

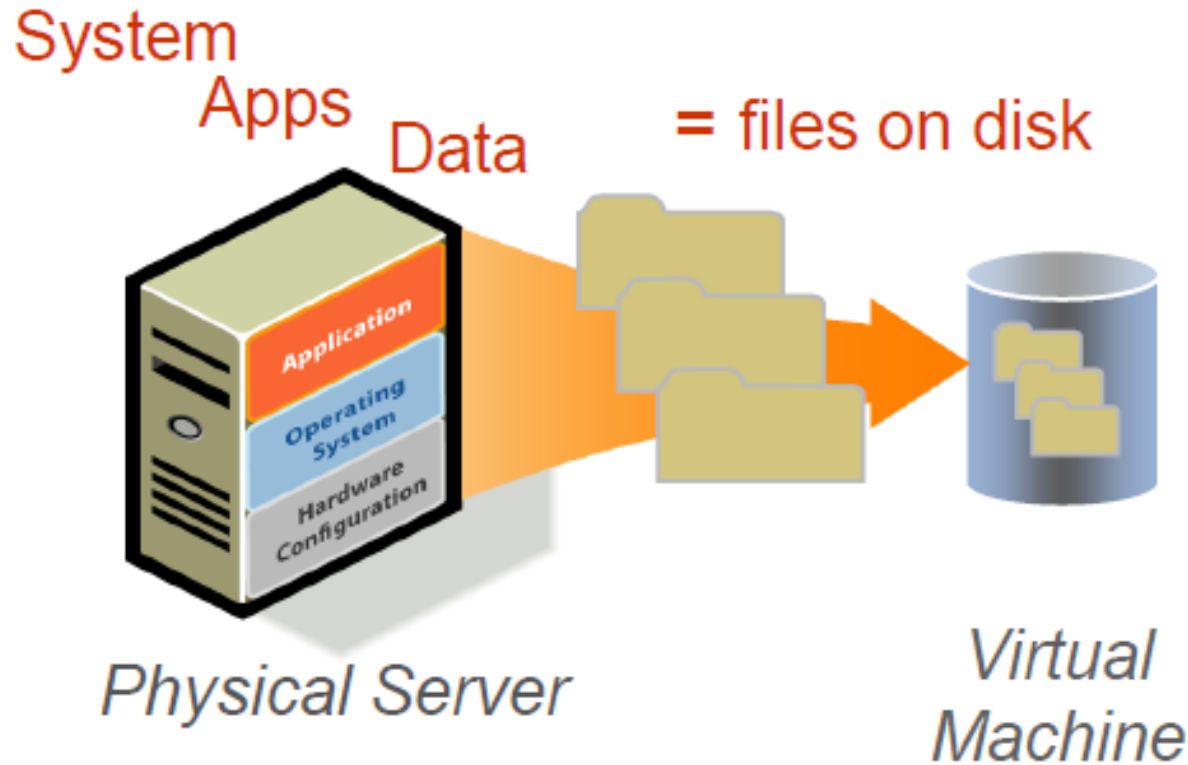
VMware storage concepts, APIs and performance best practices

Aditya Vasudevan

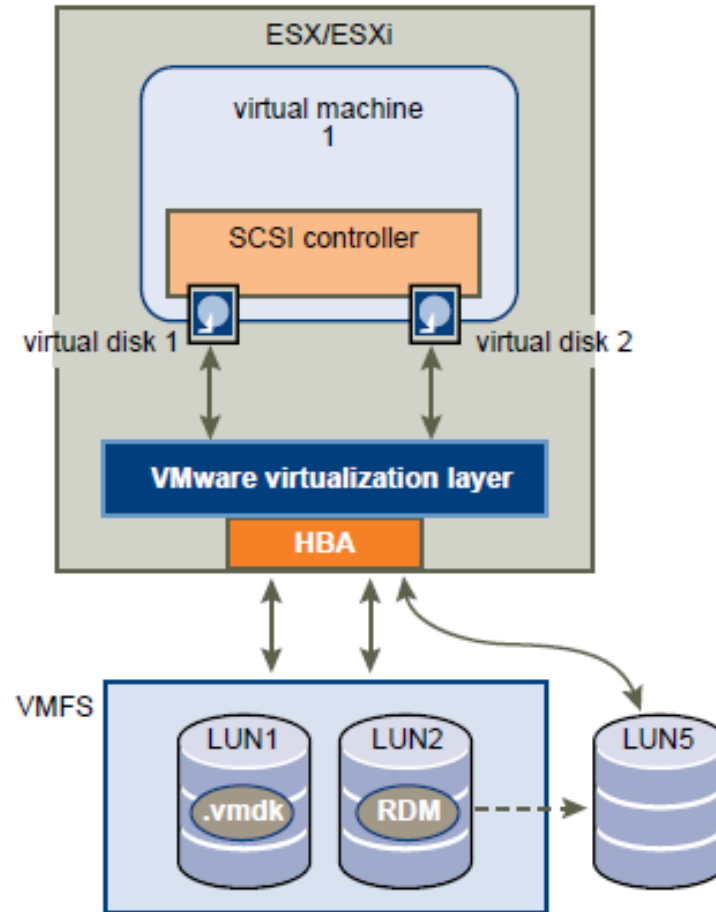
Patni Americas, Inc

- Basic VMware Storage concepts
- Storage best practices in ESX
- Storage APIs in VMware
- Q&A

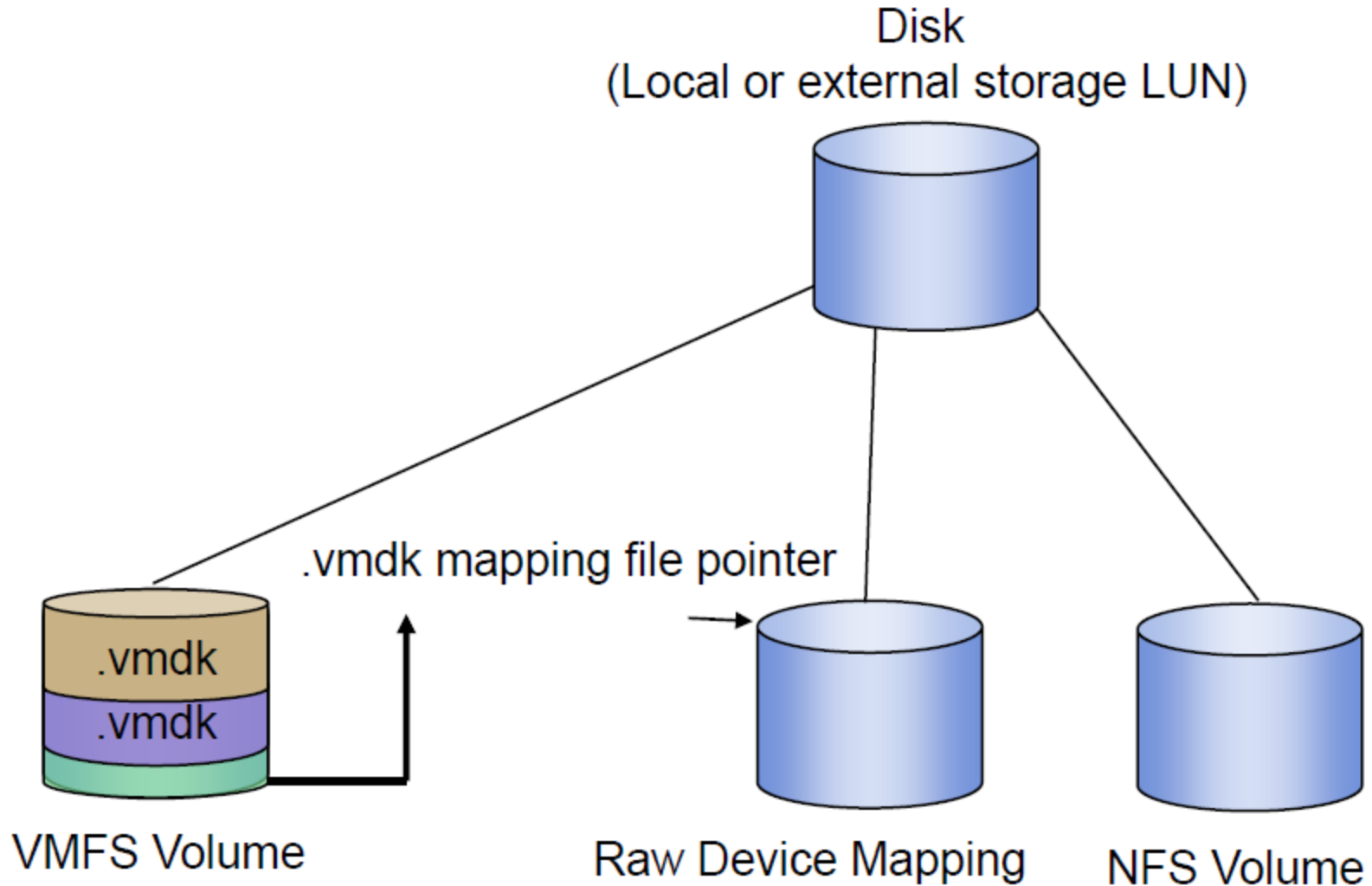
Virtual Machine is a file



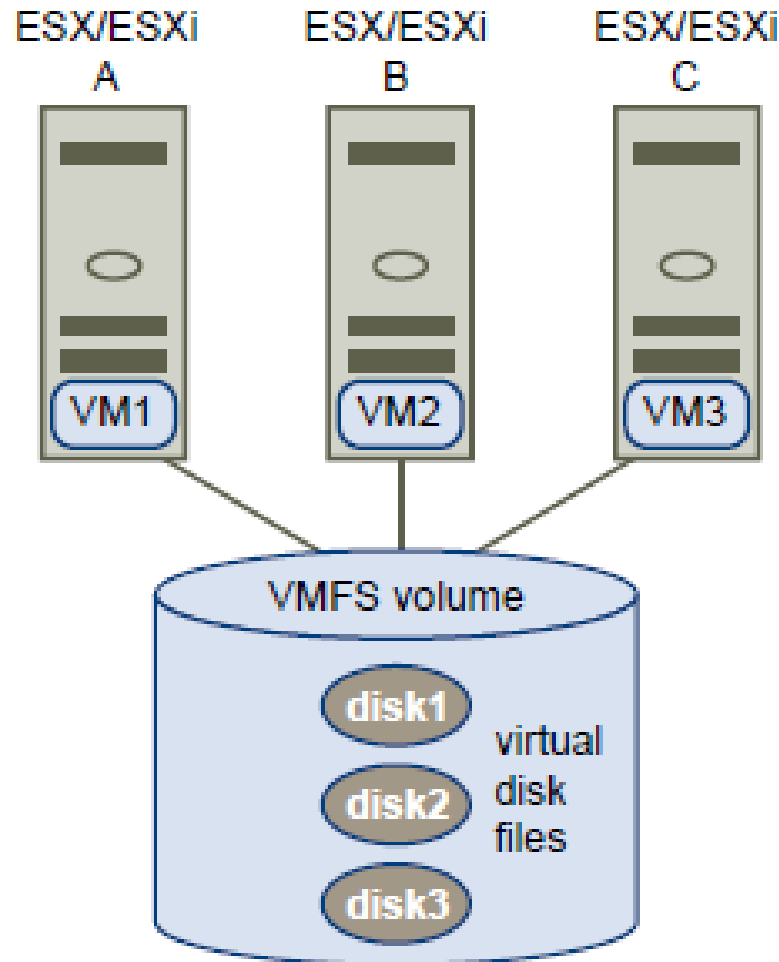
SAN storage virtualization



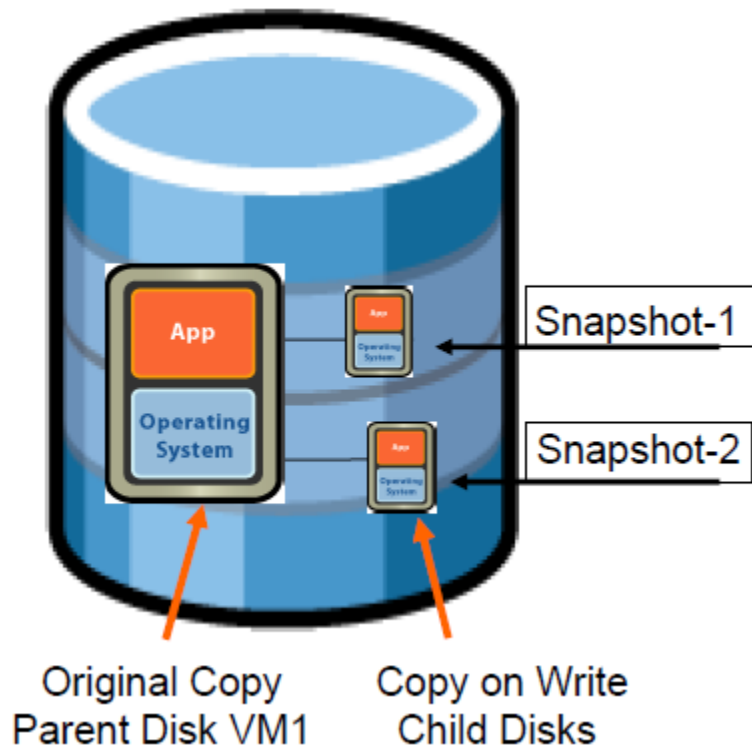
VMware disk options



Sharing VMFS across hosts

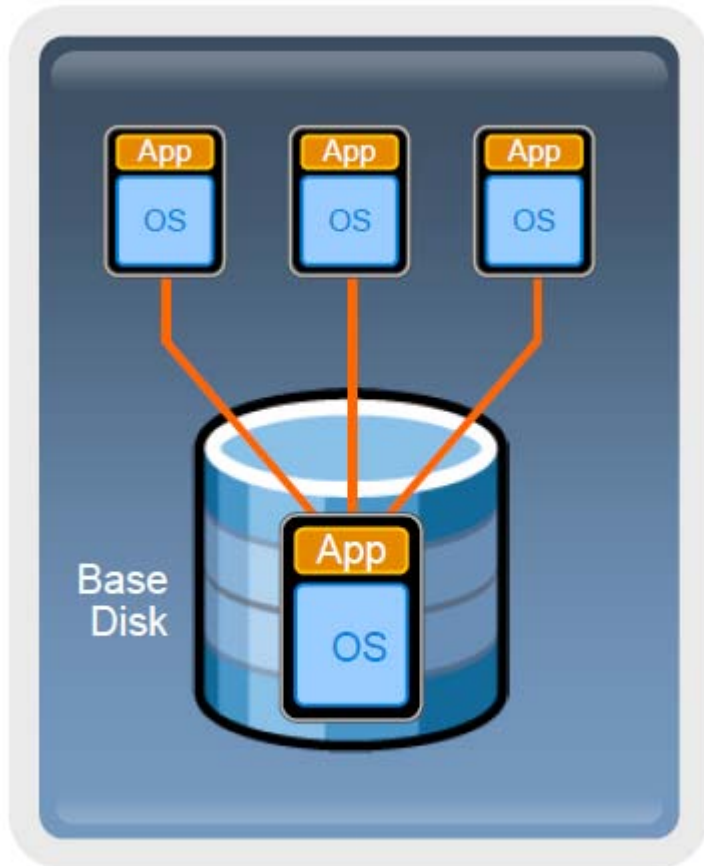


Snapshots



- ❑ VMFS feature
- ❑ Instant snap
- ❑ Copy on write
- ❑ Basis of DP API

Linked clones



- ❑ Many VMs share a disk
- ❑ Private write disk
- ❑ Updates on base disk are seen by all VMs
- ❑ Improves storage utilization & saves cost.

Importance of SAN for ESX

- ❑ These features depend on some external storage:
 - ❑ vMotion
 - ❑ High Availability
 - ❑ Distributed Resource Scheduling
 - ❑ DR using Site Recovery Manager

Some best practices

- ❑ Each LUN should have the correct RAID level and storage characteristic for applications in virtual machines that use it.
- ❑ One LUN must contain only one VMFS datastore.
- ❑ If multiple virtual machines access the same VMFS, use disk shares to prioritize virtual machines.
- ❑ Few large LUNs or many small LUNs ?
- ❑ For multipathing to work properly, each LUN must present the same LUN ID number to all ESX/ESXi hosts.

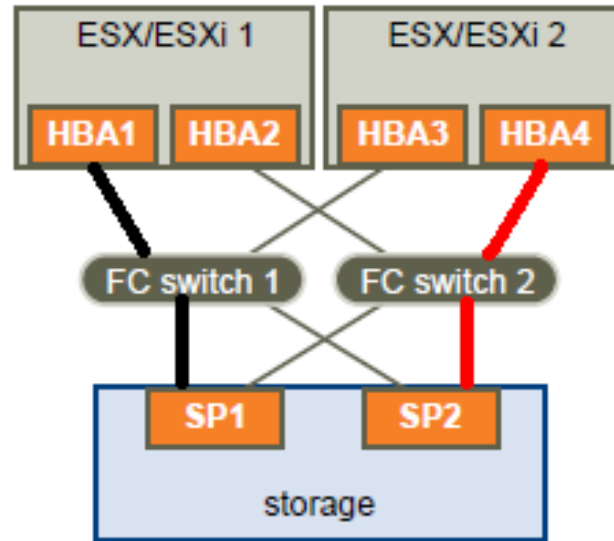
- ❑ Make sure the driver you use in the guest operating system specifies a large enough queue. You can set the queue depth for the physical HBA during system setup.
- ❑ On virtual machines running Microsoft Windows, increase the value of the SCSI TimeoutValue parameter to 60. This increase allows Windows to better tolerate delayed I/O resulting from path failover.

- ❑ You cannot use virtual machine multipathing software to perform I/O load balancing to a single physical LUN.
- ❑ When you use VMotion or DRS with an active/passive SAN storage device, make sure that all ESX/ESXi systems have consistent paths to all storage processors. Not doing so can cause path thrashing when a VMotion migration occurs.

- ❑ The RAID group containing the ESX/ESXi LUNs should not include LUNs used by other hosts that are not running ESX/ESXi for I/O intensive applications.
- ❑ Enable array cache
- ❑ Use multi-pathing for static load balancing
- ❑ Adjust LUN queue depth
- ❑ Align Partition for VMFS and guest FS.

- ❑ SIOC
- ❑ Ensure CPU cycles are available for disk IO
- ❑ Choose right disk adapter – LSI/pvscsi
- ❑ Meta-data intensive operation at off peaks hours
- ❑ Adjust Disk.SchedNumReqOutstanding
- ❑ Independent nonpersistent disk /Snapshot incur performance penalty
- ❑ Thick eager zeroed disks are better than lazy zero or thin disk

What is Path thrashing ?



Path thrashing - causes

- ❑ Array is of type Active-passive.
- ❑ Two hosts access the same LUN using different storage processors (SPs). For example, the LUN is configured to use the Fixed PSP. On Host A, the preferred path to the LUN is set to use a path through SP A. On Host B, the preferred path to the LUN is configured to use a path through SP B.

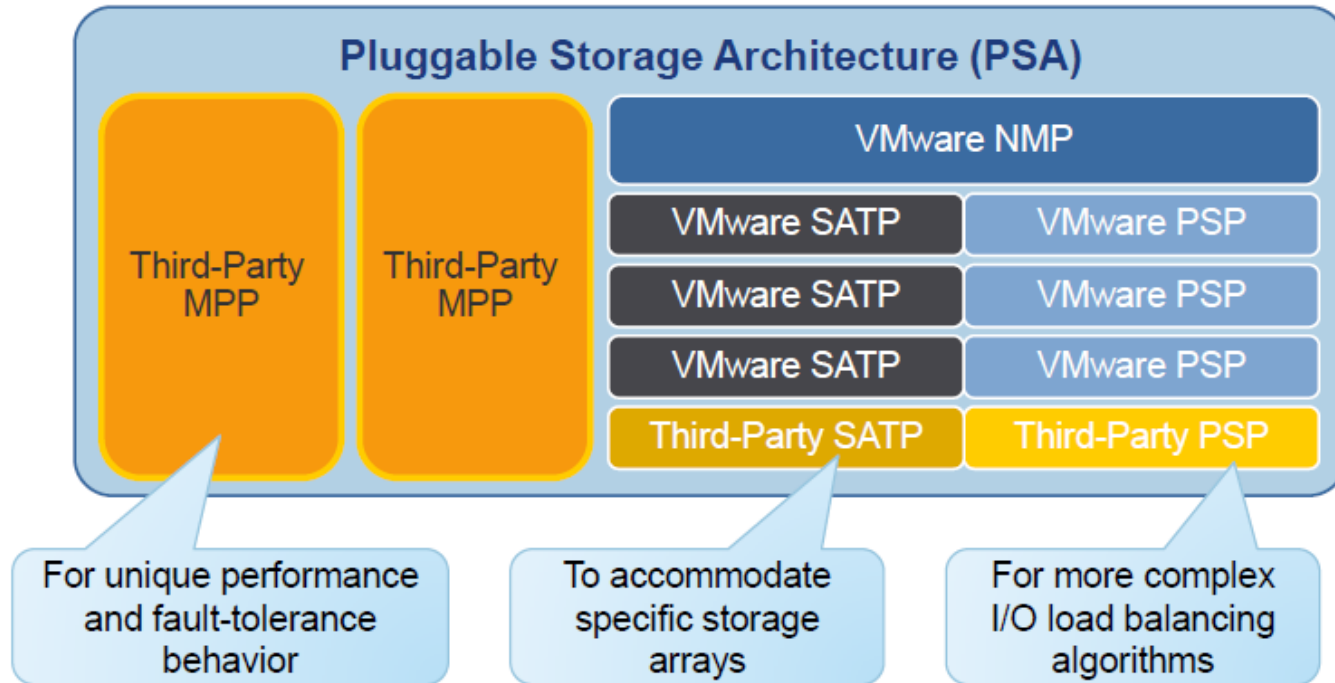
Path thrashing - resolution

- ❑ Ensure that all hosts sharing the same set of LUNs on the active-passive arrays use the same storage processor.
- ❑ Correct any cabling inconsistencies between different ESX/ESXi hosts and SAN targets so that all HBAs see the same targets in the same order.
- ❑ Configure the path to use the Most Recently Used PSP (the default).

- Data protection
- PSA
- VAAI

- ❑ Thin Provisioning Support
 - ❑ Only backup the “occupied” portion of a thin provisioned disk
 - ❑ This automatically removes the 0 byte blocks that haven’t been allocated yet
- ❑ Discovery of changed data
 - ❑ No more “snap and scan” –vStorage APIs allow vendors to identify changed data blocks
 - ❑ Near continuous replication

vStorage API - PSA



- ❑ Full copy enables the storage arrays to make full copies of data within the array without having the ESX server read and write the data.
- ❑ Copy is faster – storage vMotion/cloning/deploying VM from template
- ❑ Avoids CPU consumption on the host
- ❑ Avoids network congestion
- ❑ Avoids Storage front-end controller IO.

- ❑ To have the array complete zeroing out of a disk speeds up a standard initialization process using the copy same primitive.
- ❑ E.g. Creation of eager zero thick disk is faster
- ❑ Without this primitive zeroing is a time consuming blocking operation

- ❑ Hardware assisted locking to protect VMFS meta data instead of scsi reservation
- ❑ Its granular block level locking vs LUN level scsi reservation
- ❑ Results in better IO performance with administrative activities in parallel (VM creation, snapshots, etc)

- ❑ ESX has powerful storage capabilities
- ❑ Can achieve good performance through best practices
- ❑ Opportunities to use APIs for integration

Q&A