# SQL Server 2019 Best Practices on PowerEdge Servers and PowerStore T Model Arrays

## SQL Server 2019 best practices on Dell PowerEdge R750xs, PowerStore 5000T, Red Hat Enterprise Linux 8.5, and VMware vSphere 7.0.3

September 2022

H19332

White Paper

Abstract

This white paper provides design, configuration, and best practice guidelines for optimizing SQL Server 2019 running on Dell PowerEdge R750xs server, Dell PowerStore 5000T, Red Hat Enterprise Linux 8.5, and VMware vSphere 7.0.3.

Dell Technologies Solutions

Dell Validated Design

## Copyright

# Contents

# Introduction

## Executive summary

The number of databases and database infrastructure has increased for most enterprises. This can present challenges to IT staff and database administrators (DBAs) to keep mission-critical databases running efficiently and optimally, especially in consolidated infrastructures. One strategy to mitigate this challenge is to apply best practices for each component in the database infrastructure solution.

Dell is committed to ensuring customer success by providing best practices which were developed, validated, and consolidated by the engineers and subject matter experts (SME). The goal of this document is to improve customer outcomes when using Dell's infrastructure for SQL Server database solutions.

This paper provides best practice guidance for using Dell PowerStore T model arrays and PowerEdge R750xs servers with SQL Server database running in a virtualized environment with VMware and Red Hat Enterprise Linux.

## Document purpose

The purpose of this document is to describe the design and configuration of best practices for VMware-based virtualized SQL Server 2019 database solutions that use PowerEdge R750xs servers and PowerStore 5000T storage array. This guide provides an overview of the architectural design and best practices for the entire stack, including the compute, network, and storage. It highlights the best practices that are based on the layers in the database infrastructure stack: storage, servers, network, VMware vSphere, operating system, and database configuration.

## Audience

This document is intended for solution architects, data scientists, database administrators, VMware administrators, and storage administrators. Readers of this guide should be familiar with Dell PowerStore storage array, Dell PowerEdge R750xs servers, VMware vSphere 7.0.3, Red Hat Enterprise Linux 8.5, and SQL Server 2019.

# Business challenges

## Market environment

One of the biggest issues customers face with is how to fine-tune the recommended settings. This does not mean that the SQL Server solution will perform optimally after the first attempt. This solution performance optimization process requires constant observations and testing to come up with a set of best practices for the whole solution. To achieve this milestone, each component in the solution must be tuned individually and then combined with other components to ensure that all changes work in harmony. Those components in the infrastructure solution consist of:

- Storage Configuration
- CPU Configuration
- Linux Operating System Configuration
- Hypervisor Configuration

- Database Configuration

In addition, this tuning exercise must be done on a separate environment such as test or dev to avoid any problems that could arise when implemented on production. Since performance tuning is an art and the best practices derived from the initial testing may not fit for every workload, patience and time are required to come up with the right set of best practices that can be beneficial in the end.

# Solution overview

The development of the SQL Server best practices program has one primary goal: improve customer outcomes using Dell infrastructure. The goal of this SQL Server best practices program is to centralize and validate recommendations that will make you successful in running an SQL Server database on Dell infrastructure.

## The approach

At the beginning of the program, we gathered recommendations from multiple sources and categorized the guidelines into different layers of the database infrastructure. For example, for the optimal PowerEdge configuration for databases tests, the team pulled best practices from current manuals and held meetings with PowerEdge experts to review the findings. This meant we actively worked with specialists and engineers responsible for each part of the SQL Server infrastructure. The following list describes each infrastructure layer validated for best practices, in the order tested:

1. Optimal Storage Configuration
2. Optimal CPU configuration
3. Optimal Linux and Database Configuration

## Benchmark tool

The engineering team conducted internal load testing to determine the workload profile before starting the validation tests. HammerDB was used to generate an Online Transaction Processing Workload (OLTP) that simulates many common enterprise applications. The goal of generating a significant load on the SQL Server infrastructure was to ensure that the system was sufficiently taxed to demonstrate that best practices optimized performance. In this case, the initial target was six VMs each with ten processor cores. The HammerDB workload configuration is described in the below table.

**Table 1. HammerDB workload configuration**

| Setting name | Value |
|---|---|
| Total transactions per user | 1,000,000 |
| Number of warehouses | 5,000 |
| Number of virtual users | 80 |
| Minutes of ramp up time | 10 |
| Minutes of test duration | 50 |
| Use all warehouses | Yes |
| User delay (ms) | 500 |
| Repeat delay (ms) | 500 |
| Iterations | 1 |

With this HammerDB configuration, each best practice was validated in an hour-long workload test: 10 minutes ramp up time plus 50 minutes for test duration. We chose to run the workload for one hour to ensure that the database system reached a consistent performance state. Reaching a consistent run state determines whether the configuration is stable, and the best practice prove value over time.

The HammerDB parameter *Use all warehouses* enables increased I/O to the database area by assigning all the warehouses to the virtual users. The result of forcing the use of all warehouses means the workload will generate more I/O on the storage subsystem. The first set of best practices compares baseline database performance without an optimal storage configuration to a database configuration with an optimal storage configuration.

The metrics New Order per Minute (NOPM) and Transaction per Minute (TPM) help us interpret our results. These metrics are from the TPC-C benchmark and indicate the result of a test. During our best practice validation, we compared these metrics against the baseline to ensure that there is an increase in performance.

## Test methodology

The team sequentially tested a best practice or a set of practices during the program. For example, with the storage configuration, the team tested with multiple volumes using the PowerStore Volume Groups feature.

The purpose of the storage configuration best practice category was to optimize the SQL Server database using the PowerStore volume groups. The goal was to show how multiple volumes can improve SQL Server database performance. With each additional best practice, the working theory was that we would observe a gain in performance. For example, the last test in changing the SQL Server database should achieve the overall optimal SQL Server performance. Building upon prior best practices shows that when best practices are deployed together, overall database performance becomes optimized.

## Best practices category

All best practices are not created equal. To enhance the value of best practices, we have identified which configuration changes produced the greatest results. This should enable anyone reviewing the best practices to easily identify the recommendations that will have the most value. Best practices were categorized as follows:

- Day 1 through 3: Most enterprises implement configuration changes based on the delivery cycle:
  - **Day 1**: Indicates configuration changes that are part of provisioning a database. The business has defined these best practices as an essential part of delivering a database.
  - **Day 2**: Indicates configuration changes that are applied after the database has been delivered to the customer. These best practices address optimization steps to further improve system performance.
  - **Day 3**: Indicates configuration changes that provide small incremental improvements in database performance.
- Highly, moderately, and fine-tuning recommendations: Customers want to understand the impact of the best practices and these terms are used to indicate the value of each best practice.
  - **Highly recommended**: Indicates best practices that provided the greatest performance in our tests.

    - **Moderately recommended**: Indicates best practices that provide modest performance improvements, but which are not as substantial as the highly recommended best practices.
    - **Fine-tuning**: Indicates best practices that provide small incremental improvements in database performance.

When reviewing the best practices, the day and value of the recommendations are combined. Here are some examples:

- **Day 1, highly recommended**: Indicates these best practices should be part of provisioning an SQL Server database as they provide the greatest performance.
- **Day 2, moderately recommended**: Indicates these best practices should be considered as part of post-delivery activities that provide modest performance gains.
- **Day 3, fine-tuning**: Indicates these best practices should be considered when small incremental improvements are needed to further tune the system.

The goal of this ranking system is to help customers to quickly decide which recommendations will provide the best value. For example, two or three best practices from the section on CPU optimization might provide most of the value for a customer depending on their goals. Investing the time to implement these best practices provides the greatest return. This approach can be taken for each layer of the database system until all Day 1, highly recommended best practices are completed. This methodology makes best practices unique and provides a way for customers to have the best return on investment.

Best practices are broad recommendations that apply to most SQL Server environments using Dell infrastructure, however every database application workload and environment is different, meaning the value of these best practices will vary from system to system. As with any configuration or change to a database system, the best approach is to validate the change before implementing the best practice on a production system. We recommend testing all best practices before implementing the changes in production.

# Solution architecture

## Physical architecture

The following figure shows an overview of the architecture used for the SQL Server best practices development.

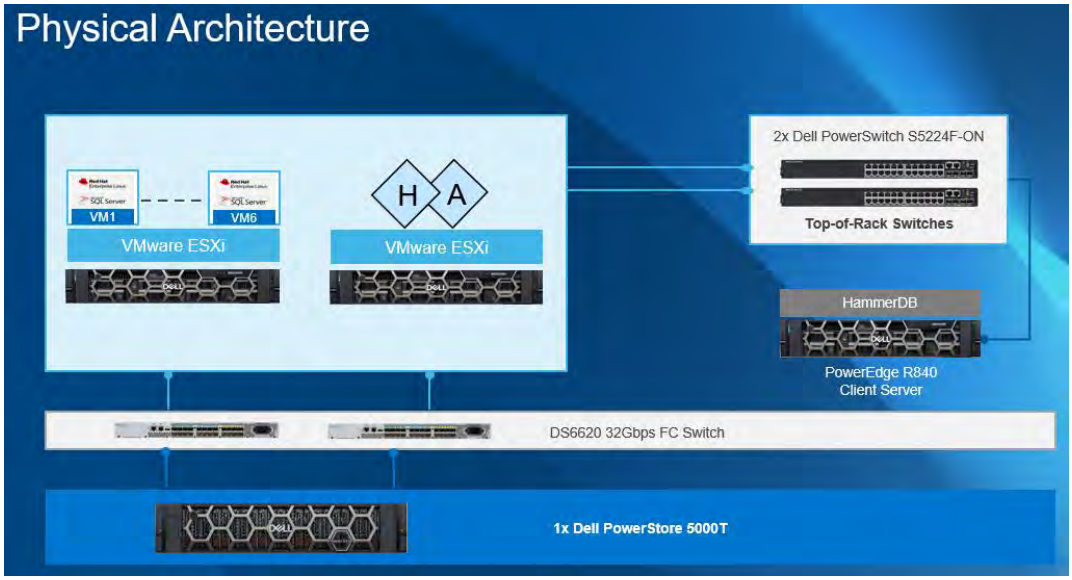


**Figure 1: Architecture Overview**

## Server

The architecture was designed to broadly represent the infrastructure customers use for their SQL Server databases. Dell PowerEdge R750xs servers were used for the compute layer. Each PowerEdge R750xs was configured the same way to ensure consistency. Two Intel® Xeon® Gold 6338 processors with 32 physical cores each, for 64 total cores, were used. The default server configuration enables logical processors which means 128 cores were available at the hypervisor level. The following table shows the detailed configuration:

**Table 2. PowerEdge R750xs Configuration details**

| | |
|---|---|
| Processors | 2 x Intel® Xeon® Gold 6338 32C CPU @ 2.00 GHz |
| Memory | 16 x 64 GB 3200MT/s memory, total of 1 TB |
| Network adapters | Embedded NIC: 1 x Broadcom BCM5720 1 GbE DP Ethernet (mgmt.)<br>Integrated NIC1: 1 x Broadcom Adv. Dual port 25 Gb Ethernet<br>NIC Slot 5: 1 x Mellanox ConnectX-5 EN 25 GbE Dual port Adapter |
| HBA | 2 x Emulex LPe35002 32 Gbps Dual Port Fibre Channel |

To learn more about the server go to the PowerEdge R750xs page and download the specification sheet.

## Fibre Channel Switch

Dell Connectrix switches were used for connectivity from the servers to the PowerStore 5000T storage array. The Connectrix DS-6620B is designed to support medium-to-large

sized database deployments. The Connectrix configuration used in our tests included eight active 32 Gb/s ports to optimize the connection to the PowerStore storage.

## Storage array

The PowerStore storage array is cloud-enabled and offers enterprise-rich data services like snapshots, replication, and many other features in a small footprint. The PowerStore storage array is highly tuned by default and does not require intensive manual tuning. In the PowerStore 5000T configuration used, there were 21 NVMe drives, each 1.92 TB in size.

Complete details of the PowerStore 5000T configuration are described in the following table:

**Table 3. PowerStore 5000T configuration details**

| | |
|---|---|
| Processors | 2 x Intel® Xeon® Gold 6130 CPU @ 2.10 GHz per Node |
| Cache size | 4 x 8.5 GB NVMe NVRAM |
| Drives | 21 x 1.92 TB NVMe SSD |
| Total usable capacity | 28.3 TB |
| Front-end I/O modules | 2 x Four-Port 32 Gb FC |

Many production database systems use dedicated infrastructure. In validating the SQL Server best practices, all the Dell infrastructure was reserved for the database. No parallel workloads were running and competing for CPU, network, and storage resources. Validating best practices in a dedicated environment helped to eliminate variables that might impact test results. We understand that many environments have been consolidated and challenges can arise in tuning one database system on shared infrastructure.

The implementation of best practices might improve performance in consolidated systems, but the positive gains may not be as significant due to the shared resources. Using these best practices can mitigate some challenges by integrating Day 1, Highly Recommended configuration practices as part of provisioning an SQL Server database. As the database ecosystem is transformed by using best practices, the overall system performance may rise, and consolidated systems might perform more efficiently. Best practices offer the enterprise the ability to deploy a database with the optimal design and ecosystems with the capability to drive improved efficiencies.

## Software architecture

This section outlines the various software that was used for this best practice development. They are:

- SQL Server 2019
- VMware vSphere 7.0.3
- Red Hat Enterprise Linux 8.5

## SQL Server 2019

Microsoft offers one of the most popular databases in the world. The SQL Server 2019 used for our best practices tests is a version of the database that many customers are using. The SQL Server 2019 has a vast array of features and capabilities. In our final validation tests the Engineering team tested changes to the database configuration to optimize performance.

## VMware

Virtualization of databases has been gaining momentum over the years. There are many advantages to virtualizing databases, including consolidation, agility, and ease of management. We chose to virtualize the SQL Server 2019 with VMware vSphere version 7.0.3. Virtualization adds another layer of configuration to the database. As virtualization impacts every part of the database infrastructure, we integrated VMware best practices to each physical layer, rather than having a dedicated section for virtualization. For example, storage best practices include recommendations for VMware ESXi Round Robin. This should simplify the program, as customers can review best practices as both physical and virtualization recommendations are presented for each part of the infrastructure.

### Red Hat Enterprise Linux

The Linux operating system has been widely accepted for running SQL Server databases. We used Red Hat Enterprise Linux version 8.5 for the tests, as this Linux flavor provides the stability, reliability, and security required for databases. The same best practice approach was taken with Red Hat Linux, as used for VMware. Linux best practices are integrated into each physical layer of recommendations. It is easy for customers to review Linux recommendations by reading CPU best practices or another physical layer.

The following table summarizes the software architecture used to validate the best practices:

**Table 4: Software architecture used**

| | |
|---|---|
| SQL Server | 2019 CU 16-15.0.4223.1 |
| Operating system | Red Hat Enterprise Linux 8.5 |
| VMware vSphere | VMware 7.0.3 with vCenter 7.0.3 |

There are many possible combinations for the software architecture. The goal for testing SQL Server, Red Hat Enterprise Linux 8.5, and VMware vSphere 7.0.3 was to have a design that applies to what database customers use today. Future best practices programs will address new versions of the database and advances in both the operating system and virtualization.

# Baseline configuration

This section covers details of the baseline configuration. A baseline configuration was used to determine the initial workload and provide a comparison for the first set of best practices. After each test, the database was refreshed with the best practices retained. This test methodology allowed the team to compare results as best practices were progressively added to the database.

### CPU reservation

To achieve a meaningful test result for a single R750xs server, we ran six virtualized databases in parallel. After several tests using HammerDB and an OLTP workload, we developed a baseline configuration. The following table shows the vCPU and memory allocation for each virtualized database:

**Table 5. vCPU and memory allocation**

| Resource reservation | Baseline configuration per virtual machine | Total across six virtualized databases |
|---|---|---|
| vCPU | 10 cores | 60 cores |
| Memory | 112 GB | 672 GB |

A VMware reservation is a guaranteed allocation of CPU or memory for the virtual machine. In the case of vCPU, each virtual machine received a ten-core reservation. With six total virtualized databases, the total vCPU reservation was 60 cores on the server (six virtual machines each with ten cores). The PowerEdge R750xs servers are two-socket servers. In the configuration for our tests, we used two Intel® Xeon® Gold 6338 CPUs. Each Intel® CPU has 32 cores, for 64 total physical cores. The total of 60 vCPU across the six virtual machines, resulted in a total of four unallocated physical cores which can be consumed by the hypervisor.

While the virtual machines used most of the physical cores by default, the PowerEdge R750xs servers enable logical processors. When logical processors are enabled, the hypervisor presents twice as many processor cores. In our configuration, the 64 physical processor cores are presented as 128 cores (64 x 2). Logical processors can boost performance by enabling more executions in parallel. In the case of our six virtual databases, we used nearly half of the available processor resources: 60 processor cores were reserved, and 128 cores were available.

### Memory reservation

The memory configuration per virtual machine included a reservation of 112 GB. A total of 672 GB of memory was reserved using six virtual machines. The total memory available in the server was 1 TB, indicating that the virtual machines used almost three-fourths of the available physical memory in the server. The goal in allocating memory to each virtual machine was to ensure that each VM fit into a physical non-uniform memory access (NUMA) node. NUMA is a multiprocessor configuration in which each physical CPU has a dedicated memory bank. Information can be retrieved faster from the local memory bank than having to pull the data from another memory bank. When a virtual machine's memory spans two memory banks, wait times are incurred, causing sub-optimal performance.

To prevent a sub-optimal NUMA design, each virtual machine was allocated 112 GB memory. This memory allocation enabled the use of the local memory bank only. The VMware ESXi hypervisor is responsible for the placement of memory on a processor and has been designed to optimize performance. By reserving memory that enables the virtual machine to use a local memory bank we optimized performance for all the virtual machines on the server.

## ESXi host connectivity

Each ESXi host (R750xs server) has two dual-port Fibre Channel HBAs. The PowerStore 5000T under test has two Fibre Channel front-end I/O modules (one per node). Each I/O module has four 32Gbps ports with a total of eight 32Gbps front-end ports. There are two Connectrix Fibre Channel switches. Each Connectrix switch is its own fabric. For connectivity, each ESXi host was zoned to all eight front-end ports of the PowerStore across the two switches. This configuration should be the default setup. Figure 2 below shows the SAN zoning configuration.

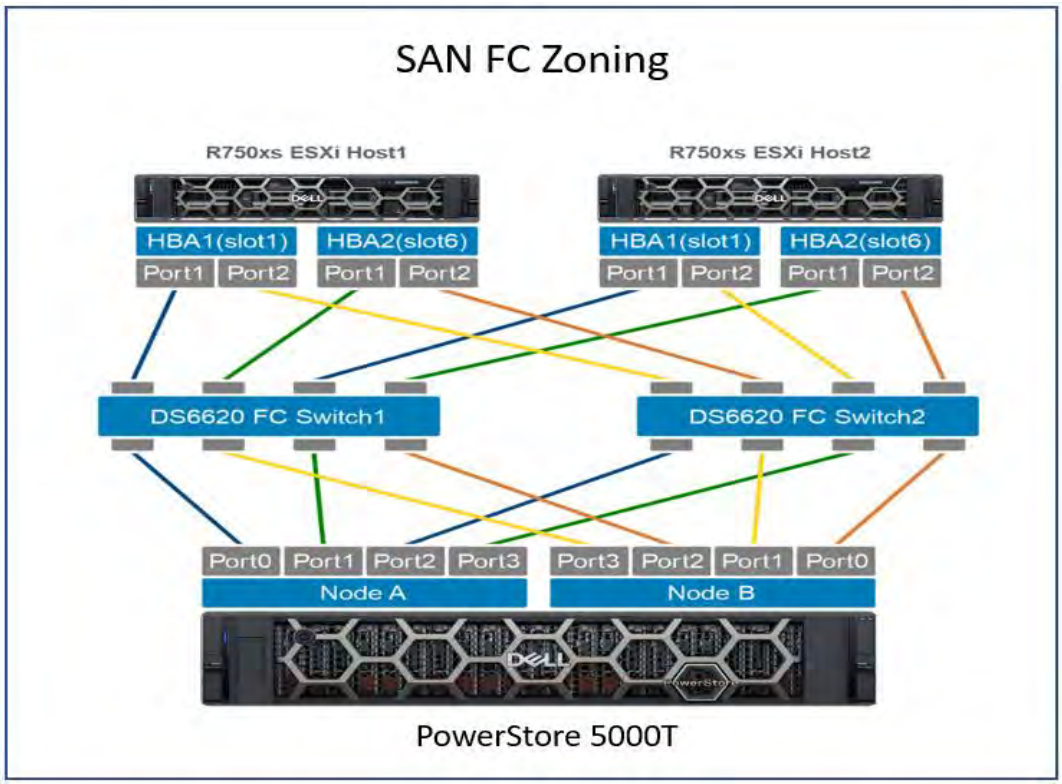


**Figure 2. SAN Zoning Configuration**

## Storage configuration

The PowerStore has the volume group feature, which enables the administrator to create a configuration that facilitates ease of management, efficient snapshots, and replication. For the baseline storage configuration, all six virtualized databases used the same design. Table 6 shows the configuration detail for the first virtualized SQL Server database.

**Table 6. Virtualized SQL Server database configuration**

| Volume Group | Volume Name | Number of LUNs | VMware Datastore | Volume size (GB) | Notes |
|---|---|---|---|---|---|
| sql_vm1_os | sql-vm1-os | 1 | sql-vm1-os-ds | 300 | Operating System |
| sql_vm1_db1 | sql-vm1-db1-data1 | 1 | sql-vm1-db1-data1-ds | 512 | Data |
| | sql-vm1-db1-log1 | 1 | sql-vm1-db1-log1-ds | 256 | Log |
| sql_vm1_db1_temp | sql-vm1-db1-temp-data1 | 1 | sql-vm1-db1-temp-data1-ds | 256 | TempData |
| | sql-vm1-db1-temp-log1 | 1 | sql-vm1-db1-temp-log1-ds | 128 | TempLog |
| sql_vm1_db1_backup | sql-vm1-db1-backup1 | 1 | sql-vm1-db1-backup1-ds | 1000 | Backup |

A naming convention using VMn and DB*n* allowed the database administrators to increment the value to create volume groups for all the databases. For example, the second copy of the database had a volume group name of SQL_VM2_DB2. Using this naming convention allowed the administration team to repurpose copies to the database quickly.

**Note**: Each LUN was configured as a VMware datastore.

For more information about PowerStore best practices for SQL Server, see Microsoft SQL Server Best Practices on PowerStore.

## SQL Server Configuration

The SQL Server instance configuration includes many default settings that reflect a new installation of the database. The following table includes a subset of the database configuration parameters to show the baseline settings.

**Table 7. Instance configuration parameters**

| Parameters | Values |
|---|---|
| Maximum Degree of Parallelism (MAXDOP) | 0 |
| Cost Threshold of Parallelism | 5 |
| Recovery mode | Full |
| Query Optimizer fixes | OFF |
| max worker threads | 0 |
| Min server memory | 98304 MB |
| Max server memory | 98304 MB |

The overall profile can be labeled as an entry-level database through a review of the configuration parameters. An entry-level database configuration provides a foundation to build upon best practices. Over the course of validating best practices the database configuration will change to optimize performance. Starting with a minimal configuration and adding best practices can help customers address database growth challenges using these validated recommendations.

# Storage best practice

This section discusses the PowerStore storage array best practice that was used for validation.

## PowerStore: Add Multiple LUNs

Storage configuration can directly influence database performance. This best practice validates the number of storage LUNs configured for SQL Server 2019 to optimize database performance.

**Table 8. Best Practice Category**

| Category | PowerStore Storage |
|---|---|
| Product | PowerStore 5000T |
| Type of best practice | Performance Optimization |
| Day and value | Day 1, Highly recommended |

**Overview**

To determine if a best practice provides value, the metrics were compared to a baseline configuration of the database to validate performance. In the baseline configuration for the database volumes, the SQL data, SQL log, temp data, temp log, and SQL backup file system each had one LUN. The below table shows the baseline PowerStore volume groups and LUNs configuration.

**Table 9. Baseline storage group configuration**

| Volume Group | Volume Name | Number of LUNs | VMware Datastore | Volume size (GB) | Notes |
|---|---|---|---|---|---|
| sql_vm1_os | sql-vm1-os | 1 | sql-vm1-os-ds | 300 | Operating System |
| sql_vm1_db1 | sql-vm1-db1-data1 | 1 | sql-vm1-db1-data1-ds | 512 | Data |
| | sql-vm1-db1-log1 | 1 | sql-vm1-db1-log1-ds | 256 | Log |
| sql_vm1_db1_temp | sql-vm1-db1-temp-data1 | 1 | sql-vm1-db1-temp-data1-ds | 256 | TempData |
| | sql-vm1-db1-temp-log1 | 1 | sql-vm1-db1-temp-log1-ds | 128 | TempLog |
| sql_vm1_db1_backup | sql-vm1-db1-backup1 | 1 | sql-vm1-db1-backup1-ds | 1000 | Backup |

The baseline storage configuration also includes separating database files into dedicated volume groups. This allows the database administrator (DBA) to only snapshot the database, without including unnecessary temp files and backup files. This configuration also provides space savings if replication is used.

The volume groups remain intact, but additional LUNs were added for the SQL User DB data area. Using multiple LUNs for the most active portion of a SQL Server database can improve performance. The primary benefit of using multiple LUNs is that the operating system creates an I/O queue path per LUN. The following table shows the optimized PowerStore storage configuration for the SQL Server database. In the optimized storage configuration, the SQL User DB Data area has four LUNs. Compared to the baseline configuration, this configuration has four times the number of I/O queue paths.

**Table 10. Optimized PowerStore storage configuration**

| Volume Group | Volume Name | VMware Datastore | Volume size (GB) | Notes |
|---|---|---|---|---|
| sql_vm1_os | sql-vm1-os | sql-vm1-os-ds | 300 | Operating System |
| sql_vm1_db1 | sql-vm1-db1-data1 | sql-vm1-db1-data1-ds | 160 | SQL Data/Log |
| | sql-vm1-db1-data2 | sql-vm1-db1-data2-ds | 160 | |
| | sql-vm1-db1-data3 | sql-vm1-db1-data3-ds | 160 | |
| | sql-vm1-db1-data4 | sql-vm1-db1-data4-ds | 160 | |
| | sql-vm1-db1-log1 | sql-vm1-db1-log1-ds | 256 | |
| sql_vm1_db1_temp | sql-vm1-db1-temp-data1 | sql-vm1-db1-temp-data1-ds | 256 | Temp Data/Log |
| | sql-vm1-db1-temp-log1 | sql-vm1-db1-temp-log1-ds | 128 | |
| sql_vm1_db1_backup | sql-vm1-db1-backup1 | sql-vm1-db1-backup1-ds | 1000 | Backup |

Once the additional LUNs are created, these LUNS can be formatted with a file system and mounted. Multiple SQL Server datafile can now be spread across all four data mountpoints.

**Recommendation**

Based on our test results, the overall database performance increased by having multiple LUNs for the SQL Data file system. This best practice of increasing the number of LUNs is highly recommended and should be considered as a Day 1 practice, as part of initial data provisioning.

For more information about PowerStore best practices with SQL Server database, see SQL Server Database Best Practices on PowerStore.

## VMware: Use vVols for SQL Server 2019

Virtual Volumes (vVols) is a new VMware technology that is supported starting in vSphere 6.0. This best practice validates that using VMware vVols for SQL Server 2019 optimizes database performance and management.

**Table 11. Best practice category**

| Category | VMware vSphere |
|---|---|
| Product | VMware vSphere 7.3 |
| Type of best practice | Performance Optimization |
| Day and value | Day 1, Highly recommended |

**Overview**

vVols are VMDK granular storage entities exported by storage arrays. vVols are exported to the ESXi host through a small set of protocol end-points (PE). Protocol Endpoints are part of the physical storage fabric, and they establish a data path from virtual machines to their respective vVols on demand. Storage systems enable data services on vVols.

One of the key features of vVols is that it allows administrators to use storage policy-based management (SPBM) for their environment. This enables you to align application needs with the appropriate storage resources in an automated manner.

**Recommendation**

Using vVols for simplified storage management automates storage provisioning and space reclamation. vVols performance should be on par with VMFS. However, vVols provides better performance over VMFS when it comes to snapshotting because it uses the array snapshot functionality. In a SQL Server environment, snapshot is frequently used, therefore, vVols should be considered over VMFS.

Based on our results, this practice is considered as a Day 1, Highly Recommended best practice.

**Additional resources**

Understanding Virtual Volumes (vVols) in VMware vSphere 6.7/7.0

## PowerStore: Volume Performance Policy

The volume performance can be managed in the PowerStore storage array by adjusting the performance policy. Selecting a performance policy sets a relative priority for the volume by using a share based QoS system that prioritizes the higher performance policy when resources are constrained. The performance policy does not impact system behavior unless some volumes have a low performance policy setting and other volumes are set to medium or high.

**Table 12. Best practice category**

| Category | PowerStore Storage |
|---|---|
| Product | PowerStore 5000T |
| Type of best practice | Performance Optimization |
| Day and value | Day 3, Fine Tuning |

**Overview**

A performance policy specifies I/O performance requirements for PowerStore storage resources. PowerStore provides three predefined performance policies:

- High
- Medium (default)
- Low

To determine if changing the volume performance policy provides value, it was tested against the baseline using the default value (medium). In the baseline configuration for the database volumes, the SQL data, SQL log, temp data, temp log, and SQL backup volumes were configured with a medium-performance policy. For this test case, the SQL

Data volumes were configured with a high-performance policy. The following table shows which volumes were configured with a high-performance policy.

**Table 13. Volume performance policy settings**

| Volume Group | Volume Name | VMware Datastore | Volume Performance Policy | Notes |
|---|---|---|---|---|
| sql_vm1_db1 | sql-vm1-db1-data1 | sql-vm1-db1-data1-ds | High | SQL Data/Log |
| | sql-vm1-db1-data2 | sql-vm1-db1-data2-ds | High | |
| | sql-vm1-db1-data3 | sql-vm1-db1-data3-ds | High | |
| | sql-vm1-db1-data4 | sql-vm1-db1-data4-ds | High | |
| | sql-vm1-db1-log1 | sql-vm1-db1-log1-ds | High | |
| sql_vm1_db1_temp | sql-vm1-db1-temp-data1 | sql-vm1-db1-temp-data1-ds | High | Temp Data/Log |
| | sql-vm1-db1-temp-log1 | sql-vm1-db1-temp-log1-ds | High | |
| sql_vm1_db1_backup | sql-vm1-db1-backup1 | sql-vm1-db1-backup1-ds | Medium | Backup |

**Recommendation**

Changing the performance policy to high for the SQL Data, SQL log, SQL Temp DB, and SQL Temp Log volumes resulted in no performance gain. Considering these results, this best practice is considered as a Day 3, Fine Tuning practice and benefit only if there are volumes with mixed performance policy on the array.

**Additional resources**

Dell PowerStore Manager Overview, Storage: Volumes

Dell PowerStore: Best Practices Guide, Block storage resources

Dell EMC PowerStore Configuring Volumes

# Server best practices

For the server configuration best practices, we compared the results to the previous test case and ensure that the performance does not decrease.

Dell PowerEdge R750xs server can be optimized by tuning the sub-components within the server hardware such as CPU and memory. As part of the PowerEdge server optimization, we tested multiple settings within the BIOS and compared it the default settings.

## PowerEdge R750xs: Database Optimized Performance Workload Profile

By default, the Database Optimized Performance (DOP) profile is not used on the PowerEdge R750XS server. In this best practice, we selected the DOP profile in the BIOS to evaluate its impact on performance.

**Table 14. Best practice category**

| Category | PowerEdge |
|---|---|
| Product | PowerEdge R750xs server |
| Type of best practice | Performance Optimization |
| Day and value | Day 3, Fine Tuning |

### Overview

A BIOS profile is an easy way to apply firmware values that optimize the server. The following list describes BIOS settings that are automatically changed by selecting the DOP Profile.

1. **Virtualization Technology (VT)**: Default setting enabled changed to disabled with the DOP profile.
   When VT is enabled, firmware settings are configured for enhanced support of hypervisors, meaning VT leads to greater integration with virtualization. The DOP profile disables VT, as many databases are not virtualized. As the SQL Server databases were virtualized in our best practice, the VT setting was changed back to enabled after being disabled by the DOP profile.
2. **System Profile:** Default setting Performance per Watt (DAPC profile) changed to Performance.
   The DAPC profile is Dell's proprietary implementation of dynamic processor management. Changes to the DOP profile alters the system profile from the default to Performance. Instead of the server balancing performance to drive a cost savings, the system is configured for maximum performance.
3. **CPU Power Management:** Default System Demand-Based Power Management (DBPM) changed to maximum performance.
   CPU Power Management controls use of processor performance states. By default, CPU Power Management uses DBPM which enables processor operating frequency and voltage to be dynamically adjusted. The DOP profile changes CPU Power Management to a static state of maximum performance.

Other system profile changes include the following:

- **C1E**: default setting enabled changed to disabled. This change disables DIMM CKE power down and QPI power management.
- **C States**: default setting enabled changed to disabled. This change disables C-states and DIMM self-refresh.
- **Uncore Frequency**: Default setting Dynamic Uncore Frequency scaling (UFS) changed to Maximum. Setting the frequency to Maximum enables best performance of the cores and uncore during runtime.
- **Energy Efficient Policy**: By default, the "Balanced Performance" policy is used. This setting was changed to "Performance," which instructs the processors to operate at the highest performance level.

- **CPU Interconnect Bus Link Power Management**: default setting enabled changed to disabled.
- **PCI ASPM L1 Link Power Management**: default setting enabled changed to disabled.

Overall, DOP benefits from setting several firmware settings to maximum performance. The only exception made in our validation tests was changing the VT setting back to the default value of enabled. However, we recommend keeping the DOP profile change for VT to disabled for customers with physical (non-virtual) database deployments.

**Recommendation**

Our test finding from validating the DOP setting showed a slight improvement in performance for the database, therefore, it is considered as a Day 3, Fine Tuning recommendation and applied only if necessary. The below metrics showed a slight increase in:

- New Order per Minute (NOPM)
- Transaction per Minute (TPM)
- PowerStore IOPS
- Server CPU utilization

**Additional Resources**

Setting up BIOS on 15th Generation (15G) Dell PowerEdge Servers

## PowerEdge R750xs: Dell Controlled Turbo

The PowerEdge servers come with the Dell Controlled Turbo feature which controls the turbo engagement of the CPUs.

**Table 15. Best practice category**

| Category | PowerEdge |
|---|---|
| Product | PowerEdge R750xs |
| Type of best practice | Performance Optimization |
| Day and value | Day 3, Fine Tuning |

**Overview**

The Dell Controlled Turbo setting allows the CPUs to run at maximum speed. This option only works if the system profile is set to performance in the BIOS. In this best practice, we enabled the Dell Controlled Turbo with the Database Optimization Performance profile.

**Recommendation**

After enabling the Dell Controlled Turbo option, the following test metrics gained a slight increase in performance and considered as a Day 3, Fine Tuning best practice.

- New Order per Minute (NOPM)
- Transactions per Minute (TPM)
- PowerStore IOPS

- Server CPU utilization

**Additional resources**

Enable Dell Controlled Turbo on 15th Generation (15G) Dell PowerEdge Server

## PowerEdge R750xs: Enabling Processor x2APIC Support

By default, Intel's Advanced Programmable Interrupt Controller (x2APIC) BIOS setting is disabled on the PowerEdge R750xs server. In this best practice, we enabled the x2APIC setting to evaluate the impact on performance.

**Table 16. Best practice category**

| Category | PowerEdge |
|---|---|
| Product | PowerEdge R750xs server |
| Type of best practice | Performance Optimization |
| Day and value | Day 3, Fine Tuning |

**Overview**

The x2APIC is designed to improve efficiency in multiprocessor systems. An interrupt is a request by the software for the processor to respond to an event. For example, if an operating system requests that the processor suspend current activities and this request is accepted by the processor, the state of the current activities is saved and the new request is processed.

The x2APIC is Intel's most recent and advanced programmable interrupt controller. Enhancements to x2APIC include support for more processors and improved performance. The PowerEdge R750xs used for testing best practices had two Intel® Xeon® Gold 6338 processors each with 32 cores, for a total of 64 cores in the server. VMware vSphere 7.0.3 was used to virtualize the SQL Server databases. Enabling x2APIC should enable efficiencies for the PowerEdge multi-processor system and optimize interrupt management of virtual machines.

**Recommendation**

Overall, the best practice of enabling the x2APIC in BIOS can provide a slight improvement in the system performance. Enabling x2APIC considered as a Day 3, Fine Tuning recommendation with these considerations:

- For new PowerEdge servers, consider enabling the x2APIC BIOS setting if the virtualization and/or operating system supports the new interrupt controller.
- For existing PowerEdge servers, consult support documentation and consider enabling this BIOS setting at the next maintenance window.

**Additional Resources**

Setting up BIOS on 15th Generation (15G) Dell EMC PowerEdge Server

## PowerEdge R750xs: Disable Unused Integrated Devices

By default, PowerEdge servers have USB and other ports enabled. In this best practice, USB and serial ports are disabled to reduce security risks.

**Table 17. Best practice category**

| Category | PowerEdge |
|---|---|
| Product | PowerEdge R750xs |
| Type of best practice | Performance Optimization |
| Day and value | Day 3, Fine Tuning |

**Overview**

Although we did not expect a significant performance improvement from disabling USB ports on the servers, it is still a good security practice. The following ports were disabled in BIOS:

- User accessible USB ports
- Internal USB ports
- iDRAC Direct USB port
- Serial communication

**Recommendation**

Although disabling unused integrated devices showed no improvement, it is still considered as a Day 3, Fine Tuning best practice for security purposes.

**Additional Resources**

Setting up BIOS on 15th Generation (15G) Dell PowerEdge Server

## VMware vSphere: Power Management Policy

The ESXi Power Policy enables the VMware vSphere administrator to assign a power profile to the virtualized database. In this best practice, we select the high-performance power policy.

**Table 18. Best practice category**

| Category | VMware vSphere |
|---|---|
| Product | VMware vSphere 7.0.3 |
| Type of best practice | Performance Optimization |
| Day and value | Day 3, Fine Tuning |

**Overview**

There are four CPU Power Management Polices that can be selected for a virtual machine:

- **High Performance**: maximize performance and disable power management features.
- **Balanced (Default)**: use power management features that will minimally impact performance.
- **Low Power**: emphasis on power management features to minimize energy consumption over performance.

- **Custom**: user-defined power management policy.

Many databases support critical business functions, meaning database performance is an enterprise priority. In this best practice, the CPU Power Management Policy was changed from the default setting of balanced to high performance to ensure the best possible performance.

**Recommendation**

The results from changing the CPU Power Management Policy from balanced to high performance showed a slight increase in performance for the database.

Although changing the CPU Power Management Policy to high performance did not lead to significant performance improvements, it is still a good practice for production databases and considered a Day 3, Fine Tuning recommendation. By changing the power management policy to high performance, the administrator ensures that no power management features will impact future performance.

**Additional Resources**

VMware vSphere Documentation: Select a CPU Power Management Policy

# Consolidated: RHEL and SQL Server 2019 best practices

This section discusses the combination of Linux best practices and SQL Server best practices. During our tests, we combined both the Linux and the SQL Server layer best practices and compared to the previous best practices test results. The following table lists the consolidated best practices for Linux and SQL Server.

**Table 19. RHEL and SQL Server 2019 best practices**

| Best Practice | Category | Product | Best Practice Type | Day and Value |
|---|---|---|---|---|
| SQL Server Tuned Profile | Operating System | RHEL 8.5 | Performance Optimization | Day 1, Highly Recommended |
| Access Time | Operating System | RHEL 8.5 | Performance Optimization | Day 1, Highly Recommended |
| CPU Affinity | SQL Server | SQL Server 2019 | Performance Optimization | Day 1, Highly Recommended |
| Forced Unit Access | SQL Server | SQL Server 2019 | Performance Optimization | Day 1, Highly Recommended |

## Red Hat Enterprise Linux: SQL Server Tuned Profile

The “tuned-adm” command line tool enables switching between different tuning profiles in Linux. In this best practice, we used the “tuned-adm” tool to switch to an optimized Linux profile for SQL Server.

**Overview**

The Linux operating system provides a system tuning tool called “tuned.” This tool is a profile-based tool that uses udev device manager. The “tuned-adm” tool enables the Linux administrator to switch between profiles for the operating system. Pre-defined profiles enable the application of Linux configuration settings in a downloadable package.

In this best practice, the TUNED-PROFILES-MSSQL was downloaded and installed in the Linux OS. This resulted in the following changes in the tuned SQL Server profile:

- FORCE_LATENCY = 5

Certain configuration parameters that adjust the PM QoS CPU DA latency according to the CPU load. For example, the force latency parameter enables the user or in this case the SQL Server profile to force a specific value. For more information about this CPU configuration parameter and similar parameters, see the Red Hat Performance Tuning Guide.

- VM.TRANSPARENT_HUGEPAGES set to ALWAYS

Transparent HugePage Support (THP) automates the promotion and demotion of memory pages. Promotion is the automation from small memory pages to larger memory page sizes and demotion is the automation from larger memory pages to smaller memory pages. SQL Server uses transparent HugePages to optimize performance thus, this parameter is set to ALWAYS.

- VM.SWAPPINESS was set to 1

Swap space is used by the Linux kernel as virtual memory. The swappiness setting defines how aggressively the kernel will swap memory pages to disk. By changing VM.SWAPPINESS to 1, the kernel is instructed to swap as little as possible to disk and keep more memory pages in RAM.

- VM.DIRTY_BACKGROUND_RATIO was set to 3

The DIRTY_BACKGROUND_RATIO is the percentage of memory that contains free pages and reclaimable pages that can be filled with dirty pages before the pages are written to disk. Databases can aggressively use the kernel page cache which lowers the background ratio forcing the kernel to flush the dirty pages to disk more frequently.

- VM.DIRTY_RATIO set to 80

The DIRTY_RATIO is the percentage of total system memory that contains dirty pages before the pages are written to disk. In setting dirty_ratio to 80, the kernel is instructed to use 80 percent of total system memory for dirty pages before having to flush the data to disk.

- VM.DIRTY_EXPIRE_CENTISECS is set to 500

The DIRTY_EXPIRE_CENTISECS define in centisecs (one hundredth of a second) when data is old enough to be written to disk. By setting DIRTY_EXPIRE_CENTISECS to 500 a dirty page can stay in memory for 500 centisecs before it is written to disk.

- VM.DIRTY_WRITEBACK_CENTISECS is set to 100

The DIRTY_WRITEBACK_CENTISECS parameter defines the frequency that the kernel will check to write dirty pages to disk. In this case, a value of 100 centisecs instructs the kernel to check every 100 centisecs to write dirty pages to disk.

- VM.MAX_MAP_COUNT is set to 1600000

The MAX_MAP_COUNT defines the number of memory map areas a process may include. For example, databases use memory extensively which means increasing this value results in more memory map areas available for the database to use.

This best practice also included network and kernel settings updated by the MSSQL tuned profile. The following list describes the network and kernel settings that were updated by the MSSQL tuned profile:

- NET.CORE.RMEM_DEFAULT = 262144
- NET.CORE.RMEM_MAX = 4194304
- NET.CORE.WMEM_DEFAULT = 262144
- NET.CORE.WMEM_MAX = 1048576
- KERNEL.NUMA_BALANCING = 0

The configuration changes made to the Linux operating system fine-tuned memory, network, and kernel configurations resulted in moderate performance improvements including more efficient use of system resources.

**Additional Resources**

Performance best practices and configuration guidelines for SQL Server on Linux

## Red Hat Enterprise Linux: Last Accessed Time (atime)

In this best practice, we changed the default Linux XFS mount option per Red Hat guidelines.

**Overview**

Atime is a specification of the Linux operating system tracks the last time a file was accessed. For example, when the Linux operating system reads data from a data file, the access time is updated. Database frequently access data files and updating the access time with each access represents an overhead on the operating system possibly impacting performance.

It is possible to specify no access time (noatime), meaning the Linux operating system will not update the access time for files. The goal of this best practice was to reduce associated overhead and increase efficiency by setting the operating system to noatime.

**Additional resources**

Optimizing RHEL 8 for Real Time for low latency operation

## SQL Server 2019: CPU Affinity

In this best practice, we used processor affinity to optimize SQL Server performance. The goal is to define a range of processors for SQL Server to reduce thread migration and improve performance.

**Overview**

CPU affinity is the capability to define which processor cores the SQL Server engine will use for multitasking. By assigning processor cores, SQL Server only uses those processors for multitasking database operations. This can improve performance by reducing processor reloads and thread migration across processors.

The virtual machine supporting SQL Server has 10 allocated vCPUs. For this reason, we implemented the following configuration changes as part of this best practice:

- Configured SQL Server Processor Affinity to use processors 0 through 9, for a total of 10

- Validated that processor 0 through 9 belong to a single NUMA node

**Additional Resources**

affinity mask Server Configuration Option

### SQL Server 2019: Forced Unit Access

In this best practice, we implemented MS-SQL Forced Unit Access (FUA) to improve write performance of the SQL Server database.

**Overview**

FUA is a bit that is sent with writes to storage that improves database performance and durability. FUA was not supported prior to Linux kernel 4.18, which resulted in device level flushing that was inefficient and could impact storage I/O performance. SQL Server addressed storage inefficiency by minimizing flushes, but there was still room for improvement.

Starting with the Linux kernel 4.18, the XFS file systems support the FUA bit. The 4.18 Linux kernel allows the XFS file system and a device supporting FUA writes to avoid the additional block device flush requests. Using the Linux XFS file system and the FUA bit can significantly improve performance for write intensive workloads.

**Additional Resources**

SQL Server On Linux: Forced Unit Access (Fua) Internals, Microsoft Ignite

### Recommendation

Results from validating the consolidated best practices of the Linux operating system and SQL Server 2019 showed a significant performance improvement. Test findings showed increases in:

- New Order per Minute (NOPM)
- Transaction per Minute (TPM)
- PowerStore IOPS

Based on these results, these best practices are considered as Day 1, Highly Recommended and should be part of the initial setup.

# Conclusion

The Dell Integrated Solutions team developed, validated, and consolidated the SQL Server 2019 database best practices on 15G PowerEdge servers with Intel® processors, PowerSwitch, and PowerStore 5000T storage. Improving customer outcomes continues to be a key tenet across all SQL Server solutions and the validated best practices demonstrate how Dell Technologies is committed to ensuring customer success.

These SQL Server 2019 best practices represent a comprehensive, consolidated set of recommendations that include everything from the physical infrastructure to the software stacks. For our customers, this simplifies best practice identification and offers significant time savings in optimizing database performance.

We categorized the best practices using performance metrics from HammerDB, the database, servers, and storage. The goal was to categorize best practices based upon their value in terms of improving performance.

By grouping best practices by their return on performance, we streamline how customers can achieve the best results in the shortest time. This reduces complexity and risk by

providing a plan our customers can follow: Day 1 followed by Day 2 and, if needed, Day 3 best practices.

## We value your feedback

Dell Technologies and the authors of this document welcome your feedback on the solution and the solution documentation. Contact the Dell Technologies Solutions team by email.

**Author:** Tom Dau

**Contributors**: Ava English

**Note**: For links to additional documentation for this solution, Dell Technologies Solutions Info Hub for SQL Server.