)	4	54124	52721	-3%
Backup with default compression settings	4	54382	52452	-4%
Backup with compression and additional switch settings	4	54868	41110	-25%

Table 12. 2 & 4 socket backup testing with concurrent workload – Backup Speed and Duration

Backup performed during active OLTP workload	# of sockets	Bandwidth (MB/Sec)	% Diff	Duration (Seconds)	% Diff
Database size=500GB					
Backup with default settings (baseline)	2	295MB	--	1294	--
Backup with default compression settings	2	225MB	-24%	1694	31%
Backup with compression and additional switch settings	2	1094MB	271%	344	-73%
Backup with default settings (baseline)	4	295MB	--	1286	--
Backup with default compression settings	4	240MB	-19%	1591	24%
Backup with compression and additional switch settings	4	1058MB	259%	357	-72%

SQL index rebuild tests

The max degree of parallelism (MAXDOP) option determines the maximum number of processors to use during an index rebuild operation. Using MAXDOP with the default value of zero (0), the server determines the number of CPUs that are used for the index operation, using only the actual available number of processors or fewer based on the current system workload. However, you can manually configure the number of processors used to run index operations by specifying the number of processors. By doing so, performance may be impacted positively or negatively during online index rebuild. To find the optimal value for our workload, we measured workload performance during concurrent online index rebuilds with varying MAXDOP values.

Our index rebuild test plan consists of running several degree of parallelism scenarios that will measure the performance of the OLTP workload and the duration of the indexing during online index rebuilds. We used one single OLTP database, with an un-partitioned index with 777,595,200 rows by default, 11,442,956 data pages, and 89.441GB in size. At first, we measured the index rebuild performance when the workload is off, with and without degree of parallelism, finding the optimal MAXDOP setting that will yield the best rebuild completion time. The first performance test was done with no max degree of parallelism, with MAXDOP set to 1. The next tests were done with 48, 32, 24, 16, 8, and at the default MAXDOP of 0. The table below shows that exploring various degree of parallelism values can definitely reduce the duration of the index rebuild. Adjusting MAXDOP value from default (zero) to 16 will reduce the time from approximately 26 minutes to 10 minutes, 2.6 times faster in rebuild duration time. Without any load on the system, changing the MAXDOP index option to 16 is recommended during online index rebuild for our workload.

Table 13. Performance results during index rebuild on a single database with workload off

MAXDOP	Duration (hh:mm:ss)	% Diff from default (0)
1	00:35:12	34%
48	00:25:08	-4%
32	00:17:39	-32%
24	00:22:21	-14%
16	00:10:09	-61%
8	00:24:09	-7%
0	00:26:06	

To simulate an off-hour index rebuild maintenance period, our next index rebuild tests were run with the workload running at 25% CPU utilization. Tests were run with the default MAXDOP value of zero then at 16. With results at hand, the table below shows that adjusting the index MAXDOP value to 16 will yield the best performance with an average decrease of 10% in batch requests per second while the index rebuild only took 31 minutes and 50 seconds. Using MAXDOP with the default value of zero (0), the server determines the number of CPUs that are used for the index operation, using only the actual available number of processors or fewer based on the current system workload; meaning, over-subscription of CPUs can cause insufficient resources for other applications and database operations for the duration of the index operation, thus performance will decline.

Table 14 . Performance results during online index rebuild on a single database with workload on at 25% CPU utilization

Index rebuild on 89.4GB index table	BRPS before rebuild	BRPS (avg) during rebuild	% Diff	BRPS (min) during rebuild	% Diff	Duration (hh:mm:ss)	% Diff
MAXDOP=0	7633	5655	-26%	4205	-45%	00:49:44	
MAXDOP=16	7960	7185	-10%	6747	-15%	00:31:50	-36%

Analysis and recommendations

- Many SQL Server best practices and configuration guidelines in Windows are available in SQL Server 2017 on Linux.
- Using CPU affinity can improve performance on your workload. With roughly 6-9% performance improvement on our workload, the use of Resource Governor enabled each database instance to be assigned to its own NUMA nodes, resulting in more efficient, local access to IO and memory.
- With HPE ProLiant DL560 Gen10 server, scaling from 2-socket to 4-socket configuration showed a 114% increase in performance.
- MAXDOP settings and compression are a viable option for backups under moderate load, reducing backup storage while running within a reasonable maintenance window with minimum impact to the workload.
- With our workload, online index rebuild showed a 36% improvement in time spent when index MAXDOP value was optimally adjusted over the default value

Summary

As customer look for database alternatives in Linux ecosystem with the latest hardware out in the market, the Microsoft SQL Server 2017 and the HPE ProLiant DL560 Gen10 server offers a great solution in providing faster transaction processing, scalability and increased flexibility.

- HPE ProLiant DL560 Gen10 server offers a powerful and scalable platform for the Microsoft SQL 2017 Server on Linux. Multiple CPU and storage options, along with a large number of PCI expansion slots provide flexibility needed to deploy high performance and scalable SQL Server deployments.
- From the entry-level 2-socket configuration to a 4-socket scale-up model, the HPE ProLiant DL560 Gen10 server improved OLTP performance by a factor of two while doubling the SQL database size capacity at a lower cost of upgrade compared to purchasing a new comparable hardware.
- Tuning backup and index operations can greatly improve performance of the operations, but those increases consume resources at a higher rate and must be balanced with concurrent production workload requirements.
- The backup and index rebuild concurrent with workload do reduce workload performance as expected. This can be mitigated by scheduling these operations during off hours and using Resource Governor if needed to protect resources from over-utilization.

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 15. Bill of materials for the HPE ProLiant DL560 Gen10 server with 2-sockets

Qty	Part number	Description
HPE ProLiant DL560 Gen10 server		
1	841730-B21	HPE DL560 Gen10 CTO Server
1	840379-L21	HPE DL560 Gen10 Intel Xeon-Platinum 8176 (2.1GHz/28-core/165W) FIO Processor Kit
1	840379-B21	HPE DL560 Gen10 Intel Xeon-Platinum 8176 (2.1GHz/28-core/165W) Processor Kit
12	815101-B21	HPE 64GB (1x64GB) Quad Rank x4 DDR4-2666 CAS-19-19-19 Load Reduced Memory Kit
1	873444-B21	HPE DL5x0 Gen10 12Gb SAS Expander Kit / Cables
1	872233-B21	HPE DL560 Gen10 8SFF Bay1 Kit
1	872235-B21	HPE DL560 Gen10 8SFF Bay2 Kit
2	873359-B21	HPE 400GB SAS 12G MU SFF SSD
6	872359-B21	HPE 800GB SATA 6G WI SFF SSD
4	873365-B21	HPE 1.6TB SAS 12G MU SFF SSD
1	789006-B21	HPE Ethernet 10Gb 2-port 562FLR-SFP+ Adapter
1	869079-B21	HPE Smart Array E208i-a SR Gen10 12G SAS Controller
1	875675-B21	HPE DL560 Gen10 4xPSU Enablement Kit
4	865414-B21	HPE 800W Flex Slot Platinum Hot Plug Low Halogen Power Supply Kit
1	733662-B21	HPE 2U LFF Easy install Rail Kit
1	BD505A	HPE iLO Adv incl 3yr TSU 1-Svr Lic

Table 16. Bill of materials for the HPE ProLiant DL560 Gen10 server with 4-sockets

Qty	Part number	Description
HPE ProLiant DL560 Gen10 server		
1	841730-B21	HPE DL560 Gen10 CTO Server
1	840379-L21	HPE DL560 Gen10 Intel Xeon-Platinum 8176 (2.1GHz/28-core/165W) FIO Processor Kit
3	840379-B21	HPE DL560 Gen10 Intel Xeon-Platinum 8176 (2.1GHz/28-core/165W) Processor Kit
1	872222-B21	HPE DL5x0 Gen10 CPU mezzanine Board kit
24	815101-B21	HPE 64GB (1x64GB) Quad Rank x4 DDR4-2666 CAS-19-19-19 Load Reduced Memory Kit
1	872225-B21	HPE DL560 Premium Gen10 6SSF and 2 NVMe or 8SFF Bay1 Kit
1	872227-B21	HPE DL560 Premium Gen10 6SSF and 2 NVMe or 8SFF Bay2 Kit
2	873359-B21	HPE 400GB SAS 12G MU SFF SSD
10	872359-B21	HPE 800GB SATA 6G WI SFF SSD
4	873365-B21	HPE 1.6TB SAS 12G MU SFF SSD
1	789006-B21	HPE Ethernet 10Gb 2-port 562FLR-SFP+ Adapter
1	873444-B21	HPE DL5x0 Gen10 12Gb SAS Expanded Card Kit with Cables
1	804394-B21	HPE Smart Array E208i-a SR Gen10 12G SAS Controller
1	875675-B21	HPE DL560 Gen10 4xPSU Enablement Kit
4	865414-B21	HPE 800W Flex Slot Platinum Hot Plug Low Halogen Power Supply Kit
1	733662-B21	HPE 2U LFF Easy install Rail Kit
1	BD505A	HPE iLO Adv incl 3yr TSU 1-Svr Lic

Appendix B: Configuration adjustments

SQL Server 2017 Installation and Configuration

For general SQL Server 2017 installation, configuration details, and best practices, go to the Microsoft [SQL Server on Linux](#) page. The following scripts are used to configure the Resource Governor's database to NUMA Node affinity for this Reference Architecture.

Figure 7 shows a sample script to perform three tasks- create a user login for a specific workload group, create a resource pool that has affinity to NUMA Node 0, and create a workload group that will use the NUMA Node 0-affinitized resource pool. Substitute the name of login user, resource pool, and workload group to create the remaining Resource Governor profiles as found in Table 5. Through Table 8.

```
USE master
GO

IF NOT EXISTS (SELECT loginname FROM master.dbo.syslogins WHERE name = 'DBO1User')
    CREATE LOGIN DBO1User
    WITH PASSWORD = 'Passwd1234',
    DEFAULT_DATABASE = master,
    DEFAULT_LANGUAGE = us_english,
    CHECK_EXPIRATION = OFF,
    CHECK_POLICY = OFF
ELSE
    ALTER LOGIN DBO1User WITH PASSWORD = 'Passwd1234'
GO

ALTER LOGIN DBO1User ENABLE
GO

ALTER SERVER ROLE sysadmin ADD MEMBER DBO1User
GO

-- Create or Alter pDBO1Queries Resource Pool
IF EXISTS ( SELECT name FROM sys.resource_governor_resource_pools WHERE name = 'pDBO1Queries' )
    ALTER RESOURCE POOL pDBO1Queries WITH(min_cpu_percent = 0,
    max_cpu_percent = 100,
    min_memory_percent = 0,
    max_memory_percent = 100,
    cap_cpu_percent = 100,
    AFFINITY NUMANODE = (0),
    min_iops_per_volume = 0,
    max_iops_per_volume = 0)
ELSE
    CREATE RESOURCE POOL pDBO1Queries WITH(min_cpu_percent = 0,
    max_cpu_percent = 100,
    min_memory_percent = 0,
    max_memory_percent = 100,
    cap_cpu_percent = 100,
    AFFINITY NUMANODE = (0),
    min_iops_per_volume = 0,
    max_iops_per_volume = 0)
GO

ALTER RESOURCE GOVERNOR RECONFIGURE;
GO

-- Create or Alter gDBO1QueryStream Workload Group
IF EXISTS ( SELECT name FROM sys.resource_governor_workload_groups WHERE name = 'gDBO1QueryStream' )
    ALTER WORKLOAD GROUP gDBO1QueryStream WITH(group_max_requests = 0,
    importance = High,
    request_max_cpu_time_sec = 0,
    request_max_memory_grant_percent = 25,
    request_memory_grant_timeout_sec = 0,
    MAX_DOP = 1) USING pDBO1Queries
ELSE
    CREATE WORKLOAD GROUP gDBO1QueryStream WITH(group_max_requests = 0,
    importance = High,
    request_max_cpu_time_sec = 0,
    request_max_memory_grant_percent = 25,
    request_memory_grant_timeout_sec = 0,
    MAX_DOP = 1) USING pDBO1Queries
GO

ALTER RESOURCE GOVERNOR RECONFIGURE;
GO
```

Figure 7. Resource Governor. Create Unique Login User, Resource Pool, and Workload Group script.

Figure 8 shows a sample classifier function script that assigns each user to a specific workload group which is defined by the script above to use the corresponding resource pool that has CPU affinity to a specific NUMA Node.

```
USE master
GO

SET ANSI_NULLS ON;
GO

SET QUOTED_IDENTIFIER ON;
GO

CREATE FUNCTION dbo.OLTP_Classifier() RETURNS sysname
WITH SCHEMABINDING
AS
BEGIN
    DECLARE @WorkloadGroup AS SYSNAME
    IF(SUSER_NAME() = 'DB01User')
        SET @WorkloadGroup = 'gDB01QueryStream'
    ELSE IF (SUSER_NAME() = 'DB02User')
        SET @WorkloadGroup = 'gDB02QueryStream'
    ELSE IF(SUSER_NAME() = 'DB03User')
        SET @WorkloadGroup = 'gDB03QueryStream'
    ELSE IF (SUSER_NAME() = 'DB04User')
        SET @WorkloadGroup = 'gDB04QueryStream'
    RETURN @WorkloadGroup
END
GO

ALTER RESOURCE GOVERNOR WITH (CLASSIFIER_FUNCTION=dbo.TPCEClassifier);
GO

ALTER RESOURCE GOVERNOR RECONFIGURE
GO
```

Figure 8. Classifier function script for Resource Governor's workload.

Appendix C: Monitoring batch requests per second in Linux

Batch Requests calculations can be automatically calculated by using the sample script below. Adjust the *SMALLINT* variable (in seconds) the interval of which the batch requests are measured. Also, adjust the *cnt* variable how many times the measurements will be calculated. The example below will measure the batch requests 36 times (or 3 hours) in 5-minute intervals (SMALLINT=300).

```
DECLARE @v1 BIGINT, @delay SMALLINT = 300, @time DATETIME, @cnt INT = 0;

WHILE @cnt < 36
BEGIN

    SELECT @time = DATEADD(SECOND, @delay, '00:00:00');

    SELECT @v1 = cntr_value
    FROM master.sys.dm_os_performance_counters
    WHERE counter_name = 'Batch Requests/sec';

    WAITFOR DELAY @time;

    SELECT (cntr_value - @v1)/@delay
    FROM master.sys.dm_os_performance_counters
    WHERE counter_name='Batch Requests/sec';

    SET @cnt = @cnt + 1;
END;
```

Figure 9. Batch requests per second calculation script

Resources and additional links

HPE Reference Architectures, hpe.com/info/ra

HPE Servers, hpe.com/server/dl560

HPE Networking, hpe.com/networking

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

HPE SSD QuickSpecs, <http://h20195.www2.hpe.com/V2/GetDocument.aspx?docname=4AA4-7186ENW>

UEFT Workload-based Performance and Tuning Guide for HPE ProLiant Gen10 servers and HPE Synergy, https://support.hpe.com/hpsc/doc/public/display?docId=a00016408en_us

HPE Serviceguard for Linux Beta Program, <http://downloads.linux.hpe.com/SDR/project/sglx/sglx-beta/>

HPE Serviceguard for Linux, <https://h20392.www2.hpe.com/portal/swdepot/displayProductInfo.do?productNumber=SGLX-DEMO>

Microsoft SQL Server on Linux, <https://docs.microsoft.com/en-us/sql/linux/sql-server-linux-overview>

Microsoft SQL Server Migration Assistant, <https://docs.microsoft.com/en-us/sql/ssma/sql-server-migration-assistant>

Microsoft SQL Server Documentation, <https://docs.microsoft.com/en-us/sql/sql-server/sql-server-technical-documentation>

Microsoft SQL Server, Linux and HPE Performance Benchmark Love Story, https://community.hpe.com/t5/Servers-The-Right-Compute/Microsoft-SQL-Server-Linux-and-HPE-Performance-Benchmark-Love/ba-p/6960184#_Wf_7L2hSxaQ

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



Sign up for updates



© Copyright 2017–2018 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.

Microsoft, Windows, Windows Server, and SQL Server are registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries. Intel, Xeon, and Intel Xeon, are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Red Hat and Red Hat Enterprise Linux Certified are trademarks of Red Hat, Inc. in the United States and other countries.

a00036336enw, April 2018, Rev. 1