Architecting Microsoft SQL Server on VMware vSphere

Table of contents

Architecting Microsoft SQL Server on VMware vSphere

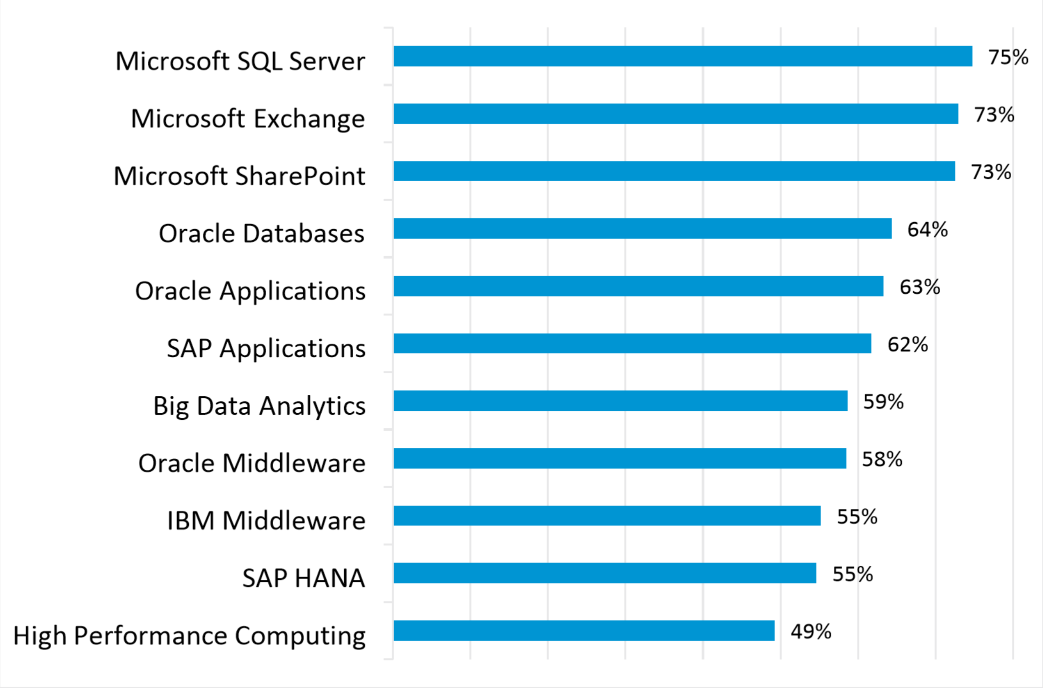
Introduction

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.

Figure 1. Percent of Customers Operating the Virtualized Instance of Applications[2]



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 SDDC (software-defined data center) 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 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. With the introduction of the "vSphere vMotion Notifications for Latency Sensitive Applications"[3] feature in vSphere 8.0, application owners and administrators are now better able to more finely control and schedule vMotion events, significantly improving SQL Server workloads' availability and resilience. vMotion notification also complements native SQL Server HA capabilities, especially for planned outages and workloads relocation.

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 SQL Server virtual machines and can be used in conjunction with SQL Server's own HA capabilities. Among other features, VMware NSX® provides network virtualization and dynamic security policy enforcement. VMware Site Recovery Manager™ provides disaster recovery plan orchestration, VMware Aria Operations provides comprehensive analytic and monitoring engine. 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 virtualization strategy to achieve the business and technical requirements while keeping operational overhead to a minimum for cost effectiveness.

Purpose

This document provides best practice guidelines for designing and implementing Microsoft SQL Server ("SQL Server") in a virtual machine to run on VMware vSphere ("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.

Target Audience

This document assumes a knowledge and understanding of vSphere and Microsoft 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.
- Database Administrators and support 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.

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 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 database workloads associated with the applications accessing SQL Server.
- Understand availability and recovery requirements, including uptime guarantees (“number of nines”) and disaster recovery for both the VM and the databases.
- Capture resource utilization baselines for existing physical servers that are hosting databases.
- Plan the migration/deployment to vSphere.

Understand the SQL Server Workload

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 databases (online transaction processing) 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 (CPU, memory, disk, and network) to achieve optimal performance. They might also be candidates for clustering with Windows Server Failover Cluster, 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.

Availability and Recovery Options

Running SQL Server on vSphere offers many options for database availability, backup and disaster recovery utilizing the best features from both VMware and Microsoft. This section provides brief overview of different options that exist for availability and recovery"[4].

VMware Business Continuity Options

VMware technologies, such as vSphere HA, vSphere Fault Tolerance, 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.

vSphere High Availability

vSphere HA provides easy to use, cost-effective high availability 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. vSphere HA protects application availability in the following ways:

vSphere Fault Tolerance

vSphere Fault Tolerance (FT) provides a higher level of availability, allowing users to protect any virtual machine 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.

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 above support vMotion of a VM with RDM disks in physical compatibility mode that are part of a Windows failover cluster.

Native SQL Server Capabilities

At the application level, all Microsoft features and techniques are supported on vSphere, including SQL Server Always on Availability Groups, database mirroring, failover cluster instances, and log shipping. These SQL Server features can be combined with vSphere features to create flexible availability and recovery scenarios, applying the most efficient and appropriate tools for each use case.

The following table lists SQL Server availability options and their ability to meet various recovery time objectives (RTO) and recovery point objectives (RPO). Before choosing any one option, evaluate your own business requirements to determine which scenario best meets your specific needs.

Table 1. SQL Server High Availability Options

Technology	Granularity	Storage Type	RPO - Data Loss	RTO - Downtime
Always On Availability Groups	Database	Non-shared	None (with synchronous commit mode)	~3 seconds or Administrator recovery
Always On Failover Cluster Instances	Instance	Shared	None	~30 seconds
Database Mirroring[5]	Database	Non-shared	None (with high safety mode)	< 3 seconds or Administrator recovery
Log Shipping	Database	Non-shared	Possible transaction log	Administrator recovery

For guidelines and information on the supported configuration for setting up any Microsoft clustering technology on vSphere, including Always on Availability Groups, see the Knowledge Base article Microsoft Clustering on vSphere: Guidelines for supported configurations (1037959) at http://kb.vmware.com/kb/1037959.

VMware Cloud on VMware-Powered Hybrid Clouds

VMware Cloud on AWS is one example of several VMware-powered Public and Hybrid Cloud offerings. VMware Cloud on AWS brings VMware's enterprise-class SDDC software to the AWS Cloud with optimized access to AWS services. 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.[6]

VMware Cloud on AWS allows to consume public cloud in the same manner and with the same toolset as the on-premises vSphere environment. VMs can be bi-directionally migrated using vSphere vMotion technology between on-premises datacenters and VMware-based Cloud services without any modification for the VM or application configuration. You can learn more about migration and optimization in this document.

Consider the following use cases enabled by VMware-based cloud infrastructure for an instance of virtualized SQL Server deployed on a VM in on-premises datacenter:

- 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
- Provide Disaster Recovery as a Service with VMware Cloud Disaster Recovery (VCDR) SaaS Solution

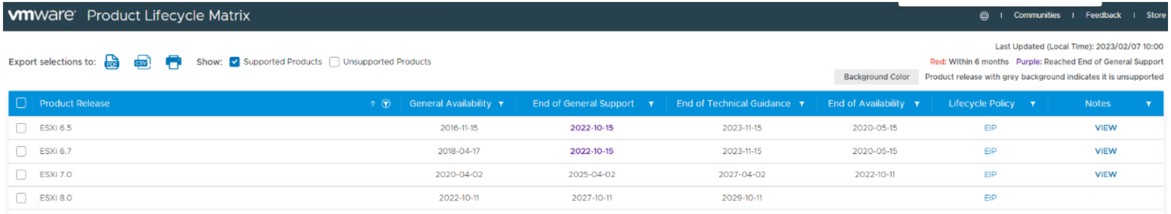
After an instance of a virtualized SQL Server is moved to a VMware vSphere-based Cloud instance, operational and configuration guidelines summarized in this document continue to apply[7].

SQL Server on vSphere Supportability Considerations

One of the goals of the purpose-built solution architecture is to provide a solution which could be easily operated and maintained. One of the cornerstones of the "Day two" operational routine is to ensure that the only supported configurations are used.

Consider following points while architecting SQL Server on vSphere

1. Use VMware Configuration Maximums Tool[8] to check the final architecture if any limits are reached or may be reached in the near future
2. Use VMware Compatibility Guide[9] to check compatibility for all components used
3. Use VMware Lifecycle Product Matrix[10] to find the End of General Support (EGS) date for solutions in use. For example, as of time of writing this document, general support ESXi/vCenter 6.7 ended 15 October 2022, and further technical guidance will stop beginning from Novermber, 15, 2023



vmware Product Lifecycle Matrix
Communities | Feedback | Store
Export selections to:
Show: ☑ Supported Products □ Unsupported Products

Last Updated (Local Time): 2023/02/07 10:00
Red: Within 6 months Purple: Reached End of General Support
Background Color Product release with grey background indicates it is unsupported

□ Product Release	General Availability	End of General Support	End of Technical Guidance	End of Availability	Lifecycle Policy	Notes
□ ESXi 6.5	2016-11-15	2022-10-15	2023-11-15	2020-05-15	EIP	VIEW
□ ESXi 6.7	2018-04-17	2022-10-15	2023-11-15	2020-05-15	EIP	VIEW
□ ESXi 7.0	2020-04-02	2025-04-02	2027-04-02	2022-10-11	EIP	VIEW
□ ESXi 8.0	2022-10-11	2027-10-11	2029-10-11		EIP	

4. Recheck Microsoft Support Knowledge Base Article "Support policy for SQL Server products that are running in a hardware virtualization environment"[11]. For now, all version of SQL Server higher than SQL Server 2008 are supported by Microsoft while running on a virtual platform.
5. Microsoft officially supports virtualizing Microsoft SQL Server on all currently-shipping and supported versions of VMware vSphere. The list of all supported VMware vSphere versions is available for easy reference on the Windows Server Virtualization Validation Program[12] website. 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, as described here[13].
6. Relaxed policies for application license mobility – Starting with the release of Microsoft SQL Server 2012, Microsoft further relaxed its licensing policies 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 r server in another server farm, or to a non - private cloud, once every 90 days.

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.

Right-Sizing

Right-sizing is a term that means allocating the appropriate amount of compute resources, such as virtual CPUs and RAM, to the virtual machine to power the database workload instead of adding more than is actively utilized, which is 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 spare 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 supposed 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 can be added non-disruptively, or with a brief reboot of the VM. To find out how many resources are required for the target SQL Server VM, monitor the source physical SQL Server (if one exists) using dynamic management views (DMV)-based tools, or leverage monitoring software, such as the VMware vRealize True Visibility Suite. The amount of collected time series data should be enough to capture all relevant workloads spikes (such as quarter-end or monthly reports), but at least two weeks at a minimum to capture enough data to be considered a true baseline.

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.

When in doubt, start with the lower amount of resources 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. Adjustments can be based on additional monitoring using a DMV-based tool, similar to monitoring a physical SQL Server deployment. VMware vRealize True Visibility Suite can perform DMV-based monitoring with ongoing capacity management and will alert if there is resource waste or contention points.

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

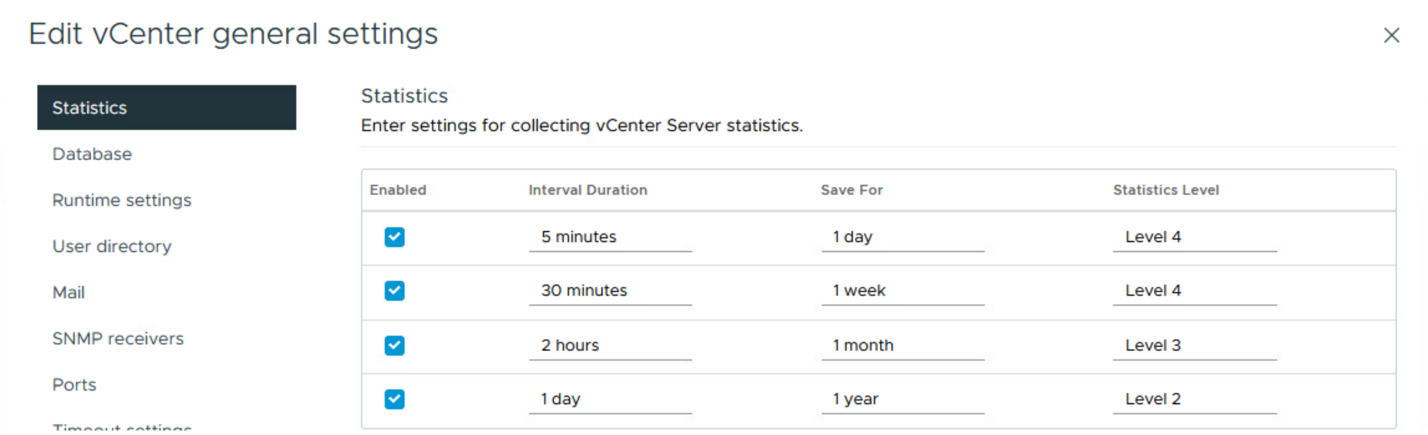
- Configuring a VM with more virtual CPUs than its workload can use might cause slightly increased resource usage, potentially impacting performance on heavily loaded systems. Common examples of this include a single-threaded workload running in a multiple-vCPU VM, or a multithreaded workload in a virtual machine with more vCPUs than the workload can effectively use. Even if the guest operating system 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 SQL Server VM with the VMware Active Memory counter[14]. Applications that contain their own memory management or use large amounts of memory as a storage read cache, 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 allocated to a SQL Server VM.
- Having more vCPUs assigned for the virtual SQL Server also has SQL Server licensing implications in certain scenarios, such as per-virtual-core licenses.

Adding resources to VMs (a click of a button) is much easier than adding resources to physical machines.

vCenter Server Configuration

The vCenter server configuration, by default, is set to a base level of statistics collection, 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. To achieve a significantly longer retention of granular metrics, the following statistics levels are recommended.

Figure 2. vCenter Server Statistics



Edit vCenter general settings

- Statistics
- Database
- Runtime settings
- User directory
- Mail
- SNMP receivers
- Ports
- Timeout settings

Statistics

Enter settings for collecting vCenter Server statistics.

Enabled	Interval Duration	Save For	Statistics Level
☑	5 minutes	1 day	Level 4
☑	30 minutes	1 week	Level 4
☑	2 hours	1 month	Level 3
☑	1 day	1 year	Level 2

ESXi Cluster Compute Resource Configuration

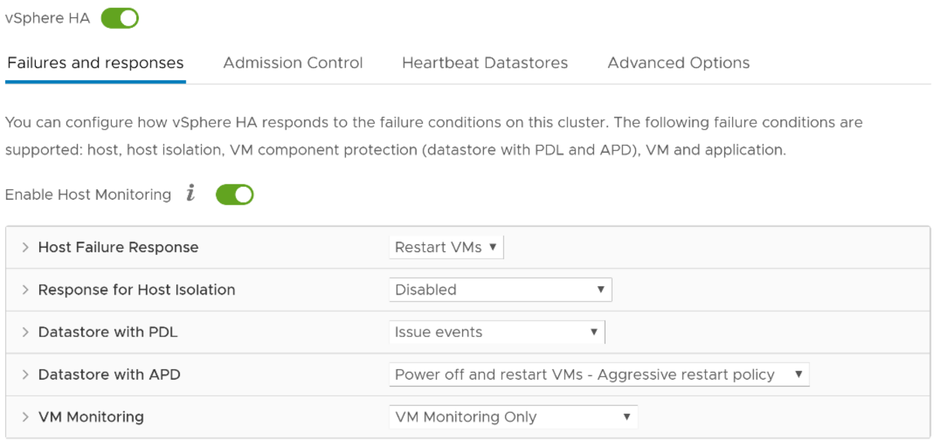
The vSphere host cluster configuration is vital for the wellbeing of a production SQL Server platform. The goals of an appropriately engineered compute resource cluster include maximizing the VM and SQL Server availability, minimizing the impact of hardware component failures, and minimizing the SQL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

vSphere HA

vSphere HA is a feature that provides resiliency to a vSphere environment. If an ESXi host were to fail suddenly, it 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[15].

Figure 3. vSphere HA Settings



vSphere HA: on

Failures and responses | Admission Control | Heartbeat Datastores | Advanced Options

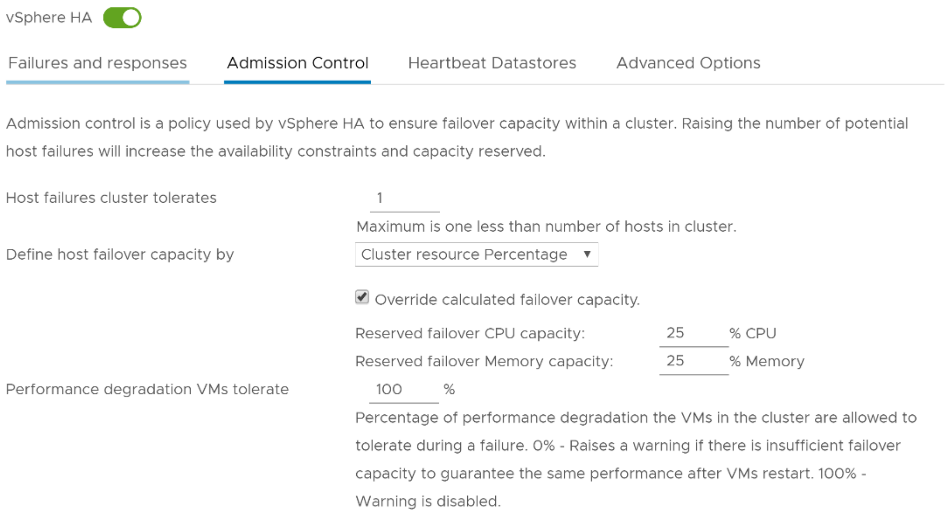
You can configure how vSphere HA responds to the failure conditions on this cluster. The following failure conditions are supported: host, host isolation, VM component protection (datastore with PDL and APD), VM and application.

Enable Host Monitoring: on

Host Failure Response	Restart VMs
Response for Host Isolation	Disabled
Datastore with PDL	Issue events
Datastore with APD	Power off and restart VMs - Aggressive restart policy
VM Monitoring	VM Monitoring Only

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. 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.

Figure 4. vSphere Admission Control Settings



vSphere HA

Failures and responses | Admission Control | Heartbeat Datastores | Advanced Options

Admission control is a policy used by vSphere HA to ensure failover capacity within a cluster. Raising the number of potential host failures will increase the availability constraints and capacity reserved.

Host failures cluster tolerates: 1

Maximum is one less than number of hosts in cluster.

Define host failover capacity by: Cluster resource Percentage

☑ Override calculated failover capacity.

Reserved failover CPU capacity: 25 % CPU

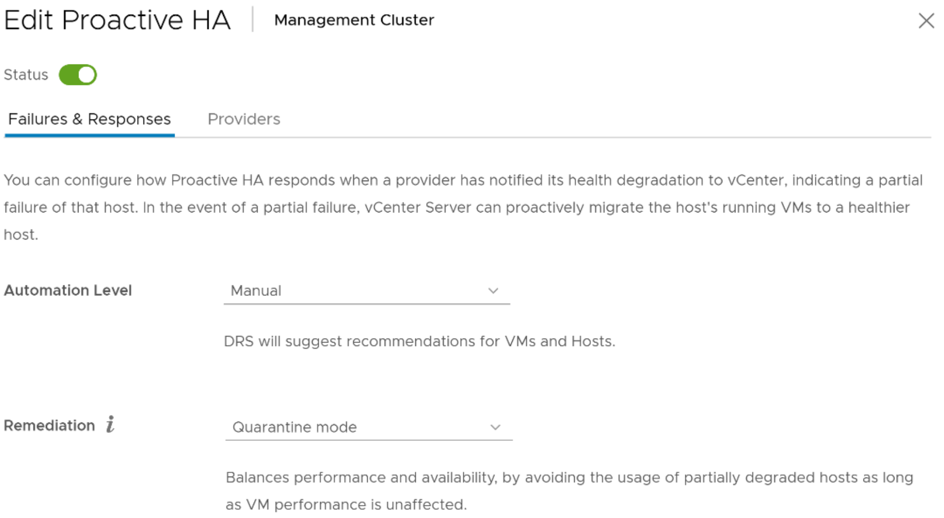
Reserved failover Memory capacity: 25 % Memory

Performance degradation VMs tolerate: 100 %

Percentage of performance degradation the VMs in the cluster are allowed to tolerate during a failure. 0% - Raises a warning if there is insufficient failover capacity to guarantee the same performance after VMs restart. 100% - Warning is disabled.

vSphere 6.5 introduced a new feature called Proactive HA. Proactive HA, when integrated with the VMware Aria Operations (formerly vRealize Operations) detects error conditions in host hardware, and can evacuate a host's VMs onto other hosts in advance of the hardware failure.

Figure 5. Proactive HA



Edit Proactive HA | Management Cluster

Status

Failures & Responses | Providers

You can configure how Proactive HA responds when a provider has notified its health degradation to vCenter, indicating a partial failure of that host. In the event of a partial failure, vCenter Server can proactively migrate the host's running VMs to a healthier host.

Automation Level: Manual

DRS will suggest recommendations for VMs and Hosts.

Remediation: Quarantine mode

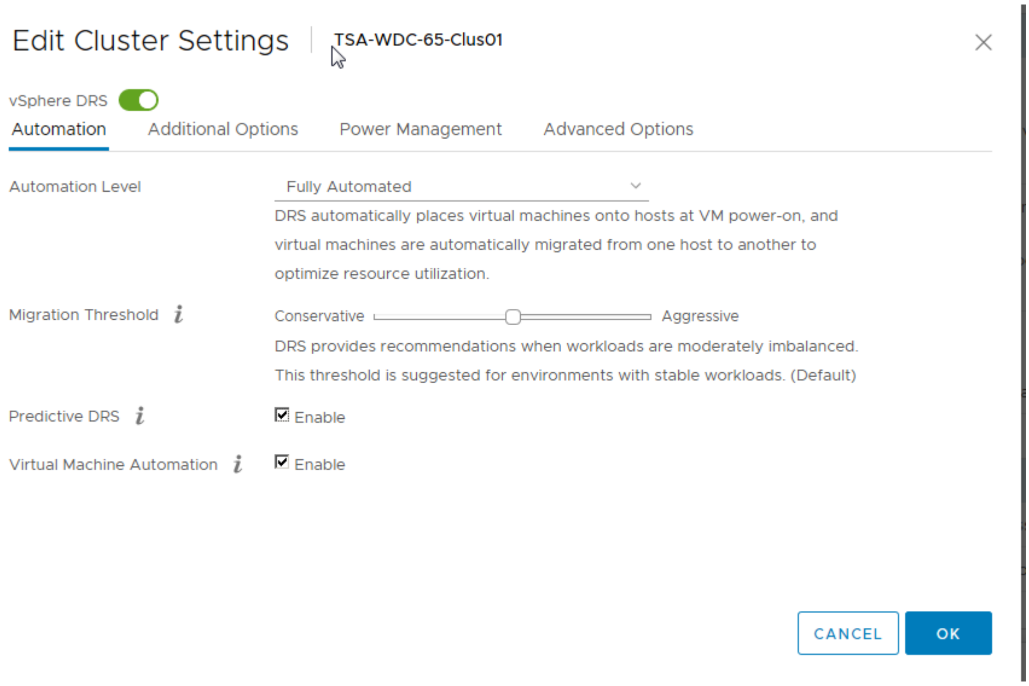
Balances performance and availability, by avoiding the usage of partially degraded hosts as long as VM performance is unaffected.

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 recommend enabling DRS functionality for a cluster hosting SQL Server VMs.[16]

Figure 6. vSphere DRS Cluster



Edit Cluster Settings | TSA-WDC-65-Clus01

vSphere DRS: on

Automation Level: Fully Automated

DRS automatically places virtual machines onto hosts at VM power-on, and virtual machines are automatically migrated from one host to another to optimize resource utilization.

Migration Threshold: Conservative — Aggressive

DRS provides recommendations when workloads are moderately imbalanced. This threshold is suggested for environments with stable workloads. (Default)

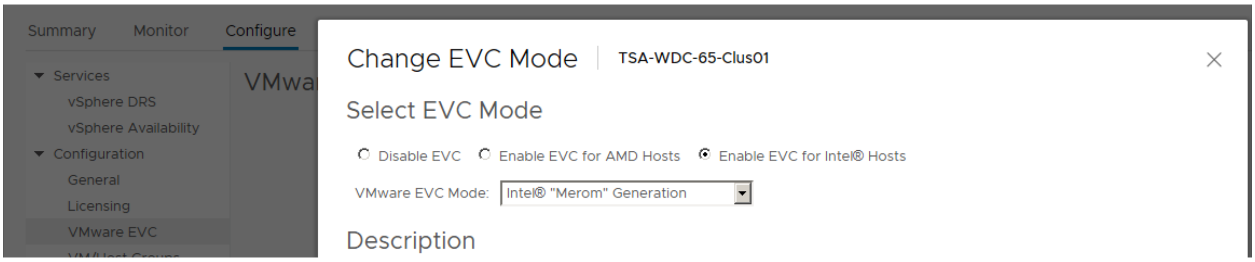
Predictive DRS: ☑ Enable

Virtual Machine Automation: ☑ Enable

VMware EVC[17]

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 CPU feature set in the cluster.

Figure 7. VMware EVC Settings



Change EVC Mode | TSA-WDC-65-Clus01

Select EVC Mode

○ Disable EVC ○ Enable EVC for AMD Hosts ● Enable EVC for Intel® Hosts

VMware EVC Mode: Intel® "Merom" Generation

Description

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

The following use cases might justify enabling EVC mode:

- A cluster consist of hosts with different CPU models and 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 models. Consider using per-VM EVC (Section 3.5.8) if only a subset of VMs might be migrated to another cluster.

Resource Pools[18]

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 VMs over production non-critical VMs over pre-production SQL Server workloads. The resources pools can be configured 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.

ESXi Host Configuration

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

BIOS/UEFI and Firmware Versions

As a best practice, update the BIOS/UEFI 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.

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 selection
- Disable any devices that are not used (for example, serial ports)
- Set Power Management (or its vendor-specific equivalent label) to “OS controlled”. This will enable the ESXi hypervisor to control power management based on the selected policy. See the following section 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 (such as Halt-to-Full), resulting in reduced performance for the VM.

Power Management

The ESXi hypervisor provides a high performance and competitive platform that efficiently runs many Tier-1 application workloads in VMs. 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.

VMware defines latency-sensitive applications as workloads that require optimizing for a few microseconds to a few tens of microseconds end-to-end latencies. This does not apply to applications or workloads in the hundreds of microseconds to tens of milliseconds end-to-end -latencies. In VMware terms of network access times, SQL Server is not typically considered a “latency sensitive” application. However, given the adverse impact of incorrect power settings in a Windows Server operating system, customers must pay special attention to power management.

Server hardware and operating systems are usually engineered to minimize power consumption for economic reasons. Windows Server and the ESXi hypervisor both favor minimized power consumption over performance. 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”

There are three distinct areas of power management in an ESXI hypervisor virtual environment: server hardware, hypervisor, and guest operating system.

ESXi Host Power Settings

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.

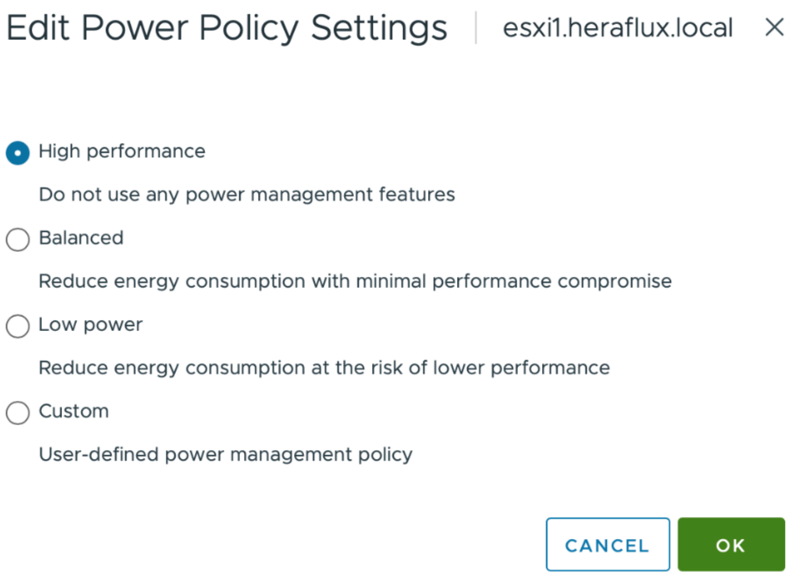
In general, selecting a high-performance policy provides more absolute performance, but at lower efficiency (performance per watt). Lower-power policies provide lower absolute performance, but at higher efficiency. ESXi provides five power management policies. If the host does not support power management, or if the BIOS/UEFI settings specify that the host operating system is not allowed to manage power, only the “Not Supported” policy is available.

Table 2. CPU Power Management Policies

Power Management Policy	Description
High Performance	The VMkernel detects certain power management features, but will not use them unless the BIOS requests them for power capping or thermal events. This is the recommended power policy for an ESXi host running a SQL Server VM.
Balanced (default)	The VMkernel uses the available power management features conservatively to reduce host energy consumption with minimal compromise to performance.
Low Power	The VMkernel aggressively uses available power management features to reduce host energy consumption at the risk of lower performance.
Custom	The VMkernel bases its power management policy on the values of several advanced configuration parameters. You can set these parameters in the vSphere Web Client Advanced Settings dialog box.
Not supported	The host does not support any power management features, or power management is not enabled in the BIOS.

VMware recommends setting the “High performance” Power policy for an ESXi host(s) hosting SQL Server VMs. You select a policy for a host using the vSphere Web Client. If you do not select a policy, ESXi uses Balanced by default.[19]

Figure 8. Recommended ESXi Host Power Management Setting



Edit Power Policy Settings | esxi1.heraflux.local

● High performance

Do not use any power management features

○ Balanced

Reduce energy consumption with minimal performance compromise

○ Low power

Reduce energy consumption at the risk of lower performance

○ Custom

User-defined power management policy

Virtual Machine CPU Configuration

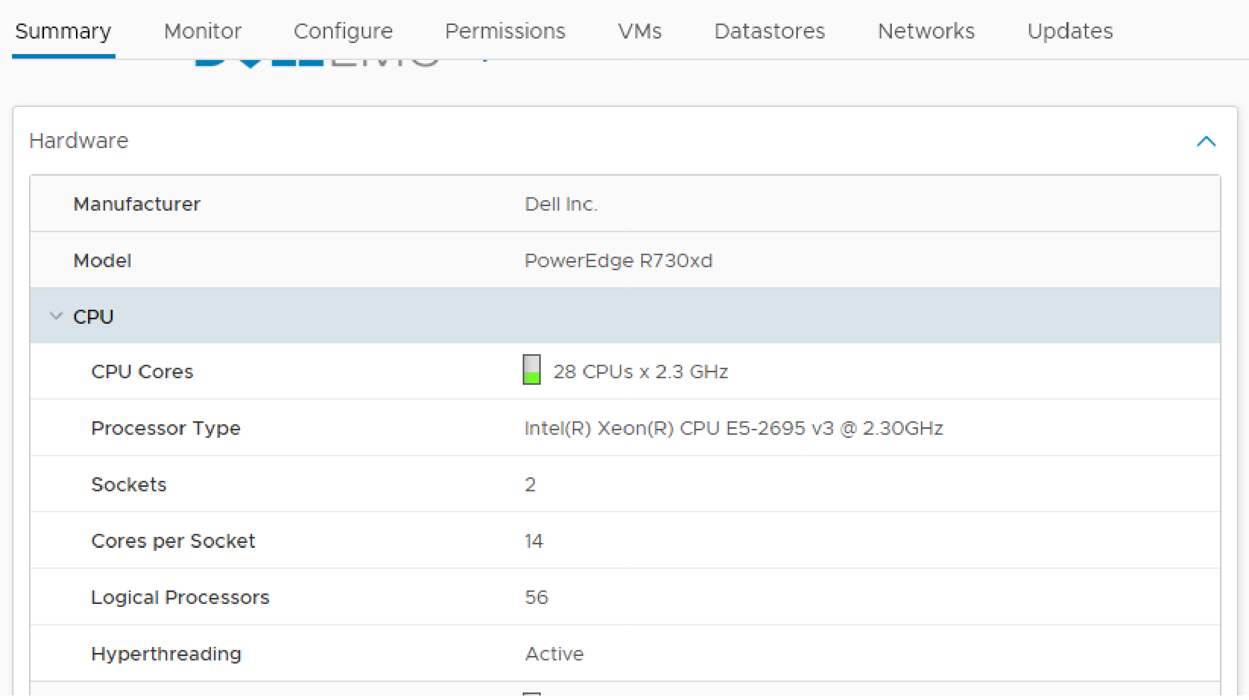
Physical, Virtual, and Logical CPUs and Cores

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[20]:

- CPU: The CPU, or processor, is the component of a computer system that performs the tasks required for computer applications to run. The CPU is the primary element that performs the computer functions. CPUs contain cores.
- CPU Socket: A CPU socket is a physical connector on a computer motherboard that connects to a single physical CPU. Some motherboards have multiple sockets and can connect multiple multicore processors (CPUs). Because of the need to make a clear distinction between “Physical Processor” and “Logical Processor”, this Guide will follow the industry standard practice and use the term “Sockets” wherever we mean “Physical Processors”
- Core: A core contains a unit containing an L1 cache and functional units needed to run applications. Cores can independently run applications or threads. One or more cores can exist on a single CPU.

- Hyperthreading and Logical Processors: Hyperthreading technology allows a single physical processor core to behave like two logical processors. The processor can run two independent applications at the same time. In hyperthreaded systems, each hardware thread is a logical processor. For example, a dual-core processor with hyperthreading activated has two cores and four logical processors[21]. Please see Figure 10 below for a visual representation. It is very important for readers to note that, while hyperthreading can improve workloads performance by facilitating more efficient use of idle resources, a hyper-thread of a Core is not a full-fledged Processor Core and does not perform like one[22]. This awareness will be very useful when making decisions about compute resource allocation and "right-sizing". For more details, see section 5.3.

Figure 9. Physical Server CPU Allocation



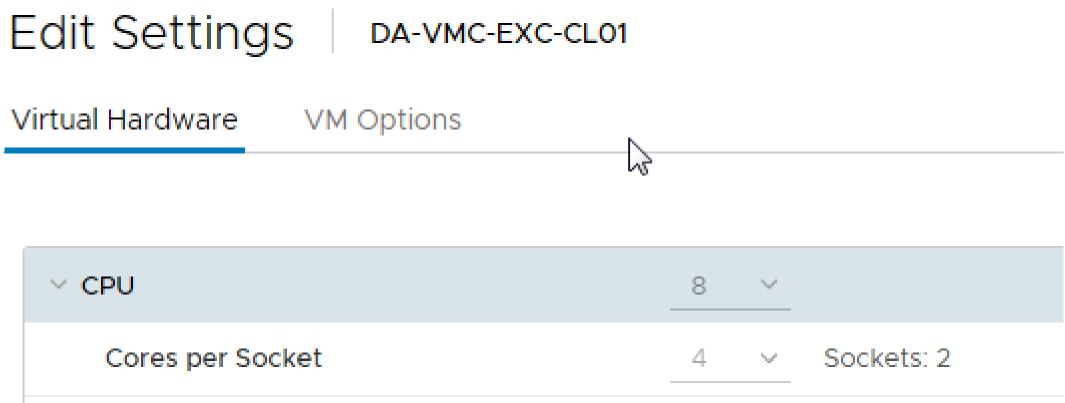
Manufacturer	Dell Inc.
Model	PowerEdge R730xd
CPU	
CPU Cores	28 CPUs x 2.3 GHz
Processor Type	Intel(R) Xeon(R) CPU E5-2695 v3 @ 2.30GHz
Sockets	2
Cores per Socket	14
Logical Processors	56
Hyperthreading	Active

As an example, a host listed on the Figure 10 above, has two pSocket (two pCPUs), 28 pCores and 56 logical Cores as a result of an active Hyper-Threading configuration. Hyperthreading is enabled by default if ESXi detects that the capability is enabled on the Physical ESXi Host. Some vendors refer to "Hyperthreading" as "Logical Processor" in the System BIOS.

- Virtual Socket –number of virtual sockets assigned to a virtual machine. Each virtual socket represents a virtualized physical CPU package and can be configured with one or more virtual cores
- Virtual Core – refers to the number of cores per virtual Socket, starting with vSphere 4.1.
- Virtual CPU (vCPU) – virtualized central processor unit assigned to a VM. Total number of assigned vCPUs to a VM is calculated as:

$$\text{Total vCPU} = (\text{Number of virtual Socket})*(\text{Number of virtual cores per socket})$$

Figure 10. Virtual Machine CPU Configuration



CPU 8

Cores per Socket 4 Sockets: 2

As an example, the virtual machine shown in Figure 10 has two virtual Sockets, each with four vCores, with total number of vCPUs being eight.

Allocating vCPU to SQL Server Virtual Machines

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 cores (rather than the 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 SQL Server workloads run on has 16 physical CPU cores, avoid allocating more than 16 virtual vCPUs for the VMs on that vSphere host during the initial virtualization effort. This initial conservative sizing approach helps rule out CPU resource contention as a possible contributing factor in the event of sub-optimal performance during and after the virtualization project. After you have determined that there is excess capacity to be used, you can increase density in that physical server by adding more workloads into the vSphere cluster and allocating virtual vCPUs beyond the available physical cores. Consider using monitoring tools capable of collecting, storing, and analyzing mid- and long-terms data ranges.

Lower-tier SQL Server workloads typically are less latency sensitive, so in general the goal is to maximize the 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 sufficient performance.

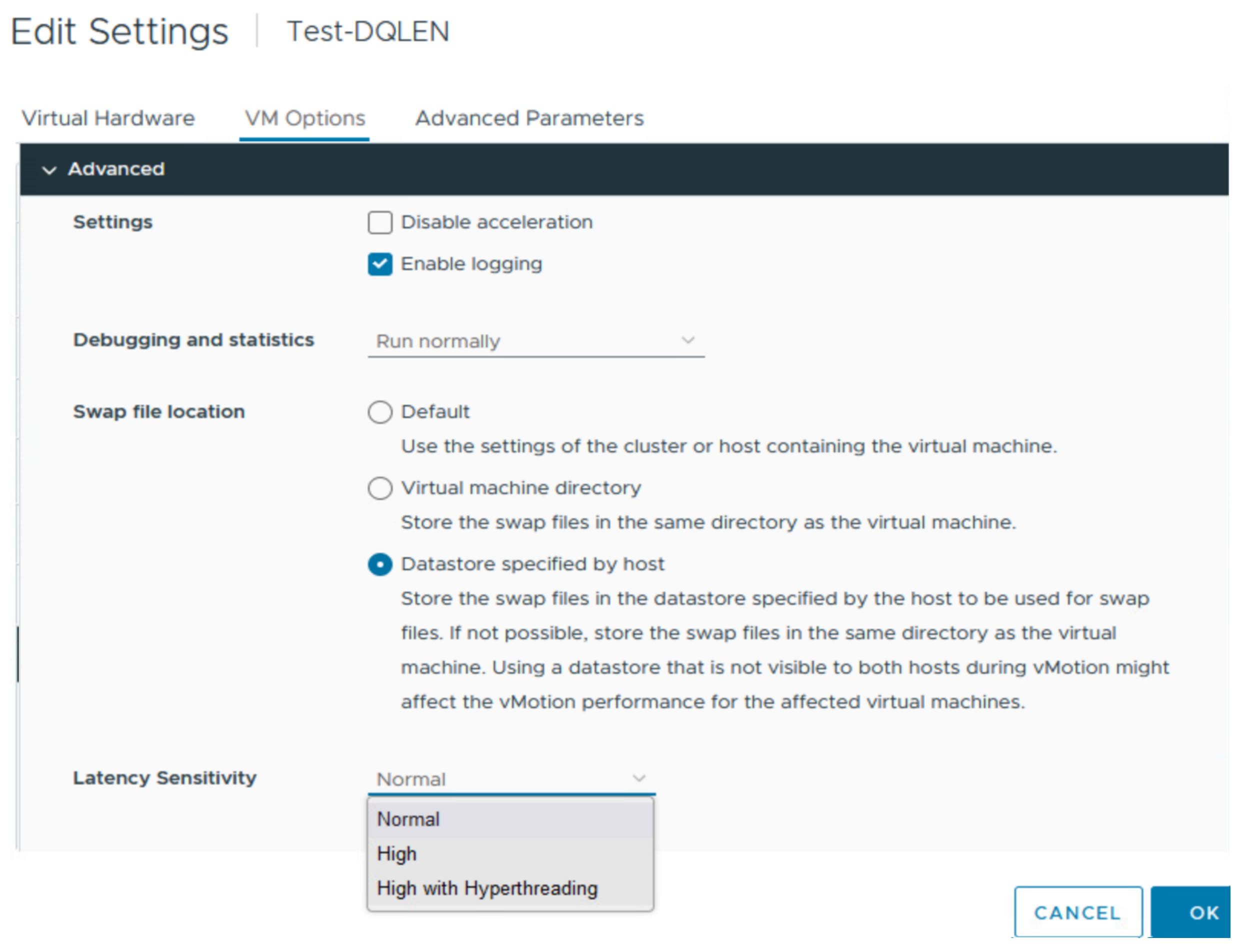
Hyper-threading[23]

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 up to 30 percent. This improvement, coupled with the reality that most workloads in the virtual environment (VMs) are highly unlikely to request and consume their full allocation of compute resources simultaneously on a regular basis is why it is possible (and supported) to over-allocate an ESXi Host’s physical compute resources by a factor of 2:1 in vSphere. Extensive testing and good monitoring tools are required when following this over-allocation 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 and takes 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.

VMware introduced virtual Hyperthreading (vHT) in vSphere 8.0 as an enhancement to the “Latency Sensitivity” setting which has long existed in vSphere. The location for controlling “Latency Sensitivity” has moved from the “Virtual Hardware” section of a VM’s Properties in vCenter to the “Advanced” section of the “VM Options” tab, as shown in the image below:

Figure 11. Setting Latency Sensitivity on a VM



Settings
☐ Disable acceleration
☑ Enable logging

Debugging and statistics: Run normally

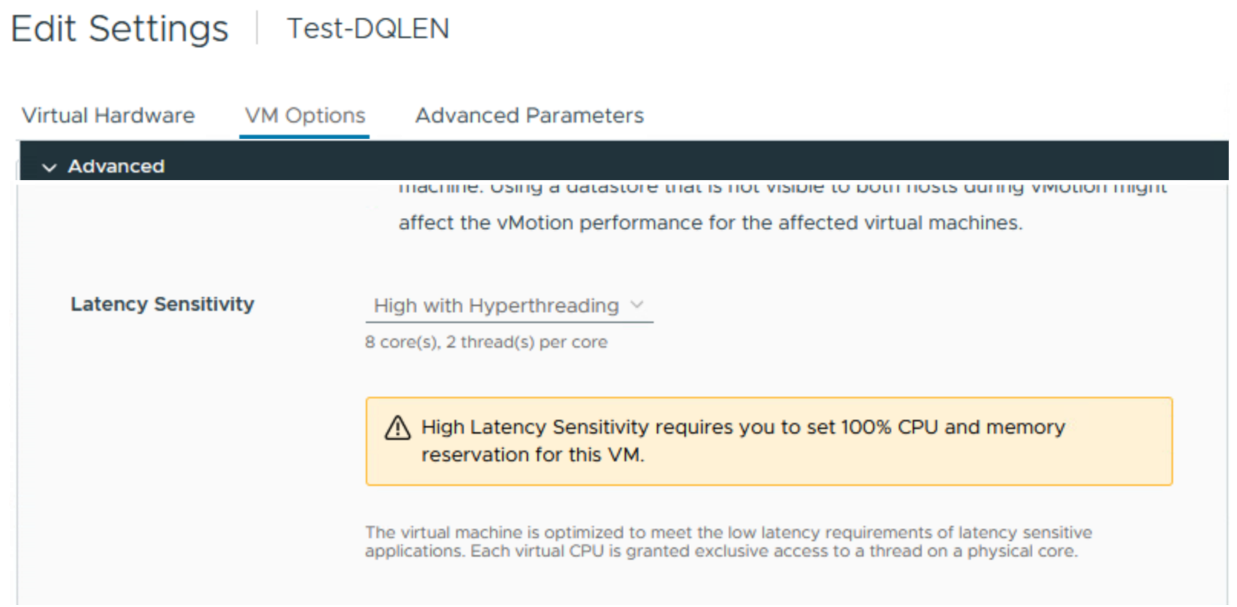
Swap file location
○ Default
Use the settings of the cluster or host containing the virtual machine.
○ Virtual machine directory
Store the swap files in the same directory as the virtual machine.
● Datastore specified by host
Store the swap files in the datastore specified by the host to be used for swap files. If not possible, store the swap files in the same directory as the virtual machine. Using a datastore that is not visible to both hosts during vMotion might affect the vMotion performance for the affected virtual machines.

Latency Sensitivity: Normal
- Normal
- High
- High with Hyperthreading

Readers would notice that a new option “High with Hyperthreading” is now available. Setting latency sensitivity to “High” for applications such as Microsoft SQL Server has traditionally led to substantial performance gain.

This gain increases more when the “High with Hyperthreading” option is selected. This option enables virtual HT for such HT-aware applications in a vSphere environment. The performance gains usually result from a combination of exclusive reservation of (and access to) physical CPUs when “Latency Sensitivity” is set to “High” on a VM, and the ability of the Guest Operating System and application to become aware of hyper-threads on allocated cores.

Figure 12. "High With Hyperthreading" Provides Exclusive CPU Access



affect the vMotion performance for the affected virtual machines.

Latency Sensitivity: High with Hyperthreading
8 core(s), 2 thread(s) per core

⚠ High Latency Sensitivity requires you to set 100% CPU and memory reservation for this VM.

The virtual machine is optimized to meet the low latency requirements of latency sensitive applications. Each virtual CPU is granted exclusive access to a thread on a physical core.

Without vHT activated on ESXi, each virtual CPU (vCPU) is equivalent to a single non-hyperthreaded core available to the guest operating system. With vHT activated, each guest vCPU is treated as a single hyperthread of a virtual core (vCore).

Virtual hyperthreads of the same vCore occupy the same physical core. As a result, vCPUs of the VM can share the same core as opposed to using multiple cores on VMs with latency sensitivity high that have vHT deactivated.

Because setting “Latency Sensitivity” to High on a VM causes the hypervisor to enable full resource reservations for that VM (therefore decreasing the amount of physical compute resources available to other VMs in the cluster), it is important to account for the differences between a processor thread (logical core) and a physical CPU/core during capacity planning for your SQL Server deployment. High latency sensitivity should be used sparingly, and only when other performance tuning options are shown to have been ineffective for the VM in question.

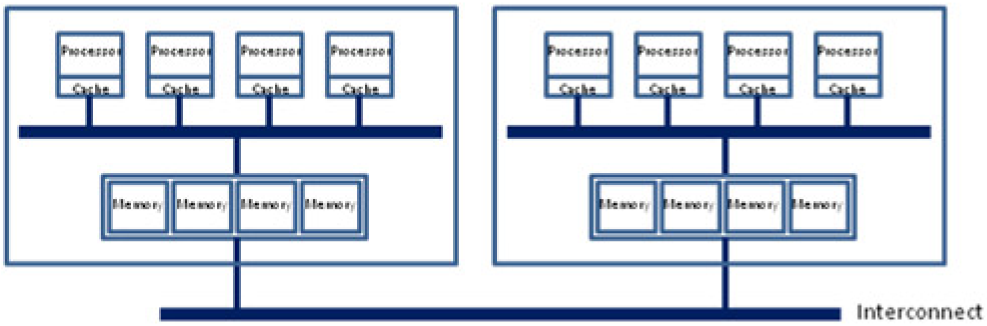
NUMA Consideration

Over the last decade, no topic has attracted so much attention as discussions about NUMA technologies and its implementation. This is expected, taking into account the complexity of the technology, particularly the variances in vendor implementations, number of configurations options and layers (from hardware through hypervisor to Guest OS and application). Without pretending to provide full overview of all available configuration options, we will concentrate on giving important guidelines which will help improve NUMA-specific performance metrics for NUMA-aware applications such as the Microsoft SQL Server instances virtualized on a vSphere platform.

Understanding NUMA[24]

Non-uniform memory access (NUMA) is a hardware architecture for shared memory implementing subdivision of physical memory banks between pCPUs (see Figure 12 for one of the possible implementations). NUMA-capable, symmetric multiprocessing (SMP) systems usually have more than one system bus with dense (large) number of processors supported by huge amounts of memory bandwidth to assist them in effectively and efficiently using their processing powers to their maximum capabilities. Without the large memory bandwidth, applications running on these dense Systems are constrained in performance and throughput. While this constraint can be somewhat mitigated by simply increasing the data bus bandwidth, this option is both expensive and limited in scalability.

Figure 13. Intel based NUMA Hardware Architecture[25]



The industry-standard approach to this problem is dividing the bus into smaller chunks (nodes) and grouping smaller number of processors and a corresponding slice of available memory into these nodes. This grouping provides highly efficient and high-performing connection between the processors and memory in the node. Applications accessing these processors perform much better and more optimally when the threads, instructions and processes executed by a processor is serviced by memory resident in the same node. The efficiency and performance improvement come mainly from the fact that, when the processor doesn’t have to cross the System’s interconnect to fetch memory to service a given instructions, the instructions are executed and completed more rapidly. Given that the NUMA interconnect itself can become a throttling point for applications with high memory bandwidth requirements, servicing execution instructions with local memories in a NUMA system is cheaper and better than doing so with memories from remote nodes.

This architecture has ultimate benefits, but also poses some trade-offs that needs to be considered. The most important of it - the time to access data in different memory cache lines varies - depending on local or remote placement of the corresponding memory cache line to a CPU core executing the request, with remote access being up to X[26] time 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.

It should be noted that, although the ESXi NUMA scheduler can automatically detect and present an optimal and efficient NUMA topology to a VM when that VM has more vCPUs allocated than is physically available in a single Socket (a condition frequently described as “wide VM”), vSphere Administrator are highly encouraged to manually review such presentations to ensure that it satisfies their workloads requirements and corporate systems administration practices.

This is also important because, even when enabled, NUMA does not automatically configure itself in a way that benefits the workloads or applications, especially in a virtual environment. Over the years, VMware has continued to improve its implementation of NUMA features in ESXi, and our guidance on how to appropriately configure memory and CPU allocations to VMs have changed periodically to account for these evolutions. We will go through different layers of NUMA implementation and provide recommended 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 that fall outside the scope of these general guidelines.

How ESXi NUMA Scheduling Works[27]

ESXi uses a sophisticated NUMA scheduler to dynamically balance processor load and memory locality or processor load balance.

- Each virtual machine managed by the NUMA scheduler is assigned a home node. A home node is one of the system’s NUMA nodes containing processors and local memory, as indicated by the System Resource Allocation Table (SRAT).
- When memory is allocated to a virtual machine, the ESXi host preferentially allocates it from the home node. The virtual CPUs of the virtual machine are constrained to run on the home node to maximize memory locality.
- The NUMA scheduler can dynamically change a virtual machine's home node to respond to changes in system load. The scheduler might migrate a virtual machine to a new home node to reduce processor load imbalance. Because this might cause more of its memory to be remote, the scheduler might migrate the virtual machine’s memory dynamically to its new home node to improve memory locality. The NUMA scheduler might also swap virtual machines between nodes when this improves overall memory locality.

Sub-NUMA Clustering (Cluster-on-Die)

We mentioned earlier that an ESXi Host's NUMA topology typically mirrors the physical CPU Socket and Memory topology. We can expect that a physical motherboard with 2 Sockets will generally contain 2 NUMA Nodes. This is largely true until we consider the increasing popularity of the not-so-new feature set in modern CPU technologies called "Cluster on Die" (CoD) or "Sub-NUMA Clustering" (SNC).

Both nomenclatures refer to the efforts by CPU vendors to squeeze more performance out of the system by further sub-dividing each CPU Socket into smaller units (or nodes). As you read this section, note that the ESXi scheduler creates NUMA clients for each of the NUMA nodes it surfaces to a VM. ESXi uses these clients internally to optimize performance for a VM and the ESXi CPU scheduler can (and does) move these clients around (for re-balancing) whenever it believes that doing so will improve performance for the VM in question. By default, the scheduler places a VM's NUMA clients as close to each other as possible, but, (especially) when there is resource contention on the Host, the scheduler might migrate the supporting NUMA clients behind the scene. While NUMA client migration is transparent to the administrator, it has some not-quite-insignificant detrimental effects on a VM's performance. The optimal state in a vSphere infrastructure is when there are as few NUMA client migration as possible.

As Server hardware and compute resource capacities become larger and denser, and as virtualized business critical workloads replace physical instances as standard deployment and configuration practices, application owners have had to choose between creating fewer large VMs (aka "Monster VMs") or creating more smaller ones. There are arguments to be made in support of either options (e.g. fewer VMs to manager equal operational efficiency while they create a larger single point-of-failure for the application they host, compared to spreading the application over multiple smaller instances, reducing the impact of a single system failure on the rest of the infrastructure).

Most Customers deploying SQL Server in a vSphere environment typically opt for larger, denser VMs for their workloads. Some do this for administrative efficiency purposes, and others do it for licensing considerations. Regardless of the reason for this choice of larger VMs, it is important for us to understand the impact of CoD/SNC on this configuration choice.

VMware reference architecture documents and best practices recommendations talk about the need to "right-size" a VM in a way that ensures that the compute resources allocated to the VM fits into as few NUMA nodes as possible. This is because applications and systems generally perform better from close CPU-to-Memory adjacency. When the memory servicing a particular CPU instruction is close to the CPU issuing the instruction, the instruction completes faster.

Administrators will typically use their knowledge of the physical hardware layout (and the information presented by ESXi) to deduce their Host's NUMA topology. By further dividing the "real" NUMA topology into smaller, CoD/SNC creates a NUMA topology which is different from (and smaller than) the physical topology. Because CoD/SNC topology is not visible in the vSphere client, administrators are unable to see this topology, so the NUMA topology presented to their VMs will be different from what is actually presented to the VM.

This configuration difference creates a situation in which a right sized VM will inherit a topology which results in sub-optimal performance for applications like SQL Server. Let's illustrate this with an example.

We are using a Dell PowerEdge R750 (Intel Xeon Gold 6338 CPU 2.00GHz) Server for our example. This host has 2 physical CPU packages, consisting of 2 CPU sockets, 32 cores in each socket and 512GB of memory. Generally, each of the socket is adjacent to half of the total available memory (256GB). For illustration purposes, this effectively creates 2 physical NUMA nodes, which is communicated up to the ESXi hypervisor.

As described, a VM with up to 32 virtual CPUs and 256GB RAM allocated will fit into a single NUMA node and can be expected to perform more efficiently that a VM with, say, 34 virtual CPUs with 312GB RAM.

When CoD/SNC is enabled on this ESXi host, each NUMA node is further divided into 2, creating a cluster of Four (4) NUMA nodes, each with 16 cores and 128GB memory.

In this example, a VM with up to 16 virtual CPUs and 128GB RAM allocated will fit into a single NUMA node and can be expected to perform optimally under normal operating conditions, chiefly because no remote memory fetching is occurring in this condition.

A VM with 20 virtual CPUs and 128GB RAM will be forced to fetch some of the memory required for its processes from a remote NUMA node because, with SNC/CoD enabled, the NUMA node now has only 16 vCPUs while the VM has 20.

While some applications will, indeed, benefit from CoD/SNC, VMware has not observed such performance benefits in large Microsoft SQL Server workloads on vSphere. VMware, therefore, strongly recommends that Customers conduct extensive testing and validation of their applications and business requirements when making a decision regarding the use of the “Cluster-on-Die” or “Sub-NUMA Clustering” feature on their ESXi Host.

Customers should particularly note that, under certain conditions, enabling SNC/CoD may result in the ESXi hypervisor presenting undesired virtual NUMA topologies to large VMs hosted on the ESXi Host.

If SNC/CoD is enabled, VMware recommends that customers should ensure that they allocate virtual CPUs to large VMs in multiples of the core size of the sub-NUMA node (16, 32, 48, in our example).

Understanding New vNUMA Options in vSphere 8[28]

With the release of vSphere 8.0, VMware made considerable improvements and changes to how the hypervisor presents NUMA topologies to the VM, and these changes, in turn, influence how the guest operating system in the virtual machine sees and consumes the vCPUs it is allocated.

ESXi has historically been able to expose NUMA topologies to a VM. When a VM is allocated more than eight virtual CPUs, virtual NUMA (vNUMA) is automatically enabled on that VM (this minimum threshold of eight vCPUs is administratively configurable), enabling the VM’s guest OS and applications to take advantage of the performance improvements provided by such topology awareness.

This feature has been further refined and enhanced in vSphere 8.0. Whereas in the past, vNUMA does not factor in the size of memory allocated to the VM or the number of virtual sockets and cores-per-socket in computing the vNUMA topology exposed to that VM, the new vNUMA algorithm and logic account for these configurations. By default, ESXi 8.0 now considers virtual memory allocation in auto-configuring vNUMA topologies, and the resulting presentation is influenced by a combination of Cores-per-Socket allocated and the need to provide optimal L3 cache size to support the topology.

Using NUMA

As mentioned in the previous section, using a NUMA architecture may provide positive influence on the performance of an application, if this application is NUMA-aware. SQL Server Enterprise edition has native NUMA support starting with the version SQL Server 2005 (and, with some limitations, this is also available on SQL Server 2000 SP3 as well)[29], that ultimately means that almost all recent deployments of modern SQL Server will benefit from a properly configured NUMA presentation[30].

With this in mind, let us now 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.

Physical Server Hardware

NUMA support is dependent on the CPU architecture and was introduced first by AMD Opteron series and then by Intel Nehalem processor family back to the year 2008. Nowadays, almost all server hardware currently available on the market today uses NUMA architecture, although (depending on the vendor and hardware) NUMA capabilities may enabled by default in the BIOS of a server. For this reason, Administrator are encouraged to verify that NUMA support is enabled in their ESXi Host’s hardware BIOS settings. Most of hardware vendors will call this setting as “Node interleaving” (HPE, Dell) or “Socket interleave” (IBM) and this setting should be set to “disabled” or “Non-uniform Memory access (NUMA)[31] to expose NUMA topology.

As rules of thumb, number of exposed NUMA nodes will be equal to the number of physical sockets for Intel processors[32] and will be two times for AMD processors. Check you server documentations for more details.

VMware ESXi Hypervisor Host

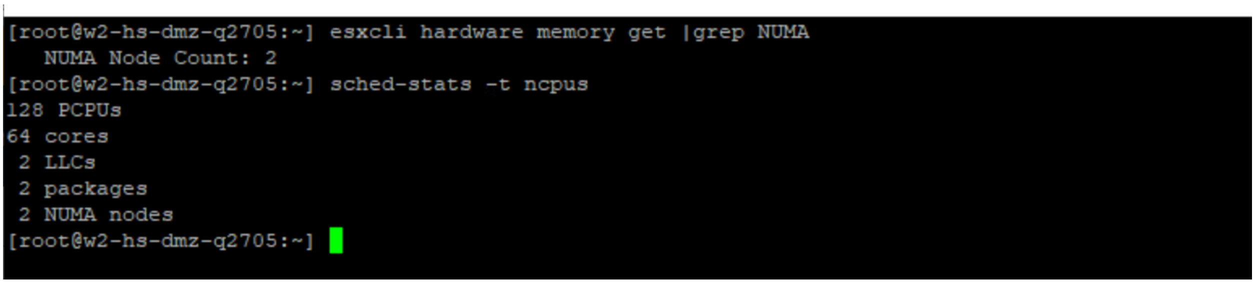
vSphere supports NUMA on the physical server starting with version 2. Moving to the current version (8.0 as of the time of writing this document), several configuration options were introduced to help manage NUMA topology, especially for “Wide” VMs. As our ultimate goal is to provide clear guidelines on how a 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 sched-stats* to obtain this information:

esxcli hardware memory get | grep NUMA

sched-stats -t ncpus

Figure 14. Using esxcli or sched-stats to Obtain NUMA Node Count on ESXi Host

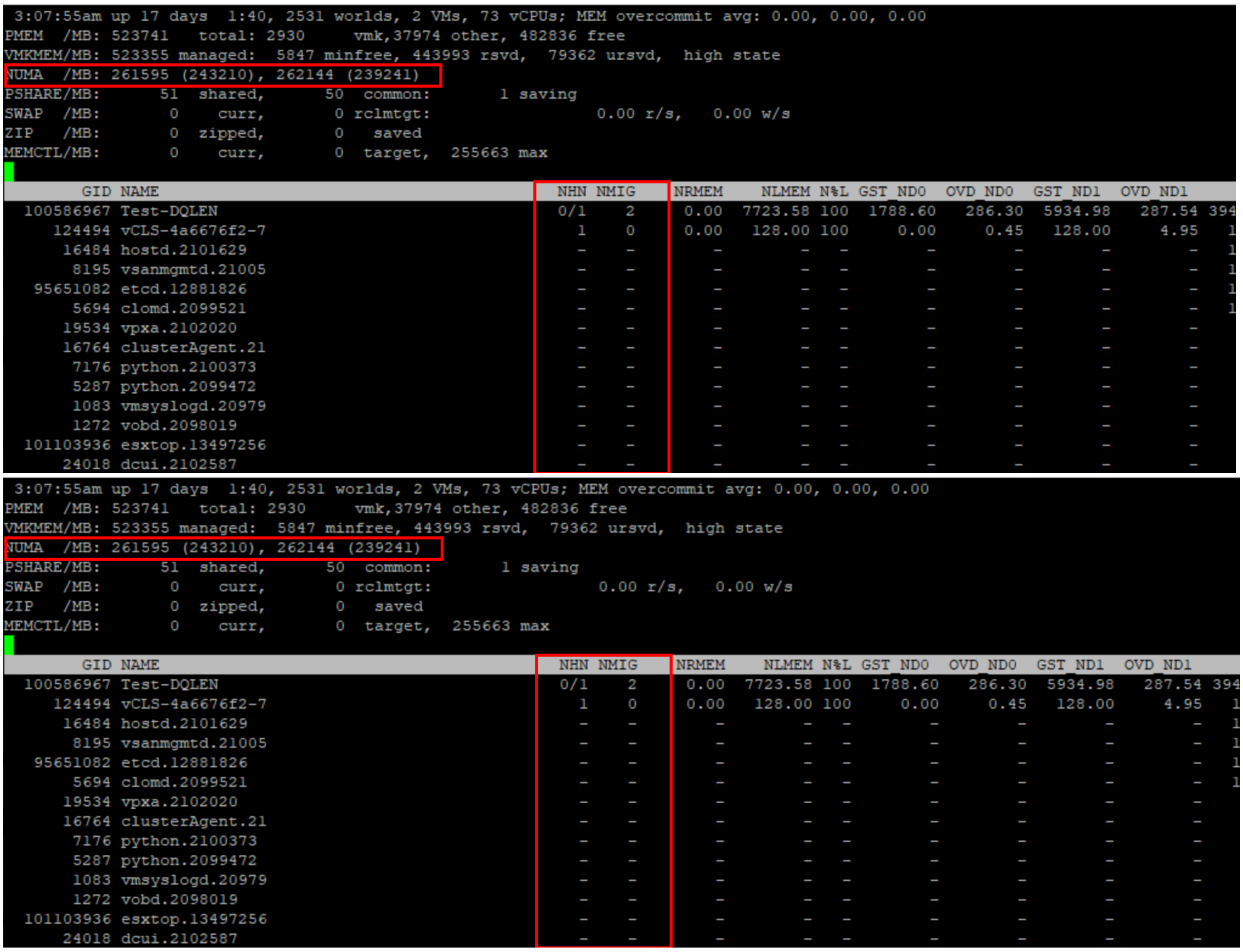


```
[root@w2-hs-dmz-q2705:~] esxcli hardware memory get |grep NUMA
   NUMA Node Count: 2
[root@w2-hs-dmz-q2705:~] sched-stats -t ncpus
128 PCPUs
64 cores
 2 LLCs
 2 packages
 2 NUMA nodes
[root@w2-hs-dmz-q2705:~]
```

or

ESXTOP, press M for memory, F to adjust fields, G to enable NUMA stats[33]*,*

Figure 15. Using esxtop to Obtain NUMA Related Information on an ESXi Host



```
 3:07:55am up 17 days  1:40, 2531 worlds, 2 VMs, 73 vCPUs; MEM overcommit avg: 0.00, 0.00, 0.00
PMEM  /MB: 523741   total: 2930     vmk,37974 other, 482836 free
VMKMEM/MB: 523355 managed:  5847 minfree, 443993 rsvd,  79362 ursvd,  high state
NUMA  /MB: 261595 (243210), 262144 (239241)
PSHARE/MB:      51  shared,      50  common:       1 saving
SWAP  /MB:       0    curr,       0 rclmtgt:                0.00 r/s,   0.00 w/s
ZIP   /MB:       0  zipped,       0   saved
MEMCTL/MB:       0    curr,       0  target,  255663 max

       GID NAME                                            NHN NMIG NRMEM    NLMEM N%L GST_ND0  OVD_ND0  GST_ND1  OVD_ND1
  100586967 Test-DQLEN                                     0/1    2  0.00  7723.58 100  1788.60   286.30  5934.98   287.54 394
     124494 vCLS-4a6676f2-7                                  1    0  0.00   128.00 100     0.00     0.45   128.00     4.95   1
      16484 hostd.2101629                                    -    -     -        -   -        -        -        -        -   1
       8195 vsanmgmtd.21005                                  -    -     -        -   -        -        -        -        -   1
   95651082 etcd.12881826                                    -    -     -        -   -        -        -        -        -   1
       5694 clomd.2099521                                    -    -     -        -   -        -        -        -        -   1
      19534 vpxa.2102020                                     -    -     -        -   -        -        -        -        -
      16764 clusterAgent.21                                  -    -     -        -   -        -        -        -        -
       7176 python.2100373                                   -    -     -        -   -        -        -        -        -
       5287 python.2099472                                   -    -     -        -   -        -        -        -        -
       1083 vmsyslogd.20979                                  -    -     -        -   -        -        -        -        -
       1272 vobd.2098019                                     -    -     -        -   -        -        -        -        -
  101103936 esxtop.13497256                                  -    -     -        -   -        -        -        -        -
      24018 dcui.2102587                                     -    -     -        -   -        -        -        -        -
```

```
 3:07:55am up 17 days  1:40, 2531 worlds, 2 VMs, 73 vCPUs; MEM overcommit avg: 0.00, 0.00, 0.00
PMEM  /MB: 523741   total: 2930     vmk,37974 other, 482836 free
VMKMEM/MB: 523355 managed:  5847 minfree, 443993 rsvd,  79362 ursvd,  high state
NUMA  /MB: 261595 (243210), 262144 (239241)
PSHARE/MB:      51  shared,      50  common:       1 saving
SWAP  /MB:       0    curr,       0 rclmtgt:                0.00 r/s,   0.00 w/s
ZIP   /MB:       0  zipped,       0   saved
MEMCTL/MB:       0    curr,       0  target,  255663 max

       GID NAME                                            NHN NMIG NRMEM    NLMEM N%L GST_ND0  OVD_ND0  GST_ND1  OVD_ND1
  100586967 Test-DQLEN                                     0/1    2  0.00  7723.58 100  1788.60   286.30  5934.98   287.54 394
     124494 vCLS-4a6676f2-7                                  1    0  0.00   128.00 100     0.00     0.45   128.00     4.95   1
      16484 hostd.2101629                                    -    -     -        -   -        -        -        -        -   1
       8195 vsanmgmtd.21005                                  -    -     -        -   -        -        -        -        -   1
   95651082 etcd.12881826                                    -    -     -        -   -        -        -        -        -   1
       5694 clomd.2099521                                    -    -     -        -   -        -        -        -        -   1
      19534 vpxa.2102020                                     -    -     -        -   -        -        -        -        -
      16764 clusterAgent.21                                  -    -     -        -   -        -        -        -        -
       7176 python.2100373                                   -    -     -        -   -        -        -        -        -
       5287 python.2099472                                   -    -     -        -   -        -        -        -        -
       1083 vmsyslogd.20979                                  -    -     -        -   -        -        -        -        -
       1272 vobd.2098019                                     -    -     -        -   -        -        -        -        -
  101103936 esxtop.13497256                                  -    -     -        -   -        -        -        -        -
      24018 dcui.2102587                                     -    -     -        -   -        -        -        -        -
```

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[34].

If 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 Virtual Machine starting with vSphere version 5.0 and above. 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 the vNUMA topology will be created and what settings can influence it.

Please bear in mind that one of the impressive NUMA/vNUMA features introduced in vSphere 8.0 is the availability of a GUI option for administrators to more granularly define the NUMA/vNUMA presentations for a VM. We encourage readers to explore this feature to determine its applicability and suitability for their desired end state and performance needs.

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

General Rules (applies to all currently supported versions of vSphere):

a. The minimum vCPU threshold at which ESXi exposes vNUMA to a VM is nine. Once a VM has more than eight vCPUs, virtual NUMA will be exposed to that VM.
    i. **NOTE:** This threshold is administratively configurable. By setting the advanced VM configuration parameter value of “numa.vcpu.min” to whatever is desired on the advanced configuration on a VM, an Administrator instructs ESXi to expose virtual NUMA to that VM once its number of allocated vCPUs exceeds this value.
b. The first time a virtual NUMA activated virtual machine is powered on, its virtual NUMA topology is based on the NUMA topology of the underlying physical host. Once a virtual machines virtual NUMA topology is initialized, it does not change unless the number of vCPUs in that virtual machine is changed.
    i. **NOTE:** This behavior is very important to keep in mind when using the Cluster-level Enhanced vMotion Compatibility (EVC)[35] feature to group ESXi Hosts with dissimilar physical NUMA topologies together in a single vSphere Cluster.
    ii. **Because VM’s vNUMA topology is not re-**evaluated after power-on, migrating a running VM from one Host to another Host with different NUMA topology does not cause the exposed topology to change on the VM. This can lead to NUMA imbalance and performance degradation until the VM is restarted on its new ESXi Host.
    iii. EVC is also available as a VM-level configuration attribute. Configuring EVC at VM-level overrides Cluster-level EVC settings. A VM which has VM-level EVC set cannot inherit the EVC mode of its new Host, even after a reboot. VM-level EVC requires a manual reconfiguration to become compatible with its new Host’s EVC mode.
c. Although NUMA/vNUMA is a function of a combination of CPU/vCPU and Memory, vSphere does not consider a VM’s allocated memory nor the number of cores-per-socket or virtual sockets when exposing vNUMA to a VM.
    i. NOTE: If a VM’s allocated vCPUs and Memory can fit into one physical NUMA node, ESXi does not expose vNUMA to the VM.
    ii. If the allocated vCPUs can fit into one physical socket, but allocated Memory exceeds what’s available in that Socket, the ESXi scheduler will create as many scheduling topologies as required to accommodate the non-local memory. This auto-created construct will not be exposed to the Guest OS.
    iii. The implication of the foregoing includes a situation where applications inside the VM/Guest OS is forced to rely on its instructions and processes being serviced by remote memory across NUMA boundaries, leading to severe performance degradation.
d. The issue described in the previous section is now effectively addressed in vSphere 8.0, with the introduction of the virtual NUMA topology definition GUI.
e. Traditionally, vNUMA is not exposed if the “CPU hot add” feature is enabled on a VM. This behavior has also changed in vSphere 8.0 (see the “CPU Hot Plug” Section below)
f. VM virtual hardware version 8 is required to have vNUMA exposed to the Guest OS. However, the new enhancements in vSphere 8.0 described previously and in other parts of this document are only available to a VM only if its virtual hardware version is 20 and above.
g. vNUMA topology will be updated if changes are made to the CPU configuration of a VM. pNUMA information from the host, where the VM that was started at the time of the change will be used for creating vNUMA topology.

vSphere Version 6.5 and Above

VMware introduced the automatic vNUMA presentation in vSphere 6.5. As previously mentioned, this feature did not factor in the “Cores per Socket” setting while creating the vNUMA topology on an eligible VM. The final vNUMA topology for a VM is computed using the number of physical cores per CPU package of the physical host where VM is about to start. The total number of vCPUs assigned to a VM will be consolidated in the minimal possible number of proximity domains (PPD), equal in size of CPU package. In most use cases (and for most workloads), using auto-sizing will create more optimized configuration compared to the previous approach.

In vSphere 8.0, VMware vSphere administrators are now able to manually configure the desired NUMA topologies for their VM and its Guest OS and applications in an intuitive way and override the auto NUMA configurations determined by ESXi. We caution that administrators should work closely with their SQL Server administrators and other stakeholders in understanding their application's usage and specific requirements to evaluate and determine the impacts of the recommendations prescribed in the section that follows below on their specific usage situations.

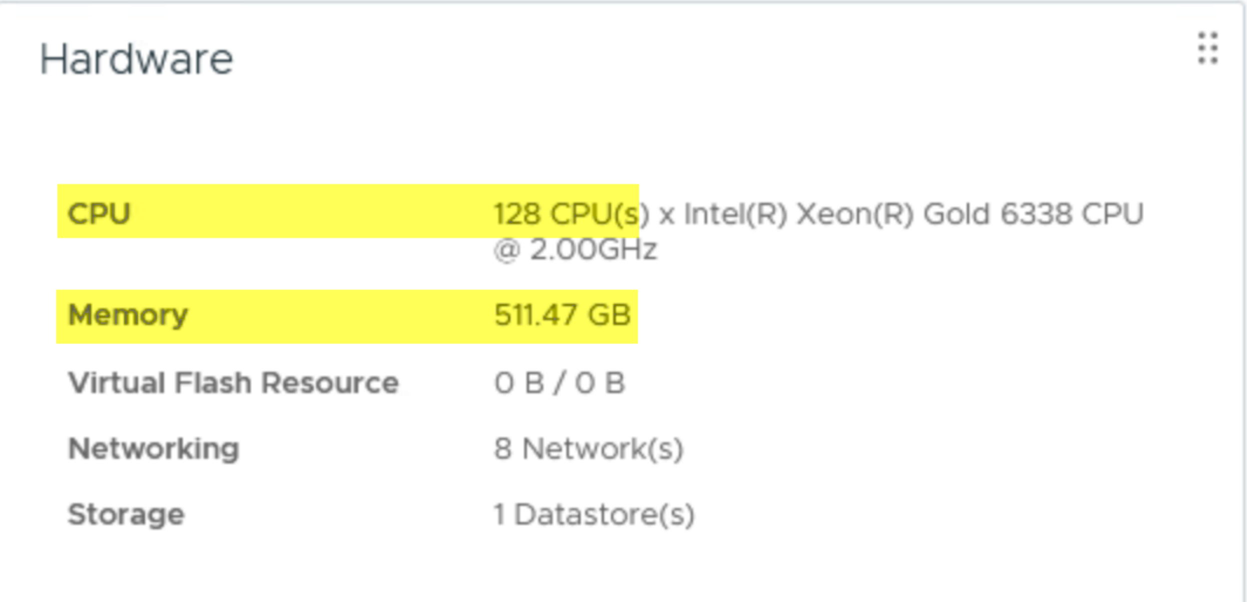
Configuring virtual NUMA for "Wide" Microsoft SQL Server VMs

vSphere 8.0 changes the standard CPU topology presentation. We have discussed much of these changes in other part of this document. This session presents a recommendation for improving NUMA-based performance metrics for a virtualized SQL Server instance in a vSphere environment.

When a VM has more one or more compute resources (Memory and/or CPU) allocated than is available in an ESXi Host's physical NUMA, it is a good practice for administrators to consider manually adjusting this presentation in a way that closely mirrors the Host's physical NUMA topology. It is important to remember that a VM is considered "Wide" as long as its allocated memory OR vCPUs cannot fit into a single physical NUMA node. For clarity, we shall now proceed to illustrate this with an example.

When is my considered "Wide"?

Figure 16. When is My Considered "Wide"?



Hardware

CPU	128 CPU(s) x Intel(R) Xeon(R) Gold 6338 CPU @ 2.00GHz
Memory	511.47 GB
Virtual Flash Resource	0 B / 0 B
Networking	8 Network(s)
Storage	1 Datastore(s)

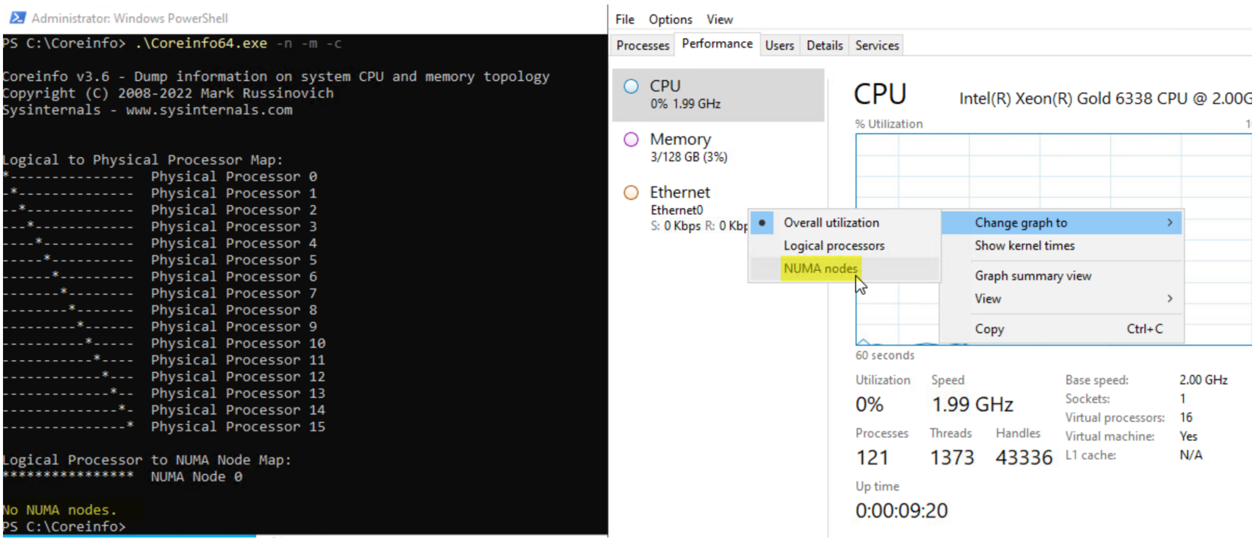
Our ESXi host has 128 logical CPUs (2 Sockets, each with 32 cores, and hyperthreading is enabled).

It also has 512GB of Memory.

Example 1

If we create a non-wide VM with 16 vCPUs and 128GB RAM and power it on, we see that ESXi has determined that, even though we have exceeded the minimum vCPU threshold beyond which vNUMA is automatically exposed to a VM, this is not a "Wide" VM because all of the compute resources allocated can fit into a single NUMA node. As a result, ESXi presents UMA to the VM, as seen below.

Figure 17. ESXi Creates UMA Topology When VM's vCPUs Fits a NUMA Node



Administrator: Windows PowerShell

```
PS C:\Coreinfo> .\Coreinfo64.exe -n -m -c

Coreinfo v3.6 - Dump information on system CPU and memory topology
Copyright (C) 2008-2022 Mark Russinovich
Sysinternals - www.sysinternals.com


Logical to Physical Processor Map:
*---------------  Physical Processor 0
-*--------------  Physical Processor 1
--*-------------  Physical Processor 2
---*------------  Physical Processor 3
----*-----------  Physical Processor 4
-----*----------  Physical Processor 5
------*---------  Physical Processor 6
-------*--------  Physical Processor 7
--------*-------  Physical Processor 8
---------*------  Physical Processor 9
----------*-----  Physical Processor 10
-----------*----  Physical Processor 11
------------*---  Physical Processor 12
-------------*--  Physical Processor 13
--------------*-  Physical Processor 14
---------------*  Physical Processor 15

Logical Processor to NUMA Node Map:
****************  NUMA Node 0

No NUMA nodes.
PS C:\Coreinfo>
```

CPU Intel(R) Xeon(R) Gold 6338 CPU @ 2.00G

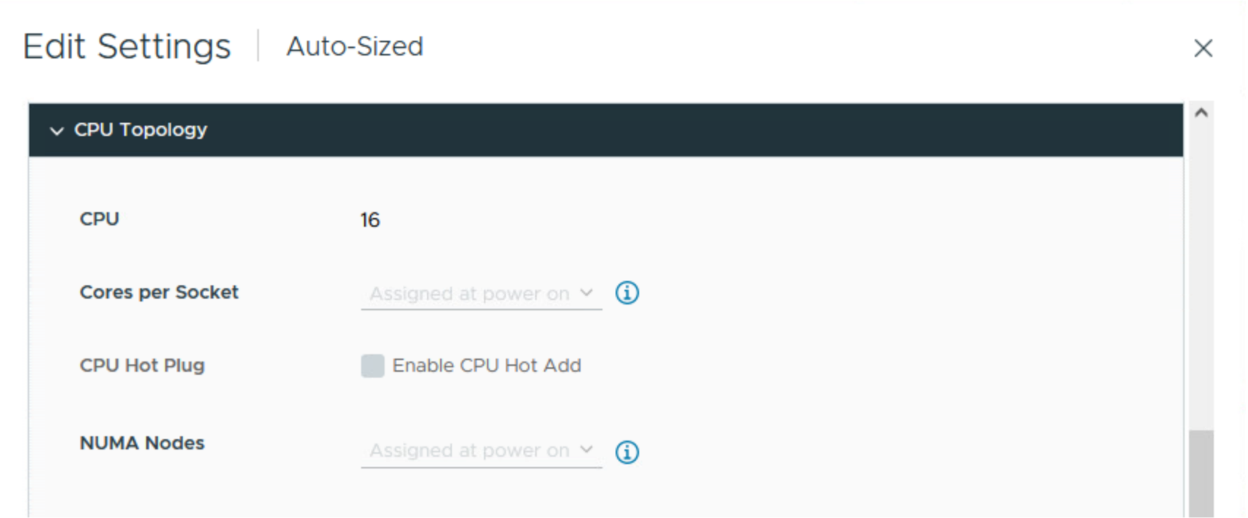
% Utilization

Utilization	Speed		Base speed:	2.00 GHz
0%	1.99 GHz		Sockets:	1
			Virtual processors:	16
Processes	Threads	Handles	Virtual machine:	Yes
121	1373	43336	L1 cache:	N/A

Up time
0:00:09:20

We have also chosen to NOT manually change this because there is no technological benefit to doing so in this case.

Figure 18. Letting ESXi Determine vNUMA Topology Automatically for Non-Wide VM



Edit Settings | Auto-Sized

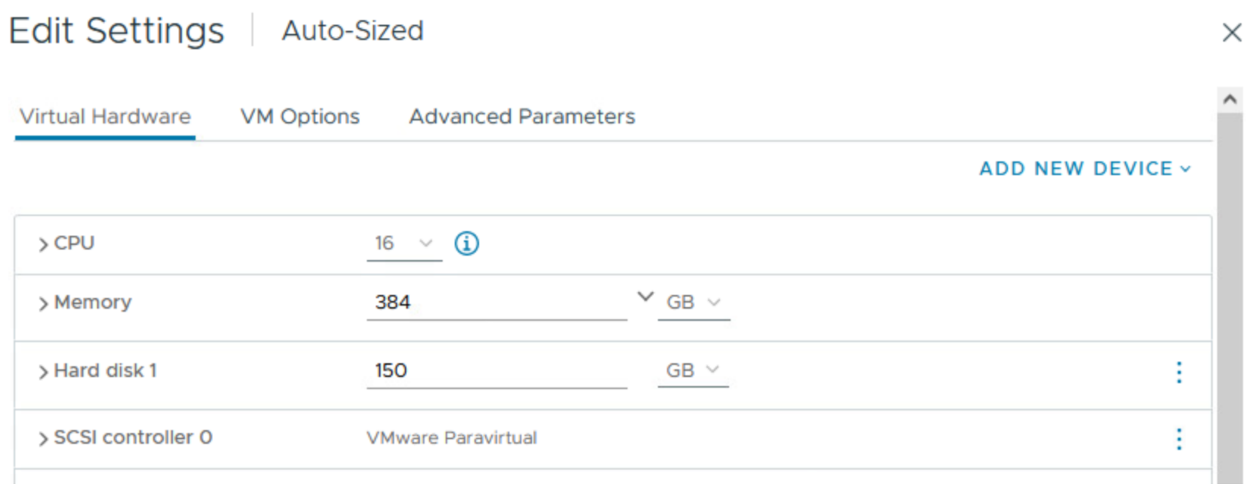
CPU Topology

CPU	16
Cores per Socket	Assigned at power on
CPU Hot Plug	□ Enable CPU Hot Add
NUMA Nodes	Assigned at power on

Example 2

If we now adjust the memory size of the VM to (say) 384GB and leave the vCPU unchanged (16 vCPUs), ESXi will still present a UMA configuration, but with a slight problem – because the VM's allocated memory has exceeded the capacity of the available memory in the physical NUMA node (256GB), the "excess" VM memory will be allocated from another physical node.

Figure 19. Letting ESXi Determine vNUMA Topology Automatically for Memory-wide VM

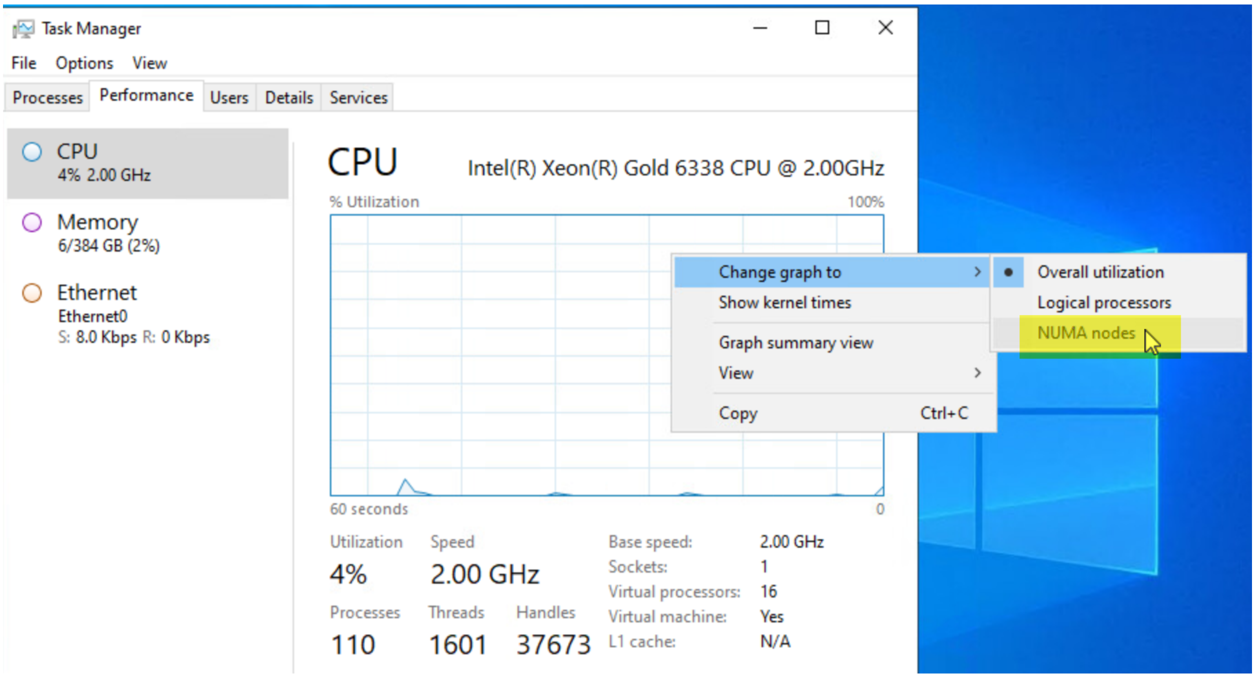


Edit Settings | Auto-Sized

CPU	16	
Memory	384	GB
Hard disk 1	150	GB
SCSI controller 0	VMware Paravirtual	

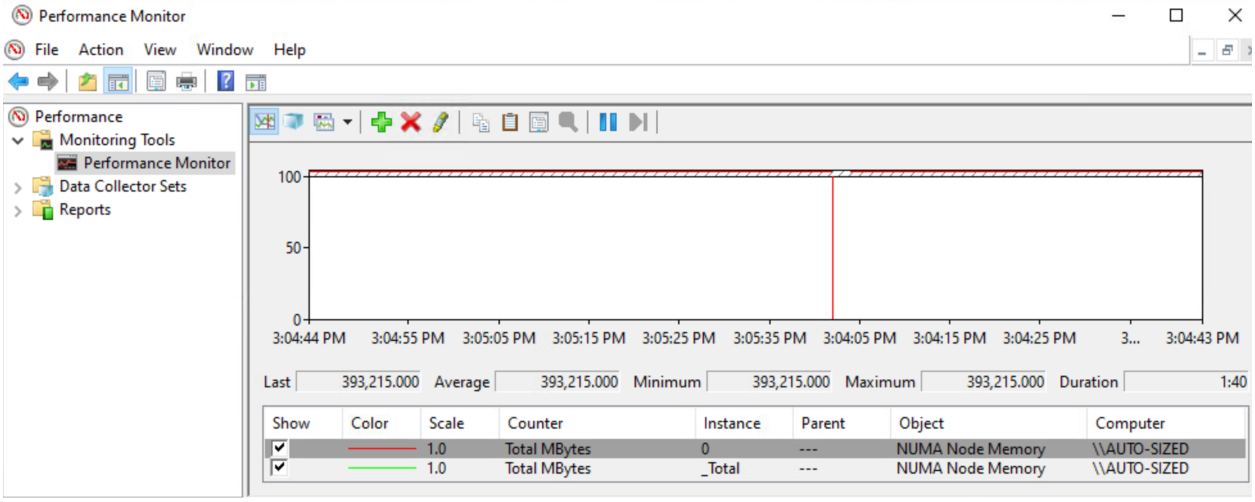
ESXi present an UMA topology to the Guest OS, as shown in the images below:

Figure 20. ESXi Doesn't Consider Memory Size in vNUMA Configuration



Utilization 4%
Speed 2.00 GHz
Processes 110
Threads 1601
Handles 37673
Base speed: 2.00 GHz
Sockets: 1
Virtual processors: 16
Virtual machine: Yes
L1 cache: N/A

Figure 21. Memory-wide VM, As Seen by Windows

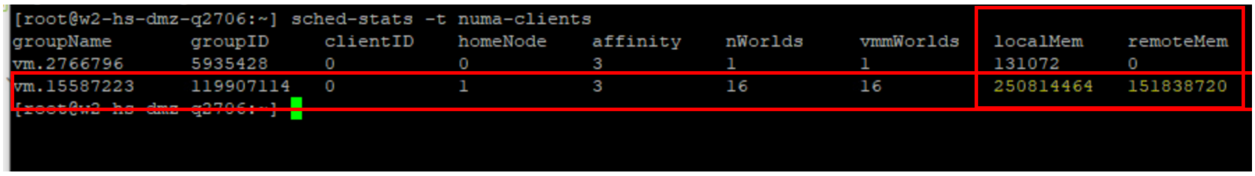


Last 393,215.000 Average 393,215.000 Minimum 393,215.000 Maximum 393,215.000 Duration 1:40

Show	Color	Scale	Counter	Instance	Parent	Object	Computer
☑	🟥	1.0	Total MBytes	0	---	NUMA Node Memory	\\AUTO-SIZED
☑	🟩	1.0	Total MBytes	_Total	---	NUMA Node Memory	\\AUTO-SIZED

In this configuration, Windows is led to believe that it has 16 CPUs and 384GB RAM, all of which are in one NUMA node. When we check with ESXi itself, we see that this is neither true nor accurate. This creates a situation where memory requests from some of the 16 vCPUs will be serviced by the remote memory, as shown in the image below:

Figure 22. Automatic vNUMA Selection Creates Unbalanced Topology on Memory-wide VM



```
[root@w2-hs-dmz-q2706:~] sched-stats -t numa-clients
groupName     groupID     clientID    homeNode    affinity    nWorlds     vmmWorlds   localMem    remoteMem
vm.2766796    5935428     0           0           3           1           1           131072      0
vm.15587223   119907114   0           1           3           16          16          250814464   151838720
[root@w2-hs-dmz-q2706:~]
```

NOTE:

- LocalMem is the amount of the VM’s allocated memory local to the allocated vCPUs (~256GB).
- RemoteMem is the amount of the VM’s allocated memory located in another node.

This is a situation which calls for manual administrative intervention, since we do not want our SQL Server’s queries and processes to be impacted by the latencies associated with vNUMA imbalance.

Example 3

In this example, we will demonstrate how to quickly correct the situation we described in the last example and mitigate the effects of such an imbalance.

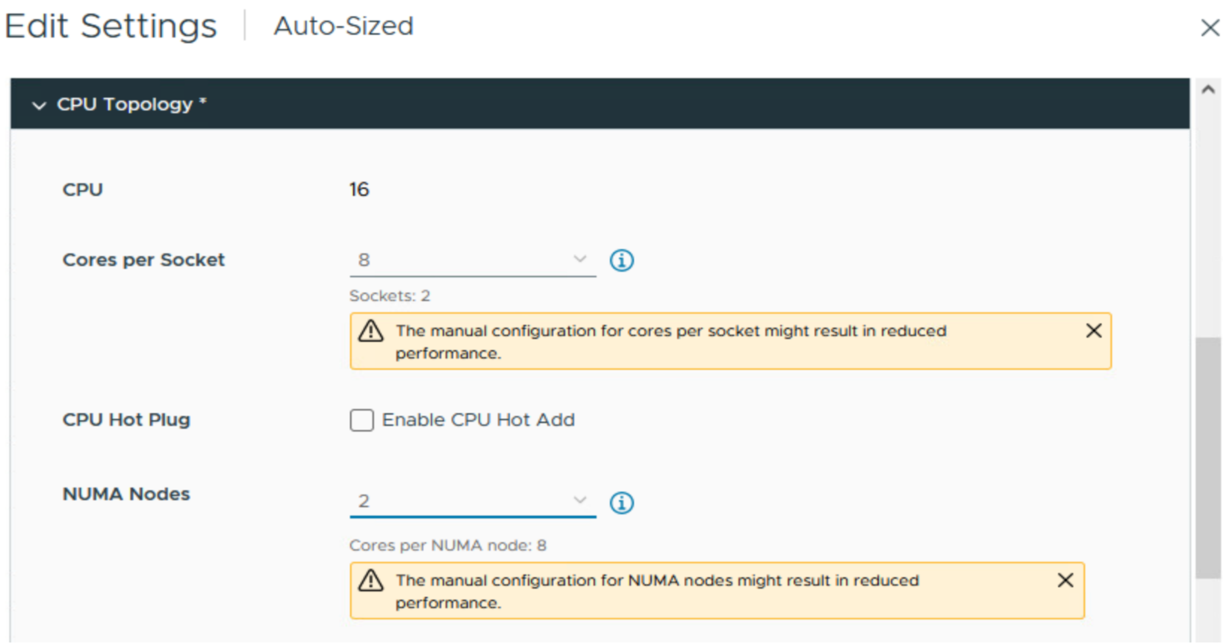
The major issue in the previous example is that we needed more memory than vCPUs for our SQL Server instance. This could be due to specific business requirements or other needs, so we will not attempt to rationalize the requirement. Our goal is to provide a properly configured VM in vSphere to optimally support the business or operational needs of our SQL Server instance.

We leave the configuration at 16 vCPUs and 384GB RAM and go over to the “VM Options” tab on the VM’s property, navigate to the “CPU Topology” section and adjust HOW we want ESXi to present the topology that reflects our desired state configuration.

We power off the VM and split the vCPUs into two Sockets, with eight cores in each socket.

We also explicitly configure two NUMA Nodes to account for the large allocated memory footprint.

Figure 23. Manual vNUMA Configuration Options in vSphere 8.0



Edit Settings | Auto-Sized

CPU Topology *

CPU: 16

Cores per Socket: 8

Sockets: 2

The manual configuration for cores per socket might result in reduced performance.

CPU Hot Plug: □ Enable CPU Hot Add

NUMA Nodes: 2

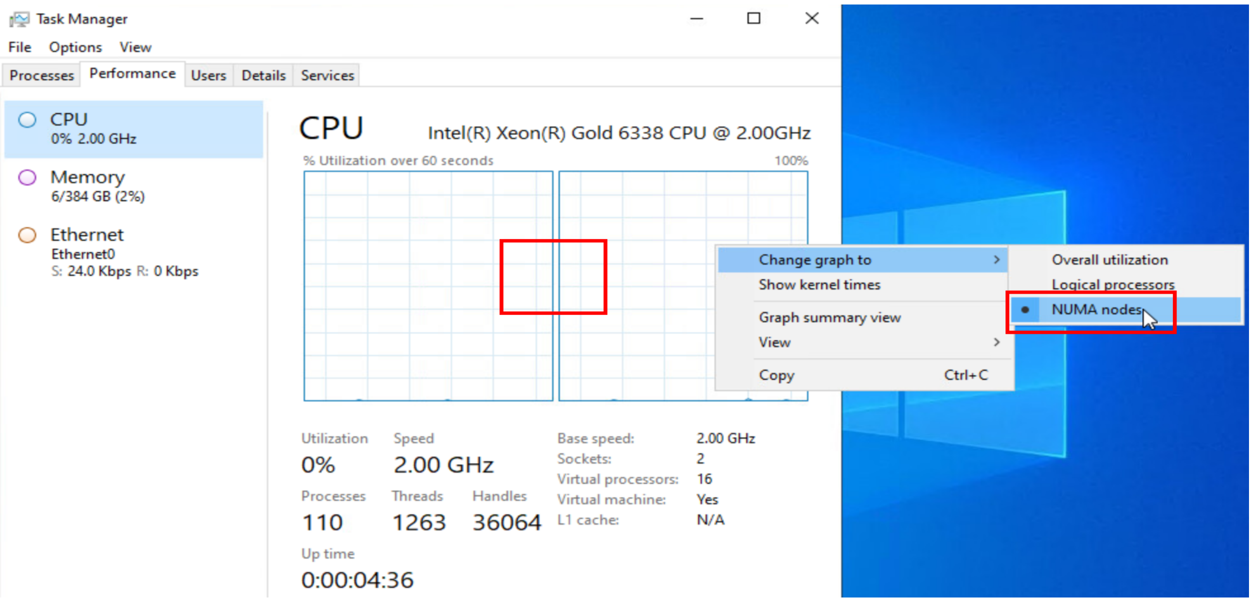
Cores per NUMA node: 8

The manual configuration for NUMA nodes might result in reduced performance.

After power on, we can immediately notice the effect of our configuration changes:

- We are now able to see TWO virtual NUMA nodes in the Guest Operating System.

Figure 24. Manual vNUMA Configuration, As Seen in Windows



Task Manager

File Options View

Processes | Performance | Users | Details | Services

CPU 0% 2.00 GHz

Memory 6/384 GB (2%)

Ethernet Ethernet0 S: 24.0 Kbps R: 0 Kbps

CPU Intel(R) Xeon(R) Gold 6338 CPU @ 2.00GHz

% Utilization over 60 seconds 100%

Change graph to > Overall utilization / Logical processors / ● NUMA nodes

Show kernel times

Graph summary view

View >

Copy Ctrl+C

Utilization 0% Speed 2.00 GHz

Processes 110 Threads 1263 Handles 36064

Up time 0:00:04:36

Base speed: 2.00 GHz

Sockets: 2

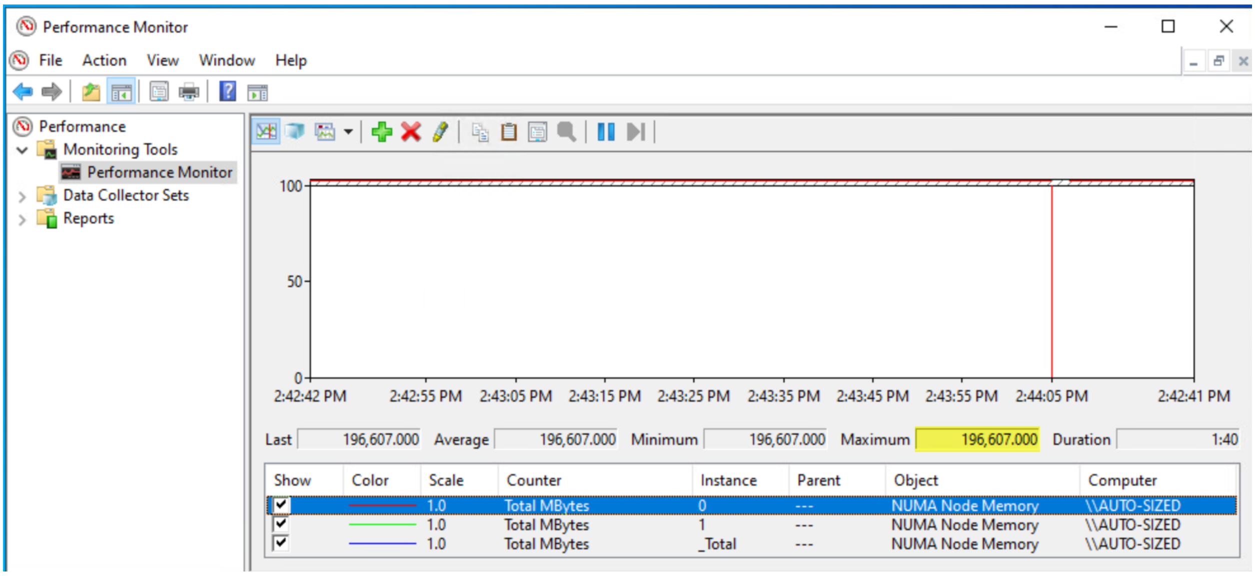
Virtual processors: 16

Virtual machine: Yes

L1 cache: N/A

- In Perfmon, we can see that half of the allocated RAM has been allocated to each of the NUMA nodes.

Figure 25. Manual vNUMA Topology Presents Balanced Topology

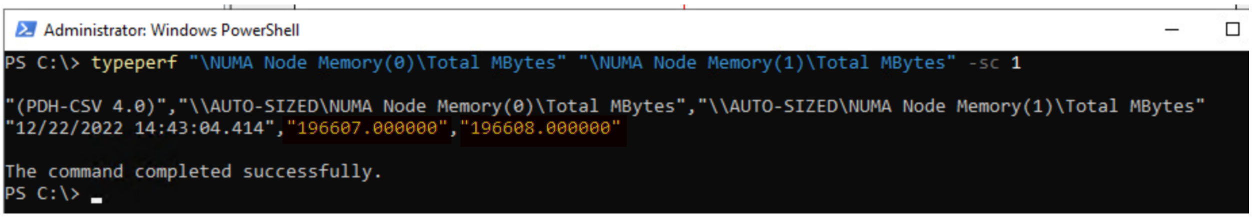


Last 196,607.000 Average 196,607.000 Minimum 196,607.000 Maximum 196,607.000 Duration 1:40

Show	Color	Scale	Counter	Instance	Parent	Object	Computer
☑	🟥	1.0	Total MBytes	0	---	NUMA Node Memory	\\AUTO-SIZED
☑	🟩	1.0	Total MBytes	1	---	NUMA Node Memory	\\AUTO-SIZED
☑	🟦	1.0	Total MBytes	_Total	---	NUMA Node Memory	\\AUTO-SIZED

- This even split of the allocated RAM into multiple nodes is more clearly illustrated when we query Perfmon from the command line.

Figure 26. Allocated Memory Evenly Distributed Across vNUMA Nodes



```
PS C:\> typeperf "\NUMA Node Memory(0)\Total MBytes" "\NUMA Node Memory(1)\Total MBytes" -sc 1

"(PDH-CSV 4.0)","\\AUTO-SIZED\NUMA Node Memory(0)\Total MBytes","\\AUTO-SIZED\NUMA Node Memory(1)\Total MBytes"
"12/22/2022 14:43:04.414","196607.000000","196608.000000"

The command completed successfully.
PS C:\> _
```

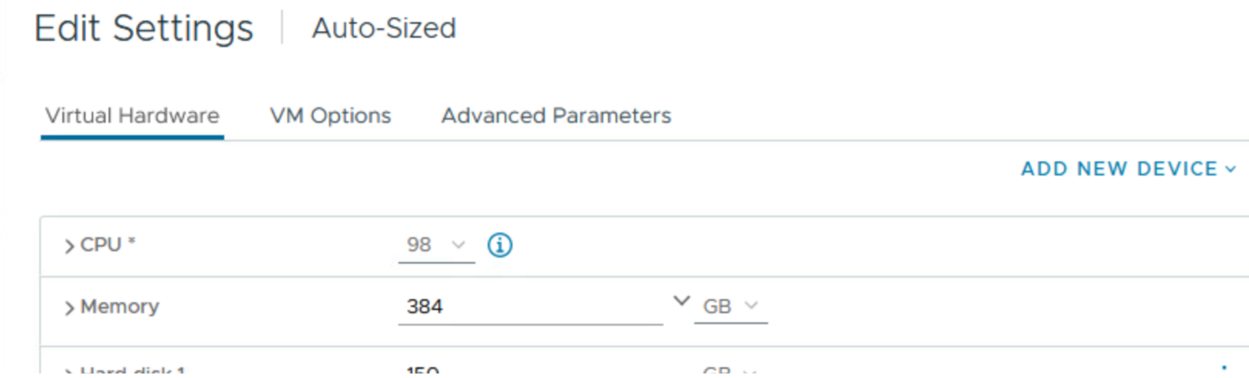
This is a much better presentation which substantially improves performance for our SQL Server instance.

Example 4

In this example, we will use the knowledge of our performance boost and symmetric topology to create a much large VM (aka “Monster VM”) with larger vCPU footprint.

Bearing in mind that our ESXi has 128 logical processors (2x32+hyperthreads), the rough math says that there are 64 logical processors in each physical NUMA node. We’re going to allocate more than 64 vCPUs to our VM this time, leaving the memory at 384GB so that both compute resources exceed what is physically available in the Host’s NUMA node.

Figure 27. Automatic vNUMA Topology for Wide VM



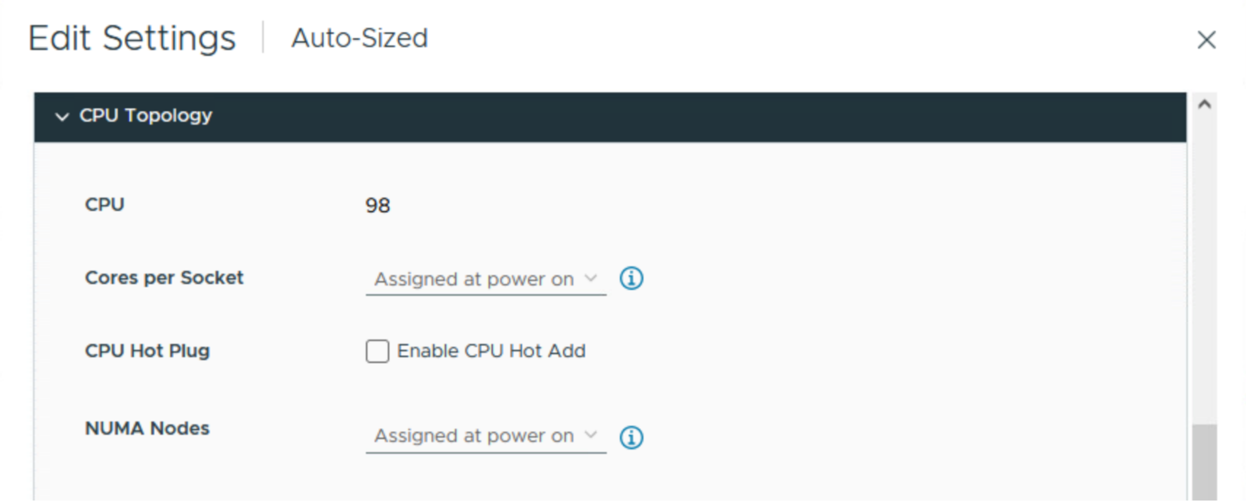
> CPU * 98

> Memory 384 GB

> Hard disk 1 150 GB

We will also accept ESXi’s default behavior and let it assign whichever topology it deems optimal for this configuration.

Figure 28. Configuring Automatic vNUMA Presentation for Wide VM



Edit Settings | Auto-Sized

CPU Topology

CPU: 98

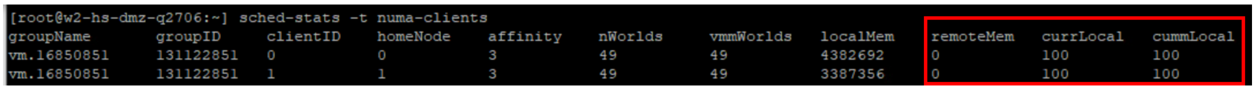
Cores per Socket: Assigned at power on

CPU Hot Plug: □ Enable CPU Hot Add

NUMA Nodes: Assigned at power on

In this configuration, and without any manual administrative intervention, ESXi tells us that everything is properly configured, as all memory allocated to each NUMA node is completely local to the node.

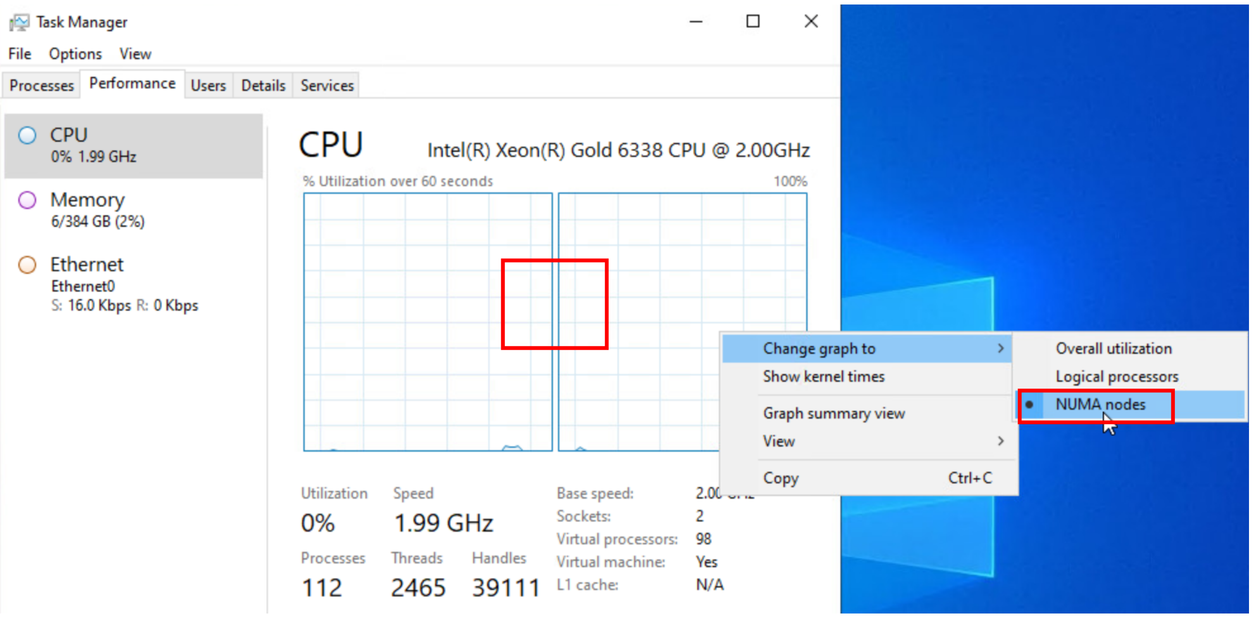
Figure 29. Balanced vNUMA Presentation, As Seen in ESXTop



```
[root@w2-hs-dmz-q2706:~] sched-stats -t numa-clients
groupName        groupID      clientID     homeNode     affinity     nWorlds      vmmWorlds    localMem     remoteMem    currLocal    cummLocal
vm.16850851      131122851    0            0            3            49           49           4382692      0            100          100
vm.16850851      131122851    1            1            3            49           49           3387356      0            100          100
```

In the Guest OS, we also see that ESXi exposes and present the expected topology.

Figure 30. vNUMA Topology, As Seen in Windows



CPU 0% 1.99 GHz

Memory 6/384 GB (2%)

Ethernet Ethernet0 S: 16.0 Kbps R: 0 Kbps

CPU Intel(R) Xeon(R) Gold 6338 CPU @ 2.00GHz

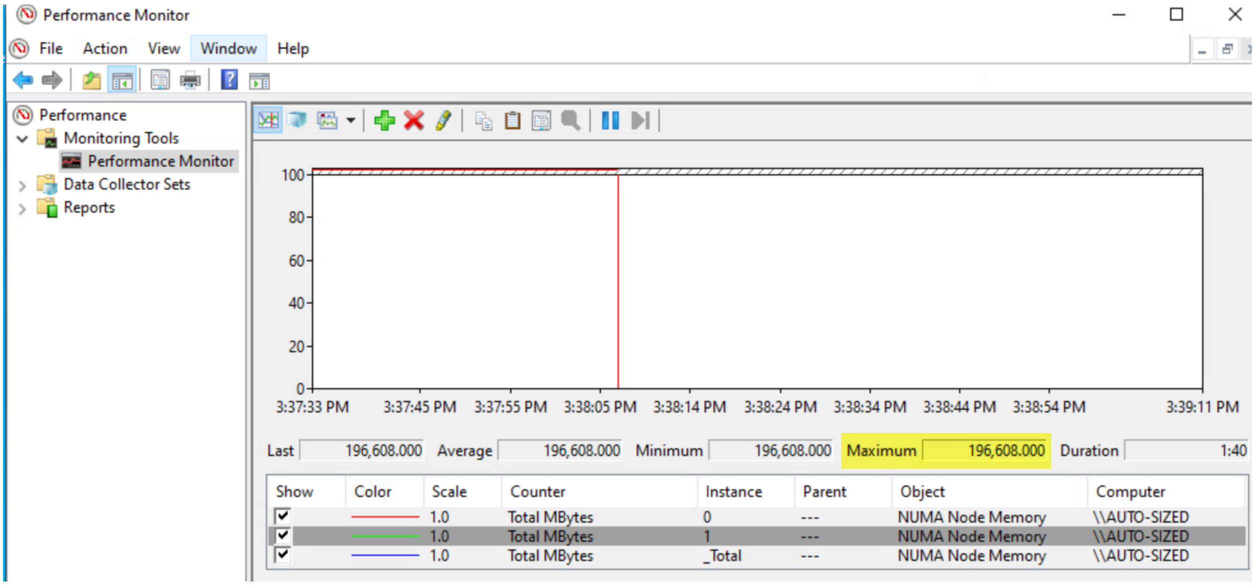
Utilization 0% Speed 1.99 GHz

Processes 112 Threads 2465 Handles 39111

Base speed: 2.00 GHz
Sockets: 2
Virtual processors: 98
Virtual machine: Yes
L1 cache: N/A

The compute resources are evenly distributed between the NUMA nodes, as each node has half of the allocated RAM, as shown in Perfmon:

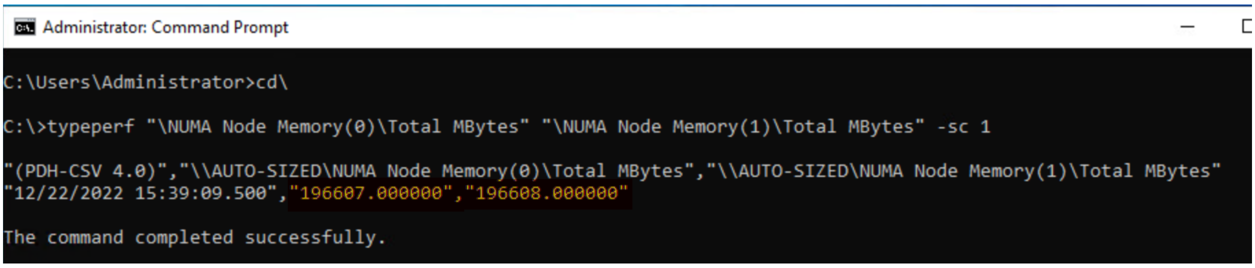
Figure 31. Wide VM Memory Distribution in Automatic vNUMA Configuration



Last 196,608.000 Average 196,608.000 Minimum 196,608.000 Maximum 196,608.000 Duration 1:40

Show	Color	Scale	Counter	Instance	Parent	Object	Computer
☑	🟥	1.0	Total MBytes	0	---	NUMA Node Memory	\\AUTO-SIZED
☑	🟩	1.0	Total MBytes	1	---	NUMA Node Memory	\\AUTO-SIZED
☑	🟦	1.0	Total MBytes	_Total	---	NUMA Node Memory	\\AUTO-SIZED

Figure 32. Wide VM Memory Distribution in Automatic vNUMA Configuration, As Seen in Windows



```
C:\Users\Administrator>cd\

C:\>typeperf "\NUMA Node Memory(0)\Total MBytes" "\NUMA Node Memory(1)\Total MBytes" -sc 1

"(PDH-CSV 4.0)","\\AUTO-SIZED\NUMA Node Memory(0)\Total MBytes","\\AUTO-SIZED\NUMA Node Memory(1)\Total MBytes"
"12/22/2022 15:39:09.500","196607.000000","196608.000000"

The command completed successfully.
```

This presentation is optimal and balanced for our SQL Server.

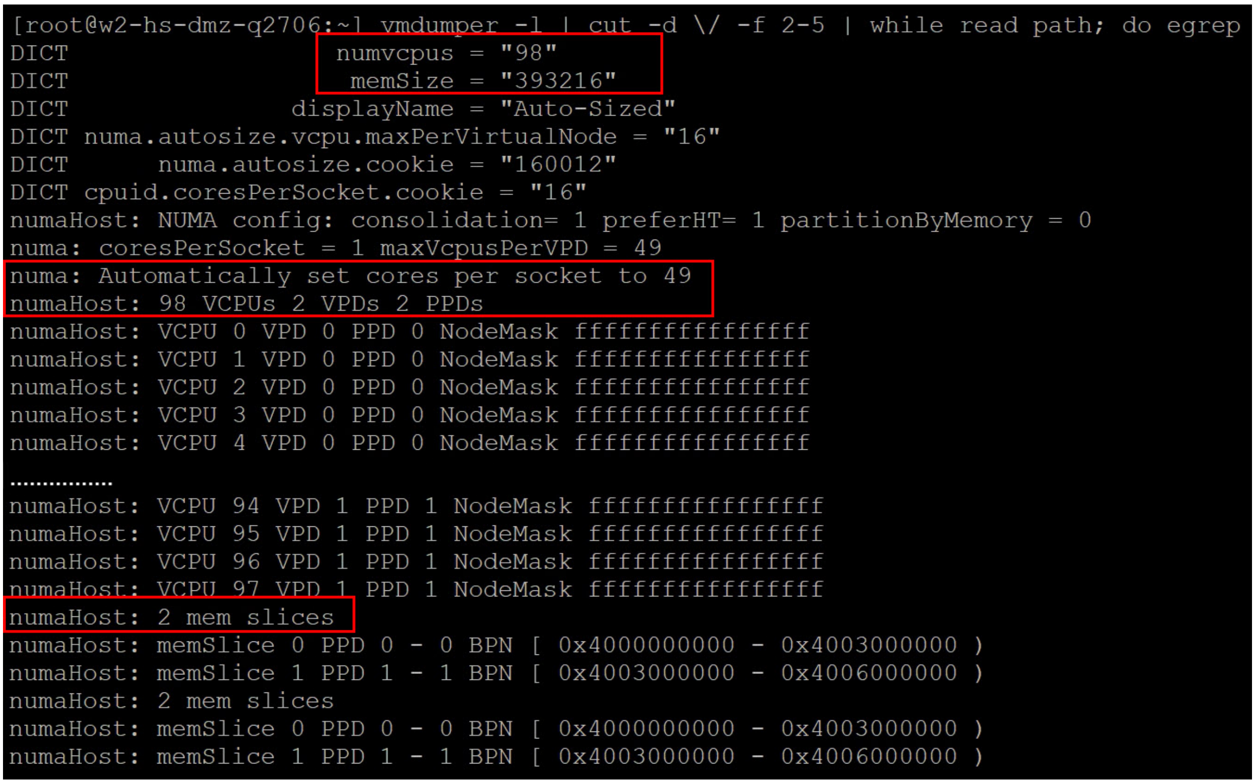
We have gone through these detailed examples to demonstrate the capabilities and behaviors of the new vNUMA topology configuration options in vSphere 8.0. It is expected that this feature will continue to be refined and optimized in subsequent versions, and it is our intention to update this guidance as warranted by any applicable changes.

Check vNUMA

After desired vNUMA topology is defined and configured, power on a VM and recheck how the final topology looks like. Following command on the ESXi host hosting the VM might be used[36]:

```
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 33. Checking NUMA Topology with vmdumper



```
[root@w2-hs-dmz-q2706:~] vmdumper -l | cut -d \/ -f 2-5 | while read path; do egrep
DICT                      numvcpus = "98"
DICT                       memSize = "393216"
DICT                   displayName = "Auto-Sized"
DICT numa.autosize.vcpu.maxPerVirtualNode = "16"
DICT      numa.autosize.cookie = "160012"
DICT cpuid.coresPerSocket.cookie = "16"
numaHost: NUMA config: consolidation= 1 preferHT= 1 partitionByMemory = 0
numa: coresPerSocket = 1 maxVcpusPerVPD = 49
numa: Automatically set cores per socket to 49
numaHost: 98 VCPUs 2 VPDs 2 PPDs
numaHost: VCPU 0 VPD 0 PPD 0 NodeMask ffffffffffffffff
numaHost: VCPU 1 VPD 0 PPD 0 NodeMask ffffffffffffffff
numaHost: VCPU 2 VPD 0 PPD 0 NodeMask ffffffffffffffff
numaHost: VCPU 3 VPD 0 PPD 0 NodeMask ffffffffffffffff
numaHost: VCPU 4 VPD 0 PPD 0 NodeMask ffffffffffffffff
................
numaHost: VCPU 94 VPD 1 PPD 1 NodeMask ffffffffffffffff
numaHost: VCPU 95 VPD 1 PPD 1 NodeMask ffffffffffffffff
numaHost: VCPU 96 VPD 1 PPD 1 NodeMask ffffffffffffffff
numaHost: VCPU 97 VPD 1 PPD 1 NodeMask ffffffffffffffff
numaHost: 2 mem slices
numaHost: memSlice 0 PPD 0 - 0 BPN [ 0x4000000000 - 0x4003000000 )
numaHost: memSlice 1 PPD 1 - 1 BPN [ 0x4003000000 - 0x4006000000 )
numaHost: 2 mem slices
numaHost: memSlice 0 PPD 0 - 0 BPN [ 0x4000000000 - 0x4003000000 )
numaHost: memSlice 1 PPD 1 - 1 BPN [ 0x4003000000 - 0x4006000000 )
```

VM vNUMA Sizing Recommendation

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

a. Best possible performance in general is observed when a VM could fit into one pNUMA node and benefit from local memory access. For example, when sizing a SQL Server VM on a host with 12 pCores per pNUMA node, the VM is more likely to perform better when allocated 12 vCPUs than it will be when allocated 14 vCPUs. This is because the allocated memory is more likely to be local to the 12 vCPUs than it would be with 14 vCPUs.
b. If a wide-NUMA configuration is unavoidable (for example, using the scenario described in (a) above), if business requirements have determined that the VM needs more than 12 vCPUs), consider double-checking the recommendations given and execute extensive performance testing before implementing the configuration. Monitoring should be implemented for important CPU counters after moving to production.

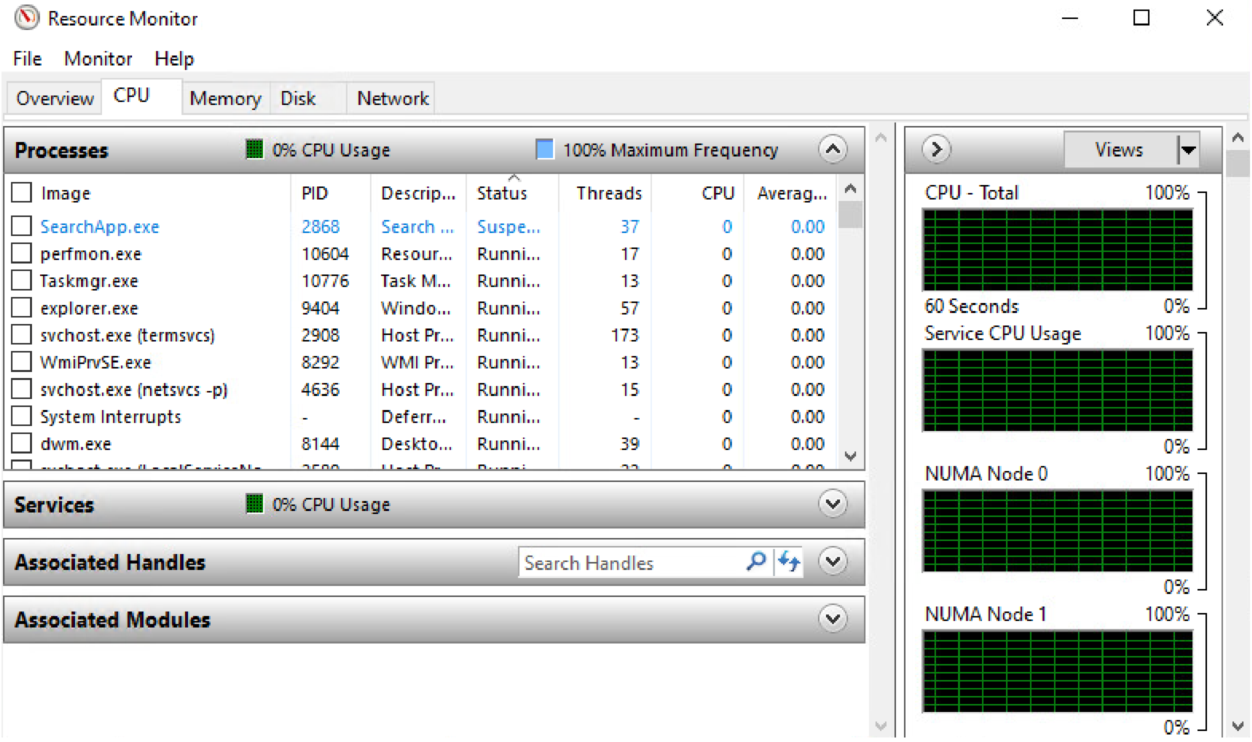
In-Guest operating system

Current editions of SQL Server could be run on Windows or Linux operating systems. In all cases, the most important part of NUMA configuration on this layer will be to re-check 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 is determined that the changes are required, bear in mind that it is not enough to restart a VM. Refer to the previous section to find how vNUMA topology can be adjusted.

Checking NUMA Topology in Windows OS

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

Figure 34. Windows Server Resource Monitor NUMA Topology View



Resource Monitor

File Monitor Help

Overview | CPU | Memory | Disk | Network

Processes 🟩 0% CPU Usage 🟦 100% Maximum Frequency

□ Image	PID	Descrip...	Status	Threads	CPU	Averag...
□ SearchApp.exe	2868	Search ...	Suspe...	37	0	0.00
□ perfmon.exe	10604	Resour...	Runni...	17	0	0.00
□ Taskmgr.exe	10776	Task M...	Runni...	13	0	0.00
□ explorer.exe	9404	Windo...	Runni...	57	0	0.00
□ svchost.exe (termsvcs)	2908	Host Pr...	Runni...	173	0	0.00
□ WmiPrvSE.exe	8292	WMI Pr...	Runni...	13	0	0.00
□ svchost.exe (netsvcs -p)	4636	Host Pr...	Runni...	15	0	0.00
□ System Interrupts	-	Deferr...	Runni...	-	0	0.00
□ dwm.exe	8144	Deskto...	Runni...	39	0	0.00

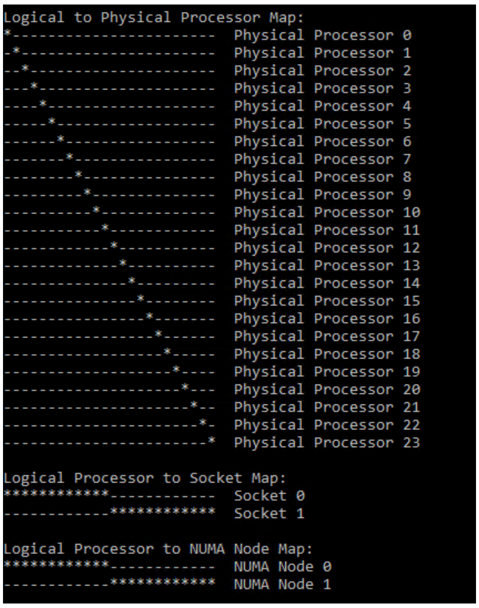
Services 🟩 0% CPU Usage

Associated Handles Search Handles

Associated Modules

Windows Resource Monitor is the quickest way to view NUMA topologies, as seen by the Operating System. Another useful tool is Coreinfo, originally from Sysinternals, now acquired and owned by Microsoft.[37]

Figure 35. Core Info Showing a NUMA Topology for 24 cores/2socket VM



```
Logical to Physical Processor Map:
*-----------------------  Physical Processor 0
-*----------------------  Physical Processor 1
--*---------------------  Physical Processor 2
---*--------------------  Physical Processor 3
----*-------------------  Physical Processor 4
-----*------------------  Physical Processor 5
------*-----------------  Physical Processor 6
-------*----------------  Physical Processor 7
--------*---------------  Physical Processor 8
---------*--------------  Physical Processor 9
----------*-------------  Physical Processor 10
-----------*------------  Physical Processor 11
------------*-----------  Physical Processor 12
-------------*----------  Physical Processor 13
--------------*---------  Physical Processor 14
---------------*--------  Physical Processor 15
----------------*-------  Physical Processor 16
-----------------*------  Physical Processor 17
------------------*-----  Physical Processor 18
-------------------*----  Physical Processor 19
--------------------*---  Physical Processor 20
---------------------*--  Physical Processor 21
----------------------*-  Physical Processor 22
-----------------------*  Physical Processor 23

Logical Processor to Socket Map:
************------------  Socket 0
------------************  Socket 1

Logical Processor to NUMA Node Map:
************------------  NUMA Node 0
------------************  NUMA Node 1
```

Checking NUMA Topology in Linux OS

Since SQL Server 2017, it is supported to run SQL Server on the selected Linux operating systems like Red Hat Enterprise Linux or Ubuntu[38]. Following utilities can be used to check the NUMA topology exposed.

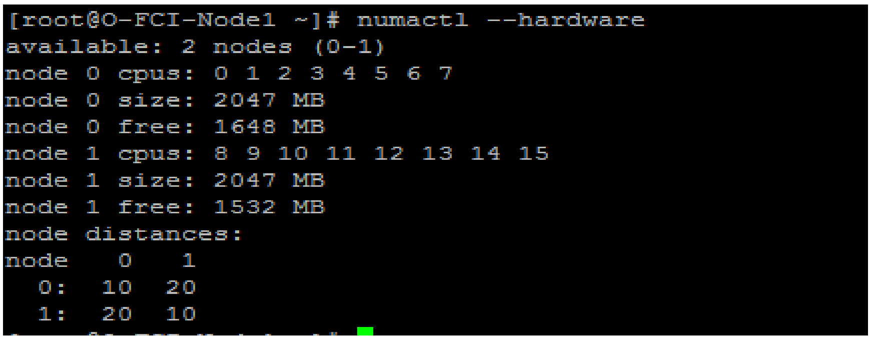
- Numactl[39]

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

Numactl –hardware

to get required information

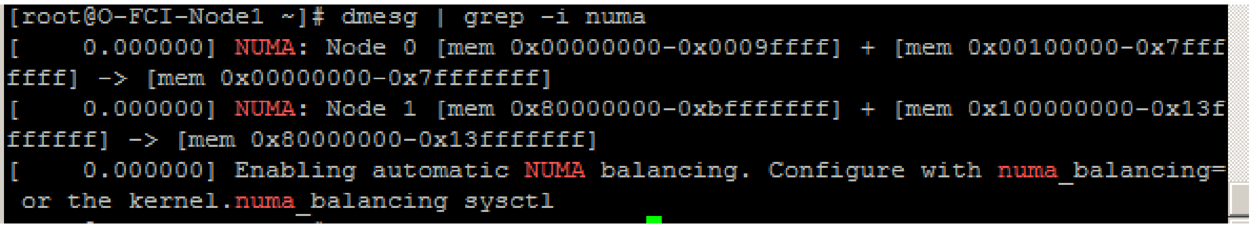
Figure 36. Using numactl to Display the NUMA Topology



```
[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 37. 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
```

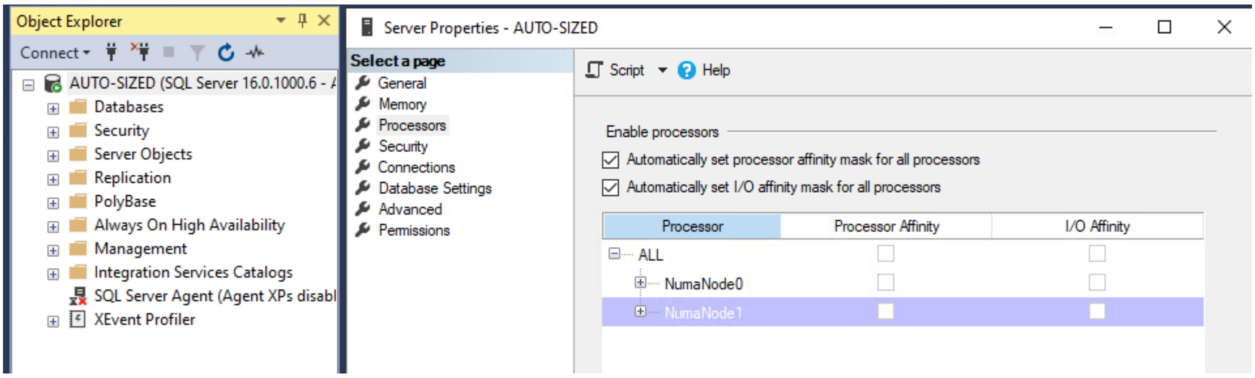
Ensure that acpi is **not** turned off, as it will disable NUMA as well: grep acpi=off /proc/cmdline

SQL Server

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

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

Figure 38. Displaying NUMA Information in the SQL Server Managemnet Studio



Enable processors

☑ Automatically set processor affinity mask for all processors

☑ Automatically set I/O affinity mask for all processors

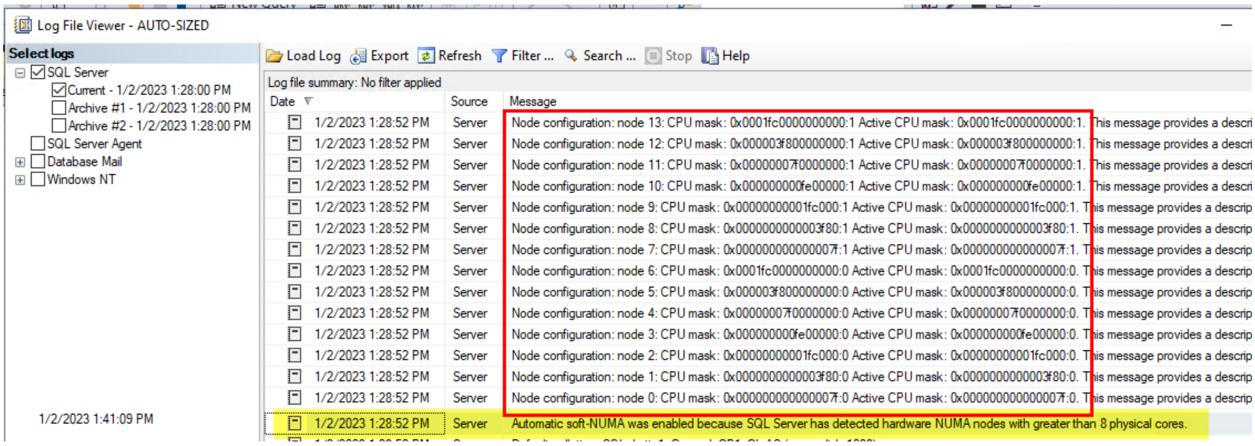
Processor	Processor Affinity	I/O Affinity
ALL	□	□
NumaNode0	□	□
NumaNode1	□	□

You can also view the information directly from SQL Server DMVs with the following command:

SELECT * FROM sys.dm_os_nodes;

Additional information can be obtained from the *errorlog* file after the restart of the VM or SQL Server services.

Figure 39. Errorlog Messages for Automatic soft-NUMA on 12 Cores per Socket VM



```
Log file summary: No filter applied
Date                  Source  Message
1/2/2023 1:28:52 PM   Server  Node configuration: node 13: CPU mask: 0x0001fc0000000000:1 Active CPU mask: 0x0001fc0000000000:1. This message provides a descri
1/2/2023 1:28:52 PM   Server  Node configuration: node 12: CPU mask: 0x000003f800000000:1 Active CPU mask: 0x000003f800000000:1. This message provides a descri
1/2/2023 1:28:52 PM   Server  Node configuration: node 11: CPU mask: 0x00000007f0000000:1 Active CPU mask: 0x00000007f0000000:1. This message provides a descri
1/2/2023 1:28:52 PM   Server  Node configuration: node 10: CPU mask: 0x000000000fe00000:1 Active CPU mask: 0x000000000fe00000:1. This message provides a descri
1/2/2023 1:28:52 PM   Server  Node configuration: node 9: CPU mask: 0x00000000001fc000:1 Active CPU mask: 0x00000000001fc000:1. This message provides a descrip
1/2/2023 1:28:52 PM   Server  Node configuration: node 8: CPU mask: 0x0000000000003f80:1 Active CPU mask: 0x0000000000003f80:1. This message provides a descrip
1/2/2023 1:28:52 PM   Server  Node configuration: node 7: CPU mask: 0x000000000000007f:1 Active CPU mask: 0x000000000000007f:1. This message provides a descrip
1/2/2023 1:28:52 PM   Server  Node configuration: node 6: CPU mask: 0x0001fc0000000000:0 Active CPU mask: 0x0001fc0000000000:0. This message provides a descrip
1/2/2023 1:28:52 PM   Server  Node configuration: node 5: CPU mask: 0x000003f800000000:0 Active CPU mask: 0x000003f800000000:0. This message provides a descrip
1/2/2023 1:28:52 PM   Server  Node configuration: node 4: CPU mask: 0x00000007f0000000:0 Active CPU mask: 0x00000007f0000000:0. This message provides a descrip
1/2/2023 1:28:52 PM   Server  Node configuration: node 3: CPU mask: 0x000000000fe00000:0 Active CPU mask: 0x000000000fe00000:0. This message provides a descrip
1/2/2023 1:28:52 PM   Server  Node configuration: node 2: CPU mask: 0x00000000001fc000:0 Active CPU mask: 0x00000000001fc000:0. This message provides a descrip
1/2/2023 1:28:52 PM   Server  Node configuration: node 1: CPU mask: 0x0000000000003f80:0 Active CPU mask: 0x0000000000003f80:0. This message provides a descrip
1/2/2023 1:28:52 PM   Server  Node configuration: node 0: CPU mask: 0x000000000000007f:0 Active CPU mask: 0x000000000000007f:0. This message provides a descrip
1/2/2023 1:28:52 PM   Server  Automatic soft-NUMA was enabled because SQL Server has detected hardware NUMA nodes with greater than 8 physical cores.
```

SQL Server Automatic Soft-NUMA and vNUMA

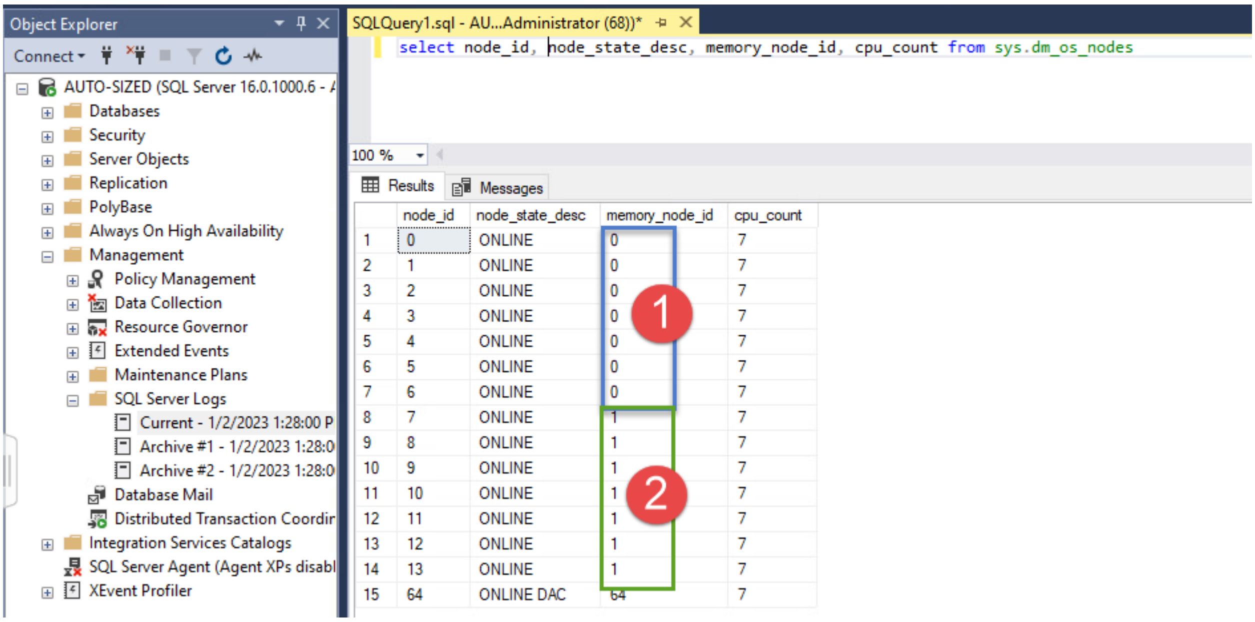
SQL Server Soft-NUMA feature was introduced to in response to the growing number of cores per pCPU. Soft-NUMA aims to partition available CPU resources inside one NUMA node into so-called “soft-NUMA” nodes. Although a cursory glance at the Soft-NUMA topologies auto-created by SQL Server may lead one to believe that they conflict with the topologies presented by ESXi and seen by the Guest Operating System, there are no technical incompatibility between the two. On the contrary, leveraging both features have been observed to further optimize scalability and performance of the database engine for most of the workload[40].

As shown in the image above, although SQL Server has partitioned the 98 vCPUs allocated to this VM into 14 “NUMA Nodes” of 7 cores each, a close examination shows that 7 of these (Soft-NUMA) Nodes are group into the same CPU mask boundary. There are two such boundaries, corresponding directly with the two vNUMA topologies presented to the Operating System.

Starting with SQL Server 2014 SP2, “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, the upper boundary for starting the partitioning and enabling “Soft-NUMA” is eight (8) cores per NUMA node, although (as shown in the image below), each Soft-NUMA node may contain less than eight cores, depending on the number of allocated vCPUs (in our case, 98 vCPUs are divided into 14 smaller Soft-NUMA nodes, each with 7 cores).

The result of having automatic soft-NUMA enabled can be noticed in the errorlog (Figure 22) and using sys.dm_os_nodes system view (Figure 24).

Figure 40. sys.dm_os_nodes Information on a System with 2 NUMA Nodes and 4 soft-NUMA Nodes


Object Explorer
Connect ▾
AUTO-SIZED (SQL Server 16.0.1000.6 - /
Databases
Security
Server Objects
Replication
PolyBase
Always On High Availability
Management
Policy Management
Data Collection
Resource Governor
Extended Events
Maintenance Plans
SQL Server Logs
Current - 1/2/2023 1:28:00 P
Archive #1 - 1/2/2023 1:28:0
Archive #2 - 1/2/2023 1:28:0
Database Mail
Distributed Transaction Coordir
Integration Services Catalogs
SQL Server Agent (Agent XPs disabl
XEvent Profiler
SQLQuery1.sql - AU...Administrator (68))*
100 %
Results
Messages
1
2

```sql
select node_id, node_state_desc, memory_node_id, cpu_count from sys.dm_os_nodes
```

	node_id	node_state_desc	memory_node_id	cpu_count
1	0	ONLINE	0	7
2	1	ONLINE	0	7
3	2	ONLINE	0	7
4	3	ONLINE	0	7
5	4	ONLINE	0	7
6	5	ONLINE	0	7
7	6	ONLINE	0	7
8	7	ONLINE	1	7
9	8	ONLINE	1	7
10	9	ONLINE	1	7
11	10	ONLINE	1	7
12	11	ONLINE	1	7
13	12	ONLINE	1	7
14	13	ONLINE	1	7
15	64	ONLINE DAC	64	7

Soft-NUMA partitions only CPU cores and does not provide the memory to CPU affinity[41]. The number of lazy writer threads created for Soft-NUMA is determined by the vNUMA nodes surfaced to the Operating System by ESXi. If the resultant topologies are deemed inefficient or less than desired, Administrators can manually modify the configuration by setting CPU Affinity mask, either programmatically through SQL statements or by editing the Windows registry. We encourage Administrators to consult Microsoft for accurate guidance on how to make this change, and to understand the effects of such changes to the stability and performance of their SQL Server instances.

Starting with vSphere 8.0, the considerations for presenting virtual CPUs to a virtual machine have changed to accommodate and reflect the increasing importance of enhanced virtual CPU topology for modern Guest Operating Systems and applications.

The virtual topology of a VM enables optimization within Guest OS for placement and load balancing. Selecting an accurate virtual topology that aligns with the underlying physical topology of the host where the VM is running is crucial for application performance.

ESXi 8.0 now automatically selects optimal coresPerSocket for a VM and optimal virtual L3 size. It also includes a new virtual motherboard layout to expose NUMA for virtual devices and vNUMA topology when CPU hotplug is enabled.

This enhanced virtual topology capability is available to a VM with hardware version 20 or above. Virtual hardware version 20 is available only for VMs created on ESXi 8.0 or later.

Cores per Socket

As it is still very common to use this setting to ensure that SQL Server Standard Edition will be able to consume all allocated vCPUs and can use up to 24 cores[42], it should be obvious after reading the previous chapter that while satisfying licensing needs, care should be taken to get the right vNUMA topology exposed, especially on vSphere 6.0 and below.

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

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. However, this default behavior can now be overridden through the new vNUMA Hot-Add feature introduced in vSphere 8.0. vNUMA HotAdd enables NUMA-aware applications such as SQL Server instances to benefit from the performance enhancements of surfacing virtual NUMA to the Operating System while simultaneously allowing for the operational efficiencies inherent in the ability to increase CPU resources for the VM during periods of increasing loads.

Theoretically, with this new vSphere vNUMA Hot-add capabilities and improvements added in the Windows Server 2022 Operating System, the issues described in the following references should no longer be applicable. Theoretically.

Unfortunately, while the original underlying root cause has been fixed in Windows Server 2022 (Note: The fix has not been back-ported to Windows Server 2019 and older versions), we continue to observe the lingering anomalies of Windows creating phantom NUMA Nodes when vNUMA HotAdd is enabled on all currently-shipping versions of Windows.

Consequently, for all versions of Windows and VMware vSphere, VMware continues to recommend that Customers do *not* enable CPU hot plug (and the new vNUMA HotAdd) as a general practice, especially for VMs that require (or can benefit from) vNUMA enlightenments. In these cases, right-sizing the VM’s CPU is always a better choice than relying on CPU hot plug, and 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.

Please refer to the following documents for background information on the impacts of CPU Hot Add on VMs and the applications hosted therein:

vNUMA is disabled if VCPU hot plug is enabled (2040375)

Enabling vCPU HotAdd creates fake NUMA nodes on Windows (83980)

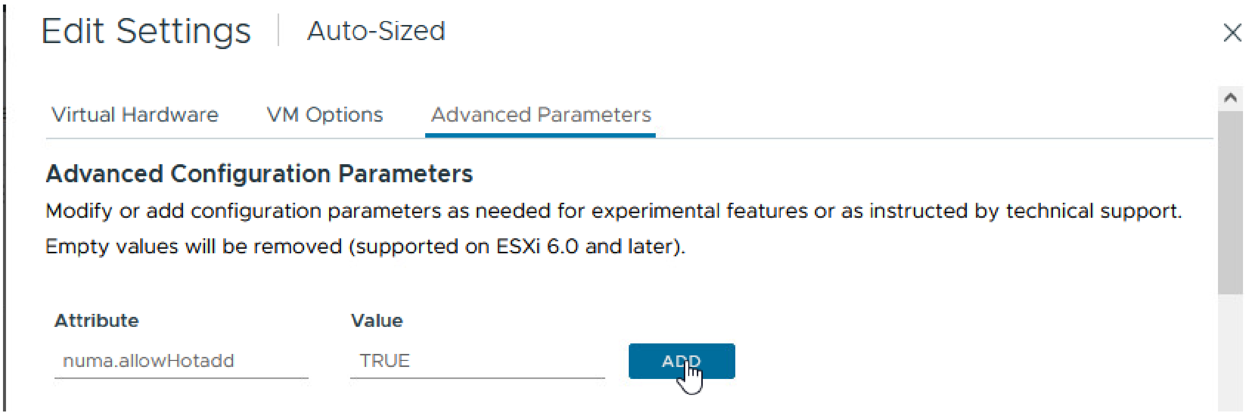
CPU HotAdd for Windows VMs: How BADLY Do You Want It?

Configuring CPU Hot Plug in vSphere 8.0

This Section provides a high-level demonstration of how Administrators can leverage the new CPU Hot add capabilities in vSphere 8.0 to improve performance and simplified administration for their SQL Server instances on vSphere. We encourage Customers to diligently validate these options in their non-production environments to be better understand their suitability for their own particular needs.

A new Virtual Machine Advanced configuration attribute (**numa.allowHotadd**) is required to enable the new vNUMA Hot Add feature on a VM, without disabling virtual NUMA for the VM.

Figure 41. CPU HotAdd VM Advanced Configuration



Edit Settings | Auto-Sized

Advanced Configuration Parameters

Modify or add configuration parameters as needed for experimental features or as instructed by technical support. Empty values will be removed (supported on ESXi 6.0 and later).

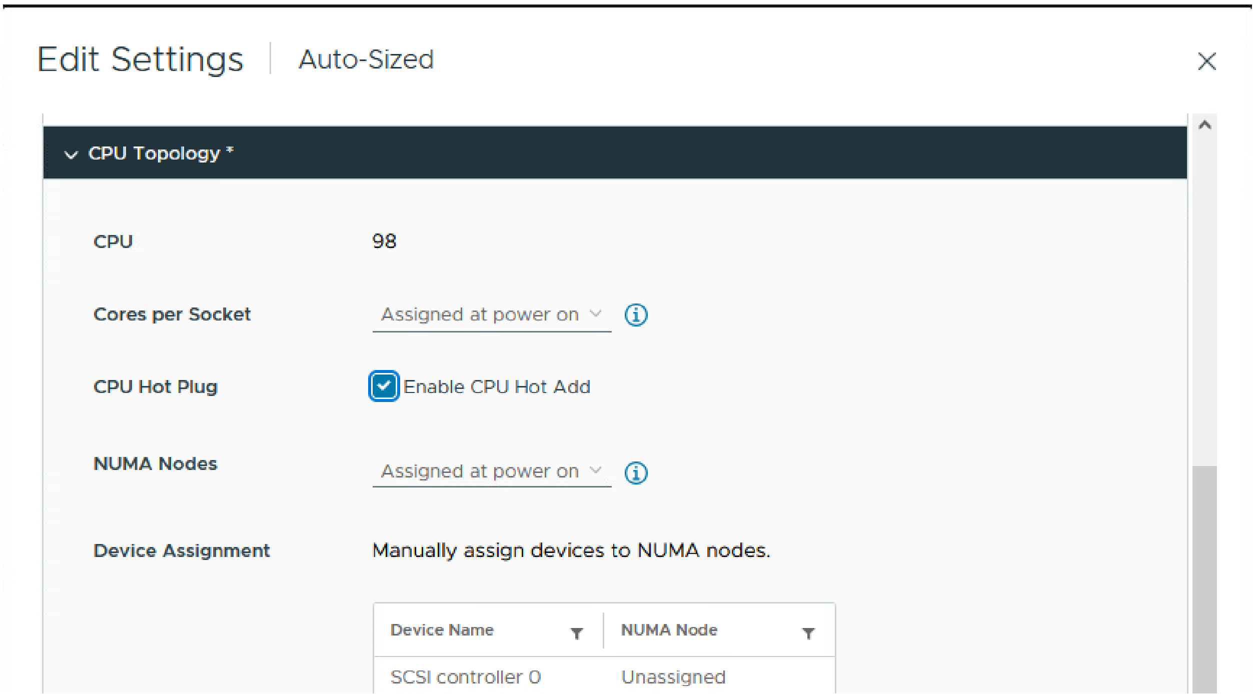
Attribute	Value
numa.allowHotadd	TRUE

Once this attribute is configured, the VM is now ready to support vNUMA Hot Add.

As with the rest of CPU-related configuration options, CPU and vCPU Hot Add settings are now configured in the “CPU Topology” section of the “VM Options” tab. The vCPU HotAdd setting is located in the “NUMA Nodes” section and, together with the advanced configuration option mentioned previously, controls the number of virtual NUMA nodes presented to the VM.

Check the box to enable CPU Hot Plug and click **Save.**

Figure 42. Enable CPU Hot Add on VM



Edit Settings | Auto-Sized

CPU Topology *

CPU: 98

Cores per Socket: Assigned at power on

CPU Hot Plug: ☑ Enable CPU Hot Add

NUMA Nodes: Assigned at power on

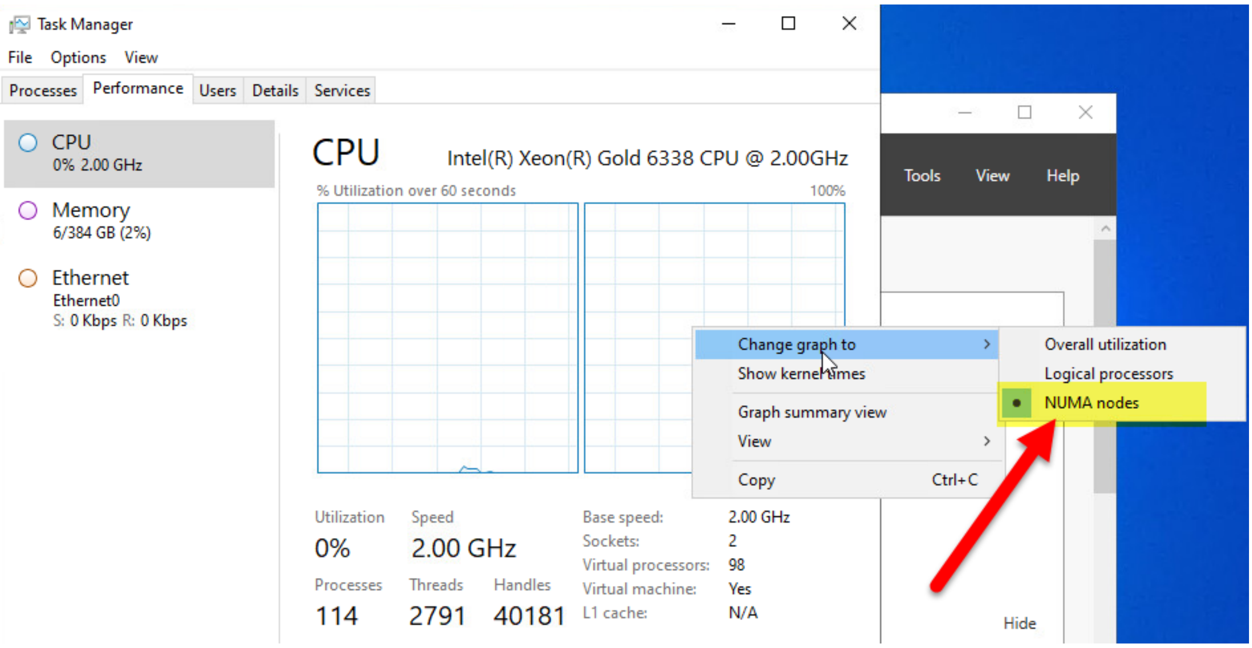
Device Assignment: Manually assign devices to NUMA nodes.

Device Name	NUMA Node
SCSI controller 0	Unassigned

Let’s power on the VM and examine the impact of the configuration in Windows.

As seen in the image below, ESXi has surfaced the vNUMA topology to the VM, in spite of the fact that CPU Hot Add is enabled.

Figure 43. vNUMA Available with CPU HotAdd



CPU
0% 2.00 GHz

Memory
6/384 GB (2%)

Ethernet
Ethernet0
S: 0 Kbps R: 0 Kbps

CPU Intel(R) Xeon(R) Gold 6338 CPU @ 2.00GHz

% Utilization over 60 seconds 100%

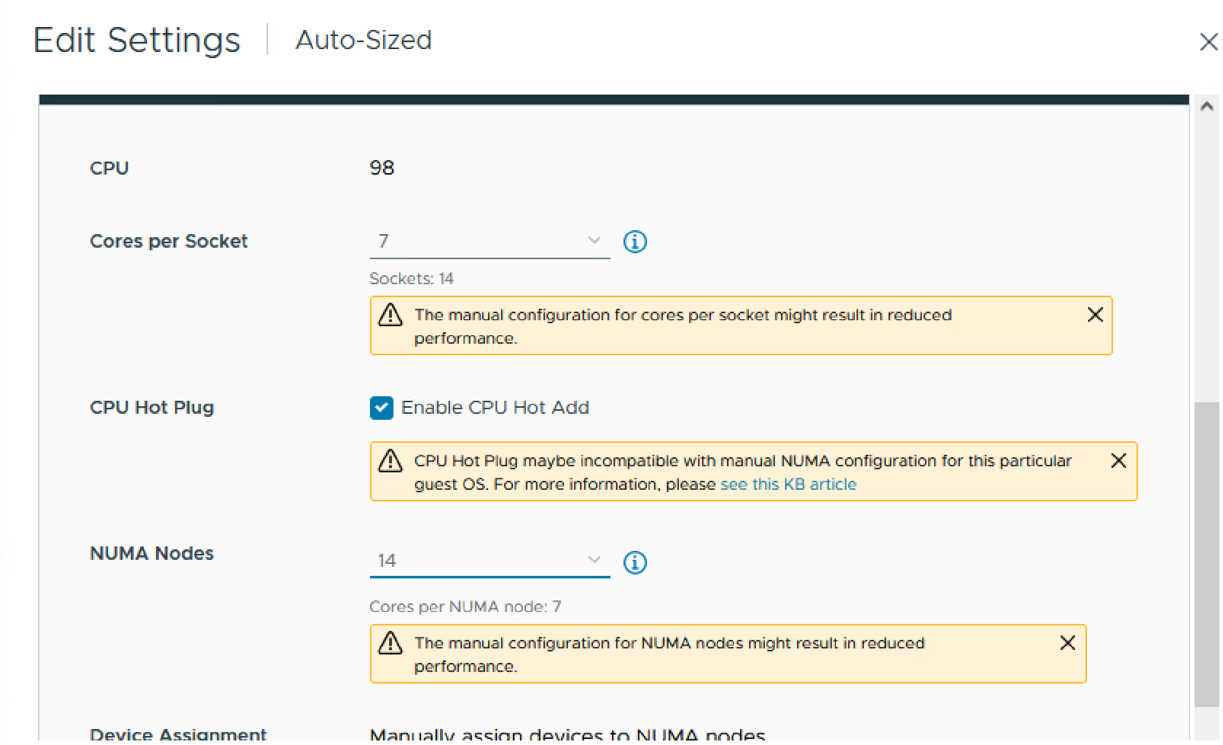
Utilization 0%
Speed 2.00 GHz
Processes 114
Threads 2791
Handles 40181

Base speed: 2.00 GHz
Sockets: 2
Virtual processors: 98
Virtual machine: Yes
L1 cache: N/A

In the image above, we have let ESXi auto-configure what it considers the most optimal vNUMA topology to the VM (two nodes).

What if we were to change the vNUMA presentation to, for example, mirror what the SQLOS is presenting with Soft-NUMA? In the images below, we see that the results are the same – Windows dutifully mirrors the NUMA topology configured in ESXi.

Figure 44. Manually Configured vNUMA Topology



Edit Settings | Auto-Sized

CPU 98

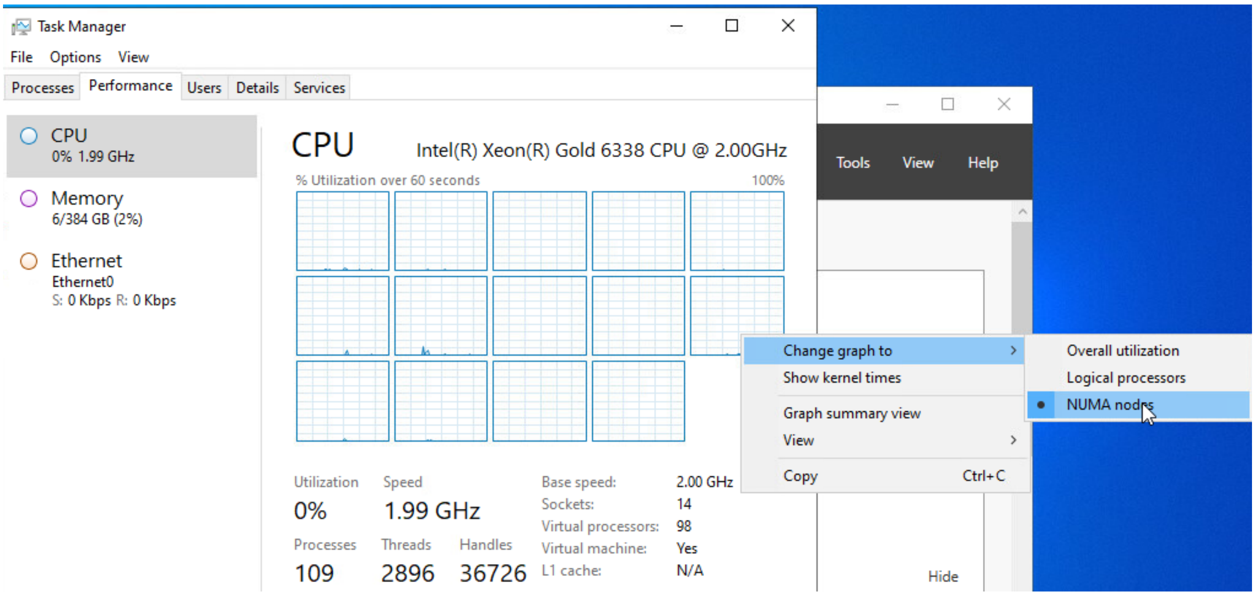
Cores per Socket 7
Sockets: 14
The manual configuration for cores per socket might result in reduced performance.

CPU Hot Plug ☑ Enable CPU Hot Add
CPU Hot Plug maybe incompatible with manual NUMA configuration for this particular guest OS. For more information, please see this KB article

NUMA Nodes 14
Cores per NUMA node: 7
The manual configuration for NUMA nodes might result in reduced performance.

Device Assignment Manually assign devices to NUMA nodes

Figure 45. Defined Topology Mirrored in Windows



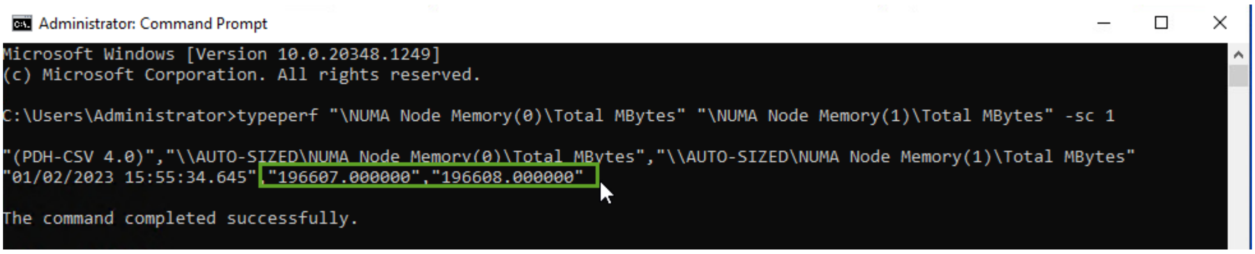
CPU Intel(R) Xeon(R) Gold 6338 CPU @ 2.00GHz

Utilization 0%
Speed 1.99 GHz
Processes 109
Threads 2896
Handles 36726
Base speed: 2.00 GHz
Sockets: 14
Virtual processors: 98
Virtual machine: Yes
L1 cache: N/A

CPU Hot Plug and “Phantom Node” in vSphere 8.0

Apart from disabling vNUMA, one of the most problematic issues with enabling CPU Hot Add for Windows VMs is that, when it is enabled, Windows creates a NUMA topology in which one or more nodes do not have any memory allocated. In this state, all CPUs in the phantom nodes execute instructions using remote memory, amplifying latencies and performance degradation as a result. As shown in the image below, this issue is not present in vSphere 8.0 and Windows Server 2022.

Figure 46. No "Phantom Node" with CPU HotAdd



```
Microsoft Windows [Version 10.0.20348.1249]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator>typeperf "\NUMA Node Memory(0)\Total MBytes" "\NUMA Node Memory(1)\Total MBytes" -sc 1

"(PDH-CSV 4.0)","\\AUTO-SIZED\NUMA Node Memory(0)\Total MBytes","\\AUTO-SIZED\NUMA Node Memory(1)\Total MBytes"
"01/02/2023 15:55:34.645","196607.000000","196608.000000"

The command completed successfully.
```

As previously mentioned, while this new feature enables a Guest Operating System and NUMA-aware applications hosted within to become aware of the virtual NUMA topology surfaced by ESXi, VMware recommends that Customer should not enable it for the Windows Operating System, due to the “Phantom NUMA Node” issue referenced earlier.

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 the 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 also disables the ability to vMotion a VM.

Per VM EVC Mode[43]

VMware introduced the ability to configure Enhanced vMotion Compatibility (EVC) mode at the VM level in vSphere Version 6.7. The per-VM EVC mode determines the set of host CPU features that a virtual machine requires to power on and migrate. The EVC mode of a virtual machine is independent of (and supersedes) 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/vCenter or to any of the multitudes of vSphere-based public cloud infrastructure.

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

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

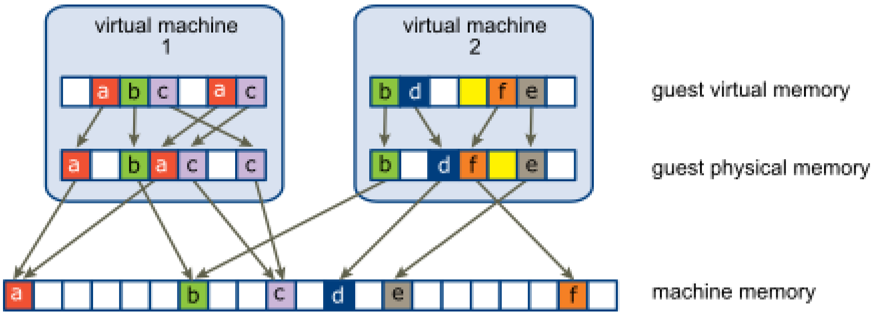
Virtual Machine Memory Configuration

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[44]. Insufficient hypervisor memory resources result in memory contention, having a significant impact on the SQL Server performance.

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 operating system's own memory management subsystem.[45]

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 48. Memory Mappings between Virtual, Guest, and Physical Memory



Memory Sizing Considerations

Memory sizing considerations include the following:

- When designing for performance to prevent memory contention between VMs, avoid overcommitment of memory at the ESXi host level (HostMem >= Sum of VMs memory – overhead). That means that if a physical server has 256 GB of RAM, do not allocate more than that amount to the virtual machines 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's native metrics query tool (the DMV) for this task. With respect to memory consumption/utilization, the sys.dm_os_process_memory counters provide the most accurate reporting. Because the SQL Server memory management operations are self-contained inside the SQLOS, SQL DMV counters provide a more reliable and authoritative measure of these metrics than what is provided by the vSphere counters ("memory consumed", "memory active", etc) or the Windows Task Manager metrics.
- Consider SQL Server version-related memory limitations while assigning memory to a VM. For example, SQL Server 2017 Standard edition supports a maximum 128 GB memory per instance, while relational database maximum memory capacity in SQL Server 2022 Enterprise edition tops out at 524PB.
- For situations where operational necessities require the creation of "unbalanced NUMA" memory configuration (this is the case when the amount of configured memory exceeds what's available within a single NUMA node while the number of allocated vCPUs fits within a NUMA node), VMware recommends that administrator should proactively configure the VM to have enough virtual NUMA nodes to accommodate the allocated memory size.

Memory Overhead[46]

Virtual machines require a certain amount of available overhead memory to power on. You should be aware of the amount of this overhead. The amount of overhead memory needed for a virtual machine depends on a large number of factors, including the number of vCPUs and memory allocation, the number and types of devices, the execution mode that the monitor is using, and the hardware version of the virtual machine.

The version of vSphere you are using can also affect the amount of memory needed. ESXi automatically calculates the amount of overhead memory needed for a virtual machine. To find out how much overhead memory is needed for your specific configuration, first power on the virtual machine in question. Look in the vmware.log file.

When the virtual machine powers on, the amount of overhead memory it needs is recorded in the log. Search within the log for "**VMMEM**" to see the initial and precise amount of overhead memory reserved for the virtual machine.

Memory Reservation

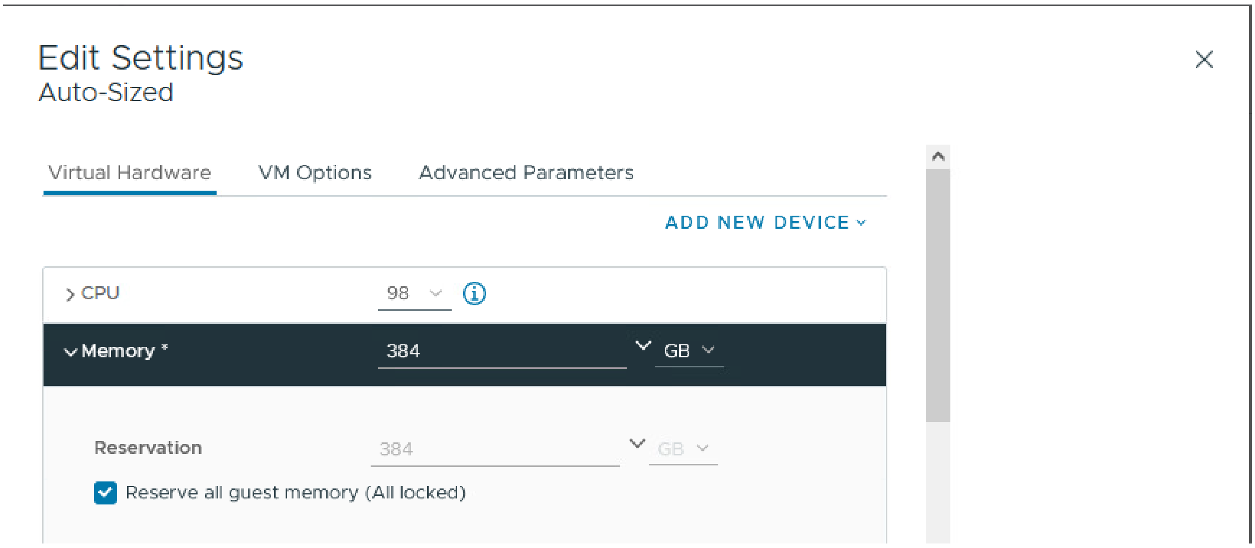
When achieving sufficient performance is the primary goal, consider setting the memory reservation equal to the provisioned memory. This will eliminate the possibility of ballooning or swapping from occurring and will guarantee that the VM will have exclusive access to all its reserved memory, even when there is more resource contention in the vSphere cluster.

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.

Figure 49. Setting Memory Reservation



Edit Settings
Auto-Sized

CPU 98

Memory * 384 GB

Reservation 384 GB

☑ Reserve all guest memory (All locked)

- **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 creation of the swap file and will save the disk space especially for VMs with big amount of memory assigned and if it will be the only one VM running on the host.**

If the “Reserve all guest memory” checkbox is NOT set, it is highly recommended to monitor host swap related counters (swap in/out, swapped). Even if swapping is the last resort for host to allocate physical memory to a VM and happens during congestion only, swapped VM memory will stay swapped even if congestion conditions are gone. If, for example, during extended maintenance or disaster recovery, an ESXi host experiences memory congestion and not all VM memory is reserved, the host will swap part of the VM memory. This memory will NOT be un-swapped automatically. If swapped memory is identified, consider either to vMotion, shut down and power on the VM or use unswap command[47] to reclaim physical memory backing for the swapped portion.

Memory Limit

In contrast with Memory Reservation, which is beneficial to VMs and the applications they host, the Memory Limit setting impedes a VM’s ability to consume all its allocated resources. This is because the limit is a rigid upper bound for a VM’s entitlement to compute resources. You can create limits on a VM to restrict how much CPU, Memory, or Storage I/O resources is allocated to the VM.

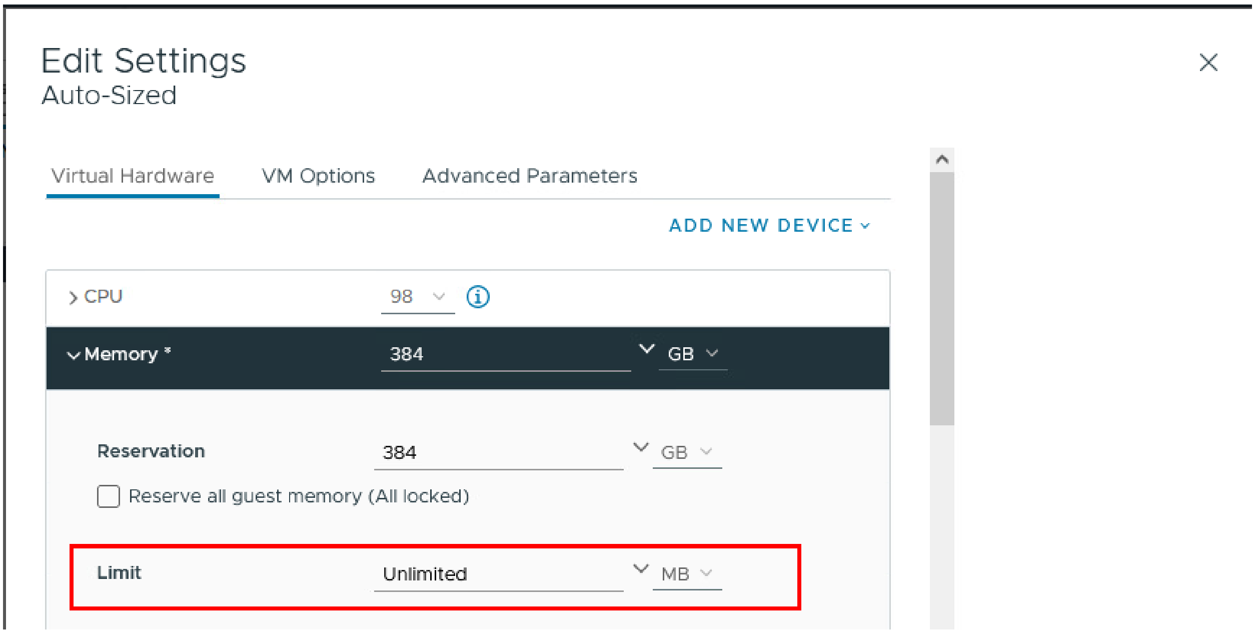
vSphere Administrators typically set limits on VM templates as part of their standard operating procedures. The intent is to avoid unnecessary resource consumptions (usually for test deployment scenarios). One of the draw-back of this practice is that

Administrators usually forget about this setting when using the same template to deploy mission-critical SQL Server workloads, which require more resources than those specified in the limits.

Because "Limit" takes priority over other considerations, a VM with, say, 64GB Memory limit will never receive anything beyond 64GB, regardless of how much more above that is allocated to it. Even setting FULL reservation on 500TB of Memory for the VM will not change the fact that there is a 64GB limit on that Memory resources for the VM.

Administrators are highly encouraged to avoid setting limits on compute resources allocated to a VM running SQL Server instances, and to always remember to check for limits when troubleshooting performance-related issues on these high-capacity VMs. Instead of using limits to control resource consumptions, consider engaging in the proper benchmarking tasks to estimate the appropriate amount of resources required to right-size the VM.

Figure 50. Configuring Memory Limit



Edit Settings
Auto-Sized

CPU 98
Memory * 384 GB
Reservation 384 GB
□ Reserve all guest memory (All locked)
Limit Unlimited MB

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 operating system's free memory table but will not delete the actual data from the page. The ESXi hypervisor does not have access to the operating system'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™[48], will 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, the memory that is paged to disk is 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 two-host failure. In case of a power outage that causes a failure of four 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 operating systems to page, allowing the important database servers to continue running in the least disruptive way to the business.

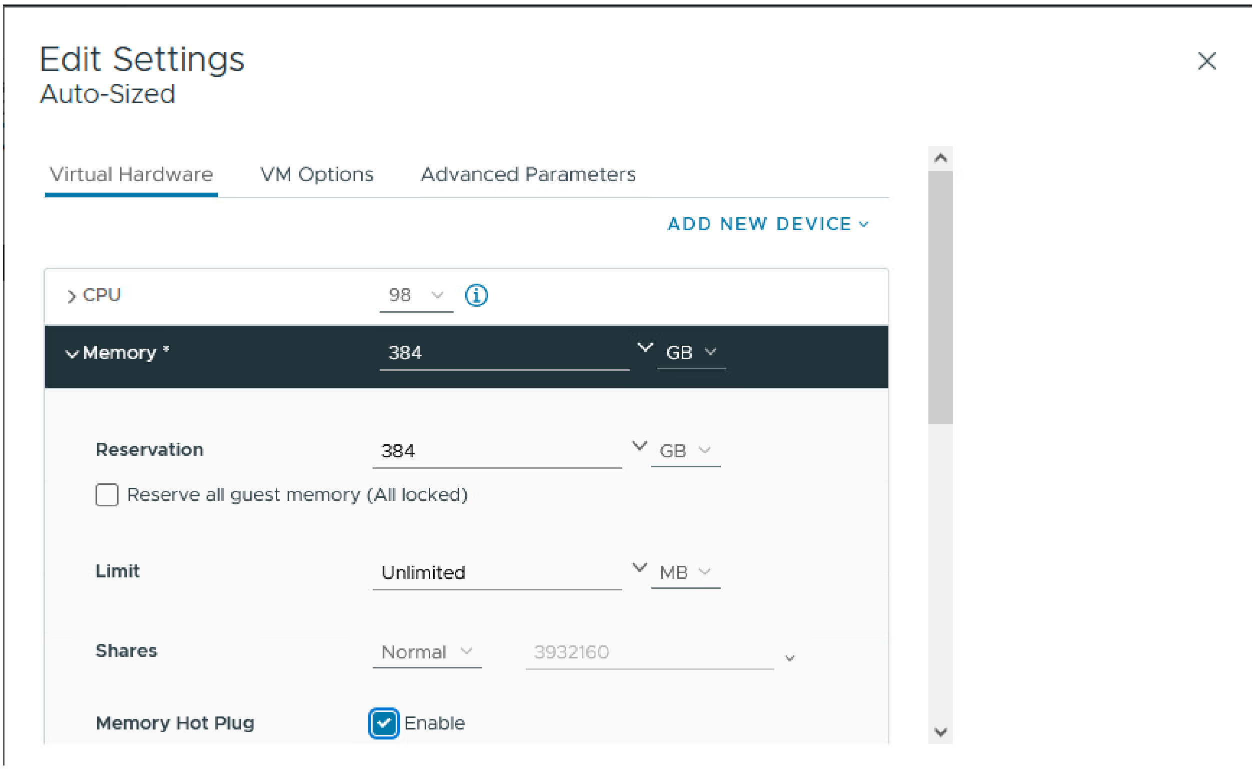
It's highly recommended to implement monitoring of the ballooned memory both on the host and VMs level. Use "ballooned memory" counter in vCenter Web Client to configure the alarm, or special tools like VMware Aria Operations.

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

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 VM with vNUMA enabled, it would always add it to node0, creating a NUMA imbalance[49]. With vSphere 6.0 and later, when enabling memory hot plug and adding memory to the VM, the memory will be added evenly to both 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 6.0 and later. After memory has been added to the VM, increase the max memory setting on the instance if one has been set. This can be done without requiring a server reboot or a restart of the SQL Server service, unless SQL Server's large memory pages is used, and a service restart is necessary. 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 51. Setting Memory Hot Plug



Edit Settings
Auto-Sized

CPU 98

Memory * 384 GB

Reservation 384 GB

□ Reserve all guest memory (All locked)

Limit Unlimited MB

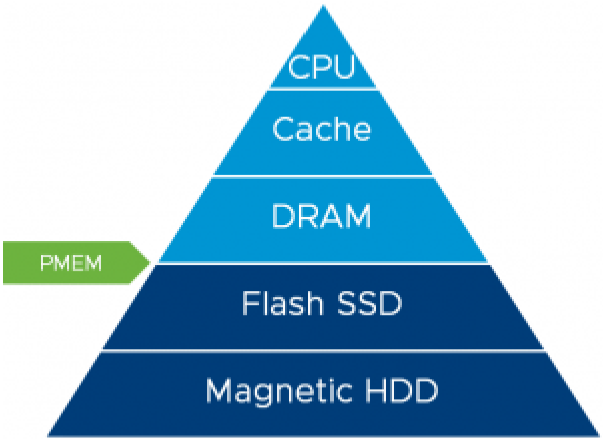
Shares Normal 3932160

Memory Hot Plug ☑ Enable

Persistent Memory

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 native PMem support in all currently shipping versions of Windows Server and SQL Server editions which support this feature, increasing the performance of high-loaded databases.

Figure 52. Positioning PMem



Persistent memory can be consumed by virtual machines in two different modes. [50]

- Virtual Persistent Memory (vPMem)[51] 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. Consider using vPMem when working with Windows 2016 Guest OS and SQL Server 2016 SP1 and above. For this combination, after a vPMem device is exposed to the Windows Guest 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 additional log file on a SCM volume configured as DAX[52].

```
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)[53], one PMem module could be efficiently shared between different VMs running on the same host, providing high saving costs by sharing one NVDIMM for many consumers.

vPMemDisk mode could be used with any version of Windows Guest OS/SQL Server as a traditional storage device, but with very low latency. Recent use cases demonstrated benefits of vPMemDisk for SQL Server backup and restore.[54]

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

With a full support of the persistent memory technology in 6.7, it looks very beneficial from SQL Server performance perspective to enhance hardware configuration of new servers with NVDIMM devices.

Virtual Machine Storage Configuration

Storage configuration is critical to any successful database deployment, especially in virtual environments where you might consolidate multiple SQL Server VMs 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.

For information about best practices for SQL Server storage configuration, please refer to Microsoft’s Performance Center for SQL Server Database Engine and Azure SQL Database. Follow these recommendations along with the best practices in this guide. Pay special attention to this section, as eight of ten performance issues are caused by storage subsystem configuration.

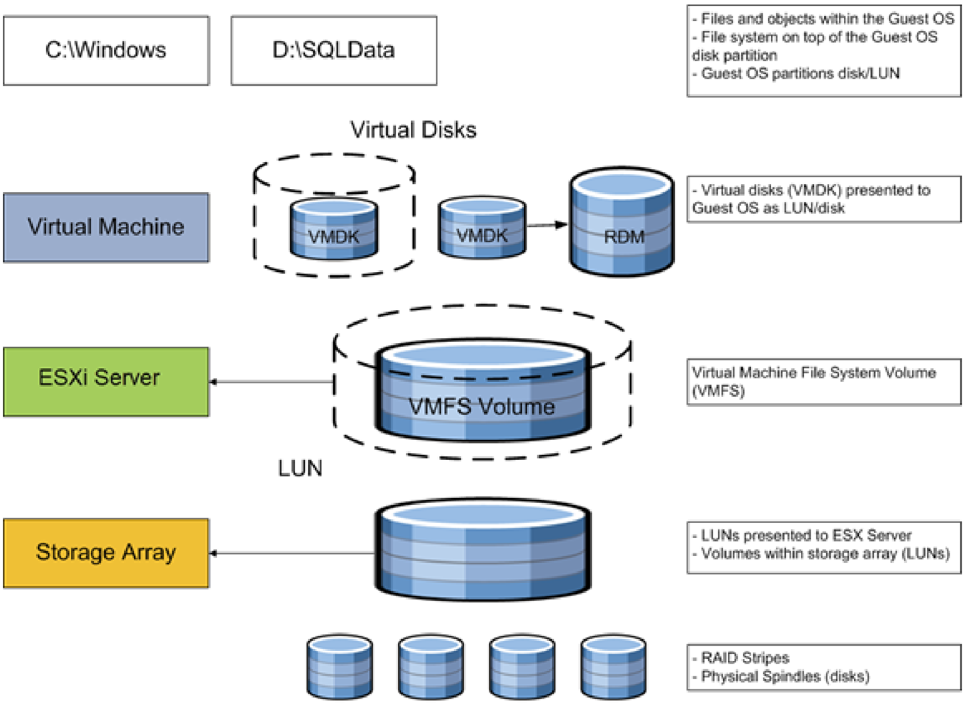
vSphere Storage Options

vSphere provides several options for storage configuration. The one that is the most widely used is a 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 takes storage management to the next level, where VMs are native objects on the storage system. Other options include hyper-converged solutions, such as VMware vSAN™ and/or all flash arrays, such as EMC’s XtremIO. This section covers the different storage options that exist for virtualized SQL Server deployments running on vSphere.

VMFS on Shared Storage Subsystem

The vSphere Virtual Machine File System (VMFS) 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 53. VMware Storage Virtualization Stack


C:\Windows
D:\SQLData
- Files and objects within the Guest OS
- File system on top of the Guest OS disk partition
- Guest OS partitions disk/LUN
Virtual Disks
Virtual Machine
VMDK
VMDK
RDM
- Virtual disks (VMDK) presented to Guest OS as LUN/disk
ESXi Server
VMFS Volume
Virtual Machine File System Volume (VMFS)
LUN
Storage Array
- LUNs presented to ESX Server
- Volumes within storage array (LUNs)
- RAID Stripes
- Physical Spindles (disks)

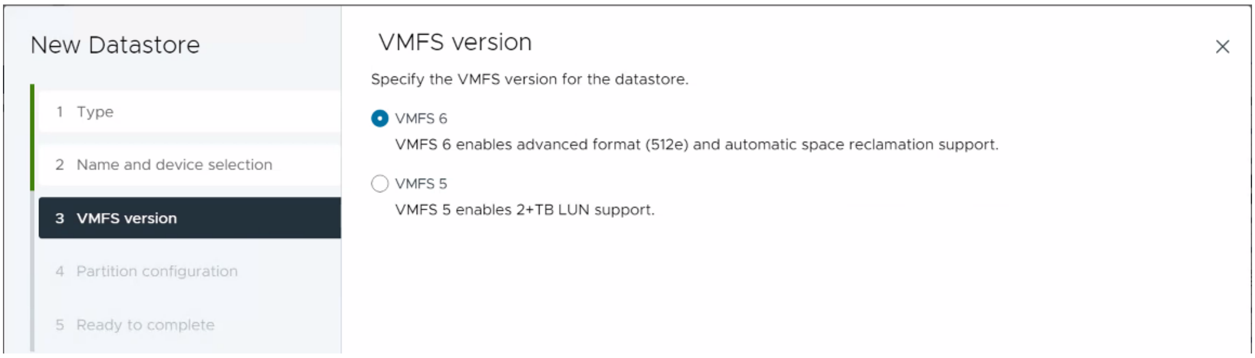
VMware Virtual Machine File System (VMFS)

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 available VMFS version supported by ESXi hosts in your environment. For a comprehensive list of the differences between VMFS5 and the latest version of VMFS (6), see Understanding VMFS Datastores.

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

Figure 54. VMFS6 vs VMFS5



VMFS version

Specify the VMFS version for the datastore.

● VMFS 6

VMFS 6 enables advanced format (512e) and automatic space reclamation support.

○ VMFS 5

VMFS 5 enables 2+TB LUN support.

Network File System (NFS)[55]

An NFS client built into ESXi 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. The main difference from the block storage is that NAS/NFS will provide file level access, VMFS formatting is not required for NFS datastores.

Although VMware vSphere supports both NFS 3 and NFS 4.1, it is important to note that there are some vSphere features and operations (e.g. SIOC, SDRS, SRM with SAN or vVols, etc.) which are currently unsupported when using NFS 4.1. For a comprehensive list of the benefits of (and differences between) each version, please see Understanding Network File System

Datastores.

vSphere 8.0 now has the ability to validate NFS mount requests and NFS mount retries on failure.

NFS Datastores considerations:

vSphere 8.0 supports up to 256 NFS mount points per each version of the NFS protocols. This means that you can have 256 NFS3 and 256 NFS4.1 datastores mounted simultaneously on each ESXi Host.

By default, the VMkernel interface with the lowest number (aka vmk0) will be used to access the NFS server. Ensure, that the NFS server is located outside of the ESXi management network (preferably, a separate non-routable subnet) and that separate VMkernel interface is created to access the NFS Server.

Consider using at least 10 Gigabit physical network adapters to access the NFS server.

For more details, consult the following references:

- NFS Storage Guidelines and Requirements
- Best Practices for Running NFS with VMware vSphere
- Best Practices for running VMware vSphere™ on Network Attached Storage

SQL Server support on NFS datastore in virtualized environment

SQL Server itself provides native (without trace flag) support for placing databases on network files starting with the version 2008, and includes support for clustered databases starting with version 2012[56]. In case of the virtualized platform, instance of SQL Server running in virtual machine, placed on NFS datastore has no knowledge of the underlying storage type. This fact imposes following supported configurations:

- NFS datastores are supported for a VM running SQL Server 2008 and above
- Always On Availability Groups (AG) using non-shared storage starting with version SQL Server 2012
- Shared Disk (FCI) clustering is not supported on NFS datastores

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. Over the years, the technical rationale for the use of RDMs for virtualized workloads on vSphere has gradually diminished, due to increasing optimization of the native VMFS and VMDKs, and due to the introduction of vVols.

From a performance perspective, both VMFS and RDM volumes can provide similar transaction throughput[57]. The following charts summarize some performance testing[58].

Figure 55. Random Mixed (50% Read/50% Write) I/O Operations per Second (Higher is Better)

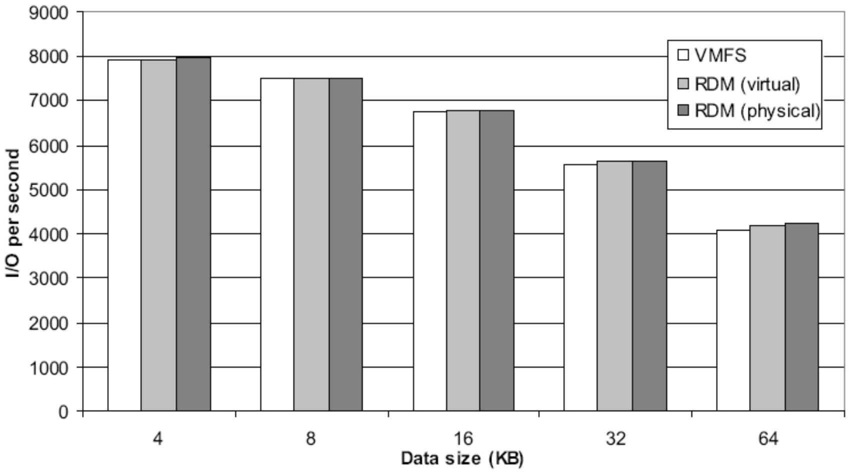
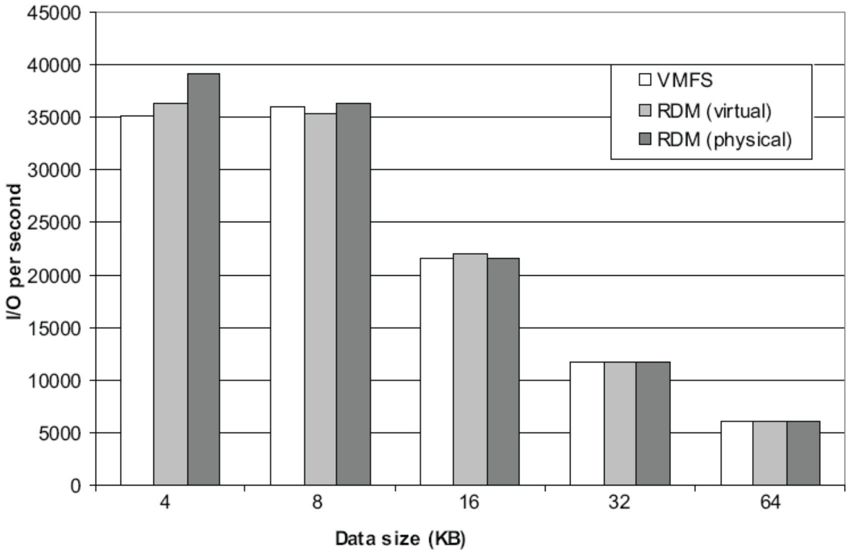


Figure 56. Sequential Read I/O Operations per Second (Higher is Better)



Clustered VMDK[59]

After closing the performance and size gaps between VMFS and RDM (both RDM and VMFS/VMDK can be up to 62TB in size)[60], the primary consideration for using RDM disks became the need to support SCSI-3 Persistent Reservation requirements for Microsoft Windows Server Failover Clustering (WSFC). WSFC is the clustering solution underpinning all application-level High Availability configuration options on the Microsoft Windows platform, including the Microsoft SQL Server Always on Failover Cluster Instance (FCI), which requires shared disks among/between participating nodes. With the release of the “Clustered VMDK” feature in vSphere 7.0, VMDKs can now be successfully shared among/between WSFC nodes, with support for SCSI-3 Persistent Reservation capabilities. RDMs are, therefore, no longer required for WSFC shared-disk clustering[61].

Considerations and limitations for using Clustered VMDKs are detailed in the “Limitations of Clustered VMDK support for WSFC” section of VMware vSphere Product Documentation[62]

With a few restrictions, you can enable Clustered VMDK support on existing VMFS Datastore. Because Clustered VMDK-enabled Datastores are not intended to be general-purpose Datastores, we recommend that, where possible and practical, Customers should create new dedicated LUNs for use when considering Clustered VMDKs.

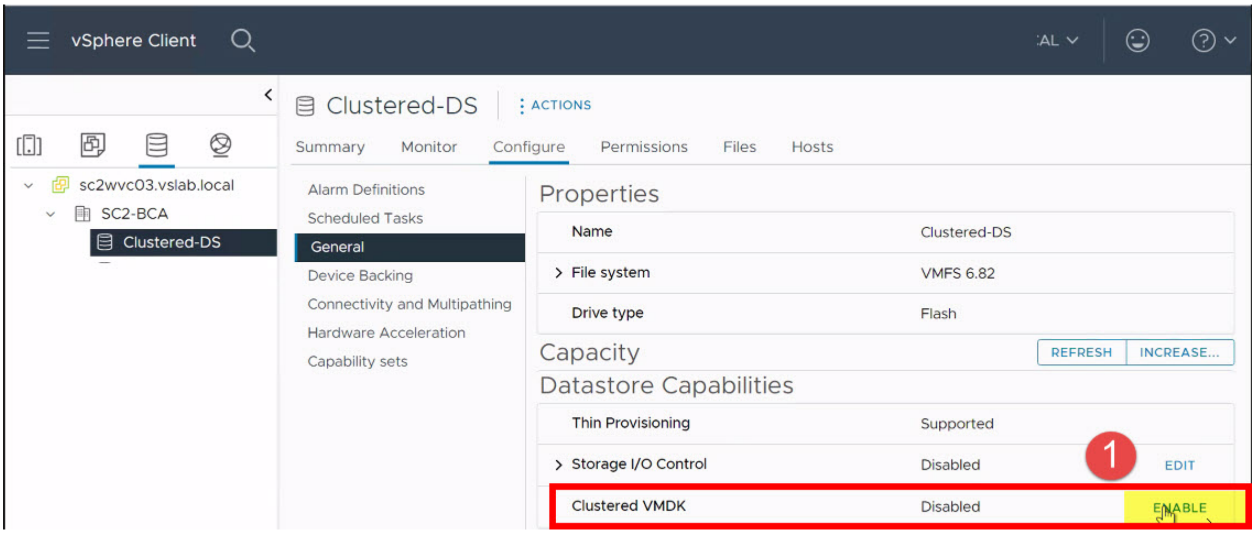
The most common use envisioned for this feature is the support for shared-disk Windows Server Failover Clustering (WSFC), which is required for creating SQL Server Failover Clustering Instance (FCI).

If you must re-use an existing Datastore for this purpose, VMware highly recommend that you migrate all existing VMDKs away from the target Datastore, especially if those VMDKs will not be participating in an FCI configuration. VMware does not support mixing shared VMDKs and non-shared VMDKs in a Clustered VMDK-enabled Datastore.

You can enable support for Clustered VMDK on a Datastore only after the Datastore has been provisioned.

The process is as shown in the images below:

Figure 57. Enabling Clustered VMDK (Step 1)



Properties

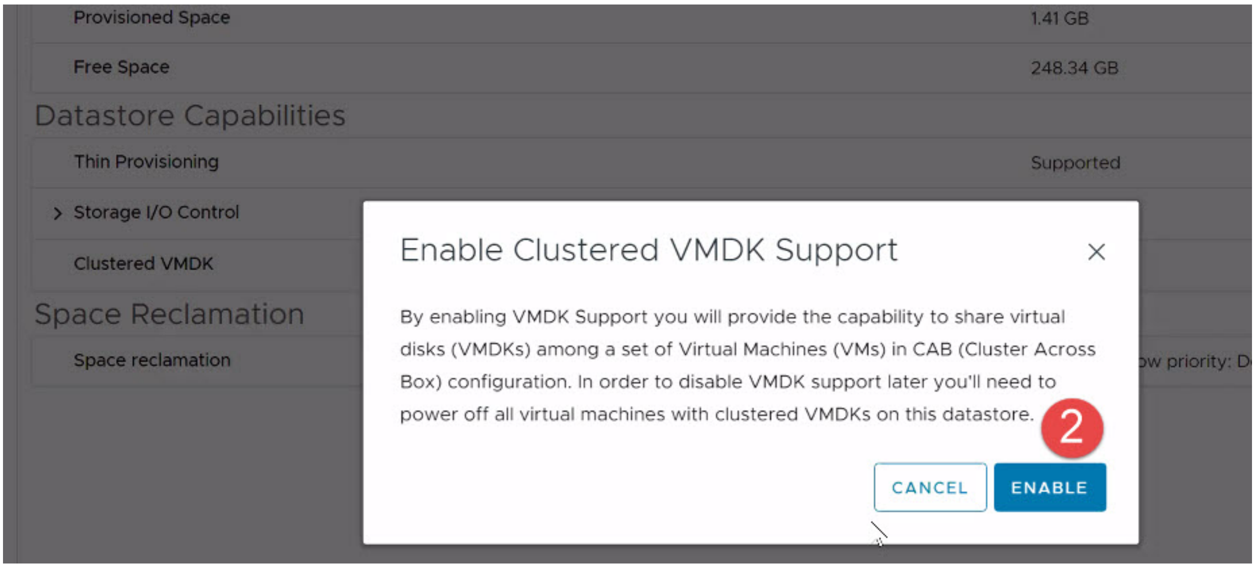
Name	Clustered-DS
File system	VMFS 6.82
Drive type	Flash

Capacity

Datastore Capabilities

Thin Provisioning	Supported
Storage I/O Control	Disabled
Clustered VMDK	Disabled

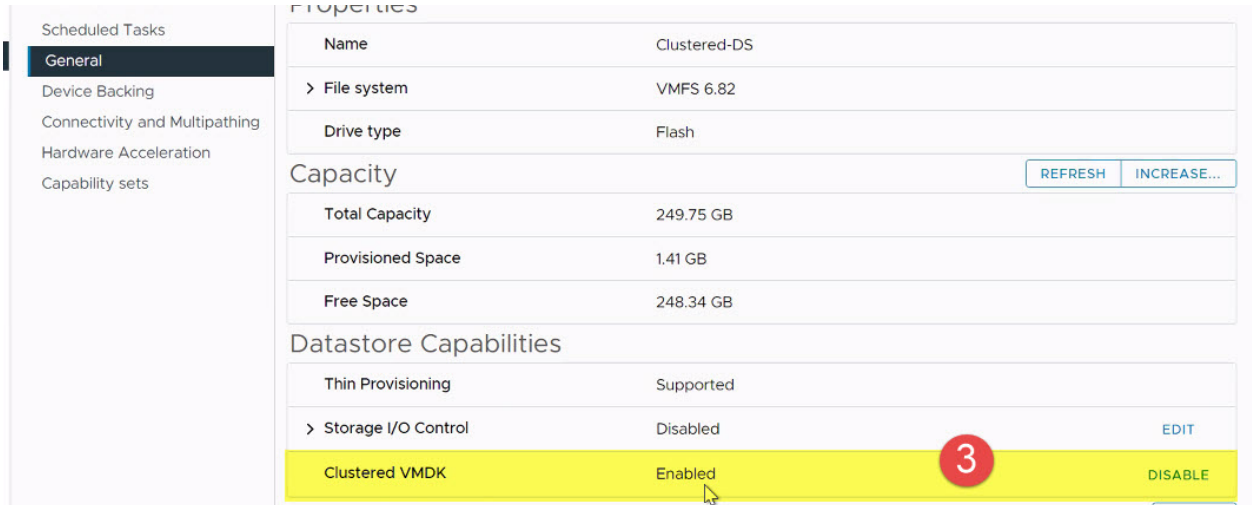
Figure 58. Enabling Clustered VMDK (Step 2)



Enable Clustered VMDK Support

By enabling VMDK Support you will provide the capability to share virtual disks (VMDKs) among a set of Virtual Machines (VMs) in CAB (Cluster Across Box) configuration. In order to disable VMDK support later you'll need to power off all virtual machines with clustered VMDKs on this datastore.

Figure 59. Enabling Clustered VMDK (Step 3)



Name	Clustered-DS
File system	VMFS 6.82
Drive type	Flash

Capacity

Total Capacity	249.75 GB
Provisioned Space	1.41 GB
Free Space	248.34 GB

Datastore Capabilities

Thin Provisioning	Supported
Storage I/O Control	Disabled
Clustered VMDK	Enabled

vSphere Virtual Volumes[63]

vSphere Virtual Volumes 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 vSphere Virtual Volumes are focused on operational efficiencies and flexible consumption models:

- Flexible consumption at the logical level – vSphere Virtual Volumes 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 – vSphere Virtual Volumes 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 vSphere Virtual Volumes, it dynamically instantiates the necessary data services. Through policy 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 started with vSphere 6.7. If the underlying storage subsystem does not support Clustered VMDK, vVol disks could be used instead of a RDM disk to provide a disk resource for the Windows failover cluster.

Figure 60. vSphere Virtual Volumes

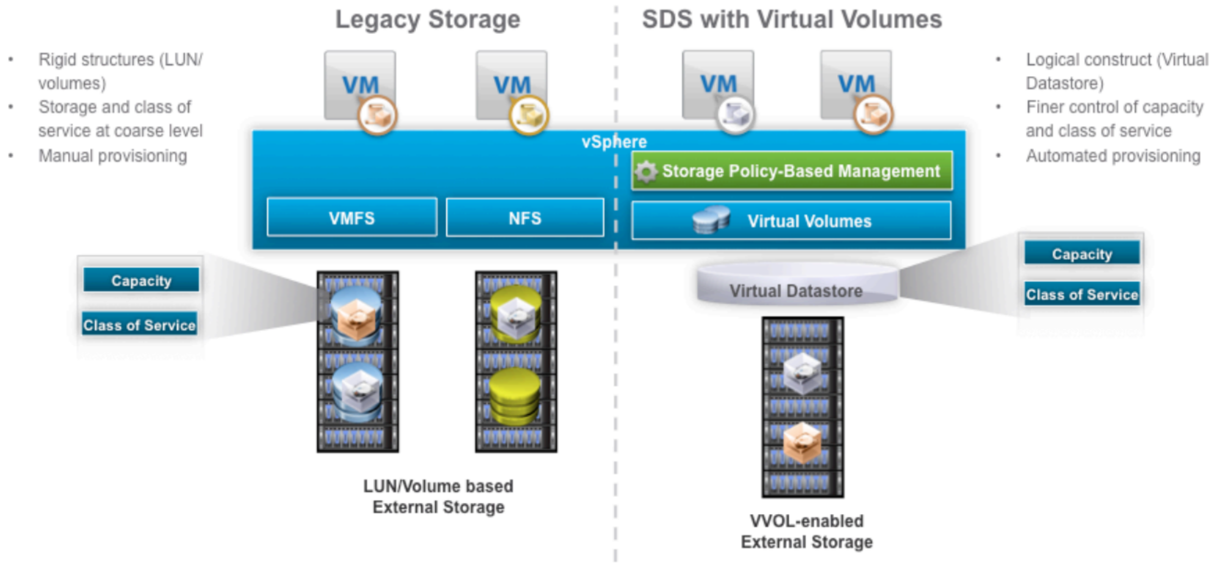

Legacy Storage
SDS with Virtual Volumes
Rigid structures (LUN/ volumes)
Storage and class of service at coarse level
Manual provisioning
Logical construct (Virtual Datastore)
Finer control of capacity and class of service
Automated provisioning
VM
VM
VM
VM
vSphere
Storage Policy-Based Management
VMFS
NFS
Virtual Volumes
Capacity
Class of Service
Virtual Datastore
Capacity
Class of Service
LUN/Volume based External Storage
VVOL-enabled External Storage

Figure 2: *Differences in Legacy storage and Software-Defined Storage with vSphere Virtual Volumes*

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

vSphere Virtual Volumes 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 vSphere Virtual Volumes 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.

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 intensive, and a misconfigured storage subsystem can increase I/O latency and significantly degrade performance of SQL Server.

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. An unaligned partition results in additional I/O operations, incurring penalties on latency and throughput. vSphere 5.0 and later automatically aligns VMFS5 partitions along a 1 MB boundary. If a VMFS3 partition was created using an earlier version of vSphere that aligned along a 64 KB boundary, and that file system is then upgraded to VMFS5, it will retain its 64 KB alignment. Such VMFS volumes should be reformatted. 1 MB alignment can only be achieved when the VMFS volume is create using the vSphere Web Client.

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.
- Consult with the storage vendor for alignment recommendations on their hardware.

For more information, see the white paper Performance Best Practices for vSphere 6.5 (https://www.vmware.com/techpapers/2017/Perf_Best_Practices_vSphere65.html).

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 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.

In most cases, 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[64].

All flash arrays utilize a 100 percent thin provisioning mechanism to be able to have storage on demand. However, vSphere requires the use of EagerZeroedThick vmdks for certain disk types, especially when such disks are shared among multiple VMs (as in an Always On Failover Clustering Instance configuration). Wherever possible, we recommend that vmdks used for Microsoft SQL Server instance's Transaction Logs, TempDB and Data file volumes be formatted as EagerZeroedThick for administrative efficiency, standardization and performance.

Optimize with Device Separation

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

Table 3. Typical SQL Server Disk Access Patterns

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

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 a substantial improvement in throughput and 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 largely sequential. Traditional storage built with spinning disk media requires repositioning of the disk head for random read and write access. 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, and thus optimizes storage performance.

Beginning with vSphere 6.7, vSphere has supported up to 64 SCSI targets per VMware Paravirtualized SCSI (PVSCSI) adapter, making it possible to have up to 256 VMDKs per VMs and up to 4096 paths per ESXi Host. In vSphere 8, the drivers required to support PVSCSI controller are now native to modern versions of the Windows OS, so there is no longer any need to use the old LSI Logic SAS controller for the Windows OS volume, as was the practice before now. By using all possible four PVSCSI controllers to distribute assigned disks across a VM, Administrators are able to leverage both the superior performance features and increased capacity of PVSCSI to optimize SQL Server storage I/O requirements.

The following guidelines can help to achieve best performance:

- Place SQL Server data (system and user), transaction log, and backup files into separate VMDKs, preferably in separate datastores. The SQL Server binaries are usually installed in the OS VMDK. 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 underlying 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 the 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 to schedule invocation of the SDRS for off-peak hours to avoid performance penalties while moving a VM.[65]

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 higher I/O rate, but also lowering CPU consumption compared to the LSI Logic SAS controller. In addition, the PVSCSI adapters provides higher queue depths, increasing I/O bandwidth for the virtualized workload. See OS Configuration section for more details.

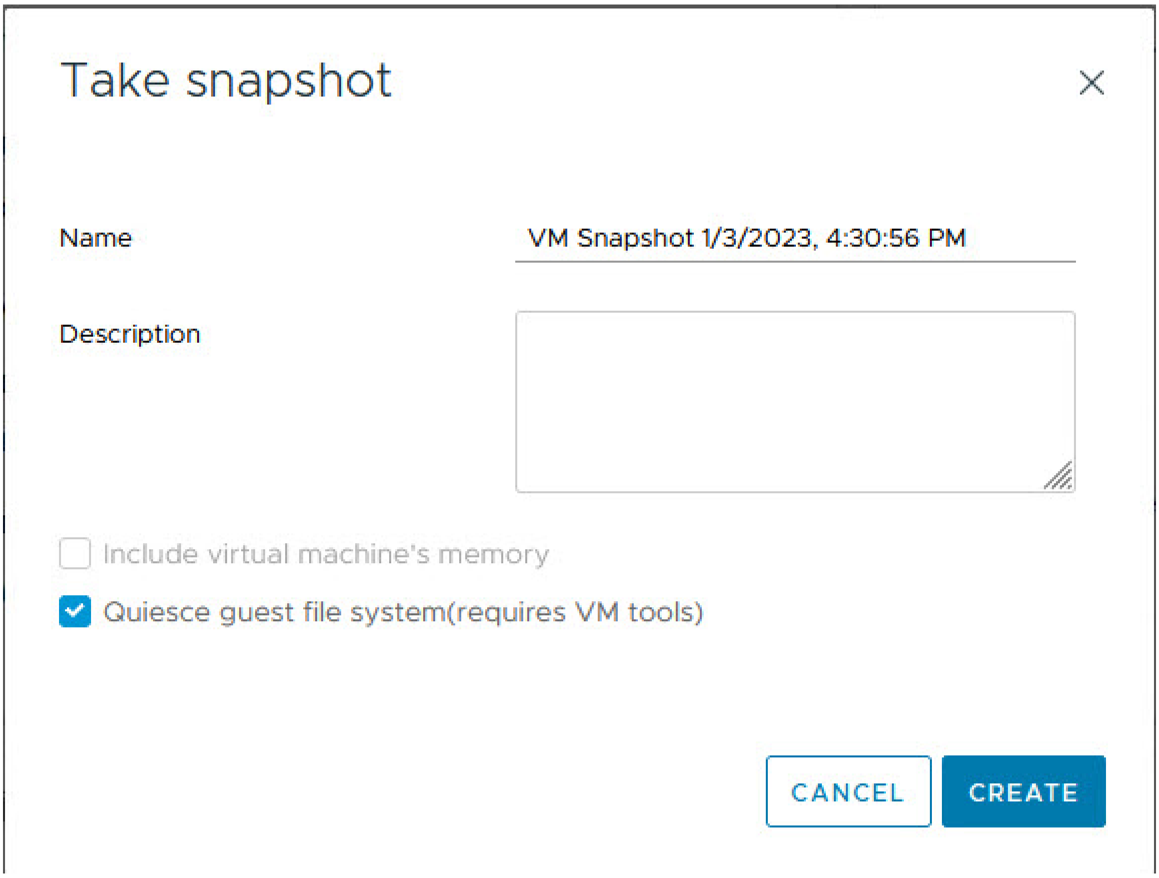
Use multiple PVSCSI adapters. It is supported to configure up to four (4) adapters per VM. Placing OS, data, and transaction logs onto a separate vSCSI adapter optimizes I/O by distributing load across multiple target devices and allowing for more queues on the operating system level. Consider to evenly distributing disks between controllers. vSphere supports up to 64 disks per controller[66].

In vSphere 6.5 the new type of virtual controller was introduced – vNVMe[67]. It has since undergone multiple significant performance enhancements with each vSphere release. NVMe controller might bring performance improvement and reduce I/O processing overhead especially in combination with low latency SSD drives on All-flash storage or Persistent Memory. Consider testing the configuration using a representative copy of your production database to check if this change will be beneficial. Virtual hardware 14 and above are strongly recommended for any implementation of vNVMe controller.

Using Snapshots

A VM snapshot preserves the state and data of a virtual machine at a specific point in time.[68] 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 after creation of a snapshot a special "delta" file is used, which contains a continuous record of the block level changes to the disk[69]. Snapshots are widely used for backup software or by infrastructure administrators and DBAs to preserve the state of a virtual machine before implementing changes (like upgrading the SQL Server application or installing patches).

Figure 61. Take Snapshot Options



Take snapshot

Name: VM Snapshot 1/3/2023, 4:30:56 PM

Description:

□ Include virtual machine's memory

☑ Quiesce guest file system(requires VM tools)

Below are some best practices and considerations for taking snapshots on a VM hosting a SQL Server instance:

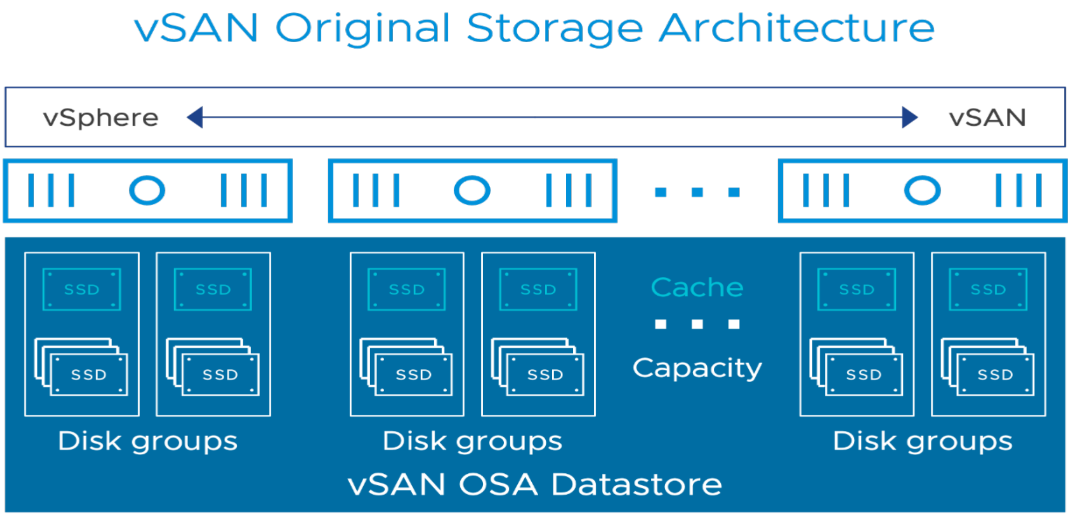
1. Offline snapshot (a VM is powered off when a snapshot is taken) can be used without special considerations.
2. If an online snapshot (VM is powered on and Guest OS is running) needs to be taken:
    a. Consider not to use "Snapshot the virtual machine's memory" option as this may stun a VM[70]. Rely on SQL Server mechanisms to prevent data loss by losing in-memory data.
    b. Use "Quiesce guest file system" option to ensure that a disk-consistent snapshot will be taken. Special notes:
        i. Consider not taking an online snapshot if VMware Tools are not installed or not functional, as this may lead to the inconsistent disk state.
        ii. Consider checking the status of the VSS service on Windows OS before taking a snapshot.
    c. Be aware that on highly active instances of SQL Server that produce a high number of disk operations, snapshot operations (creation of an online snapshot, online removal of the snapshot) may take a long time and can potentially cause performance issues[71]. Consider using snapshots operations for off-peak hours, use offline creation/removal of snapshots, or use vVol technology with the storage array level snapshot integrations.

3. Do not run a VM hosting SQL Server on a snapshot for more than 72 hours[72].
4. Snapshots are not a replacement for a backup: delta disk files contain only references to the changes and not the changes itself.
5. Consider using VMFS6 and SEsparce snapshot for performance improvements.

vSAN Original Storage Architecture (OSA)[73]

vSAN is the VMware software-defined storage solution for hyper-converged infrastructure, a software-driven architecture that delivers tightly integrated computing, networking, and shared storage from x86 servers. vSAN delivers high performance, highly resilient shared storage. 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 62. VMware vSAN Original Storage Architecture



vSAN can be configured as a hybrid or an all-flash storage. In a hybrid disk architecture, vSAN hybrid leverages flash-based devices for performance and magnetic disks for capacity. In an all-flash vSAN architecture, vSAN can use flash-based devices (PCIe SSD or SAS/SATA SSD) for both the write buffer and persistent storage. Read cache is not available nor required in an all-flash architecture. vSAN is a distributed object storage system that leverages the SPBM feature to deliver centrally managed, application-centric storage services and capabilities. Administrators can specify storage attributes, such as capacity, performance, and availability as a policy on a per-VMDK level. The policies dynamically self-tune and load balance the system so that each VM has the appropriate level of resources.

When deploying VMs with SQL Server on a hybrid vSAN, consider the following:

- Build vSAN nodes for your business requirements – vSAN is a software solution. As such, customers can design vSAN nodes from the “ground up” that are customized for their own specific needs. In this case, it is imperative to use the appropriate hardware components that fit the business requirements.
- Plan for capacity – The use of multiple disk groups is strongly recommended to increase system throughput and is best implemented in the initial stage.
- Plan for performance – It is important to have sufficient space in the caching tier to accommodate the I/O access of the OLTP application. 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, in cases where high performance is required for mostly random I/O access patterns, VMware recommends that the SSD size be at least two times that of the working set.

For the SQL Server mission critical user database, use the following recommendations to design the SSD size:

  - SSD size to cache active user database. The I/O access pattern of the TPC-E-like OLTP is small (8 KB dominant), random, and read-intensive. To support the possible read-only workload of the secondary and log hardening workload, VMware recommends having two times the size of the primary and secondary database. For example, for a 100-GB user database, design 2 x 2 x 100 GB SSD size.
  - Select appropriate SSD class to support designed IOPS. For the read-intensive OLTP workload, the supported IOPS of SSD depends on the class of SSD. A well-tuned TPC-E like workload can have ten percent write ratio.

- Plan for availability. Design more than three hosts and additional capacity that enables the cluster to automatically remediate in the event of a failure. For SQL Server mission-critical user databases, enable Always On to put the database in the high availability state when the Always On is in synchronous mode. Setting FTT greater than one means more write copies to vSAN disks. Unless special data protection is required, FTT=1 can satisfy most of the mission-critical SQL Server databases with AlwaysOn enabled.

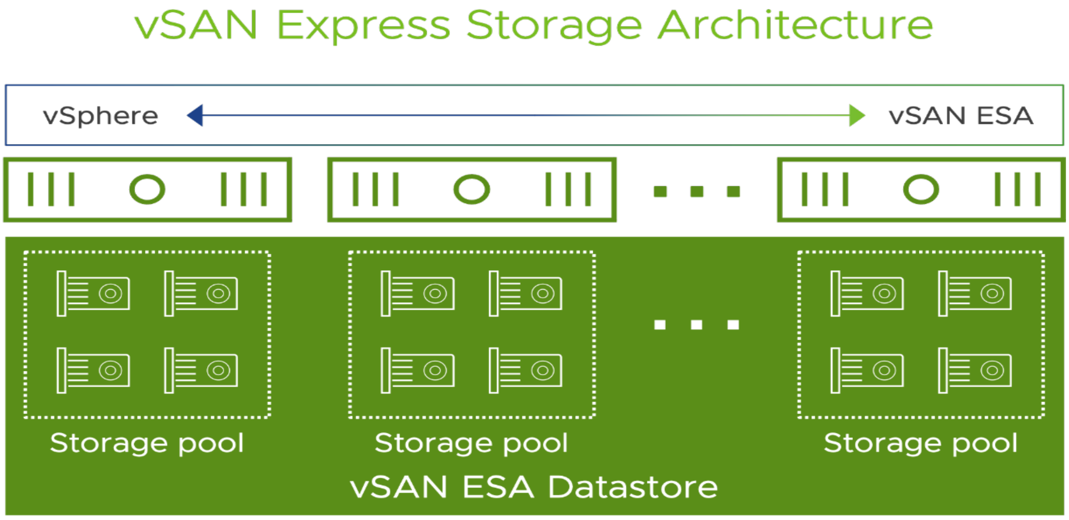
- Set proper SPBM. vSAN SPBM can set availability, capacity, and performance policies per VM:
- Set object space reservation. Set it to 100 percent. The capacity is reserved from the vSAN datastore.
- Number of disk stripes per object. The number of disk stripes per object is also referred to as stripe width. It is the setting of vSAN policy to define the minimum number of capacity devices across which replica of a storage objects is distributed. vSAN can create up to 12 stripes per object. Striping can help performance if the VM is running an I/O intensive application such as an OLTP database. In the design of a hybrid vSAN environment for a SQL Server OLTP workload, leveraging multiple SSDs with more backed HDDs is more important than only increasing the stripe width. Consider the following conditions:
    - If more disk groups with more SSDs can be configured, setting a large stripe width number for a virtual disk can spread the data files to multiple disk groups and improve the disk performance.
    - A larger stripe width number can split a virtual disk larger than 255 GB into more disk components. However, vSAN cannot guarantee that the increased disk components will be distributed across multiple disk groups with each component stored on one HDD disk. If multiple disk components of the same VMDK are on the same disk group, the increased number of components are spread only on more backed HDDs and not SSDs for that virtual disk, which means that increasing the stripe width might not improve performance unless there is a de-staging performance issue.
- Depending on the database size, VMware recommends having multiple VMDKs for one VM. Multiple VMDKs spreads the database components across disk groups in a vSAN cluster.
- In an All Flash vSAN, for read-intensive OLTP databases, such as TPC-E-like databases, the most space requirements come from the data, including table and index, and the space requirement for transaction logs is often smaller versus data size. VMware recommends using separate vSAN policies for the virtual disks for the data and transaction log of SQL Server. For data, VMware recommends using RAID 5 to reduce space usage from 2x to 1.33x. The test of a TPC-E-like workload confirmed that the RAID 5 configuration achieves good disk performance. Regarding the virtual disks for transaction log, VMware recommends using RAID 1.
- VMware measured the performance impact on All-Flash vSAN with different stripe widths. In summary, after leveraging multiple virtual disks for one database that essentially distributes data in the cluster to better utilize resources, the TPC-E-like performance had no obvious improvement or degradation with additional stripe width. VMware tested different stripe widths (1 to 6, and 12) for a 200 GB database in All-Flash vSAN and found:
    - The TPS, transaction time and response time were similar in all configurations.
    - Virtual disk latency was less than two milliseconds in all test configurations.
- VMware suggests setting the stripe width as needed to split the disk objects into multiple components to distribute the object components to more disks in different disk groups. In some situations, you might need this setting for large virtual disks.
- Use Quality of Service for Database Restore Operations. vSAN 6.2 introduces a QoS feature that sets a policy to limit the number of IOPS that an object can consume. The QoS feature was validated in the sequential I/O-dominant database restore operations in this solution. Limiting the IOPS affects the overall duration of concurrent database restore operations. Other applications on the same vSAN that has performance contention with I/O-intensive operations (such as database maintenance) can benefit from QoS.
- vSAN 6.7 expands the flexibility of the vSAN iSCSI service to support Windows Server Failover Clusters (WSFC)[74].
- vSAN 6.7 Update 3 extends support for Windows SQL Server Failover Clusters Instances (FCI) with shared target storage locations exposed using vSAN native for SQL Server[75].

vSAN Express Storage Architecture (ESA) [76]

vSAN 8.0 introduced express storage architecture (ESA) as an optional, alternative architecture in vSAN that is designed to process and store data with all new levels of efficiency, scalability, and performance. This optional architecture is optimized to exploit the full capabilities of the very latest in hardware. vSAN ESA can be selected at the time of creating a cluster.

vSAN 8 ESA evolves beyond the concept of disk groups, discrete caching, and capacity tiers, enabling users to claim storage devices for vSAN into a “storage pool” where all devices are added to a host’s storage pool to contribute to the capacity of vSAN. This will improve the serviceability of the drives and the data availability management and will help drive down costs.

Figure 63. VMware vSAN Express Storage Architecture



vSAN Express Storage Architecture is ideal for customers moving to the latest generation of hardware, while vSAN original architecture is a great way to take advantage of the existing hardware most effectively when trying to upgrade the cluster to latest version. Consider the following aspects when trying to deploy VMs with SQL Server on vSAN Express Storage Architecture:

- Erasure Coding – Express Storage Architecture delivers space efficiency of RAID-5/6 erasure coding at the performance of RAID-1 mirroring. Express Storage Architecture is recommended for a compromise of both capacity and performance consideration for SQL Server data files, transaction logs and TempDB files as well.
- Space Efficiency (compression) – For capacity sensitive cases of SQL Server databases, Express Storage Architecture achieves better compression ratio compared to compression-only feature of Original Storage Architecture. It also allows policy-based data compression for SQL Server virtual disks with smaller granularity.
- Snapshots – Express Storage Architecture delivers extremely fast and consistent performance with the new native scalable snapshots feature. It enables VM-based snapshot backup solutions possible for SQL Server VMs with minimal performance overhead.

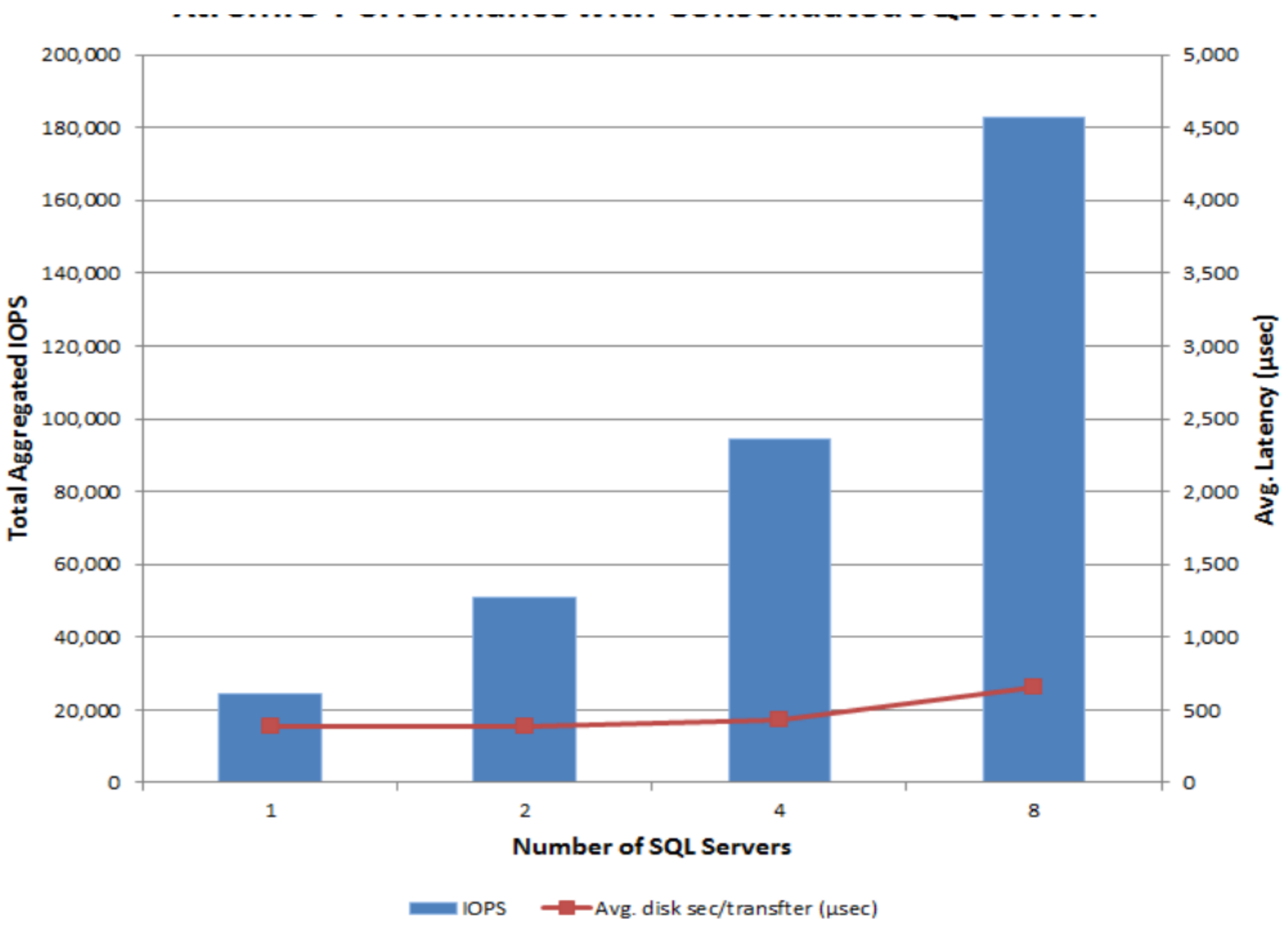
Considerations for Using All-Flash Arrays

All-flash storage is gaining increasing popularity in corporate data centers, typically because of performance, but the latest generation of all-flash storage also offers:

- Built-in data services, such as, thin provisioning, inline data deduplication, and inline data compression that provide compelling data reduction ratio.
- Flash-optimized data protection that replaces traditional RAID methodologies can simplify the database server sizing and capacity planning efforts while minimizing protection overhead and performance penalty.
- Instant space-efficient copies through VSS integration that significantly increases efficiency and operational agility for SQL Server and can be used for local data protection.

From a performance perspective, the ability to maintain consistent sub-millisecond latency under high load, and to scale linearly in a shared environment drives more and more interest in all-flash arrays. In a study of SQL Server on XtremIO performed by EMC, EMC ran eight SQL Server workloads on a dual X-Brick XtremIO cluster. Each of the OLTP-like workloads simulates a stock trading application, and generates I/O activities of a typical SQL Server online transaction workload of 90 percent read and 10 percent write. As the number of SQL Server instances increases from 1, 2, 4, and 8, the total aggregated IOPS increases from 22 K, 45 K, 95 K, and 182 K respectively, while maintaining about 500 µs consistent latency.

Figure 64. XtremIO Performance with Consolidated SQL Server



For more information about the study, see the Best Practices for Running SQL Server on EMC XtremIO document at http://www.emc.com/collateral/white-paper/h14583-wp-best-practice-sql-server-xtremio.pdf.

When designing SQL Server on all-flash array, there are considerations for storage and file layout which differ from traditional storage systems. This section refers to two aspects of the all-flash storage design:

- RAID configuration
- Separation of SQL Server files

Raid Configuration

When deploying SQL Server on an All-Flash arrays, traditional RAID configuration considerations are no longer relevant, and each vendor has its own proprietary optimizations technologies to consider. Taking XtremIO as an example, the XtremIO system has a built-in "self-healing" double-parity RAID as part of its architecture. The XtremIO Data Protection (XDP) is designed to take advantages of flash-media-specific properties, so no specific RAID configuration is needed.

Separation of Files

A very common storage I/O optimization strategy for an I/O-intensive, transactional SQL Server workload is to logically separate the various I/O file types (TempDB, data and logs) into as many multiple volumes, disks, LUNs and even physical disk groups at the array level as possible. The main rationale for this historical recommendation is the need to make the various I/O types parallel to reduce latencies, enhance responsiveness, and enable easier management, troubleshooting, and fault isolation.

All-flash storage arrays introduce a different dimension to this recommendation. All-flash arrays utilize solid state disks (SSDs), which typically have no moving parts and, consequently, do not experience the performance inefficiencies historically associated with legacy disk subsystems. The inherent optimized data storage and retrieval algorithm of modern SSD-backed arrays makes the physical location of a given block of data on the physical storage device of less concern than on traditional storage arrays. Allocating different LUNs or disk groups for SQL Server data, transaction log, and TempDB files on an all-flash array does not result in any significant performance difference on these modern arrays.

Nevertheless, VMware recommends that, unless explicitly discouraged by corporate mandates, customers should separate the virtual disks for the TempDB volumes allocated to a high-transaction SQL Server virtual machine on vSphere, even when using an all-flash storage array. The TempDB is a global resource that is shared by all databases within a SQL Server instance. It is a temporary workspace that is recreated each time a SQL Server instance starts. Separating the TempDB disks from other disk types (data or logs) allows customers to apply data services (for example, replication, disaster recovery and snapshots) to the database and transaction logs volumes without including the TempDB files which are not required in such use cases.

Additional considerations for optimally designing the storage layout for a mission-critical SQL server on an all-flash array vary among storage vendors. VMware recommends that customers consult their array vendors for the best guidance when making their disk placement decisions.

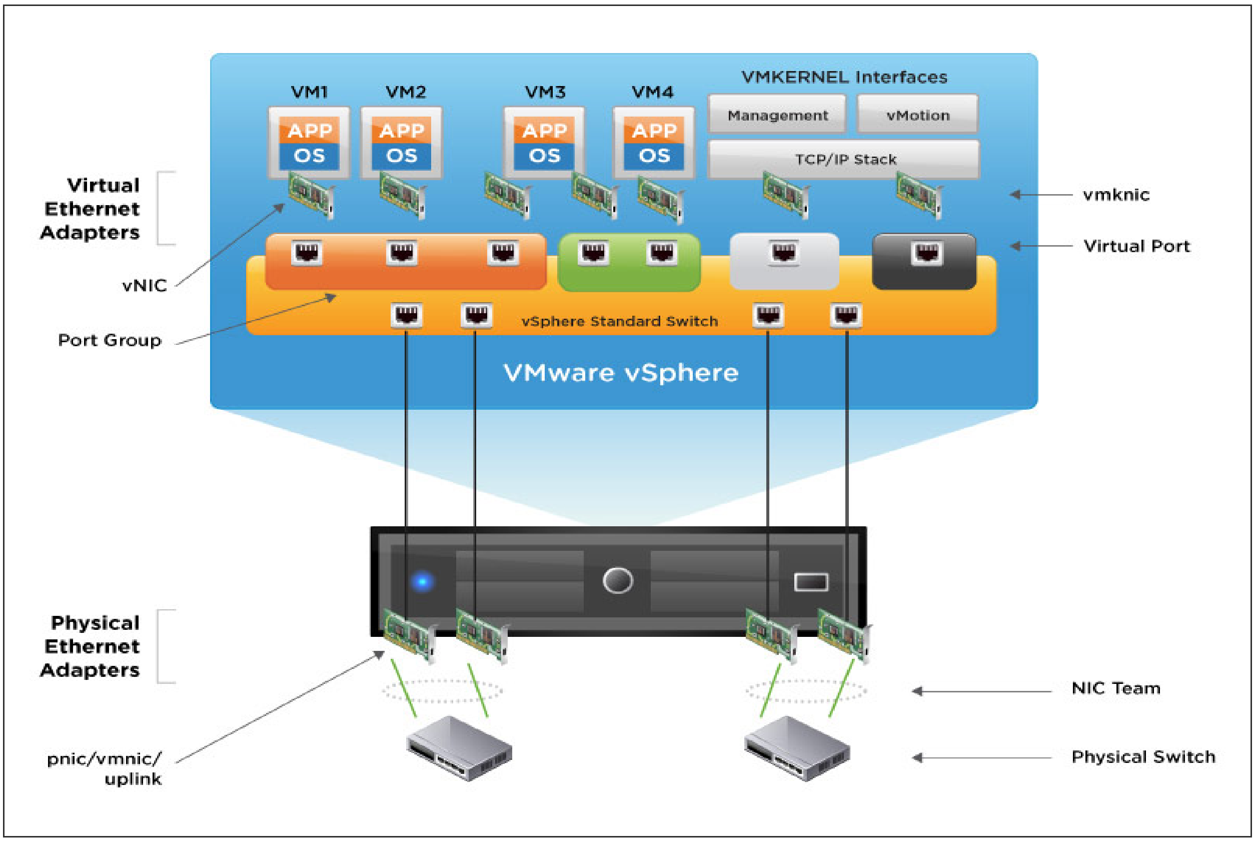
Virtual Machine Network Configuration

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.

Virtual Network Concepts

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

Figure 65. Virtual Networking Concepts



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.

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 and/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 operating system networking considerations and hardware networking considerations in the *Performance Best Practices for vSphere 6.5* guide https://www.vmware.com/techpapers/2017/Perf_Best_Practices_vSphere65.html

Using Multi-NIC vMotion for High Memory Workloads

vSphere 5.0 introduced a new feature called Stun during Page Send (SDPS), which 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, and 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 64 GB 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 for a single VM vMotion operation. This speeds up vMotion operation and reduces the risk for SDPS on large, memory intensive VMs.

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

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

Check https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/performance/vmotion-7u1-perf.pdf for more information.

Figure 66. vMotion of a Large Intensive VM with SDPS Activated

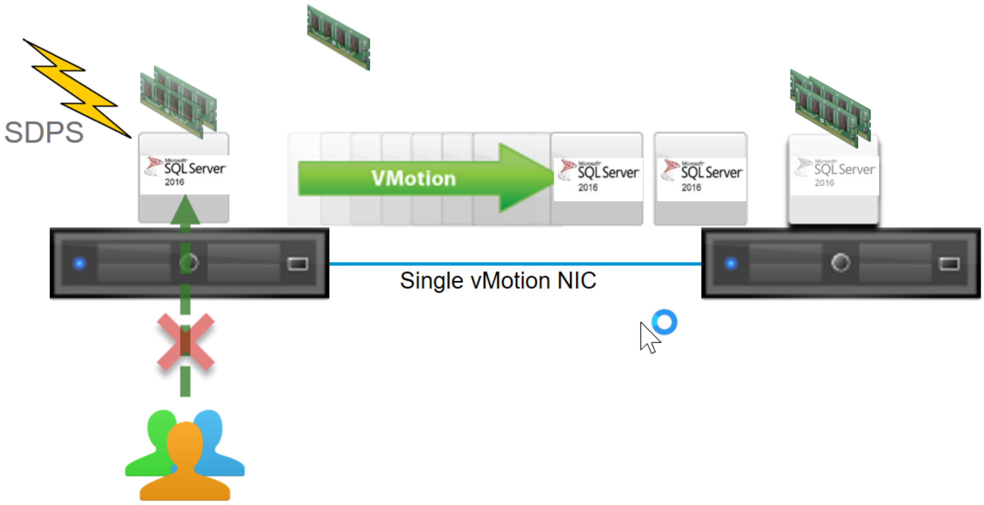
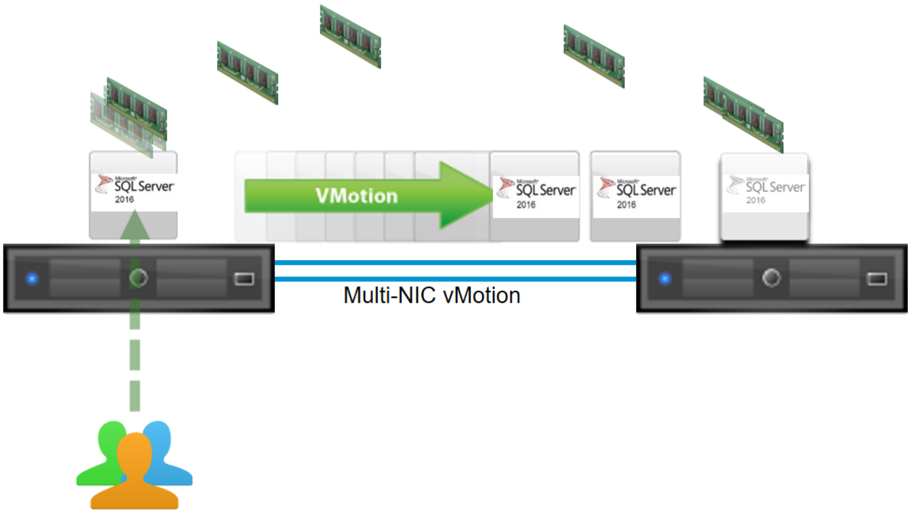


Figure 67. Utilizing Multi-NIC vMotion to Speed Up vMotion Operation



vSphere 7.0 introduced multiple vMotion enhancement features to dramatically reduce the performance impact during the live-migration and stun time. These features allow large virtual machines – also referred to as “Monster” VMs with large CPU and memory configuration of SQL Server to be live-migrated with minimized performance degradation during a vMotion event. For more details of this performance improvement, refer to vMotion to take advantage of high-speed bandwidth NICs, improving large memory VM page copies, and improvements in the fast suspend and resume process for Storage vMotion.

In most cases, SQL Server instance, or even Failover Cluster Instances on a Monster VM, will not be impacted during a vMotion event with minimal performance overhead. There’s nothing SQL Server users or DBAs need to do during this action. However, vSphere 8.0 introduced vMotion notification feature that helps those latency sensitive and clustering applications which cannot tolerate a vMotion operation’s one-second downtime SLA. In case SQL Server is implemented in a latency sensitive use case and requires taking action before a vMotion event happens, vMotion Notification can be helpful for those SQL Server virtual machines.

vMotion Notification requires vSphere to be running version 8 and above and virtual machine to be using hardware version 20. For the guest operating system, VMware Tools or Open VM tools must be installed and running with a minimum version 11.0. Applications need to use the VMware Tools vmtoolsd command line utility to register for notification. Here’s sample demo video for vMotion Notification. For more details, refer to vSphere vMotion Notifications.

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. During testing, VMware tested vSphere vMotion migrations 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 sensitivity 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.

vSphere Security Features

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

Virtual Machine Encryption[77]

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 form the storage. VM encryption is controlled on a per VM basis and is implemented in the virtual vSCSI layer using the 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.

vSphere Security Features[78]

vSphere 6.7 introduced a number of enhancements that help lower security risks in the vSphere infrastructure for a VMs hosting SQL Server. These include:

- Support for a virtual Trusted Platform Module (vTPM) for the virtual machine
- Support for Microsoft Virtualization Based Security[79]
- Enhancement for the ESXi "secure boot"
- Virtual machine UEFI Secure Boot
- FIPS 140-2 Validated Cryptographic Modules turned on by default for all operations

With the release of vSphere 8.0, additional security improvements have continued to be added to the Platform, including automatic encryption of ESXi sensitive files, secure boot for ESXi Host, deprecation of TLS 1.0 and 1.1, automatic SSH session timeout, discontinuation of TPM1.2 support, among other.

Maintaining a Virtual Machine

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

Upgrade VMware Tools[80]

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 operating systems. 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. Bug fixes, incompatibility or stability issues, security fixes and other enhancements are delivered to the VM through the facility of the VMware Tools. It is, therefore, critical that Customers ensure that they regularly upgrade or update VMware Tools for their production VMs in their vSphere infrastructure.

VMware Tools and other VM-related drivers are now available through Windows Update. This significantly reduces the complexities associated with manually updating them. VMware strongly encourages Customers to take steps to incorporate VMware Tools servicing through Windows Update into their standard lifecycle management and administrative practices.

Upgrade the Virtual Machine Compatibility [81]

The virtual machine compatibility determines the virtual hardware available to the virtual machine, 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 virtual machine[82]. 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 define maximum CPU instruction set exposed to a VM: 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. Following procedure is recommended[83]:

- Take a backup of the SQL Server databases and Operating System
- Upgrade VMware Tools
- Validate that no misconfigured/inaccessible devices (like CD-ROM, Floppy) are present
- Use vSphere Web Client to upgrade to the desired version
- **Upgrading a Virtual Machine to the latest hardware version is the physical equivalent of swapping the old mainboard on a physical system and replacing it with a newer one. Its success will depend on the resiliency of the guest operating system in the face of hardware changes. VMware does not recommend upgrading the virtual hardware version if you do not need the new features exposed by the new version. However, you should be aware that newer enhancements and capabilities added to more recent virtual hardware versions are not generally backported to older hardware versions.**

SQL Server on VMware-powered Hybrid Clouds

After migrating a Virtual Machine (VM) hosting SQL Server workloads to VMware Cloud on AWS make sure to check VM configuration settings to ensure better operations and performance of your workload. The list below should not be treated as a full list of configurations recommendations but rather depicts the configuration items that might be affected due to the migration to a new environment on VMware Cloud on AWS.

VMware vSAN is the technology providing the storage resource in VMware Cloud on AWS. Therefore, migrating to VMware Cloud on AWS might require revising the current virtual disk design of VMs hosting SQL Server workloads to achieve the best performance running on vSAN.

Note: You can use the set of recommendations created by the VMware vSAN and SQL Server experts for most of the optimization tasks. Bear in mind that these recommendations are created for on-premises deployments, and not all of them could apply to a managed service like VMware Cloud on AWS due to the nature of the environment. The bullet points below supersede the recommendations in the article.

The following configuration items should be considered:

- Use the PVSCSI virtual controller type to attach virtual disks hosting SQL Server related data (including logs and tempdb) to achieve the best throughput. Do not use the LSI Logic SAS controller type.

- Use multiply PVSCSI controllers (up to four) to balance the disk throughput between controllers.
- Consider a multiple VMDK disk layout to redistribute load between vSAN nodes. This is especially important as vSAN is much more efficient with smaller disks, so a VM with multiple small in size VMDKs distributed between multiple vSCSI controllers is expected to perform better compared to a VM with the same workload but using just a single VMDK on a single vSCSI Controller.
- We strongly advise using RAID1 for SQL Server transaction log and tempdb disks.
- RAID1 should be your primary choice for SQL Server database files if the performance of SQL Server is the main goal of your design.
- Consider setting the Object Space Reservation (OSR) Advanced Policy Setting to “Thin provisioning”. OSR controls only the space reservation and has no performance advantages. While the control of the capacity is still very important for on-premises solutions, on VMware Cloud on AWS Elastic DRS (eDRS) ensures the cluster will not run out of free capacity. You can check this blog article for more details.
- Make sure to understand the applicable VMware Cloud on AWS Configuration Maximus while planning, sizing, and running your SDDC hosting Mission Critical Applications. While many configuration maximums (or minimus) are the same as on-premises, some of them might influence the way how your SDDC should be designed.
- 2. Make sure to consult the list of unsupported VM configurations to ensure that a VM can be started on/moved to VMware Cloud on AWS.

The following additional configuration settings are strongly advisable for all SQL Server on VMware Cloud on AWS:

- Set T1800 trace flag. T1800 trace flag forces 4K IO alignment for SQL Server transaction log. vSAN efficiently greatly improves with 4K aligned IO. We recommend that you enable global trace flags at startup, by using the -T command-line startup option. This ensures the trace flag remains active after a server restart. Restart SQL Server for the trace flag to take effect. You can use procmon system utility to check the IO to make sure that the trace flag is properly enabled on your SQL Server.
- Dedicate separate disks for SQL Server transaction log. Use multiple disks spread between multiple SCSI controllers if you have multiple databases.
- Dedicate separate disks for temdb. We recommend using four VMDKs spread between four SCSI controllers with each VMDK hosting two tempdb files (with a total of eight tempdb files per SQL Server instance)
- Use Database File Group with multiple files. Depending on your database design you can either use multiple File Groups or create multiple files inside of a single primary database group. SQL Server writes parallel to all files within a file grop.
- Avoid cross-region and hybrid (on-premises to SDDC) traffic flow. Ensure that the apps and all components using the database are located within the same cluster in your SDDC. Take care of your SSIS deployment. SSIS server executing packages should be located within the same SDDC as the source and target SQL server database.

You can learn more about running SQL Server on VMware Cloud on AWS in this document.

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 layer 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 review for a vSphere virtualized environment.

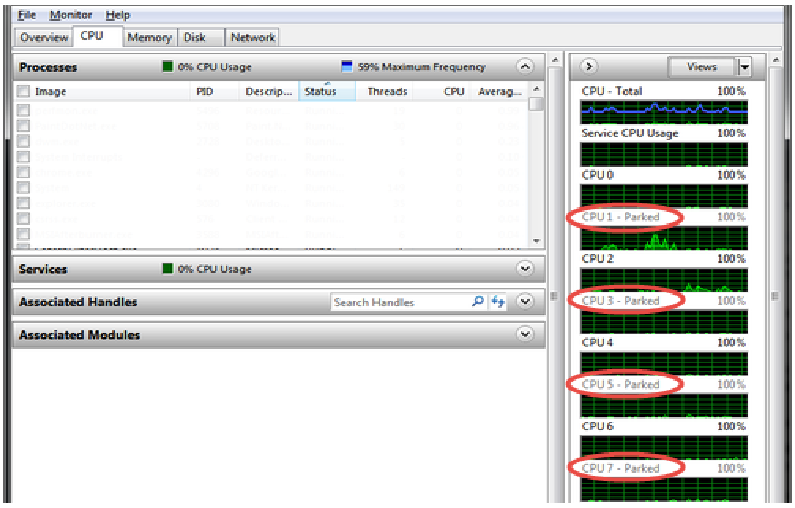
Windows Server Configuration[84]

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

Power Policy[85]

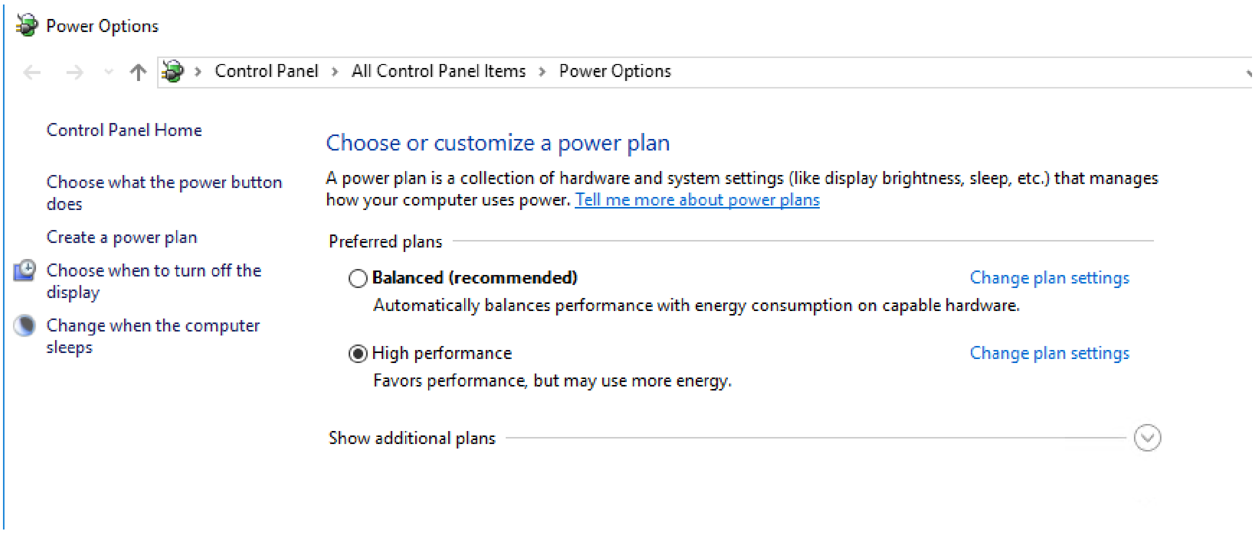
The default power policy option in Windows Server is "Balanced". This configuration allows Windows Server OS to save power consumption by periodically throttling power to the CPU and turning off devices such as the network cards in the guest when Windows Server determines that they are idle or unused. 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 68. 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 practices for virtualized deployment.

Figure 69. Recommended Windows OS Power Plan


Power Options
Control Panel › All Control Panel Items › Power Options
Control Panel Home
Choose what the power button does
Create a power plan
Choose when to turn off the display
Change when the computer sleeps

Choose or customize a power plan

A power plan is a collection of hardware and system settings (like display brightness, sleep, etc.) that manages how your computer uses power. Tell me more about power plans

Preferred plans

○ **Balanced (recommended)** Change plan settings

Automatically balances performance with energy consumption on capable hardware.

● High performance Change plan settings

Favors performance, but may use more energy.

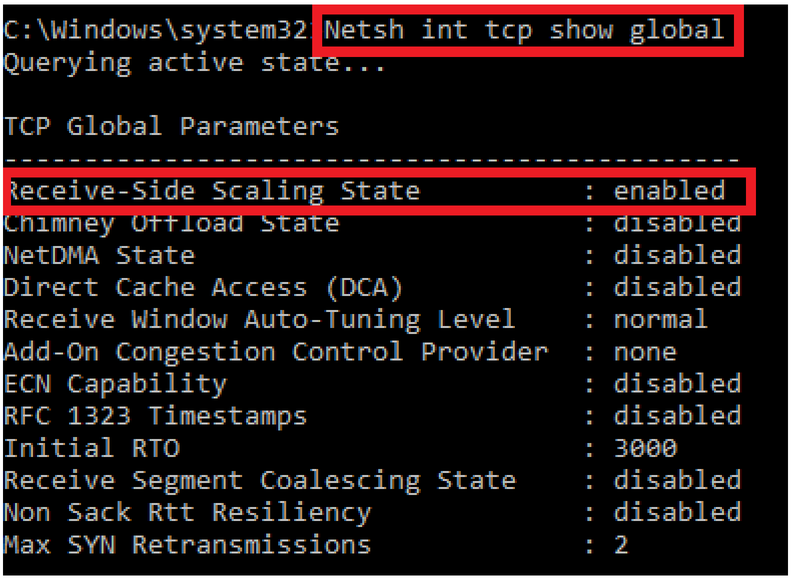
Show additional plans

Enable Receive Side Scaling (RSS)[86]

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 configured in the following two places:

- Enable RSS in the Windows kernel by running the netsh 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.

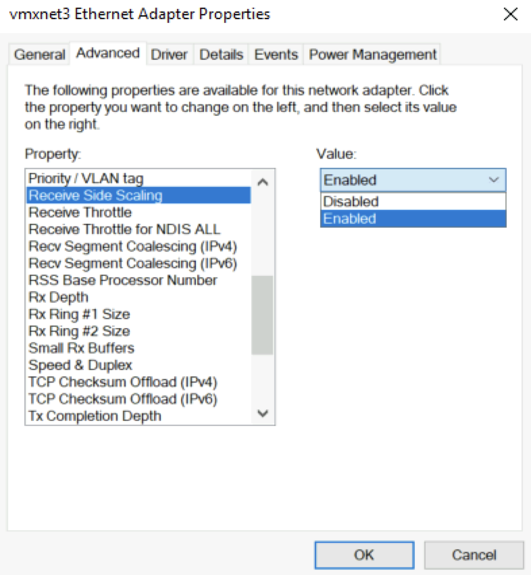
Figure 70. Enable RSS in Windows OS



```
C:\Windows\system32>Netsh int tcp show global
Querying active state...

TCP Global Parameters
----------------------------------------------
Receive-Side Scaling State          : enabled
Chimney Offload State               : disabled
NetDMA State                        : disabled
Direct Cache Access (DCA)           : disabled
Receive Window Auto-Tuning Level    : normal
Add-On Congestion Control Provider  : none
ECN Capability                      : disabled
RFC 1323 Timestamps                 : disabled
Initial RTO                         : 3000
Receive Segment Coalescing State    : disabled
Non Sack Rtt Resiliency             : disabled
Max SYN Retransmissions             : 2
```

Enable RSS on the VMXNET network adapter driver.[87] 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 71. Enabling RSS on Network Interface Card in Windows



vmxnet3 Ethernet Adapter Properties

The following properties are available for this network adapter. Click the property you want to change on the left, and then select its value on the right.

Property:

- Priority / VLAN tag
- Receive Side Scaling
- Receive Throttle
- Receive Throttle for NDIS ALL
- Recv Segment Coalescing (IPv4)
- Recv Segment Coalescing (IPv6)
- RSS Base Processor Number
- Rx Depth
- Rx Ring #1 Size
- Rx Ring #2 Size
- Small Rx Buffers
- Speed & Duplex
- TCP Checksum Offload (IPv4)
- TCP Checksum Offload (IPv6)
- Tx Completion Depth

Value: Enabled

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.

Configure PVSCSI Controller

Windows Operating systems do not include the driver for the PVSCSI controller. VMware Tools need to be installed to provide the driver for PVSCSI device[88]

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 Windows Server up 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"[89].

- **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[90]. 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.**

Using Antivirus Software[91]

Customers might have requirements that antivirus scan software must run on all servers, including those running SQL Server. 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. More information can be found at the following link.

https://support.microsoft.com/en-us/topic/how-to-choose-antivirus-software-to-run-on-computers-that-are-running-sql-server-feda079b-3e24-186b-945a-3051f6f3a95b

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.

SQL Server Configuration

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 memory on the VM. Setting the maximum server memory allows you to reserve sufficient memory for the operating system 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 server memory is a good practice to maintain SQL Server performance if under host memory pressure. When running SQL Server on vSphere, if the vSphere host is under memory pressure, the balloon driver might inflate and reclaim memory 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 operating system 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
```

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, which enables SQL Server to manage the reduction of its own working set.

Any time Lock Pages in Memory is used, because the 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.

If you are deploying a Tier-1 mission-critical SQL Server installation, consider setting the Lock Pages in Memory user right[92] 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.

Large Pages[93]

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[94]...

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. Note: SQL Server 2019 introduced trace flag 876 when columnstore indexing is used and the workload will benefit from large memory pages.
- With large pages enabled in the guest operating system, 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 start-up behaviour changes. Instead of allocating memory dynamically at runtime, SQL Server allocates all buffer pool memory during start-up. Therefore, SQL Server start-up time can be significantly delayed.
- With trace flag 834 enabled, SQL Server allocates memory in 2 MB 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, if the practice of memory overcommitment is common, 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.
```

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.

MAXDOP, or maximum degree of parallelism, is an advanced configuration option that controls the number of processors used to execute a query in a parallel plan. Setting this value to 1 disables parallel plans altogether. The default value is 5, which is usually considered too low.

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 seconds and the default is 5, which means a parallel plan for queries is used if SQL Server determines that it would take longer than 5 seconds when run serially. 5 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 configurations in a virtual environment. When low performance is observed on their database, and CXPACKET is high, many DBAs decide to disable parallelism altogether by setting MAXDOP value to 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 5 seconds to approximately 50 seconds to make sure only large queries run in parallel. Set the MAXDOP according to Microsoft's recommendation for the number of cores in the VM's NUMA node (no more than 8).

You can also set the MAXDOP to 1 and set a MAXDOP = N query hint to set parallelism in the query code. In any case, the configuration of these advanced settings is dependent on the front-end application workload using the SQL Server.

To learn more, see the Microsoft article Recommendations and guidelines for the "max degree of parallelism" configuration option in SQL Server at https://support.microsoft.com/en-us/kb/2806535.

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 software-defined data center (SDDC). This section reviews some of the VMware products that can be used in virtualized SQL Server VMs on vSphere.

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.

VMware Aria Operations

Maintaining and operating virtualized SQL Server is the vital part of the infrastructure lifecycle. It is 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 the database objects through the 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

The VMware vRealize® True Visibility™ Management Pack for Microsoft SQL Server[95] is one of the commercially-available SQL Server-aware monitoring tools that is able to fulfil all the requirements mentioned above.

The VMware vRealize® True Visibility™ Management Pack for Microsoft SQL Server is an embedded adapter for VMware Aria Operations (formerly named vRealize Operations (vROps)), collecting performance data from your Microsoft SQL Server environment and providing predictive analytics and real-time information about problems in your infrastructure.

When performance or capacity problems arise in your SQL Server environment, vRealize True Visibility Suite 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 True Visibility Suite can help minimize false alerts and proactively alert on the potential root cause of increasing performance problems before end users are impacted.

Resources

SQL Server on VMware vSphere:

- Architecting Business Critical Applications on VMware Hybrid Cloud
- Microsoft SQL Server and VMware Cloud on AWS: Design, Migration, and Configuration
- https://docs.vmware.com/en/VMware-Cloud-on-AWS/solutions/VMware-Cloud-on-AWS.919a954a9b6ca17cdc719ec42cda1401/GUID-E62521730EDBE3DC125813A448BA3B45.htmlMicrosoft SQL Server on VMware vSphere® Availability and Recovery Options
- Performance characterization of Microsoft SQL Server on VMware vSphere 6.5
- Planning highly available, mission critical SQL server deployments with VMware vSphere

VMware Blogs:

- The VMware Workloads Team Blog
- Cornac Hogan, When and why do we “stun” a virtual machine?
- Frank Denneman. NUMA Deep Dive Series
- VMware’s Microsoft SQL Server Blog Posts
- VMware Performance Team Blog Posts
- VMware vSphere Blog Posts

VMware Knowledgebase:

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

VMware Documentation

- SQL Server FCI and File Server on VMware vSAN 6.7 using iSCSI Service
- Understanding Memory Management in VMware vSphere
- VMware Tools
- VMware vCenter Server and Host Management
- VMware vSphere virtual machine encryption performance VMware vSphere 6.5
- VMDK versus RDM
- vSphere Security. vSphere 8.0

SQL Server Resources

- Compute capacity limits by edition of SQL Server
- Description of support for network database files in SQL Server
- Editions and supported features of SQL Server
- 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

- Memory Management Architecture Guide
- Performance Center for SQL Server Database Engine and Azure SQL Database
- Soft-NUMA (SQL Server)
- How It Works: How SQL Server Determines Logical and Physical Processors
- It Just Runs Faster: Automatic Soft NUMA
- SQL Server and Large Pages Explained
- Transaction Commit latency acceleration using Storage Class Memory in Windows Server 2016/SQL Server 2016 SP1
- Virtualization-based Security (VBS)

Acknowledgments

Authors:

- Deji Akomolafe – Staff Solutions Architect, Microsoft Applications Practice Lead
- Mark Xu - Sr. Technical Marketing Architect
- Oleg Ulyanov - Staff Cloud Solutions Architect

Thanks to the following people for their contributions:

- Catherine Xu - Group Manager, Workload Technical Marketing
- David Klee - Founder and Chief Architect, Microsoft MVP and vExpert at Heraflux Technologies

---

[1] Further in the document referenced as SQL Server

[2] Source: vSphere Share Tracker Survey 2017

[3] vSphere vMotion Notifications for Latency Sensitive Applications - https://docs.vmware.com/en/VMware-vSphere/8.0/vsphere-vcenter-esxi-management/GUID-0540DF43-9963-4AF9-A4DB-254414DC00DA.html

[4] . 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

[5] This feature is deprecated as of SQL Server 2012 and should not be used when possible. Consider using Always on Availability Groups instead.

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

[7] Licensing considerations are addressed here: https://www.vmware.com/products/vmc-on-aws/microsoft-licensing.html

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

[9] https://www.vmware.com/resources/compatibility/search.php?[10]
https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/support/product-lifecycle-matrix.pdf

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

[12]https://www.windowsservercatalog.com/results.aspx?&bCatID=1521&cpID=11779&avc=0&ava=136&avt=0&avq=0&OR=1&PGS=25

[13] https://learn.microsoft.com/en-us/troubleshoot/windows-server/virtualization/non-microsoft-hardware-virtualization-software

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

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

[16] 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

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

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

[19] 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”.

[20] Virtual CPU Configuration and Limitations:

https://docs.vmware.com/en/VMware-vSphere/8.0/vsphere-vm-administration/GUID-3CDA4DEF-3DE0-4A64-89C7-F31BB77222CB.html

[21] Hyperthreading: https://docs.vmware.com/en/VMware-vSphere/8.0/vsphere-resource-management/GUID-FD71CBCA-E97C-4EFA-8A1B-32C09D5DF2A1.html

[22] Intel Hyper-Threading Technology (HT) - https://en.wikipedia.org/wiki/Hyper-threading

[23] See additional information about Hyper-threading on a vSphere host in VMware vSphere Resource Management

[24] It’s recommended to review the publication: http://frankdenneman.nl/2016/07/07/numa-deep-dive-part-1-uma-numa/

[25] The figure is cited from: https://software.intel.com/en-us/articles/optimizing-applications-for-numa

[26] 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.

[27] https://docs.vmware.com/en/VMware-vSphere/8.0/vsphere-resource-management/GUID-BD4A462D-5CDC-4483-968B-1DCF103C4208.html

[28] See https://frankdenneman.nl/2022/11/03/vsphere-8-cpu-topology-for-large-memory-footprint-vms-exceeding-numa-boundaries/ for more detailed description and information

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

[30] MS SQL Enterprise edition is required to utilize NUMA awareness. https://docs.microsoft.com/en-us/sql/sql-server/editions-and-components-of-sql-server-2016?view=sql-server-2017

[31] 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

[32] 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)

[33] https://frankdenneman.nl/numa/

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

[35] https://docs.vmware.com/en/VMware-vSphere/8.0/vsphere-vm-administration/GUID-EE6F4E5A-3BEA-43DD-9990-DBEB0A280F3A.html?hWord=N4IghgNiBcIKYDcDGIC+Q

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

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

[38] For a full list of the supported OSes, check https://learn.microsoft.com/en-us/sql/linux/sql-server-linux-release-notes-2022?view=sql-server-ver16#supported-platforms

[39] Numactl is not available in the standard OS and should be installed using the applicable OS tools or commands

[40] 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

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

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

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

[44] 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

[45] 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

[46] Understanding Memory Overhead

[47] 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.

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

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

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

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

[52] 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/

[53] 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

[54] https://www.youtube.com/watch?v=jDMt9UWAPJQ&feature=youtu.be

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

[56] More details: https://support.microsoft.com/en-us/help/304261/description-of-support-for-network-database-files-in-sql-server

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

[58] For more details, see Performance Characterization of VMFS and RDM Using a SAN (

https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/performance_char_vmfs_rdm.pdf)

[59] https://docs.vmware.com/en/VMware-vSphere/8.0/vsphere-storage/GUID-710B8E1E-9B33-4229-B974-C51A316C3256.html?hWord=N4IghgNiBcIMYQK4GcAuBTATugJgAgDcBbHAaxAF8g

[60]https://configmax.esp.vmware.com/guest?vmwareproduct=vSphere&release=vSphere%208.0&categories=1-0

[61] https://docs.vmware.com/en/VMware-vSphere/8.0/vsphere-windows-server-failover/GUID-97B054E2-2EB0-4E10-855B-521A38776F39.html

[62] https://docs.vmware.com/en/VMware-vSphere/8.0/vsphere-windows-server-failover/GUID-04626D3C-A305-40BE-A7B9-4E7C7A30BA3D.html

[63] 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

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

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

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

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

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

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

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

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

[72] Depending on a workload and an environment this recommendation may vary, but in general should not exceed 72 hours

[73] More technical materials on SQL Server on vSAN can be found here. For more details: https://core.vmware.com/business-critical-application-reference-architectures

[74] 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/

[75] https://core.vmware.com/resource/sql-server-failover-cluster-instance-vmware-vsan-native

[76] https://blogs.vmware.com/virtualblocks/2022/08/30/announcing-vsan-8-with-vsan-express-storage-architecture/

[77] 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.

[78] Consult this document for a full description: https://docs.vmware.com/en/VMware-vSphere/8.0/vsphere-esxi-vcenter-80-security-guide.pdf

[79] 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/

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

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

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

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

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

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

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

[87] 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 than 10.2.5, steps listed in this document are required in order to ensure or confirm that RSS is enabled on the network card.

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

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

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

[91] See KB309422, *How to choose antivirus software to run on computers that are running SQL Server*.

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

[93] Refer to SQL Server and Large Pages Explained (

https://techcommunity.microsoft.com/t5/sql-server-support-blog/sql-server-and-large-pages-explained-8230/ba-p/315787) for additional information on running SQL Server with large pages.

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

[95] https://docs.vmware.com/en/VMware-vRealize-True-Visibility-Suite/1.0/microsoft-sql-server/GUID-21C00D61-5E63-4723-AD2A-56E8D3971C47.html