DPDK Summit 2014
DPDK in a Virtual World

Bhavesh Davda (Sr. Staff Engineer, CTO Office, VMware)

Rashmin Patel (DPDK Virtualization Engineer, Intel)

Agenda

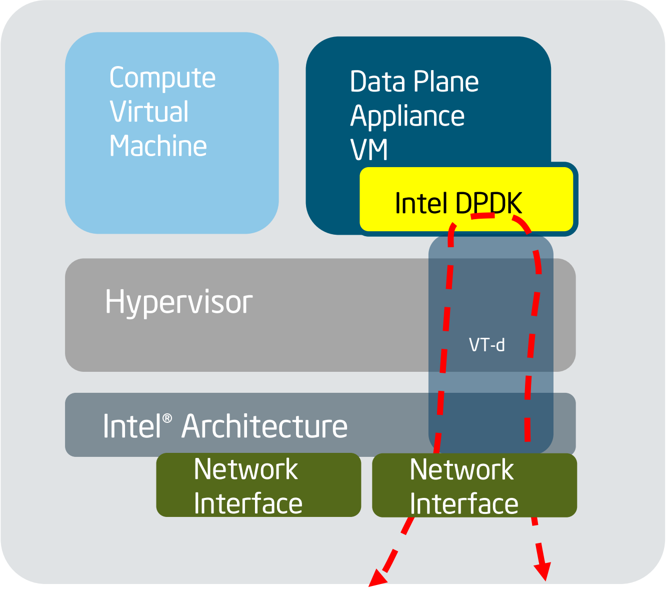
- Data Plane Virtualization Trends
- DPDK Virtualization Support
- VMware ESXi : Key Features and Performance

Agenda

- Data Plane Virtualization Trends
- DPDK Virtualization Support
- VMware ESXi : Key Features and Performance

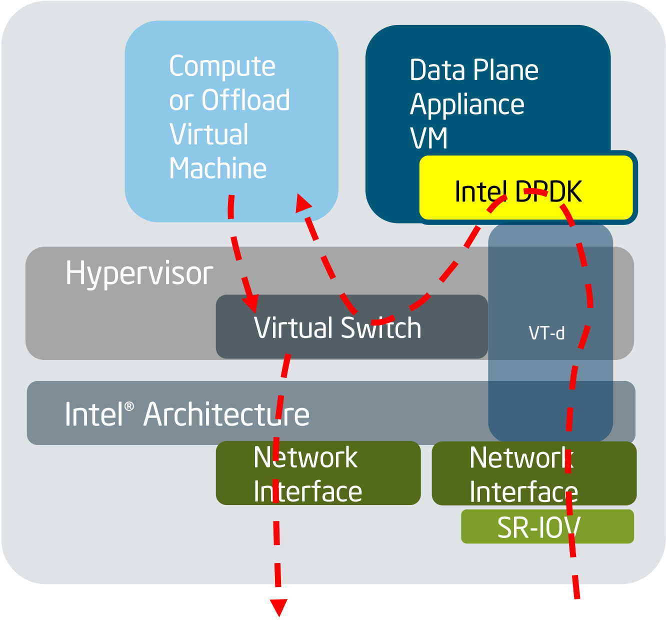
Data Plane Virtualization Trends

Virtualization for Directed I/O Packets are routed to Virtual Machine using DirectPath I/O. Limited flexibility but native performance



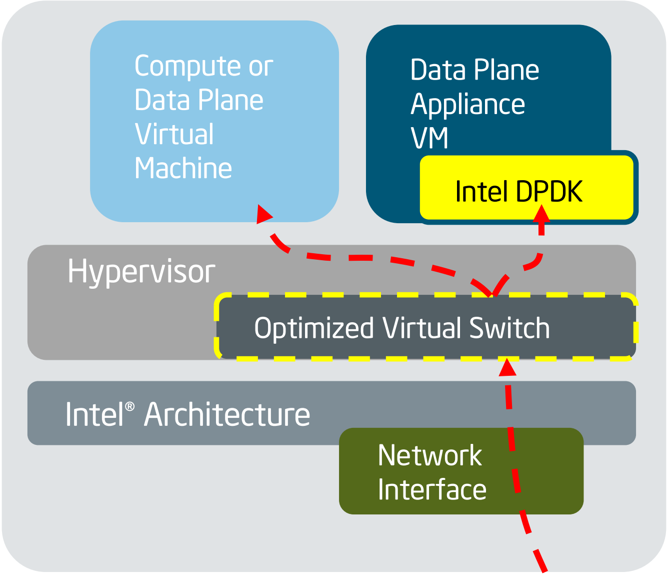
Standalone appliance integration, Firewall, WAN acceleration, Traffic Shaping

Hybrid Switching Solution, combining vSwitch support with direct assignment of SR-IOV Virtual Function



Service Chaining, Unified Threat Management, Intrusion Detection / Prevention

Optimized Virtual Switching solution, combining flexibility with performance. Support for live migration and data plane performance.



Increasing flexibility through high performance soft switching supporting both communications and compute workloads

Agenda

- Data Plane Virtualization Trends
- DPDK Virtualization Support
- VMware ESXi : Key Feature and Performance

DPDK Virtualization Support

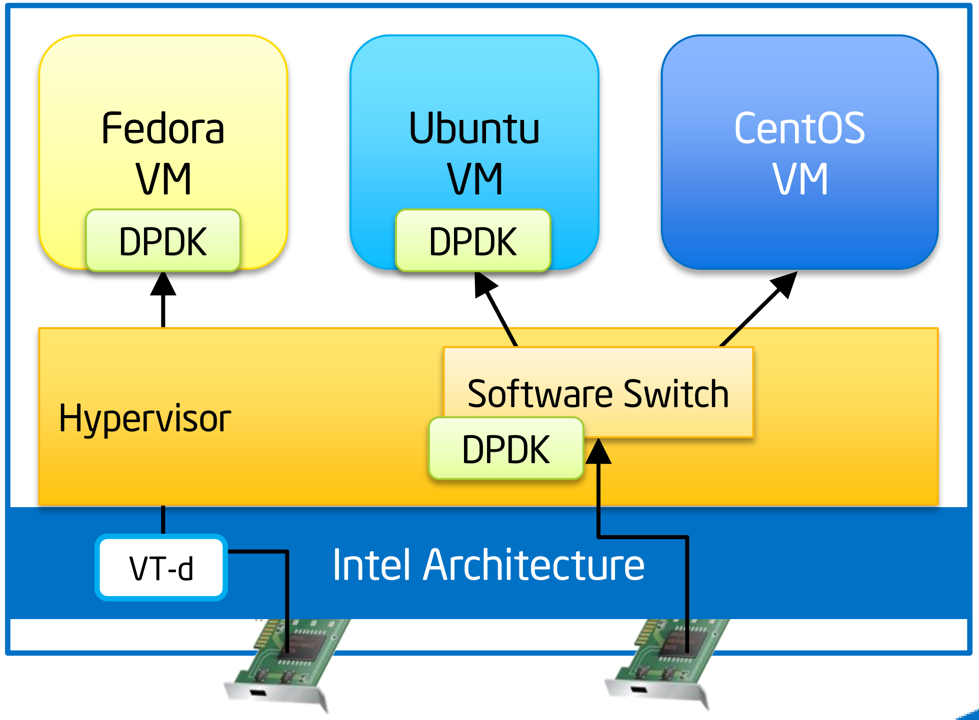
Virtual Machine

- Niantic 82599 Virtual Function PMD
- Fortville Virtual Function PMD
- E1000 Emulated Device (Intel® 82540 Ethernet Controller) PMD
- Virtio Para-virtual Device (Qumranet Device) PMD
- IVSHMEM Shared Memory Interface
- Virtqueue-GrantTable Interface in Xen DomU
- E1000 Emulated Device (Intel® 82545EM Ethernet Controller) PMD
- E1000E Emulated Device (Intel® 82574 Ethernet Controller) PMD
- Vmxnet3 Para-virtual Device PMD

VMware ESXi

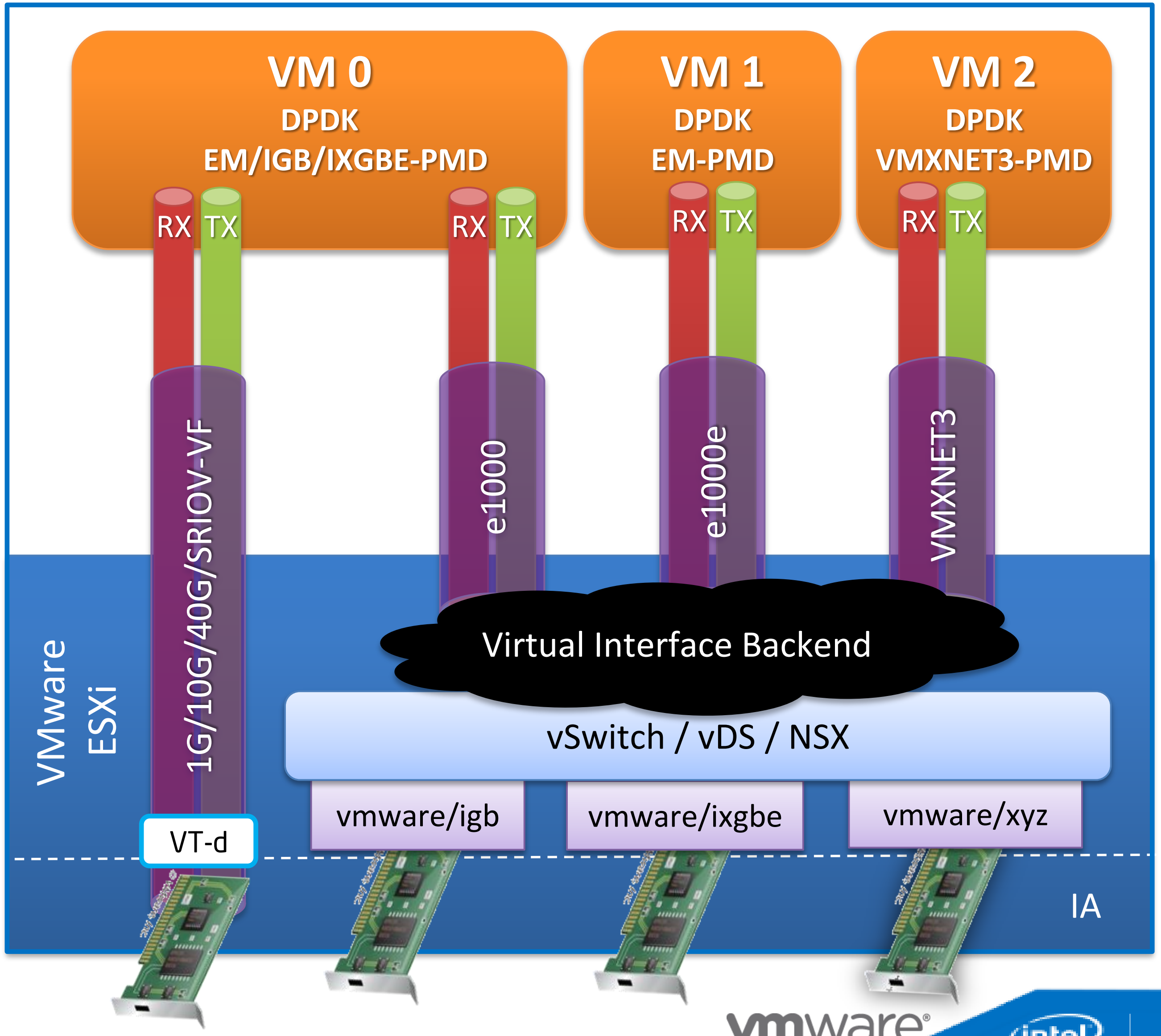
Hypervisor/Host

- Niantic 82599 Physical Function Driver
- Fortville Physical Function Driver
- Userspace-Vhost Backend support
- IVSHMEM Backend support

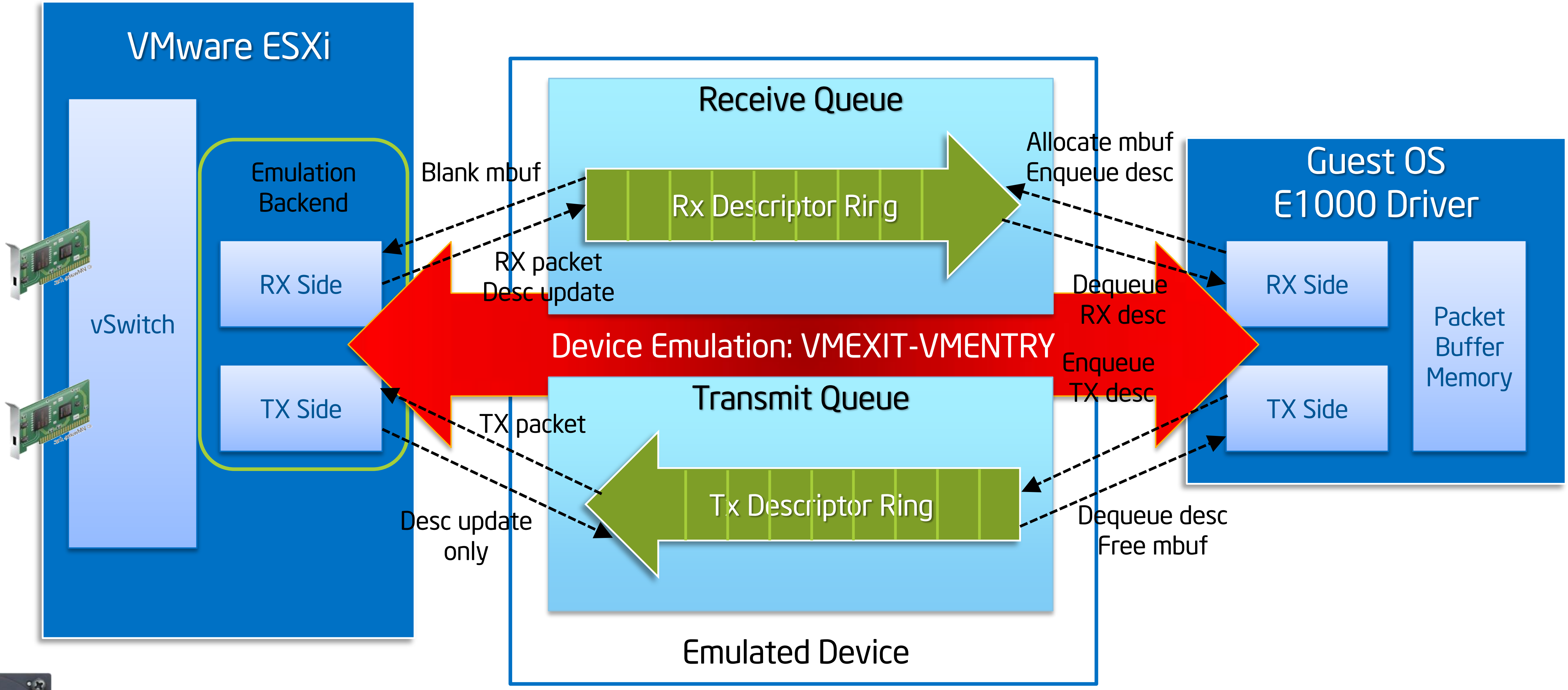


VMware ESXi Virtual Interfaces

- vSS/vDS/NSX – A software switch
  - Emulated devices
    - E1000 (Intel® 82545EM)
    - E1000E (Intel® 82574)
    - VLance (AMD PCnet32)
  - Para-virtual devices
    - VMXNET interface
    - VMXNET2 interface
    - VMXNET3 interface
- Direct assignment – vSwitch bypass
  - Intel® VT-d / IOMMU required
  - DirectPath I/O Full PCI function
  - SR-IOV PCI VF assignment
  - Incompatible with virtualization features: vMotion, HA, FT, snapshots, network virtualization overlays (VXLAN/STT/Geneve)

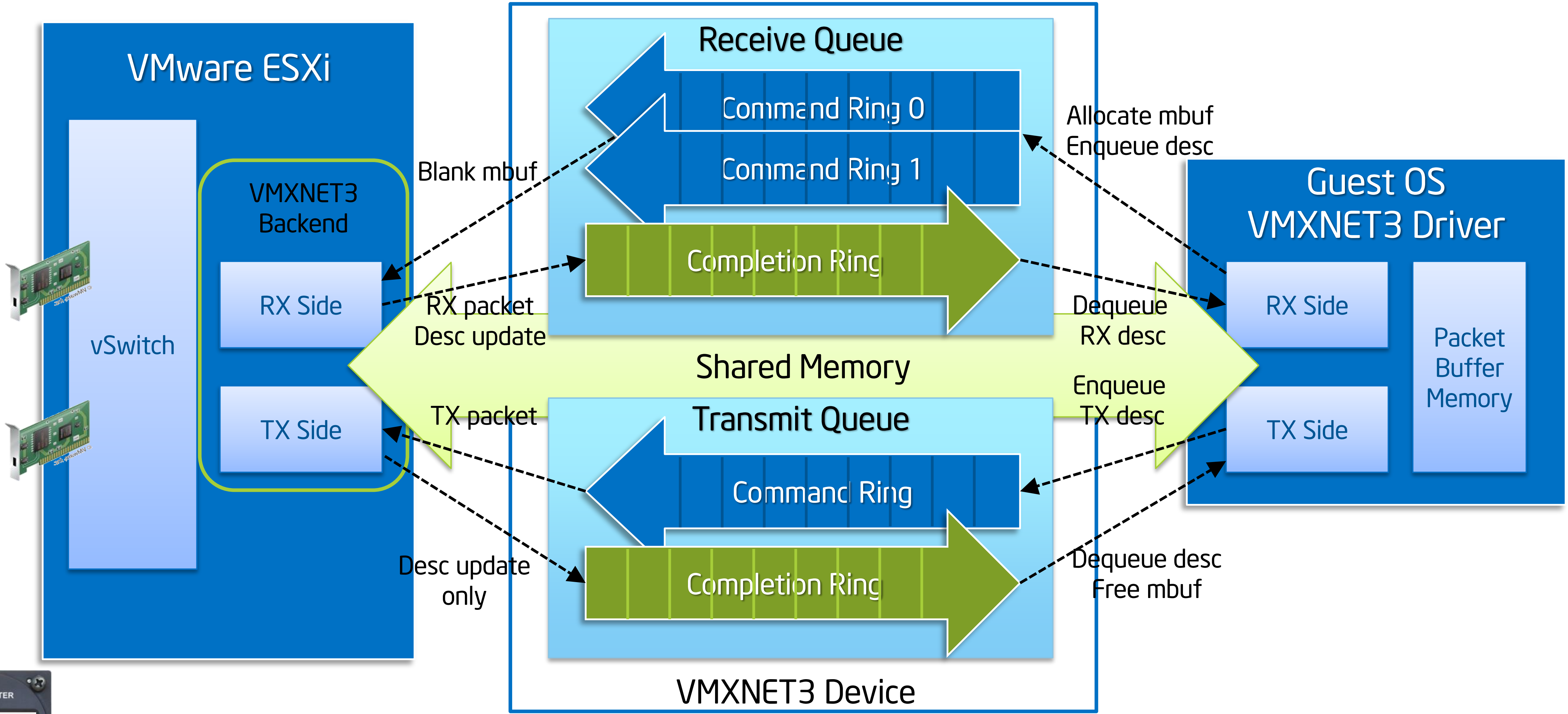


VMware ESXi Emulated Device (E1000)



Every emulated PCI MMIO access from a VM results in VM exit

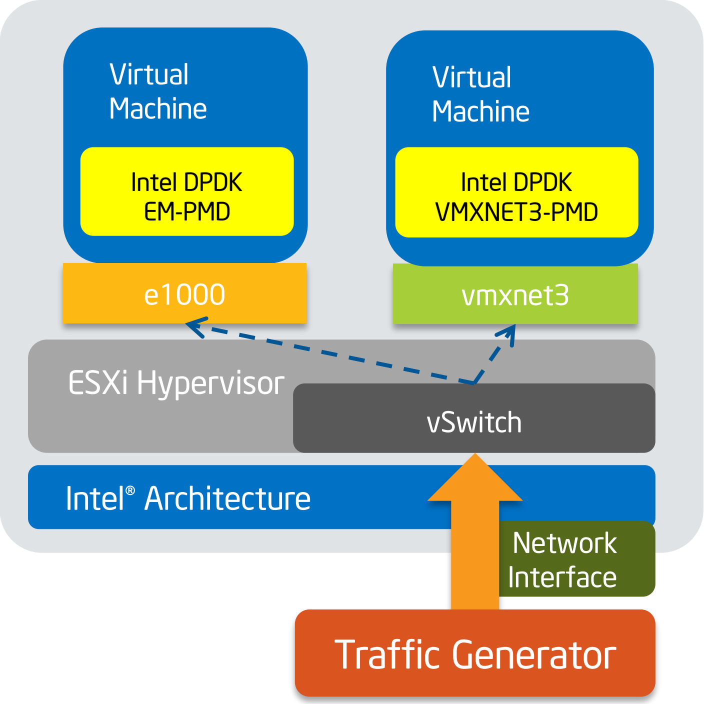
VMware ESXi Paravirtual Device (VMXNET3)



Reduced VM exits by moving Queue Control Registers in Shared Memory and passing hints from ESXi to Guest OS

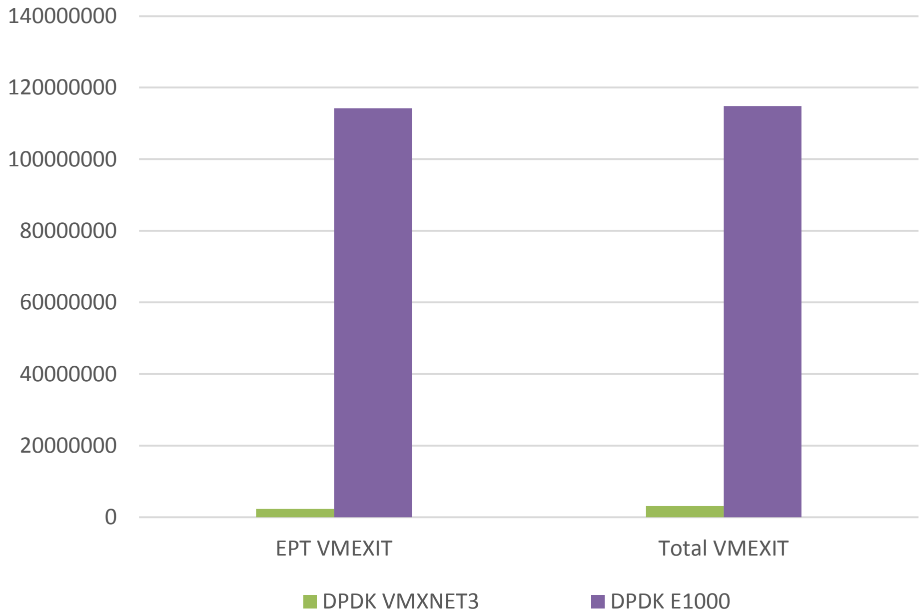
ESXi 5.5 - VMXNET3 vs E1000

- Optimized Rx/Tx queues handling in VMXNET3 controlled through shared memory region – reduced VM exits compared to E1000's inefficient MMIO emulation
- Multiqueue infrastructure of VMXNET3 with RSS capability enhance the performance with Multicores in a VM



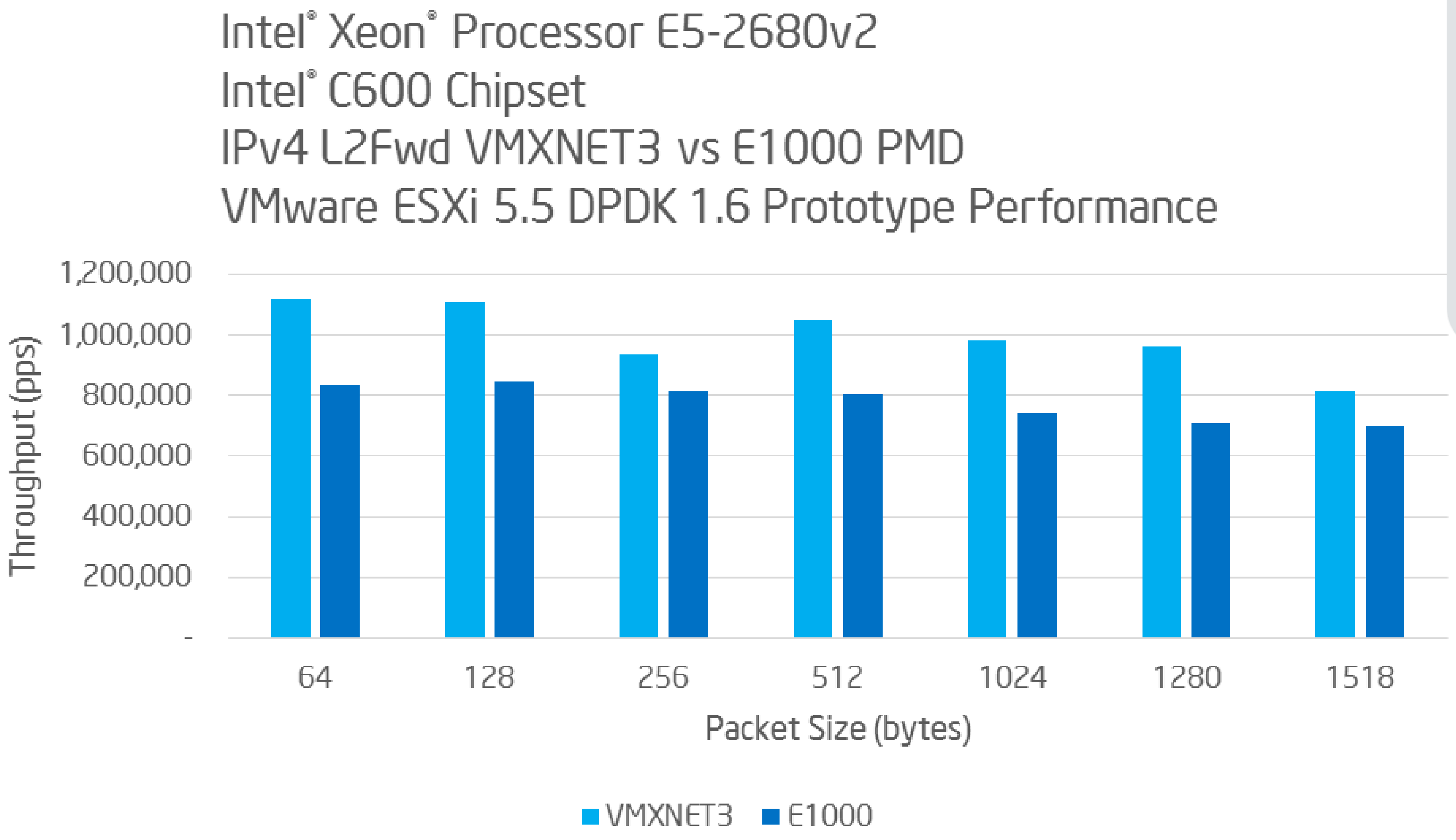
- Average cost for a VM exit/VM entry sequence includes ~600 cycles for VMCALL instruction. Average cost for EPT violation ~1000 cycles

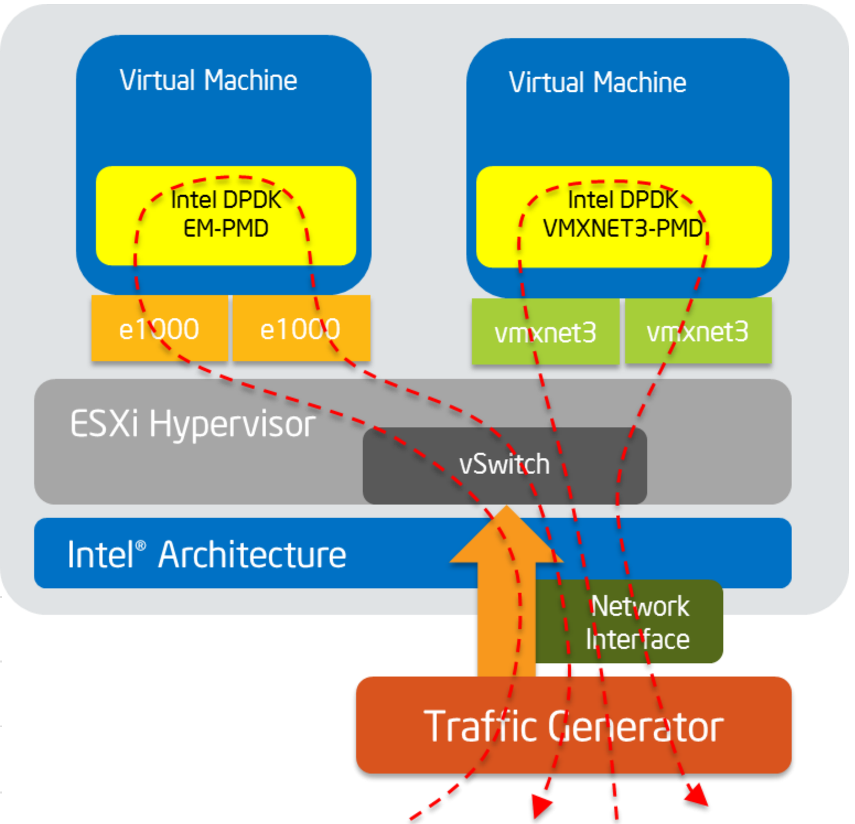
Intel® Xeon® Processor E5-4610 v2
(16M Cache, 2.30 GHz)
VM exit/ VM entry frequency



97% Reduction of VM exits associated with DPDK based Packet forwarding benchmark

Hypervisor Backend Impact





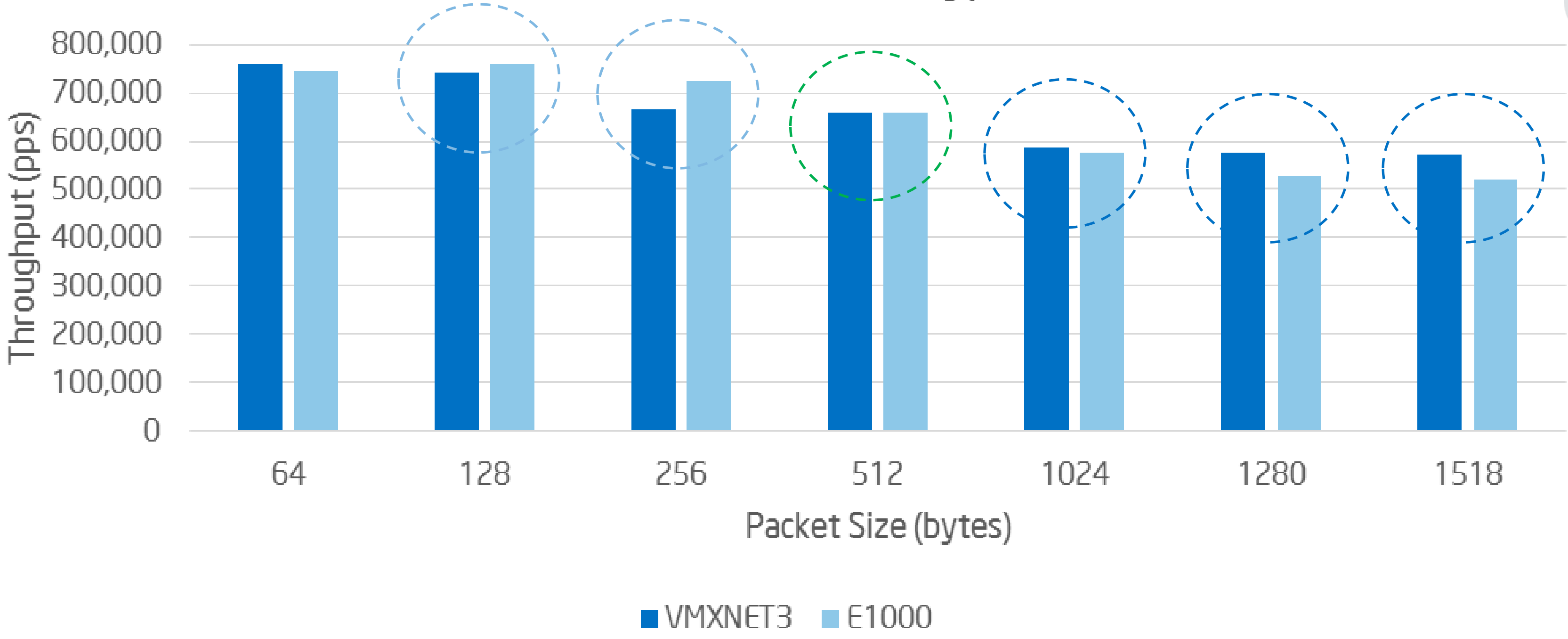
L2Fwd Loopback in VM

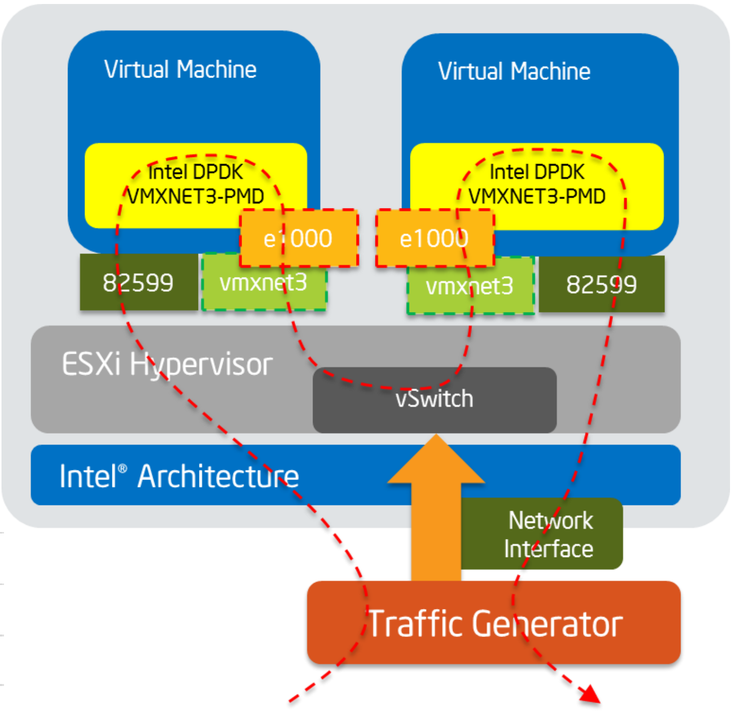
Traffic Flow:
Traffic Gen. -> vSwitch -> VMXNET3 (or E1000) -> VM (DPDK) -> VMXNET3 (or E1000) -> vSwitch -> Traffic Gen.

VM exit reduction doesn't translate to big difference in packet throughput; Hypervisor Backend and Native Networking stack needs optimizations

Device Model Research

Intel® Xeon® Processor E5-2680v2
Intel® C600 Chipset
IPv4 L2Fwd VMXNET3 vs e1000 (VM-to-VM)
VMware ESXi 5.5 DPDK 1.6 Prototype Performance





L2Fwd Chained VM-to-VM

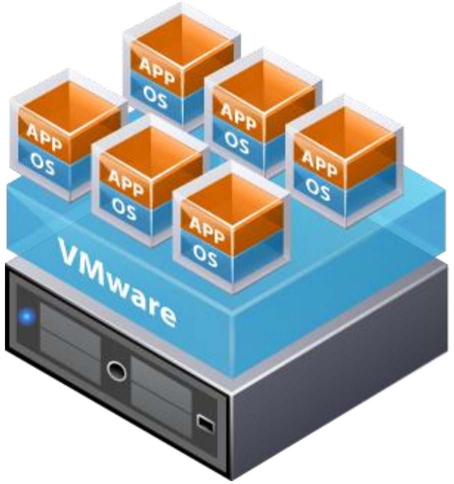
Traffic Flow:
Traffic Gen. -> 10G (VT-d) -> VM1 (DPDK)
-> VMXNET3 (or E1000) -> vSwitch
-> VMXNET3 (or E1000) -> VM2 (DPDK)
-> 10G (VT-d) -> Traffic Gen.

Important to understand for designing changes in device model for future ESXi releases

Agenda

- Data Plane Virtualization Trends
- DPDK Virtualization Support
- VMware ESXi : Key Features and Performance

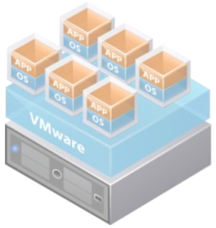
Key Properties of Virtual Machines



Partitioning

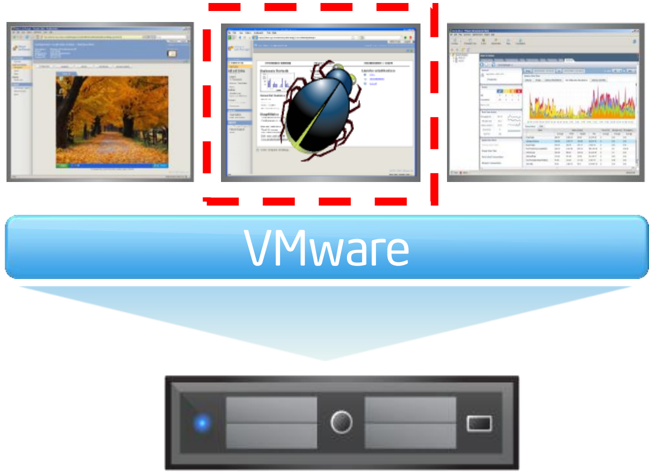
- Run multiple operating systems on one physical machine
- Divide system resources between virtual machines

Key Properties of Virtual Machines : Continued



Partitioning

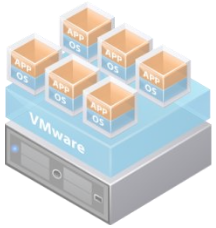
- Run multiple operating systems on one physical machine
- Divide system resources between virtual machines



Isolation

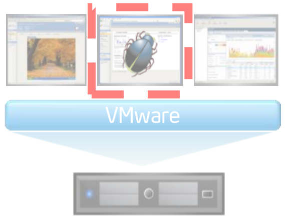
- Fault and security isolation at the hardware level
- Advanced resource controls preserve performance

Key Properties of Virtual Machines : Continued



Partitioning

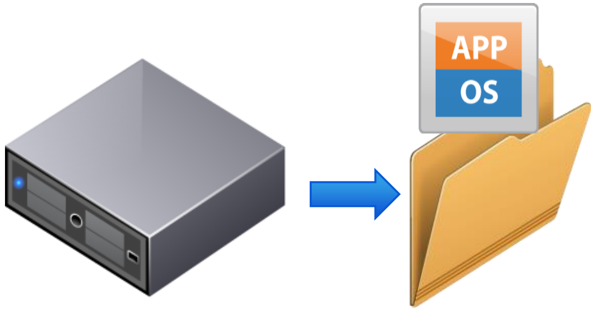
- Run multiple operating systems on one physical machine
- Divide system resources between virtual machines



Isolation

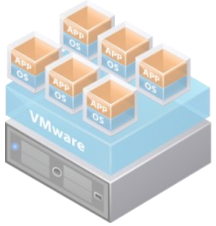
- Fault and security isolation at the hardware level
- Advanced resource controls preserve performance

Encapsulation



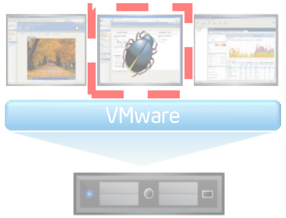
- Entire state of the virtual machine can be saved to files
- Move and copy virtual machines as easily as moving and copying files

Key Properties of Virtual Machines : Continued



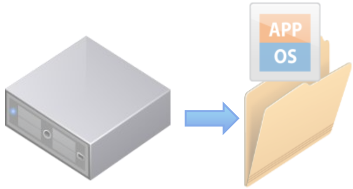
Partitioning

- Run multiple operating systems on one physical machine
- Divide system resources between virtual machines



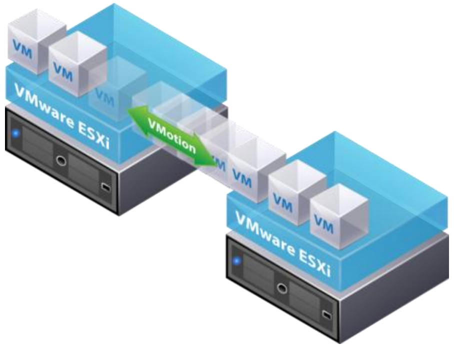
Isolation

- Fault and security isolation at the hardware level
- Advanced resource controls preserve performance



Encapsulation

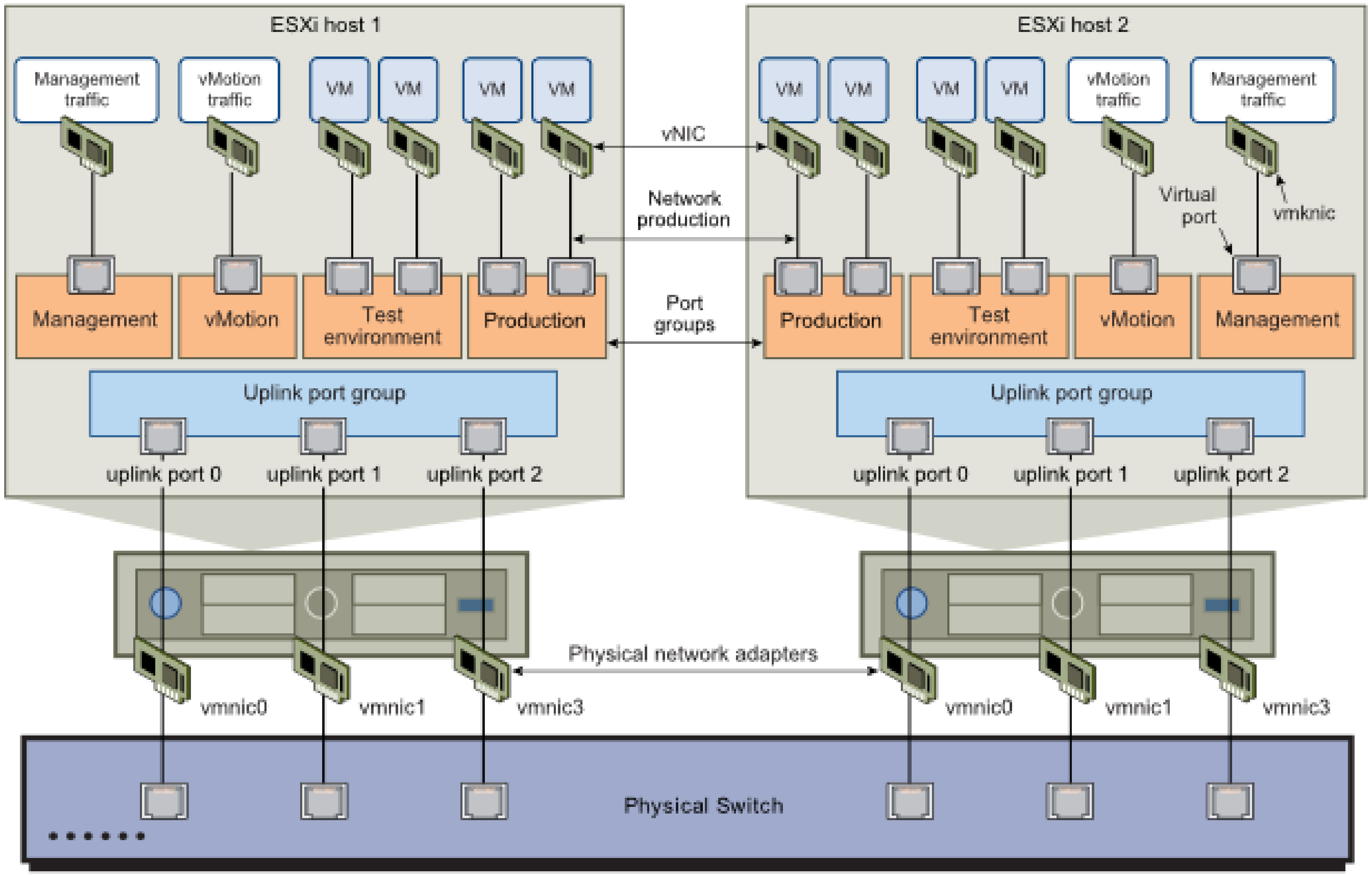
- Entire state of the virtual machine can be saved to files
- Move and copy virtual machines as easily as moving and copying files



Hardware Independence

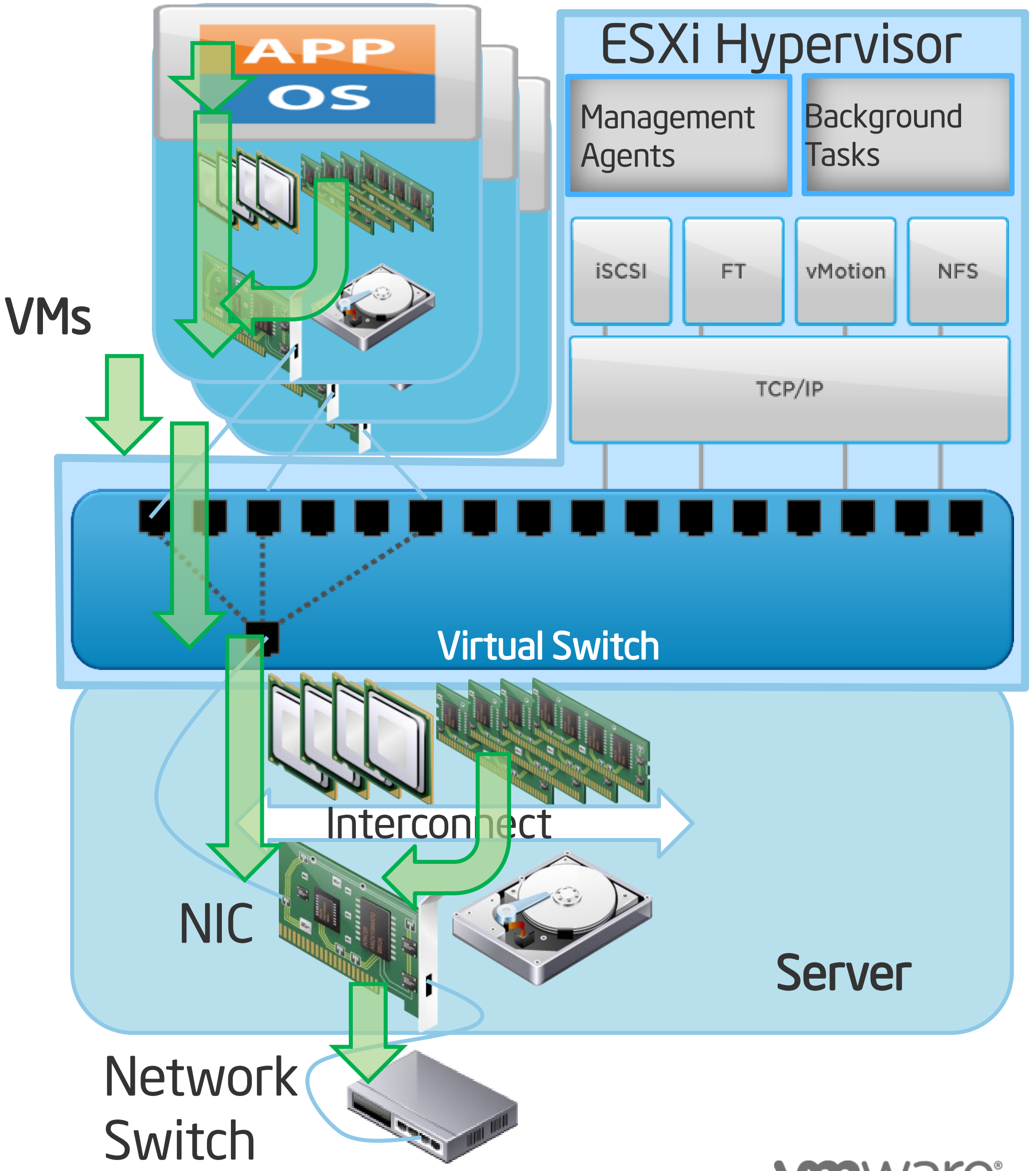
- Provision or migrate any virtual machine to any similar or different physical server

ESXi Networking Architecture Overview

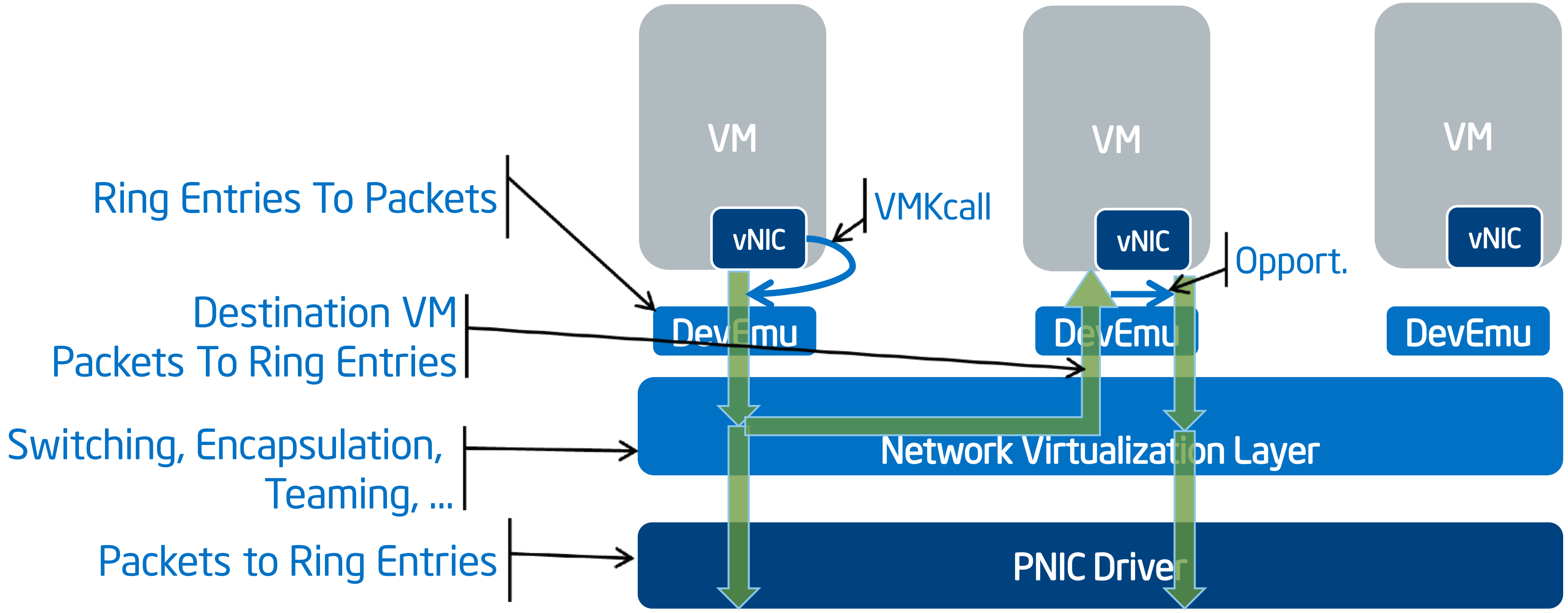


ESXi Networking Datapath Overview

- Message copy from application to GOS (kernel)
- GOS (network stack) + vNIC driver queues packet for vNIC
- VM exit to VMM/Hypervisor
- vNIC implementation emulates DMA from VM, sends to vSwitch
- vSwitch queues packet for pNIC
- pNIC DMAs packet and transmits on the wire

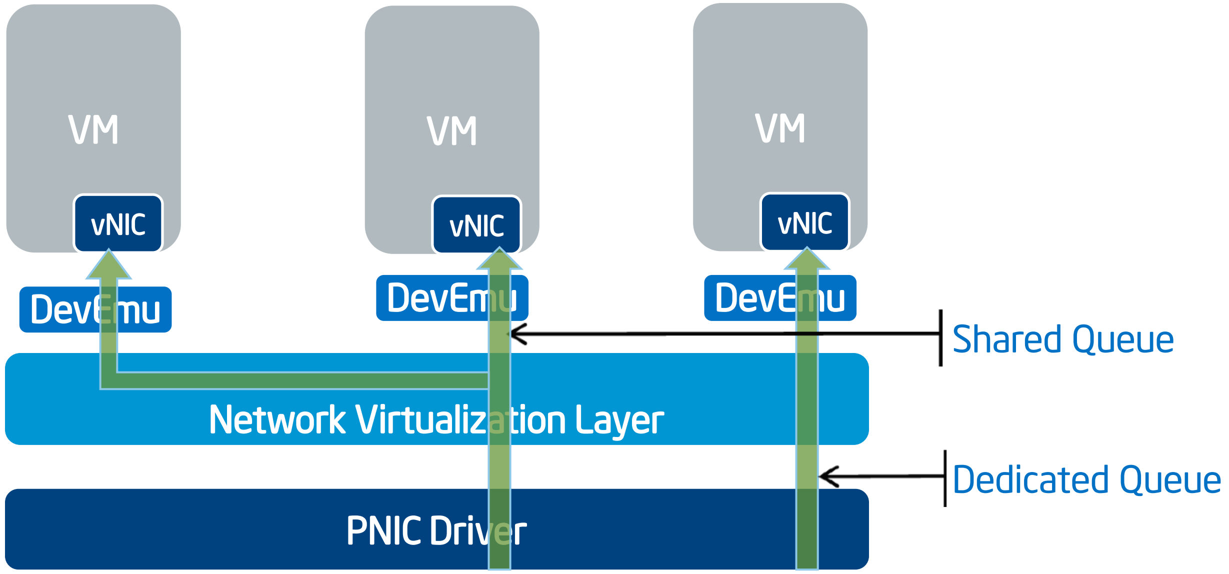


Transmit Processing for a VM



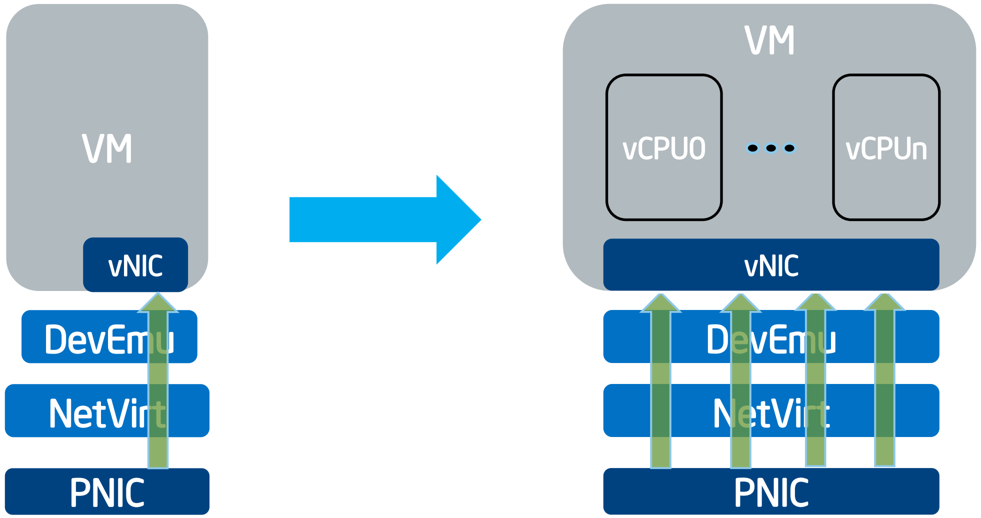
- One transmit thread per VM, executing all parts of the stack
  - Transmit thread can also execute receive path for destination VM
- Activation of transmit thread: Two mechanisms
  - Immediate, forcible activation by VM (low delay, expensive)
  - Opportunistic activation by other threads or when VM halts (potentially higher delay, cheap)

Receive Processing For a VM



- One thread per device
- NetQueue enabled devices: one thread per NetQueue
  - Each NetQueue processes traffic for one or more MAC addresses (vNICs)
  - NetQueue → vNIC mapping determined by ***unicast*** throughput and FCFS.
- vNICs can share queues
  - due to low throughput, too many vNICs or Queue type mismatch (LRO Queue vs. non-LRO VNIC)

Improve receive throughput to a single VM



Single thread for receives can become bottleneck at high packet rates (> 1 Million PPS or > 15Gbps)

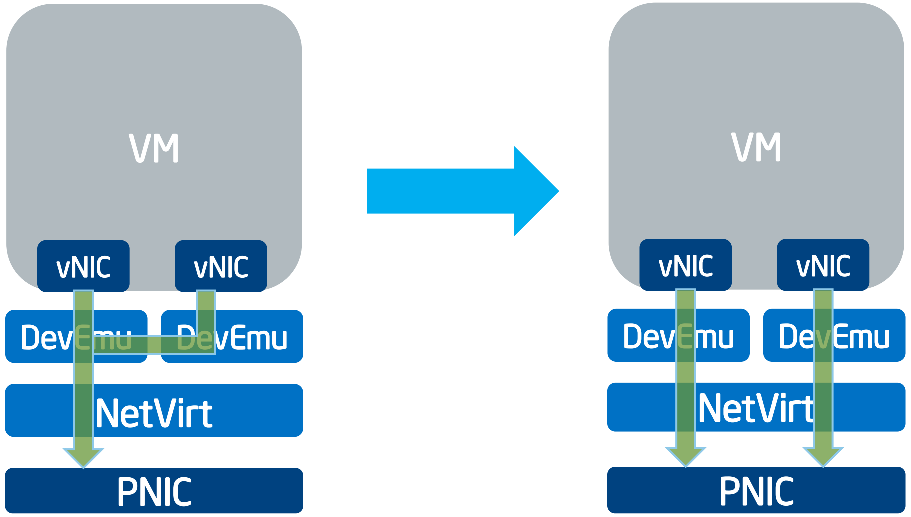
Use VMXNET3 virtual device, Enable RSS inside Guest

Enable RSS in Physical NICs (only available on some PNICs)

Add ethernetX.pnicFeatures = “4” to vmx file

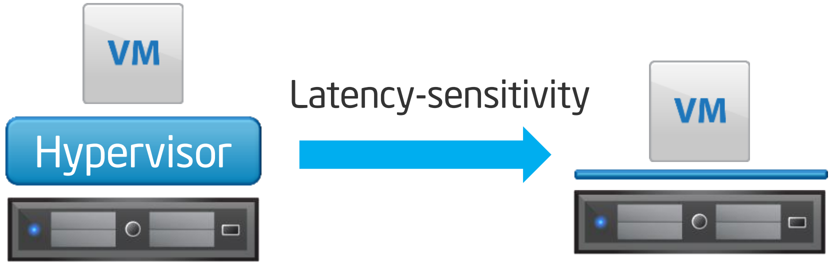
Side effects: Increased CPU Cycles/Byte

Improve transmit throughput with multiple vNICs



- Some applications use multiple vNICs for very high throughput
- Common transmit thread for all vNICs can become bottleneck
- Set ethernetX.ctxPerDev = 1 in vmx file
- Side effects: Increased CPU Cost/Byte

The Latency Sensitivity Feature in vSphere 5.5



Minimize virtualization overhead, near bare-metal performance

New virtual machine property: “Latency sensitivity”

- High => lowest latency
- Medium => low latency

Exclusively assign physical CPUs to virtual CPUs of “Latency Sensitivity = High” VMs

- Physical CPUs not used for scheduling other VMs or ESXi tasks

Idle in Virtual Machine monitor (VMM) when Guest OS is idle

- Lowers latency to wake up the idle Guest OS, compared to idling in ESXi vmkernel

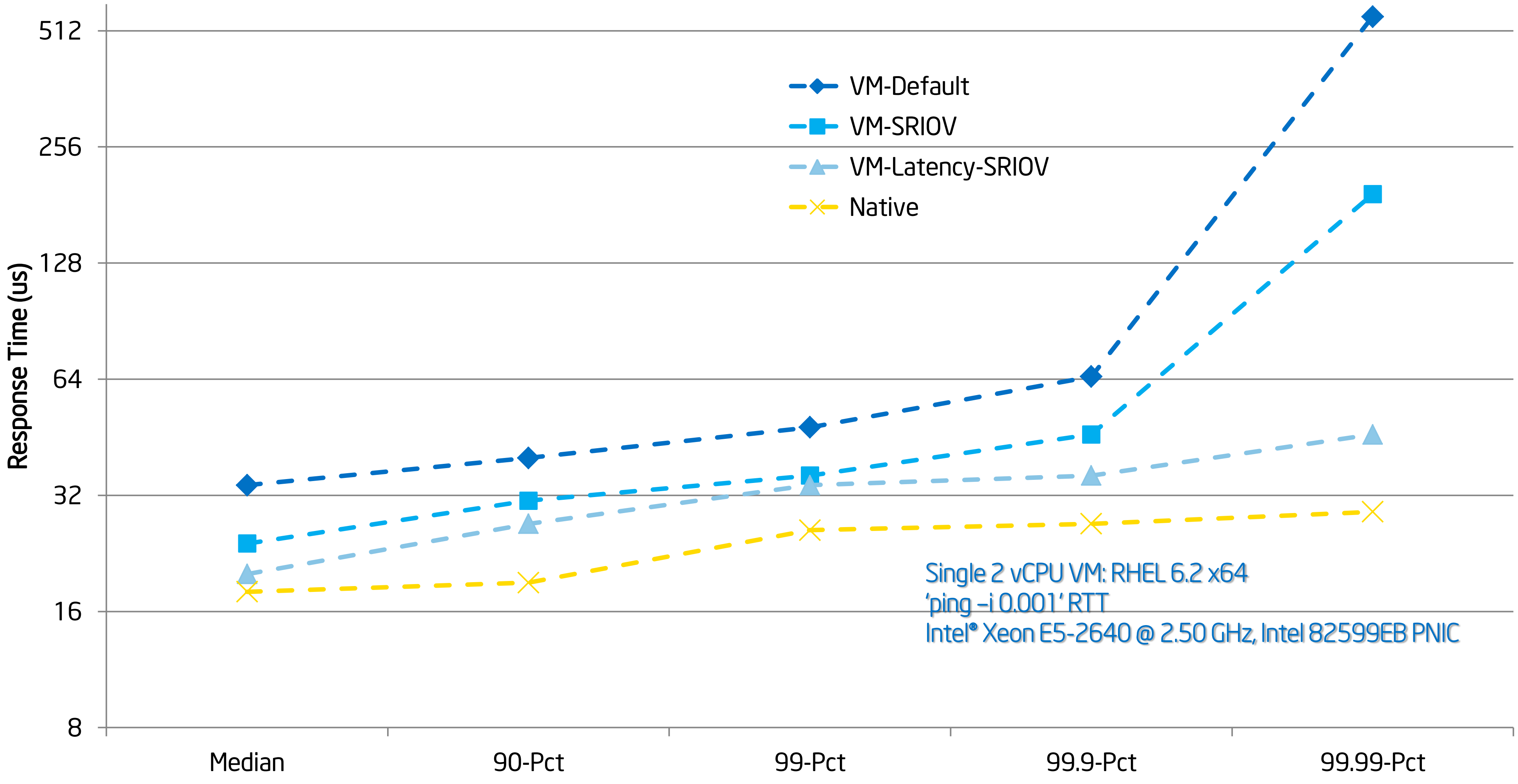
Disable vNIC interrupt coalescing

For DirectPath I/O, optimize interrupt delivery path for lowest latency

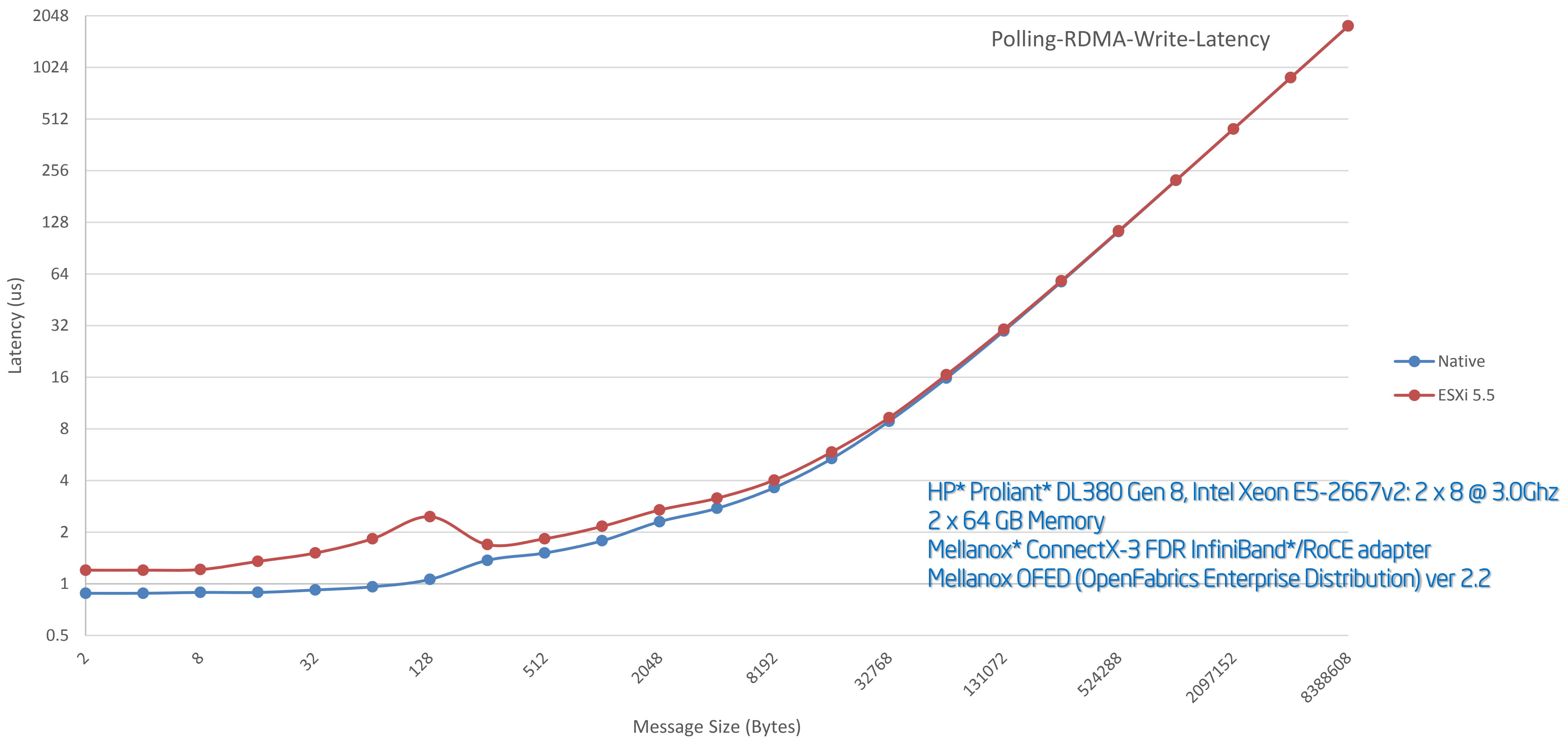
Make ESXi vmkernel more preemptible

- Reduces jitter due to long-running kernel code

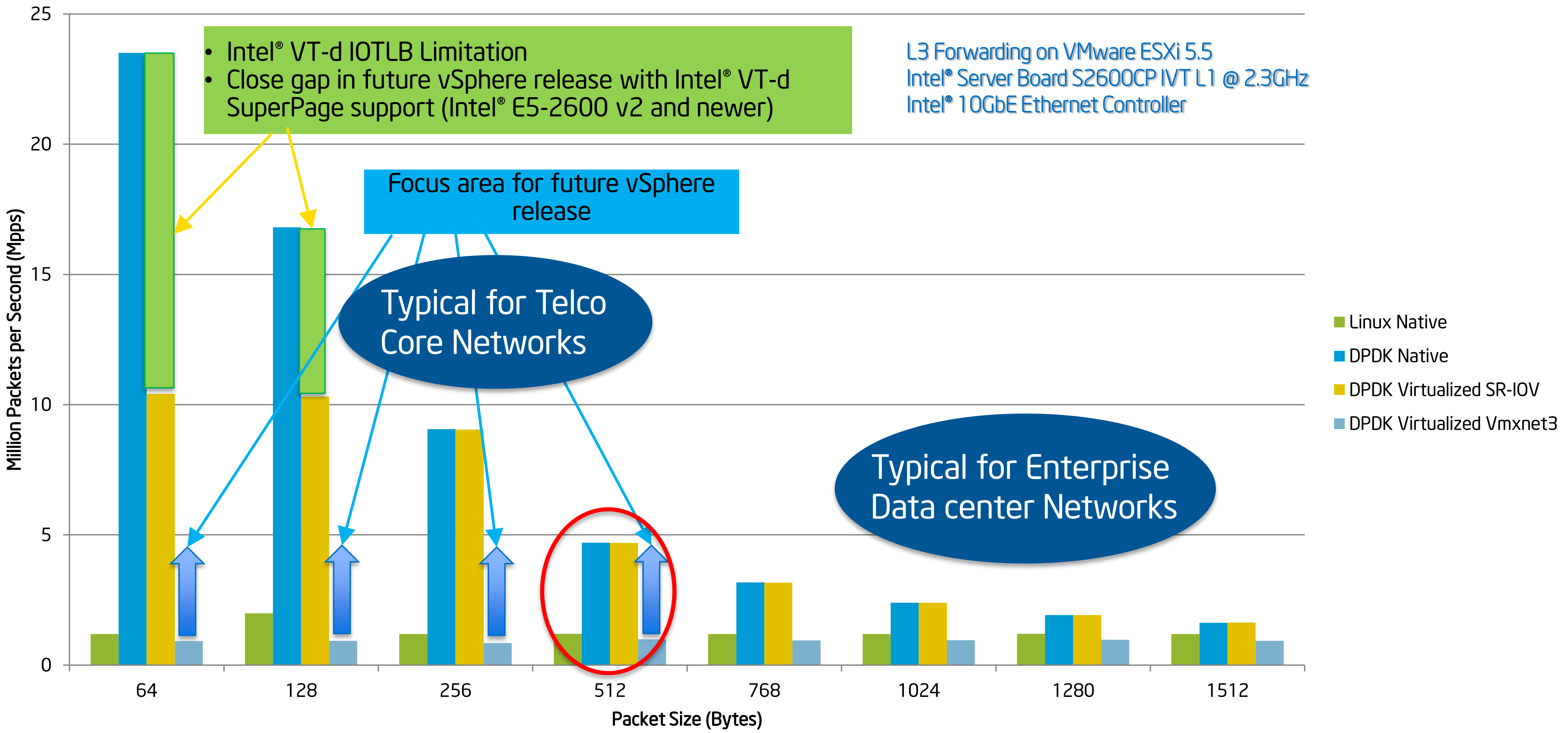
ESXi 5.5 Network Latencies and Jitter



ESXi 5.5 ultra-low latency w/ InfiniBand DirectPath I/O

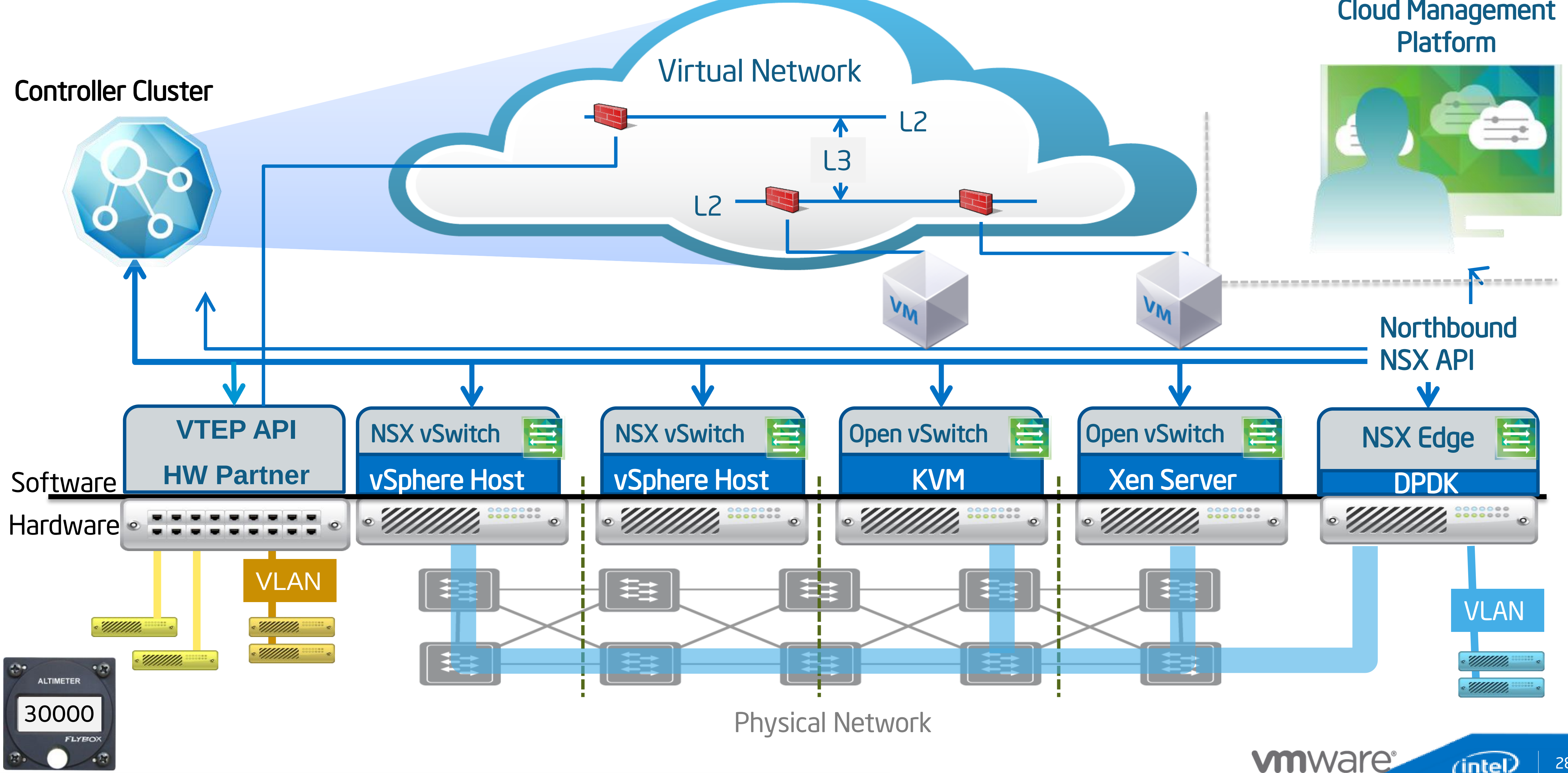


ESXi 5.5 packet rates with Intel® DPDK



Intel® Virtualization Technology (Intel® VT) for Directed I/O (Intel® VT-d)
Intel® Data Plane Development Kit (Intel® DPDK)

NSX – Network Virtualization Platform



Summary

- Virtual Appliances performing Network Functions needs high performance on commodity x86 based server platforms
- Virtualized interfaces supported in DPDK offer multiple options and flexibility for data plane application developers
- ESXi hypervisor supports multiple interfaces, including para-virtual and emulated interfaces while offering best in class virtualization features, as well as direct assigned interfaces via DirectPath I/O and SR-IOV

vSphere ESXi Performance Related References

Best Practices for Performance Tuning of Latency-Sensitive Workloads in vSphere VMs
http://www.vmware.com/files/pdf/techpaper/VMW-Tuning-Latency-Sensitive-Workloads.pdf

Intel® Data Plane Development Kit (Intel® DPDK) with VMware vSphere
http://www.vmware.com/files/pdf/techpaper/intel-dpdk-vsphere-solution-brief.pdf

Deploying Extremely Latency-Sensitive Applications in VMware vSphere 5.5
http://www.vmware.com/files/pdf/techpaper/latency-sensitive-perf-vsphere55.pdf

The CPU Scheduler in VMware vSphere 5.1
https://www.vmware.com/files/pdf/techpaper/VMware-vSphere-CPU-Sched-Perf.pdf

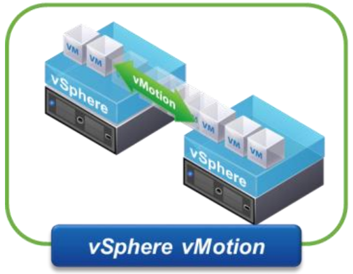
RDMA Performance in Virtual Machines using QDR InfiniBand on VMware vSphere 5
https://labs.vmware.com/academic/publications/ib-researchnote-apr2012

intel®

vmware®

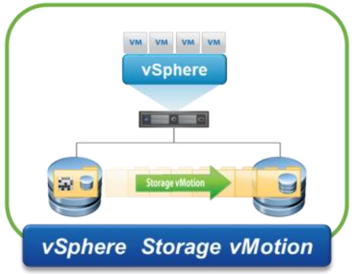
Backup

VMware vSphere Sample Features



vSphere vMotion

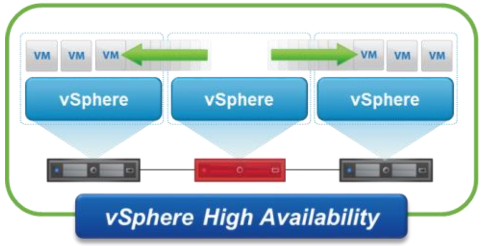
- Migrate VMs between vSphere hosts without application downtime
- Move VMs out of failing or underperforming servers without downtime
- Perform hardware maintenance without scheduling downtime or disrupting agency operations



vSphere Storage vMotion

- Perform live migration of VM disk file across heterogeneous storage array with complete transaction integrity and application availability
- Eliminate application downtime for storage maintenance
- Simplify array refresh/retirement, improve array performance and capacity balancing

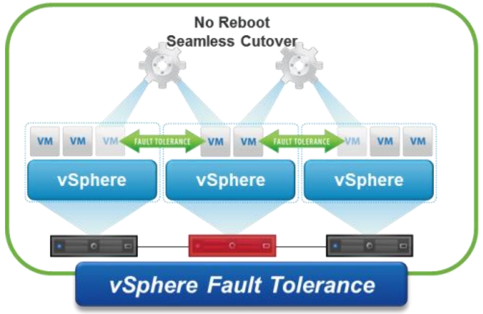
Protect Against Failures



vSphere High Availability (HA)

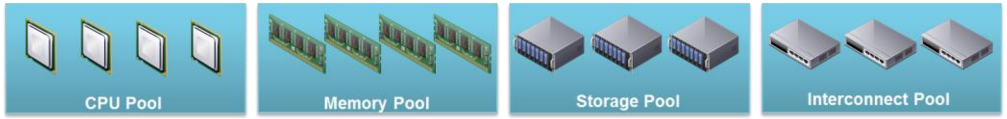
- Detect server failures and provide rapid recovery by automatically restarting VMs on available severs
- Protect applications from operating system failures by automatically restarting VMs when an operating system failure is detected.

Eliminate Planned Downtime



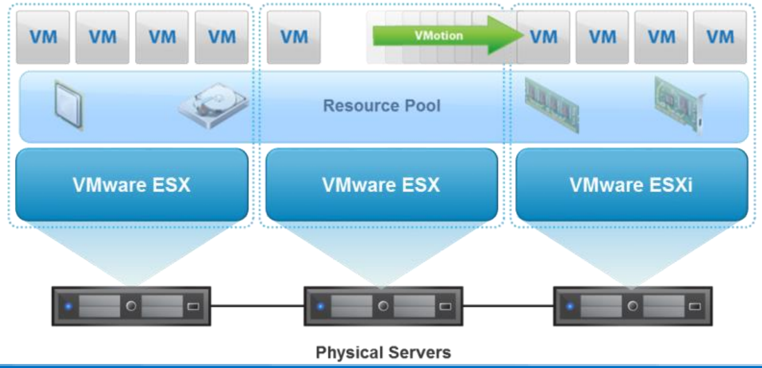
vSphere Fault Tolerance (FT)

- Single identical VMs running in lockstep on separate hosts
- Guarantee application availability and zero data loss even when a server fails
- No complex clustering or specialized hardware required



Pooled Resources Lead to Dynamic Resource Consumption via Distributed Resource Scheduler (DRS)

Distributed Resource Scheduler



Getting started: vSphere Hypervisor

Minimum Hardware Requirements:

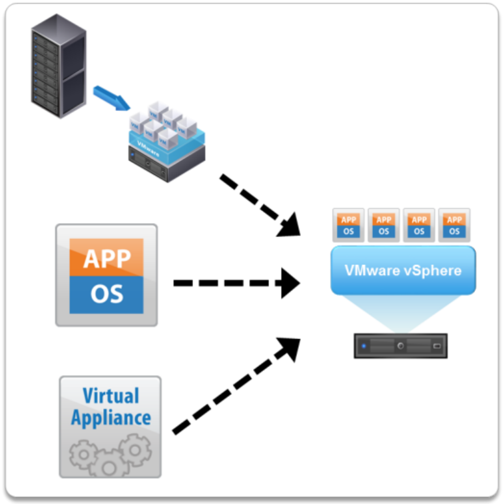
CPU Memory	▪ Single socket, dual core ▪ 4 GB
Network	▪ Two NICs (VM I/O, Management Interface)
Storage	**Local Storage (SATA/SAS):** ▪ ESXi Installable requires a 1 GB drive

Creating Virtual Machines:

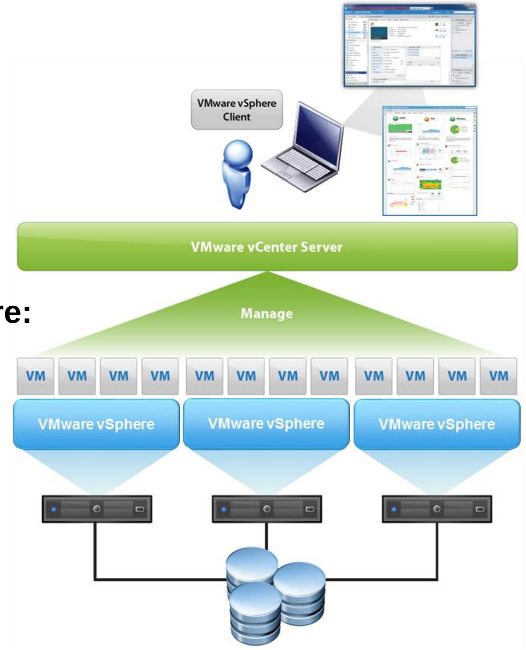
Use VMware Converter
- Transfer existing physical servers into virtual machines
- Import existing VMware and 3rd party virtual images

Create from Scratch
- Specify CPUs, Memory, Disks, Network interfaces
- Load OS from ISO image

Import a Virtual Appliance
- Hundreds to choose from on the Virtual Appliance Marketplace
- Download directly via vSphere Client and deploy on host

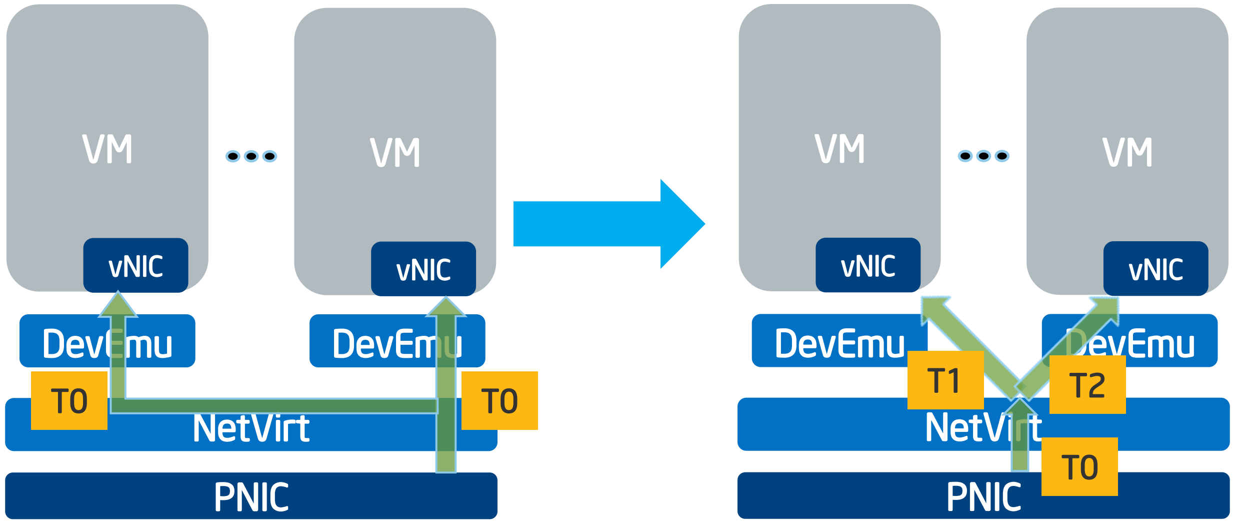


vSphere Deployment Architecture:



- Deploy vSphere on each host
- Add vCenter Server to centrally manage vSphere hosts
- Deploy vCenter Operations Management
- Upgrade license file to vSphere

Improve Multicast throughput to multiple VMs



Multicast receive traffic: single thread has to duplicate packets for ALL VMs

Set “ethernetX.emuRxMode = 1”

- Receive thread queues packets with the device emulation layer
- Per-VM thread picks up packets and carries out receive processing

Side effects: Increased receive throughput to single VM, Increased CPU Cycles/Byte