Dell Technologies

Reference Architecture

Microsoft SQL Server 2019 on RHEL

Reference Architecture for Dell EMC PowerFlex family

Abstract

This reference architecture demonstrates the deployment benefits and best practices of Microsoft SQL Server 2019 on Red Hat Enterprise Linux® platform hosted on Dell EMC PowerFlex family.

October 2020

000076

Revisions

Date	Description
October 2020	Initial release

Acknowledgements

Author: Anup Bharti and Kailas Goliwadekar

Table of contents

Executive summary

With Microsoft SQL Server 2019, it is possible to build intelligent, mission-critical applications using a scalable, hybrid database platform that has everything integrated, from in-memory performance, and advanced security to database analytics.

SQL Server 2019 (15.x) on Linux has the same underlying database engine with many similar features and services as on Microsoft Windows. With Microsoft SQL Server now available on Linux, many organizations that runs Linux can leverage SQL Server 2019 easily.

PowerFlex is a software-defined storage platform designed to deliver flexibility, elasticity, and simplicity with predictable performance and resiliency at scale by combining compute as well as high-performance storage resources in a managed unified fabric. PowerFlex comes in flexible deployment options that enable disaggregated (two-layer), HCI (single-layer), or mixed architectures.

This white paper focuses on the deployment details of Microsoft SQL Server 2019 on a Linux virtual machine using Dell EMC PowerFlex two-layer deployment architecture. In a two-layer architecture, nodes that provide storage capacity and host datasets are separated from nodes that host applications and workloads. This segregation of compute and storage resources can be helpful in minimizing software licensing costs. This architecture is suitable for hosting high-performance high value databases and application workloads.

1 Introduction

1.1 Terminology

The following table defines acronyms and terms that are used throughout this document:

Table 1 Acronyms and terms

Term	Definition
MDM	Meta Data Manager
SDC	Storage Data Client
SDS	Storage Data Server
SVM	Storage Virtual Machine
FG	Fine Granularity
MG	Medium Granularity
TPM	Transactions Per Minute
OLTP	Online Transaction Processing
DSS	Decision Support System
RHEL	Red Hat Enterprise Linux
RCM	Release Certification Matrix
VM	Virtual Machine

1.2 Objective

The primary focus of this reference architecture is to provide reader with details of the deployment of SQL Server 2019 on a Linux VM and benchmark its performance whilst running an OLTP workload. This paper also provides best practices to be followed while deploying the Microsoft SQL Server 2019 on Linux VM powered by Dell EMC PowerFlex.

1.3 Audience

This reference architecture is intended for Microsoft SQL Server database administrators, system engineers, members of Dell EMC and partner professional service community who are looking to deploy Microsoft SQL Server on Linux with Dell EMC PowerFlex.

Readers are expected to have an understanding and working knowledge of Dell EMC PowerFlex rack, PowerFlex Manager, PowerFlex (previously VxFlex OS), Red Hat® Linux administration, VMware vSphere and Microsoft SQL Server database administration.

2 Product overview

2.1 PowerFlex family

PowerFlex is a software-defined storage platform designed to significantly reduce operational and infrastructure complexity, empowering organizations to move faster by delivering flexibility, elasticity, and simplicity with predictable performance and resiliency at scale. The PowerFlex family provides a foundation that combines compute as well as high performance storage resources in a managed unified fabric. PowerFlex comes in flexible deployment options - rack, appliance or ready nodes - that enable two-layer (compute and server SAN), single-layer (HCI), and/or storage only architectures. PowerFlex is ideal for high performance applications and databases, building an agile private cloud, or consolidating resources in heterogeneous environments.

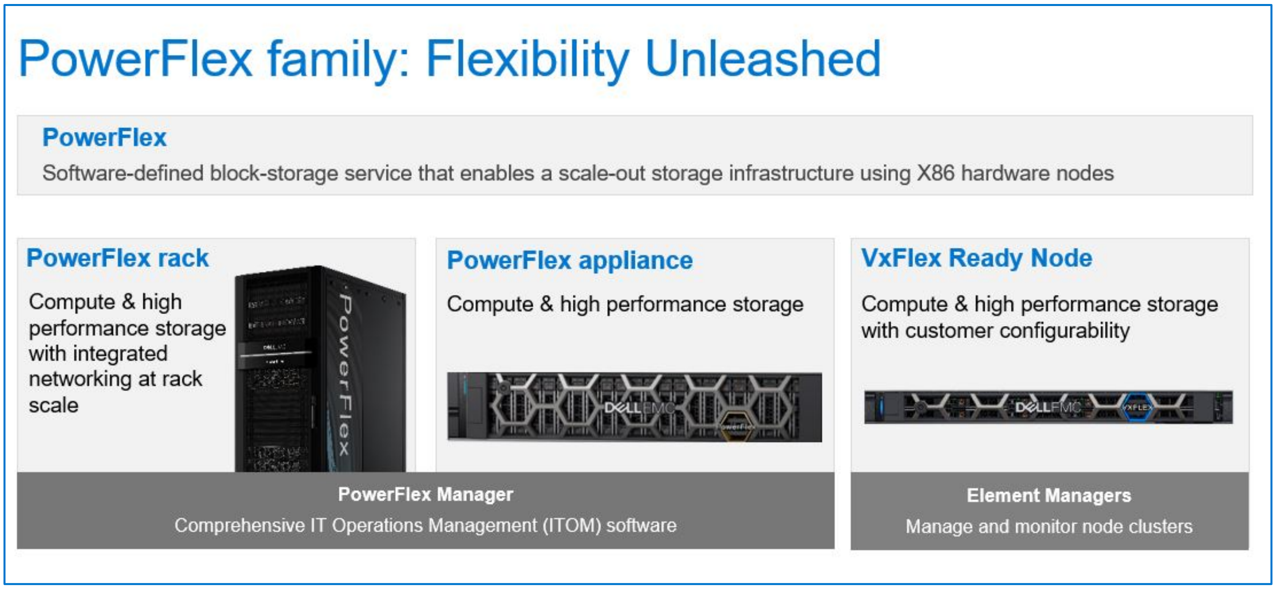


Figure 1 PowerFlex family

2.2 PowerFlex software components

Software is the key differentiation and the “secret sauce” in the PowerFlex offering. The PowerFlex software components not only provide software-defined storage services, they also help simplify infrastructure management and orchestration with comprehensive ITOM and LCM capabilities that span compute as well as storage infrastructure, from BIOS and firmware to nodes, software and networking.

The core foundational component in the PowerFlex family that enables Software Defined Storage (SDS) services is called PowerFlex, to represent the core value it enables for the platform. Additionally, PowerFlex Manager is a comprehensive IT Operational Management (ITOM) and Life Cycle Management (LCM) tool that drastically simplifies management and ongoing operation.

2.3 PowerFlex deployment architectures

PowerFlex software-define storage offers flexibility of deployment architecture to help best meet the specific deployment and architectural requirements. PowerFlex can be deployed in a two-layer (Server SAN), single layer (HCI), or in storage-only architectures. In this reference architecture, PowerFlex is deployed using two-layer architecture.

2.4 Microsoft SQL Server

Microsoft® SQL Server® 2019 delivers breakthrough mission critical capabilities with in-memory performance and operational analytics built in while comprehensive security features help protect your data at-rest, data in-motion and now support for integrated big data cluster.

SQL Server 2019 runs on Linux. It is the same SQL Server database engine, with many similar features and services regardless of your operating system.

The following table shows key features that are new or enhanced in SQL Server 2019:

Table 2 SQL Server 2019 features

Feature	Description
Replication support	SQL Server Replication on Linux
Support for the Microsoft Distributed Transaction Coordinator (MSDTC)	How to configure MSDTC on Linux
Open LDAP support for third-party AD providers	Tutorial: Use Active Directory authentication with SQL Server on Linux
Machine Learning on Linux	Configure Machine Learning on Linux
`tempdb` improvements	By default, a new installation of SQL Server on Linux creates multiple `tempdb` datafiles based on the number of logical cores (with up to eight datafile). This does not apply to in-place minor or major version upgrades. Each `tempdb` file is 8 MB with an auto growth of 64 MB. This behavior is like the default SQL Server installation on Windows.
PolyBase on Linux	Install PolyBase on Linux for non-Hadoop connectors. For details, see PolyBase type mapping.
Change Data Capture (CDC) support	CDC is now supported on Linux for SQL Server 2019.

For complete information about new features and enhancements, see SQL Server 2019 documentation.

3 Solution architecture

In this solution, Microsoft SQL Server 2019 database is deployed on RHEL 8.0 VM which is hosted on PowerFlex rack in a two-layer configuration. SQL VMs are managed by ESXi hypervisor that are installed on compute nodes whereas database volumes are provisioned and managed by PowerFlex software installed on storage nodes.

This solution also can be deployed on any of the PowerFlex portfolio.

3.1 Logical architecture

The following diagram shows logical architecture of Microsoft SQL Server 2019 deployment on Dell EMC PowerFlex rack in two-layer setup with five SDS and three SDC.

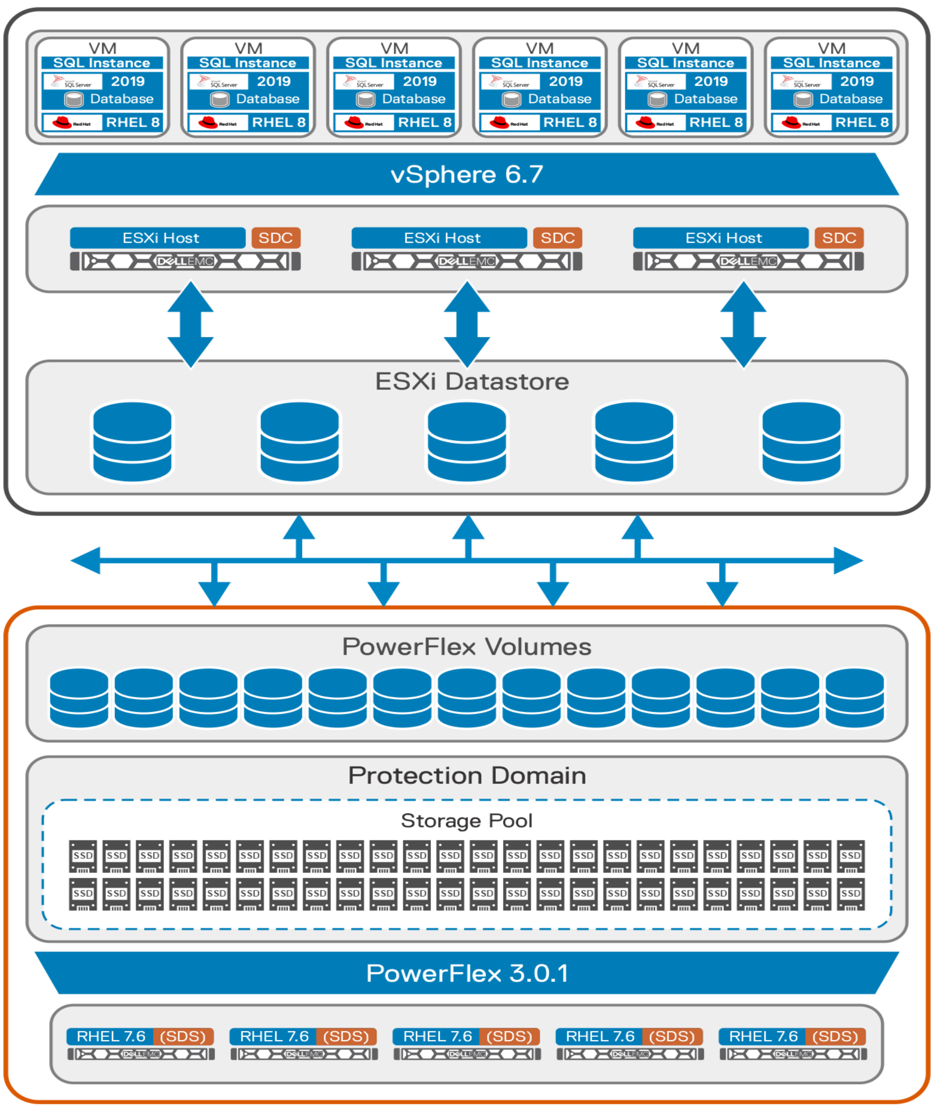


Figure 2 Logical architecture of SQL server 2019 on PowerFlex rack

The PowerFlex cluster is built from the five RHEL Storage only nodes configured in single protection domain to provide the foundation of the software defined storage. Each host is populated with ten 1.92 TB SSD drives. These 50 disks are placed in storage pool from which the usable storage volumes are created.

PowerFlex Manager supports the deployment of compute-only resources. Three compute nodes are hosting the SDC component of the PowerFlex cluster and are responsible for accessing the volume created in the storage pool hosted on the SDS (storage-only) nodes.

VMware vSphere 6.7 is used to create datastores from the volumes on SDS. vSphere is also used to create VM and to present datastores to VM. RDM disk can also be used as a database volume to the VM.

The SQL VMs are configured as per the Virtual Machine vCPU and vNUMA Rightsizing-Guidelines. SQL server is NUMA aware and it creates soft-NUMA nodes for its database engines. RHEL 8.0 has been made a default choice for operating system because of the accelerated I/O performance on XFS file system. SQL Server with NVMe results in high performance and low latency which translates into more processing of data and analysis. SQL Server thus benefits in providing higher workload consolidation and achieving greater cost savings.

For detailed configuration, see Appendix A.1: PowerFlex node configuration.

3.2 Network architecture

The following diagram depicts the two-layer network architecture that is based on PowerFlex best practices:

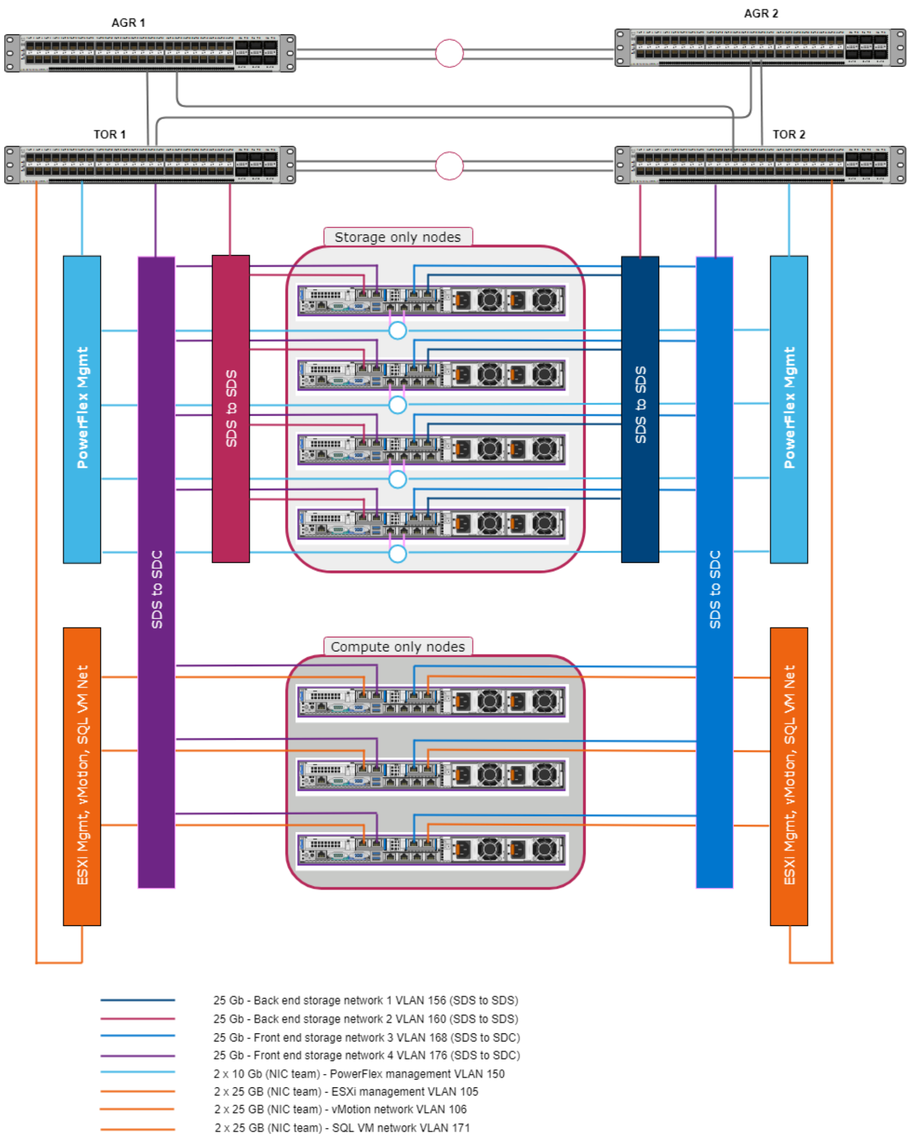


Figure 3 Network architecture for SQL server 2019 deployment on PowerFlex rack

Table 3 vLAN details

Port Group	VLAN	Purpose
esxmgmt	105	ESXi management traffic
vMotion	106	VM migration traffic
powerflex-mgmt	150	PowerFlex Management
powerflex-data1	156	SDS to SDS traffic
powerflex-data2	160	SDS to SDS traffic
SQL VM network	171	SQL VM communication

- Two Cisco 93180YC switches are configured with VCP to provide fault tolerance and enable connectivity with other switches.
- Two dual port 25 Gb Mellanox NICs on each server provide 4 x 25 Gb ports.
- VLAN 156 and 160 is dedicated to SDS-SDS communication. Both VLANs are isolated at switch level to provide high availability and avoid single point of failure.
- VLAN 168 and 176 is dedicated to SDS-SDC communication. Both VLANs are isolated at switch level to provide high availability and avoid single point of failure.
- On storage nodes, 2 x 10 Gb ports are NIC teamed on VLAN 150 to provide dedicated PowerFlex management network.
- On compute nodes, 2 x 25 Gb ports are NIC teamed to provide high availability. VLAN 171 is configured to provide connectivity with the customer network, VLAN 106 is dedicated vMotion, and VLAN 105 is dedicated to Hypervisor (ESXi) management.

3.3 Storage layout

In any software defined storage environment, the storage layout is an important aspect. The PowerFlex cluster leverages its SDS component to provide a resilient storage layout. The Storage Data Server (SDS) is a service that aggregates raw local storage of all participating nodes and presents a high-end virtual storage system. The SDS nodes may leverage SSDs, PCIe based flash, spinning media, available RAM, or any combination thereof to provide fault tolerance system. A collection of SDSs forms the PowerFlex persistence layer. SDS components can communicate directly with each other, and collections of SDSs are fully meshed.

SDSs are optimized for rebuild, rebalance, and I/O parallelism. The data layout among the SDS nodes may be organized into protection domains and storage pools, from which the volumes are allocated.

A protection domain is set of SDS configured in separate logical groups. An SDS node may belong to only a single Protection Domain, so they may be used to divide a PowerFlex system into smaller logical systems. Protection from node, device, and network connectivity failure is managed with node-level granularity through protection domains.

Client volumes used by the application workloads are allocated from a storage pool. Storage pools can be used to logically aggregate types of storage media at drive-level granularity. Storage pools can provide varying levels of storage service distinguished by capacity and performance. Following is the storage layout design for SQL Server Database:

- Single protection domain (PD-1) from five storage nodes for SQL server project
- Single storage pool (SP-1) with 50 disks.
- Six volumes for SQL database files and transaction log files on PowerFlex cluster. Map these volumes to ESXi hosts as a datastore and attach it as a disk to the SQL database VMs using para virtual (PVSCSI) adapter.

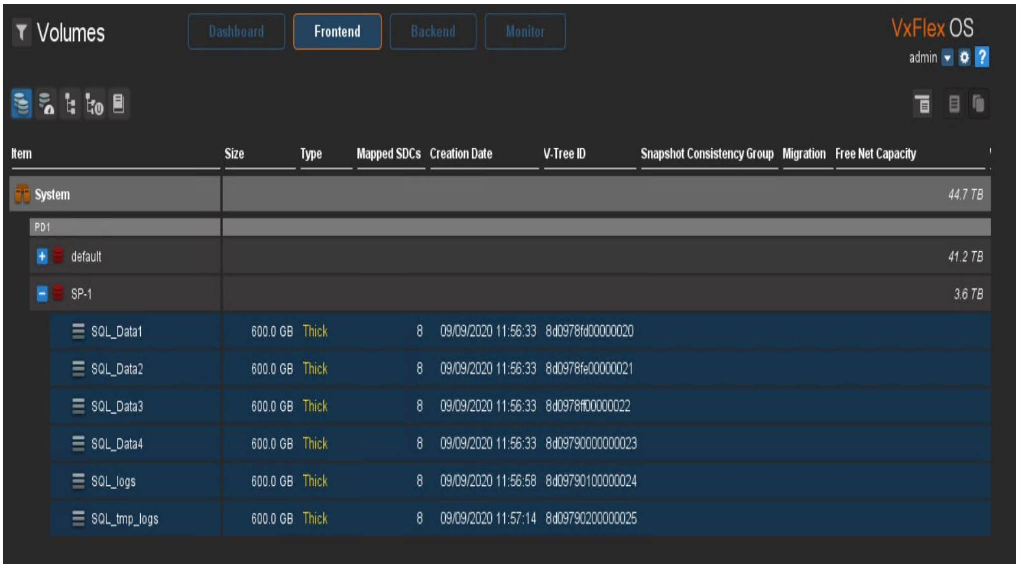


Figure 4 Database LUN on PowerFlex (previously knows as VxFlex) cluster

3.4 SQL Server layout

To get the optimal database performance it is important to design the database layout carefully by following best practices.

SQL Server database files are mostly categorized in random and sequential files, storing all the files in one partition/drive is sign of bad database design. To fully leverage SDS storage pool capacity and I/O performance, database and log files are distributed across different Linux partitions.

The following figure depicts the database files divided across different partitions for better database performance. For better database performance on VMware, see [Architecting Microsoft SQL Server on VMware.]

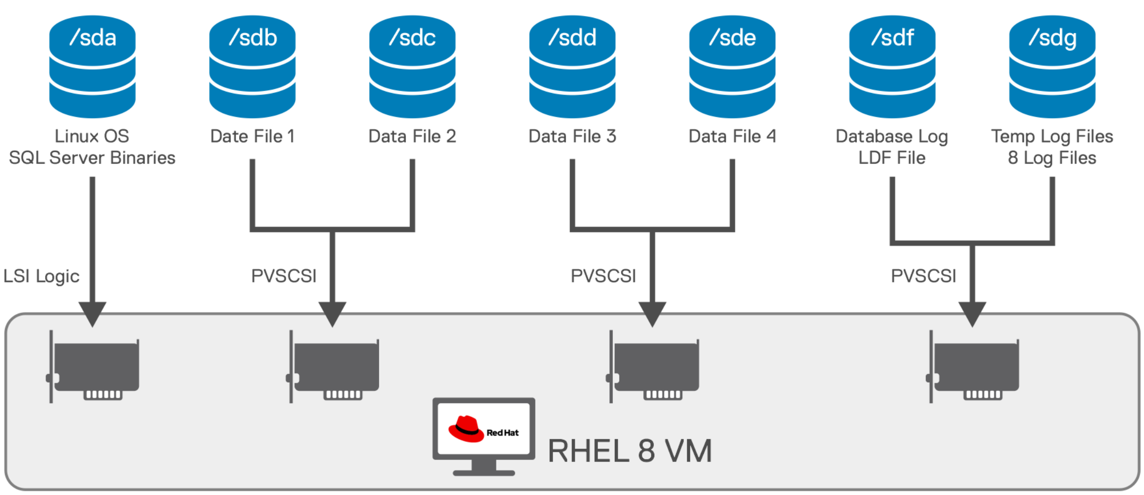


Figure 5 SQL server database layout

Further considering the best practices from the virtualization world, this solution used the PVSCSI controller for database volumes and LSI Logic for the OS volume. Due to configuration limit of four SCSI controllers for a virtual machine with vSphere 6.7, one controller is shared between operating system and SQL binaries whereas the other three controllers are shared between different database volumes. PVSCSI is high performance storage adapter that can result in greater throughput and lower CPU utilization.

As shown in the figure, database files are shared across different partitions and log files are stored separately. **`Tempdb`** files are moved from their default location to a PowerFlex volume to optimize the database performance. Pre-allocating space for **`tempdb`** and assigning a large capacity that can easily accommodate a typical workload results in better performance. In addition, by simply moving **`tempdb`** to a PMEM device results in a substantial performance increase.

The following table shows partition size and file system used for TPC-C database:

Table 4 SQL Server database layout

Volume name	Size	File system	Purpose
/sda	200 GB	xfs	OS volume
/sdb	100 GB	xfs	Database partition
/sdc	100 GB	xfs	Database partition
/sdd	100 GB	xfs	Database partition
/sde	100 GB	xfs	Database partition
/sdf	100 GB	xfs	Database logs
/sdg	100 GB	xfs	tempdb database and logs

4 Testing and Validation

This section provides a detailed summary and description of the tests that were performed to validate the SQL Server 2019 performance with Red Hat Linux Platform (RHEL) platform hosted on PowerFlex rack. HammerDB benchmark tool has been made choice for database performance test.

HammerDB is leading benchmarking tool and load testing software for world's most popular databases. It mainly focuses on schema creation, data loading and simulating any transactional and analytic workload. Result of these workload tests further derives meaningful information from the environment such as hardware performance comparison and software configurations. For more information, see HammerDB official site.

4.1 Test methodology

To validate the performance of the SQL Server 2019 on RHEL 8.0, an OLTP workload was simulated with TPC-C benchmark test using HammerDB tool. The TPC-C benchmark typically generates a real time transactional workload that runs specific set of queries in one database transaction, such database transactions are measured in TPM (Transactions per minute).

For this reference architecture, a TPC-C database was populated with 500 GB of data and multiple virtual user tests were executed in frequent intervals to generate higher TPM numbers. The lab was setup as per the Logical Architecture section in this document. A load generation VM/host was used to run the HammerDB tool to create real time client/server scenarios. Multiple runs were executed by simulating number of virtual users across TPC-C database. The load generation VMs were hosted on a different host other than the PowerFlex cluster nodes as seen in the following figure:

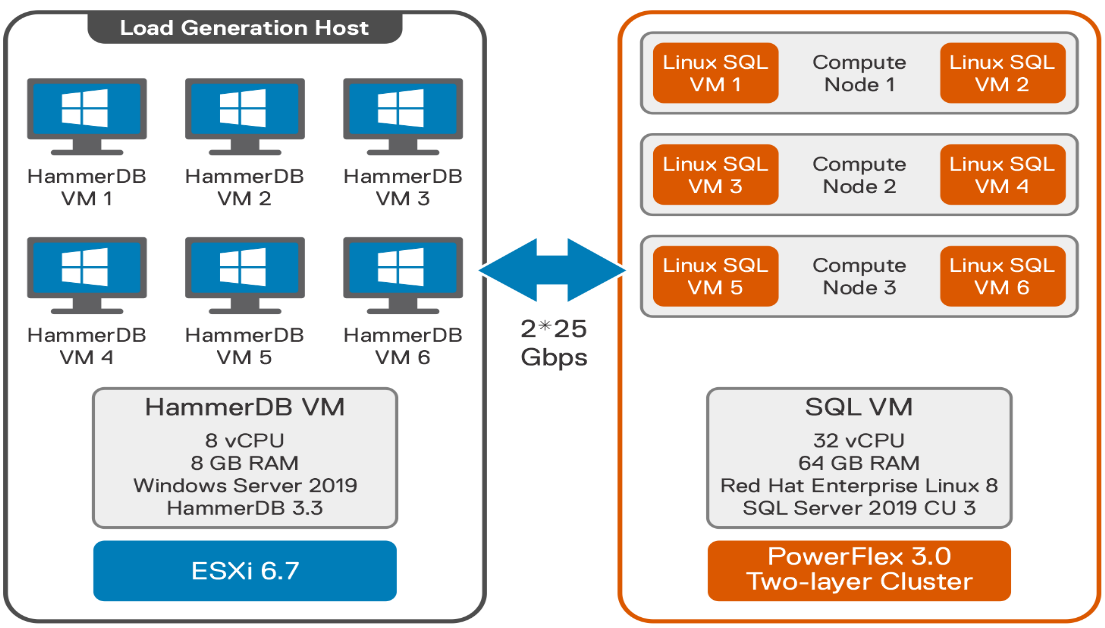


Figure 6 HammerDB load generation host depicting client/server architecture

4.2 Test results

We have considered following two deployment scenarios to test above methodology:

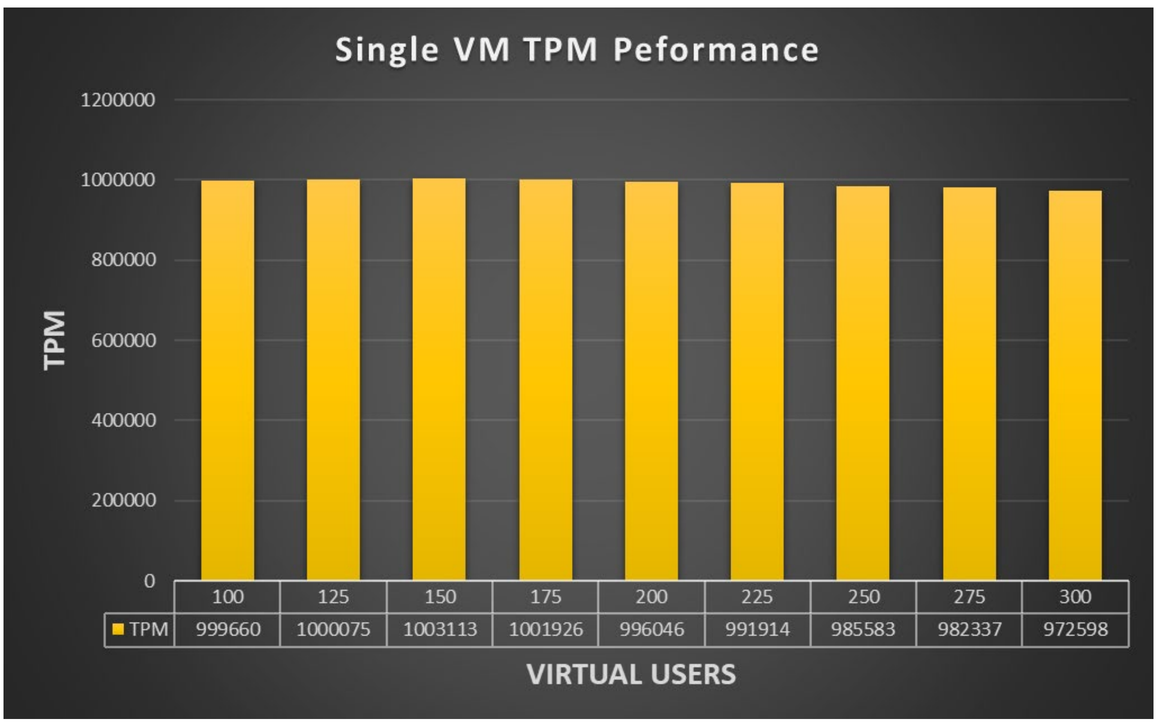
- Single VM Test
- Linear Scaling Test

To tune the VM and SQL Server for the above tests, few best practices have been followed as explained in Best practices section in this document.

4.2.1 Single VM test

The objective of this test was to check the single VM TPM performance on only one PowerFlex node. For configuration details of VM, see A2. VM configuration. Datafiles of TPCC database were distributed across four partitions. Log files are stored on separate partition and **`tempdb`** was stored on another partition. Store files on separate partition to get dedicated IOPS for the database and less contention inside database pages.

The following figure displays 16 vCPU TPM performance on single Linux VM:



	100	125	150	175	200	225	250	275	300
TPM	999660	1000075	1003113	1001926	996046	991914	985583	982337	972598

Figure 7 TPM Performance on 16 vCPU Linux VM

As seen from the above test results maximum TPM achieved was 1.03 million TPM with 150 virtual users under 1 millisecond read latency and 1.5 millisecond write latency. CPU and memory utilization of database VM for this test was approximately 70%.

4.2.2 Linear scaling test

This section focuses on scalability test up to six SQL VMs of 32 vCPU across three compute node cluster. Each VM was populated with 500 GB of data. HammerDB was deployed and configured on separate load generation VM to simulate real time client/server workload. For configuration detail of VM, see A.2 VM configuration.

Test started with one VM and with each iteration VM addition was done in following manner:

Table 5 VM scaled across three compute node cluster

	Node 1		Node 2		Node 3	
	VM 1	**VM 2**	**VM 3**	**VM 4**	**VM 5**	**VM 6**
Test 1	Active					
Test 2	Active		Active			
Test 3	Active		Active		Active	
Test 4	Active	Active	Active		Active	
Test 5	Active	Active	Active	Active	Active	
Test 6	Active	Active	Active	Active	Active	Active

The above table explains the test sequence and proves that in Test 2, VM1 and VM3 were active whereas in Test 6 shows that six VMs were active and scaled across three compute node cluster.

The following graph shows the TPM performance with linear scaling of six VMs on three nodes compute cluster. After multiple iterations with different virtual users, achieved maximum TPM numbers with 150 virtual users This test resulted in achieving more than 7 million TPM under 1 millisecond read latency and 1.5 millisecond write latency. CPU utilization of database VM for this test was approximately 100% with 70% memory utilization.

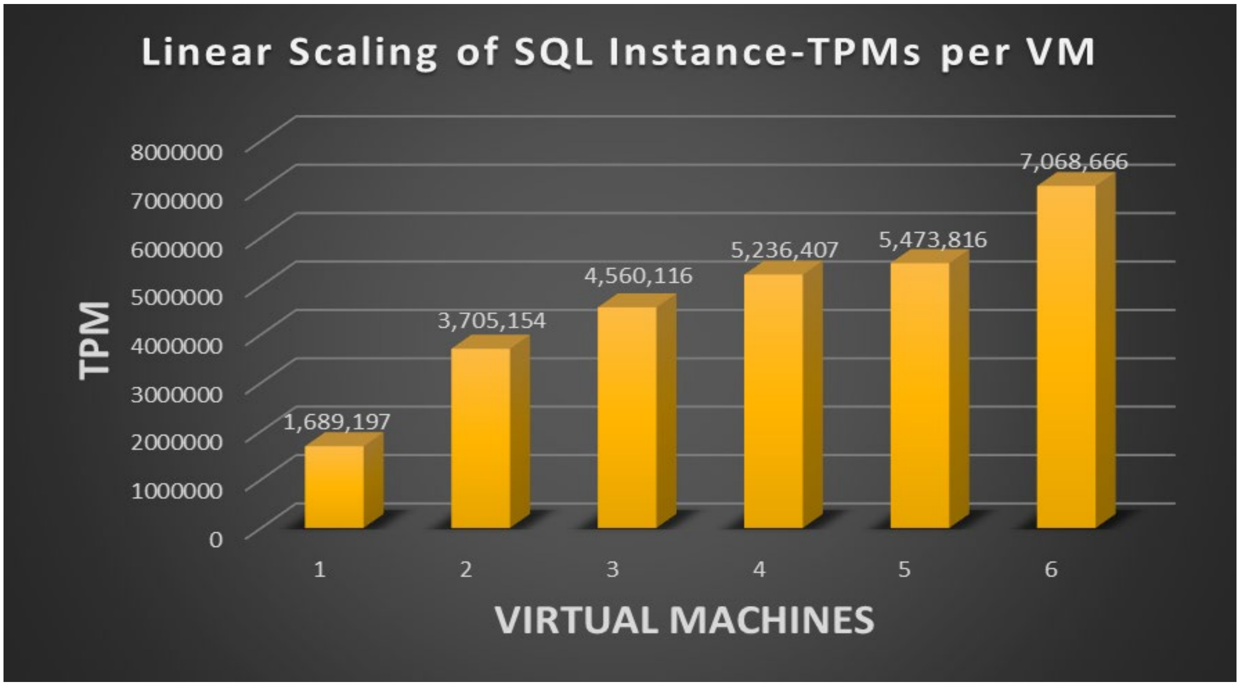


Figure 8 Total TPM Performance of SQL VM while scaling Linearly

This test proves that the performance of SQL server 2019 on Linux with Dell EMC PowerFlex increases linearly as the node scales in cluster.

5 Best practices

This section describes the best practices derived from performance test and the standard practices to deploy PowerFlex cluster. Most of the PowerFlex and VMware practices are standard proven best practices whereas SQL server and Linux best practices are derived after running multiple test by fine tuning various parameters.

5.1 PowerFlex

For better understanding of PowerFlex best practices, it has further divided into three categories:

5.1.1 PowerFlex rack

- Minimum number of nodes for production workload is seven for a PowerFlex two-layer configuration in a VMware environment.
- Homogenous node types are recommended for predictable performance. With homogeneous compute nodes and homogeneous storage nodes, compute and storage do not need to be the same.
- Change the passwords for all default accounts.
- Ensure PowerFlex rack is compliant to an RCM. For more information, see PowerFlex Specification Sheet .

5.1.2 PowerFlex networking

- Enable secure network protocol options only (for example, HTTPS and Secure Shell (SSH).
- Use VLANs to separate management and data traffic from production application traffic.
- Follow PowerFlex standards to separate VMware vSphere vMotion traffic from production traffic.

5.1.3 PowerFlex

- Configure high-performance profile for MDM, SDS, and SDC.
- Recommended maximum number of nodes in a protection domain is 32.
- Change following grub settings for Skylake processor to ensure that CPU maintain good performance in terms of latency and rebuild the grub:

```
GRUB_CMDLINE_LINUX="crashkernel=auto rd.lvm.lv=rhel/root
rd.lvm.lv=rhel/swap rhgb intel_idle.max_cstate=1 intel_pstate=disable
quiet"
grub2-mkconfig -o /boot/grub2/grub.cfg
grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg — for UEFI-based systems
```

- It is recommended to use noop I/O scheduler for all the SSD devices discovered on SDC hosts. To make the changes persistent through boot add `elevator=noop` to `GRUB_CMD_LINUX` line and rebuild the grub. Follow below snippet to change the same:

```
GRUB_CMDLINE_LINUX="crashkernel=auto rd.lvm.lv=rhel/root
rd.lvm.lv=rhel/swap rhgb quiet elevator=noop"
```

For more information, see PowerFlex tuning.

5.2 VMware

- Always use para virtualized SCSI controller on guest VMs for high performance.
- Do not overcommit the VM host CPU.
- Enable hyper-threading for each SQL server host so as to improve the overall host performance by keeping the CPU pipeline busy and allow hypervisor to schedule CPU clock cycle correctly.
- Make sure that SQL VM fits into correct NUMA node configuration so that SQL server can leverage this setting for its database engine.
- Change per device queue length i.e. "No of outstanding IOs" to 256 on each compute node to improve the IO concurrency of each data store.

For more information about SQL server best practices from VMware, see Architecting Microsoft SQL Server on VMware vSphere®.

5.3 Red Hat Linux

- Modify **max_sector_kb** to 1024 for all database volume.

```
echo 1024 > /sys/block/sdc/queue/max_sectors_kb
```

- Modify **nr_requests** to 1024 for all database volumes

```
echo 1024 > /sys/block/sdc/queue/max_sectors_kb/nr_requests
```

- Disable NUMA balancing to allow SQL server to operate at maximum efficiency on NUMA node.
- Set open file limit to maximum of 60000.
- Disable last accessed date/time on file systems for SQL Server data and log files.
- Make use of tuned profile-based system tuning tool of linux to create profile for SQL server. For sample profile see, A.3 MSSQL tuned profile.

For more information about Linux best practices, see Performance best practices and configuration guidelines for SQL Server on Linux.

5.4 SQL Server 2019

- Shard database into four or more datafiles and store it on separate database volume.
- Store database transaction logs on separate volume to leverage storage pool capacity and IO performance.
- Move **tempdb** files from the default location to separate volume or NVME persistent storage (if available) to avoid page contention. This practice will substantial improve the performance.
- Create multiple **tempdb** datafiles after installation.
- Set 85% of SQL server memory limit with **mssql-conf** tool.

5.5 Hammer DB

- Configure max degree of parallelism to 1.
- Configure max worker threads to 3000
- Keep initial data file size sufficiently large to ensure that files are not continuously growing the expense of performance.
- Auto growth settings should be configured correctly so as the file grows sufficiently without any performance impact.

For more details about HammerDB best practices, see HammerDB documentation.

6 Conclusion

The reference architecture demonstrates the deployment of SQL server 2019 on Dell EMC PowerFlex two-layer architecture with Linux Platform. In addition, it states the best practices to configure SQL server 2019 for Linux environment to meet performance, resiliency and scale.

This paper also showcases how Dell EMC PowerFlex can been leveraged to achieve superior database performance of SQL server 2019 on Linux as seen in the Test results section. In addition, it also states how easily SQL 2019 VMs can be scaled across PowerFlex cluster.

The segregation of compute and storage resources can be helpful to minimize software licensing costs in certain situations. This architecture is particularly suitable for hosting high-performance databases and application workloads to achieve optimal and fast performance.

Appendix

A.1 PowerFlex node configuration

The following table summarizes the hardware and software resources used to carry out the performance test:

Table 6 Hardware and software details

Component	Details
PowerFlex rack	**Compute Nodes** **3 x PowerFlex Nodes (PowerEdge R640 servers):** • VxFlex OS* version: R3_0.1 • ESXi version: 6.7 U2 • CPU: 2 x Intel(R) Xeon(R) Gold 6152 CPU @ 2.10 GHz, 22 cores • Memory: 384 GB RAM (12 x 32 GB DIMMs) **Storage Nodes** **5 x PowerFlex Nodes (PowerEdge R640 servers):** • VxFlex OS* version: R3_0.1 • VMware ESXi version: 6.7 U2 • CPU: 2 x Intel(R) Xeon(R) Gold 6152 CPU @ 2.10 GHz, 22 Cores • Memory: 384 GB RAM (12 x 32 GB DIMMs) • Disks: 10 x 1.92 TB SSD ***Note:** VxFlex OS rebranded as PowerFlex from version 3.5.
Network	• 2 x Mellanox ConnectX-4 Lx 25 GbE Adapter • 2 x Cisco 93180YC switches

A.2 VM configuration

Table 7 Virtual machine details

Hardware	Configuration
vCPU	16 and 32 vCPU configurations tested
Memory	64 GB
Storage	• OS volume: 200 GB • Data volume: 4*100 GB • Log volume: – 100 GB database transaction log – 100 GB temp database
Operating System	RHEL 8.0
PowerFlex Storage	600 GB

A.3 MSSQL tuned profile

- Steps to create tuned profile for SQL server database engine.
- Create a directory with name as **`mssql`** under **`/usr/lib/tuned`** directory.
- Create a file named as **`tuned.conf`** here with following snippet:

```
# A tuned configuration for SQL Server on Linux
          [main]
         summary=Optimize for Microsoft SQL Server
         include=throughput-performance
          [cpu]
         force_latency=5

         [sysctl]
         vm.swappiness = 1
         vm.dirty_background_ratio = 3
         vm.dirty_ratio = 80
         vm.dirty_expire_centisecs = 500
         vm.dirty_writeback_centisecs = 100
         vm.transparent_hugepages=always
         # For , use
         # vm.transparent_hugepages=madvice
         vm.max_map_count=1600000
         net.core.rmem_default = 262144
         net.core.rmem_max = 4194304
         net.core.wmem_default = 262144
         net.core.wmem_max = 1048576
         kernel.numa_balancing=0
         kernel.sched_latency_ns = 60000000
         kernel.sched_migration_cost_ns = 500000
         kernel.sched_min_granularity_ns = 15000000
         kernel.sched_wakeup_granularity_ns = 2000000
```

- Save the file and activate this profile by following command:

```
tuned-adm profile mssql
```

- Check your activated profiled with following command:

```
tuned-adm list
```

B Technical support and resources

B.1 Related resources

See the following referenced or recommended resources that are related to this document:

Note: The following links are open to customers although some may require registration for access.

- SQL Server on Linux
- SQL Server 2019 on VMware
- Dell EMC PowerFlex – Software-Defined Storage to Harness the Power of Change
- Quickstart: Install SQL Server and create a database on Red Hat
- Hammer DB Official Website

B.2 Additional resources

Referenced or recommended publications:

Dell.com/support is focused on meeting customer needs with proven services and support.

Storage technical documents and videos provide expertise that helps to ensure customer success on Dell EMC storage platforms.