

Architecting Business Critical Applications on VMware Hybrid Cloud

BEST PRACTICES GUIDE

Table of contents

Introduction	5
Purpose	5
Target Audience	5
Technology Overview	6
VMware vSphere	6
VMware vSAN	6
Best Practices for Deploying BCA Workloads on VMware vSphere	7
Rightsizing	7
ESXi Host Configuration	8
BIOS/UEFI and Firmware Versions	8
BIOS/UEFI Settings	8
Power Management	8
ESXi CPU Configuration	10
Physical and Virtual CPU or Core	10
Hyper-Threading	11
Understanding NUMA	12
Using NUMA Best Practices	13
ESXi Memory Configuration	16
Memory Overcommit Techniques	16
vSphere 2M Large Memory Pages	17
1GB Large Memory Pages	17
Persistent Memory	18
ESXi Storage Configuration	19
vSphere Storage Options	19
VMware Virtual Disk Provisioning Policies	26
VMware vSAN Storage Policy	27
VMware Multi-Writer Attribute for Shared VMDKs	28
Clustered VMDK Support	29
vSphere APIs for Array Integration (VAAI)	29
Storage Policy-Based Management (SPBM)	30
Automatic Space Reclamation (UNMAP)	31
Advanced Format Drive and 4kn support	31
Storage Other Features	32

ESXi Network Configuration	33
Virtual Network Concepts	33
vSphere Networking Best Practices.	34
Multi-Stream Helper for vMotion	34
Jumbo Frames for vSphere vMotion Interfaces.	35
Load-Balancing on vSphere Standard Switch and Distributed Switch.	35
RDMA (Remote Direct Memory Access) over Converged Ethernet (RoCE).	36
Network Other Features	37
vSphere Security Configuration	38
vSphere Security Features.	38
Side-Channel Vulnerability Mitigation and New CPU Scheduler	39
Virtual Machine CPU Configuration.	39
Allocating vCPU	39
vNUMA, corespersocket and PreferHT	39
CPU Hot Plug and Hot Add	42
CPU Affinity	44
Latency Sensitive Setting	44
Per Virtual Machine EVC Mode	45
Virtual Machine Memory Configuration	46
Memory Sizing Considerations.	46
Memory Reservation	47
Memory Hot Plug	49
Virtual Machine Storage Configuration	50
VM Storage Best practices.	50
VM PVSCSI Storage Controller and PVSCSI/vmdk Queue Depth	50
VM vNVMe Storage Controller	50
Partition Alignment	51
VMDK File Layout	51
Virtual Disks Hot Add and Hot Remove.	52
Virtual Disks Hot Extend	52
VM Snapshot	52
BCA Workloads on VMware vSAN	54

- Virtual Machine Network Configuration55
 - Virtual Networking Best Practices55
 - Interrupt Coalescing.56
 - Receive Side Scaling (RSS).56
 - TCP Segmentation Offload.57
 - Large Receive Offload57
- Virtual Machine Maintenance58
 - Install VMware Tools58
 - Upgrade VMware Tools59
 - Virtual Machine Compatibility 61
 - Timekeeping in Virtual Machine.62
 - Time Synchronization.62
 - VM Configuration Maximums.64
- Virtual Machine Security Features64
 - Virtual Machine Encryption64
 - Virtual Machine UEFI Secure Boot.65
- Summary65
- Appendix74
- Resources78
- Acknowledgments78

Introduction

As the success and maturity of virtualization continue to exceed all expectations, enterprises have continued to benefit well beyond its competitive operational and financial advantages. Enterprises are driving more innovations and achieving better outcomes by deploying a greater proportion of their mission-critical applications and operations in various vendor-provided virtualization platforms.

VMware is universally recognized and trusted as the leader and vendor of choice in the virtualization space. The suite of VMware virtualization technologies and solutions power these innovations and create confidence, not only in traditional datacenters but in the public and hybrid cloud space. Because VMware powers most of these cloud offerings, enterprises look to VMware to provide the important guidance necessary to ensure success in their virtualization and hybrid cloud adoption journeys.

This document provides the comprehensive and prescriptive guidance operators, administrators, architects, business owners and other stakeholders can use as a building block to reliably and successfully move mission-critical applications from traditional platforms to any of the various hybrid clouds offerings on the market today.

This guide discusses the common configuration options available in a typical VMware vSphere®-based hybrid cloud infrastructure. It provides best-fit recommendations to enterprises for avoiding common performance and reliability pitfalls when running these applications, irrespective of the infrastructure they choose. Where necessary, we have provided additional background information to explain the choice of a particular option. We have also included references to more detailed reading where we believe further information could be beneficial.

This guide assumes that the reader is conversant with the basic administration of a typical VMware vSphere-based infrastructure. References to standard VMware vSphere administration documents are provided where appropriate.

Purpose

This document provides general best practice guidelines for designing and implementing business critical application (BCA) workloads on VMware vSphere platforms. The recommendations are not specific to a particular hardware set nor to the size and scope of a particular BCA workload implementation. The examples and considerations in this document provide guidance only and do not represent strict design requirements, as varying application requirements might result in many valid configuration possibilities.

Target Audience

This document assumes a knowledge and understanding of vSphere technologies. Architectural staff can use this document to understand 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 BCA workloads 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.

Technology Overview

This section provides an overview of the technologies used in this solution:

- VMware vSphere
- VMware vSAN™

VMware vSphere

VMware vSphere, the industry-leading virtualization and cloud platform, is the efficient and secure platform for hybrid clouds, accelerating digital transformation by delivering simple and efficient management at scale, comprehensive built-in security, a universal application platform, and a seamless hybrid cloud experience. The result is a scalable, secure infrastructure that provides enhanced application performance and can be the foundation of any cloud.

As the next-generation infrastructure for next-generation applications, vSphere 7.0 has been rearchitected with native Kubernetes, enabling IT admins to use VMware vCenter Server® to operate Kubernetes clusters through namespaces. VMware vSphere with Tanzu allows IT admins to leverage their existing skillset to deliver self-service infrastructure access to their DevOps teams, while providing observability and troubleshooting of Kubernetes workloads. vSphere 7 provides an enterprise platform for both traditional and modern applications, enabling customers and partners to deliver a developer-ready infrastructure, scale without compromise, and simplify operations.

Learn more about [VMware vSphere 7.0](#).

VMware vSAN

VMware vSAN is a software-defined storage solution, built from the ground up, for vSphere VMs.

It abstracts and aggregates locally attached disks in a vSphere cluster to create a storage solution that can be provisioned and managed from vCenter and the vSphere client. vSAN is embedded within the hypervisor, hence storage and compute for VMs are delivered from the same x86 server platform running the hypervisor.

Hyperconverged infrastructure (HCI) backed by VMware vSAN provides a wide array of deployment options, from a two-node setup to a standard cluster supporting up to 64 hosts. Also, vSAN accommodates a stretched cluster topology to serve as an active-active disaster recovery solution. vSAN includes HCI Mesh, which allows customers to remotely mount a vSAN datastore to other vSAN clusters, disaggregating storage and compute. This allows greater flexibility to scale storage and compute independently.

Learn more about [VMware vSAN](#).

Best Practices for Deploying BCA Workloads on VMware vSphere

A properly designed BCA workload running on vSphere is crucial to the successful implementation of enterprise applications. A key difference between designing for performance of critical databases and designing for consolidation (i.e., the traditional practice when virtualizing) is that when designing for performance, you strive to reduce resource contention between VMs as much as possible and even eliminate it altogether. The following sections outline VMware recommendations for designing and implementing your vSphere environment so as to optimize for best practices for running BCA workloads.

Rightsizing

In the context of deploying a VM, rightsizing refers to allocating the appropriate amount of compute resources (e.g., virtual CPUs and RAM) to power a BCA workload instead of adding more than is actively utilized (a common sizing practice for physical servers). Rightsizing is imperative when sizing VMs. The rightsizing approach is different for a VM compared to physical server.

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

However, when sizing a BCA workload to run on a VM, it is important to assign that VM the exact number of resources it requires at that time. This leads to optimized performance with the lowest possible overhead and can yield licensing savings through critical production BCA workload 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 VM running a BCA workload, monitor the server using workload-specific monitor tools (e.g., Oracle Dictionary Dynamic tables [v\$ tables], Oracle AWR reports, SQL Server dynamic management views [DMVs], or Windows Performance Monitor).

Many third-party monitoring tools can be used as well, such as [VMware vRealize® True Visibility Suite](#) management pack for Oracle and Microsoft applications, which can oversee the above-mentioned BCA workload monitors with respect to ongoing capacity management and provide an alert in the event of resource waste or contention. The amount of collected time series data should be enough to capture all relevant workloads spikes, such as quarter-end or monthly reports. At least two weeks or, preferably, one full business cycle should be sampled before an analysis is performed.

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

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

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

Rightsizing a VM is a complex process and wise judgement should be used to avoid over-allocating resources or underestimating the workload requirements:

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

ESXi Host Configuration

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

BIOS/UEFI and Firmware Versions

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

For more information, review:

- [Checking your firmware and BIOS levels to ensure compatibility with ESX/ESXi \(1037257\)](#)

BIOS/UEFI Settings

Recommendation: As a best practice, use the following BIOS/UEFI settings for high performance environments (when applicable):

- Enable Turbo Boost which results in balanced workload over unused cores.
- Enable Hyper-Threading.
- Verify that all ESXi hosts have NUMA enabled in the BIOS/UEFI. In some systems (e.g., HP servers), NUMA is enabled by disabling node interleaving. Consult your server hardware vendor for the applicable BIOS settings for this feature.
- Enable advanced CPU features, such as VT-x/AMD-V, EPT, and RVI.
- Follow your server manufacturer's guidance in selecting the appropriate snoop mode.
- Disable any devices that are not used (e.g., serial ports).
- Set power management (or the equivalent vendor-specific label) to "OS controlled" (or the equivalent vendor-specific label). This will enable the ESXi hypervisor to control power management based on the selected policy.
- Disable all processor C-states (including the C1E halt state). These enhanced power management schemes can introduce memory latency and suboptimal CPU state changes (halt-to-full), resulting in reduced performance for the VM.

For more information, review:

- [Configuring the BIOS Boot Settings](#)

Power Management

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

An ESXi host can take advantage of several power management features that the hardware provides to adjust the tradeoff between performance and power use.

There are four CPU power management policies that can be selected for a VM:

- **High Performance** – Maximizes performance and disables power management features.
- **Balanced (Default)** – Uses power management features that will minimally impact performance.
- **Low Power** – Uses power management features that prioritize reducing energy consumption over performance.
- **Custom** – User-defined power management policy.

You can control how ESXi uses these features by selecting a power management policy. While previous versions of ESXi default to **High Performance** power schemes, vSphere 5.0 and later defaults to a **Balanced** power scheme.

It's crucial to follow the recommendations above and configure the server BIOS/UEFI to pass the power management to ESXi (i.e., **OS control**). If this setting is not configured, ESXi power management policies will have no effect.

Setting correct power management policies in BIOS/UEFI and in ESXi should be accomplished by configuring power policies in the OS.

Some workloads might benefit from the combination of deep C states for some cores and turbo boosting another. For this combination, custom BIOS power policy should be used with deep C states enabled and ESXi power policy should be set to **Balanced**.

Recommendation: As a best practice for BCA workloads, configure the server BIOS/UEFI to pass the power management to ESXi (i.e., OS control) and change the default Balanced power scheme to High Performance.

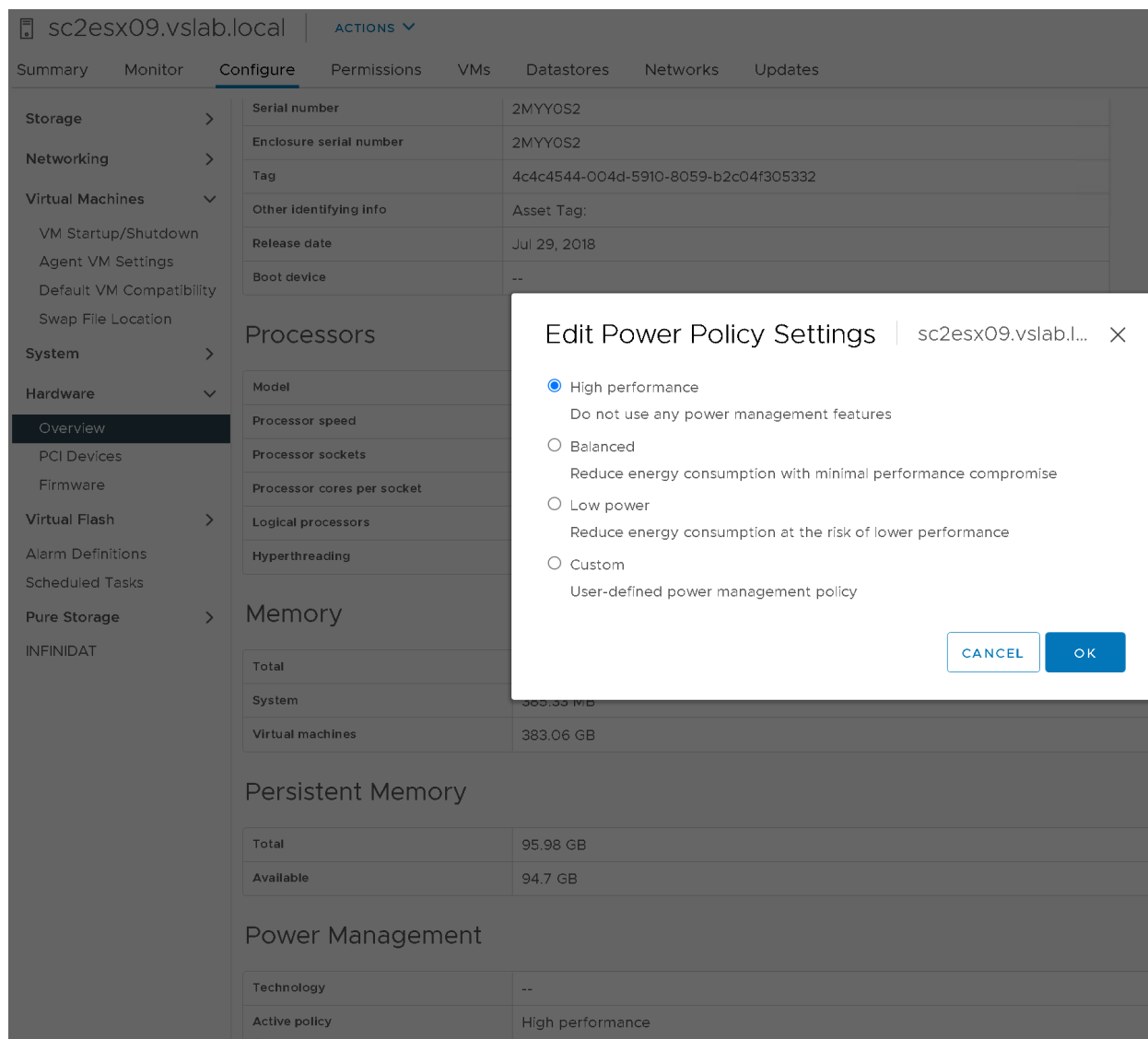


FIGURE 1. Recommended ESXi Host Power Management Setting

For more information, review:

- [Virtual machine application runs slower than expected in ESXi \(1018206\)](#)
- [Host Power Management Policies](#)

ESXi CPU Configuration

Physical and Virtual CPU or Core

VMware uses the following terms to distinguish between processors within a VM and underlying physical x86/x64-based processor cores:

Physical Server

- Physical CPU (pCPU) or physical socket – physical CPU installed in the server hardware. See **Sockets** in the illustration below.
- Physical Core (pCore) – independent processing unit residing on the same processor. See **Cores per Socket** and **CPU Cores** in the illustration below.
- Logical Core (lCore) – logical processor on a physical core with its own processor architectural state. See **Logical Processors** in the illustration below. The most widely known implementation is Intel Hyper-Threading technology.

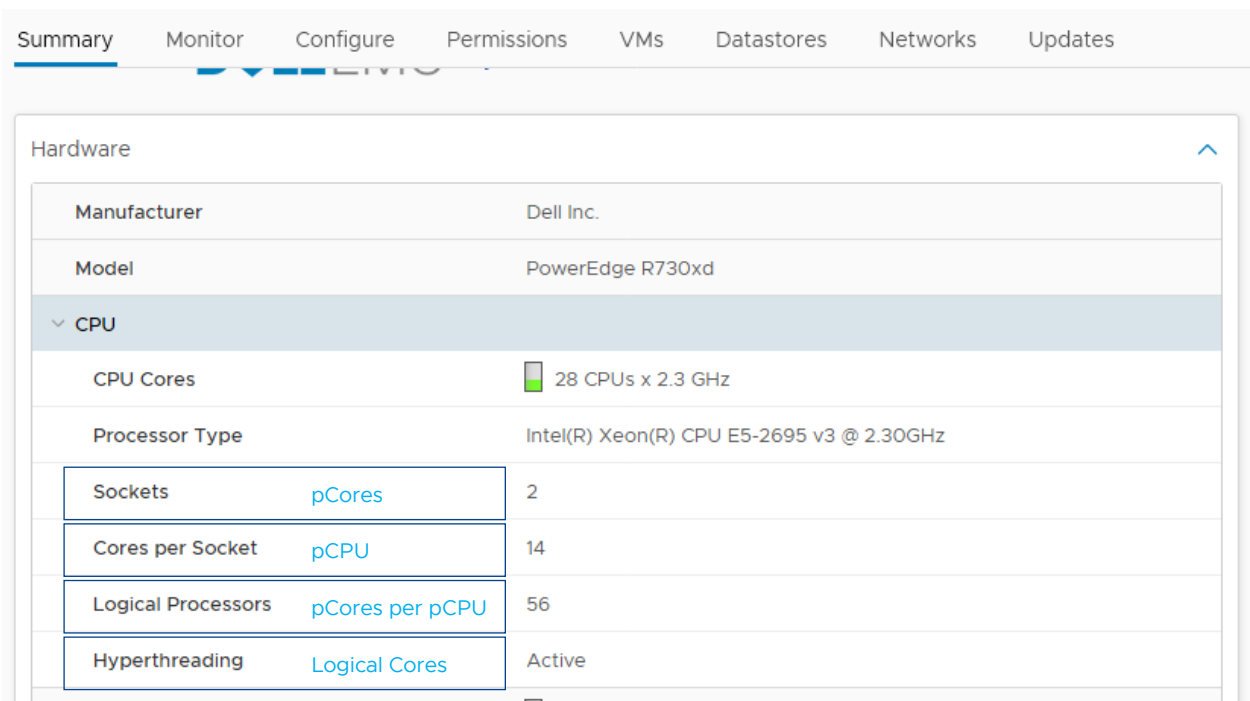


FIGURE 2. Physical Server CPU Allocation

As an example, the host listed in the illustration above has two pSockets (i.e., two pCPUs), 28 pCores and 56 logical cores as a result of active Hyper-Threading.

Virtual Machine

- Virtual Socket – each virtual socket represents a virtualized physical CPU and can be configured with one or more virtual cores. See **Sockets** in the illustration below
- Virtual Core – each virtual core is equal to a CPU and will be visible by an OS as a separate processor unit (introduced with vSphere 4.1). See **Cores per Socket** in the illustration below.
- Virtual CPU (vCPU) – virtualized central processor unit assigned to a VM. See **CPU** in the illustration below.

The total number of assigned vCPUs to a VM is calculated as:

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

As an example, a VM listed in the illustration below has two virtual sockets, each with four virtual cores, for a total number of eight vCPUs.

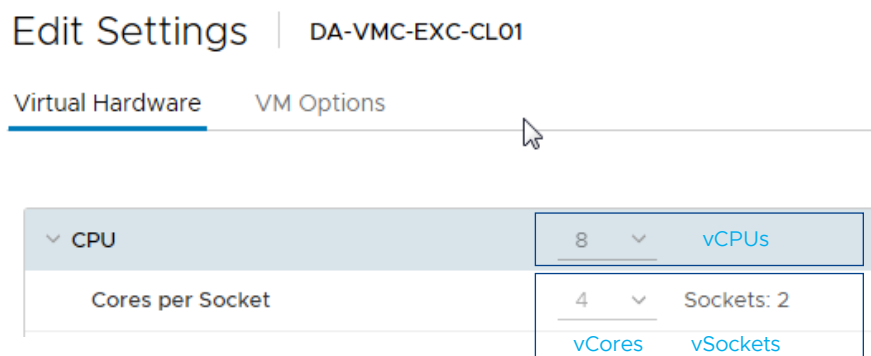


FIGURE 3. CPU Configuration of a VM

For further information, review:

- [Setting the number of cores per CPU in a virtual machine \(1010184\)](#)
- [Administering CPU Resources](#)
- [Multi-core processor](#)

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, as Hyper-Threading can provide anywhere from a slight to a significant increase (up to 24 percent) in system performance by keeping the processor pipeline busier.

By keeping the processor pipeline busier and allowing the hypervisor to have more CPU scheduling opportunities, Hyper-Threading generally improves the overall host throughput anywhere from 10 to 30 percent, allowing you to use a 1:1 to 1:3 vCPU:pCPU ratio for your VMs. Extensive testing and monitoring tools are required when following this approach.

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.

Recommendation: As a best practice, enable Hyper-Threading in the BIOS/UEFI so that ESXi can take advantage of this technology.

For further information, review:

- [Hyperthreading](#)
- [Hyper-Threading](#)
- [Intel Hyper-Threading Technology](#)

Understanding NUMA

Over last decade, few topics have raised as much attention as those surrounding non-uniform memory access (NUMA) technologies and their implementation. This is expected given the complexity of the technology, varied vendor implementations, and the number of configuration options and layers (from hardware through a hypervisor to a guest OS and an application). The consideration of NUMA hardware architecture is a must for any infrastructure architect or application owner in charge of a virtualized BCA workload.

NUMA is a hardware architecture for shared memory, implementing subdivisions of physical memory bunks between pCPUs (see **Figure 4** below for an example). In this context, local memory (i.e., on the same bus as a pCPU) and remote memory (i.e., accessed through an interconnect) concepts are introduced. Subdivision of memory is dictated by the rapidly growing number of memory consumers (i.e., CPU cores), the faster operations mode of cores, and the excessive cache-coherence traffic when two or more cores are accessing the same memory cacheline. A construct containing a pCPU, local memory and I/O modules located on the same bus is called a NUMA node.

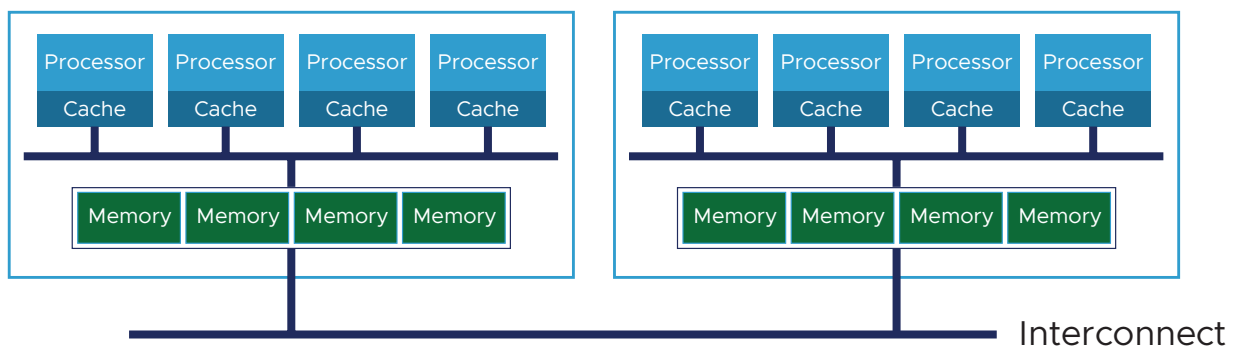


FIGURE 4. Intel based NUMA Hardware Architecture¹

While the architecture provides compelling benefits, it also poses some tradeoffs to consider. The most important of these is that the time to access data in memory will vary depending on local or remote placement of the corresponding memory cacheline to the CPU core executing the request, with remote access being up to X^2 times slower than local. This has given rise to the name *non-uniform* in defining the architecture and is the primary concern for any application deployed on top of hardware on which NUMA is implemented.

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

For further information, review:

- [Non-uniform memory access](#)

¹ [Optimizing Applications for NUMA](#)

² Depending on the implementation and the processor family, this difference could be up to 3X.
See page 6 of [Optimizing Application Performance in Large Multi-core Systems](#)

Using NUMA Best Practices

As mentioned in the previous section, using a NUMA architecture may positively influence the performance of an application if the application is NUMA-aware.

Physical Server

First introduced by AMD in its Opteron processor series and then by Intel in its Nehalem processor family in 2008, NUMA support is dependent on the CPU architecture. Today, nearly all server hardware currently available on the market uses NUMA architecture and is typically enabled by default in the BIOS of a server.

However, it's recommended to check BIOS settings to ensure they've not been modified.

Most hardware vendors will call these settings node *interleaving* (HPE, Dell) or *socket interleave* (IBM). However, it is named, this feature should be set to **disabled** or **non-uniform memory access**³ to expose the NUMA topology and get the best performance (i.e., leaving NUMA activated).

Recommendation: As a best practice, leave NUMA enabled in the BIOS for NUMA servers to be NUMA aware.

As a general rule, the number of exposed NUMA nodes will be equal to the number of physical sockets for Intel processors⁴ and will be 2x for the AMD processors. Check your server documentation for more details.

For further information, review:

- [Node Interleaving: Enable or Disable?](#)

VMware ESXi and ESXTOP

vSphere supports NUMA on the physical server starting with Version 2. Moving to the current version (vSphere 7.0 as of this writing), many configuration settings have been introduced to assist in managing a NUMA topology.

As our goal is to provide clear guidance for exposing NUMA topology to a VM hosting a BCA workload, we will skip discussion of advanced settings and will instead concentrate on the examples and relevant configuration required.

The first step toward achieving this goal is to ensure that the physical NUMA topology is exposed correctly to an ESXi host. Use `esxtop` or `esxcli` and `sched-stats` to obtain this information:

`esxcli hardware memory get | grep NUMA`

```
[root@localhost:~] esxcli hardware memory get | grep NUMA
NUMA Node Count: 2
```

`sched-stats -t ncpus`

³ Refer to your server hardware documentation for more details. The name and value of the setting could appear or be changed differently depending on the particular BIOS/UEFI implementation.

⁴ If the snooping mode **cluster-on-die** (CoD, Haswell) or **sub-NUMA cluster** (SNC, Skylake) is used with a pCPU with more than 10 cores, each pCPU will be exposed as two logical NUMA nodes. See [Intel® Xeon® Processor Scalable Family Technical Overview](#) and [Restarting Services for the Service Console to Resolve Connectivity Problems](#).

```
[root@localhost:~] sched-stats -t ncpus
56 PCPUs
28 cores
2 packages
2 NUMA nodes
```

FIGURE 5. Using esxcli and shed-stats Commands to Obtain the NUMA Node Count on an ESXi Host

Alternatively, use **esxtop** to obtain NUMA details:

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

```
5:39:11am up 1 day 14:54, 1215 worlds, 12 VMs, 56 vCPUs; MEM overcommit avg: 0.03, 0.03, 0.03
PMEM /MB: 392992 total: 4141 vmk,214832 other, 174018 free
VMKMEM/MB: 392606 managed: 4540 minfree, 56969 rsvd, 335637 ursvd, high state
NUMA /MB: 196382 (59001), 196608 (114632)
PSHARE/MB: 134797 shared, 449 common: 134348 saving
SWAP /MB: 15882 curr, 15526 rclmtgt: 0.00 r/s, 0.00 w/s
ZIP /MB: 5632 zipped, 3674 saved
MEMCTL/MB: 0 curr, 0 target, 157734 max
```

GID	NAME	NHN	NMIG	NRMEM	NLMEM	N%L	GST	ND0	OVD
40168	hana20_2	0	0	1026.53	100958.17	98	100958.17		
111067	HANA_Primary	1	0	256.29	94693.56	99	256.29	1	
38315	SAP_Windows_cln	0	0	0.00	26865.12	100	26865.12		
560651	prdrac01	1	0	0.00	1038.00	99	0.00		
40200	vcsa-67-sitea	0	0	0.00	24564.68	100	24564.68		
87162	Deji-UM01	1	0	0.00	24186.40	100	0.00		
83521	VMware-vR-Appli	0	0	0.00	18408.50	100	18408.50		
38299	DA-VMC-EXC-CL02	1	0	4.00	15328.04	99	4.00		
75240	DA-VMC-EXC-MB02	1	0	0.06	16375.39	99	0.06		
42579	TSA-65-NSXMgr-a	0	0	258.00	12876.00	98	12876.00		
38248	erphana_NFS_ser	0	0	0.00	4379.76	100	4379.76		
38264	TSALAB-DC01	1	0	0.00	4091.64	100	0.00		

FIGURE 6. Using esxtop to Obtain NUMA-related Information on an ESXi Host

The following table lists ESXTOP metrics to consider. NMIG (i.e., the number of NUMA migrations between two snapshots) is the key metric to examine for NUMA imbalances.

METRIC	EXPLANATION
NHN	Current home node for VM
NMIG	Number of NUMA migrations between two snapshots. It includes balance migration, inter-mode VM swaps performed for locality balancing, and load balancing.
NRMEM (MB)	Current amount of remote memory being accessed by a VM
NLMEM (MB)	Current amount of local memory being accessed by a VM
N%L	Current percentage memory being accessed by a VM that is local
GST_NDx (MB)	The guest memory allocated for a VM on NUMA node X, with X representing the node number
OVD_NDx (MB)	The VMM overhead memory allocated for a VM on NUMA node X

TABLE 1. ESXTOP NUMA Metrics

Recommendation: As a best practice, ensure that the physical NUMA topology is exposed correctly to an ESXi host before proceeding with configuring BCA workloads on the vSphere cluster.

For further information, review:

- [Intel Cluster-on-Die \(COD\) Technology, and VMware vSphere 5.5 U3b and 6.x \(2142499\)](#)
- [Performance Monitoring Utilities: resxtop and esxtop](#)
- [Using esxtop to Troubleshoot Performance Problems](#)
- [Using esxtop to identify storage performance issues for ESX / ESXi \(multiple versions\) \(1008205\)](#)
- [NUMA Deep Dive Part 1: From UMA to NUMA](#)
- [NUMA Deep Dive Part 5: ESXi VMkernel NUMA Constructs](#)
- [Optimizing Application Performance in Large Multi-core Systems](#)
- [Using NUMA Systems with ESXi](#)
- [NUMA Blogs](#)
- [VMware Communities: Interpreting esxtop Statistics](#)
- [NUMA Observer - VMware Fling](#)

ESXi Memory Configuration

Memory Overcommit Techniques

ESXi uses five memory management mechanisms to dynamically reduce the amount of machine physical memory required for each VM. These are page-sharing, ballooning, memory compression, swap-to-host cache, and regular swapping.

- **Page-Sharing:** ESXi can use a proprietary technique to transparently share memory pages between VMs, thus eliminating redundant copies of memory pages. While pages are shared by default *within* VMs, as of vSphere 6.0, pages are not shared by default *between* VMs for security reasons. In most environments, this change should have little effect.
- **Ballooning:** The ESXi hypervisor is not aware of the guest OS memory management tables of used and free memory. When the VM is asking for memory from the hypervisor, the ESXi will assign a physical memory page to accommodate that request. When the guest OS stops using that page, it will release it by writing it in the OS's free memory table but will not delete the actual data from the page. The ESXi hypervisor does not have access to the OS's free and used tables, and from the hypervisor's point of view, that memory page might still be in use. In case there is memory pressure on the hypervisor host, and the hypervisor requires reclaiming some memory from VMs, it will utilize the balloon driver. The balloon driver, which is installed with VMware Tools, 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. The prerequisite is VMware Tools must be installed on the guest, status of the tool service must be running, and balloon driver must not be disabled. 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*.
- **Memory Compression:** If the VM's memory usage approaches the level at which host-level swapping will be required, ESXi will use memory compression to reduce the number of memory pages it will need to swap out. Because the decompression latency is much smaller than the swap-in latency, compressing memory pages has significantly less impact on performance than swapping out those pages.
- **Swap-to-Host Cache:** If memory compression doesn't keep the VM's memory usage low enough, ESXi will next forcibly reclaim memory using host-level swapping to a host cache (if one has been configured). Swap to host cache is a feature that allows users to configure a special swap cache on SSD storage. In most cases, this host cache (residing on SSD) will be much faster than the regular swap files (typically on hard disk storage), significantly reducing access latency. Thus, although some of the pages ESXi swaps out might be active, swap-to-host cache has a far lower performance impact than regular host-level swapping.
- **Regular Host-Level Swapping:** If the host cache becomes full, or if a host cache has not been configured, ESXi will next reclaim memory from the VM by swapping out pages to a regular swap file. Like swap-to-host cache, some of the pages ESXi swaps out might be active. Unlike swap-to-host cache, however, this mechanism can cause VM performance to degrade significantly due to its high access latency (note that this swapping is transparent to the guest OS and is distinct from the swapping that can occur within the VM under the control of the guest OS).

When designing BCA workloads 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. It's highly recommended to implement monitoring of the ballooned memory at both the host and VM level. Use **ballooned memory** counter in the vCenter web client to configure an alarm or use special tools like VMware vRealize® Operation Manager™.

Recommendation: As a best practice, leave the balloon driver enabled for corner cases where it might be needed to prevent loss of service.

For further information, review:

- [Understanding Memory Resource Management in VMware vSphere](#)
- [Administering Memory Resources](#)
- [Memory Balloon Driver](#)

vSphere 2M Large Memory Pages

In addition to the usual 4KB memory pages, ESXi also provides 2MB memory pages (commonly referred to as *large pages*). ESXi assigns these 2MB machine memory pages to guest operating systems whenever possible; it does this even if the guest OS doesn't request them (though the full benefit of large pages comes only when the guest OS and applications use them as well).

The use of large pages can significantly reduce translation lookaside buffer (TLB) misses, improving the performance of most workloads, especially those with large active-memory working sets. In addition, large pages can slightly reduce per-VM memory space overhead.

Use of large pages can also change page-sharing behavior. While ESXi ordinarily uses page-sharing regardless of memory demands, ESXi does not share large pages. Therefore, with large pages, page-sharing might not occur until memory overcommitment is high enough to require the large pages to be broken into small pages.

For further information, review:

- [Transparent Page Sharing \(TPS\) in hardware MMU systems \(1021095\)](#)
- [Use of large pages can cause memory to be fully allocated \(1021896\)](#)
- [Support for Large Page Sizes](#)

1GB Large Memory Pages

Applications with large memory footprints, like SAP HANA, Oracle, or SQL Server, can often stress the hardware memory subsystem (i.e., TLB) with their access patterns. Modern processors can mitigate this performance impact by creating larger mappings to memory and increasing the memory reach of the application.

In prior releases, ESXi allowed guest OS memory mappings based on 2MB page sizes.

Beginning with version 6.7, vSphere introduces memory mappings for 1GB page sizes. The support is limited for backing guest vRAM with 1GB pages.

There are some requirements and limitations to using 1GB large memory page:

- In order to use 1GB pages for backing guest memory, set VM option `sched.mem.lpage.enable1GPage = "TRUE"`
- Full memory reservation for the VM is required.
- 1GB page vRAM backing is opportunistic and 1GB pages are allocated on a best effort basis. Hence, start VMs requiring 1GB pages on a freshly booted host because over time the host RAM is fragmented.
- A VM with 1GB pages enabled can be migrated to a different host. However, the 1GB page size might not be allocated on the destination host in the same amount as it was on the source host. You might also see that part of vRAM backed with a 1GB page on the source host is no longer backed with a 1GB page on the destination host.
- The opportunistic nature of 1GB pages extends to vSphere services such as HA and DRS that might not preserve 1GB page vRAM backing. These services are not aware of the 1GB capabilities of destination hosts and do not take 1GB memory-backing into account while making placement decisions.

Recommendation: As a best practice, use 1GB large memory pages on ESXi servers with adequate memory capacity and on any Guest OS with 1GB large memory page support.

For further information, review:

- [Backing Guest vRAM with 1GB Pages](#)
- [1 GB Large Memory Pages](#)

Persistent Memory

First introduced in vSphere 6.7, persistent memory (PMem), also known as non-volatile memory (NVM), can maintain data in memory DIMM even after a power outage. This technology layer provides the performance of memory with the persistence of traditional storage.

Support of PMem can be combined with native PMem support in newer versions of Linux and Windows operating systems, increasing performance of high-loaded databases.

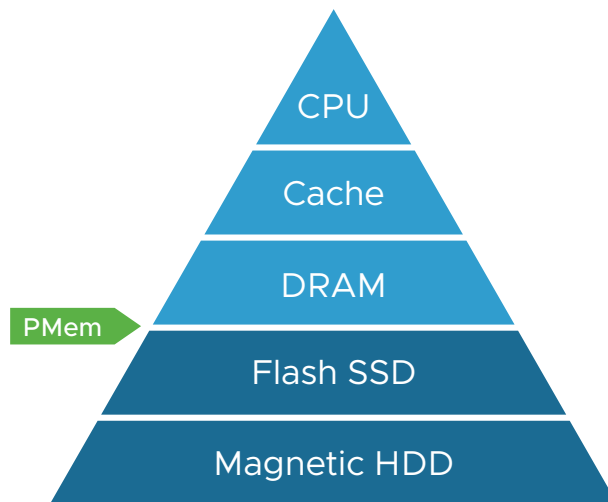


FIGURE 7. Positioning PMem

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

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

vPMemDisk mode can be used with any version of Linux or Windows as a traditional block-based storage device, but with very low latency. Recent use cases have demonstrated the benefits of vPMemDisk for BCA workload backup and restore. However, this use case requires VM hardware Version 14 and a guest OS that supports NVM technology.

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

The ESXi **memstats** tool can be used to show the DRAM cache miss rate and PMem bandwidth:

```
# memstats -r dcpmm-stats
# python memstats_postprocess.py dcpmm-stats-20190702025557
```

(Where *dcpmm-stats-20190702025557* is the output file from the `memstats -r dcpmm-stats` command).

PMem can also be preallocated as part of VM startup using `sched.pmem.prealloc`, as described in VMware KB article [78094](#). This prevents allocation overhead when the VM first uses a PMem page.

Though each DIMM slot that contains PMem is local to a NUMA node, ESXi 7.0 Update 2 doesn't automatically associate PMem with its NUMA node. The association can be manually configured, however, as follows:

1. Run the following command:

```
memstats -r pmem-extent-stats -s numaNode|grep -w "[0-9]$" | sort -n | awk 'BEGIN {nvd=0;} {printf "nvdimm0:%d.node = \"%d\\n\", nvd, $0; nvd++;}'
```

2. Using the output of the above command, configure the `nvdimm0:<devNum>.nodeAffinity` parameters, as described in VMware KB article [78094](#).

Recommendation: As a best practice, consider using persistent memory technology to accelerate BCA workloads.

For further information, review:

- [VM nvdimm config options for NUMA \(78094\)](#)
- [Persistent Memory](#)
- [Accelerating applications performance with virtualized Persistent Memory](#)
- [Persistent Memory Performance in vSphere 6.7](#)
- [Persistent Memory Performance on vSphere 6.7 Performance Study](#)
- [Persistent Memory Performance in vSphere 6.7 with Intel Optane DC persistent memory Performance Study](#)
- [Announcing VMware vSphere Support for Intel® Optane™ Persistent Memory Technology](#)
- [What Is Persistent Memory?](#)
- [Virtualized Persistent Memory with VMware vSphere 6.7 and HPE ProLiant Gen10 Servers](#)
- [DELL EMC PowerEdge Persistent Memory \(PMem\) Support \(54444\)](#)
- [Hewlett Packard Enterprise Servers Persistent Memory \(PMem\) Support \(54445\)](#)
- [How to simulate Persistent Memory \(PMem\) in vSphere 6.7 for educational purposes?](#)
- [vSphere Support for Intel's Optane Persistent Memory \(PMEM\) \(67645\)](#)

ESXi Storage Configuration

vSphere Storage Options

vSphere provides several options for storage configuration. The most widely used of these is a VMware virtual machine file system (VMFS) formatted datastore on block storage system, but this is not the only option available. Today, storage admins can utilize new technologies such as vSphere Virtual Volumes™ which takes storage management to the next level, in which VMs are native objects on storage systems.

Other options include hyper-converged solutions, such as VMware vSAN. This section covers the different storage options that exist for virtualized BCA workload deployments running on vSphere.

Virtual Machine File System (VMFS) Datastores Considerations

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

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

Consider using the highest possible VMFS version supported by ESXi hosts in the environment.

NOTE: Strictly avoid placing a VM hosting a BCA workload on VMFS3 or VMFS3 upgraded datastores, as doing so negatively affects disk performance.

VMFS remains the most widely used option 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 a physical deployment. The storage array LUNs are then formatted as VMFS volumes by the ESXi hypervisor, on which the virtual disks reside. These virtual disks are then presented to the guest OS, which can be partitioned and used in file systems or as a block.

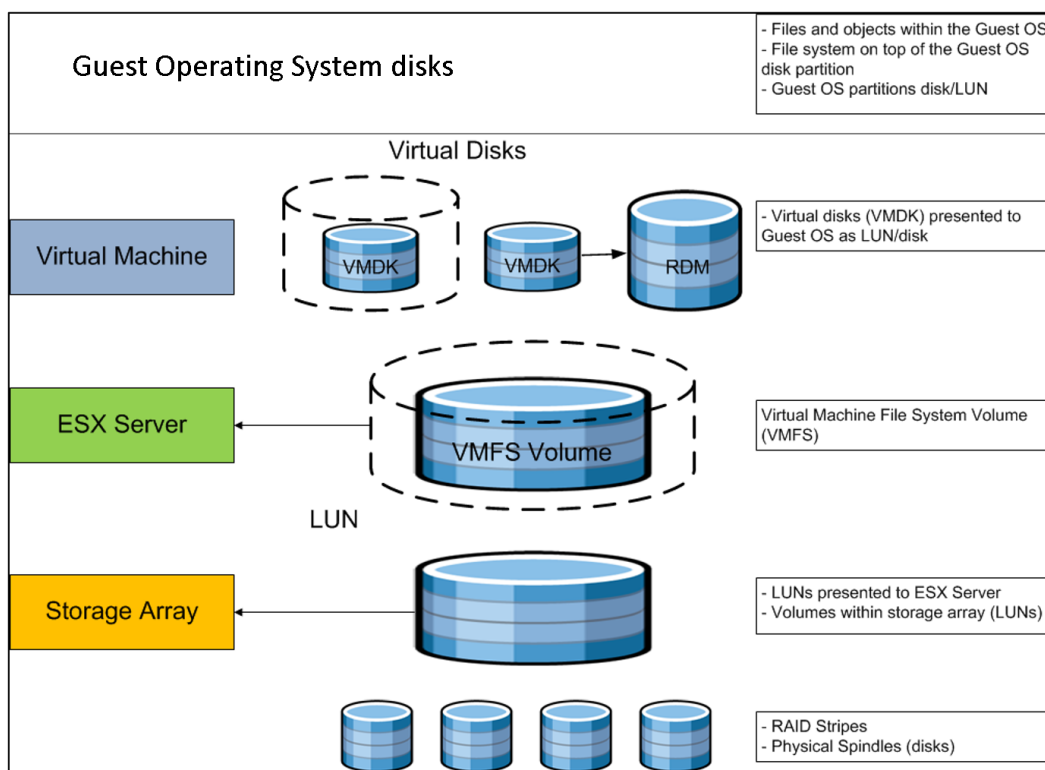


FIGURE 8. VMware Storage Virtualization Stack

Using the vSphere client to create VMFS partitions avoids this problem because, beginning with version 5.0, ESXi automatically aligns VMFS3, VMFS5, or VMFS6 partitions along the 1MB boundary.

For example, VMFS6 FC datastore has a partition alignment of 1MB assigned automatically when it was created using vSphere client.

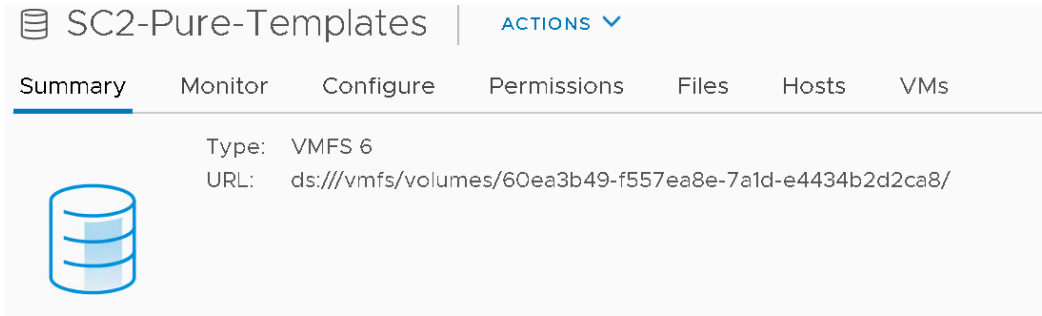


FIGURE 9. VMFS6 Datastore Example

An existing partition table on a block disk device may be examined using the `partedUtil` command-line utility as shown below:

```
[root@sc2esx09:~] partedUtil getptbl /vmfs/devices/disks/naa.624a9370a841b405a3a348ca0001592c
gpt
2673493 255 63 42949672960
1 2048 42949672926 AA31E02A400F11DB9590000C2911D1B8 vmfs 0
[root@sc2esx09:~]
```

The starting sector for the above partition is 2048 sectors which is 1MB.

For further information, review:

- [Using partedUtil command line disk partitioning utility on ESXi \(1036609\)](#)
- [Understanding VMFS Datastores](#)

Network File System (NFS) Datastores Considerations

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

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

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

For further information, review:

- [Increasing the default value that defines the maximum number of NFS mounts on an ESXi/ESX host \(2239\)](#)
- [Understanding Network File System Datastores](#)
- [Best Practices for running VMware vSphere on Network Attached Storage](#)
- [Best Practices for Running VMware vSphere® on Network-Attached Storage \(NAS\) \(2013\)](#)
- [vSphere 7.0 Configuration Limits](#)

Raw Device Mapping

Raw device mapping (RDM) allows a VM to directly access a volume on a physical storage subsystem without formatting it with VMFS. RDMs can only be used with block storage (Fiber Channel or iSCSI, FCoE). 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.

From a performance perspective, both VMFS and RDM volumes can provide similar transaction throughput. The following chart summarizes some performance testing:

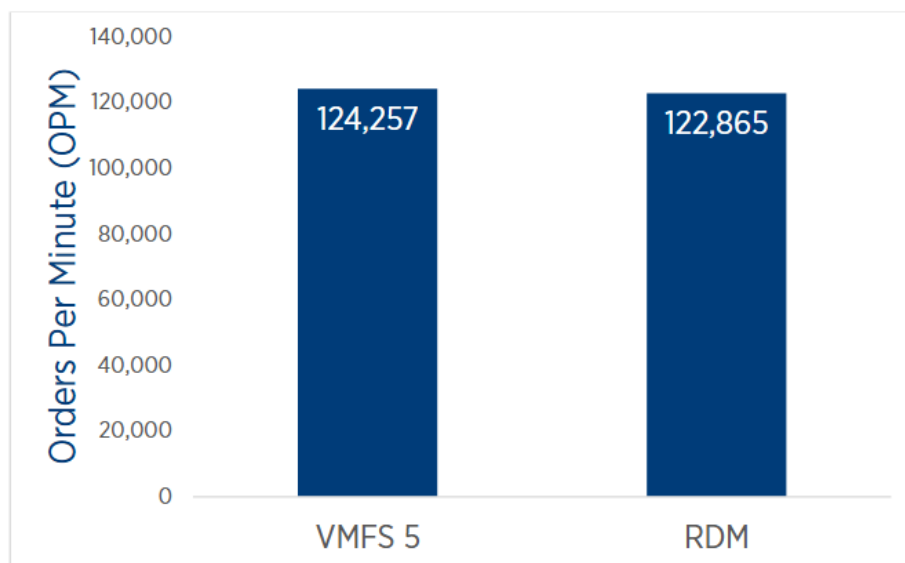


FIGURE 10. VMFS vs. RDM: DVD Store 3 Performance Comparison

Raw Device Mapping (RDM) and VMFS Trade-Off

The decision to place BCA workloads on VMFS as opposed to RDM is no longer related to performance requirements.

VMFS has been proven to provide, and in certain cases exceed, native performance. The results for VMFS, RDM (virtual), and RDM (physical) are all nearly identical in their results.

VMware generally recommends VMFS, however there might be situations in which RDMs are required. The following table summarizes some of the options and tradeoffs in choosing between VMFS and RDM.

VMFS	RDM
<ul style="list-style-type: none"> • Volume can host many VMs (or can be dedicated to one VM). • Increases storage utilization, provides better flexibility, easier administration, and management. • Can potentially support clustering software that does not issue SCSI reservations, such as Oracle Clusterware. To configure, follow the procedures given in Disabling simultaneous write protection provided by VMFS using the multi-writer flag. • Oracle RAC node live migration • Can support WSFC with shared disks starting with vSphere 7.x. Limitations apply. 	<ul style="list-style-type: none"> • Maps a single LUN to one VM, so only one VM is possible per LUN. • More LUNs are required, so it is easier to reach the current LUN limit (for the vSphere specific version) that can be presented to an ESXi host. • RDM might be required to leverage third-party storage array-level backup and replication tools. • RDM volumes can help facilitate migrating physical Oracle databases to VMs. Alternatively, enables quick migration to physical in rare Oracle support cases.

TABLE 2. VMFS and Raw Disk Mapping Trade-Offs

NOTE: Other than when explicitly requested by backup solution vendors, the use case for RDM on Windows VMs is no longer compelling on vSphere 7.x.

For further information, review:

- [Performance Characterization of VMFS and RDM Using a SAN](#)
- [Migrating virtual machines with Raw Device Mappings \(RDMs\) \(1005241\)](#)
- [Difference between Physical compatibility RDMs and Virtual compatibility RDMs \(2009226\)](#)
- [vSphere 7 Microsoft Windows Server Failover Cluster with Clustered VMDKs](#)

vSphere Virtual Volumes

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

- Flexible consumption at the logical level – vVols virtualizes SAN and NAS devices by abstracting physical hardware resources into logical pools of capacity (represented as virtual datastore in vSphere).
- Finer control at the VM level – vVols defines a new virtual disk container (the virtual volume) that is independent of the underlying physical storage representation (e.g., LUN, file system, object). 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 replication to individual VMs. This allows admins to provide the correct storage service levels to each individual VM.
- Ability to configure different storage policies for different VMs using storage policy-based management (SPBM). These policies instantiate themselves on the physical storage system, enabling VM-level granularity for performance and other data services.

- SPBM allows capturing storage service levels requirements (e.g., capacity, performance, availability) in the form of logical templates (i.e., policies) to which VMs are associated. SPBM automates VM placement by identifying available datastores that meet policy requirements and, coupled with vVols, dynamically instantiates necessary data services. Through policy enforcement, SPBM also automates service-level monitoring and compliance throughout the lifecycle of the VM.
- Array-based replication starting from vSphere Virtual Volumes 2.0 (vSphere 6.5)
- Support for SCSI-3 persistent reservation starting from vSphere 6.7. A vVols disk could be used instead of an RDM disk to provide a disk resource in a WSFC.



FIGURE 11. VMware vSphere Virtual Volumes

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

vVols 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, 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 modern databases makes it harder to clone and refresh data from production to other environments.
- Databases of different levels of criticality need different storage performance characteristics and capabilities.

When virtualizing BCA workloads on a SAN using vVols as the underlying technology, the best practices and guidelines remain the same as when using a VMFS datastore.

Make sure that the physical storage on which the VM's virtual disks reside can accommodate the requirements of the BCA workloads implementation about RAID, I/O, latency, and queue depth, as detailed in the storage best practices contained in this document.

For further information, review:

- [vVols Concepts](#)
- [What's New vSphere Virtual Volumes](#)

VMware vSAN

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

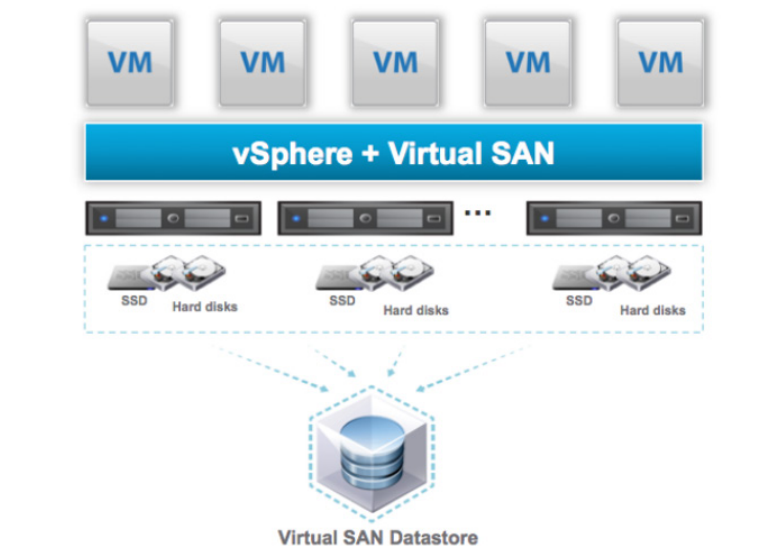


FIGURE 12. VMware vSAN

vSAN can be configured as a hybrid or 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 (e.g., 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.

For further information, review:

- [VMware vSAN Documentation](#)

VMware Virtual Disk Provisioning Policies

When performing certain VM management operations, it's possible to specify a provisioning policy for the virtual disk file. The operations include creating a virtual disk, cloning a VM to a template, or migrating a VM with VMware vSphere Storage vMotion.

You can also use VMware vSphere Storage vMotion or cross-host vSphere Storage vMotion to transform virtual disks from one format to another.

OPTION	DESCRIPTION
Thick Provision Lazy Zeroed	Creates a virtual disk in a default thick format. Space required for the virtual disk is allocated when the disk is created. Data remaining on the physical device is not erased during creation but is zeroed out on demand later on first write from the VM. VMs do not read stale data from the physical device.
Thick Provision Eager Zeroed (EZT)	A type of thick virtual disk that supports clustering features such as the multi-writer attribute for application clustering (e.g., RAC). Space required for the virtual disk is allocated at creation time. In contrast to the thick provision lazy zeroed format, the data remaining on the physical device is zeroed out when the virtual disk is created. Creating virtual disks in this format may take longer than creation of other types of disks. Increasing the size of an eager zeroed thick virtual disk causes a significant stall time for the VM.
Thin Provision	Use this format to save storage space. For the thin disk, provision as much datastore space as the disk would require based on the value entered for the virtual disk size. The thin disk starts small and, at first, uses only as much datastore space as the disk needs for its initial operations. If the thin disk needs more space later, it can grow to its maximum capacity and occupy the entire datastore space provisioned to it. Thin provisioning is the fastest method to create a virtual disk because it creates a disk with only the header information. It does not allocate or zero out storage blocks. Storage blocks are allocated and zeroed out when they are first accessed.

TABLE 3. Virtual Disk Formats Available in vSphere Storage vMotion

NOTE: Most modern all-flash arrays do not perform zeroing even when you opt to create an eager zero thick disk on them. The request is satisfied simply by updating metadata information to reflect that such a configuration is requested. This update is sufficient for the checks performed by ESXi to consider a disk to be eager provisioned.

VMDK modes are shown in the table below:

OPTION	DESCRIPTION
Dependent	Dependent disks are included in snapshots.
Independent-persistent	Disks in persistent mode behave like conventional disks on a physical computer. All data written to a disk in persistent mode is written permanently to the disk.
Independent- non-persistent	Changes to disks in non-persistent mode are discarded when the VM is turned off or reset. Non-persistent mode enables restarting of the VM with a virtual disk in the same state every time. Changes to the disk are written to and read from a redo log file that is deleted when the VM is turned off or reset.

TABLE 4. VMDK Modes

For further information, review:

- [VMware virtual disk provisioning policies](#)
- [Determining if a VMDK is zeroed thick or eager zeroed thick \(1011170\)](#)
- [Cloning and converting virtual machine disks with vmkfstools \(1028042\)](#)
- Pure Storage [Virtual Machine and Guest Configuration](#)
- Pure Storage [ZeroedThick or Eagerzeroedthick? That is the question](#)

VMware vSAN Storage Policy

vSAN requires VMs deployed on vSAN datastores be assigned at least one storage policy. When provisioning a VM, if you do not explicitly assign a storage policy to the VM, the vSAN default storage policy is assigned.

The default policy contains vSAN rule sets and a range of basic storage capabilities typically used for the placement of VMs deployed on vSAN datastores.

VM Storage Policies

CREATE EDIT CLONE CHECK REAPPLY RESET

<input type="checkbox"/>	Name	VC
<input type="checkbox"/>	vVOL Replication Policy	sc2wvc03.vslab.local
<input type="checkbox"/>	Performanceflash	sc2wvc03.vslab.local
<input type="checkbox"/>	Host-local PMem Default Storage Policy	sc2wvc03.vslab.local
<input checked="" type="checkbox"/>	vSAN Default Storage Policy	sc2wvc03.vslab.local
<input type="checkbox"/>	Management Storage Policy - Regular	sc2wvc03.vslab.local
<input type="checkbox"/>	Management Storage Policy - Stretched	sc2wvc03.vslab.local

☒ 1

Rules VM Compliance VM Template Storage Compatibility

General

Name	vSAN Default Storage Policy
Description	Storage policy used as default for vSAN datastores

Rule-set 1: VSAN

Placement

Storage Type	VSAN
Site disaster tolerance	None - standard cluster
Failures to tolerate	1 failure - RAID-1 (Mirroring)
Number of disk stripes per object	1
IOPS limit for object	0
Object space reservation	Thin provisioning
Flash read cache reservation	0%
Disable object checksum	No
Force provisioning	No
Encryption services	No preference
Space efficiency	No preference
Storage tier	No preference

FIGURE 13. Default vSAN Storage Policy

Key storage policy rules:

STORAGE POLICY	DESCRIPTION
Number of failures to tolerate	Defines the number of host, disk, or network failures a VM object can tolerate. For n failures tolerated, $n+1$ copies of the VM object are created and $2n+1$ hosts with storage are required. The settings applied to the VMs on the vSAN datastore determines the datastore's usable capacity.
Object space reservation	Percentage of the object logical size that should be reserved during the object creation. The default value is 0 percent, and the maximum value is 100 percent.
Number of disk stripes per object	This policy defines how many physical disks across each copy of a storage object are striped. The default value is 1 and the maximum value is 12.
Flash read cache reservation	Flash capacity reserved as read cache for the VM object. Specified as a percentage of the logical size of the VMDK object. It is set to 0 percent by default and vSAN dynamically allocates read cache to storage objects on demand.

TABLE 5. Key Storage Policy Rules

Object Space Reservation (OSR) – an administrator should always be aware of over-committing storage on vSAN, just as one needs to monitor over-commitment on a traditional SAN or NAS array.

By default, VM storage objects deployed on vSAN are **thinly provisioned**. This capability, *ObjectSpaceReservation*, specifies the percentage of the logical size of the storage object that should be reserved (thick provisioned) when the VM is being provisioned. The rest of the storage object will remain thin provisioned. The default value is 0 percent, implying the object is deployed as thin. The maximum value is 100 percent, meaning the space for the object is fully reserved, which can be thought of as full, thick provisioned. Since the default is 0 percent, all VMs deployed on vSAN are provisioned as thin disks unless one explicitly states a requirement for *ObjectSpaceReservation* in the policy. If *ObjectSpaceReservation* is specified, a portion of the storage object associated with that policy is reserved.

There is no eager-zeroed thick format on vSAN. OSR, when used, behaves similarly to lazy-zeroed thick.

For further information, review:

- [vSAN Default Storage Policy](#)
- [VMware vSAN Design Guide](#)

VMware Multi-Writer Attribute for Shared VMDKs

VMware vSphere VMFS is a clustered file system that disables (by default) multiple VMs from opening and writing to the same virtual disk (.vmdk file). This prevents more than one VM from inadvertently accessing the same .vmdk file. The multi-writer option allows VMFS-backed disks to be shared by multiple VMs.

As occurs with VMFS, vVols (beginning with ESXi 6.5), and NFS datastores, VMware vSAN also prevents multiple VMs from opening the same VMDK in read-write mode.

Some of the current restrictions of the multi-writer attribute for VMware non-vSAN and VMware vSAN are:

- Storage vMotion is disallowed
- Snapshots not supported (snapshots of VMs with independent-persistent disks are supported, however)
- Changed block tracking (CBT) not supported
- Cloning, hot extend virtual disk not supported

When using the multi-writer mode, the virtual disk must be eager zeroed thick (EZT). It cannot be zeroed thick or thin provisioned. Independent persistent mode is not required for enabling the multi-writer attribute.

For all flash arrays (e.g., Pure Storage AFF x50), shared VMDKs on vVols datastores has to be thin-provisioned with the multi-writer attribute.

In the case of VMware vSAN:

- Prior to vSAN 6.7 Patch P01, the virtual disk must be EZT to enable multi-writer mode.
- Beginning with VMware vSAN 6.7 Patch P01 (ESXi 6.7 Patch Release ESXi670-201912001), applications with shared VMDKs with multi-writer attribute will **NOT** require shared VMDKs on vSAN to be EZT (OSR=100) for multi-writer mode to be enabled.

NOTE: Do not enable the **multi-writer** flag on disks using Microsoft's Windows Failover Clustering Service (WSFC).

For further information, review:

- [Enabling or disabling simultaneous write protection provided by VMFS using the multi-writer flag \(1034165\)](#)
- [Using Oracle RAC on a vSphere 6.x vSAN Datastore \(2121181\)](#)
- [Attempts to enable the multi-writer virtual disk option on an NFS datastore fail \(2147691\)](#)

Clustered VMDK Support

vSphere 7.0 introduces support for the use of VMDKs on clustered datastores as shared disk resources for a WSFC. Using VMDKs reduces the extra overhead to manage the virtual disks compared to pRDMs.

Microsoft Clustering Service uses SCSI-3 PRs commands to coordinate access to a clustered disk resource. These commands (PR-IN and PR-Out) are emulated at the VSCSI layer on a datastore. The feature requires support from the datastore perspective. A datastore configured to host clustered VMDKs is referred to as a clustered VMDK datastore in this document.

You can enable clustered VMDK support when you create a new VMFS6 datastore or enable clustered VMDK support on an existing VMFS6 datastore. Before enabling clustered VMDK support, ensure all hosts connected to the datastore are using ESXi 7.0 or later and are managed by vCenter 7.0 or later. All hosts connected to the datastore must be managed by the same vCenter.

For further information, review:

- [Clustered VMDK support](#)
- [Hosting Windows Server Failover Cluster \(WSFC\) with shared disks on VMware vSphere: Doing it right!](#)

vSphere APIs for Array Integration (VAAI)

vSphere APIs for Array Integration (VAAI) is a feature introduced in ESXi and ESX 4.1 that provides hardware acceleration functionality. It enables your host to offload specific VM and storage-management operations to compliant storage hardware. With storage hardware assistance, your host performs these operations faster and consumes less CPU, memory, and storage fabric bandwidth.

These APIs, also known as VAAI, include the following components:

- Hardware Acceleration APIs – Help arrays to integrate with vSphere so that vSphere can offload certain storage operations to the array. This integration significantly reduces CPU overhead on the host.
- Array Thin Provisioning APIs – Help to monitor space use on thin-provisioned storage arrays to prevent out-of-space conditions and to perform space reclamation.

Recommendation: As a best practice, ensure that the storage array has VAAI capability to accelerate the above storage operations.

For further information, review:

- [Frequently Asked Questions for vStorage APIs for Array Integration \(1021976\)](#)
- [Storage Hardware Acceleration](#)
- [VMware vSphere Storage APIs Array Integration \(VAAI\)](#)
- [VAAI Comparison – Block versus NAS](#)

Storage Policy-Based Management (SPBM)

Within a software-defined data center (SDDC), storage policy-based management (SPBM) plays a significant role, helping to align storage with the application demands of your VMs. SPBM provides a storage policy framework that serves as a single, unified control panel across a broad range of data services and storage solutions.

As an abstraction layer, SPBM abstracts storage services delivered by vVols, vSAN, I/O filters, or other storage entities.

Rather than integrating with each individual type of storage and data services, SPBM provides a universal framework for different types of storage.

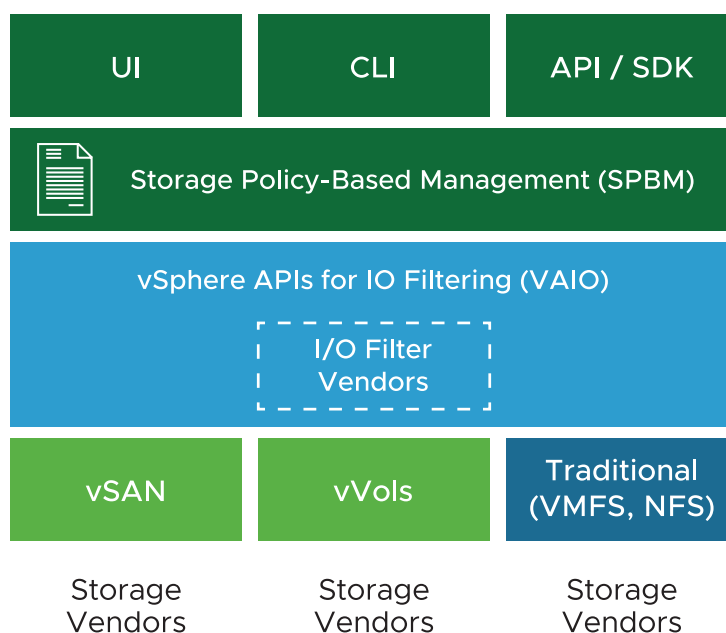


FIGURE 14. Storage Policy-Based Management (SPBM)

SPBM offers the following mechanisms:

- Promotion of storage capabilities and data services that storage arrays and other entities, such as I/O filters, offer
- Bidirectional communications between ESXi and vCenter Server on one side with storage arrays and entities on the other
- VM provisioning based on VM storage policies

For further information, review:

- [Storage Policy-Based Management](#)

Automatic Space Reclamation (UNMAP)

Beginning with vSphere 6.5, VMFS6 datastores can perform automatic space reclamation, **UNMAP**, for thin-provisioned disks. vSphere 6.5 allowed the UNMAP priority to be set.

vSphere 6.7 added the ability to set the UNMAP rate. Using these options, you can reclaim storage space much more quickly on fast arrays (especially useful for all-flash arrays) or limit the potential performance impact of performing an unthrottled burst of UNMAP operations (especially useful for slower arrays).

vSphere 6.5 revised the VMFS file system to VMFS6. VMFS6 is now 4K-aligned to support newer advanced format (AF) large-capacity drives in the future, after VMware certifies them.

vSphere 6.5 also supports 512 emulation (512e) mode for backward compatibility.

Recommendation: As a best practice, upgrade current VMFS datastores to VMFS6 to take advantage of the UNMAP feature.

For further information, review:

- [Storage Space Reclamation](#)
- [WHAT'S NEW IN PERFORMANCE VMware vSphere 6.5](#)

Advanced Format Drive and 4kn Support

vSphere 6.7 released **4kn local storage support**. Like the firmware in 512e drives, this software emulation layer exposes 512B sectors to the guest OS while using 4K native (4Kn) drives as local storage in a server.

This approach enables running legacy operating systems, applications, and existing VMs on servers with 4Kn HDD drives. This support also offers the flexibility to use new 4Kn hardware on your vSphere 6.7 host, but still run older applications in the host's VMs.

There was no perceivable difference in performance in random I/Os, reads or writes, irrespective of the percentage of 4KB alignments in the I/O blocks. This test was focused on random I/O because, generally, an ESXi host with many VMs requires the storage to be optimized for a random I/O stream.

If your storage subsystem uses 4KB native (4Kn) or 512B emulation (512e) drives, you can obtain the best storage performance if your workload issues mostly 4K-aligned I/Os.

vSphere 6.5/6.7/7.x and vSAN 6.5/6.7/7.x and later support 512e drives as direct attached drives. **vSphere and vSAN will expose 512n to the guest OS, even if the backend physical drive is 512e/4kn.**

Recommendation: As a best practice, consider 4kn local storage for capacity reasons. Consider enforcing 4K I/O alignment at both the application and guest OS layer if the underlying VMware storage devices are 512e/4kn. This is especially important for database transaction logfile I/Os on VMware vSAN.

For further information, review:

- [FAQ: Support statement for 512e and 4K Native drives for VMware vSphere and vSAN \(2091600\)](#)
- [Support for 4Kn HDDs](#)
- [What's New in Performance for VMware vSphere 6.7?](#)
- [Device Sector Formats](#)

Storage Other Features

The release of vSphere 6.x introduced many other enhancements, including:

- vSphere 6.5 revises the VMFS file system to VMFS6, with performance improvements that include faster file creation, device discovery, and device rescanning.
- VMFS6 is now 4K-aligned to support newer AF large-capacity drives. vSphere 6.5 also supports 512 emulation (512e) mode for backward compatibility.
- Software iSCSI Static Routing Support – There have been limitations in the past when using software iSCSI when the iSCSI initiator and target had to be on the same subnet. With vSphere 6.5, the software iSCSI initiator and target can be on different subnets. Static routes can be configured to route between the initiator and target subnets. vSphere 6.5 makes it easy to configure multipathing without requiring that the initiator and target be in the same network.

The release of vSphere 7.0 introduced many other enhancements, including:

- Beginning with version 7.0, ESXi supports NVMe over Fabrics (NVMe-oF), over both Fibre Channel (FC) and remote direct memory access (RDMA). Unlike NVMe, which is a protocol for accessing devices connected to the local system over a PCIe bus, NVMe-oF connects over a network to remote NVMe devices. NVMe-oF offers higher IOPS and lower latencies than many other remote storage options, with potentially lower CPU cost per I/O.

For further information, review:

- [What's New in VMware vSphere 6.5](#)
- [WHAT'S NEW IN PERFORMANCE VMware vSphere 6.5](#)
- [What's New in Performance for VMware vSphere 6.7?](#)
- [What's New in vSphere 7 Core Storage](#)
- [Performance Characterization of NVMe-oF in vSphere 7.0 U1](#)
- [vSphere 7 – Storage vMotion Improvements](#)

ESXi Network Configuration

Virtual Network Concepts

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

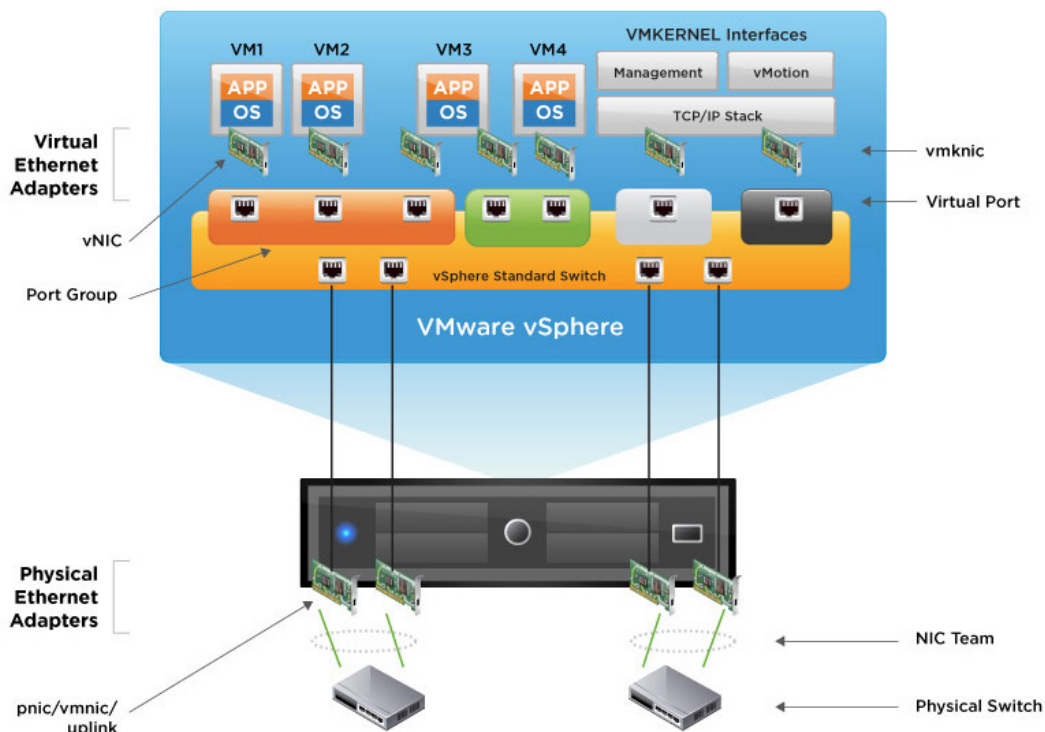


FIGURE 15. Virtual Networking Concepts

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 or 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, VM 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.

For further information, review:

- [Introduction to vSphere Networking](#)

vSphere Networking Best Practices

Some BCA workloads are more sensitive to network latency than others. To configure the network for BCA workloads, start with a thorough understanding of your workload network requirements. Monitoring the following performance metrics on the existing workload for a representative period using Linux or Windows networking monitors (sar/ perfmon), with VMware Capacity Planner™ or, preferably, with vRealize Operations can easily help determine the requirements for an BCA workload VM.

The following guidelines generally apply to provisioning the network for a BCA workload VM:

- The choice between standard and distributed switches should be made outside of the BCA workload 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 BCA workloads.
- Use NIC teaming for availability and load-balancing. NIC teaming occurs when multiple uplink adapters are associated with a single virtual switch to form a team.
- Take advantage of network I/O control to converge network and storage traffic onto 10GbE. Network I/O control was released in vSphere 4.1 and enables you to guarantee service levels (bandwidth) for particular vSphere traffic types: VM traffic, FT logging, iSCSI, NFS, management, and vSphere vMotion.
- 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.

For further information, review:

- [About vSphere Networking](#)

Multi-Stream Helper for vMotion

The streaming architecture in vMotion was first introduced in vSphere 4.1. vSphere vMotion, in versions prior to vSphere 7 Update 2, did not saturate high bandwidth NICs out-of-the-box like 25, 40, and 100 GbE. It required tuning at multiple levels such as vMotion streams, the TCP/IP VMkernel interfaces, and potentially even the NIC driver. That's because, by default, vMotion would use a single stream for handling the vMotion process.

With vSphere 7 Update 2, the vMotion process automatically spins up the number of streams according to the bandwidth of the physical NICs used for the vMotion network(s). This is now an out-of-the-box setting. The usable bandwidth for vMotion is determined by querying the underlying NICs.

This means that all VMkernel interfaces enabled for vMotion are checked, as well as the underlying physical NIC bandwidths. Depending on the outcome, a number of streams are instantiated. The baseline is one vMotion stream per 15GbE of bandwidth. This results in the following number of streams per VMkernel interface:

- 25GbE = 2 vMotion streams
- 40GbE = 3 vMotion streams
- 100GbE = 7 vMotion streams

Recommendation: As a best practice, upgrade to the latest version of vSphere to take advantage of the multi-stream helper for vMotion traffic.

For further information, review:

- [How to Tune vMotion for Lower Migration Times?](#)
- [Faster vMotion Makes Balancing Workloads Invisible](#)
- [vMotioning Monster Business Critical Applications VMs? Swim in Streams](#)
- [Learn About the vMotion Improvements in vSphere 7](#)

- [Multiple-NIC vMotion in vSphere \(2007467\)](#)
- [VMware vSphere 5.1 vMotion Architecture, Performance and Best Practices Performance Study](#)
- [vMotion Innovations in VMware vSphere 7.0 U1 Performance Study](#)
- [vSphere 7 – vMotion Enhancements](#)
- [Migration with vMotion](#)
- [Virtual Machines using large pages can temporarily become unresponsive after vMotion \(2144984\)](#)
- [Virtual machine performance degrades while a vMotion is being performed \(2007595\)](#)
- [Understanding and troubleshooting vMotion \(1003734\)](#)
- [vMotion Blogs](#)

Jumbo Frames for vSphere vMotion Interfaces

Standard ethernet frames are limited to a length of approximately 1500 bytes. Jumbo frames can contain a payload of up to 9000 bytes. This feature enables use of large frames for all VMkernel traffic, including vSphere vMotion. Using jumbo frames reduces the processing overhead to provide the best possible performance by reducing the number of frames that must be generated and transmitted by the system.

VMware tested vSphere vMotion migration of critical applications, such as Oracle and SQL, with and without jumbo frames enabled. Results showed that with jumbo frames enabled for all VMkernel ports and the vSphere distributed switch, vSphere vMotion migrations completed successfully. During these migrations, no database failovers occurred, and there was no need to modify the cluster heartbeat setting.

The use of jumbo frames requires that all network hops between 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, potentially causing 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.

Recommendation: As a best practice, consider using jumbo frames for vSphere vMotion interfaces to accelerate vMotion operation of memory-intensive BCA workloads.

For further information, review:

- [Enabling Jumbo Frames on virtual distributed switches \(1038827\)](#)
- [What's the Big Deal with Jumbo Frames?](#)

Load-Balancing on vSphere Standard Switch and Distributed Switch

Route based on originating virtual port is the default load-balancing method on vSphere Standard Switch and vSphere Distributed Switch.

Each VM running on an ESXi host has an associated virtual port ID on the virtual switch. To calculate an uplink for a VM, the virtual switch uses the VM port ID and the number of uplinks in the NIC team. After the virtual switch selects an uplink for a VM, it always forwards traffic through the same uplink for this VM as long as the machine runs on the same port. The virtual switch calculates uplinks for VMs only once unless uplinks are added or removed from the NIC team.

The port ID of a VM is fixed while the VM runs on the same host. If you migrate, power off, or delete the VM, its port ID on the virtual switch becomes free. The virtual switch stops sending traffic to this port, which reduces the overall traffic for its associated uplink. If a VM is powered on or migrated, it might appear on a different port and use the uplink, which is associated with the new port.

One of the advantages of using the **route based on originating virtual port** as the default load-balancing method is low resource consumption. This is because, in most cases, the virtual switch calculates uplinks for VMs only once.

Recommendation: As a best practice, evaluate the various load-balancing options on vSphere Standard Switch and Distributed Switch and choose route based on originating virtual port unless there is a need otherwise.

For further information, review:

- [Route Based on Originating Virtual Port](#)

RDMA (Remote Direct Memory Access) over Converged Ethernet (RoCE)

vSphere 6.5 and later releases introduced RDMA using RoCE v2 support for ESXi hosts. RDMA allows direct memory access from the memory of one computer to the memory of another computer without involving the OS or CPU. The transfer of memory is offloaded to the RDMA-capable host channel adapters (HCA).

RDMA provides low latency and higher throughput interconnects with CPU offloads between the end points. If a host has RoCE-capable network adaptor(s), this feature is automatically enabled.

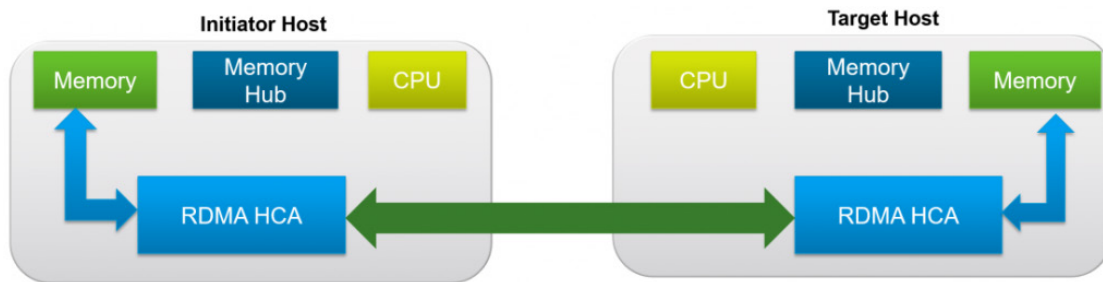


FIGURE 16. Para-Virtualized RDMA (PV-RDMA)

In this release, ESXi introduces the PV-RDMA for Linux guest OS with RoCE v2 support. PV-RDMA enables customers to run RDMA-capable applications in virtualized environments. PV-RDMA enabled VMs can also be live migrated.

The PVRDMA virtual network adapter supports remote direct memory access (RDMA) between VMs on the same ESXi host or—if both hosts have compatible network hardware—between VMs on different ESXi hosts

vSphere 6.7 introduced iSER (iSCSI Extension for RDMA) and SW-FCoE (Software Fiber Channel over Ethernet). These features enable enterprises to integrate with even more high-performance storage systems, providing greater flexibility to use the hardware that best complements their workloads.

Customers may now deploy ESXi with external storage systems supporting iSER targets. iSER takes advantage of faster interconnects and CPU offload using RDMA over converged ethernet (RoCE). We are providing iSER initiator function, which allows the ESXi storage stack to connect with iSER capable target storage systems.

ESXi introduced a software-based FCoE (SW-FCoE) initiator that can create an FCoE connection over ethernet controllers. The VMware FCoE initiator works on lossless ethernet fabric using priority-based flow control (PFC). It can work in fabric and VN2VN modes.

RDMA support is enhanced with vSphere 6.7 to bring even more performance to enterprise workloads by leveraging kernel and OS bypass, reducing latency and dependencies.

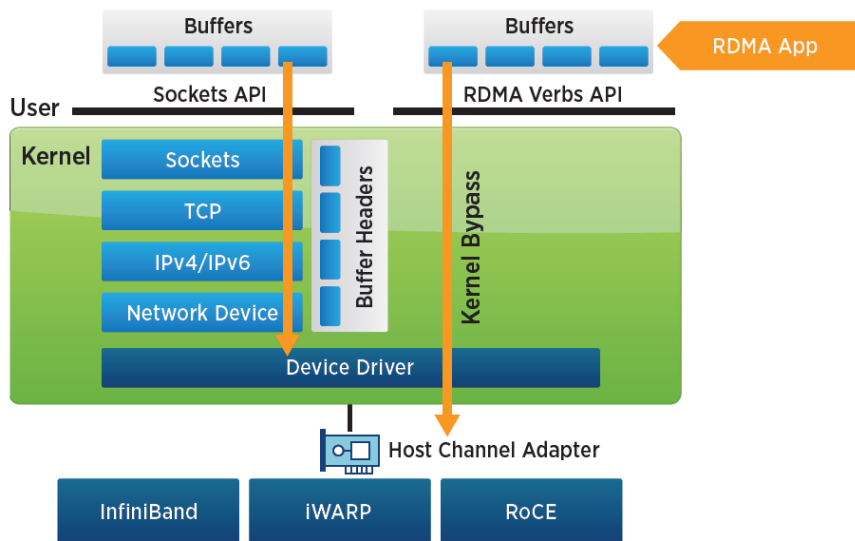


FIGURE 17. RDMA Leveraging Kernel Bypass to Improve Performance

When VMs are configured with RDMA in a passthrough mode, the workload is basically tied to a physical host with no vSphere DRS capability—that is, without vSphere vMotion migration capability. However, customers who want to harness the power of vSphere vMotion migration and vSphere DRS capability while still getting the benefits of RDMA, albeit at a very small performance penalty, can do so with paravirtualized RDMA software (PVRDMA). With PVRDMA, applications can run even in the absence of a host channel adapter (HCA) card. RDMA-based applications can be run in ESXi guests while ensuring that VMs can be live migrated. Use cases for this technology include distributed databases, financial applications, and high-performance computing.

Recommendation: As a best practice, consider using PVRDMA drivers for a VM's virtual network adapter on ESXi servers with RDMA-capable host channel adapters (HCA) to accelerate workload performance.

For further information, review:

- [Remote Direct Memory Access for Virtual Machines](#)
- [Performance of RDMA and HPC Applications in Virtual Machines using FDR InfiniBand on VMware vSphere](#)

Network Other Features

The vSphere 6.x release includes vmxnet3 version 4, which supports some new features.

- **RSS for UDP** - Receive side scaling (RSS) for the user data protocol (UDP) is now available in the vmxnet3 v4 driver. Performance testing of this feature showed a 28 percent improvement in receive packets per second. The test used 64-byte packets and four receive queues.
- **RSS for ESP** - RSS for encapsulating security payloads (ESP) is now available in the vmxnet3 v4 driver. Performance testing of this feature showed a 146 percent improvement in receive packets per second during a test that used IPSEC and four receive queues.
- **Offload for Geneve/VXLAN** - Generic network virtualization encapsulation (Geneve) and VXLAN offload is now available in the vmxnet3 v4 driver. Performance testing of this feature showed a 415 percent improvement in throughput in a test that used a packet size of 64K with eight flows.

For further information, review:

- [What's New in Performance for VMware vSphere 6.7 - Network](#)

vSphere Security Configuration

The vSphere platform has a reach set of security features which may help a DBA administrator to mitigate BCA workloads.

vSphere Security Features

vSphere 6.5 vMotion introduces encrypted vMotion, providing end-to-end encryption of vMotion traffic and protecting VM data from eavesdropping occurrences on untrusted networks. Encrypted vMotion provides complete confidentiality, integrity, and authenticity of data transferred over the vMotion network without any additional requirement for specialized hardware.

Encrypted vMotion does not rely on the secure sockets layer (SSL) and internet protocol security (IPsec) technologies for securing vMotion traffic. Instead, it implements a custom encrypted protocol above the TCP layer. To secure VM migration, vSphere 6.5 encrypted vMotion encrypts all vMotion traffic, including the TCP payload and vMotion metadata, using the most widely used AES-GCM encryption standard algorithms provided by the FIPS-certified VMKernel vmkcrypto module.

vSphere 6.5 provides several other enhancements that help to lower a security risk for a VMs hosting BCA workload. These features include:

3. Secure boot support for virtual machines
4. Secure boot plus cryptographic
5. Hypervisor assurance for ESXi

vSphere 6.7 provides several enhancements that help to lower a security risk for a VMs hosting BCA workload. This feature includes:

- Support for a virtual trusted platform module (vTPM) for the VM
- Support for Microsoft virtualization-based security
- Enhancement for ESXi secure boot
- Virtual machine UEFI secure boot
- FIPS 140-2 validated cryptographic modules turned on by default for all operations

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

vSphere 7.0 provides several enhancements that help to lower the security risk for VMs hosting BCA workloads. These features include:

- Intrinsic security
- vSphere trust authority (vTA)
- Improved certificate management

For further information, review:

- [Encrypted vSphere vMotion](#)
- [About vSphere Security](#)

Side-Channel Vulnerability Mitigation and New CPU Scheduler

The vSphere CPU scheduler in ESXi determines what VMs get to use the processors. It tries to maximize utilization of the processors while also trying to be fair to all the workloads. In determining how to place workloads on CPUs, the scheduler considers many factors such as CPU cache, memory locality, performance needs, priorities, I/O to storage and networks, and more.

The release of vSphere 6.7 Update 2 brought with it a new vSphere CPU scheduler option, the side-channel aware scheduler version 2 (SCAv2) or *sibling scheduler*. This new scheduler can help restore some performance lost to CPU vulnerability mitigations, but also carries some risk with it.

Further complicating matters is the newest Intel CPU vulnerability, [Microarchitectural Data Sampling](#) (MDS) identified by CVE-2018-12126, CVE-2018-12127, CVE-2018-12130, and CVE-2019-11091, as well as VMware Security Advisory [VMSA-2019-0008](#).

An initial version of the side-channel-aware scheduler (SCAv1) was made available in updates for previous versions of ESXi. SCAv1 limits the scheduler to use only one thread per core. A second version (SCAv2) was released with ESXi 6.7 Update 2. SCAv2 activates multithreading, but never simultaneously schedules threads from different VMs on the same core. In nearly all cases, SCAv2 imposes a lower performance penalty than SCAv1.

For further information, review:

- RedHat KB [L1TF - L1 Terminal Fault Attack - CVE-2018-3620 & CVE-2018-3646](#)
- [VMware response to 'L1 Terminal Fault - VMM' \(L1TF - VMM\) Speculative-Execution vulnerability in Intel processors for vSphere: CVE-2018-3646 \(55806\)](#)
- [VMware Performance Impact Statement for 'L1 Terminal Fault - VMM' \(L1TF - VMM\) mitigations: CVE-2018-3646 \(55767\)](#)
- [Implementing Hypervisor-Specific Mitigations for Microarchitectural Data Sampling \(MDS\) Vulnerabilities \(CVE-2018-12126, CVE-2018-12127, CVE-2018-12130, and CVE-2019-11091\) in vSphere \(67577\)](#)
- [Which vSphere CPU Scheduler to Choose](#)
- [New Scheduler Option for vSphere 6.7 U2](#)
- [Performance of vSphere 6.7 Scheduling Options](#)

Virtual Machine CPU Configuration

Allocating vCPU

Correct assignment of CPU resources is vital for BCA workloads. Occasionally, multiple virtual CPUs (vCPUs) can cause performance issues as well.

Recommendation: As a best practice, at initial sizing, ensure the total number of vCPUs assigned to all the VMs is no more than the total number of physical cores (not logical cores) available on the ESXi host machine.

For further information, review:

- [Determining if multiple virtual CPUs are causing performance issues \(1005362\)](#)

vNUMA, corespersocket and PreferHT

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

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

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

General rules

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

- vNUMA is not exposed for any VM having less than nine vCPU assigned (default).
- vNUMA is not exposed to any VM having less vCPU than the size of pNUMA of a host (default).
- vNUMA is not exposed if the **CPU hot add** feature is enabled on Linux. Before Windows 10 Build 20348, Windows will create additional fake nodes to accommodate those potential vCPUs. Read more about this in the *CPU Hot Plug/Add* section.
- The VM memory size is not considered for the creation of the vNUMA topology. For the *unbalanced NUMA* memory configuration (amount of configured memory span NUMA nodes while vCPU count stays within a NUMA node) see recommendation in *Advanced vNUMA VM configuration examples* in the Appendix.
- vNUMA topology for a VM is created only once and by default is not updated if a VM is vMotion migrated to server hardware with a different pNUMA configuration. To trigger vNUMA topology refresh, you can add/remove a vCPU or use the advanced setting.
- VM hardware version 8 is required to have vNUMA exposed to the guest OS.
- vNUMA topology will be updated if changes for the CPU configuration of a VM are completed. pNUMA information from the host, where the VM was started at the time of the change, will be used for creating vNUMA topology. Changing the memory configuration will have no effect on vNUMA topology.
- vNUMA mismatch can occur in an ESXi cluster where EVC has been enabled. If a VM started on a host with one type of NUMA topology is relocated (i.e., manually vMotion migrated or moved by DRS triggers) to another host, the VM will experience progressively worse performance until it is rebooted, at which point its correct NUMA topology will be recalculated.

Cores per Socket and vSphere Version 6.0 and Early 5.x

Cores per socket was originally intended to address licensing issues where some operating systems had limitations on the number of sockets that could be used but did not limit core count.

The vNUMA topology is directly specified by using the **Cores per Socket** setting in the VM configuration.

As a general rule, try to reflect your hardware configuration while configuring cores per socket ratio.

The number of sockets assigned will dictate the number of NUMA clients created. As a best practice, reflect your server hardware pNUMA topology configuring the cores:socket ratio. One caveat with using this setting is license affiliation when only a defined number of sockets will be accessed by the guest OS or application.

Example: A server with total of 16 CPU cores and 192GB RAM (8 cores and 96GB of RAM in each pNUMA node) is used to host a VM with 16 vCPU. **Cores per Socket** is set to two (see **Figure 18** below).

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

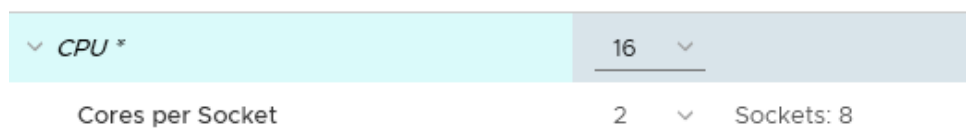


FIGURE 18. VM Cores per Socket Configuration

Cores per Socket and vSphere Version 6.5 and Later

As of vSphere 6.5, changing the `corespersocket` value no longer influences vNUMA and the configuration of the vNUMA topology.

Autosizing of vNUMA is introduced. The **Cores per Socket** setting is not taken into account while creating the vNUMA topology. The configuration of vSockets and `corespersocket` only affects the presentation of the virtual processors to the guest OS (generally required for licensing).

The final vNUMA topology for a VM will be automatically configured by ESXi using the number of physical cores per CPU package of the physical host where the VM is about to start. The total number of vCPUs assigned to a VM will be consolidated within the minimal possible number of proximity domains (PPD), equal to the size of a CPU package.

The auto-created vNUMA topology will be saved in the VM configuration file. In most cases using autosizing will create an optimized topology. Do not disable or modify vNUMA autosizing without clear use cases that require an administrator to do so. Consult the configuration examples section below to identify the best possible configuration.

To revert to the earlier behavior in vSphere 6.0, use advanced settings and set `Numa.FollowCoresPerSocket` to 1.

Example: The same host and VM with vCPU configuration shown in **Figure 18** is used. As a result, the two vNUMA nodes will be exposed to the VM.

In another example, if you create a 4-vSocket VM with four `corespersocket` (total of 16 vCPU) on a dual-socket, 16-core physical ESXi host, prior to vSphere 6.5, vNUMA would have created four vNUMA nodes based on the `corespersocket` setting. As of vSphere 6.5, the guest OS will still see four sockets and four cores per socket, but vNUMA will now only create one vNUMA node for the entire VM since it can be placed in a single physical NUMA node.

In still another example, if you create a 2-vSocket VM with six `corespersocket` (total of 12 vCPU) on a dual-socket, 24-core physical ESXi host, as of vSphere 6.5, the guest OS will still see two sockets and six cores per socket, but vNUMA will now only create one vNUMA node for the entire VM since it can be placed in a single physical NUMA node.

```
[root@oracle19c-ol8-rman ~]# lscpu
Architecture: x86_64
CPU op-mode(s): 32-bit, 64-bit
Byte Order: Little Endian
CPU(s): 12
On-line CPU(s) list: 0-11
Thread(s) per core: 1
Core(s) per socket: 6
Socket(s): 2
NUMA node(s): 1
Vendor ID: GenuineIntel
CPU family: 6
Model: 85
Model name: Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz
Stepping: 4
CPU MHz: 2693.671
BogoMIPS: 5387.34
Hypervisor vendor: VMware
Virtualization type: full
L1d cache: 32K
L1i cache: 32K
L2 cache: 1024K
L3 cache: 33792K
NUMA node0 CPU(s): 0-11
Flags: fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss ht syscall nx pdpe1gb rdtscp lm constant_tsc arch_p
erfmon nopl xtopology tsc_reliable nonstop_tsc cpuid pni pclmulqdq ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand hyperv
sor lahf_lm ahm 3dnowprefetch cpuid_fault invpcid_single pti ssbd ibrs ibpb stibp fsgsbase tsc_adjust bmi1 avx2 smep bmi2 invpcid avx512f avx512dq rdseed adx smap clflushopt
clwb avx512cd avx512bw avx512vl xsaveopt xsavec xgetbv1 xsaves arat pku ospke md_clear flush_l1d arch_capabilities
[root@oracle19c-ol8-rman ~]#
[root@oracle19c-ol8-rman ~]# numactl --hardware
available: 1 nodes (0)
node 0 cpus: 0 1 2 3 4 5 6 7 8 9 10 11
node 0 size: 96057 MB
node 0 free: 95306 MB
node distances:
node 0
0: 10
[root@oracle19c-ol8-rman ~]#
```

FIGURE 19. `lscpu` command output showing sockets and cores per socket

NOTE: Although `corespersocket` no longer directly sets the vNUMA topology, some `corespersocket` values could result in suboptimal guest OS topologies (i.e., topologies that are not efficiently mapped to the physical NUMA nodes, potentially resulting in reduced performance).

VM vNUMA Sizing Recommendation

Recommendation: Even though the introduction of vNUMA helps significantly to overcome issues with multicore VMs, best practices should be considered while architecting a vNUMA topology for a VM.

1. The best possible performance is generally observed when a VM could fit into one pNUMA node and could benefit from the local memory access. For example, when running a VM hosting a BCA workload on a host with twelve pCores per pNUMA, as a general rule, assign no more than twelve vCPUs to a VM.
2. If a wide-NUMA configuration is unavoidable, consider reevaluating the recommendations given and execute extensive performance testing before implementing the configuration. Monitoring should be implemented for important CPU counters after moving to the production.

For further information, review:

- [Setting the number of cores per CPU in a virtual machine \(1010184\)](#)
- [Setting corespersocket can affect guest OS topologies \(81383\)](#)
- [Virtual Machine vCPU and vNUMA Rightsizing – Guidelines](#)
- [Does corespersocket Affect Performance?](#)
- [What is PreferHT and When to Use It](#)
- [Virtual Machines with preallocated memory might be placed NUMA remote at random power-ons \(76362\)](#)

CPU Hot Plug and Hot Add

The hot plug and hot add CPU and memory features were added in vSphere 4.0 release. However, for best results, use VMs that are compatible with ESXi 5.0 and later.

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 the VM is not aware of which of its vCPUs are on the same NUMA node and might increase remote memory access. This removes the ability for the guest OS and applications to optimize based on NUMA and results in a possible reduction in performance.

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 BCA workloads.

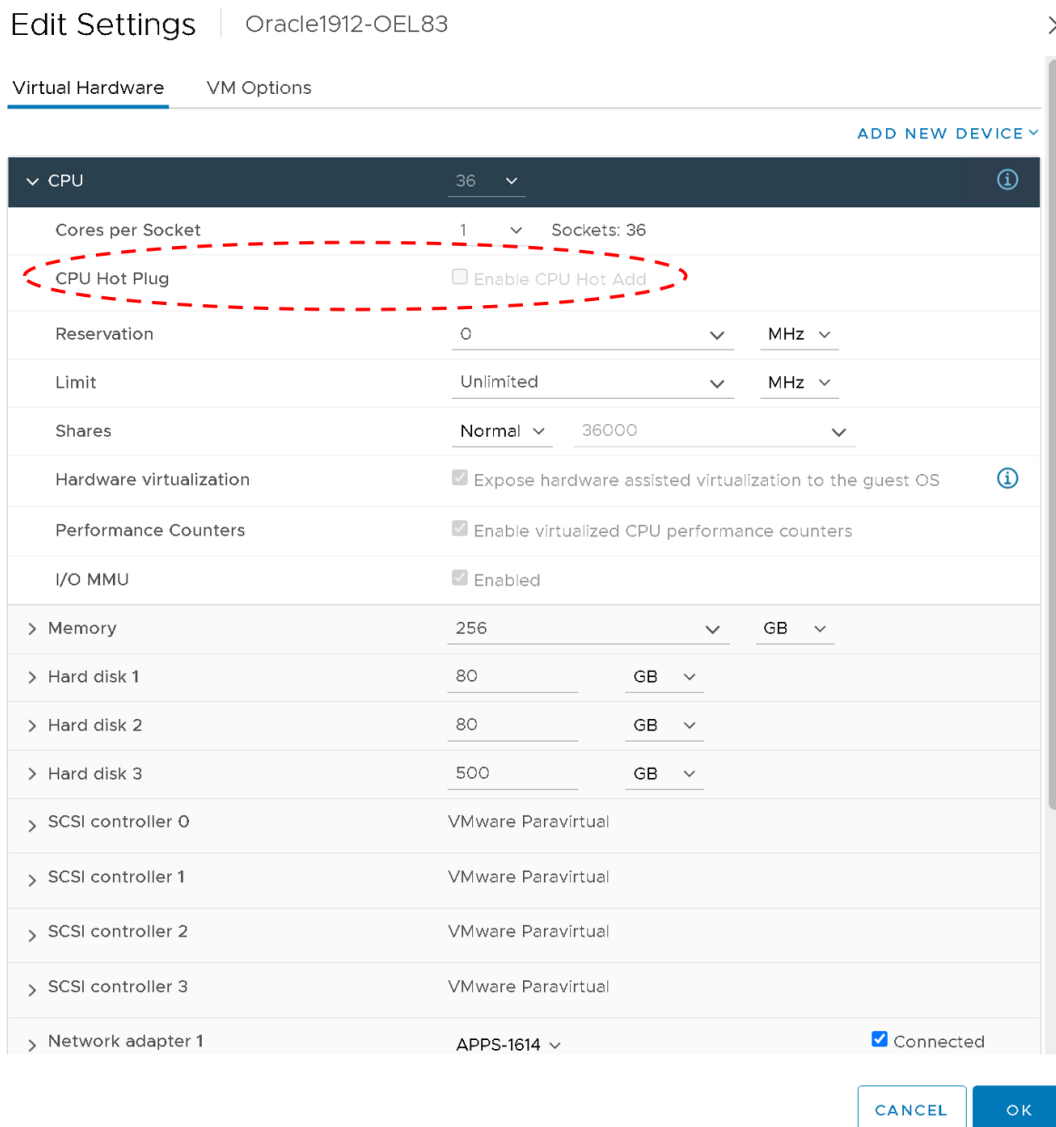


FIGURE 20. Disabling CPU Hot Plug (Uncheck Enable CPU Hot Add checkbox)

As shown in **Figure 21**, the `vmddumper` command clearly indicates that for a running VM with a feature **CPU Hot Add**, exposing of vNUMA topology will be disabled.

```

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

```

FIGURE 21. The `vmddumper` Command Provided VM Configuration for a VM with CPU Hot Add Enabled

Leaving CPU Hot Add at its default setting of disabled is a performance best practice for large VMs. Performance with the default setting of **CPU Hot Add** disabled improved from two to eight percent better than when **CPU Hot Add** was enabled.

Recommendation: As a best practice for Linux, enable CPU hot plug for VMs that do not need vNUMA optimization. For Windows versions prior to Windows 10 Build 20348, enabling CPU hot plug will create additional fake nodes to accommodate those potential vCPUs.

For further information, review:

- [vNUMA is disabled if VCPU hot plug is enabled \(2040375\)](#)
- [CPU Hot Add Performance in vSphere 6.7](#)
- [Impact of CPU Hot Add on NUMA scheduling](#)
- [Enabling vCPU HotAdd creates fake NUMA nodes on Windows \(83980\)](#)
- [CPU HotAdd for Windows VMs: How BADLY Do You Want It?](#)

CPU Affinity

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

Recommendation: As a best practice, avoid using CPU affinity in production because it limits the hypervisor's ability to efficiently schedule vCPUs on the physical server and causes poor performance. It also disables the ability to vMotion a VM.

For further information, review:

- [Scheduler operation when using CPU Affinity \(2145719\)](#)

Latency Sensitive Setting

By default, the ESXi network stack is configured to drive high network throughput at low CPU cost. While this default configuration provides better scalability and higher consolidation ratios, it comes at the cost of potentially higher network latency. vSphere provides a variety of configuration options to improve performance of workloads that are highly sensitive to network latency (i.e., workloads that are impacted by latency on the order of a few tens of microseconds).

The **Latency Sensitive** setting is a VM option which defaults to **Normal**. Setting this to **High** can yield significantly lower latencies and jitter, due to the following mechanisms that take effect in ESXi:

- Exclusive access to physical resources, including pCPUs dedicated to vCPUs with no contending threads for executing on these pCPUs
- Full memory reservation eliminates ballooning or hypervisor swapping leading to more predictable performance with no latency overheads due to such mechanisms.
- Halting in the VM monitor when the vCPU is idle, leading to faster vCPU wake-up from halt, and bypassing the VMkernel scheduler for yielding the pCPU. This also conserves power as halting makes the pCPU enter a low power mode, compared to spinning in the VM Monitor with the `monitor_control.halt_desched=FALSE` option.
- Disabling interrupt coalescing and LRO automatically for VMXNET 3 virtual NICs
- Optimized interrupt delivery path for VM DirectPath I/O and SR-IOV passthrough devices, using heuristics to derive hints from the guest OS about optimal placement of physical interrupt vectors on physical CPUs

Caution must be exercised before enabling the **Latency Sensitive** setting for BCA workloads owing to the number or restrictions imposed as listed above.

Examples of typical applications that require the setting to be **High** include VOIP, media player apps, and apps that require frequent access to the mouse or keyboard devices.

Recommendation: As a best practice, avoid enabling the Latency Sensitive setting for BCA workloads unless tests have shown it should otherwise be enabled.

For further information, review:

- [Best Practices for Performance Tuning of Latency-Sensitive Workloads in vSphere VMs](#)
- [Deploying Extremely Latency-Sensitive Applications in VMware vSphere 5.5 Performance Study](#)

Per Virtual Machine EVC Mode

vSphere previously implemented Enhanced vMotion Compatibility (EVC) as a cluster-wide attribute because, at the cluster-wide level, you can make certain assumptions about migrating a VM (e.g., even if the processor is not the same across all ESXi hosts, EVC still works). However, this policy can cause problems when trying to migrate across vCenter hosts or vSphere clusters.

vSphere 6.7 introduces a new feature – the ability to configure the EVC mode for a particular VM instead of the whole cluster (see section 3.3.3 for more details). By implementing per-VM EVC, the EVC mode becomes an attribute of the VM rather than the specific processor generation it happens to be booted on in the cluster.

The per-VM EVC mode determines the set of host CPU features that a VM requires in order to power on and migrate. The EVC mode of a VM is independent from the EVC mode defined at the cluster level.

Setting the EVC mode as a VM attribute on a VM hosting a BCA workload can help to prevent downtime while migrating a VM between datacenters or vCenters, or to a public cloud such as VMware Cloud.

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

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

Recommendation: As a best practice, consider using per-VM EVC mode to avoid downtime while migrating a VM between datacenters or vCenters, or to a public cloud such as VMware Cloud.

For further information, review:

- [Virtual EVC per VM](#)
- [Enhanced vMotion Compatibility as a Virtual Machine Attribute](#)

Virtual Machine Memory Configuration

One of the most critical system resources for BCA workloads is memory. Lack of memory resources for BCA workloads will induce the OS to page memory to disk, resulting in increased disk I/O activities, which are considerably slower than accessing memory. Lack of hypervisor memory resources results in memory contention, having a significant impact on BCA workloads performance.

When BCA workloads are virtualized, the hypervisor performs virtual memory management without the knowledge of the guest OS and without interfering with the guest OS's own memory management subsystem.

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

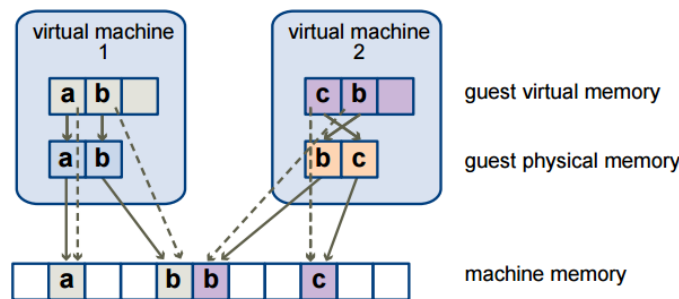


FIGURE 22. Memory Mappings between Virtual, Guest, and Physical Memory

Memory Sizing Considerations

Memory sizing considerations include the following:

- When designing for performance to prevent the memory contention between VMs, avoid overcommitment of memory at the ESXi host level ($\text{HostMem} \geq \text{Sum of (VMs memory + overhead)}$). If a physical server has 256GB of RAM, do not allocate more than that amount to the VMs residing on it, taking memory overhead into consideration as well.
- When collecting performance metrics for making a sizing decision for a VM running BCA workloads (e.g., Oracle) use Oracle AWR reports or v\$ tables for metrics. For SQL Server, use SQL Server provided metrics (available in the DMV: `sys.dm_os_process_memory`).
- Do not use vSphere or Windows Guest OS-provided memory metrics (e.g., vSphere-provided *memory consumed* or, especially, *memory active*). Active memory is defined as the *amount of memory* that is actively used, as estimated by VMkernel based on recently touched memory pages.
- Consider BCA workload memory limitations while assigning memory to a VM. In addition to what the BCA workload application needs, ensure that we also account for what the OS will need and any other software that is required.
- Consider checking the hardware pNUMA memory allocation to identify the maximum amount of memory that can be assigned to a VM without crossing the pNUMA boundaries.
- VMs require a certain amount of overhead memory to power on. You should be aware of the amount of this overhead. The following table lists the amount of overhead memory a VM requires to power on. After a VM is running, the amount of overhead memory it uses might differ from the amount listed in Table 6.

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

TABLE 6. Sample Overhead Memory on Virtual Machines

For further information, review:

- [Understanding the Memory Active and Memory Usage indicators on the Performance tab \(1002604\)](#)
- [vSphere Resource Management](#)
- [Understanding vSphere Active Memory](#)
- [Understanding Memory Resource Management in VMware vSphere Performance Study](#)

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.

Use BCA workload memory performance metrics and work with your application administrator to determine the BCA workload maximum server memory size and maximum number of worker threads. Refer to the VM overhead (Table 6) for the VM overhead.

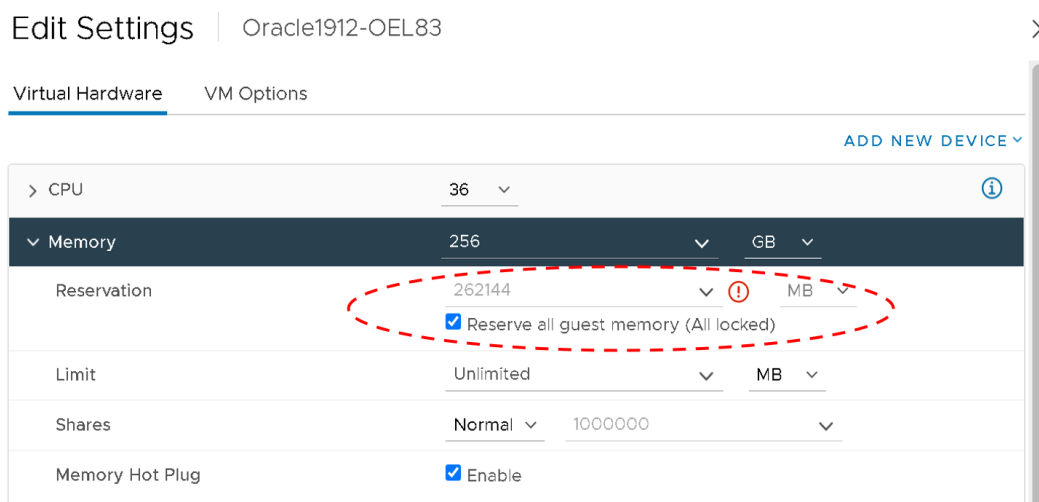


FIGURE 23. Setting Memory Reservation

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

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

The image below illustrates an example memory setting used for a VM:

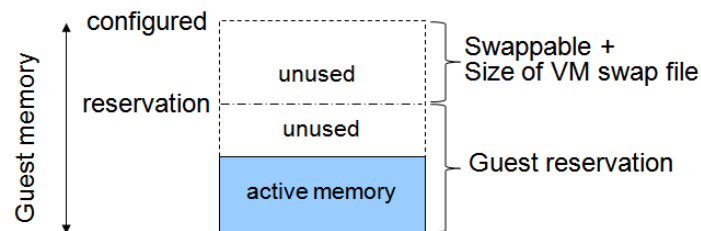


FIGURE 24. Virtual Machine Memory Settings

Definition of terms:

- Configured memory – Memory size of VM assigned at creation
- Active memory – Memory recently accessed by applications in the VM
- Reservation – Guaranteed lower bound on the amount of memory that the host reserves for the VM, which cannot be reclaimed by ESX/ESXi for other VMs. This comes into play only where there are memory resource constraints on the host or ESXi cluster.
- Swappable – Virtual machine memory that can be reclaimed by the balloon driver or, in the worst case, by ESX/ESXi swapping. This is the automatic size of the swap file that is created for each VM on the VMFS file system (`.vswp` file).

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

Recommendation: As a best practice, consider setting memory reservations for memory-intensive production BCA workloads on a case-by-case basis.

For further information, review:

- [vSphere Resource Management](#)
- [Unswapping swapped pages](#)

Memory Hot Plug

Like CPU hot plug, memory hot plug enables a VM administrator to add memory to the VM with no down time.

Before vSphere 6.5, when memory hot add was configured on a wide-NUMA VM, it would always be added to node0, creating NUMA imbalance which could be solved by rebalancing memory between vNUMA nodes, which involved powering the VM off or moving it with vMotion to a different host.

With vSphere 6.5 and later, when enabling memory hot plug and adding memory to a VM, the memory will be added evenly to all vNUMA nodes, making this feature usable for more use cases.

After memory has been added to the VM, one can increase the memory requirements for the BCA workload as needed.

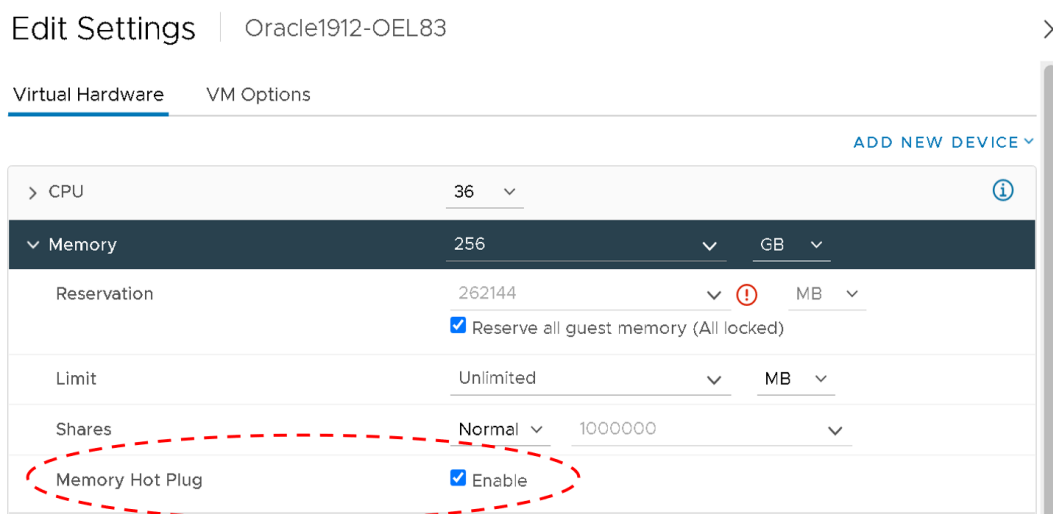


FIGURE 25. Setting Memory Hot Plug

Recommendation: As a best practice, rely on rightsizing more than on memory hot plug. The decision whether to use this feature should be made on a case-by-case basis and not be implemented in the VM template used to deploy the BCA workload.

For further information, review:

- [Change Memory Hot Add Settings](#)

Virtual Machine Storage Configuration

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

VM Storage Best Practices

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

VM PVSCSI Storage Controller and PVSCSI/VMDK Queue Depth

Utilize the **VMware Paravirtualized SCSI (PVSCSI) Controller** as the virtual SCSI controller for data and log VMDKs. The PVSCSI controller is the optimal SCSI controller for an I/O-intensive application on vSphere, allowing not only a higher I/O rate but also lowering CPU consumption compared with LSI Logic SAS.

The **queue depth** of outstanding commands in the guest OS SCSI driver can significantly impact disk performance. A queue depth that is too small, for example, limits the disk bandwidth that can be pushed through the VM.

PVSCSI adapters provide higher queue depth, increasing I/O bandwidth for the virtualized workload.

VMware supports up to four adapters per VM and as many as necessary, up to this limit, should be leveraged. Placing the database OS, data, and transaction logs onto a separate vSCSI adapter optimizes I/O by distributing load across multiple target devices and allowing for more queues on the OS level. Consider distributing disks between controllers.

Beginning with vSphere 6.7, 64 disks per PVSCSI controller can be supported.

Recommendation: As a best practice, increase the PVSCSI and VMDK queue depth to its maximum in order to push larger I/O bandwidth, unless the underlying storage vendor recommends otherwise. Use multiple PVSCSI adapters with VMDKs spread across the PVSCSI controllers for load-balancing and fairness.

For further information, review:

- [Large-scale workloads with intensive I/O patterns might require queue depths significantly greater than Paravirtual SCSI default values \(2053145\)](#)
- [Configuring disks to use VMware Paravirtual SCSI \(PVSCSI\) controllers \(1010398\)](#)
- [vSphere 7.0 Configuration Limits - VMs](#)
- [Which vSCSI controller should I choose for performance?](#)
- [SCSI, SATA, and NVMe Storage Controller Conditions, Limitations, and Compatibility](#)
- [PVSCSI and Large IO's](#)

VM vNVMe Storage Controller

In vSphere 6.5, a new type of virtual controller was introduced, **vNVMe**, undergoing significant performance enhancements with the release of vSphere 6.7. The vNVMe controller may bring performance improvement and reduce I/O processing overhead, especially in combination with low latency SSD drives on all-flash storage arrays or combined with the vPMemDisks. Consider testing the configuration using a copy of your production database to check if this change will be beneficial.

vSphere 6.7 and VM hardware 14 is strongly recommended for any implementation of vNVMe controller.

Recommendation: As a best practice, consider using vNVMe controllers for VM storage on low-latency SSD drives in all-flash storage arrays.

For further information, review:

- [Using Virtual NVMe from ESXi 6.5 and virtual machine Hardware Version 13 \(2147714\)](#)
- [Add an NVMe Controller](#)

Partition Alignment

Aligning file-system partitions is a well-known storage best practice for Oracle and SQL Server database workloads. Partition alignment on both physical machines and VMFS partitions prevents performance I/O degradation caused by unaligned I/O. vSphere 5.0 and later automatically aligns VMFS5 partitions along a 1MB boundary and most modern operating systems do the same. In rare cases, some additional manual efforts are required.⁵

Recommendation: As a best practice...

- Create VMFS partitions using the VMware vCenter web client. They are aligned by default.
- Modern Linux and Windows operating systems will automatically align a disk to a 1MB boundary. Make sure the disk partitions within the guest are aligned. Refer to the OS-specific tools for disk partition alignment.
- Consult with the storage vendor for alignment recommendations on their hardware.

For further information, review:

- [Recommendations for Aligning VMFS Partitions](#)
- [Guest OS Partition Alignment](#)

VMDK File Layout

When running on VMFS, VM disk files can be deployed in three different formats: thin, zeroed thick, and eager zeroed thick. Thin provisioned disks enable 100 percent storage on demand, where disk space is allocated and zeroed at the time the disk is written. Zeroed thick disk storage is pre-allocated, but blocks are zeroed by the hypervisor the first time the disk is written. Eager zeroed thick disks are 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 occurring during first write of the disk. Depending on the BCA workload configuration and the type of workload, the performance difference could be significant.

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

For further information, review:

- [VMware virtual disk provisioning policies](#).

⁵ Special consideration should be taken when working with the VMFS3 (and VMFS5 upgraded from VMFS3) formatted datastores. It's recommended to reformat such datastores with VMFS5 or 6.

Virtual Disks Hot Add and Hot Remove

One can add a virtual hard disk to an existing VM without shutting the VM down. You can add a virtual hard disk to a VM before or after you add a SCSI or SATA storage controller. The new disk is assigned to the first available virtual device node on the default controller, for example (0:1). Only device nodes for the default controller are available unless you add additional controllers.

For further information, review:

- [Add a Hard Disk to a Virtual Machine](#)

Virtual Disks Hot Extend

Beginning with ESX 4.1, you can extend and add virtual disks to a VM when it is powered on (after installing VMware Tools).

RHEL 7 and above allow resizing disk partitions online without any downtime. As to older style partitions, this feature has been added in the RHEL 7 current release with a feature request (i.e., RFE has been filed to add support for online resizing of disk partitions to RHEL 7 in private RHBZ#853105. With this feature, it's possible to resize the disk partitions online in RHEL 7).

For further information, review:

- [Increasing the size of a virtual disk \(1004047\)](#).

VM Snapshot

A VM snapshot preserves the state and data of a VM at a specific point in time. When a snapshot is created, it will store the power state of the VM and the state of all devices, including virtual disks. To track changes in virtual disks a special *delta* file is used, which contains a continuous record of the block level changes to a virtual disk.

Snapshots are widely used by backup software or by infrastructure administrators and DBAs to preserve the state of a VM before implementing changes (e.g., upgrade of the BCA workload, installing patches).

NOTE: Do not use snapshots taken directly from vCenter as a *recovery tool*, especially if the VM hosts database-like or time-sensitive applications.

Take Snapshot | Oleg-Jump01

Name

VM Snapshot 7/13/2018, 12:50:07 PM

Description

☐ Snapshot the virtual machine's memory
 ☒ Quiesce guest file system (Needs VMware Tools installed)

CANCEL

OK

FIGURE 26. Take Snapshot Options

Recommendation: As a best practice, consider the following when taking snapshots on a VM hosting BCA workloads:

1. Offline snapshot (i.e., a VM is powered off when a snapshot is taken) can be used without special considerations.
2. If an online snapshot (i.e., VM is powered on and guest OS is running) needs to be taken:
 - a. Avoid using the **Snapshot the VM's memory** option as this may stun a VM. Rely on BCA workload mechanisms to prevent data loss by losing in-memory data.
 - b. Use the **Quiesce guest file system** option to ensure that a disk-consistent snapshot will be taken. Special notes:
 - i. Avoid taking an online snapshot if VMware Tools are not installed or not functional as this may lead to the inconsistent disk state.
 - ii. Consider checking the status of the volume shadow copy service (VSS) on the Windows OS before taking a snapshot.
 - iii. If using a third-party utility for snapshots, make sure it is VSS compliant.
 - c. Be aware that on highly loaded BCA workloads producing a high number of disk I/O, snapshot operations (i.e., creation of an online snapshot, online removal of a snapshot) may take a long time and can potentially cause performance issues. Consider either to plan the snapshots operations for the off-peak hours, use offline creation or removal of snapshots, or use vVols with storage array-level snapshot integrations.
3. Do not run a VM hosting a BCA workload on a snapshot for more than 72 hours.⁶
4. Snapshot is not a replacement for a backup. The delta disk file contains only references to the changes and not the changes itself.
5. Consider using VMFS6 and SEsparse snapshots for performance improvements.

For further information, review:

- [High co-stop \(%CSTP\) values seen during virtual machine snapshot activities \(2000058\)](#)
- [Virtual machine becomes unresponsive or inactive when taking memory snapshot \(1013163\)](#)
- [Overview of virtual machine snapshots in vSphere \(1015180\)](#)
- [Snapshot removal stops a virtual machine for long time \(1002836\)](#)
- [Managing snapshots in vSphere Web Client \(2032907\)](#)
- [Understanding VM snapshots in ESXi \(1015180\)](#)
- [Best practices for using snapshots in the vSphere environment \(1025279\)](#)
- [VMware Snapshots](#)
- [When and why do we “stun” a virtual machine?](#)

⁶ Depending on the workload and environment, this recommendation may vary, but in general should not exceed 72 hours. Sometimes a much shorter time is preferred.

BCA Workloads on VMware vSAN

When deploying VMs with BCA workload 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 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 mission critical BCA workloads, especially Oracle and SQL Server databases, 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 (8KB 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 100GB user database, design a 2 x 2 x 100GB SSD size.
- Select the 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 a 10 percent write ratio.
- Plan for availability – Design more than three hosts and additional capacity that enable the cluster to automatically remediate in the event of a failure. Setting FTT greater than 1 means more write copies to vSAN disks. Unless special data protection is required, FTT=1 can satisfy most mission-critical BCA workloads.
- Set proper SPBM – vSAN SPBM can set availability, capacity, and performance policies per VM.
- Set Object Space Reservation (OSR) – Based on workload demands, one could set the OSR to 100 percent. The capacity is allocated up front from the vSAN datastore. Keep in mind, there is no difference in performance between a thin-provisioned and thick-provisioned VMDK once they are fully inflated to the database current capacity and as long as there is no increase to the size of the VMDK.
- 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 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 255GB 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. This 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.
- If an all flash vSAN is used 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 files and redo/transaction log of databases. 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 with cost savings but that depends on workload to workload. Regarding the virtual disks for redo/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 200GB database in all-flash vSAN and found:
 - The TPS, transaction time, and response time were similar in all configurations.
 - Virtual disk latency was less than two milliseconds in all test configurations.
- VMware suggests setting stripe width as needed to split the disk object into multiple components in order to distribute the object components to more disks in different disk groups. In some situations, this setting may be needed for large virtual disks.
- Use quality of service for database restore operations. vSAN 6.2 and later has 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 encounter performance contention with I/O-intensive operations (such as database maintenance), can benefit from QoS. Recommendation is not to use QoS for BCA workloads with performance requirements.
- Ensure that the network infrastructure used for vSAN network traffic is robust enough to sustain the planned workload; a minimum of 10Gb of dedicated network bandwidth is recommended for vSAN network traffic.

For further information, review:

- [Should you be using Eager Zero Thick on vSAN \(or VMFS\)?](#)

Virtual Machine Network Configuration

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

Virtual Networking Best Practices

Some BCA workloads are more sensitive to network latency than others. To configure the network for BCA workloads, start with a thorough understanding of your workload network requirements.

Monitoring the following performance metrics on the existing workload for a representative period using Linux or Windows networking monitors (sar/ perfmon), VMware Capacity Planner or, preferably, with vRealize Operations, can easily help determine the requirements for a BCA workload VM.

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.

Recommendation: As a best practice, consider using VMXNET3 paravirtualized network adapters for VM network traffic.

For further information, review:

- [Choosing a network adapter for your VM \(1001805\)](#)
- [Large packet loss at the guest operating system level on the VMXNET3 vNIC in ESXi \(2039495\)](#)
- [VMXNET3 resource considerations on a Linux VM that has vSphere DirectPath I/O with vMotion enabled \(2058349\)](#)

Interrupt Coalescing

Virtual network interrupt coalescing can reduce the number of interrupts, thus potentially decreasing CPU utilization. Depending on the workload, this might increase network latency by anywhere from a few hundred microseconds to a few milliseconds. By default, this feature is activated for all virtual NICs in ESXi.

VMXNET3 by default also supports an adaptive interrupt coalescing algorithm, for the same reasons that physical NICs implement interrupt moderation. This virtual interrupt coalescing helps drive high throughputs to VMs with multiple vCPUs with parallelized workloads (for example, multiple threads), while at the same time striving to minimize the latency of virtual interrupt delivery.

However, if your workload is extremely sensitive to latency, we recommend you disable virtual interrupt coalescing for VMXNET3 virtual NICs.

By default, this feature is activated for all virtual NICs in ESXi.

For VMXNET3 virtual NICs, however, this feature can be set to one of three schemes or deactivated by changing the ethernetX.coalescingScheme variable (where X is the number of the virtual NIC to configure). The feature can be deactivated by setting ethernetX.coalescingScheme to disabled. Deactivating this feature will typically result in more interrupts (and thus higher CPU utilization) but will typically also lead to lower network latency.

Recommendation: As a best practice, consider disabling virtual network interrupt coalescing for low latency requirements.

For further information, review:

- [Low throughput for UDP workloads on Windows VMs \(2040065\)](#)
- [Performance Best Practices for VMware vSphere 7.0](#)
- [Virtual network interrupt coalescing](#)

Receive Side Scaling (RSS)

Receive side scaling (RSS) is a network driver technology that enables the efficient distribution of network receive processing across multiple CPUs in multiprocessor systems.

RSS is enabled by default in modern Windows OS versions, but an additional configuration is required to enable it on the VMXNET network adapter's properties.

Receive side scaling (RSS) and multi-queue support are included in the VMXNET3 Linux device driver.

On Linux, RSS, also known as multi-queue, receive, is also implemented. In some distributions it is referred to as receive packet steering (RPS), which is the software version of hardware-based RSS.

Recommendation: As a best practice, consider enabling RSS to increase workload performance.

For further information, review:

- [What is Receive Side Scaling \(RSS\), and how do I configure it in RHEL?](#)
- [RSS and multiqueue support in Linux driver for VMXNET3 \(2020567\)](#)
- [Windows Receive Side Scaling \(RSS\)](#)
- [Windows Enabling Receive Side Scaling](#)

TCP Segmentation Offload

Use TCP segmentation offload (TSO) in VMkernel network adapters and VMs to improve the network performance in workloads that have severe latency requirements.

Enabling TSO on the transmission path of physical network adapters, and VMkernel and VM network adapters, improves the performance of ESXi hosts by reducing the overhead of the CPU for TCP/IP network operations. When TSO is enabled, the network adapter divides larger data chunks into TCP segments instead of the CPU. The VMkernel and the guest OS can use more CPU cycles to run applications.

To benefit from the performance improvement that TSO provides, enable TSO along the data path on an ESXi host including physical network adapters, VMkernel and guest operating systems. By default, TSO is enabled in the VMkernel of the ESXi host, and in the VMXNET 2 and VMXNET 3 VM adapters.

Recommendation: As a best practice, consider using TCP segmentation offload (TSO) in VMkernel network adapters and VMs to improve the network performance in workloads that have severe latency requirements.

For further information, review:

- [Understanding TCP Segmentation Offload \(TSO\) and Large Receive Offload \(LRO\) in a VMware environment \(2055140\)](#)
- [TCP Segmentation Offload](#)

Large Receive Offload

Use large receive offload (LRO) to reduce the CPU overhead required for processing packets that arrive from the network at a high rate.

LRO reassembles incoming network packets into larger buffers and transfers the resulting larger but fewer packets to the network stack of the host or VM. The CPU has to process fewer packets than when LRO is disabled, which reduces its utilization for networking especially in the case of connections that have high bandwidth.

To benefit from the performance improvement of LRO, enable LRO along the data path on an ESXi host including VMkernel and guest operating system. By default, LRO is enabled in the VMkernel and in the VMXNET3 VM adapters.

Recommendation: As a best practice, consider using large receive offload (LRO) to reduce the CPU overhead required for processing packets that arrive from the network at a high rate.

For further information, review:

- [Understanding TCP Segmentation Offload \(TSO\) and Large Receive Offload \(LRO\) in a VMware environment \(2055140\)](#)
- [Poor TCP performance might occur in Linux VMs with LRO enabled \(1027511\)](#)
- [Large Receive Offload](#)

Virtual Machine Maintenance

During the operational lifecycle of a VM hosting BCA workloads, it's expected that changes will be required. A VM might need to be moved to a different physical datacenter or virtual cluster, where physical hosts are different and a different version of vSphere is installed, or the vSphere platform will be updated to the latest version. In order to maintain best performance and be able to use new features of the physical hardware or vSphere platform, VMware strongly recommends that you:

- Check and upgrade VMware Tools
- Check and upgrade the compatibility (i.e., VM hardware)

Install VMware Tools

VMware Tools is a set of services and components that enable several features in various VMware products for better management and seamless user interactions with guest operating systems.

Open VM Tools (open-vm-tools) is the open-source implementation of VMware Tools for Linux guest operating systems. The open-vm-tools suite is bundled with Linux operating systems (e.g., RHEL 7 and above, OEL 7 and above) and is installed as a part of the OS, eliminating the need to separately install the suite on guest operating systems. VMware Tools generally ship with most of the common Linux distributions and was incorporated into the Linux kernel as of 2.6.33 as the open VM Tools package.

They are also updated via the OS vendor.

```
Setting up open-vm-tools (2:10.2.0-3~ubuntu0.16.04.1) ...
Installing new version of config file /etc/init.d/open-vm-tools ...
Installing new version of config file /etc/pam.d/vmtoolsd ...
Installing new version of config file /etc/vmware-tools/poweroff-vm-default ...
Installing new version of config file /etc/vmware-tools/poweron-vm-default ...
Installing new version of config file /etc/vmware-tools/resume-vm-default ...
Installing new version of config file /etc/vmware-tools/scripts/vmware/network ...
Installing new version of config file /etc/vmware-tools/statechange.subr ...
Installing new version of config file /etc/vmware-tools/suspend-vm-default ...

Configuration file '/etc/vmware-tools/tools.conf'
==> Modified (by you or by a script) since installation.
==> Package distributor has shipped an updated version.
What would you like to do about it ? Your options are:
  Y or I : install the package maintainer's version
  N or O : keep your currently-installed version
  D      : show the differences between the versions
  Z      : start a shell to examine the situation
The default action is to keep your current version.
*** tools.conf (Y/I/N/O/D/Z) [default=N] ? Y
Installing new version of config file /etc/vmware-tools/tools.conf ...
Installing new version of config file /etc/vmware-tools/vm-support
```

FIGURE 27. Updating VMware Tools as Part of an Ubuntu Update

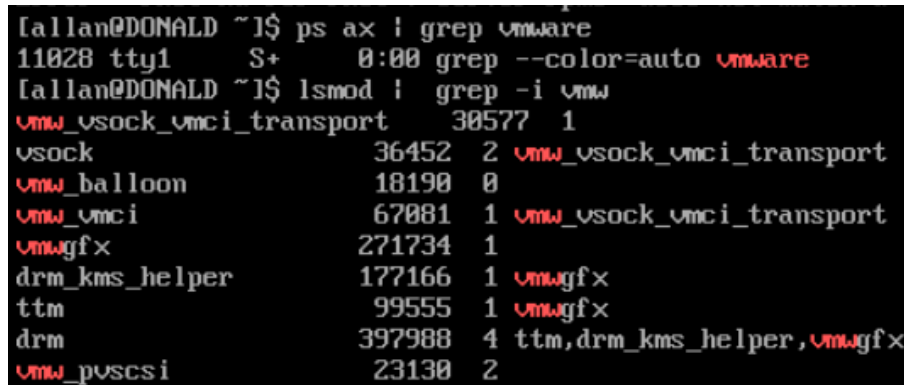
To verify VMware Tools are installed, you can use the following commands in the guest:

```
ps ax | grep vmware
```

The command can show that some processes detect that the server is virtualized (e.g., the manner in which color is or is not displayed).

```
lsmod | grep -i vmw
```

This command provides visibility into things like network and PVSCSI drivers.



```
[allan@DONALD ~]$ ps ax | grep vmware
11028 tty1      S+      0:00 grep --color=auto vmware
[allan@DONALD ~]$ lsmod | grep -i vmw
vmw_vsock_vmci_transport 30577 1
vsock                  36452 2 vmw_vsock_vmci_transport
vmw_balloon            18190 0
vmw_vmci                67081 1 vmw_vsock_vmci_transport
vmwgfx                  271734 1
drm_kms_helper         177166 1 vmwgfx
ttm                     99555 1 vmwgfx
drm                     397988 4 ttm,drm_kms_helper,vmwgfx
vmw_pvscsi              23130 2
```

FIGURE 28. Showing VMware Tools under RHEL

To get the specific package for your distribution of Linux, along with links to installation instructions, see [VMware Tools Operating System Specific Packages \(OSPs\)](#).

Recommendation: As a best practice, ensure VMware Tools are installed and up-to-date to get the best performance from BCA workloads.

For further information, review:

- [Installing VMware Tools](#)
- [Using Open VM Tools](#)
- [VMware Tools Documentation](#)

Upgrade VMware Tools

VMware Tools is a set of services, drivers and modules that enable several features for better management of, and seamless user interactions with, guest operating systems. VMware Tools is comparable to the driver's pack required for physical hardware, but in virtualized environments.

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

VMware Tools can be upgraded in many ways. However, it's extremely important to note that the VMware Tools upgrade is essentially a driver upgrade process and, as such, will influence the core OS. Hence, the upgrade should be done with care, with preliminary testing in a non-production environment.

There are two different version designations for VMware Tools: one is a human-readable number, such as 10.0.7; the other is an internal code, such as 10247. With vSphere 6.5, vSphere web client now displays both variations of the version number as well as the specific type of VMware Tools installed in the guest OS: MSI, OSP, OVT, or Tar Tools.

If you use an open-vm-tools, the VMware Tools status is **guest managed** on the VM summary tab. *Guest managed* means that you cannot use vCenter Server to manage VMware Tools and you cannot use vSphere Update Manager to upgrade VMware Tools.

For example, VM **Oracle1912-OEL83** has open-vm-tools installed and has the message displayed as shown below:

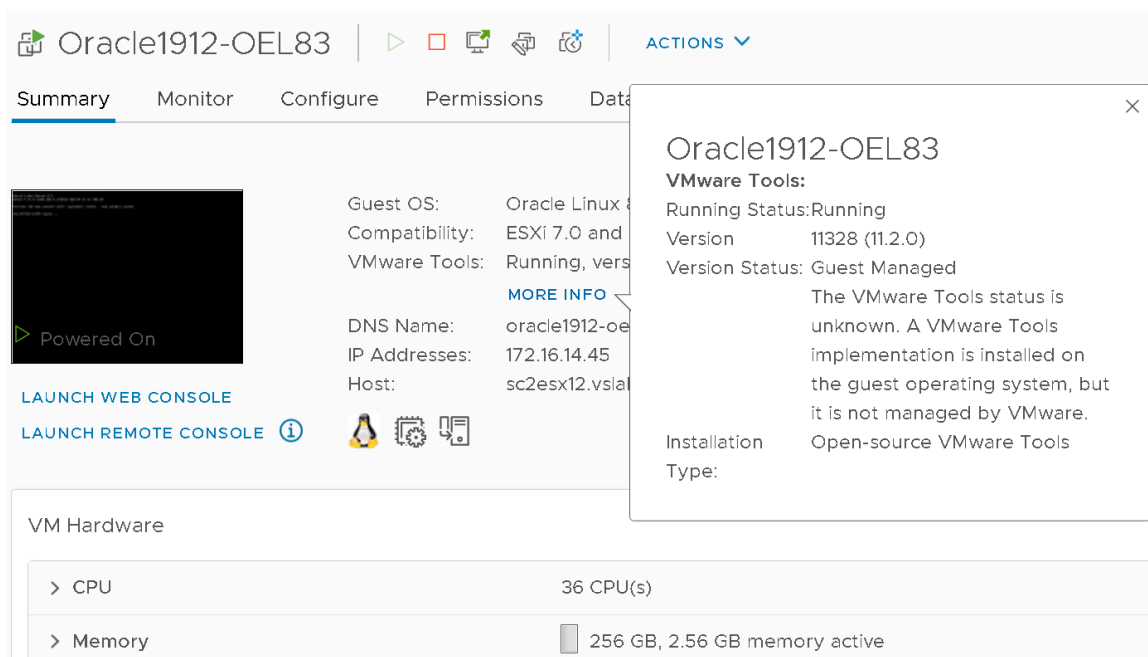


FIGURE 29. Enhanced VMware Tools Information Showing More Detailed Version and Type Data

Recommendation: As a best practice, ensure the latest version of VMware Tools is installed to get the best performance from BCA workloads.

For further information, review:

- [Upgrading VMware Tools](#)
- [Six Methods for Keeping VM Tools Up to Date](#)

Virtual Machine Compatibility

A VM's compatibility determines the hardware version of a VM, which in turn reflects the VM hardware features and functions that the VM supports, which in turn relate to the hardware available on a physical host.

When you create a VM, you can choose either the default hardware version or an earlier version. If you choose the latter, however, this may limit the VM's functionality.

Virtual hardware includes BIOS and EFI, available virtual PCI slots, maximum number of CPUs, maximum memory configuration, and other characteristics. You can upgrade the compatibility level to make additional hardware available to the VM.

For example, to be able to assign more than 1TB of memory, VM compatibility should be at least hardware version 12.

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

VMware recommends upgrading the VM compatibility when new physical hardware is introduced to the environment. Virtual machine compatibility upgrades should be planned and completed with care.

The following procedure is recommended:

- Take a backup of the Oracle or SQL databases and OS.
- Upgrade VMware Tools.
- Validate that no misconfigured or inaccessible devices (e.g., CD-ROM, floppy) are present.
- Use vSphere web client to upgrade VM hardware to the desired version.

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

NOTE: C# client has no support for the VM compatibility level 9 and above. All new features introduced after this version are exposed to the vSphere web client only.

Recommendation: As a best practice, upgrade VM compatibility when new physical hardware is introduced to the environment to get the best performance from BCA workloads. Ensure that the VM hardware version is at the minimum, set to the default version supported by the ESXi version.

For further information, review:

- [Virtual Machine Compatibility](#)
- [Hardware Features Available with Virtual Machine Compatibility Settings](#)
- [Virtual machine hardware versions \(1003746\)](#)
- [Upgrading a VM to the latest hardware version \(multiple versions\) \(1010675\)](#)
- [ESXi/ESX hosts and compatible VM hardware versions list \(2007240\)](#)

Timekeeping in Virtual Machine

Most operating systems track the passage of time by configuring the underlying hardware to provide periodic interrupts. The rate at which those interrupts are configured to arrive varies for different operating systems. High timer-interrupt rates can incur overhead that affects a VM's performance. The amount of overhead increases with the number of vCPUs assigned to a VM. The impact of these high timer-interrupt rates can lead to time-synchronization errors.

In the VMware Tools control panel, the time synchronization check box is unselected, but you might experience these symptoms:

- When you suspend a VM, it synchronizes the time to adjust it to the host the next time it is resumed.
- Time is resynchronized when you migrate the VM using vSphere vMotion, take a snapshot, restore to a snapshot, shrink the virtual disk, or restart the VMware Tools service in the VM (including rebooting the VM).

NTP is an industry standard and ensures accurate timekeeping in your guest. It may be necessary to open the firewall (UDP 123) to allow NTP traffic.

Recommendation: As a best practice, use NTP instead of VMware Tools periodic time synchronization.

For further information, review:

- [Timekeeping best practices for Linux guests](#)
- [Timekeeping best practices for Windows, including NTP](#)

Time Synchronization

The default behavior of adjusting a VM's clock through VMware Tools during various VM operations is now exposed in the VM's properties as a checkbox. As of vSphere 7.0 U1, the VMware Tools control panel provides two check-boxes that enable or disable periodic and one-off time synchronization.

- The first box, **Synchronize at Startup and Resume (recommended)**, is checked by default to reflect the default behavior. Unchecking this box effectively instructs VMware Tools to assume that the administrator has intentionally applied the settings documented in KB 1189.
- The second box, **Synchronize time periodically**, enables VMware Tools to periodically poll and resync with the host's clock if the first box is checked.

For example, on VM **Oracle1912-OEL83**, the VMware Tools control panel details are as shown below:

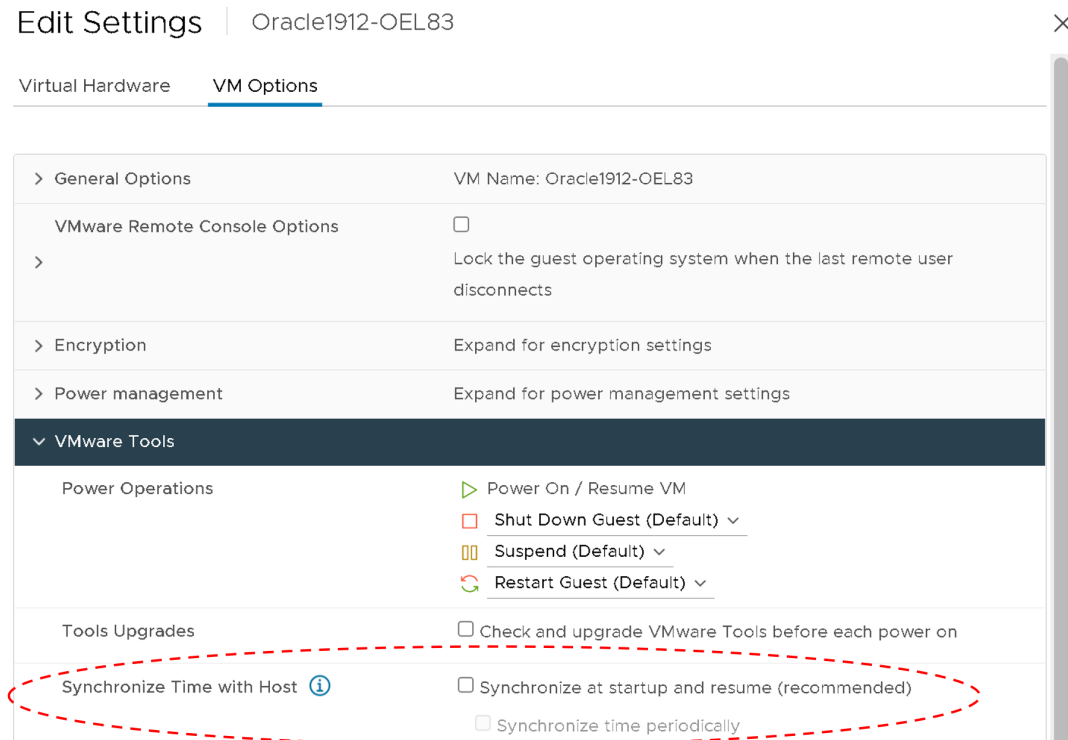


FIGURE 30. VMware Tools control panel view

VMware recommends unchecking this box for VMs running guest operating systems which have a native, reliable time-synchronization and correction mechanism.

A Windows VM joined to an active directory infrastructure is one such use case for which these boxes should be unchecked.

With the release of vSphere 7.0 U2, the VMware Time Provider (vmwTimeProvider) provides support for precision time protocol (PTP) for VMs. This is available only for a VM and only when the VM's hardware compatibility level is at least VM hardware version 17.

By default, VMware Tools will not reset a VM's clock backwards, except under the following conditions:

- The advance configuration settings in KB 1189 have **not** been applied to the VM, **and**...
- The following advanced configuration settings (not mentioned in KB 1189) have been applied to the VM:
 - synchronize.restore.backward = TRUE
 - synchronize.resume.disk.backward = TRUE
 - synchronize.tools.startup.backward = TRUE

VMware Tools cannot protect a VM against inaccurate CMOS RTC time on a host. Virtualized Windows guest OS VMs will always read and feed off of their host's CMOS clock on boot-up/power-cycle.

This is one of the reasons VMware continues to highly recommend that vSphere administrators pay close attention to possible CMOS-induced host's clock inaccuracy **and** to always synchronize their host's time with a known reliable and accurate external time source (preferably one in stratum 1).

Recommendation: As a best practice, disable time synchronization for BCA workloads.

For further information, review:

- [Disabling Time Synchronization \(1189\)](#)

VM Configuration Maximums

This configuration maximums tool provides the recommended configuration limits for VMware products. When VMs are configured and deployed, it is highly recommended that VMs stay within the limits supported by the platform. The limits presented in the tool are tested, recommended limits and are fully supported by VMware.

Recommendation: As a best practice, always check the configuration maximums tool to understand the configuration limits for VMware products.

For further information, review:

- [vSphere Configuration Limits](#)
- [Virtual machine hardware versions \(1003746\)](#)

Virtual Machine Security Features

Virtual Machine Encryption

vSphere 6.5 introduced virtual machine encryption which is a VM-agnostic method of encryption for VMs that is scalable, easy to implement, and easy to manage. Encryption protects not only your VM but also VM disks and other files. You set up a trusted connection between vCenter Server and a key management server (KMS). vCenter Server can then retrieve keys from the KMS as needed.

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

VM encryption does not impose any specific hardware requirements and using a processor that supports the AES-NI instruction set speeds up the encryption and 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 overhead largely depends on two aspects:

- The efficiency of implementation of the feature or 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.

Recommendation: As a best practice, consider virtual machine encryption for security considerations.

For further information, review:

- [How vSphere Virtual Machine Encryption Protects Your Environment](#)
- [VMware vSphere VIRTUAL MACHINE ENCRYPTION PERFORMANCE VMware vSphere 6.5](#)
- [vSphere 6.5: VM and vSAN Encryption FAQ](#)

Virtual Machine UEFI Secure Boot

UEFI secure boot is a security standard that helps ensure your PC boots using only software that is trusted by the PC manufacturer. For certain VM hardware versions and operating systems, you can enable secure boot just as you can for a physical machine.

In an OS that supports UEFI secure boot, each piece of boot software is signed, including the bootloader, the OS kernel, and OS drivers.

Recommendation: As a best practice for purposes of security, consider virtual machine UEFI secure boot.

For further information, review:

- [Enable or Disable UEFI Secure Boot for a Virtual Machine](#)

Summary

Architecting Business Critical Applications on VMware Hybrid Cloud provides comprehensive and prescriptive guidance that operators, administrators, architects, business owners and other stakeholders can use as a building block for reliably and successfully moving their mission-critical applications from traditional platforms to any of the various hybrid clouds offerings on the market today.

The best practices and guidelines discussed in the body of this document are summarized in this section.

SECTION	SUB-SECTION	RECOMMENDATION FOR BEST PRACTICES	REFERENCES
ESXi Host Configuration	BIOS/UEFI and Firmware Versions	Update the BIOS/UEFI firmware on the physical server that is running critical systems to the latest version and make sure all the I/O devices have the latest supported firmware version.	<ul style="list-style-type: none"> • Checking your firmware and BIOS levels to ensure compatibility with ESX/ESXi (1037257)
	BIOS/UEFI Settings	Follow the BIOS/UEFI settings recommended for high performance environments (when applicable) in this section.	<ul style="list-style-type: none"> • Configuring the BIOS Boot Settings
	Power Management	Configure the server BIOS/UEFI to pass power management to ESXi (OS control) and change the default Balanced power scheme to High Performance.	<ul style="list-style-type: none"> • Host Power Management Policies
ESXi CPU Configuration	Physical and Virtual CPU / Core	Informational only	<ul style="list-style-type: none"> • Setting the number of cores per CPU in a virtual machine (1010184) • Administering CPU Resources • Multi-core processor
	Hyper-Threading	Enable Hyper-Threading in the BIOS/UEFI so that ESXi can take advantage of this technology.	<ul style="list-style-type: none"> • Hyperthreading • Hyper-threading
	Understanding NUMA	Informational only	<ul style="list-style-type: none"> • Non-uniform memory access

	Using NUMA Best Practices	Leave NUMA enabled in the BIOS for NUMA servers to be NUMA aware.	<ul style="list-style-type: none"> • Node Interleaving: Enable or Disable?
	VMware ESXi and ESXTOP	Ensure that the physical NUMA topology is exposed correctly to an ESXi host before proceeding with configuring BCA workloads on the vSphere cluster.	<ul style="list-style-type: none"> • Intel Cluster-on-Die (COD) Technology, and VMware vSphere 5.5 U3b and 6.x (2142499) • Performance Monitoring Utilities: resxtop and esxtop • Using esxtop to Troubleshoot Performance Problems • Using esxtop to identify storage performance issues for ESX / ESXi (multiple versions) (1008205) • NUMA Deep Dive Part 1: From UMA to NUMA • NUMA Deep Dive Part 5: ESXi VMkernel NUMA Constructs • Optimizing Application Performance in Large Multi-core Systems • Using NUMA Systems with ESXi • NUMA Blogs • VMware Communities: Interpreting esxtop Statistics • NUMA Observer - VMware Fling
ESXi Memory Configuration	Memory Overcommit Techniques	Informational only	<ul style="list-style-type: none"> • Understanding Memory Resource Management in VMware vSphere • Administering Memory Resources
	Balloon Driver Concepts	Informational only	<ul style="list-style-type: none"> • Memory Balloon Driver
	vSphere 2M Large Memory Pages	Informational only	<ul style="list-style-type: none"> • Transparent Page Sharing (TPS) in hardware MMU systems (1021095) • Use of large pages can cause memory to be fully allocated (1021896) • Support for Large Page Sizes
	1GB Large Memory Pages	Use 1GB large memory page on ESXi servers with adequate memory capacity and on guest operating systems capable of taking advantage of the 1GB large memory page.	<ul style="list-style-type: none"> • Backing Guest vRAM with 1GB Pages • 1 GB Large Memory Pages

	Persistent Memory	Consider using persistent memory technology to accelerate BCA workloads.	<ul style="list-style-type: none"> • VM nvdimm config options for NUMA (78094) • Persistent Memory • Accelerating applications performance with virtualized Persistent Memory • Persistent Memory Performance in vSphere 6.7 • Persistent Memory Performance on vSphere 6.7 Performance Study • Persistent Memory Performance in vSphere 6.7 with Intel Optane DC persistent memory Performance Study • Announcing VMware vSphere Support for Intel® Optane™ Persistent Memory Technology • What Is Persistent Memory? • Virtualized Persistent Memory with VMware vSphere 6.7 and HPE ProLiant Gen10 Servers • DELL EMC PowerEdge Persistent Memory (PMem) Support (54444) • Hewlett Packard Enterprise Servers Persistent Memory (PMem) Support (54445) • How to simulate Persistent Memory (PMem) in vSphere 6.7 for educational purposes? • vSphere Support for Intel's Optane Persistent Memory (PMEM) (67645)
ESXi Storage Configuration	vSphere Storage Options	Informational only	<ul style="list-style-type: none"> • Understanding VMFS Datastores • Increasing the default value that defines the maximum number of NFS mounts on an ESXi/ESX host (2239) • Understanding Network File System Datastores • Best Practices for running VMware vSphere on Network Attached Storage • Best Practices for Running VMware vSphere® on Network-Attached Storage (NAS) (2013) • Performance Characterization of VMFS and RDM Using a SAN • Migrating virtual machines with Raw Device Mappings (RDMs) (1005241) • Difference between Physical compatibility RDMs and Virtual compatibility RDMs (2009226) • vVols Concepts • What's New vSphere Virtual Volumes • VMware vSAN Documentation
	VMware Virtual Disk Provisioning Policies	Informational only	<ul style="list-style-type: none"> • VMware virtual disk provisioning policies • Determining if a VMDK is zeroedthick or eagerzeroedthick (1011170) • Cloning and converting virtual machine disks with vmkfstools (1028042) • Pure Storage Virtual Machine and Guest Configuration • Pure Storage ZeroedThick or Eagerzeroedthick? That is the question

	VMware Multi-Writer Attribute for Shared VMDKs	Informational only	<ul style="list-style-type: none"> • Enabling or disabling simultaneous write protection provided by VMFS using the multi-writer flag (1034165) • Using Oracle RAC on a vSphere 6.x vSAN Datastore (2121181) • Attempts to enable the multi-writer virtual disk option on an NFS datastore fail (2147691)
	Clustered VMDK support	Informational only	<ul style="list-style-type: none"> • Clustered VMDK support • Hosting Windows Server Failover Cluster (WSFC) with shared disks on VMware vSphere: Doing it right!
	vSphere APIs for Array Integration (VAAI)	Informational only	<ul style="list-style-type: none"> • Frequently Asked Questions for vStorage APIs for Array Integration (1021976) • Storage Hardware Acceleration • VMware vSphere Storage APIs Array Integration (VAAI) • VAAI Comparison – Block versus NAS
	Storage Policy-Based Management (SPBM)	Informational only	<ul style="list-style-type: none"> • Storage Policy-Based Management
	VMware vSAN Storage Policy		<ul style="list-style-type: none"> • vSAN Default Storage Policy • VMware vSAN Design Guide
	Automatic Space Reclamation (UNMAP)	Upgrade current VMFS datastores to VMFS6 to take advantage of the UNMAP feature.	<ul style="list-style-type: none"> • Storage Space Reclamation • WHAT'S NEW IN PERFORMANCE VMware vSphere 6.5
	4kn support	vSphere and vSAN will expose 512n to the guest OS. As a best practice, consider 4kn local storage for capacity reasons. Ensure alignment of guest I/O to 4k if running on 512e/4kn drives with vSAN.	<ul style="list-style-type: none"> • FAQ: Support statement for 512e and 4K Native drives for VMware vSphere and vSAN (2091600) • Support for 4Kn HDD • What's New in Performance for VMware vSphere 6.7? • Device Sector Formats
	Storage Other Feature	Informational only	<ul style="list-style-type: none"> • What's New in VMware vSphere 6.5 • WHAT'S NEW IN PERFORMANCE VMware vSphere 6.5 • What's New in Performance for VMware vSphere 6.7? • What's New in vSphere 7 Core Storage • Performance Characterization of NVMe-oF in vSphere 7.0 U1 • vSphere 7 – Storage vMotion Improvements
ESXi Network Configuration	Virtual Network Concepts	Informational only	<ul style="list-style-type: none"> • Introduction to vSphere Networking

	Multi-NIC vMotion for High Memory Workload	Consider using multi-NIC vMotion for high memory workloads by upgrading to vSphere 7.x	<ul style="list-style-type: none"> • Multiple-NIC vMotion in vSphere (2007467) • VMware vSphere 5.1 vMotion Architecture, Performance and Best Practices Performance Study • vMotion Innovations in VMware vSphere 7.0 U1 Performance Study • vSphere 7 – vMotion Enhancements • Migration with vMotion • Virtual Machines using large pages can temporarily become unresponsive after vMotion (2144984) • Virtual machine performance degrades while a vMotion is being performed (2007595) • Understanding and troubleshooting vMotion (1003734) • vMotion Blogs
	Jumbo Frames for vSphere vMotion Interfaces	Consider using jumbo frames for vSphere vMotion interfaces to accelerate vMotion operation for memory-intensive NCA workloads.	<ul style="list-style-type: none"> • Enabling Jumbo Frames on virtual distributed switches (1038827) • What's the Big Deal with Jumbo Frames?
	Load Balancing on vSphere Standard Switch and Distributed Switch	Evaluate the various load-balancing options on vSphere Standard Switch and Distributed Switch and choose Route Based on Originating Virtual Port unless there is a need otherwise.	<ul style="list-style-type: none"> • Route Based on Originating Virtual Port
	RDMA (Remote Direct Memory Access) over Converged Ethernet (RoCE)	Consider using PVRDMA drivers for a VM's virtual network adapter on ESXi servers with RDMA-capable host channel adapters (HCA) to accelerate workload performance.	<ul style="list-style-type: none"> • Remote Direct Memory Access for Virtual Machines • Performance of RDMA and HPC Applications in Virtual Machines using FDR InfiniBand on VMware vSphere
	Network Other Features	Informational only	<ul style="list-style-type: none"> • What's New in Performance for VMware vSphere 6.7 - Network
vSphere Security Configuration	vSphere Security Features	Informational only	<ul style="list-style-type: none"> • Encrypted vSphere vMotion • About vSphere Security
	Side-Channel Vulnerability Mitigation and New CPU Schedule	Informational only	<ul style="list-style-type: none"> • RedHat KB L1TF - L1 Terminal Fault Attack - CVE-2018-3620 & CVE-2018-3646 • VMware response to 'L1 Terminal Fault - VMM' (L1TF - VMM) Speculative-Execution vulnerability in Intel processors for vSphere: CVE-2018-3646 (55806) • VMware Performance Impact Statement for 'L1 Terminal Fault - VMM' (L1TF - VMM) mitigations: CVE-2018-3646 (55767) • Implementing Hypervisor-Specific Mitigations for Microarchitectural Data Sampling (MDS) Vulnerabilities (CVE-2018-12126, CVE-2018-12127, CVE-2018-12130, and CVE-2019-11091) in vSphere (67577) • Which vSphere CPU Scheduler to Choose • New Scheduler Option for vSphere 6.7 U2 • Performance of vSphere 6.7 Scheduling Options

Virtual Machine CPU Configuration	Allocating vCPU	At initial sizing, the total number of vCPUs assigned to all VMs should be no more than the total number of physical cores (not logical cores) available on the ESXi host machine.	<ul style="list-style-type: none"> • Determining if multiple virtual CPUs are causing performance issues (1005362)
	vNUMA, corespersocket and PreferHT	Follow recommendations in this section	<ul style="list-style-type: none"> • Setting the number of cores per CPU in a virtual machine (1010184) • Setting corespersocket can affect guest OS topologies (81383) • Virtual Machine vCPU and vNUMA Rightsizing – Guidelines • Does corespersocket Affect Performance? • What is PreferHT and When to Use It • Virtual Machines with preallocated memory might be placed NUMA remote at random power-ons (76362)
	CPU Hot Plug and Hot Add	For Linux, enable CPU hot plug for VMs that do not need vNUMA optimization. For Windows versions prior to Windows 10 Build 20348, enabling CPU hot plug will create additional fake nodes to accommodate those potential vCPUs.	<ul style="list-style-type: none"> • vNUMA is disabled if VCPU hot plug is enabled (2040375) • CPU Hot Add Performance in vSphere 6.7 • Impact of CPU Hot Add on NUMA scheduling • Enabling vCPU HotAdd creates fake NUMA nodes on Windows (83980) • CPU HotAdd for Windows VMs: How BADLY Do You Want It?
	CPU Affinity	Avoid using CPU affinity in production because it limits the hypervisor's ability to efficiently schedule vCPUs on the physical server and causes poor performance. It also disables the ability to vMotion a VM.	<ul style="list-style-type: none"> • Scheduler operation when using CPU Affinity (2145719)
	Latency Sensitive Setting	Avoid enabling Latency Sensitive settings for BCA workloads unless test have shown it should otherwise be enabled.	<ul style="list-style-type: none"> • Best Practices for Performance Tuning of Latency-Sensitive Workloads in vSphere VMs • Deploying Extremely Latency-Sensitive Applications in VMware vSphere 5.5 Performance Study
	Per Virtual Machine EVC Mode	Recommendation: As a best practice, consider per VM EVC Mode to avoid downtime while migrating a VM between datacenters or vCenters, or to a public cloud, such as VMware Cloud. This is to mitigate issues during migration and is not performance related	<ul style="list-style-type: none"> • Virtual EVC per VM • Enhanced vMotion Compatibility as a Virtual Machine Attribute

Virtual Machine Memory Configuration	Memory Sizing Considerations	Informational only	<ul style="list-style-type: none"> • vSphere Resource Management • Understanding vSphere Active Memory • Understanding Memory Resource Management in VMware vSphere Performance Study
	Memory Reservation	Consider setting memory reservations for memory intensive production BCA workloads on a case-by-case basis.	<ul style="list-style-type: none"> • vSphere Resource Management • Unswapping swapped pages
	Memory Hot Plug	Rely more 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 be implemented in the VM template used to deploy the BCA workload.	<ul style="list-style-type: none"> • Change Memory Hot Add Settings
Virtual Machine Storage Configuration	VM PVSCSI Storage Controller and PVSCSI/VMDK Queue Depth	Increase the PVSCSI and VMDK queue depth to its maximum in order to push larger I/O bandwidth, unless the underlying storage vendor recommends otherwise. Use multiple PVSCSI adapters with VMDKs spread across the PVSCSI controllers for load-balancing and fairness.	<ul style="list-style-type: none"> • Large-scale workloads with intensive I/O patterns might require queue depths significantly greater than Paravirtual SCSI default values (2053145) • Configuring disks to use VMware Paravirtual SCSI (PVSCSI) controllers (1010398) • vSphere 7.0 Configuration Limits - VMs • Which vSCSI controller should I choose for performance? • SCSI, SATA, and NVMe Storage Controller Conditions, Limitations, and Compatibility • PVSCSI and Large IO's
	VM vNVMe Storage Controller	Consider using vNVMe controllers for VM storage on low latency SSD drives on all-flash storage arrays.	<ul style="list-style-type: none"> • Using Virtual NVMe from ESXi 6.5 and virtual machine Hardware Version 13 (2147714) • Add an NVMe Controller
	Partition Alignment	Follow recommendations in this section	<ul style="list-style-type: none"> • Recommendations for Aligning VMFS Partitions • Guest OS Partition Alignment
	VMDK File Layout	Informational only	<ul style="list-style-type: none"> • VMware virtual disk provisioning policies
	Virtual Disks Hot Add and Hot Remove	Informational only	<ul style="list-style-type: none"> • Add a Hard Disk to a Virtual Machine
	Virtual Disks Hot Extend	Informational only	<ul style="list-style-type: none"> • Increasing the size of a virtual disk (1004047)

	VM Snapshot	Follow recommendations in this section	<ul style="list-style-type: none"> • High co-stop (%CSTP) values seen during virtual machine snapshot activities (2000058) • Virtual machine becomes unresponsive or inactive when taking memory snapshot (1013163) • Overview of virtual machine snapshots in vSphere (1015180) • Snapshot removal stops a virtual machine for long time (1002836) • Managing snapshots in vSphere Web Client (2032907) • Understanding VM snapshots in ESXi (1015180) • Best practices for using snapshots in the vSphere environment (1025279) • VMware Snapshots • When and why do we “stun” a virtual machine?
	BCA Workloads on VMware vSAN	Follow recommendations in this section	<ul style="list-style-type: none"> • Should you be using Eager Zero Thick on vSAN (or VMFS)?
Virtual Machine Network Configuration	Virtual Networking Best Practices	Follow recommendations in this section	<ul style="list-style-type: none"> • Choosing a network adapter for your virtual machine (1001805) • Large packet loss at the guest operating system OS level on the VMXNET3 vNIC in ESXi (2039495) • VMXNET3 resource considerations on a Linux virtual machine that has vSphere DirectPath I/O with vMotion enabled (2058349)
	Interrupt Coalescing	Consider disabling virtual network interrupt coalescing for low latency requirements.	<ul style="list-style-type: none"> • Low throughput for UDP workloads on Windows virtual machines (2040065) • Performance Best Practices for VMware vSphere 7.0 • Virtual network interrupt coalescing
	Receive Side Scaling (RSS)	Consider enabling RSS to increase workload performance.	<ul style="list-style-type: none"> • What is Receive Side Scaling (RSS), and how do I configure it in RHEL? • RSS and multiqueue support in Linux driver for VMXNET3 (2020567) • Windows Receive Side Scaling (RSS) • Windows Enabling Receive Side Scaling
	TCP Segmentation Offload	Consider using TCP segmentation offload (TSO) in VMkernel network adapters and VMs to improve the network performance in workloads that have severe latency requirements.	<ul style="list-style-type: none"> • Understanding TCP Segmentation Offload (TSO) and Large Receive Offload (LRO) in a VMware environment (2055140) • TCP Segmentation Offload
	Large Receive Offload	Consider using large receive offload (LRO) to reduce the CPU overhead for processing packets that arrive from the network at a high rate.	<ul style="list-style-type: none"> • Understanding TCP Segmentation Offload (TSO) and Large Receive Offload (LRO) in a VMware environment (2055140) • Poor TCP performance might occur in Linux virtual machines with LRO enabled (1027511) • Large Receive Offload

<i>Virtual Machine Maintenance</i>	Install VMware Tools	Ensure VMware Tools are installed to get the best performance from BCA workload.	<ul style="list-style-type: none"> • Installing VMware Tools • Using Open VM Tools • VMware Tools Documentation
	Upgrade VMware Tools	Ensure the latest version of VMware Tools is installed to get the best performance from BCA workloads.	<ul style="list-style-type: none"> • Upgrading VMware Tools • Six Methods for Keeping VM Tools Up to Date
	Virtual Machine Compatibility	Upgrade VM compatibility when new physical hardware is introduced to the environment in order to get the best performance from BCA workloads.	<ul style="list-style-type: none"> • Virtual Machine Compatibility • Hardware Features Available with Virtual Machine Compatibility Settings
	Virtual Machine Hardware version	Ensure that the VM hardware version is at the minimum, set to the default version supported by the ESXi version.	<ul style="list-style-type: none"> • Virtual machine hardware versions (1003746) • Upgrading a virtual machine to the latest hardware version (multiple versions) (1010675) • ESXi/ESX hosts and compatible virtual machine hardware versions list (2007240)
	Timekeeping in Virtual Machine	Use NTP instead of VMware Tools periodic time synchronization.	<ul style="list-style-type: none"> • Timekeeping best practices for Linux guests • Timekeeping best practices for Windows, including NTP
	Time Synchronization	Disable time synchronization for BCA workloads.	<ul style="list-style-type: none"> • Disabling Time Synchronization (1189)
	VM Configuration Maximums	Informational only	<ul style="list-style-type: none"> • vSphere Configuration Limits • Virtual machine hardware versions (1003746)
<i>Virtual Machine Security Features</i>	Virtual Machine Encryption	Consider virtual machine encryption for security considerations during the design phase if needed	<ul style="list-style-type: none"> • How vSphere Virtual Machine Encryption Protects Your Environment • VMware vSphere VIRTUAL MACHINE ENCRYPTION PERFORMANCE VMware vSphere 6.5 • vSphere 6.5: VM and vSAN Encryption FAQ
	Virtual Machine UEFI Secure Boot	For purposes of security, consider virtual machine UEFI secure boot.	<ul style="list-style-type: none"> • Enable or Disable UEFI Secure Boot for a Virtual Machine

Appendix

Examples of vNUMA Configuration

vNUMA Configuration Examples

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

Standard-sized VM Configuration Examples

This section will cover standard sized VMs used to host BCA workloads. Use the configurations examples listed in the **Table 7** to properly assign vCPUs to a VM.

CONFIGURATION #	DESIRED VM CONFIGURATION	VM VCPU #	VM CORES PER SOCKET	VM SOCKET	ADVANCED SETTINGS	DIFFERENT BETWEEN 6.0/6.5	VNUMA	MEMORY
1	8 vCPU, 32 GB RAM	8	8	1	NO	NO	0	32
2	10 vCPU, 64 GB RAM	10	10	1	NO	YES	0	96
3	16 vCPU, 128 GB RAM	16	8	2	NO	NO	2	128

TABLE 7. Standard VM Configuration (Recommended vCPU Settings for Different Numbers of vCPUs)

Configuration #1 covers all use cases where the assigned number of vCPUs per VM is eight⁷ or below and the required amount of memory stays within one pNUMA node.

Expected behavior for such configuration: no vNUMA topology will be exposed to a VM. The ESXi CPU scheduler will execute all vCPUs inside of one pNUMA node.

Configuration #2 covers use cases where the number of vCPUs is nine or more but lower than the number of physical cores in a pNUMA of a server (12, in this scenario).

The expected behavior for such a VM will be to stay within one pNUMA with no vNUMA topology exposed to a VM. For a VM deployed on vSphere version 6.0 and below, its mandatory that number of cores per socket will be equal to the number of vCPU assigned to a VM, with only one socket exposed to a VM. Otherwise, an undesired wide NUMA configuration will be created.

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

⁷ Maximum number of vCPU to expose vNUMA is not reached.

Advanced vNUMA VM Configuration Examples

These special configurations are listed for reference and, if not required, one of the three configuration options from **Table 7** should be used. All configuration listed in the **Table 8** will require adding advanced settings⁸ for a VM being configured. Do not modify host-level advanced NUMA settings.

CONFIGURATION #	DESIRED VM CONFIGURATION	VM vCPU #	VM CORES PER SOCKET	VM SOCKET	ADVANCED SETTINGS	DIFFERENT BETWEEN 6.0/6.5	vNUMA	MEMORY
1	8vCPUs, 128 GB RAM	8	4	2	YES	YES	2	128
2	20vCPU, 64 GB RAM	20	20	1	YES	NO	1	64
3	10vCPU, 64 GB RAM	10	5	2	YES	YES	2	64

TABLE 8. Advanced vNUMA VM Configurations (Recommended vCPU Settings)

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

In order to enable the desired vNUMA topology, the advanced setting *vcpu.maxPerMachineNode*⁹ should be used which will specify the maximum number of vCPUs that will be scheduled inside one pNUMA. Set this value to the half of the total vCPUs assigned to a VM (it will be five in the example used):

$$numa.vcpu.maxPerMachineNode = (\text{Number of vCPU assigned to a VM})/2 = 4$$

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

$$cpuid.coresPerSocket = numa.vcpu.maxPerMachineNode = 4$$

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

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

$$numa.vcpu.preferHT = True$$

$$cpuid.coresPerSocket = 20$$

⁸ Use this procedure to add a VM advanced settings: <https://docs.vmware.com/en/VMware-vSphere/6.7/com.vmware.vsphere.vcenterhost.doc/GUID-62184858-32E6-4DAC-A700-2C80667C3558.html>. Do not modify a .vmx file manually!

⁹ Despite being often mentioned, *numa.vcpu.min* will have no effect as the size of pNUMA is still enough to accommodate all vCPUs. This advanced setting is useful on old processors where total number of cores in a pNUMA was lower than eight.

Based on our example, the size of pNUMA will be set for 24 and a VM with 20 vCPUs will be placed in one NUMA node. Due to the number of cores per socket extended to the total vCPUs assigned, all the cores will be presented as sharing the same processor cache (opposite to cores per socket set to 1, where each core will have separate cache) which could lead to the cache optimization.¹⁰ This configuration requires extensive testing and monitoring if implemented. Use with caution.

Configuration #3 Covers use cases where it's desired to split the vCPUs between NUMA nodes even if the total vCPU count on a VM will feed into one pNUMA. It might optimize the work of the ESXi NUMA- and CPU schedulers in a cluster with high VM population ratio. This configuration may make sense for a physical CPU with core count being nine and above. This configuration requires testing with the exact production workload and should be treated as exception.

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

For vSphere versions 6.5 and later, in addition to configuring **Cores per Socket**, the following VM advanced setting needs to be added to disable the autosizing:

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

Special consideration should be given for a cluster with heterogeneous hardware or when moving a VM hosting BCA workloads to a different vSphere compute cluster.

It's important to consider the following for this kind of configuration change: vNUMA topology for a VM is, by default, generated once and will not be changed upon vMotion or a power cycle of a VM.

If a VM resides in a cluster with heterogeneous hardware and a different size of pNUMA across ESXi hosts, either enable reconfiguration of vNUMA topology by each power cycle of a VM¹¹ or configure static vNUMA representation following the Configuration#1 of the advanced vNUMA configurations (listed as the Configuration#1 in **Table 8**). In general, such cluster configurations should be strictly avoided in production environments.

If a VM needs to be migrated to another cluster or moved to a new hardware, one-time reconfiguration of the vNUMA topology might be required. In this case, it's recommended to review the vNUMA topology required, recheck the hardware specification of physical hosts and then use **Tables 7 and 8** along with the description of use cases to make a decision as to what vNUMA topology should be used. If one of the standard vNUMA topologies is not sufficient, after a VM is migrated, make a slight change to a vCPU assignment (e.g., add or remove two cores) and then revert back to the original configuration. This change will instruct an ESXi host to re-create the vNUMA topology. Double check the final vNUMA topology.

Check vNUMA Topology Exposed to a VM

After the desired vNUMA topology is defined and configured, power on a VM and recheck how the final topology looks. The following command on the ESXi host hosting the VM could be used.¹²

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

¹⁰ [NUMA Deep Dive Part 5: ESXi VMkernel NUMA Constructs](#)

¹¹ Add `numa.autosize.once = FALSE` and `numa.autosize = TRUE` to a VM configuration. Use with caution, as it's not normally expected that the NUMA topology could be dynamically adjusted in the guest OS.

¹² Special thanks to V.Bondizo, Sr. Staff TSE, VMware, for sharing this vmdumper command.

```
DICT          numvcpus = "16"
DICT          memSize = "4096"
DICT          displayName = "Oleg-CentOS7-FCI-Node1"
DICT numa.autosize.vcpu.maxPerVirtualNode = "4"
DICT          numa.autosize.cookie = "40001"
DICT          cpuid.coresPerSocket = "8"
numaHost: NUMA config: consolidation= 1 preferHT= 0
numa: coresPerSocket= 8 maxVcpusPerVPD= 8
numaHost: 16 VCPUs 2 VPDs 2 PPDs
numaHost: VCPU 0 VPD 0 PPD 0
numaHost: VCPU 1 VPD 0 PPD 0
numaHost: VCPU 2 VPD 0 PPD 0
numaHost: VCPU 3 VPD 0 PPD 0
numaHost: VCPU 4 VPD 0 PPD 0
numaHost: VCPU 5 VPD 0 PPD 0
numaHost: VCPU 6 VPD 0 PPD 0
numaHost: VCPU 7 VPD 0 PPD 0
numaHost: VCPU 8 VPD 1 PPD 1
numaHost: VCPU 9 VPD 1 PPD 1
numaHost: VCPU 10 VPD 1 PPD 1
numaHost: VCPU 11 VPD 1 PPD 1
numaHost: VCPU 12 VPD 1 PPD 1
numaHost: VCPU 13 VPD 1 PPD 1
numaHost: VCPU 14 VPD 1 PPD 1
numaHost: VCPU 15 VPD 1 PPD 1
```

FIGURE 31. Checking NUMA Topology with the vmdumper Command

Resources

- [Performance Best Practices for VMware vSphere 7.0](#)
- [Performance Best Practices Guide for vSphere 6.7](#)
- [Performance Best Practices for VMware vSphere 6.5](#)
- [Performance Best Practices Guide for vSphere 6.0](#)
- [Troubleshooting a VM that has stopped responding: VMM and Guest CPU usage comparison \(1017926\)](#)
- [Understanding vSphere Active Memory](#)
- [Memory Management and the VMkernel](#)
- [Tuning ESX/ESXi for better storage performance by modifying the maximum I/O block size \(1003469\)](#)
- [Large I/Os are split into 64 KB units when using LSILogic Driver \(9645697\)](#)
- [Controlling LUN queue depth throttling in VMware ESX/ESXi \(1008113\)](#)
- [Configuring advanced driver module parameters in ESX/ESXi \(1017588\)](#)
- [Increasing the disk timeout values for a Linux 2.6 VM \(1009465\)](#)
- [Changing the queue depth for QLogic, Emulex, and Brocade HBAs \(1267\)](#)
- [Checking the queue depth of the storage adapter and the storage device \(1027901\)](#)
- [Storage Protocol Comparison](#)
- [Adaptive Queueing vs. Storage I/O Control](#)
- [DQLEN changes, what is going on?](#)
- [Understanding VMware ESXi Queueing and the FlashArray](#)
- [Troubleshooting Storage Performance in vSphere – Part 1 – The Basics](#)
- [Troubleshooting Storage Performance in vSphere – Part 2](#)
- [Troubleshooting Storage Performance in vSphere \(Part 3\) – SSD Performance](#)
- [Troubleshooting Storage Performance in vSphere – Storage Queues](#)
- [Queues, Queues and more Queues](#)
- [VMware Application Blog](#)
- [VMware Performance Team blog](#)
- [VMware vSphere Blog](#)

Acknowledgements

Author:

- **Sudhir Balasubramanian** – Senior Staff Solution Architect, Oracle Applications

Thanks to the following people for their contributions:

- **Oleg Ulyanov** – Solutions Architect, Microsoft Applications
- **Deji Akomolafe** – Staff Solutions Architect, Microsoft Applications

Thanks to the following people for their inputs:

- **Mark Achtemichuk** – Staff Engineer 2, Performance Engineer
- **Valentin Bondzio** – Sr. Staff Technical Support Engineer

