开源力量公开课

KVM性能优化

美团开放平台

邱剑

qiujian@meituan.com

https://mos.meituan.com

2013/10/22

关于美团开放平台

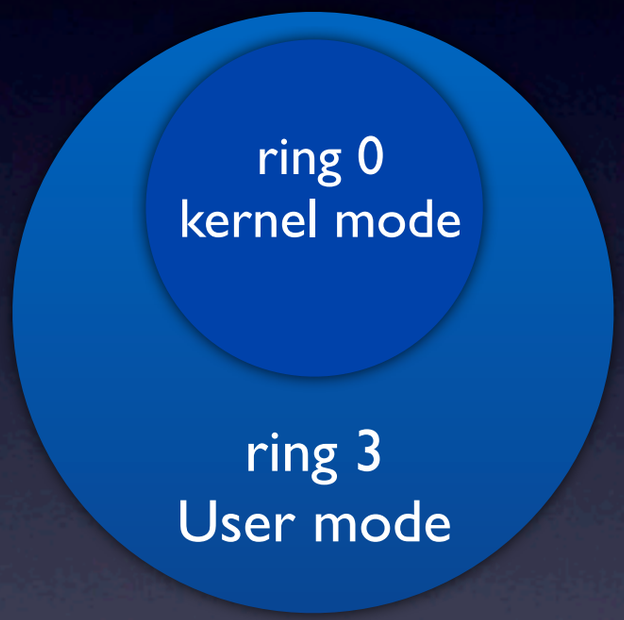
- 2012年初规划
- 基于OpenStack架构，部分组件自主开发
- 2012年9月开始逐步迁移在线服务系统到云主机
- 2013年5月推出美团开放服务(https://mos.meituan.com)，云主机为第一款产品
- 美团云主机基于KVM虚拟化技术

Agenda

- CPU
  - context switch
  - cache
- Memory
- IO
  - Storage
  - Network

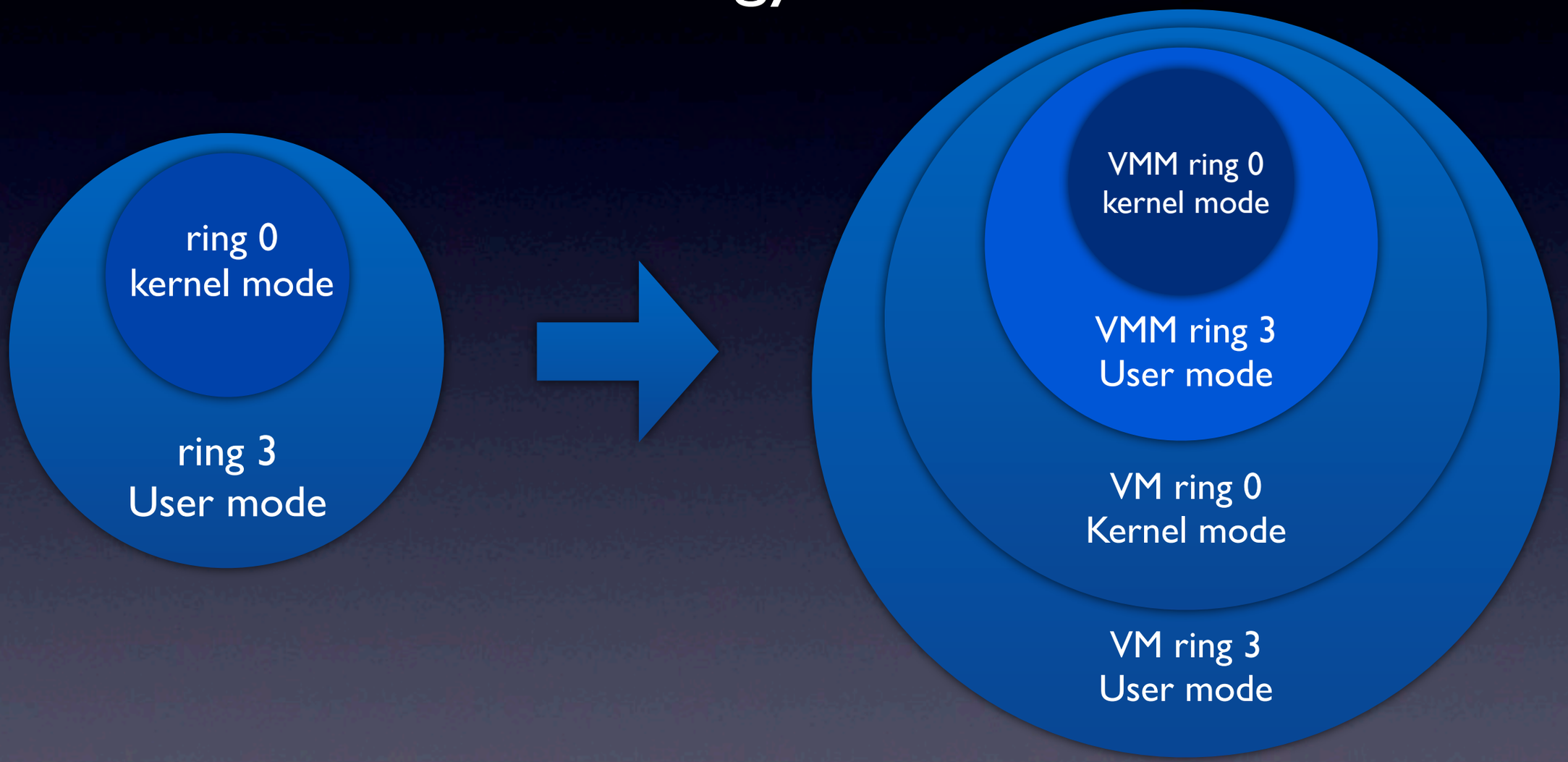
Context Switch - Intel VT-x

Virtualization Technology



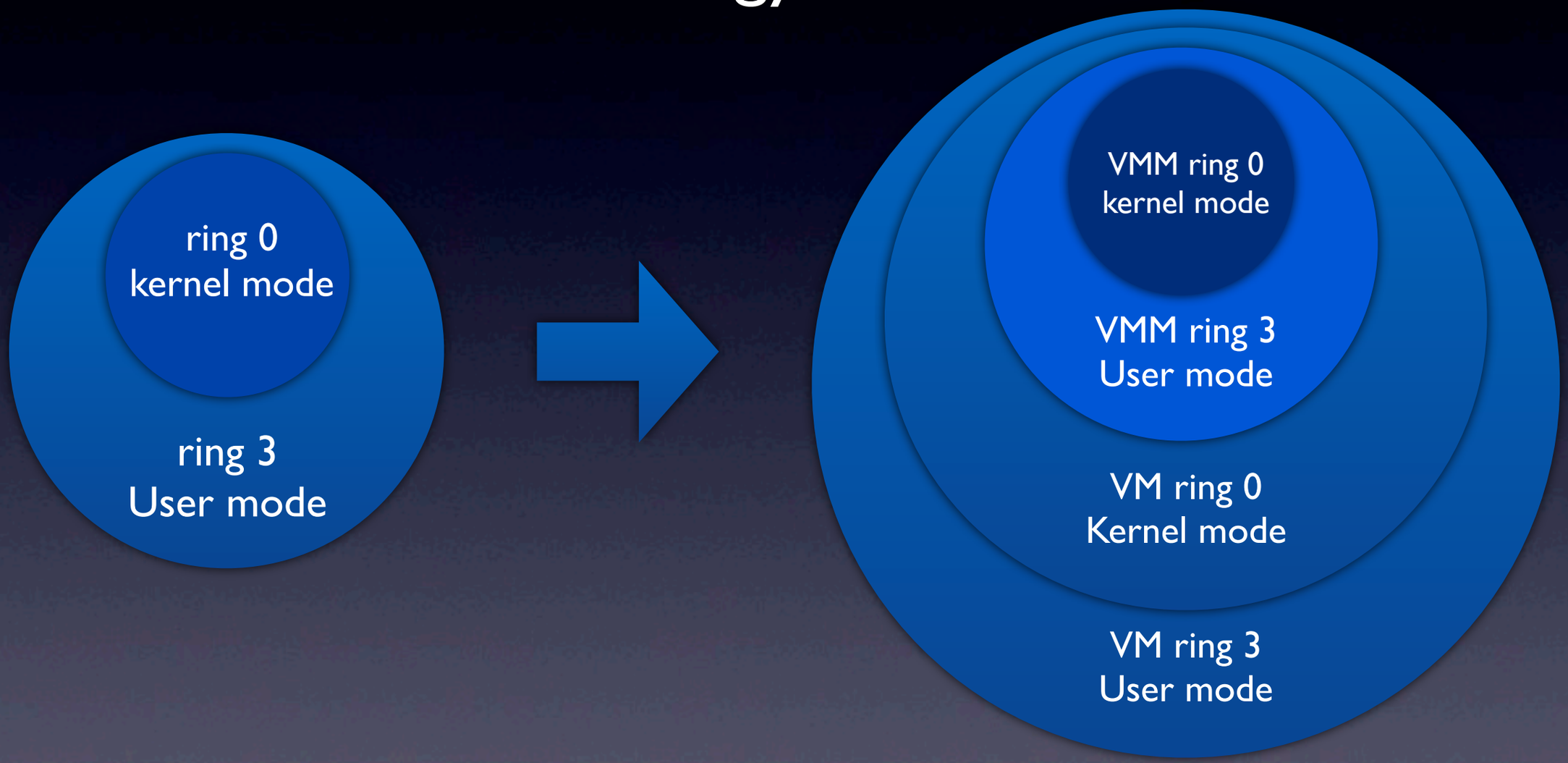
Context Switch - Intel VT-x

Virtualization Technology



Context Switch - Intel VT-x

Virtualization Technology



设置：宿主机BIOS中开启，目前默认开启

Cache - Node Binding

Cache - Node Binding

- 将qemu进程绑定到特定的CPU node或core上
  ——避免L2/L3 Cache miss

Cache - Node Binding

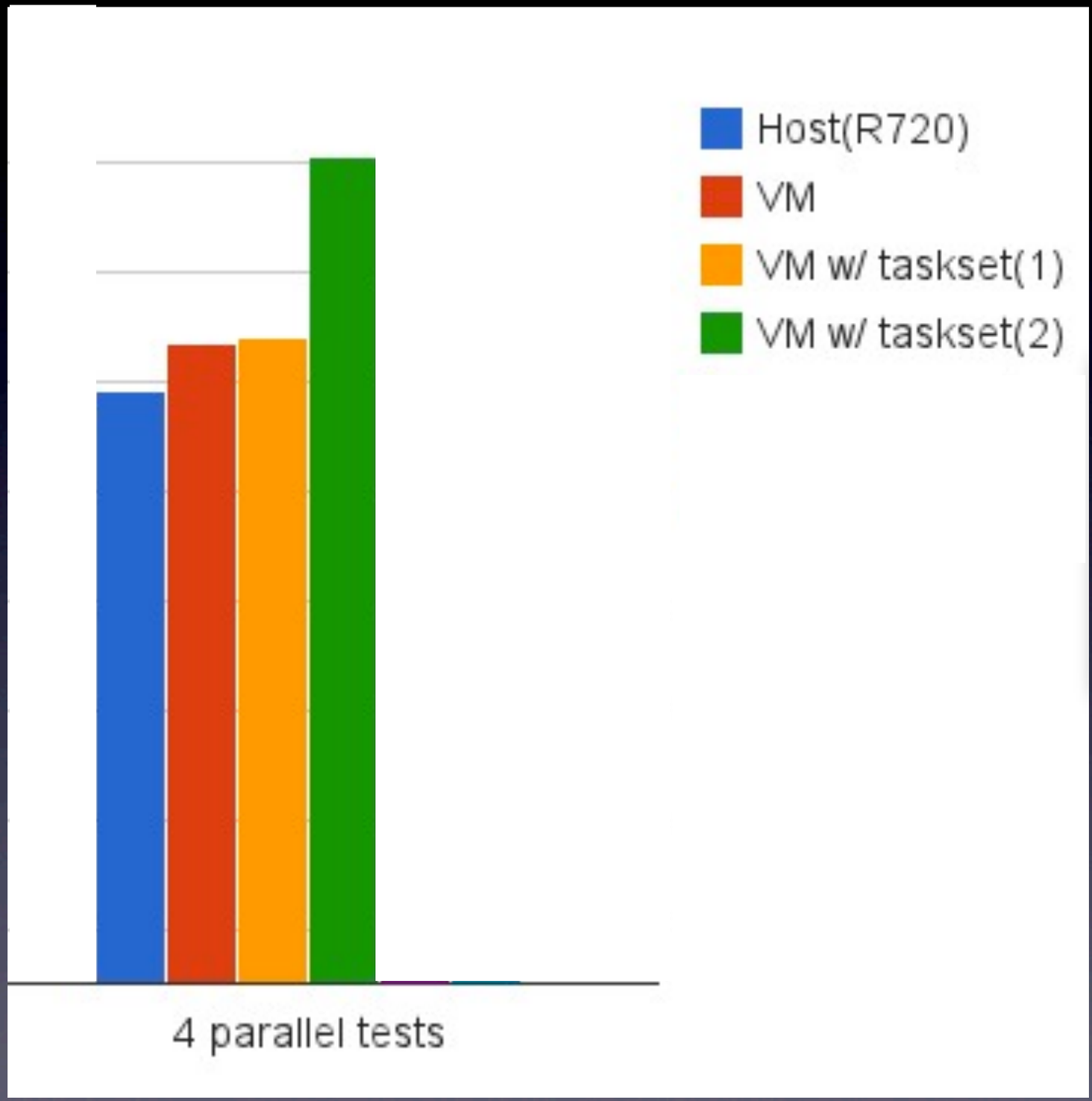
- 将qemu进程绑定到特定的CPU node或core上
  ——避免L2/L3 Cache miss
- Node binding v.s core binding

Cache - Node Binding

- 将qemu进程绑定到特定的CPU node或core上
  ——避免L2/L3 Cache miss
- Node binding v.s core binding
- 设置：

Cache - Node Binding

- 将qemu进程绑定到特定的CPU node或core上
  ——避免L2/L3 Cache miss
- Node binding v.s core binding
- 设置：
  - taskset

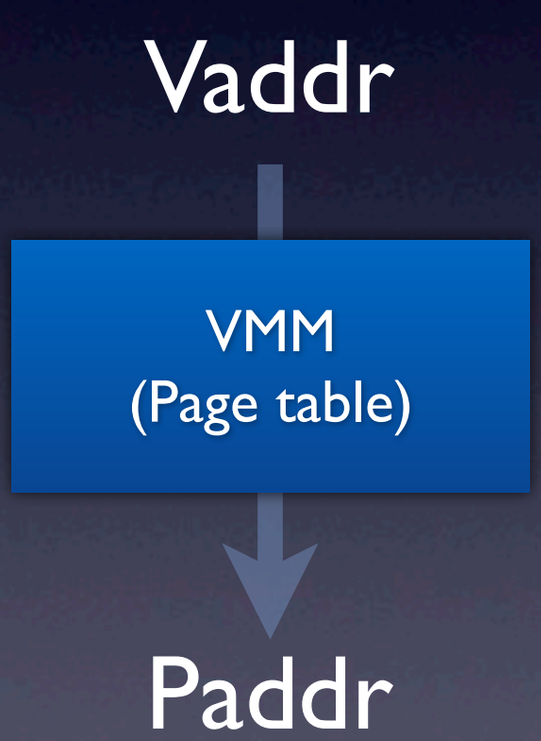

Host(R720)
VM
VM w/ taskset(1)
VM w/ taskset(2)
4 parallel tests

Agenda

- CPU
- Memory
  - Addressing
  - Space
- IO
  - Storage
  - Network

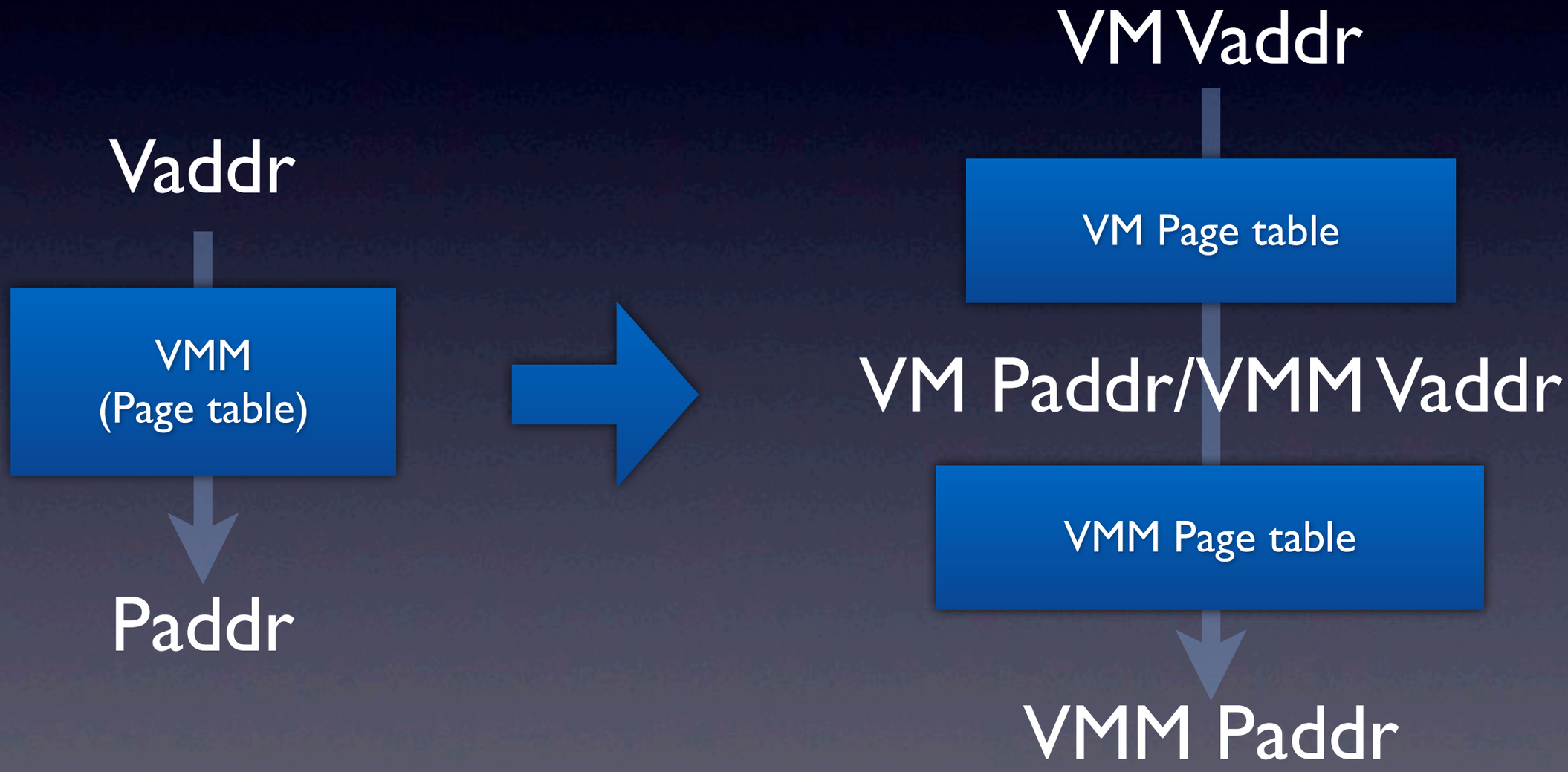
Addressing - EPT (SLAT)

- Extended page tables/second level address translation



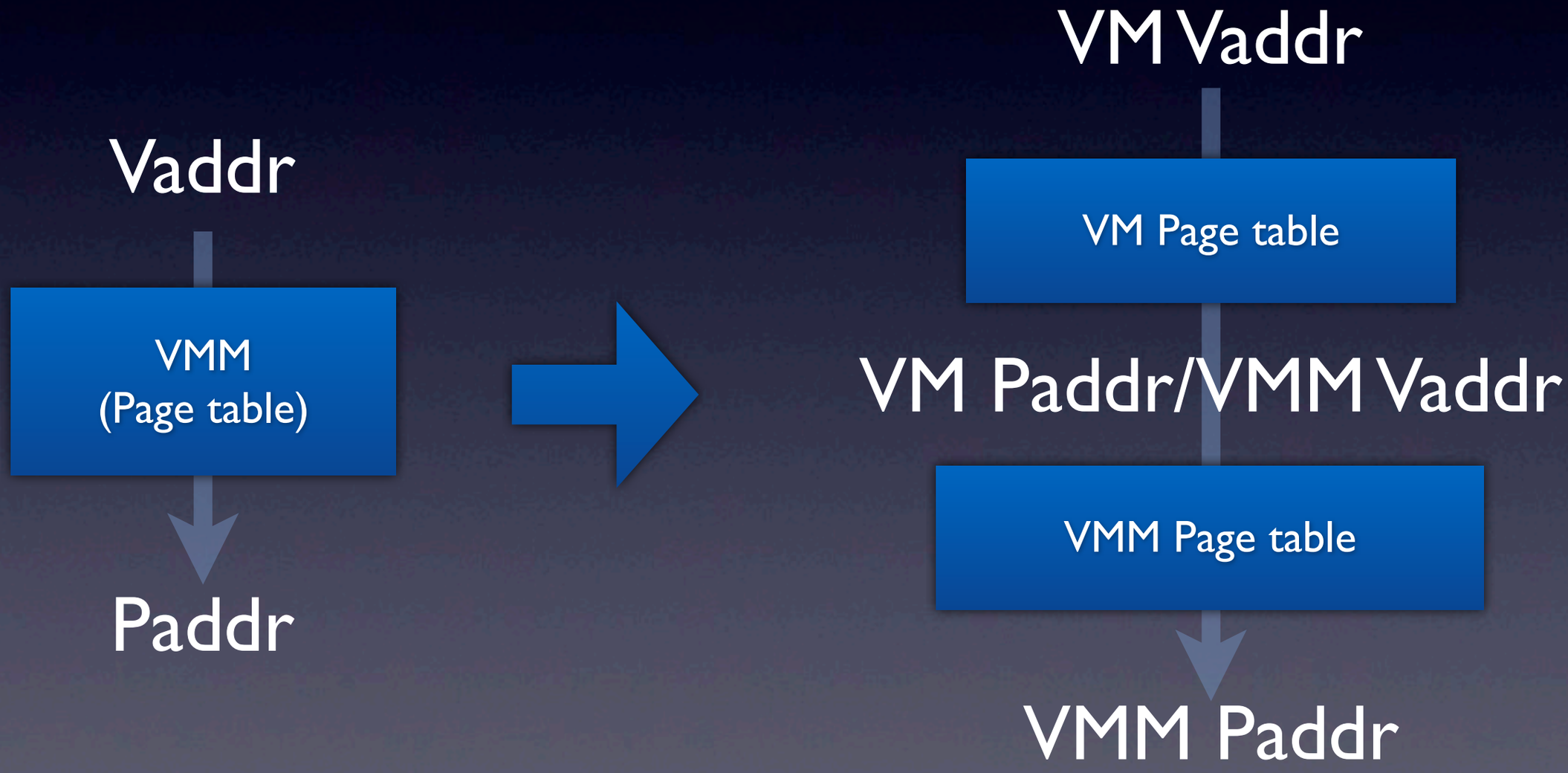
Addressing - EPT (SLAT)

- Extended page tables/second level address translation



Addressing - EPT (SLAT)

- Extended page tables/second level address translation



设置：宿主机BIOS中开启，目前默认开启

Addressing - HugePage

Addressing - HugePage

- 减少page table尺寸，降低查找缓存(TLB)的cache-miss，加速VM内存地址转换

Addressing - HugePage

- 减少page table尺寸，降低查找缓存(TLB)的cache-miss，加速VM内存地址转换
  - 默认Page size: 4KB/Hugepage size: 2M

Addressing - HugePage

- 减少page table尺寸，降低查找缓存(TLB)的cache-miss，加速VM内存地址转换
  - 默认Page size: 4KB/Hugepage size: 2M
- Transparent hugepage： kernel进程khugepaged周期性扫描内存，自动将地址连续可合并的普通4KB page合并为2MB Hugepage

Addressing - HugePage

- 减少page table尺寸，降低查找缓存(TLB)的cache-miss，加速VM内存地址转换
  - 默认Page size: 4KB/Hugepage size: 2M
- Transparent hugepage： kernel进程khugepaged周期性扫描内存，自动将地址连续可合并的普通4KB page合并为2MB Hugepage
  - 设置：

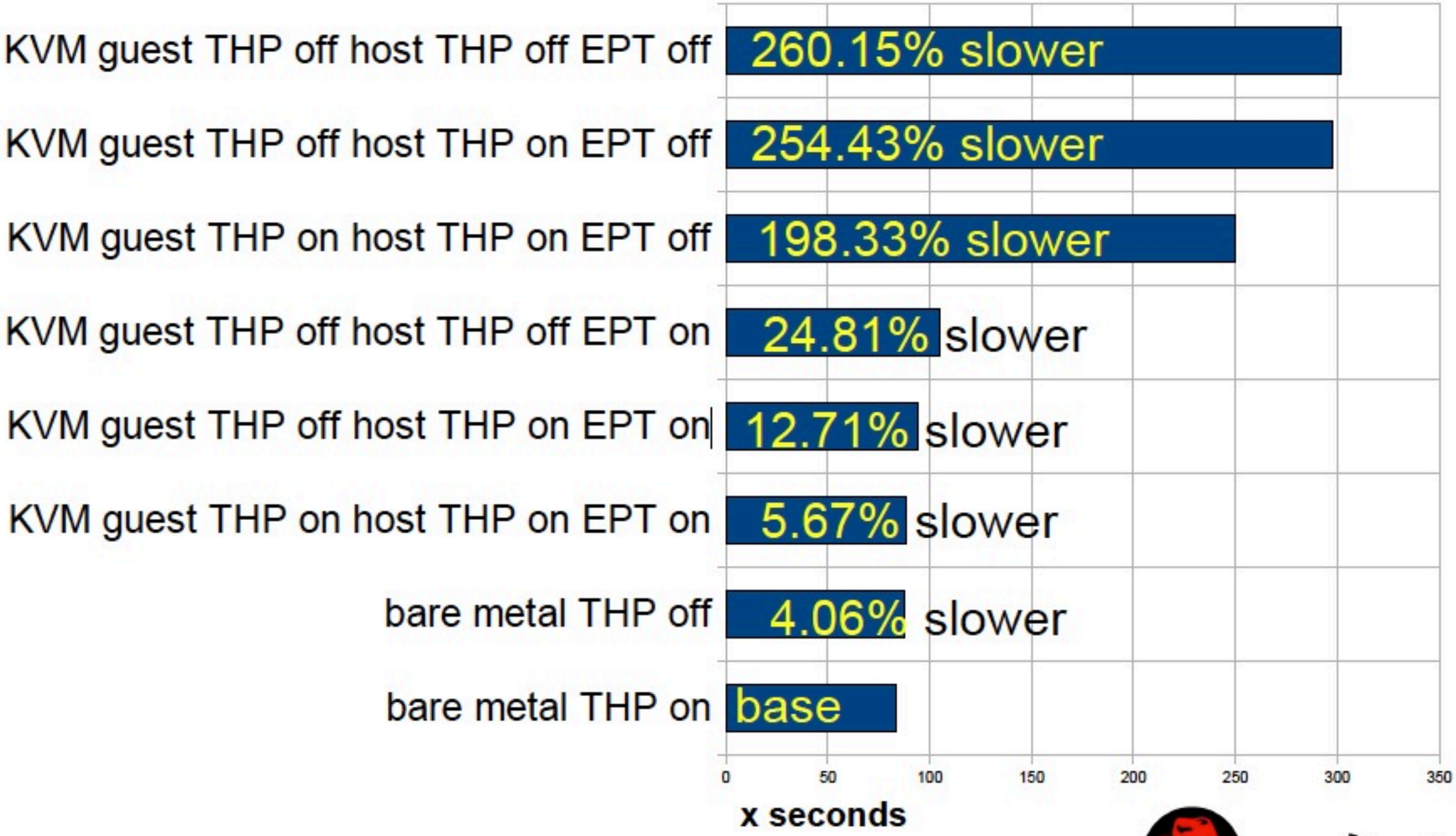
Addressing - HugePage

- 减少page table尺寸，降低查找缓存(TLB)的cache-miss，加速VM内存地址转换
  - 默认Page size: 4KB/Hugepage size: 2M
- Transparent hugepage： kernel进程khugepaged周期性扫描内存，自动将地址连续可合并的普通4KB page合并为2MB Hugepage
  - 设置：
    - sysctl -w sys.kernel.mm.transparent_hugepage.enabled=always

Addressing - HugePage

- 减少page table尺寸，降低查找缓存(TLB)的cache-miss，加速VM内存地址转换
  - 默认Page size: 4KB/Hugepage size: 2M
- Transparent hugepage：kernel进程khugepaged周期性扫描内存，自动将地址连续可合并的普通4KB page合并为2MB Hugepage
  - 设置：
    - sysctl -w sys.kernel.mm.transparent_hugepage.enabled=always
    - sysctl -w sys.kernel.mm.transparent_hugepage.defrag=always

kbuild bench (shorter is better)



Copyright © 2010 Red Hat Inc.

redhat.

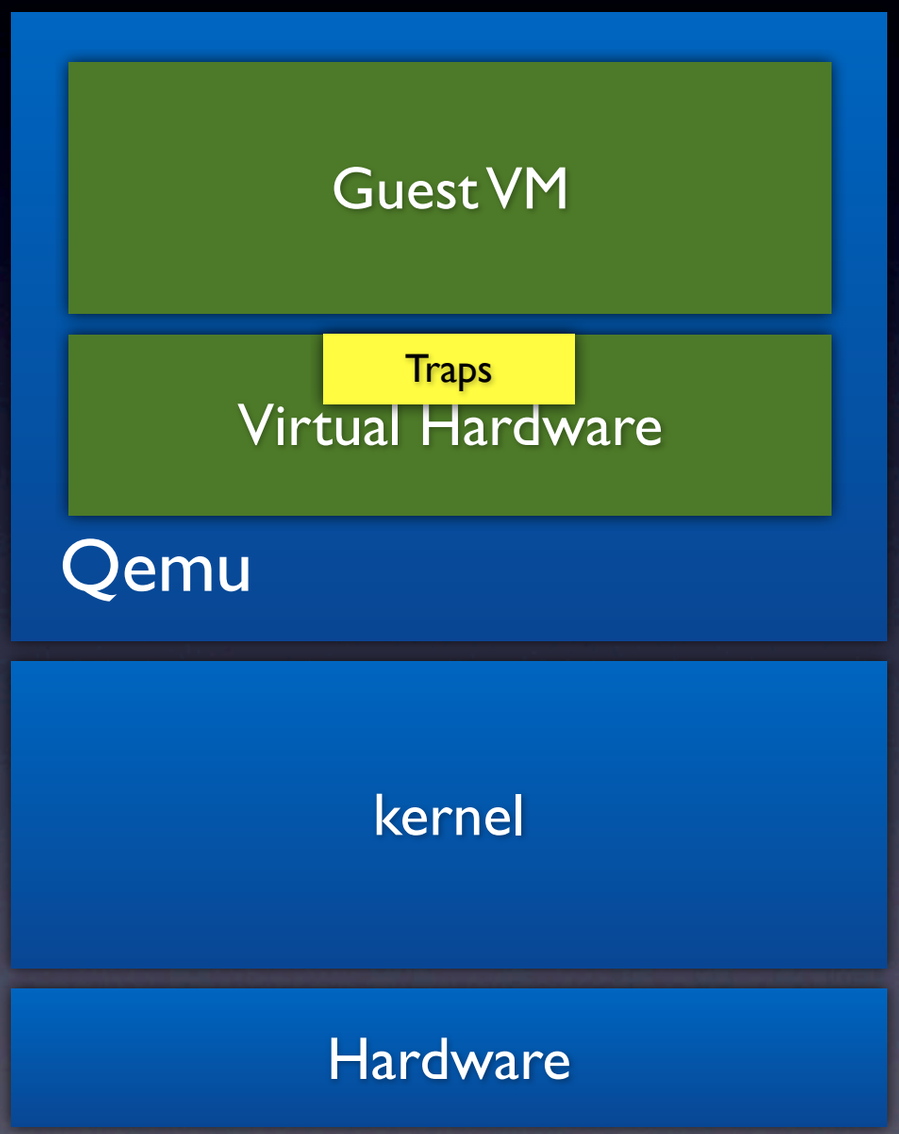
Space - KSM

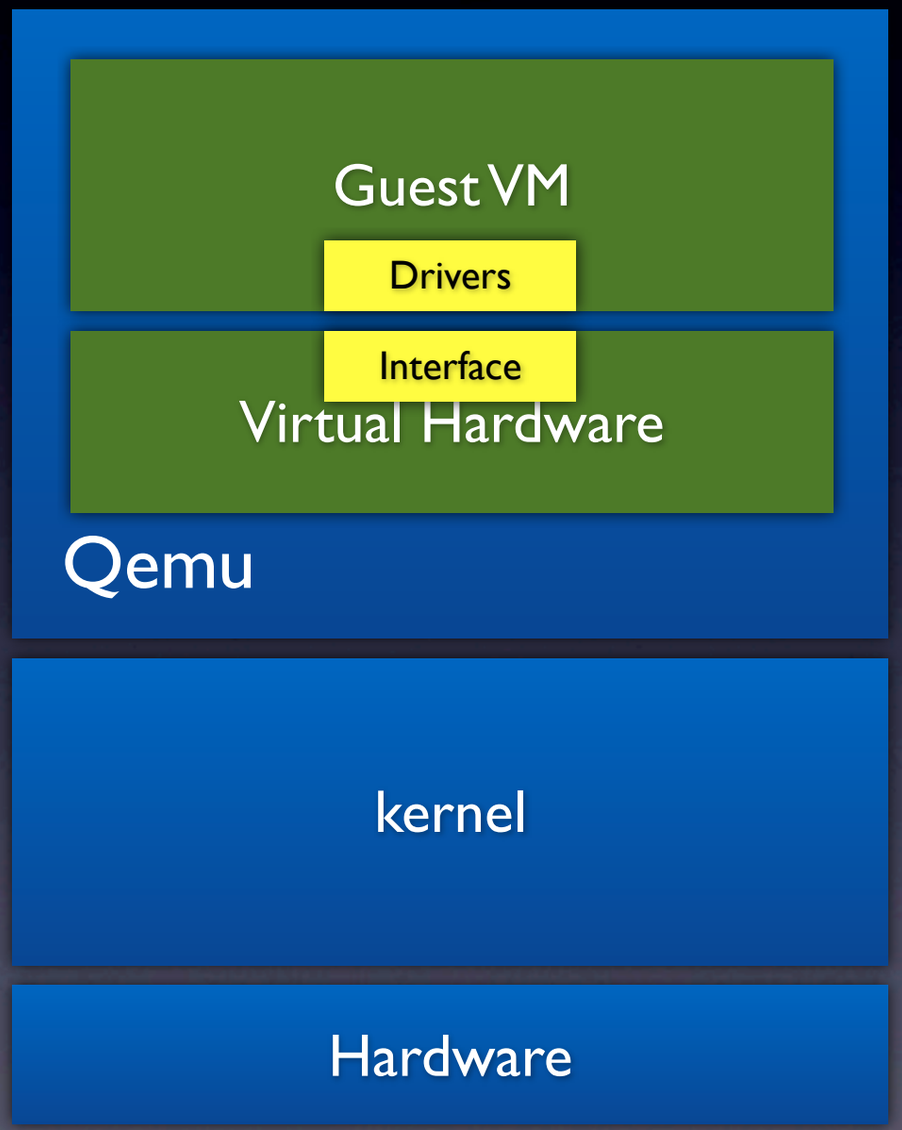
- Kernel same-page merging
- kernel进程ksmd周期性扫描内存，将内容相同的page合并，减少物理内存使用量

Agenda

- CPU
- Memory
- IO
  - Storage
  - Network

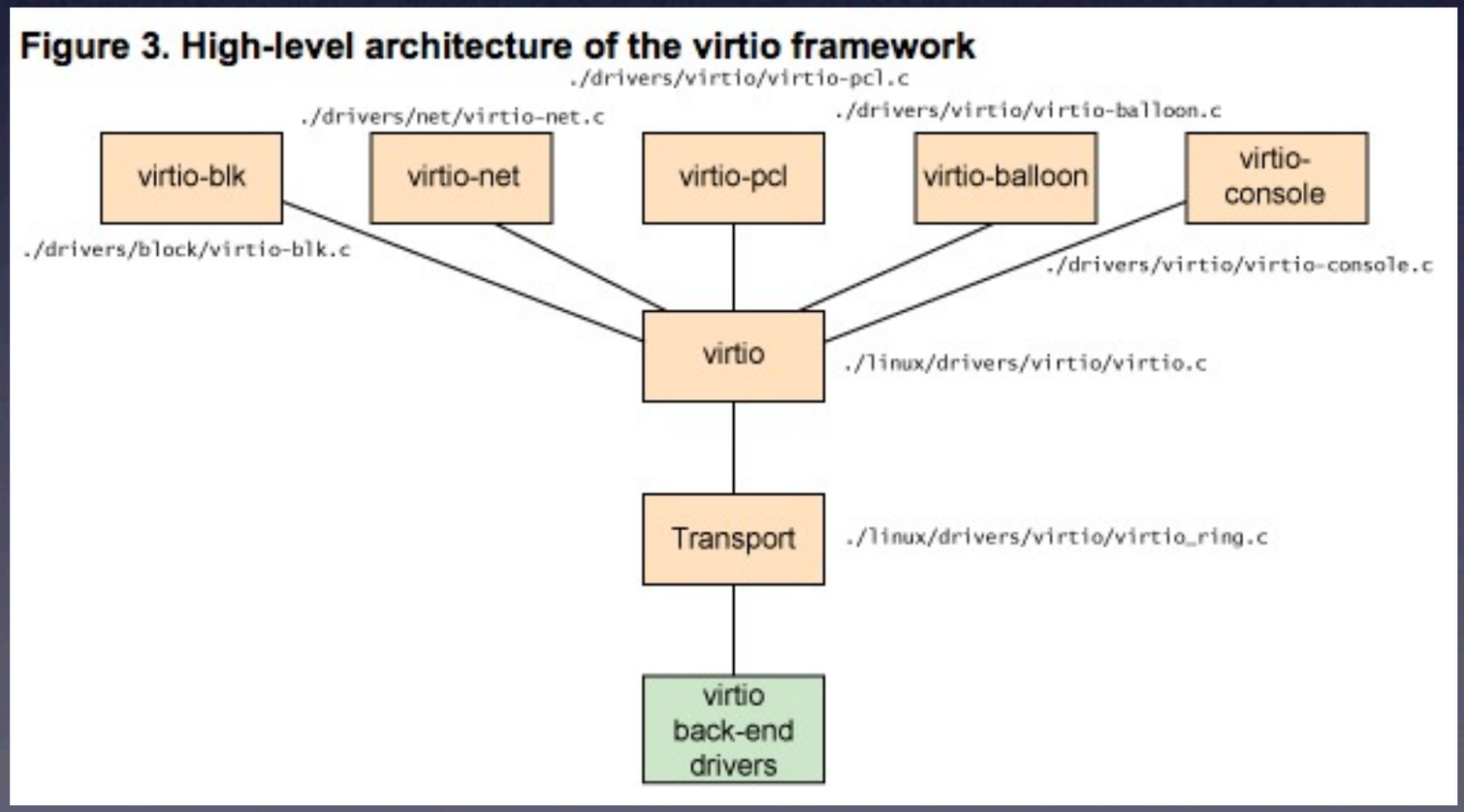
Full virtualization v.s. paravirtualization





virtio

半虚拟化I/O设备框架，标准化guest与host之间数据交换接口，简化流程，减少内存拷贝，提升虚拟机I/O效率



Agenda

- CPU
- Memory
- IO
  - Storage
  - Network

virtio-blk

- 基于virtio框架的虚拟PCI磁盘设备
- /dev/vdx

virtio-blk

- 基于virtio框架的虚拟PCI磁盘设备
- /dev/vdx

```
-drive file=win_xp.img,if=none,id=drive_0,cache=none,aio=native
-device virtio-blk-pci,drive=drive_0,bus=pci.0,addr=0x5
```

virtio-SCSI

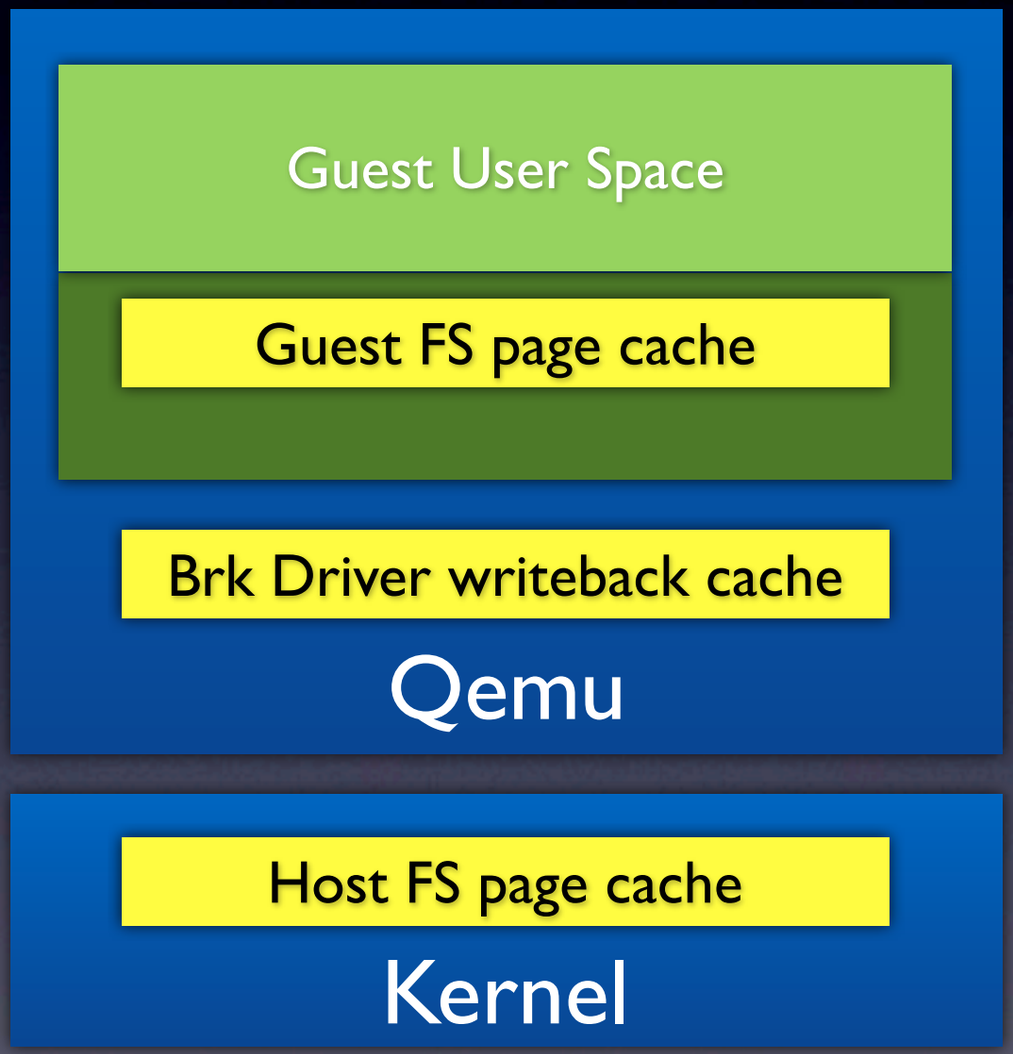
- 基于virtio框架的虚拟SCSI磁盘设备
- /dev/sdx

virtio-SCSI

- 基于virtio框架的虚拟SCSI磁盘设备
- /dev/sdx

```
-drive file=win_xp.img,if=none,id=drive_0,cache=none,aio=native
-device virtio-scsi-pci,drive=drive_0,bus=pci.0,addr=0x5
```

缓存模式


Guest User Space
Guest FS page cache
Brk Driver writeback cache
Qemu
Host FS page cache
Kernel

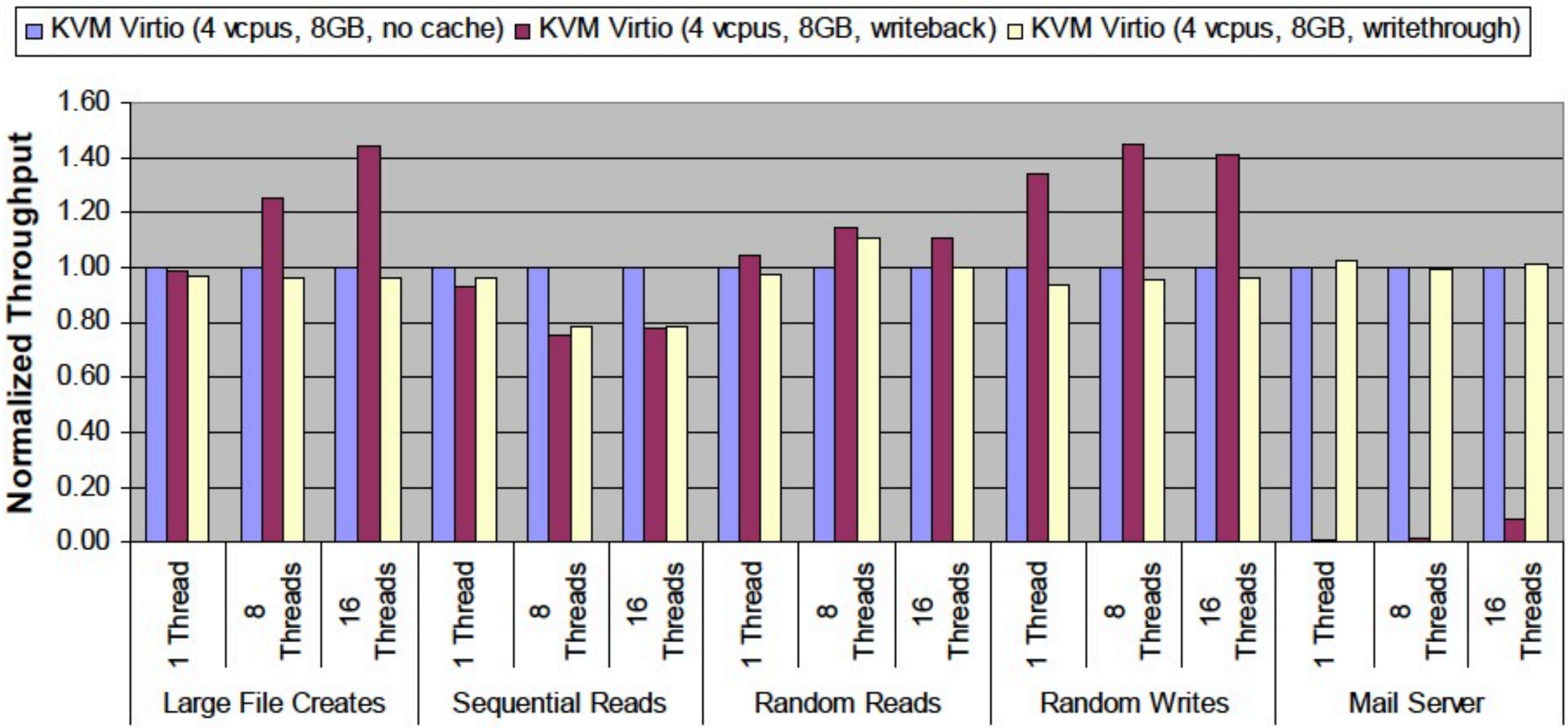
缓存模式(cont)

	page cache	writeback cache	写同步 (flush)	说明
directsync	NO	NO	N/A	无优化
writethrough	YES	NO	YES	依靠Host操作系统优化IO性能
none/off	NO	YES	N/A	关闭Host page cache. 优化写性能，并保证安全性
writeback	YES	YES	YES	优化读写性能，可能丢失数据
unsafe	YES	YES	NO	优化读写行呢干，不保证数据安全

```
-drive file=win_xp.img,if=none,id=drive_0,cache=none,aio=native
-device virtio-blk-pci,drive=drive_0,bus=pci.0,addr=0x5
```

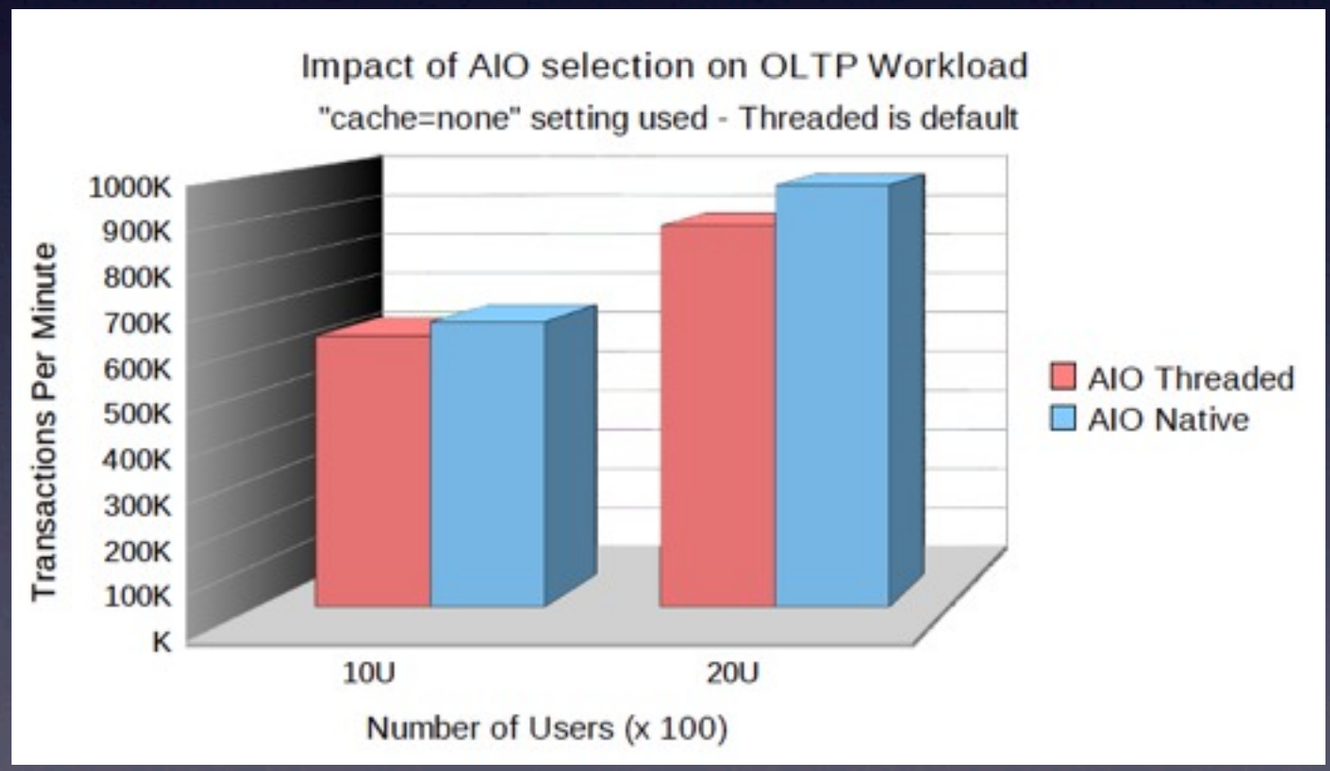
KVM Block I/O Performance - Impact of KVM Caching on Direct-Attached Storage

File System = ext3; I/O Block Size = 8KB; LVM Volume on 8 x DS3400 Disk Arrays



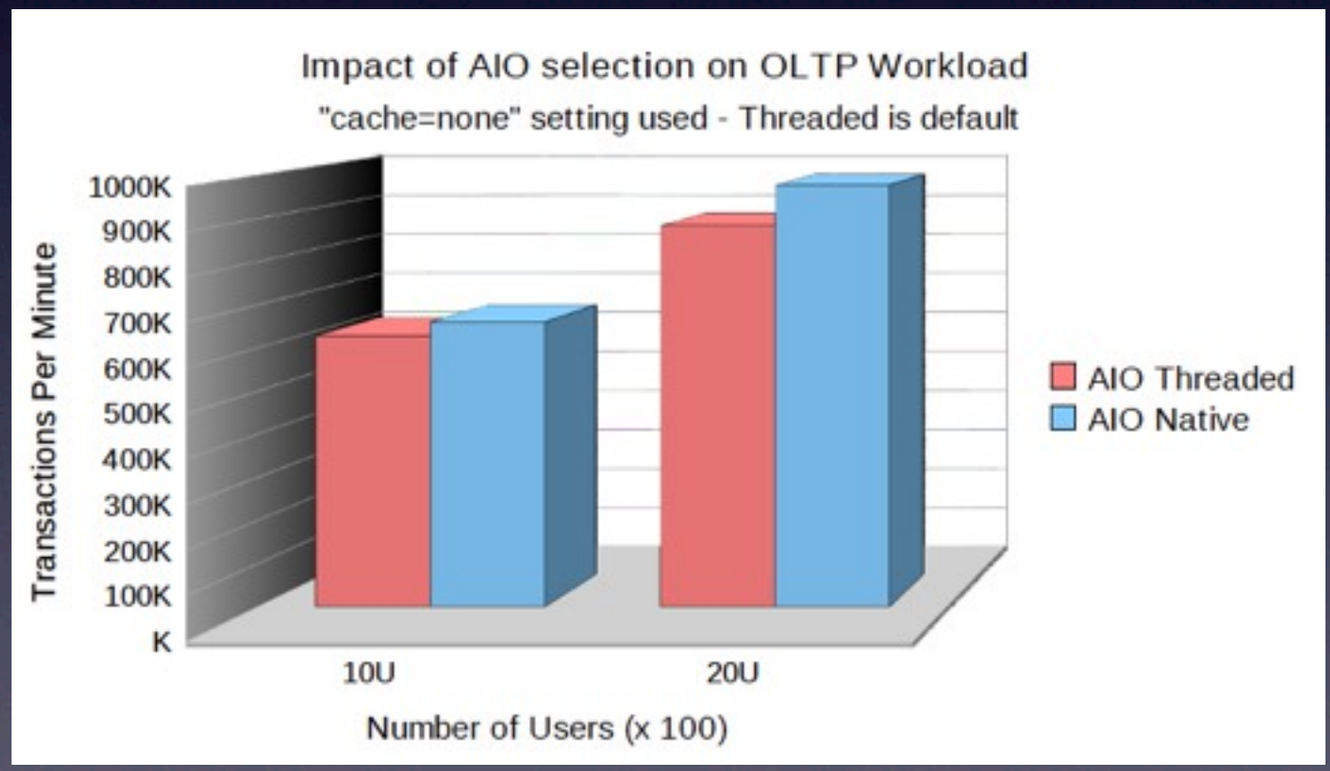
Native AIO

- Native aio: kernel AIO
- threaded aio: user space AIO emulated by posix thread workers



Native AIO

- Native aio: kernel AIO
- threaded aio: user space AIO emulated by posix thread workers




```
-drive file=win_xp.img,if=none,id=drive_0,cache=none,aio=native
-device virtio-blk-pci,drive=drive_0,bus=pci.0,addr=0x5
```

块设备IO调度器

cfq	per-process IO queue	较好公平性 较低aggregate throughput
deadline	per-device IO queue	较好实时性，较好aggregate throughput 不够公平，容易出现VM starvation

块设备IO调度器

cfq	per-process IO queue	较好公平性 较低aggregate throughput
deadline	per-device IO queue	较好实时性，较好aggregate throughput 不够公平，容易出现VM starvation

```
sysctl -w sys.block.sdb.queue.scheduler=cfq
```

Agenda

- CPU
- Memory
- IO
  - Storage
  - Network

virtio-net

- 基于virtio框架的虚拟以太网设备

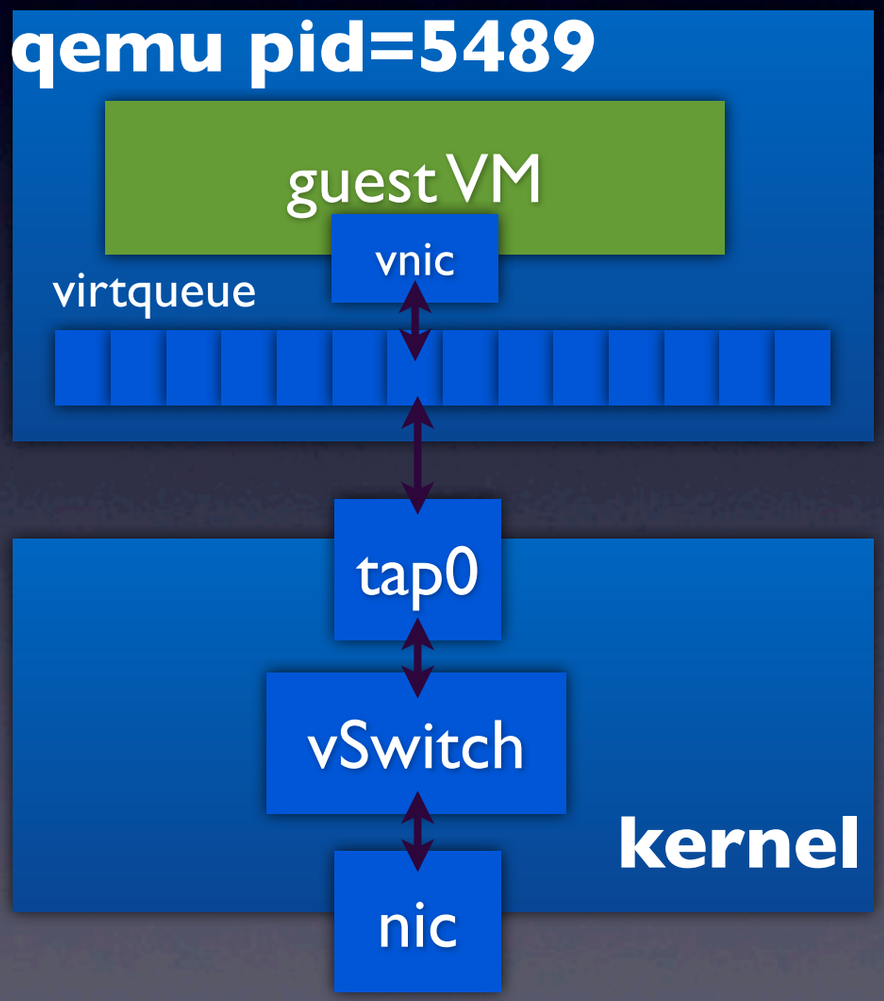
virtio-net

- 基于virtio框架的虚拟以太网设备

```
-netdev type=tap,id=pub226,ifname=pub226,vhost=on,script=up.sh,downscript=down.sh
-device virtio-net-pci,netdev=pub226,mac=00:02:dc:04:59:36,bus=pci.0,addr=0xf
```

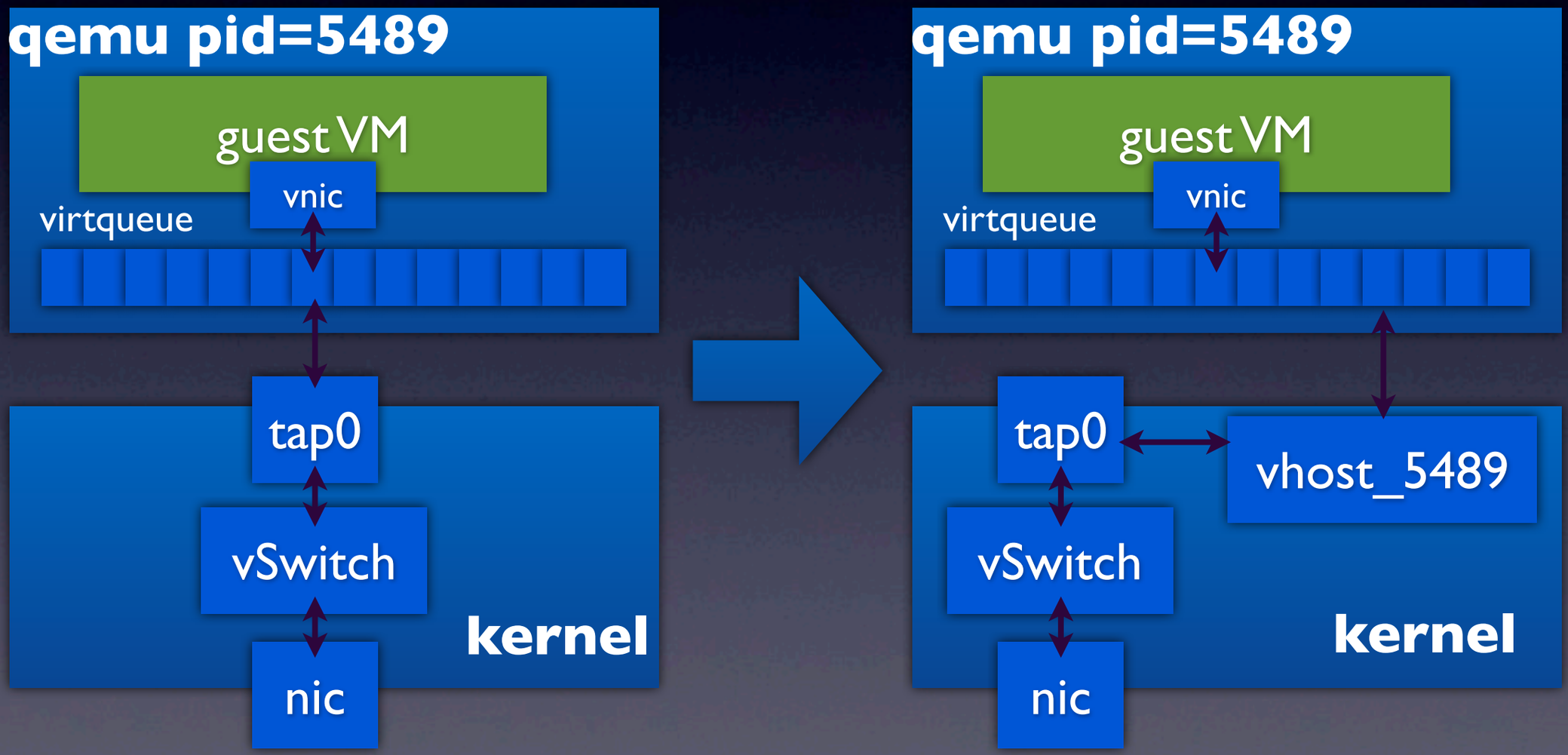
vhost_net

- 内核进程vhost_xxx负责tap设备和guest virtio queue之间的数据交换，减少qemu通过用户态和tap设备交换数据的system call和内存拷贝



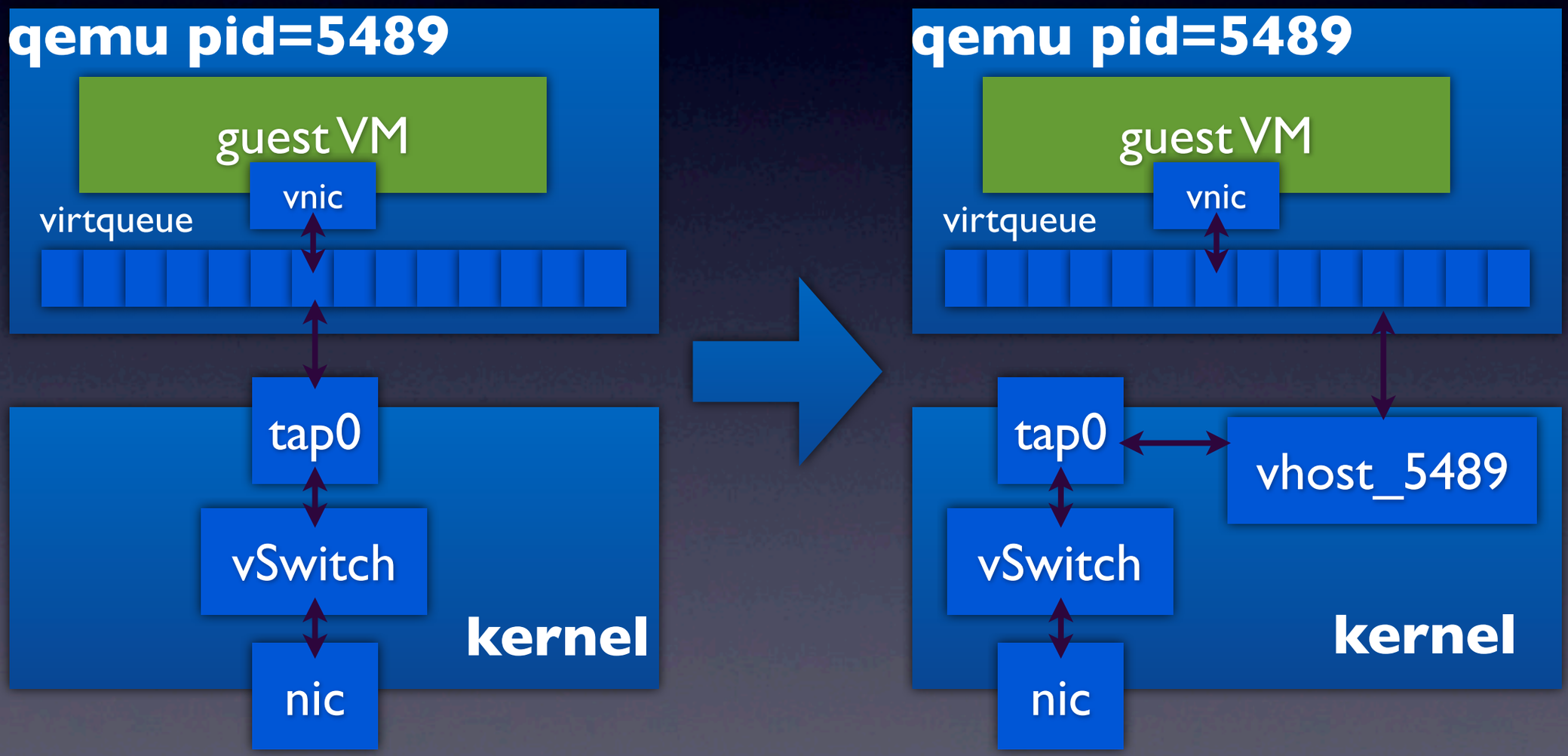
vhost_net

- 内核进程vhost_xxx负责tap设备和guest virtio queue之间的数据交换，减少qemu通过用户态和tap设备交换数据的system call和内存拷贝

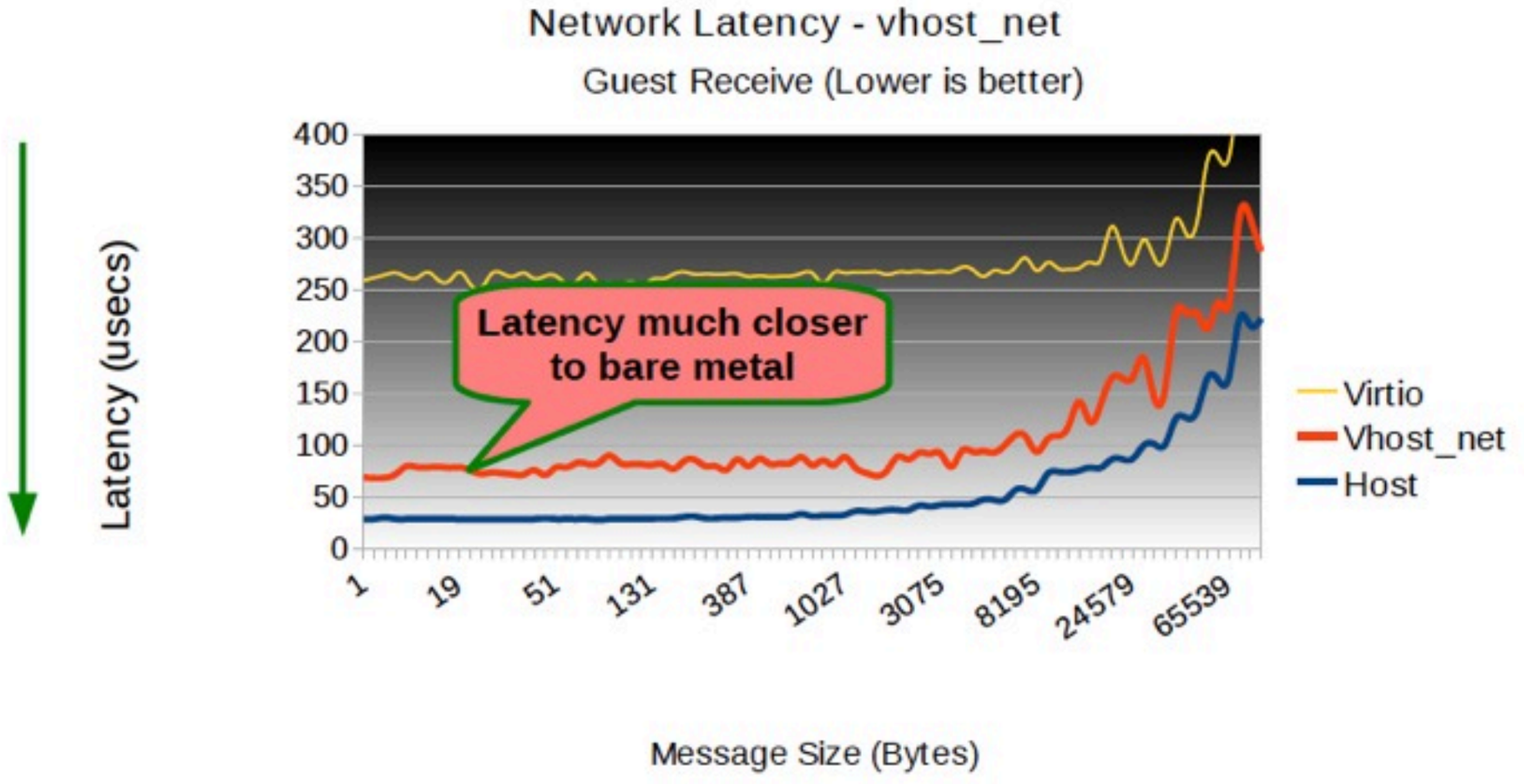


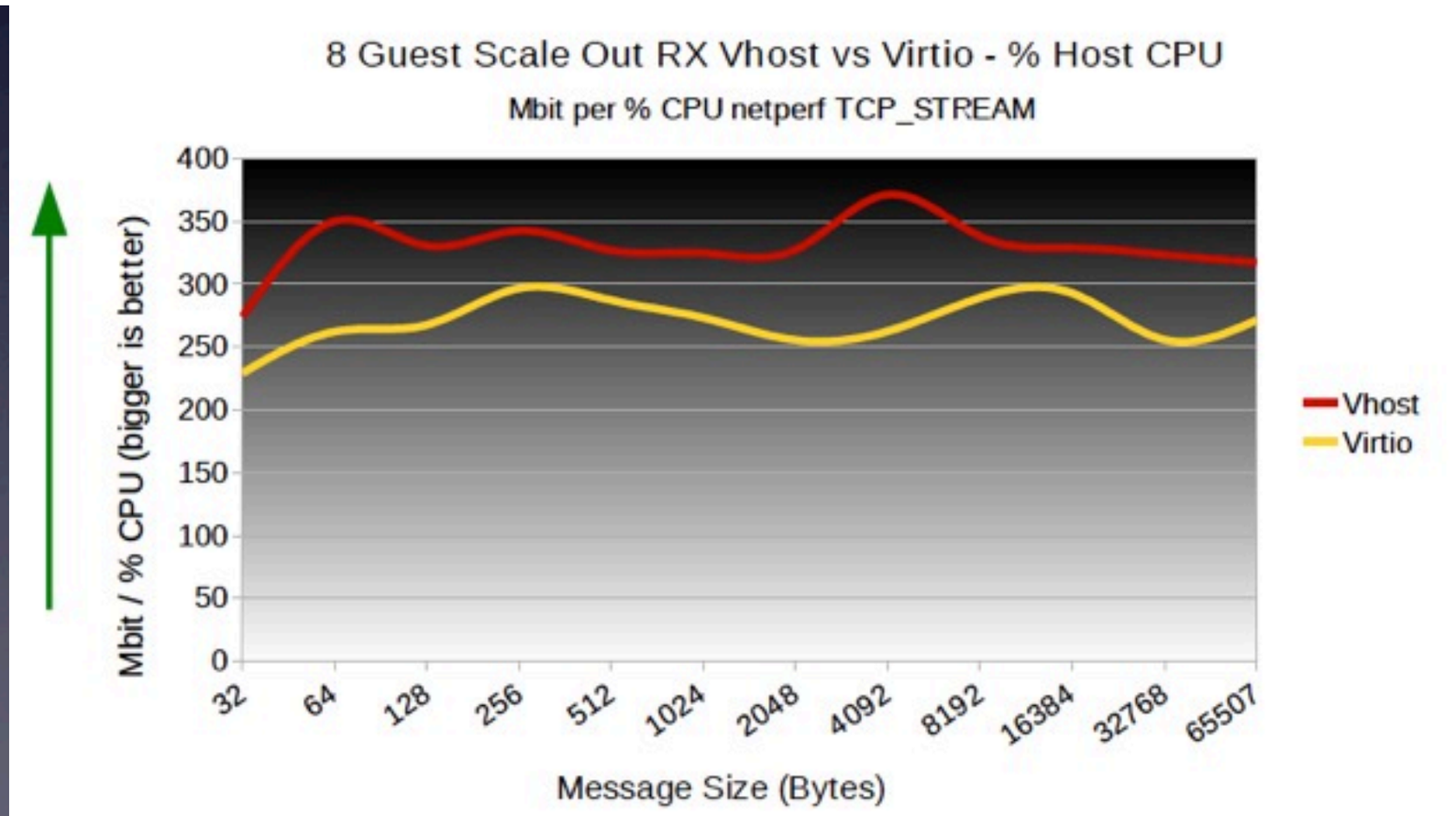
vhost_net

- 内核进程vhost_xxx负责tap设备和guest virtio queue之间的数据交换，减少qemu通过用户态和tap设备交换数据的system call和内存拷贝




```
-netdev type=tap,id=pub226,ifname=pub226,vhost=on,script=up.sh,downscript=down.sh
-device virtio-net-pci,netdev=pub226,mac=00:02:dc:04:59:36,bus=pci.0,addr=0xf
```


Network Latency - vhost_net
Guest Receive (Lower is better)
Latency (usecs)
400
350
300
250
200
150
100
50
0
Latency much closer to bare metal
1
19
51
131
387
1027
3075
8195
24579
65539
Message Size (Bytes)
Virtio
Vhost_net
Host


8 Guest Scale Out RX Vhost vs Virtio - % Host CPU
Mbit per % CPU netperf TCP_STREAM
Mbit / % CPU (bigger is better)
400
350
300
250
200
150
100
50
0
32
64
128
256
512
1024
2048
4092
8192
16384
32768
65507
Message Size (Bytes)
Vhost
Virtio

其他优化选项

- CPU: scheduler
- Memory: NUMA
- Storage: PCI-passthrough
- Network: SR-IOV, PCI-passthrough
- 提升硬件指标

Thank you

Q&A

https://mos.meituan.com