# Modernize Your Microsoft SQL Server Platform and Accelerate Deployments

## Validated Design with Dell PowerEdge, PowerSwitch and PowerStore using VMware vSphere virtualization

May 2022

H19198

Design Guide

### Abstract

This design guide describes the best practices for designing and implementing a VMWare-based virtualized SQL Server 2019 solution using Dell PowerStore 1000T storage, Dell PowerEdge servers R750, VMware vSphere 7, and Red Hat Enterprise Linux Operating system 8.

Dell Technologies Solutions

Dell
Validated Design

Copyright

# Contents

# Chapter 1 Introduction

This chapter presents the following topics:

Notice: Dell Technologies announced the availability of new PowerStore models in July 2022. The new PowerStore models offer greater performance, more features, and capacity compared to previous storage arrays. In this paper we optimized Microsoft SQL Server on the PowerStore 1000T. All the recommendations and best practices in this paper also apply to the new PowerStore arrays.

# Solution introduction

## Executive summary

As the number of applications and databases increase within businesses, infrastructure also increases. This creates database silos, management complexities, and increased costs, which can hinder businesses from accelerating their Microsoft SQL Server and IT modernization strategies.

Consolidation has historically answered many of the challenges that accompany database sprawl and data storage silos. Consolidation strategies must ensure the availability and performance of business-critical applications. The introduction of faster, more powerful systems and new storage technology has made it possible for businesses to consolidate databases with greater confidence.

However, IT staff may be unable to perform a consolidation strategy as quickly as management requires, as it can take months to plan, procure, deploy, integrate, test, and fine-tune the new consolidated architecture. A 2022 ESG study, Accelerate IT Modernization with Dell Technologies Validated Designs, revealed that 90% of the surveyed IT decision makers reported that their organizations must move faster when deploying applications, infrastructure, and services than three years ago.

A solution that is pre-designed and tested to optimize SQL Server performance deployments and supports consolidation and upgrade modernization strategies can help reduce the burden on staff and strained budgets.

## Document purpose

This document outlines the power of a Dell integrated infrastructure for SQL Server, inclusive of design concepts and validated configurations. The focus is to deploy an optimized environment through hardware, performance, and vetted best practices. This document focuses on all aspects of the underlying infrastructure, including database, server, VMware, storage, and network.

## Audience

This document is intended for solution architects, data scientists, database administrators, VMware administrators, and storage administrators.

# Chapter 2 Architecture and baseline configuration

This chapter presents the following topics:

This chapter provides architectural information for each best practice. The architecture includes different aspects of the solution such as components of physical layer, software layer, and baseline configuration. Baseline configuration provides configuration setting information related to CPU, memory, storage, and SQL Server.

# Physical architecture

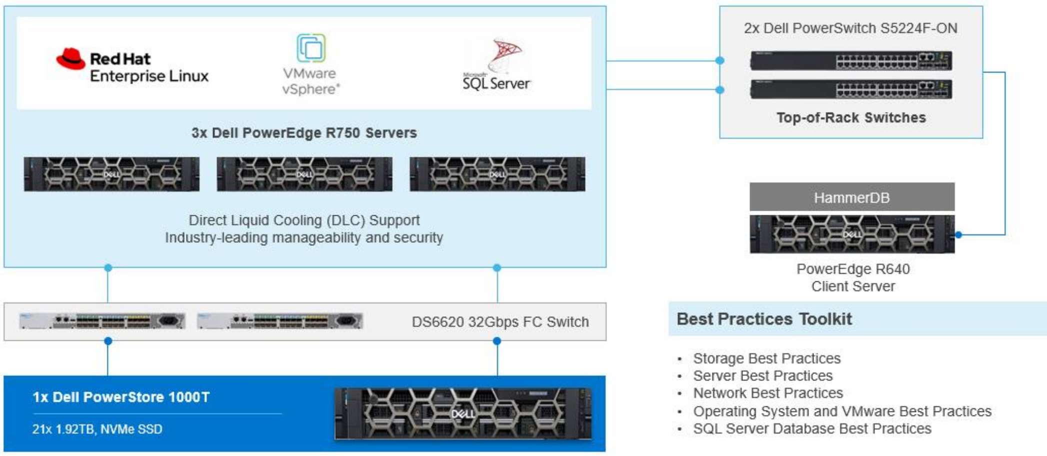


**Figure 1. Physical architecture**

The physical architecture was designed to represent the infrastructure that customers use for their SQL Server databases. Figure 1 shows that the compute layer includes three Dell PowerEdge R750 servers. Each PowerEdge R750 was configured identically to ensure consistent test results. Two Intel Xeon Gold 6334 processors with eight physical cores each were used, for a total of 16 cores. The default server configuration enables logical processors, which means at the hypervisor layer 32 cores were available. A detailed configuration is shown in the following table:

**Table 1. PowerEdge R750 server configuration**

| Component | Detail |
|---|---|
| **Processors** | **2 x Intel Xeon Gold 6334 processors** |
| Memory | 16 x 64 GB 3200 MHz Memory, Total of 1 TB |
| Network Adapters | Broadcom Gigabit Ethernet BCM5720 (embedded NIC)<br>Broadcom BCM57504 25GbE QP NetXtreme-E OCP |
| HBA | 2 x Emulex LPe35002-M2-D 2Port 32 Gb |
| Storage controller and hard drives | BOSS-S1 controller card + with 2 M.2 480 GB (RAID 1) |

To learn more about the server, see the [Dell PowerEdge R750 technical guide](#).

Dell Connectrix switches provided connectivity from the servers to the PowerStore storage array. The Connectrix DS-6620B is designed to support medium-to-large database deployments. The Connectrix configuration used in these tests included 20 active 32Gb/s ports to optimize the connection to the PowerStore storage.

A PowerStore 1000T storage array was used to validate the best practices. The PowerStore 1000T offers entry-level data services like snapshots, replication, and other features in a small footprint. The PowerStore configuration included 21 NVMe drives, each 1.9 TB in size. The PowerStore also had 2 x NVMe NVRAM drives, each 8.5 GB in size. Full details of the PowerStore 1000T configuration are described in the following table:

**Table 2. PowerStore 1000T configuration**

| Component | Detail |
|---|---|
| **Processors** | **2 x Intel(@) Xeon(R) Silver 4108 CPU @ 1.80GHz per Node** |
| Nodes | 2 Nodes (1 appliance with 2 nodes) |
| Cache size | 192 GB per node |
| Drives | 21 x 1.9 TB NVMe |
| Total usable capacity | 28.35 TB |

Many production database systems use dedicated infrastructure. To validate SQL Server best practices, all Dell infrastructure was reserved for the database. No parallel workloads were running and competing for CPU, network, and storage resources. This approach of validating best practices in a dedicated environment eliminated extraneous variables that might impact test results. Many of our customers' environments have been consolidated and challenges can arise when tuning one database system on shared infrastructure.

## Software architecture

We chose Microsoft SQL Server for our best practice tests because it is a commonly used Microsoft database by our customers. The SQL Server database also has a vast array of features and capabilities. In the final validation tests, we tested changes to the database configuration that optimize performance.

Database virtualization 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 database with VMware vSphere version 7.0U3. This requires another layer of configuration to the underlying hardware.

The Linux operating system has been widely accepted for running SQL Server databases. Red Hat Enterprise Linux version 8.5 was used for the tests as this Linux flavor provides stability, reliability, and security required for databases.

During the validation, all best practices correspond to different layers in the solution and were applied simultaneously. The database performance metrics, transactions per minute (TPM) and new orders per minute (NOPM) were collected before and after implementing the best practices. The following table summarizes the software architecture used to validate the best practices.

**Table 3. Software Details**

| Software | Version |
|---|---|
| **SQL Server** | **2019 CU15** |
| Operating system | Red Hat Enterprise Linux 8.5 |
| VMware vSphere | VMware 7.0.3d with vCenter 7.0 |

There are many possible combinations for the software architecture. In testing with SQL Server, Red Hat 8.5, and VMware vSphere 7.0 the goal was to have a design that applies to what database customers use today and in the near future.

# Baseline configuration

A baseline configuration determined the initial workload and enabled comparison with the first set of best practices.

## CPU reservation

We developed a baseline configuration for this solution, that included deployment parameters for the entire stack. This included the amount of vCPU and memory for each of the VMs, which we determined from previous experiences with similar setups and refined by performing additional performance tests and evaluations. We performed this configuration using a HammerDB benchmarking tool simulating an OLTP workload of 10,000 warehouses (approximately 1 TB in size) for each database per virtual machine.

For the final baseline configuration, two virtualized databases were run in parallel. The following table shows the vCPU and memory allocation for each virtual machine:

**Table 4. vCPU and memory allocation**

| Resource reservation | Baseline configuration per virtual machine | Total across two virtualized databases |
|---|---|---|
| vCPU | 6 cores | 12 cores |
| Memory | 384 GB | 768 GB |

The virtual machines included 6 virtual cores and 384 GB of memory. With two virtualized databases, the total vCPU was 12 cores on the server. The PowerEdge R750 servers are two socket servers. For these tests, two Intel Xeon Gold 6334 processors were used. As each Intel processor has eight cores, these tests included a total of 16 physical cores.

Although the virtual machines used most of the physical cores by default, PowerEdge R750 servers enable logical processors. When logical processors are enabled, the hypervisor presents twice as many processor cores. Logical processors can boost performance by enabling more performance in parallel. In the case of our two virtual databases, we assigned less than half of the available processor resources; we allocated 12 vCPUs and vCPU were available.

## Memory configuration

For the two virtual machines, 768 GB of memory was allocated (384 GB each). The total memory available in the server was 1 TB. This indicates that the virtual machines used less than half of the available physical memory in the server. In allocating memory to each virtual machine, the goal was to ensure that each virtual machine fit into a physical non-uniform memory access (NUMA) node and would not be NUMA-wide. 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 data retrieval from a remote memory bank. When a virtual machine's memory spans two memory banks it is called a NUMA-wide configuration and wait times increase and cause suboptimal performance. See VMware's Architecting Microsoft SQL Server on VMware vSphere Best Practices Guide for more information about NUMA considerations.

The VMware ESXi hypervisor is responsible for the management of memory and has been designed to optimize performance and if possible, prevent NUMA-wide configurations. Allocating 384 GB of memory to each virtual machine  optimized performance by enabling the use of the local memory bank only.

## Storage configuration

PowerStore features the ability to create Volume Groups that enable administrators to create a configuration that facilitates ease of management, efficient snapshots, and replication. For the baseline storage configuration, both virtualized databases used the same design. Table 5 shows the configuration detail for the first virtualized SQL Server database:

**Table 5. PowerStore configuration of first virtualized SQL Server database**

| Volume group | | Volumes | VMware volumes | Volume size (TB) | Notes |
|---|---|---|---|---|---|
| Tarragon DBs | TarragonDB1 | VM1_Data | 1.5 | All database files | |
| | | Tarragon DB2 | VM2_Data | 1.5 | |

Using this approach, the engineering team could create copies of the SQL Server database quickly and easily. For this reason, they selected this configuration for the default baseline configuration.

## SQL server configuration

The SQL Server database configuration includes many default settings that reflect a new installation of the database. Table 6 lists a subset of the database configuration parameters to show the baseline settings.

**Table 6. Database baseline configurations**

| 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 | 0 MB |
| Max server memory | 2147483647 MB |

The overall profile is the baseline database configuration. This configuration provided a foundation for the best practice tests.

# Chapter 3 Best practices and configuration guidelines

This chapter presents the following topics:

This chapter discusses the following best practice programs:

1. PowerStore Best Practices
2. PowerEdge Configuration Best Practices
3. Red Hat OS and VMware Best Practices
4. SQL Server Configuration Best Practices

# PowerStore best practices

For best practices on the storage layer, we had to

1. Additional FC ports
2. Adding Storage Volumes
3. Adding PVSCSI Controllers
4. Storage Volume Priority

## Additional FC ports

PowerStore storage arrays use appliances that enable customers to add power, performance, and capacity to an existing array. Each PowerStore appliance has two nodes. These nodes support multiple functions including front-end I/O modules. The front-end I/O modules provide connectivity to the array using various interfaces including Fibre Channel SCSI, Fibre Channel NVMe, iSCSI, and others. In the best practice validation tests, we used only the Fibre Channel interface.

Four PowerStore front-end I/O modules were allocated to each PowerEdge R750 servers. In the baseline configuration, we used one of the total four front-end I/O modules. Figure 2 shows the ports and nodes that we used. The two other front-end modules were reserved for replication. Each front-end module was a 32 Gbps Fibre Channel connection, and the baseline connection consisted of six modules.

In the baseline configuration, we used two of the total four front-end I/O modules per PowerEdge R750 server which limits the paths and the bandwidth available between the PowerEdge servers and the PowerStore storage array.

For this best practice, the two front-end modules were re-zoned and two additional front-end I/O modules were added for each of the PowerEdge servers. We also recreated our Hosts and Host Groups with the updated initiators. This resulted in four paths and bandwidth available for FC connectivity between the PowerEdge hosts and PowerStore Storage.

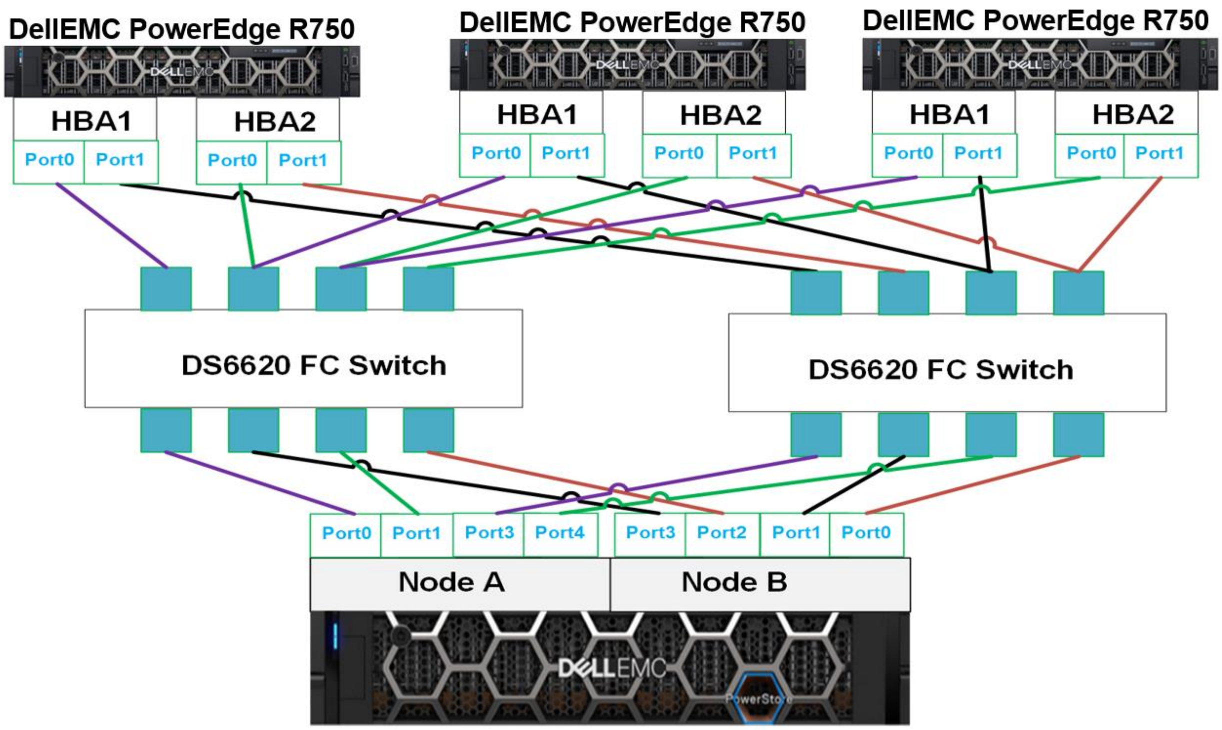


**Figure 2. Best Practice of maximizing Front-end I/O modules**

### Recommendation

Each PowerStore appliance and subsequent nodes should be zoned to maximize the FC bandwidth to make the SAN connectivity available. If the ports are not zoned correctly, it can lead to bottlenecks and ultimately performance issues.

### Implementation steps

- Add the necessary ports to the FC switches.
- Reconfigure the zoning so the mapping for initiators and targets is implemented appropriately.
- Create new Host Groups on the PowerStore Manager.
- Re-map volumes to new Host Groups.

Contact Dell support for issues related to appropriate reconfiguration.

### Additional resources

Dell PowerStore: Microsoft SQL Server Best Practices

## Adding storage volumes

Storage configuration can directly influence database performance. This best practice validates the number of storage groups that to further optimize database performance.

Volume groups were allocated to two groups: TarragonDB1 and TarragonDB2. This nomenclature was derived from the program name, Tarragon. All database files for the first of two databases were in the TarragonDB1 volume group. By incrementing the number in the volume group name (TarragonDB<#>), the engineering team created an additional copy of the first database for load tests. This storage configuration minimizes the number of PowerStore volume groups, which eases management but is not an optimized configuration. The following table shows the baseline storage group configuration:

**Table 7. Baseline storage group configuration**

| Volume group | Volume name | VMware volumes | Volume size (TB) | Notes |
|---|---|---|---|---|
| TarragonDB | TarragonDB1 | Tarragon_DB1 | 1.5 | All database files |
| | TarragonDB2 | Tarragon_DB2 | 1.5 | All database files |

Using multiple volumes for the most active portion of an SQL Server database has many benefits. The primary benefit of using multiple volumes is that the operating system creates an I/O queue path per storage group. The following table shows the optimized PowerStore storage configuration for the SQL Server database.

**Table 8. Optimized PowerStore storage group configuration**

| Volume group | Volume Name | VMware volumes | Volume size (GB) |
|---|---|---|---|
| TarragonDB# | DB#_Data1 | DB#_Data1 | 300 |
| | DB#_Data2 | DB#_Data2 | 300 |
| | DB#_Data3 | DB#_Data3 | 300 |
| | DB#_Data4 | DB#_Data4 | 300 |
| | DB#_Log | DB#_Log1 | 300 |
| TarragonTempDB | TempDB1_Data | TempDB1_Data | 300 |
| | TempDB1_Log | TempDB2_Log | 250 |
| | TempDB2_Data | TempDB2_Data | 300 |
| | TempDB2_Log | TempDB2_Log | 250 |

The goal of generating an optimized configuration was to expand the number of volumes to create more logical I/O paths, resulting in optimized storage performance. The three groups were named: TarragonDB1, TarragonDB2 and TarragonTempDB.

The optimized storage configuration improves the number of I/O paths and other actions like performance monitoring, storage snapshots, and replication. In terms of monitoring storage, the enterprise can view metrics for each part of the database with the optimized configuration. This provides the capability to continually scale capacity and performance as the database grows.

Not all parts of the SQL Server database are required for snapshots. Using this optimized configuration, the database administrator can granularly select what to snapshot, which saves storage capacity. The principle applies to storage replication; the database administrator can select the parts of the database that require protection. By replicating only part of the database, the enterprise benefits from using less network bandwidth.

### Recommendation

We recommend creating multiple separate volumes for UserDB datafile based on the workload requirements. Similarly, one separate Log volume per UserDB. TempDB data and log files should also be separate to maximize performance and assist management and scalability. As shown in table 9, each database data and log file volumes were placed in its own separate volume group

### Implementation steps

To create volumes and volume groups on the PowerStore using the PowerStore Manager, use the instructions provided in [the Dell PowerStore Manual].

## Creating A VMFS Datastore:

Once storage groups and volumes are created on the PowerStore, create respective VMFS datastores on the vSphere client using the instructions provided in the [VMware vSphere product documentation.]

When the disks are attached and visible on the operating system, use below steps to partition, format and mount the disks onto the system.

- View storage device on RHEL using the following command:

  ```
  df -h
  ```

  **or**

  ```
  cat /proc/partitions
  ```

Once you see your desired storage device listed, use the following commands to create GUID partition table (GPT) on the given storage device.

```
parted /dev/sdb mklabel gpt                    -- sdb is a storage
device example
```

Once the partition table is created, use the following command to create volume partitioning. In our case, we will be creating single primary partition with 100% capacity on each storage device.

```
parted /dev/sdb mkpart primary 2048s 100%
```

```
partprobe /dev/sdb
```

Once partition is created on the storage devices, create the file system (xfs on this case) on the same storage device using the following command:

```
mkfs.xfs /dev/sdb1
```

After file system created on the device, we will find the block id and mount on the directory of our choice.

```
mkdir -p /opt/db/data1  -- data1 is the directory where we
will be mounting the volume
mount /dev/sdb1 /opt/db/data1
```

Make an entry into /etc/fstab to for the mounting to be permanent.

```
UUID=c4cf7516-c092-4dcd-8366-df58d4f703b2    /opt/db/data1
xfs     defaults     0     0
```

Replace the UUID in above command with one you receive by running the following command.

```
blkid | grep sdb1
```

Change ownership of the mounted volume.

```
chown mssql:mssql /opt/db/data1
```

Perform all the previous steps on each of the disks attached to the virtual machine.

### Additional resources

[Dell PowerStore: Microsoft SQL Server Best Practices]

## VMware vSphere: Adding PVSCSI controllers

Paravirtual SCSI (PVSCSI) controllers offer greater throughput and overall lower CPU use. This best practice tested adding multiple PVSCSI controllers to improve storage performance.

In the baseline configuration, one PVSCSI controller was connect the virtual machine to the PowerStore storage array. Before implementing multiple PVSCSI controllers, the volume group configuration best practiced was implemented. The optimal volume group configuration consisted of adding multiple volumes for data, TempDB data, and TempDB log to create more I/O paths.

The working assumption was using multiple I/O paths and multiple PVSCSI controllers would yield better performance. For this test three additional PVSCSI controllers to each virtual machine for a total of four controllers per virtual machine.

### Implementation steps

Implement this best practice at the vCenter user interface. To add PVSCSI controllers, use the following steps:

1. From vCenter, select the VM and click **Edit Settings** > **Add Devices** > **SCSI controller**.
2. Add three more SCSI controllers, SCSI controller 1 through 3, with these settings:

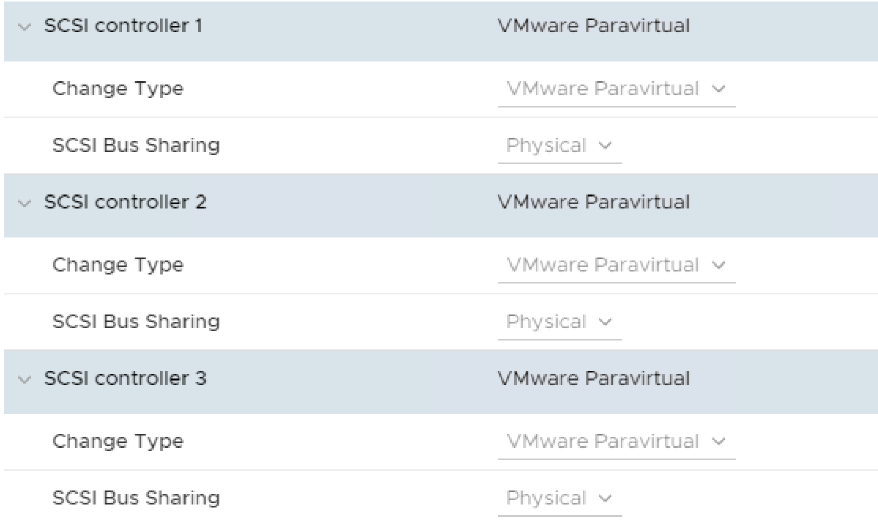


**Figure 3. Adding PVSCSI controllers to Virtual machine**

3. Re-map all existing volumes to the new controllers by selecting **Edit Settings** > **Virtual Device Node** on each VM.
4. Select the appropriate SCSI controller from the dropdown menu. Spread out the volumes across the four controllers as evenly as possible.

### Additional resources

[Configuring disks to use VMware Paravirtual SCSI (PVSCSI) controllers (1010398)]

## PowerStore performance policy

The PowerStore provides a direct method to manage volume performance relative to one another. It uses a performance policy that sets relative priority to a selected volume. The default performance policy is Medium, which provides equal performance to each volume. For SQL Server workloads, we recommend setting a High performance policy to the data volumes.

### Implementation steps

Follow the steps in the PowerStore Manual page to update this property.

### Additional resources

Dell PowerStore Configuring Volumes

# PowerEdge best practices

The following best practices were implemented on the PowerEdge servers:

1. Database Optimized Performance Profile
2. Disabling unused integrated devices

## Database optimized performance profile

PowerEdge R750 servers provide a simple and effective manner to control and optimize BIOS level parameters for different workloads. The database optimized performance profile (DBOP) once selected, automatically makes multiple selections on the BIOS layer.

**Table 9. System BIOS attribute value comparison**

| System Setup Screen | BIOS Attribute Name | BIOS Default Value | DBOP Value |
|---|---|---|---|
| Processor Settings | Virtualization Technology | Enabled | Disabled |
| System Profile | System Profile | Performance Per Watt (DAPC) | Performance |
| | CPU Power Management | System DBPM (DAPC) | Max pref |
| | C1E | Enabled | Disabled |
| | C State | Enabled | Disabled |
| | Uncore Frequency | DynamicUFS | Maximum |
| | Energy Efficient Policy | Balanced Performance | Performance |
| | Monitor/Mwait | Enabled | Enabled |
| | CPU Interconnect Bus Link Power Management | Enabled | Disabled |
| | PCI ASPM L1 Link Power Management | Enabled | Disabled |

### Implementation steps

To apply the DBOP on the iDRAC console, select **BIOS settings**. In the System Profile Settings, select **Workload Profile** as **Database Optimized Performance Profile**. These steps are described in the following figure:

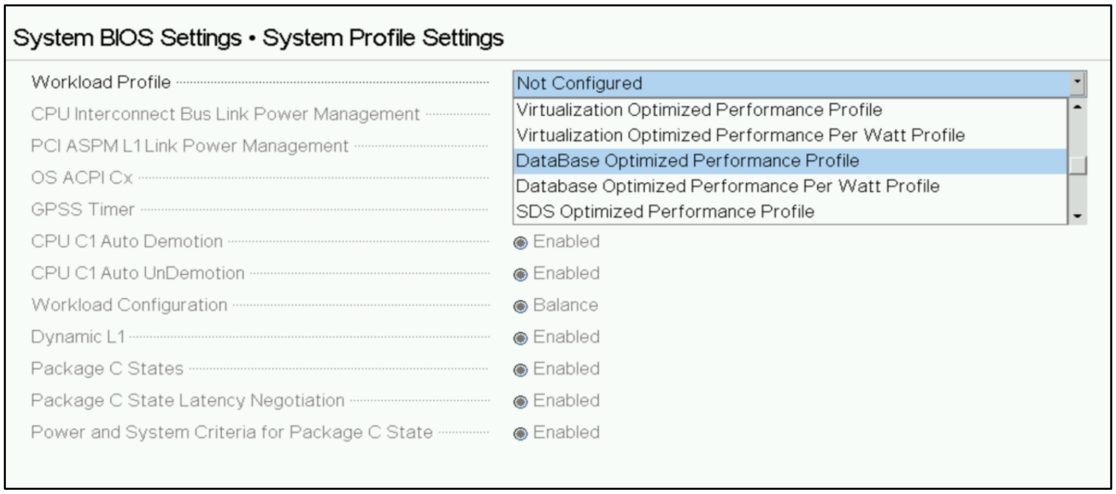


**Figure 4. Applying Workload profile in system BIOS**

**Note**: Applying the DBOP disables Virtualization Technology and x2APIC BIOS settings, so we reapply these through the same console.

## Disabling unused integrated devices

By default, PowerEdge servers have USB and other ports enabled. In this best practice, USB and serial ports are disabled for both a security and performance purposes. We disabled the following ports in BIOS:

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

### Recommendation

We recommend disabling unused integrated devices as a security precaution.

### Implementation steps

Use the following steps to update the settings for integrated devices and serial communication.

1. Turn on or restart your system.
2. Press **F2** immediately after you see the following message: **F2 = System Setup**.

**Note**: If your operating system begins to load before you press F2, wait for the system to finish booting, and then restart your system and try again.

3. On the **System Setup Main Menu** screen, click **System BIOS**.
4. Click **Integrated devices**.
5. Select the following options:
    a. Select **All ports off** from the drop-down list of User accessible USB ports parameter
    b. Select **Off** for the parameter iDRAC direct USB port
6. After updating the settings, click **Back.**

7. In **System BIOS settings** display, click Serial **communication**.
8. Select **off** from the drop-down list of Serial communication parameter.
9. After updating the setting, click **Back.**
10. Click **Finish** on the **System BIOS** screen.
11. Click **Yes** on the **Saving Changes** display, to confirm the update. You will receive a message that indicates successful updating of values.
12. Click **Finish** on the System Setup Main Menu screen**.**
13. Click **Yes** to confirm Exit and system reboot.

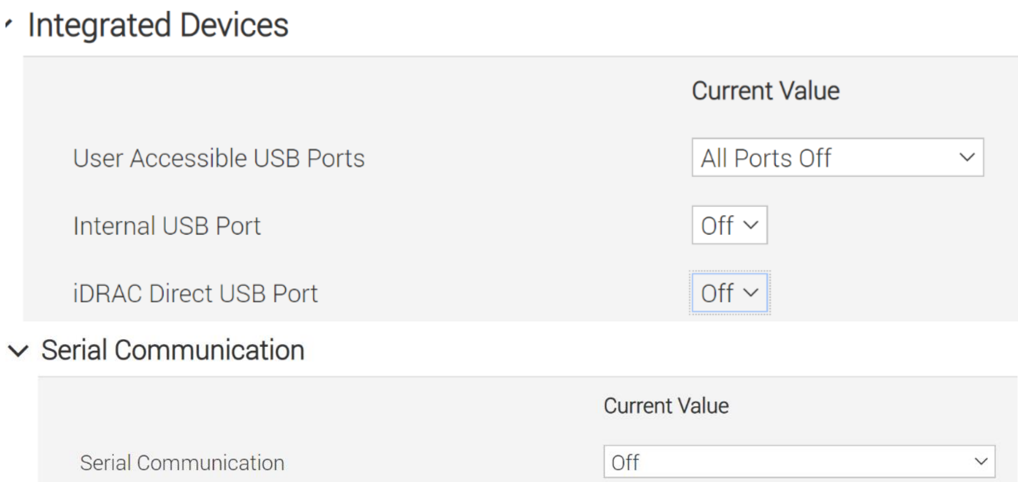


**Figure 5. Disabling USB Ports and Serial communication in system BIOS**

### Additional resources

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

# RHEL OS and VMware best practices

For best practices on the RHEL OS and VMware layer, we enabled the following settings:

1. Applying tuned-ADM profile
2. Enabling ESXi power policy

## Applying tuned-adm profile

Tuned-adm is a command line utility that allows the user to switch between multiple pre-defined and custom tuning profiles. In this best practice, we applied the tuned-adm profile recommended by Microsoft and RHEL for deployments on RHEL.

### Recommendation

We recommend using the tuned mssql profile.

### Implementation steps

To apply this profile, follow the steps and configuration guidelines that are described in [Performance best practices and configuration guidelines for SQL Server on Linux].

## Enabling ESXi power policy

There are four CPU Power Management Polices that can be selected for an ESXi host:

- **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

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

### Recommendation

We recommend updating the default power policy from balanced to high performance to ensure power management features do not impede database performance.

### Implementation steps

Use the following steps to update the power management policy:

1. Login to the VMware® vSphere Client
2. Browse to the host in the vSphere Client.
3. Click **Configure**.
4. Under Hardware, select Power Management and click **Edit Power Policy**.
5. Select High Performance and click **OK**.

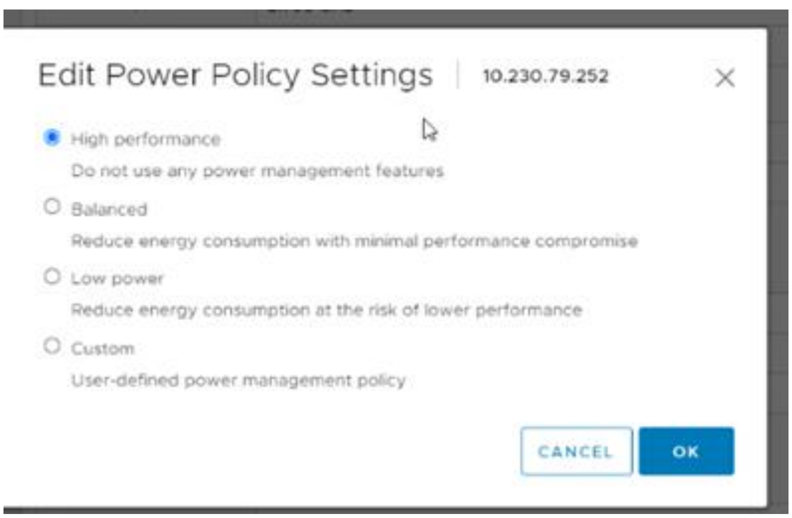


**Figure 6. Edit Power Policy Settings display**

### Additional resources

[ESXi Host Power Management Policies]
[vSphere Resource Management Update 2, VMware vSphere 7.0, VMware ESXi 7.0, vCenter Server 7.0]

# SQL Server best practices

Database best practices include:

1. Moving and resizing Tempdb
2. Autogrowth of UserDB files
3. Max worker threads
4. Process affinity for CPU and NUMA nodes
5. Memory allocated to SQL Server

## Moving and resizing Tempdb

TempDB is a SQL Server system database. As it is a resource designed to support all user databases connected to the same instance, it is important that it is appropriately sized, ideally on dedicated volumes for TempDB data and log. We created separate volumes to host both of for our best practice.

### Recommendation

Size TempDB logs appropriately and, if available, host them on dedicated volumes for TempDB data and log.

### Implementation steps

Use the steps outlined in Microsoft tempdb database to update properties for TempDB files.

## Autogrowth for database files

User database files, including data and log files, can grow during the insert, delete, or update operations stages which are often performed during an OLTP workload. To limit the need for this autogrowth being triggered often, users can set the Autogrow value for the data and log files.

The autogrow default values for UserDB and TempDB files are 64 MB.

Modifying the default values for both UserDB and TempDB files helps optimize how the database adds space and minimizes the impact on transactions. By increasing the size of autogrowth database efficiency improves by avoiding excessive autogrowths events. Database administrators must determine the proper autogrow value for their databases, as each is different. To validate autogrow best practices, we implemented the following changes to our SQL Server databases:

- Data and log files for UserDB set to FILEGROWTH = 1048576 MB (1 GB)
- Data and log files for TempDB set to FILEGROWTH = 1048576 MB (1 GB)

### Recommendation

We recommend assigning appropriate autogrowth values for database files.

### Implementation steps

Use the steps provided in Considerations for the autogrow and autoshrink settings in SQL Server to update properties for database files.

## Max Worker threads

The default value for worker threads is zero. This setting directs the SQL Server to automatically configure the number of worker threads at database startup. In our baseline tests we configured max worker threads to 864, which is the recommended value for databases using eight CPUs and 64-bit system starting with SQL Server 2016 (13.x) SP2 and SQL Server 2017 (14.x). For more information about Microsoft's configuration recommendations see, Configure the max worker threads Server Configuration Option.

After testing a few max worker thread configurations, we found that increasing the value to 864 provided a slight performance improvement.

### Recommendation

It is recommended to determine the updated max worker thread value for your environment and workload.

### Implementation steps

For information about configuring max worker threads, see Configure the max worker threads Server Configuration Option.

### Resources

Configure the max worker threads Server Configuration Option.

## Process Affinity for CPU cores and NUMA Nodes

CPU affinity describes the ability to define which processor cores the SQL Server engine uses to multitask. By assigning processor cores, the SQL Server only uses the selected processors for multitasking database operations. This can improve performance by reducing processor reloads and thread migration across processors.
The virtual machine supporting the SQL Server has 6 vCPUs. For this reason, we implemented the following configuration changes in this best practice:

- SQL Server Processor Affinity was configured to use processors 0 through 5 for a total of 6 processors.
- We ran the code below to assign NUMA NODE 0.

### Recommendation

We recommend setting SQL Server Process Affinity.

### Implementation steps

To configure SQL Server Processor Affinity to use processors zero through seven, we performed the following command:

```
ALTER SERVER CONFIGURATION SET PROCESS AFFINITY CPU=0 TO 7
```

To configure SQL Server Processor Affinity to use NUMA NODE 0 we performed the following command:

```
ALTER SERVER CONFIGURATION SET PROCESS AFFINITY NUMANODE=0
```

### Additional resources

Affinity mask Server Configuration Option

## Memory allocated to SQL server

Database administrators can modify memory allocation for SQL Server using the minimum server memory and maximum server memory parameters.

### Minimum server memory

Minimum server memory is the amount of allocated memory that the SQL Server reserves for the database. This setting is a memory allocation guarantee that the database administrator can use to optimize performance. Analysis of memory use from the operating system and other applications is required to accurately define the minimum server memory setting. Setting the minimum server memory to a high value can impact the operating system and lower overall performance.

We changed the minimum server memory from the default value to 360,448 MB (352GB).

### Maximum server memory

Maximum server memory is the maximum amount of memory that the SQL Server can use. Setting the maximum server value limits the amount of memory that the database can use. The difference between minimum and maximum server values defines the amount of memory that the SQL Server can dynamically use. For example, if the minimum is set to 64 GB and the maximum is set to 128 GB, then the SQL Server can dynamically use 64 GB (128 – 64 = 64). We highly recommend analyzing your operating system and other application memory use to ensure that the database does not impact the overall system.

We used the default value for our baseline configuration tests. For additional tests, we changed the maximum server memory to a value of 360,448 MB (352 GB). By setting the maximum and minimum server memory values to 120 GB the database will allocate all the memory and not exceed the 352 GB memory configuration.

### Recommendation

We recommend setting the minimum and maximum SQL Server memory to a value lower than that assigned to the operating system. These values should also be equal especially for predictable workloads.

### Implementation steps

In SQL the Server Management Studio we selected **Properties > Memory**, as shown in Figure 5. We set the minimum and maximum server memory values to 122880 MB (120 GB).

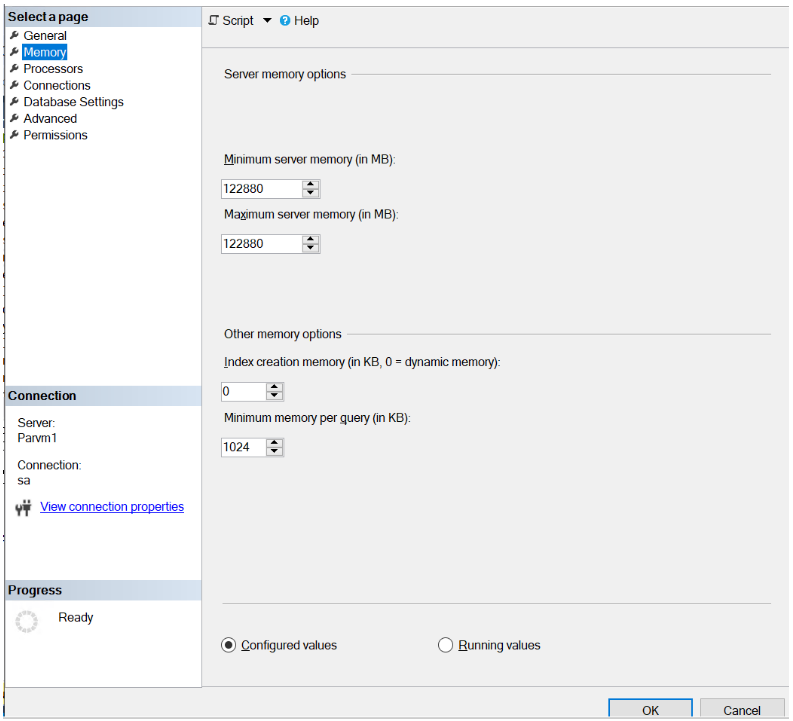


**Figure 7. Memory page from SQL Server Management Studio**

We also recommend applying this setting through the mssql config file. Run the following command to update the memory that is reserved for the SQL Server process to 120 GB:

```
/opt/mssql/bin/mssql-conf set memory.memorylimitmb 122880
```

## Resources

[Server memory configuration options]

# Chapter 4 Summary

This chapter presents the following topics:

# Database Solution Optimizer (DSO)

The Database Solution Optimizer (DSO) is a Dell toolkit that allows system administrators and database administrators to enable best practices at each layer in the solution stack. This toolkit allows the IT teams the ability to configure the solution with best practices during the initial stand-up of the infrastructure. It operates using a module-based approach allowing the IT teams the granularity and flexibility.  For more information, see the DSO github page.

# Summary

The optimized Microsoft SQL Server patform provides a validated solution for SQL Server customers that maximizes flexibility in design and robustness. It was vigorously tested with simulated SQL Server OLTP workloads to ensure all hardware and software are working according to best practices.  This validated design was purposely built for SQL Server databases. This allows customers to speed up the deployment of an SQL Server database infrastructure in less time.  This design guide also provides customers with a set of initial guidelines to deploy and configure each layer of this SQL Server deployment. Without a validated solution, customers are required to procure the necessary hardware and perform their own tests to ensure all layers are working seamlessly together. This would consume significant amounts of time from the IT teams and database administrators and could delay new business line rollouts or cause missed deadlines.

This validated design consists of Dell's state of the art PowerEdge R750 servers, PowerSwitch S5224F-ON, and a PowerStore 1000T storage system. This provides an optimized and robust infrastructure for SQL Server databases.

The PowerEdge R750 is a 2U dual-socket rack server that can be configured with various processor types. Whether it is a bare metal or virtualized deployment, you can configure the PowerEdge R750 to handle the required workloads.

The PowerSwitch S5224F-ON has 24 x SFP28 ports and 4 x QSFP28 ports and provides plenty of network bandwidth for both data networks and uplinks.

The PowerStore is a 2U two-node dual socket Intel® Xeon® platform that has features such as volume groups and snapshots that are required for SQL Server database deployments. The volume group feature allows database snapshot consistency you can use for cloning and near-line backups.

This validated design provides a simple starting point for deploying SQL Server database workloads that otherwise would be complex and time-consuming.

# 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:** Vaani Kaur, Sanjeev Ranjan

**Contributors**: Ava English

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