ARCHITECTING MICROSOFT SQL SERVER ON VMWARE VSPHERE®

Best Practices Guide

Table of Contents

Table of Contents, continued

Table of Contents, continued

List of Figures

List of Figures, continued

List of Tables

1. Introduction

Microsoft SQL Server®[1] is one of the most widely deployed database platforms in the world, with many organizations having dozens or even hundreds of instances deployed in their environments. The flexibility of SQL Server, with its rich application capabilities combined with the low costs of x86 computing, has led to a wide variety of SQL Server installations ranging from large data warehouses with business intelligence and reporting features to small, highly specialized departmental and application databases. The flexibility at the database layer translates directly into application flexibility, giving end users more useful application features and ultimately improving productivity.

Application flexibility often comes at a cost to operations. As the number of applications in the enterprise continues to grow, an increasing number of SQL Server installations are brought under lifecycle management. Each application has its own set of requirements for the database layer, resulting in multiple versions, patch levels, and maintenance processes. For this reason, many application owners insist on having a SQL Server installation dedicated to an application. As application workloads vary greatly, many SQL Server installations are allocated more hardware resources than they need, while others are starved for compute resources.

These challenges have been recognized by many organizations in recent years. These organizations are now virtualizing their most critical applications and embracing a "virtualization first" policy. This means applications are deployed on virtual machines (VMs) by default rather than on physical servers, and SQL Server is the most virtualized critical application in the past few years.

Virtualizing SQL Server with vSphere® allows for the best of both worlds, simultaneously optimizing compute resources through server consolidation and maintaining application flexibility through role isolation, taking advantage of the software-defined data center (SDDC) platform and capabilities such as network and storage virtualization. SQL Server workloads can be migrated to new sets of hardware in their current states without expensive and error-prone application remediation, and without changing operating system (OS) or application versions or patch levels. For high performance databases, VMware and partners have demonstrated the capabilities of vSphere to run the most challenging SQL Server workloads.

Virtualizing SQL Server with vSphere enables many additional benefits. For example, vSphere vMotion®, which enables seamless migration of virtual machines containing SQL Server instances between physical servers and between data centers without interrupting users or their applications. vSphere Distributed Resource Scheduler™ (DRS) can be used to dynamically balance SQL Server workloads between physical servers. vSphere High Availability (HA) and vSphere Fault Tolerance (FT) provide simple and reliable protection for virtual machines containing SQL Server and can be

[1] Further in the document referenced as SQL Server

used in conjunction with SQL Server's built-in HA capabilities. Among other features, VMware NSX® provides network virtualization and dynamic security policy enforcement. VMware Site Recovery Manager™ provides disaster recovery plan orchestration, vRealize Operations manager provides comprehensive analytic and monitoring engine, and VMware Cloud on AWS can be consumed to take the advantages of public cloud. There are many more benefits that VMware can provide for the benefit of virtualized applications.

For many organizations, the question is no longer whether to virtualize SQL Server, rather, it is to determine the best architecture design to achieve the business and technical requirements while keeping operational overhead to a minimum for cost effectiveness.

1.1 Purpose

This document provides best practice guidelines for designing and implementing SQL Server in virtual machine to run on vSphere (further referenced as vSphere). The recommendations are not specific to a particular hardware set, or to the size and scope of a particular SQL Server implementation. The examples and considerations in this document provide guidance only, and do not represent strict design requirements, as varying application requirements might result in many valid configuration possibilities.

1.2 Target Audience

This document assumes a knowledge and understanding of vSphere and SQL Server. Architectural staff can use this document to gain an understanding of how the system will work as a whole as they design and implement various components. Engineers and administrators can use this document as a catalog of technical capabilities. DBA staff can use this document to gain an understanding of how SQL Server might fit into a virtual infrastructure. Management staff and process owners can use this document to help model business processes to take advantage of the savings and operational efficiencies achieved with virtualization.

2. SQL Server Requirements Considerations

When considering SQL Server deployments as candidates for virtualization, you need a clear understanding of the business and technical requirements for each database instance. These requirements span multiple dimensions, such as availability, performance, scalability, growth and headroom, patching, and backups.

Use the following high-level procedure to simplify the process for characterizing SQL Server candidates for virtualization:

- Understand the performance characteristics and growth patterns of the workloads associated with the applications accessing SQL Server.
- Understand availability and recovery requirements, including uptime guarantees and disaster recovery for both the VM and the databases.
- Capture resource utilization baselines for existing physical server hosting databases.
- Plan the migration/deployment to vSphere.

2.1 Understand SQL Server Workloads

The SQL Server is a relational database management system (RDBMS) that runs workloads from applications. A single installation, or instance, of SQL Server running on Windows Server (or Linux) can have one or more user databases. Data is stored and accessed through the user databases. The workloads that run against these databases can have different characteristics that influence deployment and other factors, such as feature usage or the availability architecture. These factors influence characteristics like how virtual machines are laid out on VMware ESXi™ hosts, as well as the underlying disk configuration.

Before deploying SQL Server instances inside a VM on vSphere, you must understand the business requirements and the application workload for the SQL Server deployments you intend to support. Each application has different requirements for capacity, performance, and availability. Consequently, each deployment must be designed to optimally support those requirements. Many organizations classify SQL Server installations into multiple management tiers based on service level agreements (SLAs), recovery point objectives (RPOs), and recovery time objectives (RTOs). The classification of the type of workload a SQL Server runs often dictates the architecture and resources allocated to it. The following are some common examples of workload types. Mixing workload types in a single instance of SQL Server is not recommended.

- OLTP (online transaction processing) databases are often the most critical databases in an organization. These databases usually back customer-facing applications and are considered essential to the company's core operations. Mission-critical OLTP databases and the applications they support often have SLAs that require very high levels of performance and are very sensitive for performance degradation and availability. SQL Server VMs running OLTP mission-critical databases might require more careful resource allocation (central processor unit (CPU), memory, disk, and network) to achieve optimal performance. They might also be candidates for clustering with Windows Server Failover Cluster (WSFC), which run either an Always On Failover Cluster Instance (FCI) or Always On Availability Group (AG). These types of databases are usually characterized with mostly intensive random writes to disk and sustained CPU utilization during working hours.

- DSS (decision support systems) databases, can be also referred to as data warehouses. These are mission critical in many organizations that rely on analytics for their business. These databases are very sensitive to CPU utilization and read operations from disk when queries are being run. In many organizations, DSS databases are the most critical resource during month/quarter/year end.
- Batch, reporting services, and ETL databases are busy only during specific periods for such tasks as reporting, batch jobs, and application integration or ETL workloads. These databases and applications might be essential to your company's operations, but they have much less stringent requirements for performance and availability. They may, nonetheless, have other very stringent business requirements, such as data validation and audit trails.
- Other smaller, lightly used databases typically support departmental applications that may not adversely affect your company's real-time operations if there is an outage. Many times, you can tolerate such databases and applications being down for extended periods.

Resource needs for SQL Server deployments are defined in terms of CPU, memory, disk and network I/O, user connections, transaction throughput, query execution efficiency/latencies, and database size. Some customers have established targets for system utilization on hosts running SQL Server, for example, 80 percent CPU utilization, leaving enough headroom for any usage spikes and/or availability.

Understanding database workloads and how to allocate resources to meet service levels helps you to define appropriate virtual machine configurations for individual SQL Server databases. Because you can consolidate multiple workloads on a single vSphere host, this characterization also helps you to design a vSphere and storage hardware configuration that provides the resources you need to deploy multiple workloads successfully on vSphere.

2.2 Business Continuity Options

Running SQL Server under vSphere offers many options for availability, backup and disaster recovery utilizing the features from both VMware and Microsoft. This section provides brief overview of different options that exist for availability and recovery[2].

2.2.1 VMware vSphere Features for Business Continuity

VMware technologies, such as vSphere High Availability (HA),vSphere Fault Tolerance (FT), vSphere vMotion, vSphere Storage vMotion®, and VMware Site Recovery Manager™ can be used in a business continuity design to protect SQL Server instances running on top of a VM from planned and unplanned downtime. These technologies protect SQL Server instances from failure of a single hardware component, to a full site failure, and in conjunction with native SQL Server business continuity capabilities, increase availability.

[2] For the comprehensive discussion of high availability options refer to https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/solutions/sql-server-on-vmware-availability-and-recovery-options.pdf, https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/solutions/vmware-vsphere-highly-available-mission-critical-sql-server-deployments.pdf

2.2.1.1 VSPHERE HIGH AVAILABILITY

vSphere HA provides an easy-to-use, cost-effective high availability solution for applications running in virtual machines. vSphere HA leverages multiple ESXi hosts configured as a cluster to provide rapid recovery from outages and cost-effective high availability for applications running in virtual machines by graceful restart of a virtual machine.

2.2.1.2 VSPHERE FAULT TOLERANCE

vSphere FT provides a higher level of availability, allowing users to protect a VM from a physical host failure with no loss of data, transactions, or connections. vSphere FT provides continuous availability by verifying that the states of the primary and secondary VMs are identical at any point in the CPU instruction execution of the virtual machine. If either the host running the primary VM or the host running the secondary VM fails, an immediate and transparent failover occurs.

2.2.1.3 VSPHERE VMOTION AND VSPHERE STORAGE VMOTION

Planned downtime typically accounts for more than 80 percent of data center downtime. Hardware maintenance, server migration, and firmware updates all require downtime for physical servers and storage systems. To minimize the impact of this downtime, organizations are forced to delay maintenance until inconvenient and difficult-to-schedule downtime windows.

The vSphere vMotion and vSphere Storage vMotion functionality in vSphere makes it possible for organizations to reduce planned downtime because workloads in a VMware environment can be dynamically moved to different physical servers or to different underlying storage without any service interruption. Administrators can perform faster and completely transparent maintenance operations, without being forced to schedule inconvenient maintenance windows.

NOTE: vSphere version 6.0 and later support vMotion of a VM with RDM disk in physical compatibility mode, being part of a windows failover cluster.

2.2.2 SQL Server Availability Features for Business Continuity

All of SQL Server's built-in availability features and techniques are supported inside a guest on vSphere, including SQL Server Always On Availability Groups, Always On Failover Cluster Instances, database mirroring, and log shipping[3]. These native SQL Server options can be combined with vSphere features to create flexible and robust availability and recovery scenarios, applying the most efficient and appropriate tools for each use case.

[3] More details: https://kb.vmware.com/kb/2147661

The following table lists the most common SQL Server availability options and their ability to meet various RTOs and RPOs. Before choosing any one option, evaluate your own business requirements to determine which scenario best meets your specific needs.

Table 1. SQL Server 2012+ High Availability Options

TECHNOLOGY	GRANULARLY	STORAGE TYPE	RPO – DATA LOSS	RPO – DOWNTIME
Always On Availability Groups (AGs)	Database	Non-shared	None (with synchronous commit mode)	Usually measured in seconds; need to account for objects outside the database
Always On Failover Cluster Instances (FCIs)	Instance	Shared	None	~30 seconds to a few minutes depending on what is in each database's transaction log
Database Mirroring[4]	Database	Non-shared	None (with high safety mode)	Usually measured in seconds; need to account for objects outside the database
Log Shipping	Database	Non-shared	Possible transaction log	Depends on state of warm standby and if transaction logs still need to be applied; could be seconds to much longer depending

2.3 VMware Cloud on AWS

VMware Cloud on AWS brings VMware's enterprise-class SDDC software to the AWS Cloud with optimized access to AWS services. Powered by VMware Cloud Foundation, VMware Cloud on AWS integrates VMware compute, storage and network virtualization products (vSphere, VMware vSAN and VMware NSX) along with VMware vCenter management, optimized to run on dedicated, elastic, bare-metal AWS infrastructure[5].

VMware Cloud on AWS allows customers to consume the public cloud in the same manner and with the same toolset as the on-premises vSphere environment. VMs can

[4] This feature was deprecated in SQL Server 2012 and should not be used if possible. Microsoft does not always remove deprecated features. Always On Availability Groups is the official replacement including for Standard Edition as noted here https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/basic-availability-groups-always-on-availability-groups?view=sql-server-2017

[5] https://cloud.vmware.com/vmc-aws/faq

be bi-directionally migrated using vSphere vMotion technology between on premises datacenters and VMware Cloud on AWS without any modification for the VM or application configuration.

Consider among other following use cases enabled by VMware Cloud on AWS for a virtualized SQL Server instance:

- Simple application migration to place a database server near applications in the public cloud
- Benefit from the on-demand capacity available in the public cloud
- Provide disaster recovery as a service with VMware Site Recovery Manager

After an instance of a virtualized SQL Server is moved to VMware Cloud on AWS, operational and configuration guidelines summarized in this document continue to apply[6].

2.4 SQL Server on vSphere Supportability Considerations

One of the goals of the purpose build architecture is to provide a solution which can be easily operated and maintained. The supportability aspect should be given the high priority while designing a solution for mission-critical applications.

Consider following supportability points while architecting SQL Server deployments on vSphere:

- Use VMware Configuration Maximums Tool[7] to check the resulting deployment if any limits are reached or may be reached in the near future.
- Use VMware Compatibility Guide[8] to check compatibility for all components used.
- Use VMware Lifecycle Product Matrix[9] to find the End of General Support (EGS) date for solutions in use. For example, as of time of writing this document, EGS for VMware ESXi/vCenter 5.5 is due 19 Sep 2018.
- Microsoft supports the virtualized SQL Server deployments for the versions listed in the Microsoft Support Knowledge Base Article "Support policy for SQL Server products that are running in a hardware virtualization environment"[10]. VMware imposes no limitations on the version of SQL Server used inside the guest. If your requirements dictate deploying older versions of SQL Server and Windows Server, that can be done using vSphere. Consider choosing end-to-end supported solution while designing virtualized SQL Server deployments.
- Consult the Microsoft Lifecycle Policy website[11] to ensure that the version of SQL Server and Windows Server/Linux distribution are in support.
- vSphere 6.7 and earlier releases (starting with ESX 3.5 Update 2, for almost 10 years and more than any other vendor of a virtualized platform) are included in

[6] Licensing considerations are addressed here: https://blogs.vmware.com/apps/2018/06/licensing-microsoft-sql-server-in-a-vmware-cloud-on-aws-environment.html

[7] https://configmax.vmware.com/home

[8] https://www.vmware.com/resources/compatibility/search.php

[9] https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/support/product-lifecycle-matrix.pdf

[10] https://support.microsoft.com/en-us/help/956893/support-policy-for-microsoft-sql-server-products-that-are-running-in-a

[11] https://support.microsoft.com/lifecycle

the Microsoft Windows Server Virtualization Validation Program[12]. This certification provides VMware customers access to cooperative technical support from Microsoft and VMware. If escalation is required, VMware can escalate mutual issues rapidly and work directly with Microsoft engineers to expedite resolution.

- Relaxed policies for application license mobility: SQL Server 2012 further relaxed its licensing policy for customers under Software Assurance (SA) coverage. With SA, you can re-assign SQL Server licenses to different servers within a server farm as often as needed. You can also reassign licenses to another server in another server farm, or to a non-private cloud, once every 90 days.

3. Best Practices for Deploying SQL Server Using vSphere

A properly designed virtualized SQL Server instance running in a VM with Windows Server or Linux, using vSphere is crucial to the successful implementation of enterprise applications. One main difference between designing for performance of critical databases and designing for consolidation, which is the traditional practice when virtualizing, is that when you design for performance you strive to reduce resource contention between VMs as much as possible and even eliminate contention altogether. The following sections outline VMware recommended practices for designing and implementing your vSphere environment to optimize for best SQL Server performance.

3.1 Right-Sizing

Right-sizing is a term that means when deploying a VM, it is allocated the appropriate amount of compute resources, such as virtual CPUs and RAM, to power the database workload instead of adding more resources than are actively utilized, a common sizing practice for physical servers. Right-sizing is imperative when sizing virtual machines and the right-sizing approach is different for a VM compared to physical server.

For example, if the number of CPUs required for a newly designed database server is eight CPUs, when deployed on a physical machine, the DBA typically asks for more CPU power than is required at that time. The reason is because it is typically more difficult for the DBA to add CPUs to this physical server after it is deployed. It is a similar situation for memory and other aspects of a physical deployment; it is easier to build in capacity than try to adjust it later, which often requires additional cost and downtime. This can also be problematic if a server started off as undersized and cannot handle the workload it is designed to run.

However, when sizing SQL Server deployments to run on a VM, it is important to assign that VM only the exact amount of resources it requires at that time. This leads to optimized performance and the lowest overhead, and is where licensing savings can be obtained with critical production SQL Server virtualization. Subsequently, resources

[12] https://www.windowsservercatalog.com/svvp.aspx?svvppage=svvp.htm

can be added non-disruptively, or with a short reboot of the VM. To find out how many resources are required for the target VM running SQL Server, if there is an existing installation of SQL Server, monitor the server using dynamic management views (DMVs) or Performance Monitor (if on Windows Server). Many third-party monitoring tools can be used as well, such as Blue Medora's VMware vRealize Operations Management Pack for Microsoft SQL Server, which is capable of DMV-based monitoring with ongoing capacity management and will alert if there is resource waste or contention. The amount of collected time series data should be enough to capture all relevant workloads spikes such as quarter end or monthly reports. At least two weeks or, preferably, one full business cycle should be sampled before an analysis is performed.

There are two ways to size the VM based on the gathered data:

- When a SQL Server is considered critical with high performance requirements, take the most sustained peak as the sizing baseline.
- With lower tier SQL Server implementations, where consolidation takes higher priority than performance, an average can be considered for the sizing baseline. Using this approach, it's expected that the performance might be degraded during workload peaks.

When in doubt, start with the lower amount of resources, monitor consumption, and grow as necessary. After the VM has been created, continuous monitoring should be implemented and adjustments can be made to its resource allocation from the original baseline.

Right-sizing a VM is a complex process and wise judgement should be made between over-allocating resources and underestimating the workload requirements:

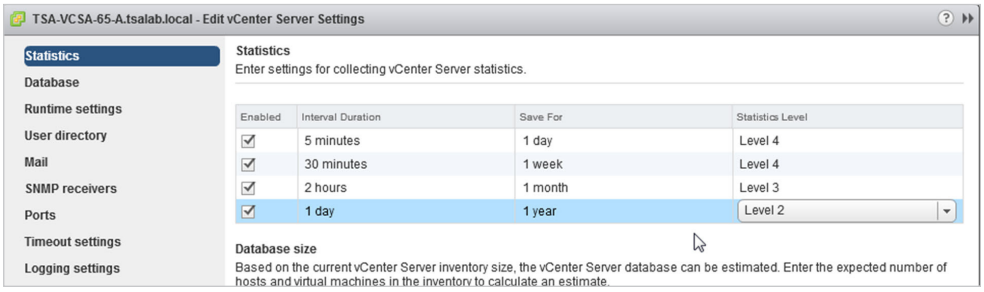
- Configuring a VM with more virtual CPUs than its workload require can cause increased resource usage, potentially impacting performance on heavily loaded systems. Common examples of this include a single-threaded workload running in a multiple-vCPU VM, or a multithreaded workload in a virtual machine with more vCPUs than the workload can effectively use. Even if the guest OS does not use some of its vCPUs, configuring VMs with those vCPUs still imposes some small resource requirements on ESXi that translate to real CPU consumption on the host.
- Over-allocating memory also unnecessarily increases the VM memory overhead and might lead to a memory contention, especially if reservations are used. Be careful when measuring the amount of memory consumed by a VM hosting SQL Server with the vSphere memory counter "active"[13]—the counter tends to underestimate memory usage. Applications that contain their own memory management, such as SQL Server, use and manage memory differently. Consult with the database administrator to confirm memory consumption rates using SQL Server-level memory metrics before adjusting the memory.
- Having more vCPUs assigned for the VM containing a virtualized SQL Server instance also has SQL Server licensing implications in certain scenarios, such as per-virtual-core licenses.

[13] More details can be found here: https://www.vmware.com/techpapers/2011/understanding-memory-management-in-vmware-vsphere-10206.html

3.2 vCenter Server Configuration

The vCenter server configuration, by default, is set to a base level of statistics collection, and useful for historical trends. Some of the real-time statistics are not visible beyond the one-hour visibility that this view provides. For the metrics that persist beyond real-time, these metrics are rolled up nightly and start to lose some of the granularity that is critical for troubleshooting specific performance degradation. The default statistics level is Level 1 for each of the four intervals (day, week, month, and year). To achieve a significantly longer retention of increased granular metrics, the following statistics levels are recommended.

Figure 1.
vCenter Server Statistics



Consider implementing a monitoring solution capable to store and analyze long time series data.

3.3 ESXi Cluster Compute Resource Configuration

The vSphere host cluster configuration is vital for the well-being of a production SQL Server platform. The goals of an appropriately engineered compute resource cluster include maximizing the VM and SQL Server availability, minimizing the impact of hardware component failures, and minimizing the SQL Server licensing footprint.

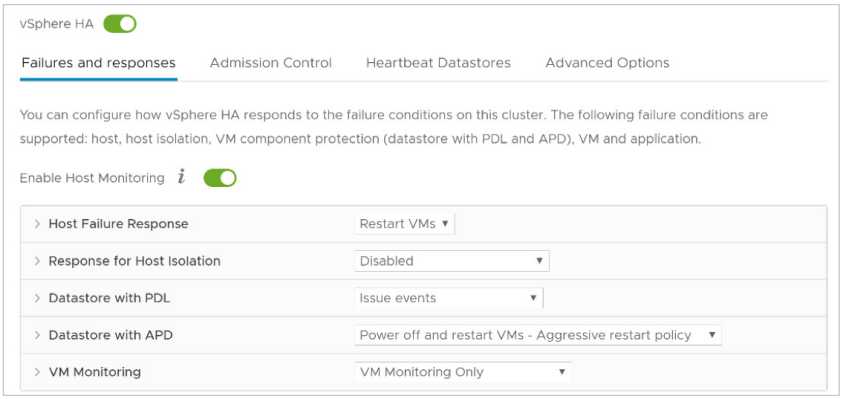
3.3.1 Sphere High Availability

vSphere HA is a feature that provides resiliency to a vSphere environment. If an ESXi host were to fail suddenly, vSphere HA will attempt to restart the virtual machines that were running on the downed host onto the remaining hosts.

vSphere HA should be enabled for SQL Server workloads unless your SQL Server licensing model could come into conflict. Make sure that an appropriate selection is configured within the cluster's HA settings for each of the various failure scenarios[14].

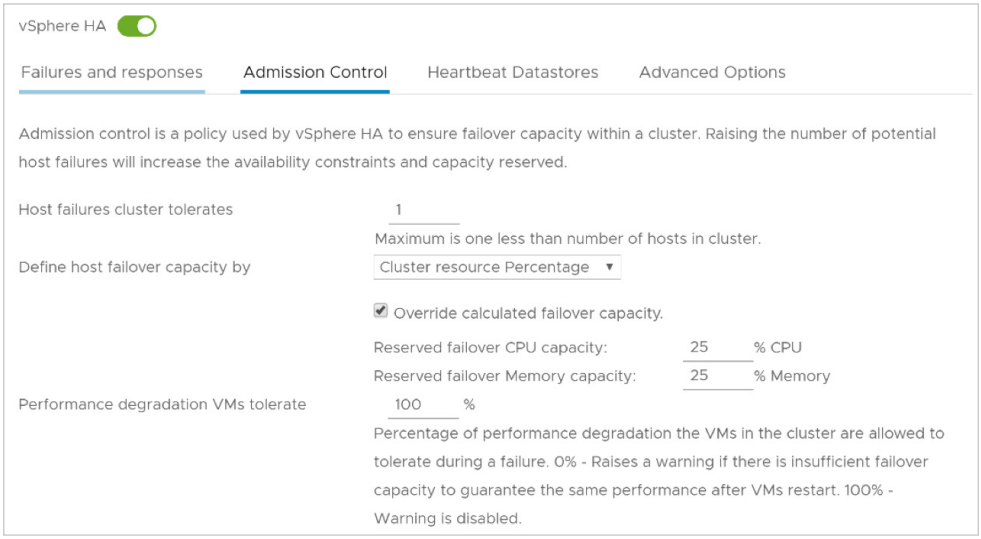
[14] Consult https://docs.vmware.com/en/VMware-vSphere/6.5/vsphere-esxi-vcenter-server-65-availability-guide.pdf for more details

Figure 2.
vSphere HA Settings



For mission-critical SQL Server workloads, ensure that enough spare resources on the host cluster exists to withstand a predetermined number of hosts removed from the cluster, both for planned and unplanned scenarios. Using a dedicated failover host might be justified for such workloads.

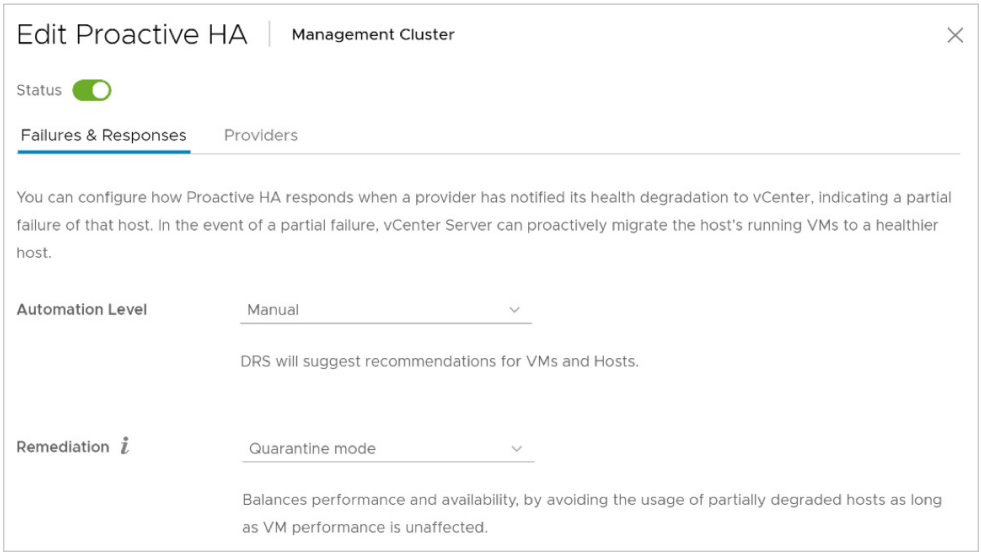
Figure 3.
vSphere Admission Control Settings



vSphere HA admission control can be configured to enforce the reservation of enough resources so that the ability to power on these VMs is guaranteed.

vSphere 6.5 introduced a new feature called Proactive HA. Proactive HA detects error conditions in host hardware, and can evacuate a host's VMs onto other hosts in advance of the hardware failure.

Figure 4.
Proactive HA

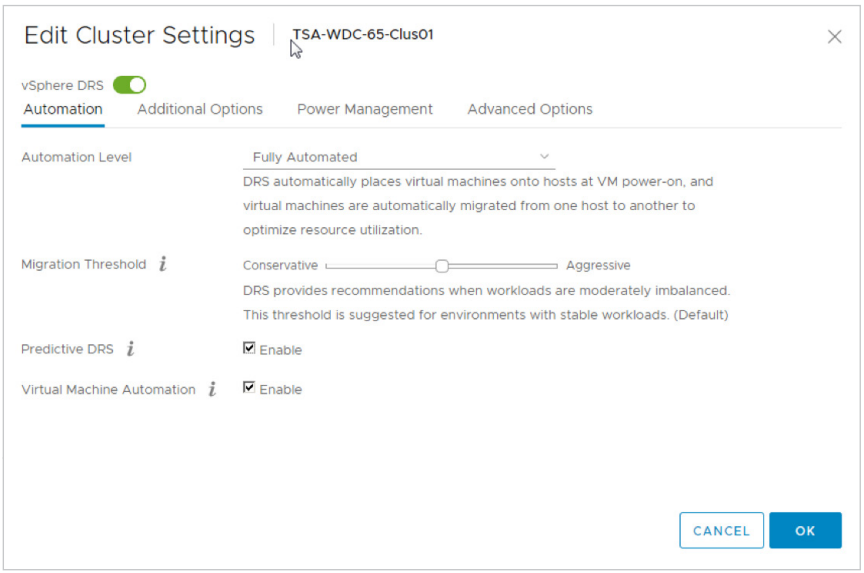


3.3.2 VMware DRS Cluster

A VMware DRS cluster is a collection of ESXi hosts and associated virtual machines with shared resources and a shared management interface. When you add a host to a DRS cluster, the host's resources become part of the cluster's resources. In addition to this aggregation of resources, a DRS cluster supports cluster-wide resource pools and enforces cluster-level resource allocation policies.

VMware recommends to enable DRS functionality for a cluster hosting SQL Server workloads[15].

Figure 5.
vSphere DRS Cluster

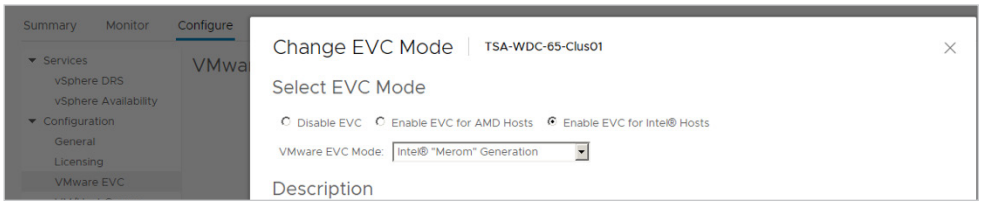


[15] More details: https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/drs-vsphere65-perf.pdf, https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/vsphere6-drs-perf.pdf

3.3.3 VMware Enhanced vMotion Compatibility[16]

The Enhanced vMotion Compatibility (EVC) feature helps ensure vMotion compatibility for the hosts in a cluster. EVC ensures that all hosts in a cluster present the same CPU feature set to virtual machines, even if the actual CPUs on the hosts differ. Using EVC prevents migrations with vMotion from failing because of incompatible CPUs. When EVC is enabled, all host processors in the cluster are configured to present the feature set of a baseline processor. This baseline feature set is called the EVC mode. EVC uses AMD-V Extended Migration technology (for AMD hosts) and Intel FlexMigration technology (for Intel hosts) to mask processor features so that hosts can present the feature set of an earlier generation of processors. The EVC mode must be equivalent to, or a subset of, the feature set of the host with the smallest feature set in the cluster.

Figure 6.
VMware EVC Settings



Consider evaluating the impact of enabling EVC mode: hiding certain CPU features may affect performance of a virtualized SQL Server instance. Avoid enabling EVC without proper use case.

The following use cases might justify enabling of EVC mode:

- A cluster consisting of hosts with different CPU microarchitectures (for example, Intel Westmere, and Intel Sandy Bridge) and where a vMotion of VMs between hosts is required. Avoid such configuration in production.
- Cross-cluster vMotion is required and hosts in different clusters have different CPU microarchitectures. Consider using per-VM EVC (Section 3.5.7) if only portion of VMs might need to be migrated to another cluster.

3.3.4 Resource Pools[17]

A resource pool is a logical abstraction for flexible management of resources. Resource pools can be grouped into hierarchies and used to hierarchically partition available CPU and memory resources.

For example, a three-tier resource pool architecture can be used for prioritizing business critical SQL Server instances running in VMs over less important deployments, such as development and test. The resources pools can be configured

[16] https://docs.vmware.com/en/VMware-vSphere/6.5/com.vmware.vsphere.vcenterhost.doc/GUID-03E7E5F9-06D9-463F-A64F-D4EC20DAF22E.html

[17] More details: https://docs.vmware.com/en/VMware-vSphere/6.7/vsphere-esxi-vcenter-server-67-resource-management-guide.pdf, chapter 9

for high, normal, and low CPU and memory share values, and VMs placed into the resource pools by priority.

Resource pools should not be used as folders for virtual machines. Incorrect usage of resource pools, especially nested resource pools, can lead to reduced performance of the virtual machines. Never combine a VM and a Resource pool in the same level of the hierarchy—it will lead to a VM having same share as the whole Resource pool.

3.4 ESXi Host Configuration

The settings configured both within the host hardware and the ESXi layers can make a substantial difference in performance of VMs with SQL Server placed on them.

3.4.1 BIOS/UEFI and Firmware Versions

As a best practice, update the BIOS/UEFI firmware on the physical server that is running critical systems to the latest version and make sure all the I/O devices have the latest supported firmware version.

3.4.2 BIOS/UEFI Settings

The following BIOS/UEFI settings are recommended for high-performance environments (when applicable):

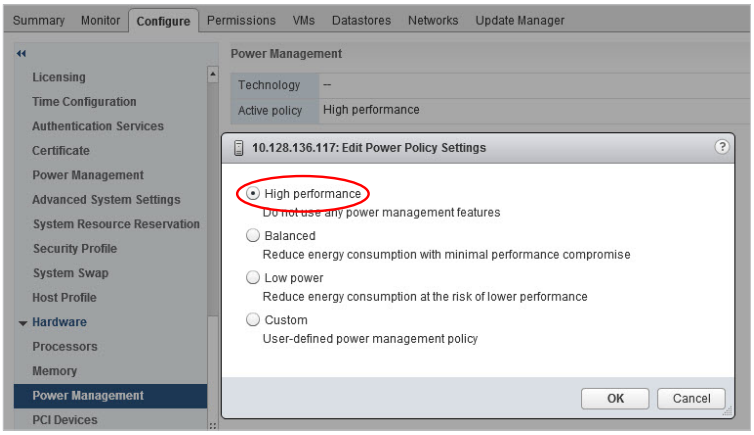
- Enable Turbo Boost.
- Enable Hyper-Threading.
- Verify that all ESXi hosts have NUMA enabled in the BIOS/UEFI. In some systems (for example, HP Servers), NUMA is enabled by disabling node interleaving. Consult your server hardware vendor for the applicable BIOS settings for this feature.
- Enable advanced CPU features, such as VT-x/AMD-V, EPT, and RVI.
- Follow your server manufacturer's guidance in selecting the appropriate Snoop Mode.
- Disable any devices that are not used (for example, serial ports).
- Set Power Management (or its vendor-specific equivalent label) to "OS controlled" (or its vendor-specific equivalent label). This wil l enable the ESXi hypervisor to control power management based on the selected policy. See the section 3.4.3 for more information.
- Disable all processor C-states (including the C1E halt state). These enhanced power management schemes can introduce memory latency and sub-optimal CPU state changes (Halt-to-Full), resulting in reduced performance for the VM.

3.4.3 Power Management

By default, ESXi has been heavily tuned for driving high I/O throughput efficiently by utilizing fewer CPU cycles and conserving power, as required by a wide range of workloads. However, many applications require I/O latency to be minimized, even at the expense of higher CPU utilization and greater power consumption.

An ESXi host can take advantage of several power management features that the hardware provides to adjust the trade-off between performance and power use. You can control how ESXi uses these features by selecting a power management policy. While previous versions of ESXi default to "High Performance" power schemes, vSphere 5.0 and later defaults to a "Balanced" power scheme. For critical applications, such as SQL Server, the default "Balanced" power scheme should be changed to "High Performance"[18].

Figure 7.
Recommended ESXi Host Power Management Setting



NOTE: It's crucial to follow the recommendation in the section 3.4.2 and configure the server BIOS/UEFI to pass the power management to ESXi ("OS control"). If this settings is not configured, ESXi power management policies will have no effect.

NOTE: Setting correct power management policies in BIOS/UEFI and in ESXi should be accomplished by configuring power policies in OS. See sections 4.1.1 and 4.2.3 for more details.

3.5 Virtual Machine CPU Configuration

Correct assignment of CPU resources are vital for SQL Server workloads. The section provides guidelines on managing CPU assignment for a VM hosting SQL Server instances.

[18] Some workloads might benefit from the combination of deep C states for some cores and Turbo boosting another. For this combination, custom BIOS power policy should be used with deep C states enabled and ESXi power policy should be set to "balanced".

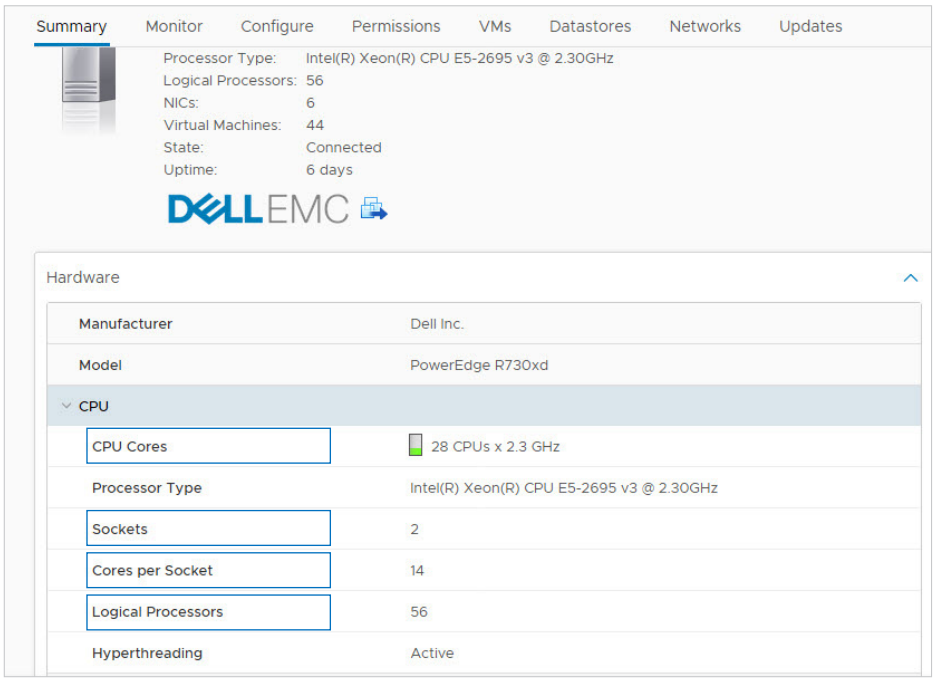
3.5.1 Physical, Virtual, and Logical CPU and Core

Let us start with the terminology first. VMware uses following terms to distinguish between processors within a VM and underlying physical x86/x64-based processor cores:

- Physical CPU (pCPU) or physical socket: Physical CPU installed in the server hardware. Refers to the "Sockets" on the Figure 8.
- Physical Core (pCore): Independent processing unit residing on the same processor[19]. Refers to the "Cores per Socket" and "CPU Cores" on the Figure 8.
- Logical Core (lCore): Logical processor on a physical core with own processor architectural state. Refers to the "Logical Processors" on the Figure 8. Most know implementation is Intel Hyper-Threading technology (HT)[20]. For more details see section 3.5.3.

As an example, the host listed on the Figure 8 has two pSocket (two pCPUs), 28 pCores, and 56 logical Cores as a result of an active Hyper-Threading.

Figure 8.
Physical Server CPU Allocation



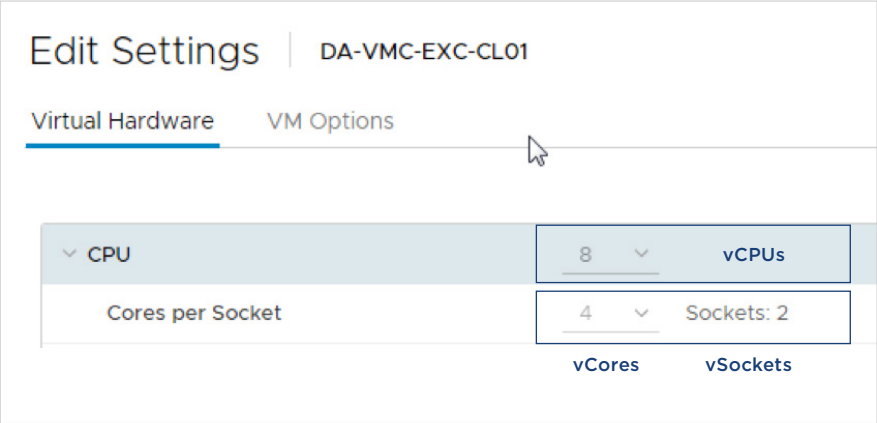
[19] More details here https://en.wikipedia.org/wiki/Multi-core_processor

[20] More details: https://en.wikipedia.org/wiki/Hyper-threading

3.5.1.2 VIRTUAL MACHINE

- Virtual Socket: Each virtual socket represents a virtualized physical CPU and can be configured with one or more virtual cores. Refers to the "Sockets" on the Figure 9.
- Virtual Core: Each virtual Core is equal to a CPU and will be visible by an OS as a separate processor unit[21]. Refers to the "Cores per Socket" on the Figure 9.
- Virtual CPU (vCPU): Virtualized central processor unit assigned to a VM. Refers to the "CPU" on the Figure 9. Total number of assigned vCPUs to a VM is calculated as:
  - Total vCPU = (Number of virtual Socket) * (Number of virtual Cores per socket)

Figure 9.
CPU Configuration of a VM



As an example, a VM listed in the Figure 9, has 2 virtual Sockets, each with 4 virtual Cores, with total number of vCPUs being 8.

3.5.2 Allocating vCPU

When performance is the highest priority of the SQL Server design, VMware recommends that, for the initial sizing, the total number of vCPUs assigned to all the VMs be no more than the total number of physical, not logical, cores available on the ESXi host machine. By following this guideline, you can gauge performance and utilization within the environment until you can identify potential excess capacity that could be used for additional workloads. For example, if the physical server that the various SQL Server workloads currently run on equates to 16 physical CPU cores, avoid allocating more than 16 virtual vCPUs for the VMs on that vSphere host during the initial virtualization effort.

Taking a more conservative sizing approach helps rule out CPU resource contention as a possible contributing factor in the event of sub-optimal performance when virtualizing SQL Server implementations. After you have determined that there is excess capacity to be used, you can consider increasing density by adding more

[21] Introduced in vSphere 4.1

workloads into the vSphere cluster and allocating virtual vCPUs beyond the available physical cores. Consider using monitoring tools capable to collect, store and analyze mid- and long-terms data ranges.

Lower-tier SQL Server workloads typically are less latency sensitive, so in general the goal is to maximize use of system resources and achieve higher consolidation ratios rather than maximize performance. The vSphere CPU scheduler's policy is tuned to balance between maximum throughput and fairness between VMs. For lower-tier databases, a reasonable CPU overcommitment can increase overall system throughput, maximize license savings, and continue to maintain adequate performance.

3.5.3 Hyper-Threading[22]

Hyper-threading is an Intel technology that exposes two hardware contexts (threads) from a single physical core, also referred to as logical CPUs. This is not the same as having twice the number of CPUs or cores. By keeping the processor pipeline busier and allowing the hypervisor to have more CPU scheduling opportunities, Hyper-threading generally improves the overall host throughput anywhere from 10 to 30 percent, allowing to use 1,1 to 1,3 vCPU:pCPU ratio for your VMs. Extensive testing and monitoring tools are required when following this approach.

VMware recommends enabling Hyper-threading in the BIOS/UEFI so that ESXi can take advantage of this technology. ESXi makes conscious CPU management decisions regarding mapping vCPUs to physical cores, taking Hyper-threading into account. An example is a VM with four virtual CPUs. Each vCPU will be mapped to a different physical core and not to two logical threads that are part of the same physical core.

3.5.4 Cores per Socket

As it's still very common to use Cores per socket setting to ensure that SQL Server Standard Edition will be able to consume all allocated vCPUs and can use up to 24 cores[23], care should be taken to get the right vNUMA topology exposed to a VM, especially on the vSphere 6.0 and below while satisfying the licensing needs.

As a rule of thumb, try to reflect your hardware configuration while configuring cores per socket ratio and revise the NUMA section (3.6) of this document for further details.

3.5.5 CPU Hot Plug

CPU hot plug is a feature that enables the VM administrator to add CPUs to the VM without having to power it off. This allows adding CPU resources "on the fly" with no disruption to service. When CPU hot plug is enabled on a VM, the vNUMA capability is disabled[24].

SQL Server Enterprise Edition supports adding a CPU in this way from version 2008 and up; it is not supported by Standard Edition. However, if a CPU is added, it will

[22] See additional information about Hyper-threading on a vSphere host in *VMware vSphere Resource Management* https://docs.vmware.com/en/VMware-vSphere/6.7/vsphere-esxi-vcenter-server-67-resource-management-guide.pdf

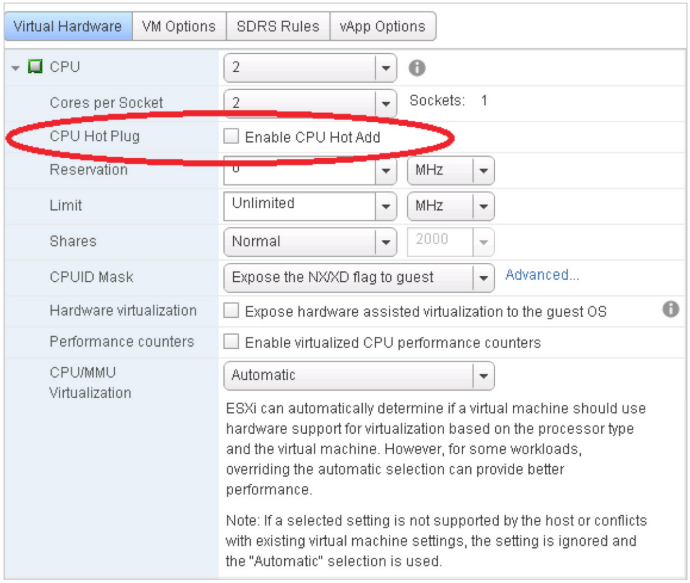
[23] https://docs.microsoft.com/en-us/sql/sql-server/compute-capacity-limits-by-edition-of-sql-server?view=sql-server-2017

[24] See the Knowledge Base article, *vNUMA is disabled if VCPU hot plug is enabled* (2040375) at http://kb.vmware.com/kb/2040375.

affect the vNUMA topology and might have degraded performance because the NUMA architecture does not reflect that of the underlying physical server.

Therefore, VMware recommends to not enable CPU hot plug by default, especially for VMs that require vNUMA. Rightsizing the VM's CPU is always a better choice than relying on CPU hot plug. The decision whether to use this feature should be made on a case-by-case basis and not implemented in the VM template used to deploy SQL.

Figure 10.
Disabling CPU Hot Plug (Uncheck Enable CPU Hot Add Checkbox)



As shown on the Figure 11, the *vmdumper* command clearly indicates that for a running VM with a feature "CPU Hot Add", exposing of vNUMA topology will be disabled.

Figure 11.
The vmdumper Command Provided VM Configuration for a VM with "CPU Hot Add" Enabled

```
DICT                numvcpus = "4"
DICT                 memSize = "4096"
DICT             displayName = "TSALAB-DC01"
DICT              vcpu.hotadd = "TRUE"
numaHost: NUMA config: consolidation= 1 preferHT= 0
numa: Hot add is enabled and vNUMA hot add is disabled, forcing UMA.
numaHost: 4 VCPUs 1 VPDs 1 PPDs
numaHost: VCPU 0 VPD 0 PPD 0
numaHost: VCPU 1 VPD 0 PPD 0
numaHost: VCPU 2 VPD 0 PPD 0
numaHost: VCPU 3 VPD 0 PPD 0
```

3.5.6 CPU Affinity

CPU affinity restricts the assignment of a VM's vCPUs to a subset of the available physical cores on the physical server on which a VM resides.

VMware recommends *not using CPU affinity* in production because it limits the hypervisor's ability to efficiently schedule vCPUs on the physical server. It's also disable the ability to vMotion a VM.

3.5.7 Per Virtual Machine EVC Mode[25]

vSphere 6.7 introduces a new feature, the ability to configure the EVC mode for a particular VM instead of the whole cluster (see section 3.3.3 for more details). The per-VM EVC mode determines the set of host CPU features that a VM requires in order to power on and migrate. The EVC mode of a VM is independent from the EVC mode defined at the cluster level.

Settings the EVC mode as a VM attribute on a VM hosting SQL Server instance can help to prevent downtime while migrating a VM between datacenters/vCenters or to a public cloud, such as VMC.

NOTE: Configuring EVC mode will reduce the list of CPU features exposed to a VM and might affect performance of SQL Server

NOTE: Virtual hardware 14 is required in order to enable the EVC mode as a VM attribute. All hosts must support a VM running this compatibility mode and be at least on vSphere version 6.7.

3.6 NUMA Considerations

Over last decade not so many topics has raised so much attention as discussions about Non-uniform memory access (NUMA) technologies and its implementation. This is expected considered complexity of the technology, particular vendor implementations, number of configurations options and layers (from a hardware through a hypervisor to a Guest OS and an application). Considering NUMA hardware architecture is a must for any infrastructure architect or SQL Server DBA in charge of a virtualized SQL Server.

3.6.1 Understanding NUMA[26]

NUMA is a hardware architecture for shared memory implementing subdivision of physical memory bunks between pCPUs (see Figure 12 for one of the possible implementations). In this term, local memory (being on the same bus as a pCPU) and remote memory (being accessed through an interconnect) concept is introduced. Subdivision of memory was dictated by the rapidly growing number of memory consumers (CPU cores), faster operations mode of cores and excessive cache coherence traffic when two or more cores accessing the same memory cacheline[27]. A Construct containing a pCPU, local memory and I/O modules located on the same bus is called a NUMA Node.

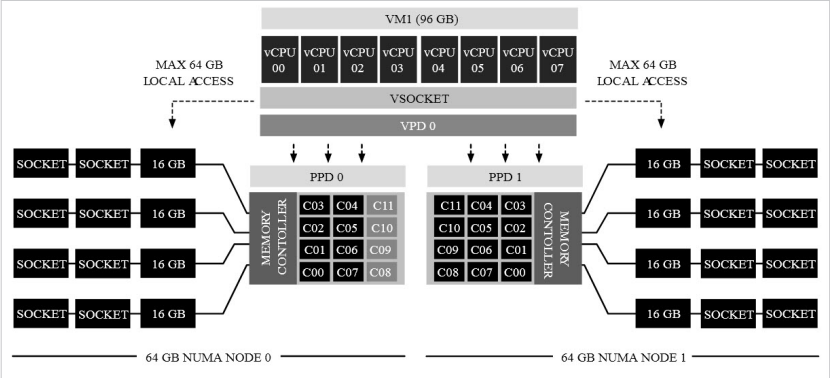
---

[25] More details: https://docs.vmware.com/en/VMware-vSphere/6.7/com.vmware.vsphere.vm_admin.doc/GUID-EE6F4E5A-3BEA-43DD-9990-DBEB0A280F3A.html

[26] This section uses the information from the research made by Frank Denneman and available here: http://frankdenneman.nl/2016/07/07/numa-deep-dive-part-1-uma-numa/

[27] https://events.static.linuxfound.org/sites/events/files/slides/Optimizing%20Application%20Performance%20in%20Large%20Multi-core%20Systems_0.pdf

Figure 12.
Intel-based NUMA Hardware Architecture[28]



This architecture having ultimate benefits also poses some trade-offs that needs to be considered and the most important of them—the time to access data in memory varies depending on local or remote placement of the corresponding memory cacheline to a CPU core executing the request, with remote access being up to X[29] times slower than local. This is what has given the name non-uniform to the whole architecture and is the primary concern for any application deployed on top of a hardware implementing NUMA.

We will go through different layers of NUMA implementation and will provide best practices suited for most of SQL Server workloads running on vSphere. As not all workloads are the same, extensive testing and monitoring are highly recommended for any particular implementation. Special high-performance optimized deployments of SQL Server may require usage of custom settings being outside of these general guidelines.

3.6.2 Using NUMA: Best Practices

As mentioned in the previous section, using of NUMA architecture may provide positive influence on the performance of an application, if this application is NUMA-aware. SQL Server has a native NUMA support starting with the version SQL Server 2005 (to some extent available to SQL Server 2000 SP3 as well)[30], that ultimately means that almost all recent deployments of modern SQL Servers will benefit from the right configured NUMA presentation.

NOTE: SQL Server Enterprise edition is required to utilize NUMA awareness. Consider avoiding the wide NUMA configuration on any other version of SQL Server[31].

Taking this fact, let us walk through how we can ensure that the correct and expected NUMA topology will be presented to an instance of the SQL Server running on a virtual machine.

---

[28] The figure is cited from: http://frankdenneman.nl/2017/10/05/vm-memory-config-exceeds-memory-capacity-physical-numa-node/

[29] Depending on the implementation and the processor family, this difference could be up to 3X (Source: https://events.static.linuxfound.org/sites/events/files/slides/Optimizing%20Application%20Performance%20in%20Large%20Multi-core%20Systems_0.pdf, p.6.

[30] https://blogs.msdn.microsoft.com/slavao/2005/08/02/sql-server-2005-numa-support-troubleshooting/

[31] https://docs.microsoft.com/en-us/sql/sql-server/editions-and-components-of-sql-server-2016?view=sql-server-2017

3.6.2.1 PHYSICAL SERVER

The NUMA support is dependent on the CPU architecture and was introduced first by AMD in Opteron series and then by Intel Nehalem processor family back to the year 2008. Nowadays, almost all server hardware currently available on the market uses NUMA architecture and NUMA is usually enabled by default in the BIOS of a server. In spite of this, it's recommended to check if a BIOS Settings was not modified. Most of hardware vendors will call this settings "Node interleaving" (HPE, Dell) or "Socket interleave" (IBM) and this setting should be set to "disabled" or "Non-uniform Memory access (NUMA)[32] to expose the NUMA topology.

As rules of thumb, number of exposed NUMA nodes will be equal to the number of physical sockets for Intel processors[33] and will be 2x for the AMD processors. Check you server documentations for more details.

3.6.2.2 VMWARE ESXI HYPERVISOR HOST

vSphere supports NUMA on the physical server starting with version 2. Moving to the current version (6.7 as of time of writing this document), many configurational settings were introduced to help to manage a NUMA topology. As our ultimate goal is to provide clear guidelines on how the NUMA topology is exposed to a VM hosting SQL Server, we will skip describing all the advanced settings and will concentrate on the examples and relevant configuration required.

First step to achieve this goal will be to ensure that the physical NUMA topology is exposed correctly to an ESXi host. Use esxtop or esxcli and shed-stats to obtain this information:

esxcli hardware memory get | grep NUMA

sched-stats -t ncpus

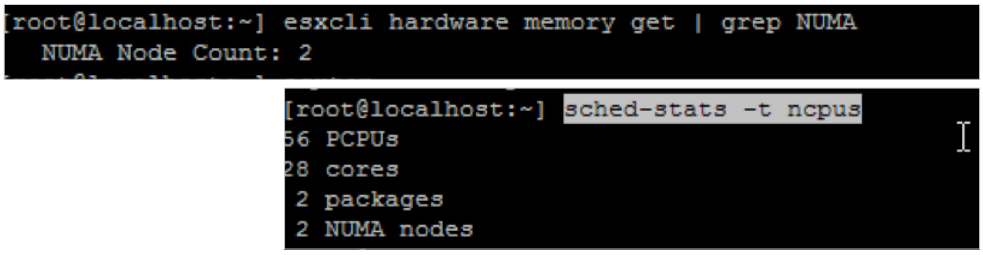


Figure 13. Using esxcli and Shed-stats Commands to Obtain the NUMA Node Count on an ESXi Host

or

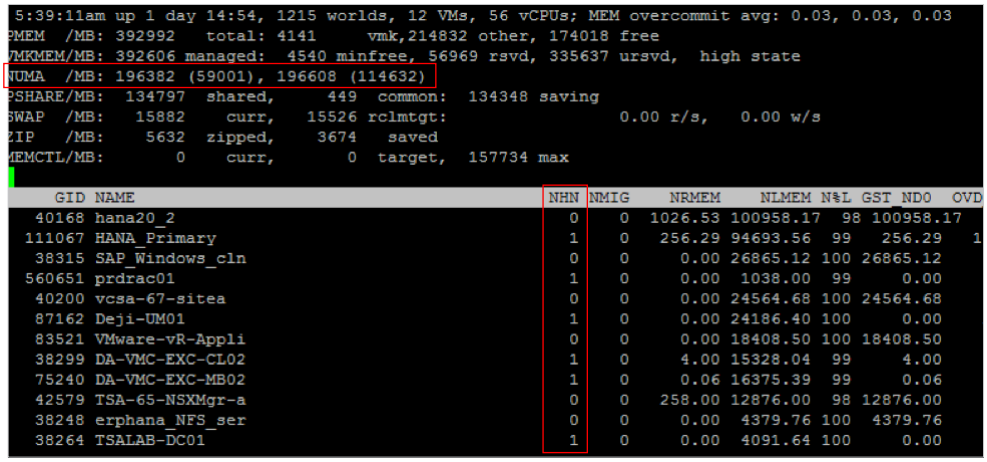
ESXTOP, press M for memory, F to adjust fields, G to enable NUMA stats[34],

[32] Refer to the documentation of the server hardware vendor for more details. Name and value of the setting could be changed or named differently in any particular BIOS/UEFI implementation

[33] If the snooping mode "Cluster-on-die" (CoD, Haswell) or "sub-NUMA cluster" (SNC, Skylake) is used with pCPU with more than 10 cores, each pCPU will be exposed as two logical NUMA nodes (https://software.intel.com/en-us/articles/intel-xeon-processor-scalable-family-technical-overview). VMware ESXi supports CoD starting with vSphere 6.0 and 6.6 U3b (https://kb.vmware.com/s/article/2142499)

[34] http://frankdenneman.nl/2016/08/22/numa-deep-dive-part-5-esxi-vmkernel-numa-constructs/

Figure 14.
Using esxtop to Obtain NUMA-related Information on an ESXi Host



If more than one NUMA node is exposed to an ESXi host, a "NUMA scheduler" will be enabled by the VMkernel. A NUMA home node (the logical representation of a physical NUMA node, exposing number of cores and amount of memory assigned to a pNUMA) and respectively NUMA clients (one per virtual machine per NUMA home node) will be created[35].

If the number of NUMA clients required to schedule a VM is more than one, such VM will be referenced as a "wide VM" and virtual NUMA (vNUMA) topology will be exposed to this VM starting with vSphere version 5.0 and later. This information will be used by a Guest OS and an instance of SQL Server to create the respective NUMA configuration. Hence it becomes very important to understand how vNUMA topology will be created and what settings can influence it.

As the creation of vNUMA topology for a VM will follow different logic starting with the vSphere 6.5, let us analyze it separately and use examples to show the difference. All settings are treated with the default values for the respective version of vSphere if not mentioned otherwise:

General Rules (applies to all versions of vSphere starting with 5.0):

- vNUMA is not exposed for any VM having less than nine (9) vCPU assigned (default).
- vNUMA is not exposed to any VM having less vCPU than the size of pNUMA of a host (default).
- vNUMA is not exposed if the "CPU hot add" feature is enabled (see the Section 3.5.5).

[35] See http://frankdenneman.nl/2016/08/22/numa-deep-dive-part-5-esxi-vmkernel-numa-constructs/ for more details

- The VM memory size is not considered for the creation of the vNUMA topology. For the "unbalanced NUMA" memory configuration (amount of configured memory span NUMA nodes while vCPU count stays within a NUMA node) see recommendation on the Section 3.6.2.4 and consider using Advanced Configuration#1 (page 32).
- vNUMA topology for a VM is created only once and by default is not updated if a VM is vMotioned to a server hardware with a different pNUMA configuration.
- VM hardware version 8 is required to have vNUMA exposed to the Guest OS.
- vNUMA topology will be updated if changes for the CPU configuration of a VM is done. pNUMA information from the host, where the VM was started at the time of the change will be used for creating vNUMA topology. Changing memory configuration will have no effect on vNUMA topology.

vSphere Version 6.0 and Earlier (5.x)

The vNUMA topology is directly specified by using the "Cores per Socket" setting in the virtual machine configuration. Number of sockets assigned will dictate the number of NUMA clients created. As a best practice, reflect you server hardware pNUMA topology configuring the cores:socket ratio. One caveat with using this setting is the affiliation with the licensing when only defined amounts of sockets will be accessed by the Guest OS/Application (consider limitations for the SQL Server Standard edition, see section 3.5.4).

Example:

A server with total of 16 CPU cores and 192 GB RAM (8 cores and 96 GB of RAM in each pNUMA node) is used to host a VM with 16 vCPU. "Cores per Socket" is set to two (Figure 15).

As a result the vNUMA topology with eight (8) NUMA nodes is exposed to the VM, which could be suboptimal.

Figure 15.
VM Cores per Socket Configuration

vSphere Version 6.5 and Later

Autosizing of the vNUMA is introduced. "Cores per Socket" setting is not taken into account while creating the vNUMA topology. The final vNUMA topology for a VM will be automatically configured by ESXi using the number of physical cores per CPU package of the physical host where VM is about to start. The total number of vCPU assigned to a VM will be consolidated in the minimal possible number of proximity domains (PPD), equal in size of a CPU package. The auto-created vNUMA topology will be saved in the VM configuration file. In most cases using autosizing will create optimized. Do not disable or modify vNUMA autosizing without clear use cases require administrator to do so. Consult the configuration examples section below to identify the best possible configuration.

Example: Same host and same VM with a vCPU configuration shown on the Figure 15 is used. As a result, the two (2) vNUMA nodes will be exposed to the VM.

vNUMA Configuration Examples[36]

For all examples a server that has a total of 24 pCPU cores and 192 GB RAM is used (12 physical cores and 96 GB of RAM in each pNUMA node). If not listed otherwise, no advanced settings were modified. Configuration listed will present the best possible vNUMA topology.

Standard-sized VM Configuration Examples

This section will cover standard sized VMs used to host SQL Server. Use configurations examples listed in the Table 2 to properly assign vCPUs to a VM.

Table 2. Standard VM Configuration: Recommended vCPU Settings for Different Number of vCPU

	Configuration 1	Configuration 2	Configuration 3
Desired VM Configuration	8 vCPU, 32 GB RAM	10 vCPU, 64 GB RAM	16 vCPU, 128 GB RAM
VM vCPU#	8	10	16
VM cores per socket	8	10	8
VM Socket	1	1	2
Advanced settings	NO	NO	NO
Different between 6.0/6.5	NO	YES	NO
vNUMA	1	1	2
Memory	32	96	128

Configuration #1: Covers all use cases where the assigned number of vCPUs per VM is eight[37] (8) or below and required amount of memory stays within one pNUMA node. Expected behavior for such configuration: no vNUMA topology will be exposed to a VM. The ESXi CPU scheduler will execute all vCPUs inside of one pNUMA node.

Configuration #2: Covers use cases where number of vCPUs is nine (9) or more, but lower than number of physical cores in a pNUMA of a server (twelve (12) in our scenario). Expected behavior for such VM will be to stay within one pNUMA with no vNUMA topology exposed to a VM. For a VM deployed on vSphere version 6.0 and below its mandatory that number of cores per socket will be equal to the number of vCPU assigned to a VM and only one socket will be exposed to a VM. Otherwise, undesired wide NUMA configuration will be created.

Configuration #3: Covers use cases where the number of vCPUs assigned to a VM is bigger than the number of cores in one pNUMA (more than twelve (12) in our example). We need to mimic the hardware socket/cores configuration for the best performance. If a server has two physical sockets, ensure that total number of sockets exposed will be no more than two. For vSphere 6.5 and later the ESXi will automatically take care of the configuration in the most efficient way.

[36] Check https://blogs.vmware.com/performance/2017/03/virtual-machine-vcpu-and-vnuma-rightsizing-rules-of-thumb.html for more details.

[37] Maximum number of vCPU to expose vNUMA is not reached

Advanced vNUMA VM Configuration Examples

These special configurations are listed for references and if not required, one of the three configuration options from the Table 2 should be used. All configuration listed in the Table 3 will require adding advanced settings[38] for a VM being configured. Do not modify host level advanced NUMA settings.

Table 3.
Advanced vNUMA VM Configurations: Recommended vCPU Settings

	Configuration 1	Configuration 2	Configuration 3
Desired VM Configuration	8VCPUs, 128 GB RAM	20vCPU, 64 GB RAM	10vCPU, 64 GB RAM
VM vCPU#	8	20	10
VM cores per socket	4	20	5
VM Socket	2	1	2
Advanced settings	YES	YES	YES
Different between 6.0/6.5	YES	NO	YES
vNUMA	2	1	2
Memory	128	64	64

Configuration #1: Covers use cases where so called "unbalanced" vNUMA topology is required (number of assigned vCPUs is within one NUMA node, but amount of memory required exceeds one pNUMA). For optimal performance, at least two vNUMA nodes should be exposed to a VM.

In order to enable the desired vNUMA topology the advanced setting *vcpu.maxPerMachineNode*[39] should be used which will specify the maximum number of vCPUs that will be scheduled inside one pNUMA. Set this value to the half of the total vCPUs assigned to a VM (it will be five (5) in the example used):

numa.vcpu.maxPerMachineNode = (Number of vCPU assigned to a VM)/2 = 5

Than reflect the required number of vNUMA nodes (two in our cases, as 128 GB RAM is not available in one pNUMA node) in the number of sockets by configuring

cpuid.coresPerSocket = numa.vcpu.maxPerMachineNode = 5

Configuration #2: Covers use cases where amount of required vCPUs for a VM is higher than a pNUMA size, but the assigned memory will fit in one pNUMA node and lots of inter-thread communications happens. For such configuration it might be beneficial to logically increase the size of a pNUMA node used for generating the vNUMA topology by taking into account Hyper-threaded threads (logical cores).

[38] Use this procedure to add a VM advanced settings: https://docs.vmware.com/en/VMware-vSphere/6.7/com.vmware.vsphere.vcenterhost.doc/GUID-62184858-32E6-4DAC-A700-2C80667C3558.html. Do not modify a .vmx file manually!

[39] Despite being often mentioned, *numa.vcpu.min* will have no effect as the size of pNUMA is still enough to accommodate all vCPUs. This advanced settings is useful on old processors where total number of cores in a pNUMA was lower than eight (8).

To achieve this, the following VM advanced settings should be added for all vSphere version:

numa.vcpu.preferHT = True
cpuid.coresPerSocket = 20

Based on our example, size of pNUMA will be set for 24 and a VM with 20 vCPU will be placed in one NUMA node. Due to cores per socket number extended to the total vCPUs assigned, all the cores will be presented as sharing the same processor cache (opposite to cores per socket set to 1, where each core will have separate cache) which could lead to the cache optimization[40]. This configuration requires extensive testing and monitoring if implemented. Use with caution.

Configuration #3: Covers use cases where it's desired to split the vCPUs between NUMA nodes even if the total vCPU count on a VM will feed into one pNUMA. It might optimize the work of the ESXi NUMA- and CPU schedulers in a cluster with high VM population ratio or it might be preferable over SQL Server Soft-NUMA[41] due to bounding vCPU with the respective memory. This configuration might make sense for a physical CPU with core count being nine (9) and above. This configuration require testing with the exact production workload and should be treated as exception.

To achieve this configuration, on vSphere 6.0 and early configure "Cores per Socket" so, that number of exposed sockets will be equal to two (it will be 5 cores per socket in our example).

For the vSphere version 6.5 and later in addition to configuring "Cores per Socket", the following VM advanced setting need to be added to disable the autosizing:

numa.vcpu.followcorespersocket = 1

Special Consideration for a Cluster with Heterogeneous Hardware or When Moving a VM Hosting an Instance of SQL Server to a Different vSphere Compute Cluster

The important consideration for such kind of configuration changes is – vNUMA topology for a VM by default is generated once and will not be changed upon vMotion or a power cycle of a VM.

If a VM resides in a cluster with heterogeneous hardware and a different size of pNUMA across ESXi hosts, either enable reconfiguration of vNUMA topology by each power cycle of a VM[42] or configure static vNUMA representation following the Configuration#1 of the advanced vNUMA configurations (listed as the Configuration#1 in Table 3). In general, such cluster configurations should be strictly avoided in production environments.

[40] More details: http://frankdenneman.nl/2016/08/22/numa-deep-dive-part-5-esxi-vmkernel-numa-constructs/

[41] https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/soft-numa-sql-server?view=sql-server-2017

[42] Add *numa.autosize.once = FALSE* and *numa.autosize = TRUE* to a VM configuration. Use with caution, as for the guest OS it's not normally expected, that the NUMA topology could be dynamically adjusted.

If a VM needs to be migrated to another cluster or moved to a new hardware, one-time reconfiguration of the vNUMA topology might be required. In this case, it's recommended to review the vNUMA topology required, recheck the hardware specification of physical hosts and then use Tables 2 and 3 with the description of use cases to make a decision of what vNUMA topology should be used. If one of the standard vNUMA topologies is not sufficient, after a VM is migrated, make a slight change to a vCPU assignment (for example, add/remove two cores) and then revert back to the original configuration. This change will instruct an ESXi host to recreate the vNUMA topology. Double check the final vNUMA topology.

Check vNUMA Topology Exposed to a VM

After the desired vNUMA topology is defined and configured, power on a VM and recheck how the final topology looks like. The following command on the ESXi host hosting the VM might be used[43]:

vmdumper -l | cut -d \/ -f 2-5 | while read path; do egrep -oi "DICT.(displayname.*|numa.*|cores.*|vcpu.*|memsize.*|affinity.*)= .*|numa:.*|numaHost:.*" "/$path/vmware.log"; echo -e; done*

Figure 16.
Checking NUMA Topology with the *vmdumper* Command

```
DICT                  numvcpus = "8"
DICT                   memSize = "153600"
DICT               displayName = "ERP_HANA3_2"
DICT numa.autosize.vcpu.maxPerVirtualNode = "8"
DICT      numa.autosize.cookie = "80001"
numaHost: NUMA config: consolidation= 1 preferHT= 0
numaHost: 8 VCPUs 1 VPDs 1 PPDs
numaHost: VCPU 0 VPD 0 PPD 0
numaHost: VCPU 1 VPD 0 PPD 0
numaHost: VCPU 2 VPD 0 PPD 0
numaHost: VCPU 3 VPD 0 PPD 0
numaHost: VCPU 4 VPD 0 PPD 0
numaHost: VCPU 5 VPD 0 PPD 0
numaHost: VCPU 6 VPD 0 PPD 0
numaHost: VCPU 7 VPD 0 PPD 0

[root@WDC-ESX17:~]
```

VM vNUMA Sizing Recommendation

Despite the fact that the introduction of vNUMA helps a lot to overcome issues with multicores VMs, following best practices should be considered while architecting a vNUMA topology for a VM.

[43] Special thanks to V.Bondizo, Sr. Staff TSE, VMware, for sharing this vmdumper command

a. Best possible performance generally is observed when a VM could fit into one pNUMA node and could benefit from the local memory access. For example, when running a VM hosting SQL Server on a host with twelve pCores per pNUMA, as a rule of thumb, assign no more than twelve vCPUs to a VM. This statement especially applies for SQL Servers licensed with the Standard Edition, which has no NUMA optimization.

b. If a wide-NUMA configuration is unavoidable, consider reevaluating the recommendations given and execute extensive performance testing before implementing the configuration. Monitoring should be implemented for important CPU counters after moving to the production.

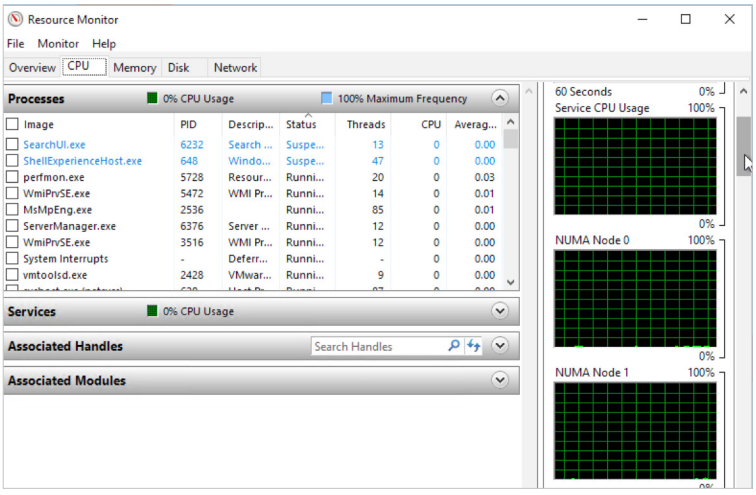
3.6.2.3. GUEST OPERATING SYSTEM

Current editions of SQL Server can be installed successfully on both the Windows and Linux OSs. Disregarding the specific Guest OS in use, the most important part of NUMA configuration on this layer will be to recheck the exposed vNUMA topology and compare it with the expectations set. If the exposed NUMA topology is not expected or not desired, changes should be made on the vSphere layer and not on the Guest OS. If it will be justified that the changes are required, bear in mind that it's not enough to restart a VM. Refer to the previous section to find, how vNUMA topology can be adjusted.

Windows OS: Check NUMA Topology

Using Windows Server 2016, the required information might be obtained through the Resource monitor.

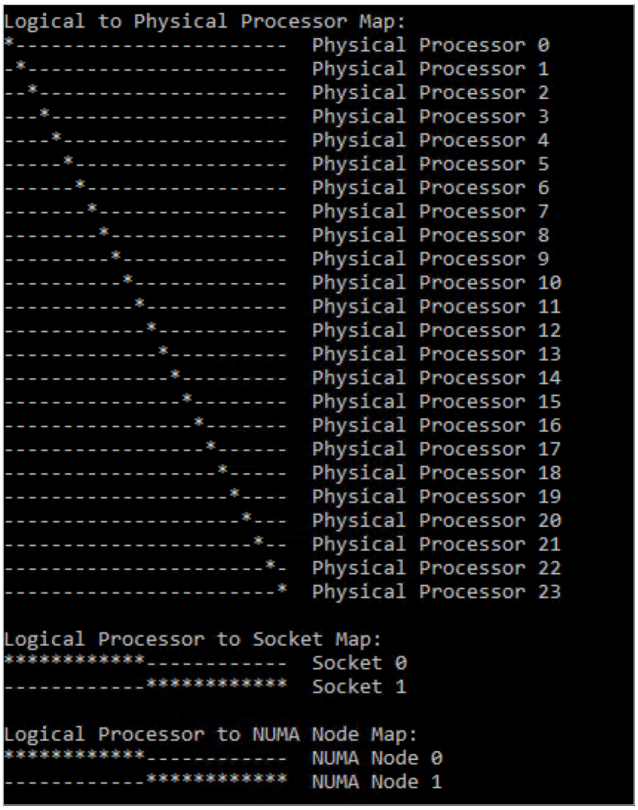
Figure 17. Windows Server 2016 Resource Monitor Exposing NUMA Information



A more comprehensive alternative would be using the coreinfo tool, available from Sysinternals[44].

[44] https://docs.microsoft.com/en-us/sysinternals/downloads/coreinfo

Figure 18.
Output of *coreinfo* Command Showing a NUMA Topology for 24 cores/2socket VM



Linux OS: Checking NUMA Topology

Since the SQL Server 2017, it's supported to run SQL Server on chosen Linux OSs like Red Hat or SuSe[45]. The following utilities can be used to check the NUMA topology exposed.

• numactl[46]

This utility provides comprehensive information about the NUMA topology and gives the ability to modify NUMA settings if required. Run the following command to get the required information:

numactl –hardware

[45] For a full list of the supported OS, check https://docs.microsoft.com/en-us/sql/linux/sql-server-linux-release-notes?view=sql-server-linux-2017

[46] Numactl is not available in the standard OS and should be installed using yum on Red Hat

Figure 19.
Using the numactl Command to Display the NUMA Topology

```
[root@O-FCI-Node1 ~]# numactl --hardware
available: 2 nodes (0-1)
node 0 cpus: 0 1 2 3 4 5 6 7
node 0 size: 2047 MB
node 0 free: 1648 MB
node 1 cpus: 8 9 10 11 12 13 14 15
node 1 size: 2047 MB
node 1 free: 1532 MB
node distances:
node   0   1
  0:  10  20
  1:  20  10
```

/var/log/dmesg with dmesg tool:

Figure 20.
Using dmesg Tool to Display the NUMA Topology

```
[root@O-FCI-Node1 ~]# dmesg | grep -i numa
[    0.000000] NUMA: Node 0 [mem 0x00000000-0x0009ffff] + [mem 0x00100000-0x7fff
ffff] -> [mem 0x00000000-0x7fffffff]
[    0.000000] NUMA: Node 1 [mem 0x80000000-0xbfffffff] + [mem 0x100000000-0x13f
ffffff] -> [mem 0x80000000-0x13fffffff]
[    0.000000] Enabling automatic NUMA balancing. Configure with numa_balancing=
 or the kernel.numa_balancing sysctl
```

Ensure to check that acpi is **not** turned off (this will disable NUMA as well): grep acpi=off /proc/cmdline

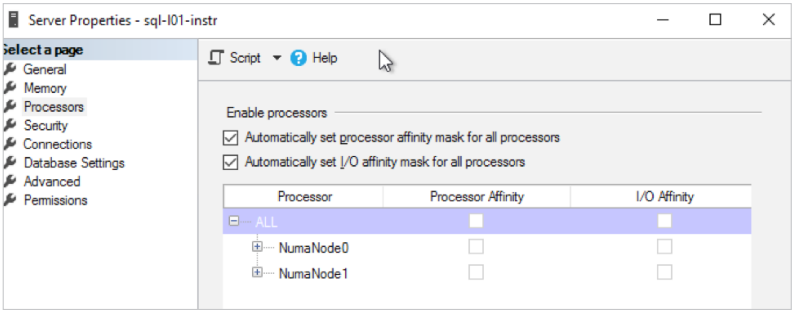
3.6.2.4 SQL SERVER

The last part of the process is to check the NUMA topology that is exposed to the instance of a SQL Server. As mentioned, SQL Server is a NUMA-aware application and require correct NUMA topology to be exposed to use it efficiently.

NOTE: SQL Server Enterprise Edition is required to benefit from NUMA topology.

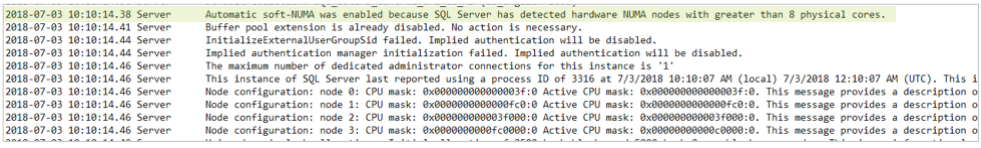
From the SQL Server Management Studio the NUMA topology could be seen in the processor properties of the server instance:

Figure 21.
Displaying the NUMA Information in the SQL Server Management Studio



Additional information can be obtained from the *errorlog* file after the restart of the database service.

Figure 22.
Errorlog Messages for Automatic Soft-NUMA on 12 Cores per Socket VM



SQL Server Automatic Soft-NUMA and vNUMA

The SQL Server Soft-NUMA feature was introduced to react on the growing number of cores per pCPU. Soft-NUMA aims to partition available CPU resources inside one NUMA node into so called "Soft-NUMA" nodes. The "Soft-NUMA" feature has no contradiction with vNUMA topology exposed to a VM, but might further optimize scalability and performance of the database engine for most of the workload[47].

Starting with the SQL Server 2014 SP2 and in all later versions (2016, 2017) the "Soft-NUMA" is enabled by default and does not require any modification of the registry or service flags for the database service startup. In SQL Server 2016 and 2017 the upper boundary for starting the partitioning and enabling "Soft-NUMA" is eight (8) cores per NUMA node.

The result of having automatic Soft-NUMA enabled can be noticed in the errorlog (Figure 22) and using the DMV *sys.dm_os_nodes* (Figure 23).

Figure 23. sys.dm_os_nodes Information on a System with Two NUMA Nodes and Four Soft-NUMA Nodes

Results | Messages

	node_id	node_state_desc	memory_node_id	cpu_count
1	0	ONLINE	0	6
2	1	ONLINE	0	6
3	2	ONLINE	1	6
4	3	ONLINE	1	2
5	64	ONLINE DAC	64	6

Soft-NUMA partitions only CPU cores and does not provide the memory to CPU affinity[48]. It will be only one lazy writer and buffer pool created per underlying NUMA node (for the configuration shown, it will be two (2) lazy writers and four (4) soft-NUMA nodes).

If using Soft-NUMA does not provide a performance benefit for the workload, it might be recommended to check if vNUMA topology could be changed to expose smaller vNUMA to a VM hosting such instance of SQL Server.

3.7 Virtual Machine Memory Configuration

One of the most critical system resources for SQL Server is memory. Lack of memory resources for the SQL Server database engine will induce Windows Server to page memory to disk, resulting in increased disk I/O activities, which are considerably slower than accessing memory[49]. Lack of hypervisor memory resources results in memory contention that has significant impact on SQL Server performance.

[47] More details can be found here: https://blogs.msdn.microsoft.com/bobsql/2016/06/03/sql-2016-it-just-runs-faster-automatic-soft-numa/ and https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/soft-numa-sql-server?view=sql-server-2017

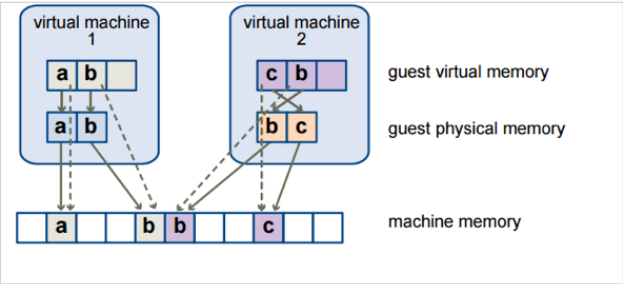
[48] https://blogs.msdn.microsoft.com/psssql/2010/04/02/how-it-works-soft-numa-io-completion-thread-lazy-writer-workers-and-memory-nodes/

[49] More details and architecture recommendation for SQL Server can be found here: https://docs.microsoft.com/en-us/sql/relational-databases/memory-management-architecture-guide?view=sql-server-2017

When a SQL Server deployment is virtualized, the hypervisor performs virtual memory management without the knowledge of the guest OS and without interfering with the guest OS's own memory management subsystem[50].

The guest OS sees a contiguous, zero-based, addressable physical memory space. The underlying machine memory on the server used by each VM is not necessarily contiguous.

Figure 24.
Memory Mappings Between Virtual, Guest, and Physical Memory



3.7.1 Memory Sizing Considerations

- When designing for performance to prevent the memory contention between VMs, avoid overcommitment of memory at the ESXi host level (HostMem >= Sum of (VMs memory + overhead). If a physical server has 256GB of RAM, do not allocate more than that amount to the VMs residing on it taking memory overhead into consideration as well.
- When collecting performance metrics for making a sizing decision for a VM running SQL Server, consider using SQL Server provided metrics (available in the DMV: sys.dm_os_process_memory). Do not use vSphere or Windows Guest OS provided memory metrics (for example, vSphere provided "memory consumed" or, especially, "memory active").
- Consider SQL Server edition-related memory limitations while assigning memory to a VM. For example, SQL Server 2017 Standard edition supports up to 128GB memory[51]. This does not account for what the OS will need, nor any other software that is required.
- Consider checking the hardware pNUMA memory allocation to identify the maximum amount of memory that can be assigned to a VM without crossing the pNUMA boundaries. See the Section 3.6.2.2 for more details and the instruction how to get the size of a pNUMA node.
- VMs require a certain amount of overhead memory to power on. You should be aware of the amount of this overhead. The following table lists the amount of overhead memory a VM requires to power on. After a VM is running, the amount of overhead memory it uses might differ from the amount listed in the Table 4.

[50] Following resource is highly recommended for deeper understanding of memory management in ESXi https://www.vmware.com/techpapers/2011/understanding-memory-management-in-vmware-vsphere-10206.html

[51] More details https://www.microsoft.com/en-us/sql-server/sql-server-2017-editions

Table 4.
Sample Overhead Memory on Virtual Machines[52]

Memory (MB)	1 VCPU	2 VCPUs	4 VCPUs	8 VCPUs
256	20.29	24.28	32.23	48.16
1024	25.90	29.91	37.86	53.82
4096	48.64	52.72	60.67	76.78
16384	139.62	143.98	151.93	168.60

3.7.2 Memory Reservation

When achieving adequate performance is the primary goal, consider setting the memory reservation equal to the provisioned memory. This will eliminate the possibility of ballooning or swapping from happening and will guarantee that the VM gets only physical memory. When calculating the amount of memory to provision for the VM, use the following formulas:

VM Memory = SQL Max Server Memory + ThreadStack + OS Mem + VM Overhead

*ThreadStack = SQL Max Worker Threads * ThreadStackSize*

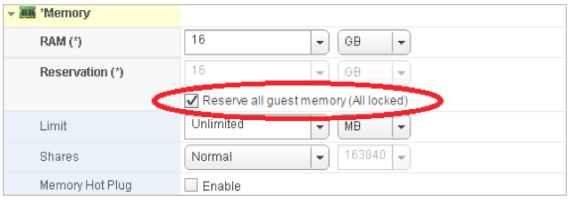
ThreadStackSize = 1MB on x86

= 2MB on x64

OS Mem: 1GB for every 4 CPU Cores

Use SQL Server memory performance metrics and work with your database administrator to determine the SQL Server maximum server memory size and maximum number of worker threads. Refer to the VM overhead (Table 4) for the VM overhead.

Figure 25.
Setting Memory Reservation



Setting memory reservations might limit vSphere vMotion. A VM can be migrated only if the target ESXi host has unreserved memory equal to or greater than the size of the reservation.

Reserving all memory will disable the creation of the swap file and will save the disk space especially for VMs with a large amount of memory assigned.

[52] For additional details, refer to *vSphere Resource Management* (https://docs.vmware.com/en/VMware-vSphere/6.7/vsphere-esxi-vcenter-server-67-resource-management-guide.pdf).

If the "Reserve all guest memory" checkbox is NOT set, it's highly recommended to monitor host swap related counters (swap in/out, swapped). Even if swapping is the last resort for a host to allocate physical memory to a VM and happens during congestion only, the swapped VM memory will stay swapped even if congestion conditions are gone. If, for example, during extended maintenance or disaster recovery situation, an ESXi host will experience memory congestion and if not all VM memory is reserved, the host will swap part of the VM memory. This amount memory will NOT be un-swapped automatically. If the swapped memory is identified, consider either to vMotion, or shut down and that power on a VM or use the *unswap* command[53].

3.7.3 The Balloon Driver

The ESXi hypervisor is not aware of the guest Windows Server memory management tables of used and free memory. When the VM is asking for memory from the hypervisor, the ESXi will assign a physical memory page to accommodate that request. When the guest OS stops using that page, it will release it by writing it in the OS's free memory table, but will not delete the actual data from the page. The ESXi hypervisor does not have access to the OS's free and used tables, and from the hypervisor's point of view, that memory page might still be in use. In case there is memory pressure on the hypervisor host, and the hypervisor requires reclaiming some memory from VMs, it will utilize the balloon driver. The balloon driver, which is installed with VMware Tools™[54], will then request a large amount of memory to be allocated from the guest OS. The guest OS will release memory from the free list or memory that has been idle. That way, memory is paged to disk based on the OS algorithm and requirements and not the hypervisor. Memory will be reclaimed from VMs that have less proportional shares and will be given to the VMs with more proportional shares. This is an intelligent way for the hypervisor to reclaim memory from VMs based on a preconfigured policy called the proportional share mechanism.

When designing SQL Server for performance, the goal is to eliminate any chance of paging from happening. Disable the ability for the hypervisor to reclaim memory from the guest OS by setting the memory reservation of the VM to the size of the provisioned memory. The recommendation is to leave the balloon driver installed for corner cases where it might be needed to prevent loss of service. As an example of when the balloon driver might be needed, assume a vSphere cluster of 16 physical hosts that is designed for a 2-host failure. In case of a power outage that causes a failure of 4 hosts, the cluster might not have the required resources to power on the failed VMs. In that case, the balloon driver can reclaim memory by forcing the guest OSs to page, allowing the important database servers to continue running in the least disruptive way to the business.

It's highly recommended to implement monitoring of the ballooned memory both on the host and VMs level. Use "ballooned memory" counter in vCenter Web Client to

[53] More details on the unswap command can be found here: http://www.yellow-bricks.com/2016/06/02/memory-pages-swapped-can-unswap/ Bear in mind, that this command is still officially not supported.

[54] VMware Tools must be installed on the guest; status of the tool service must be running and balloon driver must not be disabled

configure an alarm, or special tools like vRealize Operation Manager.

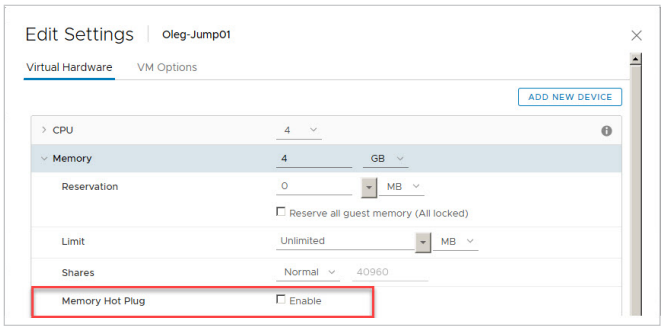
NOTE: Ballooning is sometimes confused with Microsoft's Hyper-V dynamic memory feature. The two are not the same and Microsoft recommendations to disable dynamic memory for SQL Server deployments do not apply for the VMware balloon driver.

3.7.4 Memory Hot Plug

Similar to CPU hot plug, memory hot plug enables a VM administrator to add memory to the VM with no down time. Before vSphere 6.5, when memory hot add was configured on a wide-NUMA VM, it would always be added it to the node0, creating NUMA imbalance[55]. With vSphere 6 and later, when enabling memory hot plug and adding memory to a VM, the memory will be added evenly to all vNUMA nodes which makes this feature usable for more use cases. VMware recommends using memory hot plug in cases where memory consumption patterns cannot be easily and accurately predicted only with vSphere 6 and later.

After memory has been added to the VM, increase the max memory setting in the database server properties if one has been set (Section 4.3.1). This can be done without requiring a server reboot or a restart of the SQL Server service. As with CPU hot plug, it is preferable to rely on rightsizing than on memory hot plug. The decision whether to use this feature should be made on a case-by-case basis and not implemented in the VM template used to deploy SQL Server.

Figure 26.
Setting Memory Hot Plug



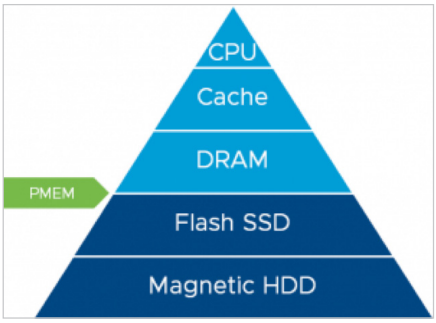
3.7.5 Persistent Memory[56]

Persistent Memory (PMem), also known as Non-Volatile Memory (NVM), is capable of maintaining data in memory DIMM even after a power outage. This technology layer provides the performance of memory with the persistence of traditional storage. Support of PMem was introduced in the vSphere version 6.7 and can be combined with a native PMem support in Windows Server 2016 and SQL Server 2016 SP1 and higher, increasing performance of high-loaded databases.

[55] To rebalance memory between vNUMA nodes, a VM should be powered off or moved with a vMotion to a different host.

[56] More information on using PMem for virtualized applications: https://blogs.vmware.com/apps/2018/07/accelerating-applications-performance-with-virtualized-pmem.html

Figure 27.
Positioning PMem



Persistent memory can be consumed by VMs in two different modes[57]:

- Virtual Persistent Memory (vPMem)[58] Using vPMem, the memory is exposed to a guest OS as a virtual NVDIMM. This enables the guest OS to use PMem in byte addressable random mode.
- Virtual Persistent Memory Disk (vPMemDisk) Using vPMemDisk, the memory can be accessed by the guest OS as a virtual SCSI device, but the virtual disk is stored in a PMem datastore.

Both modes could be profitable for a SQL Server deployment. Consider using vPMem when working with Windows Server 2016 guest OS and SQL Server 2016 SP1 and above. For this combination, after a vPMem device is exposed to the VM with Windows Server 2016 OS, it will be detected as a Storage Class Module and should be formatted as a DAX volume. SQL Server can use the DAX volume to enable "tail-of-log-cache" by placing an additional log file on a SCM volume configured as DAX[59].

ALTER DATABASE <MyDB> ADD LOG FILE (NAME = <DAXlog>,

FILENAME = '<Filepath to DAX Log File>', SIZE = 20 MB

As only 20 MB of PMem space is required (SQL Server will use only 20MB to store the log buffer)[60], one PMem module could be efficiently shared between multiple VMs running on the same host, providing high saving costs by sharing physical NVDIMMs between many consumers.

vPMemDisk mode could be used with any versions of Windows/Linux Guest OS/SQL Server as a traditional block-based storage device, but with very low latency. Recent use cases demonstrated benefits of vPMemDisk for SQL Server backup and restore.[61]

NOTE: As of time of writing, a VM with PMem devices disregards of mode, will not be covered by vSphere HA and should be excluded from the VM level backup. vMotion of a VM with PMem attached is supported (for vPMem mode destination host must have a physical NVDIMM).

[57] See https://docs.vmware.com/en/VMware-vSphere/6.7/vsphere-esxi-vcenter-server-67-resource-management-guide.pdf for more details

[58] VM hardware version 14 and a guest OS that supports NVM technology must be used

[59] More details can be found here: https://blogs.msdn.microsoft.com/sqlserverstorageengine/2016/12/02/transaction-commit-latency-acceleration-using-storage-class-memory-in-windows-server-2016sql-server-2016-sp1/ and https://blogs.msdn.microsoft.com/bobsql/2016/11/08/how-it-works-it-just-runs-faster-non-volatile-memory-sql-server-tail-of-log-caching-on-nvdimm/

[60] Size of PMem device should be at least 100 MB to allow creation of GPT partition, which is the prerequisites of the DAX enabled SCM

3.8 Virtual Machine Storage Configuration

Storage configuration is critical to any successful database deployment, especially in virtual environments where you might consolidate multiple VMs with SQL Server on a single ESXi host or datastore. Your storage subsystem must provide sufficient I/O throughput as well as storage capacity to accommodate the cumulative needs of all VMs running on your ESXi hosts. In addition, consider changes when moving from a physical to virtual deployment in terms of a shared storage infrastructure in use[61].

Follow these recommendations along with the best practices in this guide. Pay special attention to this section, as many of virtualized SQL Server performance issues are caused by storage subsystem configuration.

3.8.1 vSphere Storage Options

vSphere provides several options for storage configuration. The one that is the most widely used is a VMware Virtual Machine File System (VMFS) formatted datastore on block storage system, but that is not the only option. Today, storage admins can utilize new technologies such as vSphere Virtual Volumes™ which take storage management to the next level, where VMs are native objects on the storage systems. Other options include hyper-converged solutions, such as VMware vSAN™. This section covers the different storage options that exist for virtualized SQL Server deployments running on vSphere.

3.8.1.1 VMWARE VIRTUAL MACHINE FILE SYSTEM

VMFS is a clustered file system that provides storage virtualization optimized for VMs. Each VM is encapsulated in a small set of files and VMFS is the default storage system for these files on physical SCSI based disks and partitions. VMware supports block storage (Fiber Channel and iSCSI protocols) for VMFS.

Consider using the highest possible VMFS version supported by ESXi hosts in the environment. VMFS version 6, used as the default in vSphere 6.5 and 6.7, brings many enhancements[62], among others:

- SEsparce snapshots
- Automatic storage space reclamation
- 4kn device support

Consider upgrading a VMFS datastore only after all ESXi hosts sharing access to a datastore are upgraded to the desired vSphere version.

NOTE: Strictly avoid placing a VM hosting SQL Server on a VMFS3 or VMFS3 upgraded datastores. It will affect the disk performance.

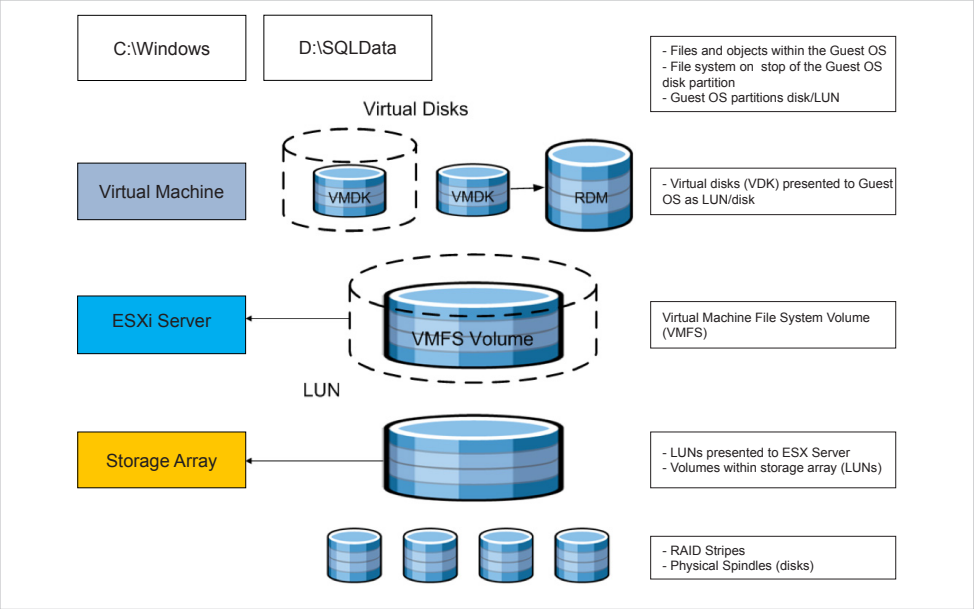
[61] For information about best practices for SQL Server storage configuration, refer to *Microsoft's Storage Top Ten Practices* (http://technet.microsoft.com/en-us/library/cc966534.aspx).

[62] Full list of features of VMFS 6 available here: https://docs.vmware.com/en/VMware-vSphere/6.7/com.vmware.vsphere.storage.doc/GUID-7552DAD4-1809-4687-B46E-ED9BB42CE277.html

3.8.1.2 VIRTUAL MACHINE FILE SYSTEM ON SHARED STORAGE SUBSYSTEM

This is still the most commonly used option today among VMware customers. As illustrated in the following figure, the storage array is at the bottom layer, consisting of physical disks presented as logical disks (storage array volumes or LUNs) to vSphere. This is the same as in the physical deployment. The storage array LUNs are then formatted as VMFS volumes by the ESXi hypervisor and that is where the virtual disks reside. These virtual disks are then presented to the guest OS.

Figure 28.
VMware Storage Virtualization Stack



3.8.1.3 NETWORK FILE SYSTEM[63]

A Network File system (NFS) client built into ESXi host uses the Network File System (NFS) protocol over TCP/IP to access a designated NFS volume that is located on a NAS server. The ESXi host can mount the volume and use it for its storage needs. vSphere supports versions 3 and 4.1 of the NFS protocol. The main difference from block storage is that NAS/NFS will provide file level access, the VMFS formatting is not required for NFS datastores.

[63] More details: https://docs.vmware.com/en/VMware-vSphere/6.7/com.vmware.vsphere.storage.doc/GUID-9282A8E0-2A93-4F9E-AEFB-952C8DCB243C.html

3.8.1.3.1 NFS DATASTORES CONSIDERATIONS

vSphere 6.7 supports up to 256 NFS datastore per ESXi host with defaults being eight (8). If more than eight (8) NFS datastores will be presented, increase relevant ESXi host advanced settings[64].

By default, the VMkernel interface with the lowest number will be used to access the NFS server. No special configuration exist to select a vmk used to access a NFS share. Ensure, that the NFS server is located outside of the ESXi management network (preferable, is a separate non-routable subnet) and that a separate VMkernel interface is created to access the NFS server.

Consider using at least 10 GbE physical network adapters to access the NFS server[65].

3.8.1.3.2 SQL SERVER SUPPORT ON NFS DATASTORE IN VIRTUALIZED ENVIRONMENT

If a datastore is a NFS datastore and VMDKs are used by a VM that has SQL Server, the guest knows nothing about NFS and this is a supported configuration since inside the guest things will look as they would if the VM was a physical server to Windows Server and SQL Server. The only caveat is that shared disks for FCI configurations are not supported using NFS datastores[66].

Inside the guest, SQL Server supports the use of network attached storage (NAS). In SQL Server 2008 and earlier, this requires the use of a trace flag and can be used natively in SQL Server 2008 R2 or later as documented in the Microsoft KB article "Description of support for network database files in SQL Server"[67]. SQL Server 2012 and later also supports SMB 3.0 shares for storing data and transaction log files.

3.8.1.4 RAW DEVICE MAPPING

Raw Device Mapping (RDM) allows a VM to directly access a volume on the physical storage subsystem without formatting it with VMFS. RDMs can only be used with block storage (Fiber Channel or iSCSI). RDM can be thought of as providing a symbolic link from a VMFS volume to a raw volume. The mapping makes volumes appear as files in a VMFS volume. The mapping file, not the raw volume, is referenced in the VM configuration. RDM in physical compatibility mode is required for Always On FCI configuration to allow the SCSI-3 persistent reservations or in order to use storage array software, for example for snapshots. RDM and virtual disks can reach the same size of 62 TB and can be increased in size without shutting down the VM.

From a performance perspective, both VMFS and RDM volumes can provide similar transaction throughput[67]. The following chart summarize some performance testing.

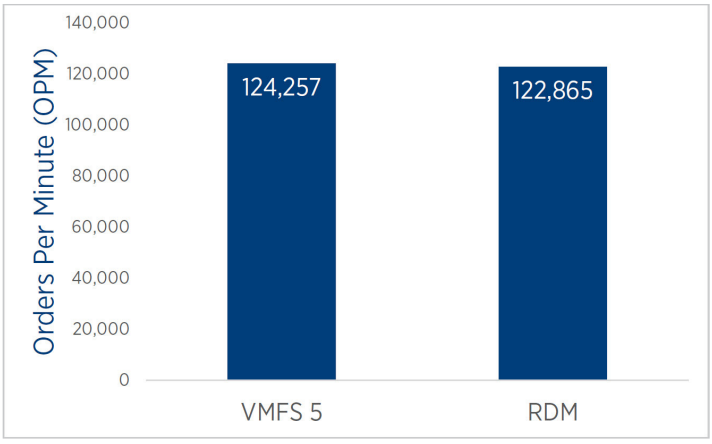
[64] Follow the kb https://kb.vmware.com/s/article/2239

[65] For more details consult: https://storagehub.vmware.com/t/vSphere-storage/best-practices-for-running-vmware-vSphere-on-network-attached-storage/

[66] https://kb.vmware.com/s/article/2147661

[67] More details: https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/performance/sql-server-vsphere65-perf.pdf, p.10-11

Figure 29.
VMFS vs. RDM: DVD Store 3 Performance Comparison[68]



3.8.1.5 VSPHERE VIRTUAL VOLUME[69]

VMware vSphere Virtual Volumes™ (VVols) enables application-specific requirements to drive storage provisioning decisions while leveraging the rich set of capabilities provided by existing storage arrays. Some of the primary benefits delivered by VVols are focused on operational efficiencies and flexible consumption models:

- Flexible consumption at the logical level: VVols virtualizes SAN and NAS devices by abstracting physical hardware resources into logical pools of capacity (represented as virtual datastore in vSphere)
- Finer control at the VM level: VVols defines a new virtual disk container (the virtual volume) that is independent of the underlying physical storage representation (LUN, file system, object, and so on.). It becomes possible to execute storage operations with VM granularity and to provision native array-based data services, such as compression, snapshots, de-duplication, encryption, replication, and so on to individual VMs. This allows admins to provide the correct storage service levels to each individual VM.
- Ability to configure different storage policies for different VMs using Storage Policy-Based Management (SPBM). These policies instantiate themselves on the physical storage system, enabling VM level granularity for performance and other data services.
- Storage Policy-Based Management (SPBM) allows capturing storage service levels requirements (capacity, performance, availability, and so on) in the form of logical templates (policies) to which VMs are associated. SPBM automates VM placement by identifying available datastores that meet policy requirements, and coupled with VVols, it dynamically instantiates necessary data services. Through policy

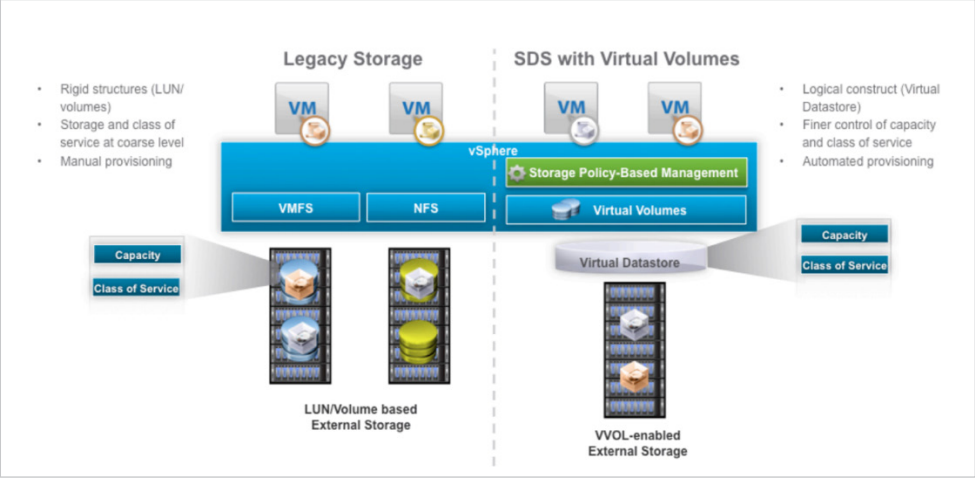
[68] https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/performance/sql-server-vsphere65-perf.pdf, p11

[69] More details can be obtained here: https://docs.vmware.com/en/VMware-vSphere/6.7/com.vmware.vsphere.storage.doc/GUID-EE1BD912-03E7-407D-8FDC-7F596E41A8D3.html and https://www.vmware.com/files/pdf/products/virtualvolumes/VMware-Whats-New-vSphere-Virtual-Volumes.pdf.

enforcement, SPBM also automates service-level monitoring and compliance throughout the lifecycle of the VM.

- Array-based replication starting from VVol 2.0 (vSphere 6.5)
- Support for SCSI-3 persistent reservation starting from vSphere 6.7. A VVol disk could be used instead of a RDM disk to provide a disk resource in a WSFC.

Figure 30.
vSphere Virtual Volumes



VASA support from the storage vendor is required for vSphere to leverage VVol.

VVol capabilities help with many of the challenges that large databases are facing:

- Business-critical virtualized databases need to meet strict SLAs for performance, and storage is usually the slowest component compared to RAM and CPU and even network.
- Database size is growing, while at the same time there is an increasing need to reduce backup windows and the impact on system performance.
- There is a regular need to clone and refresh databases from production to QA and other environments. The size of the modern databases makes it harder to clone and refresh data from production to other environments.
- Databases of different levels of criticality need different storage performance characteristics and capabilities.

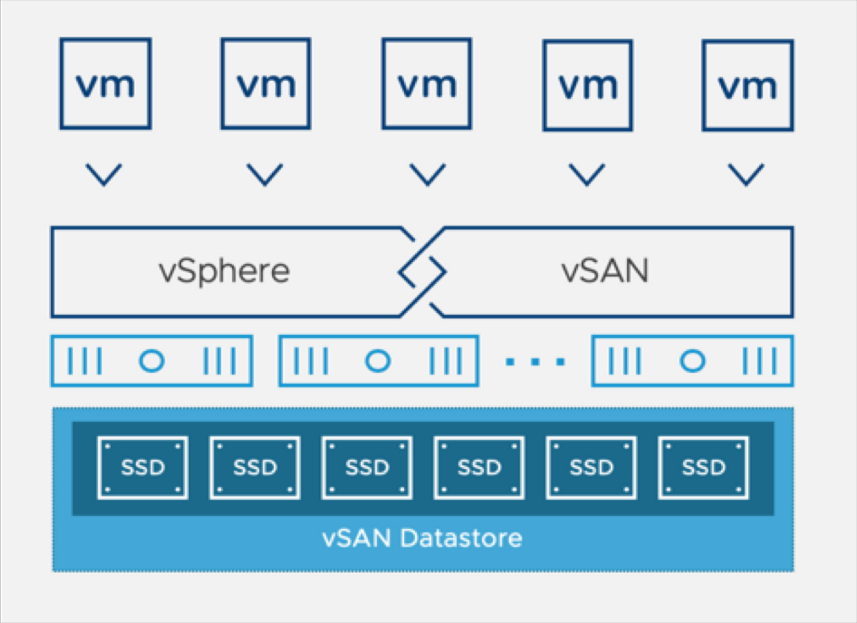
When virtualizing SQL Server on a SAN using VVols as the underlying technology, the best practices and guidelines remain the same as when using a VMFS datastore.

Make sure that the physical storage on which the VM's virtual disks reside can accommodate the requirements of the SQL Server implementation with regard to RAID, I/O, latency, queue depth, and so on, as detailed in the storage best practices in this document.

3.8.2 VMware vSAN[70]

VMware vSAN (vSAN) is the VMware software-defined storage solution for hyperconverged infrastructure (HCI), a software-driven architecture that delivers tightly integrated computing, networking, and highly resilient shared storage from x86 servers. Like vSphere, vSAN provides users the flexibility, and control to choose from a wide range of hardware options; and easily deploy and manage them for a variety of IT workloads and use cases.

Figure 31.
VMware vSAN Architecture



vSAN can be configured as a hybrid or an all-flash storage. In a hybrid disk architecture, vSAN leverages flash-based devices (SAS/SATA SSD or NVMe SSD) for cache, and magnetic disks for capacity. In an all-flash vSAN architecture, vSAN can use flash-based devices for both cache and capacity.

vSAN is a distributed object-based storage that leverages the Storage Policy Based Management (SPBM) vSphere feature to deliver centrally managed infrastructure, application-centric storage services, and storage capabilities at a granular level. Administrators can specify storage attributes, such as capacity, performance, and availability as a policy on a per-object (such as an individual VMDK) level or can also apply to all VMDKs within a virtual machine.

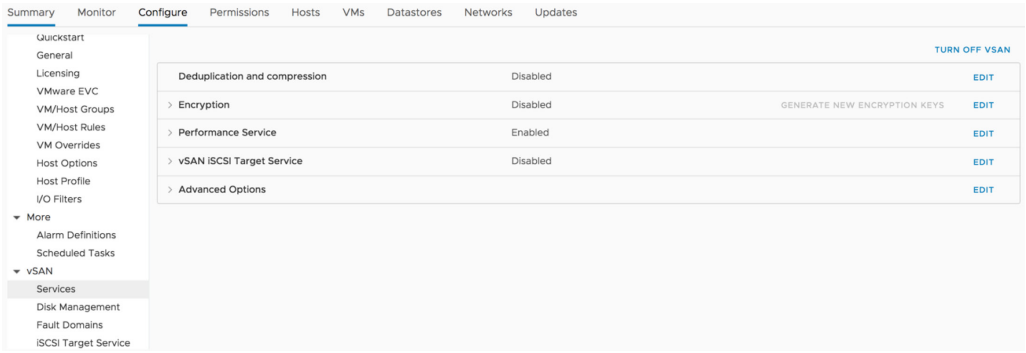
[70] More technical materials on SQL Server on vSAN can be found here: https://storagehub.vmware.com/t/vmware-vsan/microsoft/

3.8.2.1 GENERAL RECOMMENDATIONS

This common set of recommendations is relevant to any SQL Server workloads hosted on vSAN.

- Include an additional host (N+1) to the sizing results, where N = minimum number of hosts needed. For example, the minimum requirement for a vSAN cluster is to have at least three hosts, the recommendation, in this case, would be to have four or more hosts in this vSAN cluster. This additional capacity enables automatic remediation in the event of a drive, disk group, or host failure.
- Ensure that the network infrastructure used for vSAN network traffic is robust enough to sustain the planned workload: A redundant, dedicated, and minimum 10Gbit network for the backend vSAN network is required.
- VMware recommends at least two disk groups per host, or more. Multiple disk groups can improve throughput in many cases and provide better resilience to certain drive failure scenarios.
- vSAN Services on vSAN cluster level:
  - vSAN Performance service provides performance metrics and enables vROps or other monitoring tools to collect vSAN based performance data and efficiently troubleshoot the vSAN deployment. The performance service is enabled by default starting with the vSAN version 6.7. For lower vSAN versions consider enabling this service.
  - Encryption. For high security demands, data at rest encryption (FIPS 140-2 compliant) might be enabled as a cluster-wide setting. Our storage stress validation demonstrated the performance of overall IOPS is very similar between encrypted and non-encrypted runs, but it will add CPU overhead to encrypt and decrypt data. We highly recommend using physical CPUs supporting the AES-NI offloading and verifying that the AES-NI feature is enabled in the server BIOS. For an end-to-end encryption solution, consider using IPSEC for channel encryption. If only selected SQL Server databases should be encrypted, SQL Server native encryption options can be used instead[71].

Figure 32. vSAN Cluster Services



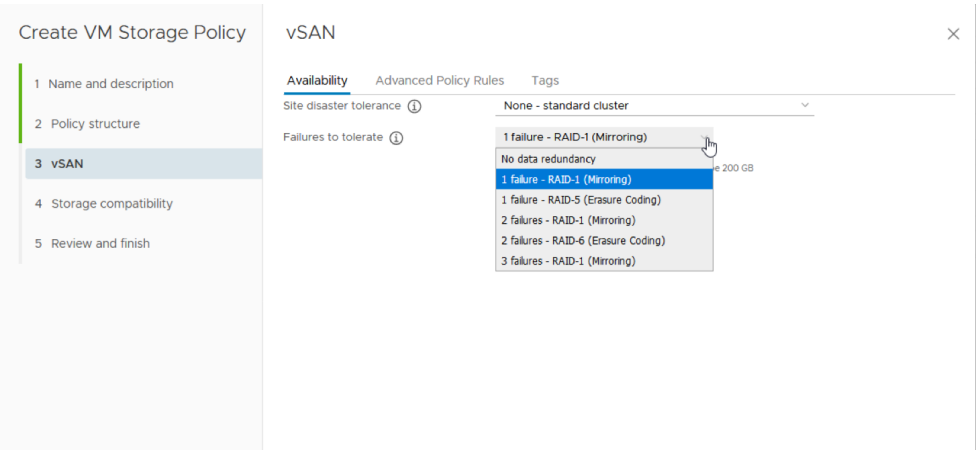
[71] https://docs.microsoft.com/en-us/sql/relational-databases/security/encryption/sql-server-encryption?view=sql-server-2017

- The vSAN iSCSI Target service at the vSAN cluster level provides the ability to use vSAN storage as an iSCSI target to be used within Guest OS. vSAN 6.7 expands the functionality of the vSAN iSCSI Target service to provide the SCSI-3 persistent reservations support for shared disks for windows failover cluster[72] if using of the SQL Server Failover Cluster Instance (FCI) is a requirement.

• Configure and set proper SPBM for VMDKs hosting SQL Server workloads

- Failures to tolerate: Ensure that an option with at least "1 (one) failure to tolerate" is selected. Do not use the option: "*No data redundancy*".

Figure 33.
Configure recommended SPBM[73]



- Number of disk stripes per object: Use default one (1), and consider spreading the data between multiple VMDKs attached to multiple vSCSI controllers. Check Section 3.8.2.3 and 3.8.2.4 for more details.
- IOPS limit per object: vSAN 6.2 and later versions have a QoS feature that sets a policy to limit the number of IOPS that an object can consume. This feature may be used to limit IOPS allocation for backup/restore operations to prevent saturating production workload. Use with caution as this will influence backup/restore time and the RTO/RPO as well. Do not use this setting for any performance demanding VMDKs hosting SQL Server data or log files.

Additional setting adjustments may help to gain more benefits to your workload. Use the section describing "Tier 1" workload for SQL Server workloads where performance is the priority, while the section "Tier 2" might fit better to the workloads where capacity and cost savings are more important.

[72] https://blogs.vmware.com/virtualblocks/2018/04/17/whats-new-vmware-vsan-6-7/ and https://storagehub.vmware.com/t/vmware-vsan/sql-server-fci-and-file-server-on-vmware-vsan-6-7-using-iscsi-service/

[73] All screenshots are taken using vSphere Client (HTML5) and vSphere version 6.7.

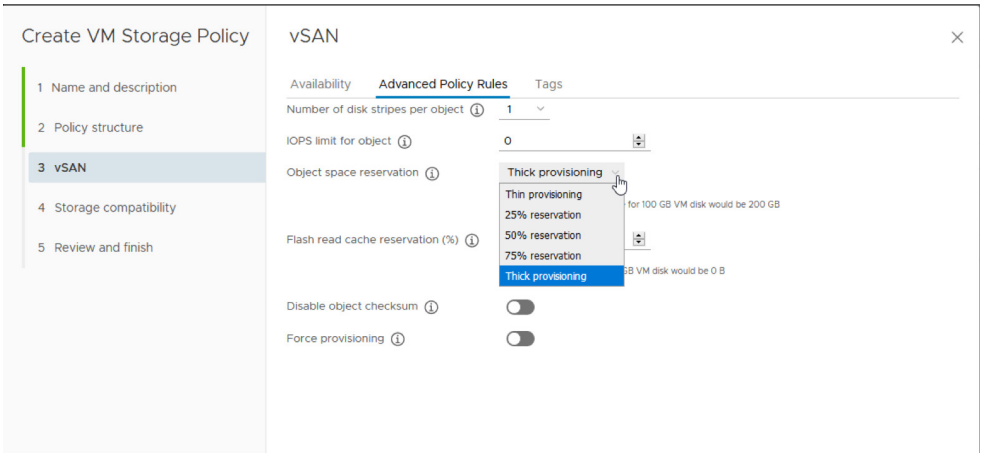
3.8.2.2 TIER 1 HIGH PERFORMANCE ONLINE TRANSACTION PROCESSING (OLTP) WORKLOAD:

Typical SQL Server I/O activities for OLTP workloads include queries with many seek operations, checkpoint activity that flushes dirty pages to disk periodically, and transaction log writes. The in-flight IO of data is fairly small in size, and typically between 8K and 64K. TPC-E benchmarks are commonly used to reproduce the OLTP-like workloads.

For such workloads where performance, and availability are highly important, consider following:

- All-flash vSAN deployments
- Consider using at least SAS SSD devices. A SAS SSD device will perform better than a SATA SSD device in most cases, and has a bigger queue depth. Newer devices such as NVMe, will yield better performance.
- Disable deduplication and compression (this is disabled by default in vSAN). You may implement compression on a per table basis at the database level (Enterprise edition of SQL Server is required) if needed.
- Set object space reservation – "Thick provisioning" for all VMDKs hosting SQL Server workloads (data, tempdb, and log files). The capacity will be reserved up front from the vSAN datastore to avoid out of space issue. Bear in mind that this setting would not create a thick (eager zeroed) vmdk, a vmdk will still be thin provisioned.

Figure 34. Configure Object Space Reservation in SPBM



- Do not use the *IOPS limit for object* option.
- Use separate VMDKs for SQL Server workloads connected via PVSCSI adapter. Check the section 3.8.3.3 for more details.
- RAID 1 mirroring and at least 1 failure to tolerate (FTT) for all VMDKs hosting SQL Server workload is the recommended failure tolerance method for this type of workload.
- Additional availability requirements might dictate the need to increase FTT and/or implement SQL Server high availability solutions like Always On Availability Groups. If FTT will be increased, make adjustments to the number of disks required to satisfy

not only capacity, but performance requirements. If Availability Groups are enabled, you can have application level high availability as well as the underlying storage high availability, however more storage space, and a well-designed virtual machine network is required.

- If a multi-site availability is required, the vSAN Stretched Cluster configuration may be used to increase the data availability across data centres.
- It is also important to think about the network switches to be used during a vSAN deployment. While using configurations such as all-NVMe vSAN clusters, the physical disk controllers are no longer part of the configuration, and allows for such NVMe devices to process a lot of IO in a small amount of time. Enterprise grade switches (10Gbit or more) capable of having large buffer sizes (non-shared) is ideal in order to allow for more IO flow to the vSAN cluster.

3.8.2.3 TIER 2 ONLINE TRANSACTION PROCESSING (OLTP) WORKLOAD:

Tier-2 workloads associate with general-purpose SQL Server deployments which do not have performance dominant requirements but need to be cost effective. All considerations posted below should not be applied to any infrastructure where performance is the primary goal.

- Hybrid vSAN might satisfy the requirements for such workloads if the following is taken into account:
  - The use of multiple disk groups is strongly recommended to increase the system throughput.
  - It is important to have enough space in the caching tier. The general recommendation of the SSD as the caching tier for each host is to be at least 10 percent of the total storage capacity. However, the recommended SSD size should be at least two times that of the working data set.
  - Select the appropriate SSD class to support planned IOPS. A SAS SSD device will perform better than a SATA SSD device in most cases, and has a bigger queue depth
- All-flash vSAN might yield satisfactory performance with better space savings that hybrid vSAN, at a lower costs with the following settings:
  - Enable deduplication and compression if the workload is low write intensive and cost/space savings outweigh performance requirements.
  - For the SQL Server data disks, using RAID 5/6 erasure coding to reduce space usage might be a choice, if space/cost savings are desired. The virtual disks for transaction logs should still be placed on a VMDK configured with RAID 1 policy. It is important to note that this recommendation applies to workloads that do not require high performance, and space savings are desired.
- Set object space reservation – "Thick provisioning" for VMDKs hosting SQL Server log files. The capacity will be reserved up front from the vSAN datastore to avoid out of space issue.

3.8.2.4 DATA WAREHOUSE AND/OR REPORTING WORKLOADS

Data warehouse applications issue scan-intensive operations that access large portions of the data at a time, as well as commonly performed bulk loading operations. These operations result in larger I/O sizes than OLTP workloads do, and require a storage subsystem that can provide the required throughput. This makes the throughput, or megabytes per second (MB/s), the critical metric. Common data

warehouse type applications include decision support applications. TPC-H benchmarks are commonly used to reproduce such workload.

For such workloads consider the following:

- In all-flash configurations, using NVMe SSD at the capacity tier may benefit the workload performance due to the read dominant query operations.
- In all-flash configurations where saving space is the priority, usage of RAID5/6 for data VMDKs may be a choice due to the predominantly read pattern on such workload type. Perform thoughtful testing as RAID1 configuration will generally provide better performance.
- Set object space reservation – "Thick provisioning" for all VMDKs hosting SQL Server workloads. The capacity will be reserved up front from the vSAN datastore to avoid out of space issue.
- Do not use the IOPS limit for object option. As throughout is the critical metrics enough bandwidth should be available for such workloads.

3.8.3 Storage Best Practices

Many of SQL Server performance issues can be traced back to the improper storage configuration. SQL Server workloads are generally I/O heavy, and a misconfigured storage subsystem can increase I/O latency and significantly degrade performance of SQL Server.

3.8.3.1 PARTITION ALIGNMENT

Aligning file system partitions is a well-known storage best practice for database workloads. Partition alignment on both physical machines and VMFS partitions prevents performance I/O degradation caused by unaligned I/O. vSphere 5.0 and later automatically aligns VMFS5 partitions along a 1 MB boundary and most modern OSs do the same. Just in rare cases some additional manual efforts are required[74].

It is considered a best practice to:

- Create VMFS partitions using the VMware vCenter™ web client. They are aligned by default.
- Starting with Windows Server 2008, a disk is automatically aligned to a 1 MB boundary. If necessary, align the data disk for heavy I/O workloads using the *diskpart* command[75].
- Consult with the storage vendor for alignment recommendations on their hardware.

3.8.3.2 VMDK FILE LAYOUT

When running on VMFS, virtual machine disk files can be deployed in three different formats: thin, zeroed thick, and eagerzeroedthick. Thin provisioned disks enable 100

[74] Special consideration should be taken when working with the VMFS3 (and VMFS5 upgraded from VMFS3) formatted datastores. It's recommended to reformat such datastores with VMFS5 or 6.

[75] More details: https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008/dd758814(v=sql.100)

percent storage on demand, where disk space is allocated and zeroed at the time disk is written. Zeroedthick disk storage is pre-allocated, but blocks are zeroed by the hypervisor the first time the disk is written. Eagerzeroedthick disk is pre-allocated and zeroed when the disk is initialized during provision time. There is no additional cost for zeroing the disk at run time.

Both thin and thick options employ a lazy zeroing technique, which makes creation of the disk file faster with the cost of performance overhead during first write of the disk. Depending on the SQL Server configuration and the type of workloads, the performance difference could be significant.

When the underlying storage system is enabled by vSphere Storage APIs - Array Integration (VAAI) with "Zeroing File Blocks" primitive enabled, there is no performance difference between using thick, eager zeroed thick, or thin, because this feature takes care of the zeroing operations on the storage hardware level. Also for thin provisioned disks, VAAI with the primitive "Atomic Test & Set" (ATS) enabled, improves performance on new block write by offloading file locking capabilities as well. Now, most storage systems support vSphere Storage APIs - Array Integration primitives[76]. All-flash arrays utilize a 100 percent thin provisioning mechanism to be able to have storage on demand.

3.8.3.3 OPTIMIZE WITH DEVICE SEPARATION

SQL Server files have different disk access patterns as shown in the following table:

Table 5. Typical SQL Server Disk Access Patterns

Operation	Random/Sequential	Read/Write	Size Range
OLTP – Transaction Log	Sequential	Write	sector-aligned, up to 64K
OLTP – Data	Random	Read/Write	8K
Bulk Insert	Sequential	Write	Any multiple of 8K up to 256K
Read Ahead (DSS, Index Scans)	Sequential	Read	Any multiple of 8K up to 512K
Backup	Sequential	Read	1MB

When deploying a Tier 1 mission-critical SQL Server, placing SQL Server binary, data, transaction log, and TempDB files on separate storage devices allows for maximum flexibility, and can improve performance. SQL Server accesses data and transaction log files with very different I/O patterns. While data file access is mostly random, transaction log file access is sequential only. Traditional storage built with spinning disk media requires repositioning of the disk head for random read and write access.

[76] Consult your storage array vendor for the recommended firmware version for the full VAAI support.

Therefore, sequential data is much more efficient than random data access. Separating files that have different random access patterns, compared with sequential access patterns, helps to minimize disk head movements, optimizing storage performance. vSphere 6.7 helps to follow this practice increasing the supported number of disks per VM to 256 and increasing the number of path per ESXi host to 4096.

Remember that Windows Server and Linux handle disks differently. Windows Server deploys with SQL Server support NTFS or ReFS (Windows Server 2012 R2+ and SQL Server 2014+) for the file system. Linux-based SQL Server implementations support EXT4 or XFS only. Linux does not have the concept of drive letters, so disks mounted using PVSCSI will be shown as a device which will need to be mounted to a folder such as /var/opt/mssql/mydata with the appropriate permissions. SQL Server's data and log files behave the same otherwise, but how things are configured underneath of SQL Server itself is slightly different due to the OS differences.

The following guidelines can help to achieve best performance:

- Place SQL Server data (system and user), transaction log, and backup files into separate VMDKs (if not using RDMs) and possibly on separate datastores.
- The SQL Server binaries should be installed in the OS VMDK. SQL Server, even if another drive is selected for binary installation, will still install things on the OS drive so there is no real point in installing elsewhere. Separating SQL Server installation files from data and transaction logs also provides better flexibility for backup, management, and troubleshooting.
- For the most critical databases where performance requirements supersede all other requirements, maintain 1:1 mapping between VMDKs and LUNs. This will provide better workload isolation and will prevent any chance for storage contention on the datastore level. Of course, the underlying physical disk configuration must accommodate the I/O and latency requirements as well. When manageability is a concern, group VMDKs and SQL Server files with similar I/O characteristics on common LUNs while making sure that the underling physical device can accommodate the aggregated I/O requirements of all the VMDKs.
- For underlying storage, where applicable, RAID 10 can provide the best performance and availability for user data, transaction log files, and TempDB.

For lower-tier SQL Server workloads, consider the following:

- Deploying multiple, lower-tier SQL Server systems on VMFS facilitates easier management and administration of template cloning, snapshots, and storage consolidation.
- Manage performance of VMFS. The aggregate IOPS demands of all VMs on the VMFS should not exceed the IOPS capability the underlying physical disks.
- Use vSphere Storage DRS™ (SDRS) for automatic load balancing between datastores to provide space and avoid I/O bottlenecks as per pre-defined rules. Consider scheduling invocation of the SDRS for off-peak hours to avoid performance penalties while moving a VM.[77]

[77] More details: https://docs.vmware.com/en/VMware-vSphere/6.7/com.vmware.vsphere.resmgmt.doc/GUID-DBCAA3F6-D54A-41DA-ACFC-57CCB7E8DF2A.html

3.8.3.4 USING STORAGE CONTROLLER

Utilize the VMware Paravirtualized SCSI (PVSCSI) Controller as the virtual SCSI Controller for data and log VMDKs. The PVSCSI Controller is the optimal SCSI controller for an I/O-intensive application on vSphere allowing not only a higher I/O rate but also lowering CPU consumption compared with the LSI Logic SAS. In addition, the PVSCSI adapters provide higher queue depth, increasing I/O bandwidth for the virtualized workload. See section 4.1.3 for more details.

Use multiple PVSCSI adapters. VMware supports up to four (4) adapters per a VM and as many as necessary, up to this limit, should be leveraged. Placing OS, data, and transaction logs onto a separate vSCSI adapter optimizes I/O by distributing load across multiple target devices and allowing for more queues on the OS level. Consider distributing disks between controllers. vSphere 6.7 supports up to 64 disks per controller[78].

In vSphere 6.5 the new type of virtual controller was introduced—vNVMe[79]. It has undergone significant performance enhancement with the vSphere 6.7 release. The vNVMe controller might bring performance improvement and reduce I/O processing overhead especially in combination with low latency SSD drives on all-flash storage arrays or combined with the vPMemDisks. Consider testing the configuration using a copy of your production database to check if this change will be beneficial. vSphere 6.7 and virtual hardware 14 are strongly recommended for any implementation of vNVMe controller.

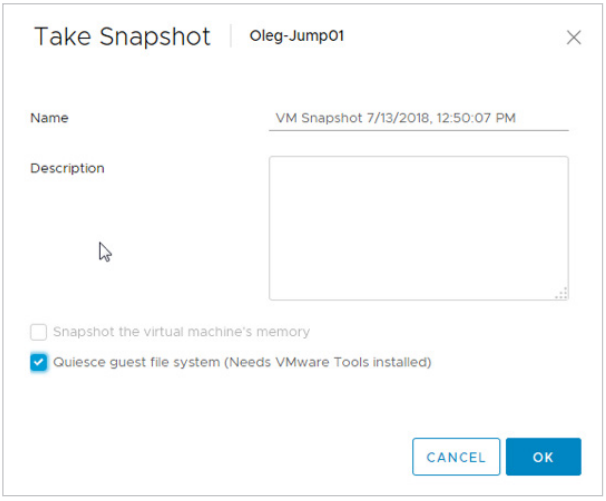
[78] Consult https://configmax.vmware.com/guest for more details

[79] https://docs.vmware.com/en/VMware-vSphere/6.5/com.vmware.vsphere.vm_admin.doc/GUID-63E09187-0E75-405B-97C7-B48DA1B1734F.html

3.8.2.5 USING SNAPSHOTS

A VM snapshot preserves the state and data of a VM at a specific point in time[80]. When a snapshot is created, it will store the power state of the virtual machine and the state of all devices, including virtual disks. To track changes in virtual disks a special "delta" file is used that contains a continuous record of the block level changes to a virtual disk[81]. Snapshots are widely used by backup software or by infrastructure administrators and DBAs to preserve the state of a virtual machine before implementing changes (like upgrade of the SQL Server, installing patches etc.).

Figure 35.
Take Snapshot Options



In general, snapshot usage should be limited and possibly avoided on VMs running production SQL Server workloads. For backup purposes, consider to use in-guest, agent-based backup. If snapshots should be taken, the following best practices should be considered:

- Offline snapshot (a VM is powered off when a snapshot is taken) can be used without special considerations.
- If an online snapshot (VM is powered on and Guest OS is running) needs to be taken:
  - Consider not using "Snapshot the virtual machine's memory" option as this may stun a VM[82]. Rely on SQL Server mechanisms to prevent data loss by losing in-memory data.
  - Use "Quiesce guest file system" option to ensure that a disk consistent snapshot will be taken. Special notes:
    - » Consider not taking an online snapshot if VMware Tools are not installed or not functional as this may lead to the inconsistent disk state.

[80] https://kb.vmware.com/s/article/1015180

[81] https://blogs.vmware.com/kb/2010/06/vmware-snapshots.html

[82] https://kb.vmware.com/kb/1013163

  - » Consider checking the status of the Volume Shadow Copy Service (VSS) on the Windows OS before taking a snapshot.
  - » If using a third-party utilities for snapshots, make sure there are VSS compliant.
- Be aware that on a highly loaded instance of SQL Server producing high number of disk I/O, snapshot operations (creation of an online snapshot, online removal of a snapshot) may take a long time and can potentially cause performance issues[83]. Consider planning the snapshots operations for non-peak hours, use offline creation/removal of snapshots, or use VVol technology with the storage array level snapshot integrations.

- Do not run a VM hosting a SQL Server instance on a snapshot for more than 72 hours[84].
- Snapshot is not a replacement for a backup. The delta disk file contains only references to the changes and not the changes itself.
- Consider using VMFS6 and SEsparce snapshot for performance improvements.

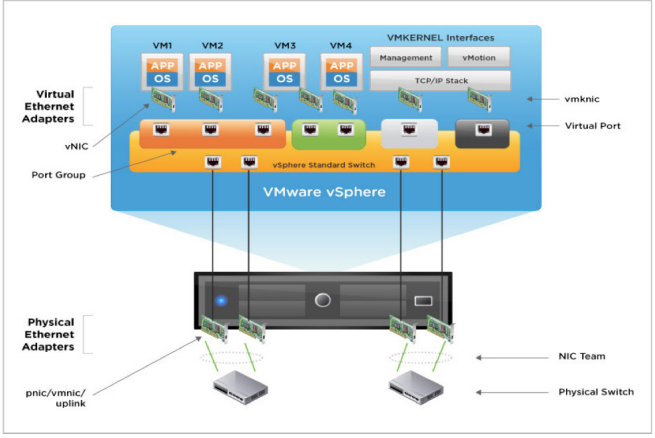
3.9 Virtual Machine Network Configuration

Networking in the virtual world follows the same concepts as in the physical world, but these concepts are applied in software instead of through physical cables and switches. Many of the best practices that apply in the physical world continue to apply in the virtual world, but there are additional considerations for traffic segmentation, availability, and for making sure that the throughput required by services hosted on a single server can be distributed.

3.9.1 Virtual Network Concepts

The following figure provides a visual overview of the components that make up the virtual network.

Figure 36
Virtual Networking Concepts



[83] http://kb.vmware.com/kb/1002836, https://cormachogan.com/2015/04/28/when-and-why-do-westun-a-virtual-machine/

[84] Depending on a workload and an environment this recommendation may vary, but in general should not exceed 72 hours, sometimes much shorter time is preferred.

As shown in the figure, the following components make up the virtual network:

- Physical switch: vSphere host-facing edge of the physical local area network.
- NIC team: Group of NICs connected to the same physical/logical networks to provide redundancy and aggregated bandwidth.
- Physical network interface (pnic/vmnic/uplink): Provides connectivity between the ESXi host and the local area network.
- vSphere switch (standard and distributed): The virtual switch is created in software and provides connectivity between VMs. Virtual switches must uplink to a physical NIC (also known as vmnic) to provide VMs with connectivity to the LAN. Otherwise, virtual machine traffic is contained within the VM.
- Port group: Used to create a logical boundary within a virtual switch. This boundary can provide VLAN segmentation when 802.1q trunking is passed from the physical switch, or it can create a boundary for policy settings.
- Virtual NIC (vNIC): Provides connectivity between the VM and the virtual switch.
- VMkernel (vmknic): Interface for hypervisor functions, such as connectivity for NFS, iSCSI, vSphere vMotion, and vSphere Fault Tolerance logging.
- Virtual port: Provides connectivity between a vmknic and a virtual switch.

3.9.2 Virtual Networking Best Practices

Some SQL Server workloads are more sensitive to network latency than others. To configure the network for your SQL Server-based VM, start with a thorough understanding of your workload network requirements. Monitoring the following performance metrics on the existing workload for a representative period using Windows Perfmon or VMware Capacity Planner™ or preferably with vROPs can easily help determine the requirements for an SQL Server VM.

The following guidelines generally apply to provisioning the network for an SQL Server VM:

- The choice between standard and distributed switches should be made outside of the SQL Server design. Standard switches provide a straightforward configuration on a per-host level. For reduced management overhead and increased functionality, consider using the distributed virtual switch. Both virtual switch types provide the functionality needed by SQL Server.
- Traffic types should be separated to keep like traffic contained to designated networks. vSphere can use separate interfaces for management, vSphere vMotion, and network-based storage traffic. Additional interfaces can be used for VM traffic. Within VMs, different interfaces can be used to keep certain traffic separated. Use 802.1q VLAN tagging and virtual switch port groups to logically separate traffic. Use separate physical interfaces and dedicated port groups or virtual switches to physically separate traffic.
- If using iSCSI, the network adapters should be dedicated to either network communication or iSCSI, but not both.
- VMware highly recommends considering enabling jumbo frames on the virtual switches where you have enabled vSphere vMotion traffic or iSCSI traffic. Make sure that jumbo frames are also enabled on your physical network infrastructure end to end before making this configuration on the virtual switches. Substantial performance penalties can occur if any of the intermediary switch ports are not configured for jumbo frames properly.

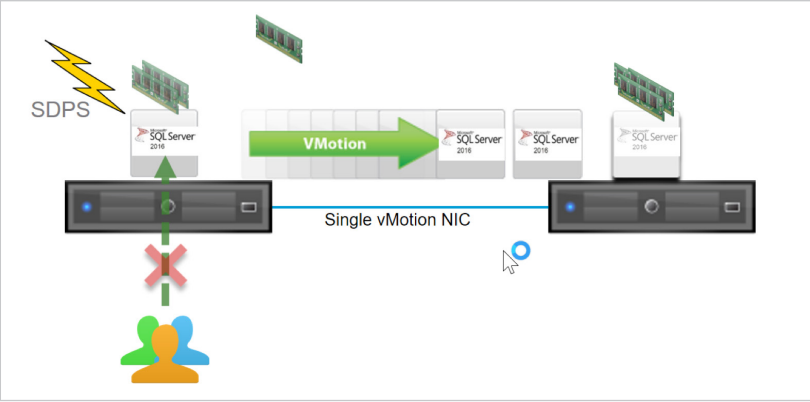
- Use the VMXNET3 paravirtualized NIC. VMXNET 3 is the latest generation of paravirtualized NICs designed for performance. It offers several advanced features including multi-queue support, Receive Side Scaling, IPv4/IPv6 offloads, and MSI/MSI-X interrupt delivery.
- Follow the guidelines on guest OS networking considerations and hardware networking considerations[85].

3.9.3 Using multi-NIC vMotion for High Memory Workloads

vSphere 5.0 and later have a feature called Stun during Page Send (SDPS)[86] that helps vMotion operations for large memory intensive VMs. When a VM is being moved with vMotion, its memory is copied from the source ESXi host to the target ESXi host iteratively. The first iteration copies all the memory, subsequent iterations copy only the memory pages that were modified during the previous iteration. The final phase is the switchover, where the VM is momentarily quiesced on the source vSphere host and the last set of memory changes are copied to the target ESXi host, and the VM is resumed on the target ESXi host.

In cases where a vMotion operation is initiated for a large memory VM and its large memory size is very intensively utilized, pages might be "dirtied" faster than they are replicated to the target ESXi host. An example of such a workload is a 64GB memory optimized OLTP SQL Server that is heavily utilized. In that case, SDPS intentionally slows down the VM's vCPUs to allow the vMotion operation to complete. While this is beneficial to guarantee the vMotion operation to complete, the performance degradation during the vMotion operation might not be an acceptable risk for some workloads. To get around this and reduce the risk of SDPS activating, you can utilize multi-NIC vMotion. With multi-NIC vMotion, every vMotion operation utilizes multiple port links, even a single VM vMotion operation. This speeds up vMotion operation and reduces the risk for SDPS on large, memory intensive VMs[87].

Figure 37
vMotion of a Large Intensive VM with SDPS Activated

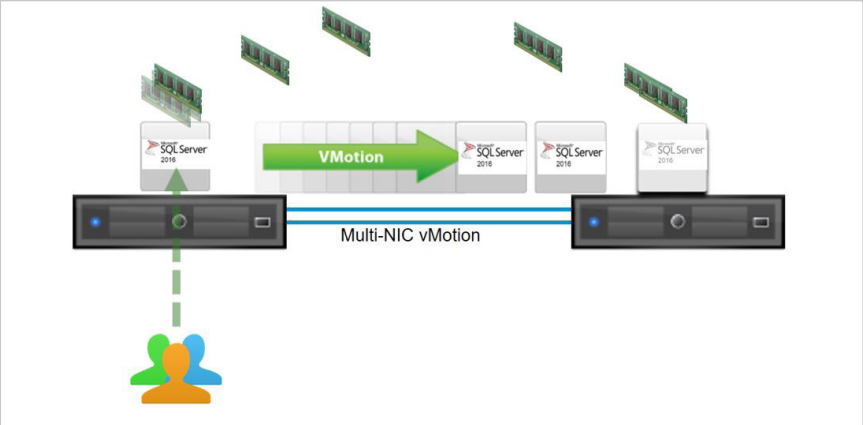


[85] *Performance Best Practices for vSphere 6.5* guide https://www.vmware.com/techpapers/2017/Perf_Best_Practices_vSphere65.html

[86] For more information about vMotion architecture and SDPS, see the *vSphere vMotion Architecture, Performance and Best Practices* https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/vmware-vSphere51-vmotion-performance-white-paper.pdf

[87] For more information on how to set multi-NIC vMotion refer to the following kb article: https://kb.vmware.com/kb/2007467

Figure 38
Utilizing Multi-NIC vMotion to Speed Up vMotion Operation



3.9.4 Enable Jumbo Frames for vSphere vMotion Interfaces

Standard Ethernet frames are limited to a length of approximately 1500 bytes. Jumbo frames can contain a payload of up to 9000 bytes. This feature enables use of large frames for all VMkernel traffic, including vSphere vMotion. Using jumbo frames reduces the processing overhead to provide the best possible performance by reducing the number of frames that must be generated and transmitted by the system. VMware tested vSphere vMotion migration of critical applications, such as SQL Server, with and without jumbo frames enabled. Results showed that with jumbo frames enabled for all VMkernel ports and on the vSphere Distributed Switch, vSphere vMotion migrations completed successfully. During these migrations, no database failovers occurred, and there was no need to modify the cluster heartbeat setting.

The use of jumbo frames requires that all network hops between the vSphere hosts support the larger frame size. This includes the systems and all network equipment in between. Switches that do not support (or are not configured to accept) large frames will drop them. Routers and Layer 3 switches might fragment the large frames into smaller frames that must then be reassembled, and this can cause both performance degradation and a pronounced incidence of unintended database failovers during a vSphere vMotion operation.

Do not enable jumbo frames within a vSphere infrastructure unless the underlying physical network devices are configured to support this setting.

3.10 vSphere Security Features

vSphere platform has a reach set of security features which may help a DBA administrator to mitigate SQL Server security risks in a virtualized environment.

3.10.1 Virtual Machine Encryption[88]

VM encryption enables encryption of the VM's I/Os before they are stored in the virtual disk file. Because VMs save their data in files, one of the concerns starting from the earliest days of virtualization, is that data can be accessed by an unauthorized entity, or stolen by taking the VM's disk files from the storage. VM encryption is controlled on a per VM basis, and is implemented in the virtual vSCSI layer using an IOFilter API. This framework is implemented entirely in user space, which allows the I/Os to be isolated cleanly from the core architecture of the hypervisor.

VM encryption does not impose any specific hardware requirements, and using a processor that supports the AES-NI instruction set speeds up the encryption/decryption operation.

Any encryption feature consumes CPU cycles, and any I/O filtering mechanism consumes at least minimal I/O latency overhead.

The impact of such overheads largely depends on two aspects:

- The efficiency of implementation of the feature/algorithm.
- The capability of the underlying storage.

If the storage is slow (such as in a locally attached spinning drive), the overhead caused by I/O filtering is minimal, and has little impact on the overall I/O latency and throughput. However, if the underlying storage is very high performance, any overhead added by the filtering layers can have a non-trivial impact on I/O latency and throughput. This impact can be minimized by using processors that support the AES-NI instruction set.

3.10.2 vSphere 6.7. New Security Features[89]

vSphere 6.7 provides a number of enhancement that help to lower a security risks for a VMs hosting SQL Server. This features includes:

- Support for a virtual Trusted Platform Module (vTPM) for the virtual machine.
- Support for Microsoft Virtualization Based Security[90].
- Enhancement for the ESXi "secure boot."
- Virtual machine UEFI Secure Boot.
- FIPS 140-2 Validated Cryptographic Modules turned on by default for all operations.

NOTE: vHardware version 14 must be used to allow these features to be enabled.

3.11 Maintaining a Virtual Machine

During the operational lifecycle of a VM hosting SQL Server it's expected that changes will be required. A VM might need to be moved to a different physical datacenter or

[88] For the latest performance study of VM encryption, see the following paper: http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/vm-encryption-vsphere65-perf.pdf.

[89] Consult this document for a full description: https://docs.vmware.com/en/VMware-vSphere/6.7/vsphere-esxi-vcenter-server-67-security-guide.pdf

[90] More details https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/oem-vbs and here https://blogs.msdn.microsoft.com/sqlsecurity/2017/10/05/enabling-confidential-computing-with-always-encrypted-using-enclaves-early-access-preview/

virtual cluster, where physical hosts are different and different version of the vSphere is installed, or the vSphere platform will be updated to the latest version. In order to maintain best performance and be able to use new features of the physical hardware or vSphere platform VMware strongly recommend to:

- Check and upgrade VMware Tools
- Check and upgrade the compatibility (aka "virtual hardware")

3.11.1 Upgrade VMware Tools[91]

VMware Tools is a set of services, drivers and modules that enable several features for better management of, and seamless user interactions with, guest's OSs. VMware Tools can be compared with the drivers pack required for the physical hardware, but in virtualized environments.

Upgrading to the latest version will provide the latest enhancements and bug and security fixes for virtual hardware devices like VMXNET3 network adapter or PVSCSI virtual controller. For example, VMware Tools version 10.2.5 enables RSS Scaling be default for any new installation.

VMware Tools can be upgraded in a many ways[92], but for a VM hosting a Tier1 mission critical SQL Server, a manual update through the vSphere Web Client is recommended accomplished with comprehensive testing process. It should be double underscored, that the VMware Tools upgrade is essentially a driver upgrade process and it will influence the core OS, hence should be done with care and preliminary testing in a non-production environment.

3.11.2 Upgrade the Virtual Machine Compatibility[93]

A virtual machine's compatibility determines the virtual hardware available to the VM, which corresponds to the physical hardware available on the host machine. Virtual hardware includes BIOS and EFI, available virtual PCI slots, maximum number of CPUs, maximum memory configuration, and other characteristics. You can upgrade the compatibility level to make additional hardware available to the VM[94]. For example, to be able to assign more than 1TB of memory, virtual machine compatibility should be at least hardware version 12.

It's important to mention, that the hardware version also defines the maximum CPU instruction set exposed to a VM: for example, a VM with the hardware level 8 will not be able to use the instruction set of the Intel Skylake CPU.

VMware recommends upgrading the virtual machine compatibility when new physical hardware is introduced to the environment. Virtual machine compatibility upgrade should be planned and taken with care.

[91] More details: https://docs.vmware.com/en/VMware-Tools/

[92] https://blogs.vmware.com/vsphere/2016/03/six-methods-for-keeping-vm-tools-up-to-date.html

[93] https://docs.vmware.com/en/VMware-vSphere/6.7/com.vmware.vsphere.vm_admin.doc/GUID-64D4B1C9-CD5D-4C68-8B50-585F6A87EBA0.html

[94] https://docs.vmware.com/en/VMware-vSphere/6.7/com.vmware.vsphere.vm_admin.doc/GUID-789C3913-1053-4850-A0F0-E29C3D32B6DA.html

The following procedures are recommended[95]:

- Take a backup of the SQL Server databases and OS.
- Upgrade VMware Tools.
- Validate that no misconfigured/inaccessible devices (like CD-ROM, Floppy) are present.
- Use vSphere Web Client to upgrade virtual hardware to the desired version

NOTE: Upgrading a Virtual Machine to the latest hardware version is the physical equivalent of swapping the drive out of one system and placing it into a new one. Its success will depend on the resiliency of the guest OS in the face of hardware changes. VMware does not recommend upgrading virtual hardware version if you do not need the new features exposed by the new version.

NOTE: C# Client has no support for the virtual machine compatibility level 9 and above. All new features introduced after this version are exposed to the vSphere Web Client only.

[95] https://kb.vmware.com/s/article/1010675

4. SQL Server and In-Guest Best Practices

In addition to the previously mentioned vSphere best practices for SQL Server, there are configurations that can be made on the SQL Server and Windows Server/Linux side to optimize its performance within the virtual machine. Many of these settings are described by Microsoft, and generally none of our recommendations contradict Microsoft recommendations, but the following are important to mention for a vSphere virtualized environment.

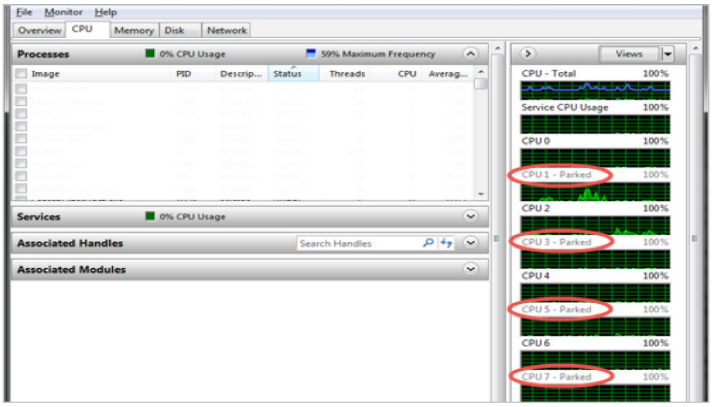
4.1 Windows Server Configuration[96]

The following list details the configuration optimization that can be done on the Windows OS.

4.1.1 Power Policy[97]

The default power policy option in Windows Server 2012 and above is "Balanced." This configuration allows Windows Server OS to save power consumption by periodically throttling power to the CPU and turning off devices such as the network cards in the guest when Windows Server determines that they are idle or unused. This capability is inefficient for critical SQL Server workloads due to the latency and disruption introduced by the act of powering-off and powering-on CPUs and devices. Allowing Windows Server to throttle CPUs can result in what Microsoft describes as core parking and should be avoided. For more information, see *Server Hardware Power Considerations* at https://msdn.microsoft.com/en-us/library/dn567635.aspx.

Figure 39.
Windows Server CPU Core Parking

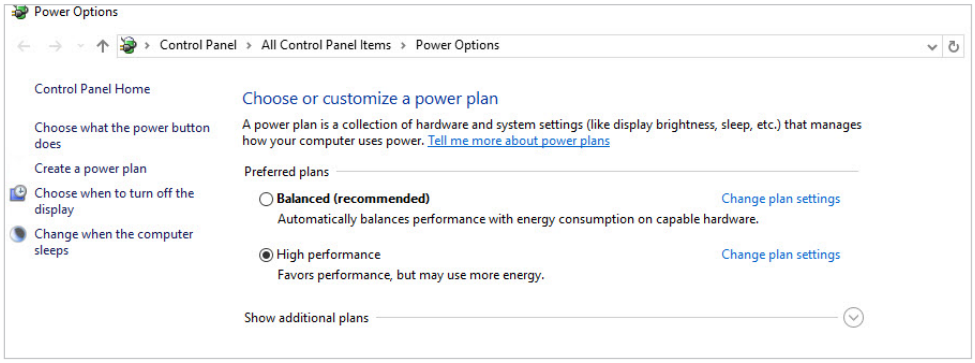


Microsoft recommends the high-performance power management plan for applications requiring stability and performance. VMware supports this recommendation and encourages customers to incorporate it into their SQL Server tuning and administration practice for virtualized deployment.

[96] Consult the document for more details: https://blogs.msdn.microsoft.com/docast/2018/02/01/operating-system-best-practice-configurations-for-sql-server/

[97] https://support.microsoft.com/en-au/help/2207548/slow-performance-on-windows-server-whenusing-the-balanced-power-plan

Figure 40.
Recommended Windows OS Power Plan

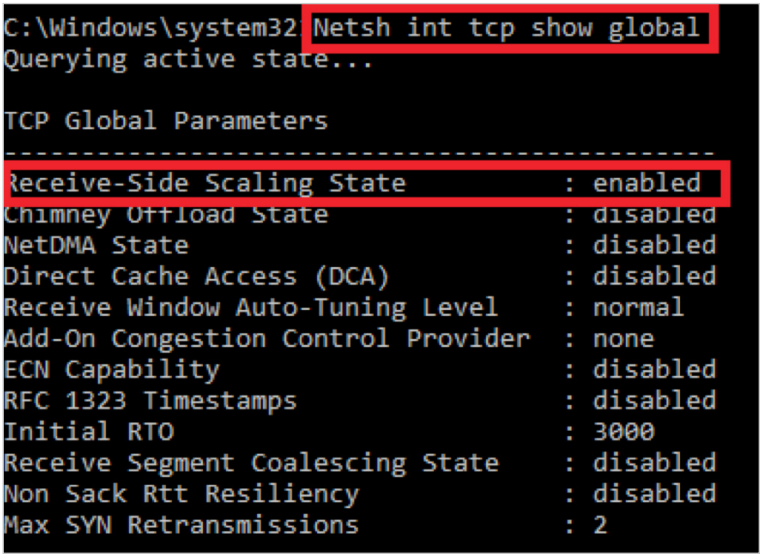


4.1.2 Enable Receive Side Scaling (RSS)[98]

Enable RSS (Receive Side Scaling): This network driver configuration within Windows Server enables distribution of the kernel-mode network processing load across multiple CPUs. Enabling RSS is done in the following two places:

- Enable RSS in the windows kernel by running the n*etsh interface tcp set global rss=enabled* command in elevated command prompt. You can verify that RSS is enabled by running the *Netsh int tcp show global* command. The following figure provides an example of this:

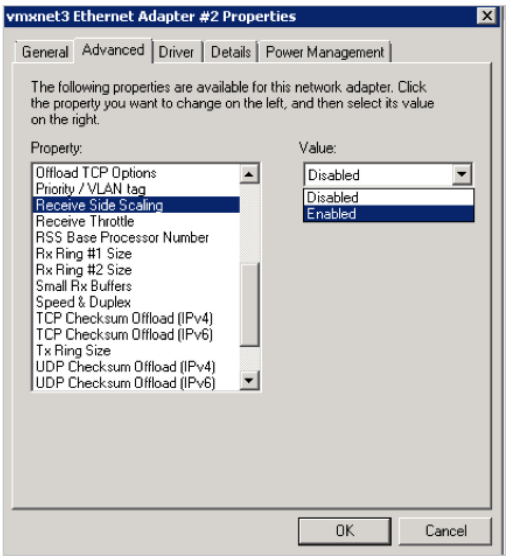
Figure 41.
Enable RSS in Windows OS



[98] https://support.microsoft.com/en-au/help/2207548/slow-performance-on-windows-server-when-using-the-balanced-power-plan

- Enable RSS on the VMXNET network adapter driver[99]. In Windows in **Network adapters**, right-click the VMXNET network adapter and click **Properties**. On the **Advanced** tab, enable the setting Receive-side scaling.

Figure 42.
Enable RSS in VMware Tools



For more information about RSS, see https://technet.microsoft.com/en-us/library/hh997036.aspx. To enable RSS, see https://technet.microsoft.com/en-us/library/gg162712(v=ws.10).aspx.

4.1.3 Configure PVSCSI Controller

4.1.3.1 PVSCSI CONTROLLER DRIVER

Windows Operating systems do not include the driver for the PVSCSI controller[100].

If the boot disk is placed on PVSCSI controller, drivers should be provided during the OS installation process. The drivers are available within VMware Tools. If only data disks are located on PVSCSI controller(s), VMware Tools should be installed after the OS installation is complete. Starting with Windows Server 2016, the PVSCSI driver is added to the Windows Update—this integration will allow you to update the PVSCSI driver during a Windows update process[101].

4.1.3.2 PVSCSI CONTROLLER QUEUE DEPTH

Using the PVSCSI virtual storage controller, Windows Server is not aware of the increased I/O capabilities supported. The queue depth can be adjusted for PVSCSI in

[99] Starting with the VMware Tools version 10.2.5, RSS is enabled by default for VMXNET3 adapter for the new installation of tools. If VMware Tools that were upgraded from lower version as 10.2.5 steps listed in this document is required.

[100] http://kb.vmware.com/kb/1010398

[101] https://docs.vmware.com/en/VMware-Tools/10.3/rn/vmware-tools-1030-release-notes.html

Windows Server to 254 for maximum performance. This is achieved by adding the following key in the Windows Server registry "HKLM\SYSTEM\CurrentControlSet\services\pvscsi\Parameters\Device /v DriverParameter /t REG_SZ /d "RequestRingPages=32,MaxQueueDepth=254"[102].

NOTE: While increasing the default queue depth of a virtual SCSI controller can be beneficial to an SQL Server-based VM, the configuration can also introduce unintended adverse effects in overall performance if not done properly[103]. VMware highly recommends that customers consult and work with the appropriate storage vendor's support personnel to evaluate the impact of such changes and obtain recommendations or other adjustments that may be required to support the increase in queue depth of a virtual SCSI controller.

4.1.4 Using Antivirus Software[104]

Customers might have requirements that antivirus scanning software must run on all servers, including those running SQL Server instances. Microsoft has published strict guidelines if you need to run antivirus where SQL Server is installed specifying exceptions for the on-line scan engine to be configured.

4.1.5 Other Applications

The use of secondary applications on the same server as a SQL Server should be scrutinized, as misconfiguration or errors in these applications can cause availability and performance challenges for the SQL Server.

4.2 Linux Server Configuration

This section will cover the configuration considerations for the Linux distributions supported by SQL Server 2017 and later. The same general best practices such as using PVSCSI controllers and VMXNET3 for the vNICs are true for Linux as well. The major differences will be discussed.

4.2.1 Supported Linux Distributions

As of the writing of this document[105], the supported Linux distributions and versions for SQL Server deployments are:

- Red Hat Enterprise Linux (RHEL) 7.3 or 7.4;
- SUSE Linux Enterprise Server (SLES) v12 SP2;
- Ubuntu 16.04.

4.2.2 VMware Tools

Unlike Windows Server, the VMware Tools generally ship with most of the common Linux distributions and was incorporated into the Linux kernel as of 2.6.33 as the open VM Tools package[106]. They are also updated via the OS vendor.

---

[102] http://kb.vmware.com/kb/2053145

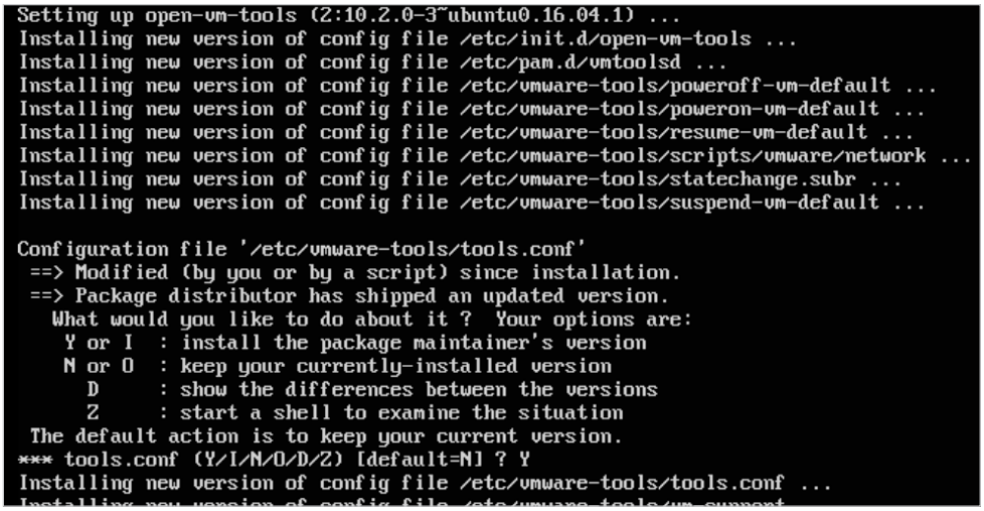
[103] https://docs.vmware.com/en/VMware-vSphere/6.5/com.vmware.vsphere.troubleshooting.doc/GUID-53B382A8-0330-47C1-8E43-94125BCA8AD0.html

[104] See KB309422, How to choose antivirus software to run on computers that are running SQL Server at https://support.microsoft.com/en-us/help/309422/how-to-choose-antivirus-software-to-run-on-computers-that-are-running-sql-server

[105] Consult the official Microsoft documentation at https://docs.microsoft.com/en-us/sql/linux/sql-server-linux-setup?view=sql-server-linux-2017 to see any updates or changes.

[107] https://docs.vmware.com/en/VMware-Tools/10.3.0/com.vmware.vsphere.vmwaretools.doc/GUID-8B6EA5B7-453B-48AA-92E5-DB7F061341D1.html

Figure 43.
Updating the VMware Tools as Part of an Ubuntu Update



To verify that the VMware Tools are installed, you can use the following commands in the guest:

ps ax | grep vmware

That command can show that some processes detect that the server is virtualized (for example, how color is or is not displayed)

lsmod | grep -I vmw

That command allows end-users to see things like the network and PVSCSI driver.

Figure 44.
Showing the VMware Tools Under RHEL

If for some reason the VMware Tools are not installed, ensure they are installed to get the best performance from SQL Server. To get the specific package for your distribution of Linux, it can be found on the "VMware Tools Operating System Specific Packages (OSPs)" web page https://www.vmware.com/support/packages.html. Links to installation instructions are also on that page.

4.2.3 Power Scheme

Similar to Windows Server, the hypervisor, and the underlying hardware, ensure that the distribution of Linux is set to run at full performance. Each distribution has its own method for how to configure the power scheme.

- RHEL https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/power_management_guide/index
- SLES https://en.opensuse.org/Powersaving
- Ubuntu https://askubuntu.com/questions/410860/how-to-permanently-set-cpu-power-management-to-the-powersave-governor

4.2.4 Receive Side Scaling

RSS, also known as multi-queue receive, is also implemented in Linux. In some distributions it is referred to as Receive Packet Steering (RPS), which is the software version of hardware-based RSS. The links below discuss RSS for supported distribution for SQL Server.

- RHEL https://access.redhat.com/solutions/62877
- SLES https://www.suse.com/documentation/sles11/book_sle_tuning/data/sec_tuning_network_rps.html

4.3 SQL Server Configuration

4.3.1 Maximum Server Memory and Minimum Server Memory

SQL Server can dynamically adjust memory consumption based on workloads. SQL Server maximum server memory and minimum server memory configuration settings allow you to define the range of memory for the SQL Server process in use. The default setting for minimum server memory is 0, and the default setting for maximum server memory is 2,147,483,647 MB. Minimum server memory will not immediately be allocated on startup. However, after memory usage has reached this value due to client load, SQL Server will not free memory unless the minimum server memory value is reduced.

SQL Server can consume all the memory on the VM if left unchecked. Setting the maximum SQL Server instance memory allows you to reserve sufficient memory for the OS and other applications running on the VM. In a traditional SQL Server consolidation scenario where you are running multiple instances of SQL Server on the same VM, setting maximum server memory will allow memory to be shared effectively between the instances.

Setting the minimum instance memory is a good practice to maintain SQL Server performance under host memory pressure. When running SQL Server on vSphere, if the vSphere host is under memory pressure, the balloon drive might inflate and take memory back from the SQL Server VM. Setting the minimum server memory provides SQL Server with at least a reasonable amount of memory.

For Tier 1 mission-critical SQL Server deployments, consider setting the SQL Server memory to a fixed amount by setting both maximum and minimum server memory to the same value. Before setting the maximum and minimum server memory, confirm that adequate memory is left for the OS and VM overhead. For performing SQL Server maximum server memory sizing for vSphere, use the following formulas as a guide:

SQL Max Server Memory = VM Memory - ThreadStack - OS Mem - VM Overhead

*ThreadStack = SQL Max Worker Threads * ThreadStackSize*

ThreadStackSize = 1MB on x86

= 2MB on x64

OS Mem: 1GB for every 4 CPU Cores

4.3.2 Lock Pages in Memory

Granting the Lock Pages in Memory user right to the SQL Server service account prevents SQL Server buffer pool pages from paging out by Windows Server. This setting is useful and has a positive performance impact because it prevents Windows Server from paging a significant amount of buffer pool memory out of the process, enabling SQL Server to manage the reduction of its own working set.

Any time Lock Pages in Memory is used, because SQL Server memory is locked and cannot be paged out by Windows Server, you might experience negative impacts if the vSphere balloon driver is trying to reclaim memory from the VM. If you set the SQL Server Lock Pages in Memory user right, also set the VM's reservations to match the amount of memory you set in the VM configuration. Used incorrectly and during times of memory overcommit pressure, SQL Server instability could occur as a result.

If you are deploying a Tier 1 mission-critical SQL Server installation, consider setting the Lock Pages in Memory user right[107] and setting VM memory reservations to improve the performance and stability of your SQL Server running vSphere.

Lock Pages in Memory should also be used in conjunction with the Max Server Memory setting to avoid SQL Server taking over all memory on the VM.

For lower-tiered SQL Server workloads where performance is less critical, the ability to overcommit to maximize usage of the available host memory might be more important. When deploying lower-tiered SQL Server workloads, VMware recommends that you do not enable the Lock Pages in Memory user right. Lock Pages in Memory causes conflicts with vSphere balloon driver. For lower tier SQL Server workloads, it is better to have balloon driver manage the memory dynamically for the VM containing that instance. Having balloon driver dynamically manage vSphere memory can help maximize memory usage and increase consolidation ratio.

4.3.3 Large Pages[108]

Hardware assist for MMU virtualization typically improves the performance for many workloads. However, it can introduce overhead arising from increased latency in the processing of TLB misses. This cost can be eliminated or mitigated with the use of large pages[109].

---

[107] http://msdn.microsoft.com/en-us/library/ms190730.aspx

[108] Refer to *SQL Server and Large Pages Explained* (http://blogs.msdn.com/b/psssql/archive/2009/06/05/sql-server-and-large-pages-explained.aspx) for additional information on running SQL Server with large pages.

[109] http://www.vmware.com/resources/techresources/1039

SQL Server supports the concept of large pages when allocating memory for some internal structures and the buffer pool, when the following conditions are met:

- You are using SQL Server Enterprise Edition.
- The computer has 8 GB or more of physical RAM.
- The Lock Pages in Memory privilege is set for the service account.

As of SQL Server 2008, some of the internal structures, such as lock management and buffer pool, can use large pages automatically if the preceding conditions are met. You can confirm that by checking the *ERRORLOG* for the following messages:

2009-06-04 12:21:08.16 Server Large Page Extensions enabled.

2009-06-04 12:21:08.16 Server Large Page Granularity: 2097152

2009-06-04 12:21:08.21 Server Large Page Allocated: 32MB

On a 64-bit system, you can further enable all SQL Server buffer pool memory to use large pages by starting SQL Server with trace flag 834. Consider the following behavior changes when you enable trace flag 834:

- With SQL Server 2012 and later, it is not recommended to enable the trace flag 834 if using the Columnstore feature.
- With large pages enabled in the guest OS, and when the VM is running on a host that supports large pages, vSphere does not perform Transparent Page Sharing on the VM's memory.
- With trace flag 834 enabled, SQL Server startup behaviour changes. Instead of allocating memory dynamically at runtime, SQL Server allocates all buffer pool memory during startup. Therefore, SQL Server startup time can be significantly delayed.
- With trace flag 834 enabled, SQL Server allocates memory in 2MB contiguous blocks instead of 4 KB blocks. After the host has been running for a long time, it might be difficult to obtain contiguous memory due to fragmentation. If SQL Server is unable to allocate the amount of contiguous memory it needs, it can try to allocate less, and SQL Server might then run with less memory than you intended.

Although trace flag 834 improves the performance of SQL Server, it might not be suitable for use in all deployment scenarios. With SQL Server running in a highly-consolidated environment, this setting is not recommended. This setting is more suitable for high performance Tier 1 SQL Server workloads where there is no oversubscription of the host, and no overcommitment of memory. Always confirm that the correct large page memory is granted by checking messages in the SQL Server ERRORLOG. See the following example:

2009-06-04 14:20:40.03 Server Using large pages for buffer pool.

2009-06-04 14:27:56.98 Server 8192 MB of large page memory allocated.

4.3.4 CXPACKET, MAXDOP, and CTFP

When a query runs on SQL Server using a parallel plan, the query job is divided to multiple packets and processed by multiple cores. The time the system waits for the query to finish is calculated as CXPACKET wait time[110].

MAXDOP, or maximum degree of parallelism, is an advanced configuration option that controls the number of cores used to execute a query in a parallel plan. The MAXDOP setting can be defined on the instance, database[111], and query level. Setting the value to one disables parallel plans altogether. Setting the value to zero lets SQL Server engine to use all available cores if a query runs in parallel.

CTFP, or cost threshold for parallelism, is an option that specifies the threshold at which parallel plans are used for queries. The value is specified in cost units and is not a unit of time[112]. A default value of five is typically considered too low for today's CPU speeds.

There is a fair amount of misconception and incorrect advice on the Internet regarding the values of these advanced configuration options in a virtual environment. When low performance is observed on their database and CXPACKET wait time is high, many DBAs decide to disable parallelism altogether by setting MAXDOP value to one (1). This is not recommended because there might be large jobs that will benefit from processing on multiple CPUs. The recommendation instead is to increase the CTFP value from five to approximately 50 to make sure only large queries run in parallel. Set the MAXDOP according to Microsoft's recommendation[113] for the number of cores in the VM's vNUMA node (no more than eight).

In any case, the configuration of these advanced settings is dependent on the front-end application workload using the SQL Server and should be done after thoughtful testing.

4.3.5 Instant file Initialization

SQL Server engine produces a lot of disk operations, most of them (for example, INSERT rows in tables) will use already preallocated disk space, but some of them require a new, previously unused by SQL Server, disk space to be added. Such operations include creation of a database, adding space or file to an existing database (either manual or due to auto growth) or restore a database operation. By default, SQL Server engine will first clean erase the space to be added by writing zeroes (zeroing), which require additional time and disk IO. To speed up disk operations an option was introduced to allow an instant file initialization. As per Microsoft, "in SQL Server, data files can be initialized instantaneously to avoid zeroing operations. Instant file initialization allows for fast execution of the previously mentioned file operations"[114].

[110] In SQL Server version 2016 SP2 and 2017 RTM CU3 and higher the CXPACKET wait time is split between CXPACKET and CXCONSUMER, see more details here: https://blogs.msdn.microsoft.com/sqlreleaseservices/sql-server-2016-service-pack-2-sp2-released/

[111] Starting with SQL Server version 2016.

[112] More details: https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/configure-the-cost-threshold-for-parallelism-server-configuration-option?view=sql-server-2017

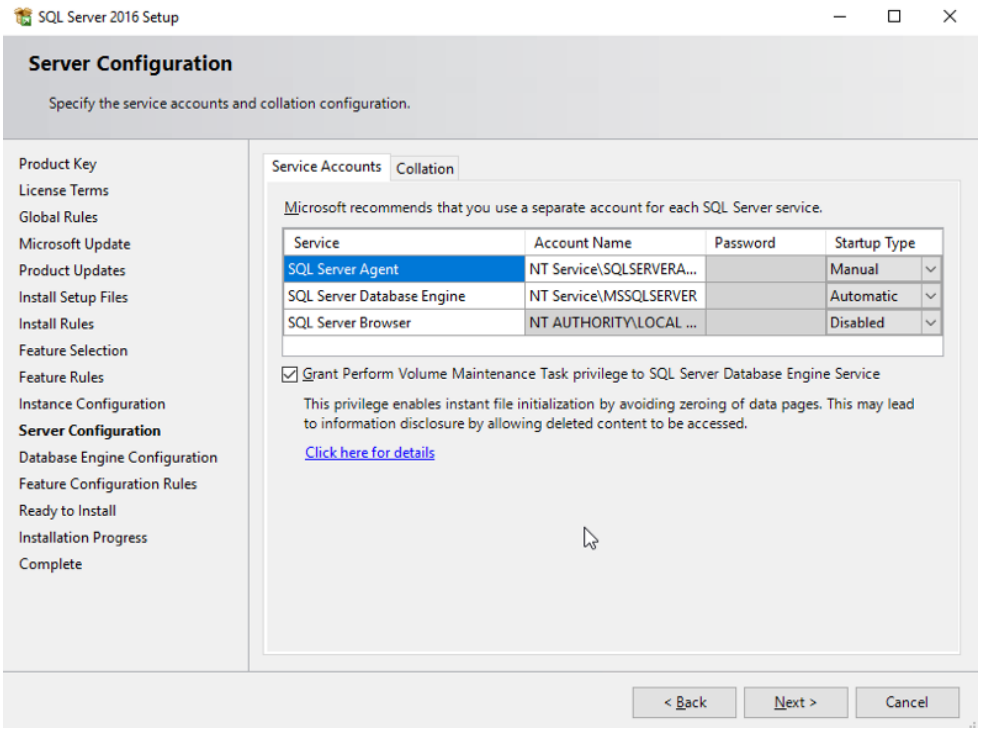
[113] More information available here: https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/configure-the-max-degree-of-parallelism-server-configuration-option?view=sql-server-2017

[114] https://docs.microsoft.com/en-us/sql/relational-databases/databases/database-instant-file-initialization?view=sql-server-2017

Enabling Instant file initialization provides positive effect on disk operations, especially database restore, and where performance of the database engine is the primary goal, should be enabled. It should be mentioned that this option is not affect space allocation for database log files. It also might pose a security risk which generally exists in all situations where a disk space is re-used without zeroing[115].

In order to enable instant file initialization either manually assign Perform Volume Maintenance Tasks security policy to a SQL Server service account or, starting with SQL Server 2016, use the checkbox in SQL Server instance installation wizard as shown on the figure below.

Figure 45.
Enable Instant File Initialization



[115] For more details see https://blogs.msdn.microsoft.com/sql_pfe_blog/2009/12/22/how-and-why-to-enable-instant-file-initialization/

5. VMware Enhancements for Deployment and Operations

vSphere provides core virtualization functionality. The extensive software portfolio offered by VMware is designed to help customers to achieve the goal of 100 percent virtualization and the SDDC. This section reviews some of the VMware products that can be used with virtualized SQL Server implementations in VMs on vSphere.

5.1 Network Virtualization with VMware NSX for vSphere

Although virtualization has allowed organizations to optimize their compute and storage investments, the network has remained mostly physical. VMware NSX® for vSphere solves data center challenges found in physical network environments by delivering software-defined networking and security. Using existing vSphere compute resources, network services can be delivered quickly to respond to business challenges. VMware NSX is the network virtualization platform for the SDDC. By bringing the operational model of a VM to your data center network, you can transform the economics of network and security operations. NSX lets you treat your physical network as a pool of transport capacity, with network and security services attached to VMs with a policy-driven approach.

5.2 VMware vRealize Operations Manager

Maintaining and operating virtualized SQL Server is the vital part of the infrastructure lifecycle. It's very important that the solution architecture already includes all necessary steps to ensure proper operations of the environment.

For the virtualized SQL Server, consider following requirements for a monitoring tool:

- Ability to provide end-to-end monitoring from a database objects through virtual machine back to the physical hosts and storage in use
- Ability to maintain, visualize, and dynamically adjust the relationships between the components of the solution
- Ability to maintain mid- and long-term time series data
- Ability to collect the data from virtualized and non-virtualized SQL Server instances

VMware vRealize® Operations Manager™ (vROPs) is one of the tools that is able to fulfil all the requirements mentioned above when combined with vital extensions such as Blue Medora SQL Server Management Pack.

When performance or capacity problems arise in your SQL Server environment, vRealize Operations Manager can analyze metrics from the application all the way through to the infrastructure to provide insight into problematic components, whether they are compute (physical or virtual), storage, networking, OS, or application related. By establishing trends over time, vRealize Operations Manager can minimize false alerts and proactively alert on the potential root cause of increasing performance problems before end users are impacted.

To monitor the SQL Server application and ingest data from the SQL Server database into vRealize Operations Manager dashboards, there are two options:

- Utilize the EPO management pack for SQL Server provided by VMware: This management pack is included with vRealize Operations™ Enterprise and can be implemented by the customer or VMware services. The EPO management pack is collecting information from SQL Server deployments using an agent and does not include capacity management information.
- VMware vRealize Operations Management Pack for Microsoft SQL Server from Blue Medora[116]: While this solution incurs additional cost, it provides added value with agentless integration and includes performance metric from the SQL Server database instances, both physical and virtual.

[116] For more details see https://blogs.msdn.microsoft.com/sql_pfe_blog/2009/12/22/how-and-why-to-enable-instant-file-initialization/

6. Resources

SQL Server on VMware vSphere:

- Microsoft SQL Server on VMware vSphere® Availability and Recovery Options
  https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/solutions/sql-server-on-vmware-availability-and-recovery-options.pdf
- Performance characterization of Microsoft SQL Server on VMware vSphere 6.5
  https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/performance/sql-server-vsphere65-perf.pdf
- Planning highly available, mission-critical SQL Server deployments with VMware vSphere
  https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/solutions/vmware-vsphere-highly-available-mission-critical-sql-server-deployments.pdf

VMware Blogs:

- Cornac Hogan, "When and why do we 'stun' a virtual machine?"
  https://cormachogan.com/2015/04/28/when-and-why-do-we-stun-a-virtual-machine/
- Frank Deneman. NUMA Deep Dive Series.
  http://frankdenneman.nl/2016/07/06/introduction-2016-numa-deep-dive-series/
- VMware Application Blog
  https://blogs.vmware.com/apps/microsoft/sql
- VMware Performance Team Blog
  https://blogs.vmware.com/performance
- VMware vSphere Blog
  https://blogs.vmware.com/vsphere
- Virtual Machine vCPU and vNUMA Rightsizing: Rules of Thumb
  https://blogs.vmware.com/performance/2017/03/virtual-machine-vcpu-and-vnuma-rightsizing-rules-of-thumb.html

VMware Knowledgebase:

- A snapshot removal can stop a virtual machine for long time
  http://kb.vmware.com/kb/1002836
- Configuring disks to use VMware Paravirtual SCSI (PVSCSI) adapters
  http://kb.vmware.com/kb/1010398
- Large-scale workloads with intensive I/O patterns might require queue depths significantly greater than Paravirtual SCSI default values
  http://kb.vmware.com/kb/2053145
- Understanding VM snapshots in ESXi / ESX
  https://kb.vmware.com/kb/1015180
- Upgrading a virtual machine to the latest hardware version
  https://kb.vmware.com/kb/1010675
- Virtual machine becomes unresponsive or inactive when taking a snapshot
  https://kb.vmware.com/kb/1013163

VMware Documentation:

- DRS performance VMware vSphere 6.5
  https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/drs-vsphere65-perf.pdf

- SQL Server FCI and File Server on VMware vSAN 6.7 using iSCSI Service
https://storagehub.vmware.com/t/vmware-vsan/sql-server-fci-and-file-server-on-vmware-vsan-6-7-using-iscsi-service/
- Understanding Memory Management in VMware vSphere 5
https://www.vmware.com/techpapers/2011/understanding-memory-management-in-vmware-vsphere-10206.html
- Understanding vSphere DRS performance VMware vSphere 6
https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/vsphere6-drs-perf.pdf
- VMware Tools
https://docs.vmware.com/en/VMware-Tools/
- VMware vCenter Server and Host Management
https://docs.vmware.com/en/VMware-vSphere/6.7/com.vmware.vsphere.vcenterhost.doc/GUID-3B5AF2B1-C534-4426-B97A-D14019A8010F.html
- VMware vSphere virtual machine encryption performance VMware vSphere 6.5
http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/vm-encryption-vsphere65-perf.pdf
- vSphere 5.1 – VMDK versus RDM
https://blogs.vmware.com/vsphere/2013/01/vsphere-5-1-vmdk-versus-rdm.html
- vSphere Availability 6.5
https://docs.vmware.com/en/VMware-vSphere/6.5/vsphere-esxi-vcenter-server-65-availability-guide.pdf
- vSphere Resource Management. vSphere 6.7
https://docs.vmware.com/en/VMware-vSphere/6.7/vsphere-esxi-vcenter-server-67-resource-management-guide.pdf
- vSphere Security. vSphere 6.7
https://docs.vmware.com/en/VMware-vSphere/6.7/vsphere-esxi-vcenter-server-67-security-guide.pdf

SQL Server Resources:

- Compute capacity limits by edition of SQL Server
https://docs.microsoft.com/en-us/sql/sql-server/compute-capacity-limits-by-edition-of-sql-server?view=sql-server-2017
- Description of support for network database files in SQL Server
https://support.microsoft.com/en-us/help/304261/description-of-support-for-network-database-files-in-sql-server
- Editions and supported features of SQL Server 2017
https://docs.microsoft.com/en-us/sql/sql-server/editions-and-components-of-sql-server-2017?view=sql-server-2017
- How It Works (It Just Runs Faster): Non-Volatile Memory SQL Server Tail Of Log Caching on NVDIMM
https://blogs.msdn.microsoft.com/bobsql/2016/11/08/how-it-works-it-just-runs-faster-non-volatile-memory-sql-server-tail-of-log-caching-on-nvdimm/
- How It Works: Soft NUMA, I/O Completion Thread, Lazy Writer Workers and Memory Nodes
https://blogs.msdn.microsoft.com/psssql/2010/04/02/how-it-works-soft-numa-io-completion-thread-lazy-writer-workers-and-memory-nodes/

- Memory Management Architecture Guide
  https://docs.microsoft.com/en-us/sql/relational-databases/memory-management-architecture-guide?view=sql-server-2017
- Operating System Best Practice Configurations for SQL Server
  https://blogs.msdn.microsoft.com/docast/2018/02/01/operating-system-best-practice-configurations-for-sql-server/
- Soft-NUMA (SQL Server)
  https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/soft-numa-sql-server?view=sql-server-2017
- SQL 2016 – It Just Runs Faster: Automatic Soft NUMA
  https://blogs.msdn.microsoft.com/bobsql/2016/06/03/sql-2016-it-just-runs-faster-automatic-soft-numa/
- SQL Server and Large Pages Explained -
  http://blogs.msdn.com/b/psssql/archive/2009/06/05/sql-server-and-large-pages-explained.aspx
- Transaction Commit latency acceleration using Storage Class Memory in Windows Server 2016/SQL Server 2016 SP1
  https://blogs.msdn.microsoft.com/sqlserverstorageengine/2016/12/02/transaction-commit-latency-acceleration-using-storage-class-memory-in-windows-server-2016sql-server-2016-sp1/
- Virtualization-based Security (VBS)
  https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/oem-vbs

7. Acknowledgments

Author: Oleg Ulyanov, Sr. Solutions Architect, Microsoft Applications

Special contributors:

- David Klee: Founder and Chief Architect at Heraflux Technologies, Microsoft MVP, vExpert. Provided thoughtful review of the document and special contribution on the NUMA Considerations and SQL Server Configuration (4.3) sections.
- Allan Hirt: Managing Partner at SQLHA LLC, Dual Microsoft MVP, and vExpert. Authored Linux Server Configuration section (4.2) and provided valuable comments to the whole document.
- Frank Denneman: Chief Technologist, VMware. The vNUMA section of this document is based on his research available online here: http://frankdenneman.nl/2016/07/06/introduction-2016-numa-deep-dive-series/
- Dave Morera: Sr. Solution Architect and Tony Wu: Sr. Solution Architect. The VMware vSAN section (3.8.2) was developed in a close collaboration with Dave and Tony.

Thanks to the following people for their inputs:

- Deji Akomolafe: Staff Solutions Architect, Microsoft Applications
- Mark Achtemichuk: Staff Engineer, Performance Engineering
- Sudhir Balasubramanian: Staff Solution Architect, Data Platforms
- Vas Mitra: Staff Solutions Architect
- Mohan Potheri: Sr. Solutions Architect, Technical Marketing
- Valentin Bondzio: Sr. Staff Technical Support Engineer

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com
Copyright © 2019 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at http://www.vmware.com/go/patents. VMware is a registered trademark or trademark of VMware, Inc. and its subsidiaries in the United States and other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies. Item No: VMW_0818_1864_WPS_Architecting Microsoft SQL Server on VMware vSphere_WhitePaper_5.2_YC
11/18