SDC
STORAGE DEVELOPER CONFERENCE
SNIA SANTA CLARA, 2015

Avoiding common storage development pitfalls:

Approaches and Lessons Learned with the VMware vSphere platform

Scott H. Davis, CTO

Infinio Systems, Inc.

Welcome!

Scott H. Davis

- CTO, Infinio
- 25+ year IT veteran
- Former VMware EUC CTO & Chief Data Center/Storage Architect
- Founder, President, CTO of Virtual Iron
- 17 Patents for Virtualization, Storage, Clustering, and EUC technologies
- vExpert 2015

www.TalkingTechwithSHD.com

@shd_9

Agenda

- Introduction: Decoupled Architecture
- Storage & virtualization overview
- VMware storage stack & I/O intercept techniques
- Infinio solution & architecture
- Infinio intercepts & lessons

Storage in 2004

File services in front of SAN arrays
(EMC Celerra and Clariion)

Early storage tiering (Compellent)

NetApp SAN with 2 FAS Filers

HP server with EVA SAN

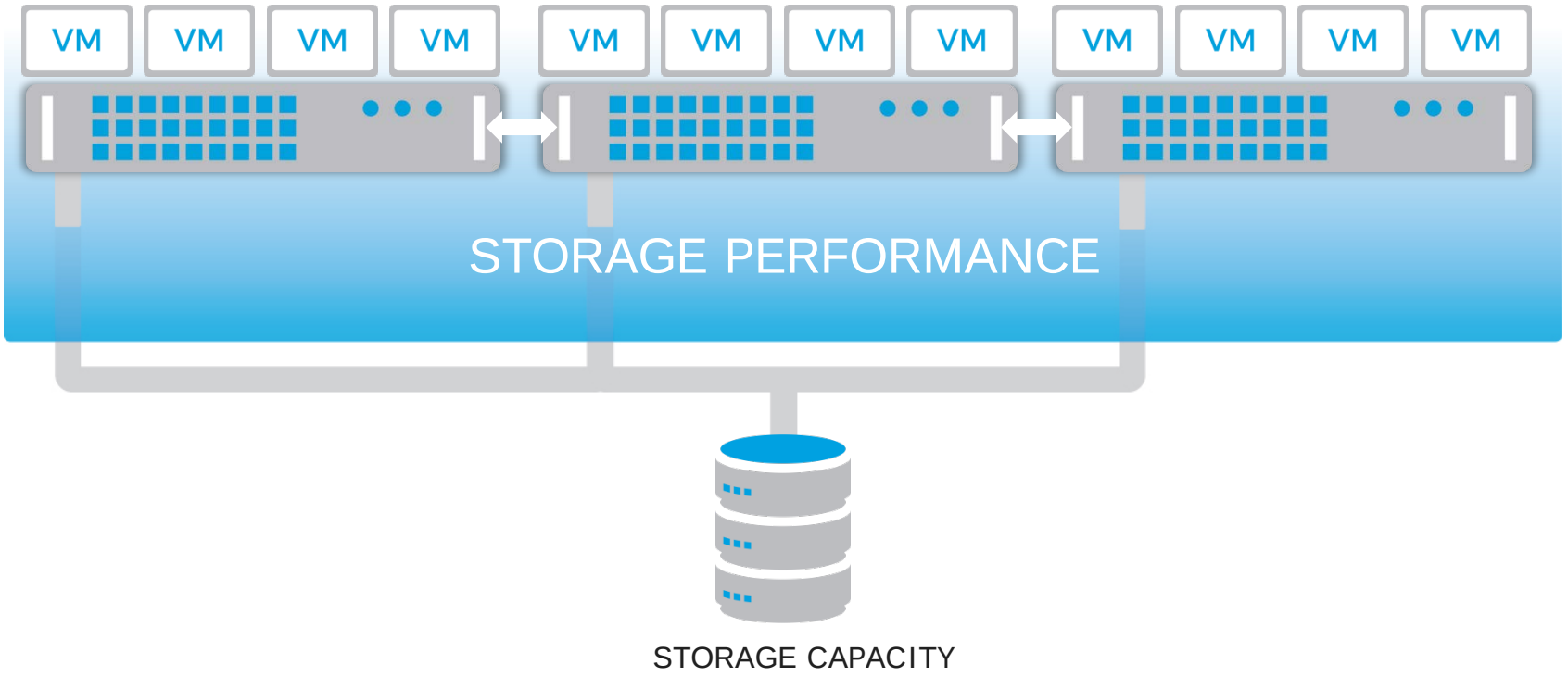
Storage in 2015. . .

But storage has just two jobs that have to get done:

- ❑ Provide the **storage performance** that applications need
- ❑ Provide the **storage capacity** that applications need

These two jobs are best done separately

Infinio's Decoupled Architecture



10x improvement in latency	**Reduced performance costs ($/IOPS)**	**Scale-out I/O with application growth**	**Reduced capacity costs for any array ($/GB)**

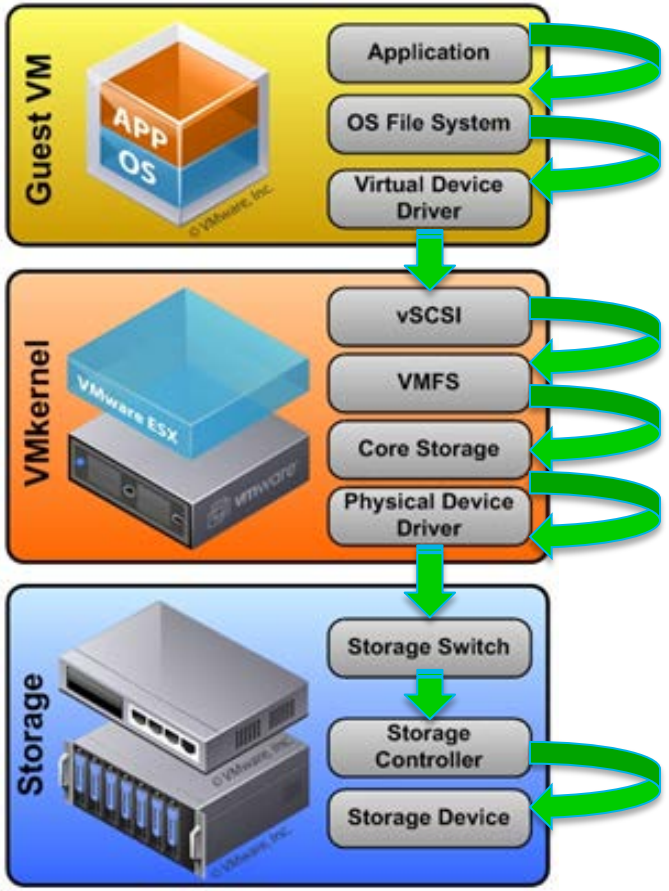
Virtualization Impact

General virtualization storage challenges

- Large set of physical machines consolidated
- Diverse set of applications
- More workload variety and concentration
- Blender effect, mix of read/write ratios

VMware specific storage challenges

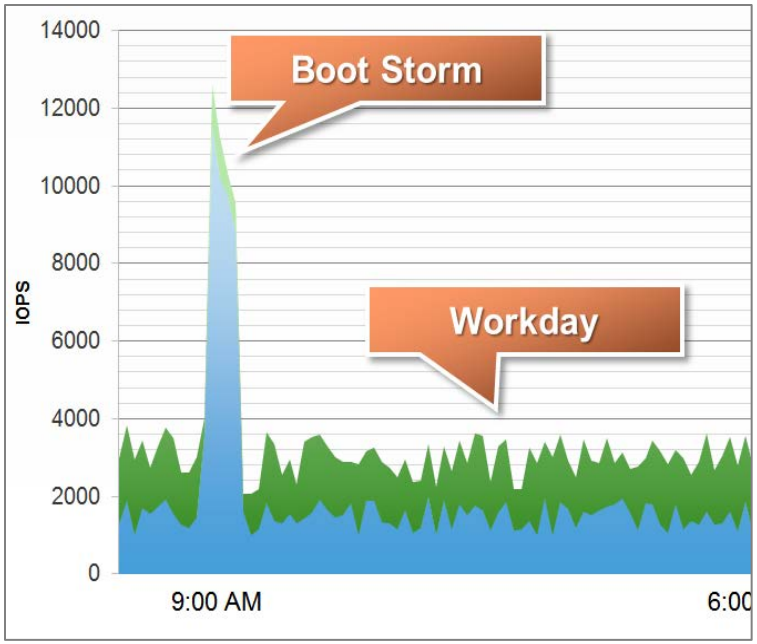
- VM provisioning operations mixed with applications
- More abstractions/layers means less visibility
- Clustered locking protocols
- Workload mobility (vMotion, DRS)



Virtual storage I/O path

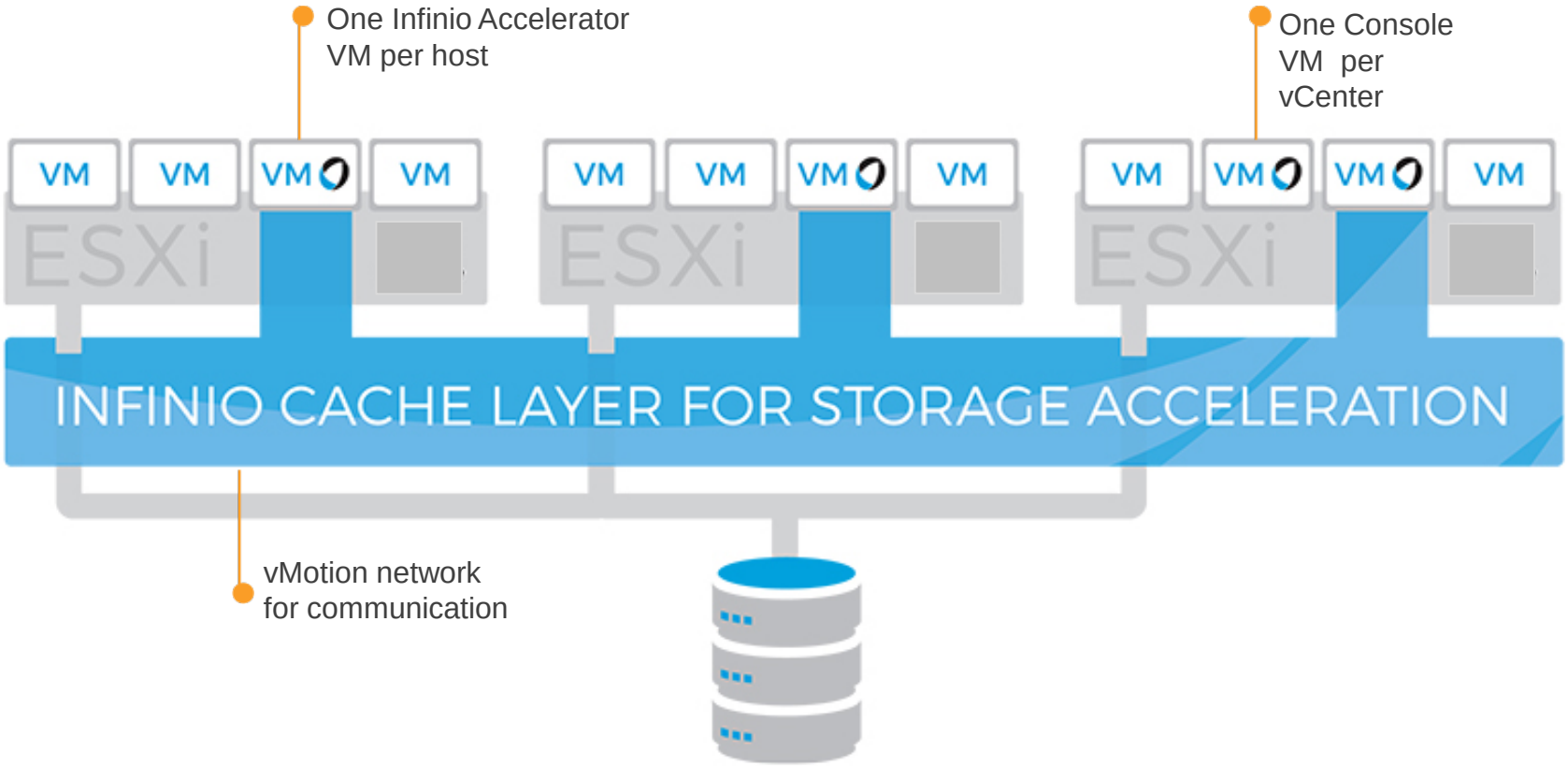
Unique VDI storage challenges

- Capacity and performance cost
  - Virtual desktop should cost less than a physical desktop = high consolidation ratio
  - Shared data center hardware vs. dedicated desktops
  - User experience is key - 25-75 IOPs per desktop
- VDI
  - Desktop OS assumptions
  - Impact of synchronized peaks (e.g., boot storms, login storms, virus scans)

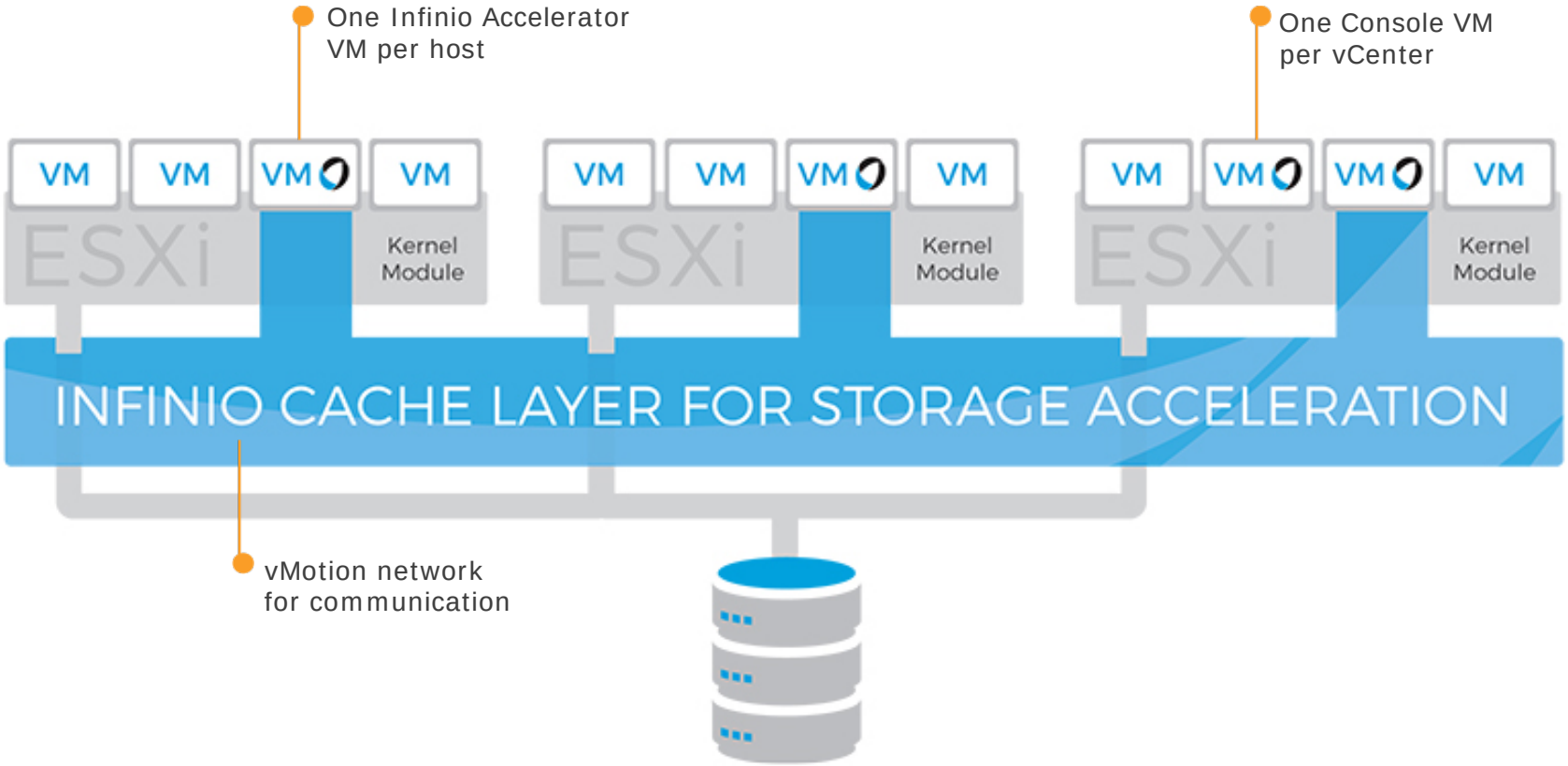


Infinio Decoupled Architecture

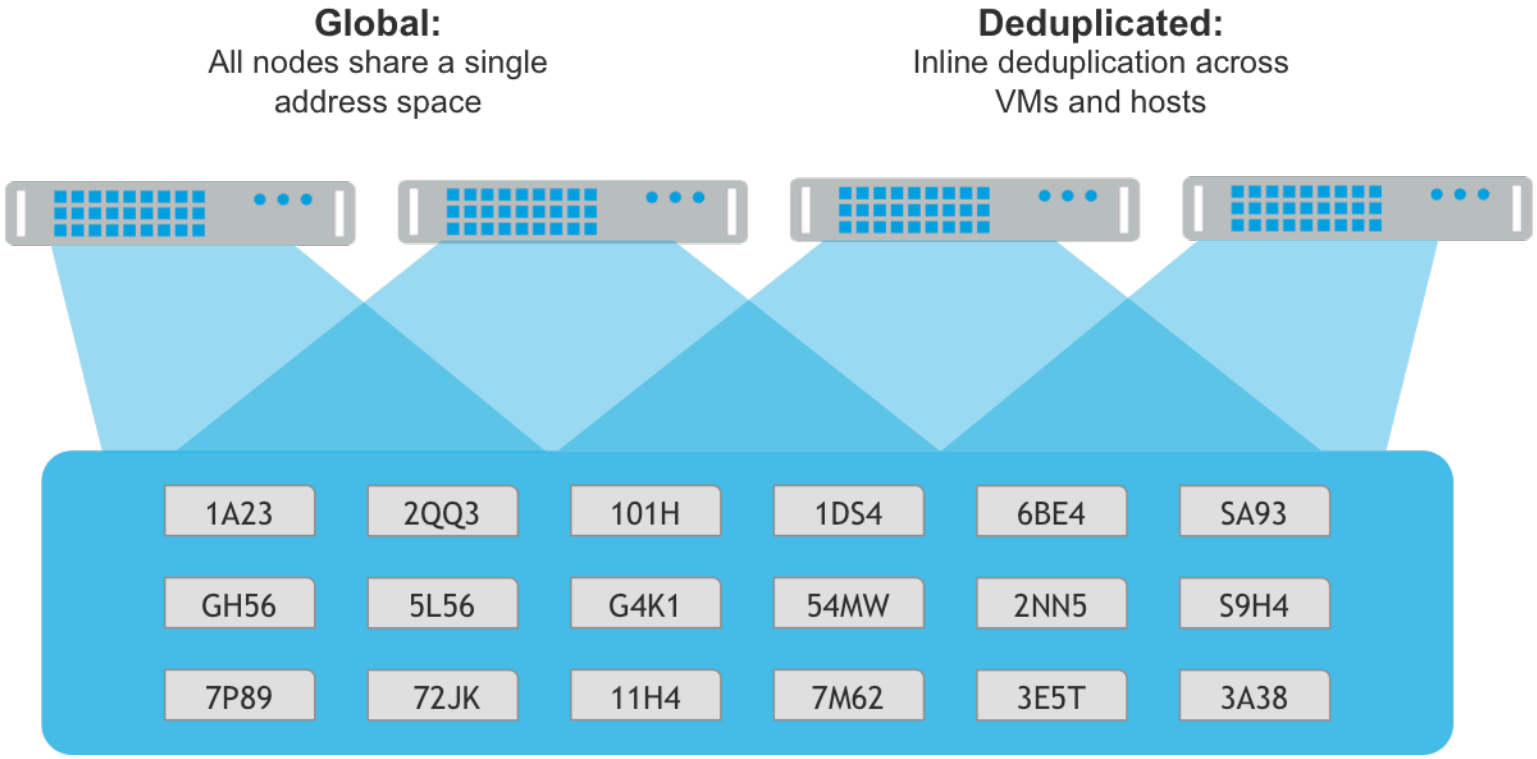
Infinio's Storage 1.0 Acceleration platform



Infinio's Storage 2.0 Acceleration platform



Infinio's cache architecture

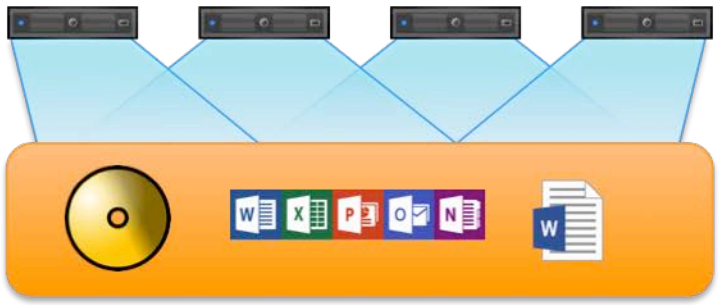


With a 5:1 dedupe rate, an 4-node Infinio cluster starts with an effective size of 160GB and can grow much larger.

Infinio's global deduplication in action

VDI

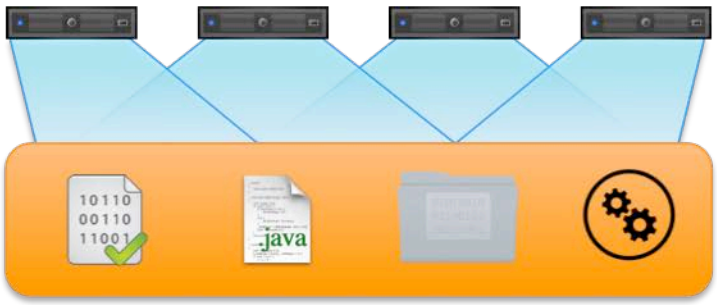
- ❑ Gold images
- ❑ Common application executables
- ❑ Common user files



Customer *National Specialty Alloy* saw sustained offload rates of 61%

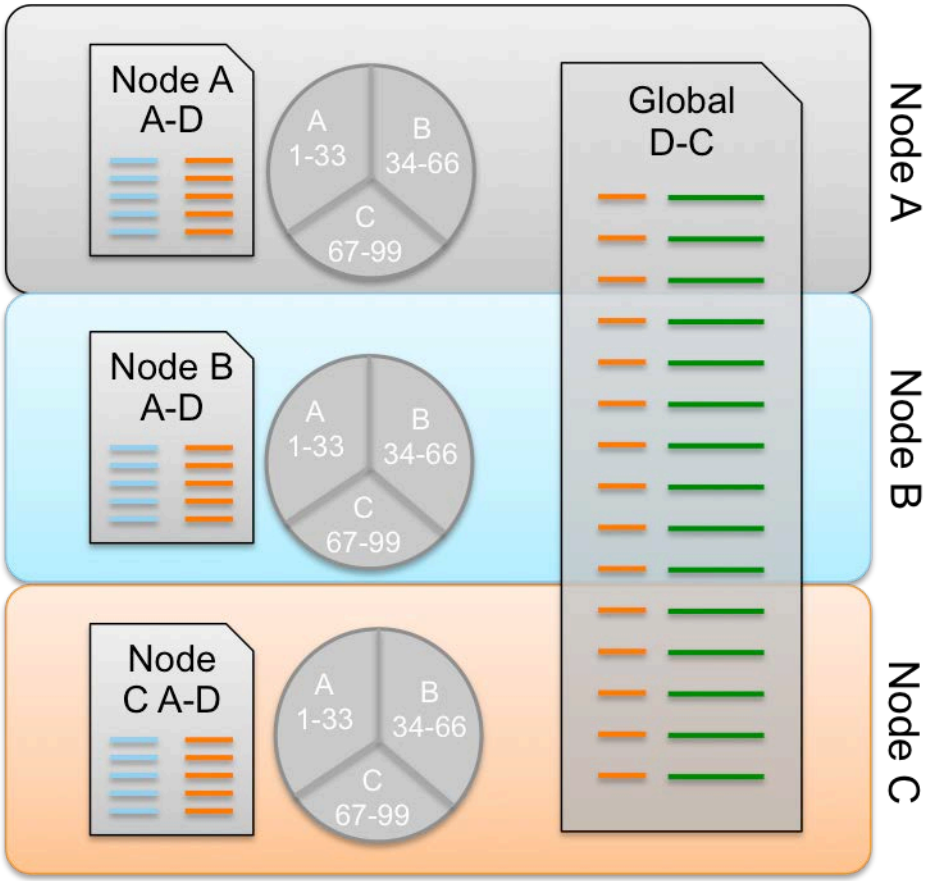
Software Development

- ❑ Source code for slightly different versions
- ❑ Test automation on the same code
- ❑ Exemplar data

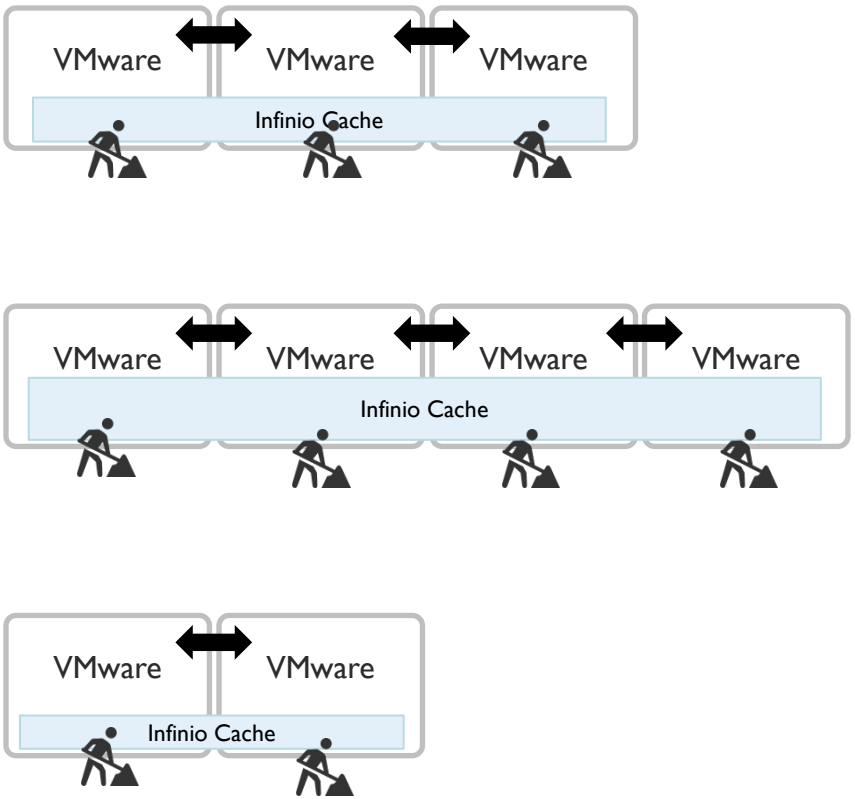


A large consumer goods company saw build time drop from 2 hours to 15 minutes

Infinio's directory architecture

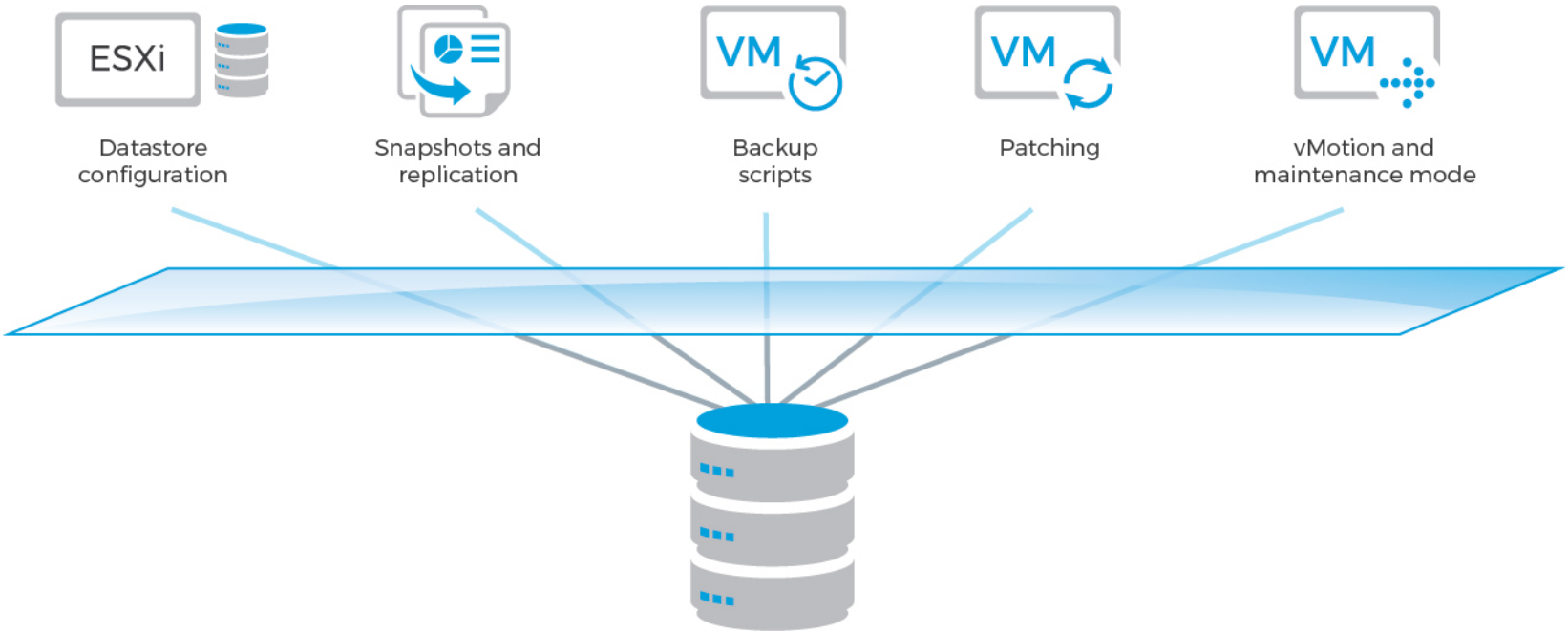


Infinio's true scale-out design

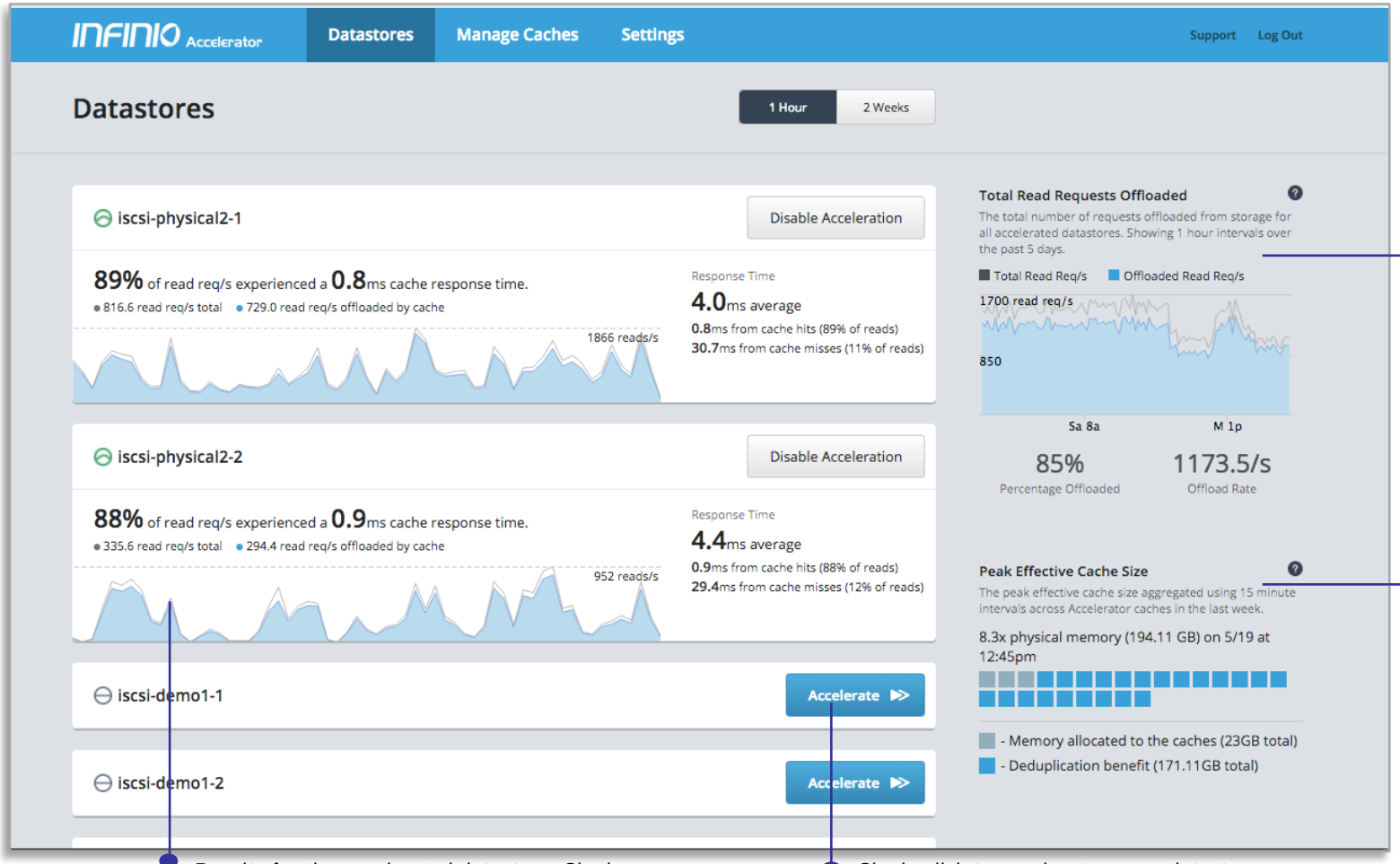


- Work is shared across nodes equitably
- Linear performance increases as you add more nodes
- Uniform distribution of work preserved during cluster changes:
  - Add or remove a node
  - Add or remove a datastore
- Cache contents are accessible across nodes (but not replicated)
- Cache consistency is built in

Infinio's operational transparency



UI: simple, intuitive management



Global statistics for all accelerated datastores

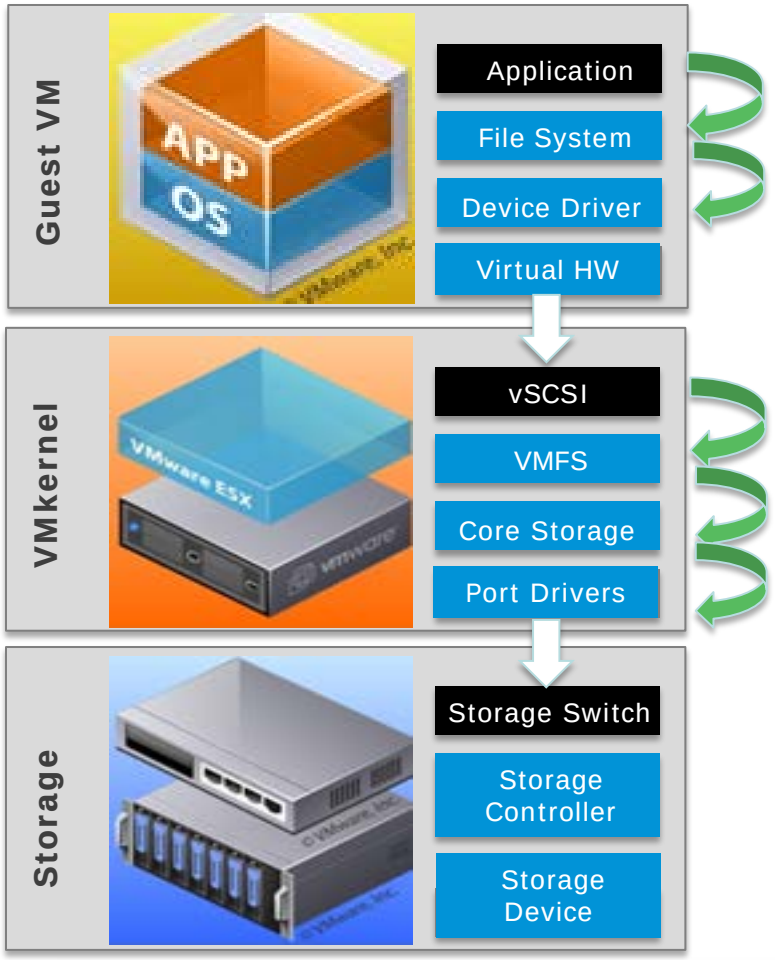
Global view of the peak effective cache size. Deduplication benefit at-a-glance

Results for the accelerated datastore. Single-click to disable acceleration.

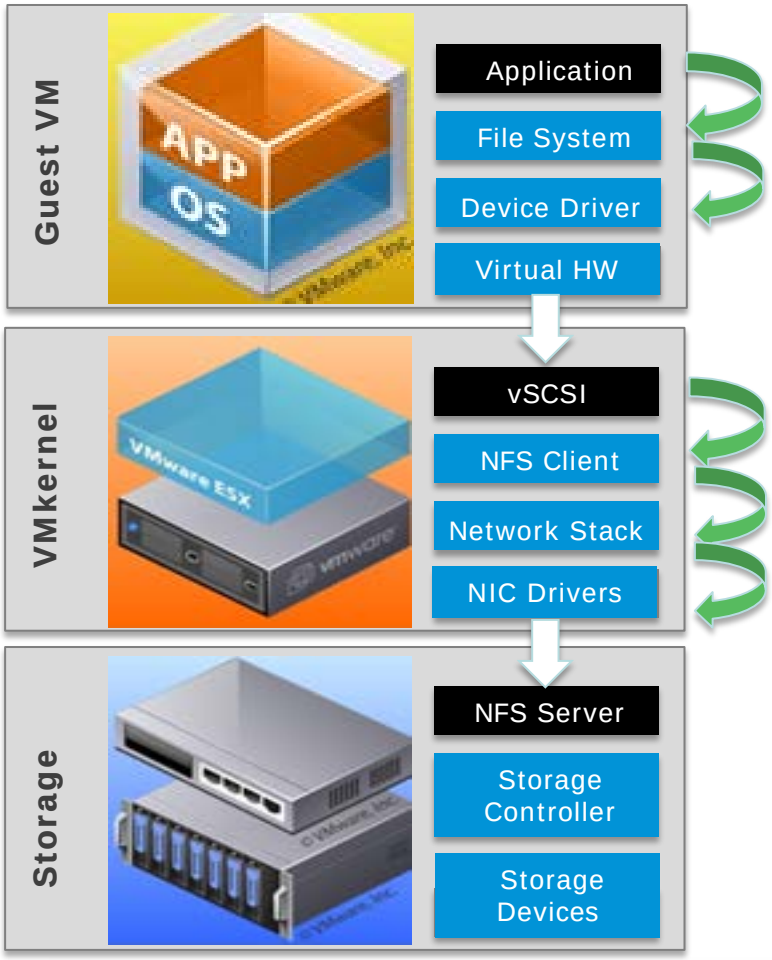
Single click to accelerate a new datastore.

VMware Storage & Intercept Techniques

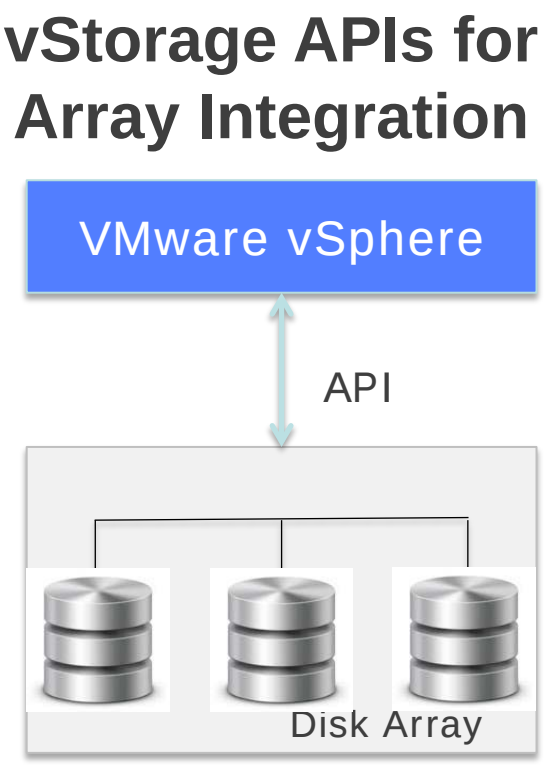
Virtual storage block I/O path



Virtual storage NFS NAS I/O path



VAAI

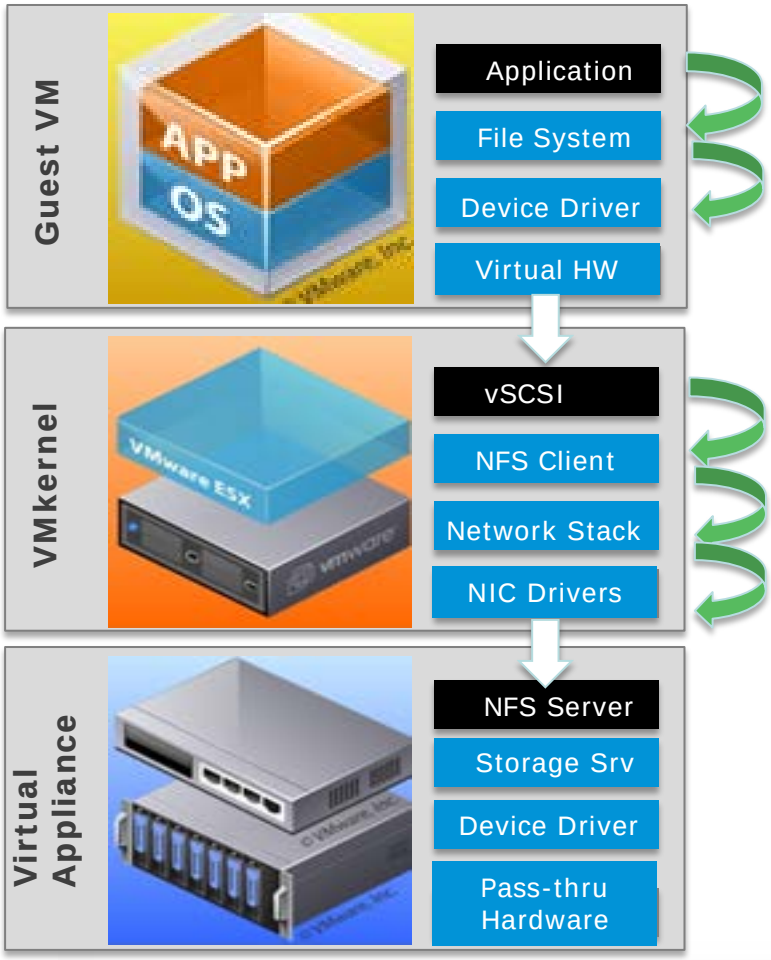


- T-10 primitives to offload VMware server resource intensive storage operations to array
- Primitives
  - Xcopy
  - Write-Same
  - Trim/Unmap
  - Compare-and-Swap
- Snapshots
- All filter types/storage intercepts must carefully consider

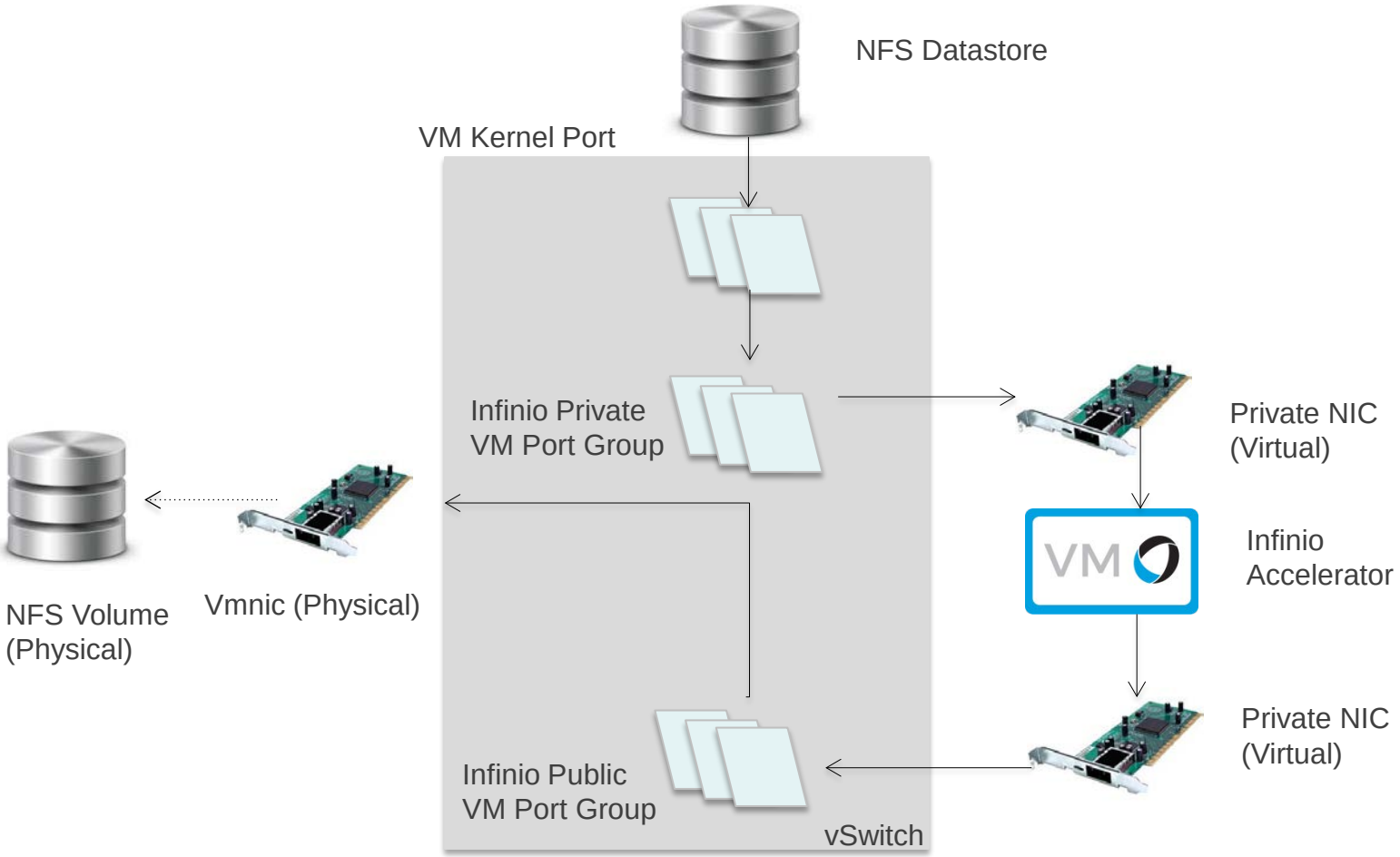
Intercept techniques

- Not explicitly covered:
  - Para-virt in guest interface
  - Proxy device
- Virtual Appliance
- VMkernel Drivers
- PSA Plug-in
- VAIO

NFS Intercept – virtual appliance



Infinio V1.0 NFS Intercept

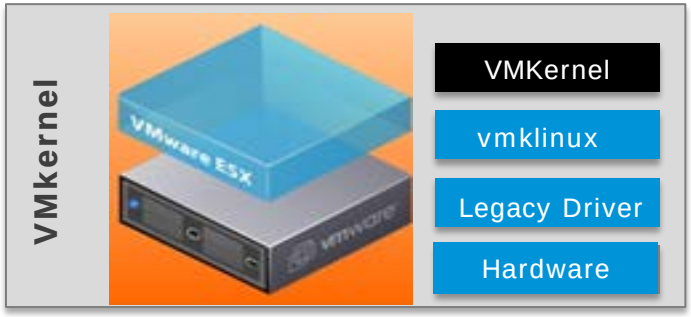


Lessons: Virtual Appliances

- Virtual Appliance
  - Simplest to implement
  - Typically an NFS Server
  - Heaviest resource usage, least balanced
  - VM scheduling & context switch on IO operation
  - Fewest constraints
- Infinio-specific lessons:
  - Extra overhead – context switch on all I/Os
  - Datastore offline events

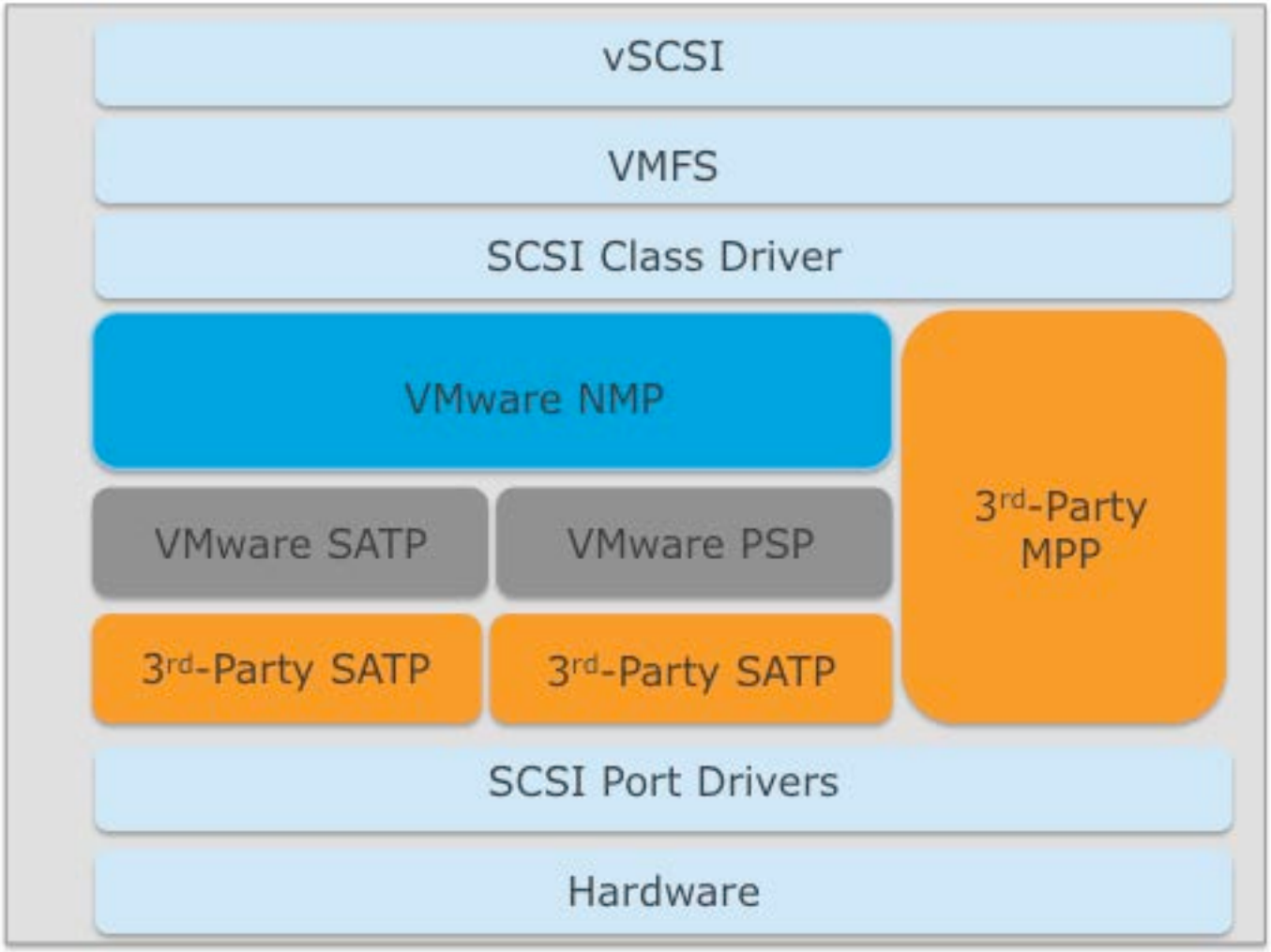
ESX Kernel drivers

Vmklinux driver model

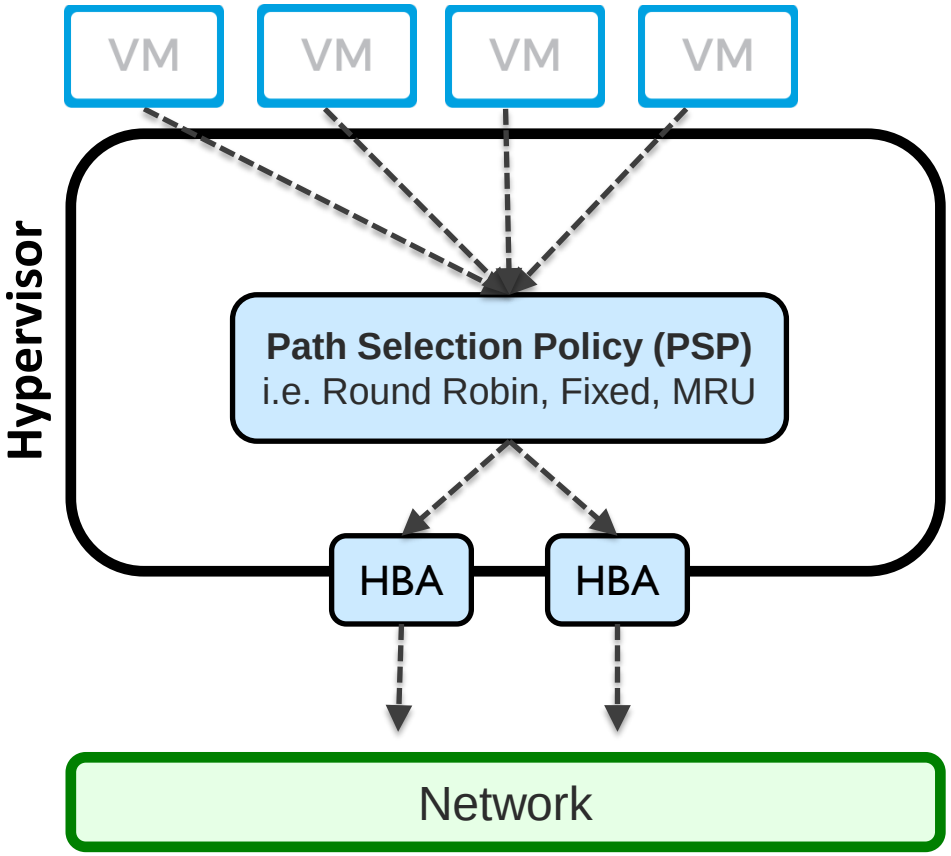


ESX 5.5 Native Driver Model

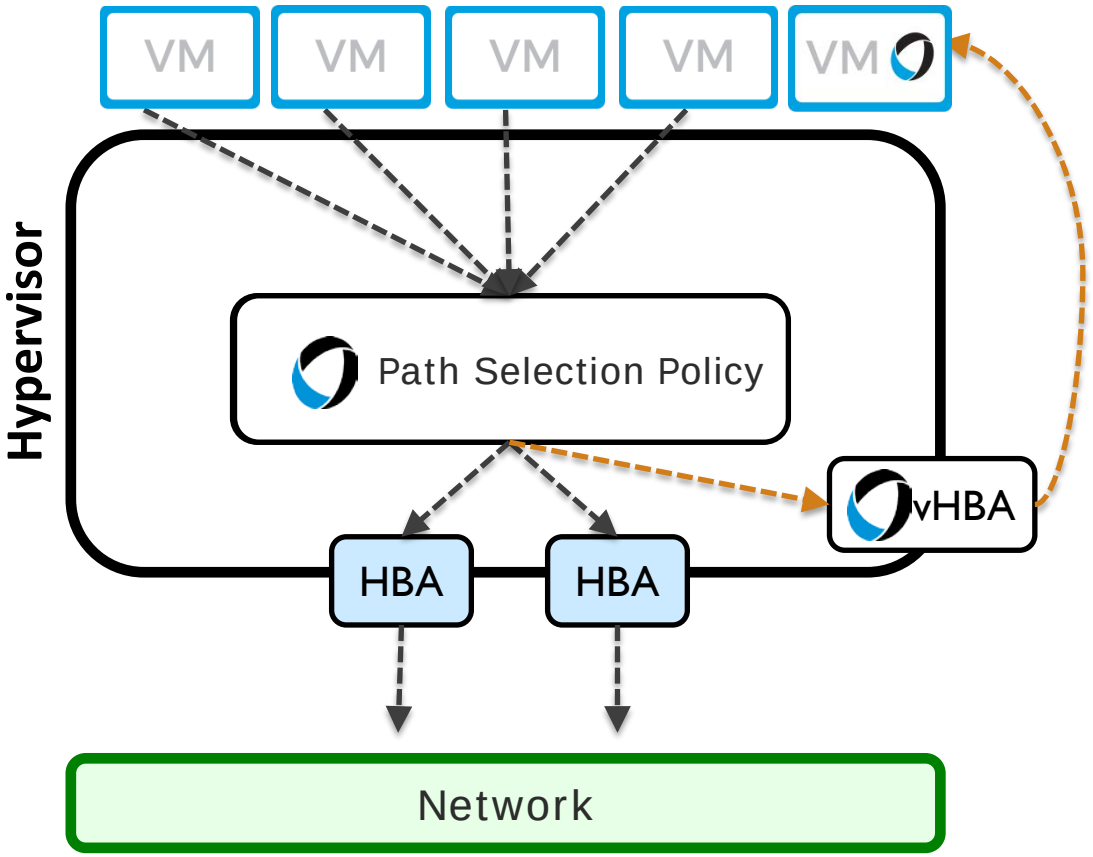
Pluggable Storage Architecture (PSA)



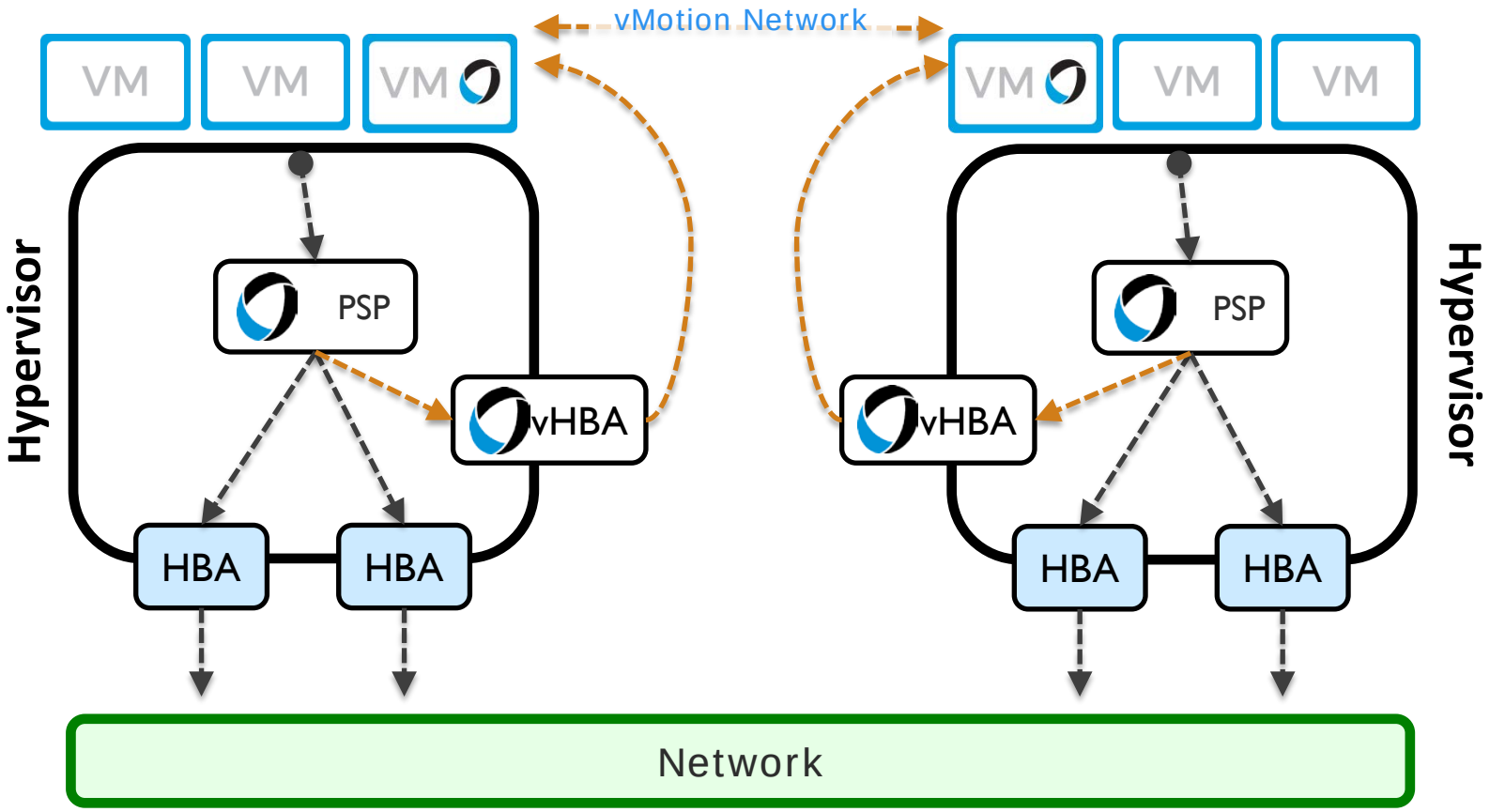
Block-based storage path



Infinio PSA Intercept



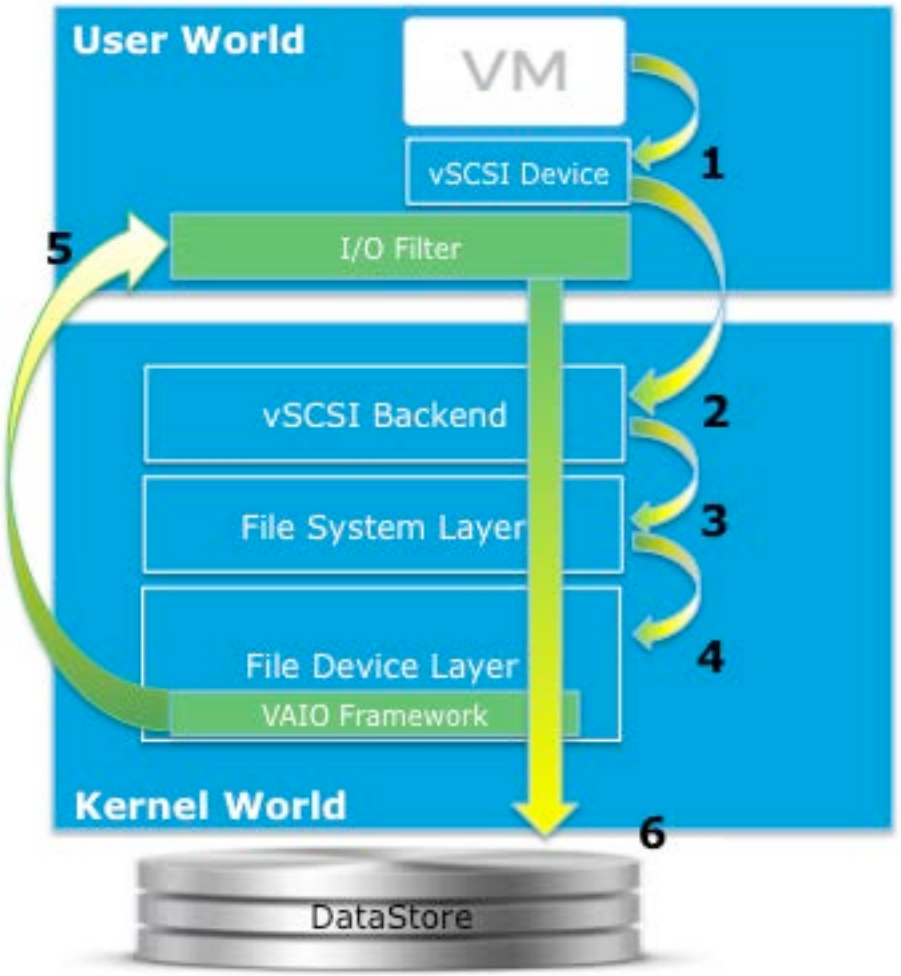
Infinio's 2.0 PSA Intercept



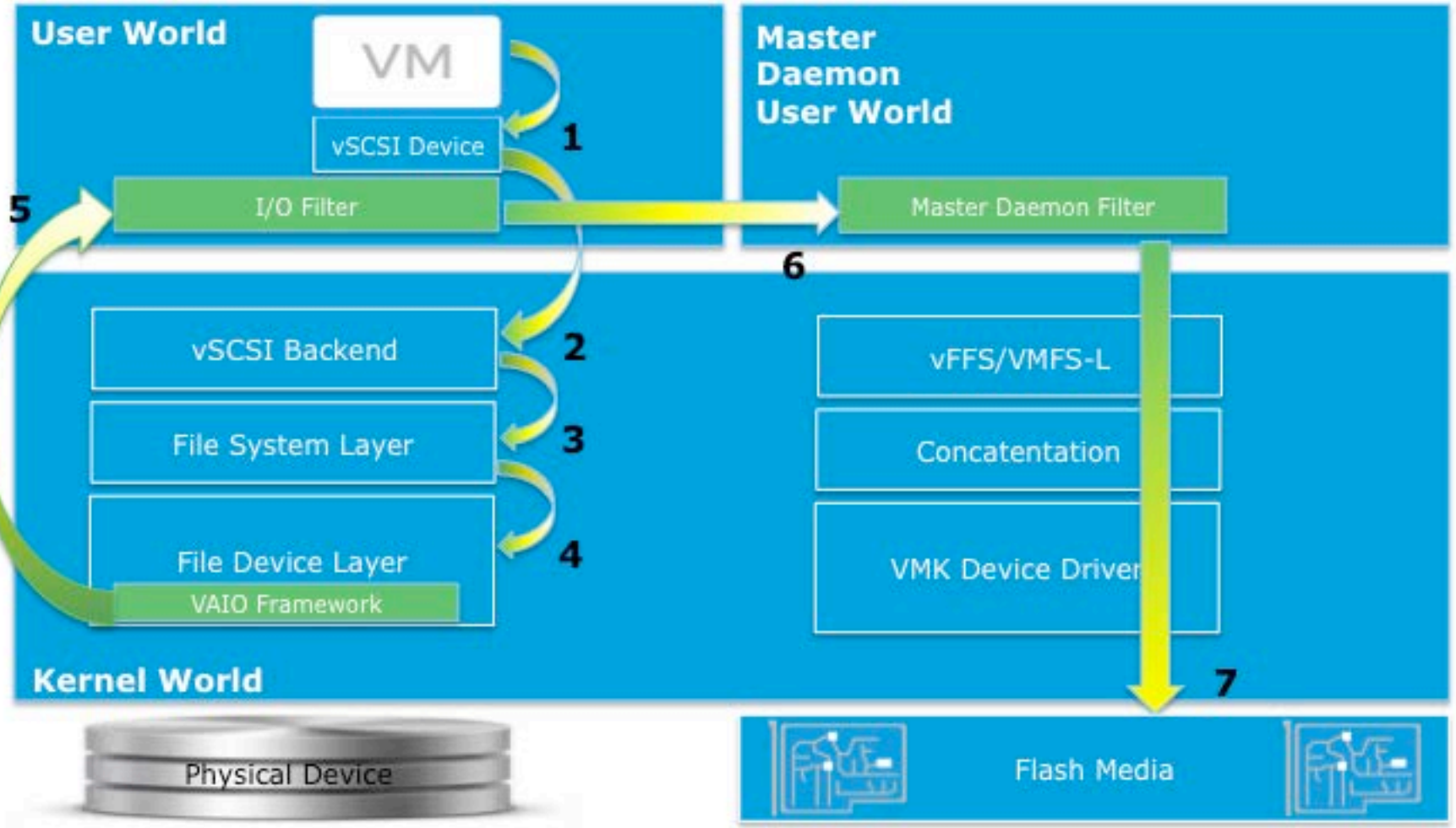
Lessons: PSA

- Harder to implement
  - Less Context - intercepting SCSI commands
  - Efficient; runs on the VMkernel Userworld thread
  - Care must be taken on potential collisions
    - Out of band operations such as VAAI
    - Multi-host VMFS meta-data blocks and protocols
    - Shared Writer VMDKs
  - Constrained, kernel-mode environment
- Infinio specific lessons
  - Good results, better product with much lower latency
    - No context switch on cache miss or writes
  - Getting VAAI, VMFS metadata handling correct added significantly to development time

VAIO Architecture



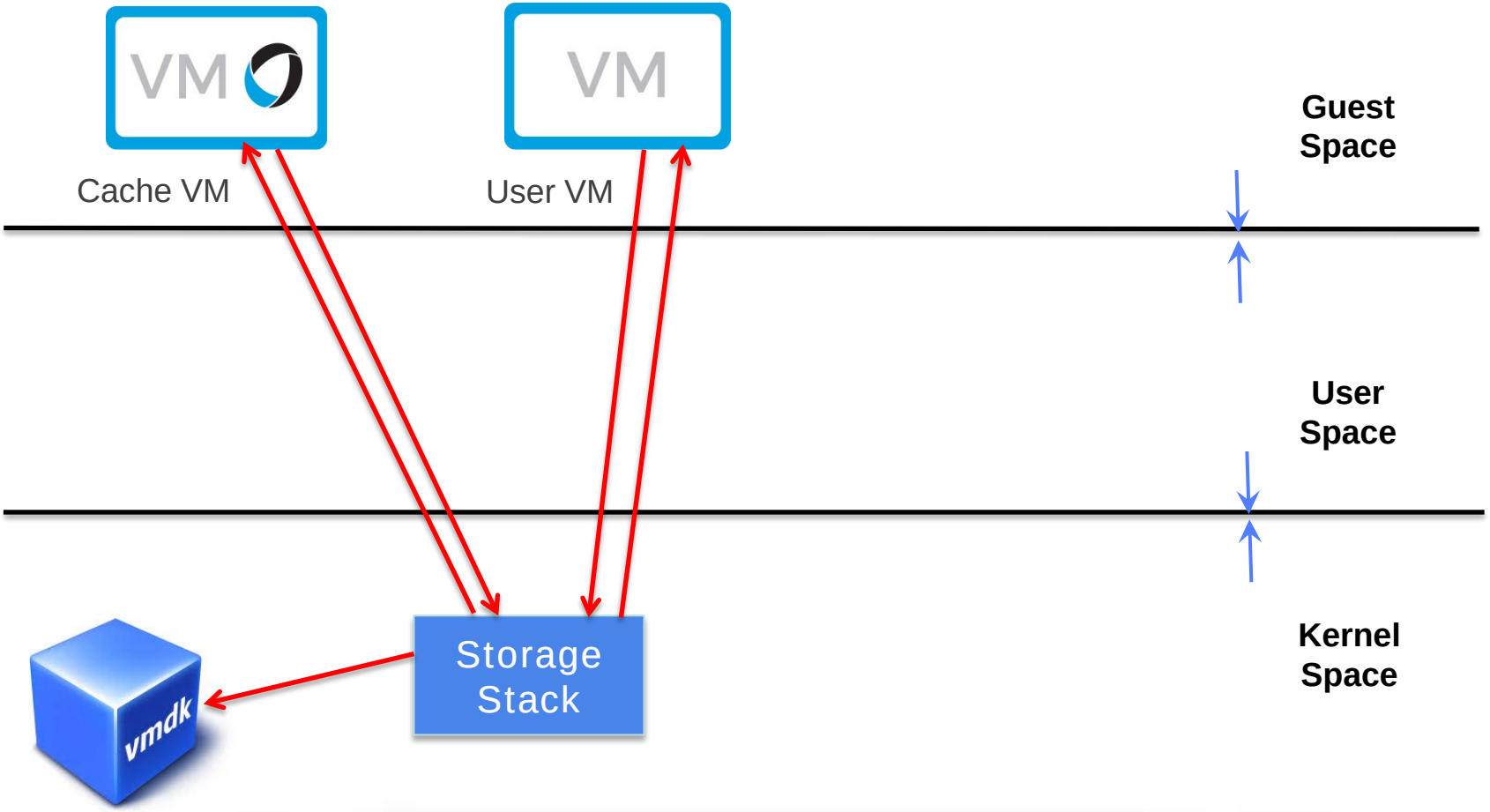
VAIO Architecture



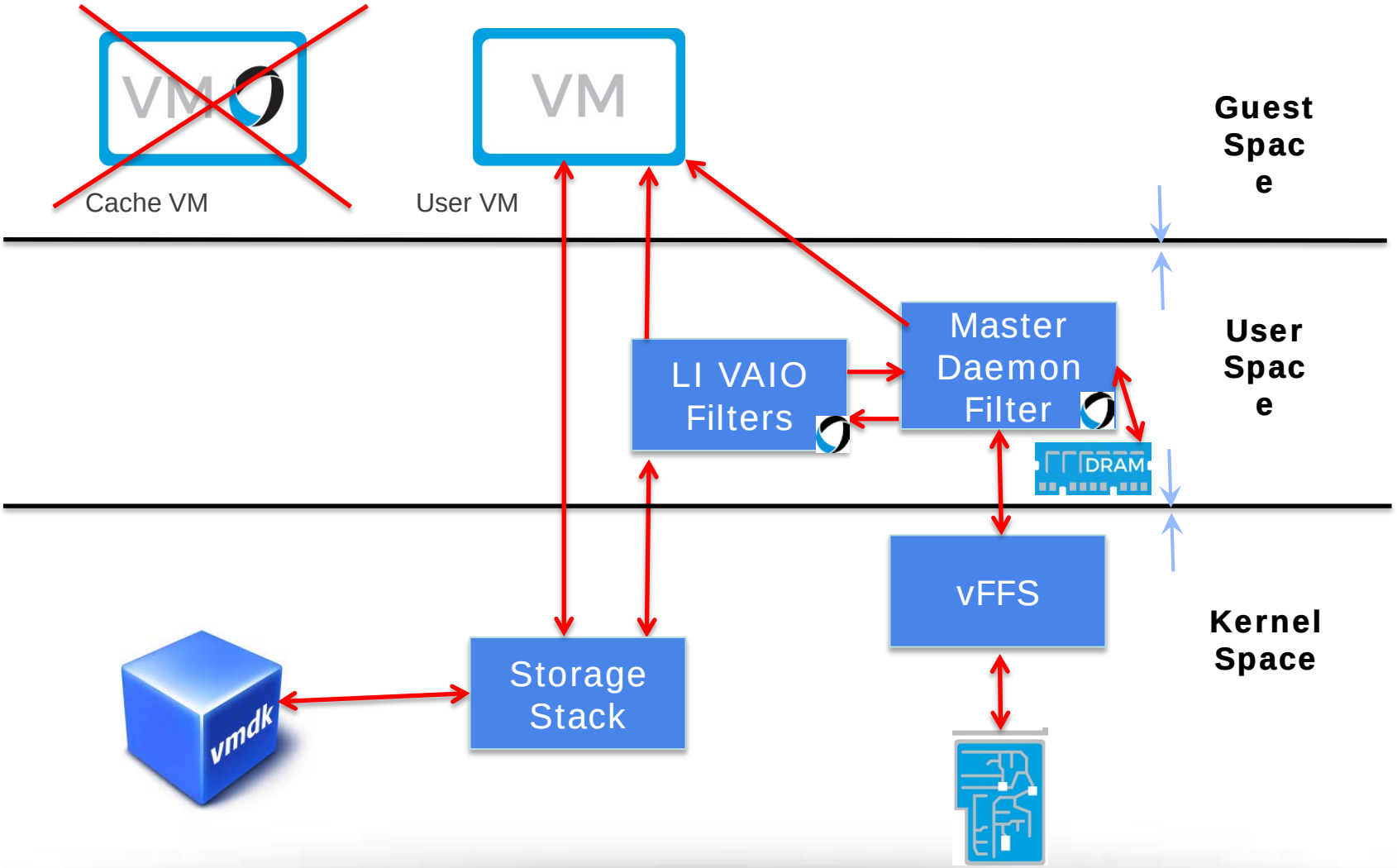
VAIO

- Library Instance Filter
  - Runs in User Space; Instance per VMX Userworld
  - Filter per VMDK
  - Upcall above storage stack, below VM/ESX File System
  - Event call backs – Snapshots, vMotion
  - C Language
- Master Daemon Filter
  - Peer to LI Filter, one per volume/datastore
  - Intended for shared resources spanning LI Filters
  - Separate Cartel/Userworld
  - IP Sockets and limited shared memory
  - vFFS

Infinio's Cache VM



Infinio's VAIO-based architecture



Lessons: VAIO

- Library instance vs. Master Daemon
  - Context Switch
  - Shared Memory vs. Data copy
  - Threading model
  - Resource bookkeeping
- Moderate constraints, improved robustness
- Infinio Learnings:
  - IO Stack intercept simplifies VAAI/meta-data handling
  - Events simplify vMotion/Warm vmotion
  - Intercept overhead is better than VSA, close to PSA
  - Cache hit times benefit from VSA elimination

Summary/Conclusions

- Per I/O latency & throughput, scale all matter
- VM/VSA
  - Richest environment, most overhead
- PSA/Driver
  - Most constrained environment, least overhead
- VAIO
  - Strong hybrid – less constraints, low overhead
  - VM granularity
  - Robust