What's New in vSphere 6.7 Core Storage

VMware Storage

© VMware LLC.

Table of contents

© VMware LLC.

What's New in vSphere 6.7 Core Storage

Core Storage Whitepaper

UNMAP

On thin-provisioned datastores, reclaiming deleted space from various operating systems can be a challenge. How does the GOS (guest operating system) notify the underlying filesystem it no longer needs unused or deleted space? There are two techniques used to pass this data down to the filesystem; SCSI UNMAP and SSD Trim.

In vSphere 6.5, UNMAP added an automated crawler mechanism for reclaiming dead or stranded space on VMFS-6 datastores. This is a big change from previous versions of vSphere, where UNMAP had to be manually run. With vSphere 6.5, UNMAP runs continuously in the background at the default rate of 25MBps. Many flash storage vendors requested higher rates as they can easily handle UNMAP rates higher than 25MBps.

In vSphere 6.7, we have introduced new configurable parameters for automatic UNMAP to enable more granularity in the rate at which UNMAP recovers unused space in VMFS-6. You now have the option to configure an “UNMAP Method” being either “Priority” or “Fixed.” With Priority, the options are none or low with low being the default and set to 25MBps. If Fixed is selected, you can set an UNMAP rate between 100MBps and 2000MBps in 100MB increments. These setting can be changed via CLI or via the UI.

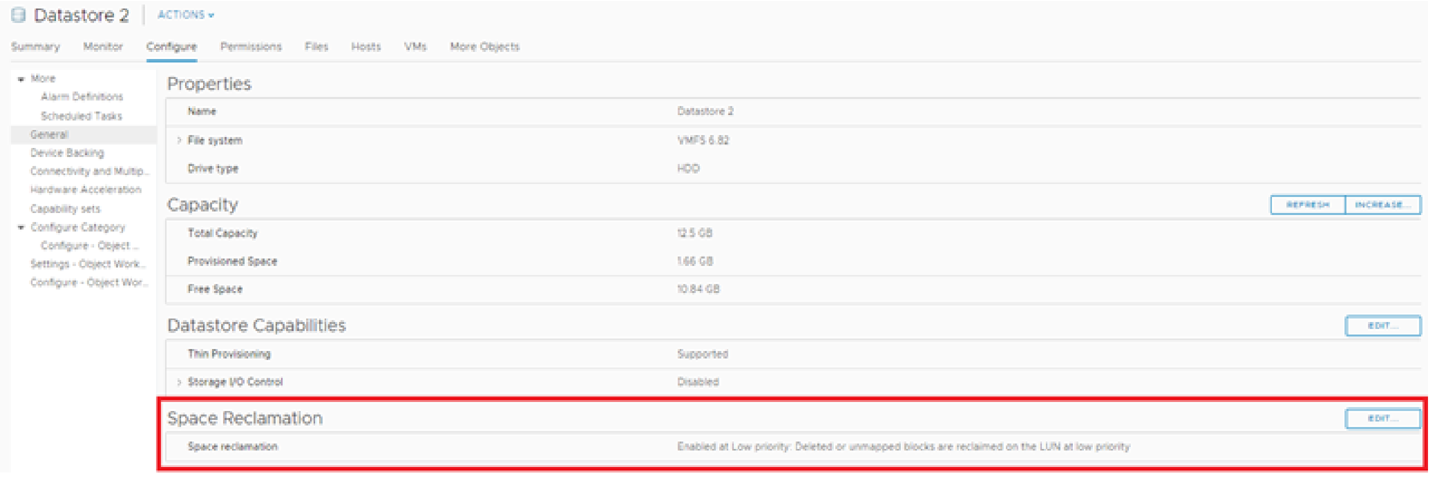
Regarding the method, the values available are “priority” or “fixed.” When UNMAP priority is set, (default) then the rate is the default 25 MBps. When you configure a specific rate, you are changing the method to fixed along with the rate value. For example, if you set the UNMAP rate to 500 MBps, then your method would be changed to fixed.

UNMAP settings and progress may be set and monitored in the UI and via CLI.

The UNMAP setting in the UI are available as follows:

- Select a host, and go to “monitor” -> “performance” -> “advanced” in the VC UI.
- Click on “chart options” and select “datastore” and at the bottom of the list, “UNMAP size” and “UNMAP IOs” are available to be selected. Selecting these starts showing interval stats for UNMAP on available datastores.

Example of UNMAP Priority



Properties

Name	Datastore 2
File system	VMFS 6.82
Drive type	HDD

Capacity

Total Capacity	12.5 GB
Provisioned Space	1.66 GB
Free Space	10.84 GB

Datastore Capabilities

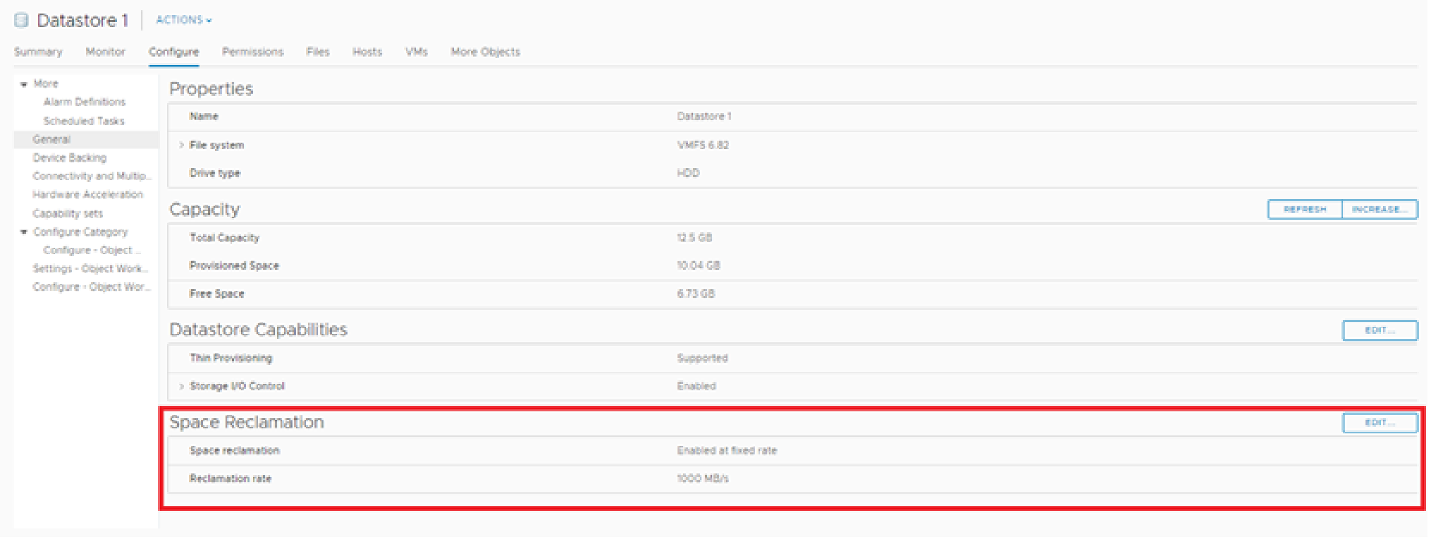
Thin Provisioning	Supported
Storage I/O Control	Disabled

Space Reclamation

Space reclamation	Enabled at Low priority: Deleted or unmapped blocks are reclaimed on the LUN at low priority

Figure 1

Example of UNMAP Fixed



Properties

Name	Datastore 1
File system	VMFS 6.82
Drive type	HDD

Capacity

Total Capacity	12.5 GB
Provisioned Space	10.04 GB
Free Space	6.73 GB

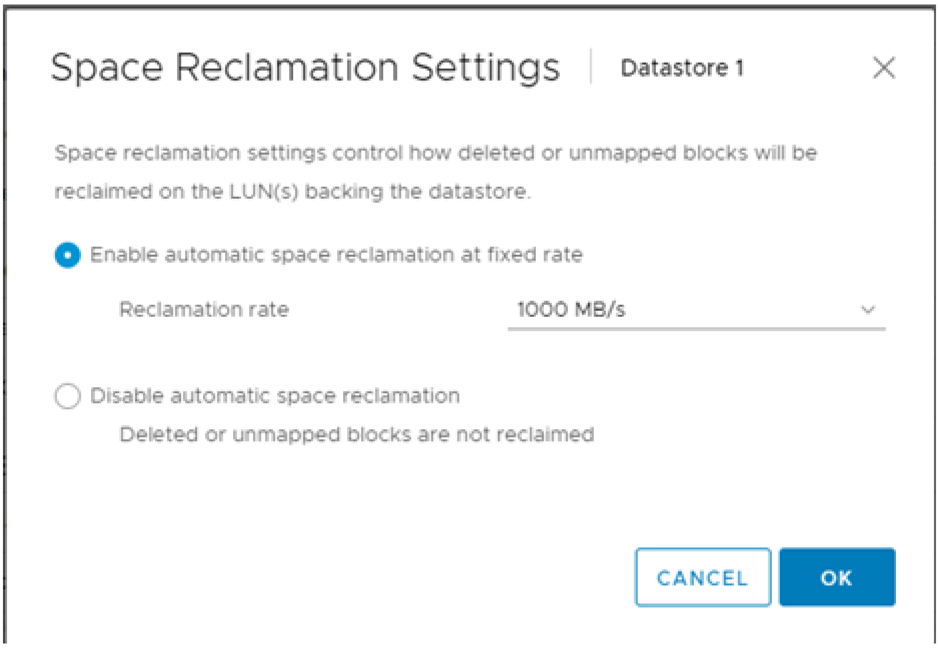
Datastore Capabilities

Thin Provisioning	Supported
Storage I/O Control	Enabled

Space Reclamation

Space reclamation	Enabled at fixed rate
Reclamation rate	1000 MB/s

Figure 2



Space Reclamation Settings | Datastore 1

Space reclamation settings control how deleted or unmapped blocks will be reclaimed on the LUN(s) backing the datastore.

● Enable automatic space reclamation at fixed rate

Reclamation rate: 1000 MB/s

○ Disable automatic space reclamation

Deleted or unmapped blocks are not reclaimed

Figure 3

UNMAP may also be configured via CLI or API; below are some examples of configuring UNMAP via esxcli.

To check the current configuration:

```
esxcli storage vmfs reclaim config get --volume-label <volume-name>
```

To set the rate to 100 MBps:

```
esxcli storage vmfs reclaim config set --volume-label <volume-name> --reclaim-method fixed -b 100
```

To set the rate to 1GBps:

```
esxcli storage vmfs reclaim config set --volume-label <volume-name> --reclaim-
method fixed -b 1000
```

For additional monitoring of UNMAP, there have been two fields added to ESXTOP under the disk monitoring screen. Now you can monitor UNMAPSTATS = Unmap Stats and UNMAPRESULTS = Unmap Results.

esxtop virtual machine, disk field ordering screen:

```
Current Field order: aBCDEfghIJK

  A:  ID = Vscsi Id
* B:  GID = Grp Id
* C:  VMNAME = VM Name
* D:  VDEVNAME = Virtual Device Name
* E:  NVDISK = Num of Virtual Disks
  F:  NUMIOFILTERS = Num of IOFilters
  G:  IOFILTERCLASS = IOFILTERCLASS TYPE
  H:  IOFILTERSTATS = IOFILTER STATS
* I:  IOSTATS = I/O Stats
* J:  LATSTATS/rd = Read Latency Stats (ms)
* K:  LATSTATS/wr = Write Latency Stats (ms)
  L:  UNMAPSTATS = Unmap Stats       <------ NEW VSCSI STATS GROUP
  M:  UNMAPRESULTS = Unmap Results   <------ NEW VSCSI STATS GROUP

GID VMNAME  VDEVNAME NVDISK   CMDS/s  READS/s WRITES/s MBREAD/s MBWRTN/s LAT/rd LAT/wr
|<------ STATS (L) ------>|  |<------ RESULTS (M) ----->|
UNMAPS/s MBUNMAP/s UNMAPFAIL/s UNMAPOK/s
```

UNMAP SESparse

In vSphere 6.7, we now have UNMAP for SESparse disks on VMFS-6. SESparse is a sparse virtual disk format used for snapshots as a default for VMFS-6. In this release, we are providing automatic space reclamation for VM's with SESparse snapshots on VMFS-6. Previously, UNMAP from GOS were processed by SESparse, and its metadata was updated to indicate grains have been unmapped. However, they were not unmapped VMFS and the storage. In vSphere 6.5, the only way to UNMAP from VMFS and storage was a call to an internal API which was used by Horizon Connection Broker. Consequently, it only worked in VDI environments. Now, in vSphere 6.7, UNMAP for SESparse now supports automatic UNMAP. For example, in Windows 10 and above or Linux GOS using -o discard mount option will allow UNMAP for SESparse to reclaim unused space in the SESparse snapshot. It will also work with GOS that do manual UNMAP, for example, optimize drive on Windows, and fstrim on Linux.

In SESparse snapshots, the data is tightly packed, and it may not be possible to get one contiguous filesystem block to UNMAP. In vSphere 6.7, for UNMAP SESparse to get the contiguous space, we need to de-fragment the VMDK. At the end of de-fragment operation, we end up with all the used blocks in the beginning of the file and all un-used blocks at the end of the file; the file is then truncated up to the end of valid data, freeing space on the filesystem. In supported filesystems, the freed space triggers an UNMAP, this workflow is known as "shrink" which is automatically completed in the background.

Before vSphere 6.7, this was a manual process. In vSphere 6.7, auto-UNMAP, also known as auto-shrink, is only done when the VM is powered on and only to the top-most level of the SESParse snapshot in the snapshot hierarchy. The shrink workflow is highly IO intensive and time-consuming. To make sure there is no impact to the running VMs, it is done after accumulating a reasonable amount of free space in the VMDK and is done in smaller fragments. This is controlled by the SESparse config-option /SE/ SEAtioshrinkThresholdMB. It is set to 2GB by default and can be changed based on requirements using CLI "esxcli system settings advanced set –option /SE/ SEAutoshrinkThresholdMB –I <value in MB>." This is a system-wide setting.

A common helper queue is used to handle the background UNMAP. The helper does the UNMAP in time-slice for all the VMDKs. A maximum of 512MB worth of UNMAP is processed for a given VMDK in one-time slice, then the process moves to the next VMDK and so on. Once all the VMDK's are processed, it repeats the same process for all the VMDK's until all the VMDK's done with UNMAP processing.

Below is the State Machine process for reclaim space, this shows how space is monitored and what happens when triggered.

- **IDLE**: When there is not enough free space accumulated. The auto UNMAP currently is not active.

- **INITIATED**: When the freespace threshold is reached, the VMDK moves to this state. This state indicates, the VMDK is queued to helper for auto UNMAP processing. From this state, the VMDK moves to INPROGRESS when the helper picks up for processing.
- **INPROGRESS**: When helper picks up the VMDK and starts the auto UNMAP, VMDK moves to this state. From this, the VMDK can either move to PAUSED or SUSPEND or IDLE (i.e., auto UNMAP is done).
- **PAUSED**: This is when the VMDK uses up its timeslice and needs to wait for its next turn to process the UNMAP. From this state, the VMDK moves to INPROGRESS when it turns arrives.

SUSPEND: This is when a manual “shrink” is issued or for some other reason, we need to temporarily stop the auto-UNMAP processing. From this state, the VMDK moves to INPROGRESS when it is ready to process the auto- UNMAP again

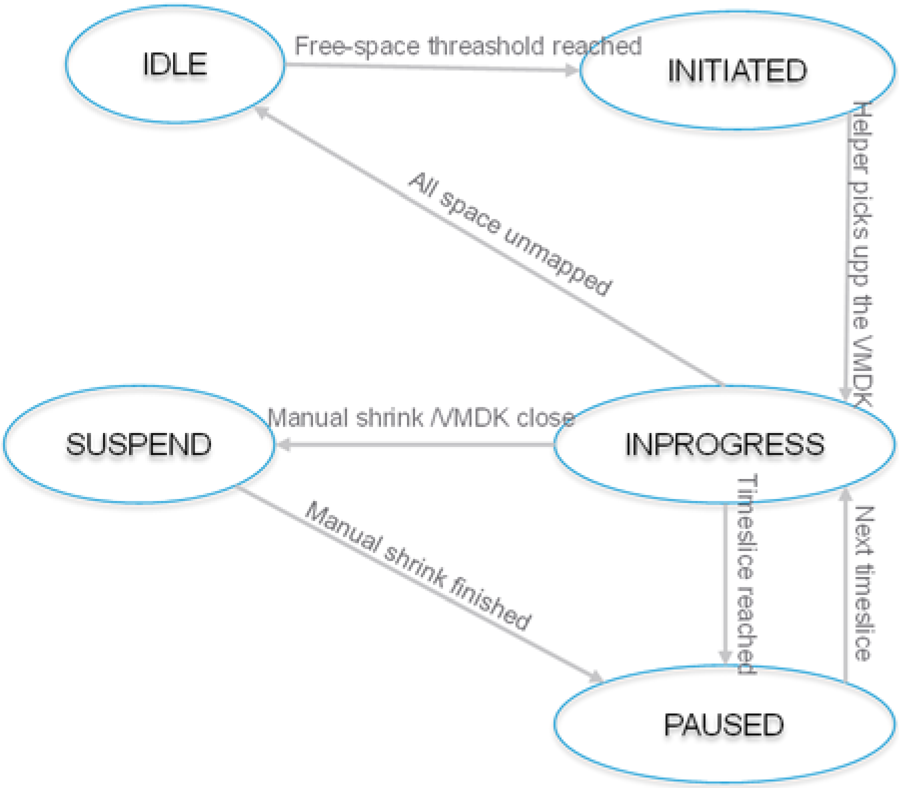


Figure 4

4kn SWE (Software Emulated)

Storage Vendors are moving towards cost-efficient 4K Native drives. The migration to 4K-sized sectors will provide quicker paths to higher areal densities and hard drive capacities as well as more robust error correction. The HDD vendors have been manufacturing 4K sectored drives but using emulation (a.k.a 512e) in the firmware to reduce the impact of the format change to the host clients. 512e drives were introduced to enable the transition to 4Kn drives. Vendors expect mass adoption of 4Kn within the next few years. Subsequently, VMware has been working to enable 4Kn drives in vSphere to ensure utilization of the latest technology.

4Kn drives have various benefits over 512 sector size drives. Higher capacity and improved performance from the more optimized placement of data on the drive. Efficient space utilization with optimized meta-data giving up to 10% more available data. Improved drive reliability and error correction with larger meta-data by increasing the ECC block from 50 to 100 bytes. This provides a much-needed improvement in error correction efficiency.

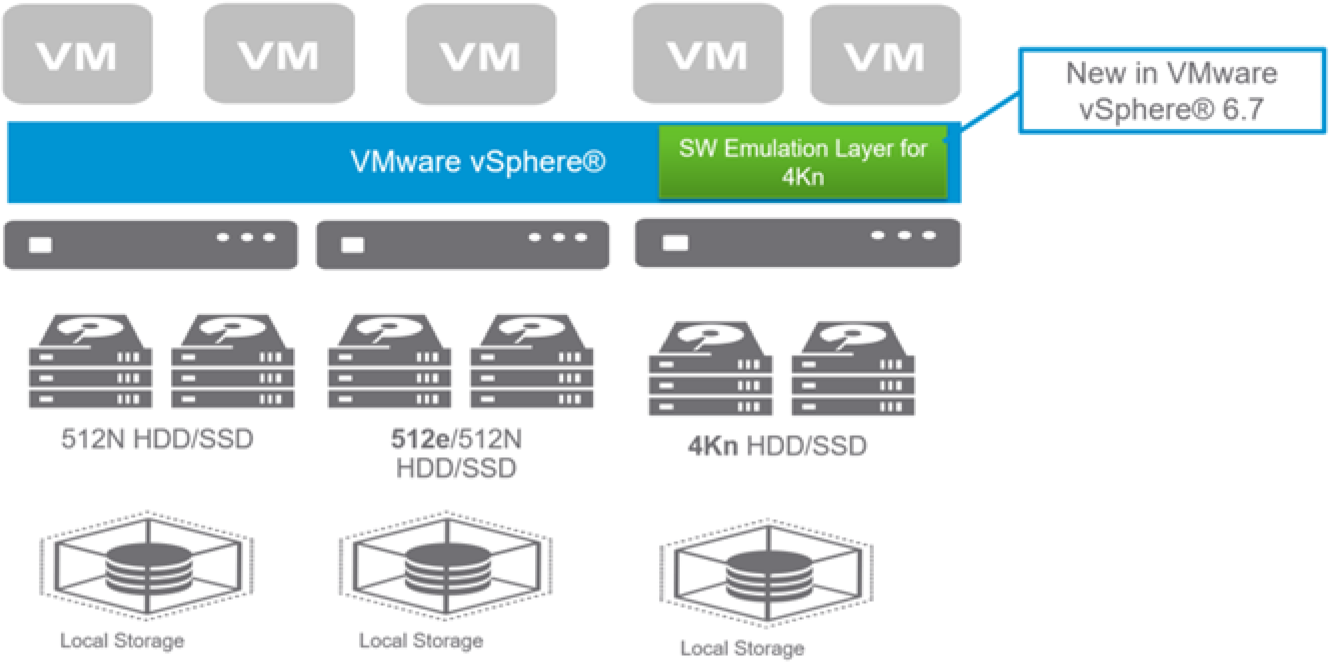


Figure 5

The release of vSphere 6.7 4Kn (4K native) direct attached drives are now supported natively via 4Kn SWE (Software Emulation). The software emulation layer allows the use of 4Kn drives while still allowing legacy OS, applications, and existing VMs to run on newer 4Kn drives.

There are some limitations for 4Kn drives; only local SAS, SATA HDDs are supported, must use VMFS6, and booting from 4Kn drives requires UEFI. Also, 4Kn SSD, NVMe, and RDM to GOS are not supported. 4K VMDKs may only be created on datastores made up of 4Kn disks or 512e disks. vSAN and VVOL may declare themselves as 512e if they can handle both 512 byte and 4k I/Os without any atomicity issues. Third party multi-pathing plugins are not supported.

Datastore	VMDK		
	512n	512e	4K
512n (physical sector=512, logical sector=512)	Yes	Yes	No
512e (physical sector=4096, logical sector=512)	Yes	Yes	Yes
4K (physical sector=4096, logical sector=4096)	Yes (With RMW)	Yes (With RMW)	No

Table 1

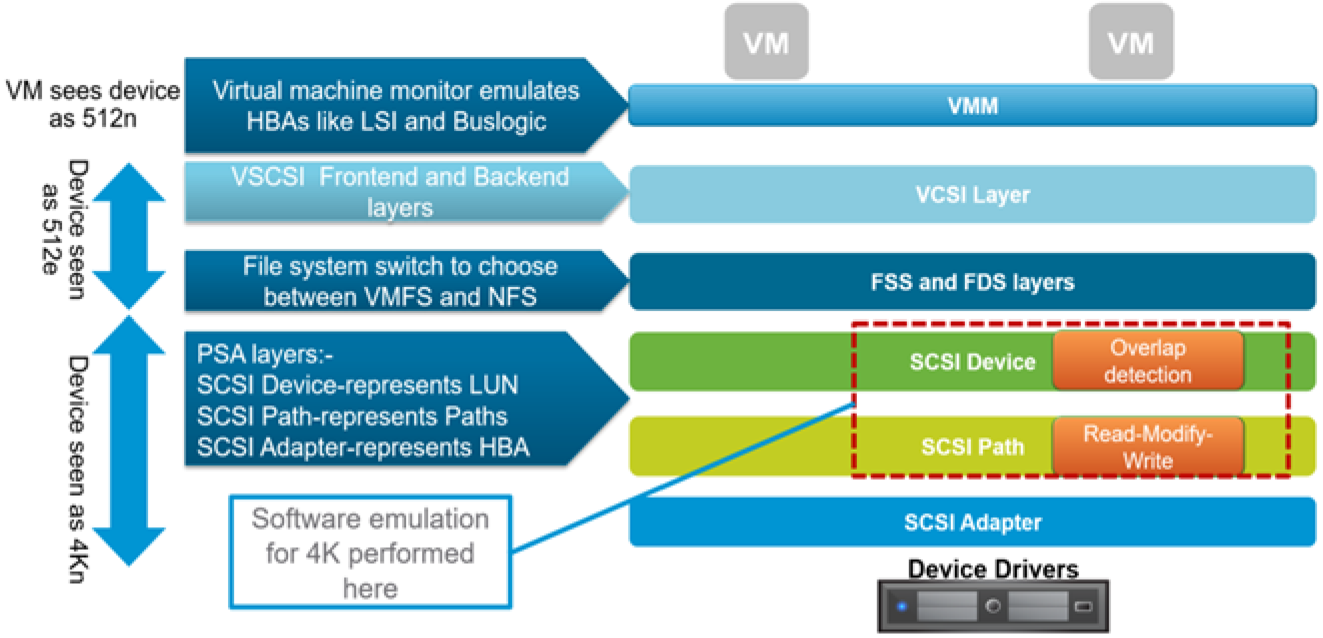

VM
VM
VM sees device as 512n
Virtual machine monitor emulates HBAs like LSI and Buslogic
VMM
Device seen as 512e
VSCSI Frontend and Backend layers
VCSI Layer
File system switch to choose between VMFS and NFS
FSS and FDS layers
Device seen as 4Kn
PSA layers:-
SCSI Device-represents LUN
SCSI Path-represents Paths
SCSI Adapter-represents HBA
SCSI Device
Overlap detection
SCSI Path
Read-Modify-Write
Software emulation for 4K performed here
SCSI Adapter
Device Drivers

Figure 6

For writes, if the WRITE I/O is less than 4K or I/O is misaligned (logical to physical), then an RMW (Read-Modify-Write) action is triggered otherwise, the WRITE I/O goes straight to disk. If an RMW is triggered, it reads the entire 4K sector from disk into memory, modifying the 512 sector, then writes the 4k sector of data to disk. If a second I/O request occurs on the same 4K sector, the original RMW is completed before allowing the second I/O to write the same block. This is called overlap detection and prevents parallel writes and data loss.

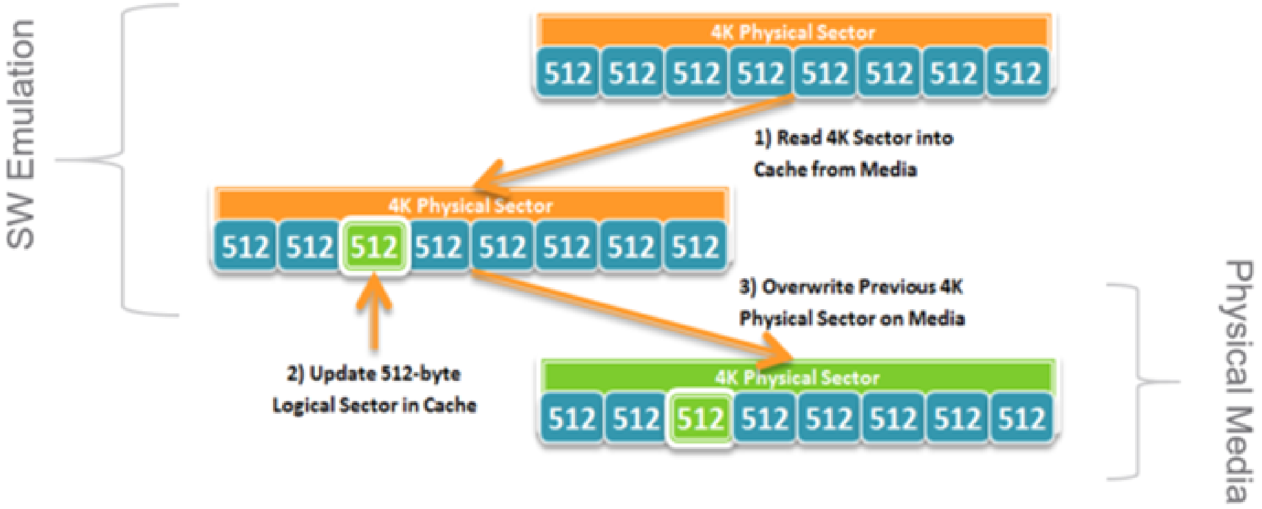
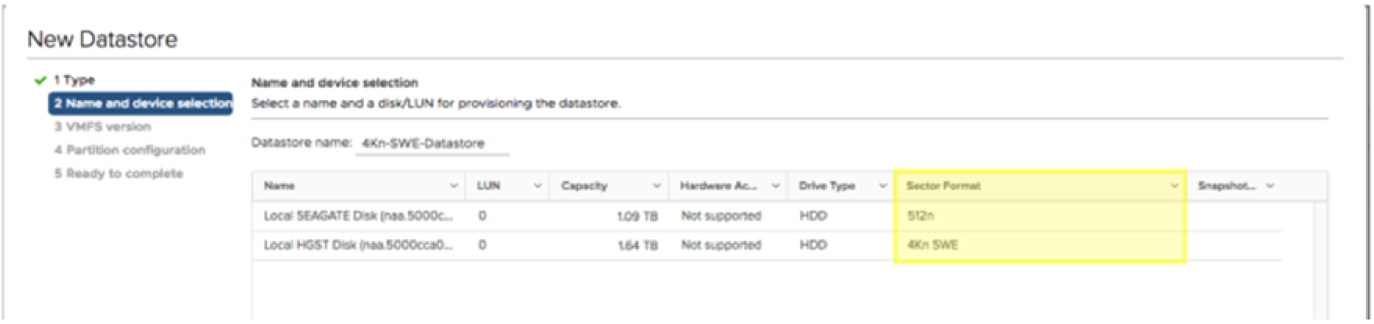


Figure 7

For reads, if the READ I/O is less than 4K or I/O is misaligned (logical to physical), then an RMW (Read-Modify-Write) action is triggered otherwise the READ I/O goes straight to memory. If an RMW is triggered for READ I/O, the 4K sector is read into memory; the RMW completes the original 512 byte command, then the additional allocated memory is freed. The overlap detection also monitors for READ I/O overlaps to ensure memory is not overwritten.

Creating a new datastore:



New Datastore

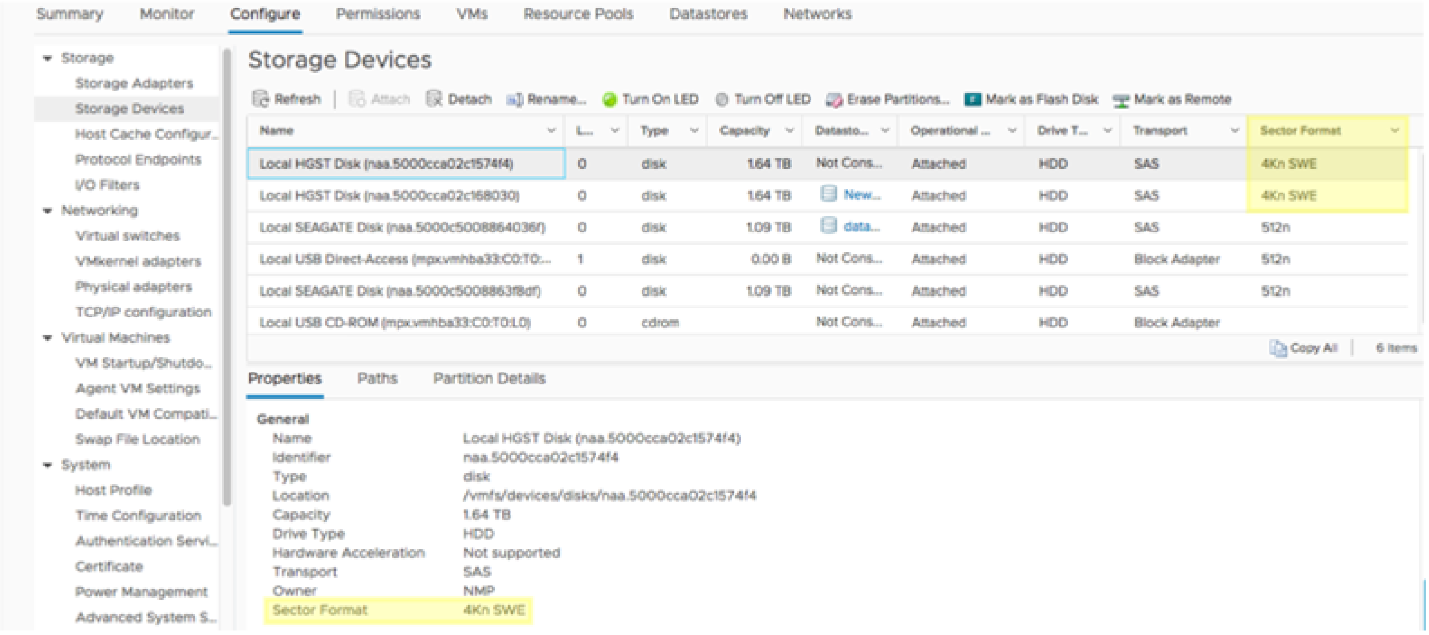
Name and device selection

Select a name and a disk/LUN for provisioning the datastore.

Datastore name: 4Kn-SWE-Datastore

Name	LUN	Capacity	Hardware Ac...	Drive Type	Sector Format	Snapshot...
Local SEAGATE Disk (naa.5000c...	0	1.09 TB	Not supported	HDD	512n	
Local HGST Disk (naa.5000cca0...	0	1.64 TB	Not supported	HDD	4Kn SWE	

Figure 8



Storage Devices

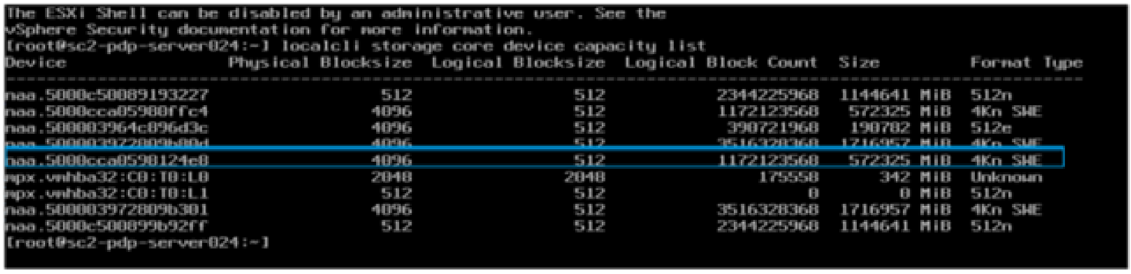
Name	L...	Type	Capacity	Datasto...	Operational ...	Drive T...	Transport	Sector Format
Local HGST Disk (naa.5000cca02c1574f4)	0	disk	1.64 TB	Not Cons...	Attached	HDD	SAS	4Kn SWE
Local HGST Disk (naa.5000cca02c168030)	0	disk	1.64 TB	New...	Attached	HDD	SAS	4Kn SWE
Local SEAGATE Disk (naa.5000c5008864036f)	0	disk	1.09 TB	data...	Attached	HDD	SAS	512n
Local USB Direct-Access (mpx.vmhba33:C0:T0:...	1	disk	0.00 B	Not Cons...	Attached	HDD	Block Adapter	512n
Local SEAGATE Disk (naa.5000c5008863f8df)	0	disk	1.09 TB	Not Cons...	Attached	HDD	SAS	512n
Local USB CD-ROM (mpx.vmhba33:C0:T0:L0)	0	cdrom		Not Cons...	Attached	HDD	Block Adapter	

General

Name	Local HGST Disk (naa.5000cca02c1574f4)
Identifier	naa.5000cca02c1574f4
Type	disk
Location	/vmfs/devices/disks/naa.5000cca02c1574f4
Capacity	1.64 TB
Drive Type	HDD
Hardware Acceleration	Not supported
Transport	SAS
Owner	NMP
Sector Format	4Kn SWE

Figure 9

ESXi shell output for 4Kn DiskID to be selected for ESXi Installer



```
The ESXi Shell can be disabled by an administrative user. See the
vSphere Security documentation for more information.
[root@sc2-pdp-server024:~] localcli storage core device capacity list
Device                    Physical Blocksize  Logical Blocksize  Logical Block Count  Size         Format Type
-------------------------------------------------------------------------------------------------------------
naa.5000c50089193227                     512                512           2344225968  1144641 MiB  512n
naa.5000cca05900ffc4                    4096                512           1172123568   572325 MiB  4Kn SWE
naa.500003964c096d3c                    4096                512            390721968   190782 MiB  512e
naa.50000397280?b00d                    4096                512           3516328368  1716957 MiB  4Kn SWE
naa.5000cca0590124e0                    4096                512           1172123568   572325 MiB  4Kn SWE
mpx.vmhba32:C0:T0:L0                    2048               2048               175558      342 MiB  Unknown
mpx.vmhba32:C0:T0:L1                     512                512                    0        0 MiB  512n
naa.500003972809b301                    4096                512           3516328368  1716957 MiB  4Kn SWE
naa.5000c500899b92ff                     512                512           2344225968  1144641 MiB  512n
[root@sc2-pdp-server024:~]
```

Figure 10

vSAN and 4Kn

4Kn drives with disk group format 4 and above are supported for capacity only. There are no backward compatibility concerns since 4Kn HDDs have not been supported by vSphere or vSAN before vSphere 6.7. vSAN addresses this concern by already being 4Kn aware of vSAN format 4 capacity tier disks.

This is done by having all higher layer vSAN data I/O be 4KB-aligned and 4KB-multiple-sized. Reading/writing from/to the VSAN caching tier disks using 4KB-aligned and 4KB-multiple-sized I/O, and de-staging I/O from the caching tier to capacity tier disks as 4KB-aligned and 4KB-multiple-sized I/O. All vSAN control plane I/O that is issued as raw SCSI/ATA commands to a vSAN disks are changed to utilize the physical sector size of the disk. Therefore, all 4Kn disks will be vSAN format 4 (or greater) and all I/O to these disks will be 4KB-aligned and 4KB-multiple-sized. vSAN disks with vSAN format 3 or below will be aligned and sized according to the logical block size of the vSAN capacity tier physical disk. That is 512 bytes for vSAN disk format versions 3 and earlier. Hardware upgrade from 512n/512e to 4Kn HDDs has no restrictions. A vSAN disk group may contain a mix of disks with different sector sizes. 4Kn HDDs utilized as vSAN capacity tier disks are not required to be formatted as vSAN version 4 disks so 4Kn disks may be added to existing disk groups that are not formatted as version 4. A vSAN Health UI check is provided to alert a customer whenever a 4Kn disk claimed by vSAN is not formatted as a version 4 disk.

To enable 4Kn drives for vSAN you must enable using the command below. Once enabled, on the vSAN management page, “4Kn SWE” is displayed for the format type.

```
#esxcfg-advcfg --set <1|0> /VSAN/Device4KSupport
```

1K/4K LUN/Path Maximum Increase

With the release of vSphere 6.7, the maximum number of LUNs and paths per host has increased. Now, you can have up to 1024 LUNs and 4096 Paths per host. This is double the previous 512 LUNs/2048 Paths. At the VM level, the number of vDisks has increased from 16 to 64 disks. This equates to 256 disks using PVSCSI adapters. This increase is a major enhancement for Microsoft WSFC. Previously Microsoft WSFC VM pRDMs were limited to 45 LUNs, with vSphere 6.7 that number has increased to 192. This enables the customer to have larger clusters and reduce the overhead of managing multiple smaller clusters. The reason there are only 192 disks/LUNs per VM for Microsoft WSFC is one of the four controllers is used for management leaving three controllers with 64 vDisks/LUNs each. Applications not requiring SCSI controller 0 for management may have up to 256 PVSCSI devices.

Old PVSCSI device limit - 45 per WSFC

New PVSCSI device limit - 192 per WSFC

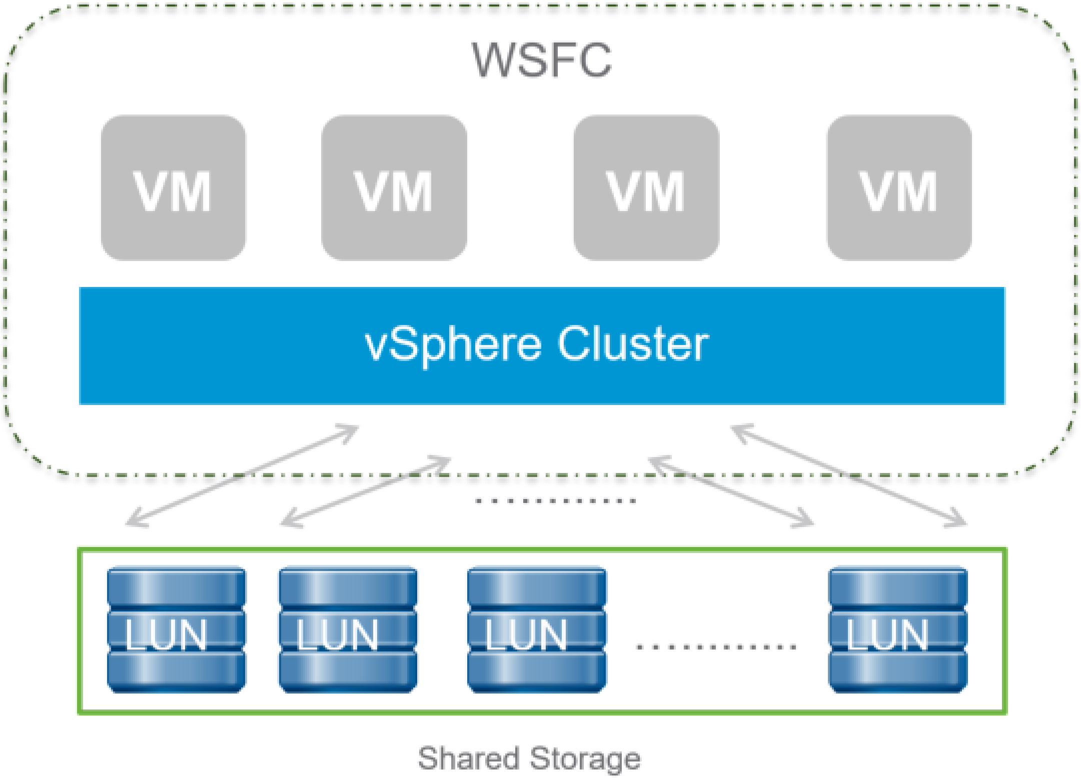


Figure 11

XCOPY Enhancements

Within the VAAI (vStorage API for Array Integration) primitives, XCOPY is used to offload storage-intensive operations such as copying, cloning, and zeroing to the storage array instead of the ESXi host. This frees up host CPU, buffers, and network resources to be utilized elsewhere. Previously, specific transfer sizes were limited to EMC VMAX arrays. With vSphere 6.7, we have enabled XCOPY to be configured with specific parameters to optimize the storage array's XCOPY operations. We have implemented this in two stages; Support for arrays using the standard SCSI T10 plugin (VMW_VAAIP_T10) and support for vendor-specific plugins (vmkAPI).

To enable support and optimize XCOPY, we use a claim rule with the specific values and array details.

Parameters introduced (These parameters can be set via esxcli storage core claimrule add)

- **xcopy-use-array-values(-a)**: Enable usage of storage reported parameters to size the XCOPY segment.
- **xcopy-use-multi-segs(-s)**: Enable usage of multiple data segments in single XCOPY command
- **xcopy-max-transfer-size(-m)**: Specify the maximum transfer Size in MBs in case the user wants to use a different transfer Size than the array reported.

With Vshpere 6.7, a new option is added to specify max transferSize in KBs. This allows setting max transfer Sizes lower than 1MB (For example, preferred transfer size for XtremeIO is 64K/128K).

- **--xcopy-max-transfer-size-kib(-k):** Maximum transfer size in KiB to use for XCOPY commands if the admin wants to use a transfer size different than array reported. This option takes precedence over –xcopy-max-transfer-size option

A few Examples

Add the VAAI claimrule specifying XCOPY parameters (with vendor-specific VAAI plugin).

```
esxcli storage core claimrule add -r 914 -t vendor -V XtremIO -M XtremApp -P
VMW_VAAIP_T10 -c VAAI -a -s
```

Add the VAAI claim rule specifying XCOPY Parameters (without vendor-specific VAAI plugin).

```
esxcli storage core claimrule add -r 65430 -t vendor -V EMC -M SYMMETRIX -P
VMW_VAAIP_SYMM -c VAAI -a -s -m 200
```

Virtual Volumes (vVols)

Virtual Volumes (vVols) offers a new paradigm in which an individual virtual machine and its disks, rather than a LUN, becomes a unit of storage management for a storage system. vVols encapsulate virtual disks and other virtual machine files and natively store the files on the storage system. With Virtual Volumes, the model changes from managing space inside datastores (tied to LUNS) to managing abstract storage objects handled by storage arrays. Each vVol is exported to the ESXi host through a small set of protocol end-points (PE). Protocol Endpoints are part of the physical storage fabric, and they establish a data path from virtual machines to their respective vVols on demand.

With vSphere 6.7, three features added to enhance the functionality of vVols.

1. IPv6 support for management access to the VASA provider
2. SCSI-3 Persistent Group Reservations (PGRs) support for supported arrays.
3. TLS 1.2 default VP security.

With end to end support for IPv6, this enables many organizations including the government, to implement vVols using IPv6.

With vVols supporting SCSI-3 reservations this adds support for Microsoft WSFC clusters. What are SCSI reservations? This allows multiple nodes to access a LUN at a storage level while restricting simultaneous access to that LUN from other nodes to avoid data corruption. By allowing multiple nodes access to an individual LUN, this allows Microsoft WSFC (Windows Server Failover Clustering) to manage the locking along with the storage. This is called registration and reservation, which allows one node to eject another node. Other features of SCSI-3 include persistent reservations across reboots and supporting multi-pathing to disks.

With SCSI-2 reservations, only one node can access a LUN at a time. Consequently, to use Microsoft WSFC, you had to use RDMs to the virtual machines because the storage arrays support physical SCSI-3 reservations. Also, SCSI-2 does not allow multipath from host to disk, and rebooting loses the reservation (non-persistent).

Why is this a big deal? With SCSI-3 reservation in vVols, you can now eliminate RDMs for Microsoft WSFC clusters! SCSCi-3 vVols allows shared virtual disks/volumes across nodes. By eliminating RDM, you now have access to virtualized benefits such as vMotion, snapshots, DRS, HA, etc.

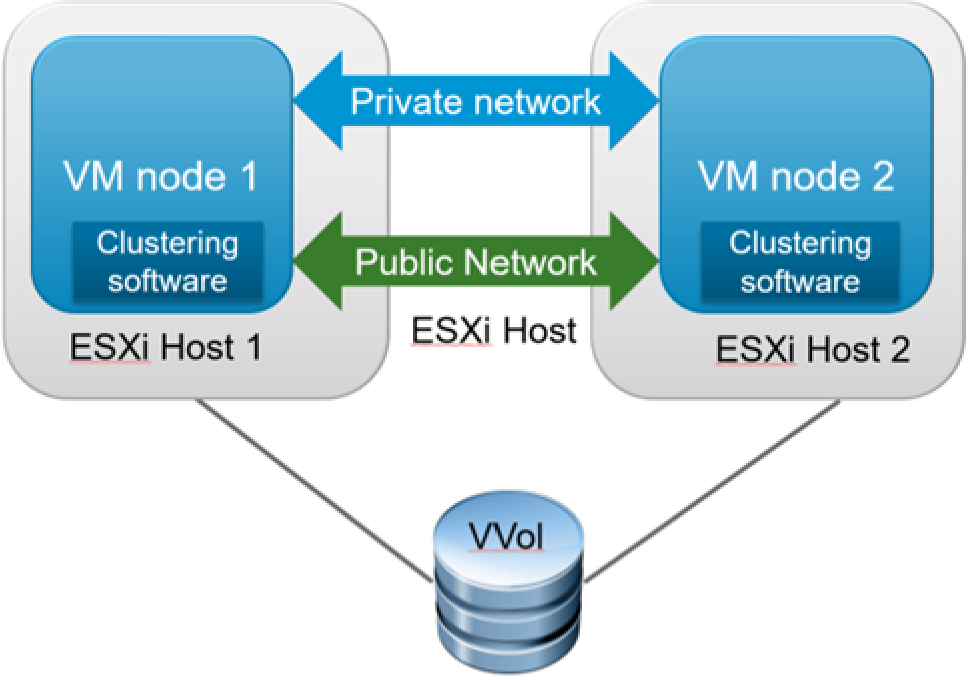


Figure 12

There are some caveats specific to WSFC; the cluster must be across nodes and not in the same node or host unless part of a separate cluster. Virtual hardware must be version 14 or above, NFS vVols are not supported, and the array must support SCSI-3 Persistent Reservation at the secondary LUN (vVol) level. When setting up the virtual machine, the SCSI controller should have the “SCSI bus sharing” field set to physical.

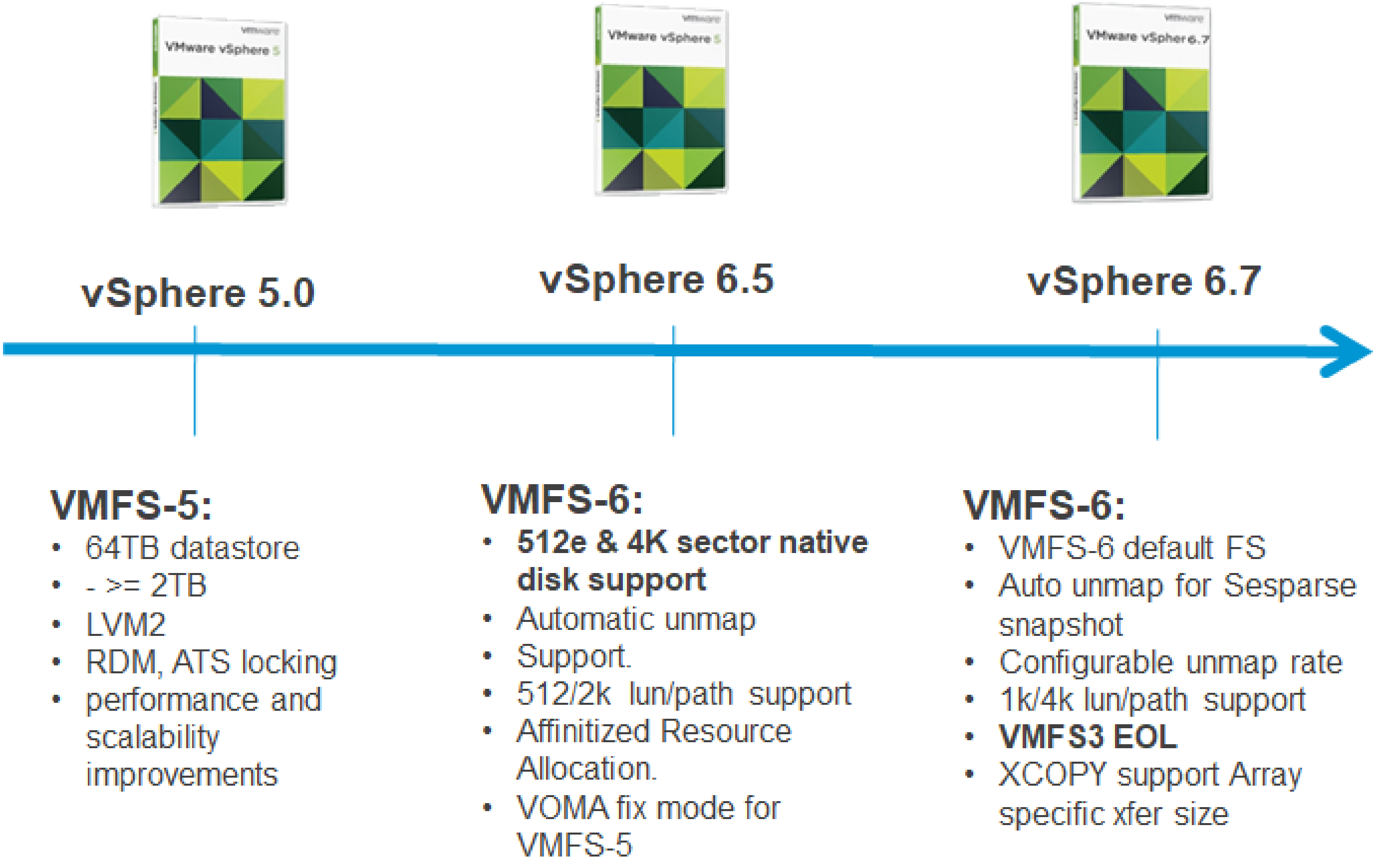
Secondary LUN ID

vVols with block-based storage (FC & iSCSI) use something called a secondary LUN ID to communicate with the virtual volumes on the array. This enables ESXi hosts to connect to a very large number of vVols (up to 64,000 per host) via a small number of Protocol Endpoint LUNs. This scaled out approach to I/O does rely on your HBAs supporting this secondary LUN ID so a PE can distinguish the I/O to individual vVols associated with it.

You can check that your HBAs support vVol by going to the VMware Compatibility Guide for IO devices, selecting your HBA vendor, and then selecting the feature support “Secondary LUNID (Enables vVol)”. If your HBA vendor supports the Secondary LUN ID feature you can then drill down into the search results to view specific driver and firmware versions supporting this feature.

VMFS-3 EOL

Up to vSphere 6.5, VMFS-3 volumes are supported, from vSphere 6, and above VMFS-3 volumes cannot be created. In vSphere 6.7, VMFS-3 has been End Of Lifed and will not be supported. Subsequently, if you do not upgrade your VMFS-3 volume(s) to VMFS-5 or above, when installing vSphere 6.7 your VMFS-3 volume(s) will automatically be upgraded to VMFS-5 when mounted. If VMFS-3 volume is not upgraded, it will not be automounted, and files cannot be created or opened on a VMFS-3 volume. If the automatic upgrade fails, the volume will need to be retired manually. Failures from the upgrade will remain unmounted and flagged (FAILED_TO_UPGRADE). VMFS-3 volumes will remain accessible from hosts running vSphere 6.5 or below. The upgrade will occur on the first host upgraded to vSphere 6.7. Any new datastores/volumes created under vSphere 6.7 will use VMFS-6 as the default.



vSphere 5.0

VMFS-5:
- 64TB datastore
- - >= 2TB
- LVM2
- RDM, ATS locking
- performance and scalability improvements

vSphere 6.5

VMFS-6:
- **512e & 4K sector native disk support**
- Automatic unmap
- Support.
- 512/2k lun/path support
- Affinitized Resource Allocation.
- VOMA fix mode for VMFS-5

vSphere 6.7

VMFS-6:
- VMFS-6 default FS
- Auto unmap for Sesparse snapshot
- Configurable unmap rate
- 1k/4k lun/path support
- **VMFS3 EOL**
- XCOPY support Array specific xfer size

Figure 13

Upgraded VMFS-5 partitions will retain the partition characteristics of the original VMFS-3 datastore, including file block-size, sub-block size of 64K, etc. To take full advantage of all the benefits of VMFS-5, migrate the virtual machines to another datastore(s), delete the existing datastore, and re-create it using VMFS-5. (KB 2003813)

Note: Increasing the size of an upgraded VMFS datastore beyond 2TB changes the partition type from MBR to GPT. However, all other features/characteristics continue to remain the same.

Intel VMD

Intel has released a new technology called Intel VMD (Volume Management Device) which enables the serviceability of NVMe drives. Intel VMD is a hardware logic provided inside the Intel Xeon processor. It aggregates the NVMe PCIe SSDs attached to its root port and acts like an HBA does for SATA and SAS.

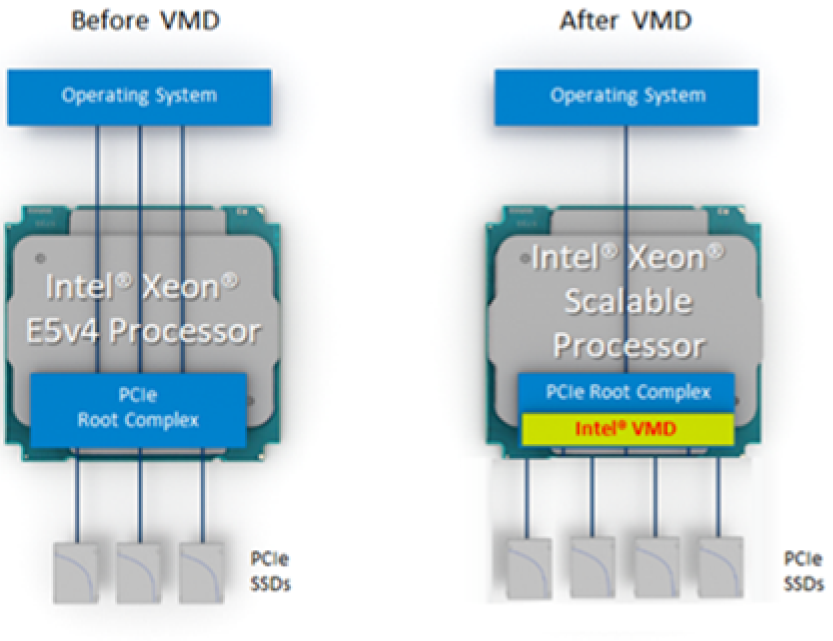


Figure 14

Intel VMD provides Error management, Surprise Hot-plug, and LED Management. With compatible hardware, you may now enable status LEDs and hot-swap NVMe drives as you can with SAS or SATA drives. Subsequently, in the event of a drive needing to be located or replaced you can turn on the status LED, locate and swap the drive without shutting down the system. Now with the release of vSphere 6.7, VMware can also use these functions with vSAN and/or other local or DAS NVMe drives without the need to install VIBS. All Intel VMD function will be in box and support vSAN and DAS storage. This is important as you need to be able to replace individual disks in case of upgrades or in the event of a failure.

System Requirements

- Intel Xeon Scalable platform with The Intel VMD enabled on NVMe PCIe SSD connected PCIe lanes
- Intel VMD-enabled NVMe driver is installed.
- Intel VMD LED management command-line tool is installed on the following VMware* ESXi operating system versions: 6.0 U3 & 6.5.
- Before using the command line tools, the ESXi command-line shell should be enabled from either the vSphere client or from the direct console of the ESXi host system.

Note: On ESXi systems, the tools only work on PCIe NVMe drives using the Intel VMD-enabled NVMe driver.

RDMA (Remote Direct Memory Access)

RDMA or Remote Direct Memory Access allows the transfer of memory from one computer to another. This is a direct transfer and minimizing CPU/ kernel involvement. By bypassing the kernel, we get extremely high I/O bandwidth and, low latency. To use RDMA, you must have HCA (Host Channel Adapter) device on both the source and destination.

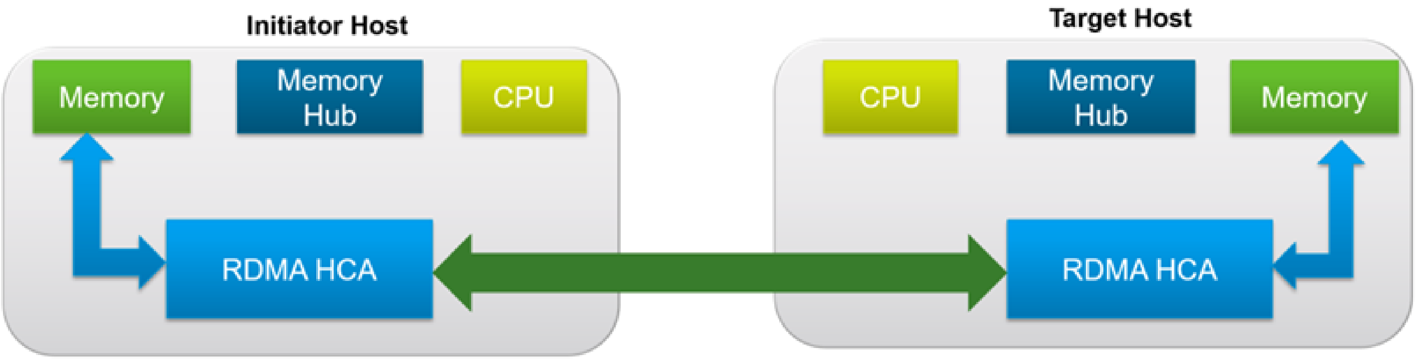


Figure 15

The figure below shows the different paths between a standard transfer and an RDMA transfer. As you can see the RDMA bypasses several layers by bypassing the kernel. There are a few types of network technologies that RDMA supports; iWARP, Infiniband, and RoCE.

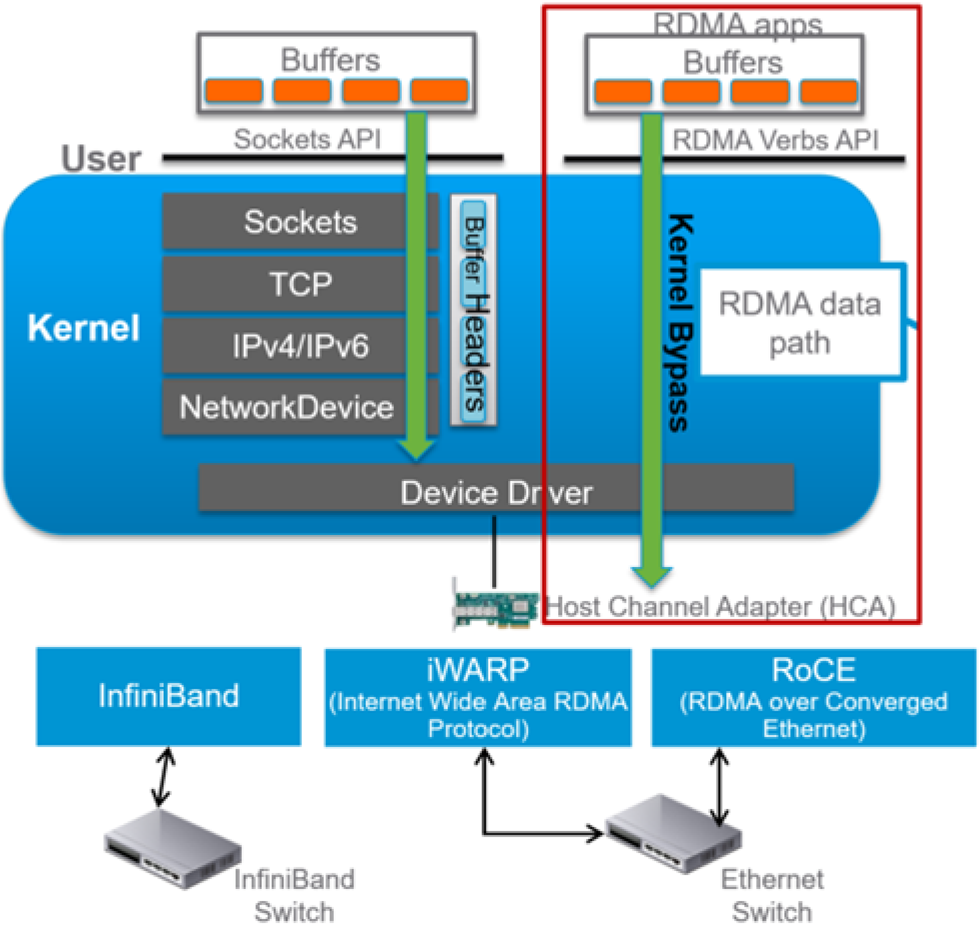


Figure 16

With the release of vSphere 6.7, VMware now supports RDMA and iSER. With vSphere 6.7, VMware is only supporting RoCE (RDMA over Converged Ethernet). RDMA may be used in two forms;

RDMA in ESXi

- RDMA can be useful to accelerate many of the hypervisor services, including; vMotion, SMP-FT, NFS, and iSCSI.

RDMA between Virtual Machines Referred to as vRDMA. There is a wide range of applications that may use RDMA for low latency and high throughput communication.

- High-performance computing, databases, trading application, etc.
- Emerging big data applications such as Hadoop

With vRDMA, there are three transport modes that can occur and are selected automatically.

1. Memcpy – RDMA between VMs on the same host.
2. TCP – RDMA between hosts without Host Channel Adapters(HCA).
3. RDMA – Fast Path RDMA between hosts with HCAs

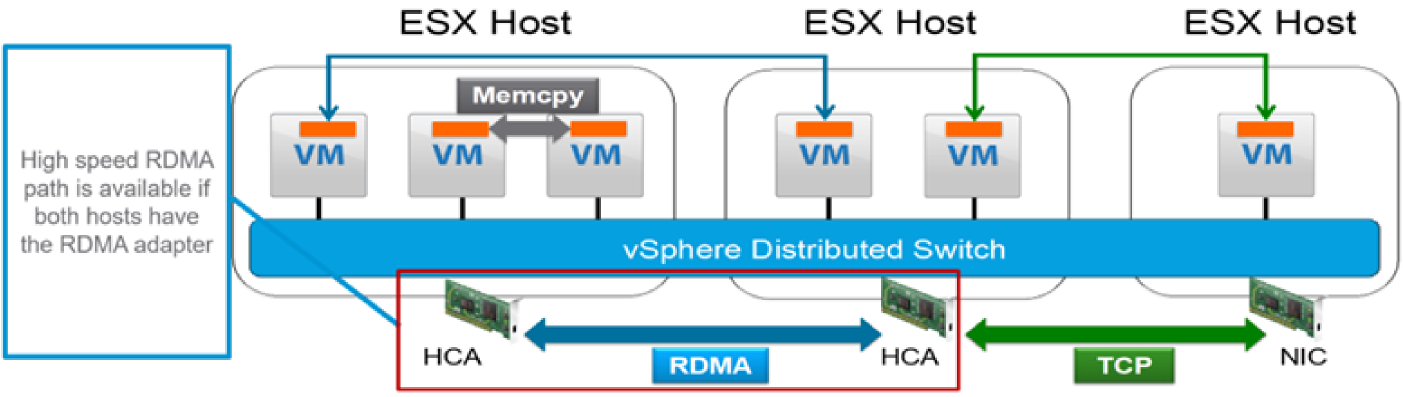

ESX Host
ESX Host
ESX Host
High speed RDMA path is available if both hosts have the RDMA adapter
Memcpy
VM
VM
VM
VM
VM
VM
vSphere Distributed Switch
HCA
RDMA
HCA
TCP
NIC

Figure 17

iSER (iSCSI Extensions for RDMA)

How is RDMA relevant to storage? With vSphere 6.7, iSER (iSCSI Extensions for RDMA) provides high-performance iSCSI connectivity with standard RDMA HCAs (NICs). It lowers CPU utilization by replacing the TCP transport protocol with RDMA transport. Using RDMA, the data is sent to the destination without the overhead of copying from the TCP packet buffers. iSER maintains comprehensive iSCSI management capabilities such as discovery, naming, security, error-recovery, and boot. It allows the use of standard management tools and interfaces while leveraging well-known iSCSI practices. With this release of iSER, we will support; RoCE protocol versions 1 & 2, adapters speeds of 40Gbps and 100Gbps, and IPv4 and IPv6.

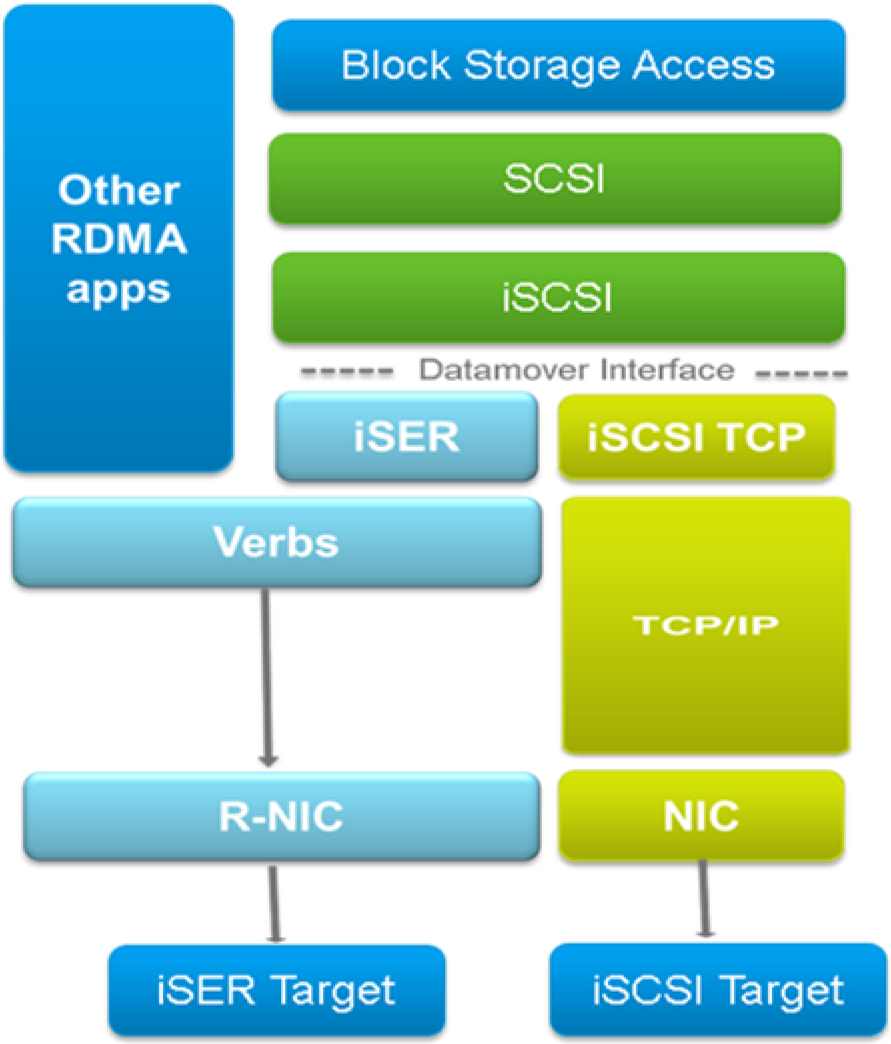
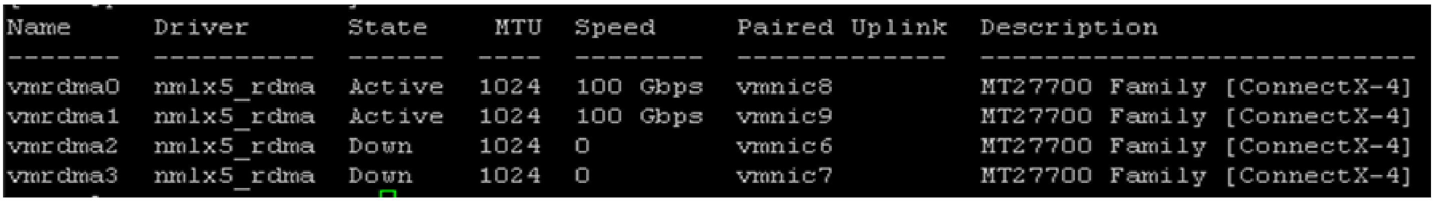


Figure 18

Architecture

iSER driver sits between the PSA and the Networking (RDMA) layers, providing the transport mechanism for ESXi's iSCSI (transport) layer. iSER driver load creates a new iSER logical device. iSER driver creates/registers a PSA device (vmhba), and it registers that vmhba with iSCSI transport layer to become the (RDMA) transport mechanism. iSCSI target portal is discovered on TCP/IP network. Once vmkiscsid discovers the target, it calls into iSER to start iSCSI session(s) to the target. iSCSI sessions from iSER are maintained in the RDMA context via RDMA CM (Connection Manager) The actual data transfer for Reads/Writes happens via pure RDMA via Core RDMA Interfaces (PostSend and PostReceive). RDMA NIC (R-NIC) registers two logical devices.

- Logical uplink (vmnic #)
- Logical RDMA device (vmrdma#)



```
Name     Driver      State   MTU  Speed     Paired Uplink  Description
-------  ----------  ------  ---- --------  -------------  --------------------------
vmrdma0  nmlx5_rdma  Active  1024 100 Gbps  vmnic8         MT27700 Family [ConnectX-4]
vmrdma1  nmlx5_rdma  Active  1024 100 Gbps  vmnic9         MT27700 Family [ConnectX-4]
vmrdma2  nmlx5_rdma  Down    1024 0         vmnic6         MT27700 Family [ConnectX-4]
vmrdma3  nmlx5_rdma  Down    1024 0         vmnic7         MT27700 Family [ConnectX-4]
```

Figure 19

Logical uplink and the logical RDMA device are mapped one-to-one. Connections with the target are established using iSCSI target IP addresses. Multipathing of iSER LUNs are managed by NMP.

PFC (Priority Flow Control) must be enabled end-to-end. Including RoCE driver, RDMA switch ports for initiator and target, and for RoCE drivers on the targets if applicable. Two vSwtiches are needed with one RDMA capable NIC per vSwitch. NIC teaming is not supported with R-NICs.

vSphere support for Persistent Memory (PMem) NVDIMM

Persistent Memory or PMem is a type of non-volatile DRAM (NVDIMM) that has the speed of DRAM but retains contents through power cycles. Subsequently, resuming functionality is significantly faster as the content of RAM does not need to be reloaded. In addition to being non-volatile, it’s byte addressable, meaning it can be used as storage.

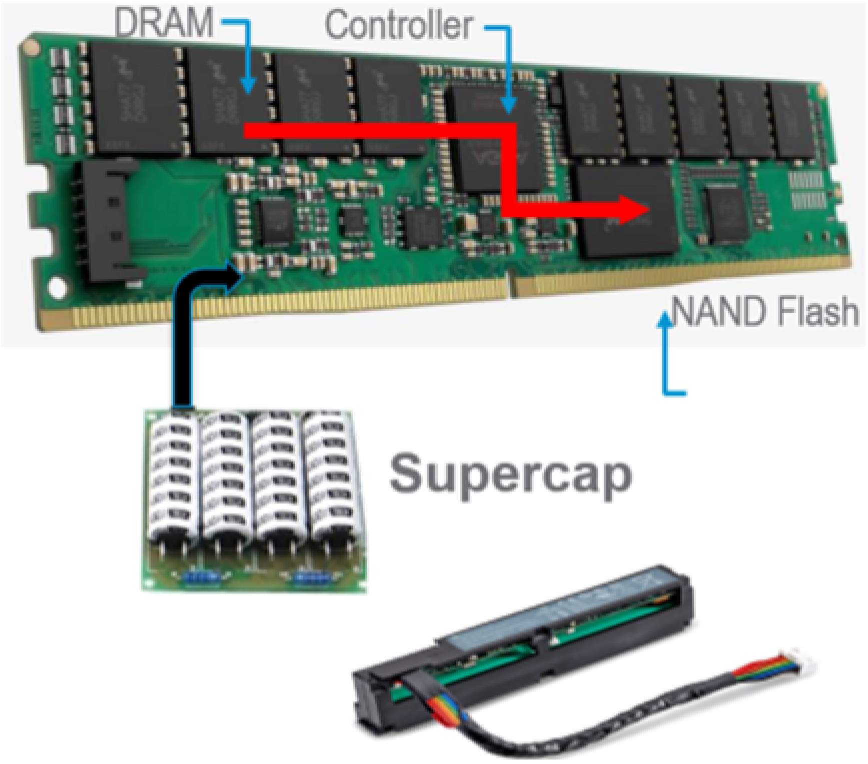


Figure 20

This brings incredible performance possibilities as NVDIMMs are equal to or near the speed of DRAM; almost 100 times faster than SSDs! By moving the data closer to where the analysis is done, the CPU can access the data as if it was in RAM with DRAM-like latencies.

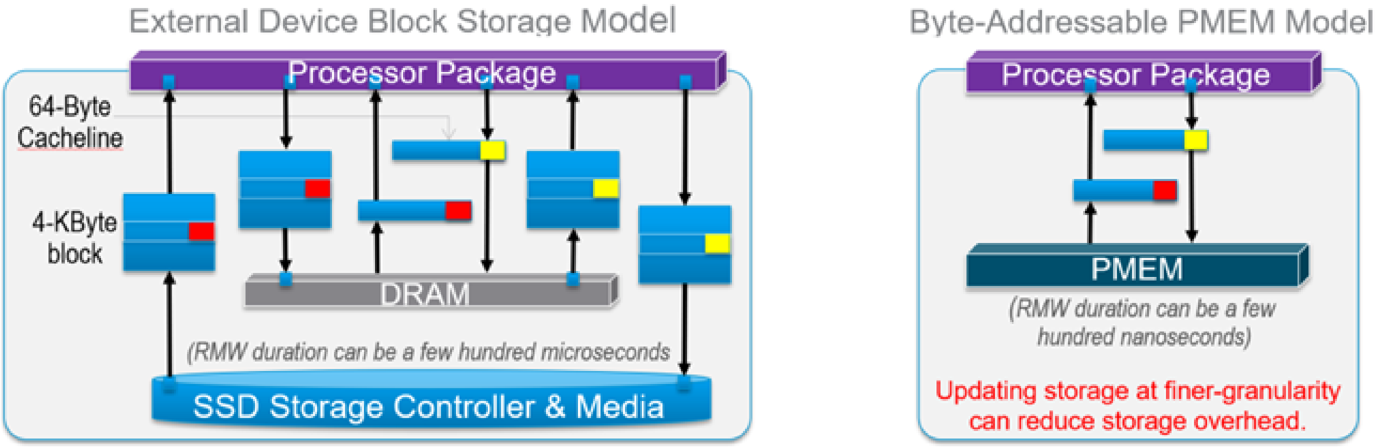
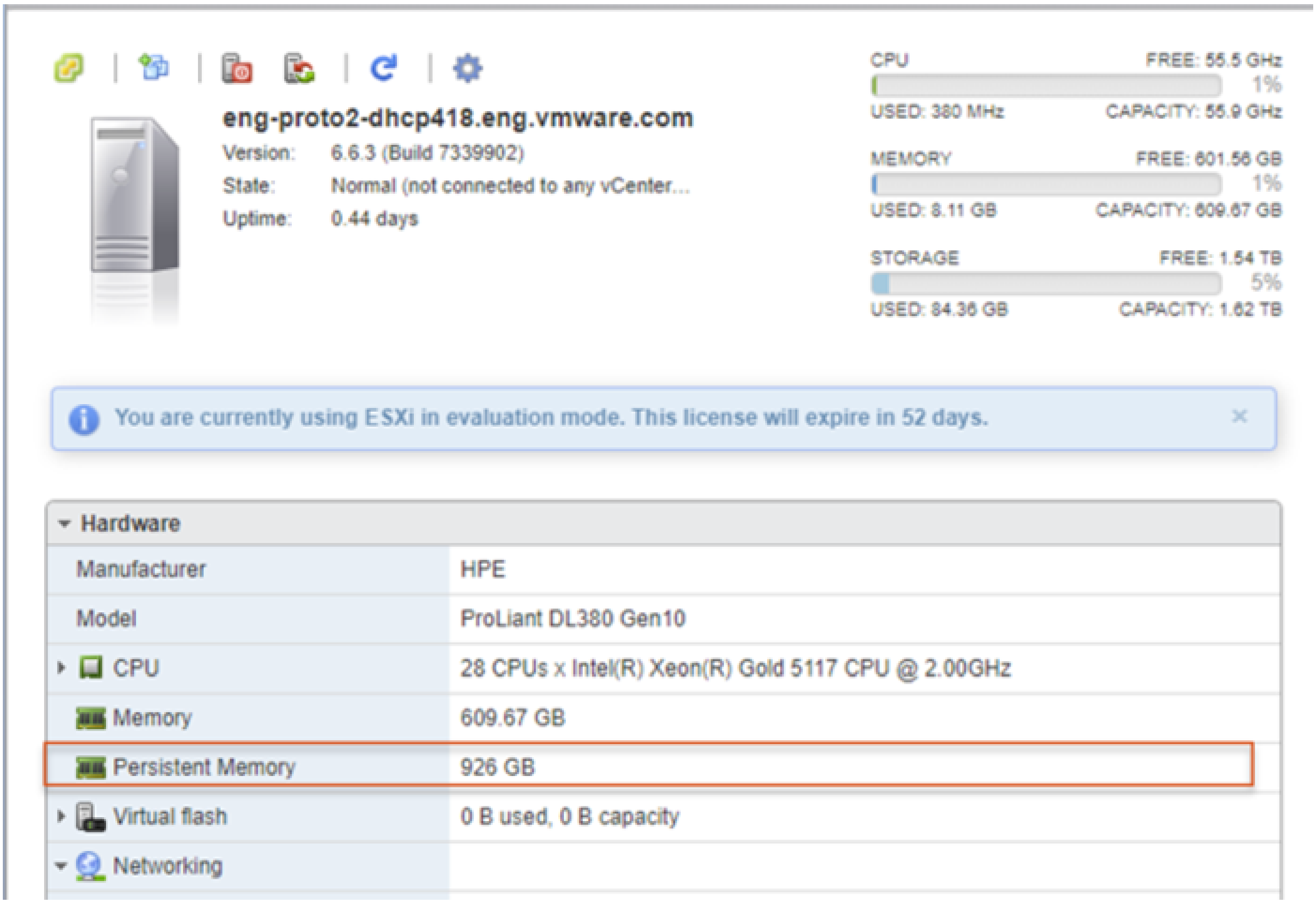

External Device Block Storage Model
Processor Package
64-Byte Cacheline
4-KByte block
DRAM
(RMW duration can be a few hundred microseconds
SSD Storage Controller & Media
Byte-Addressable PMEM Model
Processor Package
PMEM
(RMW duration can be a few hundred nanoseconds)
Updating storage at finer-granularity can reduce storage overhead.

Figure 21

With the release of vSphere 6.7, pMEM or NVDIMMs are now supported and can be used for the host and or a VM. Now applications, whether modified to use NVDIMMs or legacy VMs, can take advantage of PMEM on VMware vSphere.

When NVDIMM modules are installed in supported hardware and with vSphere 6.7, a PMem datastore is automatically created on the host. That datastore is managed by the Virtual Center and DRS, no action is required to manage.



eng-proto2-dhcp418.eng.vmware.com

Version: 6.6.3 (Build 7339902)
State: Normal (not connected to any vCenter...
Uptime: 0.44 days

CPU FREE: 55.5 GHz 1%
USED: 380 MHz CAPACITY: 55.9 GHz

MEMORY FREE: 601.56 GB 1%
USED: 8.11 GB CAPACITY: 609.67 GB

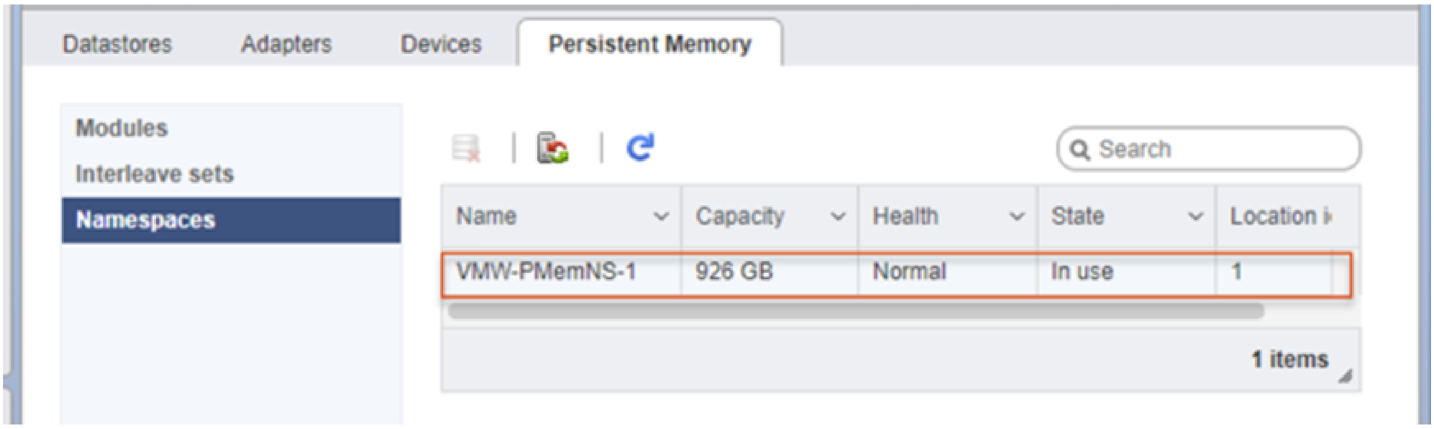
STORAGE FREE: 1.54 TB 5%
USED: 84.36 GB CAPACITY: 1.62 TB

You are currently using ESXi in evaluation mode. This license will expire in 52 days.

Hardware

Manufacturer	HPE
Model	ProLiant DL380 Gen10
CPU	28 CPUs x Intel(R) Xeon(R) Gold 5117 CPU @ 2.00GHz
Memory	609.67 GB
Persistent Memory	926 GB
Virtual flash	0 B used, 0 B capacity
Networking	

Figure 22

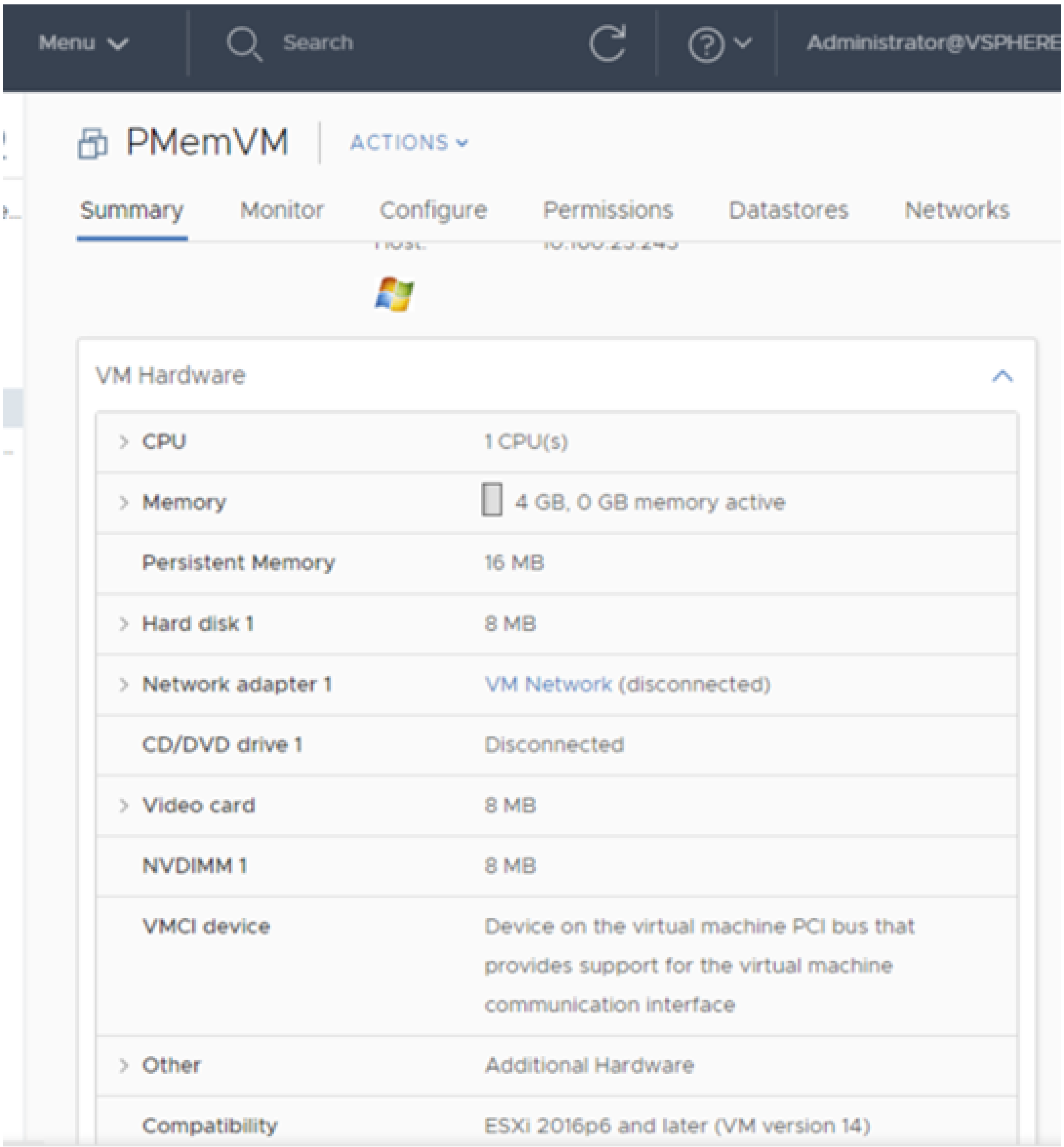


Name	Capacity	Health	State	Location i
VMW-PMemNS-1	926 GB	Normal	In use	1

1 items

Figure 23

Virtual machine support: PMem storage is supported by all hardware versions, and legacy guest OS may use the storage. Virtual NVDIMM requires hardware version 14 or higher. The OS must also support the use of PMem, for example, Windows Server 2016 and Enterprise RedHat 7.4 or later. With DRS support for PMem, VM's utilizing PMem will still have most of the virtualization features available. HA and snapshots are currently unsupported.



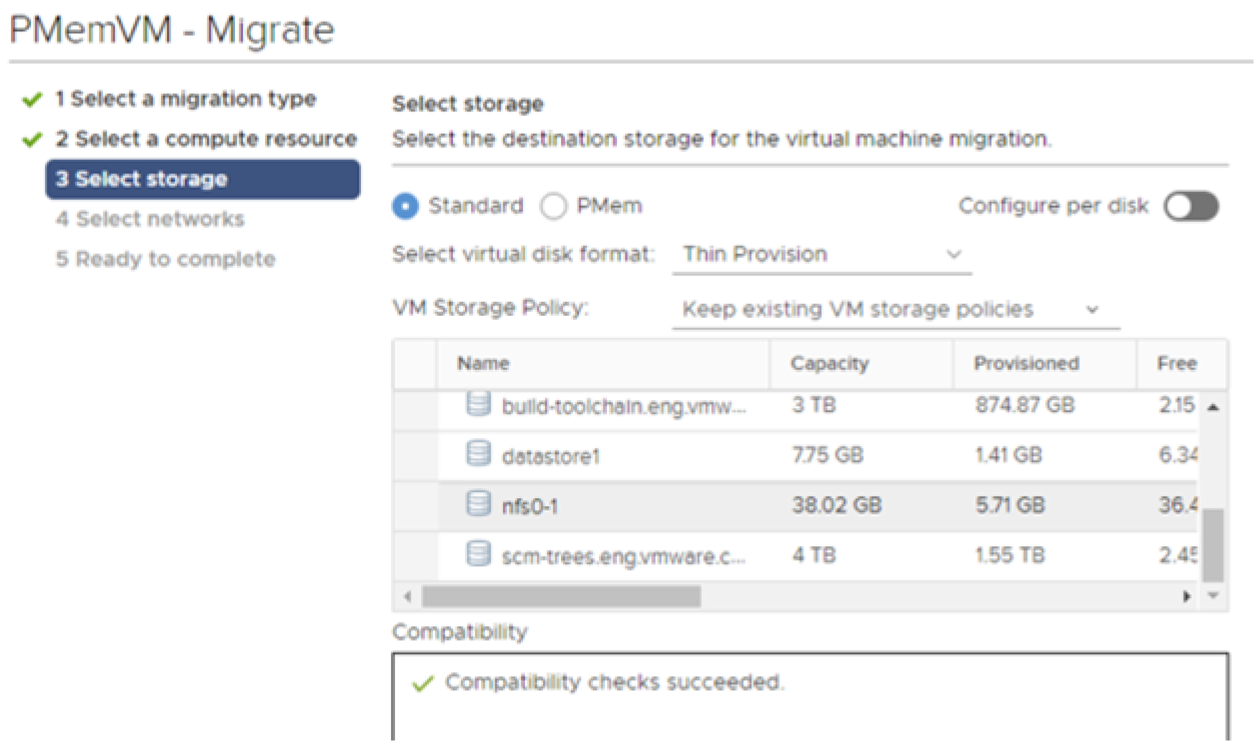
VM Hardware

CPU	1 CPU(s)
Memory	4 GB, 0 GB memory active
Persistent Memory	16 MB
Hard disk 1	8 MB
Network adapter 1	VM Network (disconnected)
CD/DVD drive 1	Disconnected
Video card	8 MB
NVDIMM 1	8 MB
VMCI device	Device on the virtual machine PCI bus that provides support for the virtual machine communication interface
Other	Additional Hardware
Compatibility	ESXi 2016p6 and later (VM version 14)

Figure 24

Virtual NVDIIMs always use persistent memory and cannot be moved to a regular datastore. They can be migrated to another host if that host also contains PMem.

Persistent Memory Disks use migrate workflows and can be moved to or from a regular datastore and can be moved between hosts as we as datastores.



PMemVM - Migrate

✓ 1 Select a migration type
✓ 2 Select a compute resource
3 Select storage
4 Select networks
5 Ready to complete

Select storage
Select the destination storage for the virtual machine migration.

● Standard ○ PMem   Configure per disk

Select virtual disk format: Thin Provision

VM Storage Policy: Keep existing VM storage policies

Name	Capacity	Provisioned	Free
build-toolchain.eng.vmw...	3 TB	874.87 GB	2.15
datastore1	7.75 GB	1.41 GB	6.34
nfs0-1	38.02 GB	5.71 GB	36.4
scm-trees.eng.vmware.c...	4 TB	1.55 TB	2.45

Compatibility

✓ Compatibility checks succeeded.

Figure 25

Putting a host in maintenance mode with the use of NVDIMMs requires a few more steps. All VMs, including those powered-off, must be moved. Ensure the PMem datastore is empty, then remove all namespaces. Then the host may be powered off, enabling you to add, remove or reconfigure NVDIMMs. Once the host has been powered back on, the PMem datastore will again be created, and DRS can be used to move VMs back to the host.

SW-FCoE (Fiber Channel over Ethernet)

Software FCoE (vmkfcoe) driver with no hardware offloaded cards required. This eliminates the need for expensive FCoE capable CNA (Converged Network Adapter). SW-FCoE will work with any L2 NIC with DCB (Data Center Bridging) capability. Software FCoE Driver can work at many speeds, including 10Gb, 20Gb, 40Gb, and 100Gb. The new driver works in a native device driver framework. The SW-FCoE driver also supports fabric and Vn2Vn (Virtual node to virtual node) connections.

Architecture

Component Interactions and communication channels involved

- Uplink creates vmkfcoe device
    - “com.vmware.com” device is a child device to uplink
- Device manager loads vmkfcoe driver
- vmkfcoe creates a vmhba (scsi adapter) on its successful loading
- “esxcli fcoe” command is used to create/delete fcoe HBAs.
- State full installation maintains config information in “esx.conf”
- Stateless installation uses host profiles to maintain config information.

The native FCoE requires no changes to “esxcli fcoe” or to the GUI/UI. The driver module registers the vmhba with the PSA layer and maintains the internal mapping between the vmnic# and vmhba#. The configurations are persistent across reboots, and the vmhba is instantiated only when the corresponding uplink device is ready.

Example of how the FCoE components interact:

- VMKCTL
  - ESXCLI issues fcoe device Create/Delete operations to Uplink Driver using vmkctl code.
- Uplink Native Driver
  - Uplink Creates FCoE (com.vmware.fcoe) Child Device Object
- FCoE Native Driver (vmkfcoe)
  - Claims FCoE device created by Native Uplink Driver
- FCoE transport functionality is managed by vmkfcoe software driver.
  - SCSI_Adapter to VMKernel

FCoE Native Driver registers vmk_ScsiAdapter (aka vmhba) using existing VMKAPI with PSA

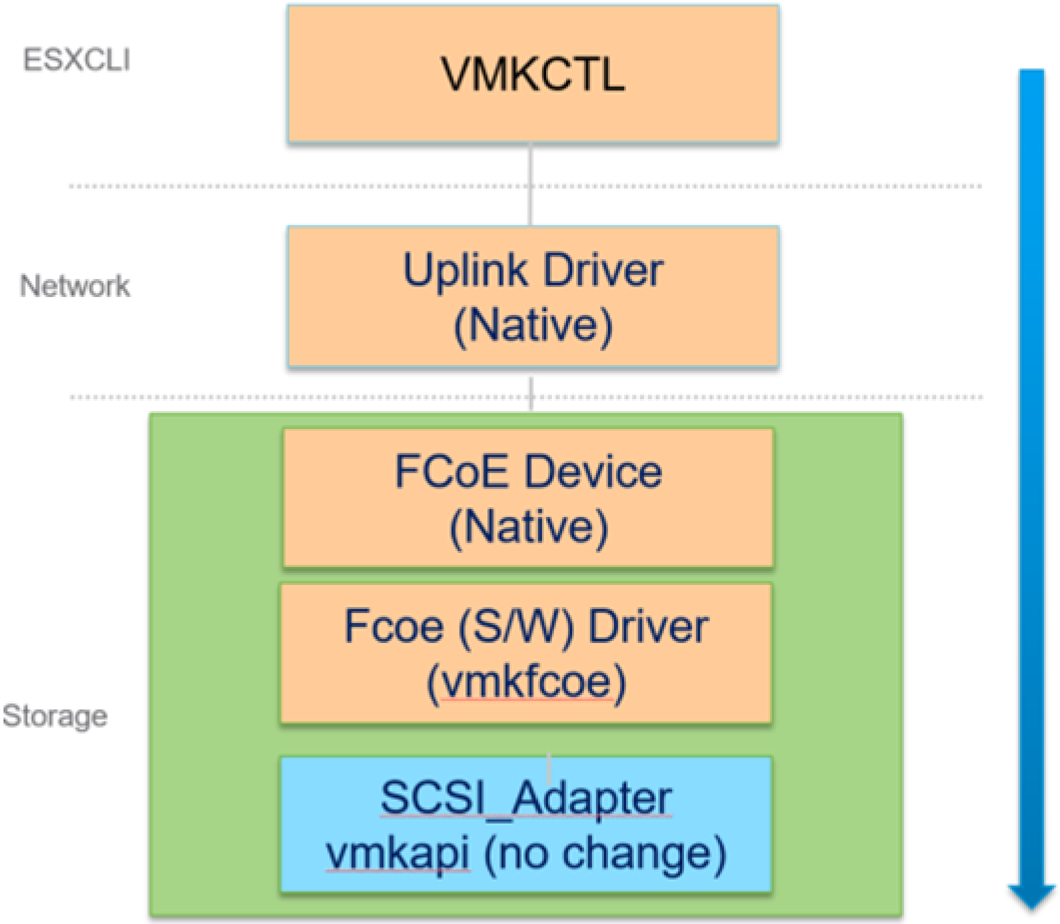
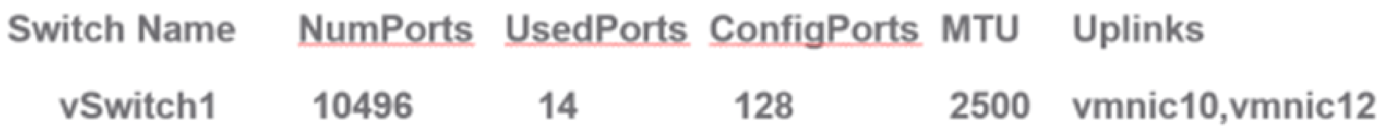


Figure 26

SW-FCoE communicates using vSwitch Uplink and requires a vSwitch to be created and paired with the uplink. Because of the stack limitations and frame size limitation of 2112 bytes, FCoE needs an MTU size of 2500 (MiniJumbo Frame). Minimum NIC/link speed supported is 10Gb.

Example switch configuration:

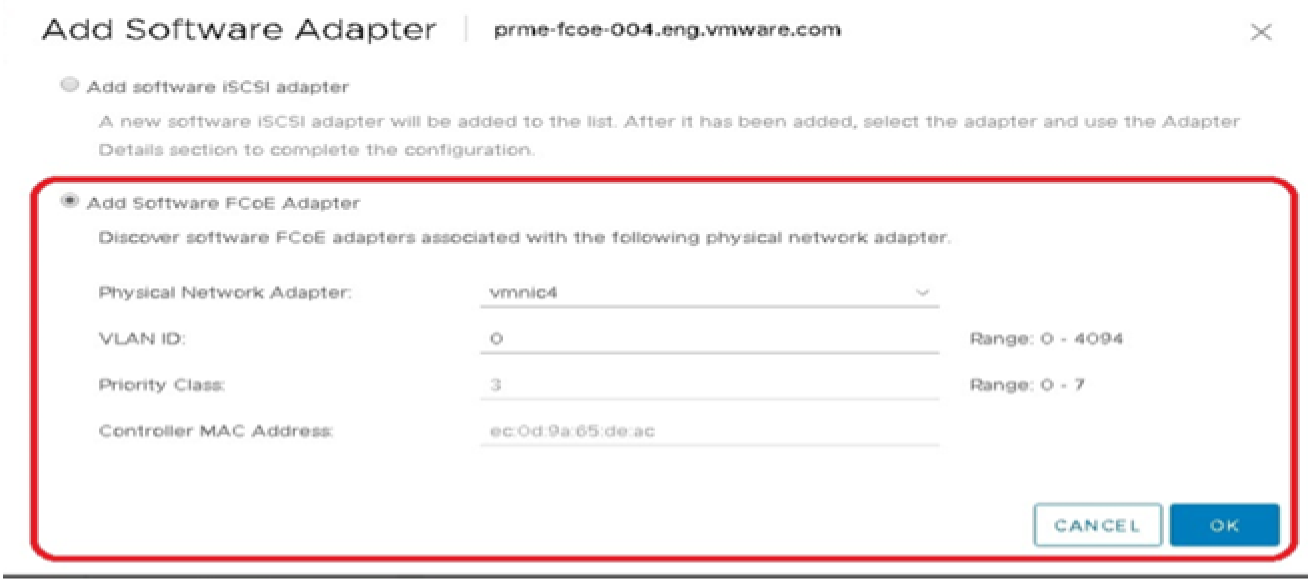


Switch Name	NumPorts	UsedPorts	ConfigPorts	MTU	Uplinks
vSwitch1	10496	14	128	2500	vmnic10,vmnic12

Figure 27

FCoE Configuration

Create HBA (VN2VN setup in esxcli only)



Add Software Adapter | prme-fcoe-004.eng.vmware.com

○ Add software iSCSI adapter

A new software iSCSI adapter will be added to the list. After it has been added, select the adapter and use the Adapter Details section to complete the configuration.

● Add Software FCoE Adapter

Discover software FCoE adapters associated with the following physical network adapter.

Physical Network Adapter: vmnic4

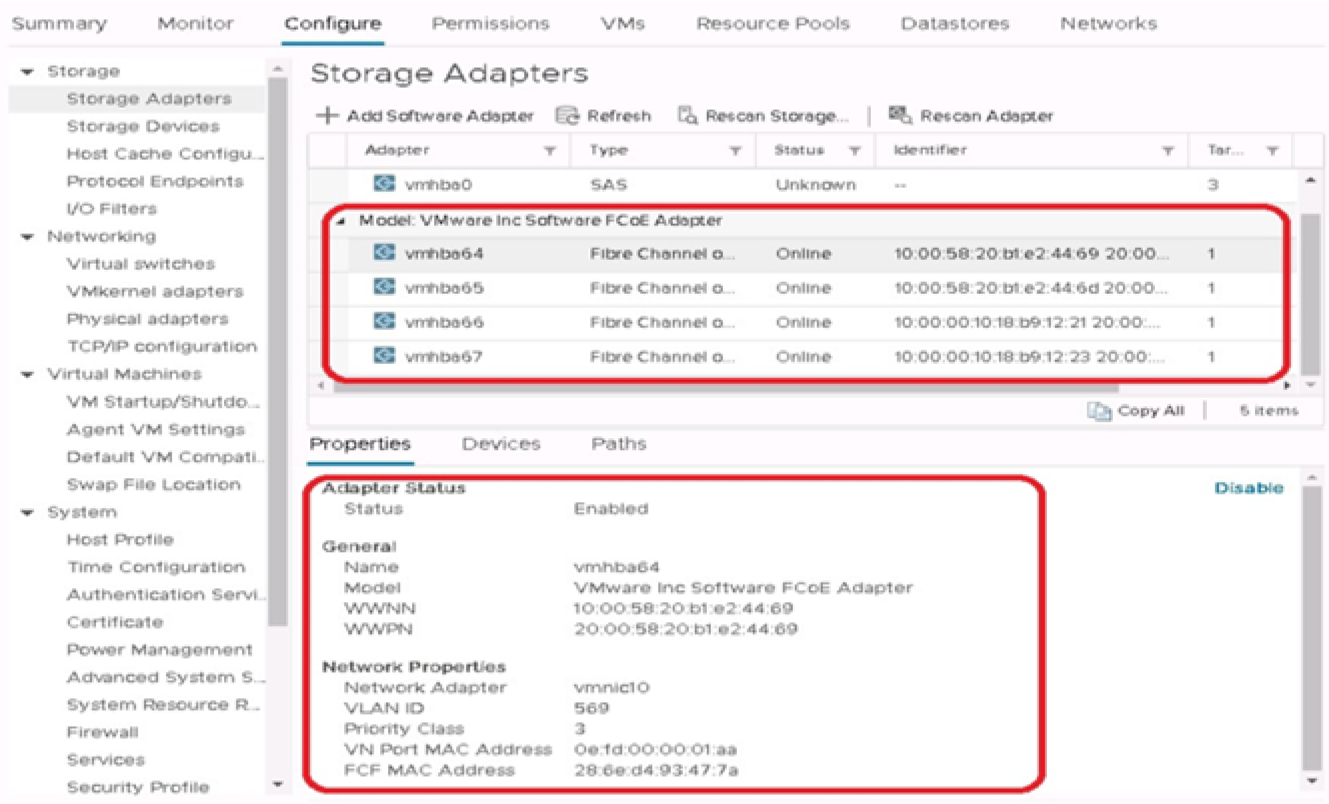
VLAN ID: 0 Range: 0 - 4094

Priority Class: 3 Range: 0 - 7

Controller MAC Address: ec:0d:9a:65:de:ac

Figure 28

FCOE Configuration – VC UI – Storage Adapters



Storage Adapters

Adapter	Type	Status	Identifier	Tar...
vmhba0	SAS	Unknown	--	3
Model: VMware Inc Software FCoE Adapter				
vmhba64	Fibre Channel o...	Online	10:00:58:20:bl:e2:44:69 20:00...	1
vmhba65	Fibre Channel o...	Online	10:00:58:20:bl:e2:44:6d 20:00...	1
vmhba66	Fibre Channel o...	Online	10:00:00:10:18:b9:12:21 20:00:...	1
vmhba67	Fibre Channel o...	Online	10:00:00:10:18:b9:12:23 20:00:...	1

Adapter Status

Status	Enabled

General

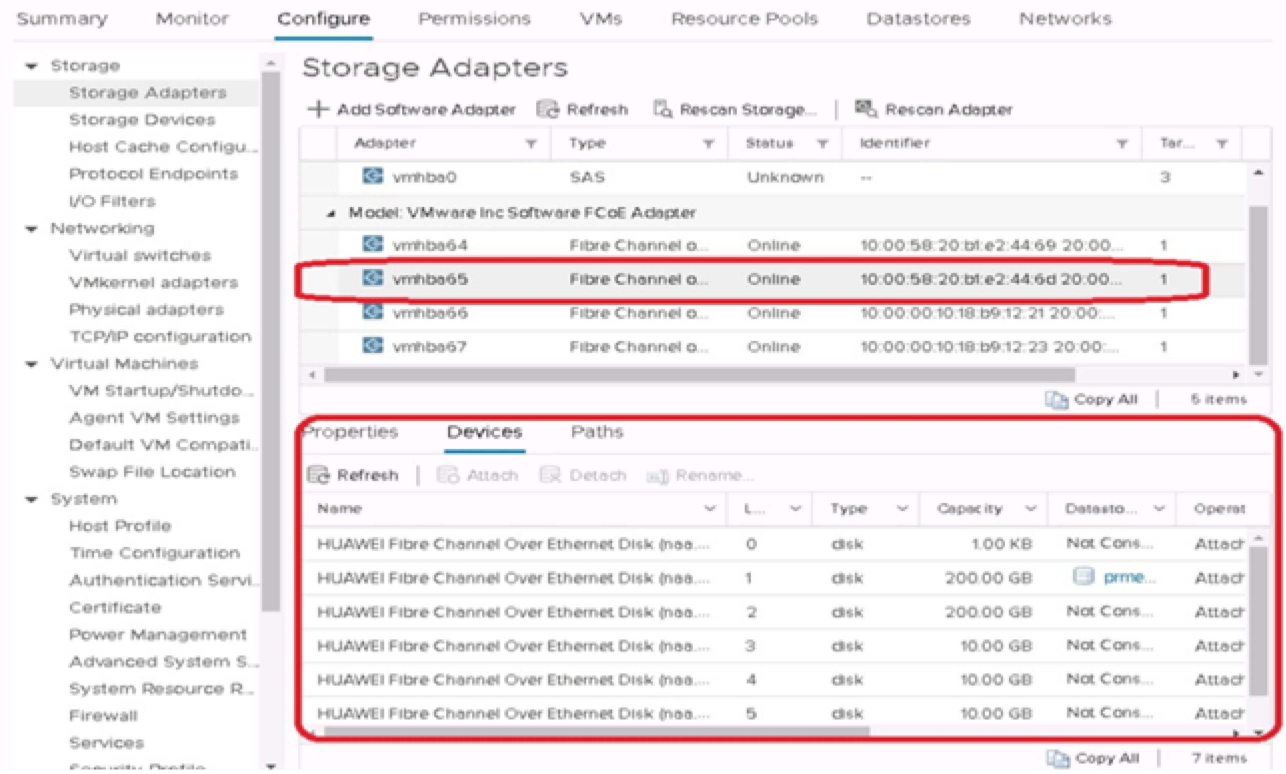
Name	vmhba64
Model	VMware Inc Software FCoE Adapter
WWNN	10:00:58:20:b1:e2:44:69
WWPN	20:00:58:20:b1:e2:44:69

Network Properties

Network Adapter	vmnic10
VLAN ID	569
Priority Class	3
VN Port MAC Address	0e:fd:00:00:01:aa
FCF MAC Address	28:6e:d4:93:47:7a

Figure 29

FCOE Configuration – VC UI – Devices



Storage Adapters

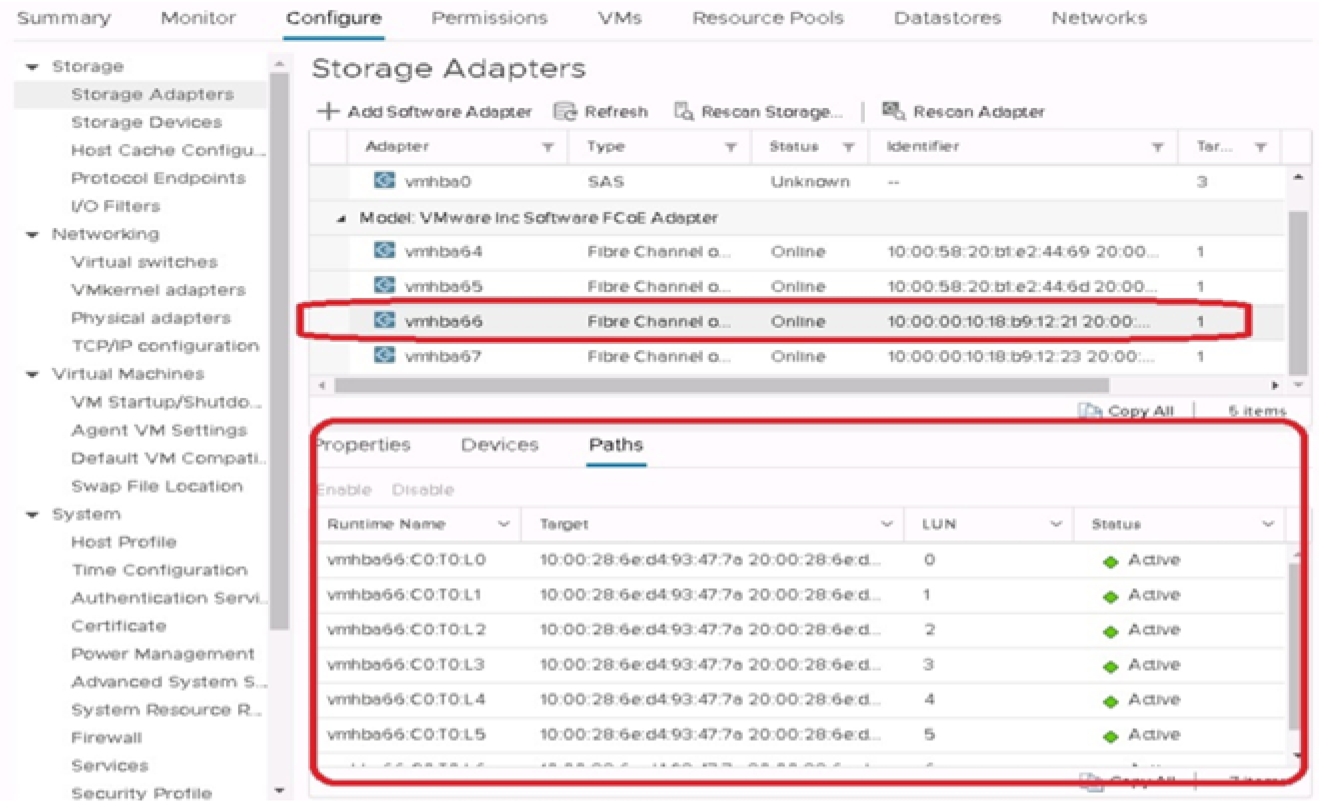
Adapter	Type	Status	Identifier	Tar...
vmhba0	SAS	Unknown	--	3
Model: VMware Inc Software FCoE Adapter				
vmhba64	Fibre Channel o...	Online	10:00:58:20:bl:e2:44:69 20:00...	1
vmhba65	Fibre Channel o...	Online	10:00:58:20:bl:e2:44:6d 20:00...	1
vmhba66	Fibre Channel o...	Online	10:00:00:10:18:b9:12:21 20:00:...	1
vmhba67	Fibre Channel o...	Online	10:00:00:10:18:b9:12:23 20:00:...	1

Name	L...	Type	Capacity	Datasto...	Operat
HUAWEI Fibre Channel Over Ethernet Disk (naa....	0	disk	1.00 KB	Not Cons...	Attach
HUAWEI Fibre Channel Over Ethernet Disk (naa....	1	disk	200.00 GB	prme...	Attach
HUAWEI Fibre Channel Over Ethernet Disk (naa....	2	disk	200.00 GB	Not Cons...	Attach
HUAWEI Fibre Channel Over Ethernet Disk (naa....	3	disk	10.00 GB	Not Cons...	Attach
HUAWEI Fibre Channel Over Ethernet Disk (naa....	4	disk	10.00 GB	Not Cons...	Attach
HUAWEI Fibre Channel Over Ethernet Disk (naa....	5	disk	10.00 GB	Not Cons...	Attach

Figure 30

© VMware LLC.

FCOE Configuration – VC GUI – Paths



Adapter	Type	Status	Identifier	Tar...
vmhba0	SAS	Unknown	--	3
Model: VMware Inc Software FCoE Adapter				
vmhba64	Fibre Channel o...	Online	10:00:58:20:bf:e2:44:69 20:00...	1
vmhba65	Fibre Channel o...	Online	10:00:58:20:bf:e2:44:6d 20:00...	1
vmhba66	Fibre Channel o...	Online	10:00:00:10:18:b9:12:21 20:00:...	1
vmhba67	Fibre Channel o...	Online	10:00:00:10:18:b9:12:23 20:00:...	1

Runtime Name	Target	LUN	Status
vmhba66:C0:T0:L0	10:00:28:6e:d4:93:47:7a 20:00:28:6e:d...	0	◆ Active
vmhba66:C0:T0:L1	10:00:28:6e:d4:93:47:7a 20:00:28:6e:d...	1	◆ Active
vmhba66:C0:T0:L2	10:00:28:6e:d4:93:47:7a 20:00:28:6e:d...	2	◆ Active
vmhba66:C0:T0:L3	10:00:28:6e:d4:93:47:7a 20:00:28:6e:d...	3	◆ Active
vmhba66:C0:T0:L4	10:00:28:6e:d4:93:47:7a 20:00:28:6e:d...	4	◆ Active
vmhba66:C0:T0:L5	10:00:28:6e:d4:93:47:7a 20:00:28:6e:d...	5	◆ Active

Figure 31

vSphere 6.7 U1 Enhanced Round Robin Load Balancing

Round Robin PSP

For many vendors, the default policy is VMW_PSP_RR (Round Robin) for their PSP. With RR, the policy uses an algorithm for path selection rotating through the configured paths. It can be used with active/active and active/passive connectivity to implement balancing across paths for different LUNs. PSP_RR shares IO equally between paths in a Round Robin sequence.

Problem

With the current PSP policies, none of them consider the latency of the paths when making a path selection for IO. There could be latency due to switch congestion, overloaded targets, or hops in a path that can add to latency and are not considered. Consequently, we end up with sub-optimal bandwidth utilization, and slow response times for guest applications.

Solution

Consider both latency and pending IOs to decide the optimal path for IO. By doing this we get a bigger picture of the path's bandwidth and latency.

New with vSphere 6.7 U1 Enhanced Load Balancing Path Selection Policy

New Policy uses logic to monitor available paths to determine the path(s) with the lowest latency. It then directs IO down those paths for a set amount of time before re-evaluating latency and load to make another path selection.

For a deep dive into the Round Robin Latency Path Selection Policy, see the Deep Dive Latency Round Robin Path Selection Policy article.

Acknowledgments and Author

Acknowledgments

Shashank Rajvanshi, Satya Kanungo, Prasanna Aitha, Rajeev Raghotham, Dipankar Sinha, Darshan Murthy, Bharat Tewari, Arun Lakshmipathy, Thorbjoern Donbaek, Ika Vaze, Adam Eckerle, Shahnawaz Pahelwa, Ramya Victor, Rahul Dev, Nitish Kumar, Ramya Manjanna Gowda, Vinay Pravin Bapa, Gautham Swamy, Bipin Krishnappa, Bulat Shelepov, Jinpyo Kim, Saurabh Goswami, Divya Sethi, Cormac Hogan, Duncan Epping, Sudhir Balasubramanian, Kristopher Groh, the Storage GSS team in Bangalore

About the Author

Jason Massae (Twitter @jbmassae) is the vVols and Core Storage Staff Technical Marketing Architect for VMware. Focusing on vSphere vVols and Core Storage technologies as well as working with VMware's storage partners to deliver storage collateral. He came from one of the largest flash and memory manufacturers in the world where he architected and lead global teams in virtualization strategies. Working with VMware, he helped test and validate SSDs for VMware vSAN.

© VMware LLC.

Copyright © 2005-2024 Broadcom. All Rights Reserved. The term “Broadcom” refers to Broadcom Inc. and/or its subsidiaries.