DPDK USERSPACE SUMMIT

September 6-8, 2022 • Arcachon, France • **#dpdkuserspace**

DPDK USERSPACE SUMMIT
September 6-8, 2022 • Arcachon, France
#dpdkuserspace

Virtualization of DPDK applications using virtio-vhost-user

Presenter: Usama Arif

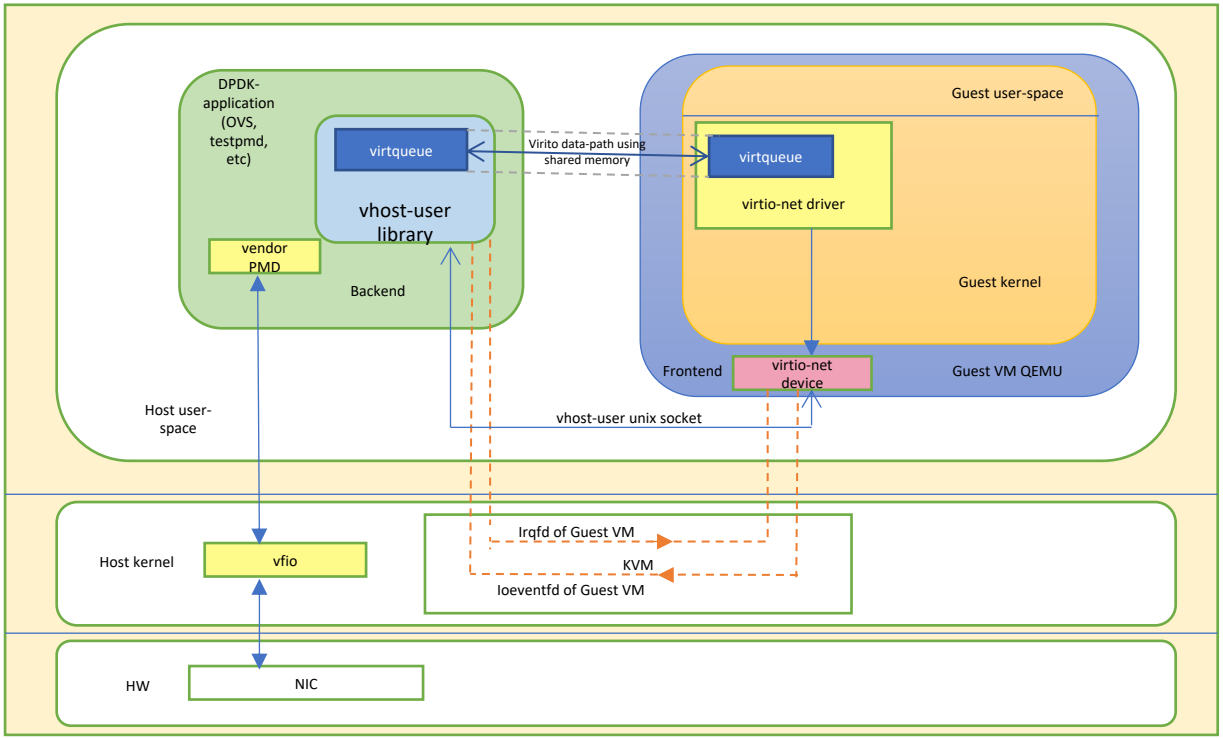
TikTok

Agenda

- Vhost-user introduction
- Why virtualize applications using vhost-user?
- Previous work
- Virtio-Vhost-User (VVU) design details
- Live update downtime
- Current status and next steps

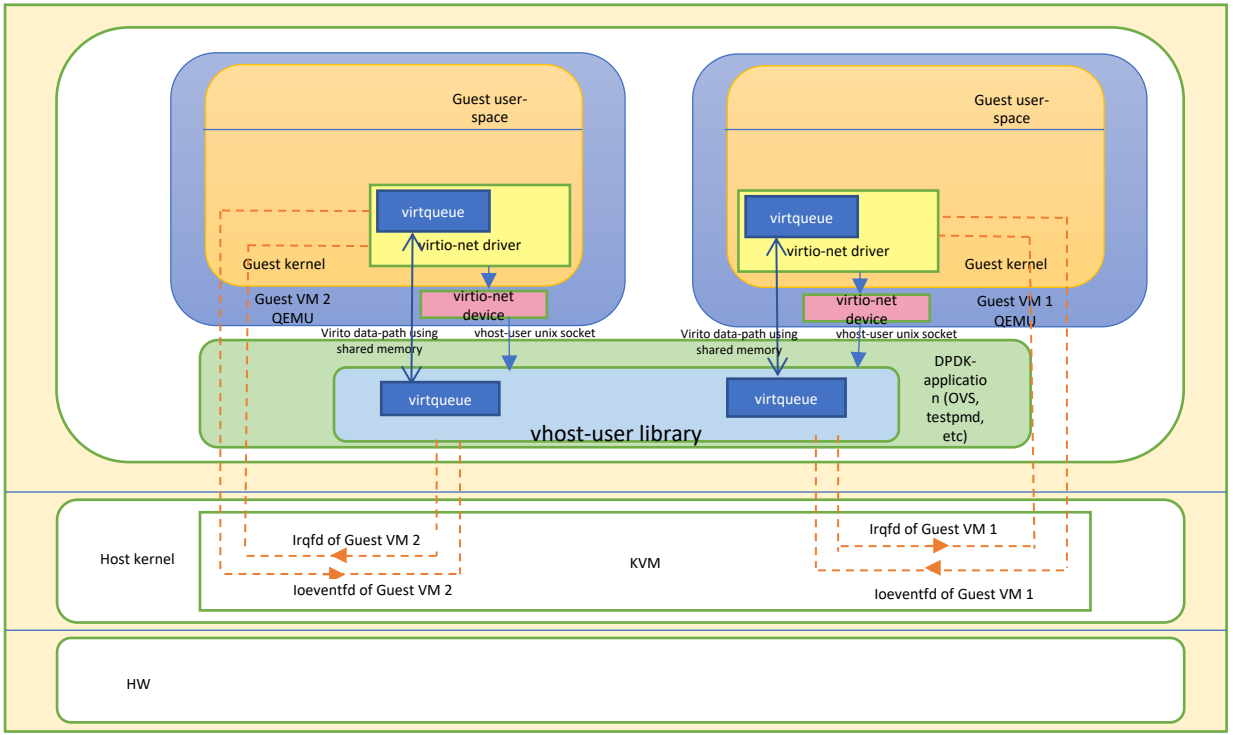
Introduction: vhost-user

- Vhost allows hypervisor to offload the data plane to either kernel driver (vhost-net) or userspace (vhost-user).
- Focus on vhost-user.



Introduction: vhost-user

- Vhost-user can be used to forward packets from one Virtual Machine to another.
- Used in Virtual Network Functions (VNF)



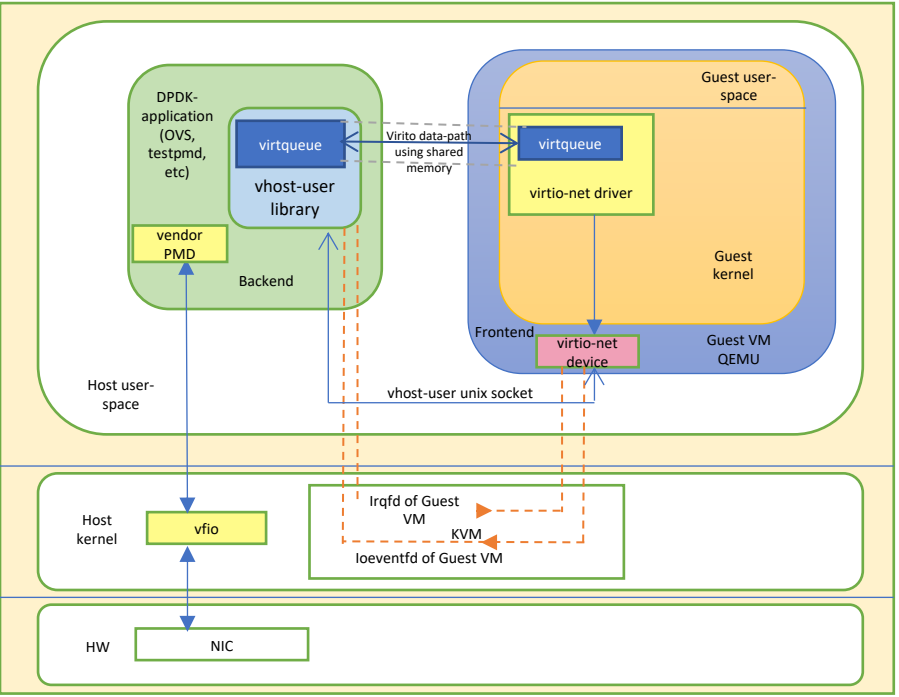
Why virtualize applications using vhost-user?

- Current implementation uses an “intermediate”
- State of host userspace process cannot be saved and restored
  - Increases downtime for host kernel live update
    1. Snapshot guest VMs
    2. Reboot into updated host kernel
    3. Resume guest VMs from snapshot
- Security benefits of running inside VM

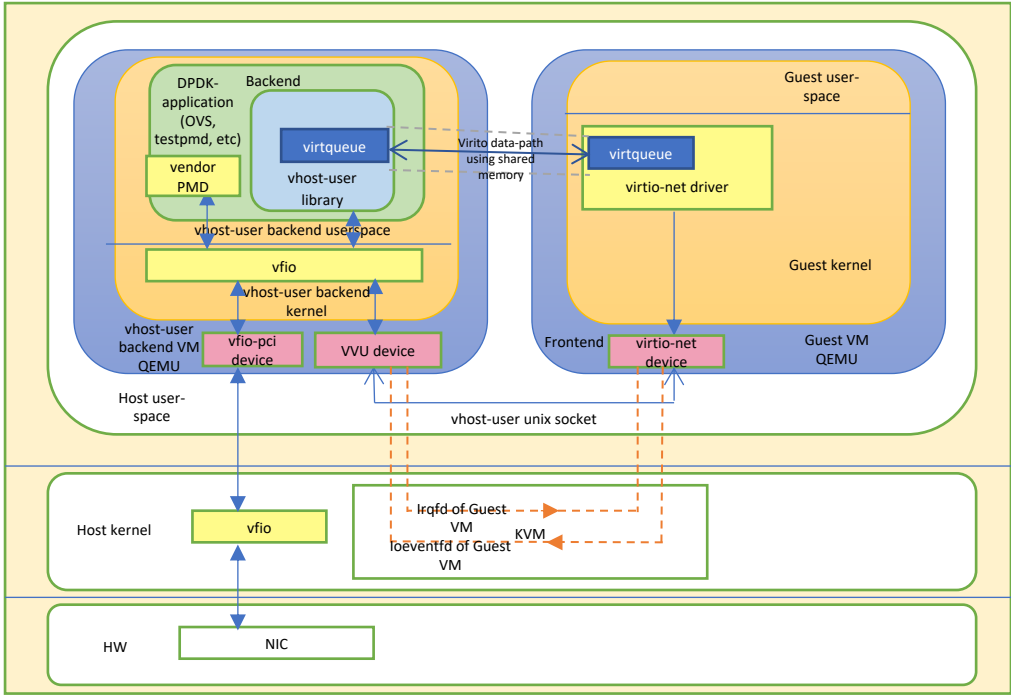
Previous work

- Vhost-pci (Wei Wang - Intel)
  - Needed a different kernel driver for each device: vhost-pci-blk, vhost-pci-net, etc.
- Virtio Vhost User: VVU (Stefan Hajnoczi - Redhat)
  - Single virtio-device type that exposes the vhost-user protocol instead of a family of new virtio device types, one for each vhost-user device type.

Virtio-vhost-user (VVU)



Existing vhost-user implementation

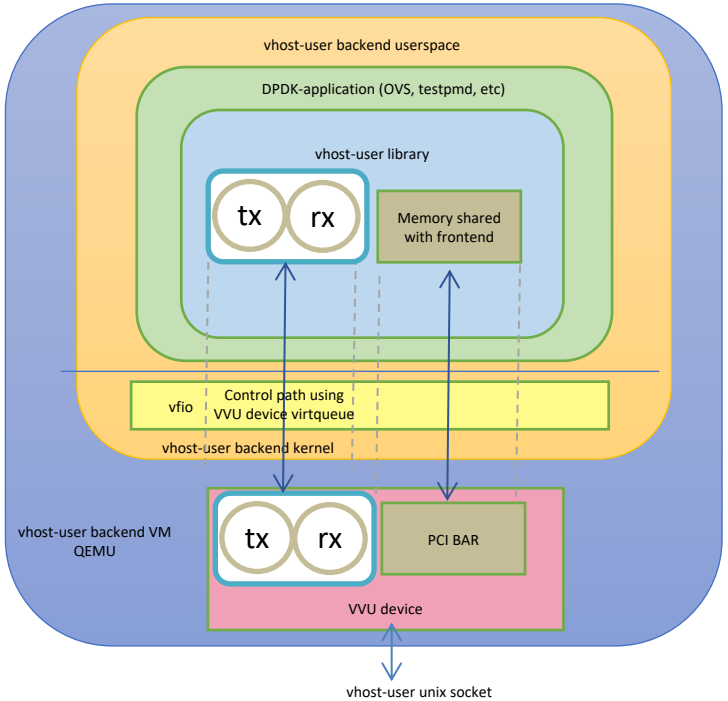


Virtualized vhost-user implementation
Virtio-vhost-user

virtio-vhost-user: Backend

Communication methods

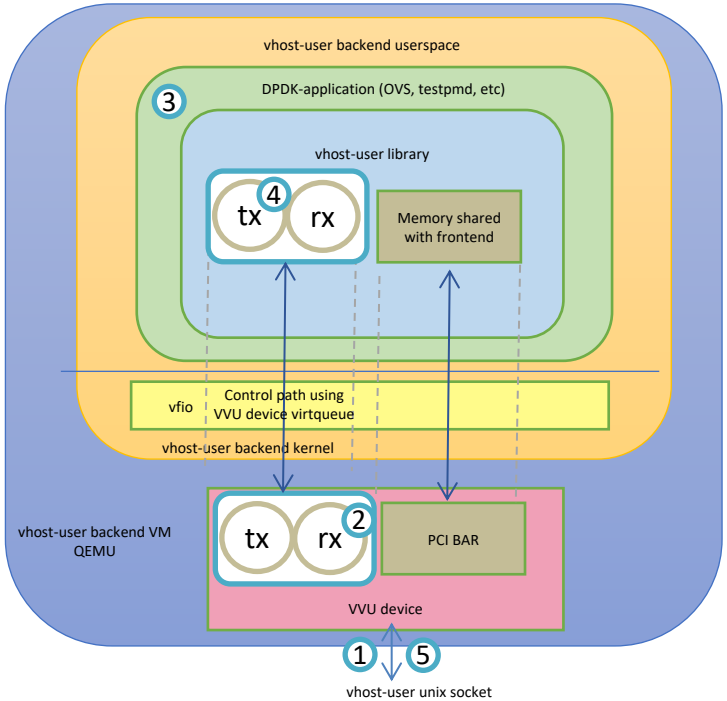
- Virtio virtqueue for forwarding parsed messages from vhost-user socket
- PCI BAR:
  - Notifications
  - Memory sharing
- Config change notification:
  - Interrupt handler in driver
  - Memory region I/O in device



virtio-vhost-user: vhost-user protocol

VHOST_USER_GET_FEATURES

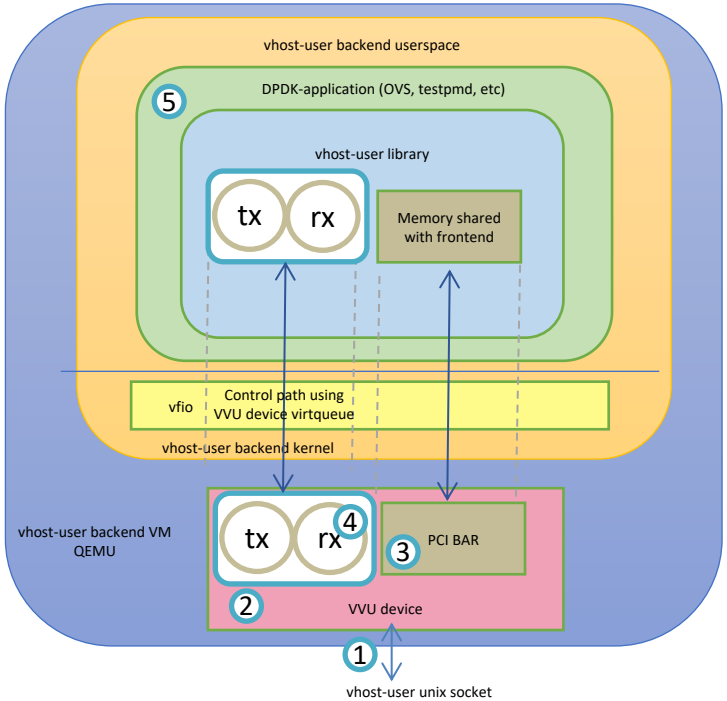
1. Message received by VVU device from vhost-user UNIX socket.
2. VVU device puts message on rxq for DPDK driver to read.
3. DPDK lib/vhost fills msg.payload.u64 with the features it supports.
4. DPDK sends reply on txq to VVU device.
5. VVU devices writes the reply back to UNIX socket.



virtio-vhost-user: vhost-user protocol

VHOST_USER_SET_MEM_TABLE

1. Message received from vhost-user UNIX socket, ancillary data contains an array of file descriptors for each memory mapped region.
2. The VVU device maps each of these vhost memory regions into its own address space using mmap.
3. The device presents those memory regions through PCI BAR of the vfio-pci device presented to the driver in the vhost-user library of DPDK.
4. VVU device puts message on rxq for DPDK driver to read.
5. The DPDK driver then saves the information so that it can translate the vring addresses.

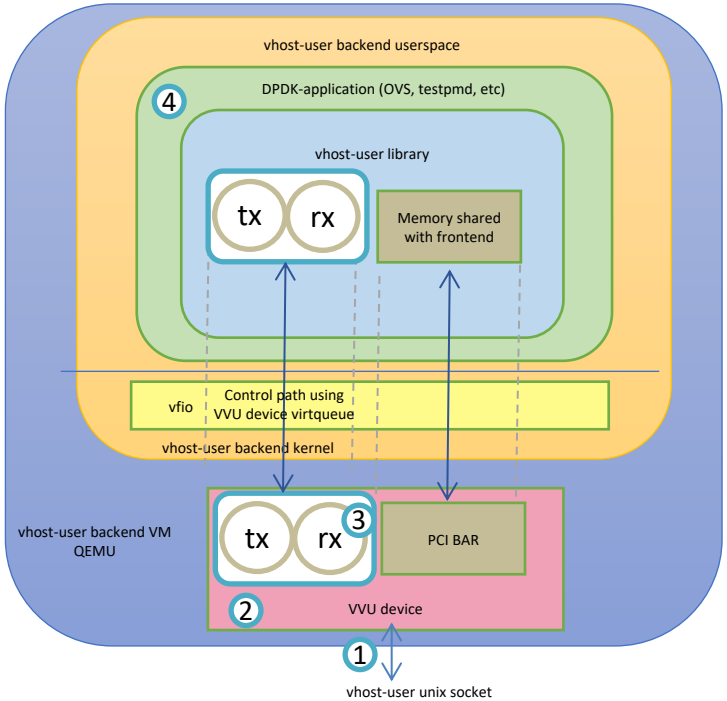


virtio-vhost-user: vhost-user protocol

VHOST_USER_SET_VRING_CALL

1. Message received from vhost-user UNIX socket, ancillary data contains the file descriptor to set when buffers are used.
2. The VVU device registers a doorbell (device auxiliary notification) within the PCI bar. Writes to that region triggers the eventfd.
3. VVU device puts message on rxq for DPDK driver to read.
4. The DPDK driver then saves the information in virtqueue callfd.

DPDK driver writes to doorbell register for the specific virtqueue index when buffers are used.



Live update downtime

- Same single guest VM run for both existing vhost-user downtime and VVU downtime test
- Existing vhost-user implementation in client mode:
  1.085 seconds
- VVU:
  0.480 seconds (55.7% faster)

Current status

- Virtio-spec changes: Introducing device/driver auxiliary notifications and VVU
  - https://uarif1.github.io/vvu/v2/virtio-v1.1-cs01.html
  - https://lists.oasis-open.org/archives/virtio-dev/202204/msg00022.html
- Working prototype:
  - https://uarif1.github.io/vvu/dpdk-vvu-instructions.md

Next steps

- Discuss approach with upstream community
- QEMU patches (VVU device): https://github.com/uarif1/qemu/tree/vvu_7.0.50
- DPDK patches (VVU driver): https://github.com/uarif1/dpdk/tree/vvu

Thank you!