TECHNICAL WHITE PAPER
September 2026

Architecting Microsoft SQL Server on VMware Cloud Foundation®

BEST PRACTICES GUIDE

SEPTEMBER 2026

Table of contents

1. Introduction 5
   1.1 Purpose 6
   1.2 Target Audience 6
2. SQL Server Requirements Considerations 7
   2.1 Understand the SQL Server Workload 7
   2.2 Availability and Recovery Options 8
   2.3 SQL Server on VCF Supportability Considerations 9
3. Best Practices for Deploying SQL Server Using VCF 11
   3.1 Right-Sizing 11
   3.2 vCenter Server Configuration 12
   3.3 ESXi Cluster Compute Resource Configuration 12
   3.4 ESXi Host Configuration 23
   3.5 Virtual Machine CPU Configuration 25
   3.7 Virtual Machine Storage Configuration 59
   3.8 Storage Best practices 65
   3.9 Virtual Machine Network Configuration 74
   3.10 VCF Security Features 77
   3.11 Maintaining a Virtual Machine 78
   3.12 Deploying SQL Server with VMware Data Services Manager 79
4. SQL Server and In-Guest Best Practices 87
   4.1 Windows Server Configuration 87
   4.2 SQL Server Configuration 91
5. VMware Enhancements for Deployment and Operations 94
   5.1 Network Virtualization with VMware NSX for VCF 94
6. Resources 94
7. Acknowledgments 95

List of Figures

Figure 1. Percent of Customers Operating the Virtualized Instance of Applications 5
Figure 2. vCenter Server Statistics 12
Figure 3. vSphere HA Settings 13
Figure 4. vSphere Admission Control Settings 14
Figure 5. Proactive HA 14
Figure 6. vSphere DRS Cluster 15
Figure 7. Pre-Defined VMware EVC Settings 21
Figure 8 - Custom VMware EVC Settings 21

Figure 9. Recommended ESXi Host Power Management Setting 25
Figure 10. Physical Server CPU Allocation 26
Figure 11. Virtual Machine CPU Configuration 27
Figure 12. Setting Latency Sensitivity on a VM 28
Figure 13. "High With Hyperthreading" Provides Exclusive CPU Access 29
Figure 14. Intel based NUMA Hardware Architecture 30
Figure 15. Using esxcli or sched-stats to Obtain NUMA Node Count on ESXi Host 35
Figure 16. Using esxtop to Obtain NUMA Related Information on an ESXi Host 35
Figure 17. When is My Considered “Wide”? 38
Figure 18. ESXi Creates UMA Topology When VM’s vCPUs Fits a NUMA Node 38
Figure 19. Letting ESXi Determine vNUMA Topology Automatically for Non-Wide VM 39
Figure 20. Letting ESXi Determine vNUMA Topology Automatically for Memory-wide VM 39
Figure 21. ESXi Doesn't Consider Memory Size in vNUMA Configuration 40
Figure 22. Memory-wide VM, As Seen by Windows 40
Figure 23. Automatic vNUMA Selection Creates Unbalanced Topology on Memory-wide VM 40
Figure 24. Manual vNUMA Configuration Options in VMware Cloud Foundation (VCF) 41
Figure 25. Manual vNUMA Configuration, As Seen in Windows 42
Figure 26. Manual vNUMA Topology Presents Balanced Topology 42
Figure 27. Allocated Memory Evenly Distributed Across vNUMA Nodes 42
Figure 28. Allocating Super-cluster (or Cross-cluster) CPUs to Wide VM 43
Figure 29. Configuring Automatic vNUMA Presentation for Wide VM 43
Figure 30. Balanced vNUMA Presentation, As Seen in ESXTop 43
Figure 31. vNUMA Topology, As Seen in Windows 44
Figure 32. Wide VM Memory Distribution in Automatic vNUMA Configuration 44
Figure 33. Wide VM Memory Distribution in Automatic vNUMA Configuration, As Seen in Windows 45
Figure 34. Checking NUMA Topology with vmdumper 45
Figure 35. Windows Server Resource Monitor NUMA Topology View 46
Figure 36. Windows Task Manager NUMA Topology View 47
Figure 37. Core Info Showing a NUMA Topology for 24 cores/2socket VM 47
Figure 38. Using numactl to Display the NUMA Topology 48
Figure 39. Using dmesg Tool to Display the NUMA Topology 48
Figure 40. Displaying NUMA Information in the SQL Server Managemnet Studio 49
Figure 41. Errorlog Messages for Automatic soft-NUMA on 48 Cores per Socket VM 49
Figure 42. sys.dm_os_nodes Information on a System with 2 NUMA Nodes and 6 soft-NUMA Nodes 50
Figure 43. CPU HotAdd VM Advanced Configuration 51
Figure 44. Enable CPU Hot Add on VM 52
Figure 45. vNUMA Available with CPU HotAdd 52
Figure 46. Manually Configured vNUMA Topology 53
Figure 47. Defined Topology Mirrored in Windows 53

Figure 48. No "Phantom Node" with CPU HotAdd 54
Figure 49. Still no "Phantom Node" in Manual vNUMA Topology Configuration 54
Figure 50. Memory Mappings between Virtual, Guest, and Physical Memory 55
Figure 51. Setting Memory Reservation 56
Figure 52. Configuring Memory Limit 57
Figure 53. Setting Memory Hot Plug 59
Figure 54. VMware Storage Virtualization Stack 60
Figure 55. VMFS6 vs VMFS5 60
Figure 56. Random Mixed (50% Read/50% Write) I/O Operations per Second (Higher is Better) 62
Figure 57. Sequential Read I/O Operations per Second (Higher is Better) 62
Figure 58. Enabling Clustered VMDK (Step 1) 63
Figure 59. Enabling Clustered VMDK (Step 2) 63
Figure 60. Enabling Clustered VMDK (Step 3) 64
Figure 61. vSphere Virtual Volumes 65
Figure 62. Take Snapshot Options 68
Figure 63. VMware vSAN Original Storage Architecture 69
Figure 64. VMware vSAN Express Storage Architecture 71
Figure 65. XtremIO Performance with Consolidated SQL Server 73
Figure 66. Virtual Networking Concepts 74
Figure 67. vMotion of a Large Intensive VM with SDPS Activated 76
Figure 68. Utilizing Multi-NIC vMotion to Speed Up vMotion Operation 76
Figure 69. Data Service Manager - SQL Server Cluster Deployment Architecture 82
Figure 70. Image Registry Configuration for SQL Server 85
Figure 71. Directory Service Settings in Data Services Manager 86
Figure 72. Windows Server CPU Core Parking When Power Scheme set to “Balanced” 89
Figure 73 - No Cores Parked When Power Set Scheme to "High Performance" 90
Figure 74. Recommended Windows OS Power Plan 90
Figure 75. Enable RSS in Windows OS 91
Figure 76. Enable RSS in Windows Network Card Properties 92

List of Tables

Table 1. SQL Server High Availability Options 9
Table 2. CPU Power Management Policies 24
Table 3. Typical SQL Server Disk Access Patterns 66
Table 4. SQL Server Deployment Model in Data Services Manager 84
Table 5. Configuration Mode for Active Directory Integration 85
Table 6. Cluster Management Technology for SQL Server AlwaysOn Availability Group 86

1. Introduction

VMware Cloud Foundation (VCF) is a comprehensive private cloud platform designed to build and manage cloud infrastructure on-premises, at the edge, or as a managed service across supported cloud endpoints. It delivers a full-stack solution with a consistent infrastructure layer that integrates best-in-class compute, storage, networking, and cloud management components. With automated deployment and lifecycle management, VCF provides centralized administration, automation capabilities, and integrated cybersecurity features—making it ideal for organizations operating complex private cloud environments that extend to edge locations.

VCF Edge extends these capabilities to geographically distributed sites, enabling unified management of Edge and Remote Office/Branch Office (ROBO) locations from a central data center. By handling provisioning, lifecycle operations, and cluster management from SDDC Manager, VCF Edge removes the need for dedicated on-site administrators. This centralization streamlines operations and delivers a consistent, cloud-like experience across all sites.

.

Microsoft SQL Server®[1] is one of the most widely deployed database platforms, supporting environments ranging from large data warehouses with business intelligence workloads to small, specialized departmental databases. Its flexibility, powerful feature set, and the low cost of x86 hardware make it a popular choice for applications that demand both performance and scalability. This flexibility at the database layer directly enhances application design options and can significantly improve end-user productivity.

However, this flexibility often increases operational complexity. As the number of applications grows, more SQL Server instances fall under lifecycle management, each with unique requirements, patch levels, and maintenance needs. Many application owners prefer dedicated SQL Server instances, resulting in uneven resource allocation—some servers remain overprovisioned while others lack sufficient compute capacity.

To address these challenges, many organizations have adopted a "virtualization-first" strategy, deploying applications on virtual machines by default. SQL Server has become one of the most virtualized critical applications in recent years.

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

Top Applications and Workload Types

Application Name	Number of vCenters	Number of Customers	NUmber of VMs
microsoft sql server	154k	14.4k	15M
postgresql	183k	15.7k	2.54M
sql browser service exe	124k	12.6k	1.84M
ssms - sql server management studio	74.6k	8.39k	693k
mongodb	59.5k	6.7k	518k
mysql	67k	6.93k	864k
microsoft sql server analysis services	60.9k	7.15k	454k
mysql server	55.8k	6.52k	347k
couchbase server	34.5k	4.72k	245k
oracle database	25.5k	3.23k	294k
redis	25.2k	3.25k	126k
oracle rnts recovery service	22.1k	2.83k	525k

Virtualizing SQL Server with VCF® delivers resource optimization through consolidation while preserving application isolation and flexibility. SQL Server workloads can be migrated to new hardware without application remediation or changes to operating system versions or patch levels. VMware and its partners have demonstrated that vSphere is capable of supporting even the most demanding SQL Server workloads.

VCF provides additional advantages such as vSphere vMotion®, which enables live migration of SQL Server VMs across hosts and data centers with no user disruption. VMware Cloud Foundation enhances this with the new vSphere vMotion Notifications for Latency-Sensitive Applications[3] feature, allowing application teams to better control and schedule vMotion events. This improves availability and resilience and complements SQL Server's native high-availability tools—particularly during planned maintenance or workload relocation.

[1] Henceforth simply referenced as SQL Server
[2] Source: VMware VCF Usage Telemetry
[3] vSphere vMotion Notifications for Latency Sensitive Applications

Core VCF technologies such as vSphere Distributed Resource Scheduler™ (DRS), vSphere High Availability (HA), and vSphere Fault Tolerance (FT) deliver dynamic workload balancing and reliable VM-level protection. VMware NSX® adds network virtualization and adaptive security, while VMware Live Site Recovery™ provides disaster-recovery orchestration. VCF Operations delivers analytics and monitoring that further enhance operational efficiency.

For many organizations, the focus is no longer on *whether* to virtualize SQL Server, but on *how* to virtualize it effectively—achieving business and technical goals while minimizing operational overhead.

1.1 Purpose

This document outlines best-practice guidelines for designing and deploying Microsoft SQL Server in virtual machines running on VCF. The recommendations are not tied to a specific hardware configuration or a particular SQL Server footprint. Instead, they provide general guidance that can be adapted to a wide range of application requirements and deployment scenarios.

1.2 Target Audience

This document is intended for readers who already understand VCF and Microsoft SQL Server.

- **Architects** can use this material to understand how the overall system functions when integrating the various components.
- **Engineers and administrators** can reference the technical capabilities and configuration considerations.
- **Database administrators** can use it to understand how SQL Server fits into a virtualized infrastructure.
- **Management and process owners** can use the information to align operational models with the efficiencies and cost benefits of virtualization.

2. SQL Server Requirements Considerations

Successfully virtualizing SQL Server begins with a clear understanding of the business and technical requirements for each deployment. Requirements typically span availability, performance, scalability, growth expectations, patching approaches, and backup strategies. Taking a structured approach ensures that each SQL Server instance is placed in the right configuration and receives the resources it needs.

A high-level process for evaluating SQL Server candidates for virtualization includes:

- **Analyze performance characteristics and growth patterns** for the database workloads tied to each application.
- **Define availability and recovery needs**, including uptime expectations, disaster recovery requirements, and recovery objectives for both the VM and the databases.
- **Collect baseline resource-utilization metrics** from any physical servers currently hosting the workloads.
- **Plan the migration or new deployment** onto VCF.

2.1 Understand the SQL Server Workload

SQL Server is a relational database management system (RDBMS) designed to run diverse workloads across one or more user databases within a single instance. Applications interacting with these databases can impose very different workload profiles, and these differences drive architectural, performance, and availability design decisions. They also inform how virtual machines are sized and distributed across VMware ESXi™ hosts, along with the underlying storage layout.

Before deploying SQL Server inside VMs on VCF, it is essential to understand both the business requirements and the technical characteristics of each workload. Different applications demand different levels of capacity, responsiveness, and resilience. Many organizations therefore segment SQL Server deployments into tiers based on SLAs, recovery point objectives (RPOs), and recovery time objectives (RTOs). These tiers help determine the architecture, resource allocation, and operational model used.

Workload categories commonly include the following. It is generally not advisable to mix different workload types within a single SQL Server instance.

OLTP (Online Transaction Processing)

OLTP workloads typically support customer-facing or revenue-critical applications, making them some of the most important databases in the enterprise. They require consistently high performance and are extremely sensitive to latency and downtime. SQL Server VMs running OLTP workloads often need precise CPU, memory, storage, and network allocations. They are also frequent candidates for clustering via Windows Server Failover Clustering, using either Always On Failover Cluster Instances (FCIs) or Always On Availability Groups (AGs).

Characteristics include:

- Heavy, random write activity
- Steady CPU utilization during business hours
- Strong sensitivity to performance fluctuations

DSS / Data Warehouse / Decision Support

DSS workloads drive analytics, reporting, and strategic decision making. Many organizations treat these databases as mission critical—especially during financial close periods such as month-end and quarter-end. DSS databases depend heavily on CPU throughput and high-performance read activity from storage, especially during complex or long-running queries.

SQL Server in AI

Vector database and index support in AI use cases[4]:

SQL Server 2025 introduces native support for vector databases and vector index, using the Disk Approximate Nearest Neighbor (DiskANN) algorithm. It enables the storage and efficient querying of vector embeddings directly

[4] Vector functions

within the database engine. This integrated capability powers AI-driven functionalities such as recommendation systems and semantic search.

Batch, Reporting, and ETL Workloads

These workloads are active only during defined windows, such as batch processing cycles, scheduled report runs, or data integration processes. While they may not require stringent performance or availability guarantees, they often have strict business rules around data accuracy, consistency, or auditability.

Departmental and Lightly Used Databases

Smaller database workloads supporting departmental applications generally pose less operational risk. In many cases, short or even extended outages can be tolerated without major business impact.

Resource Requirements

Resource requirements for SQL Server deployments typically involve:

- CPU capacity and utilization targets
- Memory sizing for buffer cache efficiency
- Disk performance (IOPS, throughput, latency)
- Network I/O patterns
- User connection counts
- Transaction throughput and query-execution characteristics
- Overall database size and expected growth

Some organizations adopt utilization targets—such as maintaining host CPU usage around 80%—to ensure adequate performance headroom for spikes and to maintain availability.

Understanding these workload demands allows designers to size individual SQL Server VMs appropriately and to consolidate multiple workloads safely onto shared ESX hosts. A well-defined workload profile guides both virtualization design and the underlying physical infrastructure required to support it successfully.

2.2 Availability and Recovery Options

Running SQL Server on VCF 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"[5].

2.2.1 VMware Business Continuity Options

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

2.2.1.1. 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:

[5] For the comprehensive discussion of high availability and recovery options, refer to:
Planning Highly Available and Mission Critical Microsoft SQL Server on Windows Deployments with VMware vSphere
Protecting and Recovering Mission-Critical Applications in a VMware Hybrid Cloud with Site Recovery Manager

2.2.1.2. 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.

2.2.1.3. vSphere vMotion and vSphere Storage vMotion

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

The vSphere vMotion and vSphere Storage vMotion functionalities in VCF make 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.

2.2.2 Native SQL Server Capabilities

At the application level, all Microsoft features and techniques are supported on VCF, including SQL Server Always on Availability Groups, database mirroring, failover cluster instances, and log shipping. These SQL Server features can be combined with VCF 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[6]	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 VCF, including Always on Availability Groups, see the Knowledge Base article *Microsoft Clustering on vSphere: Guidelines for supported configurations (1037959)* at https://knowledge.broadcom.com/external/article?legacyId=1037959.

2.3 SQL Server on VCF 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.

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

Consider following points while architecting SQL Server on VCF

1. Use VMware Configuration Maximums Tool[7] to check the final architecture if any limits are reached or may be reached in the near future
2. Use VMware Compatibility Guide[8] to check compatibility for all components used
3. Use VMware Lifecycle Product Matrix[9] to find the End of General Support (EGS) date for solutions in use. For example, as of time of writing this document, EGS for ESXi/vCenter 5.5 is due 19 Sep 2018.
4. Recheck Microsoft Support Knowledge Base Article "Support policy for SQL Server products that are running in a hardware virtualization environment"[10]. 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 and VCF. The list of all supported VMware vSphere and VCF versions is available for easy reference on the Windows Server Virtualization Validation Program 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[11].
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.

[7] VMware Configuration Maximums
[8] VMware by Broadcom Compatibility Guide
[9] Product lifecycle and end of life information for Broadcom, Symantec, and VMware products
[10] Support policy for Microsoft SQL Server products that are running in a hardware virtualization environment
[11] Support partners for non-Microsoft hardware virtualization software

3. Best Practices for Deploying SQL Server Using VCF

A properly designed virtualized SQL Server instance running in a VM with Windows Server or Linux using VCF 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 VCF environment to optimize for best SQL Server performance.

3.1 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. 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 number of allocated 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..

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[12]. 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.

[12] More details can be found here: Administer Your Memory Resources with vSphere

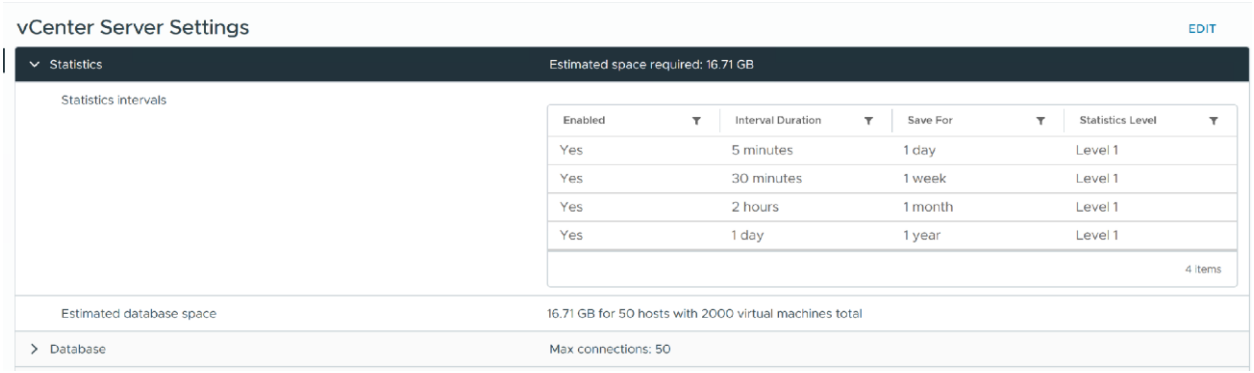
- Having more vCPUs assigned for the virtual SQL Server also has SQL Server licensing implications in certain scenarios, such as per-virtual-core licenses.

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

3.2 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



3.3 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.

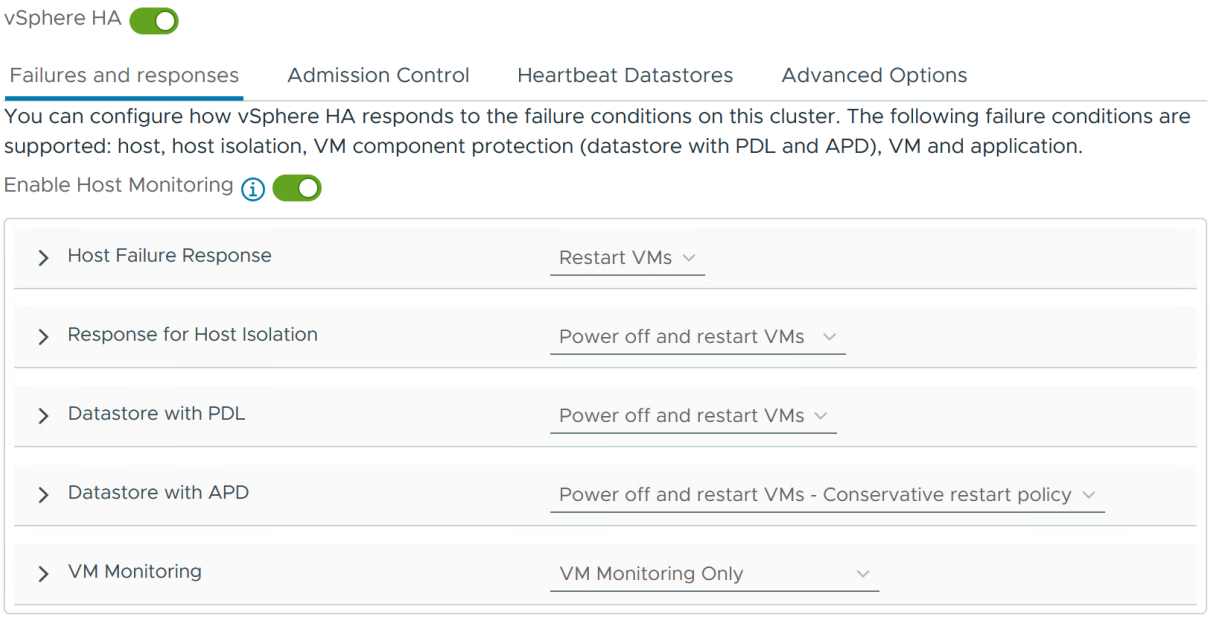
3.3.1 vSphere HA

vSphere HA is a feature that provides resiliency to a VCF 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[13].

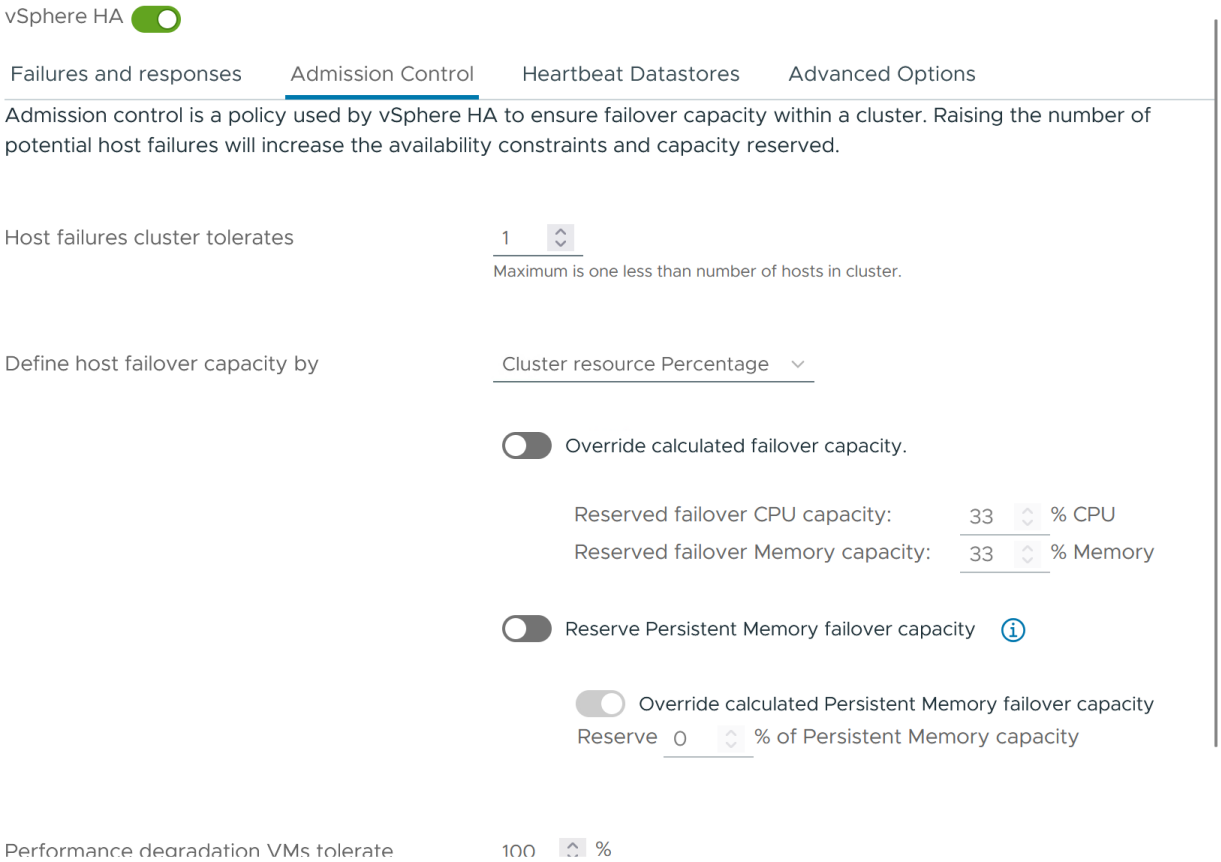
[13] Consult vCenter High Availability for more details.

Figure 3. vSphere HA Settings



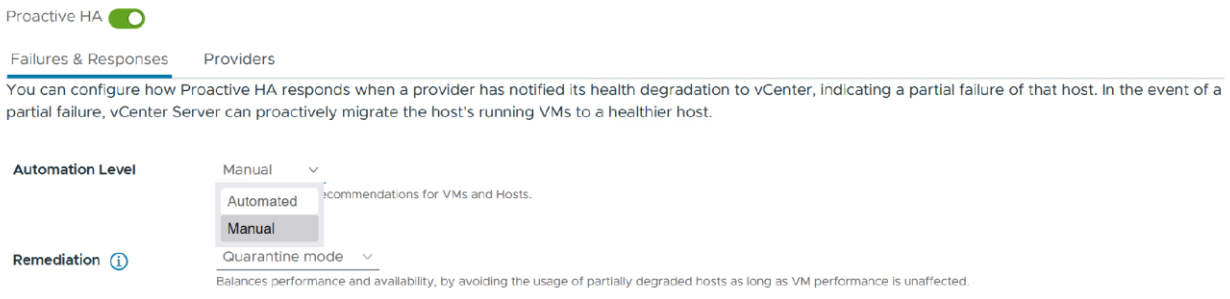
For mission-critical SQL Server workloads, ensure that enough spare resources on the host cluster exists to withstand a predetermined number of hosts removed from the cluster, both for planned and unplanned scenarios. vSphere HA admission control can be configured to enforce the reservation of enough resources so that the ability to power on these VMs is guaranteed.

Figure 4. vSphere Admission Control Settings



Starting with vSphere 6.5, VMware introduced a new feature called Proactive HA. Proactive HA, when integrated with the VCF 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

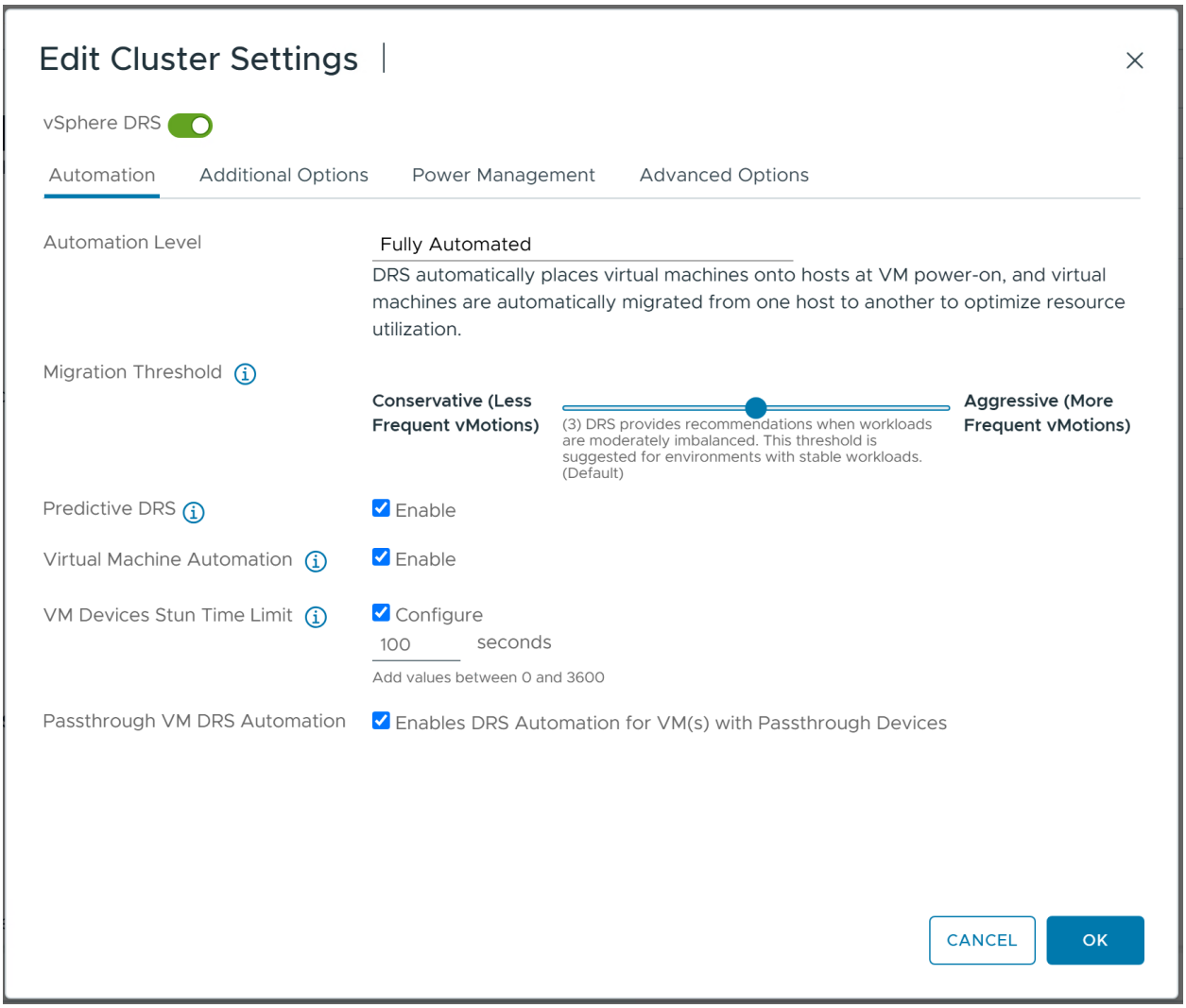


3.3.2 VMware DRS Cluster

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

VMware recommend enabling DRS functionality for a cluster hosting SQL Server VMs.[14]

Figure 6. vSphere DRS Cluster



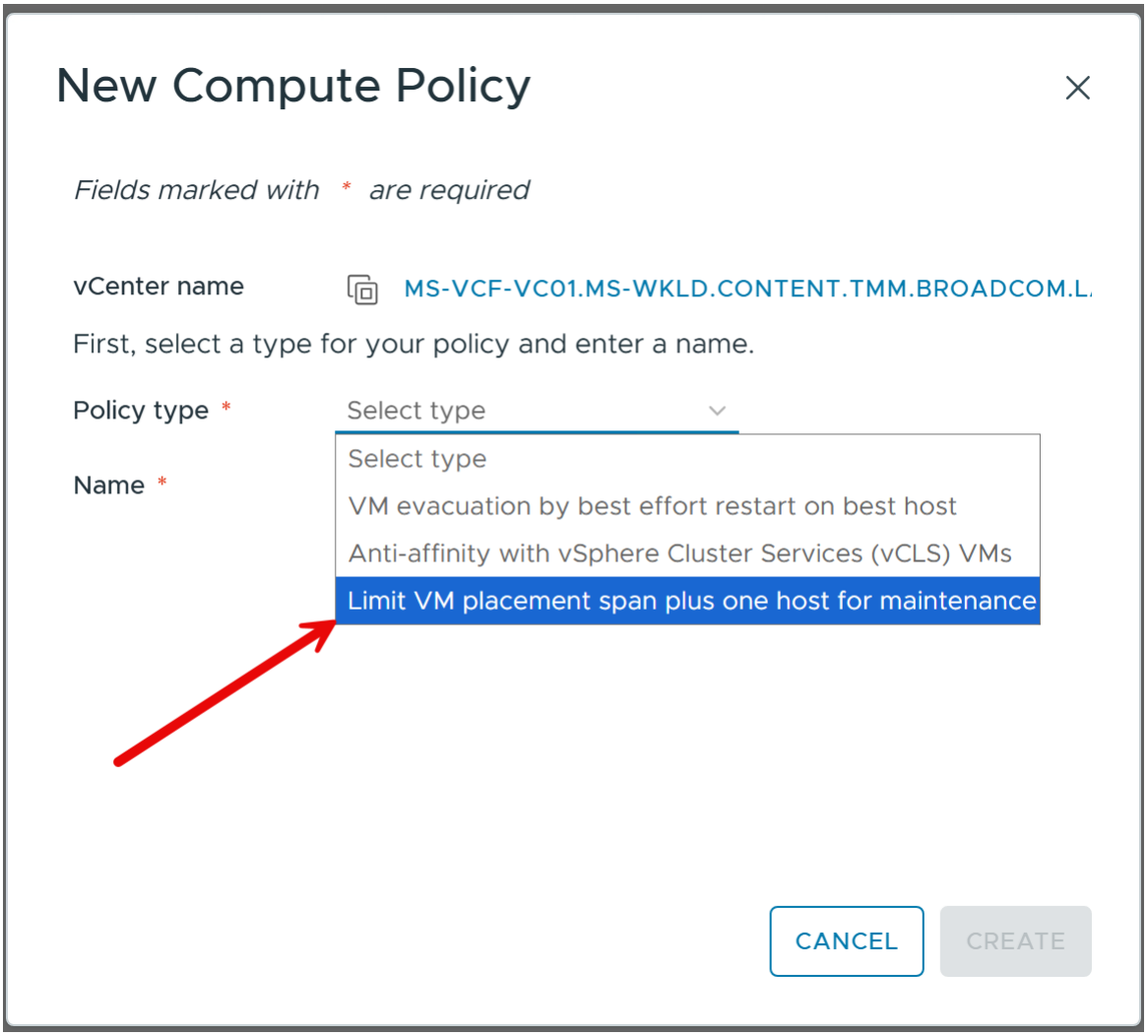
3.3.3 Host Span Limit Compute Policy

License compliance is a core requirement in the enterprise, especially in a virtualized environment. Because a single physical hardware's compute resources can be pooled and shared among multiple entities (VMs, Guest Operating Systems, Applications, etc) which operate independent of each other, tracking and accurately reporting on how the associated licenses are consumed in such configurations can be difficult. Historically, Customers have either had to create purpose-built ESXi Clusters which are dedicated to hosting VMs running these applications or create DRS affinity Rules which restrict these VMs to a subset of ESXi Hosts in the vSphere Cluster.

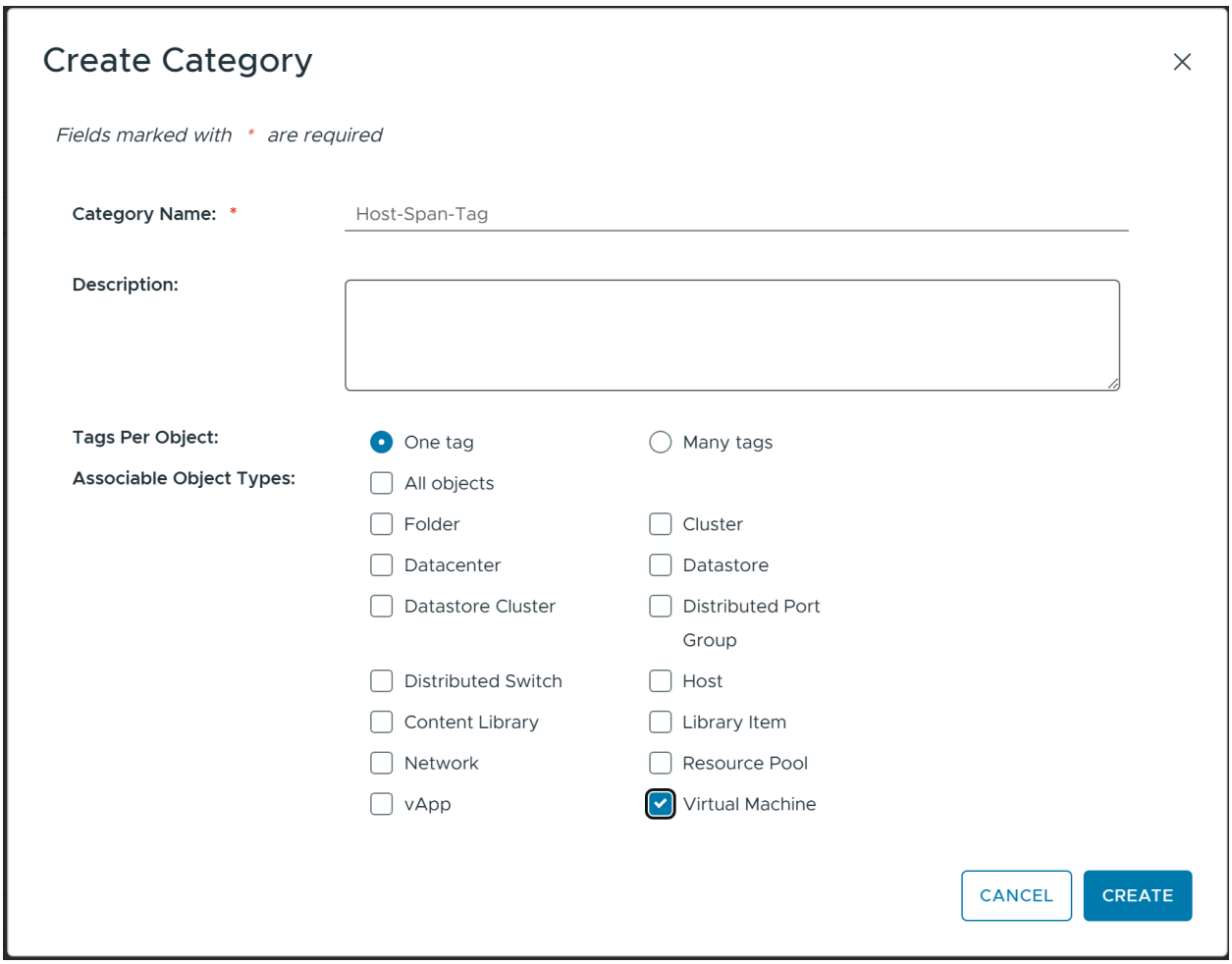
To address the inefficiencies associated with either of the foregoing options (and the fact that such configurations may not be universally sufficient to satisfy some application Vendors' more stringent license auditing, reporting and compliance requirements), Broadcom has introduced a new compute policy in VCF 9.0.

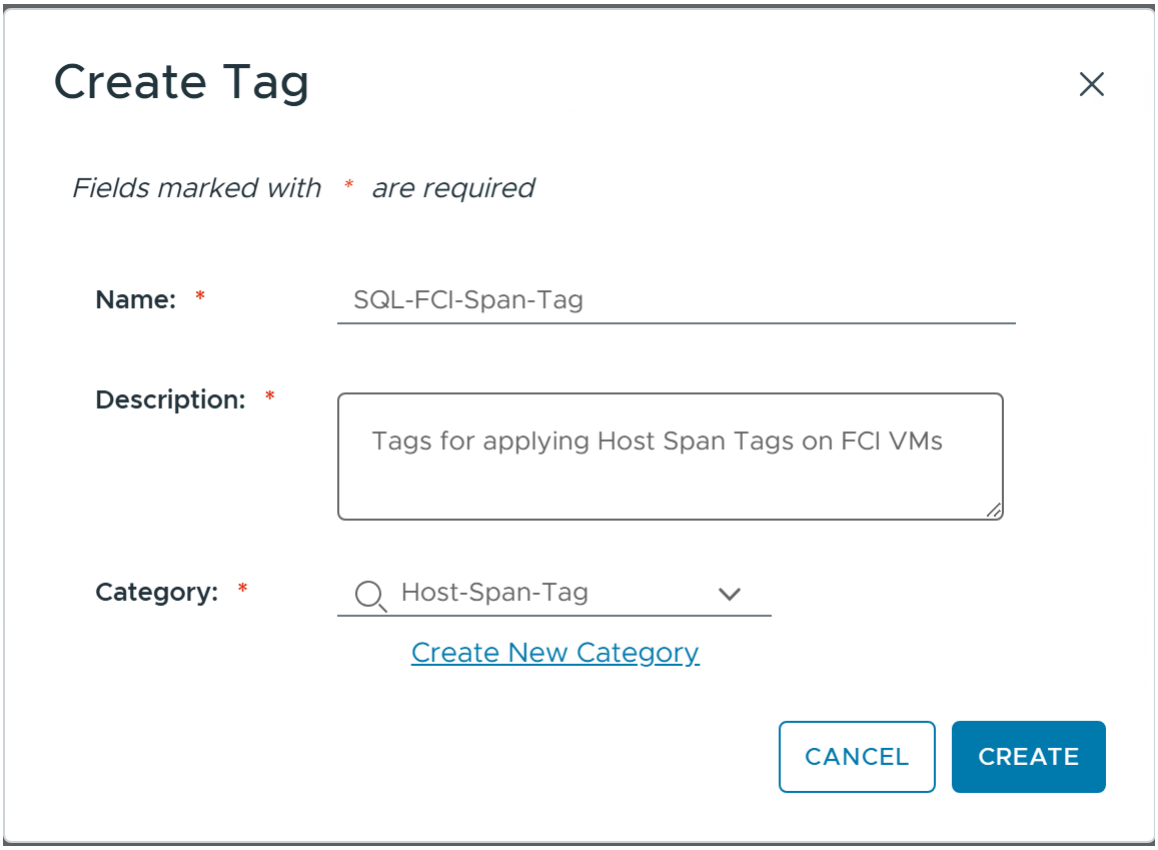
[14] More details:
VMware Distributed Resource Scheduler (DRS)
DRS Overhead Memory Management for VMs
VMware DRS Overview: Optimizing Resource Allocation in vSphere Cluster

Host Span Limit policy is a compute policy which allows VI Administrators to establish implicit affinity relationship between a group of virtual machines and a specified number of hosts, mandating those VMs to run only on those hosts, without explicitly identifying the hosts.

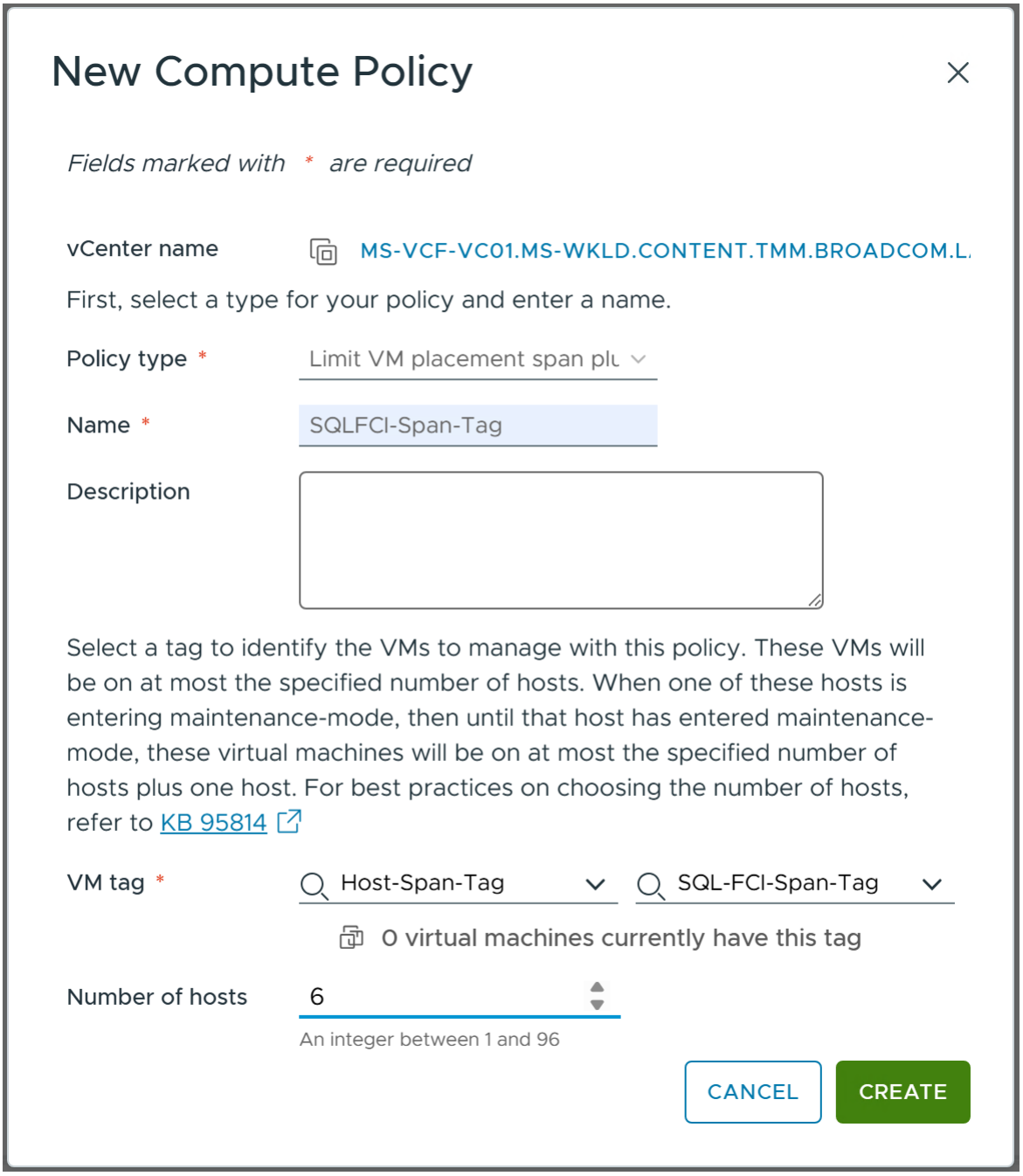


Using VM tags, Admins can identify VMs running a particular application for which they want to apply stringent license compliance restrictions.


Create Category
Fields marked with * are required
Category Name: *
Host-Span-Tag
Description:
Tags Per Object:
One tag
Many tags
Associable Object Types:
All objects
Folder
Cluster
Datacenter
Datastore
Datastore Cluster
Distributed Port Group
Distributed Switch
Host
Content Library
Library Item
Network
Resource Pool
vApp
Virtual Machine
CANCEL
CREATE



They can then create a Span Limit Compute Policy which contains the number of hosts for which they have procured licenses, associate the tagged VMs with the Policy and expect that, as long as the configuration remains in place, the tagged VMs will never be powered on or be allowed to run on any Host not included in the defined policy.



With the Span Limit policy, you can optimize the application licenses assigned to a fixed number of hosts, usually based on the number of cores.

Span Limit Policy Considerations and Limitations

Certain considerations and limitations exist when you use the Span Limit policy.

- To avoid violating the licensing instructions for the workloads, you must configure the Span Limit policy by setting the number of hosts (the group size) to be one less than the number of licenses.
  For example, if you have 5 licenses, you must configure the Span Limit policy with a group size of 4.
- VMs with the Span Limit policy (policy VMs) will be placed across no more than the group size number of hosts. This includes DRS operations like load-balancing, moving VMs in a cluster, and so on.

VM Operations

- If policy VMs are part of a cluster with DRS deactivated, the compliance status of such VMs with the Span Limit policy is "Not Applicable".

- If you power on a policy VM directly on the host, bypassing vCenter, the VM might be placed on a host that is not compliant with the Span Limit policy. The compliance status of such a VM is non-compliant, and the DRS will try to remediate the non-compliance.
- The cloning of a VM does not clone its tags to the cloned VM.
- If one of the hosts is in Maintenance mode, the VMs can spread across the group size plus one host. For example, you set the policy to have a group of 3 hosts and one of the policy hosts is in maintenance mode. While the host is entering maintenance, the policy VMs can spread across at most 4 hosts. Once the host successfully enters maintenance, the policy VMs will run on at most three hosts.

Maintenance Operations

- When using this policy, ensure you have sufficient spare capacity to perform maintenance operations.
- To ensure compliance with the policy, if a host selected by DRS for the policy is placed in maintenance mode, the policy VMs are moved to other hosts, including the additional replacement host selected by DRS for maintenance.
- If VMs cannot be moved to other hosts within the group, DRS will report a fault.

Host Operations

- You can add a host with VMs tagged with the policy's VM tag. However, the move into the cluster fails if the host violates the rules of an existing group of hosts.
- A host is removed from the group only when explicitly removed from the cluster.

HA Failover Operations for Policy VMs

- DRS ensures that policy VMs are failed over to other hosts within the hosts group.
- If a host with policy VMs fails and HA cannot communicate with DRS, the policy VMs will be restarted on other compatible hosts in the HTP group.
- If a host with policy VMs fail and HA can communicate with DRS but there is not enough failover capacity, the policy VMs remain powered off on the failed host.

DRS Rules

- This policy overrides the DRS soft rules, such as the VM-Host soft affinity or anti-affinity rules.
- If a policy VM is part of a VM-Host hard affinity rule, its compatible hosts are the ones that are both in the hard affinity rule and selected by DRS for this policy.

Cross vCenter vMotion

If a cross vCenter vMotion involves a tagged VM part of a span limit policy on the target vCenter, the vMotion will fail if the selected host on the target vCenter does not meet the policy requirements.

3.3.4 VMware EVC[15]

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.

VCF 9 introduces the concept of "Custom EVC Mode", providing fine-grained specificity for VI Admins and Operators who prefer to manually specify the configuration option they desire.

[15] VMware EVC and CPU Compatibility FAQ
KB - ERROR: The target host does not support the virtual machine's current hardware requirements

Figure 7. Pre-Defined VMware EVC Settings

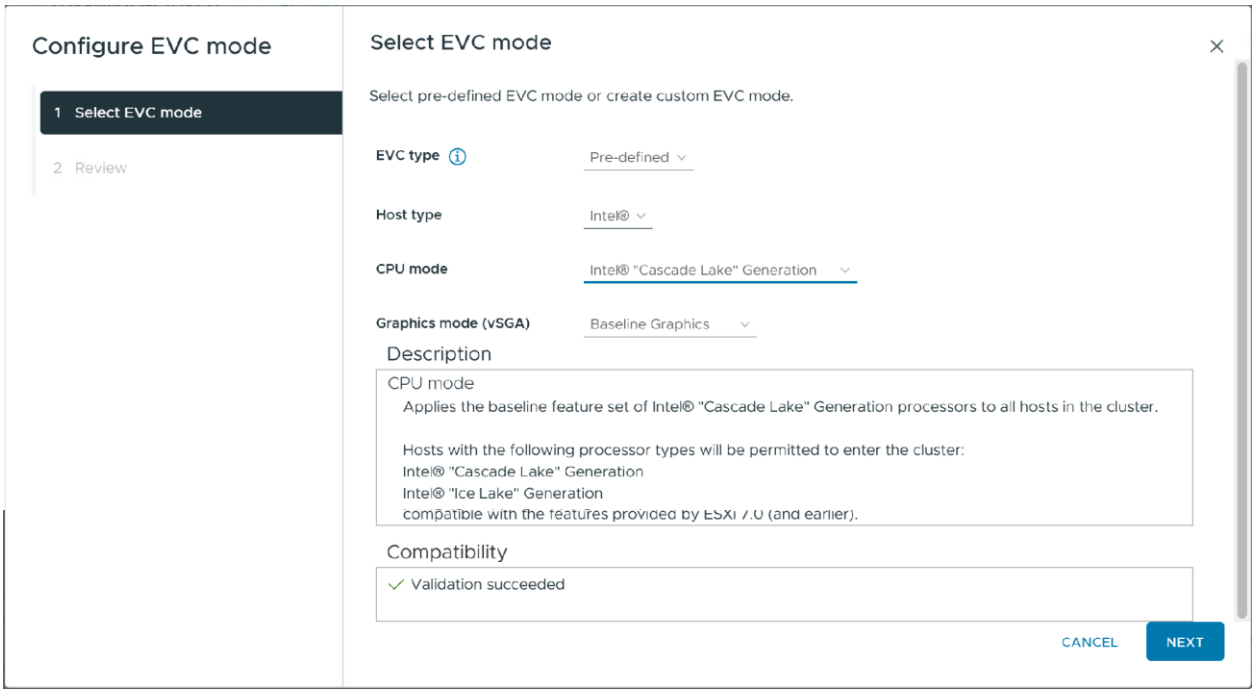
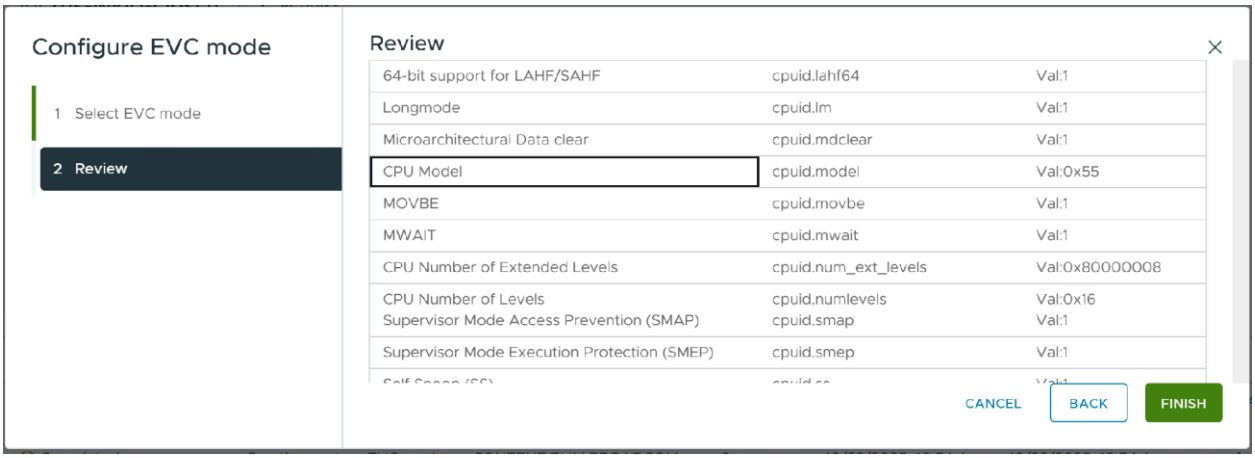


Figure 8 - Custom VMware EVC Settings



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:

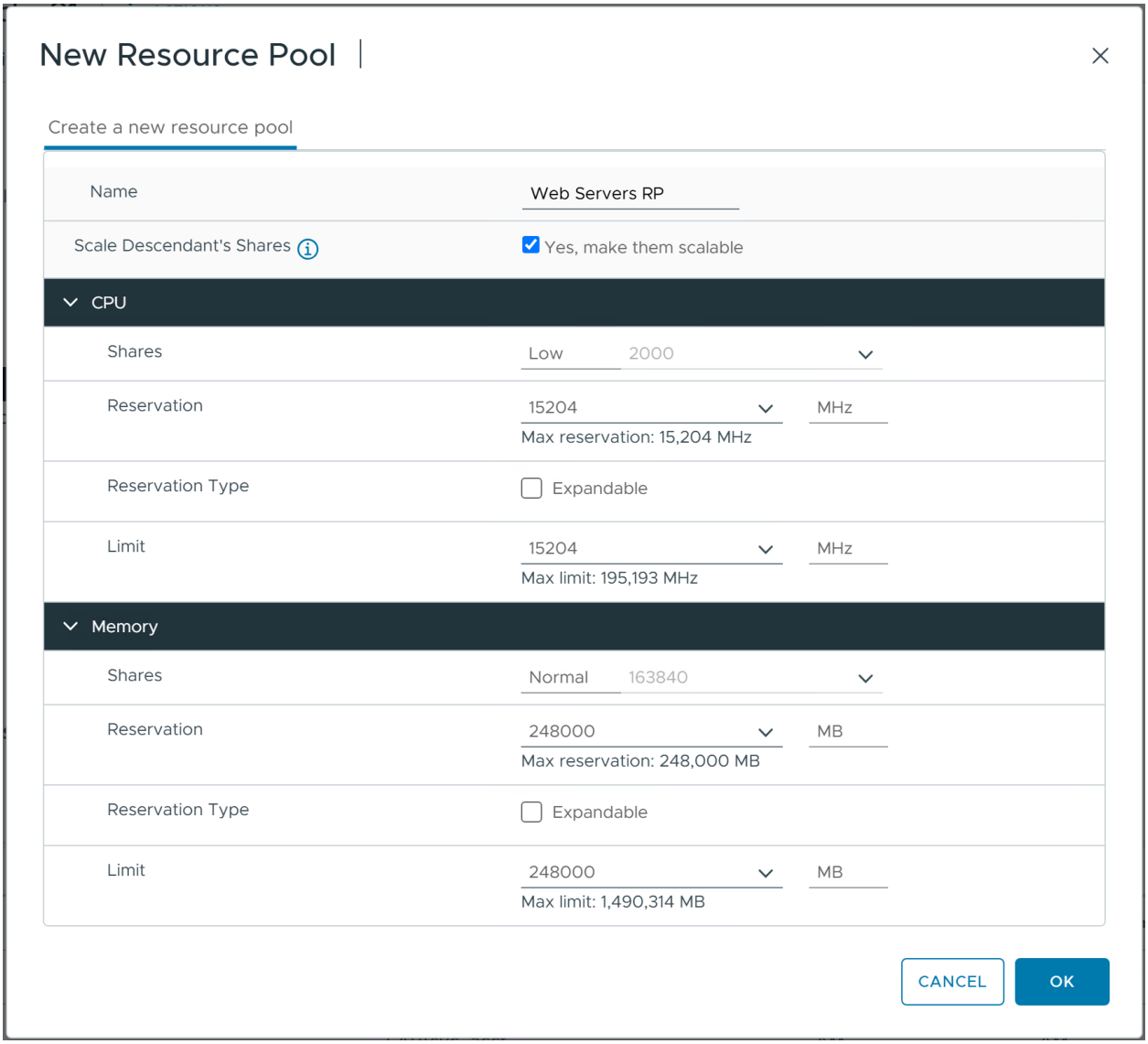
- A cluster consist of hosts with different CPU models and a vMotion of VMs between hosts is required. Avoid such configuration in production.
- Cross-cluster vMotion is required and hosts in different clusters have different CPU models. Consider using per-VM EVC (Section 3.5.8) if only a subset of VMs might be migrated to another cluster

3.3.5 Resource Pools[16]

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.



[16] More details: vSphere Resource Management

3.4 ESXi Host Configuration

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

3.4.1 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.

3.4.2 BIOS/UEFI Settings

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

- Enable Turbo Boost
- Enable hyper-threading
- Verify that all ESXi hosts have NUMA enabled in the BIOS/UEFI. In some systems (for example, HP Servers), NUMA is enabled by disabling node interleaving. Consult your server hardware vendor for the applicable BIOS settings for this feature.
- Enable advanced CPU features, such as VT-x/AMD-V, EPT, and RVI
- Follow your server manufacturer's guidance in selecting the appropriate Snoop Mode 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.
- While “Cluster on Die” or “Sub-NUMA Clustering” configurations benefit some applications who depend on finer latency and memory proximity provided by the configuration, Microsoft SQL Server does not require such dependencies and both CoD and SNC have not been demonstrated to be beneficial for SQL Server in terms of performance improvement.

  NOTE: Readers are reminded that performance-tuning recommendation or guidance for in-Memory OLTP Microsoft SQL Server configuration is outside the scope of this Document.
- 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.

3.4.3 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. In VCF 9, the default power scheme is “High Performance”. VI Admins are highly encouraged to visually confirm this settings on ESXi Hosts running Microsoft SQL Server workloads.

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

3.4.3.1. 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.

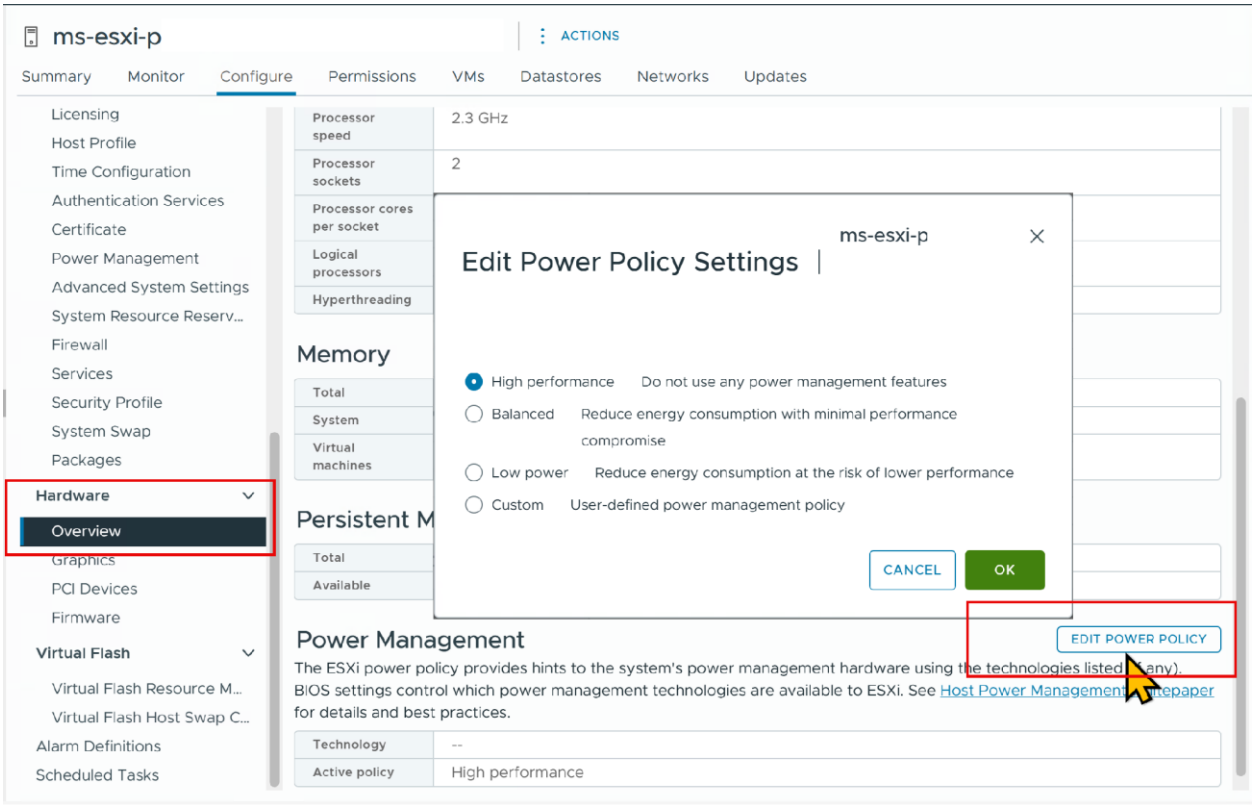
In general, selecting a high-performance policy provides more absolute performance, but at lower efficiency (performance per watt). Lower-power policies provide lower absolute performance, but at higher efficiency. ESXi provides five power management policies. If the host does not support power management, 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. This option is no longer present in VCF 9.0

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.

Figure 9. Recommended ESXi Host Power Management Setting



3.5 Virtual Machine CPU Configuration

3.5.1 Physical, Virtual, and Logical CPUs and Cores

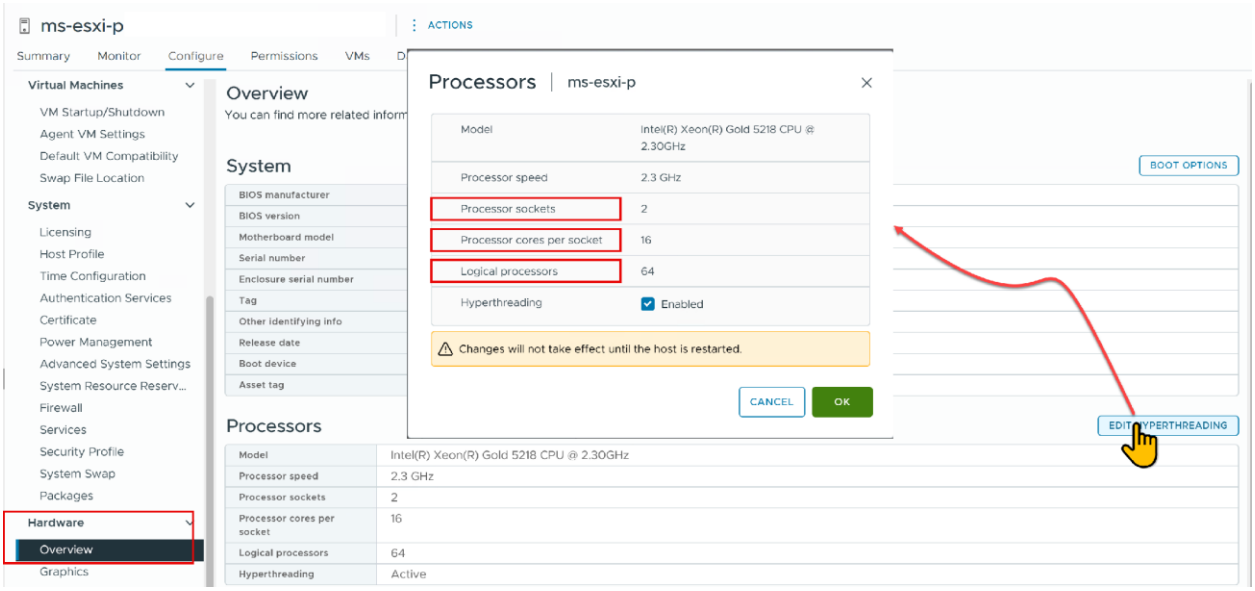
Let us start with the terminology first. VMware uses following terms to distinguish between processors within a VM and underlying physical x86/x64-based processor cores[17]:

- 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[18]. 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

[17] Virtual CPU Configuration and Limitations
[18] Hyperthreading with vSphere

Processor Core and does not perform like one[19]. This awareness will be very useful when making decisions about compute resource allocation and "right-sizing". For more details see the section 5.3

Figure 10. Physical Server CPU Allocation



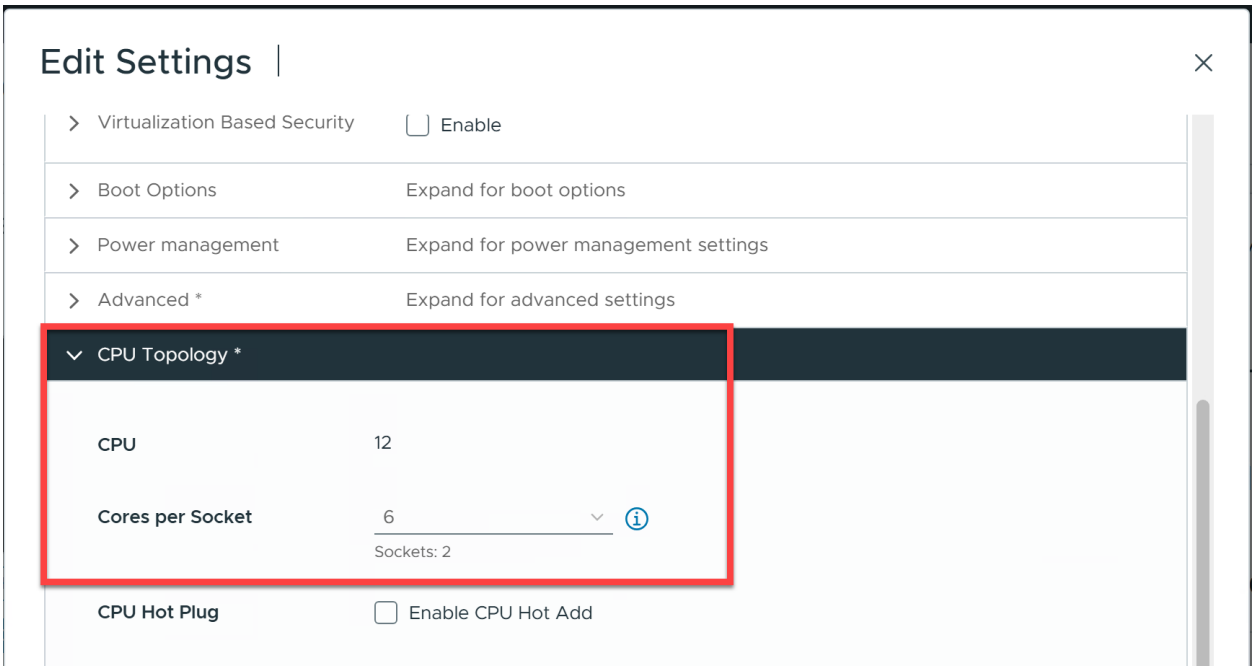
As an example, a host listed on the Figure 10 above, has two pSocket (two pCPUs), 16 pCores per Socket, and 64 logical Cores as a result of an active Hyper-Threading configuration. Hyperthreading is enabled by default if ESXi detects that the capability is enabled on the Physical ESXi Host. Some vendors 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:

Total vCPU = (Number of virtual Socket) * (Number of virtual cores per socket)

[19] Intel Hyper-Threading Technology (HT) Reference (Wikipedia)

Figure 11. Virtual Machine CPU Configuration



As an example, the virtual machine shown in Figure 10 above has two virtual Sockets, each with 6 vCores, with total number of vCPUs being 12 cores.

3.5.2 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.

3.5.3 Hyper-threading[20]

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

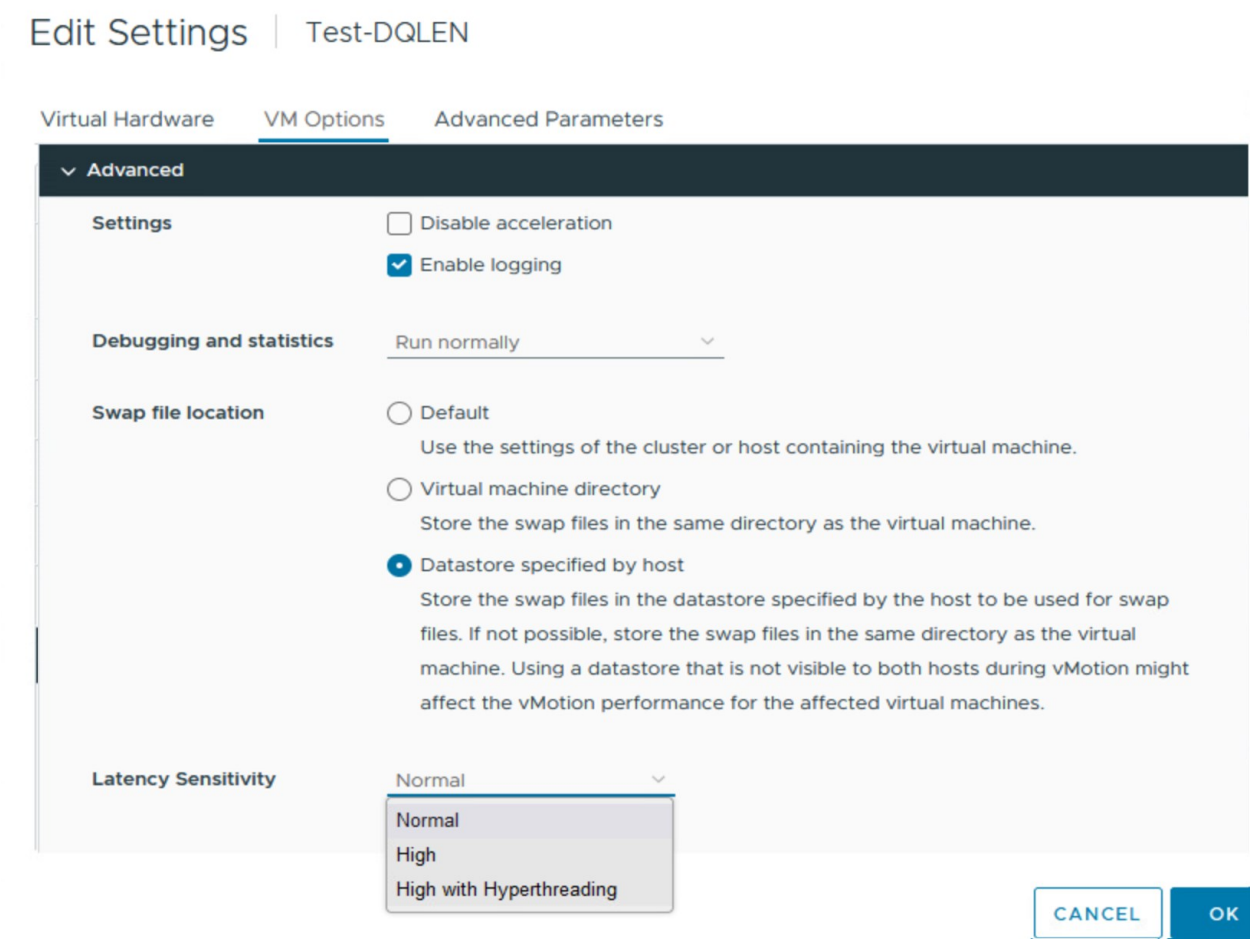
[20] See additional information about Hyper-threading on a vSphere Host

physical compute resources by a factor of 2:1 in VCF. 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 VMware Cloud Foundation (VCF) as an enhancement to the "Latency Sensitivity" setting which has long existed in VCF. 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:

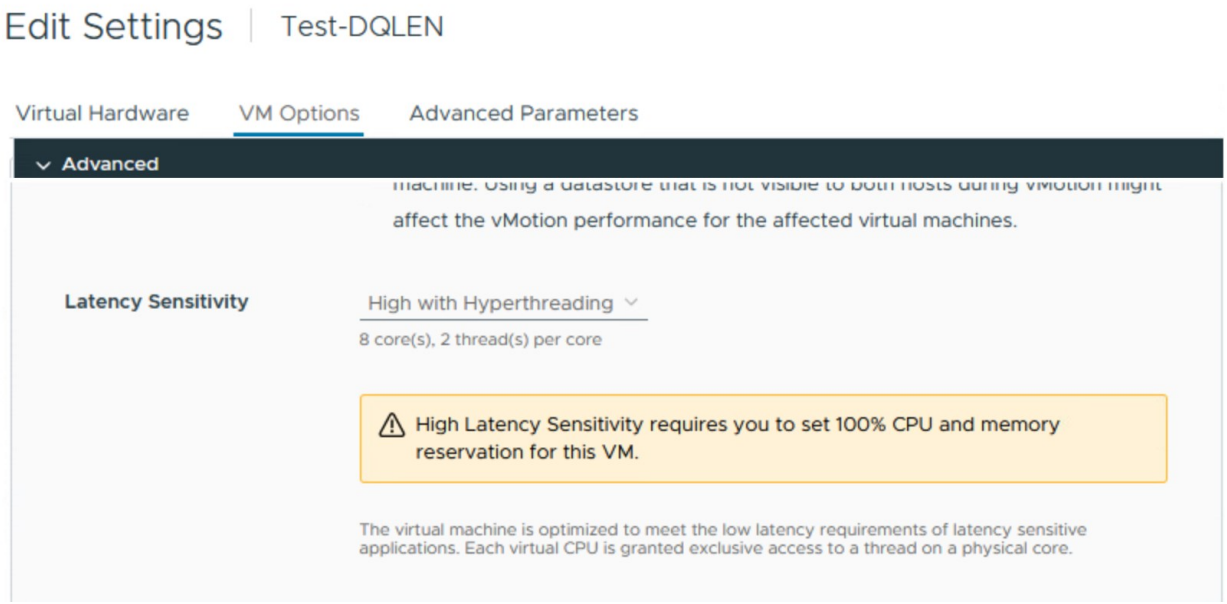
Figure 12. Setting Latency Sensitivity on a VM



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 VCF 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 13. "High With Hyperthreading" Provides Exclusive CPU Access



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.

3.5.4 NUMA Consideration

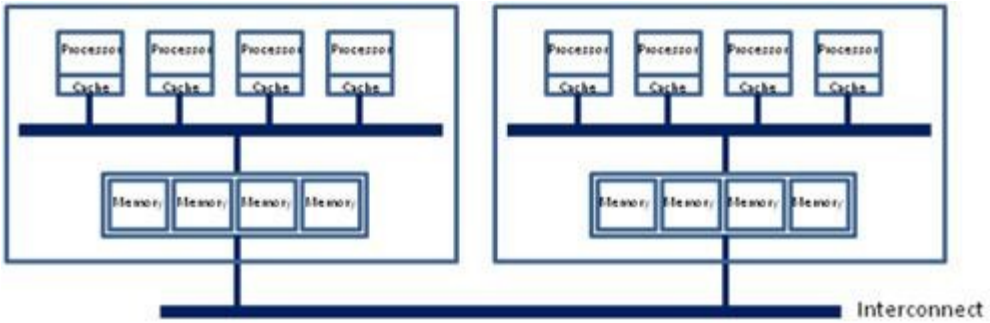
Over the last decade, no topic has attracted so much attention as discussions about NUMA technologies and its implementation. This is expected, taking into account the complexity of the technology, particularly the variances in vendor implementations, number of configurations options and layers (from 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 VCF platform.

3.5.4.1. Understanding NUMA[21]

Non-uniform memory access (NUMA) is a hardware architecture for shared memory implementing subdivision of physical memory banks between pCPUs (see Figure 14 below 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.

[21] Deep Dive into NUMA on vSphere

Figure 14. Intel based NUMA Hardware Architecture[22]



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 set, 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 3[23] 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 VCF. 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.

3.5.4.2. How ESXi NUMA Scheduling Works[24]

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).

[22] Optimizing Applications for NUMA - Intel and How To Enable and Verify NUMA and Intel® SGX on Intel® Xeon® Processors
[23] Up to 3 Times Performance Degradation (see "Optimizing Application Performance in Large Multi-core Systems")
[24] vSphere Resource Management

- 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.

3.5.4.3. Sub-NUMA Clustering (Cluster-on-Die)

An ESXi host's NUMA topology typically mirrors the physical CPU socket and memory topology. For example, a motherboard with two CPU sockets will generally contain two NUMA nodes. This holds true until we consider a feature set found in modern CPU architectures—Cluster-on-Die (CoD) or Sub-NUMA Clustering (SNC).

Both terms describe CPU-level optimizations aimed at improving performance by subdividing each CPU socket into smaller compute domains (sub-NUMA nodes). In ESXi, the CPU scheduler creates NUMA clients corresponding to each NUMA node presented to a VM. These clients are used internally to optimize VM performance. The scheduler may move these NUMA clients between nodes to rebalance workloads when it determines such migration could improve overall performance.

While this migration process is transparent to administrators, it can negatively impact VM performance. Ideally, NUMA client migration should occur as rarely as possible in a well-tuned VCF environment.

The Challenge with Large Virtual Machines

As server hardware becomes more powerful and capable of hosting denser compute resources, many organizations have shifted toward virtualizing business-critical workloads. This trend introduces a key design question: Should you deploy fewer, larger virtual machines ("Monster VMs"), or more, smaller ones?

Both approaches have merit:

- Fewer large VMs simplify administration but increase the potential impact of a single point of failure.
- More, smaller VMs reduce failure impact but add operational overhead.

In VCF environments, customers deploying Microsoft SQL Server often favor larger, denser VMs. This choice is usually driven by administrative simplicity or licensing considerations. Regardless of the motivation, understanding how CoD/SNC affects performance in such configurations is essential.

NUMA Awareness and Right-Sizing

VMware's reference architectures and best practices emphasize "right-sizing" VMs—allocating resources so that each VM fits within as few NUMA nodes as possible. This design principle leverages CPU-to-memory locality, which reduces latency and improves performance. When a CPU accesses memory local to its NUMA node, instructions complete faster than when accessing remote memory across interconnects.

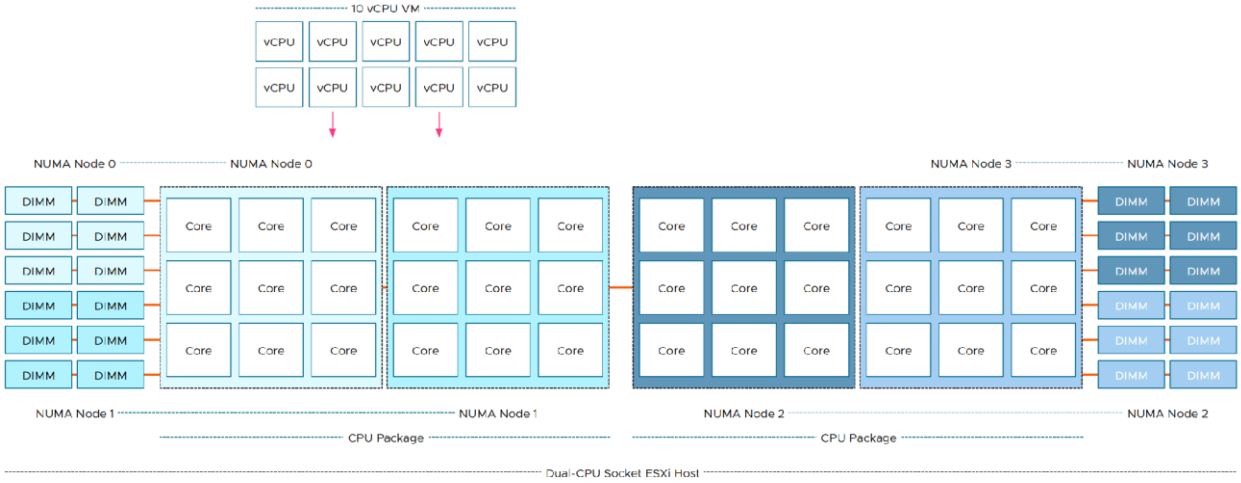
Administrators typically infer a host's NUMA layout based on hardware specifications and ESXi information.

Example: Without CoD/SNC Enabled

Consider a system with:

- 2 CPU sockets
- 36 total cores (18 per socket)
- 512 GB of RAM (256 GB local to each socket)

In this setup, a VM configured with 10 vCPUs and ≤256 GB of memory fits neatly within a single NUMA node. All CPU instructions and memory operations occur locally, minimizing interconnect latency and ensuring optimal performance.



How CoD/SNC Changes the Topology

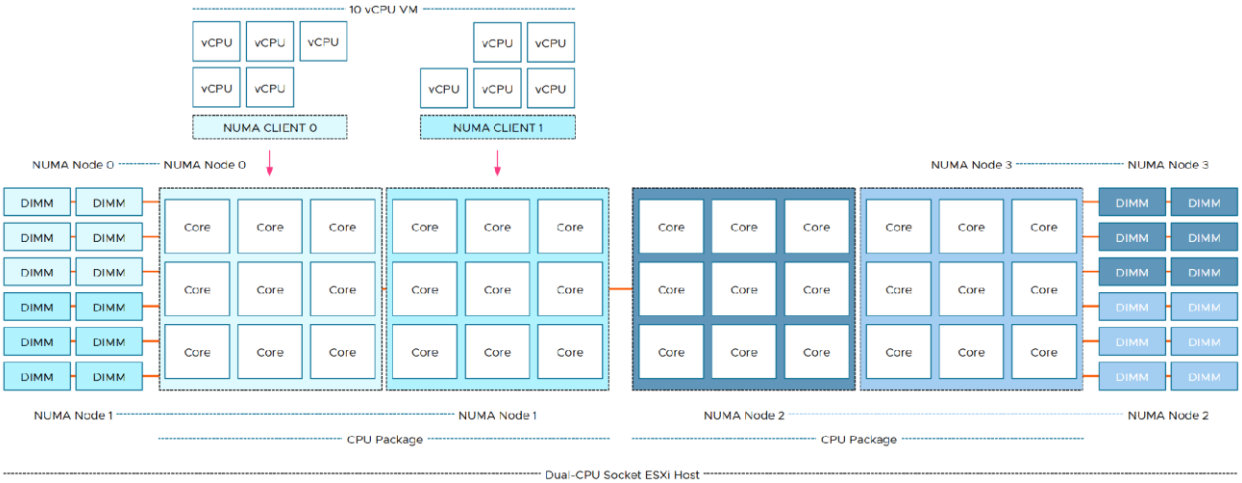
When CoD or SNC is enabled, each physical NUMA node is further divided into smaller sub-NUMA clusters. The result is a logical topology that no longer directly matches the physical one inferred by the Administrator.

Importantly, this sub-NUMA topology is not visible in the vSphere Client. As a result, administrators may misjudge how resources are being allocated to VMs.

In our example system, enabling CoD/SNC changes the effective topology so that:

- The 36-core, 2-socket system now presents 4 smaller sub-NUMA nodes.
- A 10-vCPU VM that previously fit within one NUMA node now spans two sub-NUMA boundaries.

This situation causes ESXi to create two NUMA clients for the VM, placing vCPUs across sub-NUMA boundaries. The result is a "right-sized" VM that nonetheless performs sub-optimally—particularly for workloads like SQL Server that are sensitive to NUMA locality.



Practical Performance Implications

In the previous example, with CoD/SNC enabled, a VM configured with 8 vCPUs may perform better than a 10-vCPU VM. This illustrates why administrators must closely evaluate system configurations and resource allocations—especially for large, business-critical "Monster VMs."

Although certain workloads may benefit from CoD/SNC, VMware has not observed performance gains for large Microsoft SQL Server workloads running on VCF under this configuration. VMware therefore recommends extensive testing and validation before enabling Cluster-on-Die or Sub-NUMA Clustering in production environments.

Key Recommendations

- Understand your hardware topology — Be aware of how CoD/SNC alters NUMA structures beneath the hypervisor.
- Test before deployment — Benchmark workloads to identify whether SNC/CoD provides measurable benefits or penalties.
- Right-size VMs carefully — Align vCPU and memory allocations to the actual NUMA structure, not just physical socket counts.
- If SNC/CoD is enabled, allocate vCPUs to large VMs in multiples of the sub-NUMA core size.
- Monitor NUMA client migrations — Excessive migration indicates possible NUMA misalignment.

Checking to see if CoD/SNC is enabled with ESXTop:

- SSH into the ESXi host
- Run *echo "CPU Packages";vsish -e dir /hardware/cpu/packageList;echo "NUMA nodes";vsish -e dir /hardware/cpuTopology/numa/nodes*

```
[root@lvnlvcfstgtmmc21:~] echo "CPU Packages";vsish -e dir /hardware/cpu/packageList;echo "NUMA nodes";vsish -e dir /hardware/cpuTopology/numa/nodes
CPU Packages
0
1
NUMA nodes
0
1
```

The result above reveals a System in which the reported NUMA nodes match the number of CPU packages in the System. This indicates the CoD/SNC is not enabled on the System

Checking NUMA Client Migrations with ESXTop:

- SSH into the ESXi host.
- Run `esxtop`
- Press `m` to switch to the memory view
- Observe the fields under the NUMA section (`NUMA Home Node`, `NUMA Remote`, etc.)
- Look at the `NMIG` (NUMA Migrations) counter to identify whether memory pages are being moved between NUMA nodes
- Frequent increments in the `NMIG` counter indicate possible NUMA client migration or misalignment
- Look at the NHN (NUMA Home Node) and N%L (NUMA Locality) fields to identify whether memory accesses are local or remote
- Frequent changes in NHN or low N%L values indicate possible NUMA client migration or misalignment

If you observe significant remote memory activity or frequent NUMA home migrations, consider reconfiguring VM vCPU or memory allocations to better align with physical NUMA nodes.

3.5.4.4. Understanding New vNUMA Options in VCF[25]

With the release of VMware Cloud Foundation (VCF), 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 VMware Cloud Foundation (VCF). 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. Manual administrative intervention ("numa.consolidate = FALSE") is required to configure the exposed NUMA topology in a way that distributes the allocated memory optimally across the vNUMA nodes. VCF now provides a more intuitive GUI-based option enabling Administrators to make this configuration more easily and correctly. We will discussed this new option in more details later in this document.

3.5.4.5. 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, meaning that all deployments of modern SQL Server will benefit from a properly configured NUMA presentation[26].

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 not be 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)[27] to expose NUMA topology.

[25] See vSphere 8 CPU Topology for Large Memory Footprint VMs Exceeding NUMA Boundaries for more detailed description and information
[26] 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
[27] 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

As rules of thumb, the number of exposed NUMA nodes will be equal to the number of physical sockets for Intel processors[28] and will be two times for AMD processors.

Note: Please check you server documentations for more details.

VMware ESXi Hypervisor Host

vSphere supports NUMA on the physical server starting with version 2. In VCF, 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 15. Using esxcli or sched-stats to Obtain NUMA Node Count on ESXi Host

```
[root@lvnlvcfstgtmmc21:~] esxcli hardware memory get | grep NUMA
   NUMA Node Count: 2
[root@lvnlvcfstgtmmc21:~] sched-stats -t ncpus
48 PCPUs
24 cores
 2 LLCs
 2 packages
 2 NUMA nodes
```

or

- *ESXTOP*
- *Press M for memory*
- *F to adjust fields*
- *G to enable NUMA statistics,* for the view and result shown in the image below.

Figure 16. 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                  -    -        -         -   -        -        -        -        -
```

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

[28] 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 (Intel® Xeon® Processor Scalable Family Technical Overview). VMware ESXi supports CoD starting with vSphere 6.0 and 5.5 U3b (Intel Cluster-on-Die (COD) Technology, and VMware vSphere 5.5 U3b and 6.x)

memory assigned to a pNUMA) and respectively NUMA clients (one per virtual machine per NUMA home node) will be created[29].

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 VMware Cloud Foundation (VCF) 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 and VCF if not mentioned otherwise:

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

a. The minimum vCPU threshold at which ESXi exposes vNUMA to a VM is 9. 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 machine to which virtual NUMA is exposed 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)[30] feature to group ESXi Hosts with dissimilar physical NUMA topologies together in a single vSphere Cluster.

   ii. **Because a VM's vNUMA topology is not re-evaluated after power-on**, migrating a running VM from one Host to another Host with dissimilar 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 in order to become compatible with its new Host's EVC mode.

c. Neither a VM's allocated memory nor the number of cores-per-socket or virtual sockets is taken into consideration when exposing vNUMA to a VM, even though NUMA/vNUMA is theoretically a function of a combination of CPU/vCPU and Memory,.

   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.

[29] See NUMA Deep Dive Part 5: ESXi VMkernel NUMA Constructs for more details
[30] Configure the EVC Mode of a Virtual Machine

d. The issue described in the previous section is now effectively addressed in VMware Cloud Foundation (VCF), 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 VMware Cloud Foundation (VCF) (see the "CPU Hot Plug" Section below)

f. A minimum of VM virtual hardware version 8 is required to have vNUMA exposed to the Guest OS. However, the new enhancements in VMware Cloud Foundation (VCF) 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. The pNUMA information from the host, where the VM was powered on 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 VMware Cloud Foundation (VCF), 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 in order to evaluate and determine the impacts of the recommendations prescribed in the section that follows below on their specific usage situations.

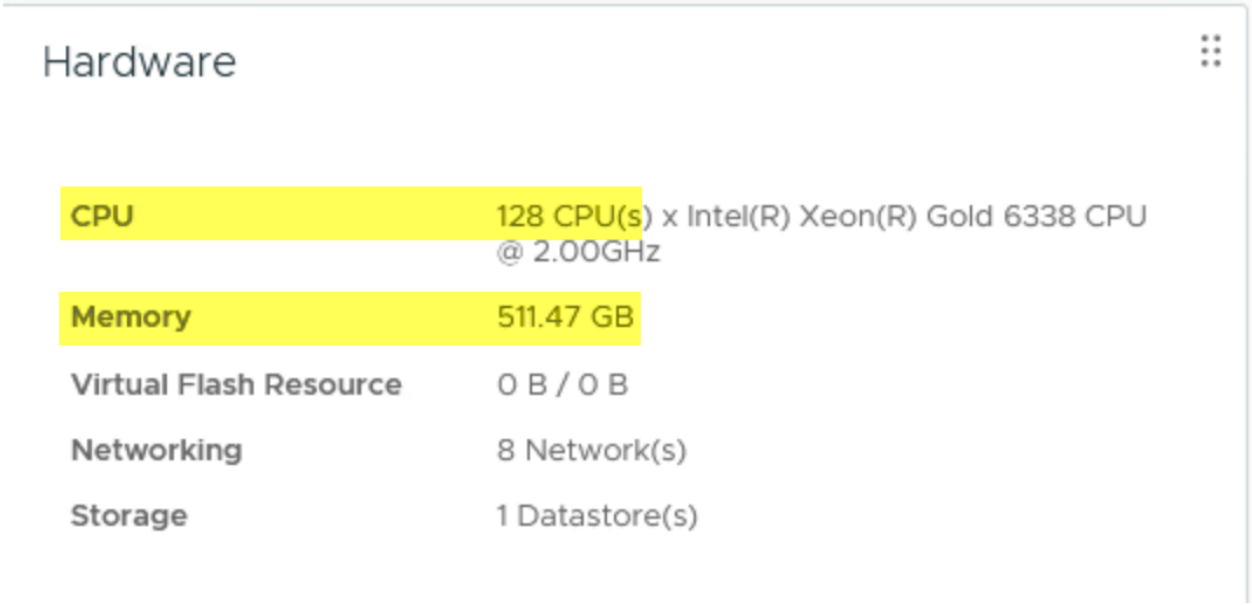
Configuring virtual NUMA for "Wide" Microsoft SQL Server VMs

VMware Cloud Foundation (VCF) 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 VCF 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 17. When is My Considered "Wide"?



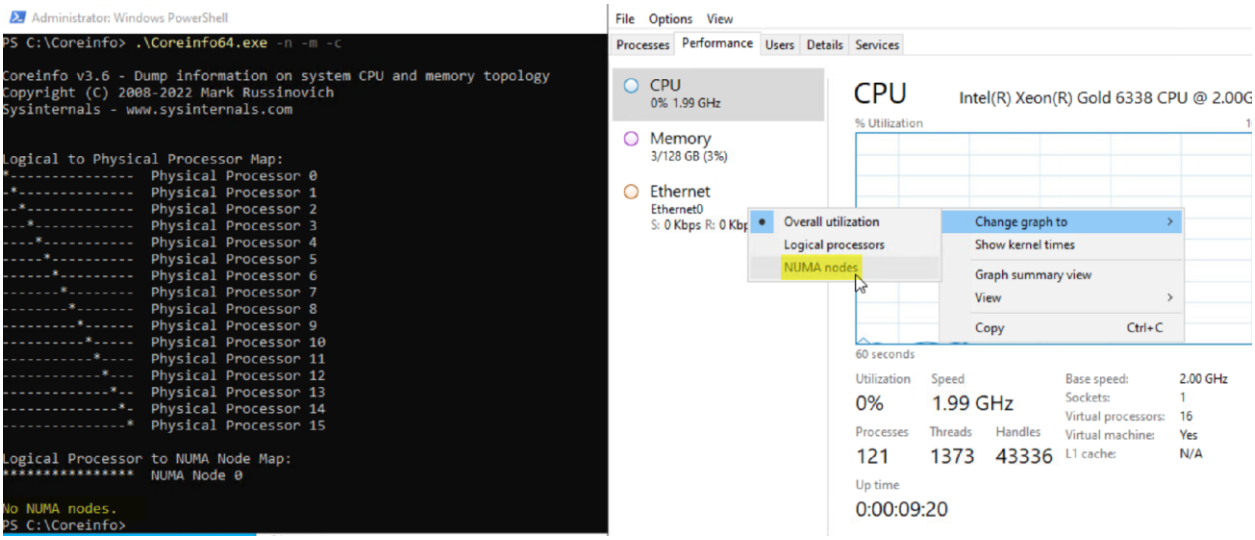
Our ESXi host has 128 logical CPUs (2 Sockets, each with 32 cores, and hyperthreading is enabled).

It also has 512GB of Memory.

Example 1

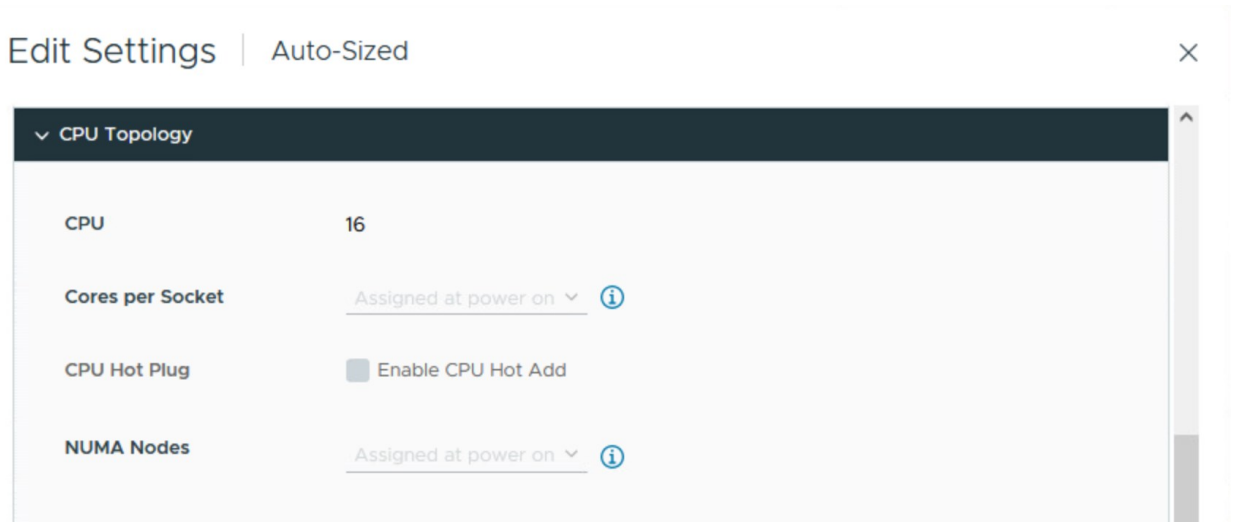
If we create a non-wide VM with 16 vCPUs and 128GB RAM and power it on, we see that ESXi has determined that, even though we have exceeded the minimum vCPU threshold beyond which vNUMA is automatically exposed to a VM, this is not a "Wide" VM because ALL of the compute resources allocated can fit into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

Figure 18. ESXi Creates UMA Topology When VM's vCPUs Fits a NUMA Node



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

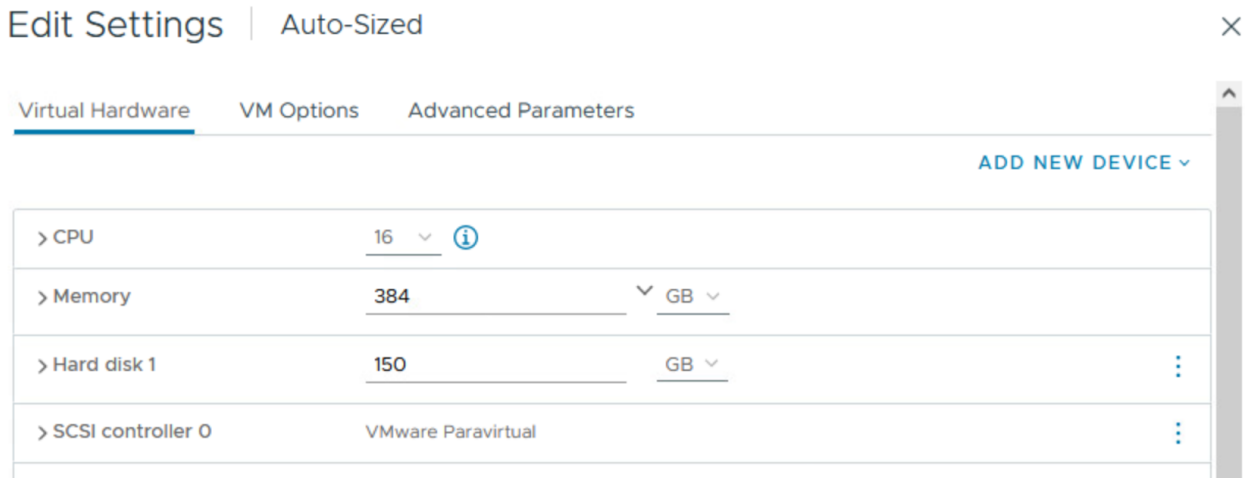
Figure 19. Letting ESXi Determine vNUMA Topology Automatically for Non-Wide VM



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



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

Figure 21. ESXi Doesn't Consider Memory Size in vNUMA Configuration

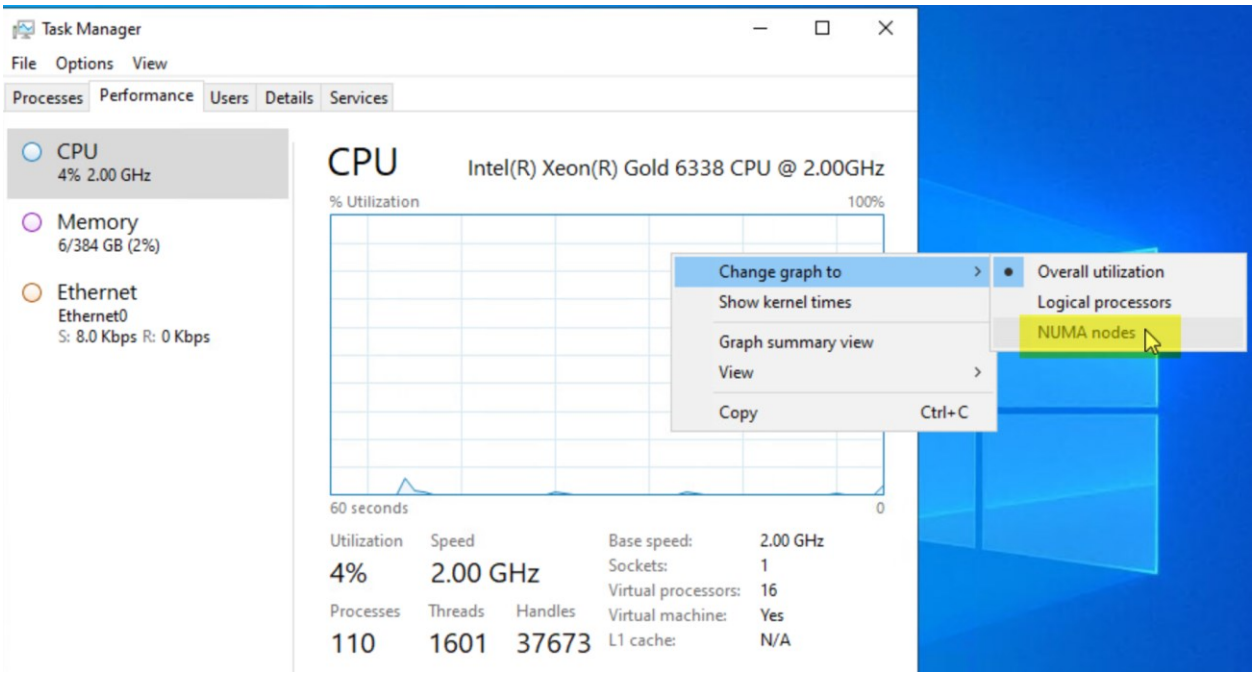
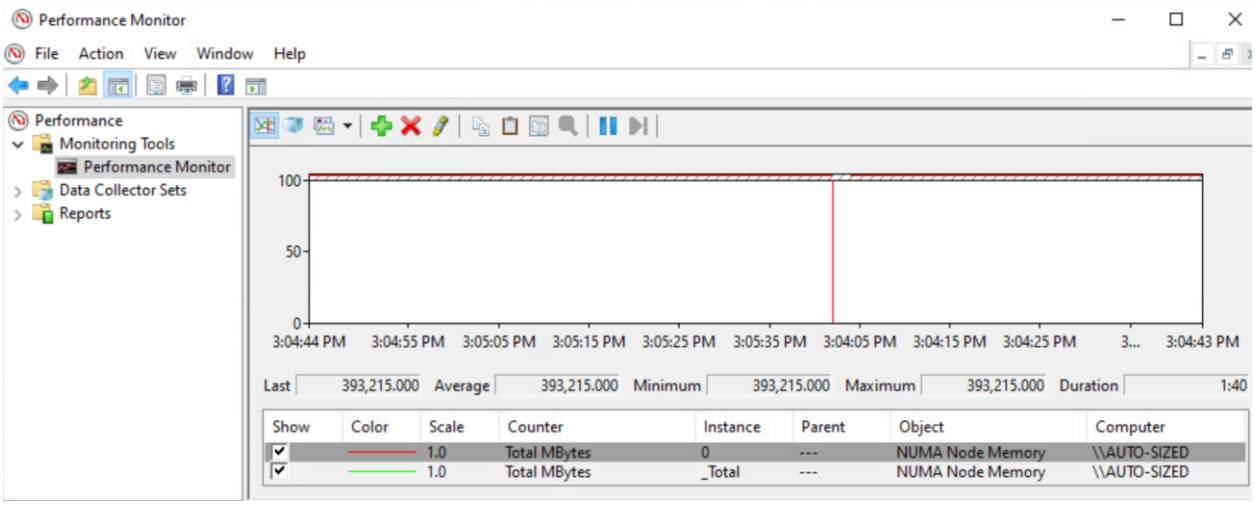
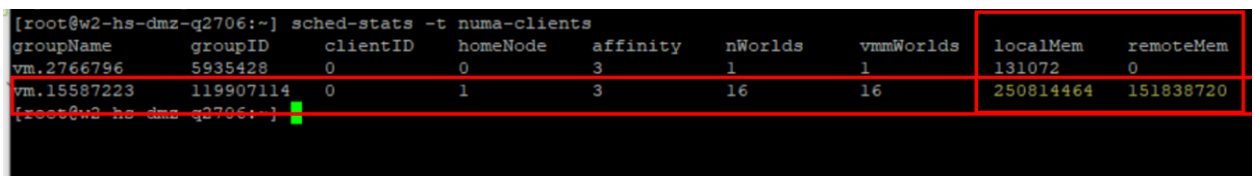


Figure 22. Memory-wide VM, As Seen by Windows



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



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.

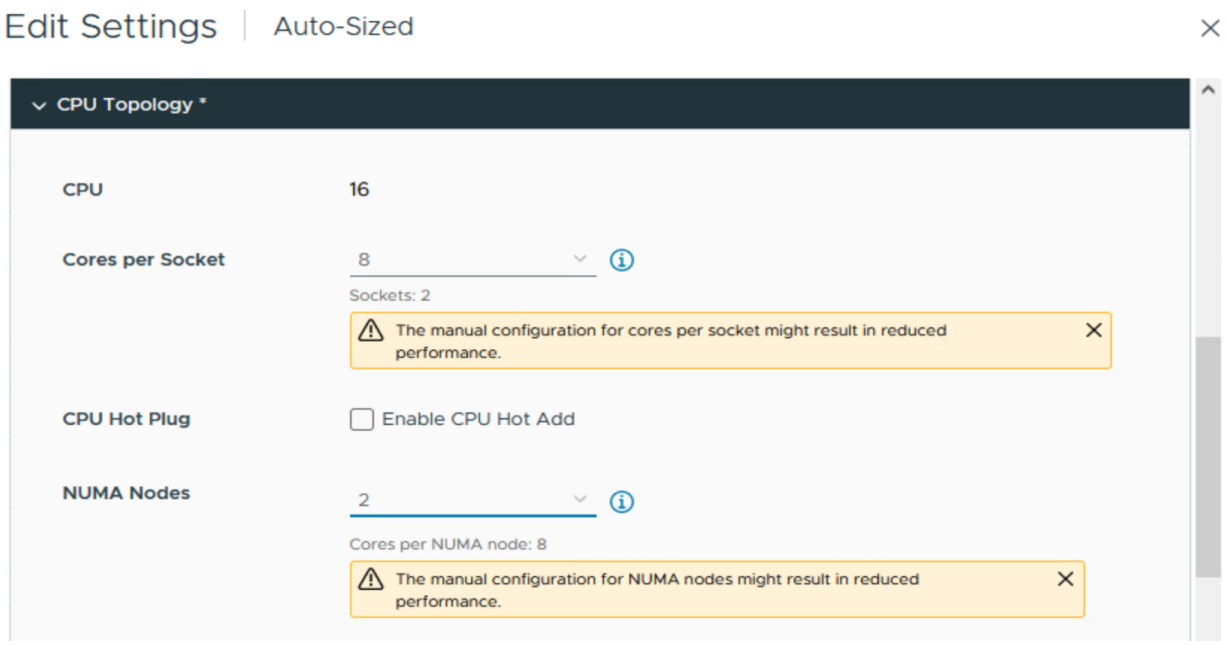
The major issue in the previous example is that we needed more memory than vCPUs for our SQL Server instance. This could be due to specific business requirements or other needs, so we will not attempt to rationalize the requirement. Our goal is to provide a properly configured VM in VCF to optimally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

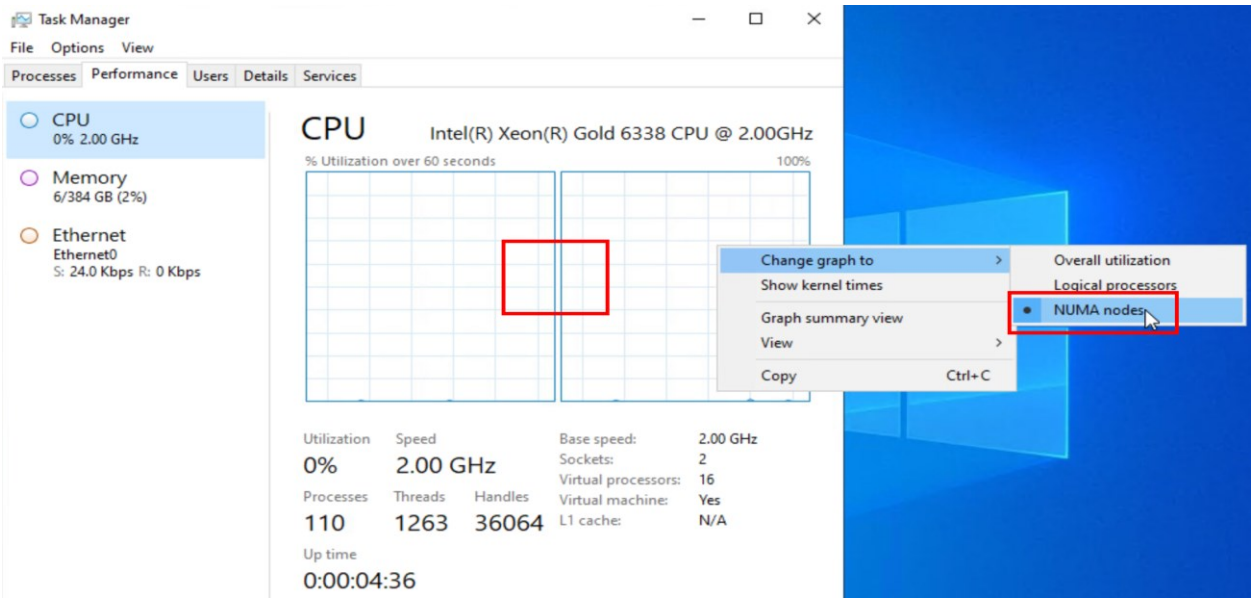
Figure 24. Manual vNUMA Configuration Options in VMware Cloud Foundation (VCF)



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

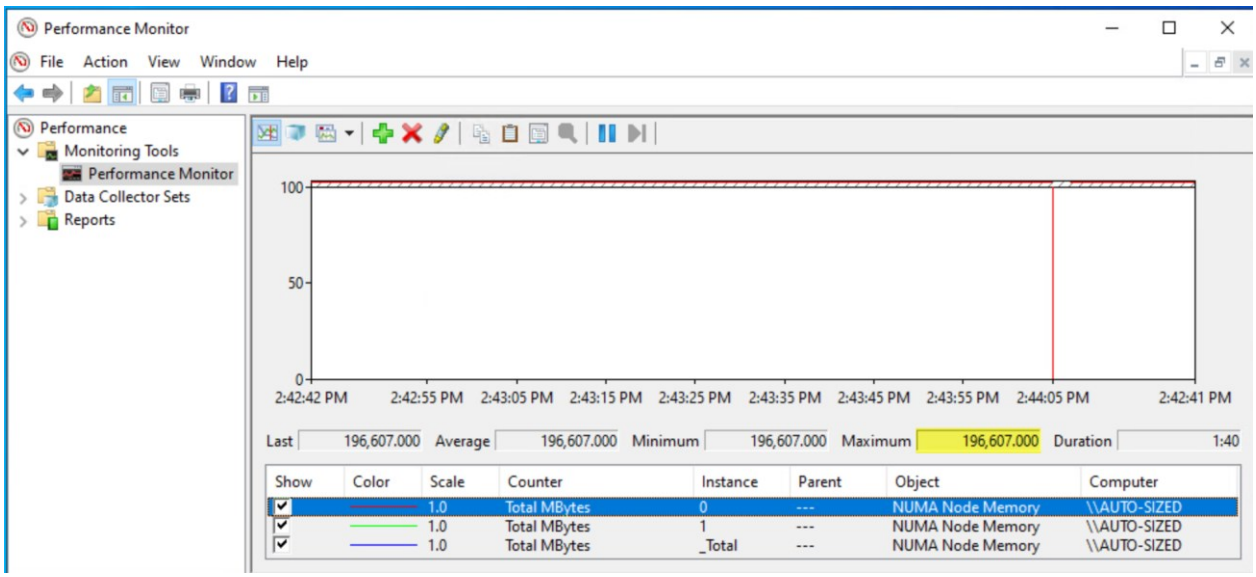
- We are now able to see TWO virtual NUMA nodes in the Guest Operating System.

Figure 25. Manual vNUMA Configuration, As Seen in Windows



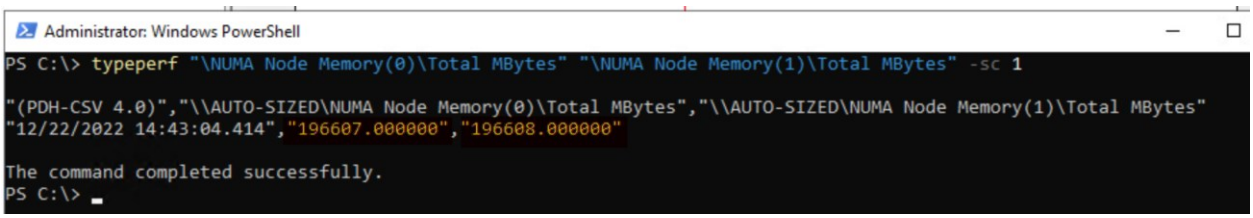
- In Windows Performance Monitor, we can see that half of the allocated RAM has been allocated to each of the NUMA nodes.

Figure 26. Manual vNUMA Topology Presents Balanced Topology



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

Figure 27. Allocated Memory Evenly Distributed Across vNUMA Nodes



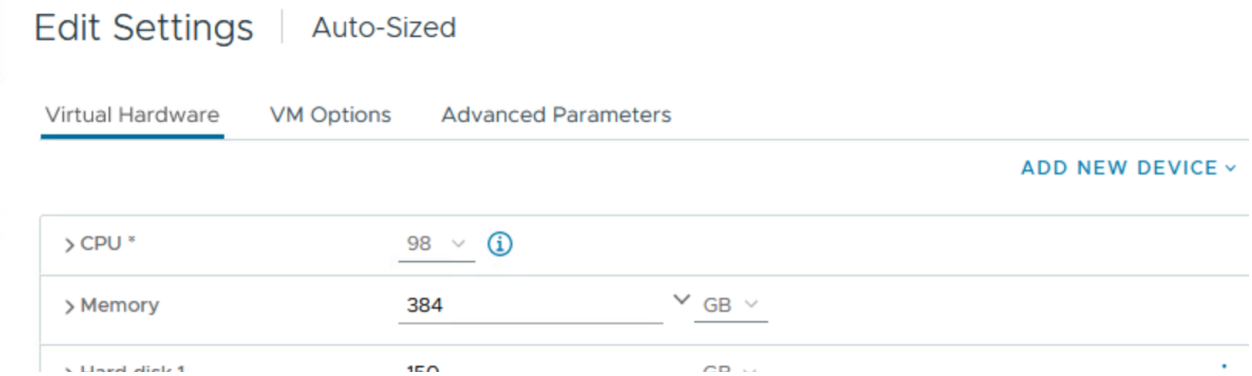
This is a much better presentation which substantially improves performance for our SQL Server instance.

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.

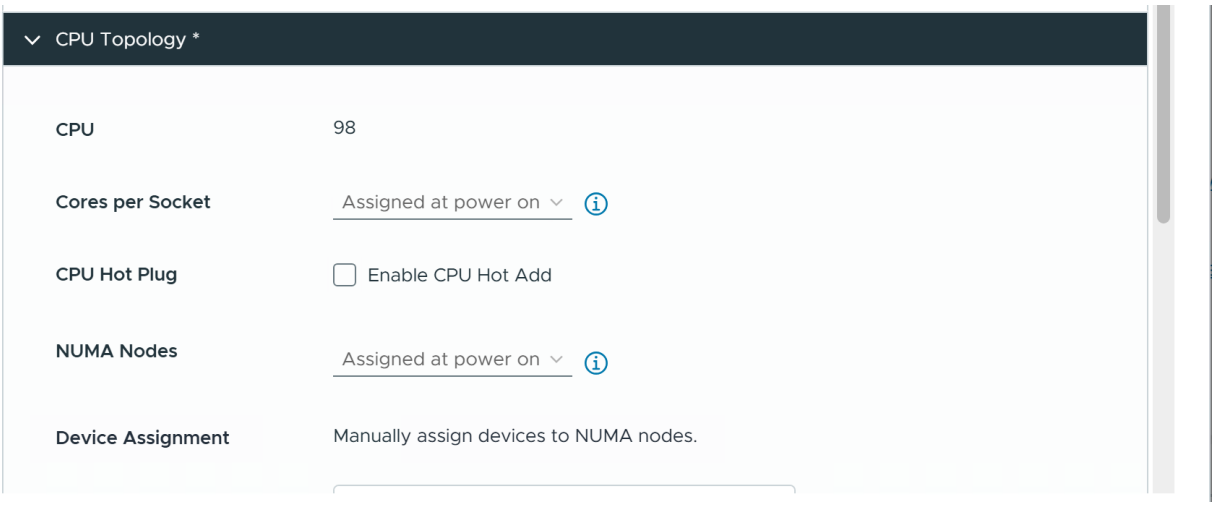
Bearing in mind that our ESXi has 128 logical processors (2x(32+hyperthreads)), the rough math says that there are 64 logical processors in each physical NUMA node. We’re going to allocate more than 64 vCPUs to our VM this time, leaving the memory at 384GB so that both compute resources exceed what is physically available in the Host’s NUMA node.

Figure 28. Allocating Super-cluster (or Cross-cluster) CPUs to Wide VM



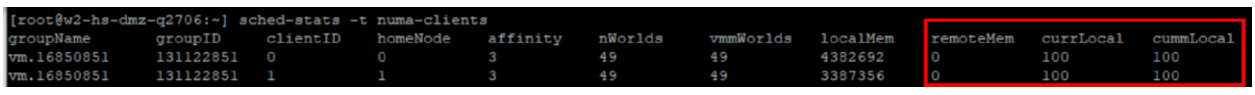
We will also accept ESXi’s default behavior and let it assign whichever topology it deems optimal for this configuration.

Figure 29. Configuring Automatic vNUMA Presentation for Wide VM



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.

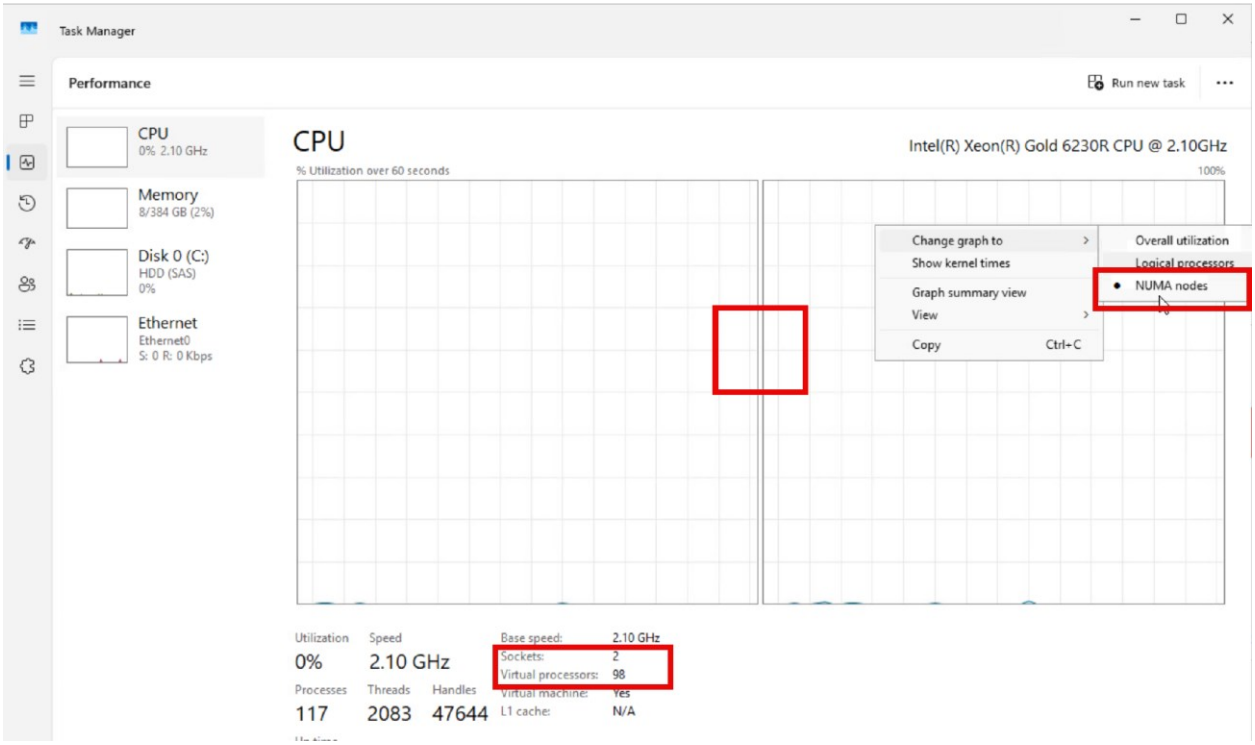
Figure 30. Balanced vNUMA Presentation, As Seen in ESXTop



```
[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 31. vNUMA Topology, As Seen in Windows



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

Figure 32. Wide VM Memory Distribution in Automatic vNUMA Configuration

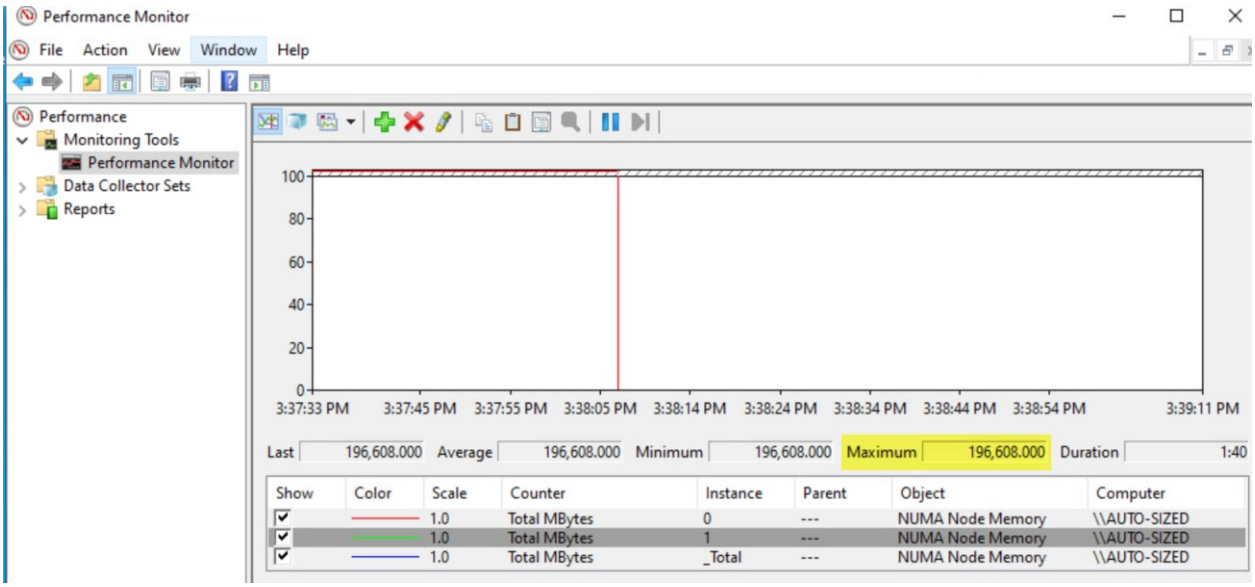
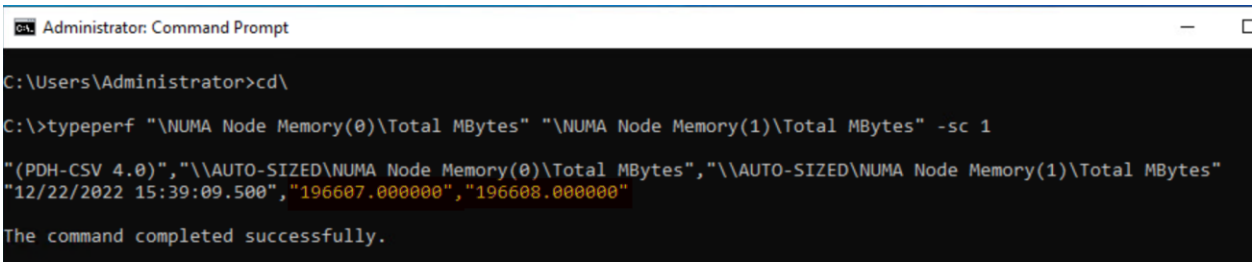


Figure 33. Wide VM Memory Distribution in Automatic vNUMA Configuration, As Seen in Windows



This presentation is optimal and balanced for our SQL Server.

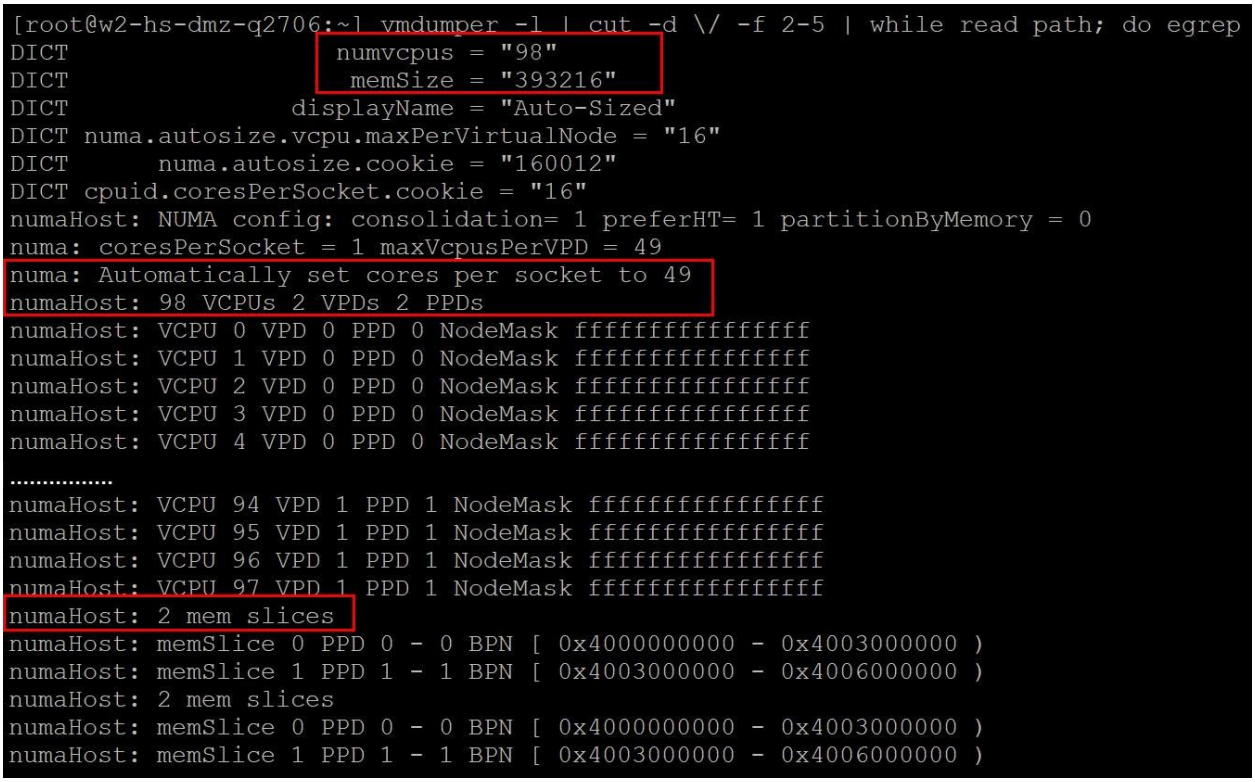
We have gone through these detailed examples to demonstrate the capabilities and behaviors of the new vNUMA topology configuration options in VMware Cloud Foundation (VCF). It is expected that this feature will continue to be refined and optimized in subsequent versions, and it is our intention to update 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:

```
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 34. Checking NUMA Topology with vmdumper



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.

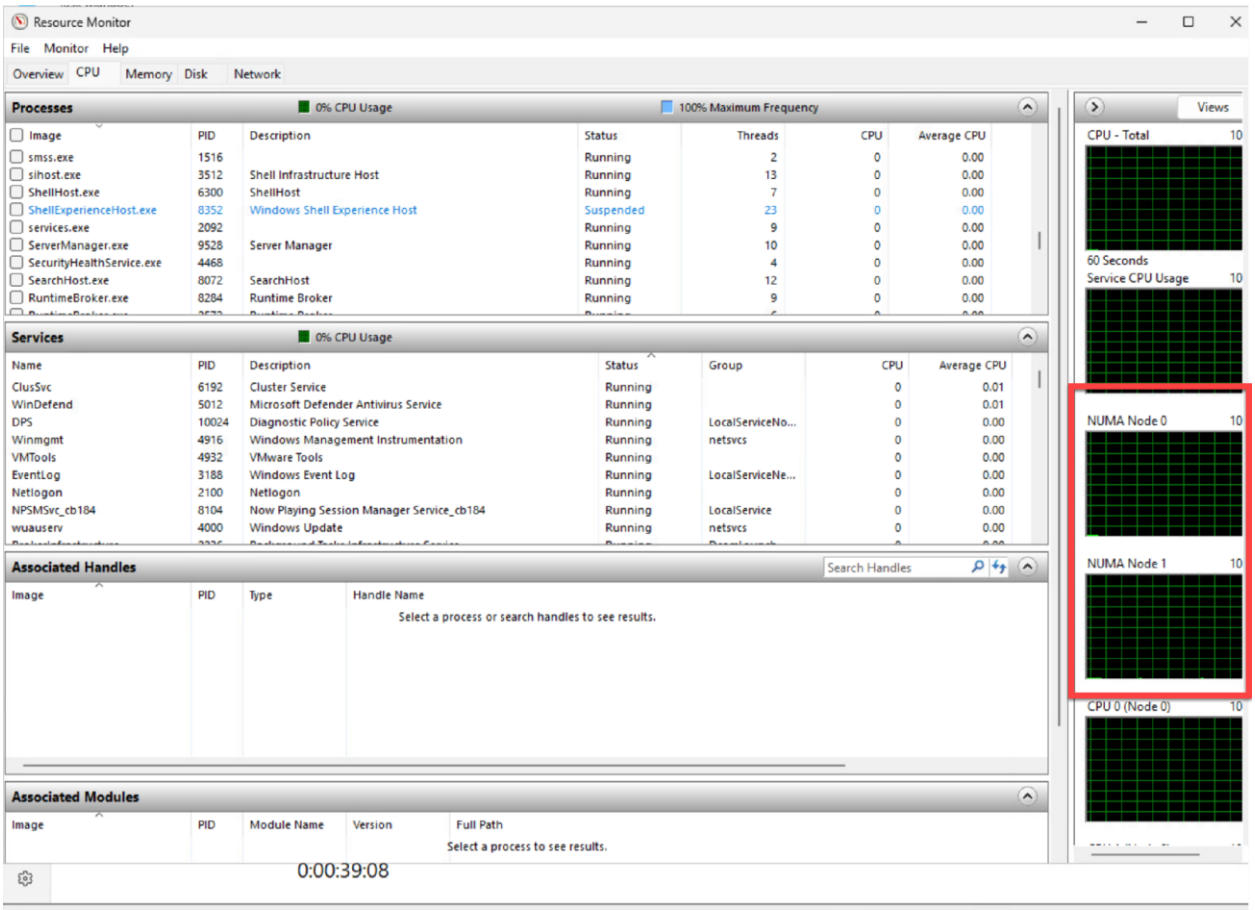
In-Guest operating system

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 through Windows Task Manager, Resource monitor or Coreinfo.exe (a downloadable tool originally from Sysinternals, now owned by Microsoft).[31].

Figure 35. Windows Server Resource Monitor NUMA Topology View



[31] Coreinfo

Figure 36. Windows Task Manager NUMA Topology View

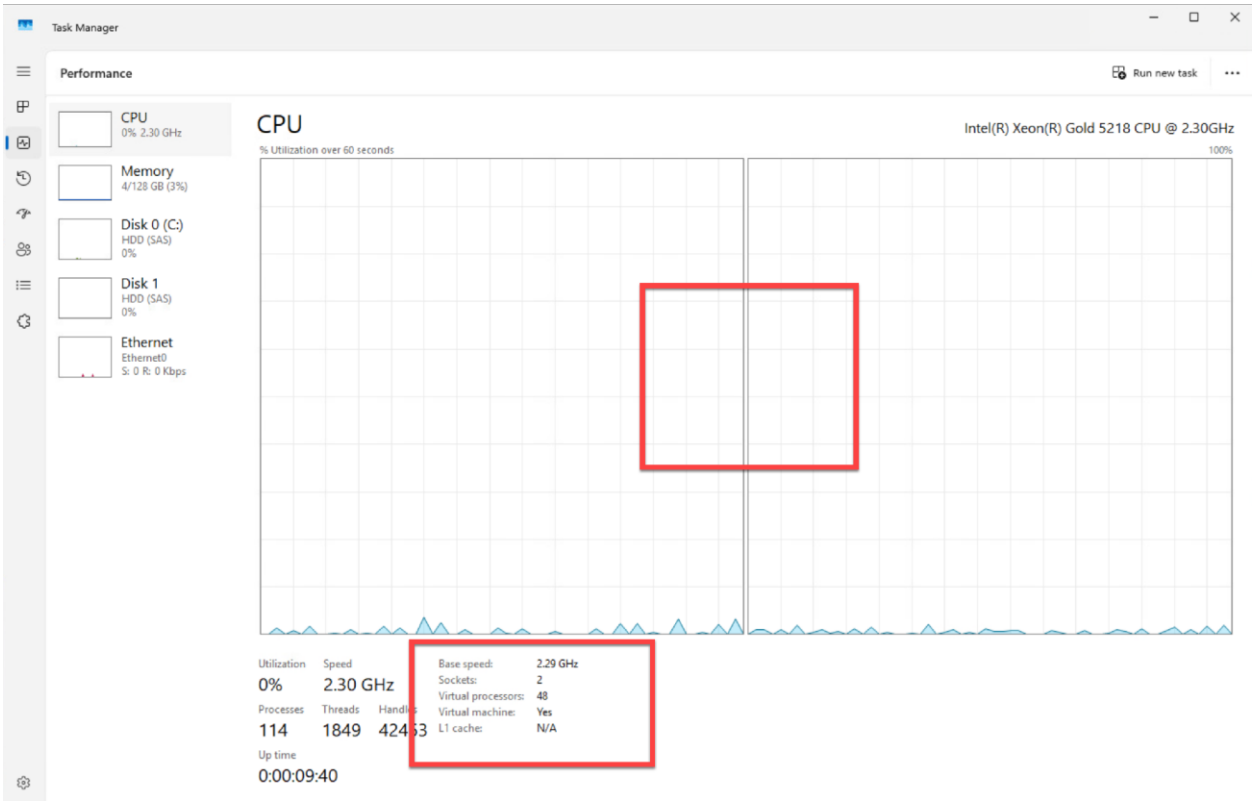
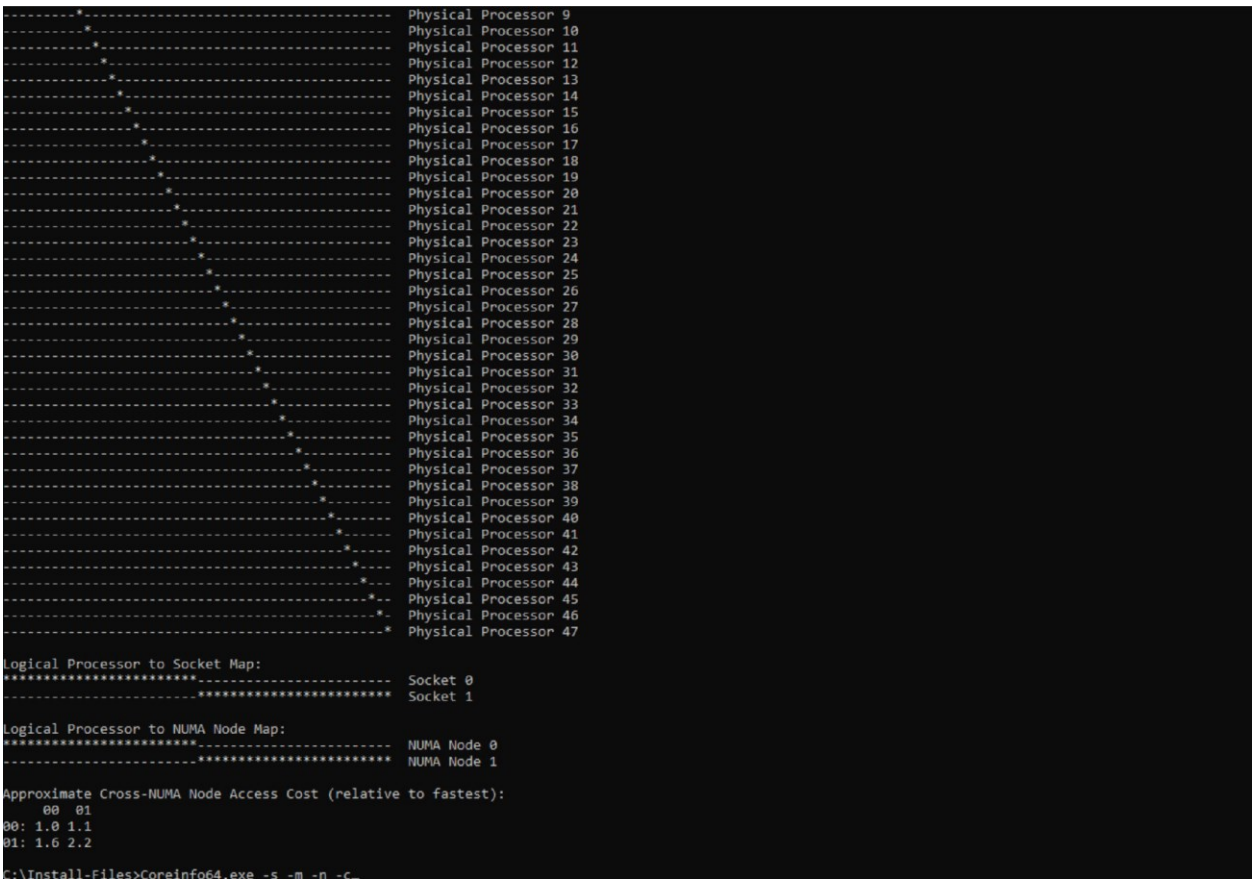


Figure 37. Core Info Showing a NUMA Topology for 24 cores/2socket VM



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[32]. Following utilities can be used to check the NUMA topology exposed.

- Numactl[33]

  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 38. 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 39. 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
fffffff] -> [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

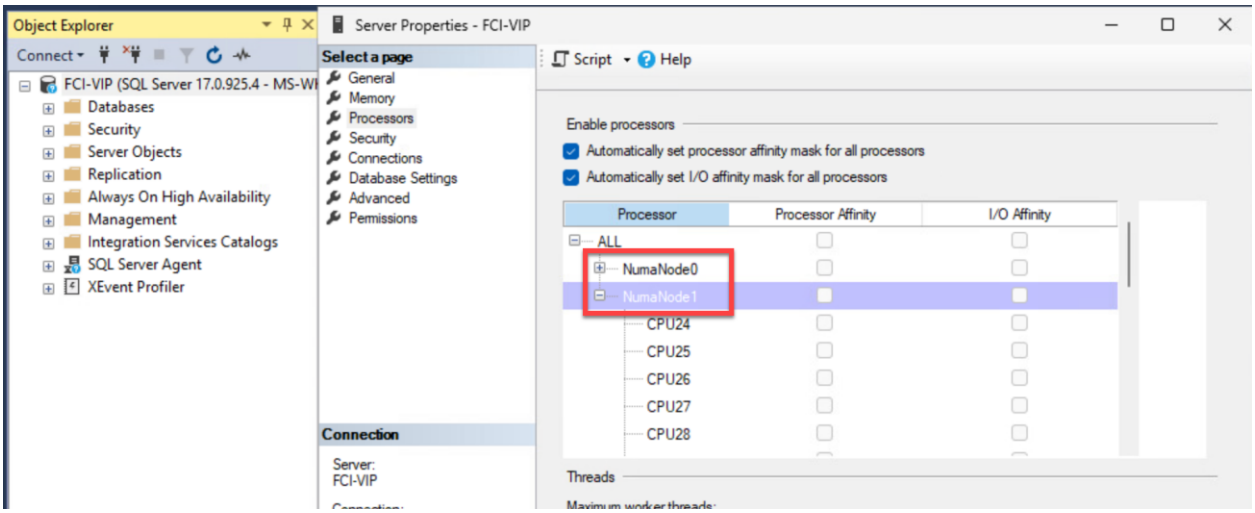
SQL Server

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:

[32] See here for a full list of the supported Operating Systems
[33] Numactl is not available in the standard OS and should be installed using the applicable OS tools or commands

Figure 40. Displaying NUMA Information in the SQL Server Managemnet Studio

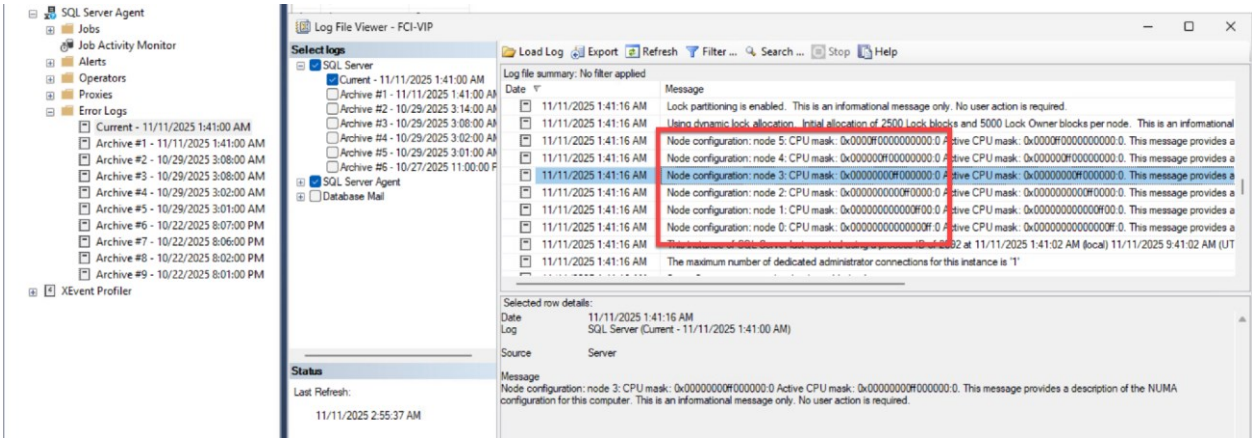


You can also view the information directly from SQL Server DMVs with the following command:

- *select memory_node_id,cpu_count from sys.dm_os_node*;

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

Figure 41. Errorlog Messages for Automatic soft-NUMA on 48 Cores per Socket VM



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[34].

As shown in the image above, although SQL Server has partitioned the 48 vCPUs allocated to this VM into 6 “Soft NUMA Nodes” of 8 cores each, a close examination shows that 3 of these (Soft-NUMA) Nodes are grouped into the same CPU mask boundary. There are two such boundaries, corresponding directly with the two vNUMA topologies presented to the Operating System.

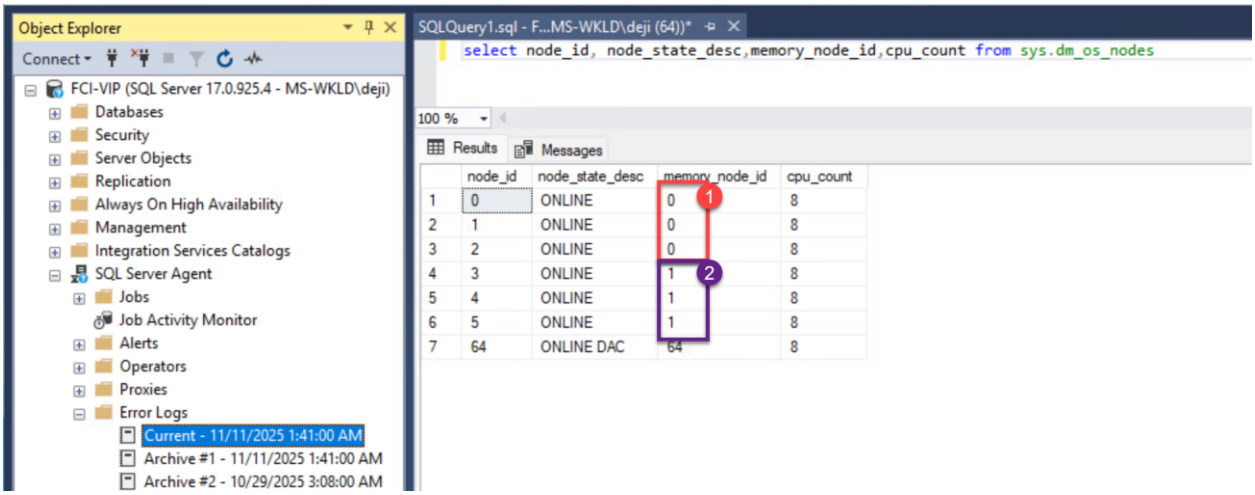
Starting with SQL Server 2014 SP2, “Soft-NUMA” is enabled by default and does not require any modification of the registry or service flags for the database service startup. In SQL Server, the upper boundary for starting the partitioning and enabling “Soft-NUMA” is eight (8) cores per NUMA node, although (as shown in the image below),

[34] See: SQL 2016 - It Just Runs Faster: Automatic Soft NUMA and Soft-NUMA (SQL Server)

each Soft-NUMA node may contain less than eight cores, depending on the number of allocated vCPUs (in our case, 48 vCPUs are divided into 6 smaller Soft-NUMA nodes, each with 8 cores).

The result of having automatic soft-NUMA enabled can be noticed in the errorlog (Figure 41 above) and using *sys.dm_os_nodes* system view (Figure 42 below).

Figure 42. sys.dm_os_nodes Information on a System with 2 NUMA Nodes and 6 soft-NUMA Nodes



Soft-NUMA partitions only CPU cores and does not provide the memory to CPU affinity[35]. 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 VMware Cloud Foundation (VCF), 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.

VCF 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.

3.5.5 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[36], 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.

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.

3.5.6 CPU Hot Plug

NOTE: *The CPU Hot-Add feature is no longer supported in Microsoft SQL Server 2025. This feature has been deprecated and will be removed entirely in future versions. This Section is retained solely for references, as the feature remains supported in SQL Server versions up to 2022.*

Do not enable CPU Hot Add on a VM running Microsoft SQL Server 2025.

[35] How It Works: Soft NUMA, I/O Completion Thread, Lazy Writer Workers and Memory Nodes
[36] Compute capacity limits by edition of SQL Server

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 in VMware Cloud Foundation (VCF), enabling 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.

With this new vSphere Hot-add capabilities and improvements added in the Windows Server 2022 Operating System, the issues described in the following references are no longer applicable.

For older versions of Windows and VMware vSphere, VMware continues to recommend that Customers do *not* enable CPU hot plug 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)

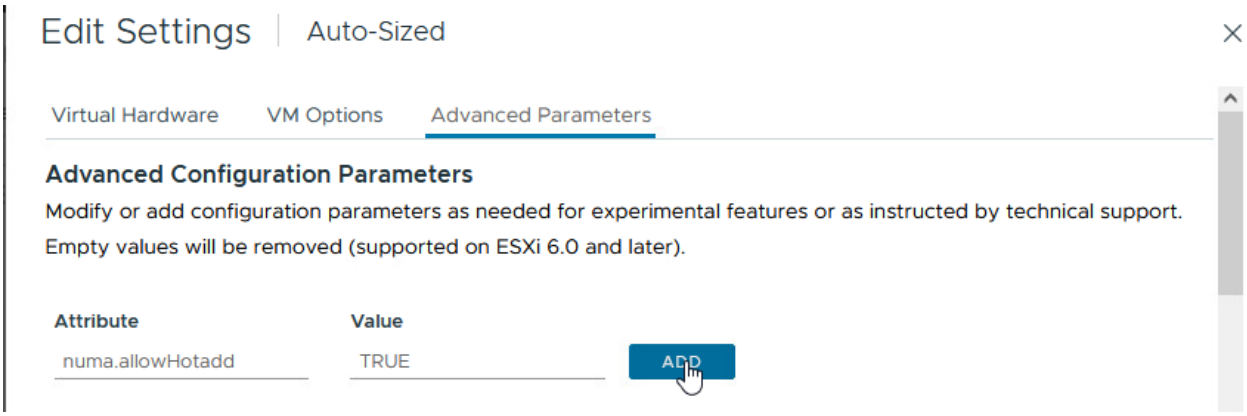
CPU HotAdd for Windows VMs: How BADLY Do You Want It?

3.5.7 Configuring CPU Hot Plug in VMware Cloud Foundation (VCF)

This Section provides a high-level demonstration of how Administrators can leverage the new CPU Hot add capabilities in VMware Cloud Foundation (VCF) to improve performance and simplified administration for their SQL Server instances on VCF. 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 in order to enable CPU Hot Add on a VM without disabling virtual NUMA for the VM.

Figure 43. CPU HotAdd VM Advanced Configuration

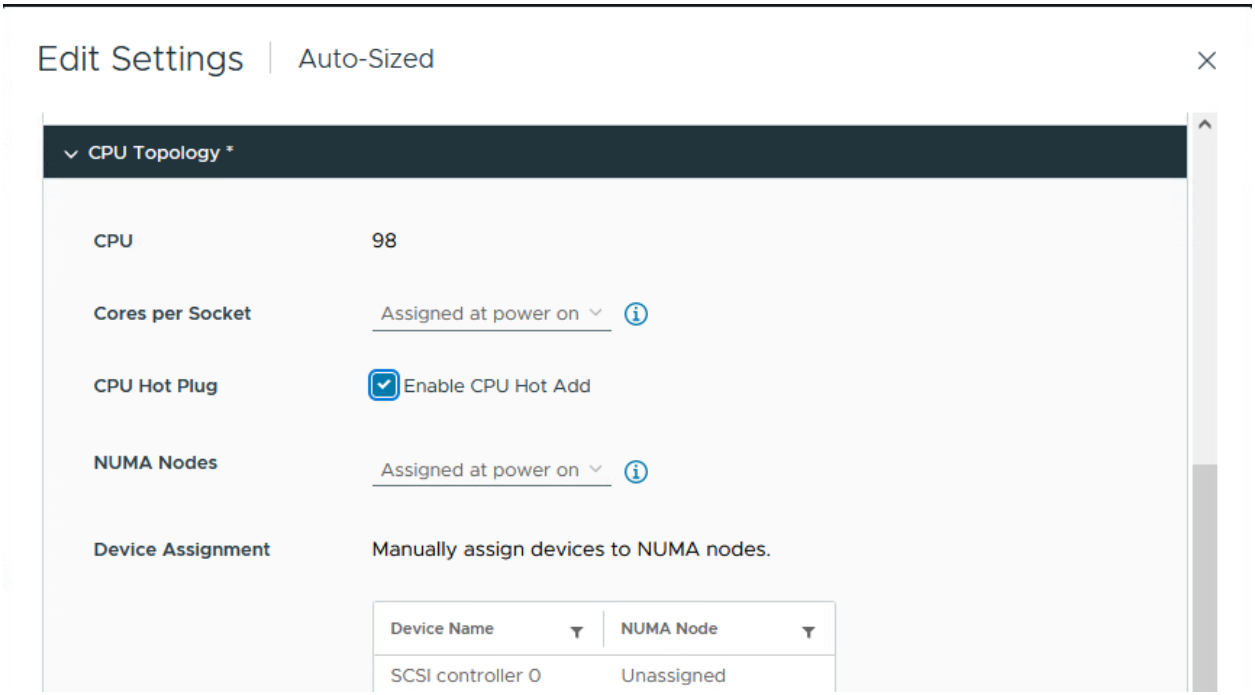


Once this attribute is configured, the VM is now ready to support CPU Hot Add.

As with the rest of CPU-related configuration options, CPU Hot Add is now configured in the "CPU Topology" section of the "VM Options" tab.

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

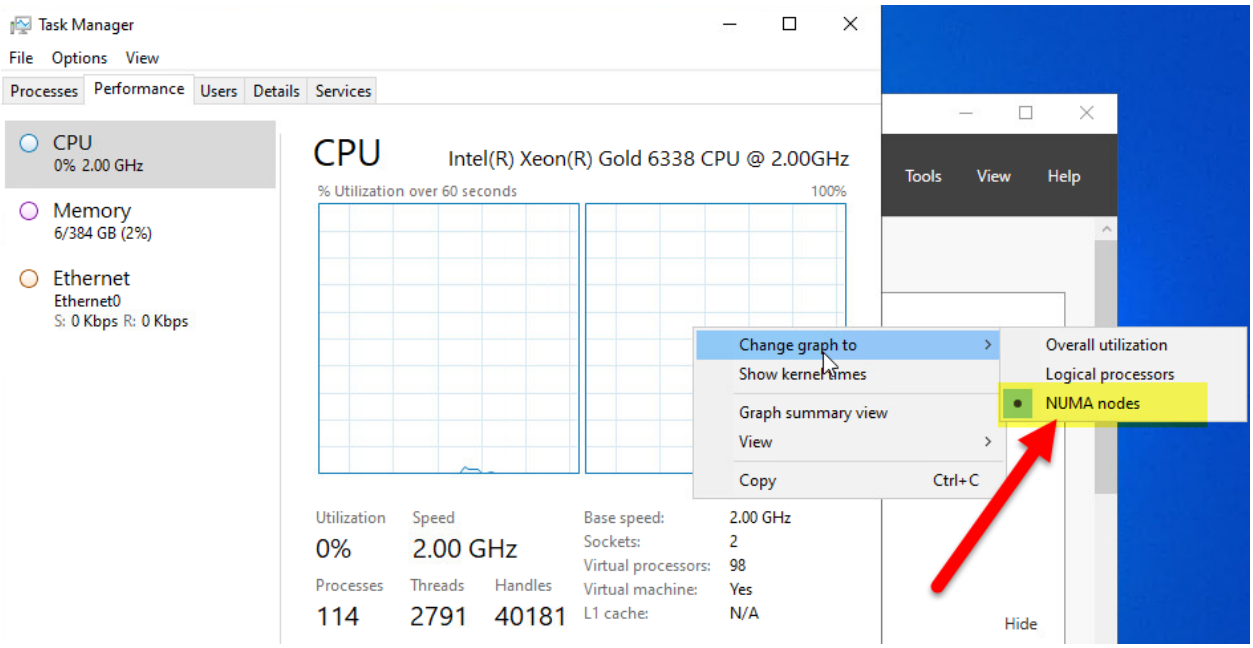
Figure 44. Enable CPU Hot Add on VM



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 45. vNUMA Available with CPU HotAdd



In the images 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 46. Manually Configured vNUMA Topology

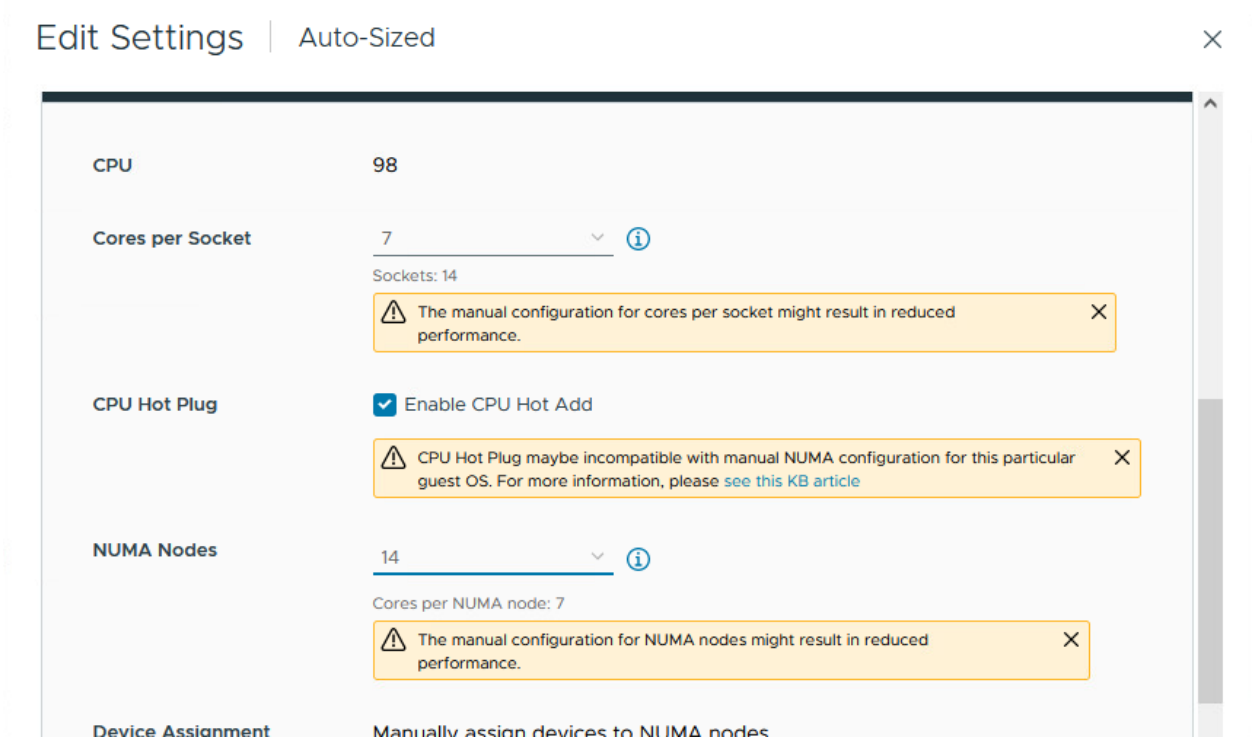
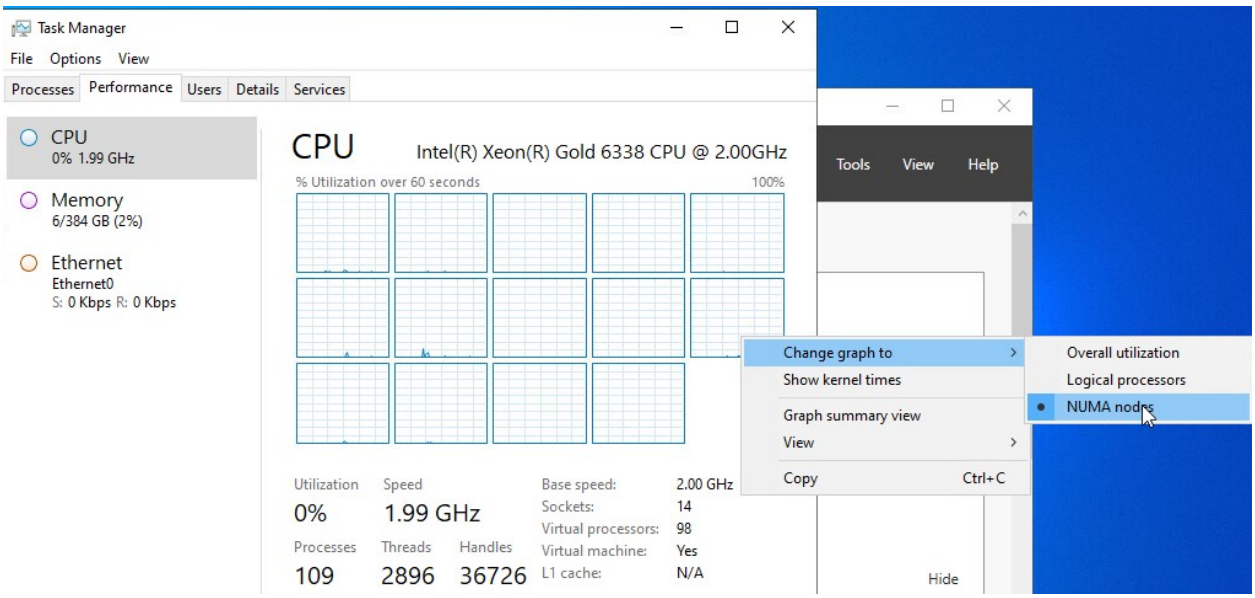


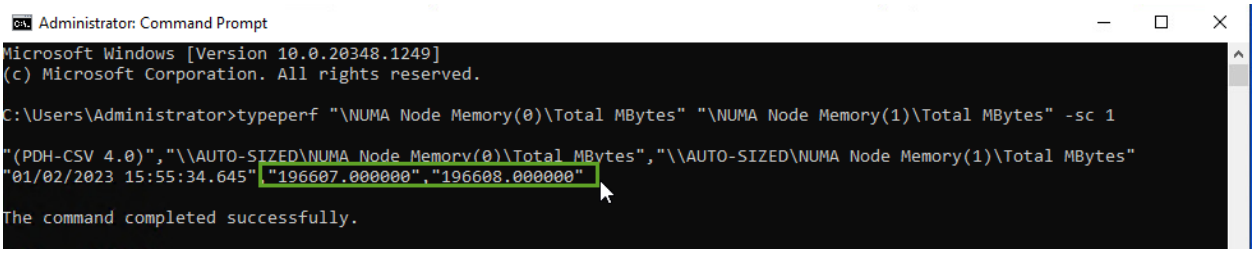
Figure 47. Defined Topology Mirrored in Windows



3.5.7.1. CPU Hot Plug and "Phantom Node" in VMware Cloud Foundation (VCF)

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 VMware Cloud Foundation (VCF) and Windows Server 2022.

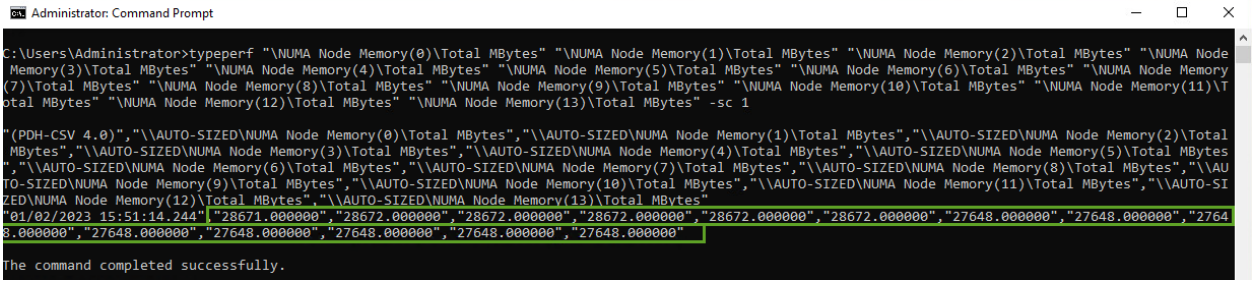
Figure 48. No "Phantom Node" with CPU HotAdd



Here, we see just two nodes created by Windows, exactly as presented by ESXi. In addition, the allocated memory is evenly divided between the Nodes, ensuring that the possibility of remote memory fetching is minimized, or entirely avoided.

We also see that the same is true for the corner cases in which we override ESXi's default choices and manually specified our desired vNUMA configurations:

Figure 49. Still no "Phantom Node" in Manual vNUMA Topology Configuration



3.5.8 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.

3.5.9 Per VM EVC Mode[37]

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 in order 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 in order 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.

[37] More details: How to Configure the EVC Mode of a Virtual Machine

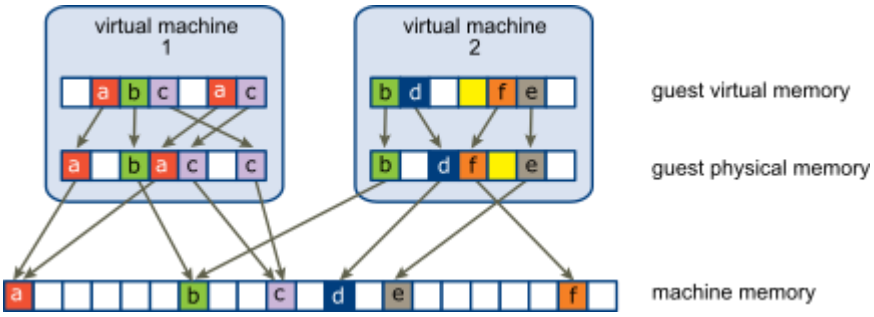
3.6 Virtual Machine Memory Configuration

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[38]. 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.[39]

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



3.6.1 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.

3.6.2 Memory Overhead[40]

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.

[38] More details and architecture recommendation for SQL Server can be found here: Memory management architecture guide
[39] Memory Virtualization with vSphere

[40] Understanding Memory Overhead

The version of vSphere or VCF deployed can also affect the amount of memory needed. ESX automatically calculates the amount of overhead memory needed for a virtual machine. In order 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.

3.6.3 Memory Reservation

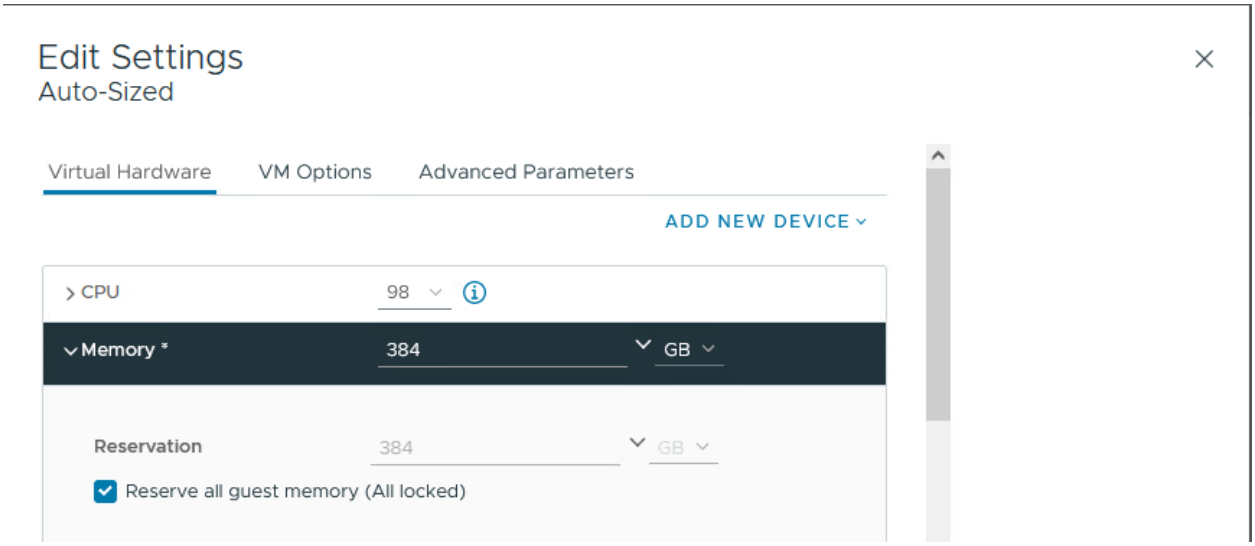
When achieving sufficient performance is the primary goal, consider setting the memory reservation equal to the provisioned memory. This will eliminate the possibility of ballooning or swapping from occurring and will guarantee that the VM will have exclusive access to all its reserved memory, even when 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 51. Setting Memory Reservation



Note 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.

Note 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[41] to reclaim physical memory backing for the swapped portion.

3.6.4 Memory Limit

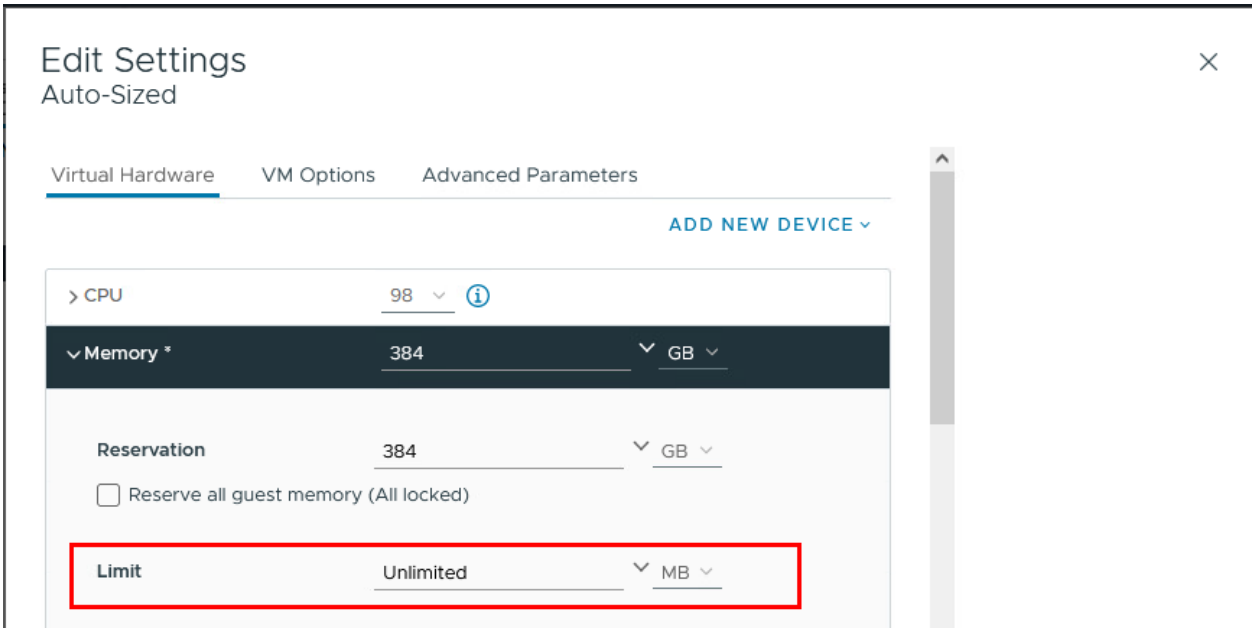
In contrast with Memory Reservation, which is beneficial to VMs and the applications they host, the Memory Limit setting impedes a VM's ability to consume all its allocated resources. This is because the limit is a rigid upper bound for a VM's entitlement to compute resources. You can create limits on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

vSphere and VCF 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 drawbacks 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 52. Configuring Memory Limit



3.6.5 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.

[41] The process and methods described here is not officially supported by Broadcom.

The balloon driver, which is installed with VMware Tools™[42], 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 VCF Operations Manager.

Note 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.

3.6.6 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[43]. With vSphere 6.5 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.

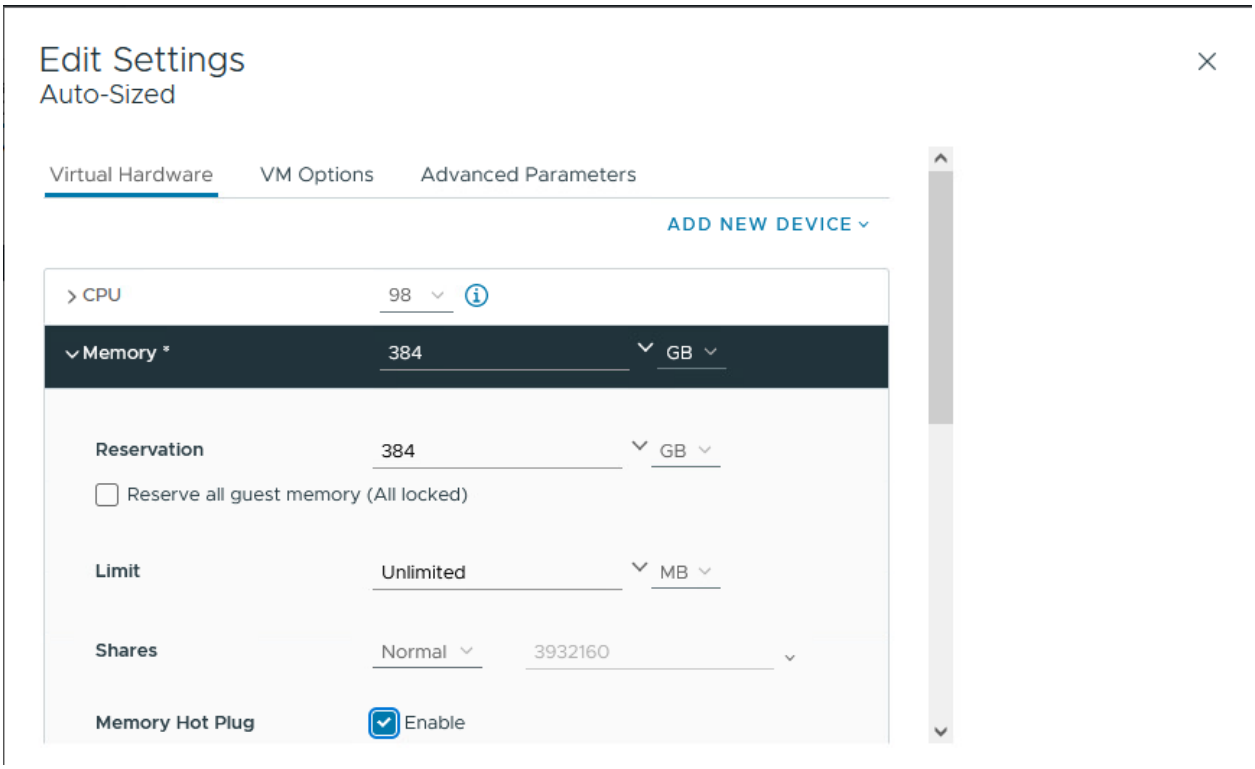
VMware recommends using memory hot plug-in cases where memory consumption patterns cannot be easily and accurately predicted. After memory has been added to the VM, increase the max memory setting on the instance if one has been set. This can be done without requiring a server reboot or a restart of the SQL Server service, unless SQL Server's large memory page 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.

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

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

Figure 53. Setting Memory Hot Plug



3.6.7 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.

Starting with VCF 9.0, this feature is now deprecated and no longer supported.

3.7 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.

3.7.1 VCF Storage Options

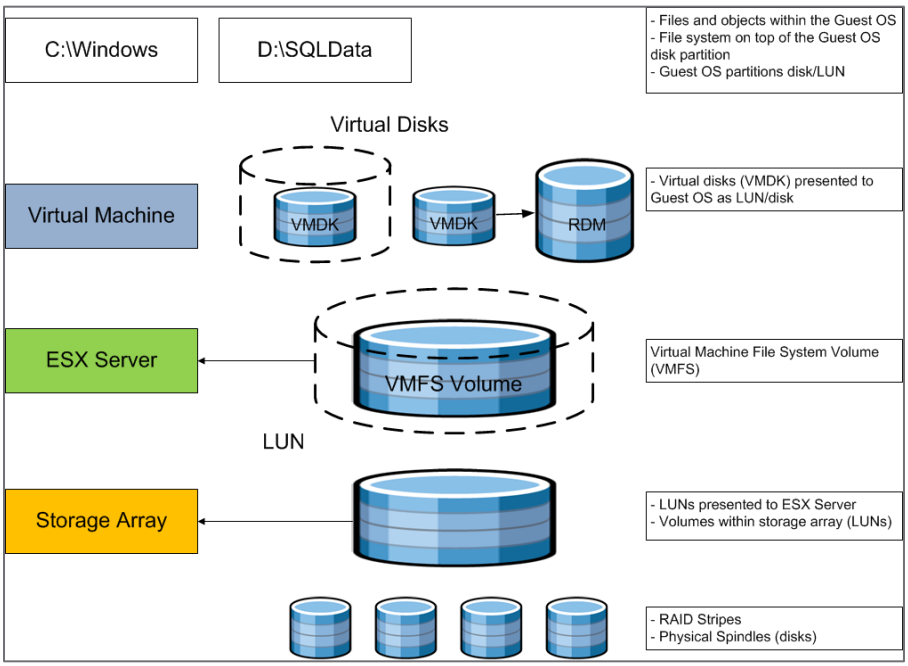
VCF provides several options for storage configuration. The one that is the most widely used is a VMFS formatted datastore on block storage system, but that is not the only option. Today, storage admins can utilize new technologies such as vSphere Virtual Volumes™ which takes storage management to the 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 VCF.

3.7.1.1. 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 54. VMware Storage Virtualization Stack



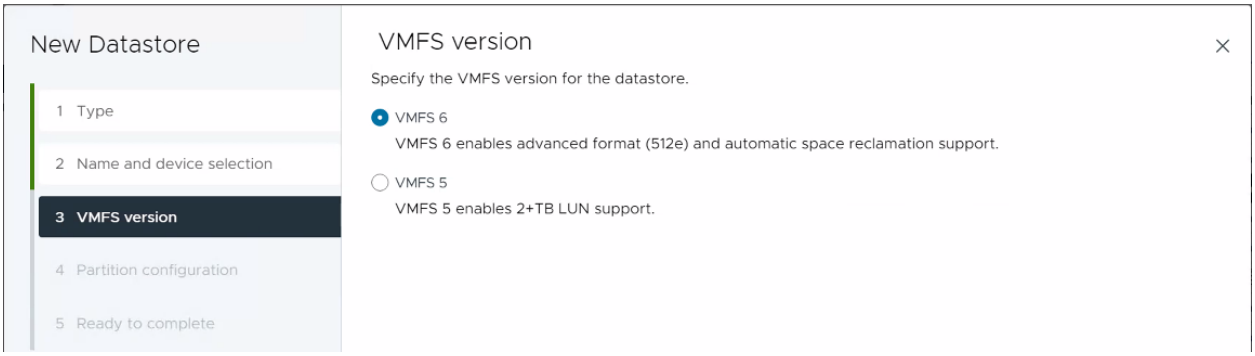
3.7.1.2. VMware Virtual Machine File System (VMFS)

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 and vSphere VMFS Datastore Concepts and Operations

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

Figure 55. VMFS6 vs VMFS5



3.7.1.3. Network File System (NFS)[44]

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.

[44] More details: Guidelines and Requirements for NFS Storage with ESX

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 Working with Datastores in vSphere Storage Environment.

VMware Cloud Foundation (VCF) now has the ability to validate NFS mount requests and NFS mount retries on failure.

3.7.1.4. NFS Datastores considerations:

VMware Cloud Foundation (VCF) 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

3.7.1.4.1. 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[45]. 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.

3.7.1.5. 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 VCF 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[46]. The following charts summarize some performance testing[47].

[45] More details: Description of support for network database files in SQL Server
[46] More details: PERFORMANCE CHARACTERIZATION OF MICROSOFT SQL SERVER ON VMWARE vSPHERE
[47] Mode details: Performance Characterization of VMFS and RDM Using a SAN

Figure 56. Random Mixed (50% Read/50% Write) I/O Operations per Second (Higher is Better)

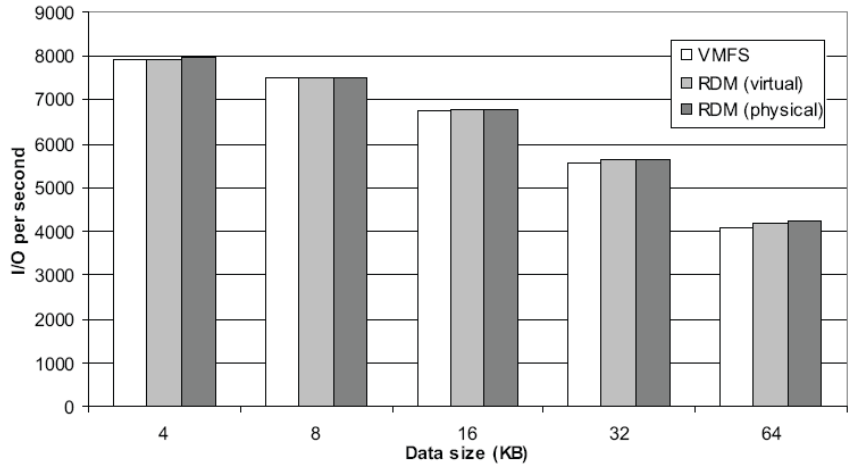
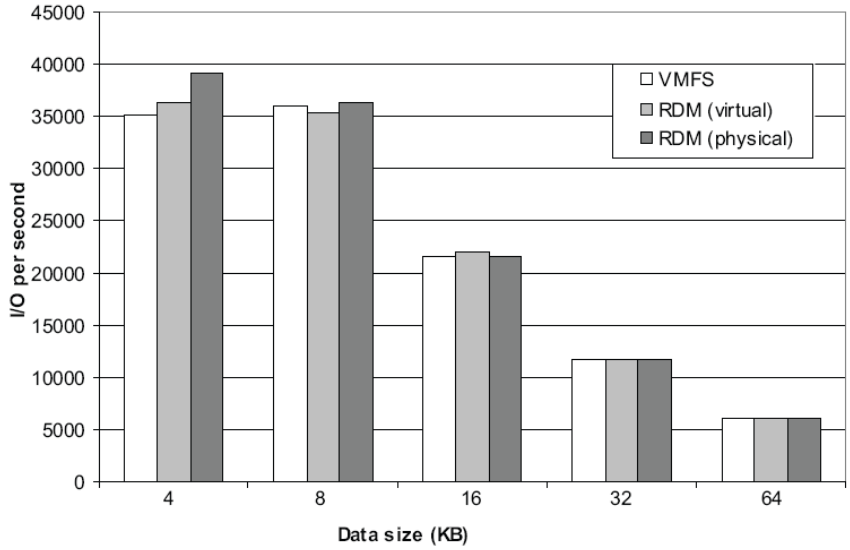


Figure 57. Sequential Read I/O Operations per Second (Higher is Better)



3.7.1.6. Clustered VMDK[48]

After closing the performance and size gaps between VMFS and RDM (both RDM and VMFS/VMDK can be up to 62TB in size)[49], 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[50].

Considerations and limitations for using Clustered VMDKs are detailed in the "Limitations of Clustered VMDK support for WSFC" section of VCF Product Documentation[51]

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.

[48] vSphere Storage
[49] VMware Configuration Maximums
[50] Setup for Windows Server Failover Clustering on VMware vSphere
[51]

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 58. Enabling Clustered VMDK (Step 1)

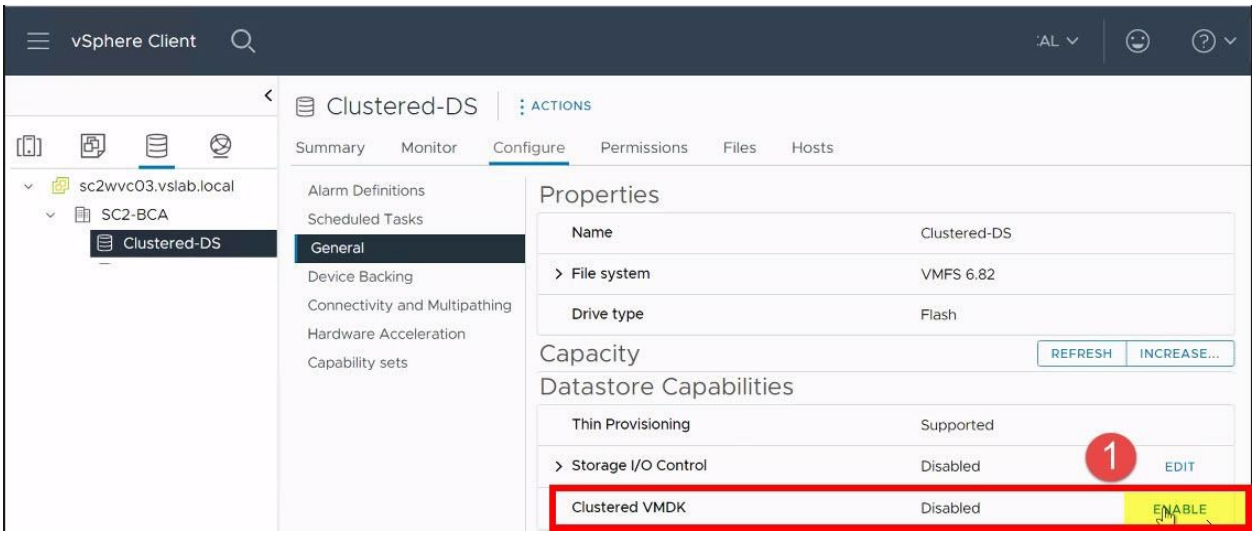
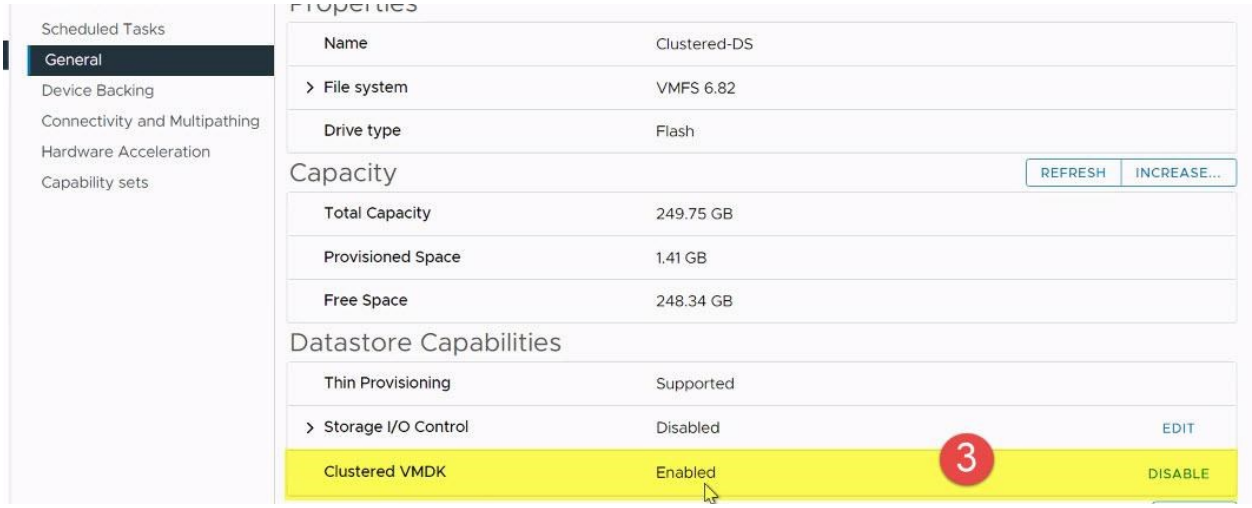


Figure 59. Enabling Clustered VMDK (Step 2)

Figure 60. Enabling Clustered VMDK (Step 3)



3.7.1.7. vSphere Virtual Volumes[52]

Important:

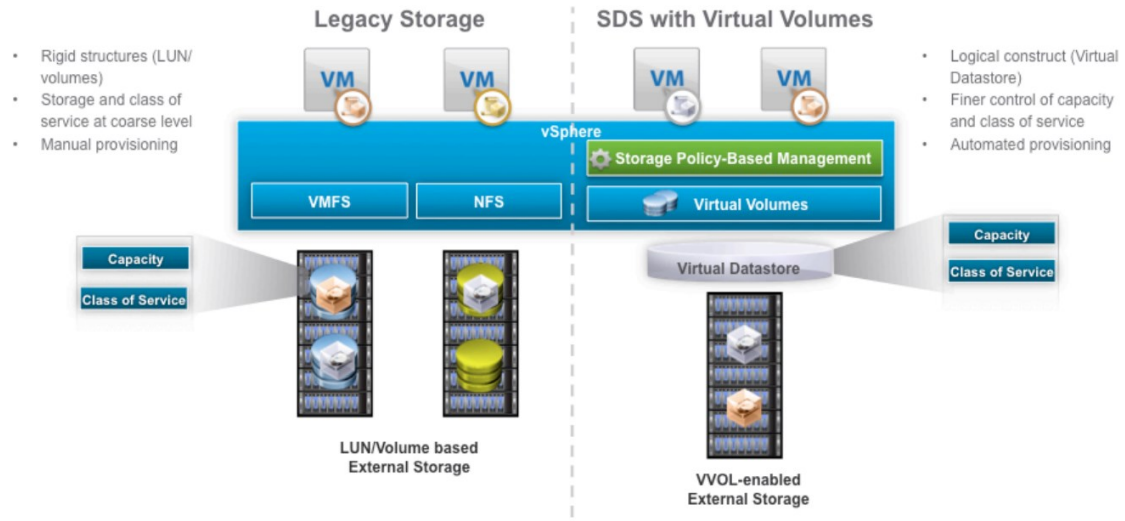
Starting with VCF 9.0 and VMware vSphere Foundation 9.0, the Virtual Volumes capability, also known as vVols, is deprecated and will be removed in a future release of VCF and VMware vSphere Foundation. Support for Virtual Volumes will continue for critical bug fixes only for versions of vSphere 8.x, VCF and VMware vSphere Foundation 5.x, and other supported versions until end-of-support of the respective release.

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.

[52] More details can be obtained here: Working with VMware vSphere Virtual Volumes

Figure 61. 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.

3.8 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.

3.8.1 Partition Alignment

Aligning file system partitions is a well-known storage best practice for database workloads. Partition alignment on both physical machines and VMFS partitions prevents performance I/O degradation caused by unaligned I/O. 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 9.0.

3.8.2 VMDK File Layout

When running on VMFS, virtual machine disk files can be deployed in three different formats: thin, zeroed thick, and eagerzeroedthick. Thin provisioned disks enable 100 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[53].

All flash arrays utilize a 100 percent thin provisioning mechanism to be able to have storage on demand. However, vSphere and VCF require 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.

3.8.3 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

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

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 VCF, 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.[54]

3.8.4 Using Storage Controller

Utilize the VMware Paravirtualized SCSI (PVSCSI) Controller as the virtual SCSI Controller for data and log VMDKs. The PVSCSI Controller is the optimal SCSI controller for an I/O-intensive application on VCF, 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[55].

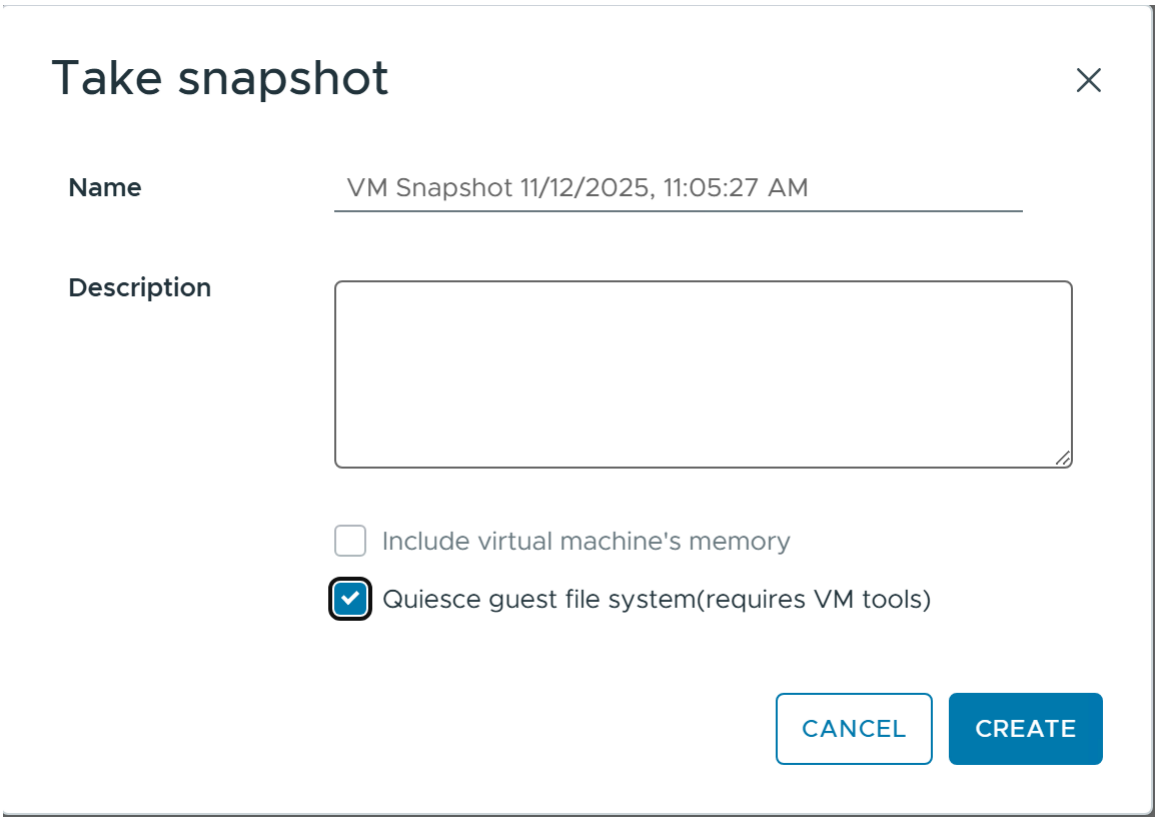
In vSphere 6.5 the new type of virtual controller was introduced – vNVMe[56]. It has since undergone multiple significant performance enhancements with each vSphere and VCF release. NVMe controller might bring performance improvement and reduce I/O processing overhead especially in combination with low 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.

[54] More details: vSphere Resource Management
[55] Consult 9.0.0 Configuration Limits for more details
[56] About VMware NVMe Storage

3.8.5 Using Snapshots

A VM snapshot preserves the state and data of a virtual machine at a specific point in time.[57] 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. 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 62. Take Snapshot Options



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[58]. Rely on SQL Server mechanisms to prevent data loss when in-memory data is lost.
    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

[57] https://kb.vmware.com/s/article/1015180
[58] Virtual machine becomes unresponsive or inactive when taking memory snapshot

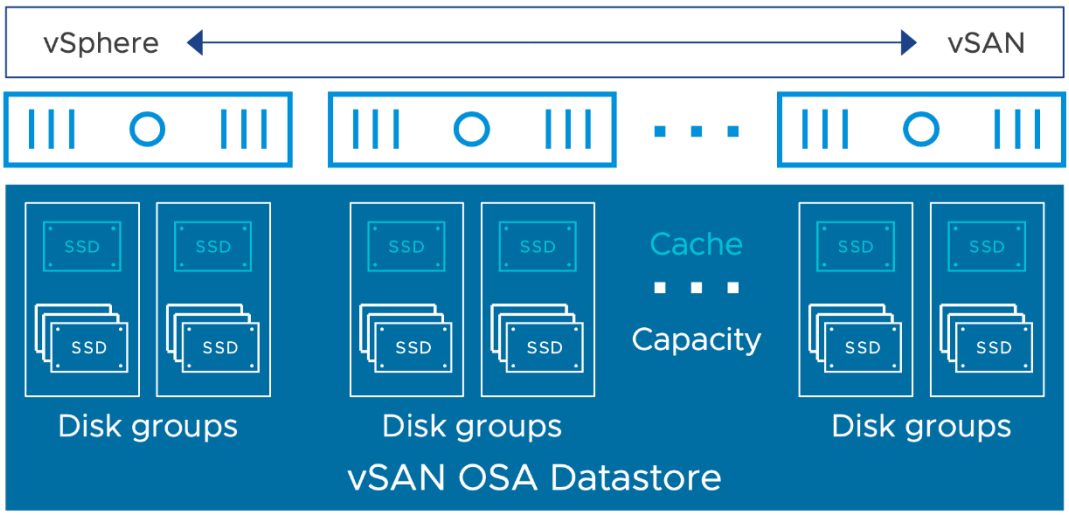
take a long time and can potentially cause performance issues[59]. Consider using snapshots operations for off-peak hours, or (better) use the offline for creating/removing snapshots.

3. Do not run a VM hosting SQL Server on a snapshot for more than 72 hours[60].
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.

3.8.6 vSAN Original Storage Architecture (OSA)[61]

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 VCF, 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 63. 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

[59] See Snapshot removal stops a virtual machine for long time and When and why do we "stun" a virtual machine?
[60] Best practices for using VMware snapshots in the vSphere environment
[61] More technical materials on SQL Server on vSAN can be found here.

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 Always On 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 allocated up front 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.

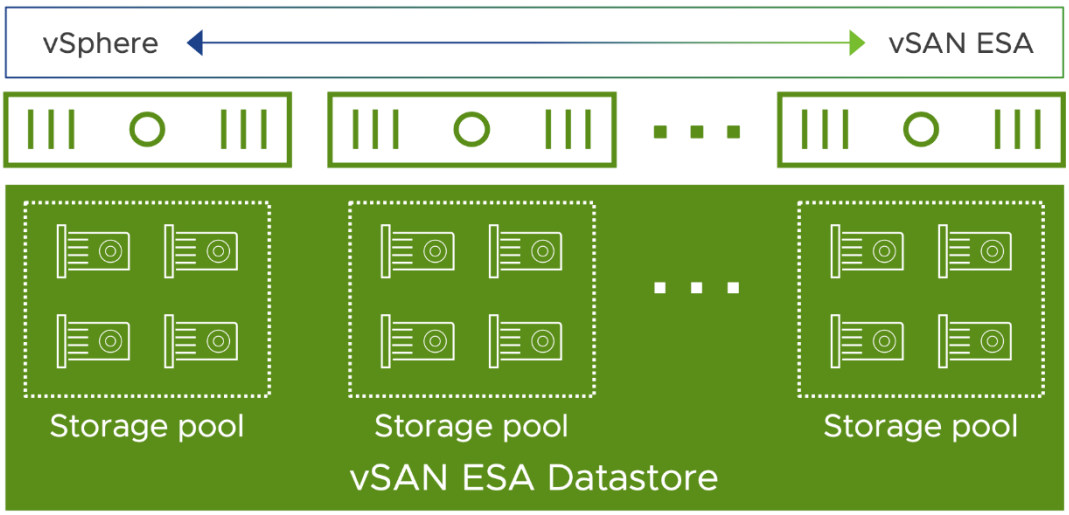
- VMware suggests setting the stripe width as needed to split the disk objects into multiple components to distribute the object components to more disks in different disk groups. In some situations, you might need this setting for large virtual disks.
- Use Quality of Service for Database Restore Operations. 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)[62].
- 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[63].

3.8.7 vSAN Express Storage Architecture (ESA)[64]

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 64. 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:

- vSAN Max - As the new disaggregated storage offering, vSAN Max provides Petabyte-scale centralized shared storage for your virtualized data assets. It delivers new capabilities, cost savings, and flexibility to the workloads running on VMware Cloud Foundation platform.

[62] What is vSAN, vSAN iSCSI Target Usage Guide and Using SQL Server Failover Clustering with vSAN iSCSI Target Service: Guidelines for supported configurations

[63] SQL Server Failover Cluster Instance on VMware vSAN Native

[64] Supported changes in a vSAN ESA or vSAN SC (Storage Clusters) ReadyNodes

- 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.
- Space Efficiency (global deduplication) – vSAN ESA in VMware Cloud Foundation introduces the new global deduplication technology, which is a simple and effective way to drive down storage costs in your virtualized environment without any significant tradeoff in performance for database workloads.
- 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.

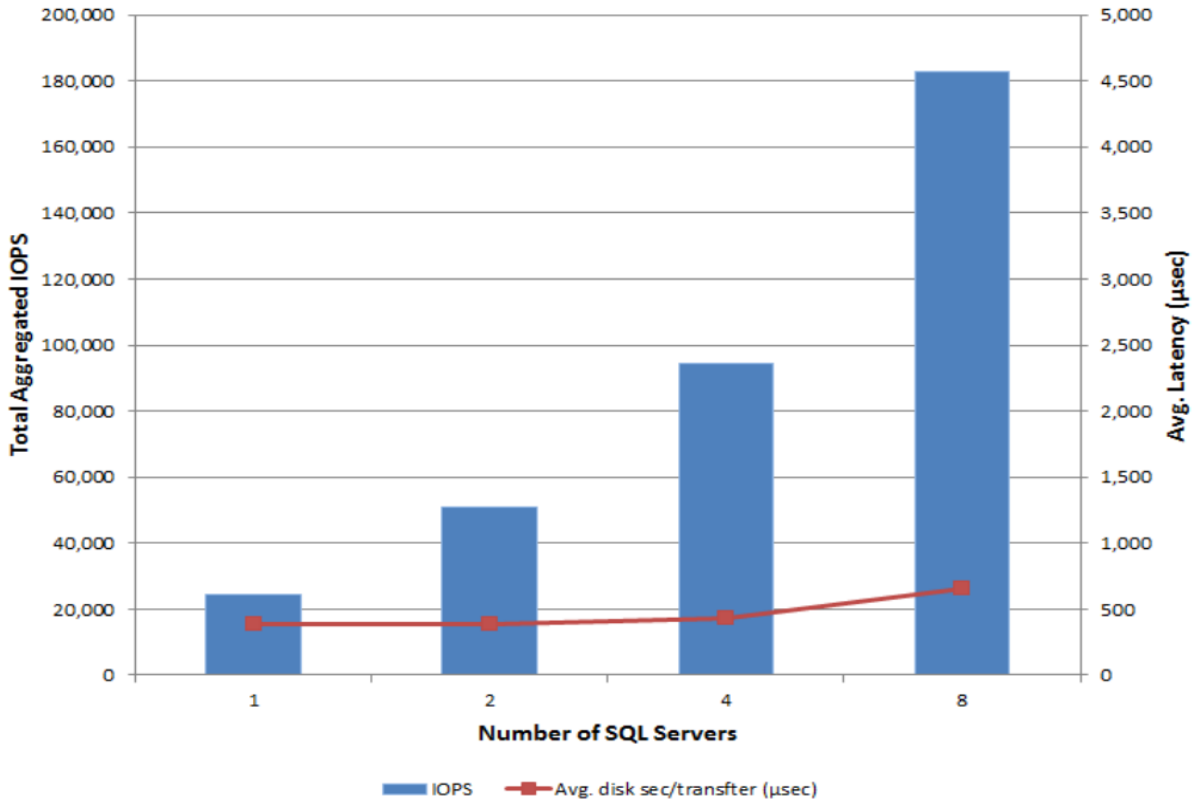
3.8.8 Considerations for Using All-Flash Arrays

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 65. XtremIO Performance with Consolidated SQL Server



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

3.8.8.1. 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.

3.8.8.2. 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

VCF, 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.

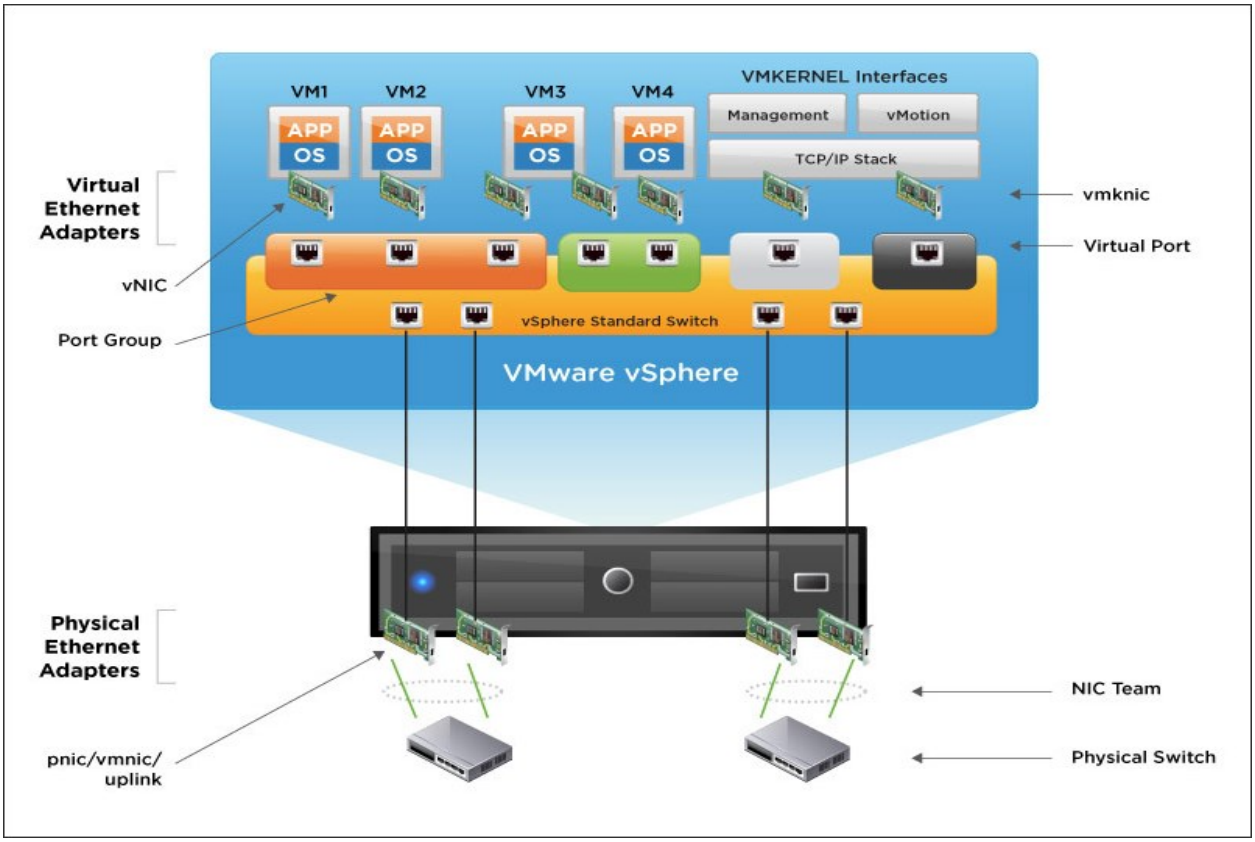
3.9 Virtual Machine Network Configuration

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

3.9.1 Virtual Network Concepts

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

Figure 66. 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.

3.9.2 Virtual Networking Best Practices

Some SQL Server workloads are more sensitive to network latency than others. To configure the network for your SQL Server-based VM, start with a thorough understanding of your workload network requirements. Monitoring the following performance metrics on the existing workload for a sample period using Windows Performance Monitor or the VMware Capacity Management feature within the VCF Operations Manager can help Administrators determine the resource requirements for an SQL Server VM very easily.

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. VCF 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 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 VMware vSphere 9.0 Guide.

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 Host Configuration for vSphere vMotion Guide

Figure 67. vMotion of a Large Intensive VM with SDPS Activated

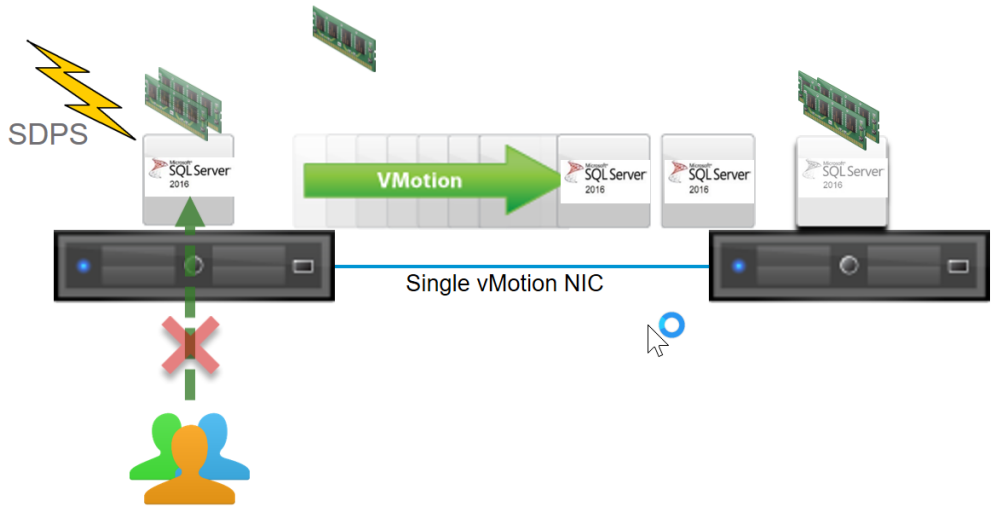
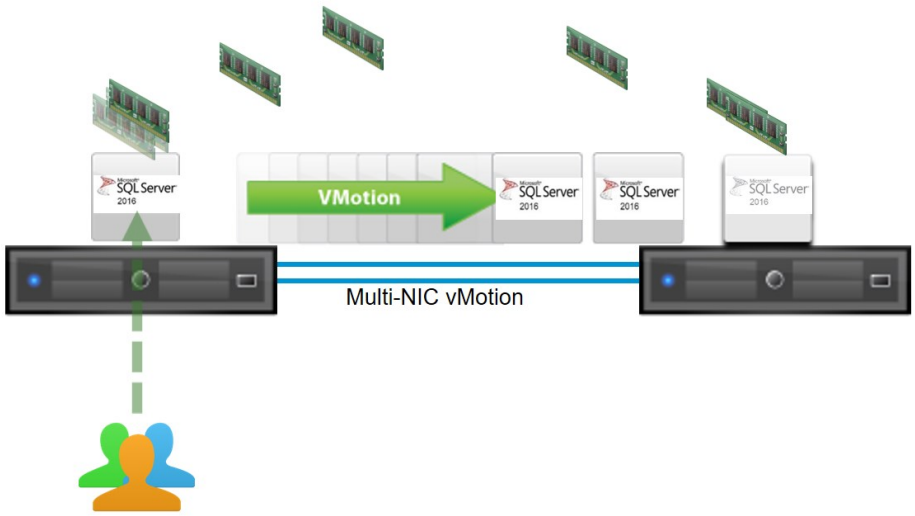


Figure 68. Utilizing Multi-NIC vMotion to Speed Up vMotion Operation



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 How to Tune vMotion for Lower Migration Times? and the "VMware Storage vMotion Recommendations" Section in this Guide.

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, VMware Cloud Foundation (VCF) introduced the vMotion Notifications 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 a detailed description of (and sample scripts for enabling and configuring) vMotion Notifications. For more details, refer to vMotion-App-Notif

3.9.3 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 VCF infrastructure unless the underlying physical network devices are configured to support this setting.

3.10 VCF Security Features

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

3.10.1 Virtual Machine Encryption[65]

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.

3.10.2 VCF Security Features[66]

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

---

[65] See for the latest performance study of VM encryption

[66] vSphere Security

- Support for a virtual Trusted Platform Module (vTPM) for the virtual machine
- Support for Microsoft Virtualization Based Security[67]
- 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 VMware Cloud Foundation (VCF), 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.

3.11 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 VCF 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 VCF platform VMware strongly recommends that Administrators:

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

3.11.1 Upgrade VMware Tools[68]

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 VCF 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.

3.11.2 Upgrade the Virtual Machine Compatibility [69]

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[70]. 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[71]:

- Take a backup of the SQL Server databases and Operating System
- Upgrade VMware Tools

[67] Virtualization-based Security (VBS)
[68] Overview of VMware Tools
[69] vSphere Virtual Machine Administration
[70] Upgrading a virtual machine to the latest hardware version (multiple versions)
[71]

- Validate that no misconfigured/inaccessible devices (like CD-ROM, Floppy) are present
- Use vSphere Web Client to upgrade to the desired version

Note 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.

3.12 Deploying SQL Server with VMware Data Services Manager

VMware Data Services Manager (DSM) automates the provisioning, management, and maintenance of Microsoft SQL Server databases. It runs SQL Server on Linux within Kubernetes to provide a standardized database-as-a-service environment.

DSM 9.1 introduces general availability of Microsoft SQL Server 2022 as a fully managed database-as-a-service (DBaaS) offering within VMware Cloud Foundation. DSM automates the complete SQL Server lifecycle — provisioning, patching, backup and restore, scaling, and clustering — without requiring specialized database administration expertise for routine operational tasks. SQL Server runs on Ubuntu nodes within managed Kubernetes clusters, a configuration that complies with the Microsoft technical support policy for SQL Server on Linux.

DSM separates server and cluster management from database management, reflecting the SQL Server architecture in which multiple independently managed databases share a single instance. Infrastructure administrators can provision and manage licensed SQL Server instances and server-level configurations such as Active Directory integration and high availability. Application teams create and manage databases on provisioned servers through a self-service interface. This model applies to both single-node configurations for non-critical workloads and three-node Always On availability group configurations for production environments

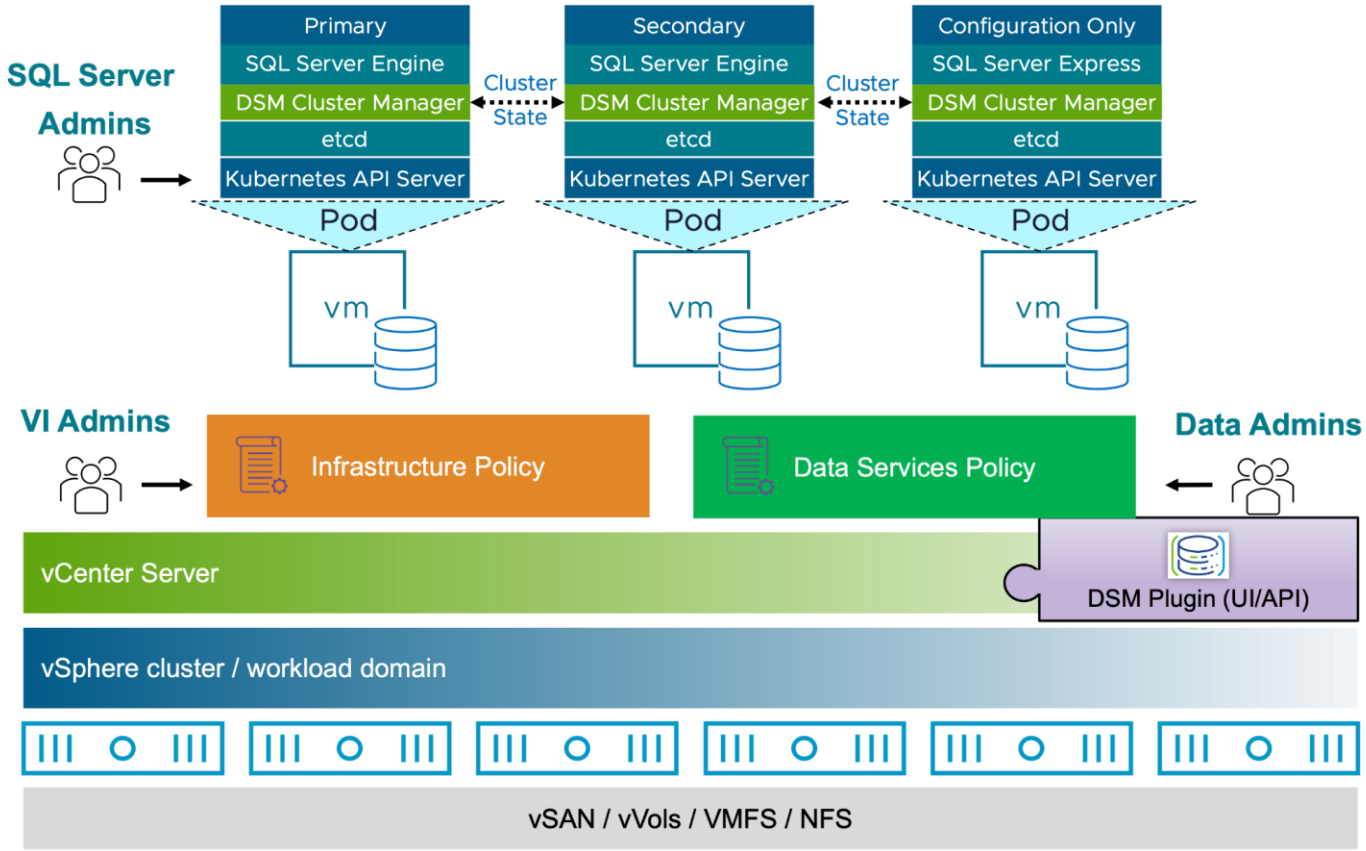
DSM provisions SQL Server VMs using predefined VM classes associated with vSphere Infrastructure Policies or Supervisor Services. A VM class defines the vCPU and memory combination available to a database VM. The infrastructure administrator defines and controls the set of VM classes, storage policies, and network resources that DSM-managed SQL Server workloads can consume.

3.12.1 SQL Server Deployment Architecture in Data Services Manager

SQL Server in Data Services Manager uses a distinct two-tier provisioning model in architectural design which decouples server and cluster management from database management. Under this separation, infrastructure administrators provision licensed SQL Server instances and handle server-level governance, such as Active Directory integration, resource boundaries, and High Availability (HA) topologies, while application teams independently create and manage user databases on those pre-configured servers. This model aligns with enterprise SQL Server usage patterns by allowing a single cluster—whether a single-node development instance or a multi-node production HA deployment—to efficiently host dozens of user databases. Additionally, DSM automates database backup and restore workflows for each database by leveraging integrated Microsoft native backup technologies, eliminating the need for manual setup or external tooling.

Figure 69 shows the cluster deployment for SQL Server in Data Services Manager. It utilizes a unified, Kubernetes-native DB Cluster Manager (dsm-cluster-manager) built on a shared control loop for SQL Server AlwaysOn Availability Group. This approach helps address the pinpoints of setting up traditional Linux and container based clustering solutions to ensure the database availability. Embedded alongside etcd Raft consensus and Kubernetes API components within the database VM nodes, this single cluster manager maintains leader leases, manages virtual IP endpoint floating, guarantees split-brain protection, and orchestrates zero-data-loss Microsoft Always On Availability Group failovers automatically.

Figure 69. Data Service Manager - SQL Server Cluster Deployment Architecture



3.12.2 Infrastructure & Sizing Guidelines

Compute sizing of SQL Server

Data Services Manager manages the compute resource sizing of SQL Server by VM Classes. The general guidance described in Section 3.3 also applies to SQL Server deployed with Data Services Manager.

A good starting point to configure the CPU-to-memory ratio within a VM class is 1 to 4 - one vCPU for every 4 GiB of memory - for general data services workloads. This ratio should be adjusted based on the specific workload profile: OLTP workloads with high concurrency may benefit from a higher vCPU allocation relative to memory, while analytics or data warehouse workloads typically demand a higher memory-to-vCPU ratio.

When the SQL Server VM is allocated more than eight vCPUs, ESXi automatically enables virtual NUMA (vNUMA) on that VM. vNUMA exposes the host's physical NUMA topology to the guest operating system, enabling SQL Server and the OS kernel to optimize memory access and CPU scheduling based on NUMA node boundaries. It is recommended to align VM vCPU counts with physical NUMA node boundaries to maximize memory locality and minimize cross-NUMA memory traffic. Refer to Section 3.5 for detailed guidance on vNUMA configuration for SQL Server virtual machines.

Data Services Manager enforces platform requirements for SQL Server memory. The VM class configured for any SQL Server cluster must allocate more than 4 GiB of memory. If the SQL Server cluster needs to host more than 20 databases the minimum memory requirement is 32 GiB, with a maximum of 50 databases per cluster regardless of memory. These thresholds are mandatory for SQL Server provisioned by Data Services Manager to function within its containerized runtime environment and cannot be bypassed.

Storage Policy-Based Management (SPBM) & vSAN Configurations

Data Services Manager governs SQL Server VM storage placement and resource guarantees through vSphere storage policies associated with DSM infrastructure policies. The infrastructure administrator defines and assigns storage policies; the appropriate policy is selected at provisioning time based on the target workload tier and availability requirements.

For VCF environments running vSAN Express Storage Architecture (ESA), VMware recommends erasure coding RAID-5 or RAID-6 as the storage policy configured for DSM-managed SQL Server workloads. Do not use RAID-1 Mirroring since it will consume more capacity but with minimal performance gain compared to RAID-5 and RAID-6 for vSAN ESA. Starting from VCF 9.1 release, vSAN ESA introduces Auto-RAID to determine the optimal data placement. Table XX summarizes the recommended vSAN ESA storage policy options.

- Use RAID-5 with FTT=1 for 3 to 5 hosts in a vSAN cluster, which is 1.5x (2+1) object capacity overhead
- Use RAID-6 with FTT=2 for 6 or more hosts in a vSAN cluster, which is 1.5x (4+2) object capacity overhead.

Network Prerequisites and configurations

Data Services Manager requires a structured and reliable port group topology on vSphere Standard (VSS), Distributed (VDS), or NSX Edge (N-VDS) switches, connecting components via a primary network interface (eth0). The Data Service Manager Appliance uses a manually assigned static IP on eth0 for the DSM UI, while the SQL Server database VMs draw IP addresses from configured IP pools to handle both management and client data traffic.

By default, DSM uses the following networks. You must not use these networks for any other purpose:

- Docker networks: 172.17.0.0/16, 172.18.0.0/16, and 172.19.0.0/16
- Workload cluster networks: 192.168.0.0/21 and 10.96.0.0/24

To maintain cluster health and security, deployed database nodes require unhindered routing to external infrastructure endpoints alongside synchronized time keeping and strict firewall rules. All deployed virtual machines must synchronize with a common NTP time source to ensure log sequencing, cluster quorum, and certificate validity across the environment. For more details about inter-service connectivity and network security requirements, please refer to Data Services Manager network requirements article.

Object Storage for database backup

Data Services Manager's backup architecture requires S3-compatible object storage external to VCF storages - such as vSAN Object Storage - and registering it as a TLS-enabled backup location in the Data Services Manager provider portal before provisioning any SQL Server instances. Deploying the object storage VM on vSAN provides storage-layer resiliency for database templates, backups, and provider repositories through the same storage policy management used for SQL Server workload VMs.

For clustered SQL Server deployment with HA, configuring a backup location is mandatory. Data Services Manager enforces this process in the data service policy and will not allow HA cluster provisioning without a valid and secure backup endpoint assigned to the governing data service policy.

3.12.3 Deploy SQL Server with Data Services Manager

SQL Server with Data Services Manager supports the following three types of deployment. Use the appropriate deployment model based on the SLA required by your organization.

Table 4. SQL Server Deployment Model in Data Services Manager

Deployment Model	Description
Single Server	Deploy standalone SQL Server instance with no clustering. An instance outrage will result in potential downtime
Single vSphere Cluster HA	Deploy three-node SQL Server Alwayson Availability Group with two synchronous-commit replicas and another configuration-only replica. DSM automatically manages primary election and failover operations. All replica nodes share the same vSphere Cluster and underlying storage. vSphere HA and DRS and any underlying storage redundancy make each individual database node highly available.
Cross vSphere Cluster HA	Deploy three-node SQL Server Alwayson Availability Group with two synchronous-commit replicas and another configuration-only replica across multiple vSphere clusters, providing enhanced availability and protection.

3.12.4 SQL Server Licensing consideration

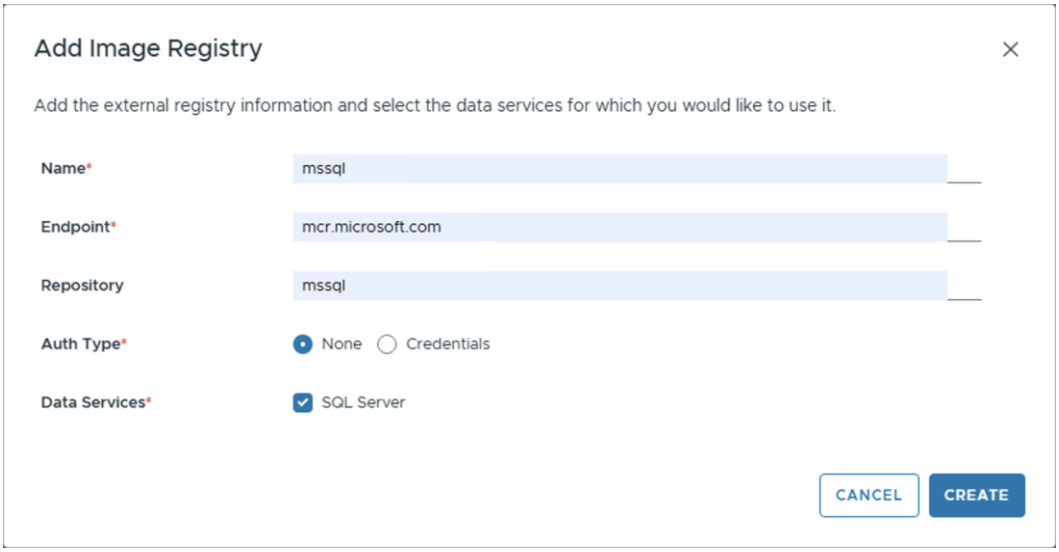
Data Services Manager orchestrates the SQL Server deployment with a declarative, Kubernetes-native API. Plan the SQL Server edition with the appropriate licensing before you deploy the SQL Server instances in Data Services Manager.

- Single-node / Standalone deployment: you may start with Express edition or Standard edition with up to 24 cores support. Use Enterprise edition for more cores demand within a single SQL Server instance.
- AlwaysOn Availability Group cluster HA deployment: use Enterprise edition for HA deployment for SQL Server in Data Services Manager. The configuration-only node in the SQL Server cluster setup is configured with express edition by DSM.
- Other test-dev deployment: evaluation edition supports 180 days trial only with complete feature set available. Use this only for temporary proof-of-concept deployments.

3.12.5 SQL Server Repository

The SQL Server Repository must be explicitly configured in the Data Services Manager portal to associate with the SQL Server data service. The following example configured the official SQL Server repository. DSM will automatically pull the required image and populate the available data service for SQL Server.

Figure 70. Image Registry Configuration for SQL Server



3.12.6 Active Directory Integration

Data Services Manager supports integration with existing active directory domain resources to enable Kerberos-based authentication for SQL Server. Administrators can choose between the following configuration mode:

- Automated mode - DSM leverages a privileged AD account to automatically handle SPNs, DNS updates, and service account management
- Manual Mode - can be used in restricted environments requiring pre-provisioned AD assets.

Table 5. Configuration Mode for Active Directory Integration

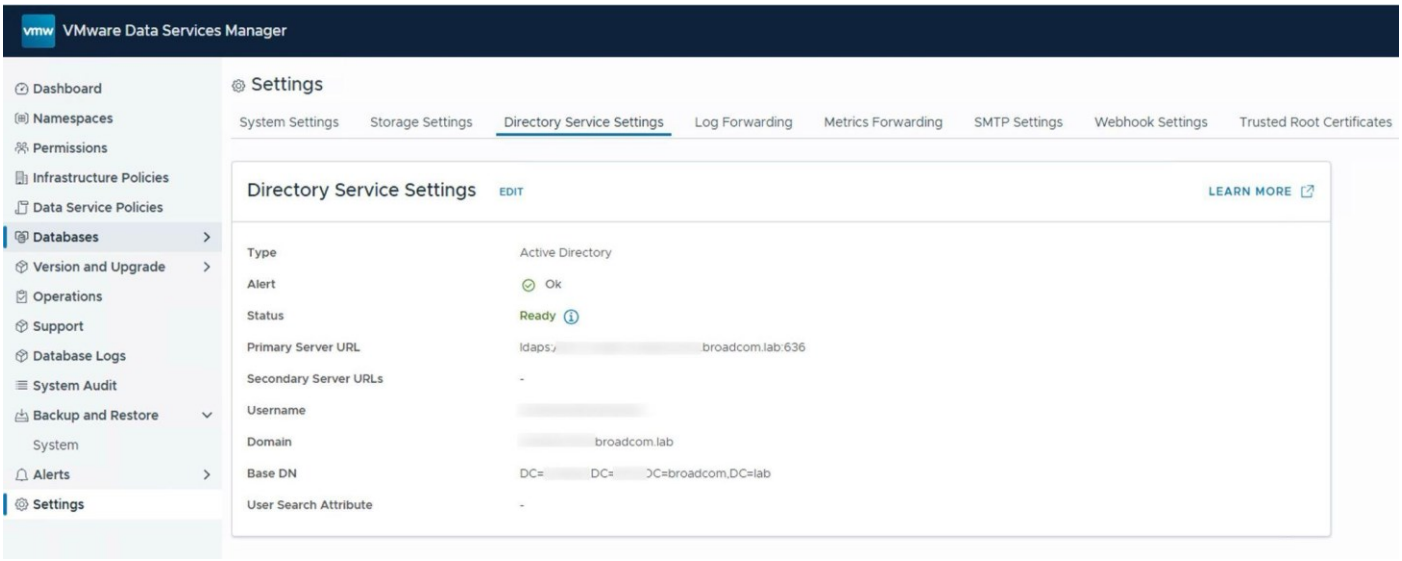
Item	Automated Mode	Manual Mode
AD Account	Pre-created privileged AD Account for DSM (SPN write, DNS update, gMSA access) and pre-created unprivileged AD account for SQL Server	Pre-created unprivileged AD account for SQL Server
SPN Management	Handled automatically by DSM	Pre-created manually by AD admin
DNS Management	Dynamic update	Static DNS records pre-configured by AD admin
Operational Overhead	Low (Zero-touch AD setup per DB)	High (Requires AD admin intervention per DB)
Security Alignment	Streamlined enterprise automation	Complies with strict least-privilege and zero-trust AD policies

To ensure seamless operation, the environment requires proper DNS resolution, secure LDAPS certificate trust, and open network access to essential Active Directory domain services. Follow Data Services Manager documentation for more details:

- Create Active Directory Domain for SQL Server
- Create an Active Directory Privileged Account
- Authorize Windows Authentication for SQL Server Users

- Grant DNS Update Permissions to the Privileged Account

Figure 71. Directory Service Settings in Data Services Manager



3.12.7 Deploy SQL Server with Always On Availability Groups in Data Services Manager

Traditionally, SQL Server AlwaysOn Availability Group relies either on Windows Server Failover Clustering (WSFC) on Windows or Pacemaker/Corosync on Linux. Data Services Manager replaces these complex OS-level clustering frameworks with a unified, cloud-native Cluster Manager architecture that handles leader election, health monitoring, failover orchestration, virtual IP of the primary replica and state reconciliation directly.

Considering the complexity of Pacemaker and other clustering technologies, DSM SQL Server abstracts this entirely away using its own cluster manager. It also automates the end-to-end lifecycle with minimal user interference from initial provisioning to various day-2 operations. By standardizing on a lightweight containerized deployment on Kubernetes, organizations can benefit from cloud-native, API-driven DBaaS capabilities on VMware Cloud Foundation.

Table 6. Cluster Management Technology for SQL Server AlwaysOn Availability Group

Item	Windows AG (WSFC)	Linux AG (Pacemaker)	DSM SQL Server Cluster
Operating System	Windows Server	Linux (RHEL, SLES, Ubuntu)	Optimized Linux (Photon OS / Ubuntu)
Clustering Technology	Windows Server Failover Cluster	Pacemaker & Corosync + resource agents	DSM built-in Cluster Manager
Deployment	Moderate manual setup	Complex manual setup	Fully Automated
Fencing / Quorum	WSFC Quorum	Corosync, fencing agents, STONITH	Integrated consensus-based quorum
Active Directory & SPN Management	Integrated	Manual	Automated by DSM
Day-2 Operations	Manual patching	Manual patching	Non-disruptive rolling upgrade and auto-healing

Item	Windows AG (WSFC)	Linux AG (Pacemaker)	DSM SQL Server Cluster
Infrastructure Integration	Bare metal / Generic VM	Bare metal / Generic VM	Native vSphere integration (vSphere HA, DRS, SPBM)

3.12.8 Post Deployment Capabilities and Day-2 Operations

Beyond initial provisioning, VMware Data Services Manager provides robust post deployment and day-2 lifecycle management for Microsoft SQL Server, enabling agile resource scaling and streamlined operations. Administrators can easily scale compute capacity up by adjusting the VM class with desired vCPU and memory allocations to adapt to shifting workload demands, as well as dynamically expand underlying storage volumes without service disruption. DSM further simplifies maintenance by automating scheduled and on-demand backups, point-in-time recovery (PITR), non-disruptive rolling updates and patches across cluster replicas, and database parameter tuning directly from the UI or API. By automating resource resizing, health monitoring, and high-availability failover management, DSM eliminates routine DBA operational overhead and ensures enterprise SQL Server environments remain resilient, optimized, and continuously available.

3.12.9 SQL Server Backup configuration

In VMware Data Services Manager, backup management for Microsoft SQL Server is designed to provide enterprise-grade data protection, compliance, and disaster recovery with minimal DBA operational overhead. DSM offloads backups to external S3-compatible object storage and supports both automated scheduled backups and on-demand backups.

Automated backups are configured as part of the data services policy creation wizard for SQL Server or adjusted post-deployment. DSM implements SQL Server's standard multi-tiered backup model to balance storage efficiency and recovery speed:

1. Full Backups: A complete copy of the database and its data structures.
2. Differential Backups: Captures only the extents that have changed since the last full backup, reducing backup size and completion time during the week.
3. Transaction Log Backups: Continuously archives the active transaction log to enable Point-in-Time Recovery (PITR).

The default backup schedule is configured to run a full backup weekly and differential backup daily. The transaction log backup frequency can be also tuned with a maximum interval of 30 minutes. Administrators can define a retention period of minimum 1 day and DSM will automate the backup purge operation for expired backup chains and transaction logs once they fall outside the retention window.

On-demand backup is also supported in DSM to provide immediate, user-triggered full backup taken outside of the regular backup schedule. On-demand backup is ideal before performing risky operations such as application releases, major schema migrations, batch data transformations, or database upgrades. It helps create an immediate point of recovery that does not rely on waiting for the next nightly or weekly scheduled backup window. Once the on-demand backup is created, it can be visible alongside scheduled backups in the DSM UI where administrators can monitor completion progress, backup size, and status.

Starting with Data Services Manager 9.1.1, it supports registering an externally created SQL Server full backup that is stored in S3-compatible storage as a backup resource in Data Services Manager to make it available for onboarding to Data Services Manager. This helps the data scientists and DBAs to migrate the existing SQL Server databases to DSM and plan Brownfield implementation of the current datasets. For more details, refer to Onboard an Existing SQL Server Database Backup into Data Services Manager.

3.12.10 Restore database for SQL Server

VMware Data Services Manager provides robust recovery options for Microsoft SQL Server by enabling non-destructive database restorations directly from S3-compatible backup storage. Rather than overwriting production instances, DSM provisions fresh database environments from either continuous automated backup timelines or on-demand backups, ensuring critical data can be quickly recovered, audited, or repurposed with minimal operational impact. You can restore a SQL Server database in DSM using the following method:

- Point-in-Time Recovery (PITR): Replays automated full, differential, and continuous transaction log backups to restore a database to any specific time within the retention window, drastically reducing data loss after accidental drops or data corruption while deploying safely into a newly provisioned VM that inherits the source configuration.
- On-Demand Recovery: Provisions a new database directly from a designated on-demand backup set to its exact captured state - bypassing transaction log replay chains and supporting restoration to the same or an alternative SQL Server cluster within the namespace for pre-change rollbacks, migrations, or dev/test validation.

3.12.12 Monitoring SQL Server

VMware Data Services Manager provides comprehensive, full-stack observability for Microsoft SQL Server by combining native in-console monitoring with enterprise metrics forwarding. Through the DSM UI, administrators gain real-time visibility into cluster health, resource consumption (CPU, memory, and storage utilization), I/O throughput, active connections, and operational alerts. For broader enterprise observability, DSM supports automated metrics forwarding—streaming database performance indicators directly to external platforms such as Prometheus or VCF Operations. This dual-layer approach allows infrastructure teams and DBAs to correlate database KPIs with the underlying vSphere virtualization layer, proactively detect bottlenecks, track availability, and ensure performance SLAs without requiring invasive third-party agents. For more details, refer to Monitoring Your Data Services Manager Environment.

4. SQL Server and In-Guest Best Practices

In addition to the previously mentioned VCF 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 VCF virtualized environment.

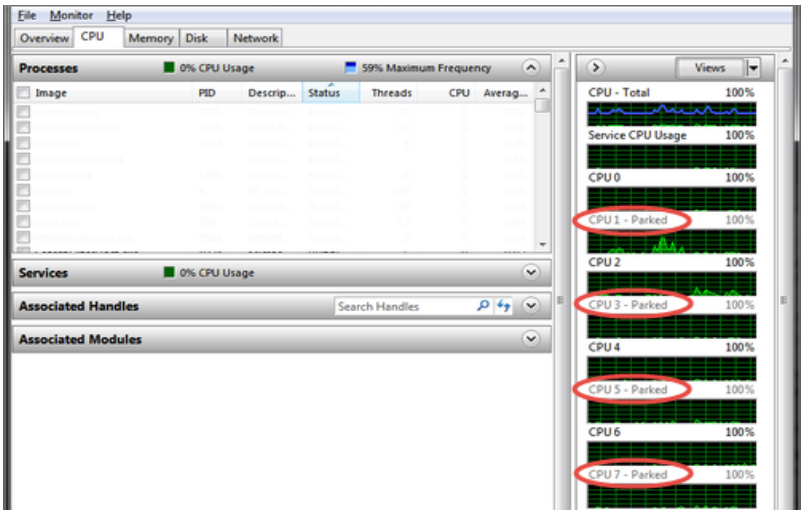
4.1 Windows Server Configuration[72]

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

4.1.1 Power Policy[73]

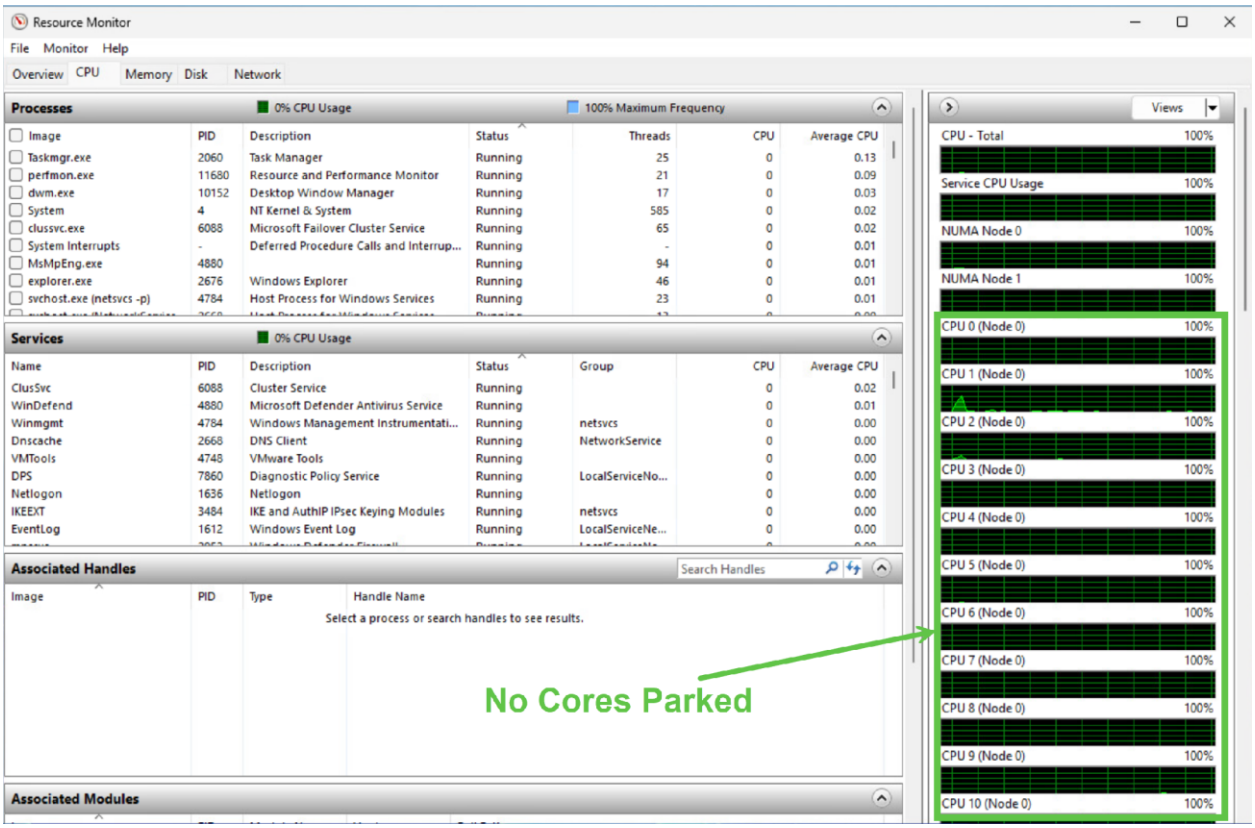
The default power policy option in Windows Server is “Balanced”. This configuration allows Windows Server OS to save power consumption by periodically throttling power to the CPU and turning off devices such as the network cards in the guest when Windows Server determines that they are idle or unused.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

Figure 72. Windows Server CPU Core Parking When Power Scheme set to “Balanced”



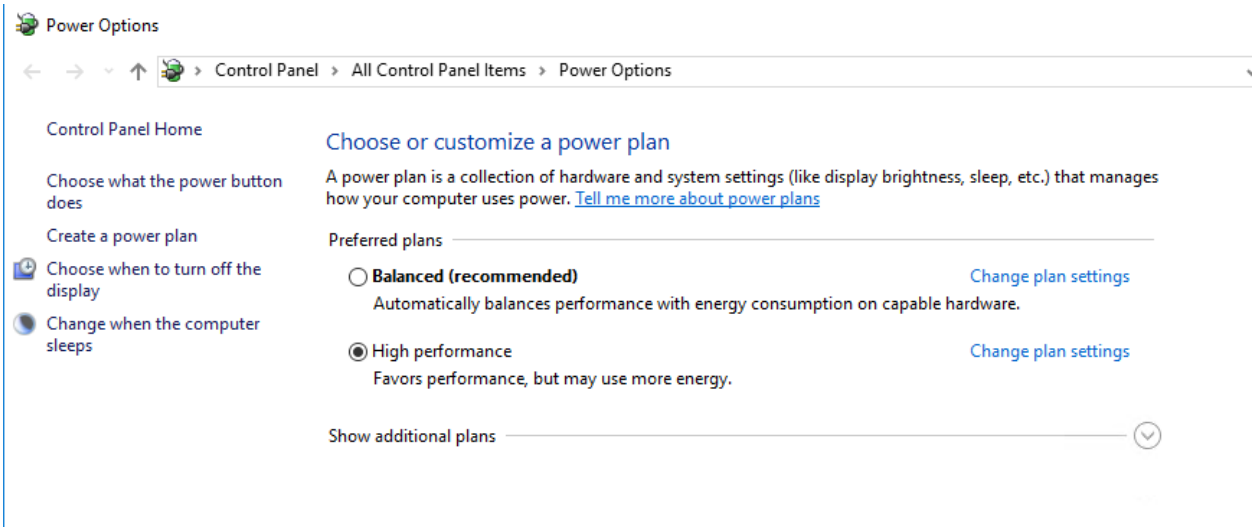
[72] Consult Operating System Best Practice Configurations for SQL Server for more information
[73] Slow performance on Windows Server when using the Balanced power plan

Figure 73 - No Cores Parked When Power Set Scheme to "High Performance"



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 74. Recommended Windows OS Power Plan

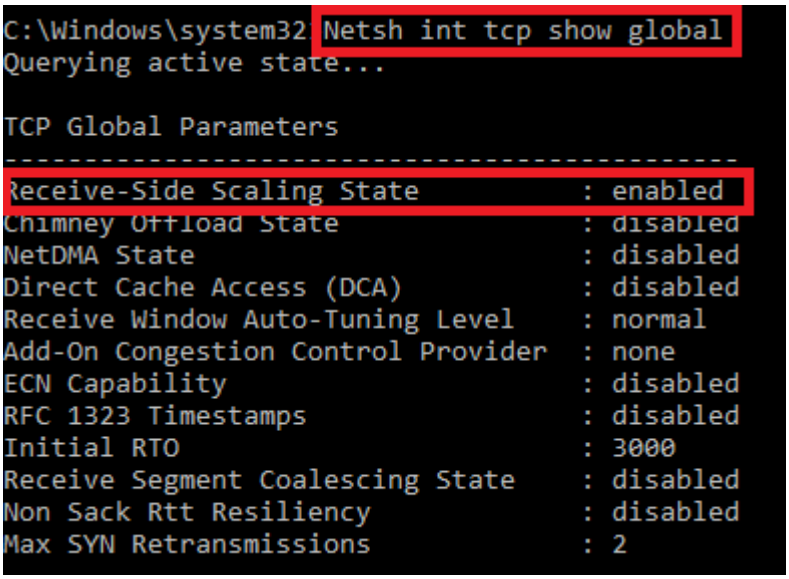


4.1.2 Enable Receive Side Scaling (RSS)[74]

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.

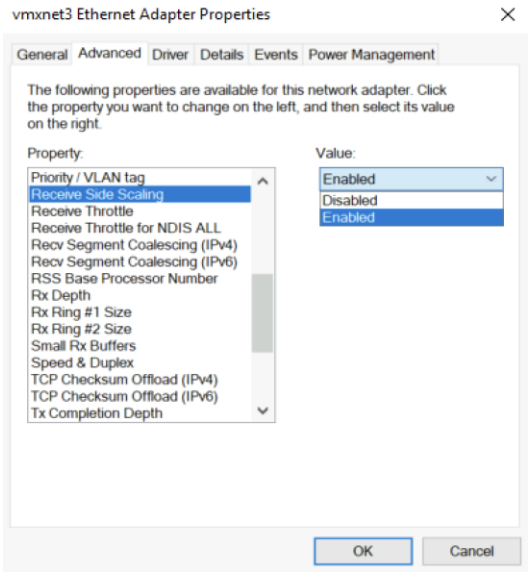
Figure 75. Enable RSS in Windows OS



Enable RSS on the VMXNET network adapter driver.[75] In Windows in **Network adapters**, right-click the VMXNET network adapter and click **Properties**. On the **Advanced** tab, enable the setting **Receive-side scaling**.

[74] Receive Side Scaling (RSS)
[75] 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 were upgraded from versions older 10.2.5, steps listed in this document are required in order to ensure or confirm that RSS is enabled on the network card.

Figure 76. Enable RSS in Windows Network Card Properties[76]



4.1.3 Configure PVSCSI Controller

The VMware Paravirtualized SCSI Controller (PVSCSI) 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"[77].

Note While increasing the default queue depth of a virtual SCSI controller can be beneficial to an SQL Server-based VM, the configuration can also introduce unintended adverse effects in overall performance if not done properly[78]. VMware highly recommends that customers consult and work with the appropriate storage vendor's support personnel to evaluate the impact of such changes and obtain recommendations or other adjustments that may be required to support the increase in queue depth of a virtual SCSI controller.

4.1.4 Using Antivirus Software[79]

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.

4.1.5 Other Applications

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

---

[76] Enabling Receive Side Scaling

[77] Large-scale workloads with intensive I/O patterns might require queue depths significantly greater than Paravirtual SCSI default values

[78] Guidelines for Troubleshooting Your Implementation of vSphere

[79] Configure antivirus software to work with SQL Server

4.2 SQL Server Configuration

4.2.1 Maximum Server Memory and Minimum Server Memory

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

SQL Server can consume all 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 VCF, 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
```

4.2.2 Lock Pages in Memory

Granting the Lock Pages in Memory user right to the SQL Server service account prevents SQL Server buffer pool pages from paging out by Windows Server. This setting is useful and has a positive performance impact because it prevents Windows Server from paging a significant amount of buffer pool memory out, 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[80] and setting VM memory reservations to improve the performance and stability of your SQL Server running on VCF.

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.

[80] Enable the Lock pages in memory option (Windows)

4.2.3 Large Pages[81]

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[82]...

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.
```

4.2.4 CXPACKET, MAXDOP, and CTFP

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

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.

[81] SQL Server and Large Pages Explained
[82] Support for Large Page Sizes in ESXi

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 following Microsoft article Server configuration: max degree of parallelism.

5. VMware Enhancements for Deployment and Operations

VCF 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 VCF.

5.1 Network Virtualization with VMware NSX for VCF

Although virtualization has allowed organizations to optimize their compute and storage investments, the network has remained mostly physical. VMware NSX® solves the datacenter 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.

6. Resources

SQL Server on VMware VCF:

- Architecting Business Critical Applications on VMware Hybrid Cloud
- 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 Cloud Foundation 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

- Using SQL Server Failover Clustering with vSAN iSCSI Target Service: Guidelines for supported configurations
- Overhead Memory on Virtual Machines
- Overview of VMware Tools
- VMware vCenter Server and Host Management
- Virtual Machine Encryption in VMware vSphere
- Comparing Available SCSI Device Access Modes

- Security in the vSphere Environment

SQL Server Resources

- Deprecated Database Engine features in SQL Server 2025 (17.x)
- 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)

7. Acknowledgments

Authors:

- Deji Akomolafe – Staff Solutions Architect, Microsoft Applications Practice Lead

vmware by Broadcom

Copyright © 2025 Broadcom. All rights reserved.
The term “Broadcom” refers to Broadcom Inc. and/or its subsidiaries. For more information, go to www.broadcom.com. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies. Broadcom reserves the right to make changes without further notice to any products or data herein to improve reliability, function, or design. Information furnished by Broadcom is believed to be accurate and reliable. However, Broadcom does not assume any liability arising out of the application or use of this information, nor the application or use of any product or circuit described herein, neither does it convey any license under its patent rights nor the rights of others.
Item No: vmw-bc-wp-tech-a4-word-2025 Feb-26