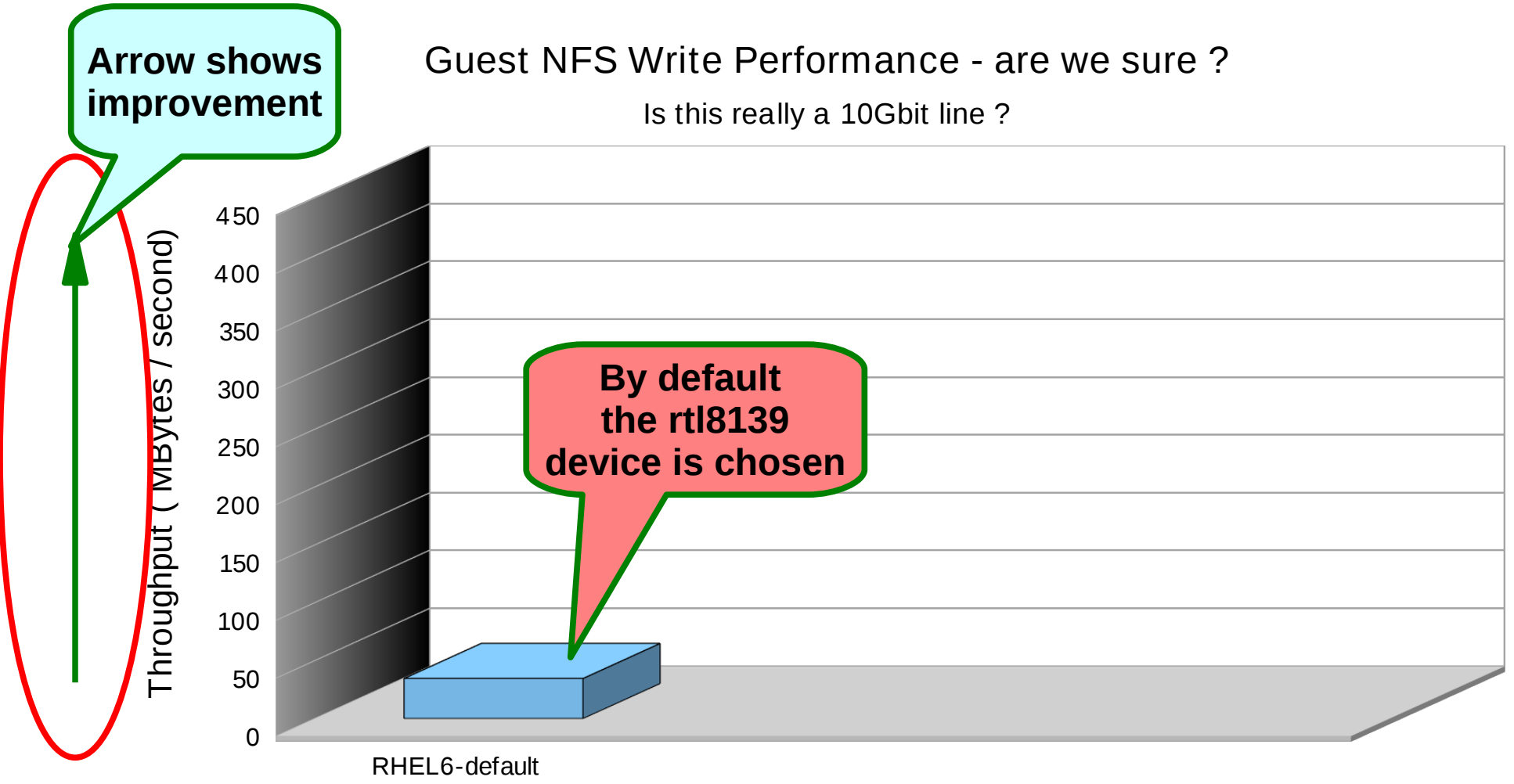
# KVM PERFORMANCE IMPROVEMENTS AND OPTIMIZATIONS

Mark Wagner
Principal SW Engineer, Red Hat
August 14, 2011

# Overview

- Discuss a range of topics about KVM performance
  - How to improve out of the box experience
  - But crammed into 30 minutes
- Use libvirt where possible
  - Note that not all features in all releases

# Before we dive in...



# Agenda

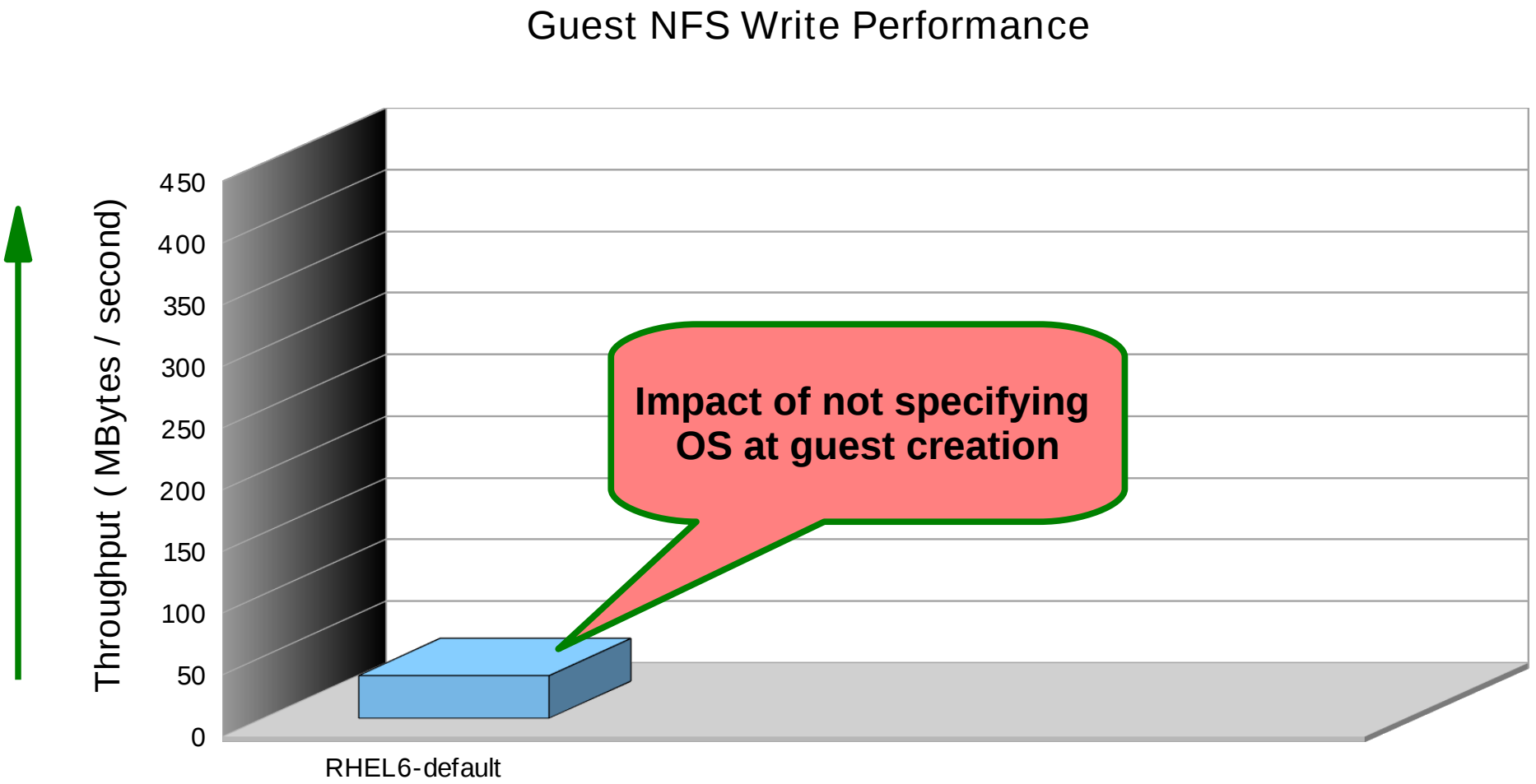
- Low hanging fruit
- Memory
- Networking
- Block I/O basics
- NUMA and affinity settings
- CPU Settings
- Wrap up

# Recent Performance Improvements

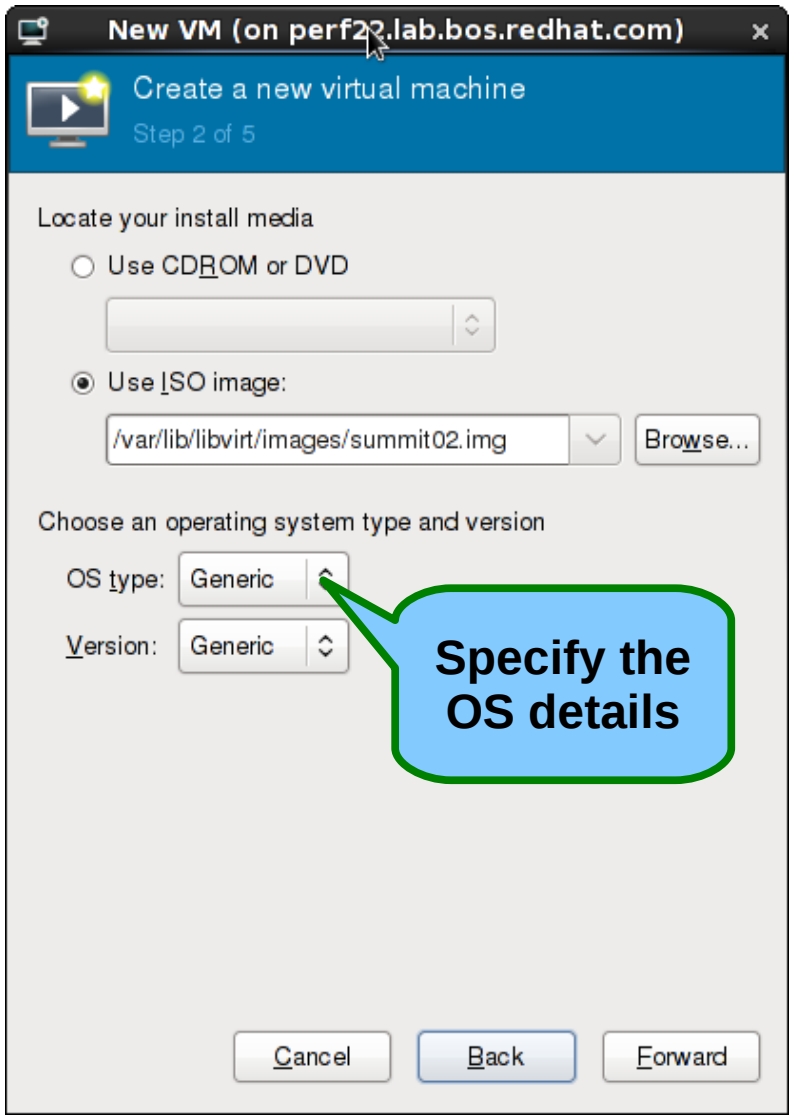
- *Performance enhancements in every component*

| Component | Feature |
|---|---|
| CPU/Kernel | NUMA – Ticketed spinlocks; Completely fair scheduler; Extensive use of Read Copy Update (RCU) Scales up to 64 vcpus per guest |
| Memory | Large memory optimizations: Transparent Huge Pages is ideal for hardware based virtualization |
| Networking | Vhost-net – a kernel based virtio w/ better throughput and latency. SRIOV for ~native performance |
| Block | AIO, MSI, scatter gather. |

# Remember this ?



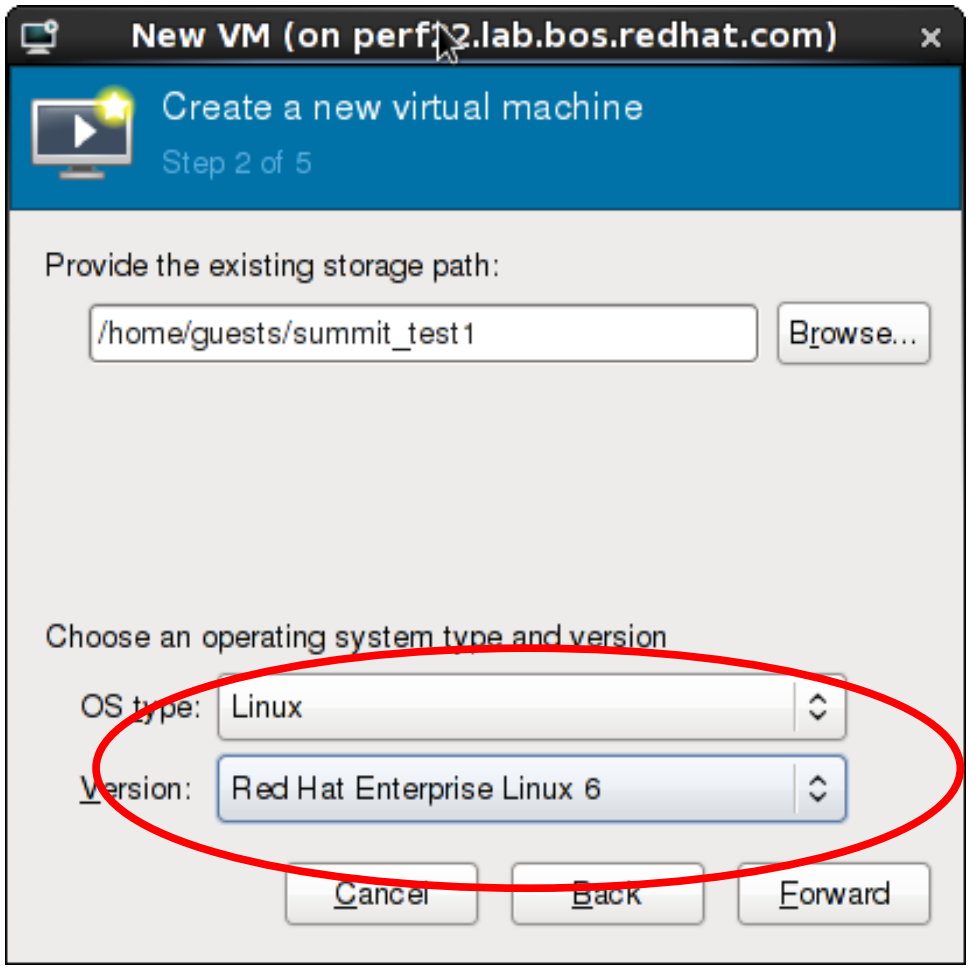
# Be Specific !



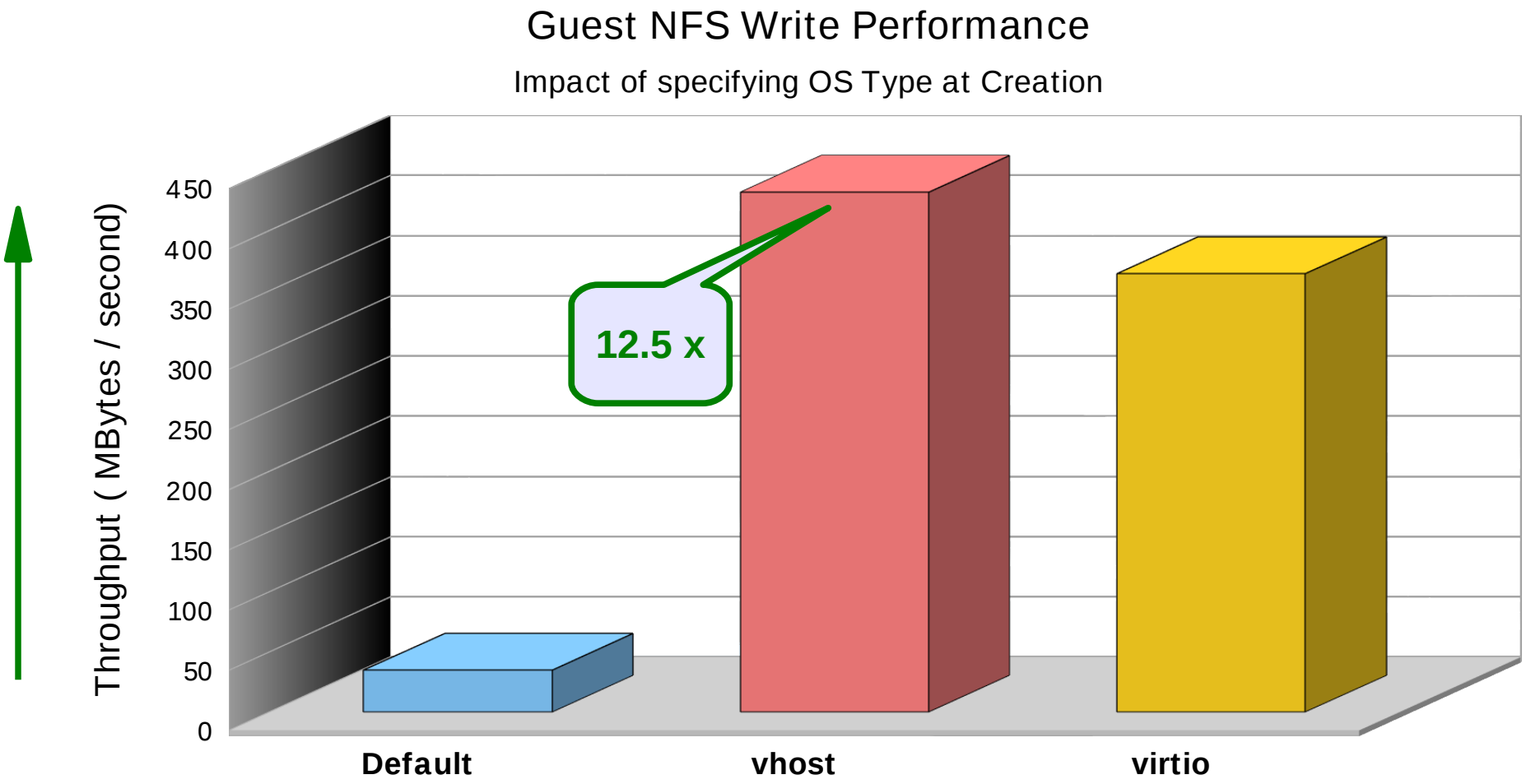

- virt-manager will:
  - Make sure the guest will function
  - Optimize as it can
- The more info you provide the more tailoring will happen

# Specify OS + flavor

- Specifying Linux will get you:
  - The virtio driver
  - If the kernel is recent enough the vhost_net drivers



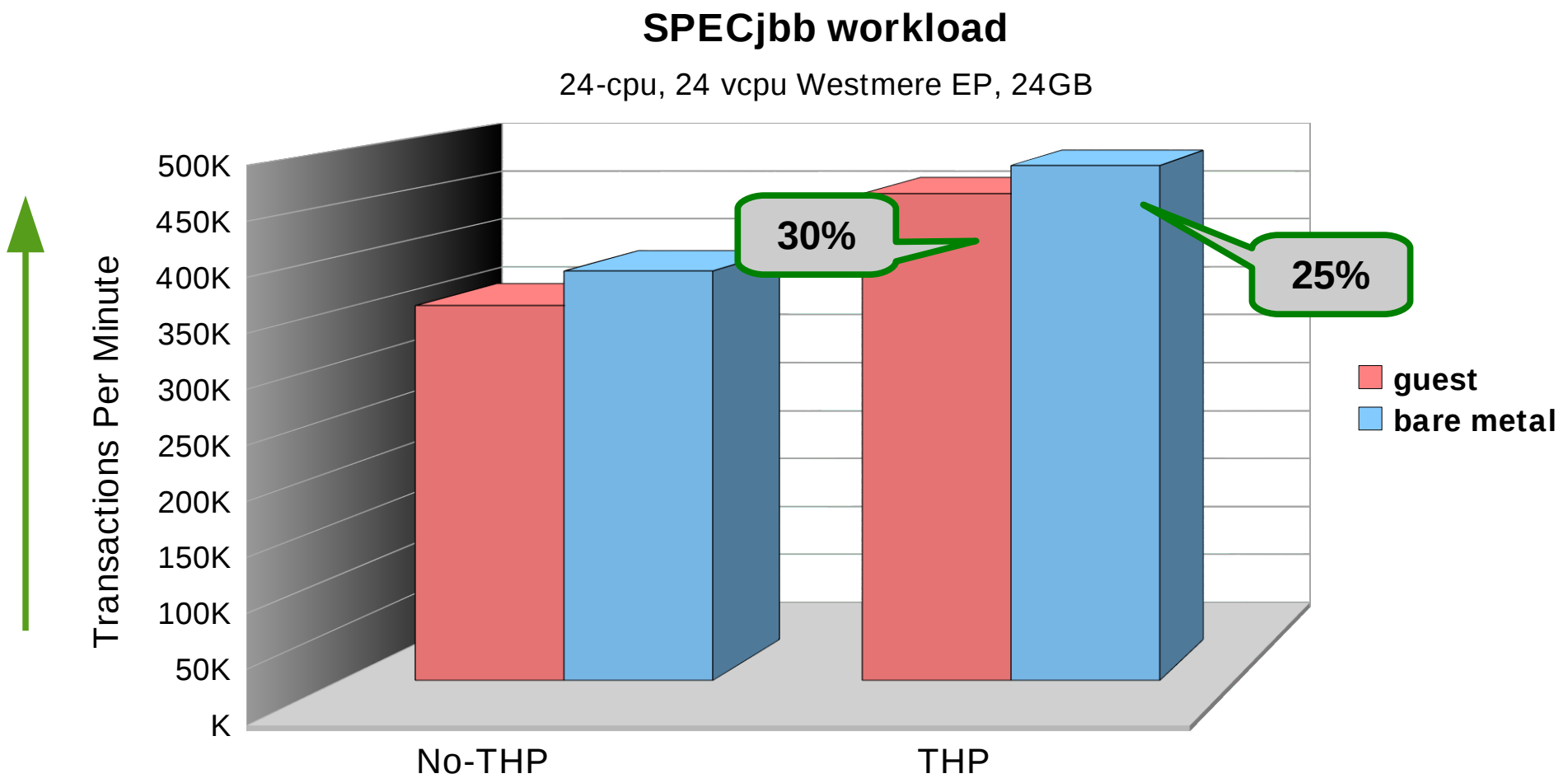
# I Like This Much Better



# Memory Tuning – Huge Pages

- 2M pages vs 4K standard Linux page
  - Virtual to physical page map is 512 times smaller
  - TLB can map more physical page resulting fewer misses
- Traditional Huge Pages always pinned
- We now have Transparent Huge Pages
- Most databases support Huge Pages
- Benefits not only Host but guests
  - Try them in a guest too !

# Transparent Huge Pages

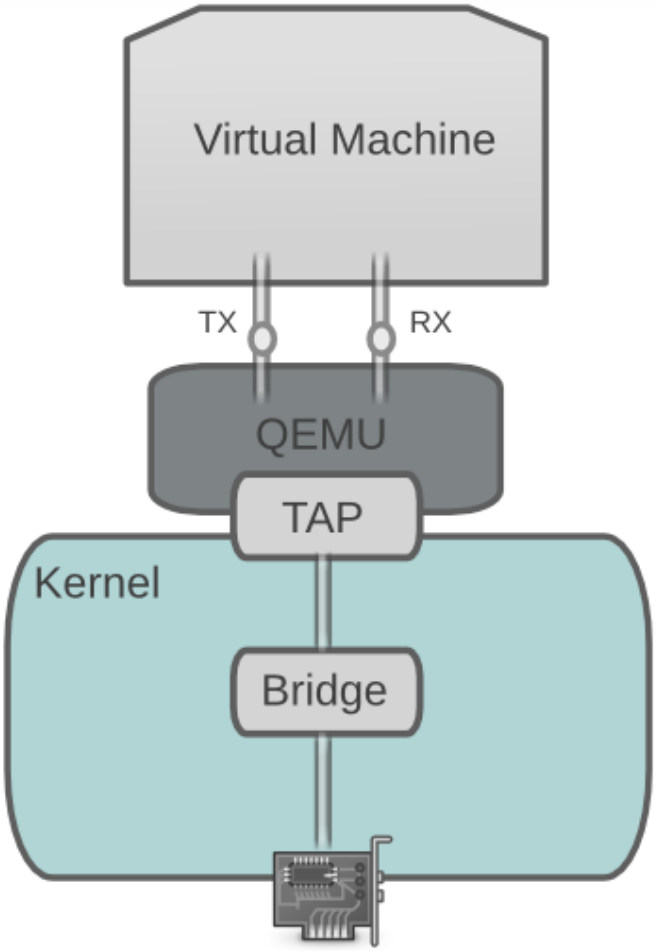


# Network Tuning Tips

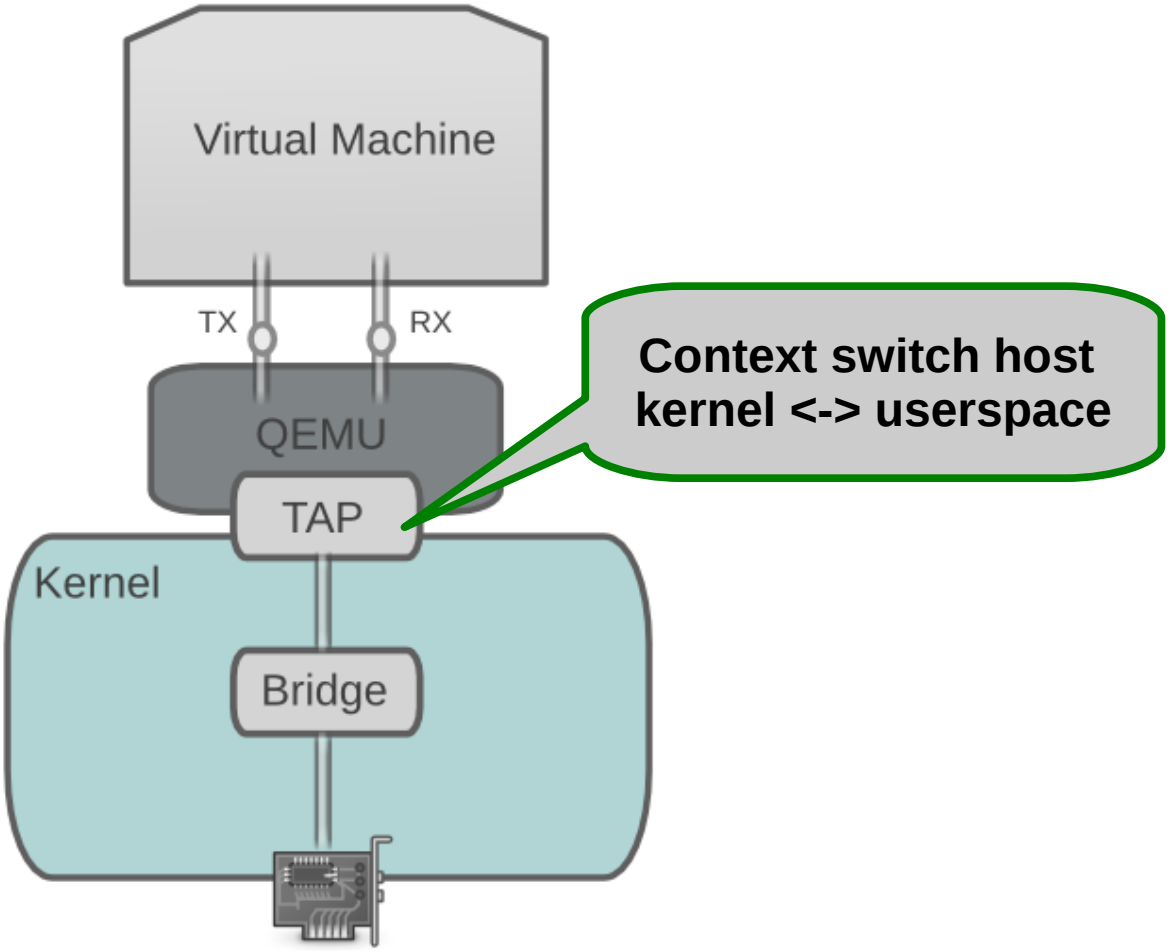
- Separate networks for different functions
  - Use arp_filter to prevent ARP Flux
    - echo 1 > /proc/sys/net/ipv4/conf/all/arp_filter
    - Use /etc/sysctl.conf for permanent
- Packet size - MTU
  - Need to make sure it is set across all components
- Don't need HW to bridge  intra-box communications
  - VM traffic never hits the HW on same box
  - Can really kick up MTU as needed

# KVM Network Architecture - VirtIO

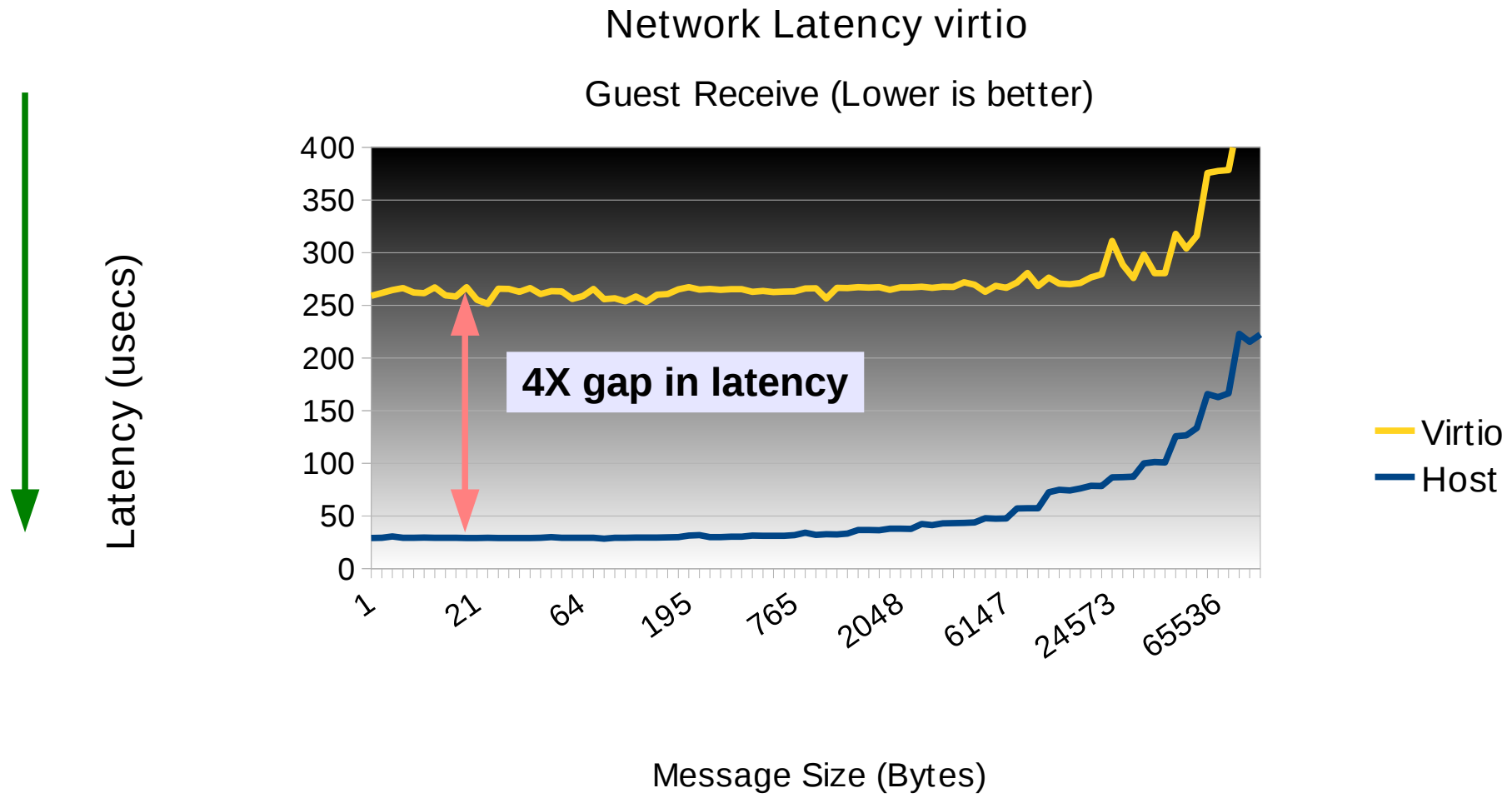
- Virtual Machine sees paravirtualized network device – VirtIO
  - VirtIO drivers included in Linux Kernel
  - VirtIO drivers available for Windows
- Network stack implemented in userspace



## Virtio

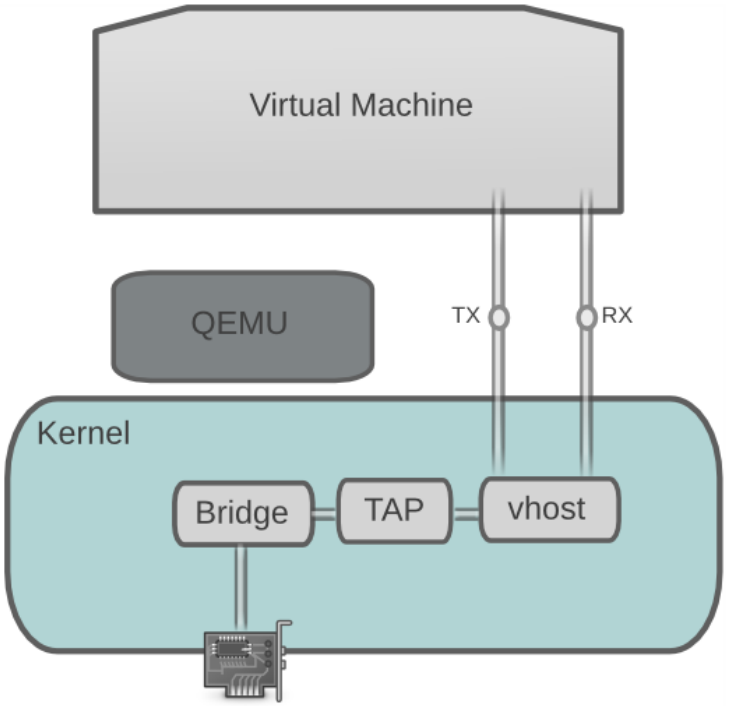

Virtual Machine
TX
RX
QEMU
TAP
Kernel
Bridge
Context switch host
kernel <-> userspace

# Latency comparison

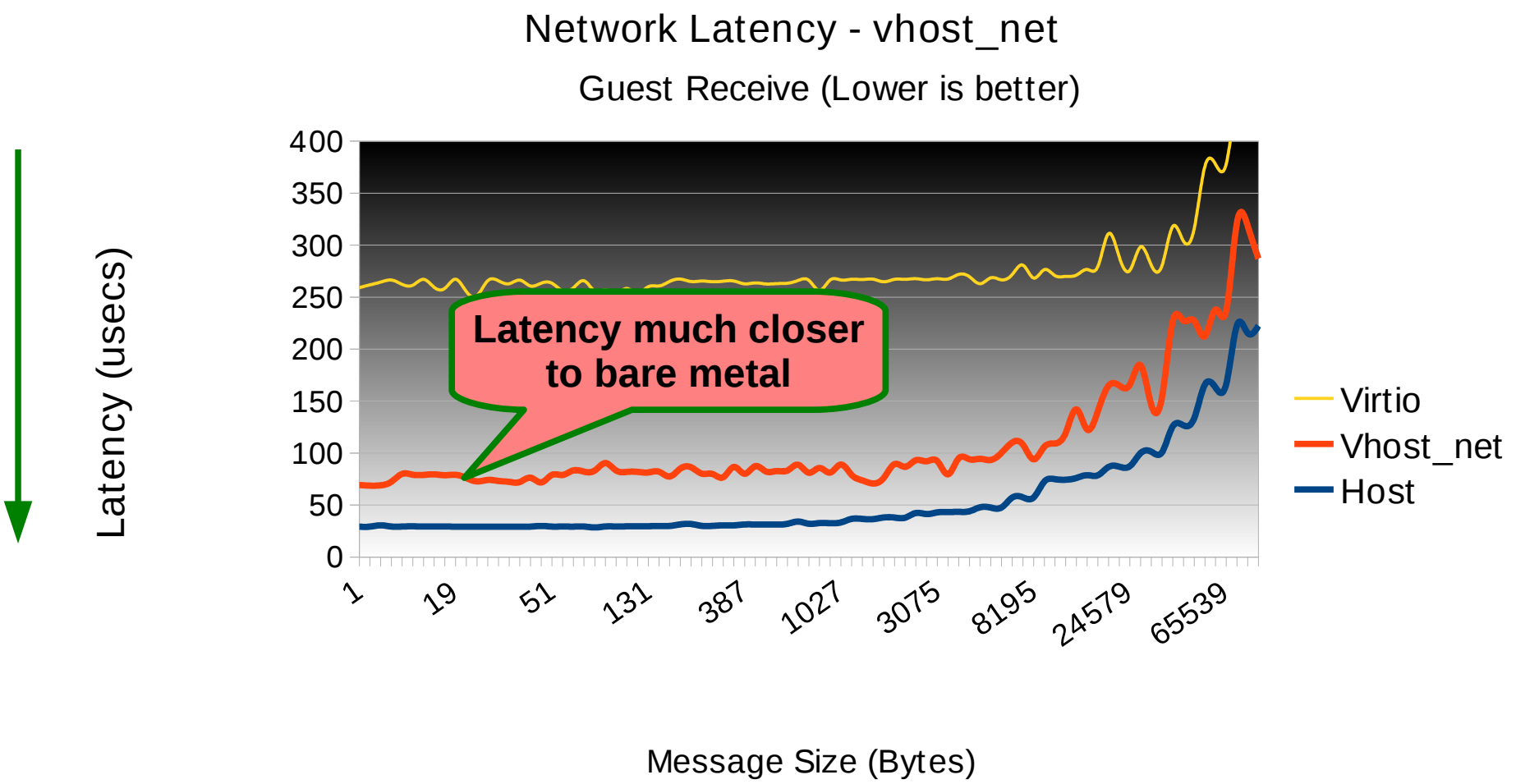


# KVM Network Architecture – vhost_net

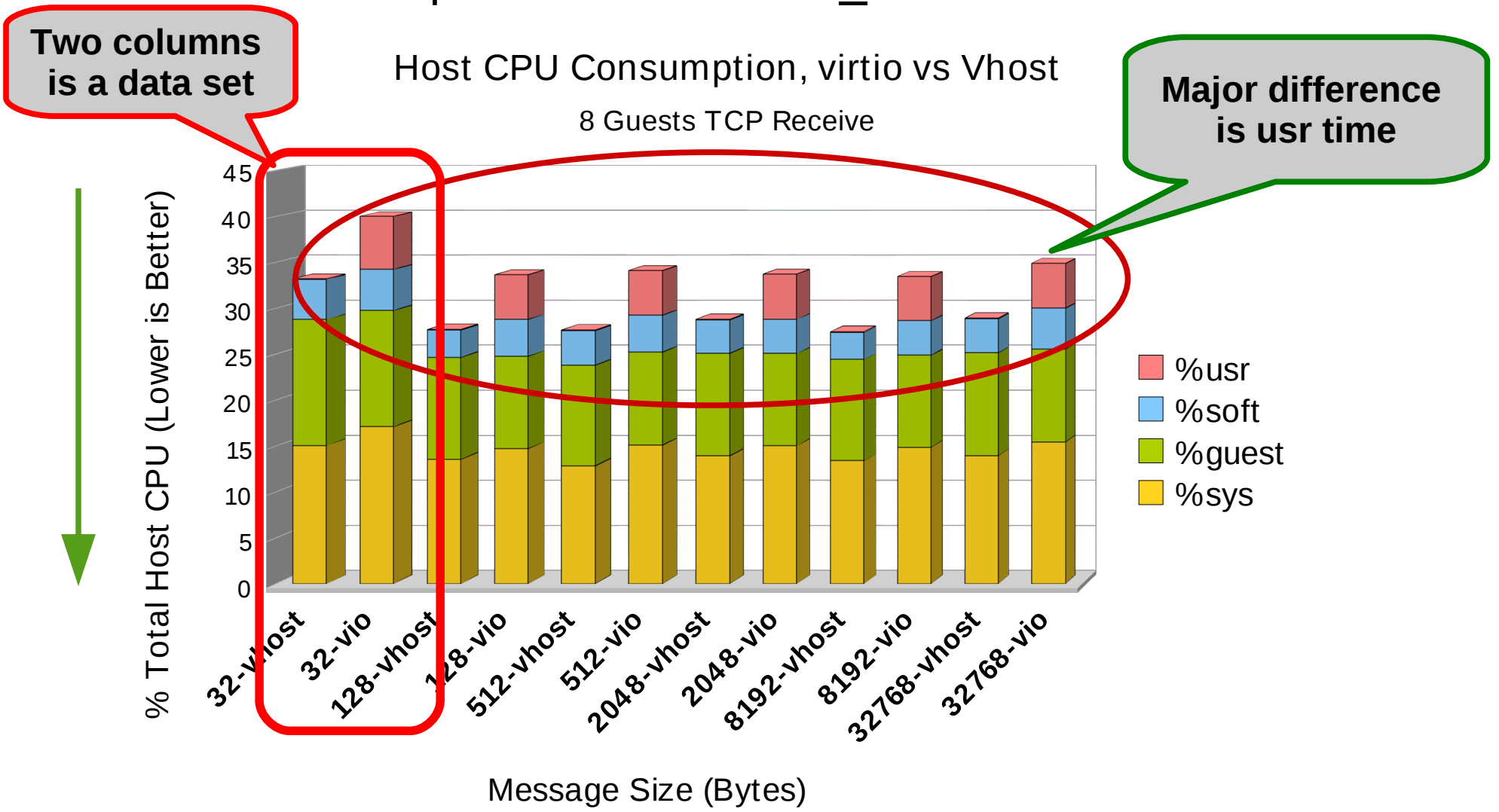
- Moves QEMU network stack from userspace to kernel
- Improved performance
- Lower Latency
- Reduced context switching
- One less copy



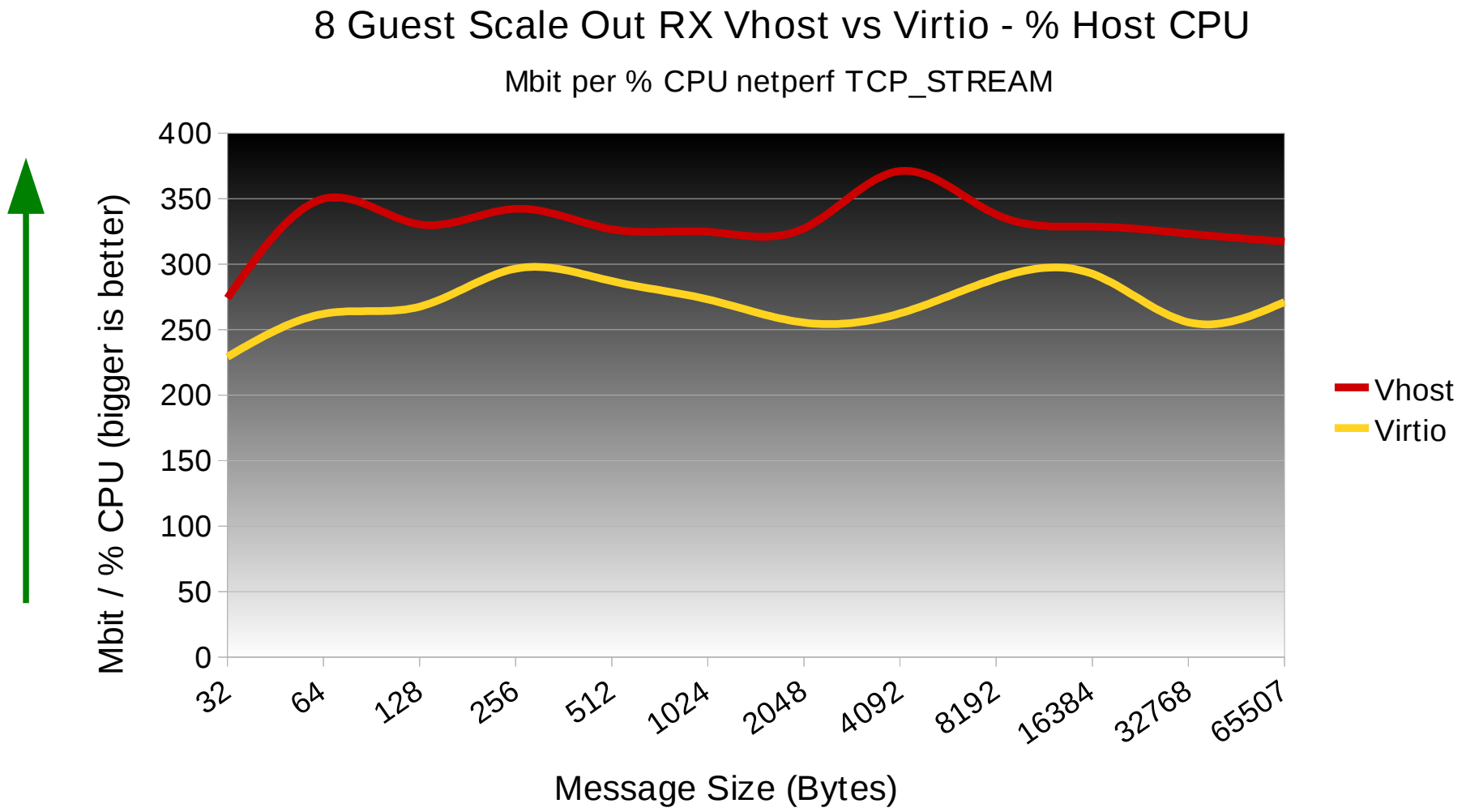
# Latency comparison



# Host CPU Consumption virtio vs vhost_net

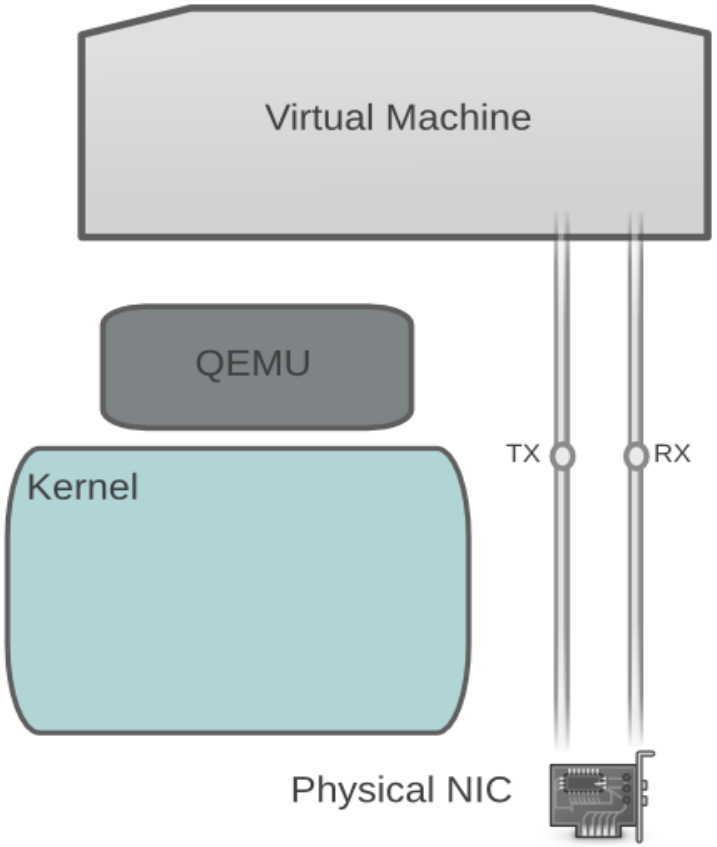


# vhost_net Efficiency

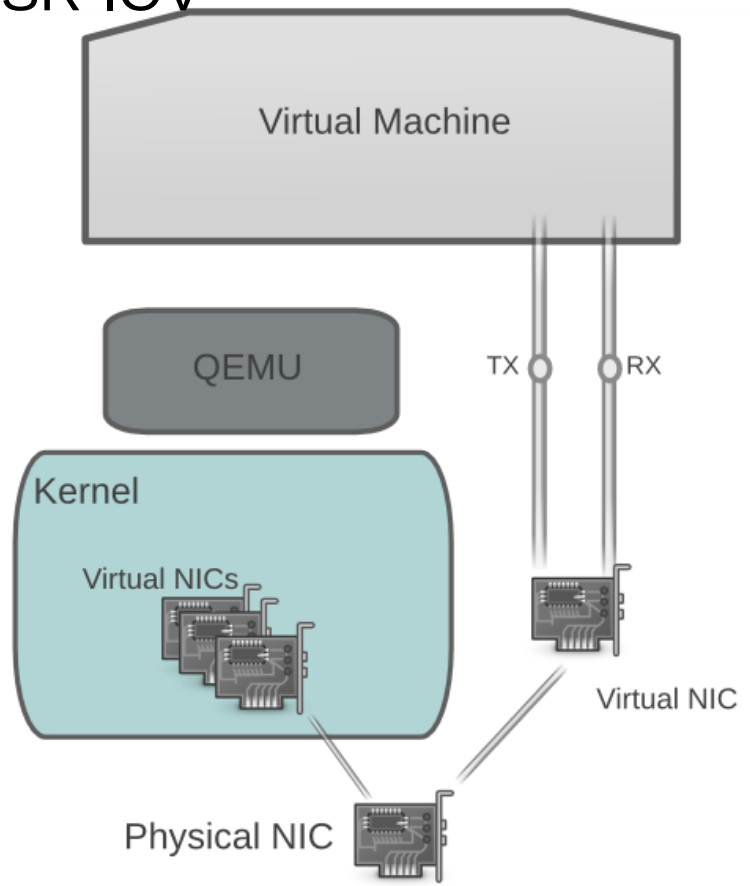


# KVM Architecture – Device Assignment vs SR/IOV

## Device Assignment




## SR-IOV



# KVM Network Architecture – PCI Device Assignment

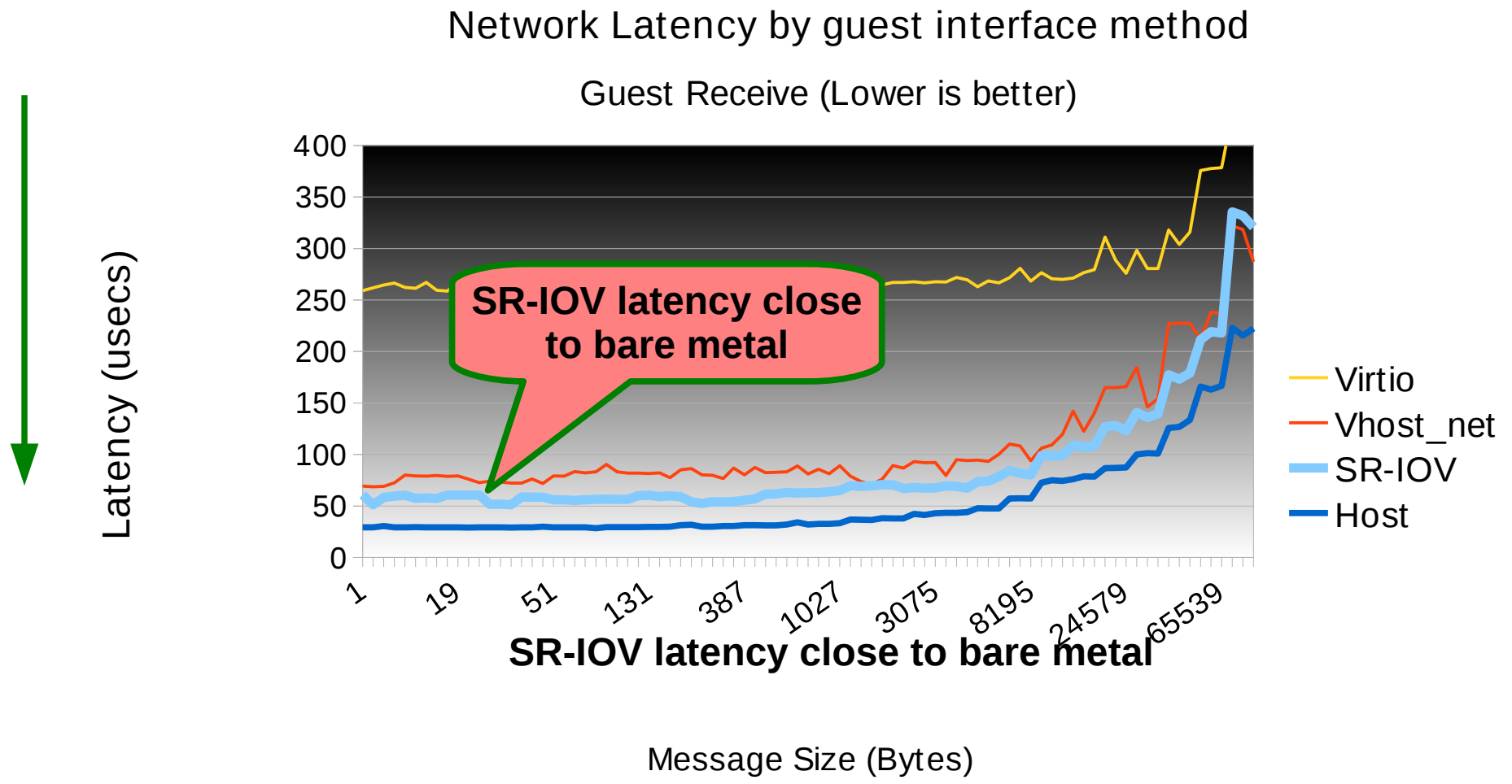
- Physical NIC is passed directly to guest
  - Device is not available to anything else on the host
- Guest sees real physical device
  - Needs physical device driver
- Requires hardware support
  - Intel VT-D or AMD IOMMU
- Lose hardware independence
- 1:1 mapping of NIC to Guest
- BTW - This also works on some I/O controllers

# KVM Network Architecture – SR-IOV

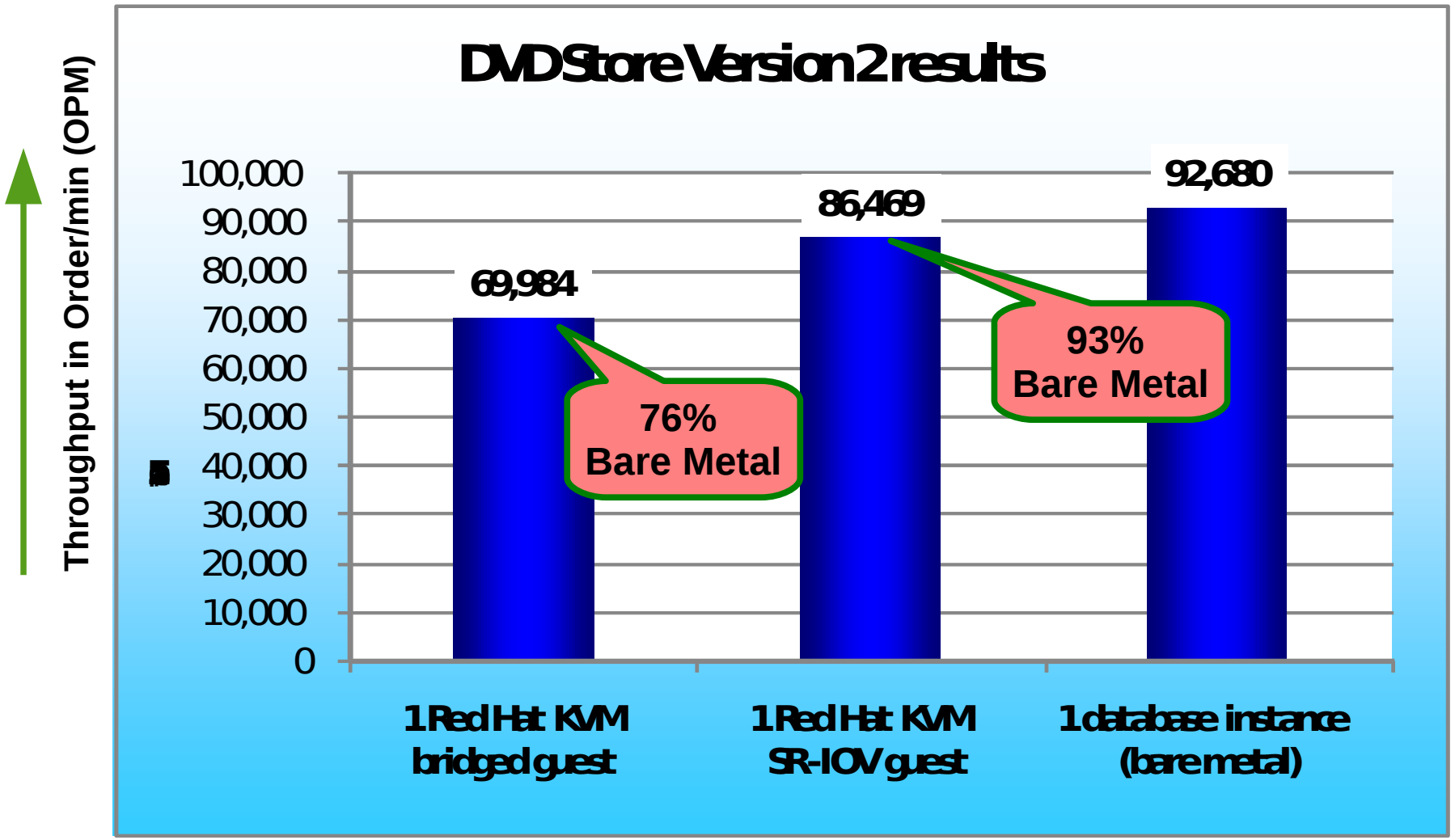
- Single Root I/O Virtualization

  New class of PCI devices that present multiple virtual devices that appear as regular PCI devices
- Guest sees real physical device
  - Needs physical (virtual) device driver
- Requires hardware  support
- Actual device can still be shared
- Low overhead, high throughput
- No live migration – well its difficult
- Lose hardware independence

# Latency comparison



# KVM w/ SR-IOV Intel Niantic 10Gb Postgres DB



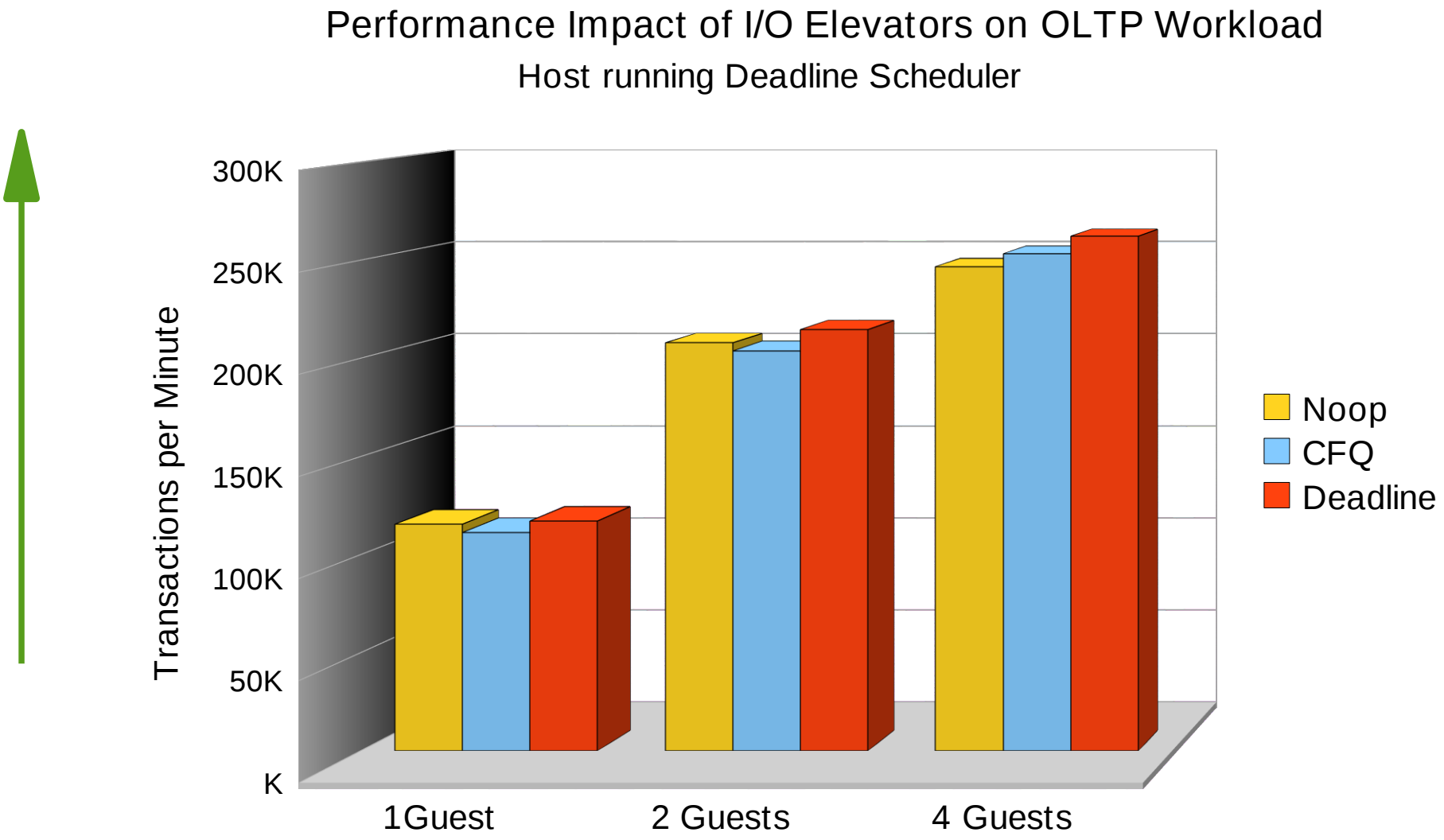
# I/O Tuning - Hardware

- Know your Storage
  - SAS or SATA?
  - Fibre Channel, Ethernet or SSD?
  - Bandwidth limits
- Multiple HBAs
  - Device-mapper-multipath
  - Provides multipathing capabilities and LUN persistence
- How to test
  - Low level I/O tools – dd, iozone, dt, etc

# I/O Tuning – Understanding I/O Elevators

- Deadline
  - Two queues per device, one for read and one for writes
  - IOs dispatched based on time spent in queue
- CFQ
  - Per process queue
  - Each process queue gets fixed time slice (based on process priority)
- Noop
  - FIFO
  - Simple I/O Merging
  - Lowest CPU Cost
- Can set at Boot-time
  - Grub command line – elevator=deadline/cfq/noop
- Or Dynamically – per device
  - echo “deadline” > /sys/class/block/sda/queue/scheduler

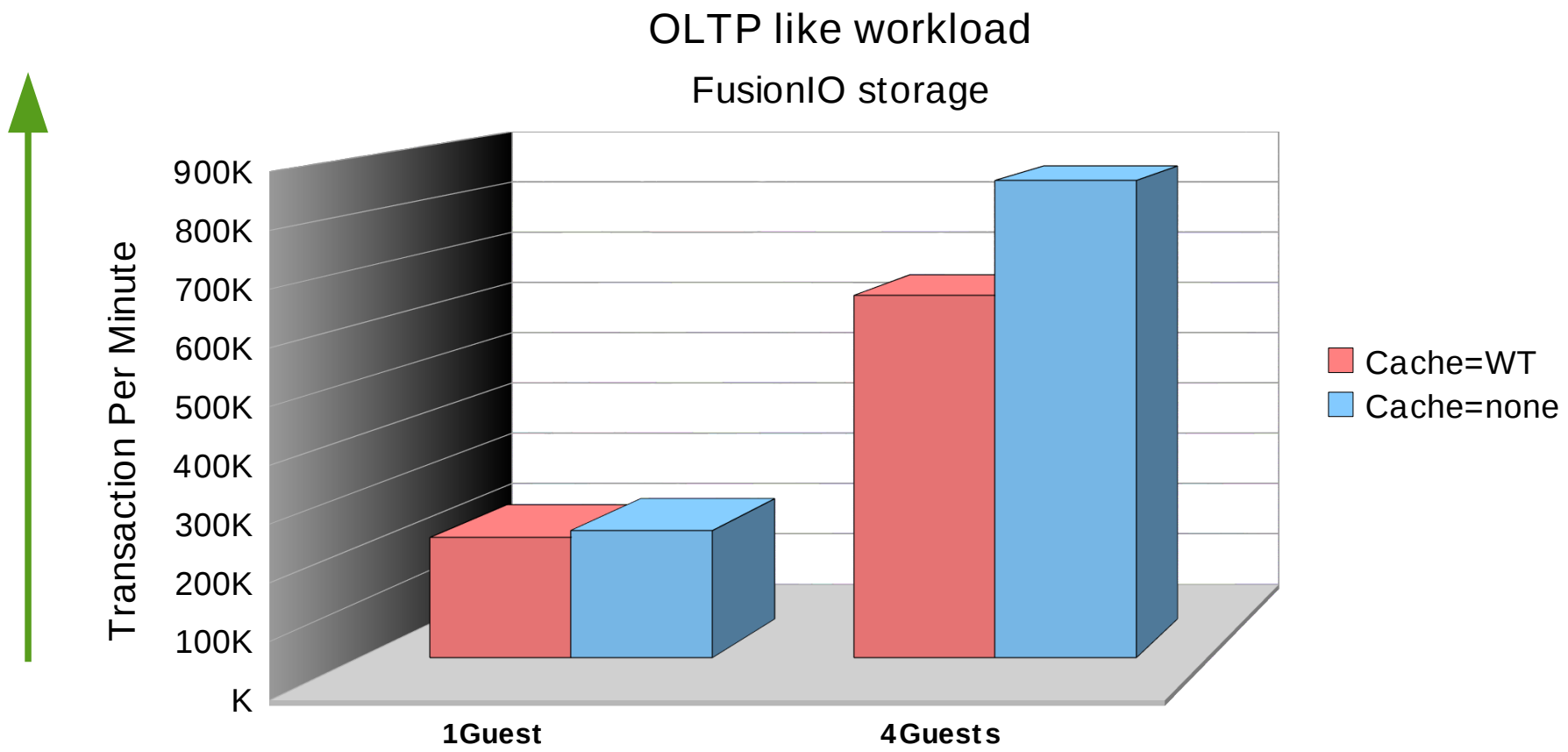
# Virtualization Tuning – I/O elevators - OLTP



# Virtualization Tuning - Caching

- Cache=none
  - I/O from the guest in not cached
- Cache=writethrough
  - I/O from the guest is cached and written through on the host
  - Potential scaling problems with this option with multiple guests (host cpu used to maintain cache)
- Cache=writeback - Not supported

# Effect of I/O Cache settings on Guest performance

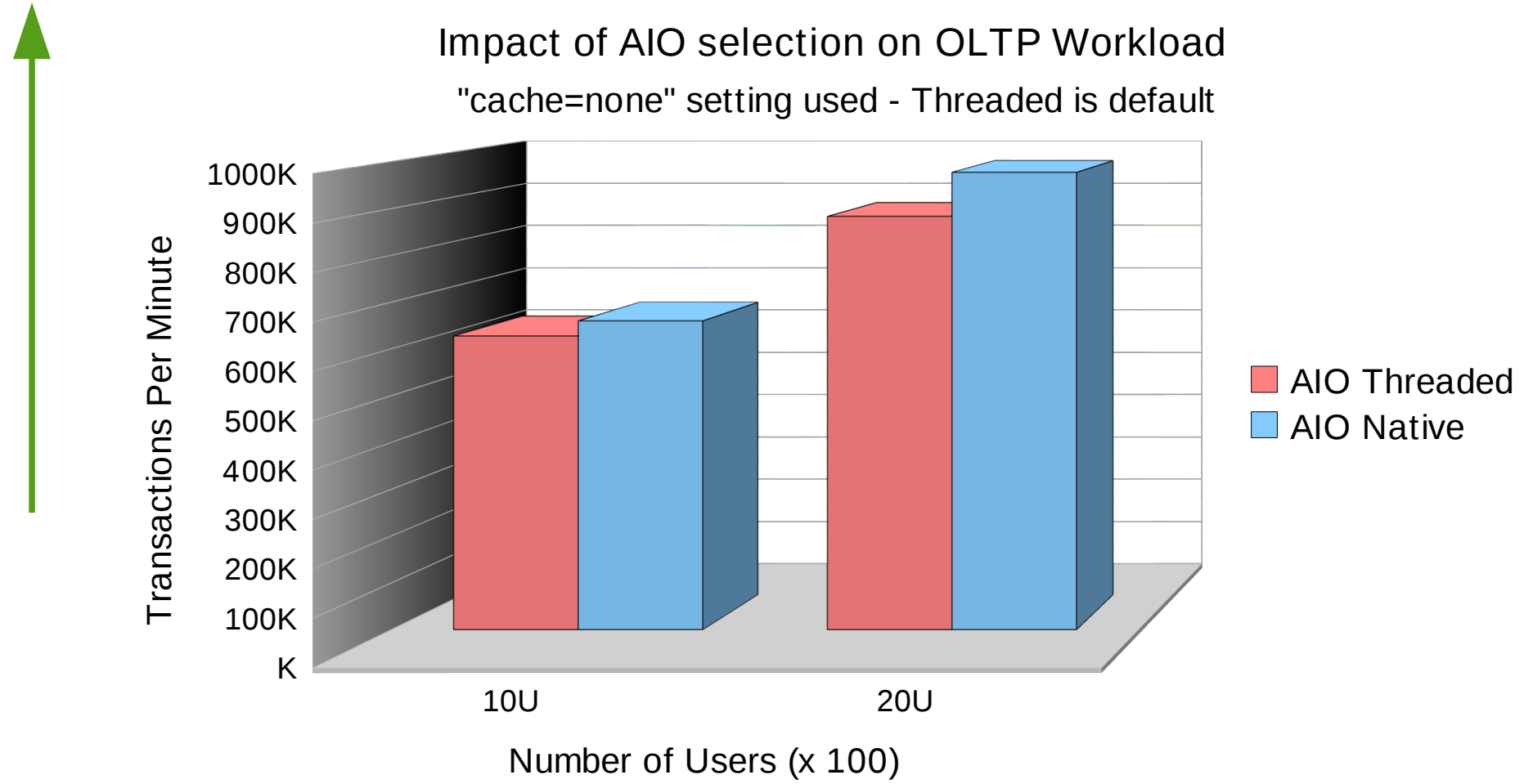


# I/O Tuning - Filesystems

- Configure read ahead
  - Database ( parameters to configure read ahead)
  - Block devices ( getra , setra )

- Asynchronous I/O
  - Eliminate Synchronous I/O stall
  - Critical for I/O intensive applications

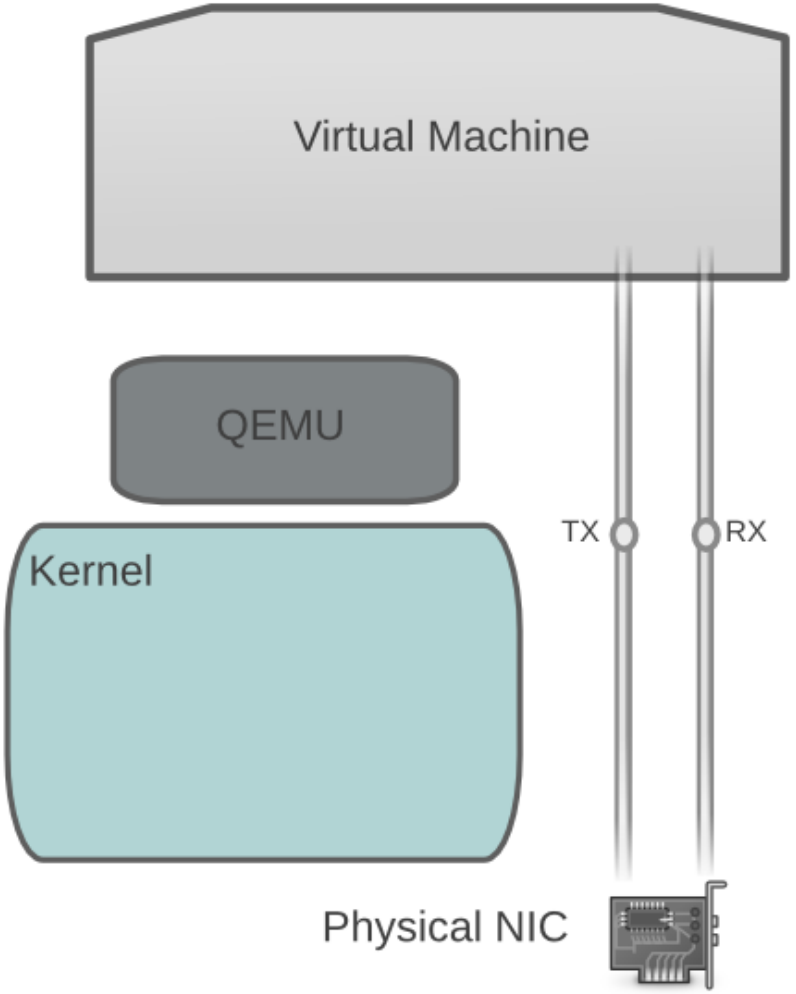
# AIO – Native vs Threaded (default)




**Configurable per device (only by xml configuration file)**

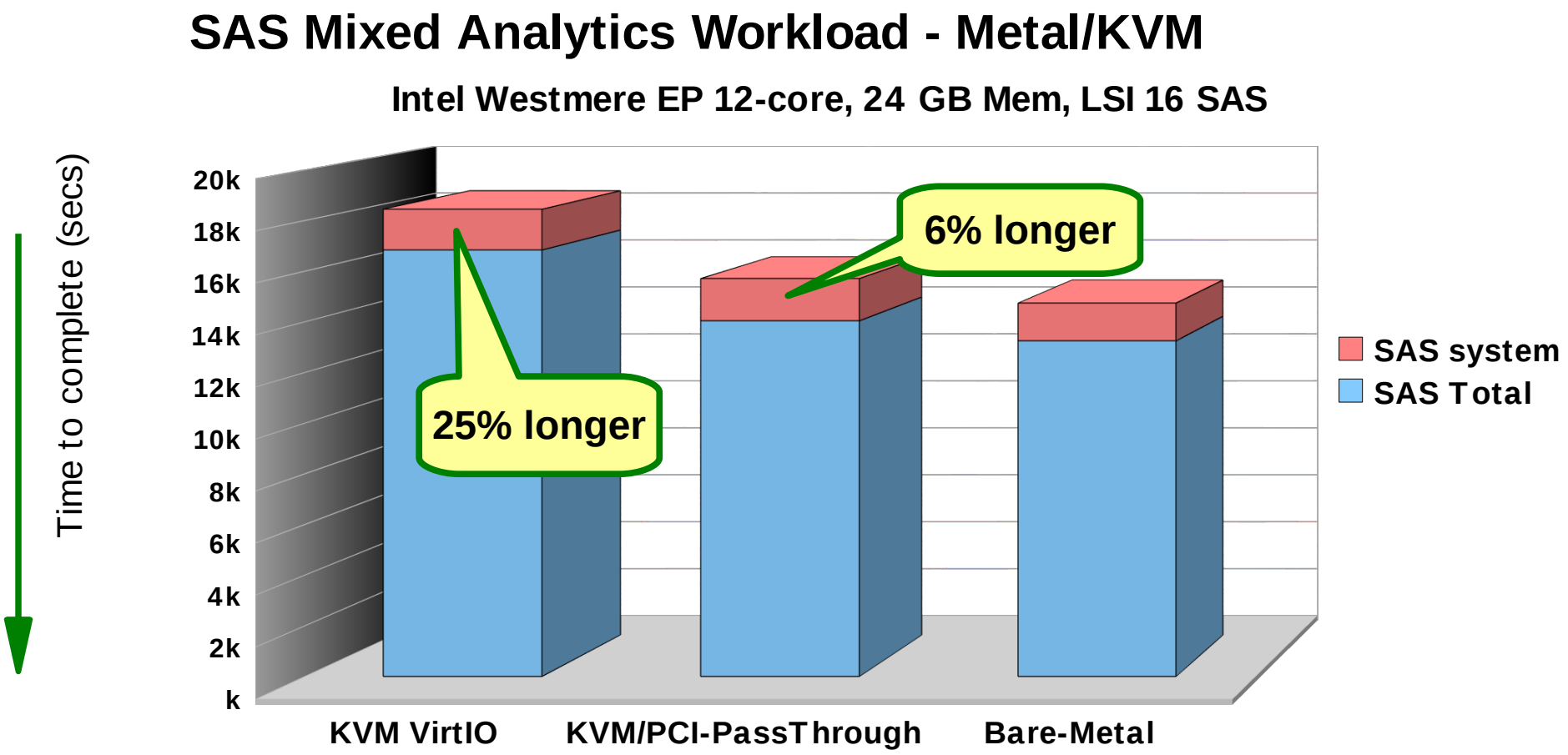
**Libvirt xml file - driver name='qemu' type='raw' cache='none' io='native'**

# Remember Network Device Assignment ?




- Device Assignment
  - It works for Block too !
  - Device Specific
  - Similar Benefits
  - And drawbacks...

- Block Device Passthrough - SAS Workload



# NUMA (Non Uniform Memory Access)

- Multi Socket – Multi core architecture
  - NUMA is needed for scaling
    - Keep memory latencies low
  - Linux completely NUMA aware
  - Additional performance gains by enforcing NUMA placement
  - Still some “out of the box” work is needed

- How to enforce NUMA placement
  - numactl – CPU and memory pinning
- One way to test if you get a gain is to **mistune** it.
- Libvirt now supports some NUMA placement

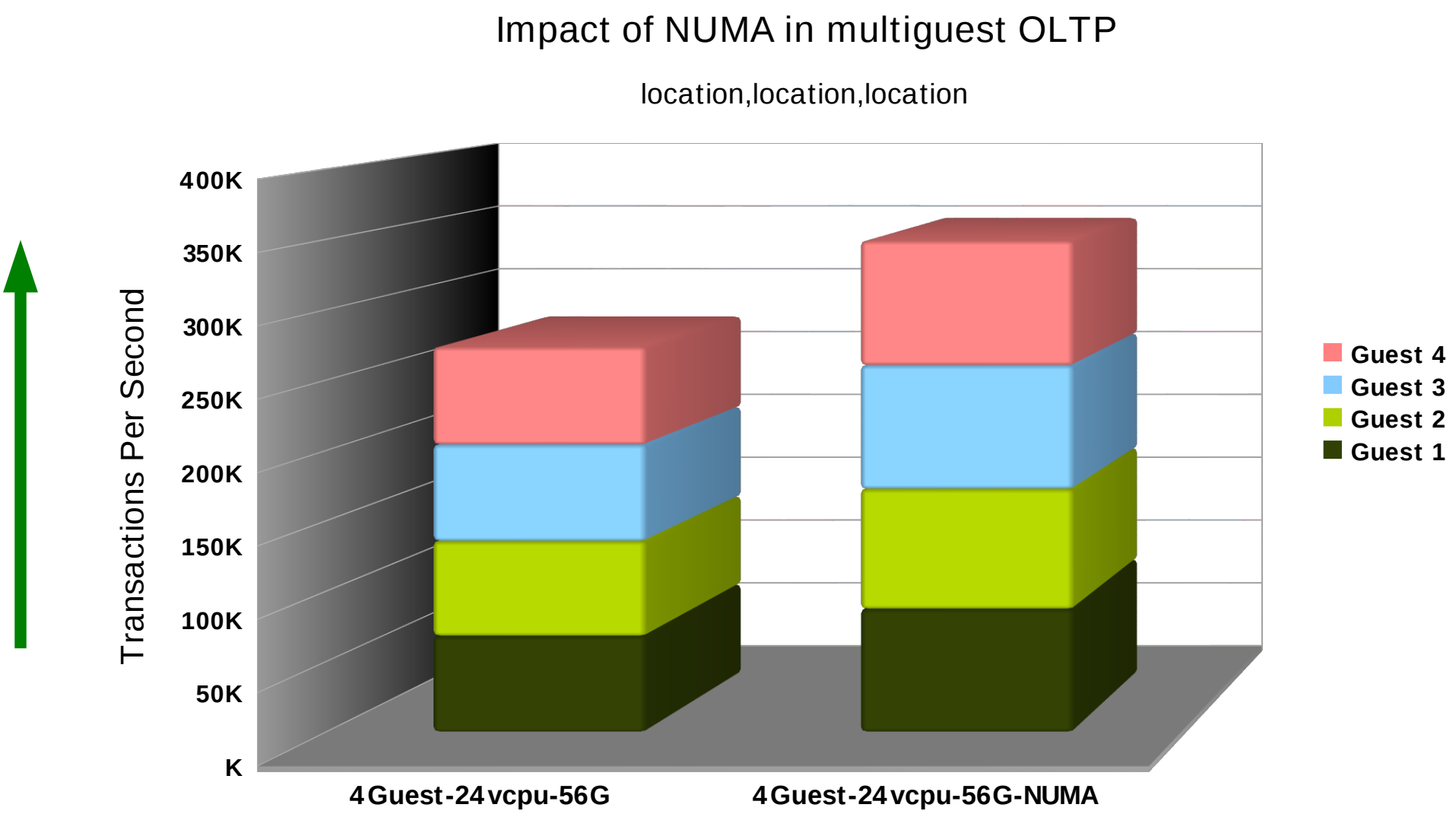
# Memory Tuning - NUMA

```
# numactl --hardware
available: 8 nodes (0-7)
node 0 cpus: 0 1 2 3 4 5
node 0 size: 8189 MB
node 0 free: 7220 MB
node 1 cpus: 6 7 8 9 10 11
node 1 size: 8192 MB
...
node 7 cpus: 42 43 44 45 46 47
node 7 size: 8192 MB
node 7 free: 7816 MB
node distances:
node   0   1   2   3   4   5   6   7
  0:  10  16  16  22  16  22  16  22
  1:  16  10  22  16  16  22  22  16
  2:  16  22  10  16  16  16  16  16
  3:  22  16  16  10  16  16  22  22
  4:  16  16  16  16  10  16  16  22
  5:  22  22  16  16  16  10  22  16
  6:  16  22  16  22  16  22  10  16
  7:  22  16  16  22  22  16  16  10
```

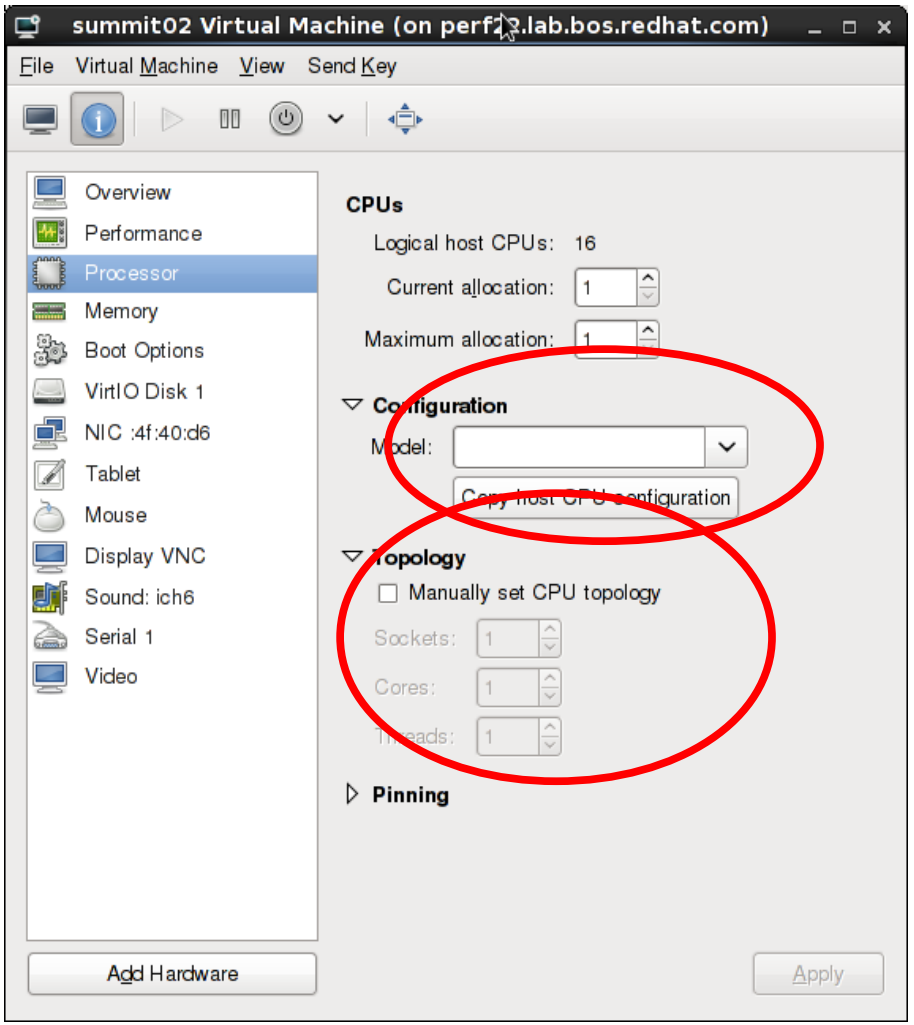
Internode Memory distance From SLIT table

Note variation in internode distances

# Virtualization Tuning – Using NUMA

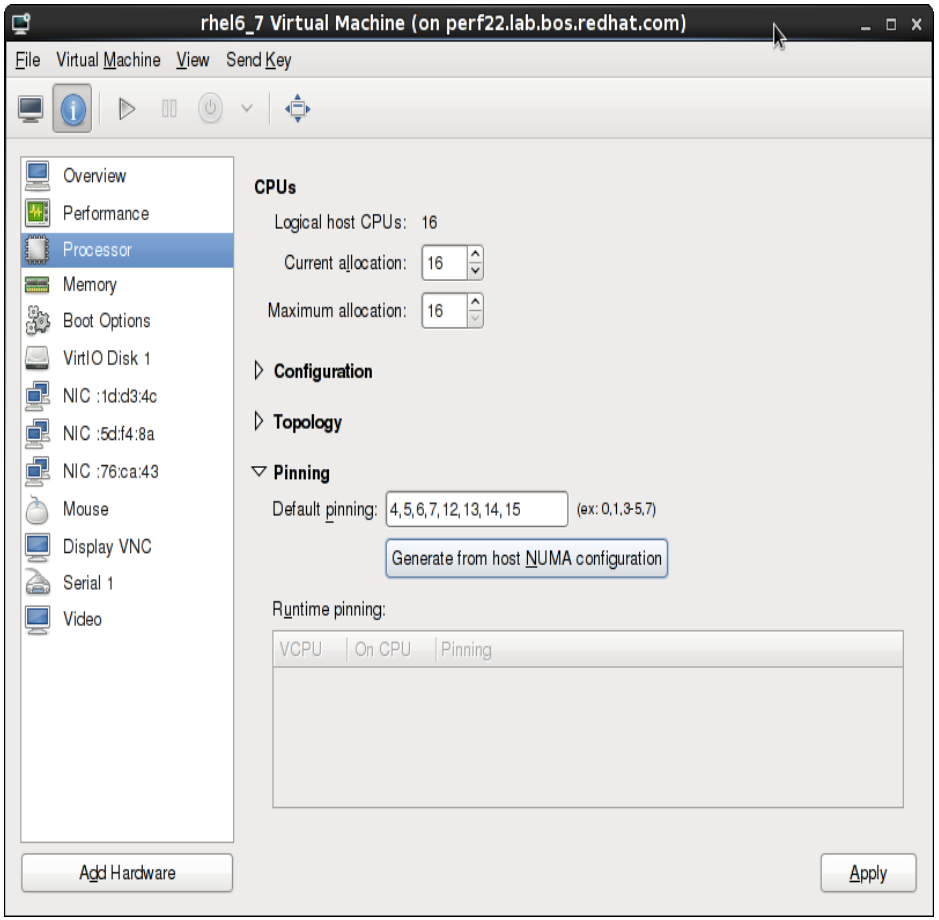


# Specifying Processor Details




- Mixed results with CPU type and topology
- The Red Hat team is still exploring some topology performance quirks
  - Both model and topology
- Experiment and see what works best in your case

# CPU Pinning - Affinity




- Virt-manager allows CPU selection based on NUMA topology
  - True NUMA support in libvirt
- Virsh pinning allows finer grain control
  - 1:1 pinning
- Good gains with pinning

# Performance monitoring tools

- Monitoring tools
  - top, vmstat, ps, iostat, netstat, sar, perf
- Kernel tools
  - /proc, sysctl, AltSysrq
- Networking
  - ethtool, ifconfig
- Profiling
  - oprofile, strace, ltrace, systemtap, perf

# Wrap up

- KVM can be tuned effectively
  - Understand what is going on under the covers
  - Turn off stuff you don't need
  - Be specific when you create your guest
  - Look at using NUMA or affinity
  - Choose appropriate elevators (Deadline vs CFQ)
  - Choose your cache wisely

# For More Information

- KVM Wiki
  - http://www.linux-kvm.org/page/Main_Page
- irc, email lists, etc
  - http://www.linux-kvm.org/page/Lists%2C_IRC
- libvirt Wiki
  - http://libvirt.org/
- New, revamped edition of the “Virtualization Guide”
  - http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/index.html
  - Should be available soon !