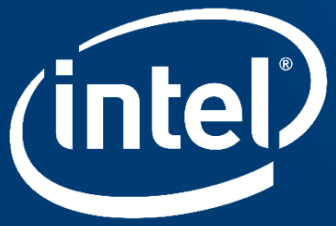


vhost-user-scsi: offloading virtio-scsi to userspace

Felipe Franciosi (AHV Engineering, Nutanix)
Jim Harris (SPDK Architect, Intel)

27 October 2017

NUTANIX™

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.

NUTANIX

Legal Disclaimer

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

All products, computer systems, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice.

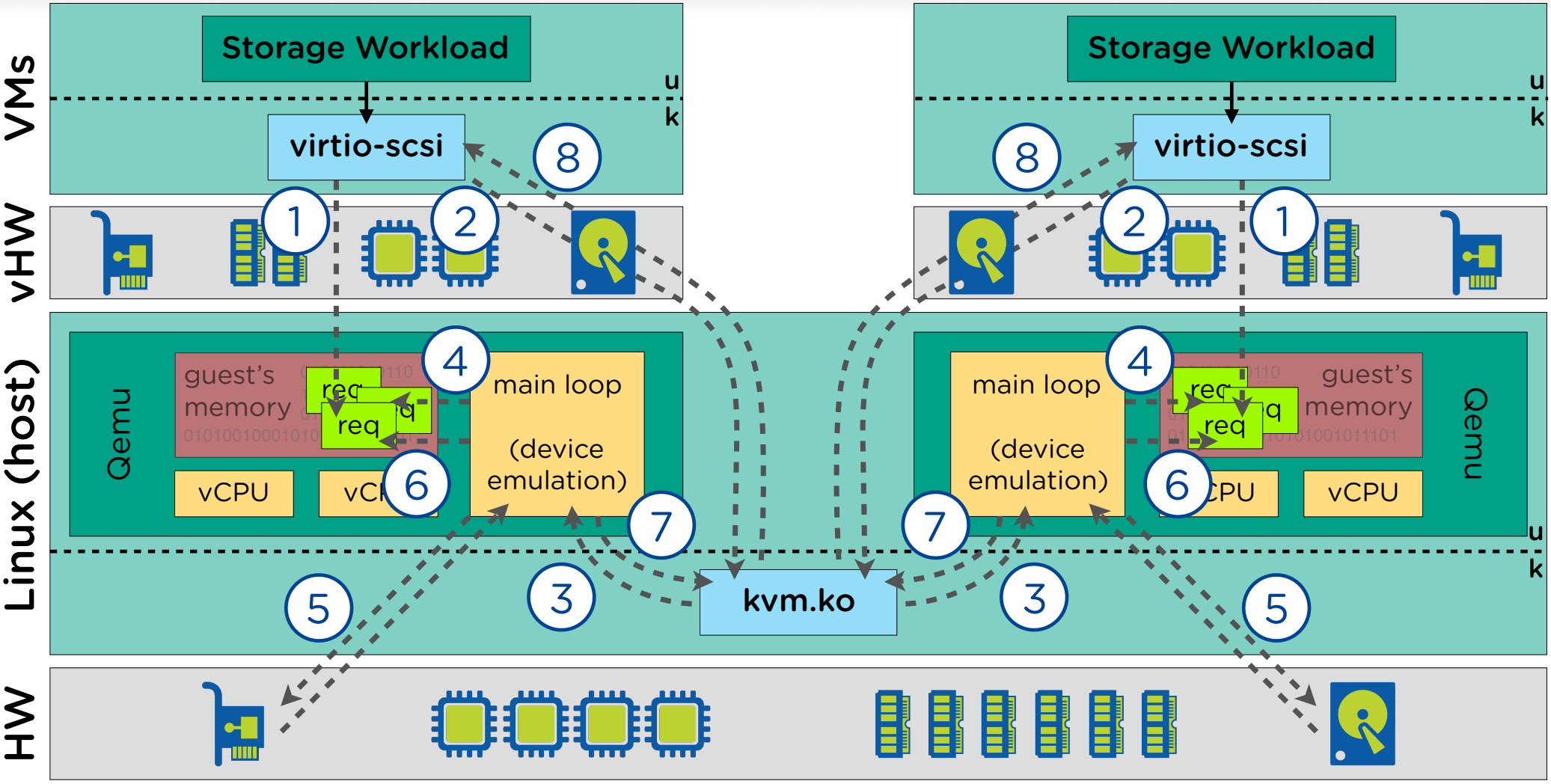
This document contains information on products in the design phase of development.

Applies only to halogenated flame retardants and PVC in components. Pursuant to JEP-709, halogens are below 1,000ppm bromine and 1,000ppm chlorine. The replacement of halogenated flame retardants and/or PVC may not be better for the environment.

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

*Other names and brands may be claimed as the property of others.

State of the Art: Qemu handles VQs

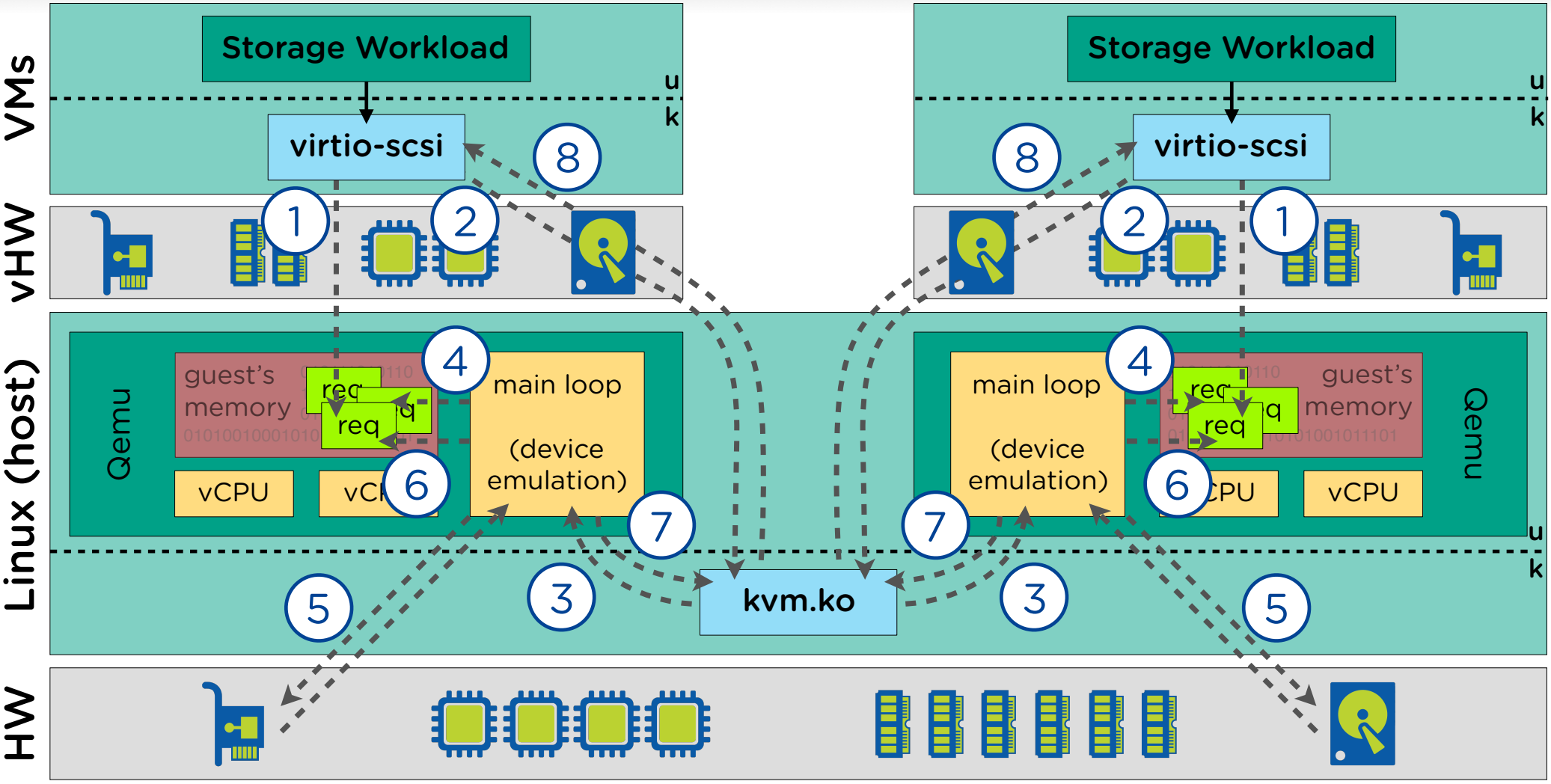


Motivation

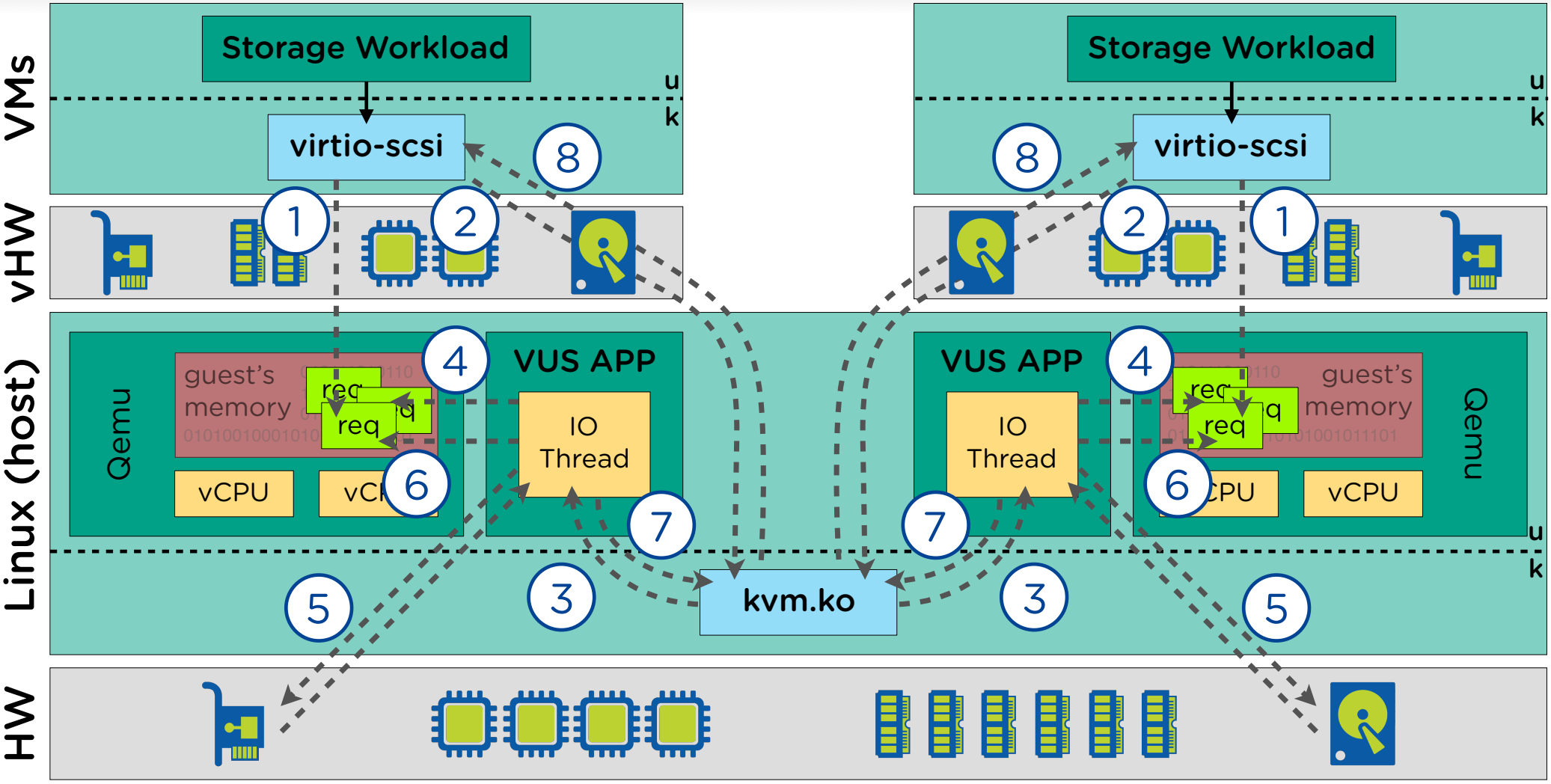
- Current design imposes certain limitations
  - A 1:1 mapping on VM to host process for handling VQs
  - Impossible to have a single process handling multiple VMs
  - Impossible to efficiently poll on multiple VMs
  - Non-trivial to virtualise a single NVMe to multiple VMs from userspace

- Solution
  - Create a mechanism to offload the datapath to a separate process
  - Precedence: vhost-user-net

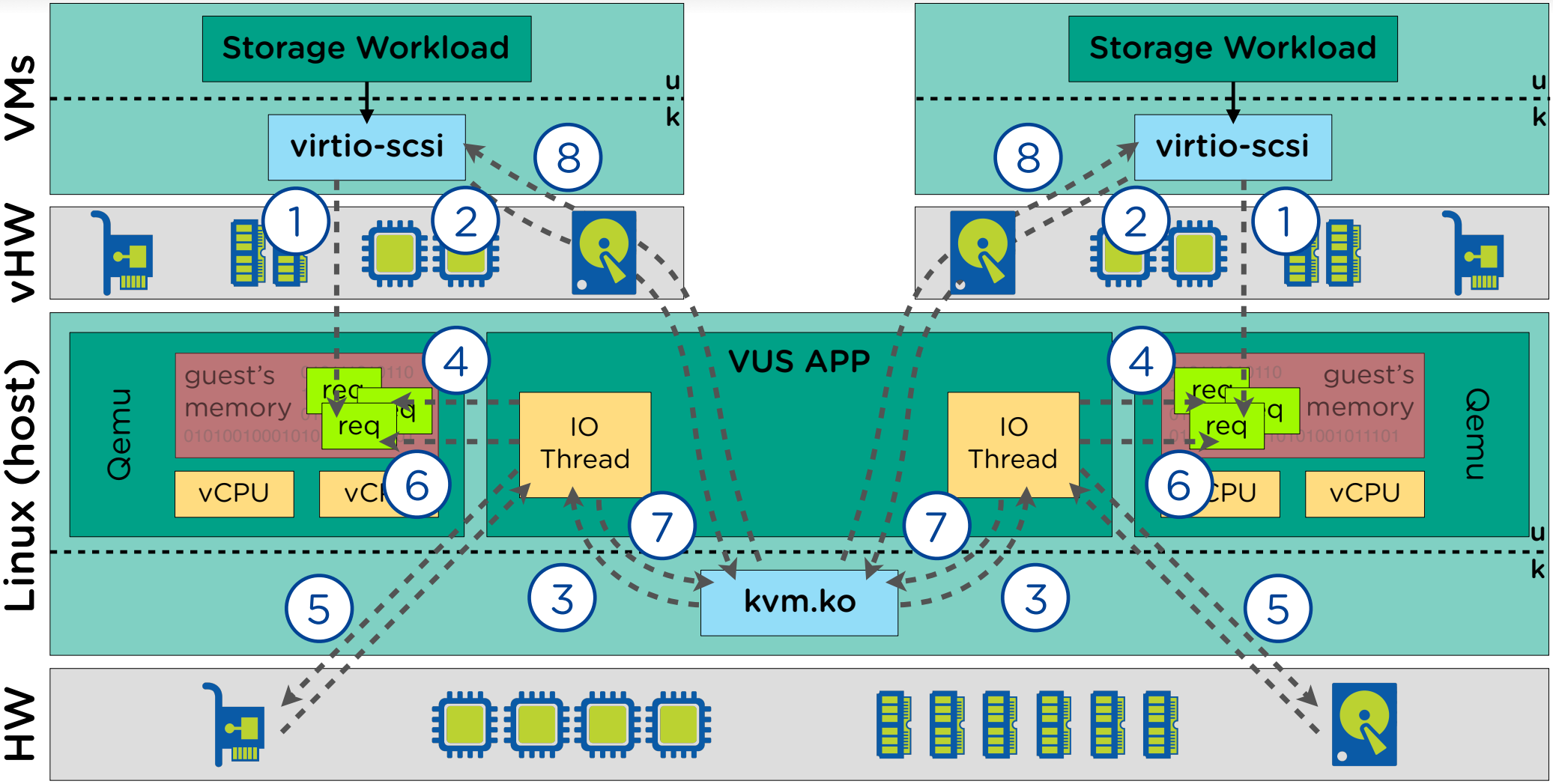
State of the Art: Qemu handles VQs



State of the Art: Qemu handles VQs



State of the Art: Qemu handles VQs

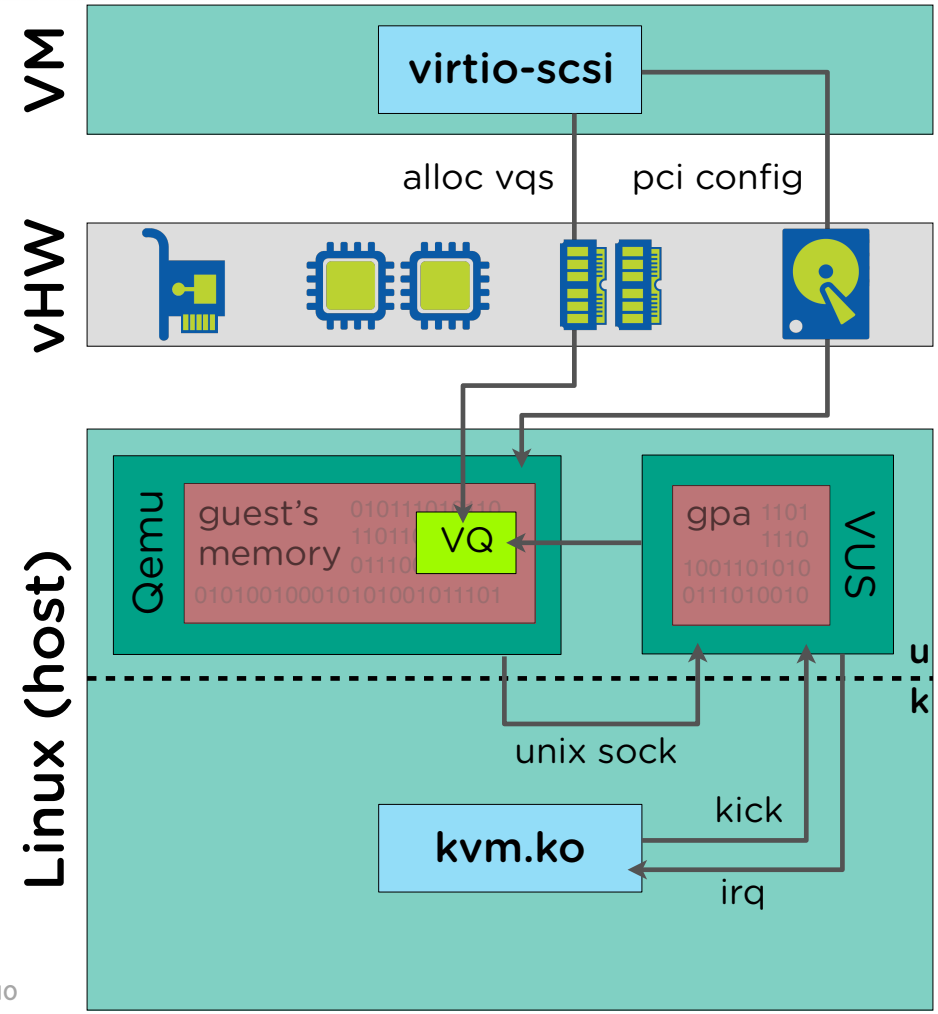


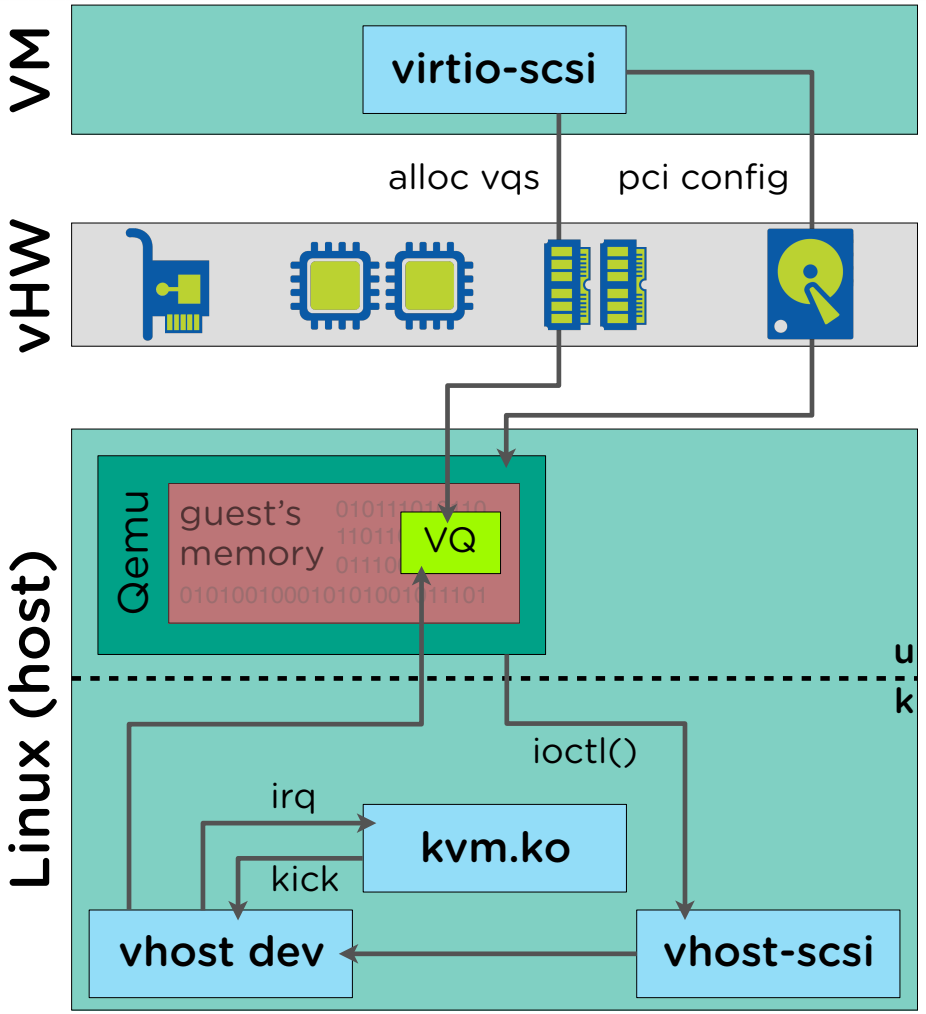
Trade-offs

- Main benefit: better performance
  - Easier to implement MQ support (with multiple IO threads)
  - Easier to optimise for better batching
    - First pick up everything from all VQs (multiple VMs)
    - Then submit it altogether to your backend
  - Simple to implement VQ polling for multiple VMs

- Drawbacks
  - When using a single backend process, security and stability
    - With separate processes, these are no different than today
  - LUN management is independent (Qemu is not aware of LUNs anymore)
  - Features generally available from Qemu block layer are lost

Side-by-side: vhost-user and vhost





Design/Implementation Highlights

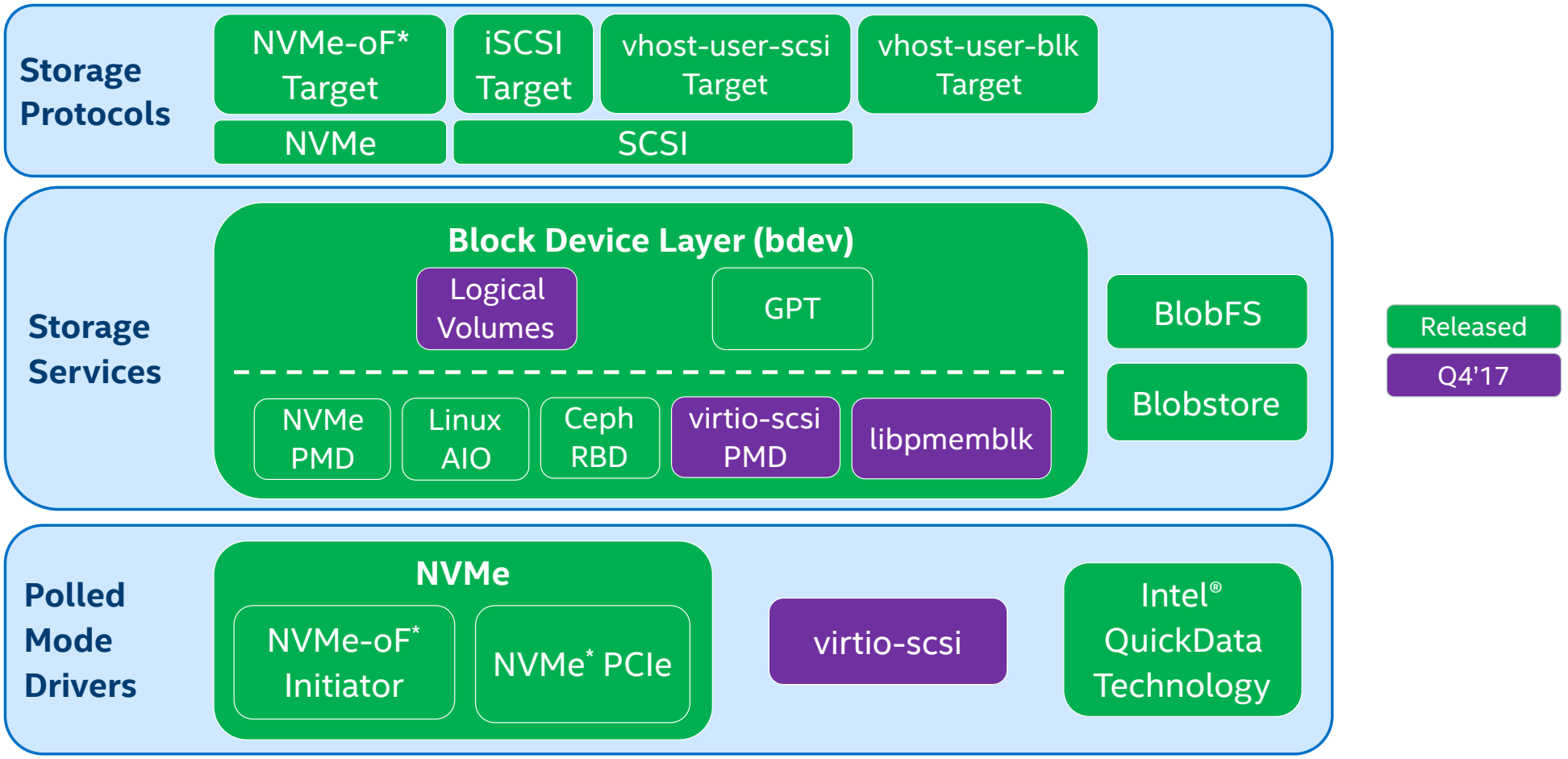
- Small refactor of vhost-scsi (in Qemu)
  - Original vhost-scsi split into: vhost-scsi-common and vhost-scsi
  - New vhost-user-scsi introduced under vhost-scsi-common
- Live Migration
  - Very straightforward, but ended up dropped from merge
    - Trick: flush the device on GET_VRING_BASE
  - Future work: throttling vhost(-user) devices for convergence
- Key difference with vhost: unix msgs are async
  - ioctl() blocks until the kernel module finished processing the command
  - On vhost-user, use F_REPLY_ACK to wait for completion

Storage Performance Development Kit (SPDK)

What is SPDK?

- Userspace polled-mode drivers, libraries and applications for storage, storage networking and storage virtualization
- Leverages DPDK for hugepage memory management and PCI device enumeration
- Started in 2013, open sourced in 2015
- BSD licensed
- http://SPDK.io

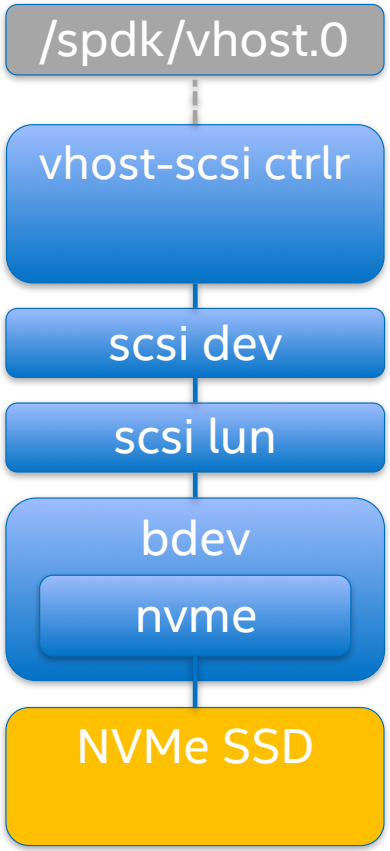
Storage Performance Development Kit (SPDK)

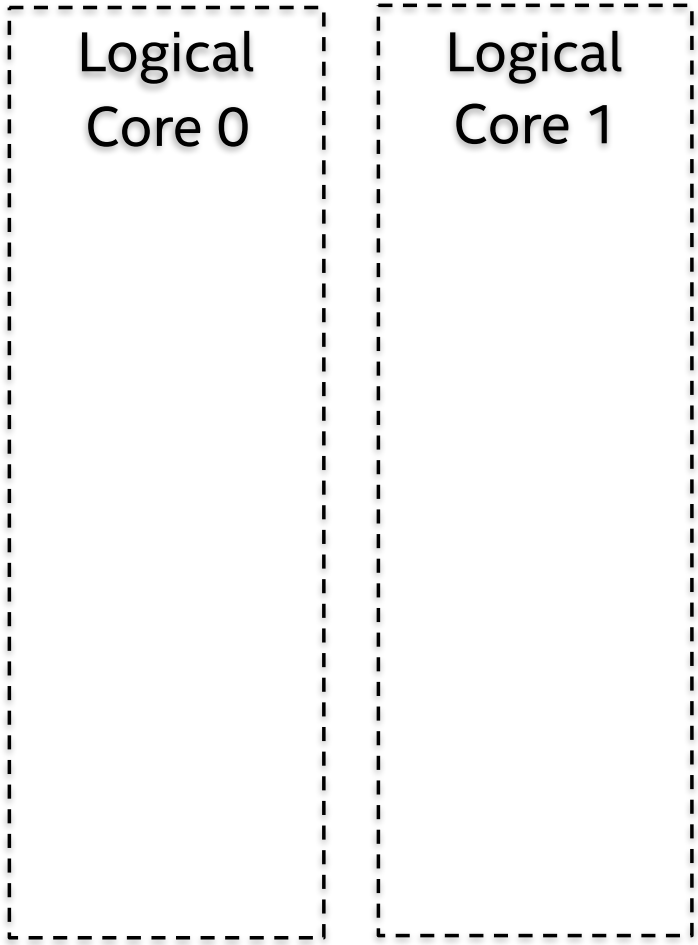


Basic Architecture

Configure vhost-scsi controller

- JSON RPC
- creates SPDK constructs for vhost device and backing storage
- creates controller-specific vhost domain socket





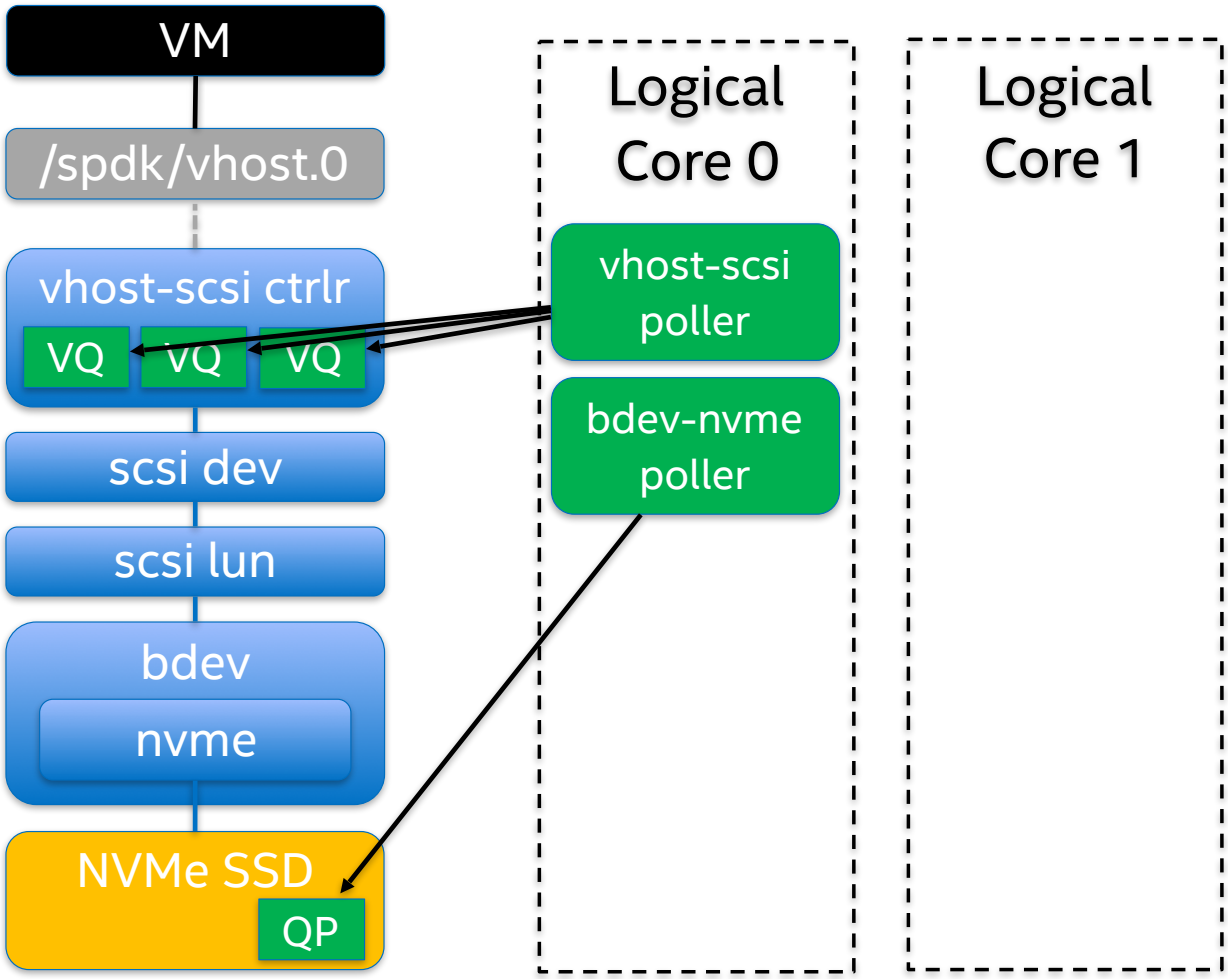
Basic Architecture

Launch VM

- QEMU connects to domain socket

SPDK

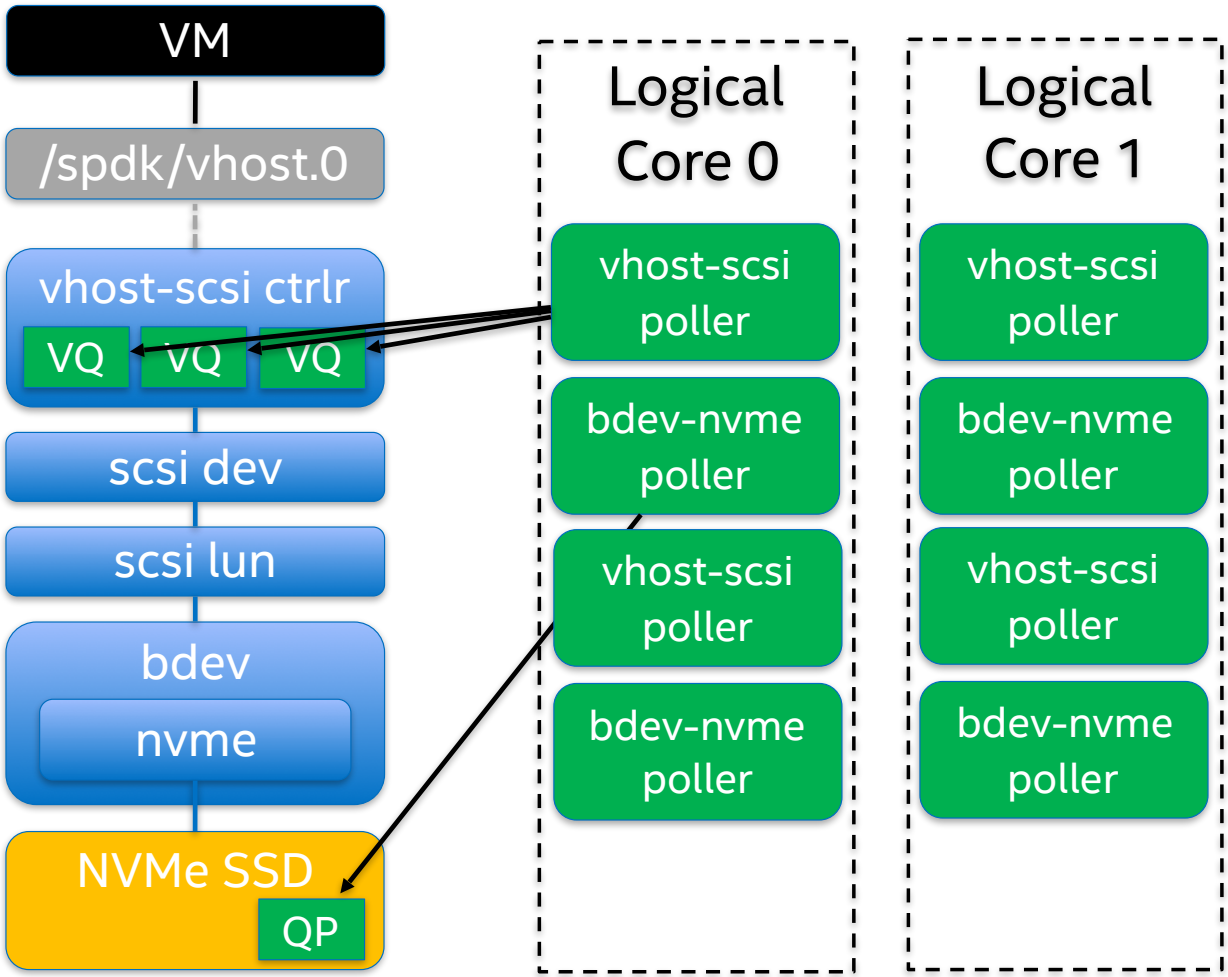
- Assigns logical core
- Starts vhost-scsi poller
- Allocates NVMe queue pair
- Starts bdev-nvme poller



Basic Architecture

Repeat for additional VMs

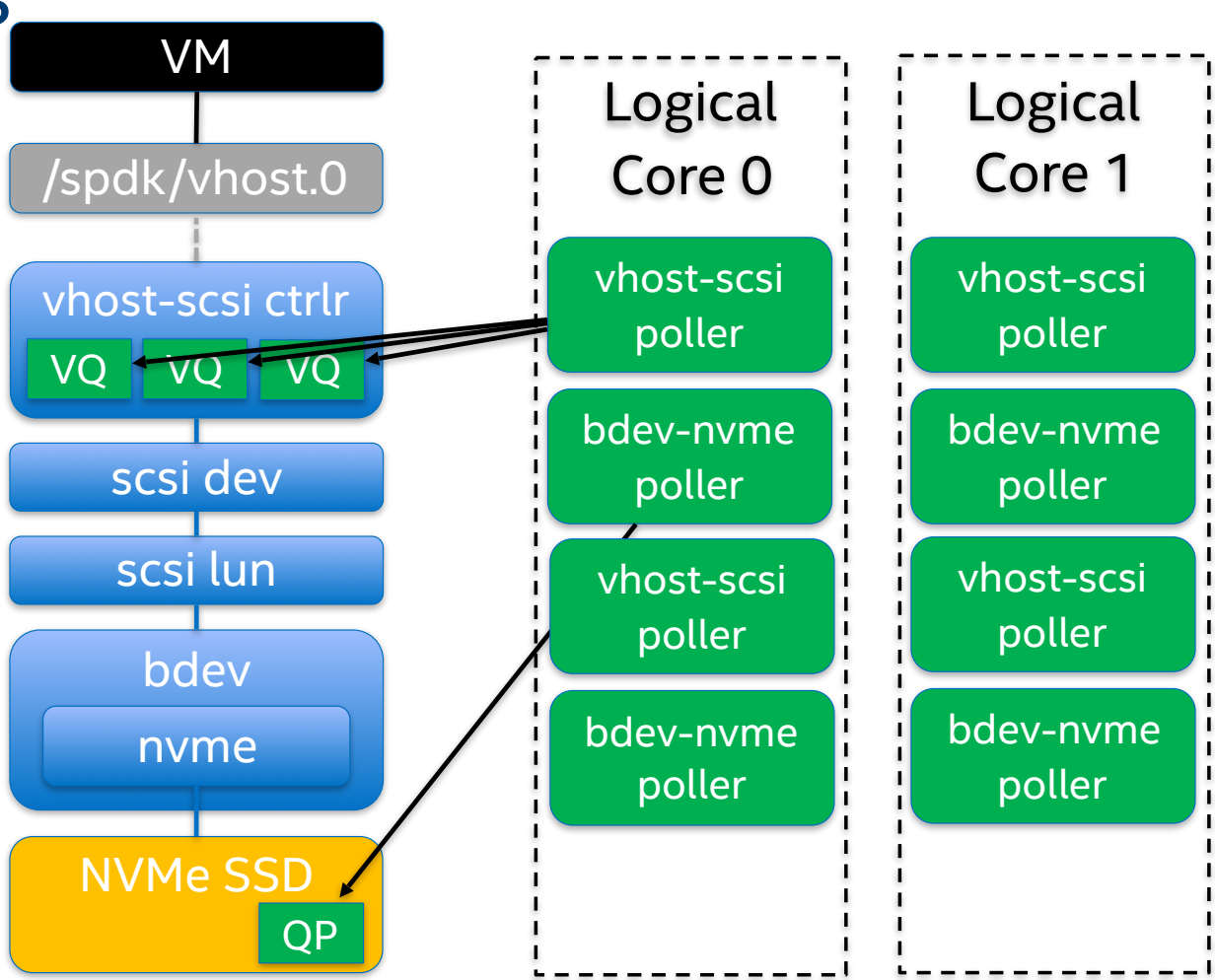
- pollers spread across available cores



Asynchronous Polling

Poller execution

- Reactor on each core
- Iterates through pollers round-robin
- vhost-scsi poller
  - poll for new I/O requests
  - submit to NVMe SSD
- bdev-nvme poller
  - poll for I/O completions
  - complete to guest VM



Sharing SSDs in userspace

Typically not 1:1 VM to local attached NVMe SSD

- otherwise just use PCI direct assignment

What about SR-IOV?

- SR-IOV SSDs not prevalent yet
- precludes features such as snapshots

What about LVM?

- LVM depends on Linux kernel block layer and storage drivers (i.e. nvme)
- SPDK wants to use userspace polled mode drivers

SPDK Blobstore and Logical Volumes!

Blobstore Design – Design Goals

- Minimalistic for targeted storage use cases like Logical Volumes and RocksDB
- Deliver only the basics to enable another class of application
- Design for fast storage media

Blobstore Design – High Level

Application interacts with chunks of data called blobs

- Mutable array of pages of data, accessible via ID

Asynchronous

- No blocking, queuing or waiting

Fully parallel

- No locks in IO path

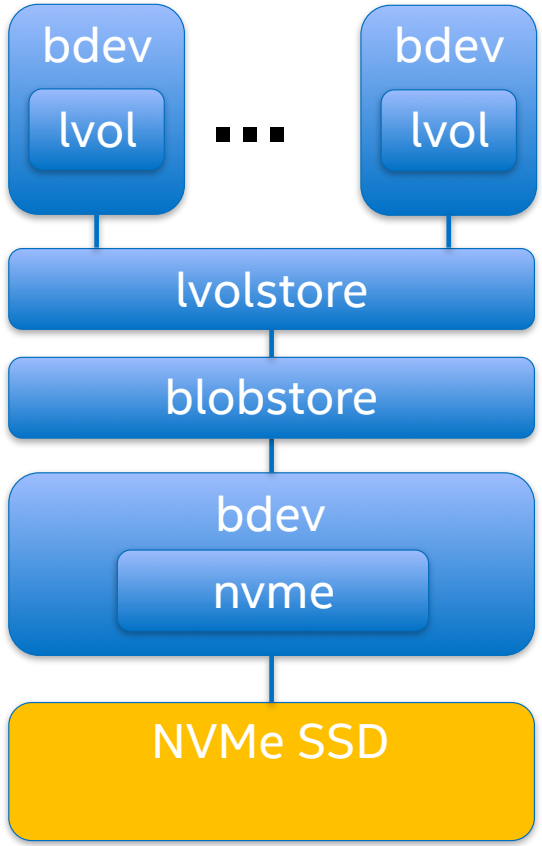
Atomic metadata operations

- Depends on SSD atomicity (i.e. NVMe)
- 1+ 4KB metadata pages per blob

Logical Volumes

Blobstore plus:

- UUID xattr for lvolstore, lvols
- Friendly names
  - lvol name unique within lvolstore
  - lvolstore name unique within application
- Future
  - snapshots (requires blobstore support)



SPDK vhost Performance - Configuration

2x Intel Xeon Platinum 8180 (28 cores each)

- VMs: 46 cores
- vhost: 10 cores

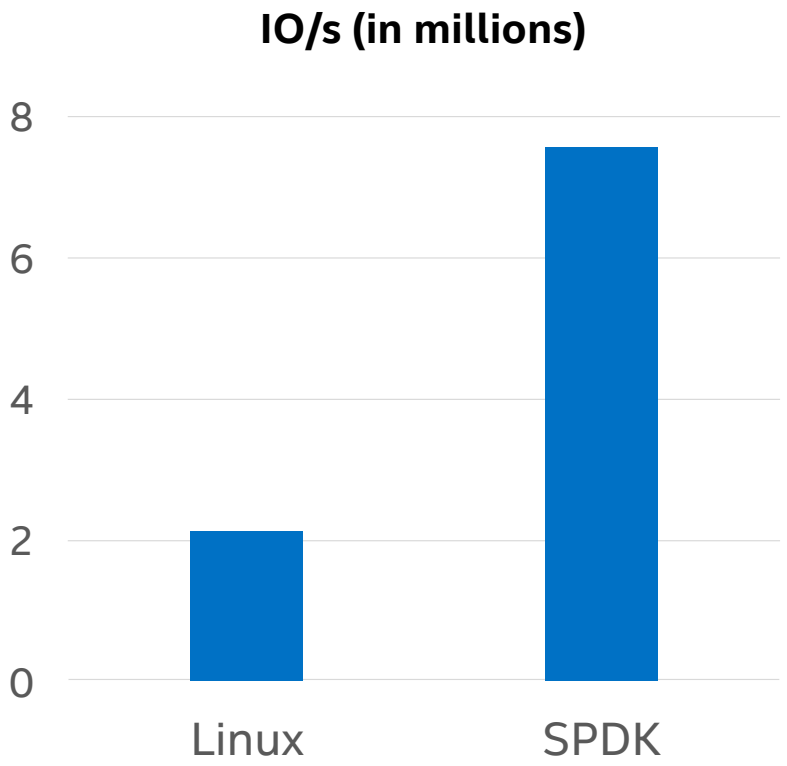
23x Intel P4800x Optane SSD

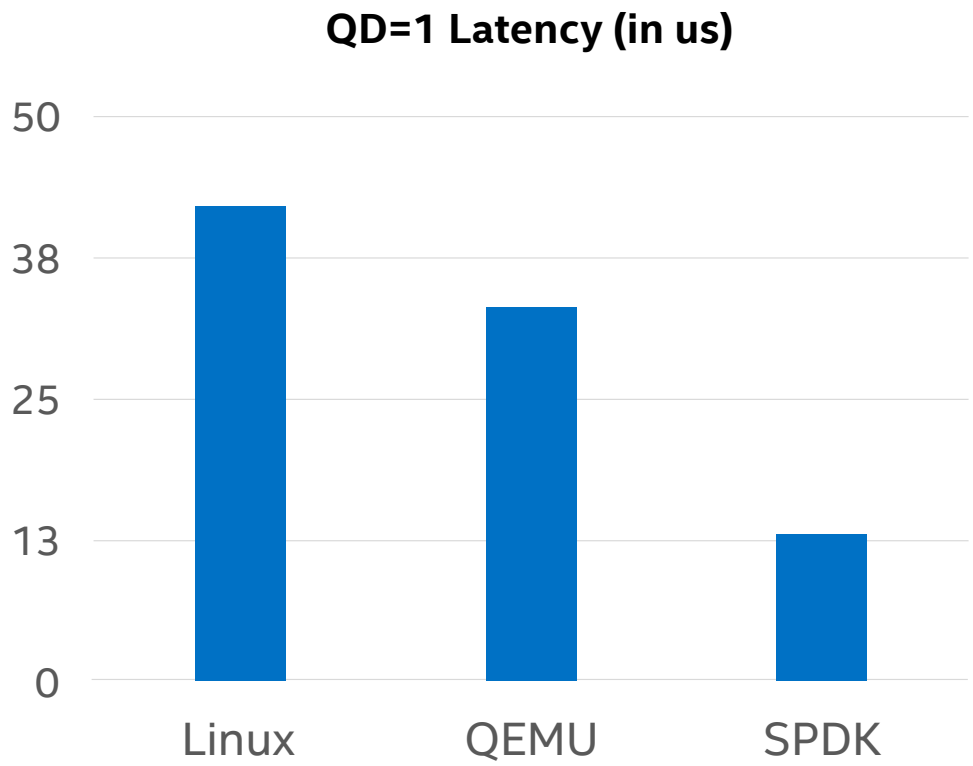
- SPDK lvolstore/LVM lvgroup per SSD

46 VMs – 1 vCPU, 2GB DRAM, 100GB logical volume

- 2 VMs per lvolstore/lvgroup

SPDK vhost Performance





System Configuration: 2S Intel® Xeon® Platinum 8180: 28C, E5-2699v3: 18C, 2.5GHz (HT off), Intel® Turbo Boost Technology enabled, 12x16GB DDR4 2133 MT/s, 1 DIMM per channel, Ubuntu* Server 16.04.2 LTS, 4.11 kernel, 23x Intel® P4800x Optane SSD – 375GB, 1 SPDK lvolstore or LVM lvgroup per SSD, SPDK commit ID c5d8b108f22ab, 46 VMs (CentOS 3.10, 1vCPU, 2GB DRAM, 100GB logical volume), vhost dedicated to 10 cores
As measured by: fio 2.10.1 – Direct=Yes, 4KB random read I/O, Ramp Time=30s, Run Time=180s, Norandommap=1, I/O Engine = libaio, Numjobs=1
Legend: Linux: Kernel vhost-scsi QEMU: virtio-blk dataplane SPDK: Userspace vhost-scsi

Future

vhost-user-blk

Live Migration

Logical Volume snapshots

SPDK Community

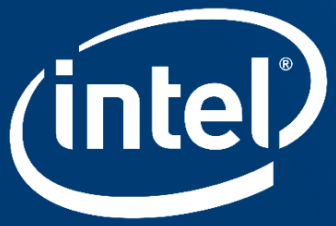
Home Page : http://www.SPDK.io/

Github : https://github.com/spdk/spdk

Trello : https://trello.com/spdk

GerritHub : https://review.gerrithub.io/#/q/project:spdk/spdk+status:open

IRC : https://freenode.net/ we're on #spdk



Thank you!

Questions?

james.r.harris@intel.com
felipe@nutanix.com

NUTANIX™