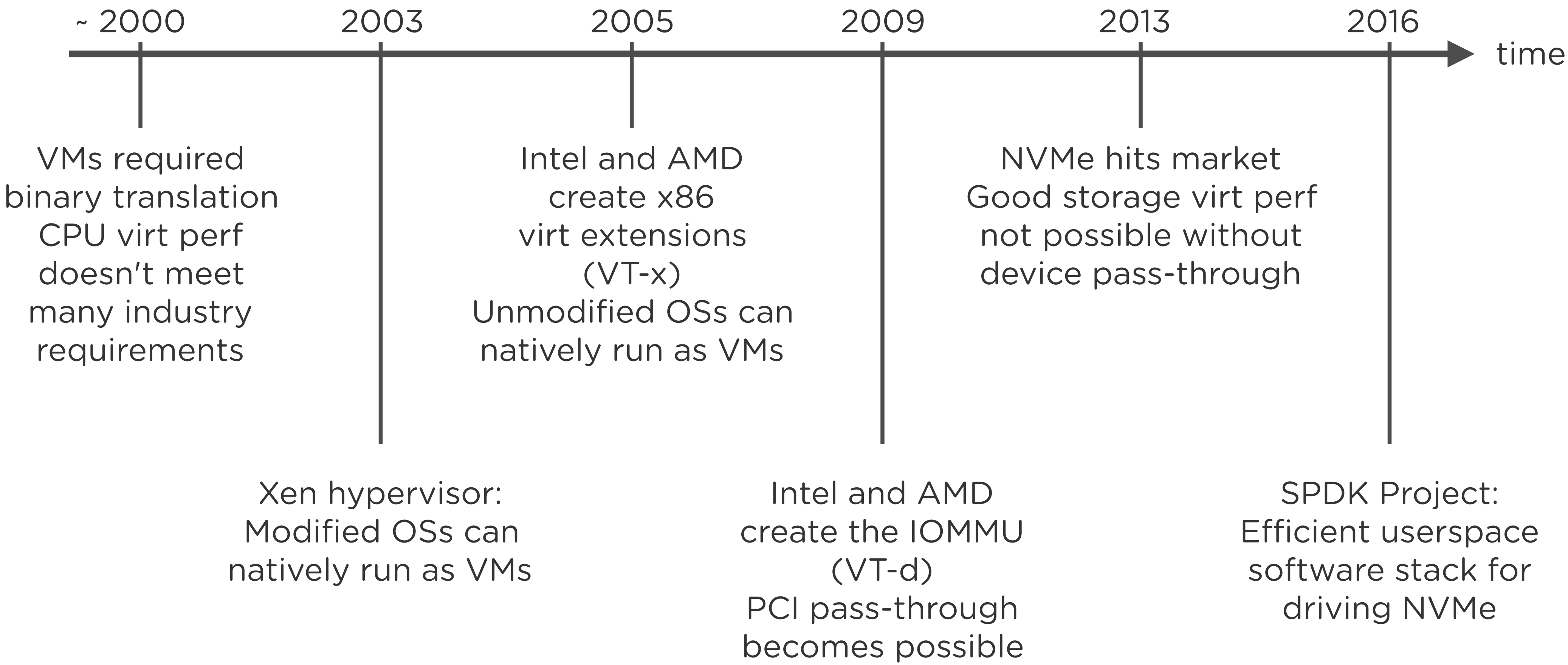
SPDK and Nutanix AHV: Minimising the Virtualisation Overhead

Dr Felipe Franciosi

AHV Engineering Lead

NOVEMBER 2018 | LINUX PITER #4

Timeline



~ 2000
2003
2005
2009
2013
2016
time
VMs required binary translation CPU virt perf doesn't meet many industry requirements
Xen hypervisor: Modified OSs can natively run as VMs
Intel and AMD create x86 virt extensions (VT-x) Unmodified OSs can natively run as VMs
Intel and AMD create the IOMMU (VT-d) PCI pass-through becomes possible
NVMe hits market Good storage virt perf not possible without device pass-through
SPDK Project: Efficient userspace software stack for driving NVMe

Disclaimer

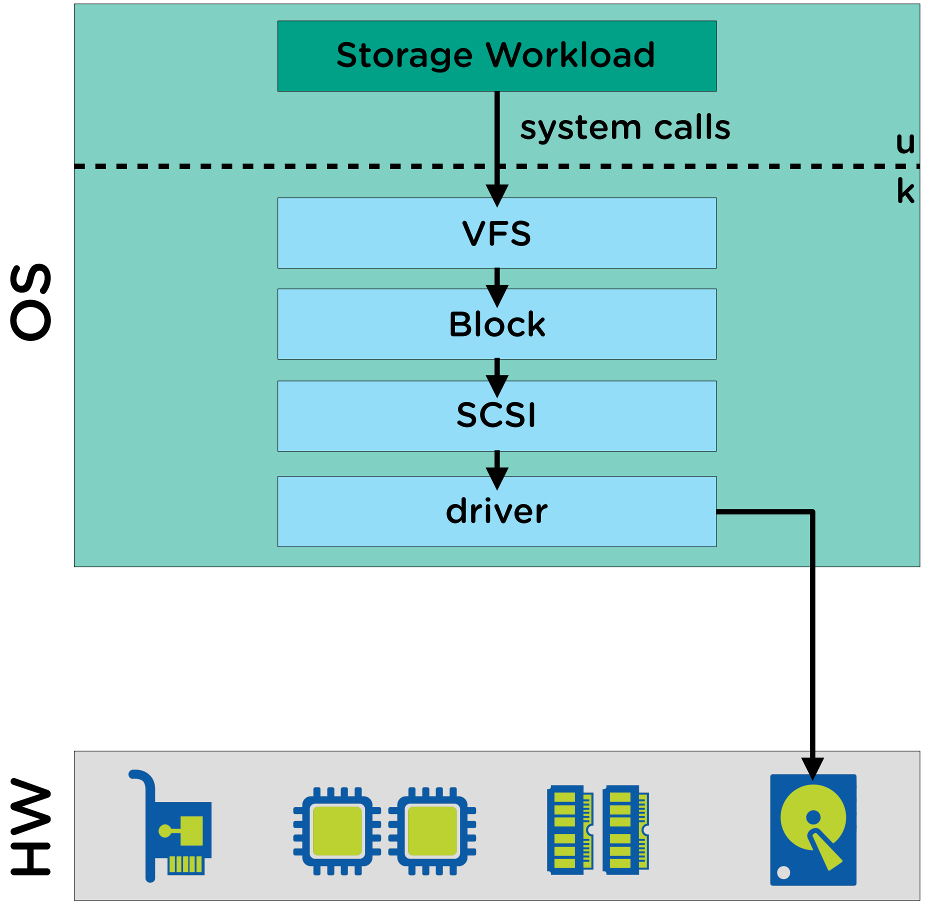
This presentation and the accompanying oral commentary may include express and implied forward-looking statements, including but not limited to statements concerning our business plans and objectives, product features and technology that are under development or in process and capabilities of such product features and technology, our plans to introduce product features in future releases, the implementation of our products on additional hardware platforms, strategic partnerships that are in process, product performance, competitive position, industry environment, and potential market opportunities. These forward-looking statements are not historical facts, and instead are based on our current expectations, estimates, opinions and beliefs. The accuracy of such forward-looking statements depends upon future events, and involves risks, uncertainties and other factors beyond our control that may cause these statements to be inaccurate and cause our actual results, performance or achievements to differ materially and adversely from those anticipated or implied by such statements, including, among others: failure to develop, or unexpected difficulties or delays in developing, new product features or technology on a timely or cost-effective basis; delays in or lack of customer or market acceptance of our new product features or technology; the failure of our software to interoperate on different hardware platforms; failure to form, or delays in the formation of, new strategic partnerships and the possibility that we may not receive anticipated results from forming such strategic partnerships; the introduction, or acceleration of adoption of, competing solutions, including public cloud infrastructure; a shift in industry or competitive dynamics or customer demand; and other risks detailed in our Form 10-Q for the fiscal quarter ended April 30, 2017, filed with the Securities and Exchange Commission. These forward- looking statements speak only as of the date of this presentation and, except as required by law, we assume no obligation to update forward- looking statements to reflect actual results or subsequent events or circumstances. Any future product or roadmap information is intended to outline general product directions, and is not a commitment, promise or legal obligation for Nutanix to deliver any material, code, or functionality. This information should not be used when making a purchasing decision. Further, note that Nutanix has made no determination as to if separate fees will be charged for any future product enhancements or functionality which may ultimately be made available. Nutanix may, in its own discretion, choose to charge separate fees for the delivery of any product enhancements or functionality which are ultimately made available.

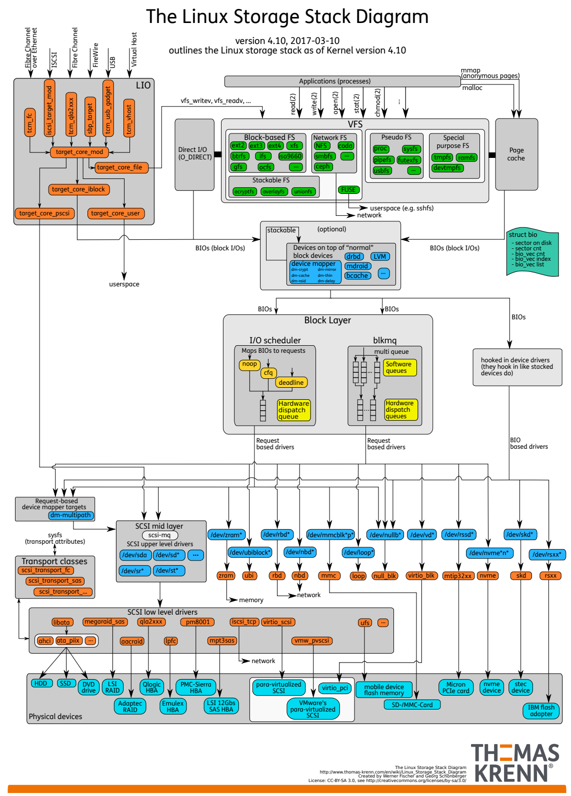
Certain information contained in this presentation and the accompanying oral commentary may relate to or be based on studies, publications, surveys and other data obtained from third-party sources and our own internal estimates and research. While we believe these third-party studies, publications, surveys and other data are reliable as of the date of this presentation, they have not independently verified, and we make no representation as to the adequacy, fairness, accuracy, or completeness of any information obtained from third-party sources.

Agenda

1. Understanding Overhead in Storage Performance
2. Hypervisor Analysis
3. AHV and SPDK: Userspace "FTW"
4. Towards Millions of IOPS on a Single Virtual Disk

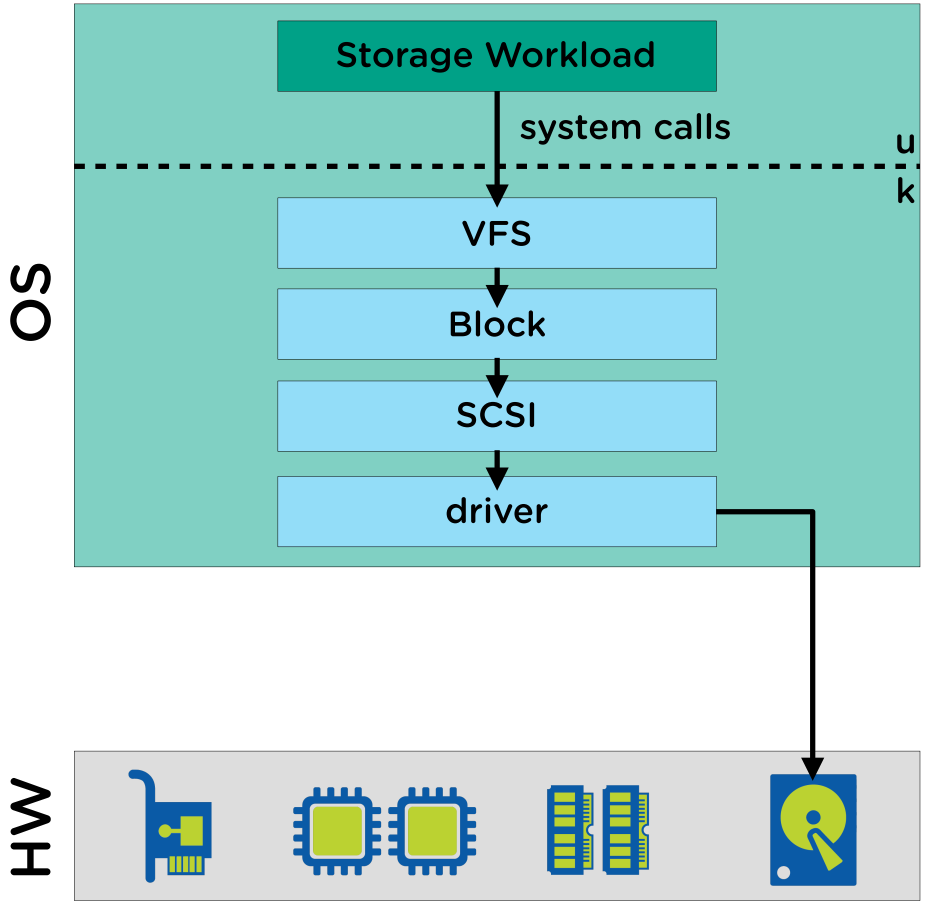
Storage Datapath





Strongly encouraged read:
https://www.thomas-krenn.com/en/wiki/Linux_Storage_Stack_Diagram

Storage Datapath

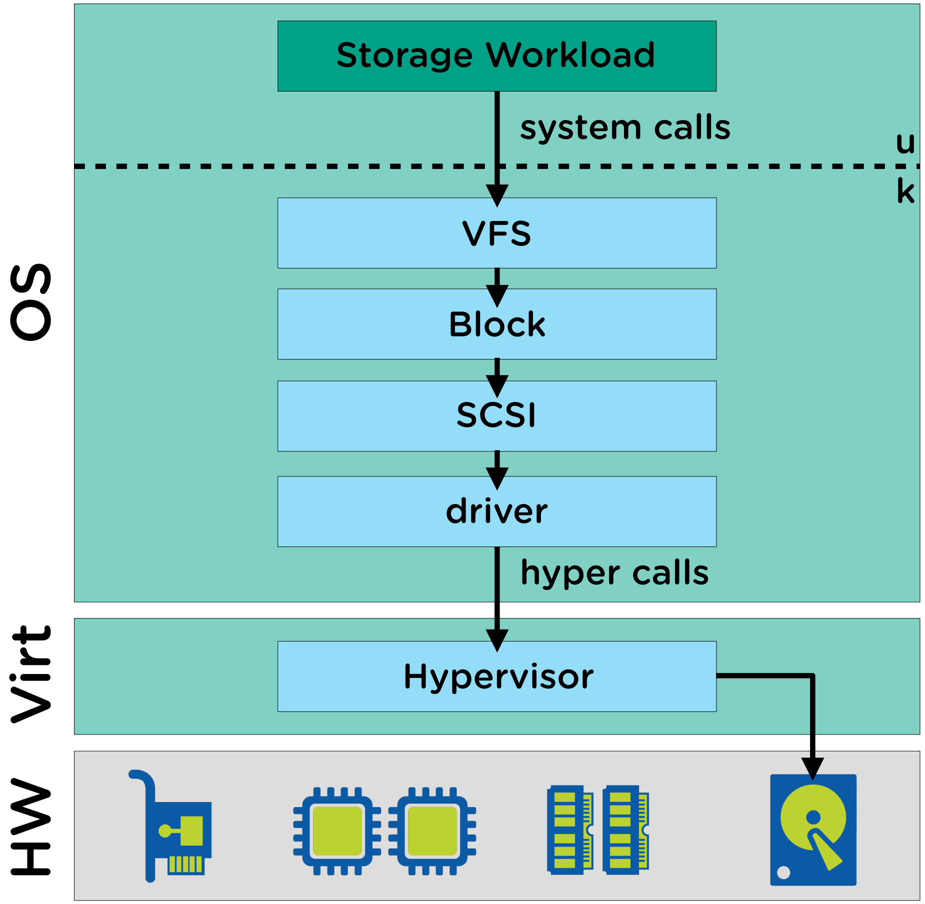


Where did time go?

~ µs Time spent on CPU is in order of **microseconds**.

~ ms Time spent on disks is in order of **milliseconds**.

Storage Datapath



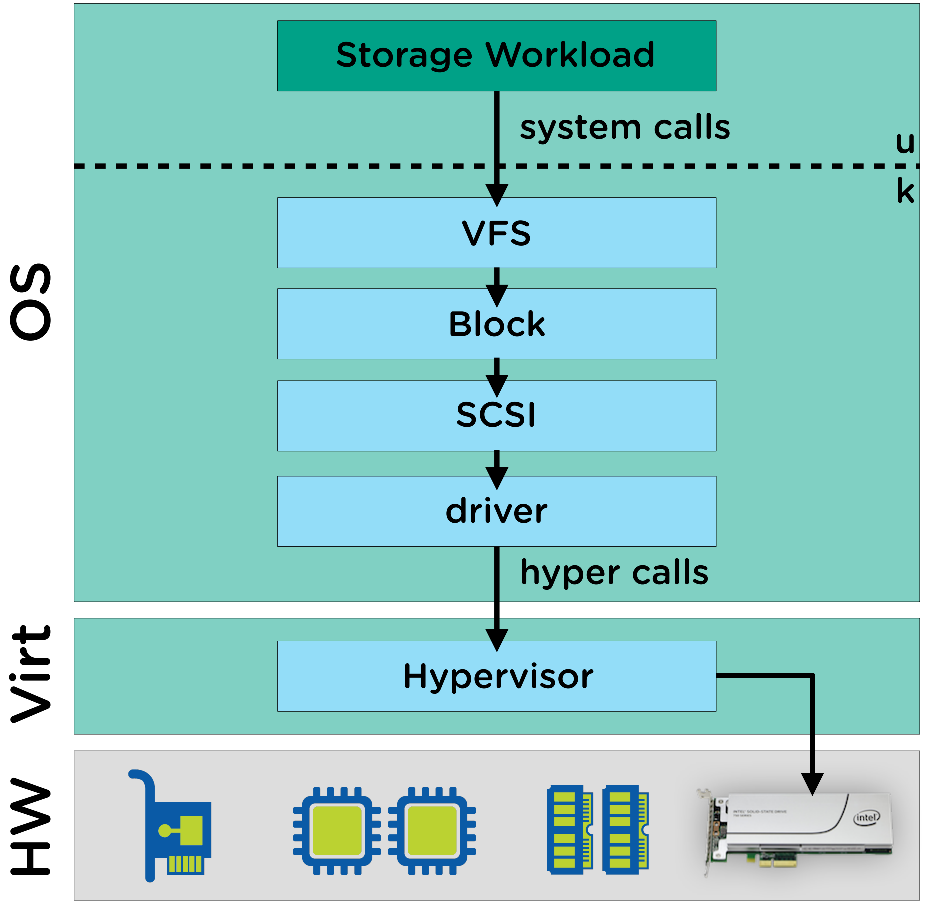
Where did time go?

~ µs Time spent on CPU is in order of **microseconds**.

~ µs Hypervisor adds some more **microseconds**.

~ ms Time spent on disks is in order of **milliseconds**.

Storage Datapath



Where did time go?

~ µs Time spent on CPU is in order of **microseconds**.

~ µs Hypervisor adds some more **microseconds**.

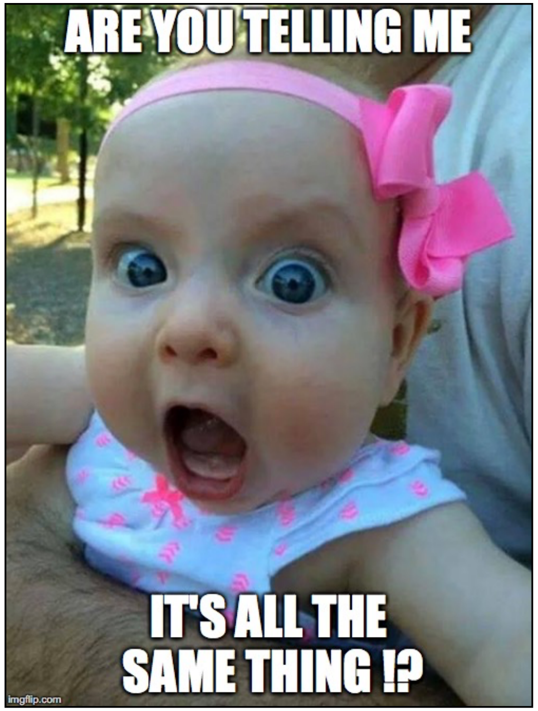
~ µs Most NVMe: latency is in order of **microseconds**.

Storage Performance Metrics

What are we really measuring?

- Bandwidth or Throughput (MB/s)
- IOPS (reqs/s)
- Latency (ms)

(MB/s)	(reqs/s)	(s/req)
$\frac{\text{data}}{\text{time}}$	$\frac{\text{data}}{\text{time}}$	$\frac{\text{time}}{\text{data}}$



Storage Performance Metrics

Yes! Well, in a way...

They all measure the same thing, but differently.

- Each metric shows different aspects of the system
- For each one, the system must be stressed differently

Think of a runner:

What does it mean to be fast?

- Long steps
- Fast steps

Driving Load on Storage Devices

There are **two ways** of loading a storage device:

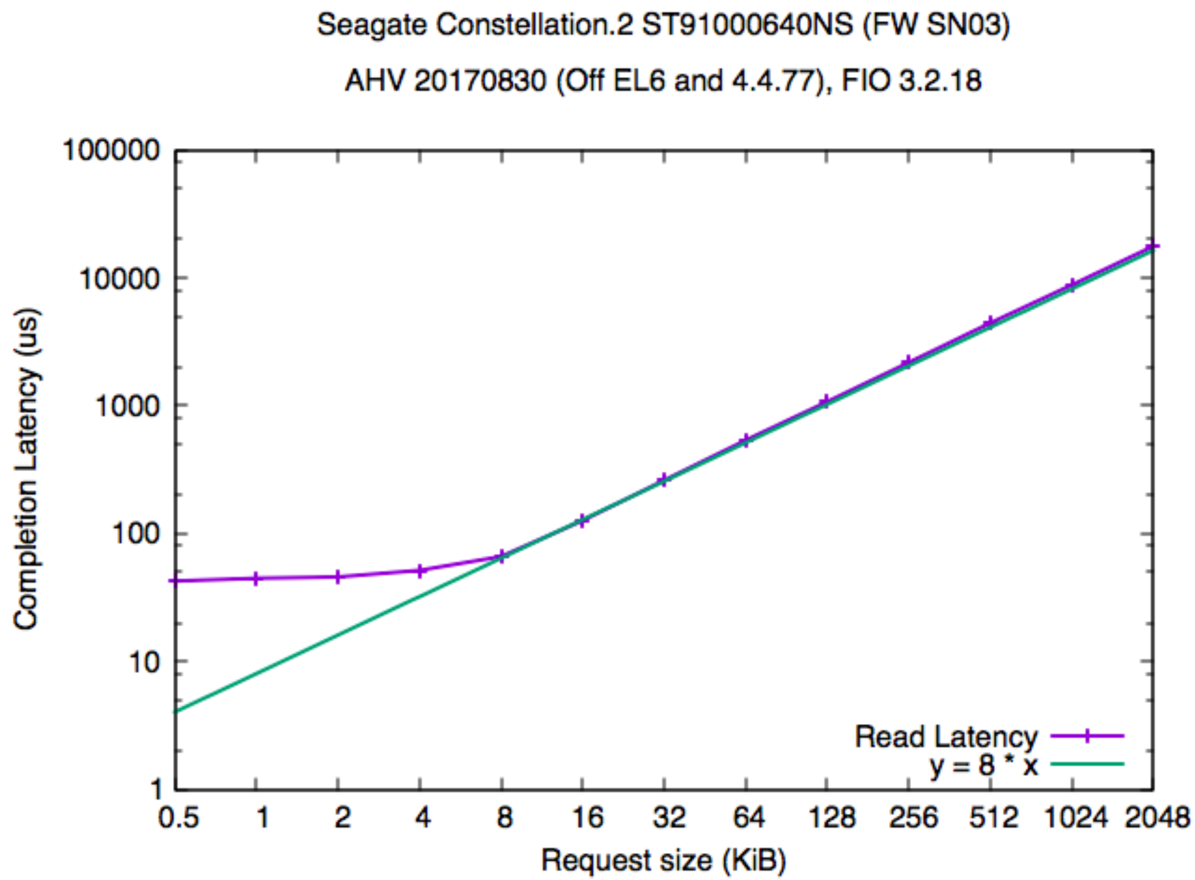
- **Increasing the size of requests**
  - Large requests are associated with sequential workloads
  - They are usually throughput-sensitive
- **Increasing the number of requests**
  - Many requests are associated with random workloads
  - They are relatively small, and latency/IOPS -sensitive

Storage Performance and Virtualisation

Latency as request size increases (HDD):

- Mechanical drive
- Sequential reads
- Queue depth = 1
- Varying request size

Storage is saturated.



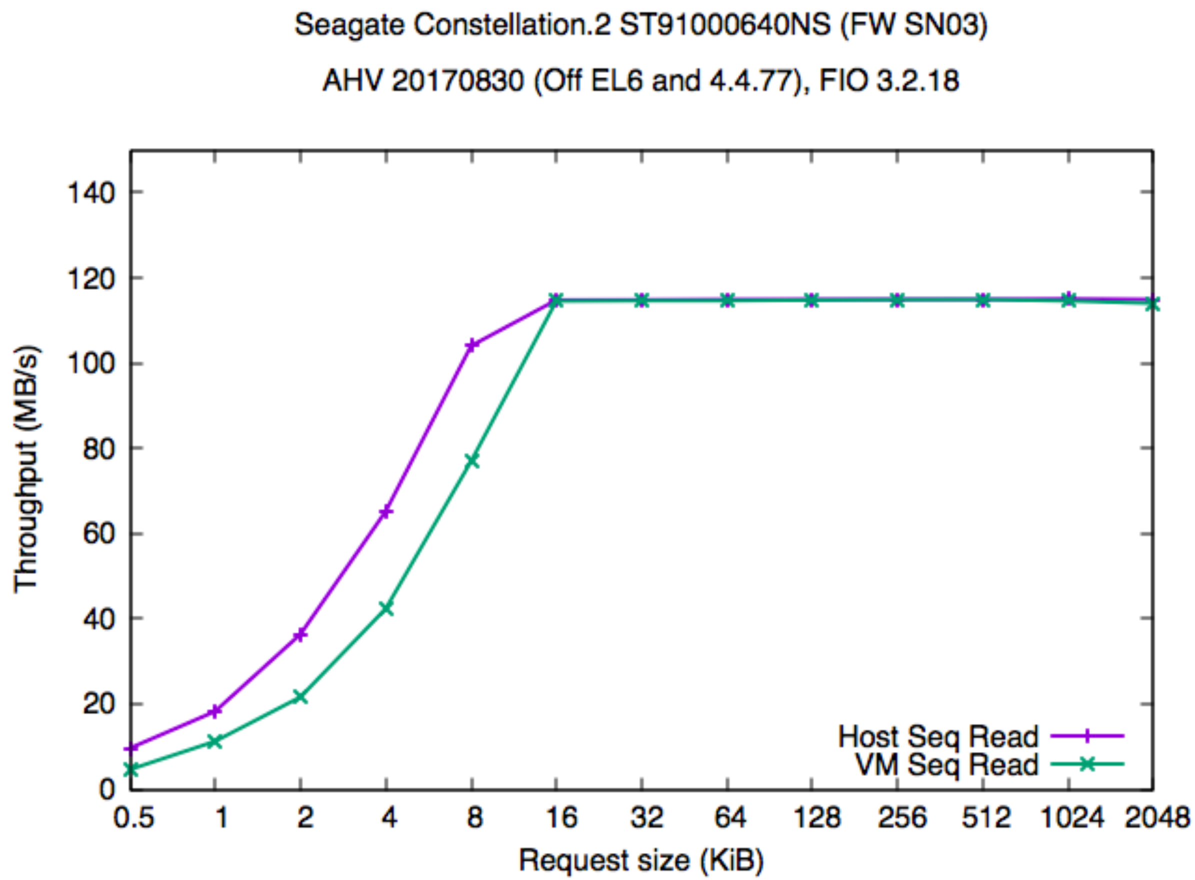
Storage Performance and Virtualisation

Translating that to throughput:

- Mechanical drive
- Sequential reads
- Queue depth = 1
- Varying request size

And from a VM ?

- Debian 9.4 VM (FIO 3.2.18)
- Host with Qemu 2.6
- Disk over virtio-scsi

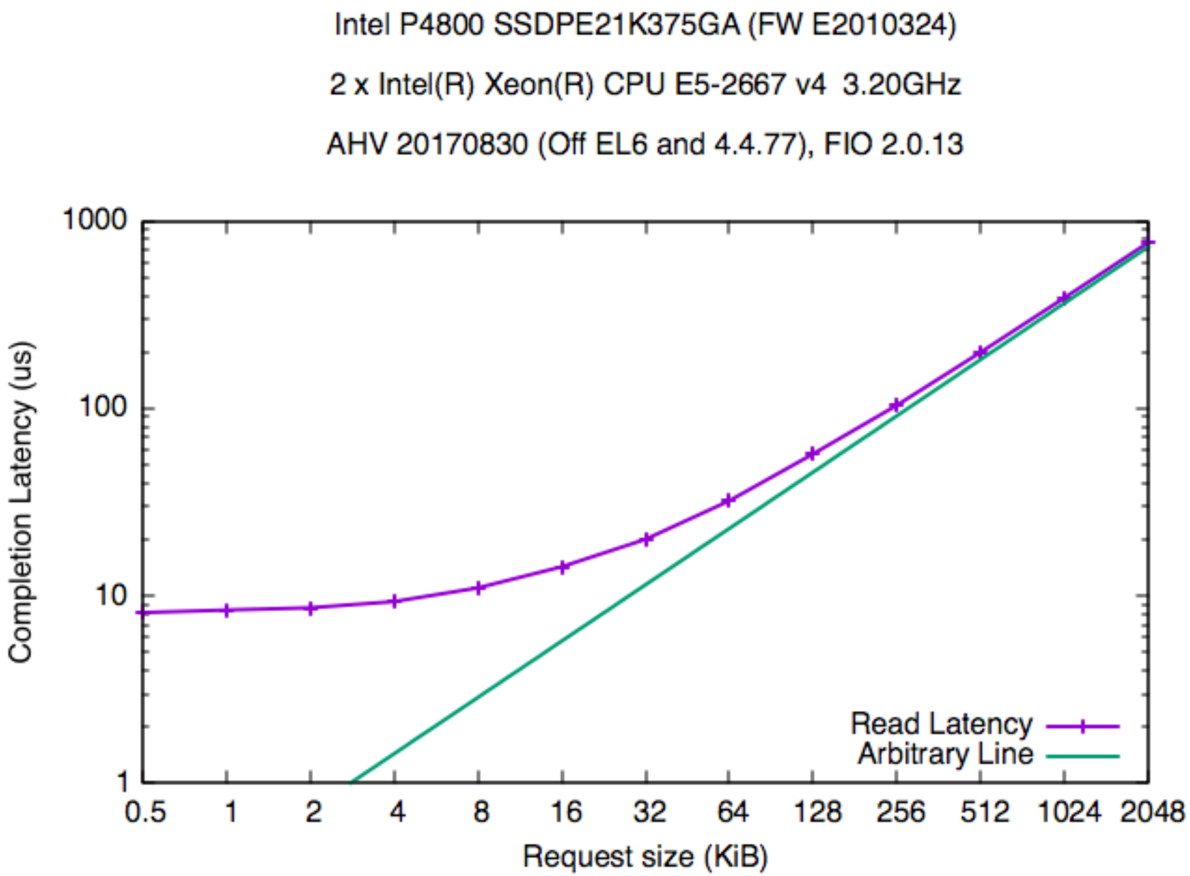


Storage Performance and Virtualisation

Latency as request size increases (NVMe w/ 3DXP):

- NVMe w/ 3DXP
- Random reads
- Queue depth = 1
- Varying request size

Storage is NOT saturated



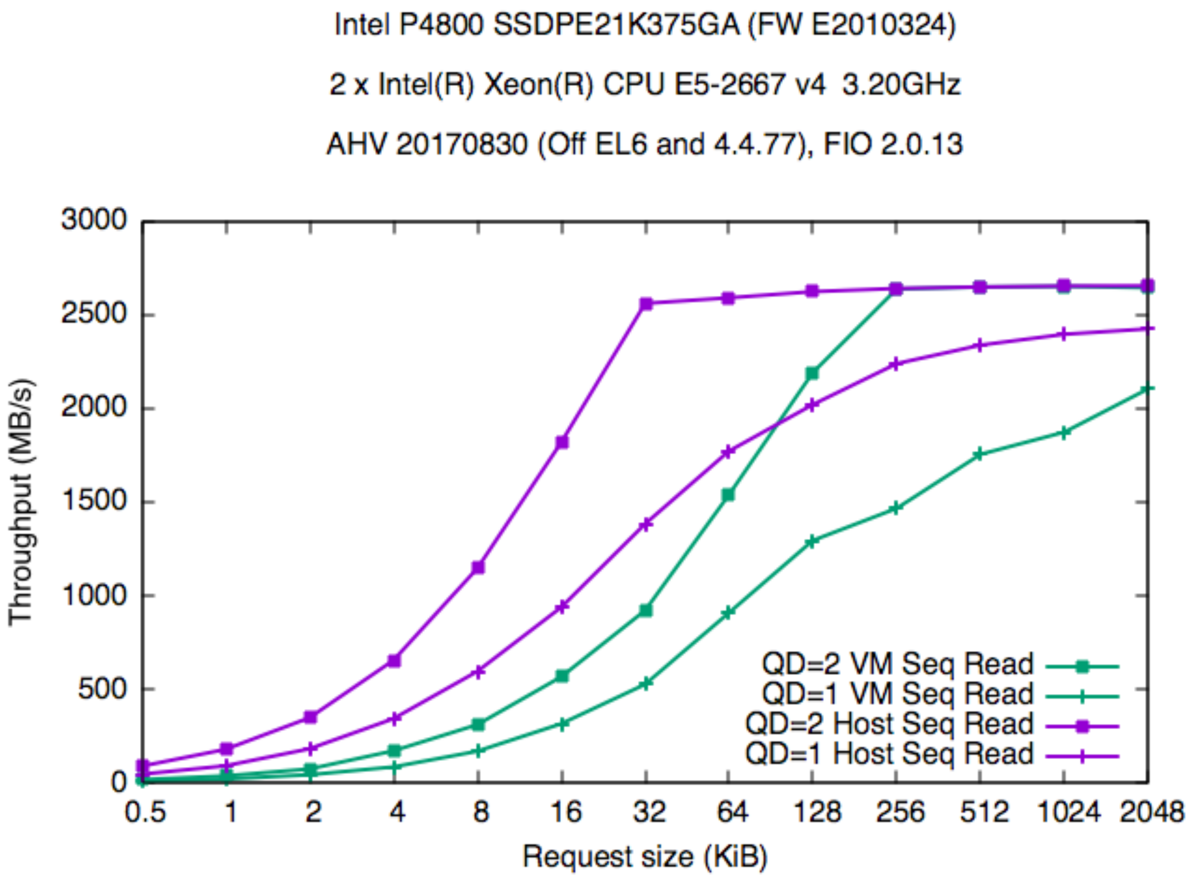
Storage Performance and Virtualisation

Translating that to throughput:

- NVMe w/ 3DXP
- Random reads
- Queue depth = 1 (or 2)
- Varying request size

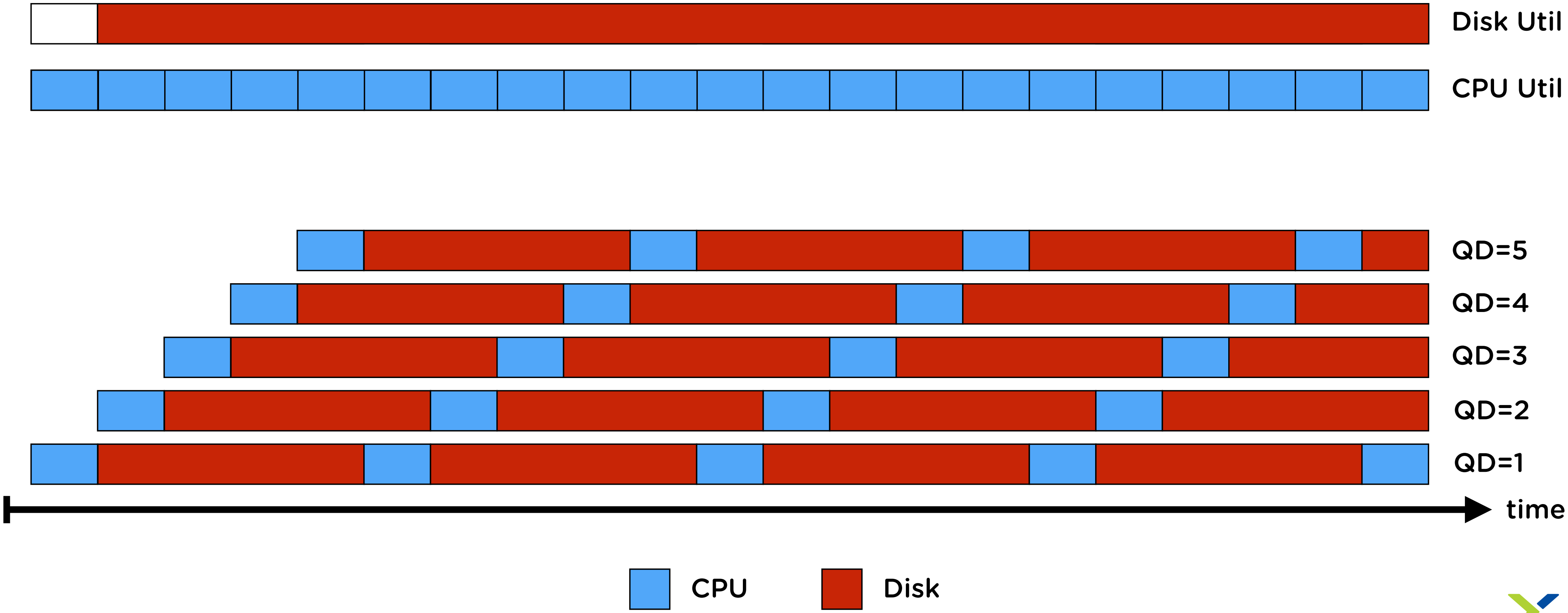
And from a VM ?

- Debian 9.4 VM (FIO 2.16)
- Host with Qemu 2.6
- vDisk over virtio-scsi



Saturating CPUs and Storage Devices

NVMe is "parallel", a single CPU is not.



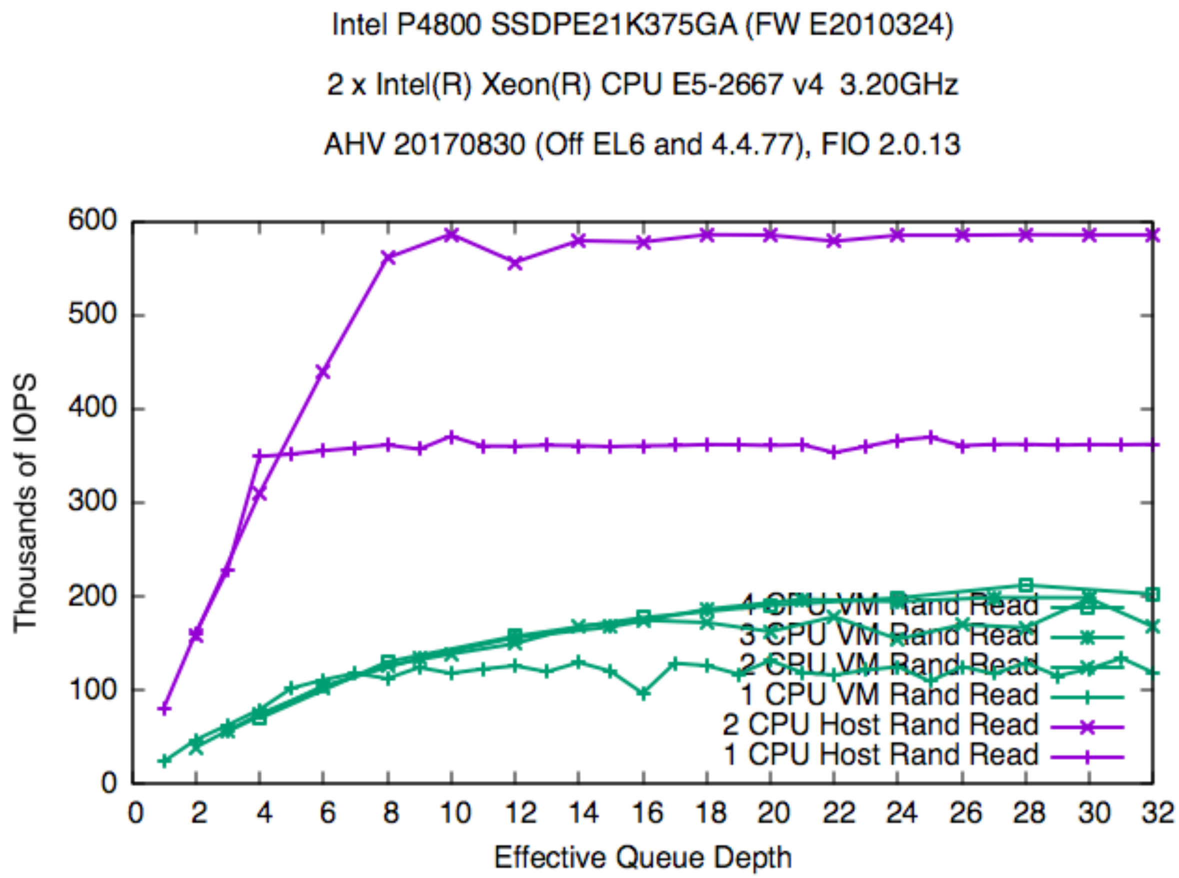
Storage Performance and Virtualisation

A more challenging metric: IOPS

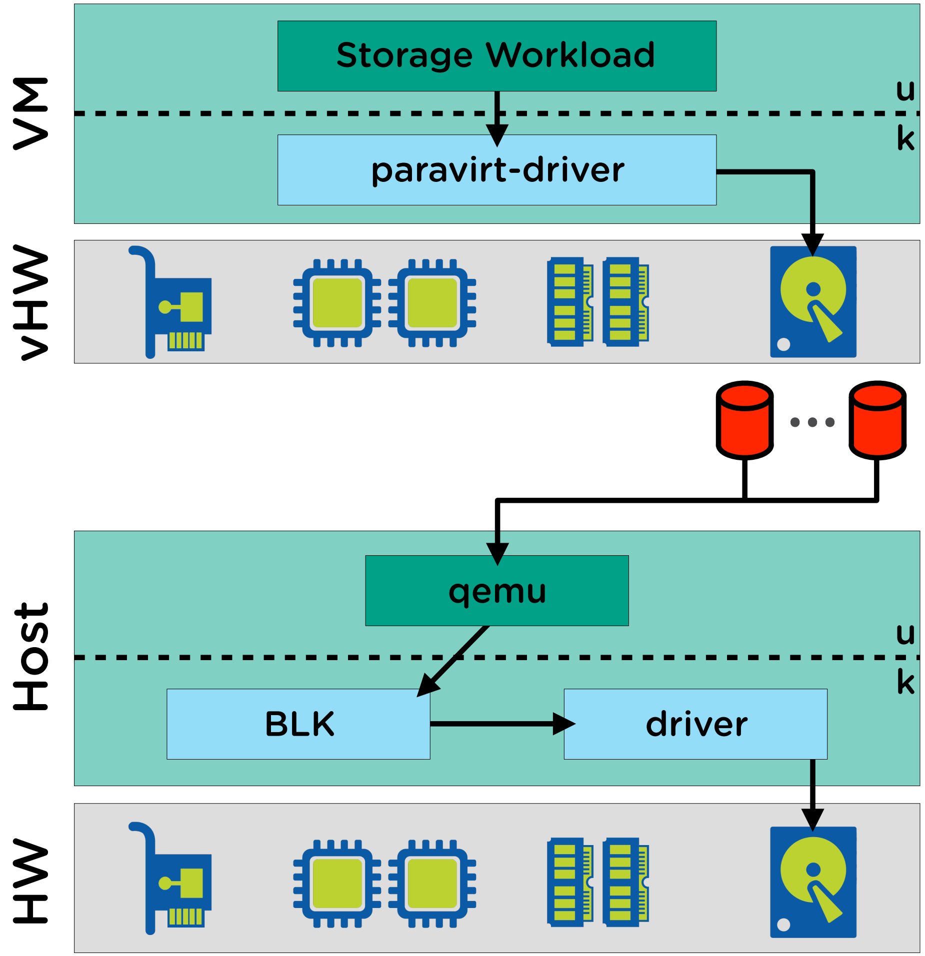
- NVMe w/ 3DXP
- Random reads
- Varying queue depth
- 4 KiB request size

And from a VM ?

- Debian 9.4 VM (FIO 3.2.18)
- Host with Qemu 2.6
- Disks over virtio-scsi



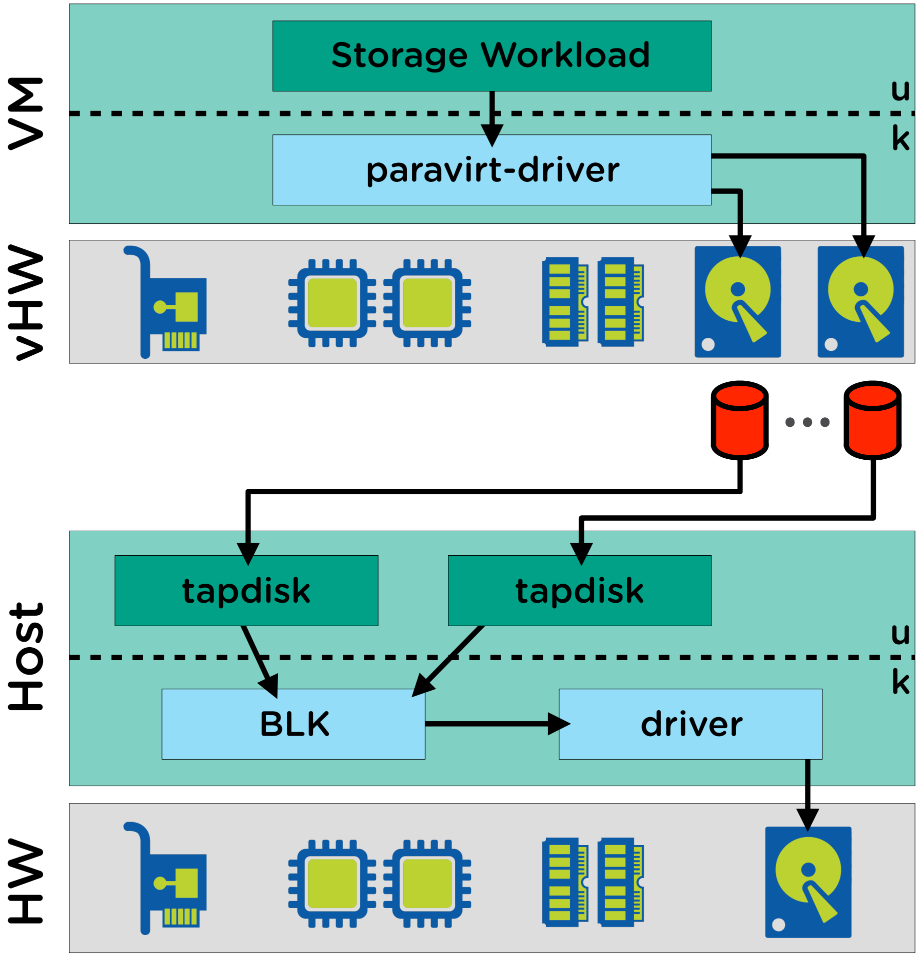
Hypervisor Analysis



Typical virtio-scsi deployment

- One controller presented to VM
- Disks are luns under targets
- One qemu thread handles ctrl
- Qemu bottlenecks on CPU
- Adding more disks won't help
- Adding more ctrls won't help

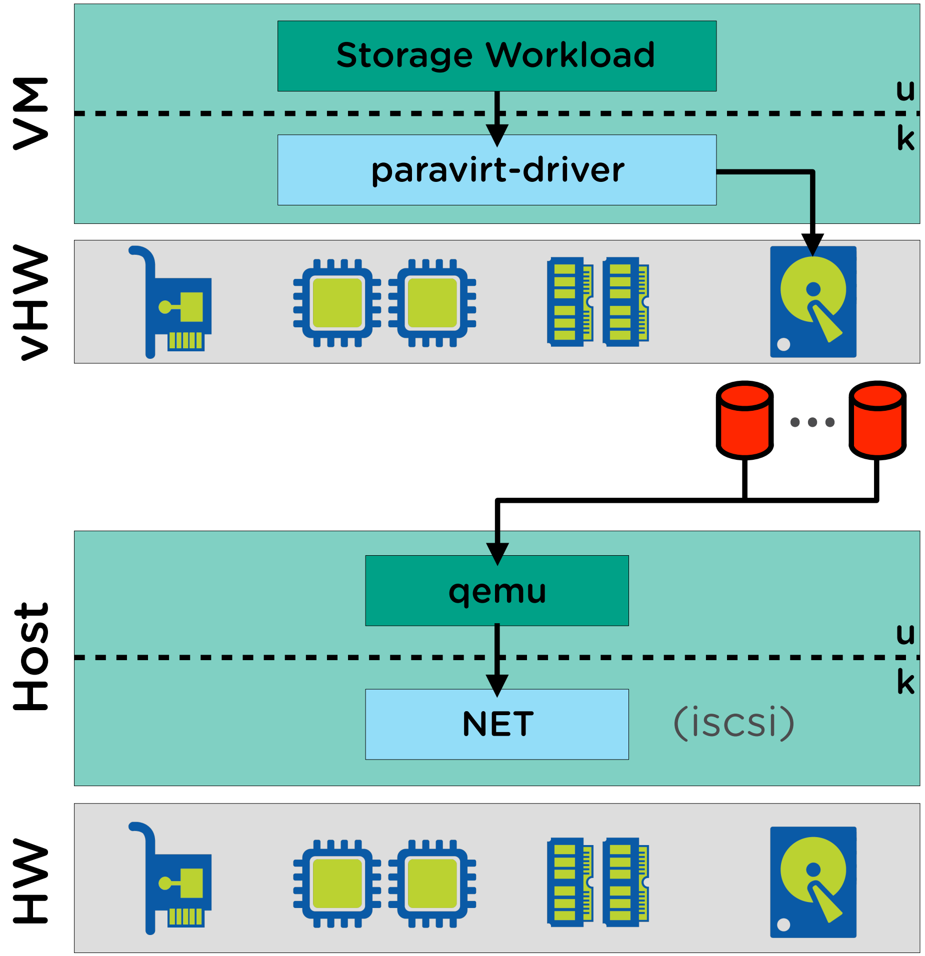
Hypervisor Analysis



Typical XenServer deployment

- Each vdisk is a block device
- Each vdisk backed by a tapdisk
- Tapdisk bottlenecks on CPU
- Bad scalability:
  - Requires more vdisks
  - Too much CPU consumption
  - Doesn't scale with VM size
  - Incompatible with workloads

Hypervisor Analysis

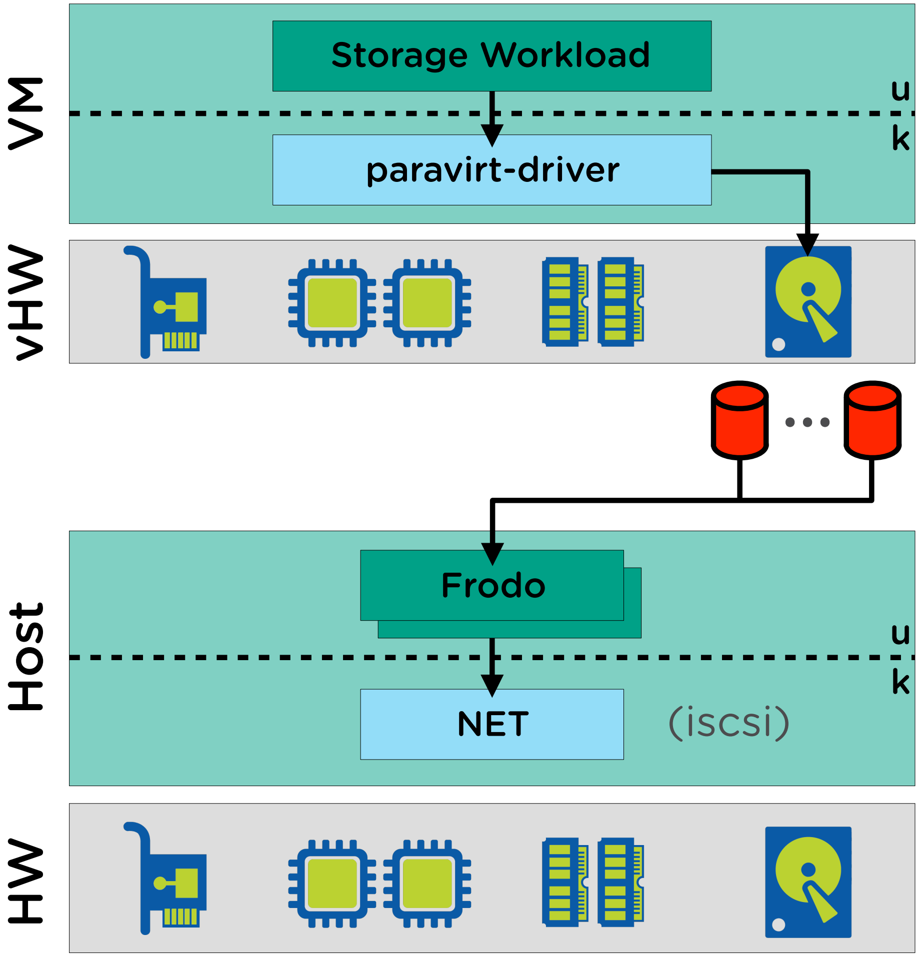


Nutanix AHV up to 5.1

- Qemu handles storage datapath
- With fast devices, Qemu bottlenecks on CPU

- Qemu dataplane meant to provide more threads
- Some hypervisors recommend more controllers (similar to XS)

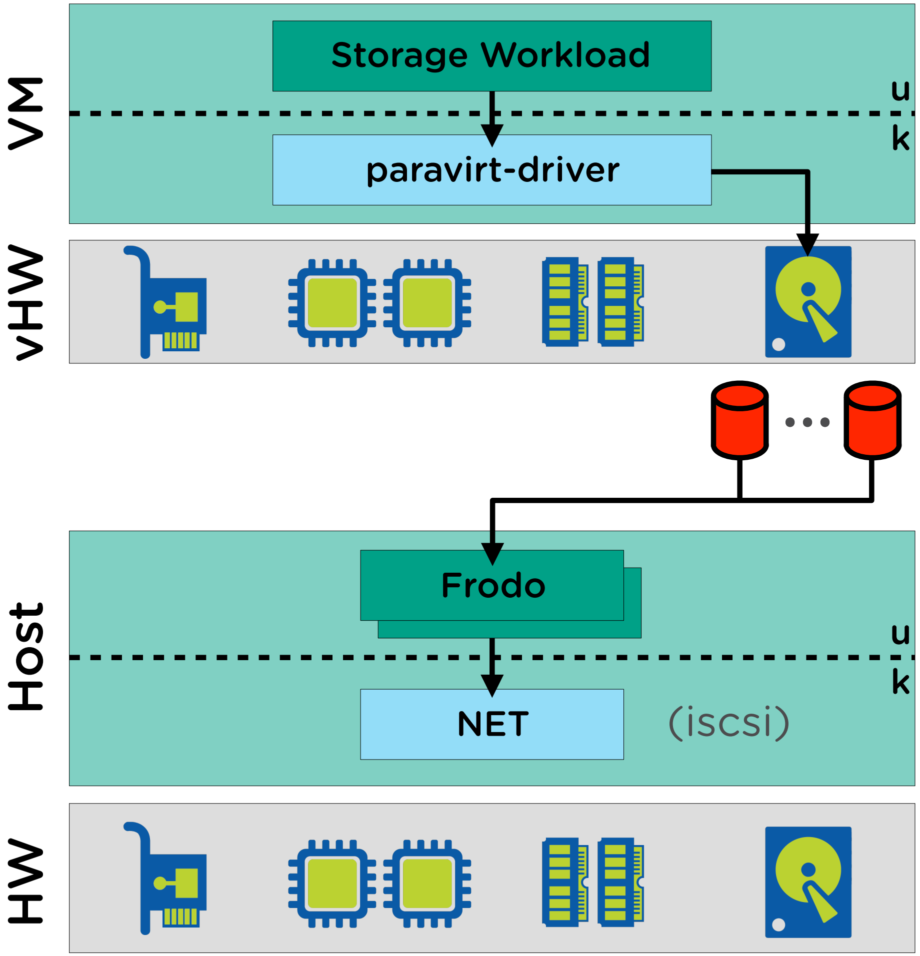
Hypervisor Analysis



Nutanix AHV 5.5 onwards

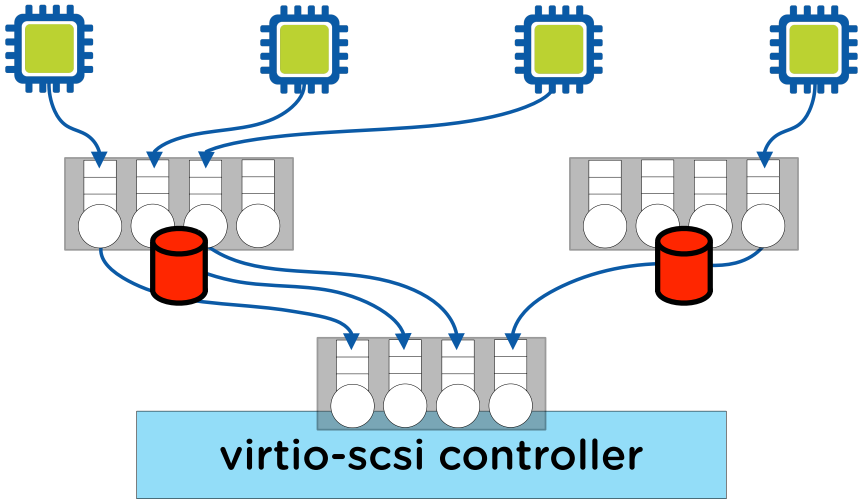
- Frodo handles storage datapath (offloaded by Qemu: vhost-user)
- Frodo presents a MQ controller
- Frodo is multi-threaded, using different threads for different VQs
- Frodo's code is very lean, each thread performs better than Qemu (160k+ IOPS/thread vs 80k IOPS @4k Random Reads on NTNX)

Hypervisor Analysis



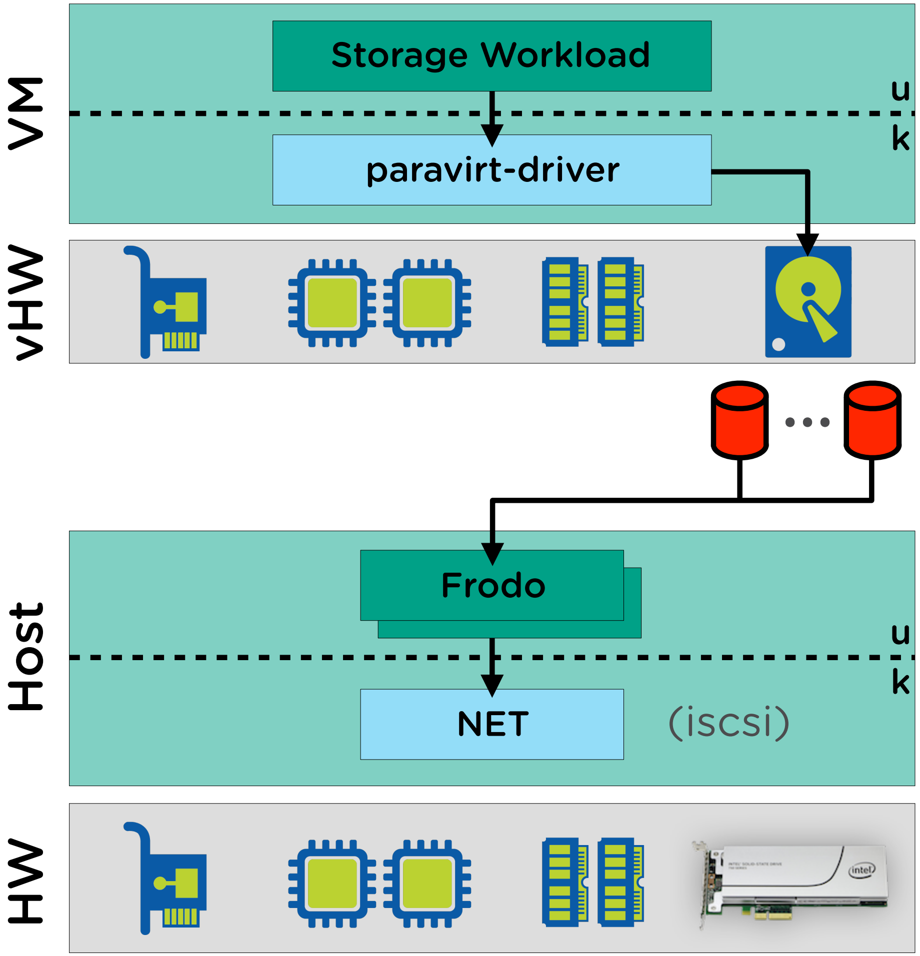
Nutanix AHV 5.5 onwards

- VM gets 1 (vHW) VQ per vCPU
- OS creates 1 (SW) VQ/vCPU/vDisk



- Higher number of inflight requests
- Lock-free datapath

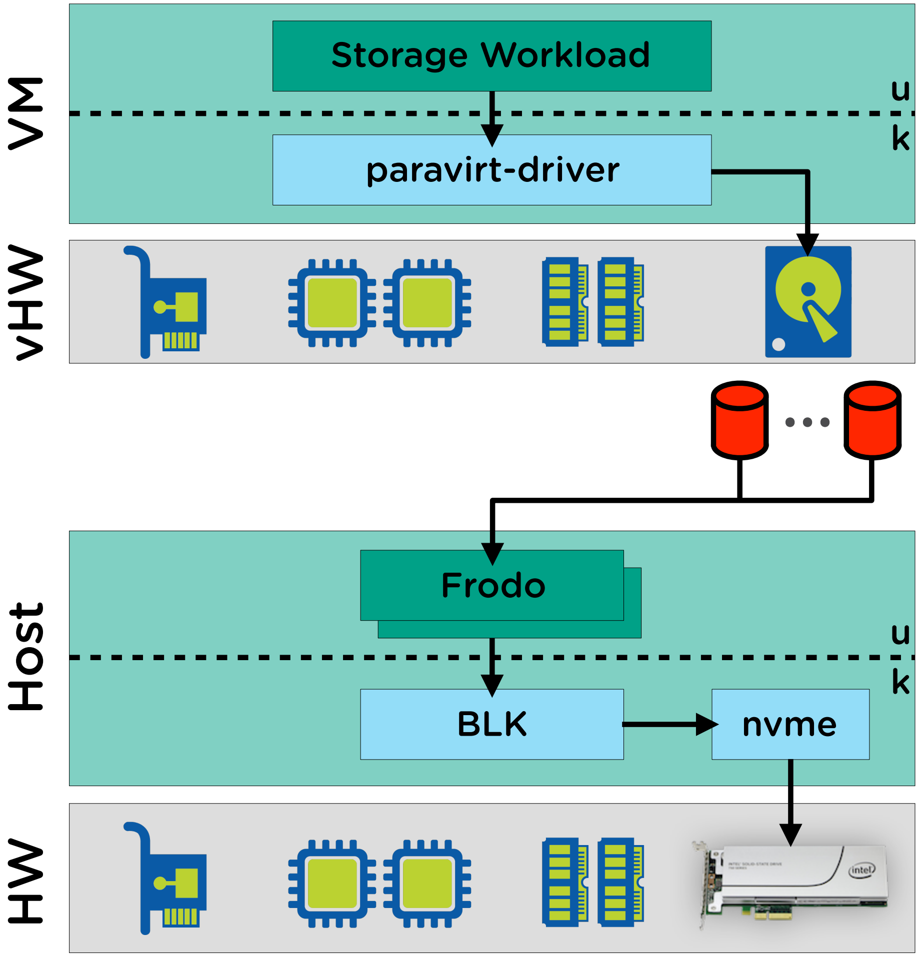
Hypervisor Analysis



Nutanix AHV under research

- Current datapath (iSCSI) too long to benefit from NVMe lower latency
- Let's bring NVMe closer to VM
- Minimise virtualisation overhead

Hypervisor Analysis

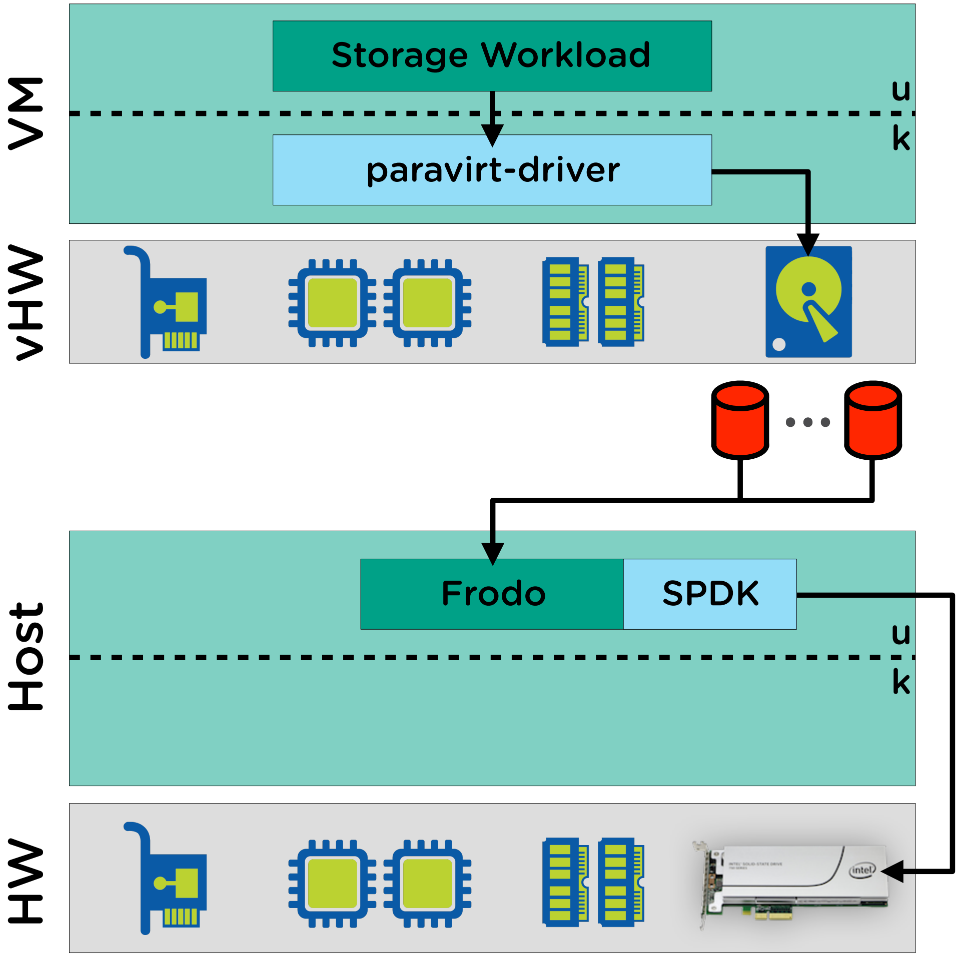


Nutanix AHV under research

- Current datapath (iSCSI) too long to benefit from NVMe lower latency
- Let's bring NVMe closer to VM
- Minimise virtualisation overhead

- One way of doing that is to use libaio and submit requests through the kernel… not.

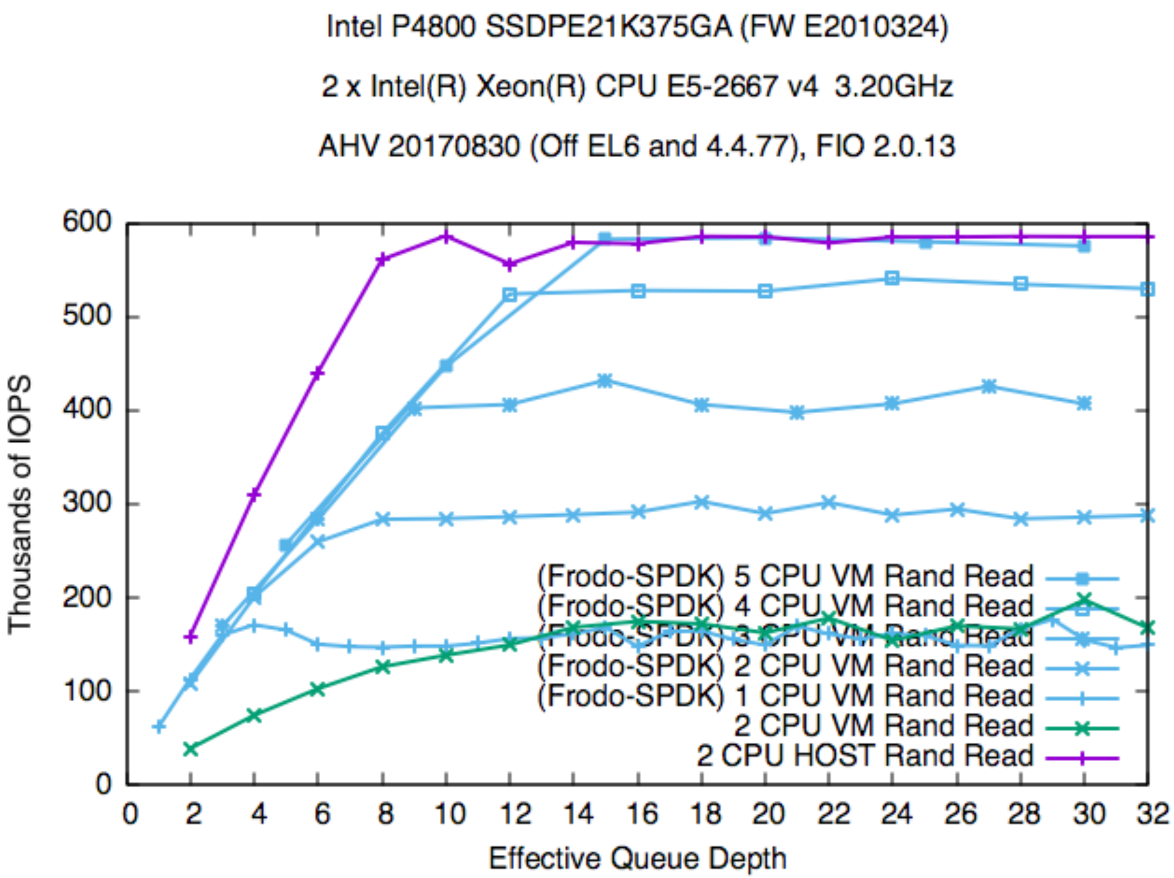
Hypervisor Analysis



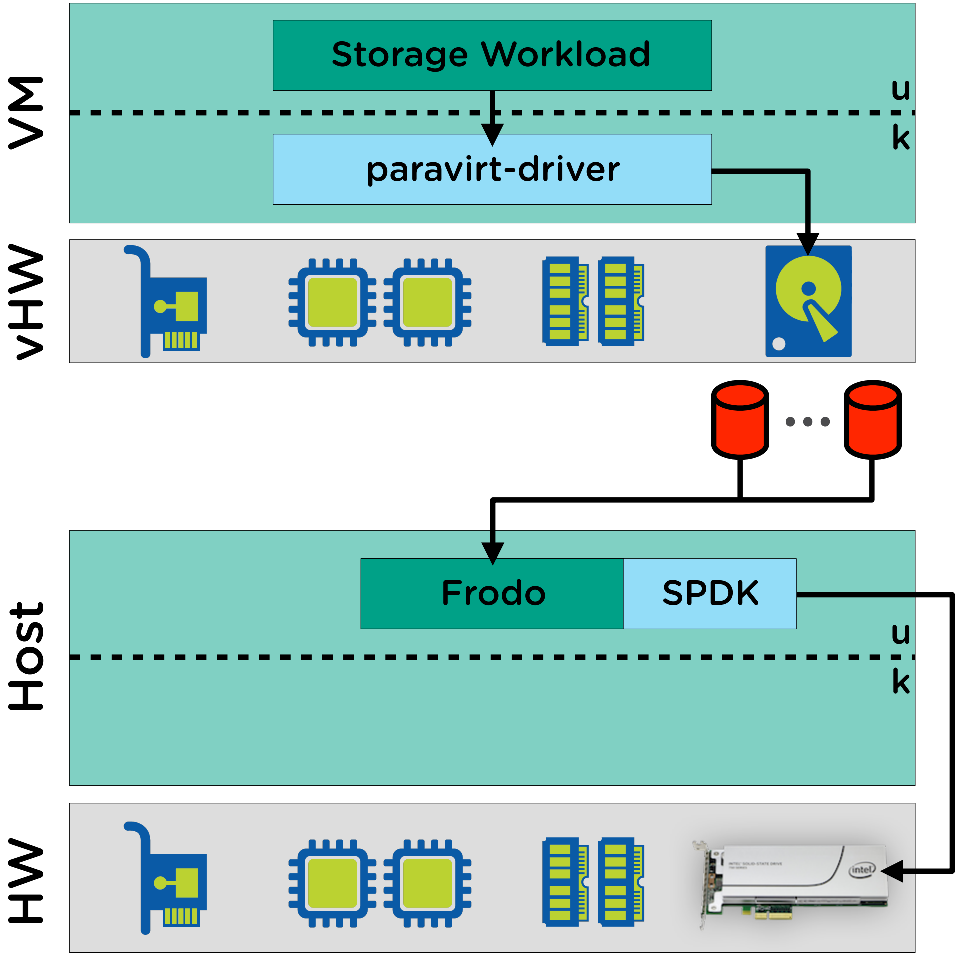
Nutanix AHV under research

- SPDK is a userspace framework for driving storage controllers (NVMe)
- It means the controller must be used exclusively by one process
- Much better performance:
  - Super lean
  - No IRQs

Nutanix AHV and SPDK



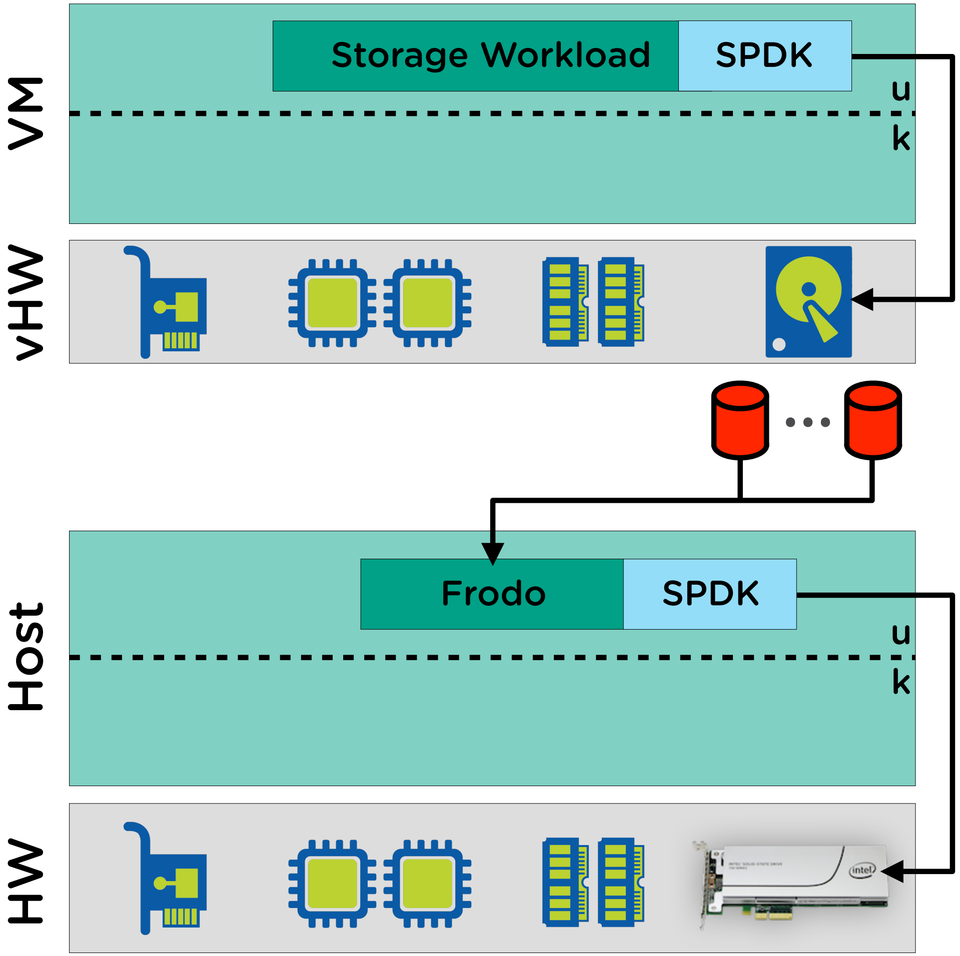
Hypervisor Analysis



Nutanix AHV under research

- SPDK is a userspace framework for driving storage controllers (NVMe)
- It means the controller must be used exclusively by one process
- Much better performance:
  - Super lean
  - No IRQs

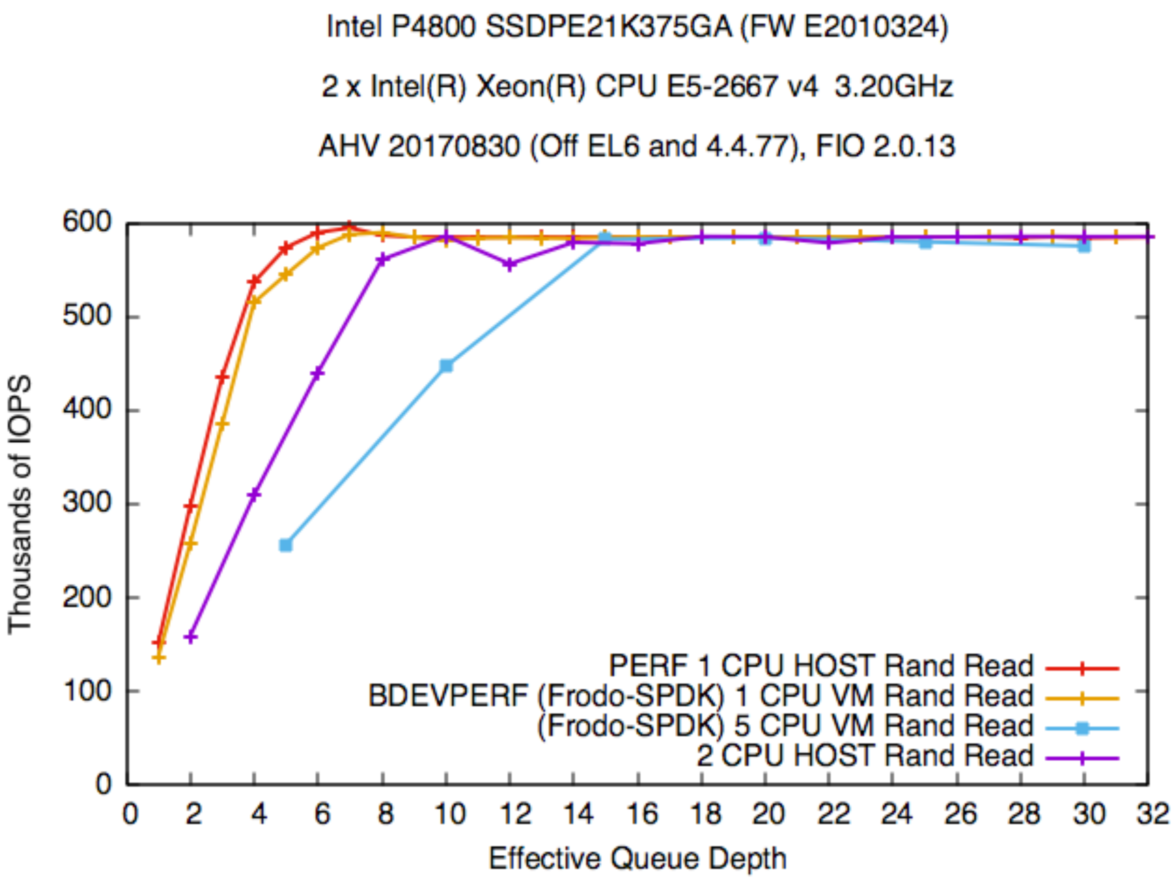
Hypervisor Analysis



Nutanix AHV under research

- VMs can also use SPDK!
- On AHV with virtio-scsi PMD
- Spins when reqs are outstanding
- Just like controllers don't have to IRQ the host, the hypervisor doesn't have to IRQ the VMs!

Nutanix AHV and SPDK



SPDK-Enabled Frodo

- Frodo (SPDK-enabled) adds ~700 ns overhead (prototype)
  - Guests using a kernel-datapath will hardly notice an overhead
  - Guests using SPDK may experience a minimal overhead

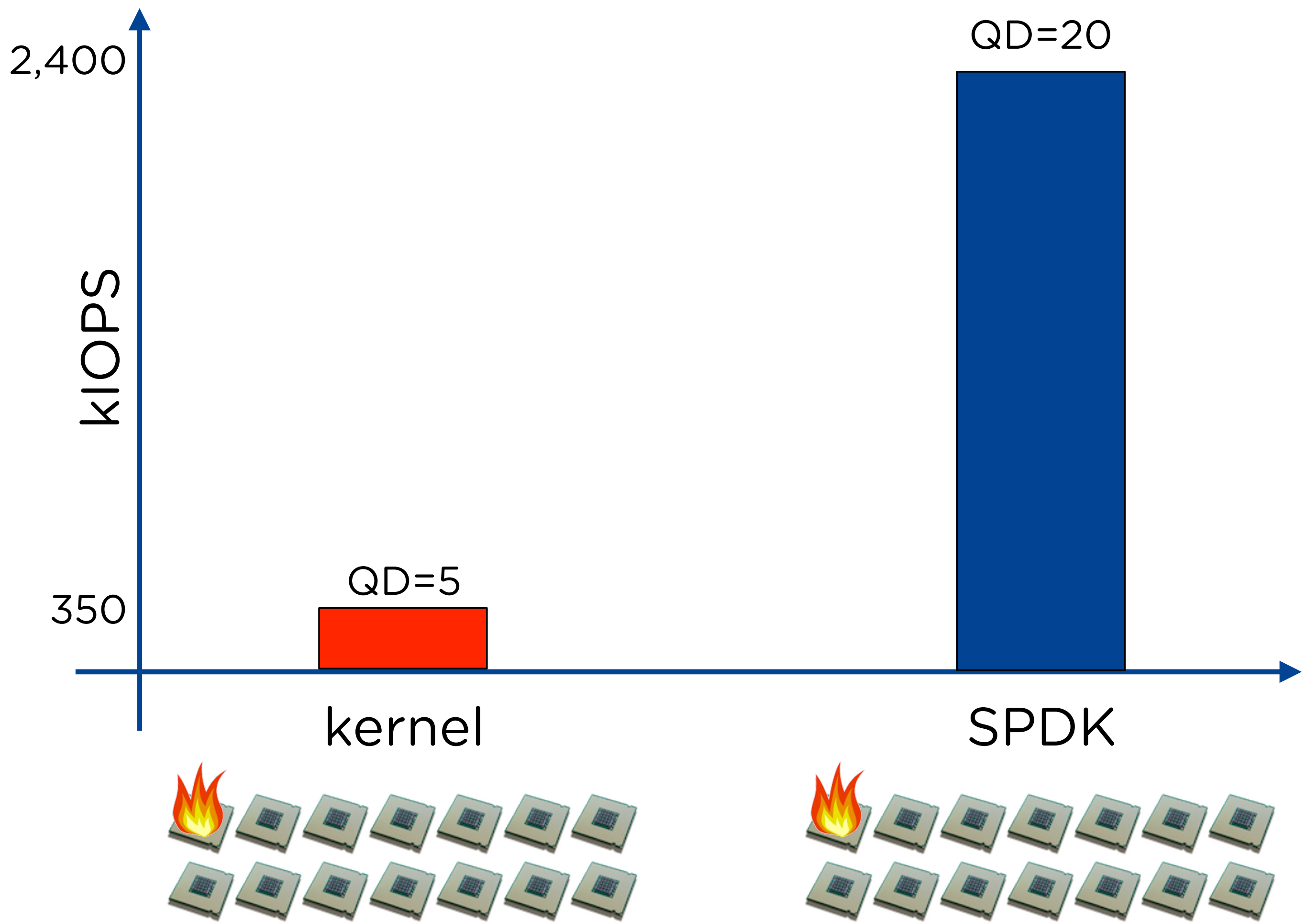
- SCSI (via virtio-scsi) becomes a bottleneck
  - At such speeds, virtio-scsi (via kernel) is very expensive
  - Emulating NVMe for guests becomes critical

- SPDK is less expensive (CPU-wise) than kernel
  - Cases where non-polling is cheaper are insignificant (eg. QD=1..3)
  - Saying "SPDK burns too much CPU" is a misconception

Driving FOUR Optane Drives

- What's the difference?
  - Kernel datapath is expensive: 350,000 IOPS using 1 core **
  - Userspace datapath: 2,400,000 IOPS using 1 core **

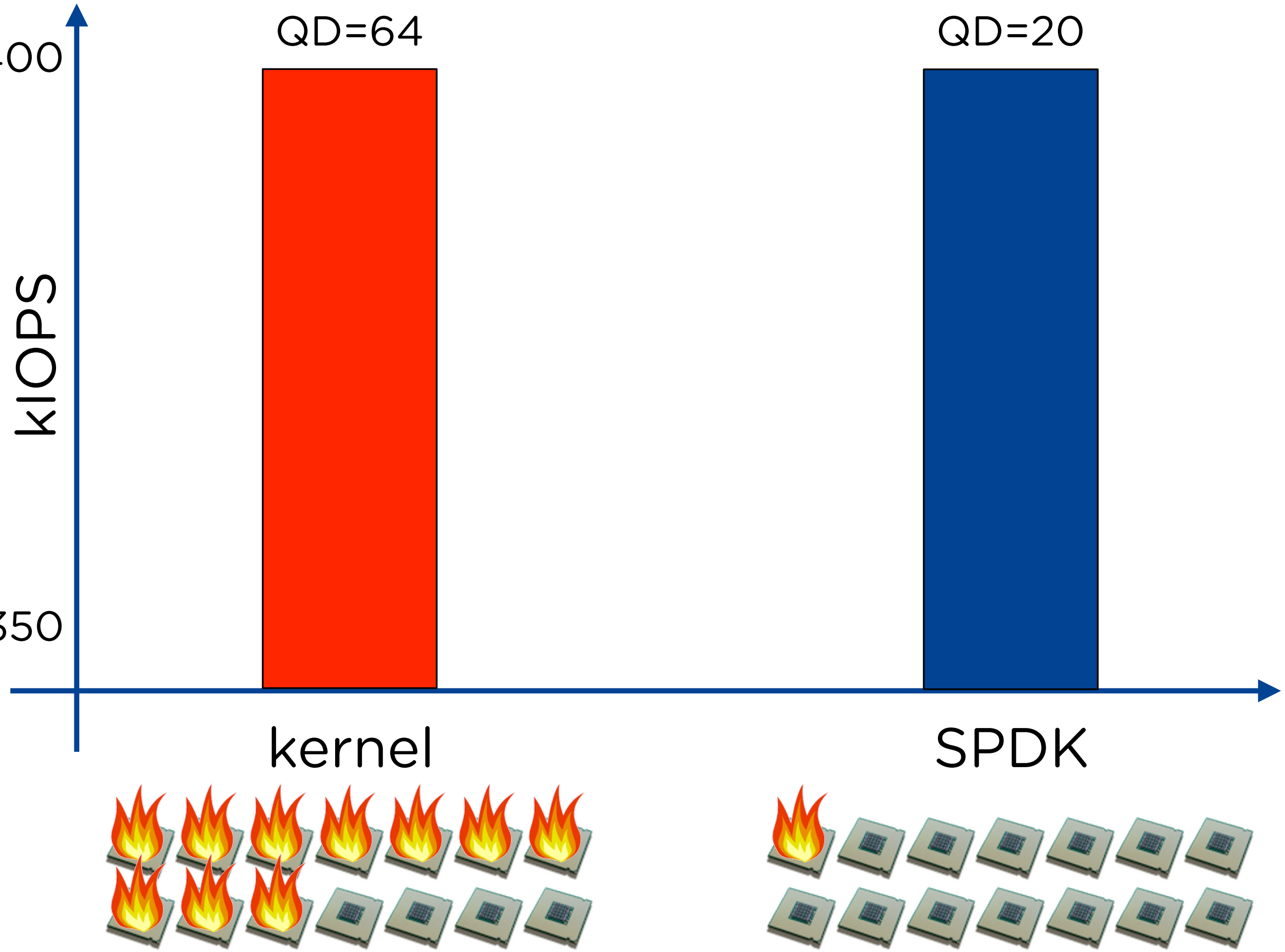
- Can kernel do the same?



** Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz,
Host Kernel 4.4.77-1.el6.nutanix.20170830.53.x86_64,
SPDK v17.10.1

Driving FOUR Optane Drives

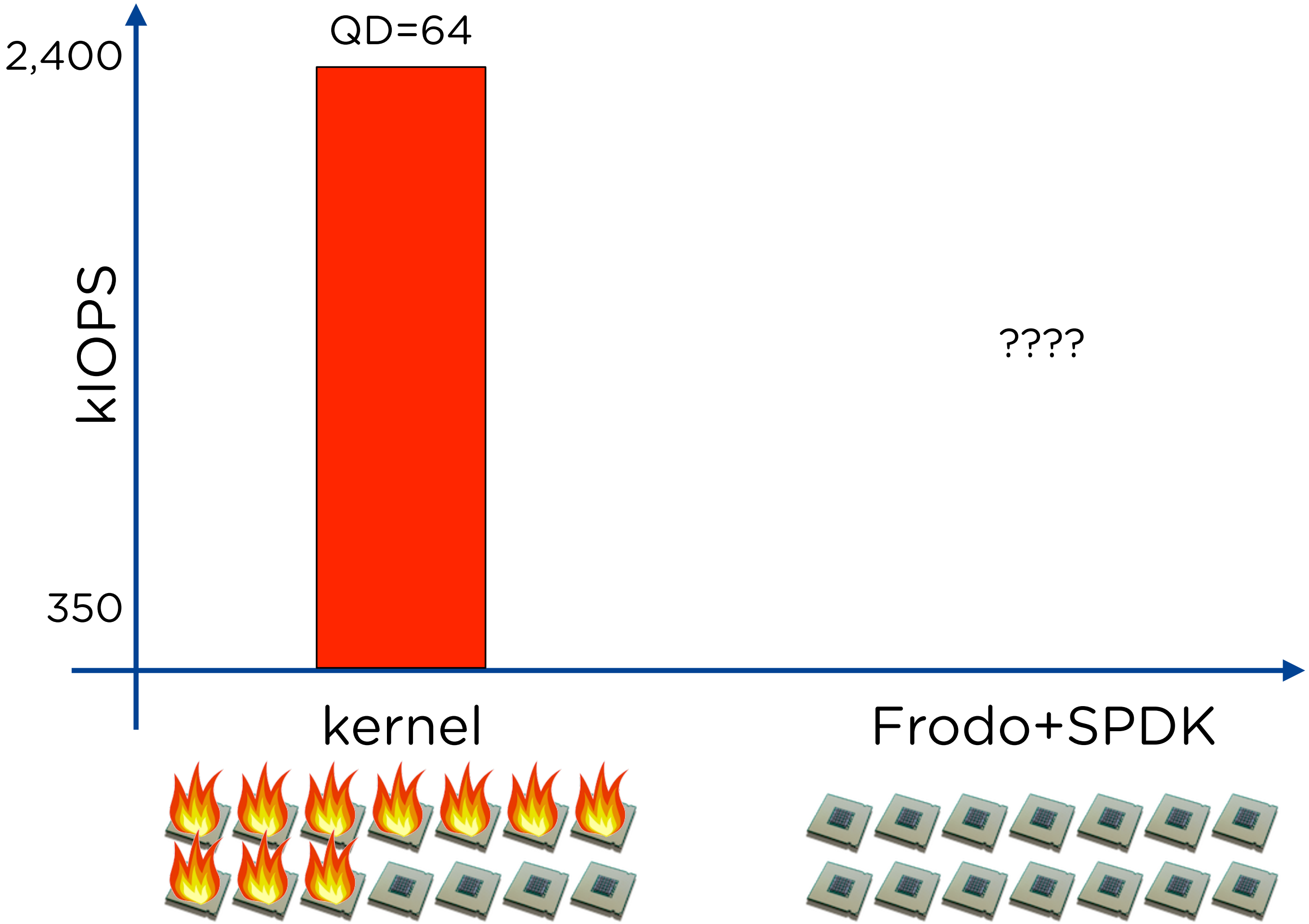
- What's the difference?
  - Kernel datapath is expensive: 350,000 IOPS using 1 core **
  - Userspace datapath: 2,400,000 IOPS using 1 core **
- Can kernel do the same?
  - Sure...
  - Require 13x the QD
  - Require 10x the CPU power



** Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz, Host Kernel 4.4.77-1.el6.nutanix.20170830.53.x86_64, SPDK v17.10.1

What about Frodo+SPDK?

- Link Frodo with SPDK
  - Frodo spins on guests' VQs
  - Frodo spins on the device
  - Guest uses SPDK and spins on virtio-scsi device



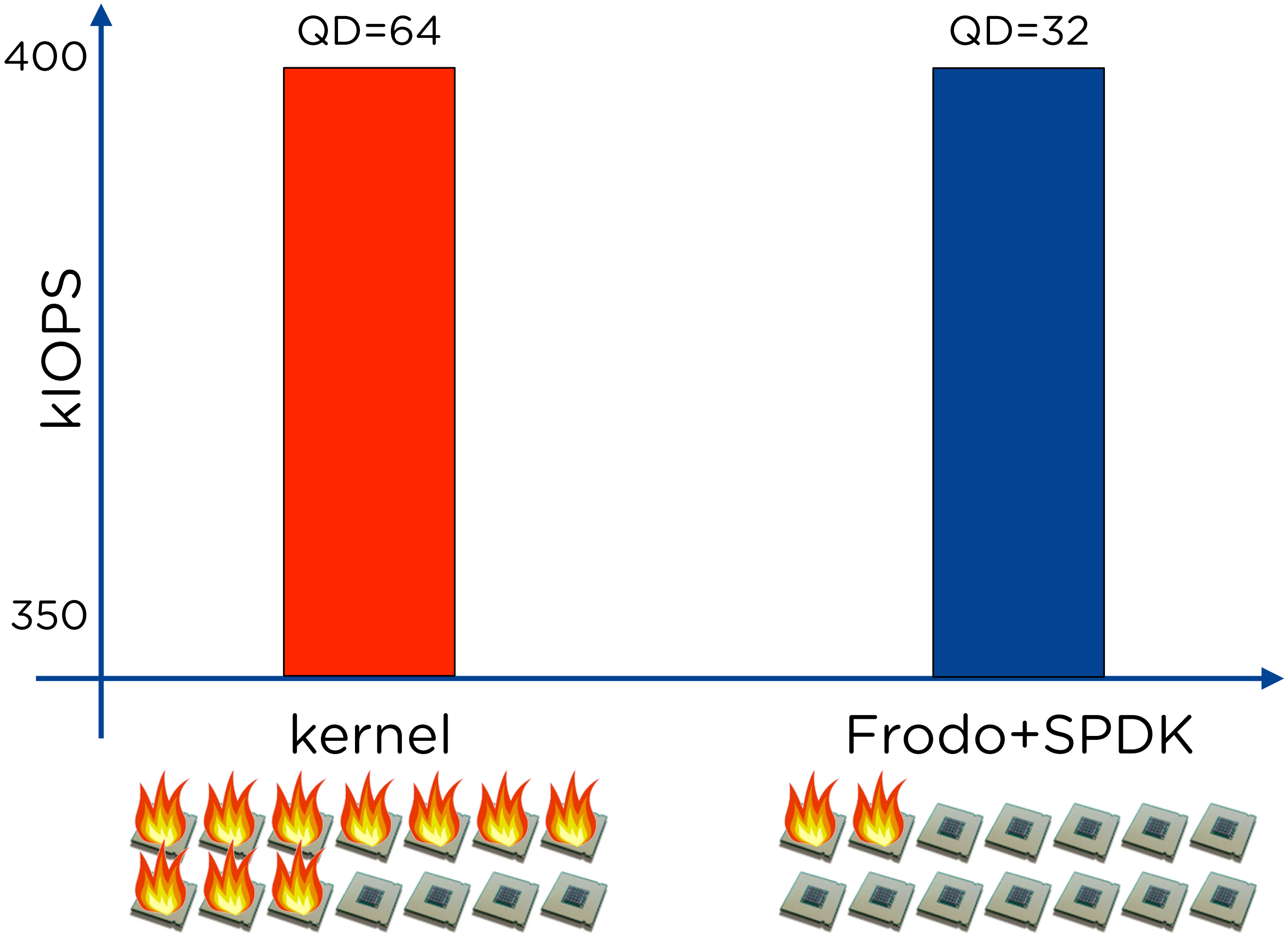
** Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz,
Host Kernel 4.4.77-1.el6.nutanix.20170830.53.x86_64,
SPDK v17.10.1

What about Frodo+SPDK?

- Link Frodo with SPDK
  - Frodo spins on guests' VQs
  - Frodo spins on the device
  - Guest uses SPDK and spins on virtio-scsi device

- Early-stage prototype!
  - We can probably do better

- Half the QD
- A FIFTH of the CPU Power



** Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz,
Host Kernel 4.4.77-1.el6.nutanix.20170830.53.x86_64,
SPDK v17.10.1

Summary

- Faster storage devices = Harder to virtualise
  - Time spent on CPU more noticeable, results in higher overhead
  - Require careful design for parallel storage access (MQ)

- Userspace-only leaner stack with SPDK
  - Leaner software = lower (CPU) latency
  - Spinning also cuts notification overhead between VM and HOST

- Hypervisors can share NVMe between VMs efficiently
  - Hypervisor uses SPDK for fast and efficient NVMe access
  - VMs can access the same NVMe, using SPDK or not

NUTANIX™

Thank you