OPENFABRICS ALLIANCE

vmware®

# RDMA on vSphere: Update and Future Directions

Bhavesh Davda & Josh Simons

Office of the CTO, VMware

# Agenda

- Guest-level InfiniBand preliminary results
- Virtual Machine / Guest OS level RDMA
- Hypervisor level RDMA

# Guest-level InfiniBand with VM DirectPath I/O: Initial Results

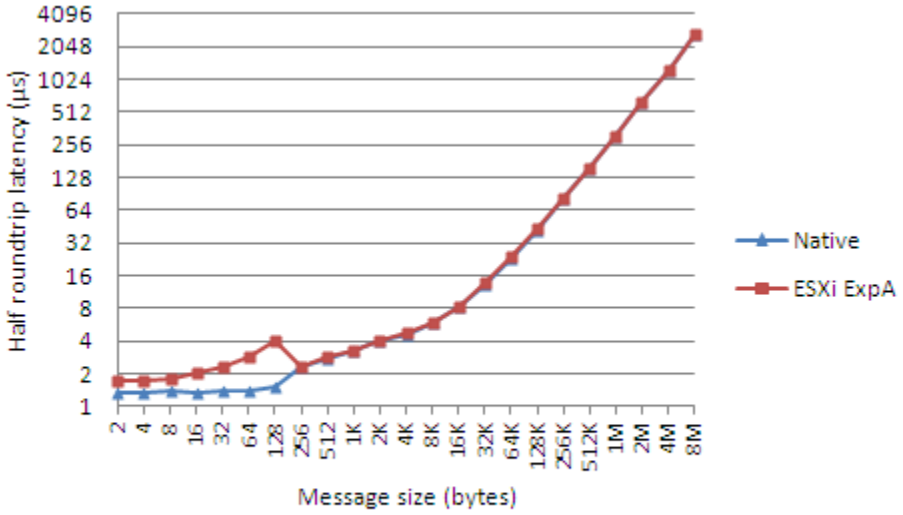



Figure 4: Send latencies using polling completions

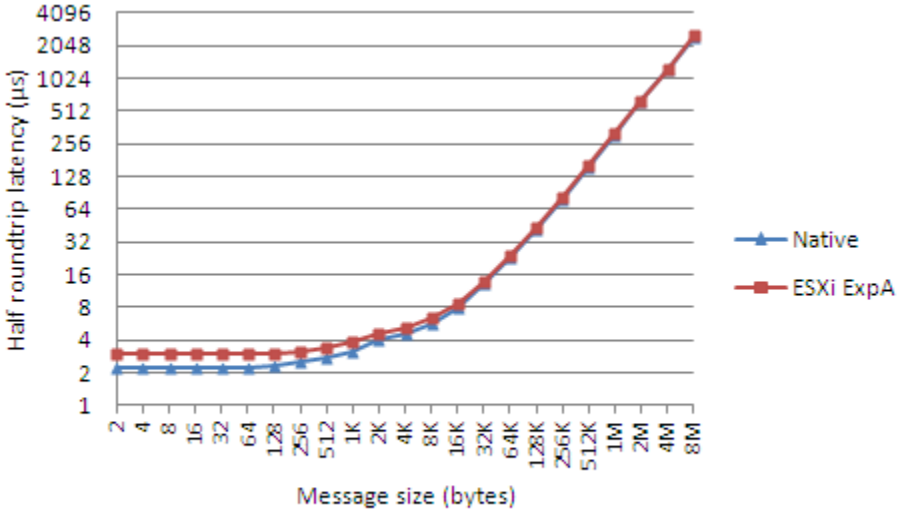



Figure 5: RDMA Read latencies using polling completions

# Guest-level InfiniBand with VM DirectPath I/O: Initial Results

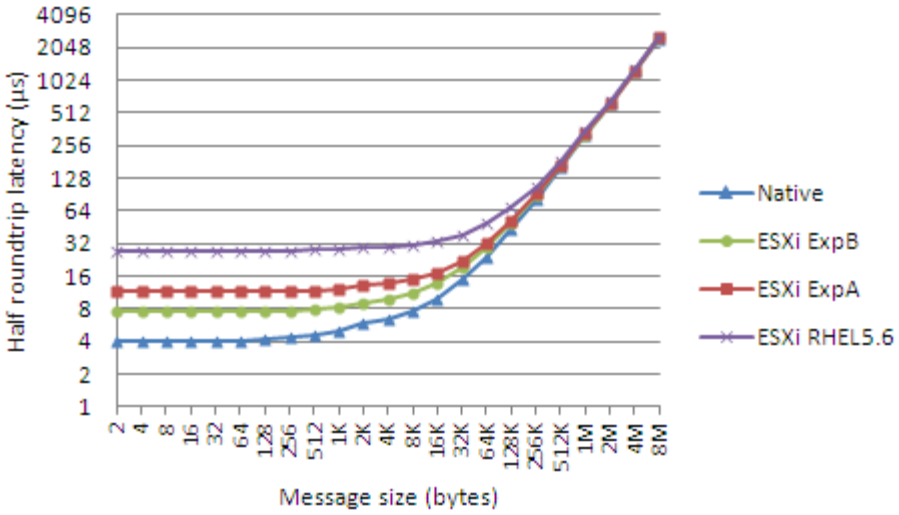



Figure 6: RDMA Read latencies using interrupt completions

Research Note currently under review:

RDMA Performance in Virtual Machines using QDR InfiniBand on VMware vSphere® 5

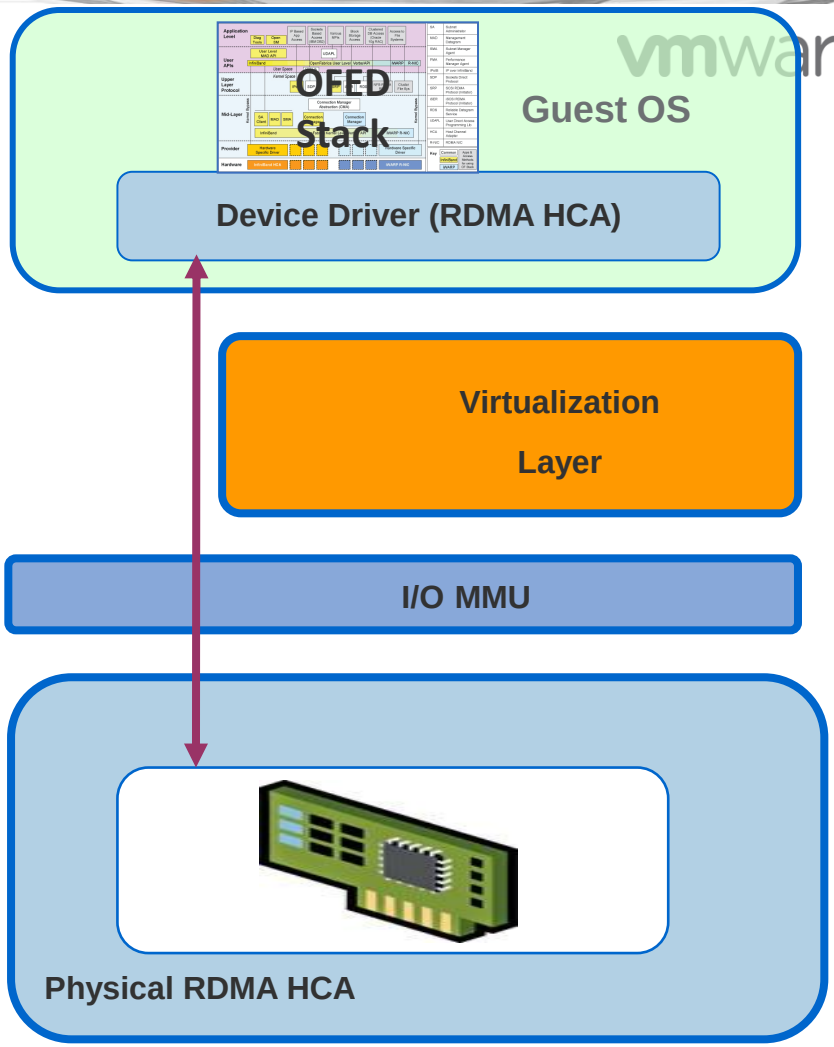
# Options to offer RDMA to vSphere Virtual Machines



A. Full-function VM DirectPath (passthrough)
B. SR-IOV VF VM DirectPath (passthrough)
C. SoftRoCE over 10GbE in VM DirectPath mode
D. SoftRoCE over paravirtual Ethernet vNIC over 10GbE uplink
E. SoftRoCE over paravirtual Ethernet vNIC between VMs
F. Paravirtual RDMA HCA (vRDMA) offered to VM

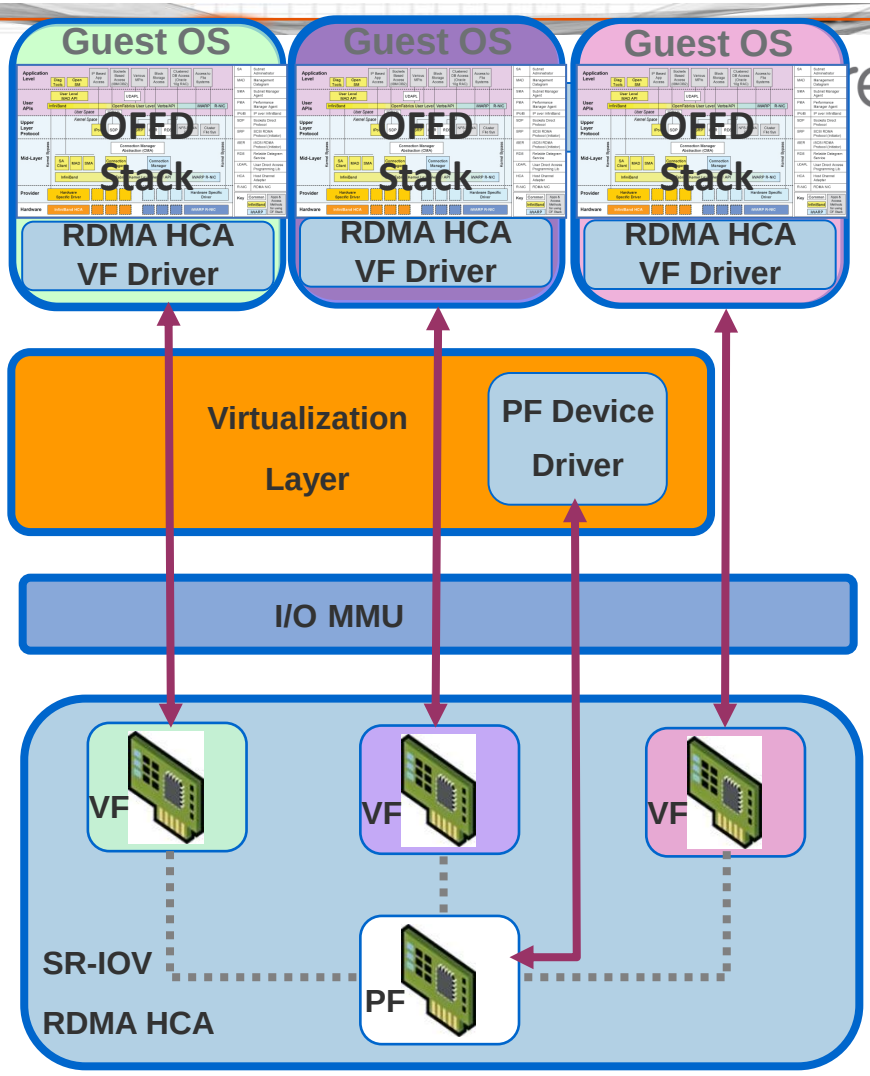
# Option A: Full-function VM DirectPath

- Direct assign physical RDMA HCA (IB/RoCE/iWARP) to VM
- Physical HCA cannot be shared between VMs or by the ESXi hypervisor
- VM DirectPath is incompatible with many important vSphere features:
  - Memory Overcommit, Fault Tolerance, Snapshots, Suspend/Resume, vMotion



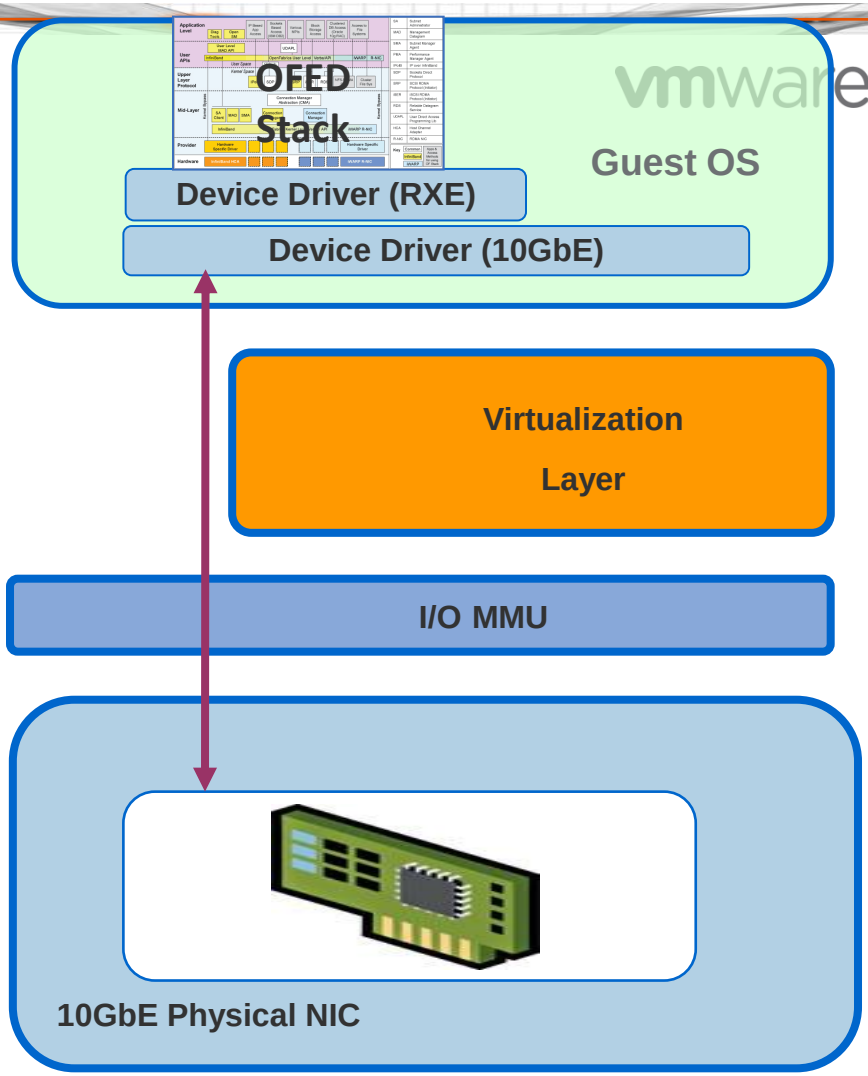
# Option B: SR-IOV VF VM DirectPath

- Single-Root IO Virtualization (SR-IOV): PCI-SIG standard
- Physical (IB/RoCE/iWARP) HCA can be shared between VMs or by the ESXi hypervisor
  - Virtual Functions direct assigned to VMs
  - Physical Function controlled by hypervisor
- Still VM DirectPath, which is incompatible with many important vSphere features



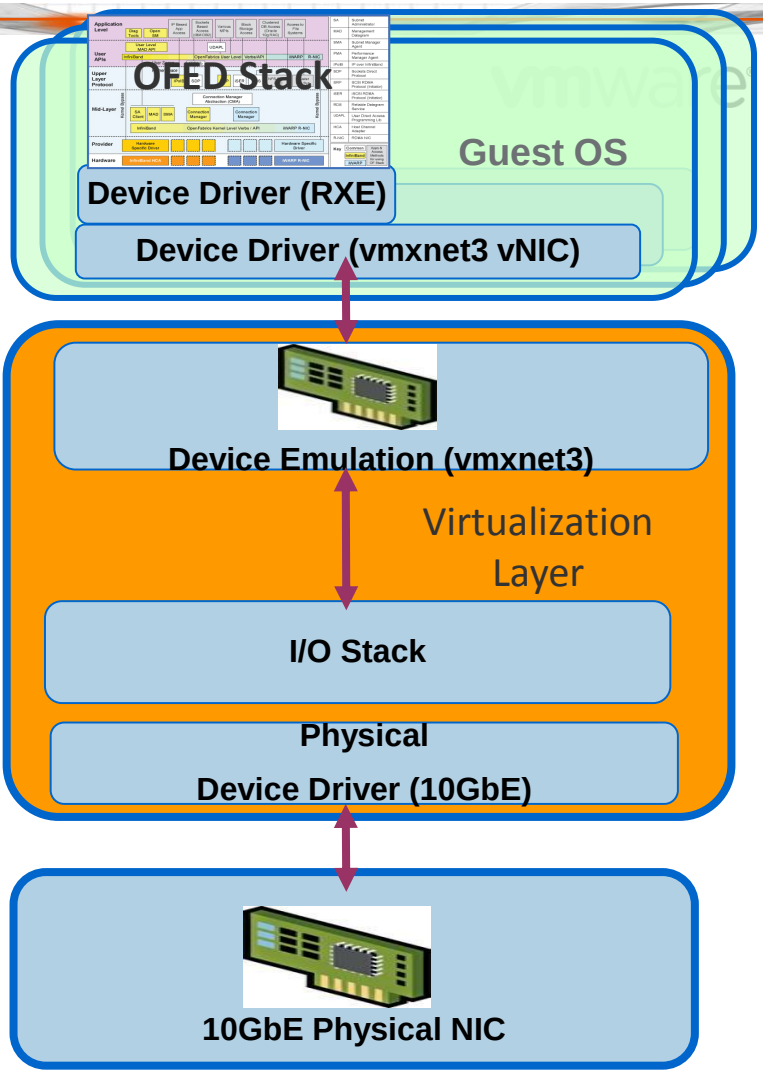
# Option C: SoftRoCE over 10GbE with VM DirectPath

- RXE driver in Guest OS bonds to 10GbE passthrough NIC (potentially SR-IOV)
- Can be used to communicate with RDMA applications on other hosts over Ethernet
  - Requires Soft/Hard RoCE in other Ethernet connected hosts as well
- VM DirectPath == Sacrifice vSphere features
- Didn't work (needs debugging)
  - Completion with error at client: Failed status 12: wr_id 1 syndrom 0x81 scnt=1, ccnt=0



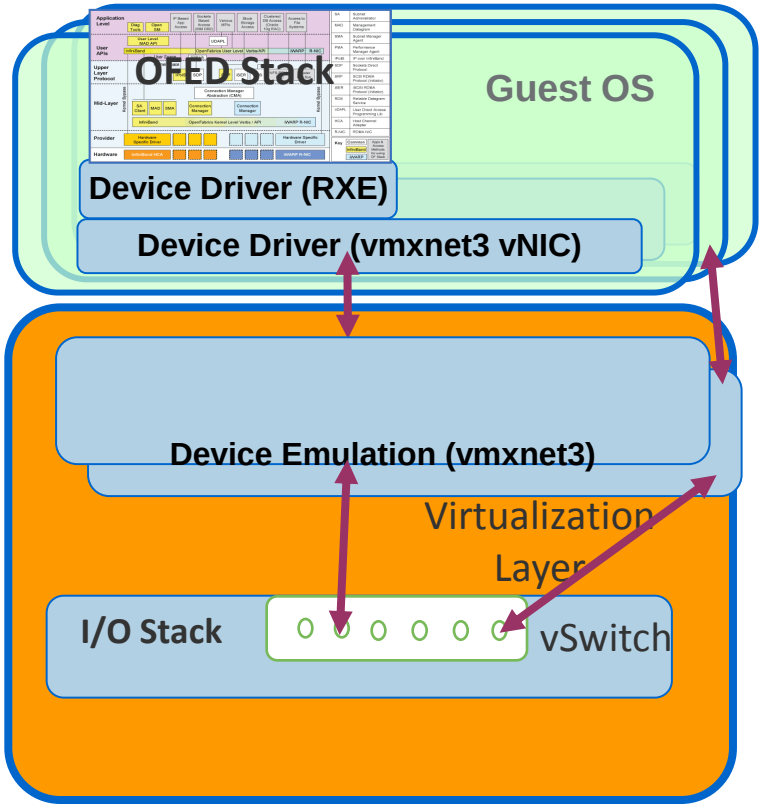
# Option D: SoftRoCE / Paravirtual vNIC / 10GbE uplink

- RXE driver in Guest OS bonds to paravirtual vmxnet3 vNIC
  - vmxnet3 emulation in hypervisor
  - Connected to a vSwitch with a10GbE physical NIC uplink
- Can be used to communicate with RDMA applications on other hosts (Soft/Hard RoCE) over Ethernet
- Doesn't rely on VMDirectPath
  - All vSphere features usable



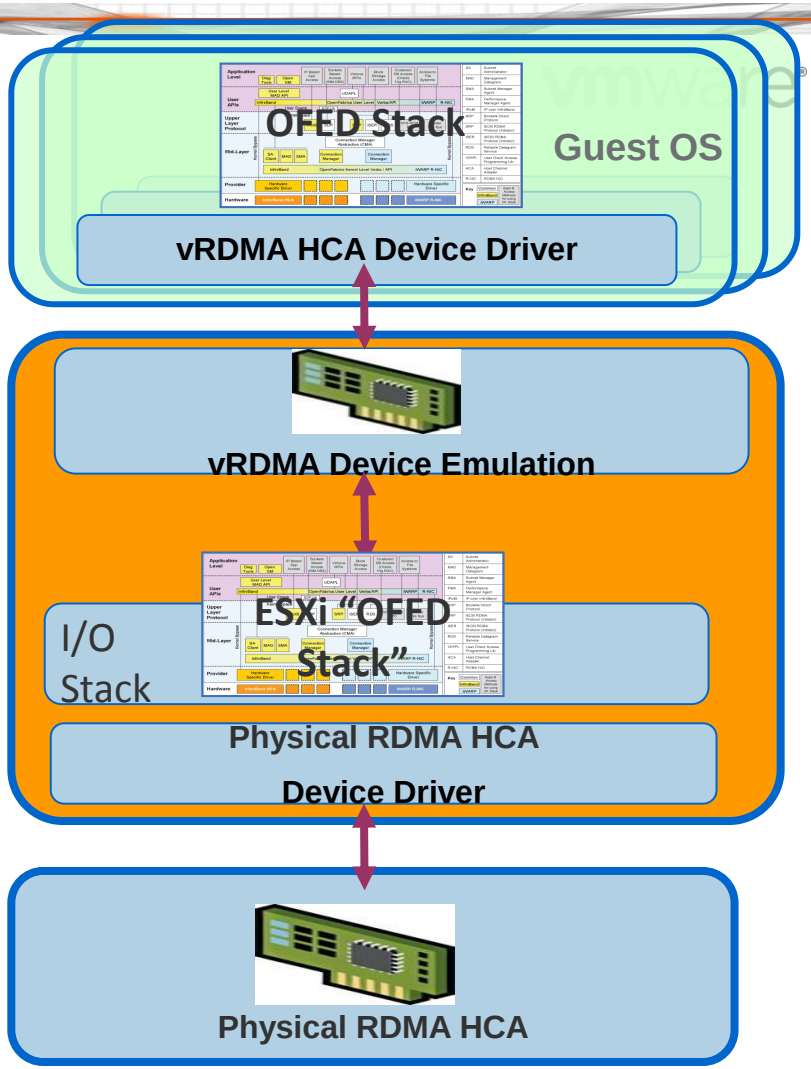
# Option E: VM-VM SoftRoCE / Paravirtual vNIC

- RXE driver in Guest OS bonds to paravirtual vmxnet3 vNIC
- Can be used to communicate with RDMA applications in other VMs over Ethernet over vSwitch
  - Requires SoftRoCE in other VMs as well
- Doesn't rely on VMDirectPath
  - All vSphere features usable

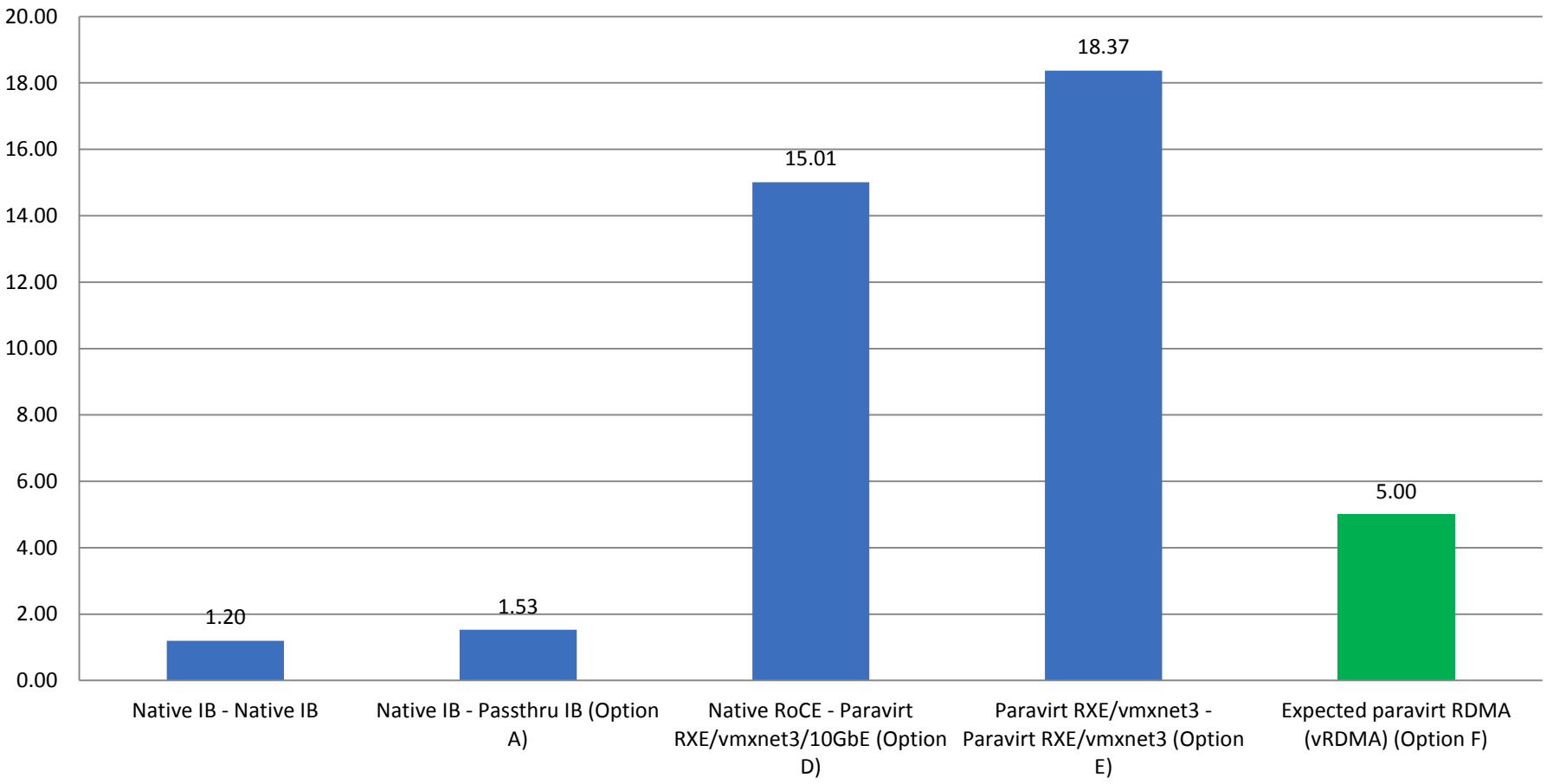


# Option F: Paravirtual RDMA HCA (vRDMA) offered to VM

- New paravirtualized device exposed to Virtual Machine
  - Implements “Verbs” interface
- Device emulated in ESXi hypervisor
  - Translates Verbs from Guest to Verbs to ESXi “OFED Stack”
  - Guest physical memory regions mapped to ESXi and passed down to physical RDMA HCA
  - Zero-copy DMA directly from/to guest physical memory
  - Completions/interrupts “proxied” by emulation
- “Holy Grail” of RDMA options for vSphere VMs



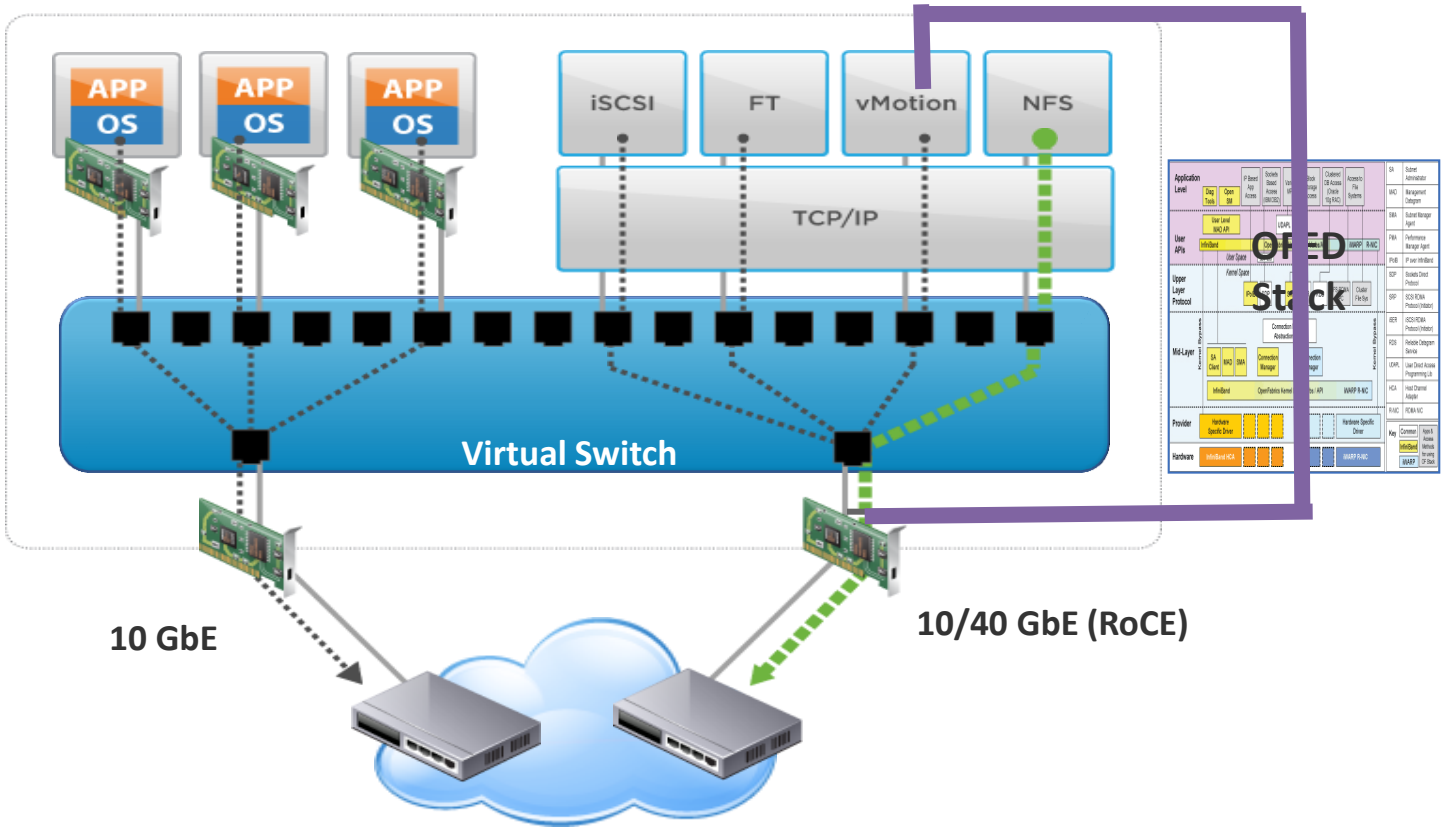
# Performance Comparison



# Summary: VM/Guest Level RDMA

- Option A: Full function VMDirectPath already used by some customers
- Option B: SR-IOV VMDirectPath will be included in an upcoming vSphere release
- Options C/D/E: SoftRoCE can enable RDMA-based guest apps today at the cost of higher latencies
- Option F: vRDMA paravirtual device being prototyped in Office of CTO

# RDMA for vSphere hypervisor services



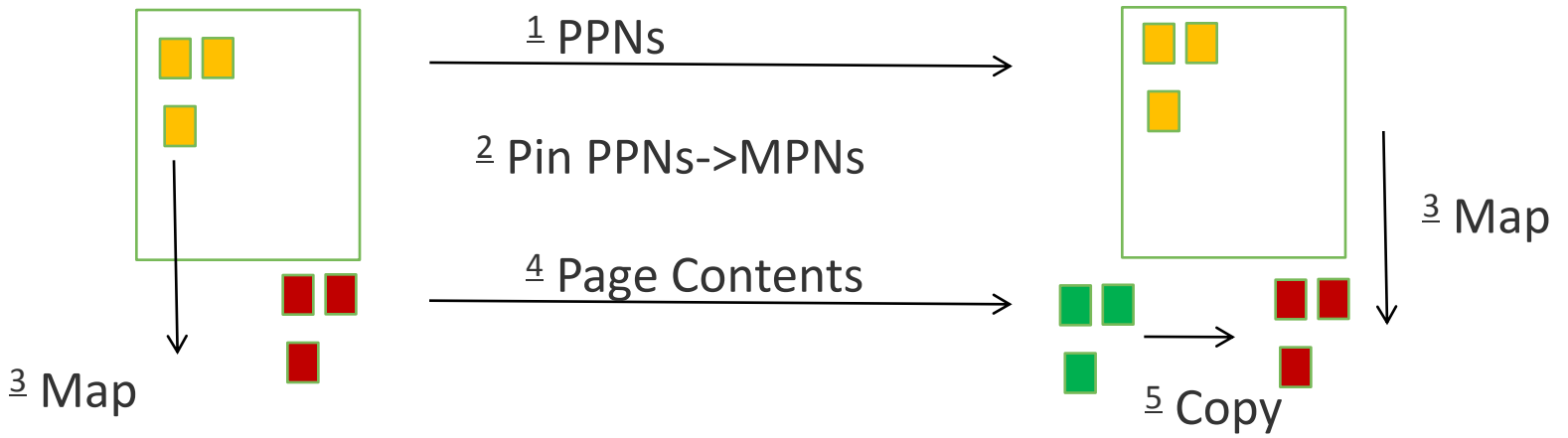
# vMotion Using RDMA Transport

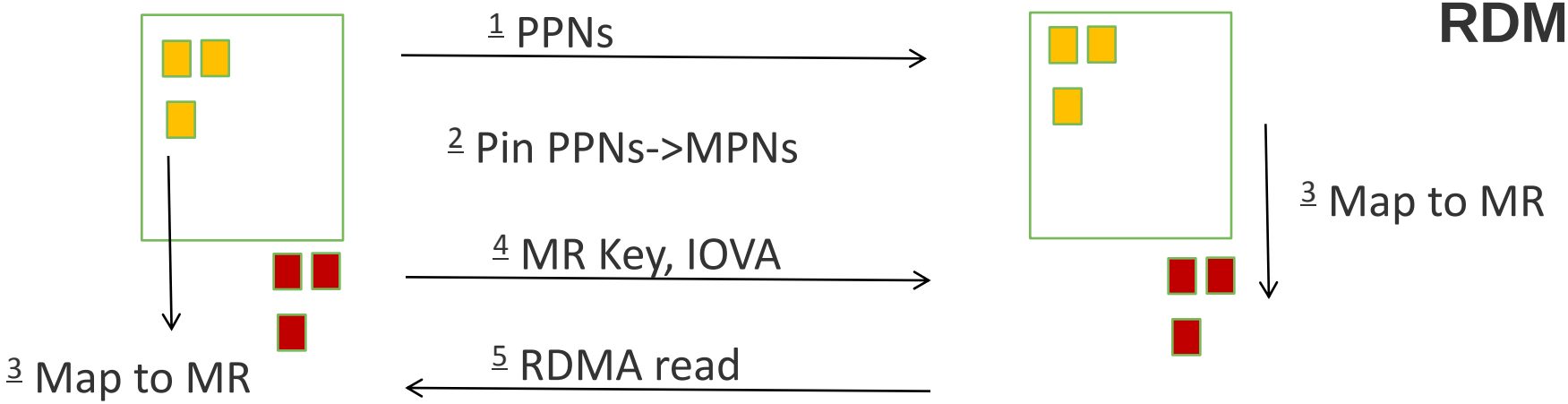
- vMotion: live migration of virtual machines
  - Key enabler for Distributed Resource Scheduler (DRS) and Distributed Power Management (DPM)
- vMotion data transfer mechanism
  - Currently network (TCP/IP/BSD sockets/mbufs) based
- Why RDMA? Performance
  - Zero copy send & receive == Reduced CPU utilization
  - Lower latency
  - Eliminate TCP/IP protocol & processing overheads

# vMotion (ESXi 5.0) Basics

- Stages: Pre copy -> Quiesce -> Page in -> Unstun
- VM memory transfer during Pre copy & Page in
  - Source: “Send Pages”; Destination: “Receive Pages”
- Multiple vmkernel helper tasks for send and receive
- SDPS (Stun During Page Send)
  - Stall vCPUs if page dirty rate is greater than page transfer rate
  - Facilitates pre copy convergence so few pages outstanding during page-in

# Send Pages and Receive Pages Operations over RDMA

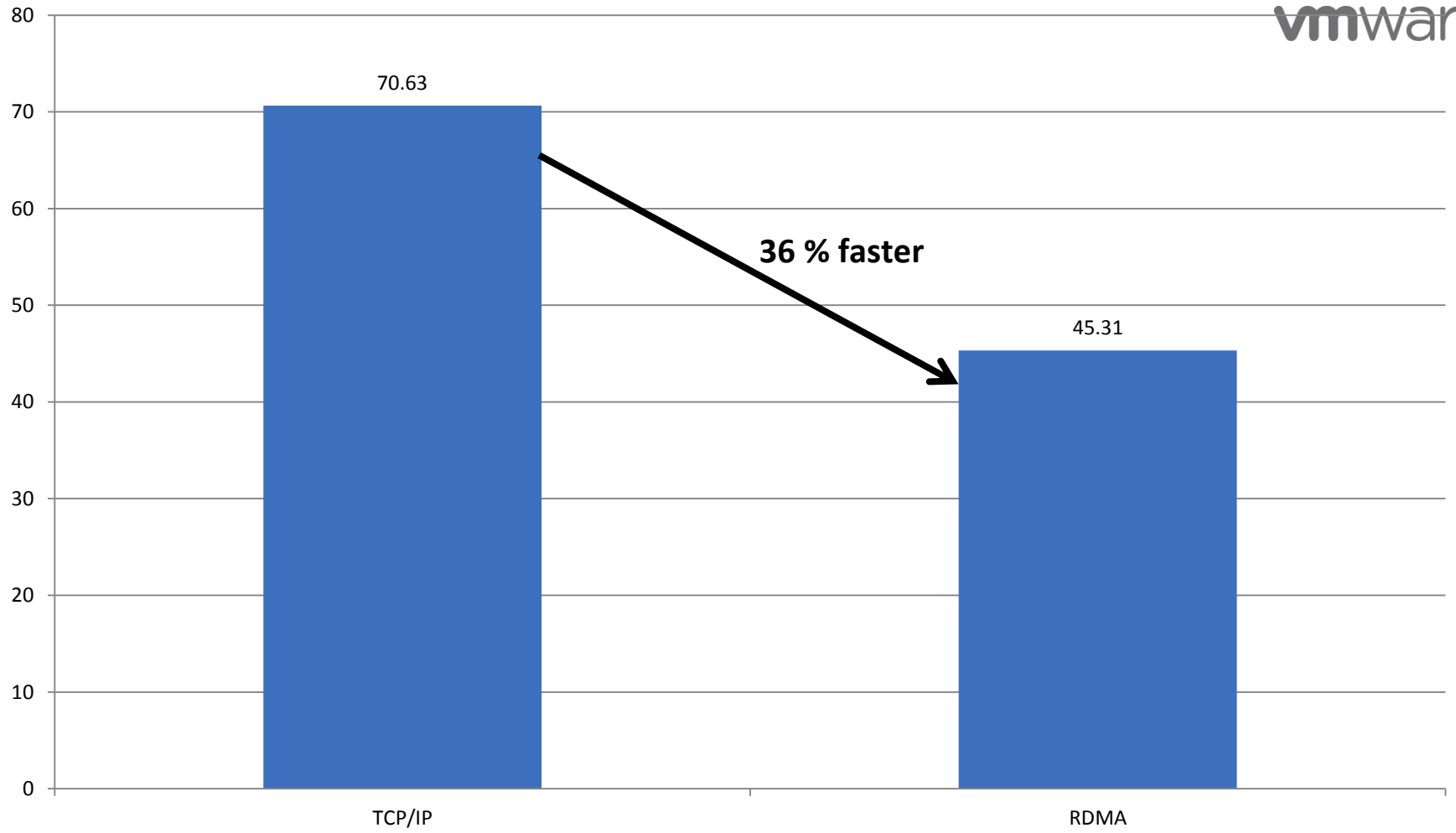


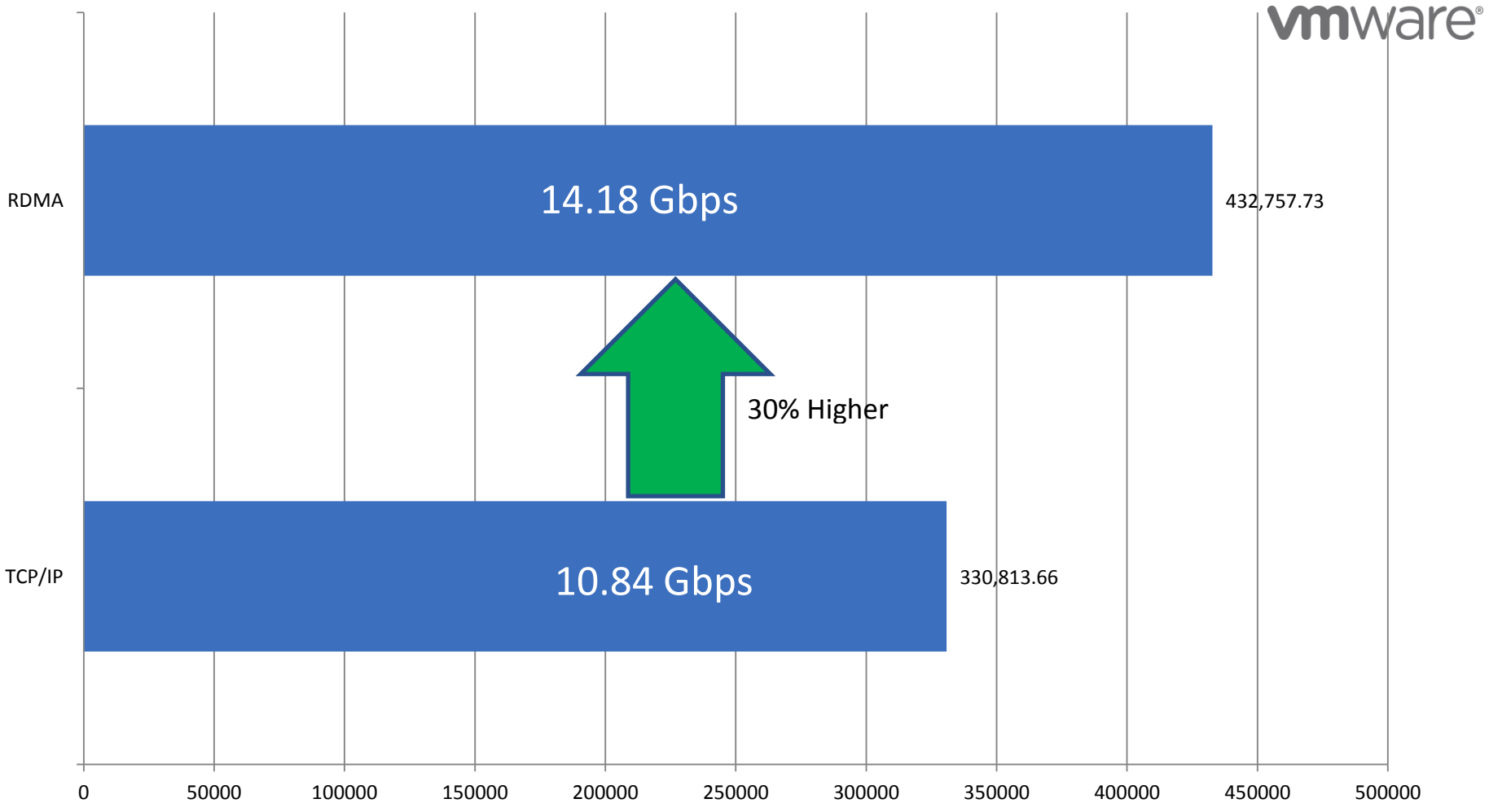
# Test setup

- Two HP ProLiant ML 350 G6 machines, 2x Intel Xeon (E5520, E5620), HT enabled, 60 GB RAM
- Mellanox 40GbE RoCE cards
  - ConnectX-2 VPI PCIe 2.0 x8, 5.0 GT/s
- SPECjbb2005 50GB workload
  - 56 GB, 4 vCPU Linux VM
  - Run-time config switch for TCP/IP or RDMA transport
  - Single stream helper for RDMA transport vs. two for TCP/IP transport
  - SDPS disabled to reduce variance in results

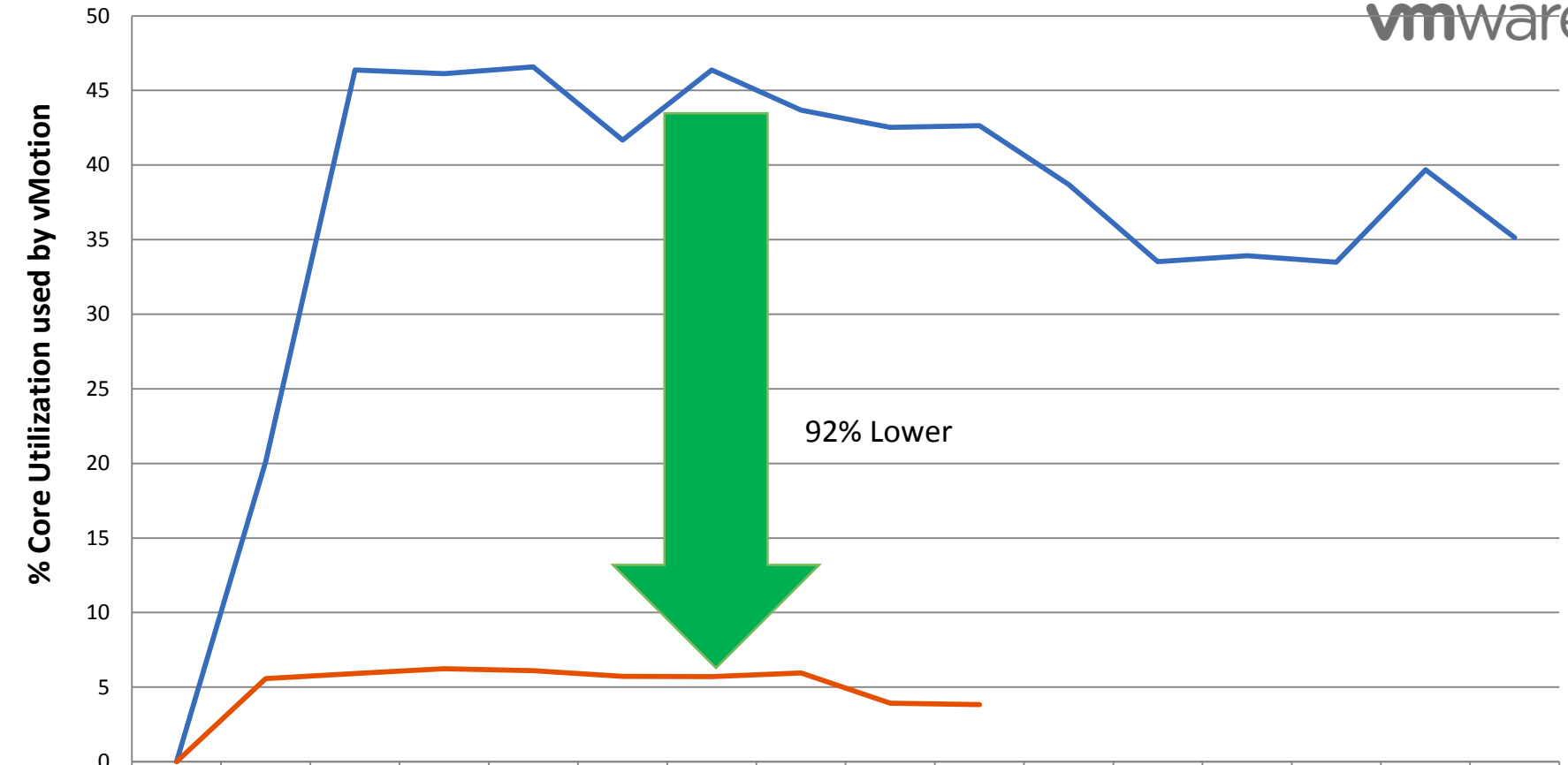
# Total vMotion Time (seconds)



# Precopy bandwidth (Pages/sec)

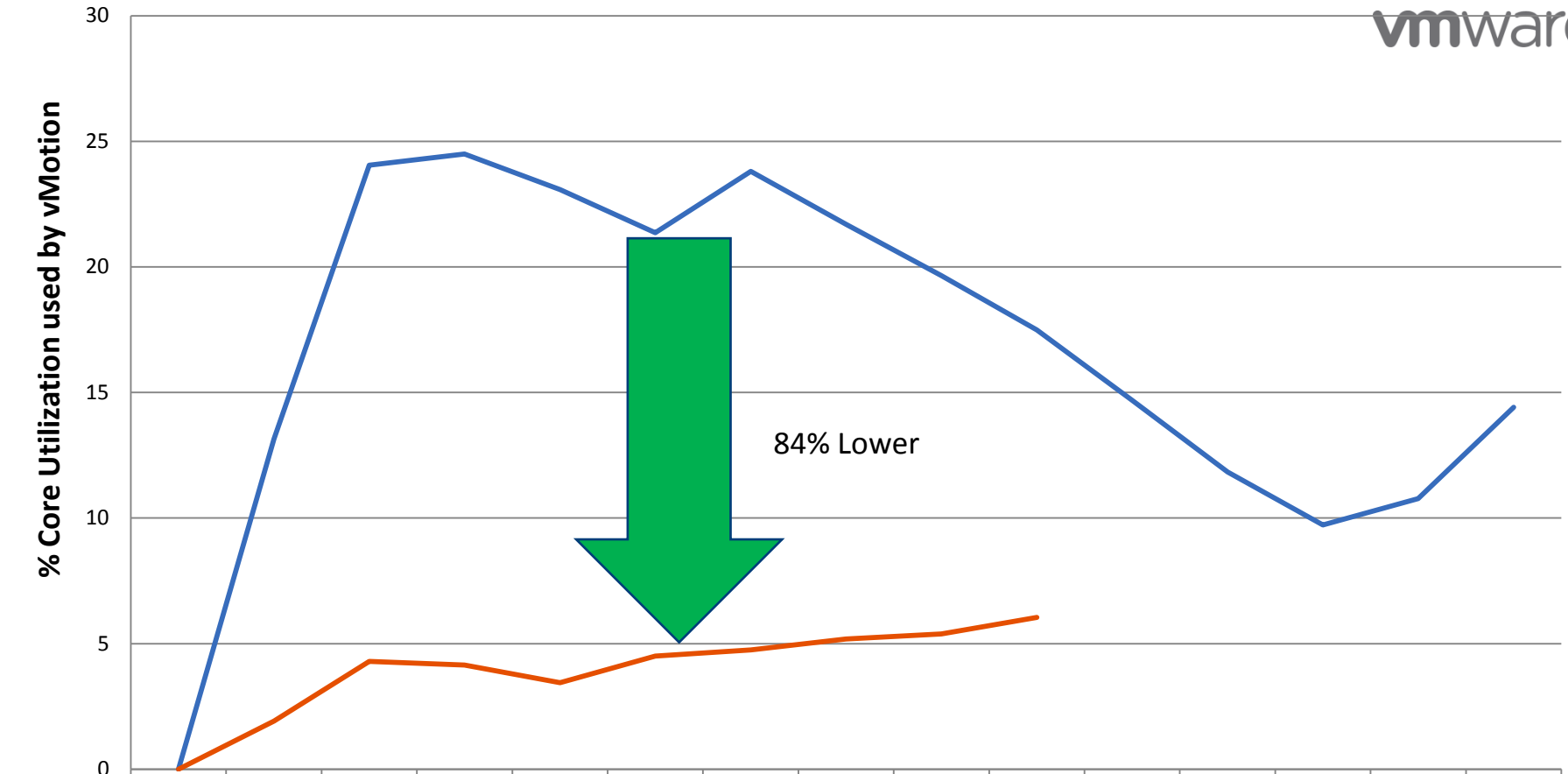


# Destination CPU Utilization




| | 0:00 | 0:05 | 0:11 | 0:16 | 0:21 | 0:26 | 0:31 | 0:36 | 0:41 | 0:46 | 0:51 | 0:56 | 1:01 | 1:06 | 1:11 | 1:16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TCP/IP | 0 | 20.11 | 46.38 | 46.13 | 46.58 | 41.68 | 46.38 | 43.69 | 42.53 | 42.64 | 38.69 | 33.53 | 33.93 | 33.49 | 39.68 | 35.13 |
| RDMA | 0 | 5.58 | 5.92 | 6.24 | 6.11 | 5.73 | 5.71 | 5.96 | 3.92 | 3.83 | | | | | | |

# Source CPU Utilization




| | 0:00 | 0:05 | 0:11 | 0:16 | 0:21 | 0:26 | 0:31 | 0:36 | 0:41 | 0:46 | 0:51 | 0:56 | 1:01 | 1:06 | 1:11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TCP/IP | 0 | 13.15 | 24.05 | 24.5 | 23.08 | 21.36 | 23.8 | 21.7 | 19.65 | 17.49 | 14.69 | 11.83 | 9.72 | 10.77 | 14.41 |
| RDMA | 0 | 1.92 | 4.29 | 4.15 | 3.44 | 4.5 | 4.75 | 5.19 | 5.39 | 6.04 | | | | | |

# Hypervisor level RDMA Requirements

- Integrate OFED Kernel Space Mid-Layer and Provider components into ESXi hypervisor
  - IP and licensing issues with OFED components need sorting out
- Add RDMA Verbs support for other hypervisor services: FT, NFS, iSCSI, others
- Create abstraction layer from common patterns to ease porting other TCP/IP/BSD mbuf-based hypervisor services
- Under discussion with vSphere product teams

# Summary

- Hypervisor-level RDMA
  - Proven benefits for hypervisor services (vMotion, etc.)
  - Currently under discussion internally
- Passthrough IB delivers credible performance
- Paravirtual vRDMA most attractive option for VM-level RDMA
  - Maintains key virtualization capabilities
  - Delivers better latencies than other approaches
  - Prototyping underway

# Acknowledgements


- VMDirectPath InfiniBand
  - VMware intern from Georgia Tech: Adit Ranadive
- vMotion over RDMA
  - VMware: Anupam Chanda, Gabe Tarasuk-Levin, Scott Goldman
  - Mellanox: Ali Ayoub
- RDMA for Virtual Machines
  - Mellanox: Ali Ayoub, Motti Beck, Dror Goldenberg
  - SFW: Paul Grun, Bob Pearson