Architecting Microsoft SQL Server on VMware vSphere®

BEST PRACTICES GUIDE

MARCH 2017

vmware®

Table of Contents

List of Figures

1. Introduction

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

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

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

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

Virtualizing Microsoft SQL Server with vSphere enables many additional benefits. For example, VMware vSphere vMotion®, which enables seamless migration of virtual machines containing Microsoft SQL Server instances between physical servers and between data centers without interrupting users or their applications. VMware vSphere Distributed Resource Scheduler™ (DRS) can be used to dynamically balance Microsoft SQL Server workloads between physical servers. VMware vSphere High Availability (HA) and VMware vSphere Fault Tolerance (FT) provide simple and reliable protection for SQL Server virtual machines and can be used in conjunction with SQL Server’s own HA capabilities. Among other features, VMware NSX® provides network virtualization and dynamic security policy enforcement. VMware Site Recovery Manager™ provides disaster recovery plan orchestration. There are many more benefits that VMware can provide for the benefit of virtualized applications.

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

1.1 Purpose

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

1.2 Target Audience

This document assumes a basic knowledge and understanding of VMware vSphere and Microsoft SQL Server.

Architectural staff can use this document to gain an understanding of how the system will work as a whole as they design and implement various components.

Engineers and administrators can use this document as a catalog of technical capabilities.

DBA staff can use this document to gain an understanding of how SQL server might fit into a virtual infrastructure.

Management staff and process owners can use this document to help model business processes to take advantage of the savings and operational efficiencies achieved with virtualization.

2. Application Requirements Considerations

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

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

1. Understand the characteristics of the database workload associated with the application accessing SQL Server.
2. Understand availability and recovery requirements, including uptime guarantees (“number of nines”) and disaster recovery.
3. Capture resource utilization baselines for existing physical databases.
4. Plan the migration/deployment to vSphere.

2.1 Understand the Application Workload

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

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

- OLTP databases (online transaction processing) are often the most critical databases in an organization. These databases usually back customer-facing applications and are considered essential to the company’s core operations. Mission-critical OLTP databases and the applications they support often have SLAs that require very high levels of performance and are very sensitive for performance degradation and availability. SQL Server VMs running OLTP mission critical databases might require more careful resource allocation (CPU, memory, disk, and network) to achieve optimal performance. They might also be candidates for clustering with Windows Server Failover Cluster, which run either an AlwaysOn Failover Cluster Instance (FCI) or AlwaysOn Availability Group (AG). These types of databases are usually characterized with mostly intensive random writes to disk and sustained CPU utilization during working hours.
- DSS (decision support systems) databases, can be also referred to as data warehouses. These are mission critical in many organizations that rely on analytics for their business. These databases are very sensitive to CPU utilization and read operations from disk when queries are being run. In many organizations, DSS databases are the most critical during month/quarter/year end.

- Batch, reporting services, and ETL databases are busy only during specific periods for such tasks as reporting, batch jobs, and application integration or ETL workloads. These databases and applications might be essential to your company's operations, but they have much less stringent requirements for performance and availability. They may, nonetheless, have other very stringent business requirements, such as data validation and audit trails.
- Other smaller, lightly used databases typically support departmental applications that may not adversely affect your company's real-time operations if there is an outage. Many times, you can tolerate such databases and applications being down for extended periods.

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

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

2.2 Availability and Recovery Options

Running Microsoft SQL Server on vSphere offers many options for database availability, backup and disaster recovery utilizing the best features from both VMware and Microsoft. This section covers the different options that exist for availability and recovery.

2.2.1 VMware Business Continuity Options

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

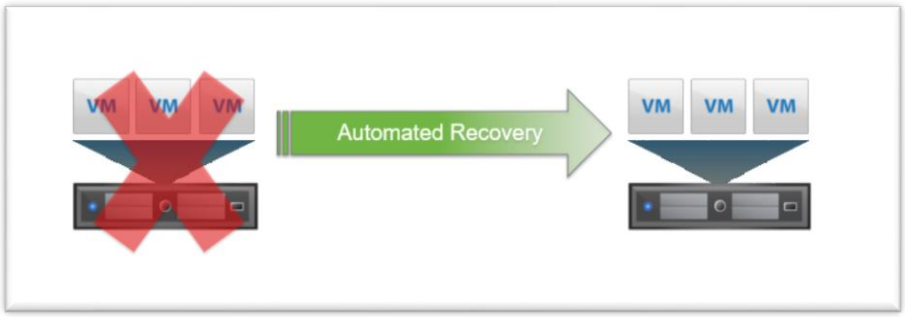
2.2.1.1. VMware vSphere High Availability

vSphere HA provides easy to use, cost-effective high availability for applications running in virtual machines. vSphere HA leverages multiple ESX hosts configured as a cluster to provide rapid recovery from outages and cost-effective high availability for applications running in virtual machines. vSphere HA protects application availability in the following ways:

- Protects against a physical server failure by restarting the virtual machines on other hosts within the cluster in case of a failure.
- Protects against operating system failure by continuous monitoring using a heartbeat and by restarting the virtual machine in case of a failure.
- Detects the hardware conditions of a host proactively and enables evacuation of the virtual machines before an issue causes an outage.
- Protects against datastore accessibility failures by restarting affected virtual machines on other hosts that still have access to their datastores.
- Protects virtual machines against network isolation by restarting them if their host becomes isolated on the management network. This protection is provided even if the network has become partitioned.

- Provides APIs for protecting against application failure allowing third-party tools to continuously monitor an application and reset the virtual machine if a failure is detected.

Figure 1. vSphere HA

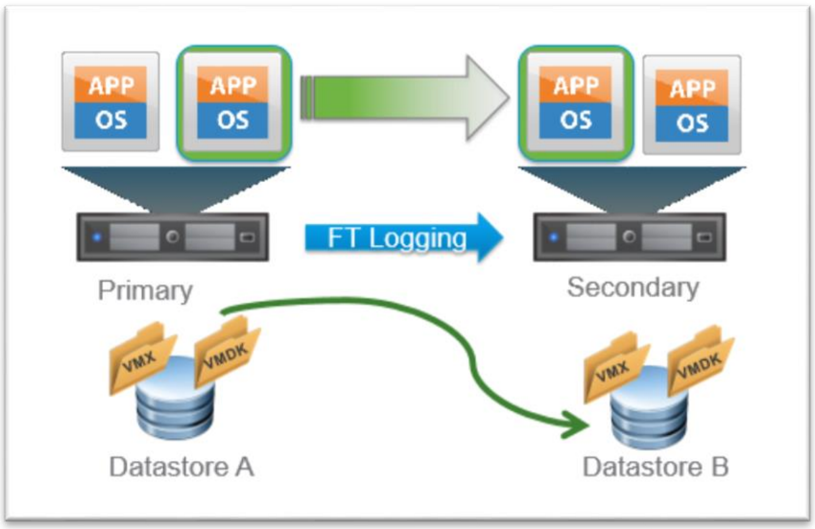


2.2.1.2. vSphere Fault Tolerance

vSphere Fault Tolerance 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 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. The functioning ESXi host seamlessly becomes the primary VM host without losing network connections or in-progress transactions. With transparent failover, there is no data loss, and network connections are maintained. After a transparent failover occurs, a new secondary VM is respawned and redundancy is re-established. The entire process is transparent, fully automated, and occurs even if VMware vCenter Server™ is unavailable.

There are licensing requirements and interoperability limitations to consider when using fault tolerance as detailed in *the VMware vSphere Availability* guide at https://pubs.vmware.com/vsphere-60/topic/com.vmware.ICbase/PDF/vsphere-esxi-vcenter-server-60-availability-guide.pdf:

Figure 2. vSphere FT



2.2.1.3. vSphere vMotion and vSphere Storage vMotion

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

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

vSphere 6 introduced three new vSphere vMotion capabilities:

- Cross vCenter vSphere vMotion – Allows for live migration of virtual machines between vCenter instances without any service interruption.
- Long-distance vSphere vMotion – Allows high network round-trip latency times between the source and destination physical servers of up to 150 millisecond RTT. This technology allows for disaster avoidance of much greater distances and further removes the constraints of the physical world from virtual machines.
- vMotion FCI node – Starting with vSphere 6, it is possible to use vSphere vMotion to migrate virtual machines that are part of Microsoft failover clustering using a shared physical RDM disk.

These new vSphere vMotion capabilities further remove the limitations of the physical world allowing customers to migrate SQL servers between physical servers, storage systems, networks, and even data centers with no service disruption.

2.2.1.4. Site Recovery Manager

Site Recovery Manager is a business continuity and disaster recovery solution that helps you to plan, test, and run the recovery of virtual machines between a protected site and a recovery site.

You can use Site Recovery Manager to implement different types of recovery from the protected site to the recovery site.

- Planned migration – The orderly evacuation of virtual machines from the protected site to the recovery site. Planned migration prevents data loss when migrating workloads in an orderly fashion. For planned migration to succeed, both sites must be running and fully functioning.
- Disaster recovery – Like planned migration except that disaster recovery does not require that both sites be up and running. For example, if the protected site goes offline unexpectedly. During a disaster recovery operation, failure of operations on the protected site are reported but otherwise ignored.

Site Recovery Manager orchestrates the recovery process with the replication mechanisms to minimize data loss and system down time.

A recovery plan specifies the order in which virtual machines start up on the recovery site. A recovery plan specifies network parameters, such as IP addresses, and can contain user-specified scripts that Site Recovery Manager can run to perform custom recovery actions on virtual machines.

Site Recovery Manager lets you test recovery plans. You conduct tests by using a temporary copy of the replicated data in a way that does not disrupt ongoing operations at either site.

For more details and best practices on vSphere business continuity capabilities, see the *VMware vSphere Availability* guide at https://pubs.vmware.com/vsphere-60/topic/com.vmware.ICbase/PDF/vsphere-esxi-vcenter-server-60-availability-guide.pdf.

For more details and best practices on Site Recovery Manager, see Section **Error! Reference source not found.**, **Error! Reference source not found.**, and the Site Recovery Manager documentation at https://pubs.vmware.com/srm-60/index.jsp.

2.2.2 Native SQL Server Capabilities

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

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

Table 1. ***SQL Server 2012 High Availability Options***

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

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

For a more detailed look at options requirements and how to plan mission critical deployments, see the following guides:

- http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/solutions/sql-server-on-vmware-availability-and-recovery-options.pdf
- *Planning a Highly Available, Mission Critical SQL Server Deployments with VMware vSphere*

3. Best Practices for Deploying SQL Server Using vSphere

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

3.1 Rightsizing

Rightsizing is a term that means when deploying a VM, it is sized properly instead of adding too many, which is a common sizing practice for physical servers. Rightsizing is imperative when sizing virtual machines. 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. This is because it is typically more difficult for the DBA to add CPUs to this physical server after it is deployed. It is a similar situation for memory and other aspects of a physical deployment – it is easier to build in capacity than try to adjust it later, which often requires additional cost and downtime. This can also be problematic if a server started off as undersized and cannot handle the workload it is 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 short 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. There are two ways to size the VM based on the requirements:

- When an SQL server is considered critical with high performance requirements, take the most sustained peak as the sizing baseline.
- With lower tier SQL Server implementations, where consolidation takes higher priority than performance, an average can be considered for the sizing baseline.

When in doubt, start with the lower amount of resources and grow as necessary.

After the VM has been created, adjustments can be made to its resource allocation from the original base line. Adjustments can be based on additional monitoring using a DMV-based tool, similar to monitoring a physical SQL Server deployments. VMware vRealize® Operations Manager™ can perform DMV-based monitoring with ongoing capacity management and will alert if there is resource waste or contention points.

Rightsizing and not over allocating resources is important for the following reasons:

- 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. While ESXi can typically reclaim the over-allocated memory, it cannot reclaim the overhead associated with this over-allocated memory, thus consuming memory that could otherwise be used to support more VMs.

- Be careful when measuring the amount of memory consumed by a SQL Server VM with the VMware Active Memory counter. Applications that contain their own memory management, such as SQL Server, can skew this counter. Consult with the database administrator to confirm memory consumption rates before adjusting the memory allocated to a SQL Server VM.
- Having more vCPUs assigned for the virtual SQL Server also has licensing implications in certain scenarios, such as per-core licenses.
- Adding resources to VMs (a click of a button) is much easier than adding resources to physical machines.

For more information about sizing for performance, see *Performance Best Practices for VMware vSphere 6.0* at http://www.vmware.com/files/pdf/techpaper/VMware-PerfBest-Practices-vSphere6-0.pdf.

3.2 Host Configuration

3.2.1 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.

3.2.2 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.
- It is a good practice to disable any devices that are not used as serial ports.
- Set Power Management (or its vendor-specific equivalent label) to “OS controlled”. This will enable the ESXi hypervisor to control power management based on the selected policy. See the following section for more information.

Disable all processor C-states (including the C1E halt state). These enhanced power management schemes can introduce memory latency and sub-optimal CPU state changes (Halt-to-Full), resulting in reduced performance for the VM.

3.2.3 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. See *Best Practices for Performance Tuning of Latency-Sensitive Workloads in vSphere VMs* (http://www.vmware.com/files/pdf/techpaper/VMW-Tuning-Latency-Sensitive-Workloads.pdf).

Server hardware and operating systems are usually engineered to minimize power consumption for economic reasons. Windows Server and the ESXi hypervisor both favor minimized power consumption over performance. While previous versions of ESXi default to “high performance” power schemes, vSphere 5.0 and later defaults to a “balanced” power scheme. For critical applications, such as SQL Server, the default power scheme in vSphere 6.0 is not recommended.

There are three distinct areas of power management in an ESXI hypervisor virtual environment: server hardware, hypervisor, and guest operating system. The following section provides power management and power setting recommendations covering all of these areas.

3.2.3.1. 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.

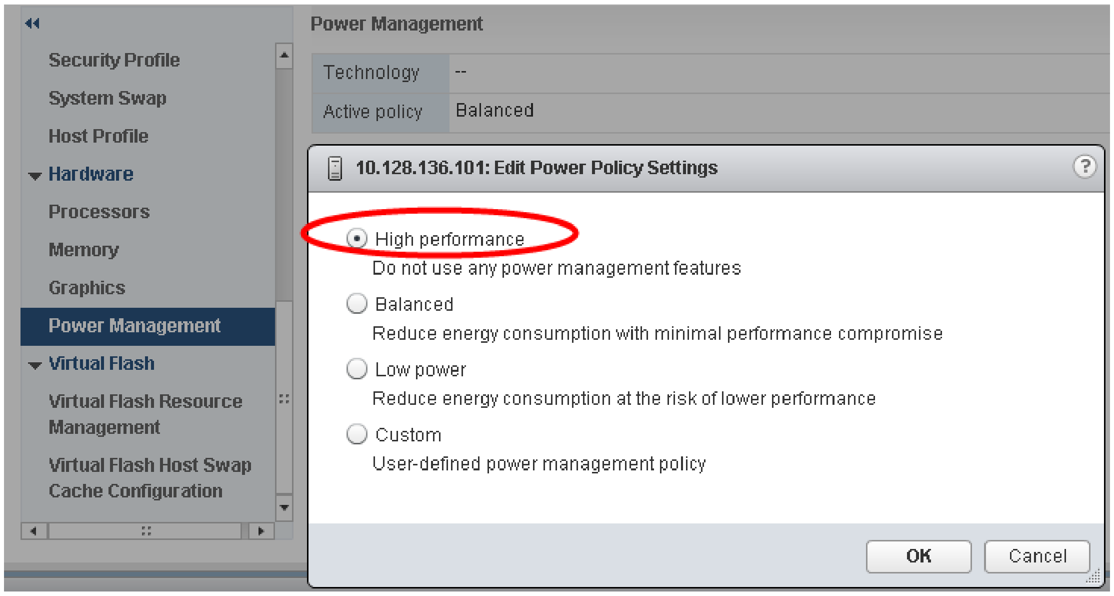
In general, selecting a high-performance policy provides more absolute performance, but at lower efficiency (performance per watt). Lower-power policies provide lower absolute performance, but at higher efficiency. ESXi provides five power management policies. If the host does not support power management, 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 SQL Server running on ESXi.*
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 VMware vSphere Web Client Advanced Settings dialog box.
Not supported	The host does not support any power management features, or power management is not enabled in the BIOS.

VMware recommends setting the high performance ESXi host power policy for critical 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 3. Recommended ESXi Host Power Management Setting



Power Management

Technology	--
Active policy	Balanced

10.128.136.101: Edit Power Policy Settings

[✓ High performance]
Do not use any power management features

Balanced
Reduce energy consumption with minimal performance compromise

Low power
Reduce energy consumption at the risk of lower performance

Custom
User-defined power management policy

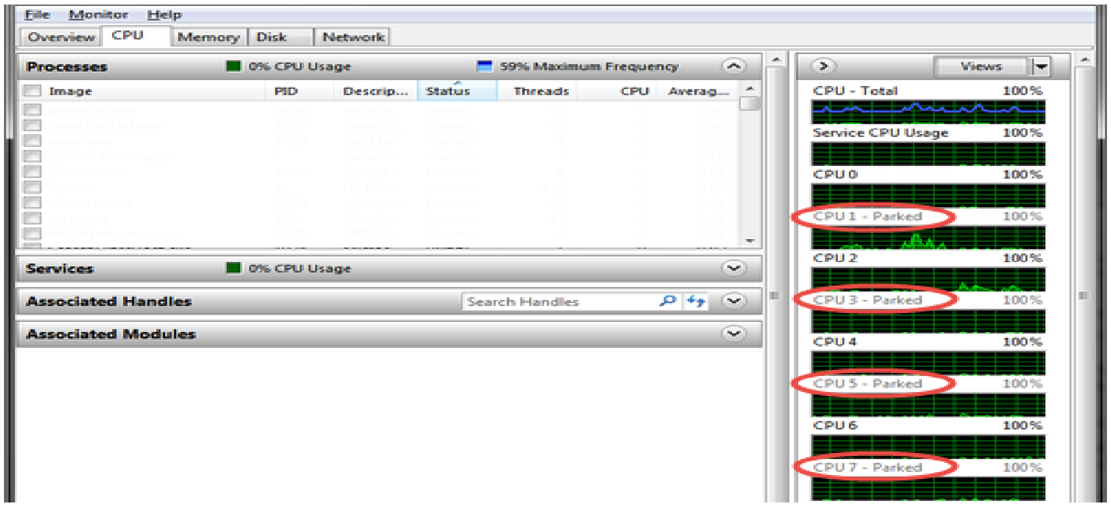
When a CPU runs at lower frequency, it can also run at lower voltage, which saves power. This type of power management is called dynamic voltage and frequency scaling (DVFS). ESXi attempts to adjust CPU frequencies so that VM performance is not affected.

When a CPU is idle, ESXi can take advantage of deep halt states (known as C-states). The deeper the C-state, the less power the CPU uses, but the longer it takes for the CPU to resume running. When a CPU becomes idle, ESXi applies an algorithm to predict how long it will be in an idle state and chooses an appropriate C state to enter. In power management policies that do not use deep C-states, ESXi uses only the shallowest halt state (C1) for idle CPUs.

3.2.3.2. Windows Guest Power Settings

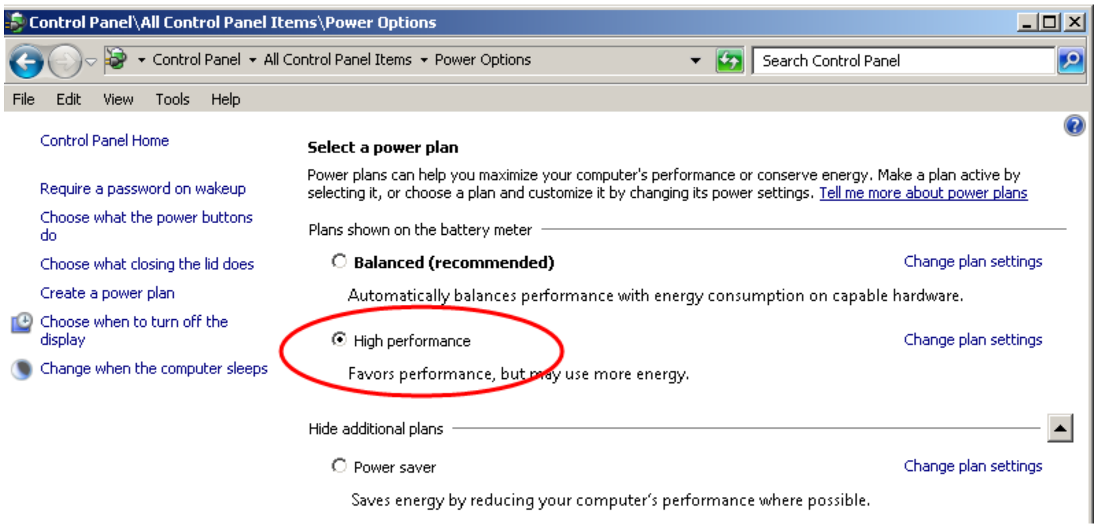
The default power policy option in Windows Server 2016 is “Balanced”. This configuration allows Windows Server OS to save power consumption by periodically throttling power to the CPU and turning off devices such as the network cards in the guest when Windows Server determines that they are idle or unused. This capability is inefficient for critical SQL Server workloads due to the latency and disruption introduced by the act of powering-off and powering-on CPUs and devices. Allowing Windows Server to throttle CPUs can result in what Microsoft describes as core parking and should be avoided. For more information, see *Server Hardware Power Considerations* at https://msdn.microsoft.com/en-us/library/dn567635.aspx.

Figure 4. Windows Server CPU Core Parking



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

Figure 5. Recommended Windows Guest Power Scheme


Control Panel\All Control Panel Items\Power Options
Control Panel ▸ All Control Panel Items ▸ Power Options
Search Control Panel
File Edit View Tools Help
Control Panel Home
Require a password on wakeup
Choose what the power buttons do
Choose what closing the lid does
Create a power plan
Choose when to turn off the display
Change when the computer sleeps

Select a power plan

Power plans can help you maximize your computer's performance or conserve energy. Make a plan active by selecting it, or choose a plan and customize it by changing its power settings. Tell me more about power plans

Plans shown on the battery meter

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

Automatically balances performance with energy consumption on capable hardware.

[✓ High performance] — Change plan settings

Favors performance, but may use more energy.

Hide additional plans

○ Power saver — Change plan settings

Saves energy by reducing your computer's performance where possible.

3.3 *CPU Configuration*

3.3.1 Physical, Virtual, and Logical CPUs and Cores

VMware uses the terms virtual CPU (vCPU) and physical CPU (pCPU), and virtual core (vCore) and physical core (pCore) to distinguish between the processors within the VM and the underlying physical x86/x64-based processor cores. VMs with more than one vCPU are called *symmetric multiprocessing* (SMP) virtual machines. Other terms that can be used are logical CPU, which refers to Hyper-threading, virtual cores, and physical cores.

3.3.2 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.

Lower-tier SQL Server workloads typically are less latency sensitive, so in general the goal is to maximize use of system resources and achieve higher consolidation ratios rather than maximize performance.

The vSphere CPU scheduler's policy is tuned to balance between maximum throughput and fairness between VMs. For lower-tier databases, a reasonable CPU overcommitment can increase overall system throughput, maximize license savings, and continue to maintain adequate performance.

3.3.3 Hyper-threading

Hyper-threading is an Intel technology that exposes two hardware contexts (threads) from a single physical core, also referred to as logical CPUs. This is not the same as having twice the number of CPUs or cores. By keeping the processor pipeline busier and allowing the hypervisor to have more CPU scheduling opportunities, Hyper-threading generally improves the overall host throughput anywhere from 10 to 30 percent.

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

Hyper-threading can be controlled on a per VM basis in the hyper-threading Sharing section on the **Properties** tab of a VM. This setting provides control of whether a VM should be scheduled to share a physical core if Hyper-threading is enabled on the host.

Any – This is the default setting. The vCPUs of this VM can freely share cores with other virtual CPUs of this or other virtual machines. VMware recommends leaving this setting to allow the CPU scheduler the maximum scheduling opportunities.

None – The vCPUs of this VM have exclusive use of a processor whenever they are scheduled to the core. Selecting **None** in effect disables Hyper-threading for your VM.

Internal – This option is similar to **None**. vCPUs from this VM cannot share cores with vCPUs from other VMs. They can share cores with the other vCPUs from the same VM.

See additional information about Hyper-threading on a vSphere host in *VMware vSphere Resource Management* (https://pubs.vmware.com/vsphere-60/topic/com.vmware.ICbase/PDF/vsphere-esxi-vcenter-server-60-resource-management-guide.pdf).

It is important to remember to account for the differences between a processor thread and a physical CPU/core during capacity planning for your SQL Server deployment.

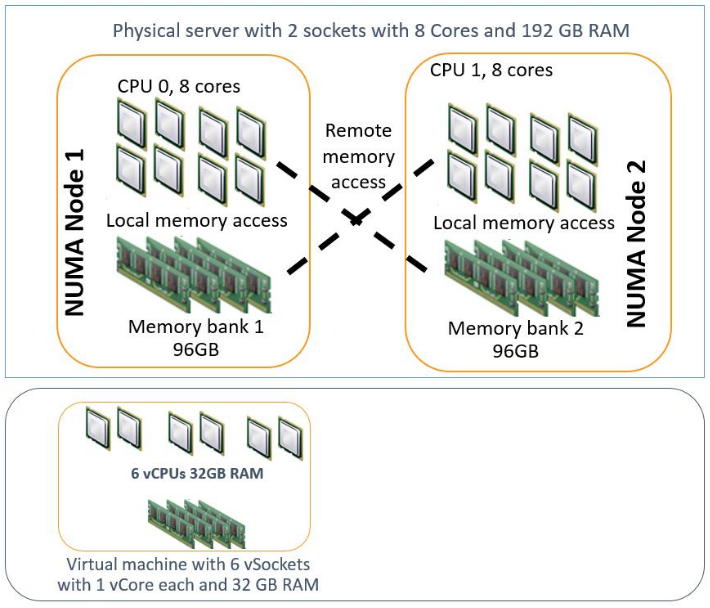
3.3.4 NUMA Consideration

vSphere supports non-uniform memory access (NUMA) systems. In a NUMA system, there are multiple NUMA nodes that consist of a set of processors and the memory. The access to memory in the same node is local. The access to the other node is remote. The remote access takes more cycles because it involves a multi-hop operation. Due to this asymmetric memory access latency, keeping the memory access local or maximizing the memory locality improves performance. On the other hand, CPU load balancing across NUMA nodes is also crucial to performance. The CPU scheduler achieves both aspects of performance.

The intelligent, adaptive NUMA scheduling and memory placement policies in vSphere can manage all VMs transparently, so administrators do not have to deal with the complexity of balancing VMs between nodes by hand. To reduce memory access latency, consider the following information.

For small VMs running SQL Server, allocate vCPUs equal to or less than the number of cores in each physical NUMA node. When you do this, the guest operating system or SQL Server does not need to consider NUMA because ESXi makes sure memory accesses are as local as possible. In the following example, there is a VM with 6 vCPUs and 32 GB of RAM residing in a server that has 8 cores and 96 GB of RAM in each NUMA node, with a total of 16 CPU cores and 192 GB RAM. The ESXi server places the VM entirely in one NUMA node, making sure all processes stay local and performance is optimized.

Figure 6. An Example of a VM with NUMA Locality

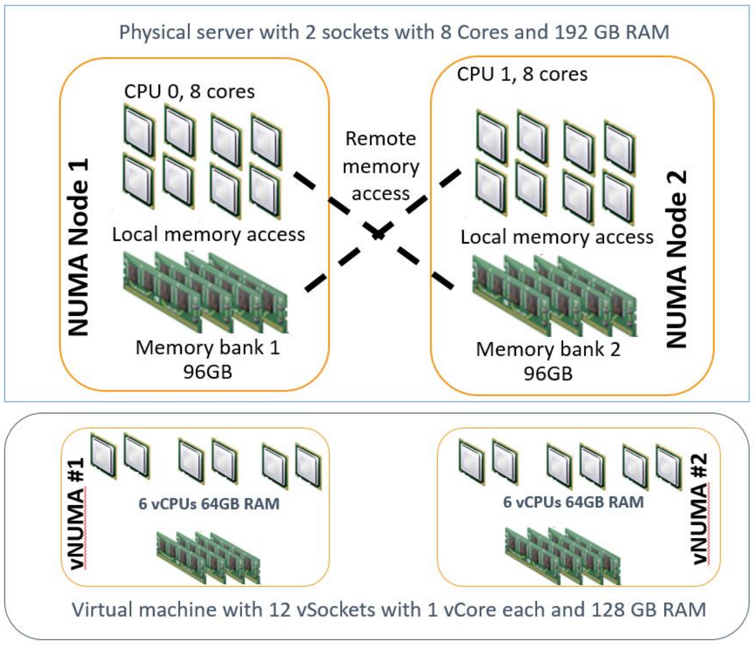


For wide SQL Server virtual machines, where the number of allocated vCPUs is greater than the number of cores in the NUMA node, ESXi divides the CPU and memory of the VM into two or more virtual NUMA nodes and places each vNUMA on a different physical NUMA node. The vNUMA topology is exposed to the guest OS and SQL Server to take advantage of memory locality.

In the following example, there is a single VM with 12 vCPUs and 128 GB of RAM residing on the same physical server that has 8 cores and 96 GB of RAM in each NUMA node, with a total of 16 CPU cores and 192 GB RAM. The VM will be created as a wide VM with a vNUMA architecture that is exposed to the underling guest server OS.

Note By default, vNUMA is enabled only for a VM with nine or more vCPUs.

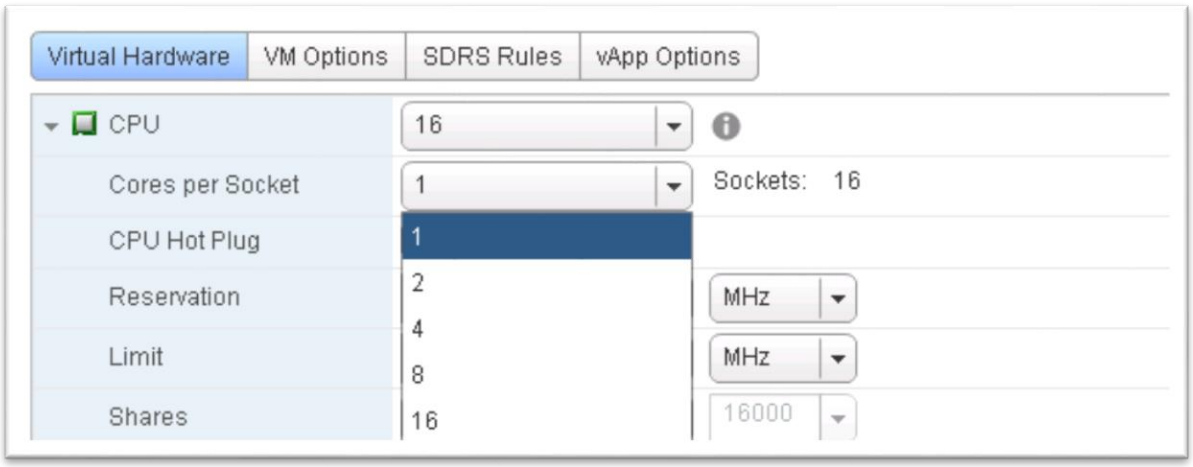
Figure 7. An Example of a VM with vNUMA



3.3.5 Cores per Socket

It is possible to assign multiple virtual cores (vCores) per virtual sockets (vSockets) to a VM. You can do this by setting the **Cores per sockets** setting to more than one in the VM configuration editor.

Figure 8. Cores per Sockets



CPU: 16
Cores per Socket: 1 Sockets: 16
CPU Hot Plug
Reservation
Limit
Shares: 16000

For example, a VM can have 4 vCPUs (sockets) each with 4 vCores, or it can have 2 vCPUs each with 8 vCores. Both options result in the VM having 16 vCores that are mapped to 16 pCores or logical Hyperthreads. This advanced setting was created to assist with licensing limitations for certain applications and operating systems that limit the number of cores and sockets. This is very useful for virtualized SQL Server deployments due to SQL Server's upper limits on the number of allowed CPU sockets and cores. The limits are different for the versions and editions of the software as detailed in the Microsoft article *Compute Capacity Limits by Edition of SQL Server* https://msdn.microsoft.com/en-us/library/ms143760(v=sql.130).aspx .

In the preceding example, if this VM is a SQL Server 2014 Standard Edition, it is limited to the lesser 4 sockets or 16 cores. Anything more than 4 sockets and 16 cores is not recognized by SQL Server. To maximize the compute allocation to the VM, either 4 vSockets and 4 vCores each (4x4), 2 vSockets and 8 vCores each (2x8), or 1 vSocket and 16 vCores (1x16) assigns access to 16 pCores to the VM.

Changing the number of vCores per vSockets has implications on the vNUMA topology and must be done with care. In the following examples, it is assumed that there is a physical server with 2 physical CPU sockets and 12 cores in each physical NUMA node:

- When the VM running SQL Server is assigned fewer CPU cores than the number cores in the physical NUMA node, any configuration of cores per socket is acceptable because even though vNUMA will be enabled with more than 9 vCPUs by default only one vNUMA node will be configured for the VM. In the preceding example of a 2x12 physical server, both a 2x4 or a 1x8 VM configurations will not affect vNUMA because both are lower than the physical cores per physical NUMA node (8<12). Generally, when the VM CPU count is lower than the physical NUMA node size, try to use the fewest number of vSockets possible.
- For configuration with more vCPUs than physical cores on the physical NUMA, such as 16 vCPU VM on a 2x12 host, always assign a number of vCPUs that can be divided evenly between physical NUMA nodes. Do not assign an odd number of vCPUs as that can result in a sub-optimal configuration. Also, the following considerations needs to be taken into account:
    - Prior to vSphere 6.5, the cores per socket configuration directly affected the vNUMA topology of the VM. Because of that, you must be aware of the underlying physical NUMA topology when configuring the "cores per socket." For example, assuming a requirement of 16 CPUs for a SQL Server 2014 Standard VM, it will not be able to take advantage of all the vCPUs assigned to it if it is configured with 1 core per socket (16 vSockets > limit of 4 sockets). In the example of a physical server with 2x12, a VM with either 4x4 or 2x8 configuration is acceptable because it allows ESXi to place each of the vNUMA nodes within a physical NUMA node.
    - Starting with vSphere 6.5 the number of cores per socket does not affect the vNUMA configuration by default. That means that any cores per sockets configuration can be set, and ESXi always tries to create the optimized vNUMA configuration in the backend.

A few things to note:

- A VM that was upgraded from earlier versions of vSphere than 6.5 has the following advanced setting:

  ```
  numa.vcpu.followcorespersocket  = 1 set
  ```

  This setting forces the old vNUMA behavior, which respects the cores per socket for vNUMA topology. This is done for backward compatibility. To make the VM to correspond to the new behavior, change this setting to 0.
- Memory size is not considered for vNUMA topology, only the number of CPUs, even if the amount of RAM assigned to the VM is more than the physical memory in each physical NUMA node. A VM with more memory than the physical NUMA node, but with fewer CPU cores than the physical NUMA node, forces memory to be fetched remotely, degrading performance. If there is a need for more memory than the physical NUMA, then it is best to force the hypervisor to create a vNUMA topology by either assigning more cores than the physical NUMA configuration (and more than 9 by default), or

by lowering the advanced setting `numa.vcpu.min` from 9 to the number of CPUs assigned to the virtual machine.

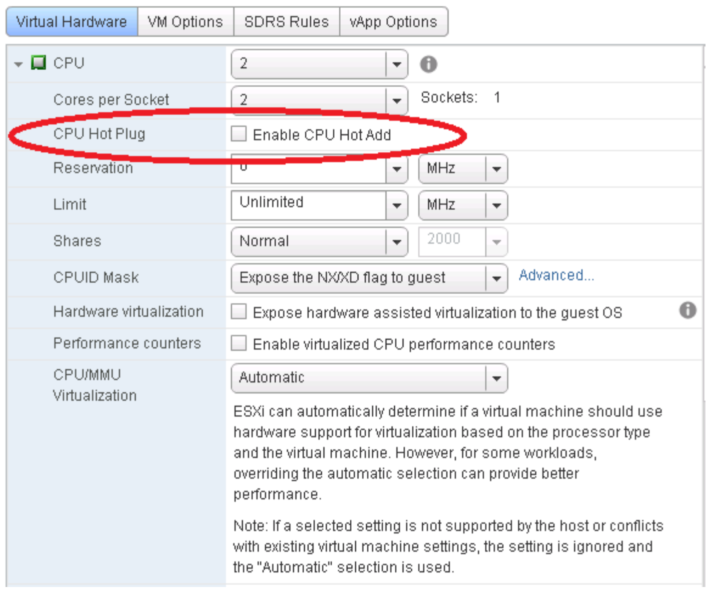
3.3.6 CPU Hot Plug

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

This means that SQL Server inside the VM cannot utilize vNUMA and will have degraded performance because the NUMA architecture exceeds that of the underlying physical server.

There is a use case for using hot plug especially for dynamic resource management and the ability to dynamically add CPUs when vNUMA is not required (which is usually because of smaller VMs). Therefore, VMware recommends to *not* enable CPU hot plug by default, especially for VMs that require vNUMA. Rightsizing the VM's CPU is always a better choice than relying on CPU hot plug. The decision whether to use this feature should be made on a case-by-case basis and not implemented in the VM template used to deploy SQL. See the Knowledge Base article, *vNUMA is disabled if VCPU hotplug is enabled (2040375)* at http://kb.vmware.com/kb/2040375.

Figure 9. Enabling CPU Hot Plug



CPU	2
Cores per Socket	2 Sockets: 1
CPU Hot Plug	☐ Enable CPU Hot Add
Reservation	0 MHz
Limit	Unlimited MHz
Shares	Normal 2000
CPUID Mask	Expose the NX/XD flag to guest Advanced...
Hardware virtualization	☐ Expose hardware assisted virtualization to the guest OS
Performance counters	☐ Enable virtualized CPU performance counters
CPU/MMU Virtualization	Automatic

ESXi can automatically determine if a virtual machine should use hardware support for virtualization based on the processor type and the virtual machine. However, for some workloads, overriding the automatic selection can provide better performance.

Note: If a selected setting is not supported by the host or conflicts with existing virtual machine settings, the setting is ignored and the "Automatic" selection is used.

3.3.7 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.

3.3.8 Virtual Machine Encryption

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 and 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.

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

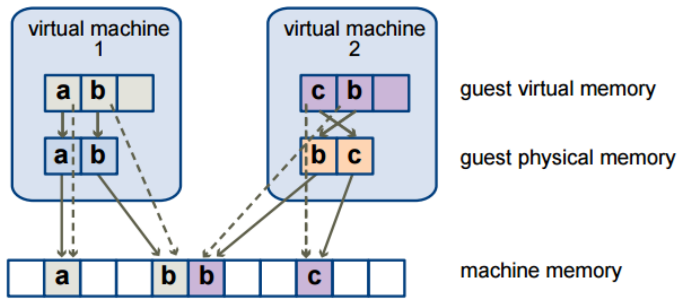
3.4 Memory Configuration

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.

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.

The guest OS, which is Windows Server in the case of SQL Server through SQL Server 2016, 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 10. Memory Mappings Between Virtual, Guest, and Physical Memory



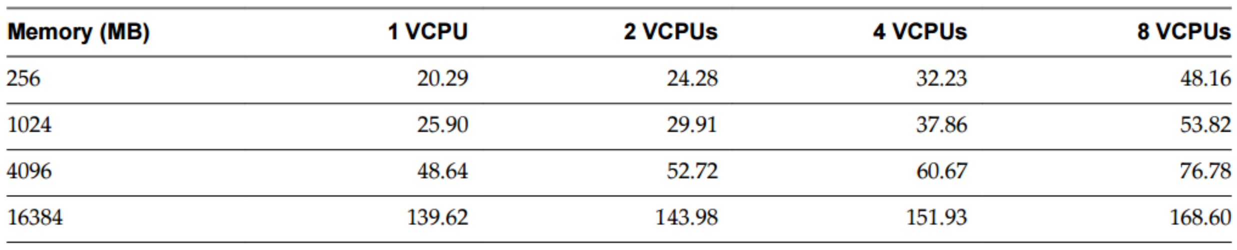
3.4.1 Memory Sizing Considerations

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.
- Similar to CPU NUMA consideration, with Microsoft SQL Server, NUMA is less of a concern because both vSphere and SQL Server support NUMA. As with SQL Server, vSphere has intelligent, adaptive NUMA scheduling and memory placement policies that can manage all VMs transparently. If the VMs memory is sized less than the amount available per NUMA node, ESXi will avoid remote memory access as much as possible. However, if the VM is sized larger than the NUMA node memory size, NUMA can be exposed to the underlying Windows Server guest OS with vNUMA allowing SQL Server to take advantage of NUMA.
- VMs require a certain amount of available overhead memory to power on. You should be aware of the amount of this overhead. The following table lists the amount of overhead memory a VM requires to power on. After a VM is running, the amount of overhead memory it uses might differ from the amount listed in the following snapshot.

This snapshot provides a sample of overhead memory values and does not apply to all possible configurations. You can configure a VM to have up to 128 vCPUs and up to 4 TB of memory.

Figure 11. Sample Overhead Memory on Virtual Machines



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

For additional details, refer to *vSphere Resource Management* (https://pubs.vmware.com/vsphere-60/topic/com.vmware.ICbase/PDF/vsphere-esxi-vcenter-server-60-resource-management-guide.pdf).

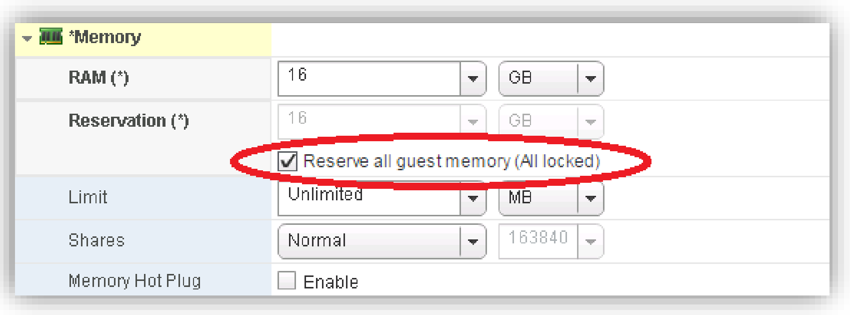
3.4.2 Memory Reservation

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

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

ThreadStack = SQL Max Worker Threads * ThreadStackSize

ThreadStackSize      = 1MB on x86

                     = 2MB on x64

                     = 4MB on IA64

OS Mem: 1GB for every 4 CPU Cores
```

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

Figure 12. Setting Memory Reservation



*Memory		
RAM (*)	16	GB
Reservation (*)	16	GB
	☑ Reserve all guest memory (All locked)	
Limit	Unlimited	MB
Shares	Normal	163840
Memory Hot Plug	☐ Enable	

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.

3.4.3 The Balloon Driver

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

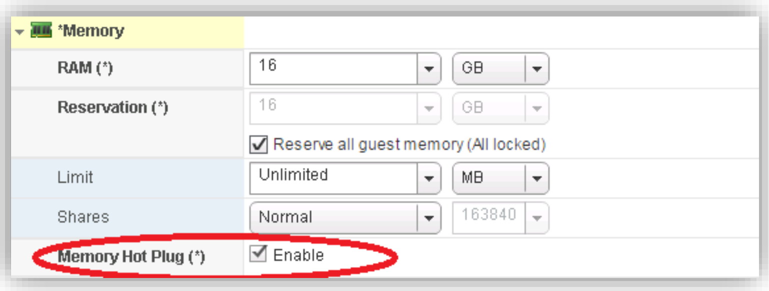
When designing SQL Server for performance, the goal is to eliminate any chance of paging from happening. Disable the ability for the hypervisor to reclaim memory from the guest OS by setting the memory reservation of the VM to the size of the provisioned memory. The recommendation is to leave the balloon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

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

3.4.4 Memory Hot Plug

Similar to CPU hot plug, memory hot plug enables a VM administrator to add memory to the VM with no down time. Before vSphere 6, when memory hot add was configured on a VM with vNUMA enabled, it would always add it to node0, creating NUMA imbalance. With vSphere 6 and later, when enabling memory hot plug and adding memory to the VM, the memory will be added evenly to both vNUMA nodes which makes this feature usable for more use cases. VMware recommends using memory hot plug in cases where memory consumption patterns cannot be easily and accurately predicted only with vSphere 6 and later. After memory has been added to the VM, increase the max memory setting on the database if one has been set. This can be done without requiring a server reboot or a restart of the SQL Server service. As with CPU hot plug, it is preferable to rely on rightsizing than on memory hot plug. The decision whether to use this feature should be made on a case-by-case basis and not implemented in the VM template used to deploy SQL Server.

Figure 13. Setting Memory Hot Plug



*Memory		
RAM (*)	16	GB
Reservation (*)	16	GB
	✓ Reserve all guest memory (All locked)	
Limit	Unlimited	MB
Shares	Normal	163840
Memory Hot Plug (*)	✓ Enable	

3.5 *Storage Configuration*

Physical SQL Server environments use resources that are isolated and not shared. When you move to virtualized SQL Server deployments, a shared storage model strategy provides many benefits, such as more effective storage resource utilization, reduced storage white space, better provisioning, and improved mobility using vSphere vMotion and vSphere Storage vMotion. However, note that some things are different under virtualization, such as needing to be sure that one data store is not a single point of failure and that the storage data stores can be shared between multiple SQL Serve deployments.

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. 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.

For information about best practices for SQL Server storage configuration, refer to Microsoft's *Storage Top Ten Practices* (http://technet.microsoft.com/en-us/library/cc966534.aspx). Follow these recommendations along with the best practices in this guide.

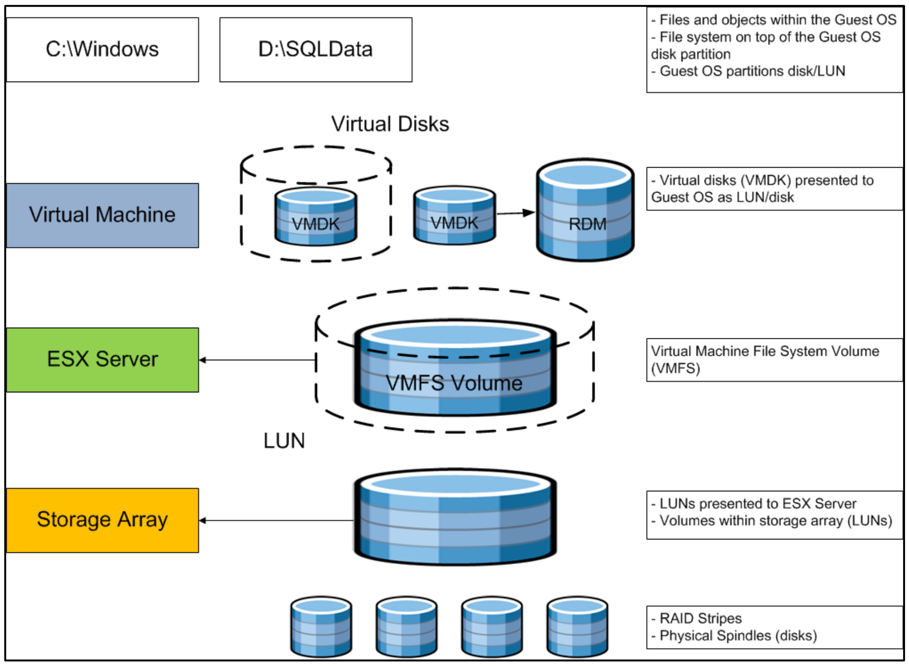
3.5.1 vSphere Storage Options

vSphere provides several options for storage configuration. The one that is the most widely used is a VMFS formatted datastore on central storage system, but that is not the only option. Today, storage admins can utilize new technologies such as VMware vSphere Virtual Volumes™ which takes storage management to the next level, where VMs are native objects on the storage system. Other options include hyper-converged solutions, such as VMware vSAN™ and/or all flash arrays, such as EMC's XtremIO. This section covers the different storage options that exist for virtualized SQL Server deployments running on vSphere.

3.5.1.1. VMFS on Central Storage Subsystem

This is 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 14. VMware Storage Virtualization Stack


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

3.5.1.2. 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 Fiber Channel and iSCSI protocols for VMFS.

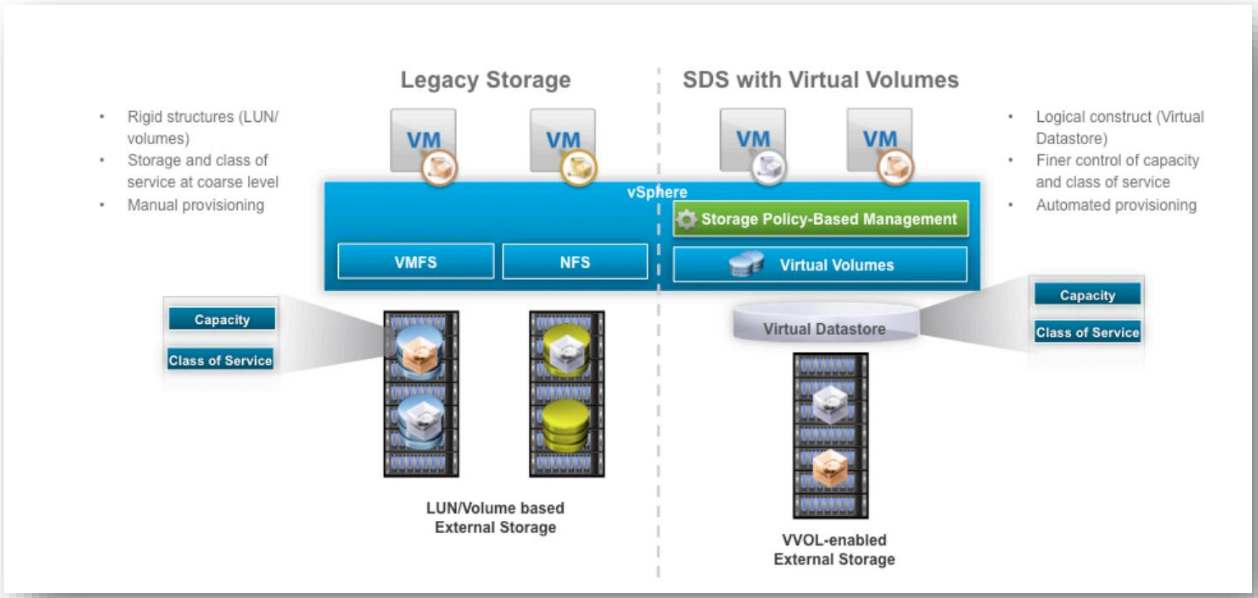
3.5.1.3. vSphere Virtual Volumes

vSphere Virtual Volumes implements the core tenants of the VMware software-defined storage vision to enable a fundamentally more efficient operational model for external storage in virtualized environments centering it on the application instead of the physical infrastructure. 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.

vSphere Virtual Volumes is a new VM disk management and integration framework that exposes virtual disks as primary units of data management for storage arrays. This new framework enables array-based operations at the virtual disk level that can be precisely aligned to application boundaries eliminating the use of LUNs and VMFS datastores. vSphere Virtual Volumes is composed of these key implementations:

- 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) that can be more flexibly consumed and configured to span a portion of one or several storage arrays.
- 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.). In other terms, with vSphere Virtual Volumes, the virtual disk becomes the primary unit of data management at the array level. This turns the virtual datastore into a VM-centric pool of capacity. 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, and so on to individual VMs. This allows admins to provide the correct storage service levels to each individual VM.
- Efficient operations through automation – 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 necessary data services. Through policy enforcement, SPBM also automates service-level monitoring and compliance throughout the lifecycle of the VM.

Figure 15. vSphere Virtual Volumes


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

The goal of vSphere Virtual Volumes is to provide a simpler operational model for managing VMs in external storage while leveraging the rich set of capabilities available in storage arrays.

For more information about virtual volumes, see the *What's New: vSphere Virtual Volumes* white paper at https://www.vmware.com/files/pdf/products/virtualvolumes/VMware-Whats-New-vSphere-Virtual-Volumes.pdf.

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.

It is a challenge to back up multi-terabyte databases due to the restricted backup windows and the data churn which itself can be quite large. It is not feasible to make full backups of these multi-terabyte backups in the allotted backup windows.

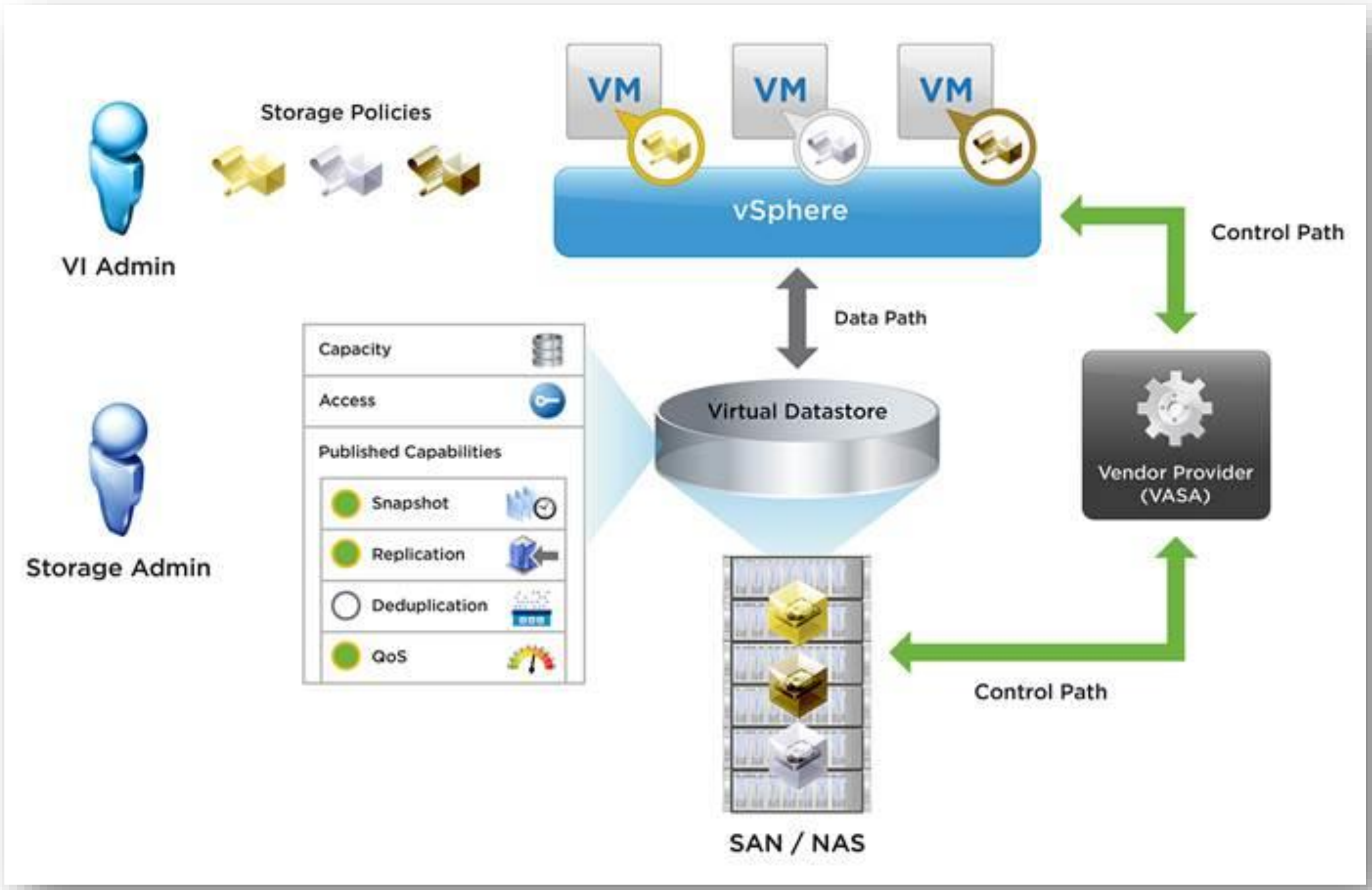
Backup solutions, such as native SQL Server backup, provide a fine level granularity for database backups but they are not always the fastest.

A VM snapshot containing the SQL Server backup file is ideal for solving this issue, but as indicated in the Knowledge Base article, *A snapshot removal can stop a virtual machine for a long time* (1002836) at http://kb.vmware.com/kb/1002836 , the brief stun moment of the VM can potentially cause performance issues.

Storage based snapshots would be the fastest, but unfortunately storage snapshots are taken at the datastore and LUN levels and not at the VM level. Therefore, there is no VMDK level granularity with traditional storage level snapshots.

vSphere Virtual Volumes is an ideal solution that combines snapshot capabilities at the storage level with the granularity of a VM level snapshot.

Figure 16. vSphere Virtual Volumes High Level Architecture



With vSphere Virtual Volumes, you can also set up different storage policies for different VMs. These policies instantiate themselves on the physical storage system, enabling VM level granularity for performance and other data services.

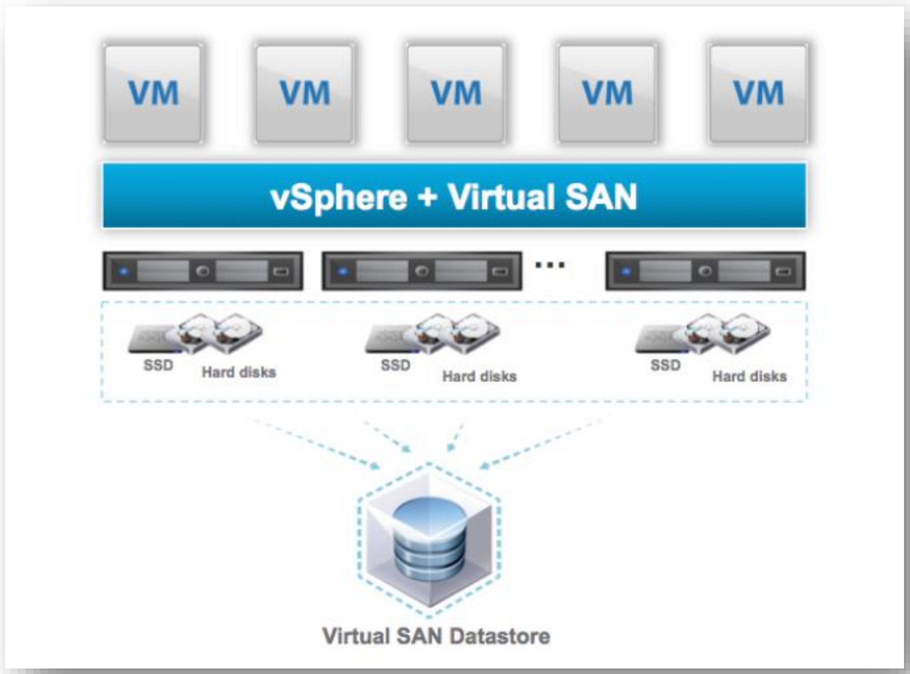
When virtualizing SQL Server on a SAN using vSphere Virtual Volumes as the underlying technology, the best practices and guidelines remain the same as when using a VMFS datastore.

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.

3.5.1.4. vSAN

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. vSAN provides enterprise-class storage services for virtualized production environments along with predictable scalability and all-flash performance at a fraction of the price of traditional, purpose-built storage arrays. Like vSphere, vSAN provides users the flexibility and control to choose from a wide range of hardware options and easily deploy and manage them for a variety of IT workloads and use cases.

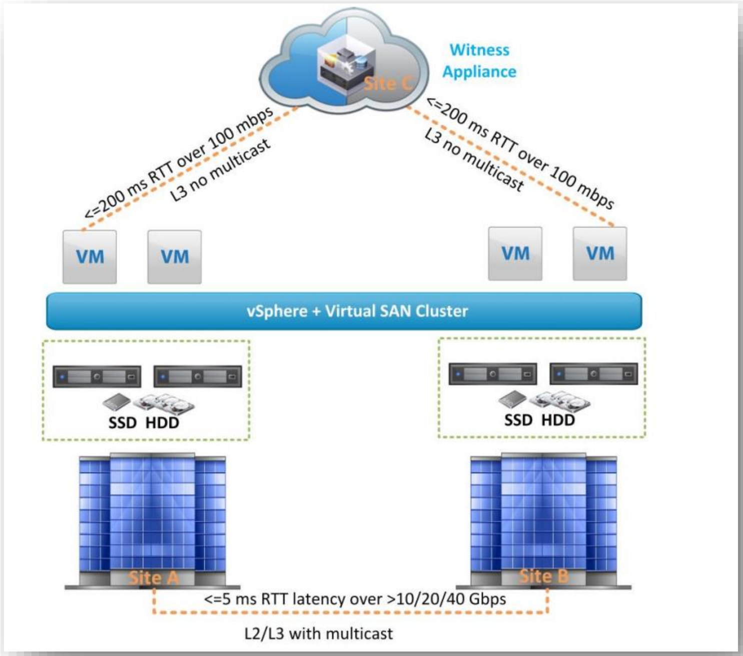
Figure 17. VMware vSAN



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.

vSAN 6.1 introduced the stretched cluster feature. vSAN stretched clusters provide customers with the ability to deploy a single vSAN cluster across multiple data centers. vSAN stretched cluster is a specific configuration implemented in environments where disaster or downtime avoidance is a key requirement. vSAN stretched cluster builds on the foundation of fault domains. The fault domain feature introduced rack awareness in vSAN 6.0. The feature allows customers to group multiple hosts into failure zones across multiple server racks to ensure that replicas of VM objects are not provisioned on to the same logical failure zones or server racks. vSAN stretched cluster requires three failure domains based on three sites (two active/active sites and one witness site). The witness site is only utilized to host witness virtual appliances that store witness objects and cluster metadata information and provide cluster quorum services during failure events.

Figure 18. vSAN Stretched Cluster


Witness Appliance
Site C
<=200 ms RTT over 100 mbps
L3 no multicast
<=200 ms RTT over 100 mbps
L3 no multicast
VM
VM
VM
VM
vSphere + Virtual SAN Cluster
SSD HDD
SSD HDD
Site A
Site B
<=5 ms RTT latency over >10/20/40 Gbps
L2/L3 with multicast

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

- Build vSAN nodes for your business requirements – vSAN is a software solution. As such, customers can design vSAN nodes from the "ground up" that are customized for their own specific needs. In this case, it is imperative to use the appropriate hardware components that fit the business requirements.
- Plan for capacity – The use of multiple disk groups is strongly recommended to increase system throughput and is best implemented in the initial stage.
- Plan for performance – It is important to have sufficient space in the caching tier to accommodate the I/O access of the OLTP application. The general recommendation of the SSD as the caching tier for each host is to be at least 10 percent of the total storage capacity. However, in cases where high performance is required for mostly random I/O access patterns, VMware recommends that the SSD size be at least two times that of the working set.

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

  - SSD size to cache active user database – The I/O access pattern of the TPC-E like OLTP is small (8 KB dominant), random, and read-intensive. To support the possible read-only workload of the secondary and log hardening workload, VMware recommends having two times the size of the primary and secondary database. For example, for a 100-GB user database, design 2 x 2 x 100 GB SSD size.

- Select appropriate SSD class to support designed IOPS – For the read-intensive OLTP workload, the supported IOPS of SSD depends on the class of SSD. A well-tuned TPC-E like workload can have ten percent write ratio.
- The *VMware Compatibility Guide* at https://www.vmware.com/resources/compatibility/search.php specifies the following designated flash. For optimal performance, VMware recommends using a flash-device class that meets workload performance requirements:
  - Class A: 2,500–5,000 writes per second
  - Class B: 5,000–10,000 writes per second
  - Class C: 10,000–20,000 writes per second
  - Class D: 20,000–30,000 writes per second
  - Class E: 30,000+ writes per second

- 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 AlwaysOn to put the database in the high availability state when the AlwaysOn in synchronous mode. Setting FTT greater than 1 means more write copies to vSAN disks. Unless special data protection is required, FTT=1 can satisfy most of the mission-critical SQL Server databases with AlwaysOn enabled.
- Set proper SPBM – vSAN SPBM can set availability, capacity, and performance policies per VM:
- Set object space reservation – Set 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 spread database components across disk groups in a vSAN cluster.

- In All Flash vSAN, for read-intensive OLTP databases, such as TPC-E-like databases, the most space requirement comes from data including table and index, and the space requirement for transaction log 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 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 width (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 2 milliseconds in all test configurations.
- VMware suggests setting stripe width as needed to split the disk object 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.

For more information about the implementation of hybrid vSAN with SQL Server solution, see the *Microsoft SQL Server 2014 on VMware Virtual SAN 6.1 Hybrid* white paper at http://www.vmware.com/files/pdf/products/vsan/microsoft-sql-on-vrtual-san61-hybrid.pdf.

For more information about the implementation of All-Flash vSAN with SQL Server solution, see the Microsoft SQL Server 2014 on VMware vSAN 6.2 All-Flash paper at

https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/products/vsan/vmware-microsoft-sql-on-all-flash-virtual-san-6-2.pdf

3.5.1.5. Raw Device Mapping

VMware also supports Raw Device Mapping (RDM). 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 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. Today there is no reason to use RDMs except in certain circumstances, such as when using FCI clustering between hosts, or when there is a snapshot software that requires it. RDM in physical compatibility mode is required for AlwaysOn FCI configuration to allow the persistent SCSI reservations and the cluster to function correctly. RDM and virtual disks can reach the same size of 62 TB and can be increased in size without shutting down the virtual machine.

From a performance perspective, both VMFS and RDM volumes can provide similar transaction throughput. The following charts summarize some performance testing. For more details, see *Performance Characterization of VMFS and RDM Using a SAN* (http://www.vmware.com/files/pdf/performance_char_vmfs_rdm.pdf).

Figure 19. Random Mixed (50% Read/50% Write) I/O Operations per Second (Higher is Better)

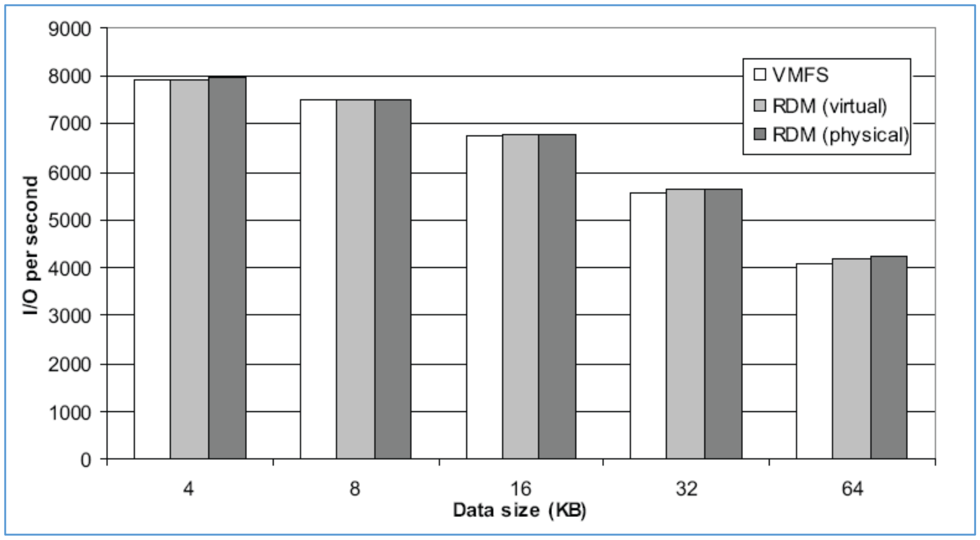
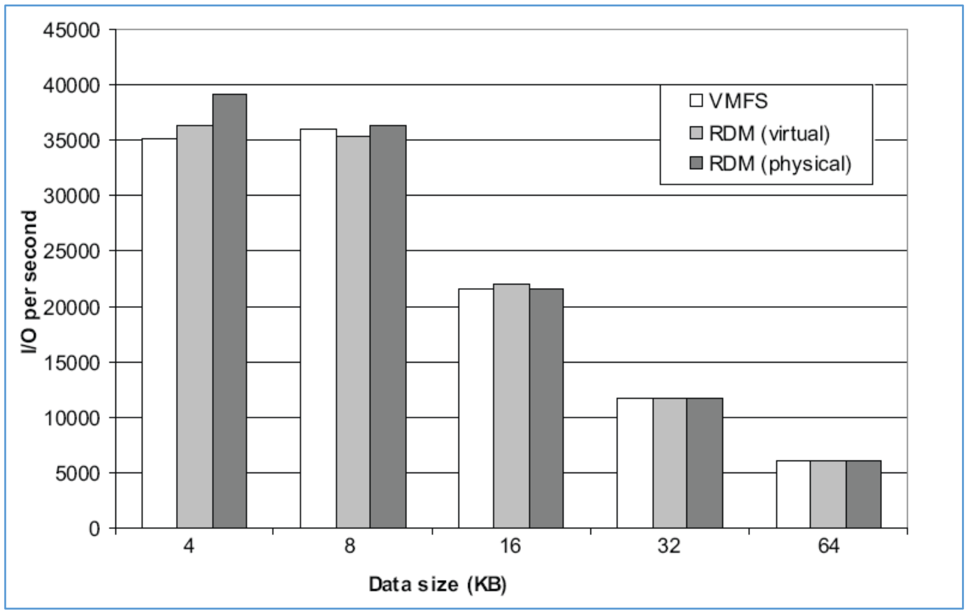


Figure 20. Sequential Read I/O Operations per Second (Higher is Better)



3.5.2 Allocating Storage

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

3.5.2.1. 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 penalty 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. 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 VMware vSphere 6.0* (http://www.vmware.com/files/pdf/techpaper/VMware-PerfBest-Practices-vSphere6-0.pdf).

3.5.2.2. VMDK File Layout

When running on VMFS, virtual machine disk files can be deployed in three different formats: thin, zeroedthick, 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 could be significant.

When the underlying storage system is enabled by VMware vSphere Storage APIs - Array Integration 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, vSphere Storage APIs - Array Integration 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. All flash arrays utilize a 100 percent thin provisioning mechanism to be able to have storage on demand.

3.5.2.3. 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	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 can improve performance. SQL Server accesses data and transaction log files with very different I/O patterns. While data file access is mostly random, transaction log file access is sequential only. Traditional storage built with spinning disk media requires repositioning of the disk head for random read and write access. 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.

The following guidelines can help to achieve best performance:

- Place SQL Server data (system and user), transaction log, and backup files into separate VMDKs (if not using RDMs). 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 for I/O and latency requirements as well. When manageability is a concern, group VMDKs and SQL Server files with similar I/O characteristics on common LUNs while making sure that the underling physical device can accommodate the aggregated I/O requirements of all the VMDKs.
- For underlying storage, where applicable, RAID 10 can provide the best performance and availability for user data, transaction log files, and TempDB.

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

- Deploying multiple, lower-tier SQL Server systems on VMFS facilitates easier management and administration of template cloning, snapshots, and storage consolidation.

- Manage performance of VMFS. The aggregate IOPS demands of all VMs on the VMFS should not exceed the IOPS capability the physical disks.
- Use VMware vSphere Storage DRS™ for automatic load balancing between datastores to provide space and avoid I/O bottlenecks as per pre-defined rules.

3.5.2.4. Using the PVSCSI Virtual Adapter

Utilize the VMware Paravirtualized SCSI (PVSCSI) Controller as the virtual SCSI Controller for data and log VMDKs. The PVSCSI Controller is the optimal SCSI controller for an I/O-intensive application on vSphere. This controller has a queue depth of 64 (Per Device) and 254 (Per Controller) by default (double the size of an LSI Logic SAS controller). The PVSCSI controller's per-device and per-controller queue depths can also be increased to 254 and 1024 respectively, providing even more increased I/O bandwidth for the virtualized workload. See *Configuring disks to use VMware Paravirtual SCSI (PVSCSI) adapters (1010298)* at http://kb.vmware.com/kb/1010398. Also, see the KB article at http://kb.vmware.com/kb/2053145 for information on how to increase queue depths in a VMware vSphere environment.

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. 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.

Use multiple vSCSI adapters. 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.

Spread the I/O load across all PVSCSI adapters to help optimize the I/O from the guest. In cases where there is a requirement for many data and transaction log disks, it will be beneficial to set the OS boot disk to use PVSCSI as well. To do that, during the OS installation you can provide the PVSCSI adapter driver to the OS installation.

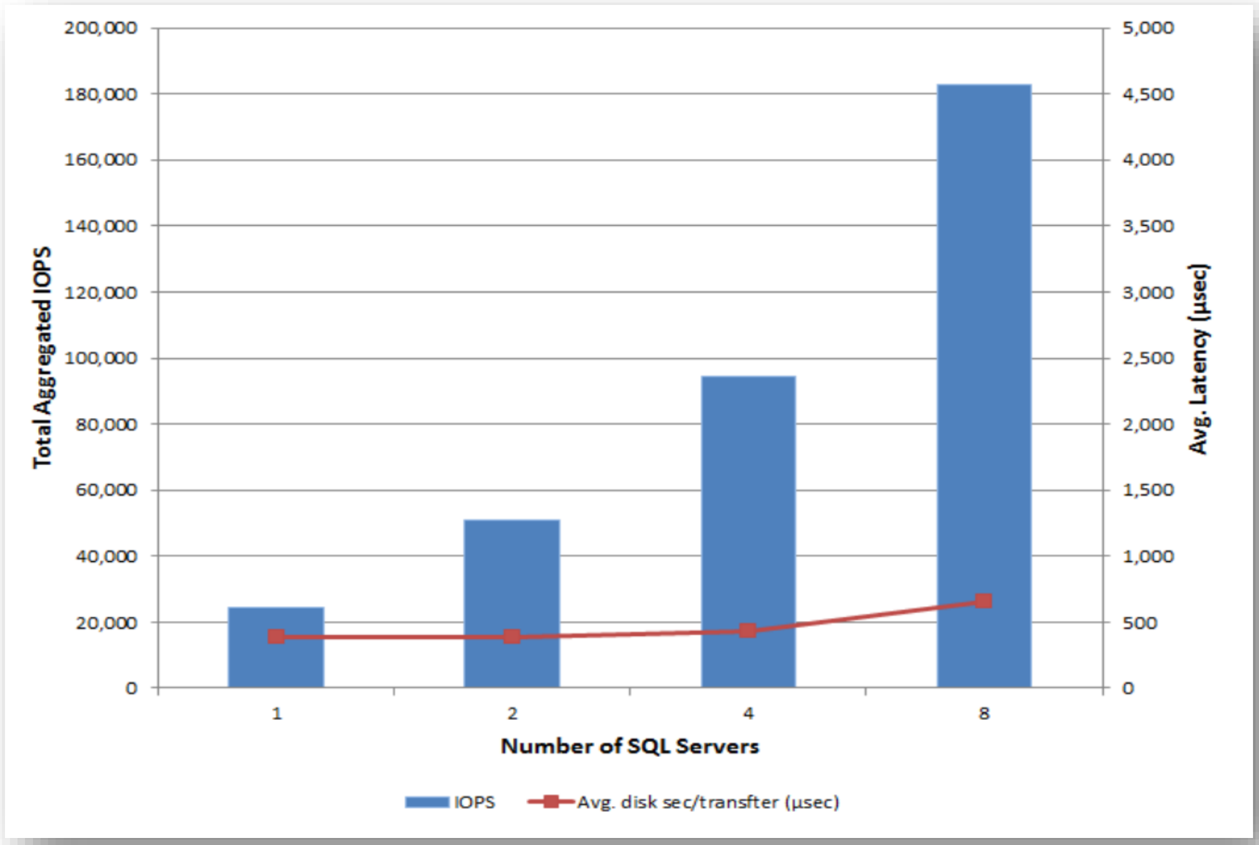
3.5.3 Considerations for Using All-Flash Arrays

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 on XtremIO done 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 21. XtremIO Performance with Consolidated SQL Server



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

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

- RAID configuration
- Separation of SQL Server files

3.5.3.1. 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 RAID configuration is needed.

3.5.3.2. 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, TempDB files on an all-flash array does not result in any significant performance difference on these modern arrays.

Nevertheless, VMware recommends that, unless explicitly discouraged by corporate mandates, customers should separate the virtual disks for the TempDB volumes allocated to a high-transaction SQL Server virtual machine on vSphere, even when using an all-flash storage array. The TempDB is a global resource that is shared by all databases within an SQL Server instance. It is a temporary work space that is recreated each time an 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.

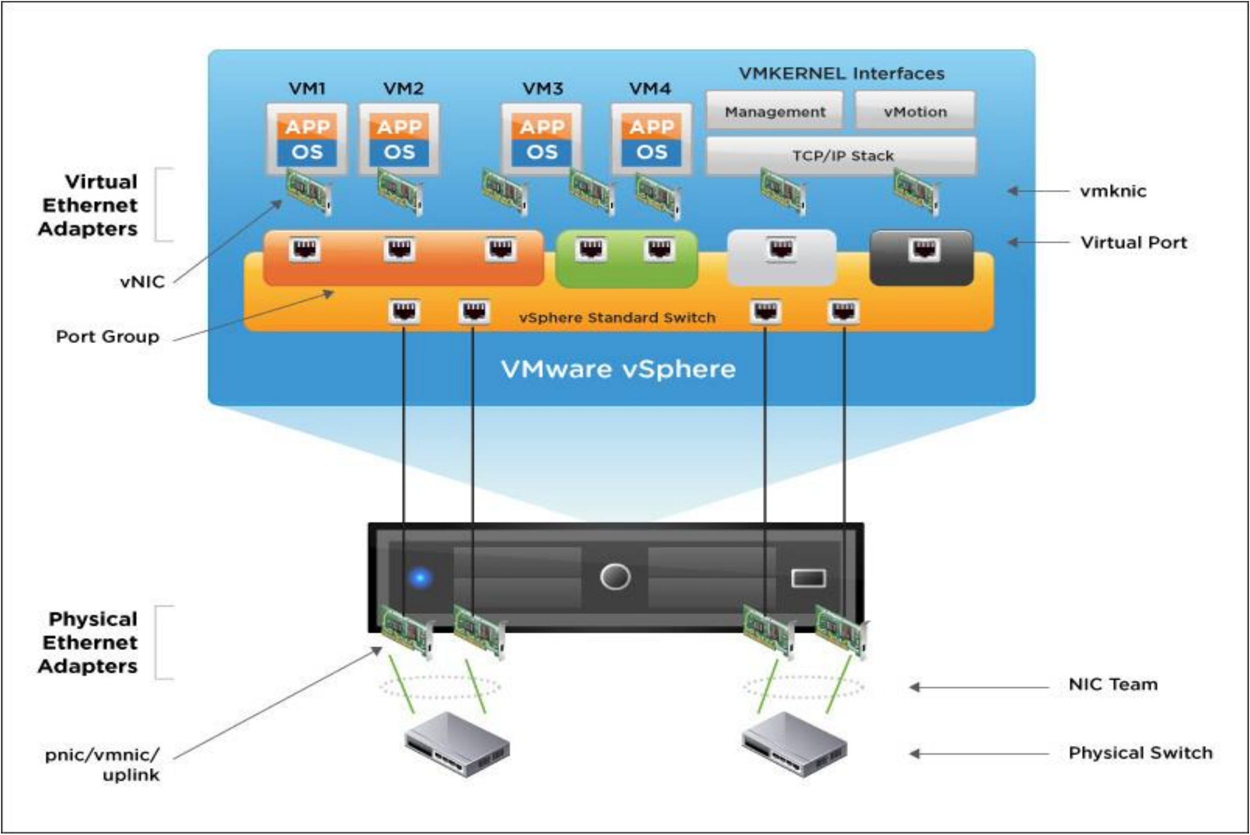
3.6 Network Configuration

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.

3.7 Virtual Network Concepts

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

Figure 22. 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.

3.7.1 Virtual Networking Best Practices

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

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

- The choice between standard and distributed switches should be made outside of the SQL Server design. Standard switches provide a straightforward configuration on a per-host level. For reduced management overhead and increased functionality, consider using the distributed virtual switch. Both virtual switch types provide the functionality needed by SQL Server.
- Traffic types should be separated to keep like traffic contained to designated networks. vSphere can use separate interfaces for management, vSphere vMotion, and network-based storage traffic. Additional interfaces can be used for VM traffic. Within VMs, different interfaces can be used to keep certain traffic separated. Use 802.1q VLAN tagging and virtual switch port groups to logically separate traffic. Use separate physical interfaces and dedicated port groups or virtual switches to physically separate traffic.
- If using iSCSI, the network adapters should be dedicated to either network communication or iSCSI, but not both.
- VMWare highly recommends considering enabling jumbo frames on the virtual switches where you have enabled vSphere vMotion traffic or iSCSI traffic. Make sure that jumbo frames are also enabled on your physical network infrastructure before making this configuration on the virtual switches.
- 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 6.0* guide (http://www.vmware.com/files/pdf/techpaper/VMware-PerfBest-Practices-vSphere6-0.pdf?vmw_so_vex_mande_12).

3.7.2 Using multinic 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 vSphere host to the target vSphere host iteratively. The first iteration copies all the memory, subsequent iterations copy only the memory pages that were modified during the previous iteration. The final phase is the switchover, where the VM is momentarily quiesced on the source vSphere host and the last set of memory changes are copied to the target vSphere host, and the VM is resumed on the target vSphere 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 vSphere 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 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 refer to the following kb article: https://kb.vmware.com/kb/2007467

For more information about vMotion architecture and SDPS, see the *VMware vSphere vMotion Architecture, Performance and Best Practices in VMware vSphere 5* paper at https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/vmware-vmotion-performance-vsphere5.pdf

Figure 24. vMotion of a Large Intensive VM with SDPS Activated

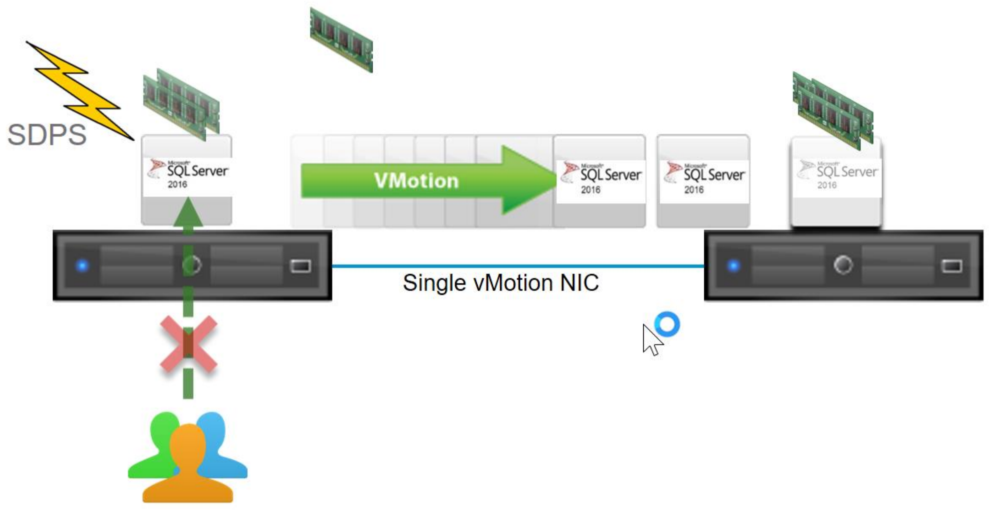
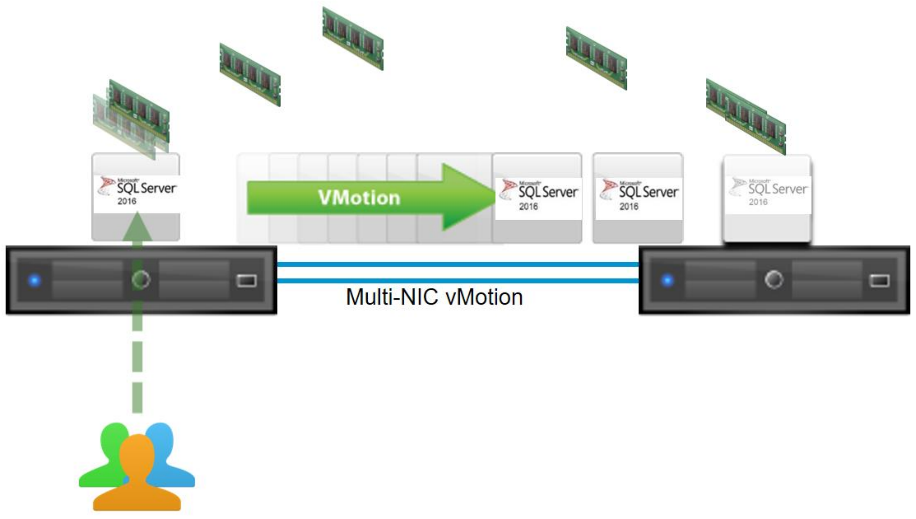


Figure 25. Utilizing Multi-nic vMotion to Speed Up vMotion Operation



3.6.4. Enable Jumbo Frames for vSphere vMotion Interfaces

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

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

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

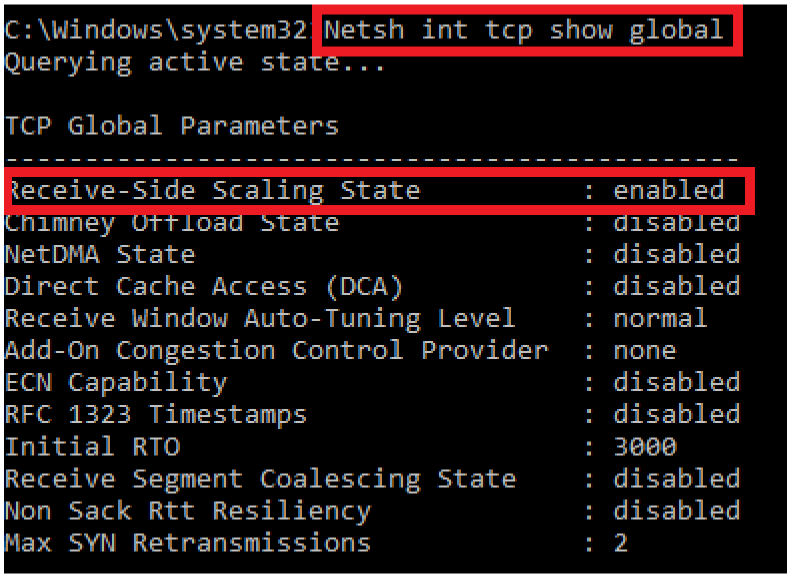
4. SQL Server and In-Guest Best Practices

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

4.1 Windows Server Configuration

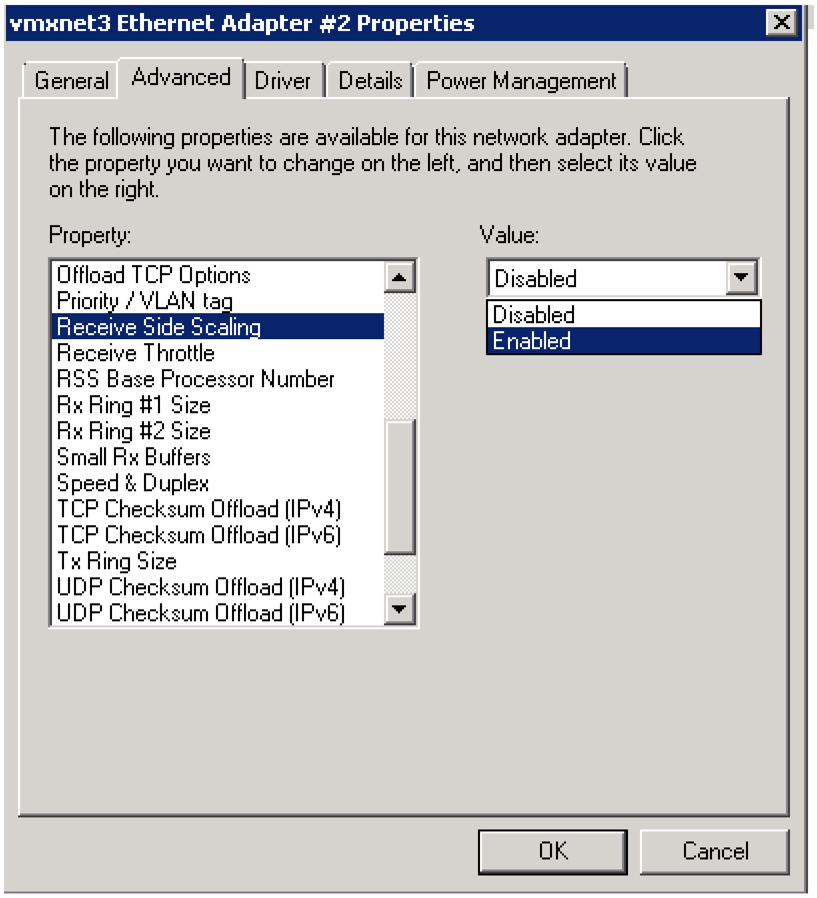
The following list details the configuration optimization that can be done on the Windows operating system:

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



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

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

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



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

Property:

Offload TCP Options
Priority / VLAN tag
Receive Side Scaling
Receive Throttle
RSS Base Processor Number
Rx Ring #1 Size
Rx Ring #2 Size
Small Rx Buffers
Speed & Duplex
TCP Checksum Offload (IPv4)
TCP Checksum Offload (IPv6)
Tx Ring Size
UDP Checksum Offload (IPv4)
UDP Checksum Offload (IPv6)

Value:

Disabled

Disabled
Enabled

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

- As detailed in Section 3.2.3.2, Windows Guest Power Settings, set the Windows power configuration to “High Performance”.
- As detailed in Section 3.5.2.4, when using PVSCSI, Windows Server is not aware of the increased I/O capabilities. The queue depth needs to be adjusted for PVSCSI in Windows Server to 254 for maximum performance. This is achieved by adding the following key in the Windows Server registry “HKLM\SYSTEM\CurrentControlSet\services\pvscsi\Parameters\Device /v DriverParameter /t REG_SZ /d "RequestRingPages=32,MaxQueueDepth=254". See the Knowledge Base article at http://kb.vmware.com/kb/2053145.
- Use the latest VMware Tools and the latest VM hardware versions because this can affect the VM’s performance. For example, in vSphere 5.5 and VM hardware version 10, VMware improved Microsoft Windows VMs timekeeping which improved the Windows VMs performance. This improvement was following Microsoft’s release of the requirements to implement the Microsoft hypervisor interface. For more information about this improvement, see the *Microsoft Operating System Time Sources and Virtual Hardware 10* blog post at https://blogs.vmware.com/vsphere/2013/10/microsoft-operating-system-time-sources-and-virtual-hardware-10.html.

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

For Tier 1 mission-critical SQL Server deployments, consider setting the SQL Server memory to a fixed amount by setting both maximum and minimum server memory to the same value. Before setting the maximum and minimum server memory, confirm that adequate memory is left for the 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

                    = 4MB on IA64

OS Mem: 1GB for every 4 CPU Cores
```

4.3 Lock Pages in Memory

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

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

If you are deploying a Tier 1 mission-critical SQL Server installation, consider setting the Lock Pages in Memory user right and setting VM memory reservations to improve the performance and stability of your SQL Server running vSphere. Setting VM memory reservations eliminates contention for the VM on memory resources. This prevents the balloon driver from inflating into the SQL Server VM's memory space. For instructions on enabling Lock Pages in Memory, refer to *Enable the Lock Pages in Memory Option (Windows)* (http://msdn.microsoft.com/en-us/library/ms190730.aspx). Lock Pages in Memory should also be used in conjunction with the Max Server Memory setting to avoid SQL Server taking over all memory on the VM.

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

4.4 Large Pages

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. Refer to *Large Page Performance* http://www.vmware.com/resources/techresources/1039 for additional information.

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 834 if using the Columnstore feature.
- 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 startup behavior changes. Instead of allocating memory dynamically at runtime, SQL Server allocates all buffer pool memory during startup. Therefore, SQL Server startup time can be significantly delayed.
- With trace flag 834 enabled, SQL Server allocates memory in 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, 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.
```

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

4.5 CXPACKET, MAXDOP, and CTFP

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

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

CTFP, or cost threshold for parallelism, is an option that specifies the threshold at which parallel plans are used for queries. The value is specified in seconds and the default is 5, which means a parallel plan for queries is used if SQL estimates it would take longer than 5 seconds when run serially. 5 is typically considered too low for today's CPU speeds.

There is a fair amount of misconception and incorrect advice on the Internet regarding the values of these configurations in a virtual environment. When low performance is observed on their database, and CXPACKET is high, many DBAs decide to disable parallelism altogether by setting MAXDOP value to 1. This is not recommended because there might be large jobs that will benefit from processing on multiple CPUs. The recommendation instead is to increase the CTFP value from 5 seconds to approximately 50 seconds to make sure only large queries run in parallel. Set the MAXDOP according to Microsoft's recommendation for the number of cores in the VM's NUMA node (no more than 8).

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

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

4.6 Using Antivirus

Customers might have requirements that virtual 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. See KB309422, *How to choose antivirus software to run on computers that are running SQL Server* at https://support.microsoft.com/en-us/help/309422/how-to-choose-antivirus-software-to-run-on-computers-that-are-running-sql-server

5. VMware Enhancements for Deployment and Operations

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

5.1 Network Virtualization with VMware NSX for vSphere

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

5.1.1 NSX Edge Load balancing

VMware NSX Edge™ provides load balancing for VMs through a virtual appliance. NSX Edge can be deployed in a high availability pair, providing better protection than hardware load balancing solutions without the additional hardware or management overhead. The NSX Edge load balancer enables application or service requests to be distributed across multiple backend servers in a pool. NSX Edge load balancer includes the following functions:

- Virtual IP (VIP) Address – An IP address and service port number used by the user to access the service.
- Server Pool – The pool of backend servers that need to be load balanced. A VIP address is associated with the server pool.
- Service Monitor – Defines the health check parameters for a particular type of network traffic. A service monitor is associated with the server pool to monitor the pool members.
- Application Profile – Defines the behavior of a particular type of network traffic. Examples include the session persistence parameter and SSL parameters.

VMware NSX 6.2 introduces support for more VIPs, from 64 to 1024 per edge. This reduces the number of edges that need to be deployed, thereby reducing the data center footprint and improving scalability in terms of manageability. VMware NSX 6.2 provides monitoring and failure reporting, support for port range when creating VIPs and tighter integration with third-party load balancers. VMware NSX 6.2 also adds support for F5 integration.

Prior to VMware NSX 6.2, health monitoring checks were configured with intervals and retries. Probes were sent at configured intervals to test the connectivity with the specific application. The default interval was 5 seconds and the default settings for retries was 3. This type of configuration cannot detect intermittent failures in the load balancer pool, making troubleshooting difficult.

With VMware NSX 6.2, health monitoring is more granular. The edge can immediately report specific probe failures. The system can also track last health check and status changes that a pool has experienced. It can also report the reason for failures. All these improvements simplify troubleshooting.

Prior to VMware NSX 6.2, a VIP was a combination of the VIP IP address and a port. Applications that must listen on multiple ports or port ranges require the creation of multiple VIPs—one for every port. This significantly affects the scalability of the solution in terms of operations and troubleshooting and increases the number of edges that need to be deployed.

VMware NSX 6.2 adds support for listening on multiple port ranges. A VIP can be associated with multiple ports or port ranges, thereby improving the scalability and reducing the number of edges that need to be deployed.

NSX Edge supports both Layer 7 (the recommended load-balancing option without session affinity requirements in Exchange Server 2016) and Layer 4 load balancing of HTTP and HTTPS protocols. It supports multiple load balancing methods, such as round-robin and least connection. Layer 7 HTTP/HTTPS VIP addresses are processed after passing the NSX Edge firewall. NSX Edge uses the faster Layer 4 load balancer engine. The Layer 4 VIP address is processed before passing the NSX Edge firewall.

The NSX Edge services gateway supports the following deployment models for load-balancer functionality:

- One-armed load balancer
- Inline load balancer

5.1.2 VMware NSX Distributed Firewall

Customers can leverage VMware NSX for vSphere to provide application layer isolation against unauthorized access. Isolation at this level typically requires hardware firewalls and multiple VLANs in a physical networking environment. With NSX distributed firewall, this capability is delivered in software through an ESXi kernel module.

The distributed firewall provides security filtering and service chaining functions on every host prepared for VMware NSX. The throughput scales as hypervisors are added.

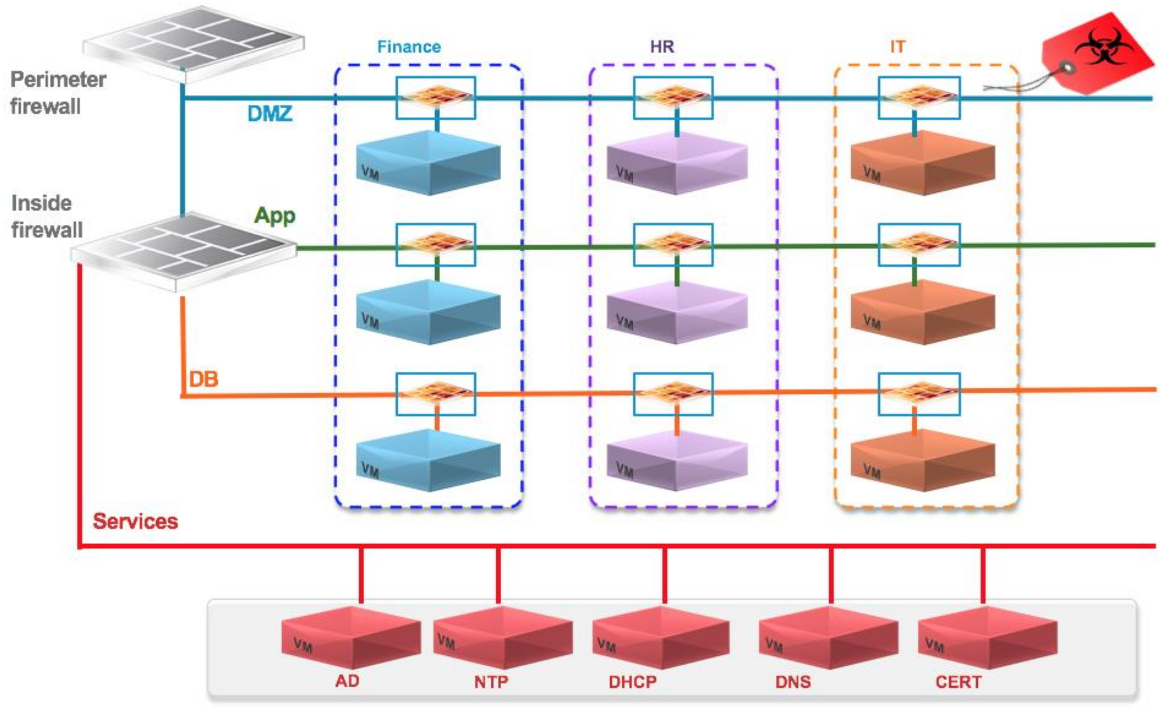
The distributed firewall kernel module is embedded in the VMkernel. It is deployed as a VMware infrastructure bundle (VIB) during the host preparation process.

The firewall rules are configured by the administrator using the vSphere Web Client or REST APIs. The VMware NSX Manager™ stores the rules in its database and pushes the policies to the ESXi hosts through the message bus agent.

You can create different types of firewall rules with VMware NSX.

Application-aware policies can use dynamic tagging and user identity. Infrastructure-aware policies can use vCenter inventory objects as the basis for rules. Network-based policies are the most common (and traditional) types of policies that use Layer 2 and Layer 3 data for the rules.

Figure 23. NSX Distributed Firewall Capability

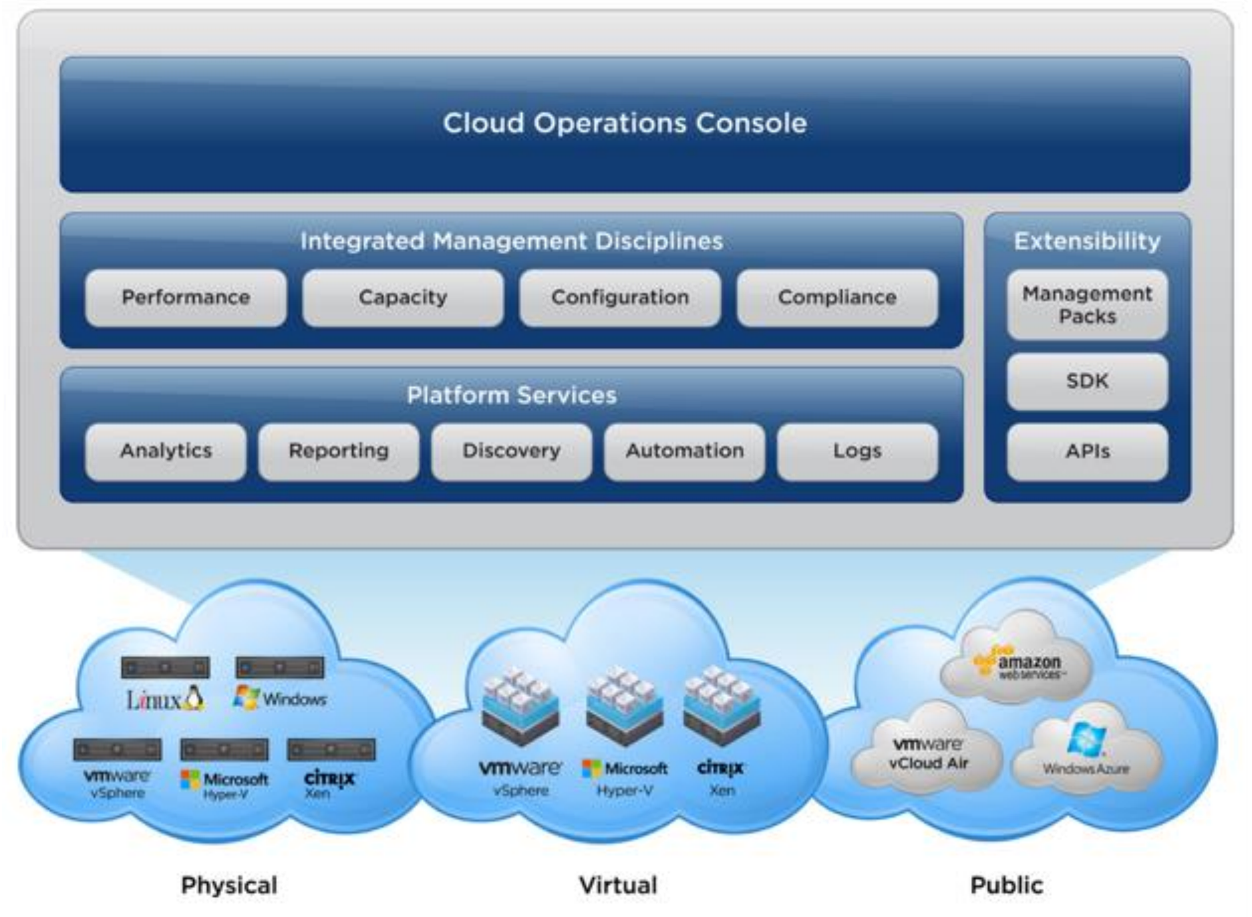


5.2 *VMware vRealize Operations Manager*

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

In an SQL Server deployment, constant monitoring is required to maintain acceptable service levels. vRealize Operations Manager includes patented capacity analytics that can eliminate the need for spreadsheets, scripts, or rules of thumb. Quickly run through "what if" capacity scenarios to understand growth trends and identify upcoming compute power shortages or over-provisioned resources. As an application comprising multiple components, SQL Server performance and functionality can be affected by changes made at many levels. vRealize Operations Manager monitors configurations across VM and detects unwanted changes to help maintain continuous compliance with operational best practices.

Figure 24. vRealize Operations



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

- Utilize the EPO management pack for MS SQL Server provided by VMware – This management pack is included with vRealize Operations™ Enterprise and can be implemented by the customer or VMware services. The EPO management pack is collecting information from SQL Server deployments using an agent and does not include capacity management information.
- BlueMedora management pack for SQL Server– While this solution incurs additional cost, it provides added value with agentless integration and includes capacity information from which you can build "what if" scenario analysis for SQL Server.

For more information about Blue Medora management packs, see http://www.bluemedora.com/wp-content/uploads/2015/06/vROps-MP-MSSQL-Server-June-2015.pdf.

6. Acknowledgments

Author: Niran Even-Chen, Staff Solutions Architect – Microsoft Applications.

Thanks to the following people for their inputs:

- Deji Akomolafe – Staff Solutions Architect, Microsoft Applications
- Mark Achtemichuk – Sr. Technical Marketing Architect, Performance Specialist
- Scott Salyer – Director, Enterprise Application Architecture
- Sudhir Balasubramanian – Senior Solution Architect – Data Platforms
- Vas Mitra – SAP Solutions Architect
- Agustin Malanco - Solutions Architect - global services advanced architecture support
- Oleg Ulyanov – Sr. Consultant, VMware PSO
- Mohan Potheri – Sr. Solutions Architect, Technical Marketing
- Allan Hirt – Managing Partner, Dual Microsoft MVP, and vExpert at SQLHA LLC
- David Klee - Founder and Chief Architect, Heraflux Technologies