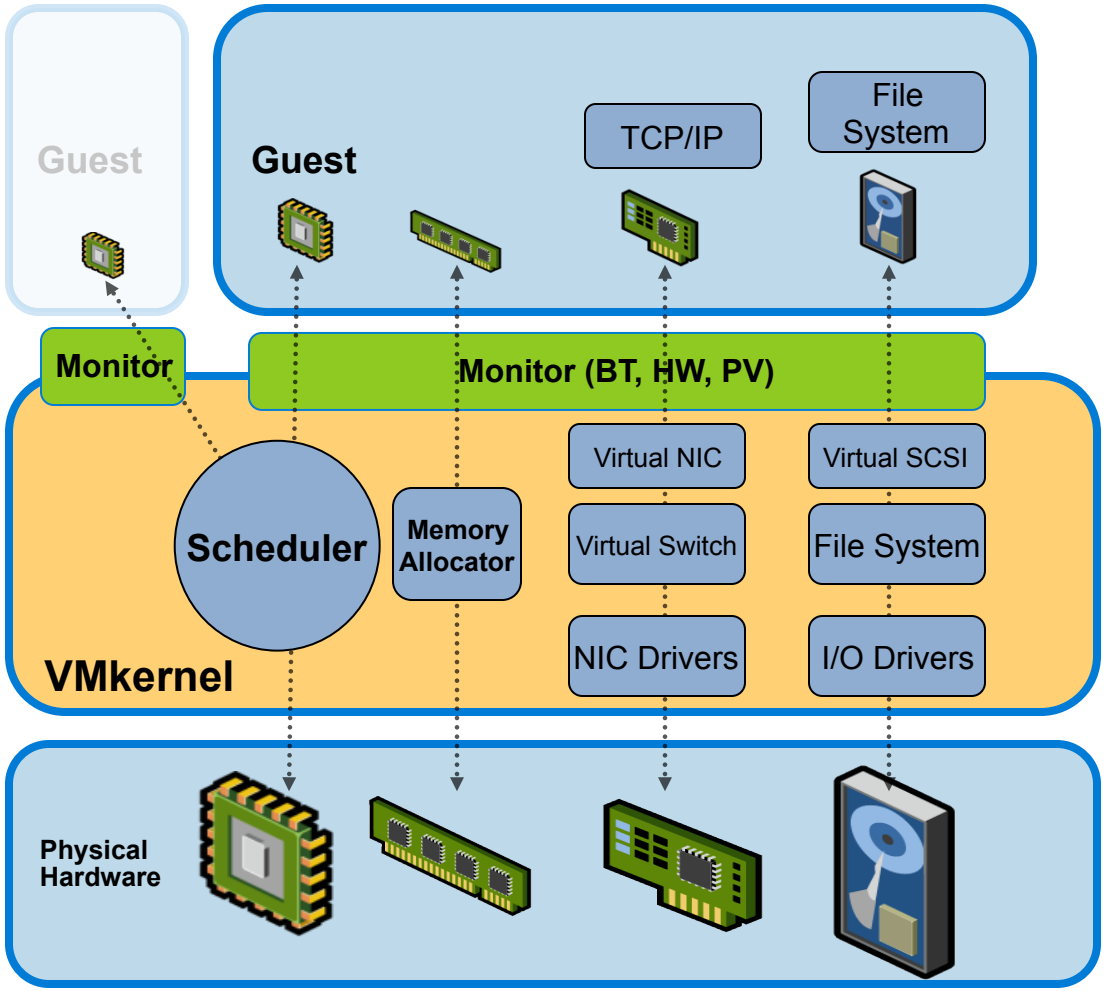
VMware Performance Overview

Virtualizing Demanding Applications

Scott Drummonds

Group Manager, Technical Marketing

VMware ESX Architecture



CPU is controlled by scheduler and virtualized by monitor

Monitor supports:
- BT (Binary Translation)
- HW (Hardware assist)
- PV (Paravirtualization)

Memory is allocated by the VMkernel and virtualized by the monitor

Network and I/O devices are emulated and proxied though native device drivers

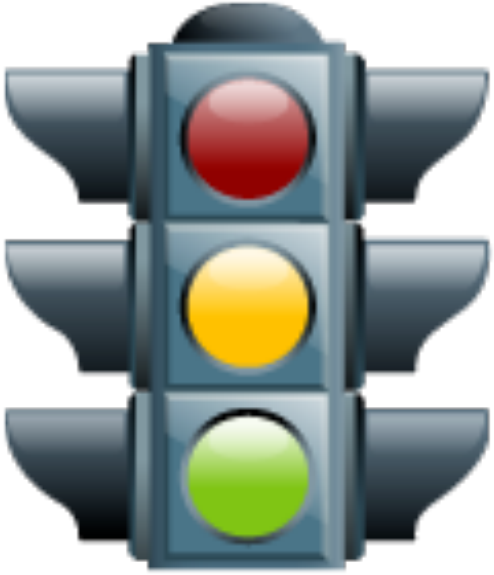
Can *Your* Application Be Virtualized?

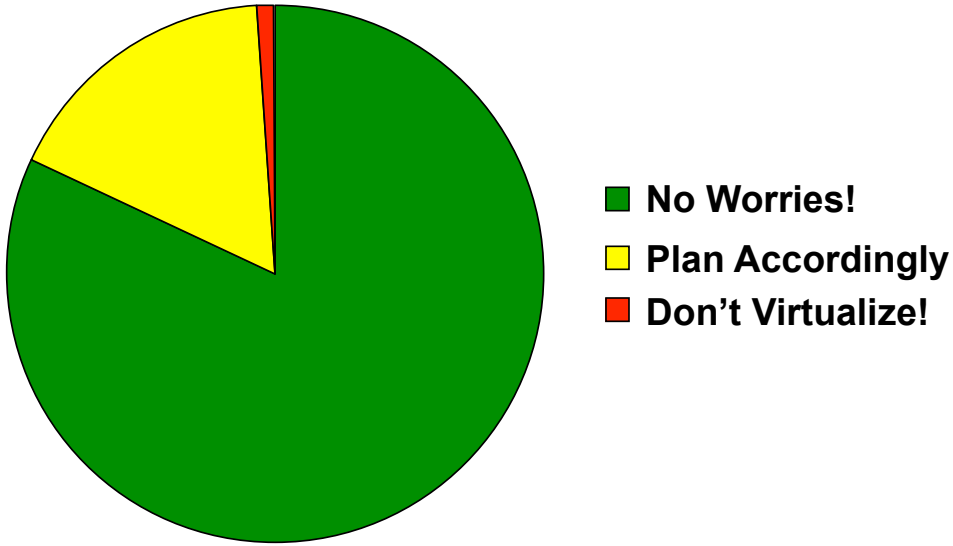
Red:
Exceeds capabilities of virtual platform

Yellow:
Runs well under right conditions

Green:
Runs perfectly out-of-box

Can *Your* Application Be Virtualized?





Characterizing and Categorizing Applications

CPU

- CPU- and memory-intensive—Green with latest hardware assist technology

Network bandwidth

- Current high water mark at 16 Gb/s (SPECweb2005 on VI3)
- 1-16 Gb/s: Yellow, >16 Gb/s: Red

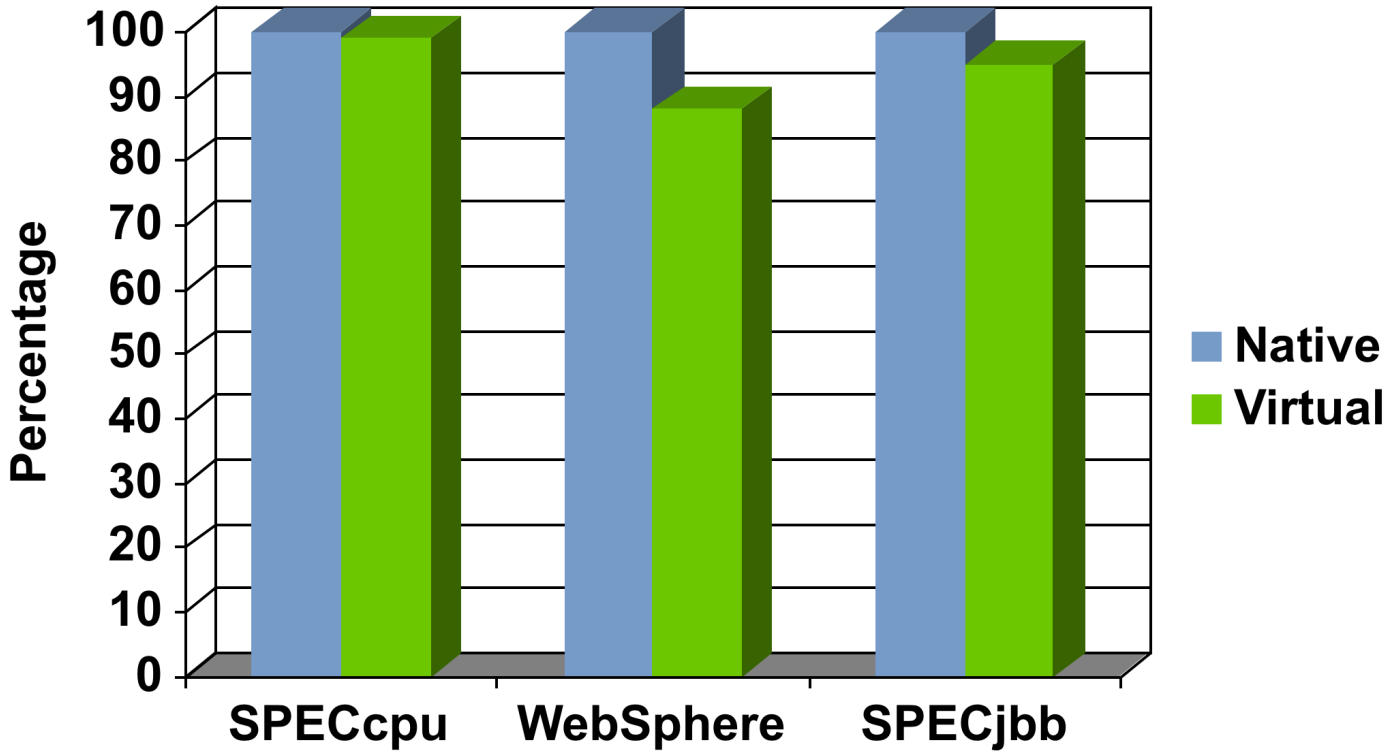
Storage bandwidth

- Maximum measured: 100,000 IOPS on 3.5, 300,000 IOPS on 4.0
- 10-300K IOPS: Yellow, > 300K IOPS: Red

Resource Configuration

- ESX 3.5: Greater than four vCPUs, 64 GB RAM: Red
- ESX 4.0: Greater than eight vCPUs, 255 GB RAM: Red

CPU Bound Workloads Usually “Green”



SPECcpu results: http://www.vmware.com/pdf/asplos235_adams.pdf

Websphere results published jointly by IBM/VMware

SPEC results used for comparison only and not submitted to SPEC

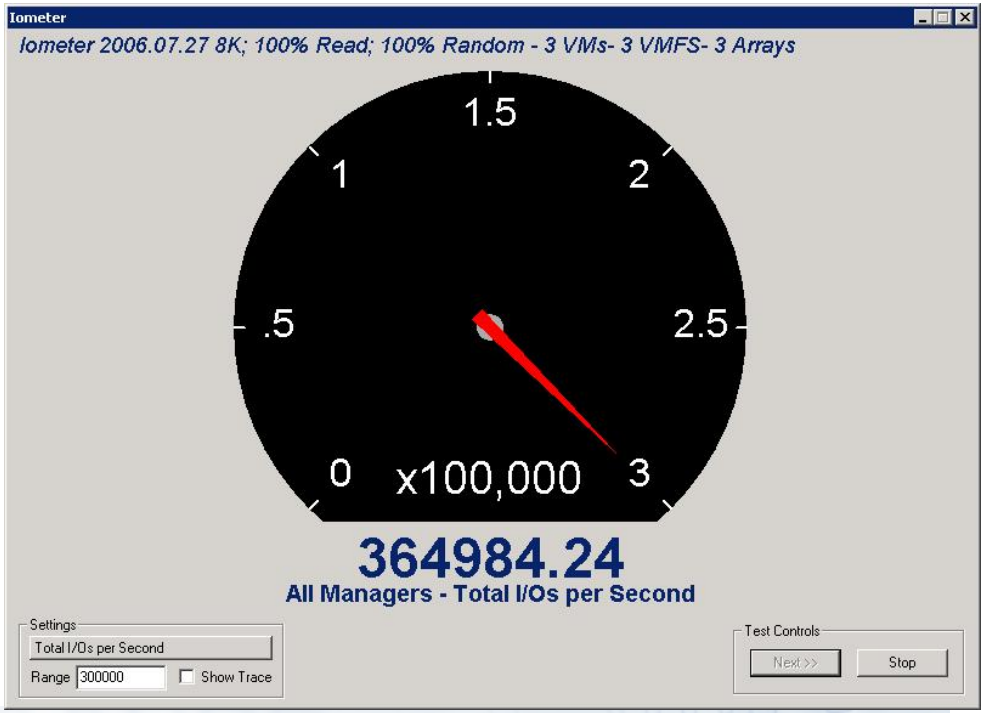
I/O Utilization Above Maximums: Usually "Red"

Maximum reported storage: 365K IOPS

- 100K on VI3

Maximum reported network: 16 Gb/s

- Measured on VI3



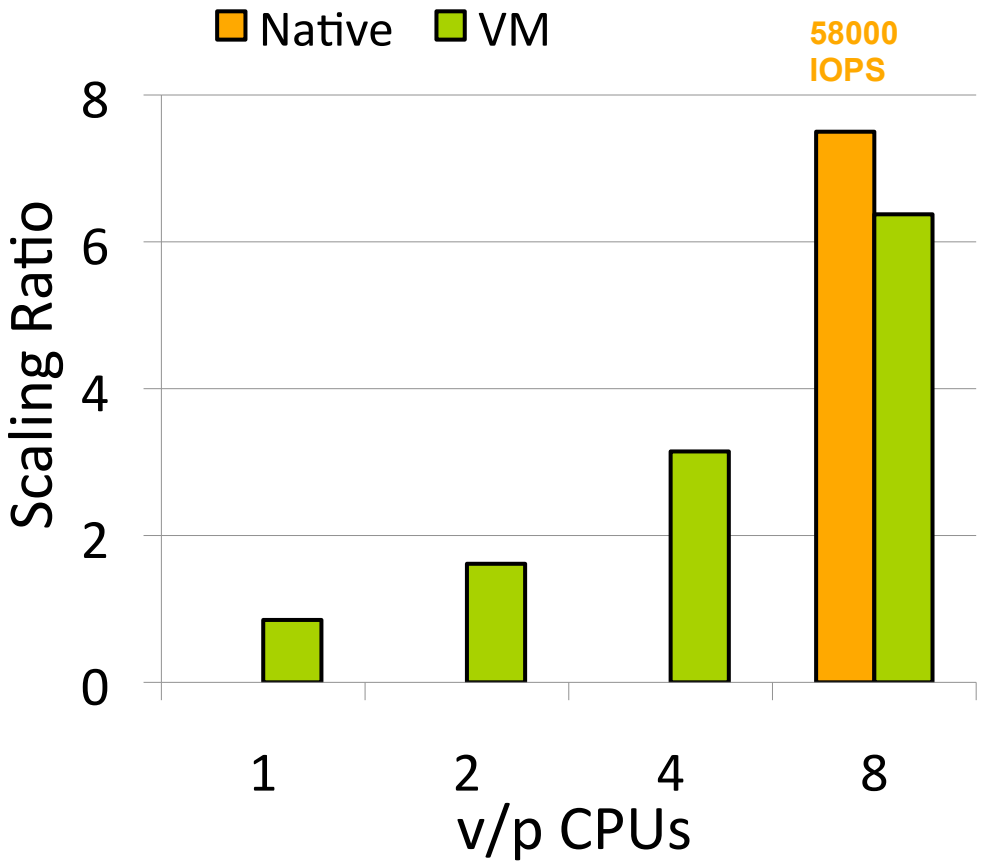
VMware Infrastructure Sets World Record for Web Server Performance

Virtualization Platform Beats Native Performance in SPECweb®2005 Benchmark

PALO ALTO, Calif., February 17, 2009 — VMware, Inc. (NYSE: VMW) the global leader in virtualization solutions from

IO In Action: Oracle/TPC-C*

- ESX achieves 85% of native performance with an industry standard OLTP workload on an 8-vCPU VM
- 1.9x increase in throughput with each doubling of vCPUs



vmware

Eight vCPU Oracle System Characteristics

Metric	8 vcpu VM
Business transactions per minute	250,000
Disk IOPS	60,000
Disk Bandwidth	258 MB/s
Network Packets/sec	27,000
Network Throughput	77 Mb/s

* Our benchmark was a fair-use implementation of the TPC-C business model;
our results are not TPC-C compliant results, and not comparable to official TPC-C results

Oracle/TPC-C* Experimental Details

- Host was an 8 CPU system with an Xeon 5500
- OLTP Benchmark: fair-use implementation of TPC-C workload
- Software stack includes: RHEL5.1, Oracle 11g R1, internal build of ESX (ESX 4.0 RC)
- Were there many Tweaks in getting this result? Not really…
  - ESX development build with these features
    - Async I/O, pvscsi driver, virtual Interrupt coalescing, topology-aware scheduling
    - EPT: H/W MMU enabled processor
  - The only ESX "tunable" applied: static vmxnet TX coalescing
    - 3% improvement in performance

Hardware Selection

Platform: Choose Newer Hardware

If Possible Choose Latest Hardware

- Older processors with longer pipelines and smaller caches can be particularly challenging for virtualized workloads

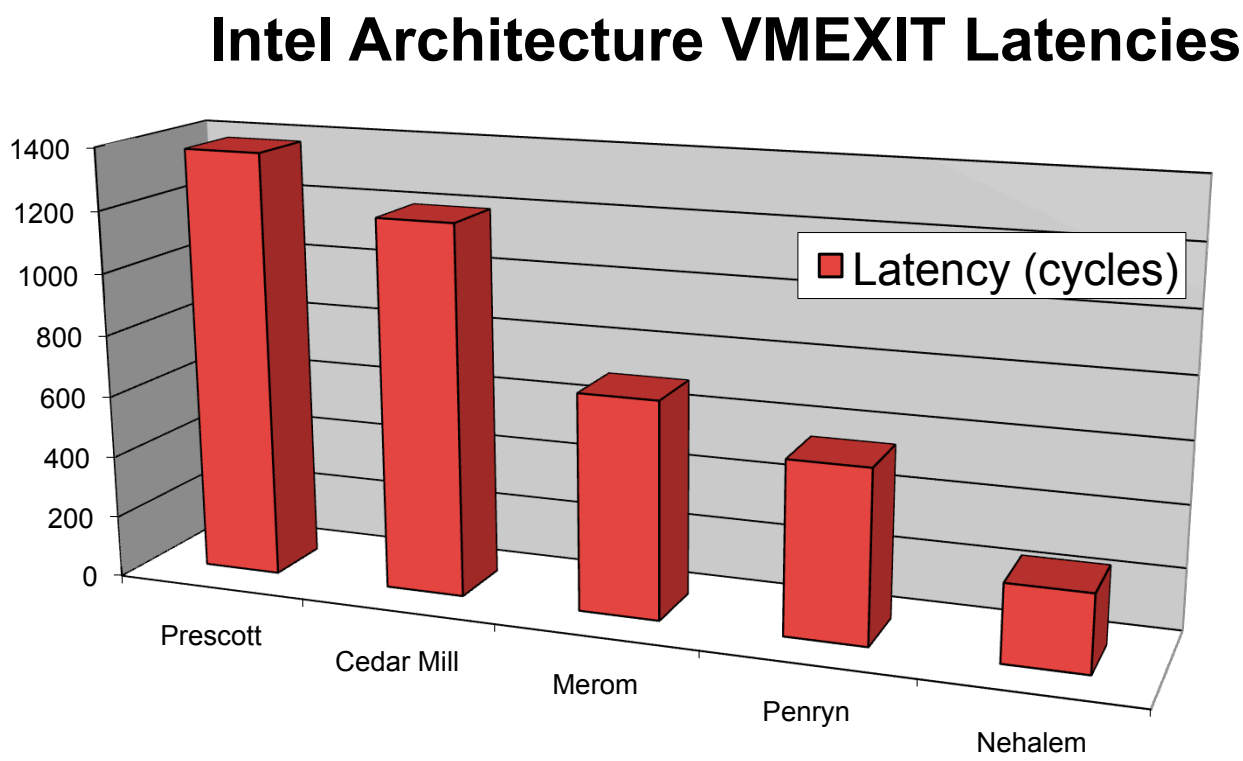
Newer processors have hardware virtualization support for

- Privileged instructions
- Virtual machine memory management

Most applications perform better with Hardware-assisted monitors (Intel VT, AMD RVI)

- Enable hardware virtualization in BIOS.

Intel Architecture Virtualization Performance



HW virtualization support improving from CPU generation to generation

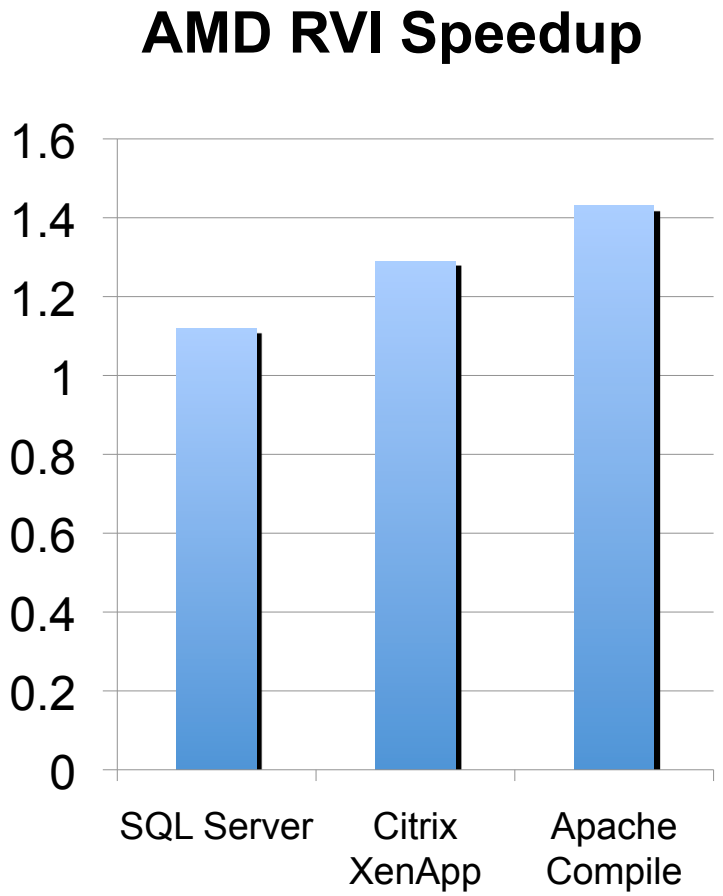
Memory Virtualization in Hardware

Hardware memory management units (MMU) improve efficiency

- AMD RVI currently available
- Dramatic gains can be seen

But some workloads see little or no value

- And a small few actually slow down



Optimal Virtual Machine Setup

General Best Practices: VM Setup

During VM creation select right guest OS type

- Determines the monitor type and related optimizations
- Determines default optimal devices and their settings
- Do not choose ‘other’

> Install 64-bit OS if large amounts of memory are needed

> Choose a OS version with fewer timer interrupts

- Windows, Linux 2.4 100/sec per vCPU
- Some Linux 2.6 250/sec per vCPU
- Some Linux 2.6 1000/sec per vCPU

Disable unused devices that use a polling scheme

> USB, CDROM

> Consume CPU when idle

Large Pages

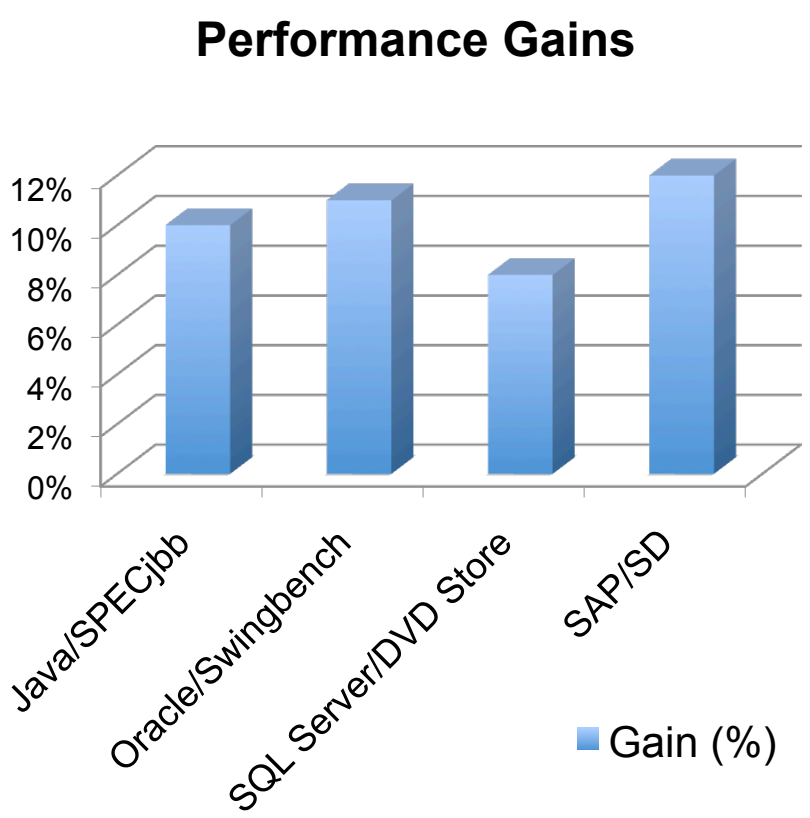
Increases TLB memory coverage

- Removes TLB misses, improves efficiency

Improves performance of applications that are sensitive to TLB miss costs

Configure OS and applicatic to leverage large pages

- LP will not be enabled by default



VM Configuration: HW or SW Memory Management?

Sensitive to TLB miss costs?	App is memory management intensive? No	App is memory management intensive? Yes
No	Example: number crunching financial software *SW and HW virtualizations perform equally well*	Example: Citrix, Apache web server *HW virtualization performs better*
Yes	Example: Java applications *With large pages, HW, with small pages, SW*	Example: databases *Depends on which cost is higher: memory virt overhead or TLB cost?* *Benchmark!*

Platform Optimization: Network

Use a network adapter that supports:

- Checksum offload, TCP segmentation offload (TSO), Jumbo frames (JF)
- Enable JF when hardware is available (default is off!)
- Capability to handle high memory DMA (64-bit DMA addresses)
- Capability to handle multiple scatter/gather elements per Tx frame

Check configuration

- Ensure host NICs are running with highest supported speed and full-duplex
- NIC teaming distributes networking load across multiple NICs
  - Better throughput and allows passive failover

Use separate NICs to avoid traffic contention

- For Console OS (host management traffic), VMKernel (vmotion, iSCSI, NFS traffic), and VMs

Jumbo Frames

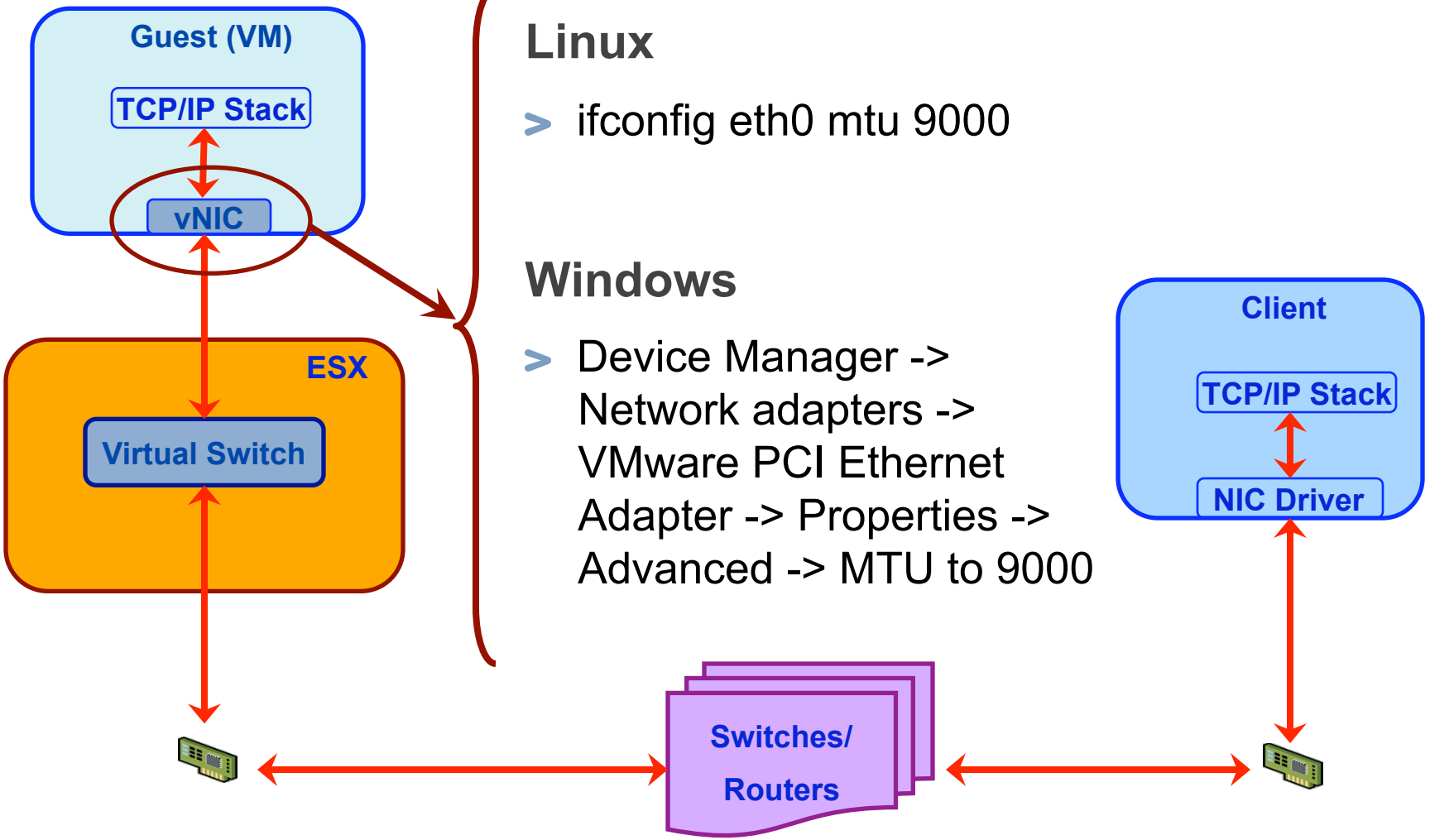
Before transmitting, IP layer fragments data into MTU (Maximum Transmission Unit) sized packets

- Ethernet MTU is 1500 bytes
- Receive side reassembles the data

Jumbo Frames

- Ethernet frame with bigger MTU
- Typical MTU is 9000 bytes
- Reduces number of packets transmitted
- Reduces the CPU utilization on transmit and receive side

Jumbo Frames



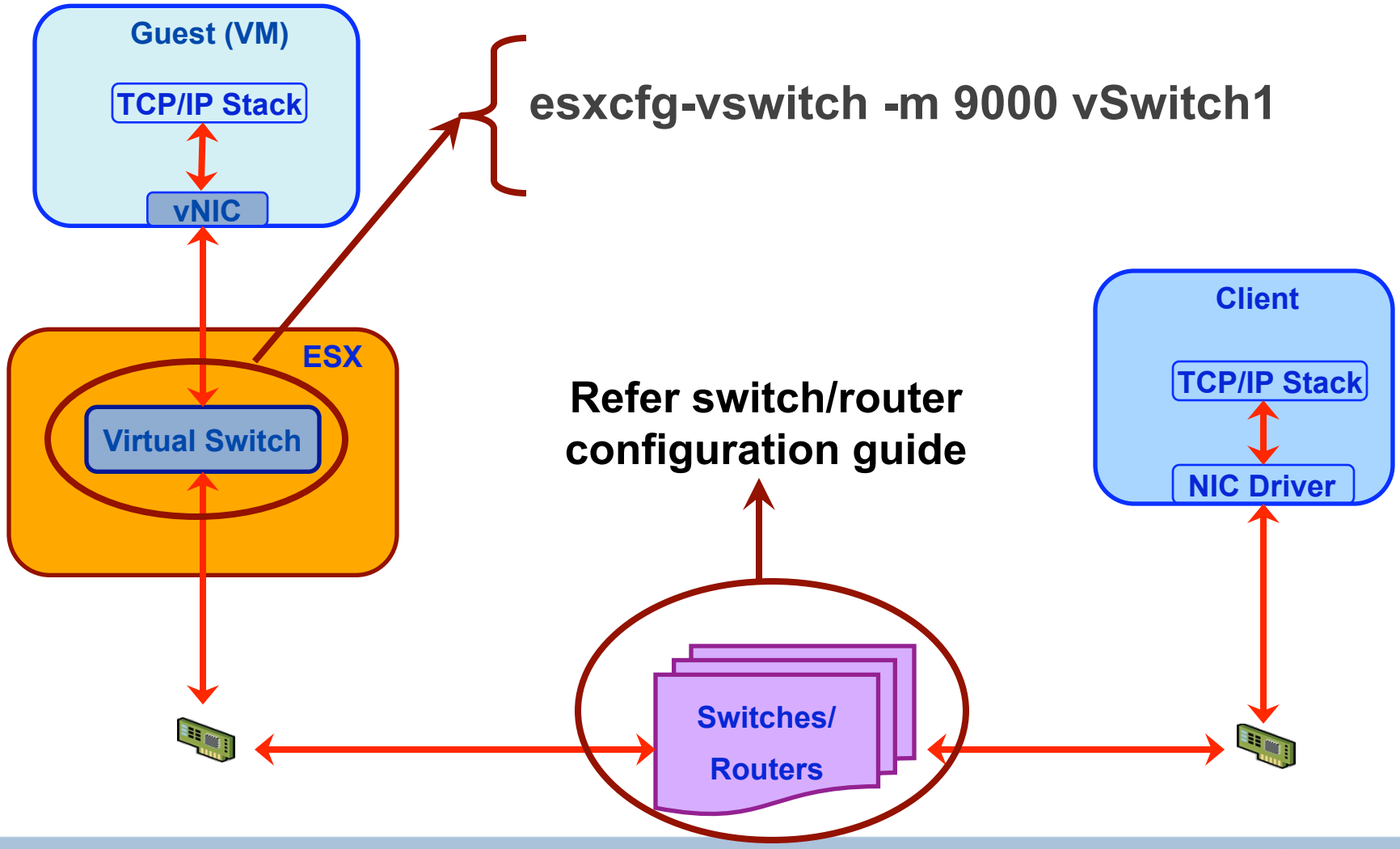
Linux

- ifconfig eth0 mtu 9000

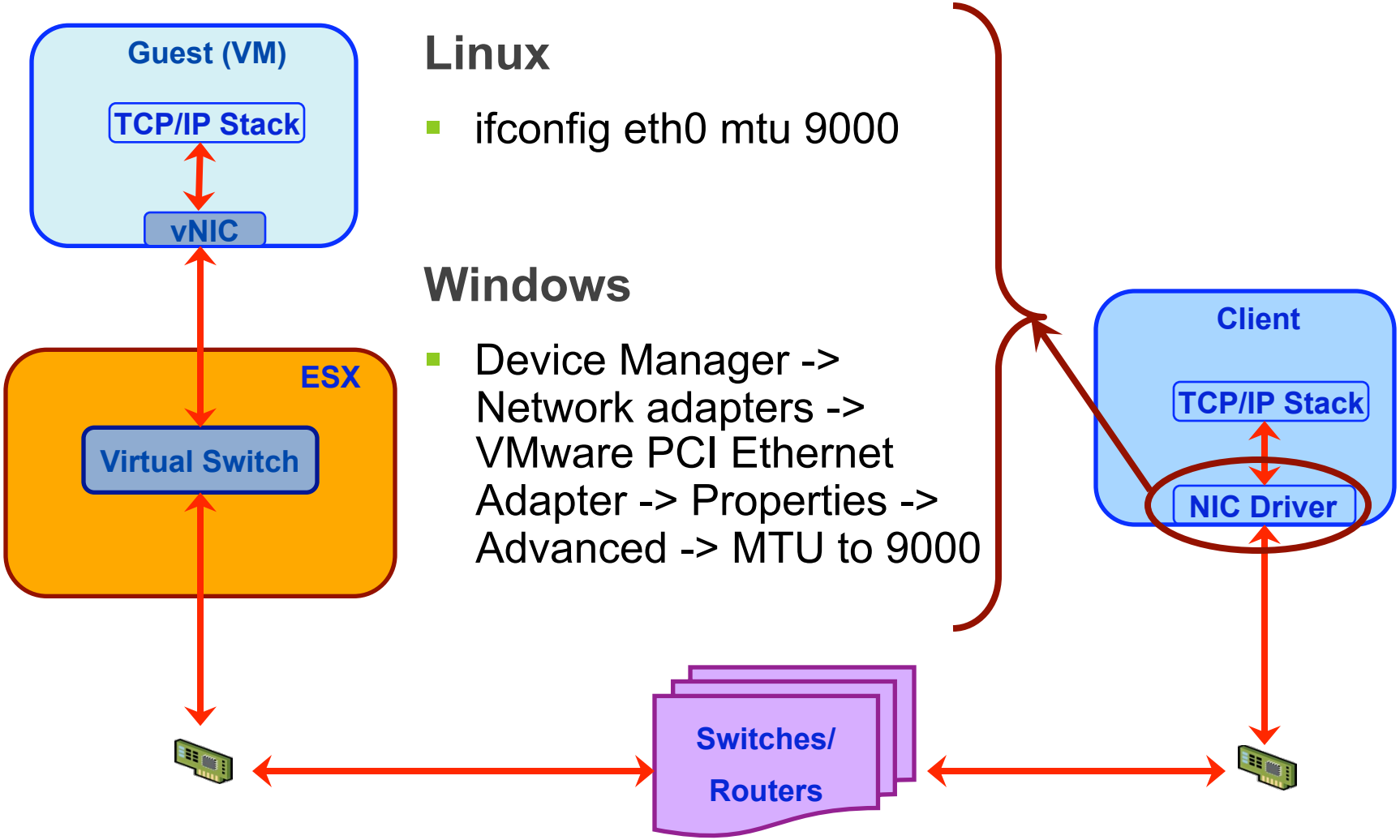
Windows

- Device Manager -> Network adapters -> VMware PCI Ethernet Adapter -> Properties -> Advanced -> MTU to 9000

Jumbo Frames



Jumbo Frames



Linux

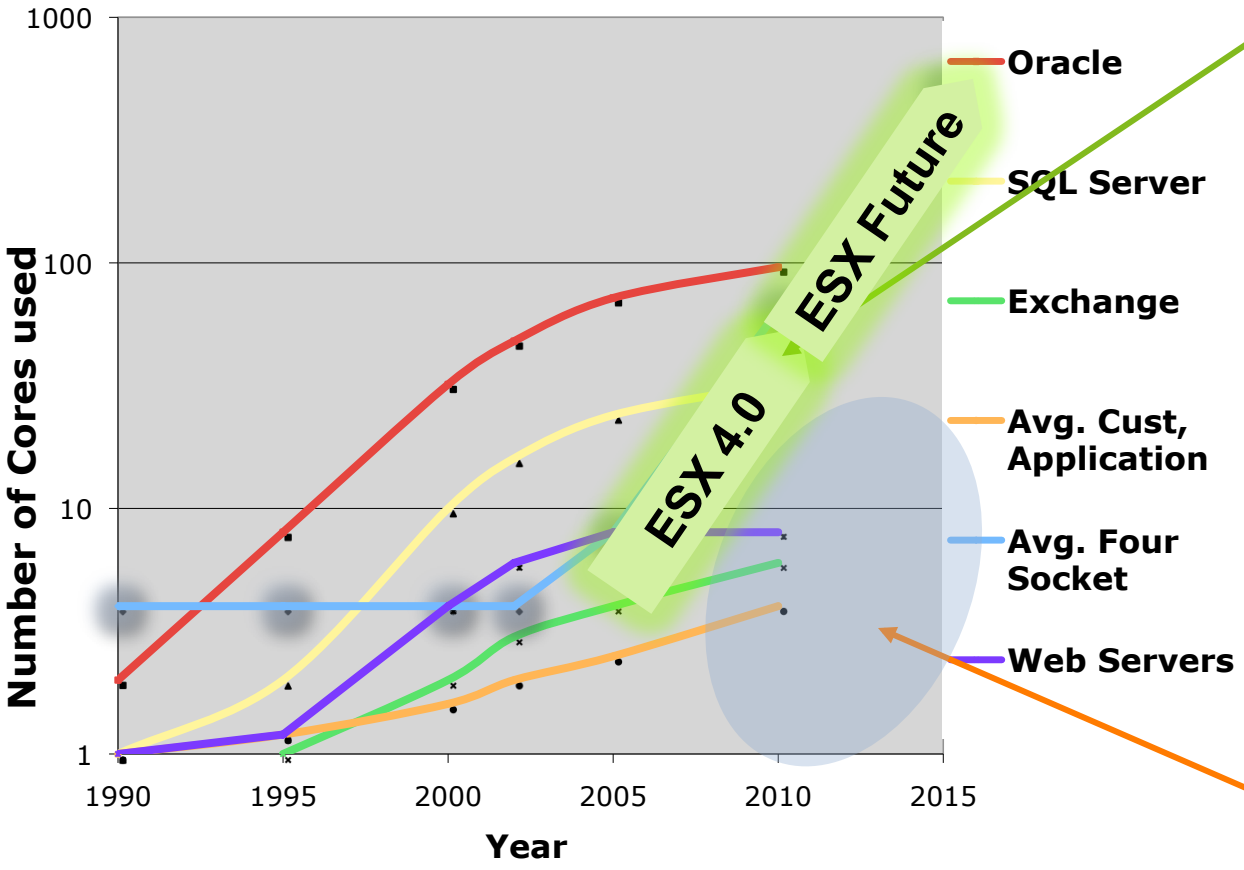
- ifconfig eth0 mtu 9000

Windows

- Device Manager -> Network adapters -> VMware PCI Ethernet Adapter -> Properties -> Advanced -> MTU to 9000

SMP and the Scheduler

VMware vSphere enables you to use all those cores…



VMware ESX Scaling: Keeping up with core counts

Virtualization provides a means to exploit the hardware's increasing parallelism

Most applications don't scale beyond 4/8 way

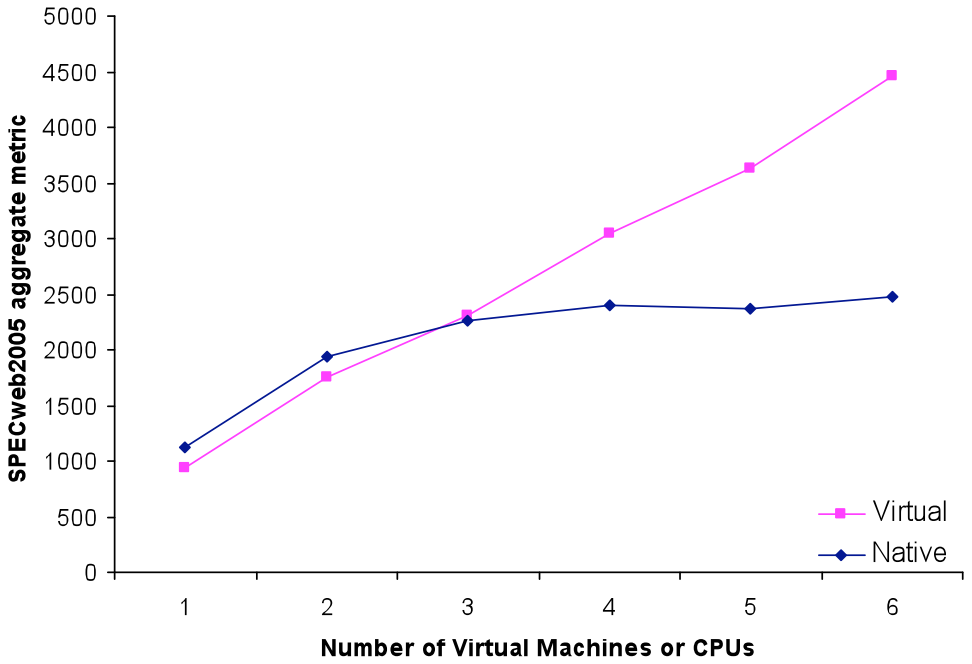
vmware

Virtualization-aware Architecture: Building Blocks

Many applications lack scalability beyond certain CPUs

- Apache web server, WebSphere, Exchange
- Configure vCPUs to application scalability limits
- For additional capacity instantiate more of such VMs

SPECweb2005 Native and Virtual Scaling



http://www.vmware.com/files/pdf/consolidating_webapps_vi3_wp.pdf

Scheduler Opportunities

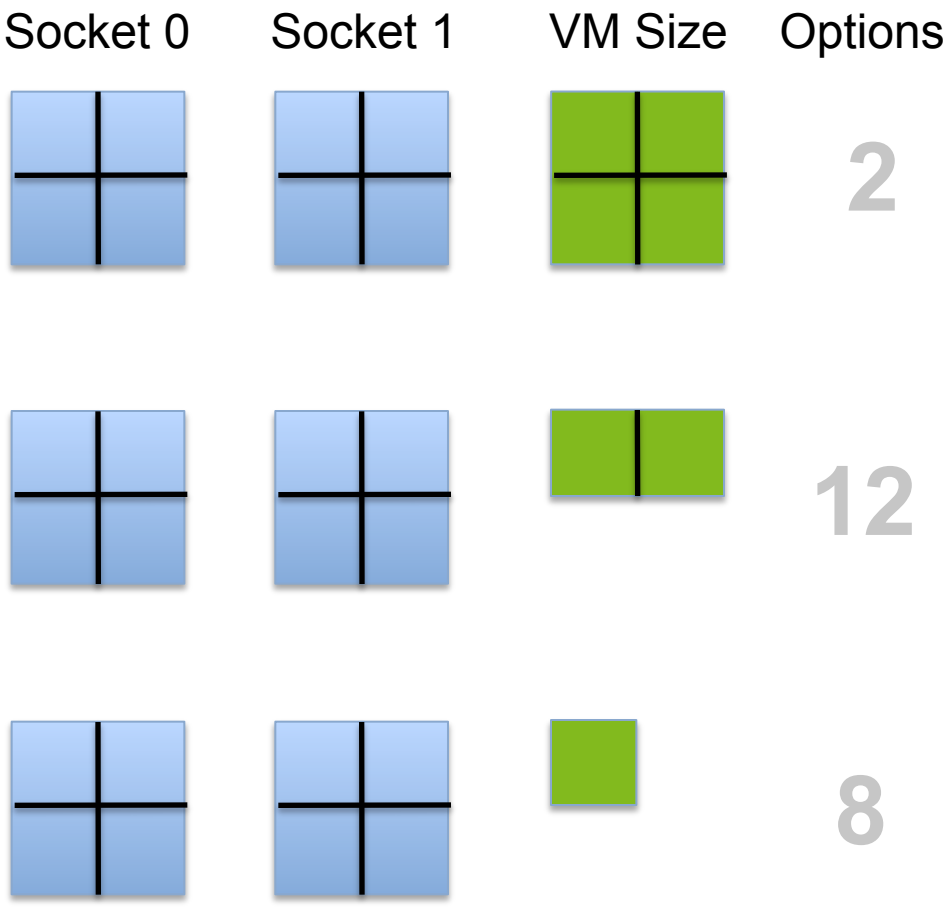
vCPUs from one VM stay on one socket*

With two quad-core sockets, there are only two positions for a 4-way VM

1- and 2-way VMs can be arranged many ways on quad core socket

Newer ESX schedulers more efficiency use fewer options

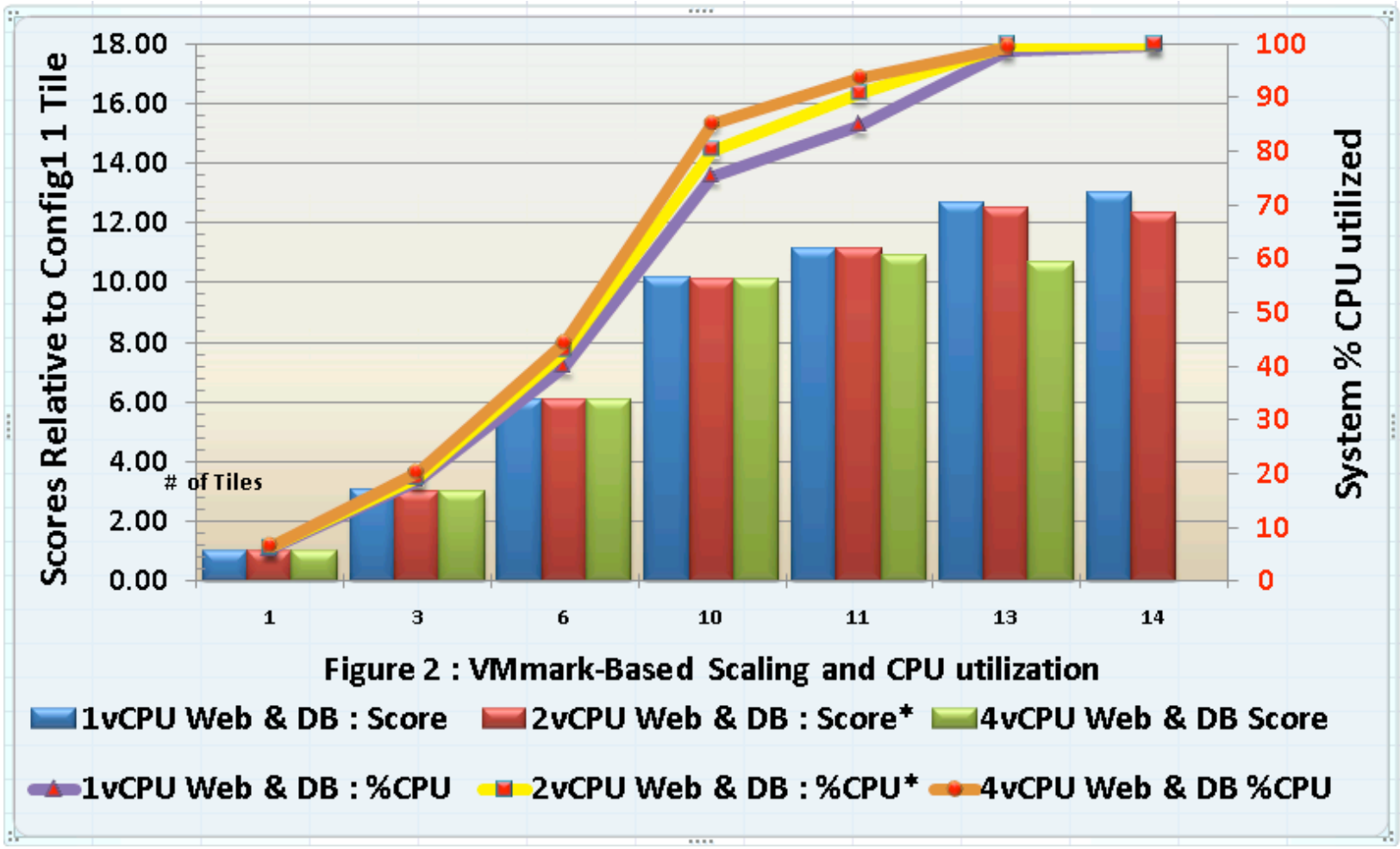
- Relaxed co-scheduling



(*) The cell limit has been removed in vSphere

vmware

The Performance Cost of SMP



From: http://blogs.vmware.com/performance/2009/06/measuring-the-cost-of-smp-with-mixed-workloads.html

Memory Management

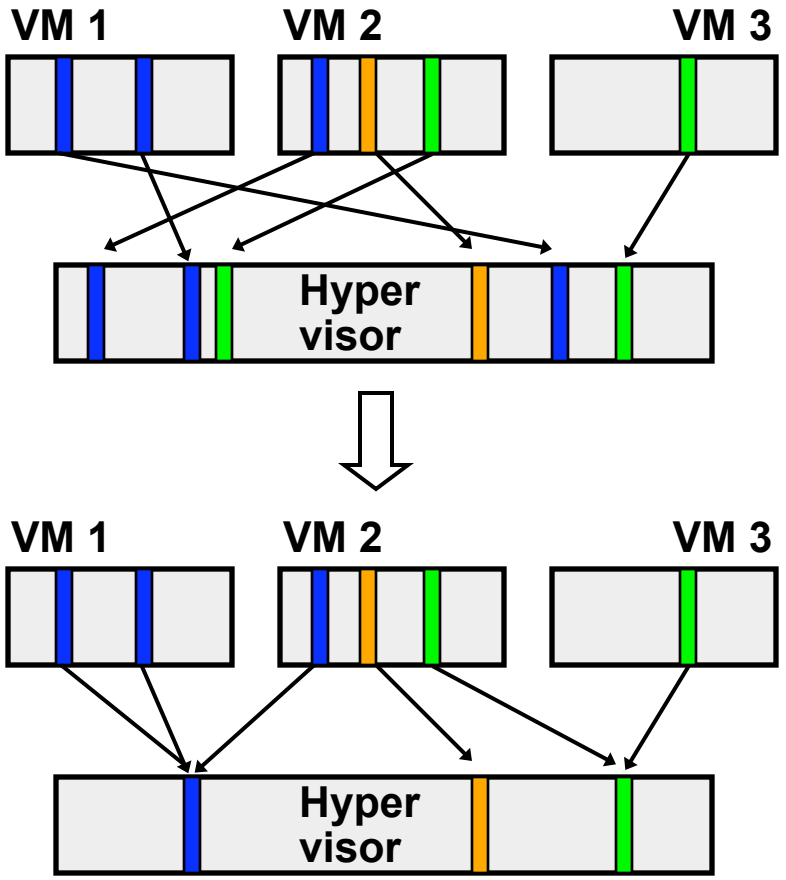
“Bonus” Memory During Consolidation: Sharing!

Content-based

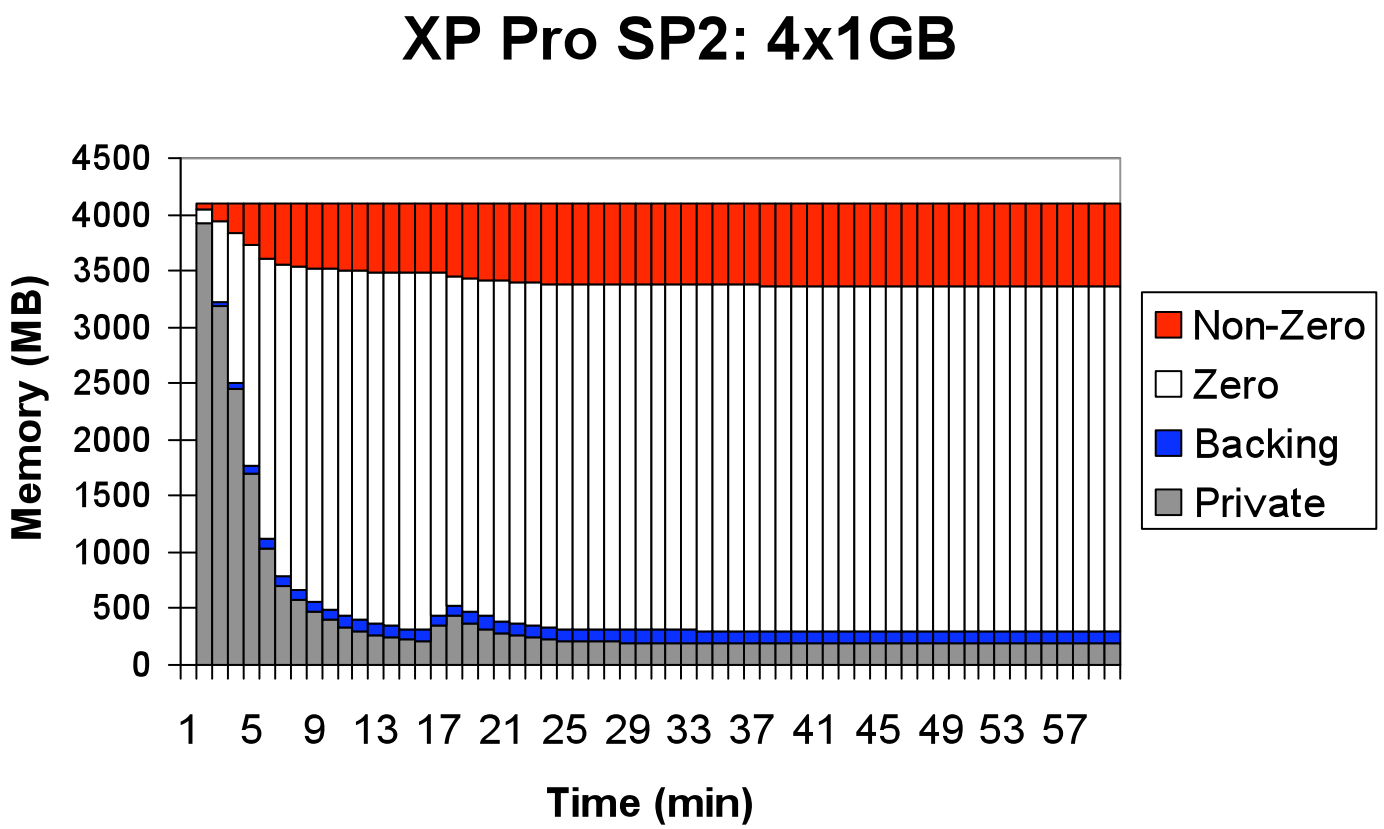
- Hint (hash of page content) generated for 4K pages
- Hint is used for a match
- If matched, perform bit by bit comparison

COW (Copy-on-Write)

- Shared pages are marked read-only
- Write to the page breaks sharing

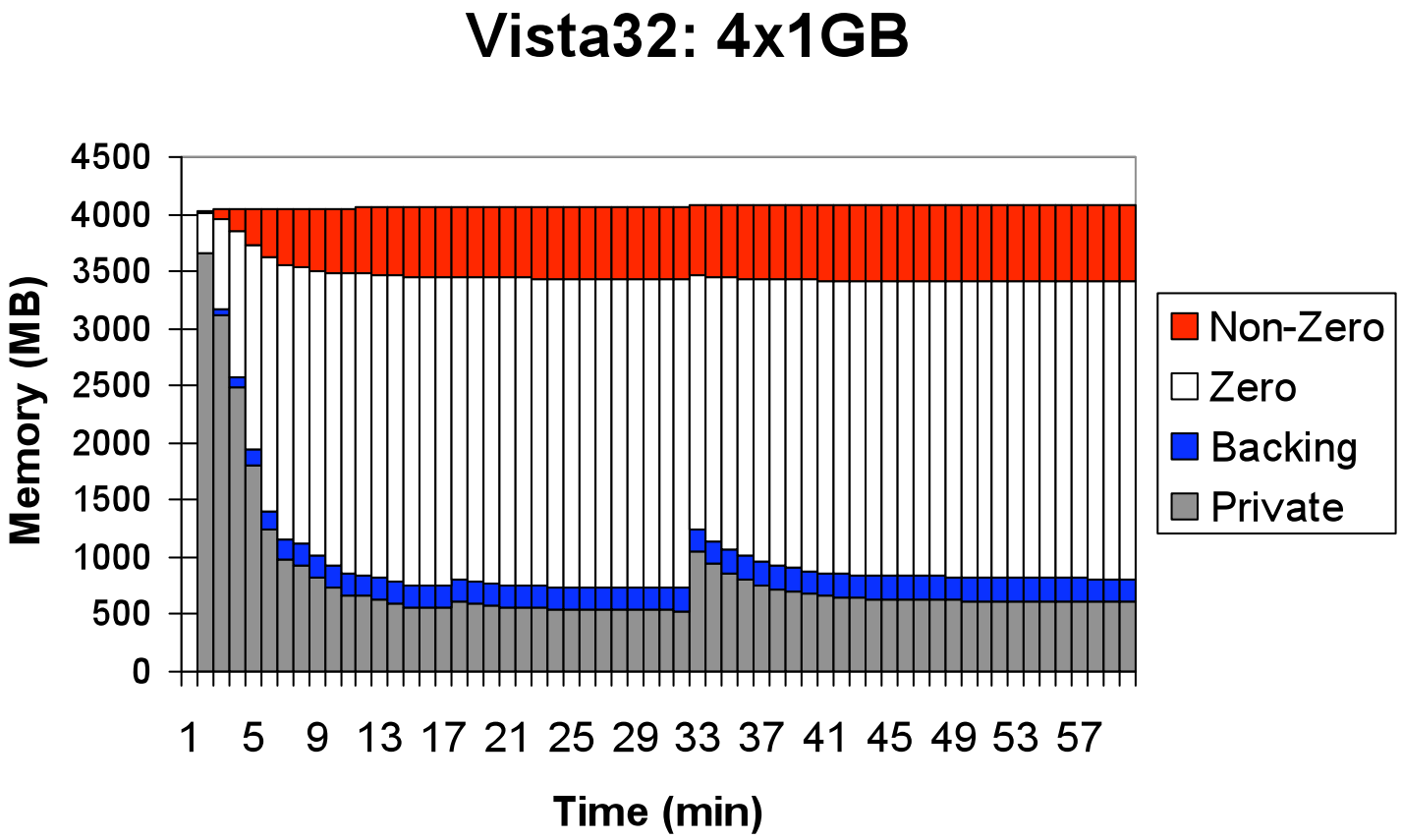


Page Sharing in XP



Memory footprint of four idle VMs quickly decreased to 300MB due to aggressive page sharing.

Page Sharing in Vista



Memory footprint of four idle VMs quickly decreased to 800MB. (Vista has larger memory footprint.)

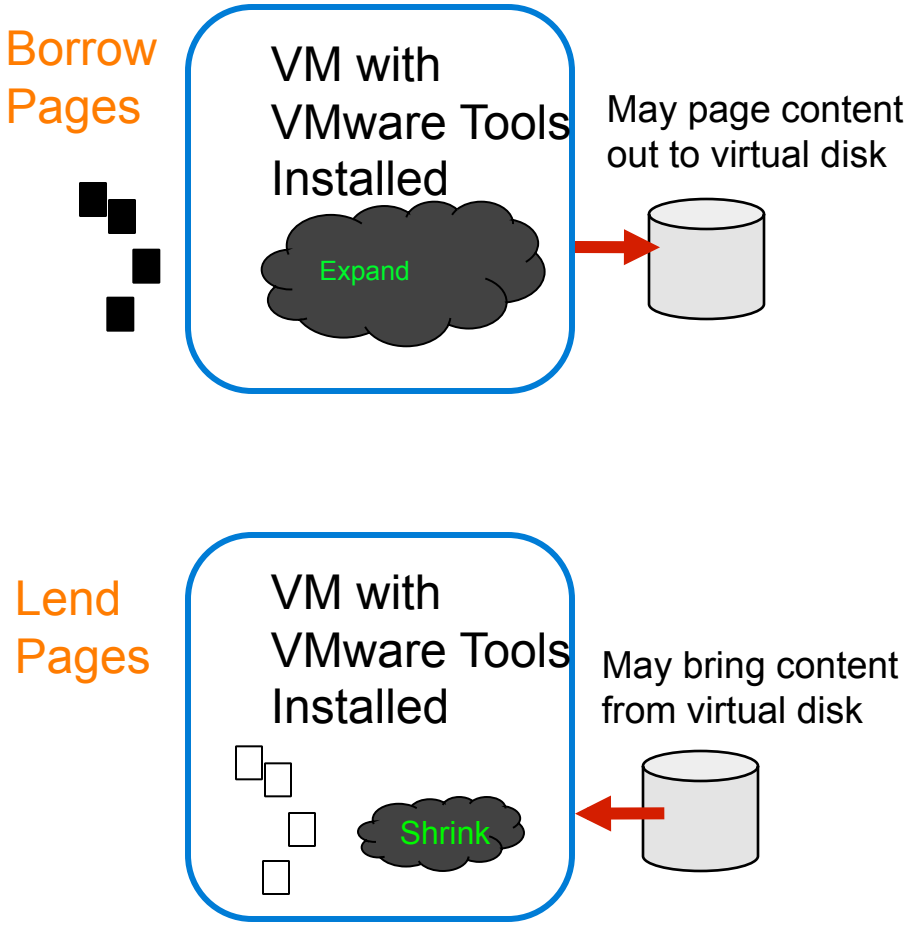
ESX Server Memory Ballooning

Guest OS has better information than VMkernel

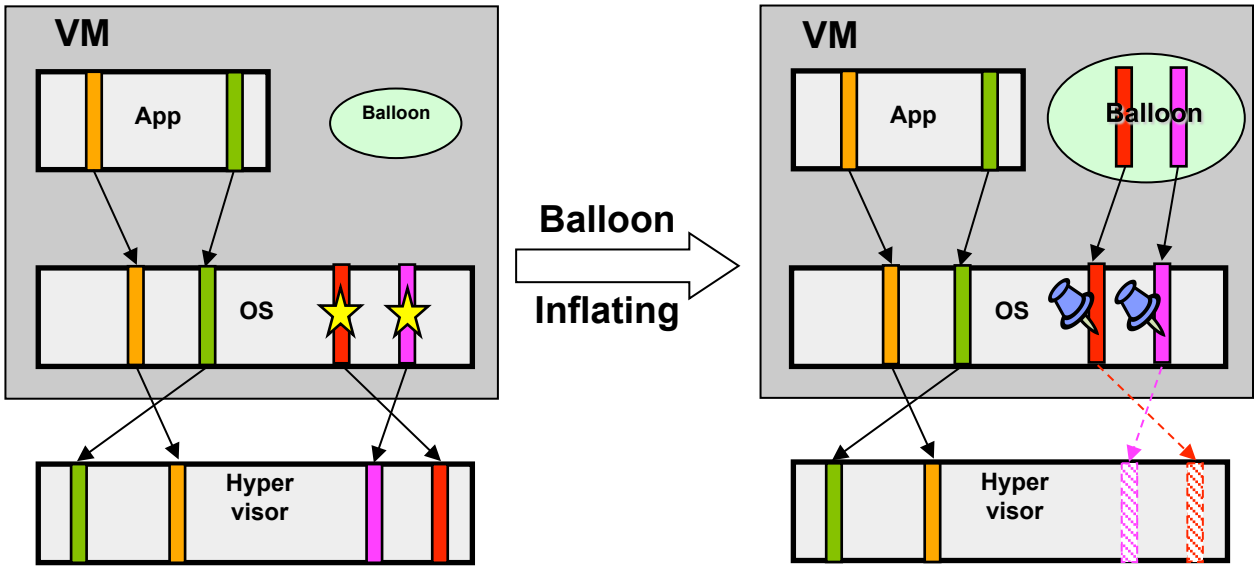
- Which pages are stale
- Which pages are unused

Guest Driver installed with VMware Tools

- Artificially induces memory pressure
- VMkernel decides how much memory to reclaim, but guest OS gets to choose particular pages



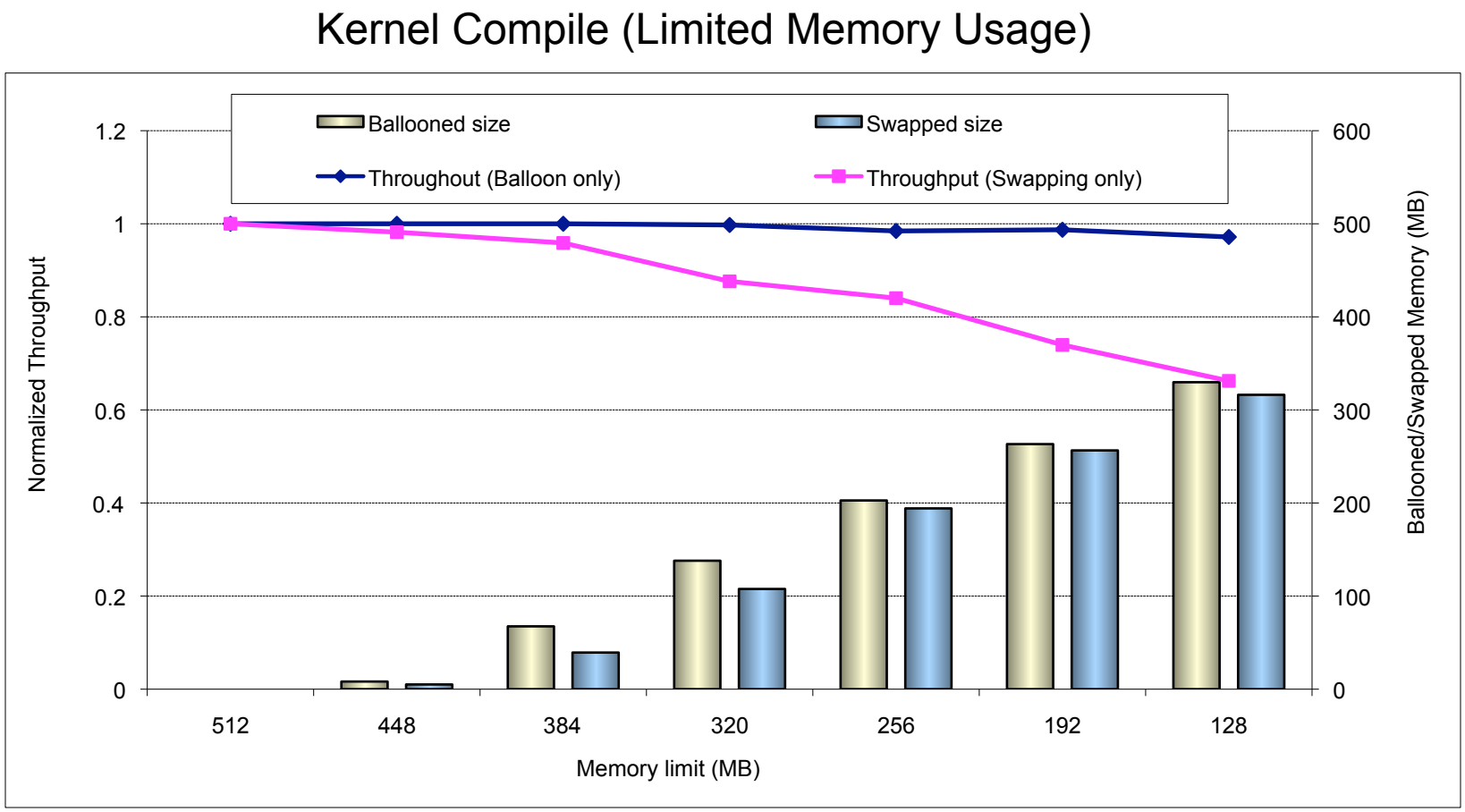
Ballooning Pins Pages



Memory has been reduced and pinned to induce guest to page, if needed

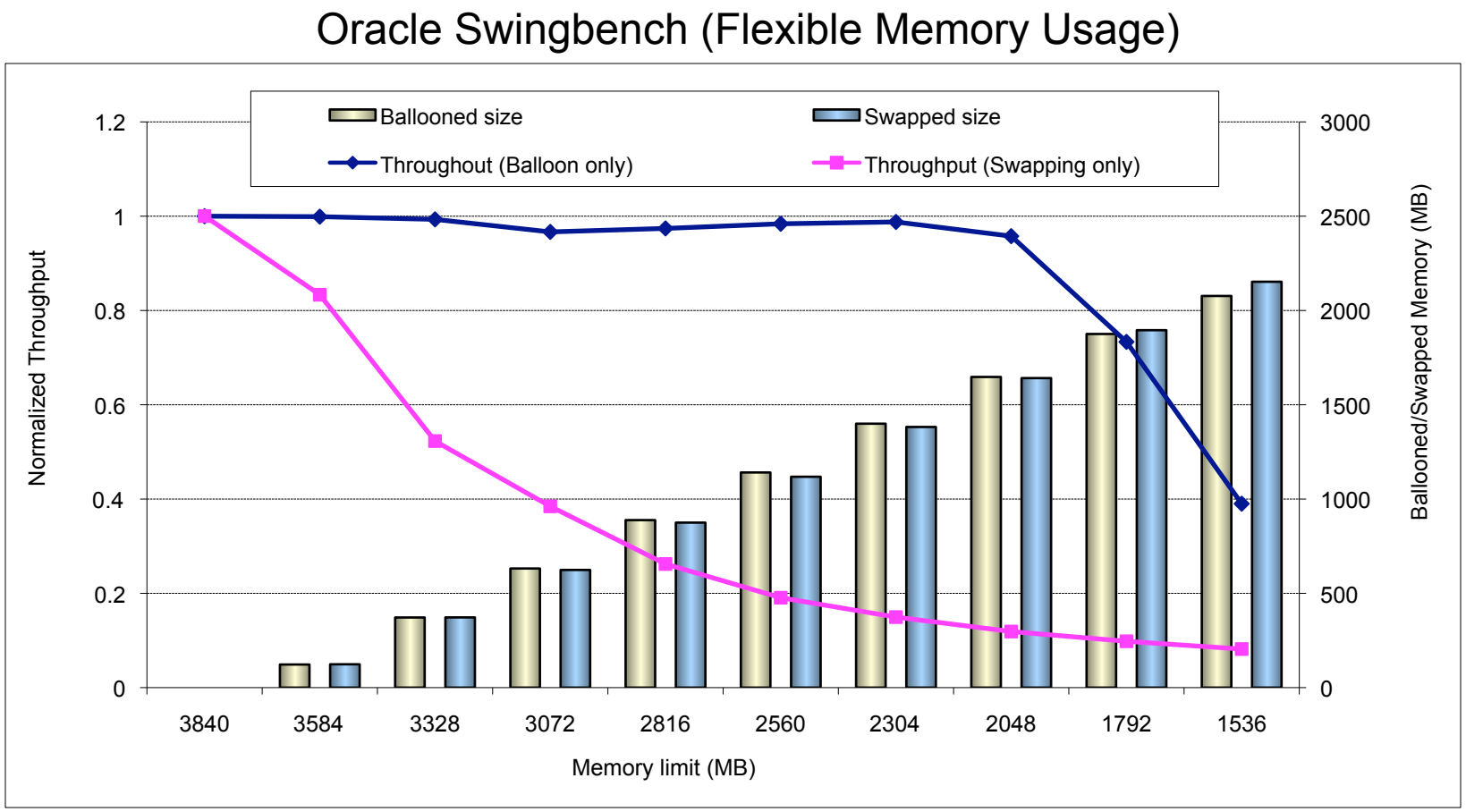
If memory is short, ESX must choose which pages to swap to disk

Ballooning Can Induce Non-harmful Guest Paging



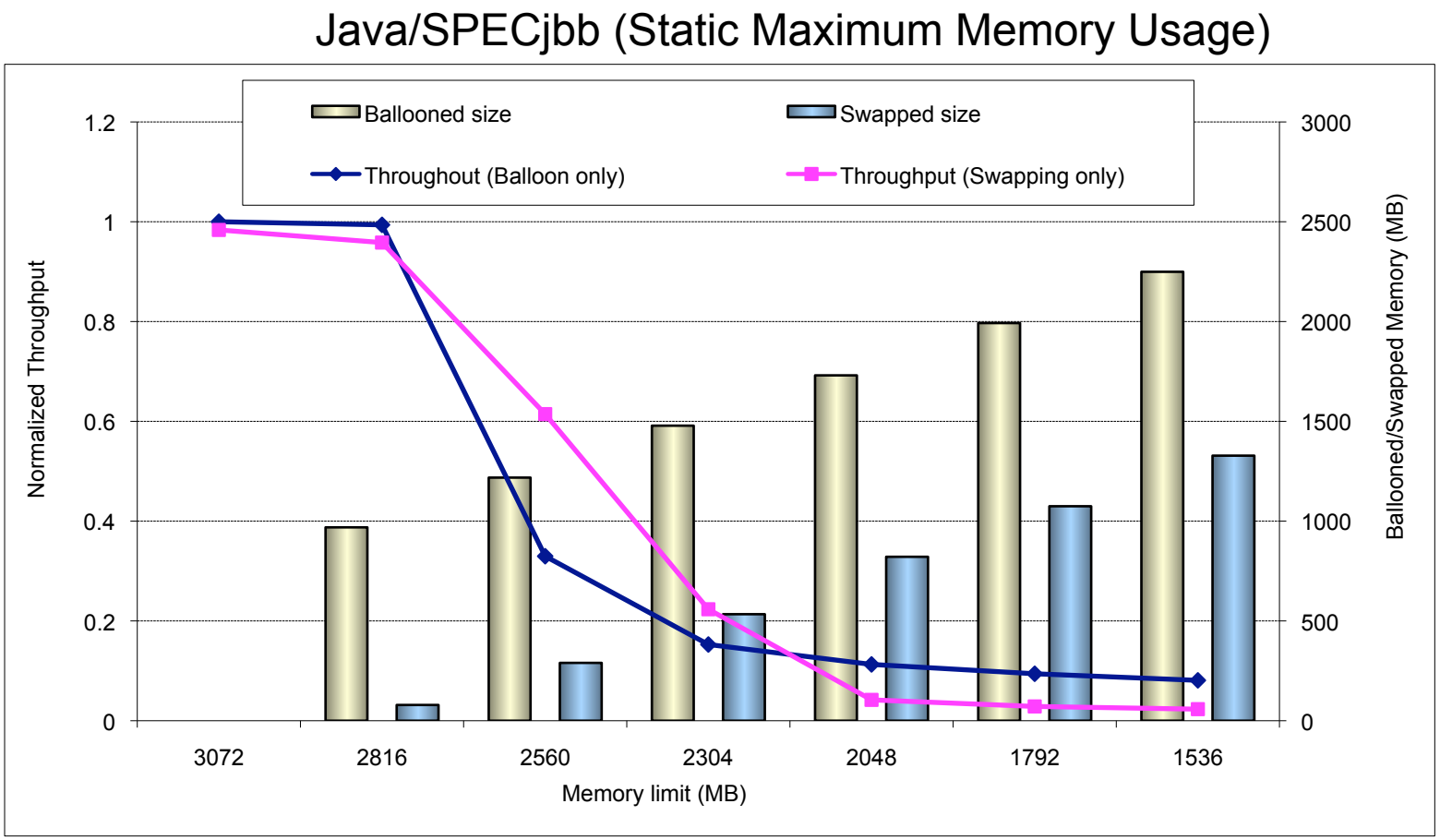
vmware

Ballooning Can Be More Effective Than Swapping



vmware

Java Requires Careful Memory Management



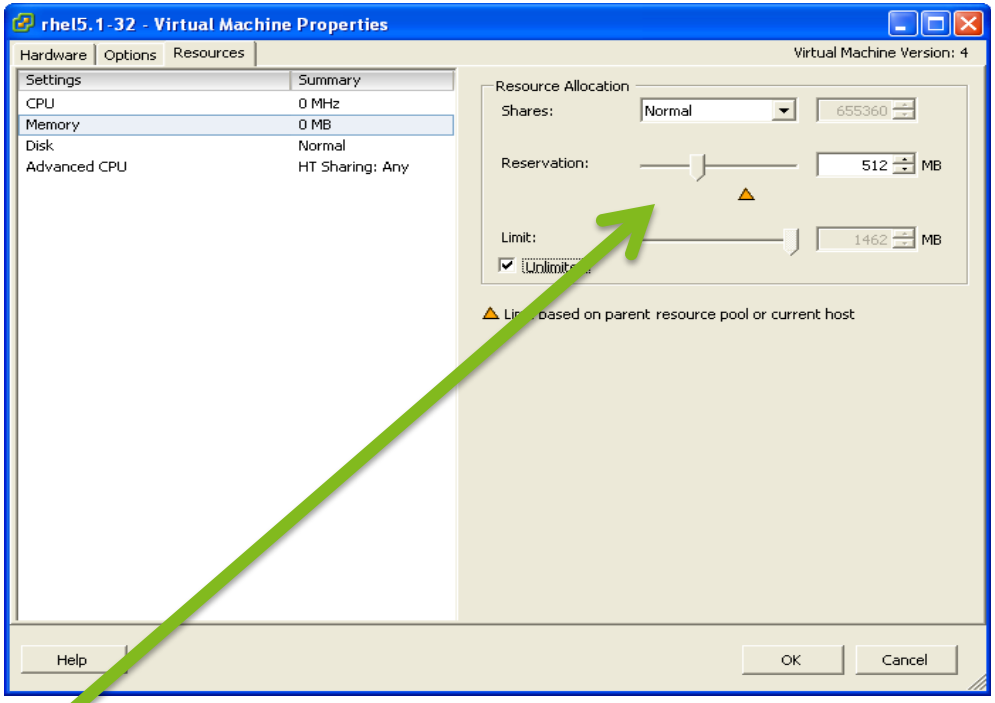
vmware

Managing Memory in Java Environments

Calculate OS memory

Estimate JVM needs

Specify heap exactly



Reservations = OS + JVM + heap

vmware

Getting Memory Sizing Just Right

Monitor guest paging using traditional tools

- Consider putting guest swap file on its own VMDK
- Put all guest swap VMDKs on the same LUN
- vSphere client can then monitor guest paging by watching that LUN's traffic

Use vSphere Client to track host memory usage

- There is no way to predict this before hand
- Run workloads and analyze performance

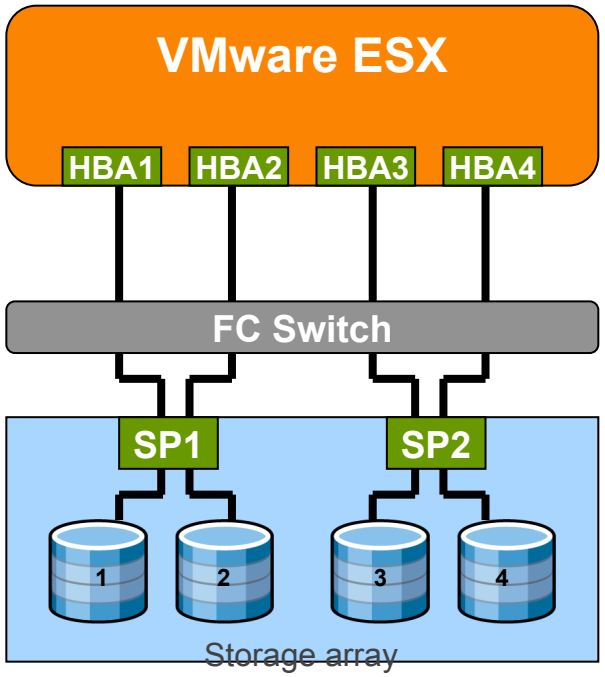
Statistic	VirtualCenter	esxtop
Active Memory (recently used by guest OS)	Active Memory	%ACTV, %ACTVS, %ACTVS
Swap rate (VC on VI3 reports swap magnitude)	VI3: Swap In/Out vSphere: Swap In/Out Rate	SWW/s, SWR/s

Understanding and Correcting Storage Performance

Platform Optimization: Storage

Over 90% of storage related performance problems stem from misconfigured storage hardware

- Consult SAN Configuration Guides
- Ensure disks are correctly distributed
- Ensure caching is enabled
- Consider tuning layout of LUNs across RAID sets
- Spread I/O requests across available paths



Platform Optimization: File System

Always use VMFS

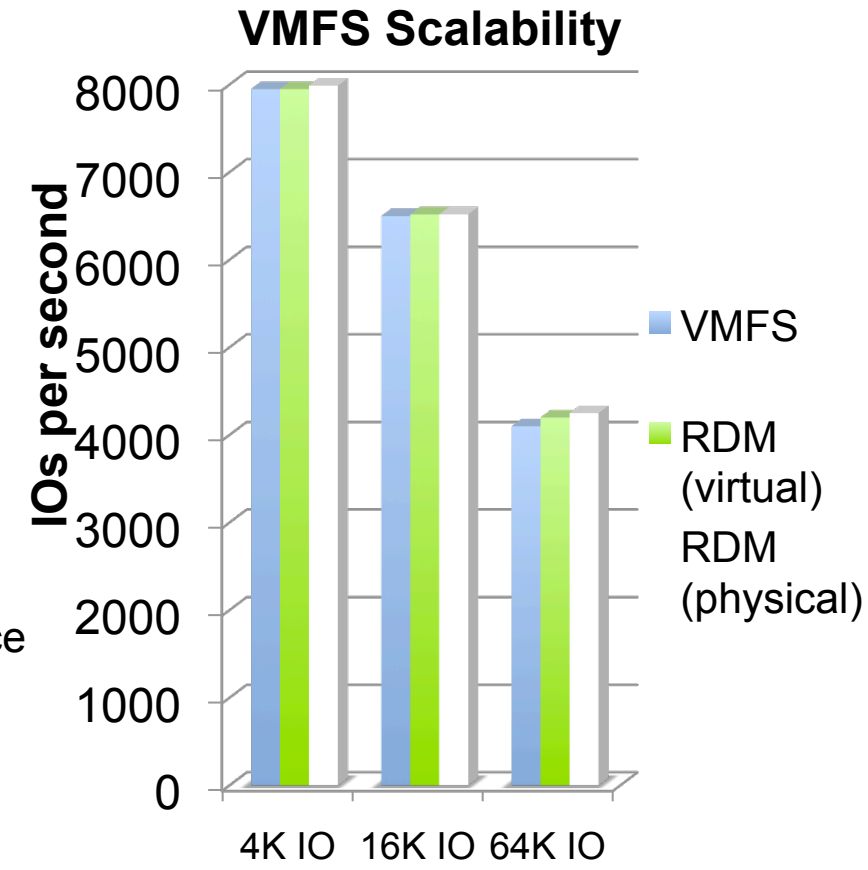
- Negligible performance cost and superior functionality

Align VMFS on 64K boundaries

- Automatic with vCenter
- www.vmware.com/pdf/esx3_partition_align.pdf

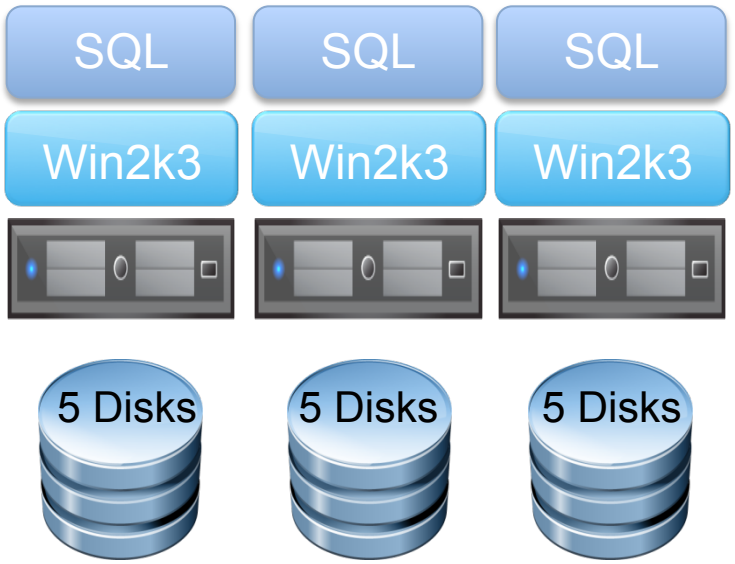
VMFS is a distributed file system

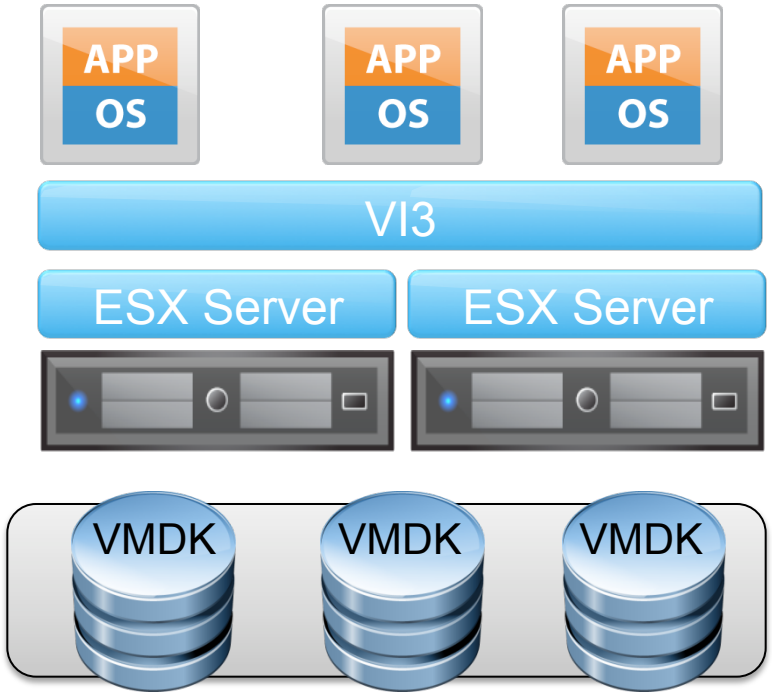
- Be aware of the overhead of excessive metadata updates
  - If possible schedule maintenance for off-peak hours



Server Consolidation: Storage Planning

Physical setup: each instance provided 5-spindle LUN

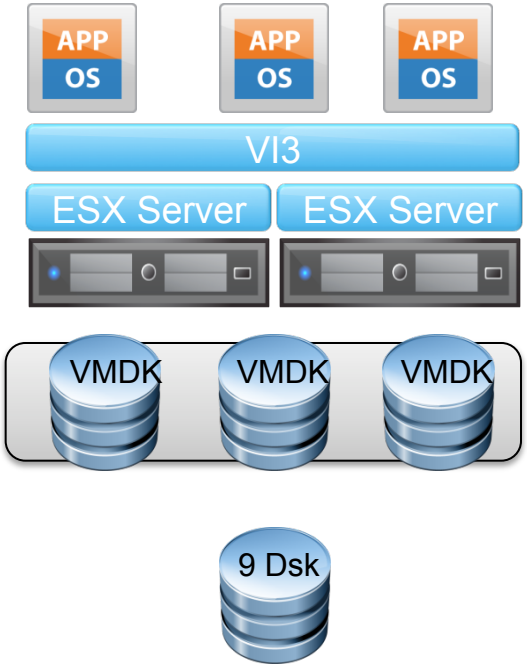




Virtual architecture: Each VM provided its own VMDK

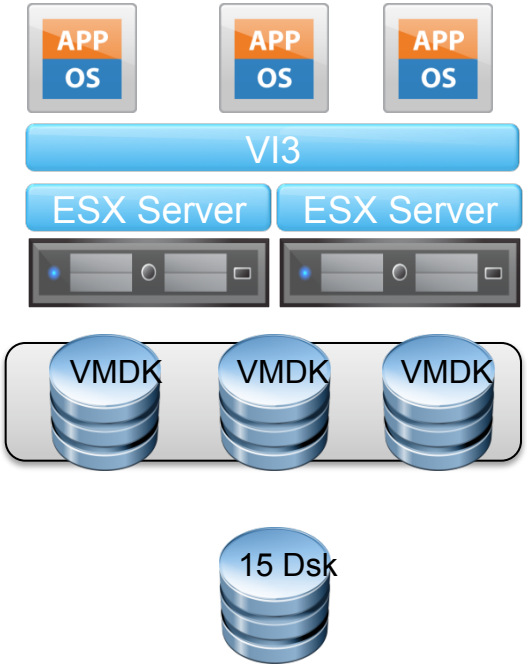
- But now do they map to disks?

Server Consolidation: Storage Planning



Nine spindles for VMFS volume

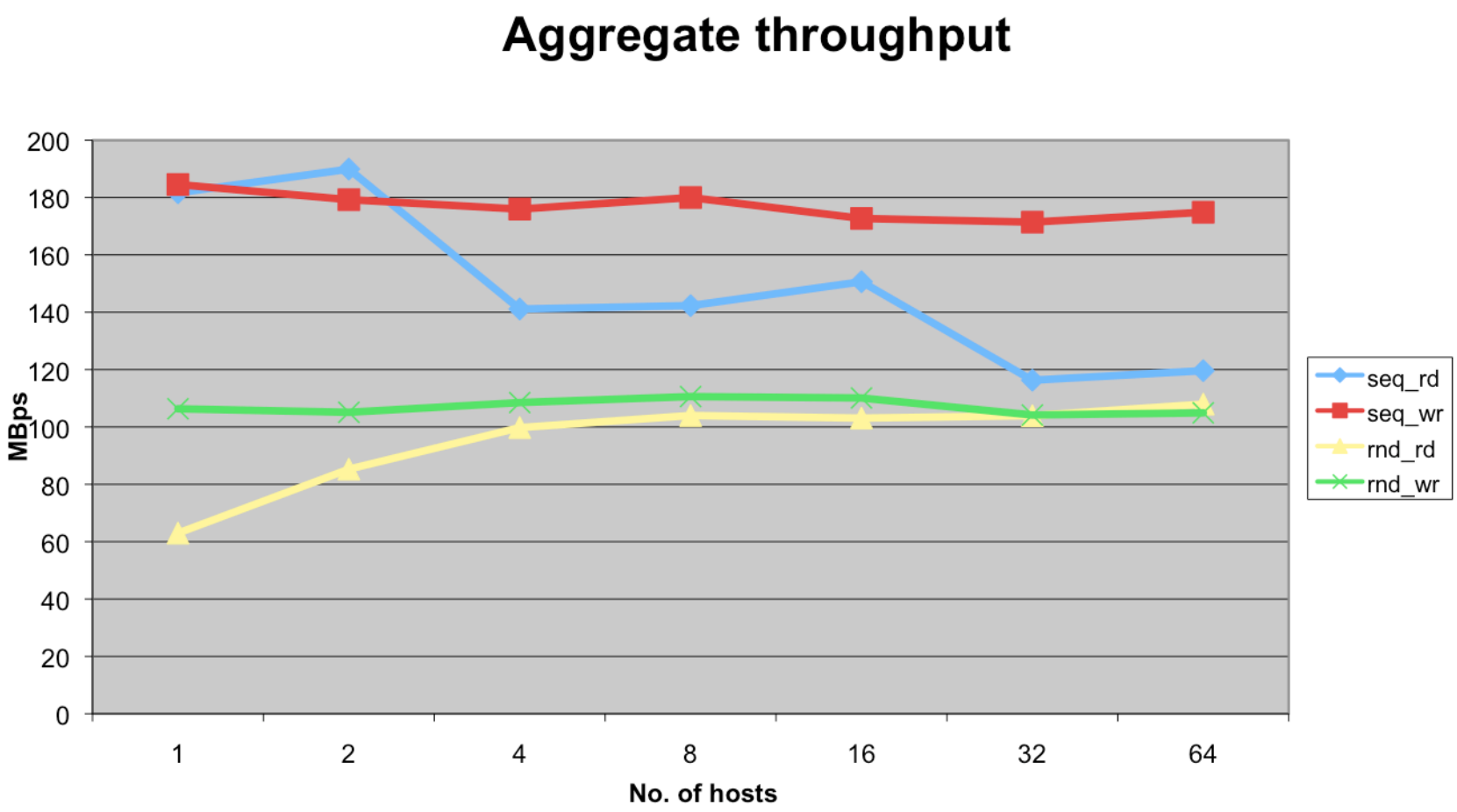
This is clearly less than the 15 disks in the physical deployment



15 spindles for virtual deployment matches physical

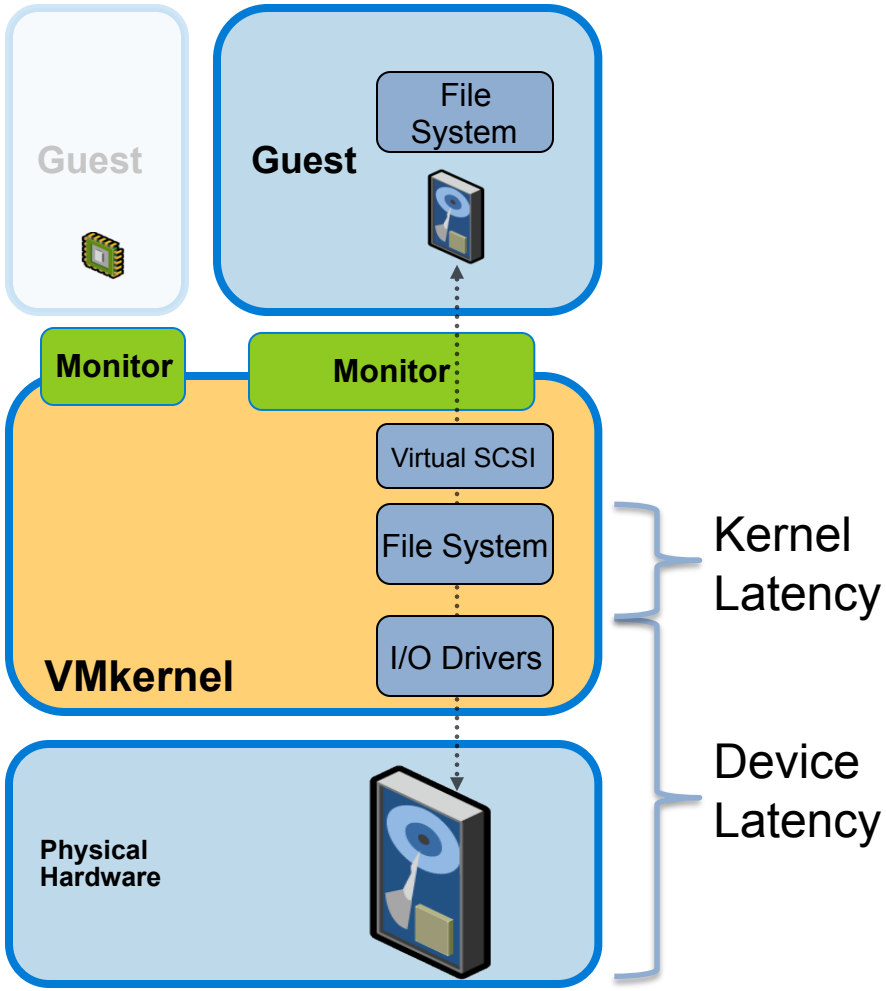
But this configuration inferior to multiple LUNs and access pattern changes (see following)

Sequential Workloads Generate Random Access
As observed in VMFS scalability tests



vmware

Storage Analysis and vscsiStats



vCenter reports latencies for FC and iSCSI only

- **Device latency for hardware**
- **Kernel latency for queuing**

VI3 and vSphere have instrumented the virtual SCSI bus for stats on all VMs

- **vscsiStats**

Workload Characterization Using vscsiStats

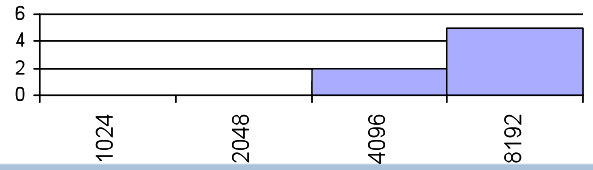
vscsiStats characterizes IO for each virtual disk

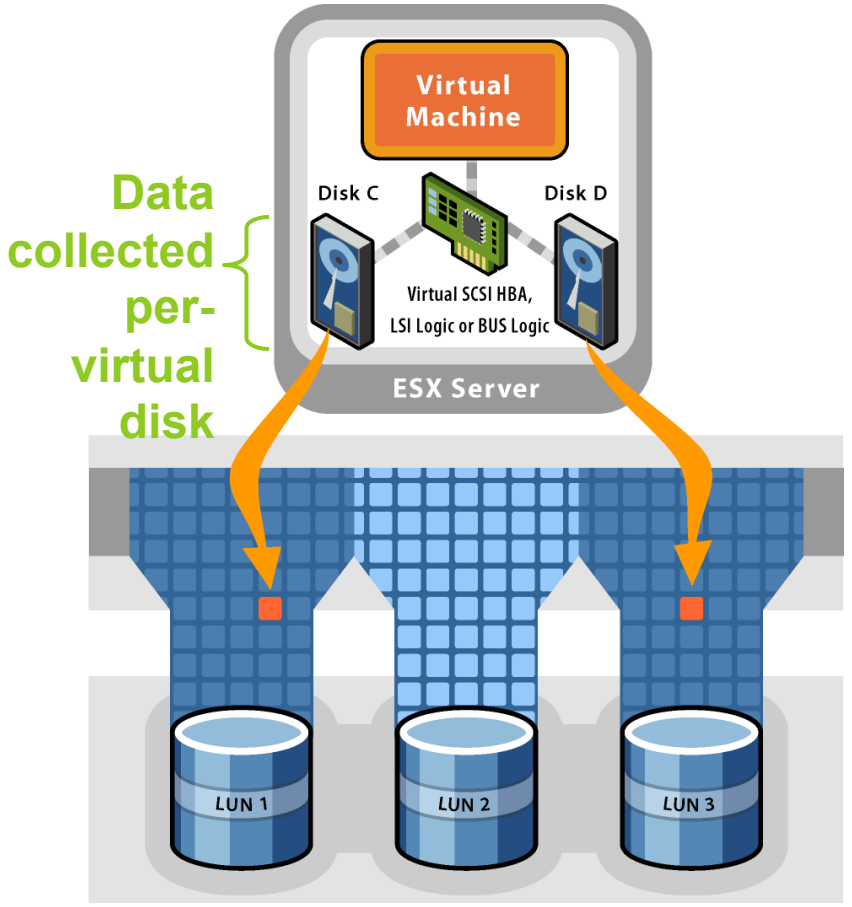
- Allows us to separate out each different type of workload into its own container and observe trends

Histograms only collected if enabled; no overhead otherwise

Technique:

- For each virtual machine I/O request in ESX, we insert some values into histograms
  - E.g., size of I/O request → 4KB





vscsiStats Reports Results Using Histograms

Read/Write Distributions are available for our histograms

- Overall Read/Write ratio?
- Are Writes smaller or larger than Reads in this workload?
- Are Reads more sequential than Writes?
- Which type of I/O is incurring more latency?

In reality, the problem is not knowing which question to ask

- Collect data, see what you find

I/O Size

- All, Reads, Writes

Seek Distance

- All, Reads, Writes

Seek Distance Shortest Among Last 16

Outstanding IOs

- All, Reads, Writes

I/O Interarrival Times

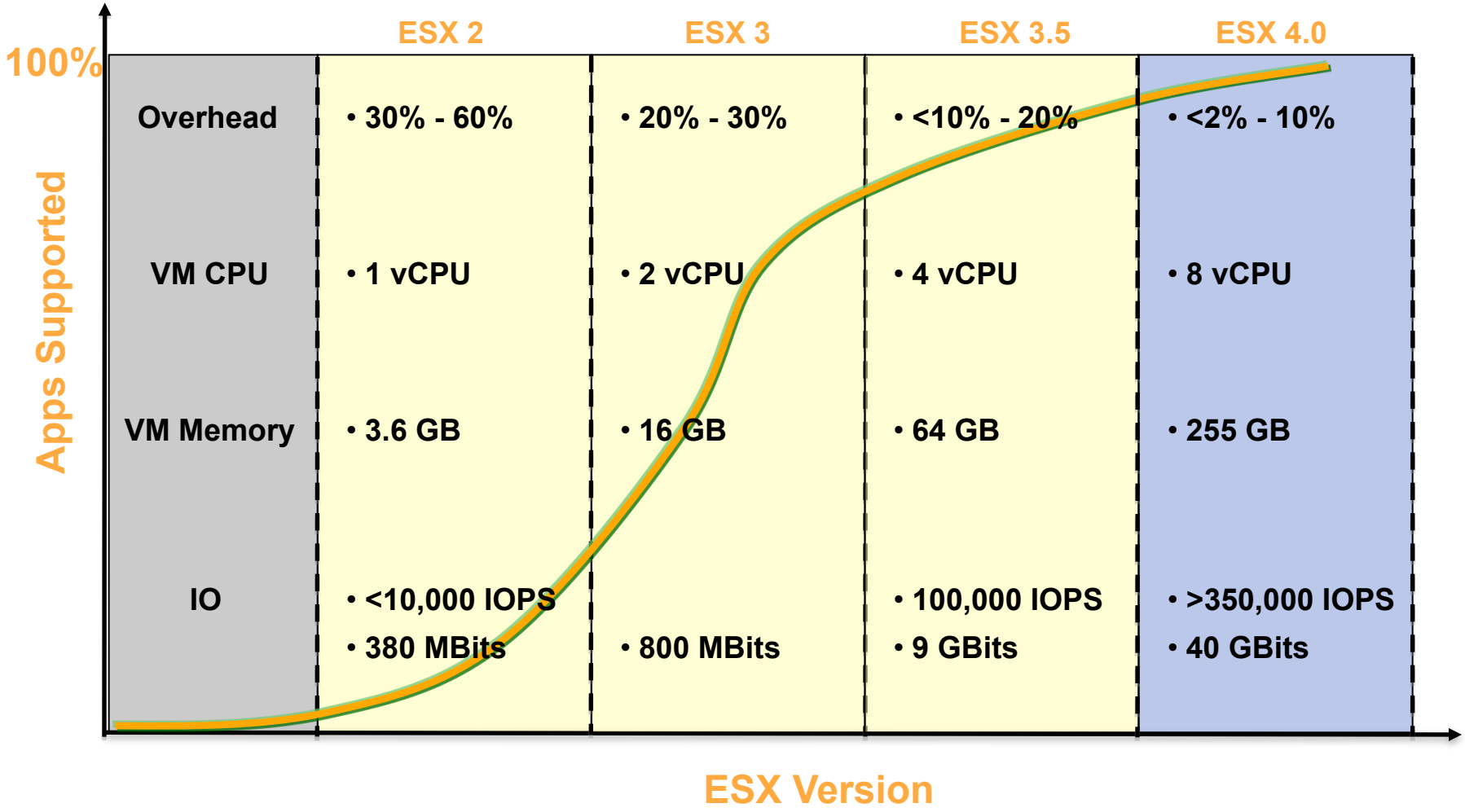
- All, Reads, Writes

Latency

- All, Reads, Write

vSphere Update

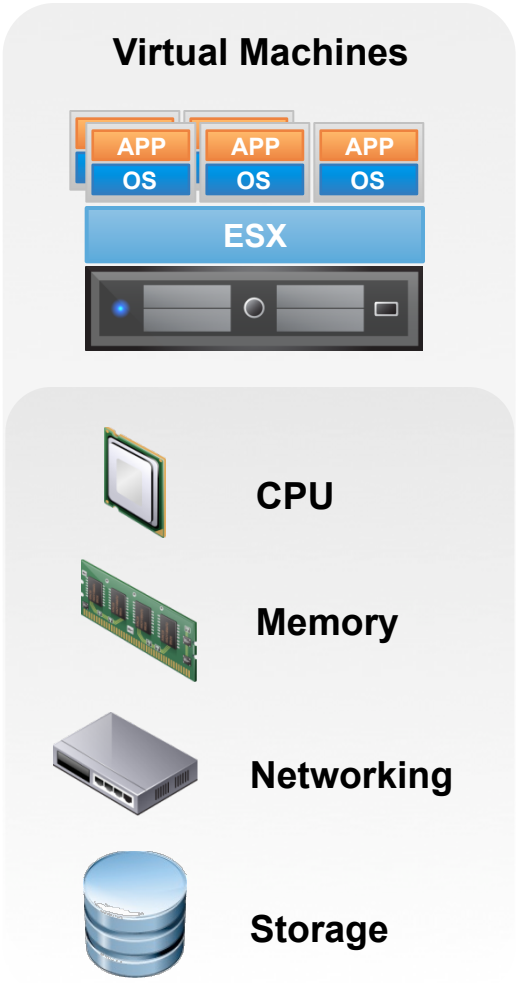
>95% of Applications Match or Exceed Native Performance on VMware Infrastructure



Source: VMware Capacity Planner analysis of > 700,000 servers in customer production environments

vmware

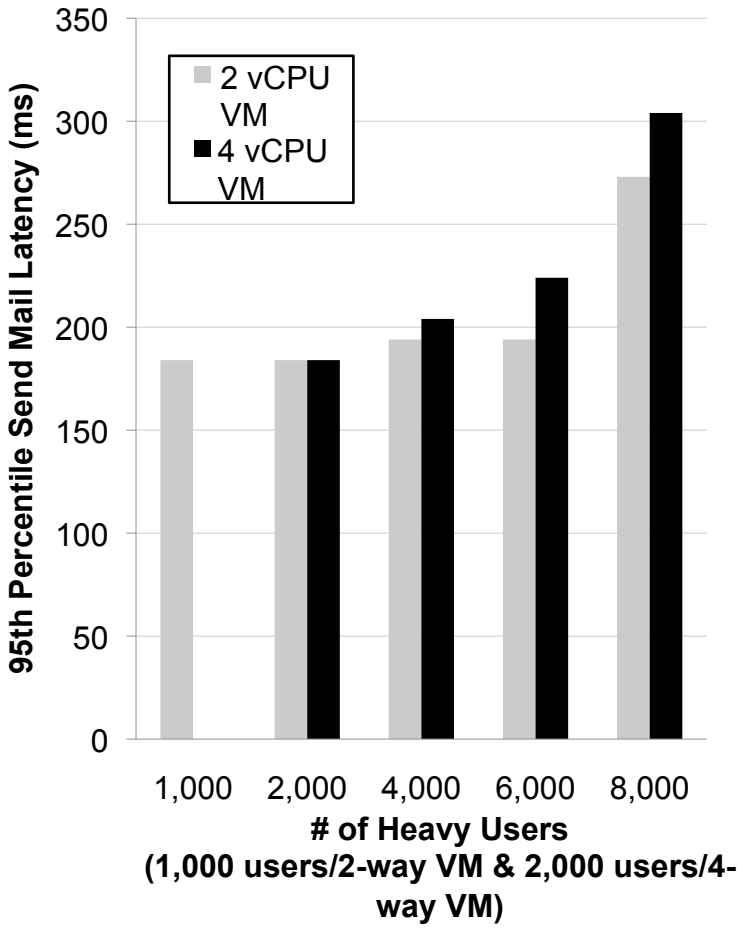
"Speeds and Feeds" Optimization for the Highest Consolidation Ratios

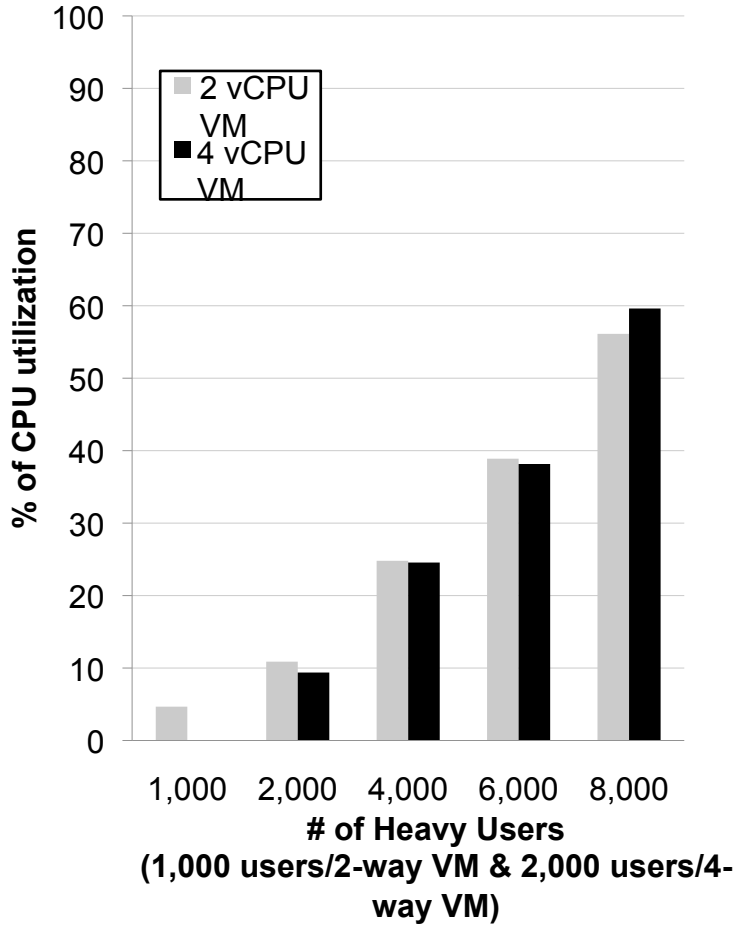


VM Scale Up	8-way vSMP and 255 GB of RAM per VM
Hardware Scale Up	64 cores and 1 TB physical RAM
Hardware Assist Purpose Built Scheduler	Lowest CPU overhead
Hardware Assist Page Sharing Ballooning	Maximum memory efficiency
VMXNET3 VMDirectPath I/O	Wirespeed network access
Storage stack optimization VMDirectPath I/O	Greater than 350k iops per second Lower than 2 millisecond latency

Current NEW

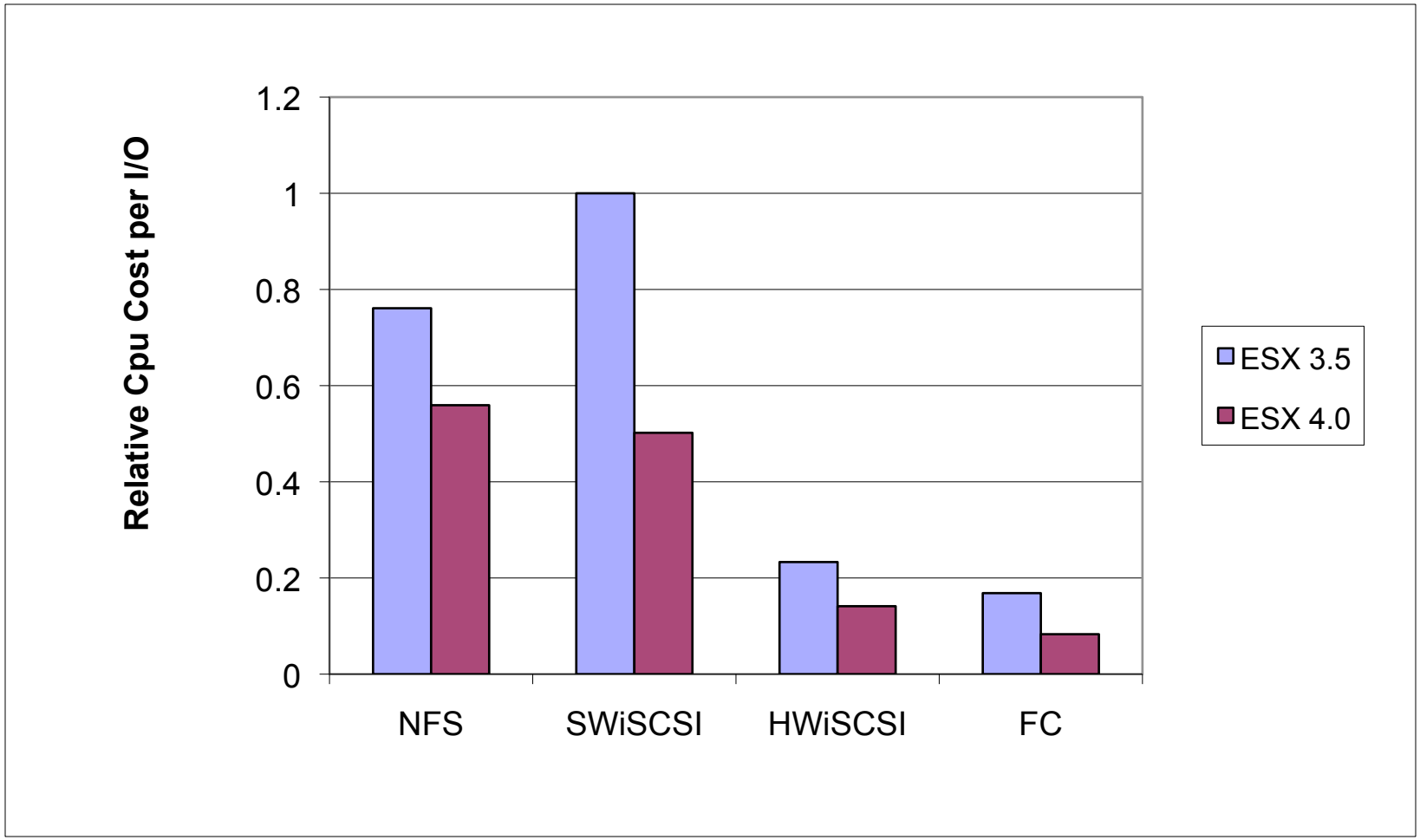
Exchange 2007 on vSphere: SMP Efficiency



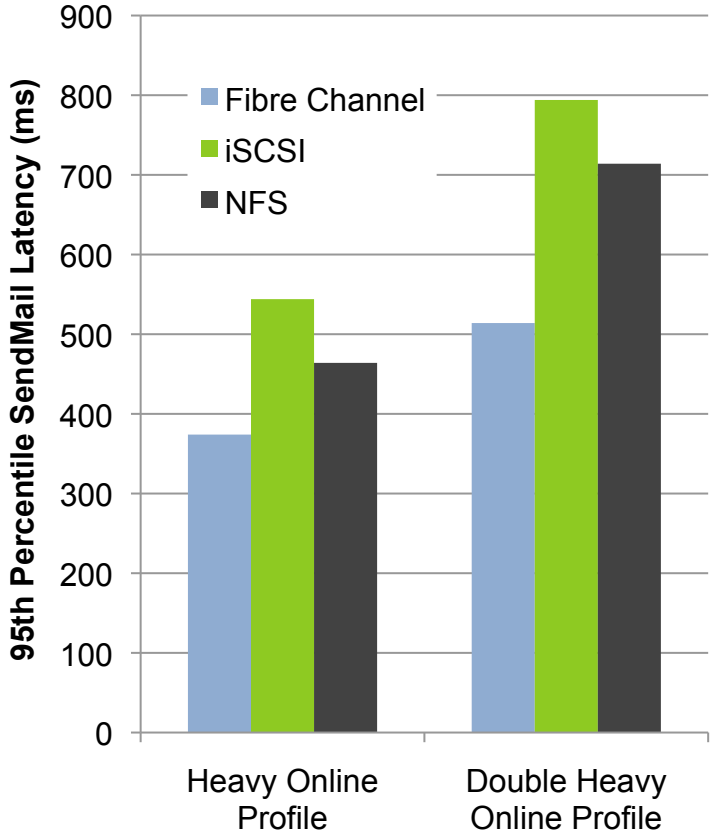


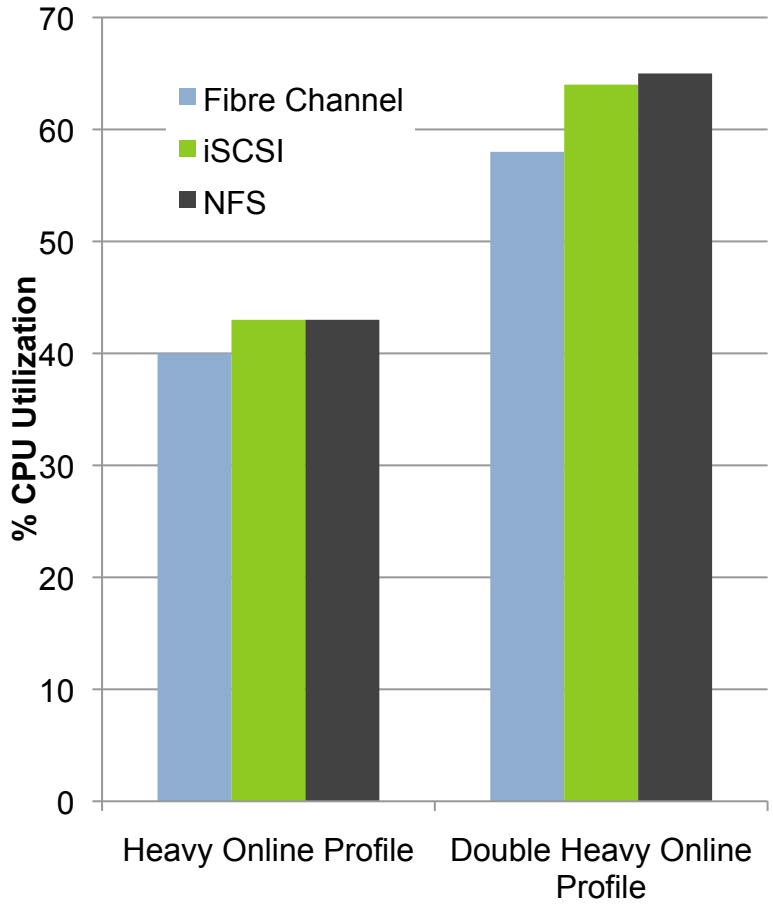
vmware

Storage Protocols: vSphere versus VI3



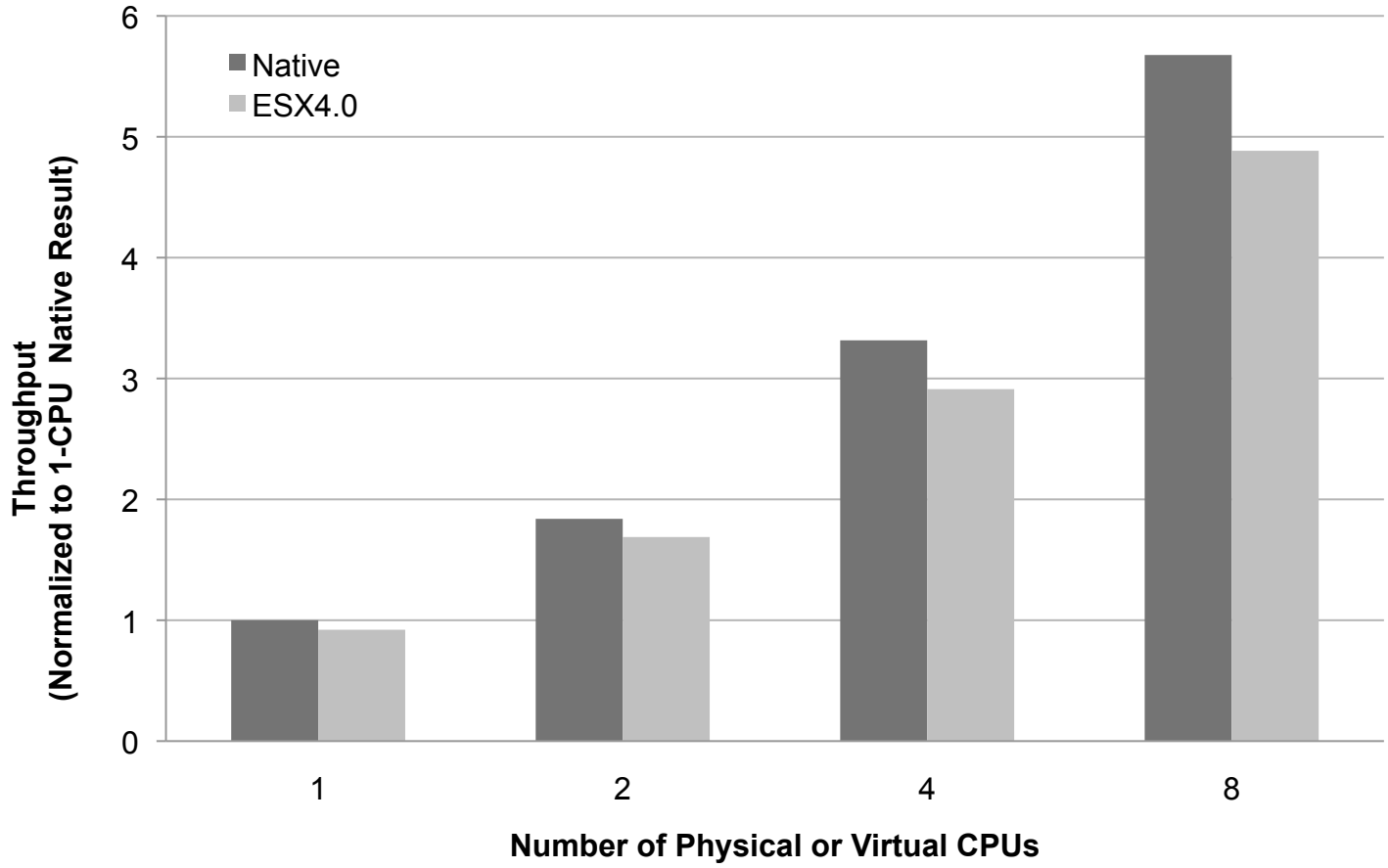
Storage Protocols and Exchange on vSphere





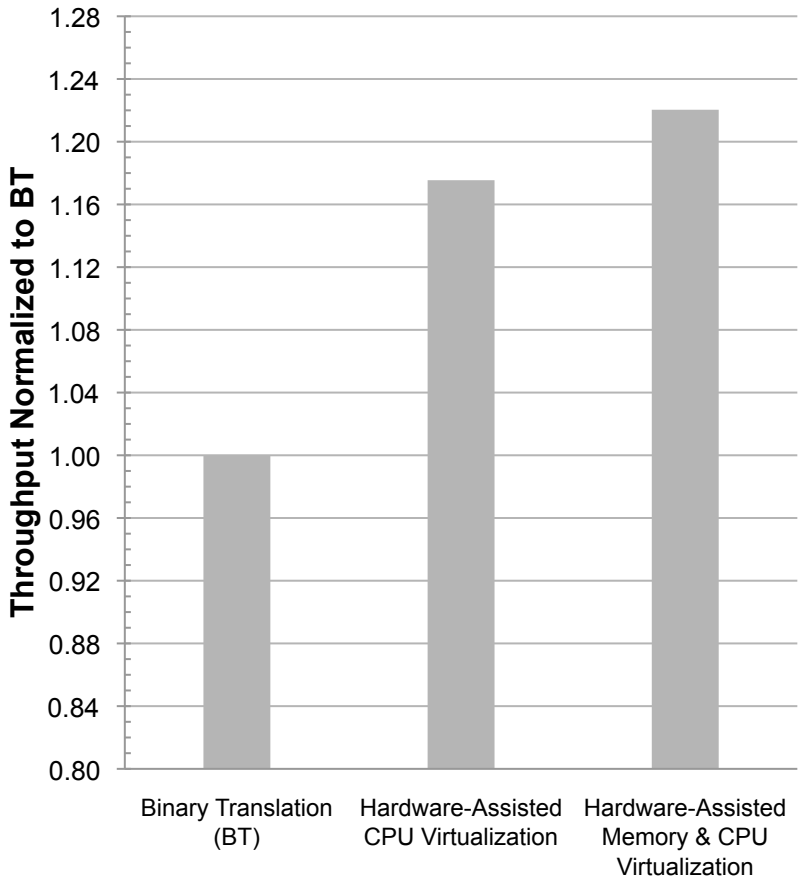
vmware

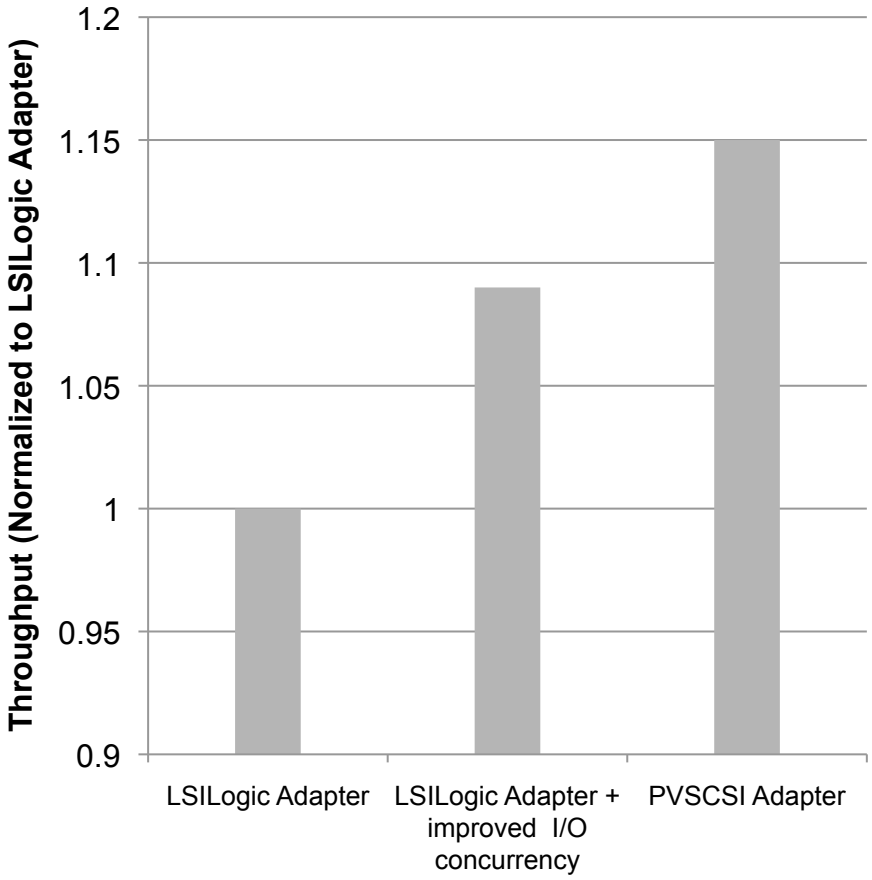
SQL Server 2005 on vSphere: Efficiency



vmware

SQL Server 2005: vSphere Features





vmware

Summary

Newer hardware improves virtualization performance

Traditional application, storage, networking best practices must be followed

Consolidation provides new challenges and opportunities that must be planned for

Performance Resources

The performance community

- http://communities.vmware.com/community/vmtn/general/performance

Performance web page for white papers

- http://www.vmware.com/overview/performance

VROOM!—VMware performance blog

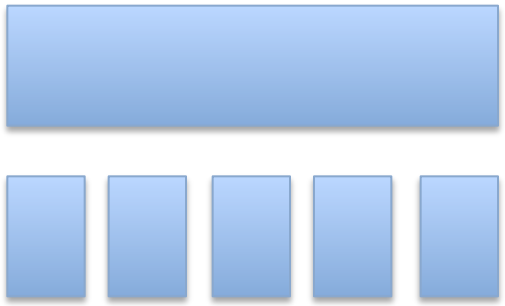
- http://blogs.vmware.com/performance

Backup

Large Pages

Guest/Host	Small	Large
Small		
Large		

Fragmentation



vSphere Thin Provisioning

Virtual Machine Sizing—NUMA

Memory accesses from CPU 0

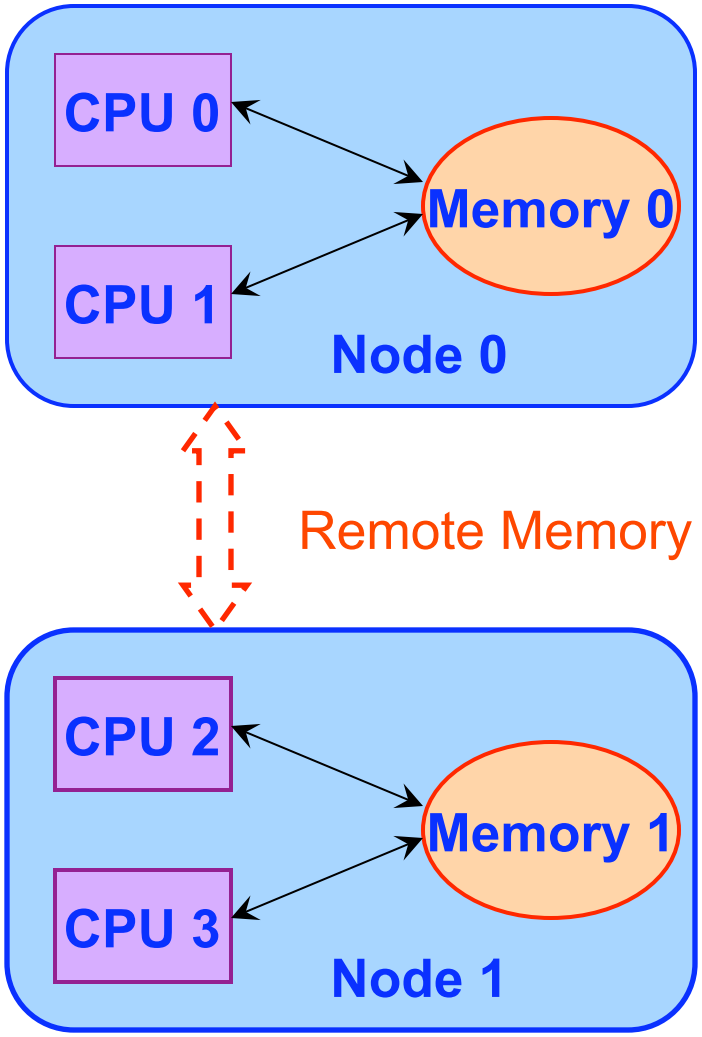
- To Memory 0 is local
- To Memory 1 is remote
- Remote access latency >> local access latency

# of vCPUs ≤ # of CPUs / node

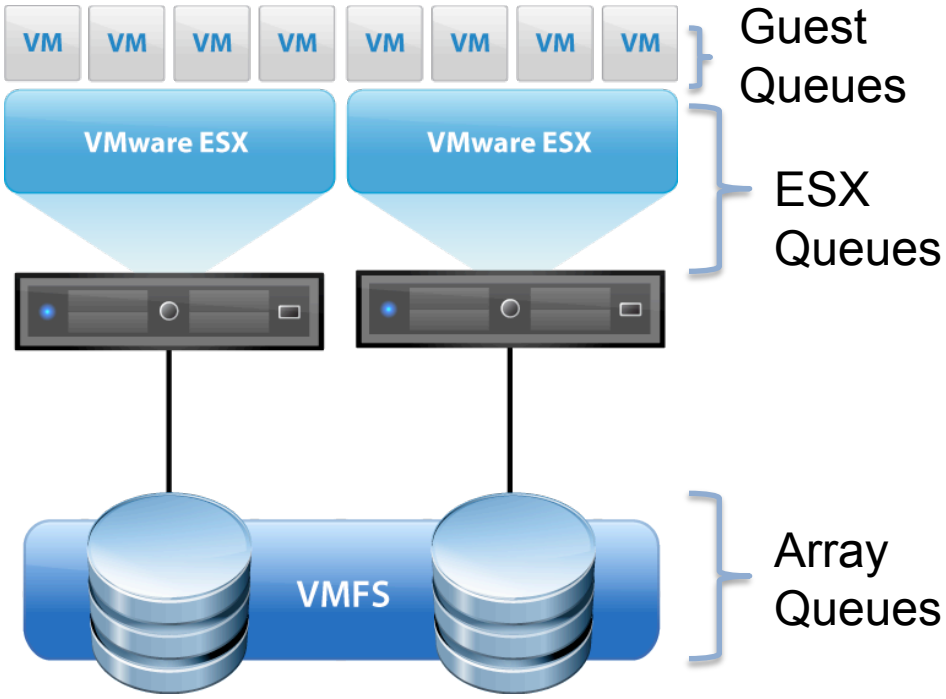
- ESX enables NUMA scheduling

If VM MemSize < Node Memory size

- No remote access penalty



Host Configuration: Storage Queues



ESX queues can be modified to increase throughput

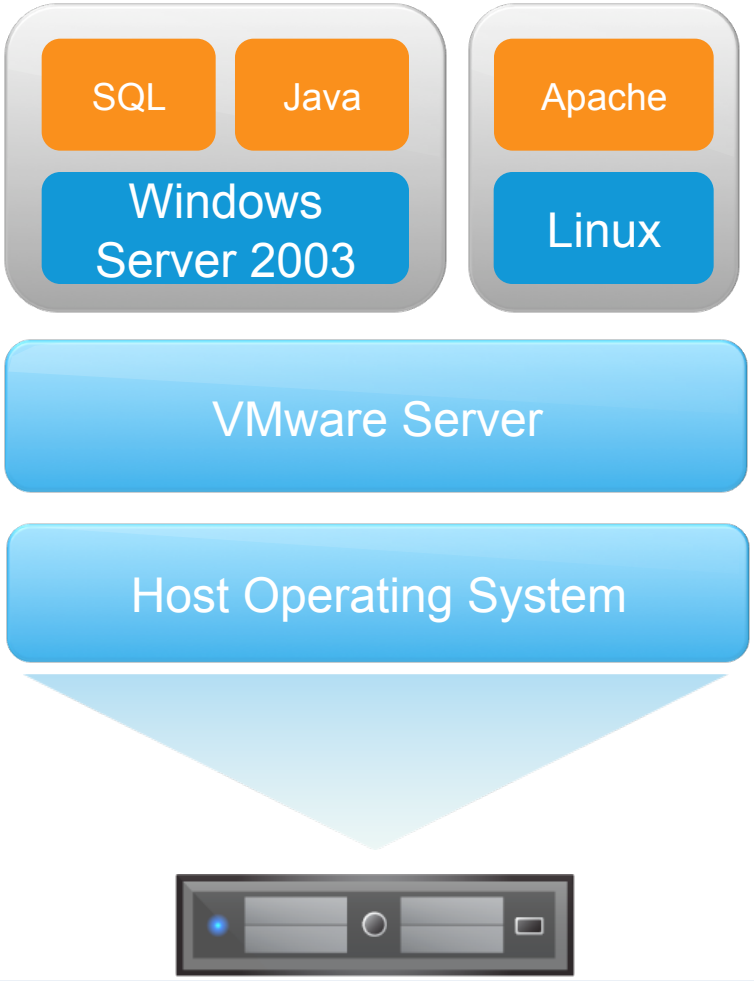
- This can benefit benchmarks to a single LUN
- Rarely required in production systems
- Oversized ESX queues on multiple servers can overload array

Kernel latency is a sign that ESX queues should be increased

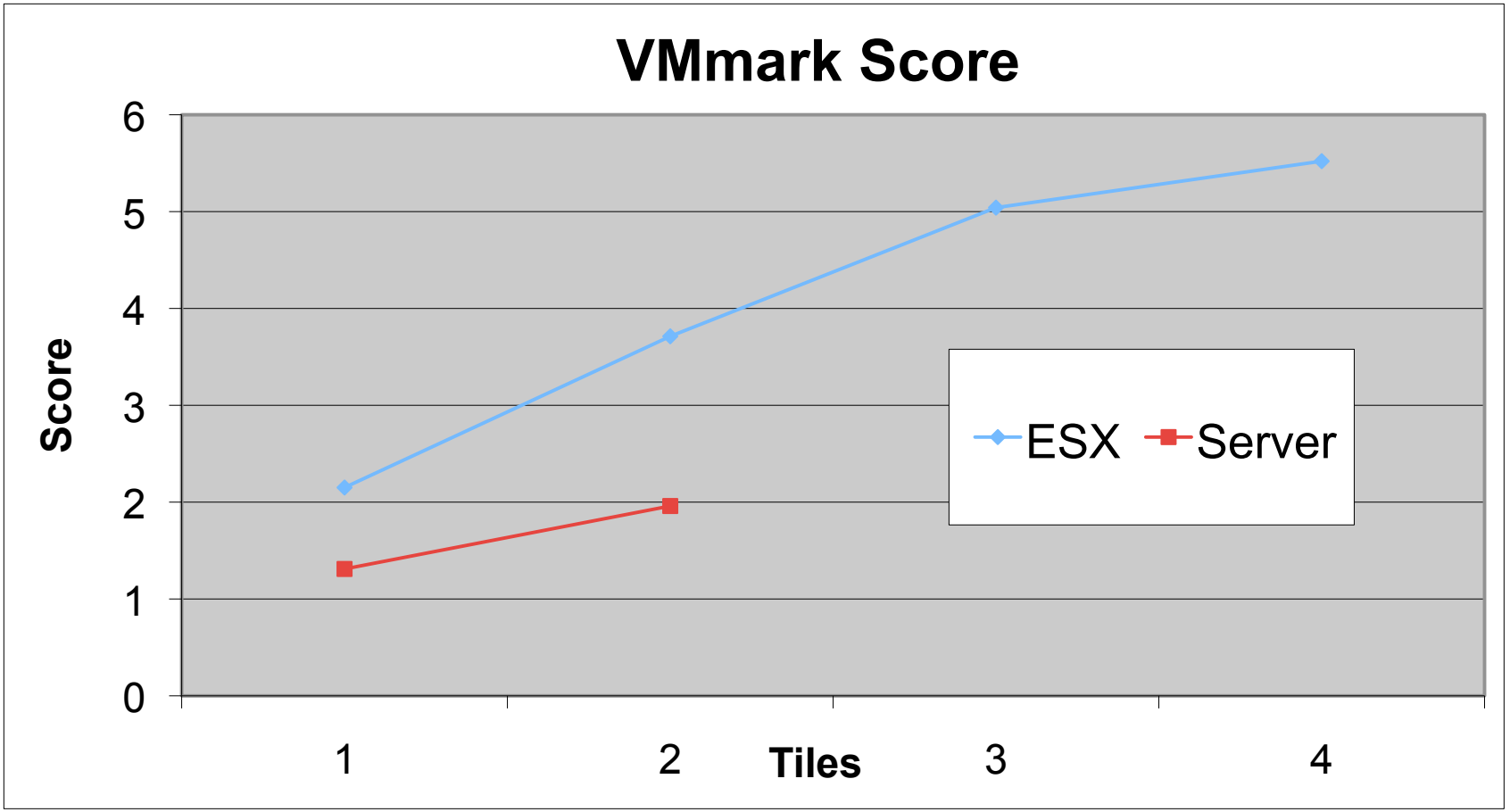
Choose the Right Virtualization Software

Hosted products aren't designed for meet the most extreme needs

- ESX demonstrates better host and VM scaling

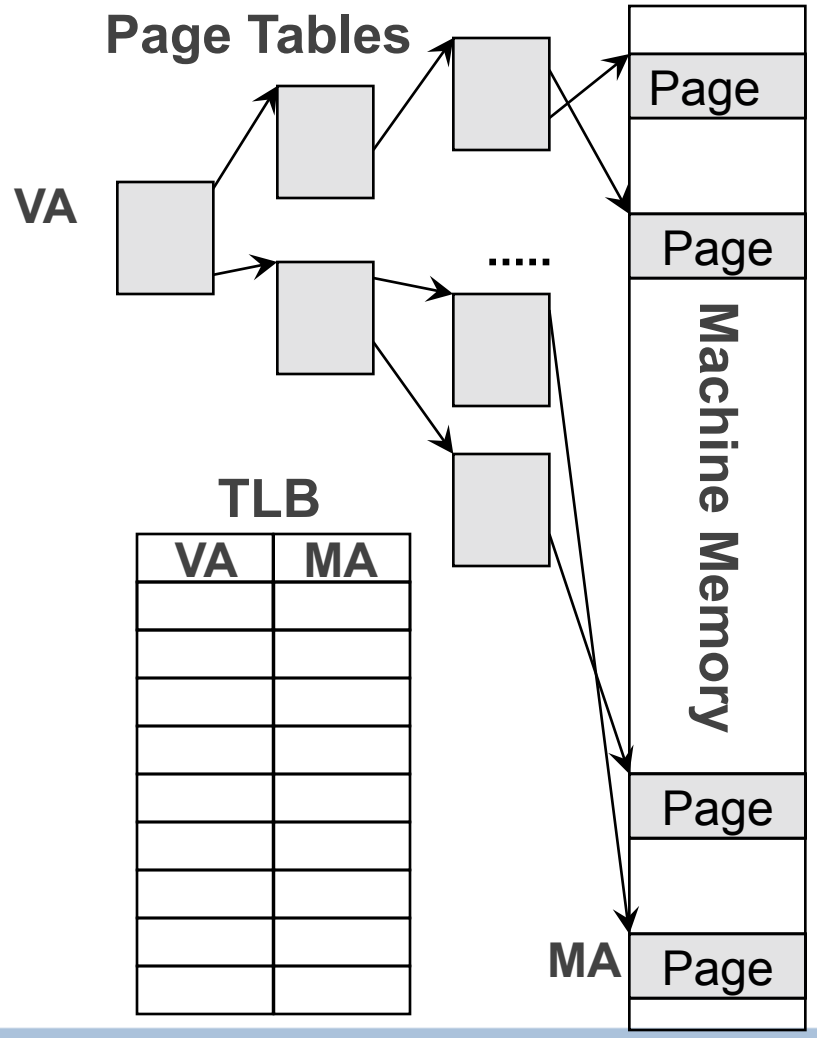


VMware ESX Compared to VMware Server



Single tile score higher than reference system

Address Translation



Virtual addresses (VA) mapped to machine addresses (MA) via page tables

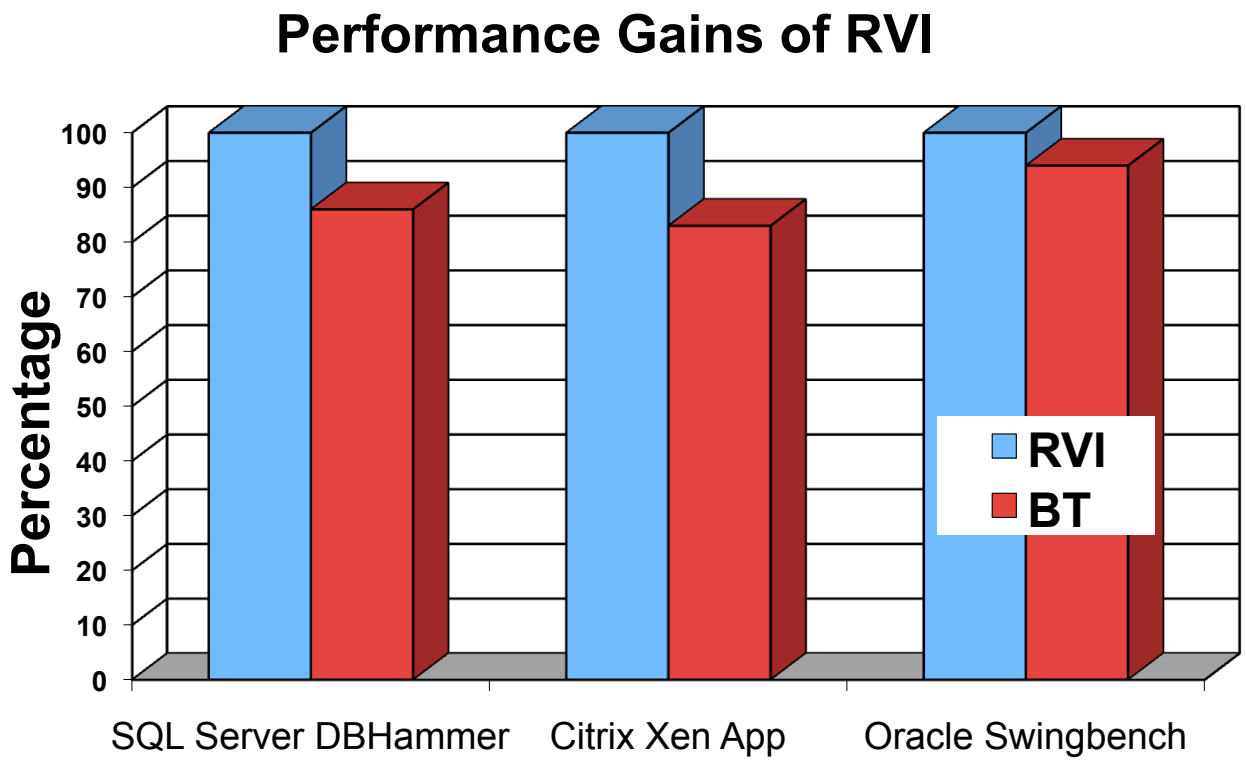
- Page table walks are expensive

Translation look-aside buffer (TLB) stores recent mappings and avoids page walks

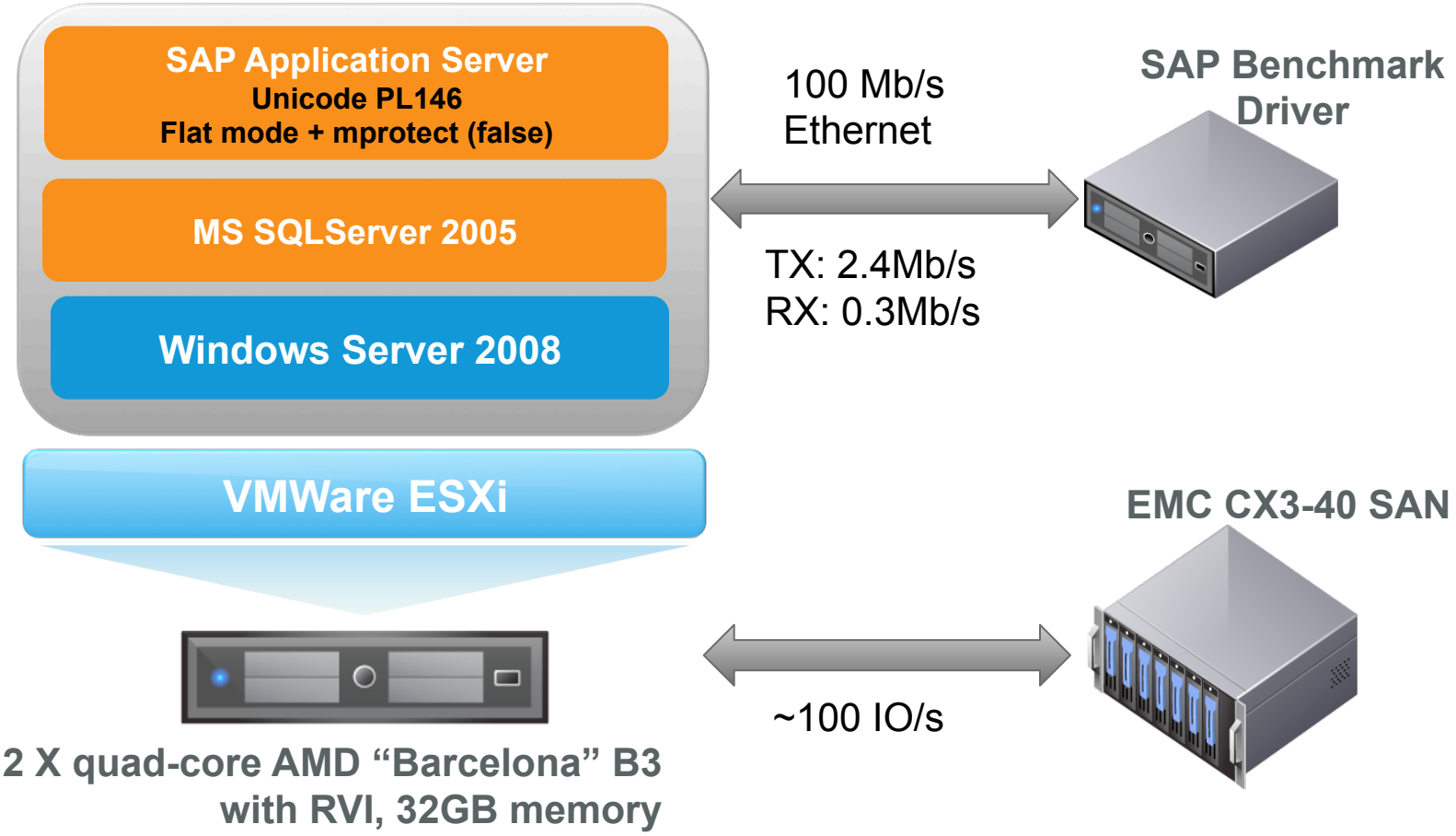
Improvements:

- Larger pages means more TLB hits
- Hardware assistance to virtual mapping means more efficient page table and TLB maintenance

AMD Hardware-assisted MMU Support (RVI)

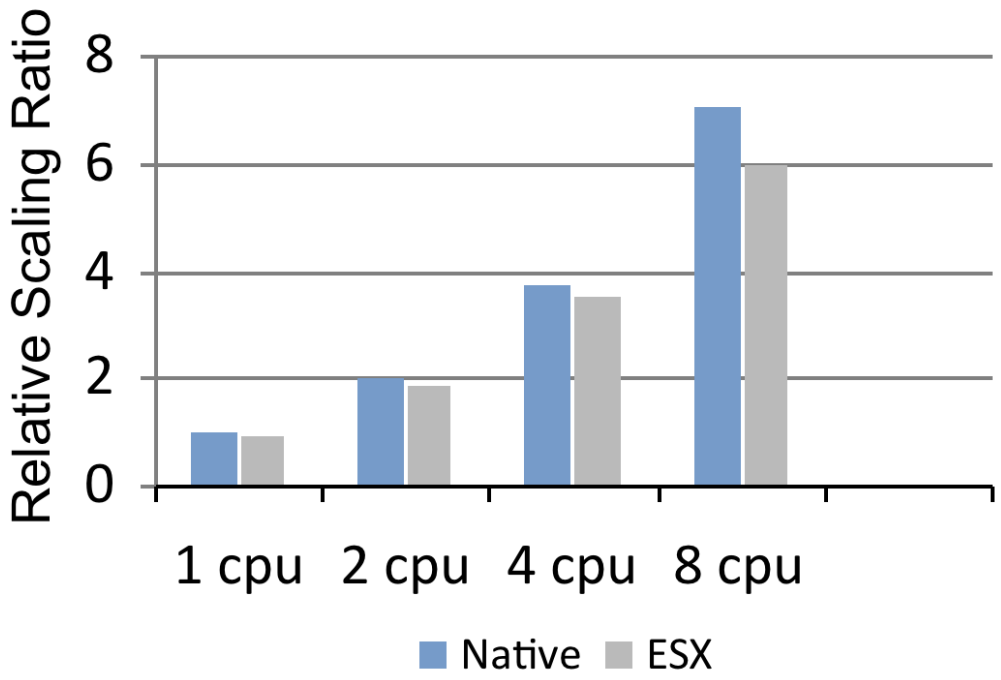


Hardware Configuration In Action: SAP



SAP SD Performance on ESX

- ESX achieves 95% of native performance on a 4vCPU VM
- 85% of native performance on an 8 vCPU VM on 8 pCPU host
- Linear scaling from 1 vCPU -> 4 vCPU



SAP SD 2-Tier performance on ESX

- SAP SD performance sensitive to software configuration and ESX monitor type:

SAP configuration Mode	Deployment	Recommended Monitor Type	Guest tunable	Effect
View Model	Production	RVI (Default)	Large pages	H/W assist reduces MMU overheads
Flat model + mprotect = true	Production	RVI (Default)	Large pages	H/W assist reduces MMU overheads
*Flat model + mprotect = false	Mostly benchmark	SVM (UI Option)	Larges pages up to 12% benefit	S/W MMU benefits up to 5%

* Configuration used in our experiments

- In most cases, default H/W MMU provides best results
  - Experiment with your individual workloads