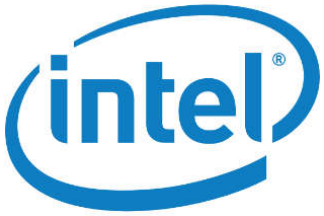


VDPA: VHOST-MDEV AS NEW VHOST PROTOCOL TRANSPORT

CUNMING(Steve) LIANG, Intel

cunming.liang AT intel.com

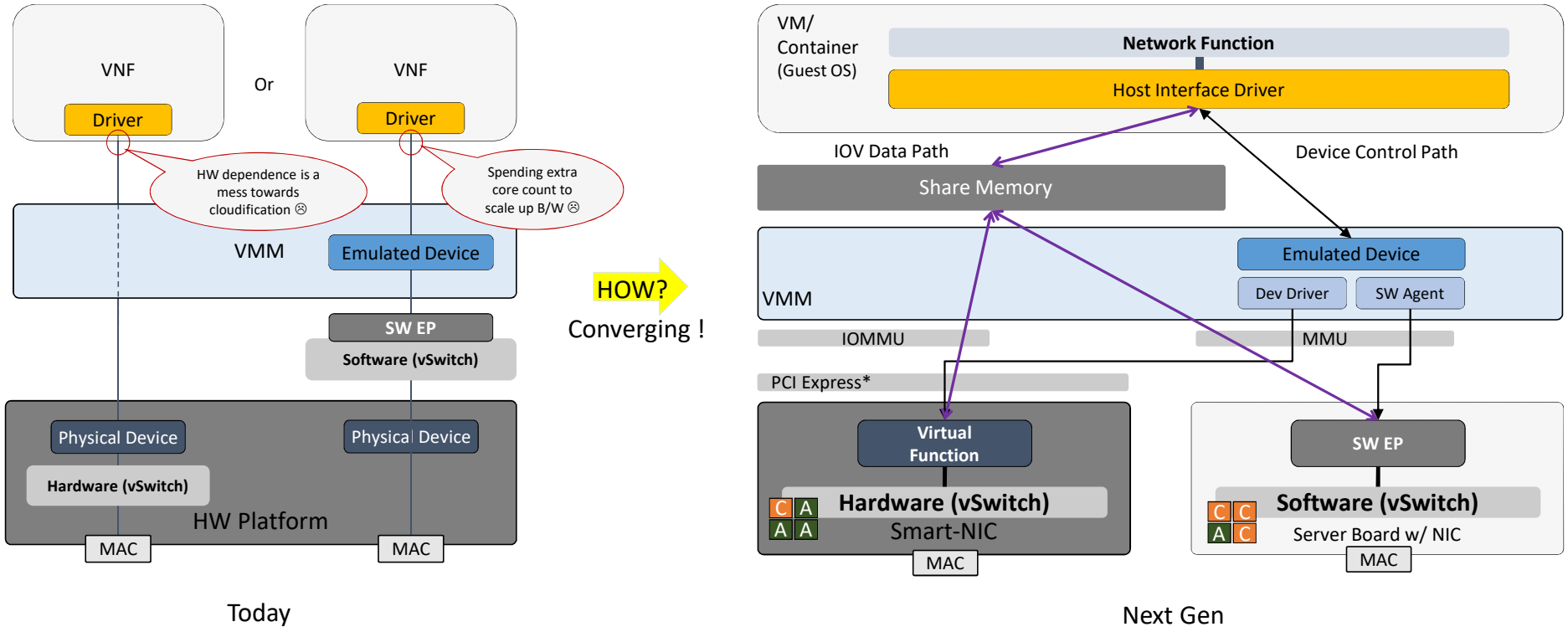
KVM Forum 2018, Edinburgh

October, 2018

Background

VISION: NFV NEXT-GEN VIRTUAL END POINT

- Transforming from virtualization to **Cloud-Ready** **Challenges**
- Incremental demands of VNF **B/W** (10Gb/s -> 25Gb/s -> 100Gb/s)



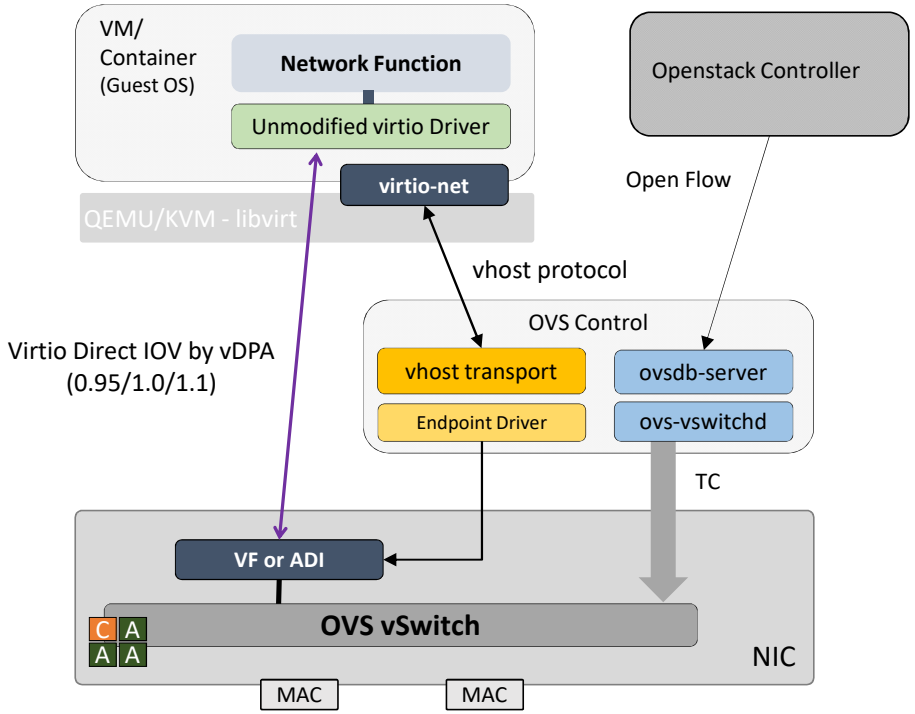
Today

Next Gen

VHOST DATA PATH ACCELERATION

- Direct Virtio IOV – I/O device read/write directly from/to the VM memory, performs like device assignment
- VDPA enables direct I/O to guest over a standard virtio device being emulated
- Benefits:
  - Device pass-thru equivalent packet rate
  - Live migration support
  - Unmodified virtio driver on uest
- DPDK 18.05, QEMU 3.0 as Pilot
  - Effort remains on vIOMMU supporting

http://lists.nongnu.org/archive/html/qemu-devel/2018-09/msg02216.html

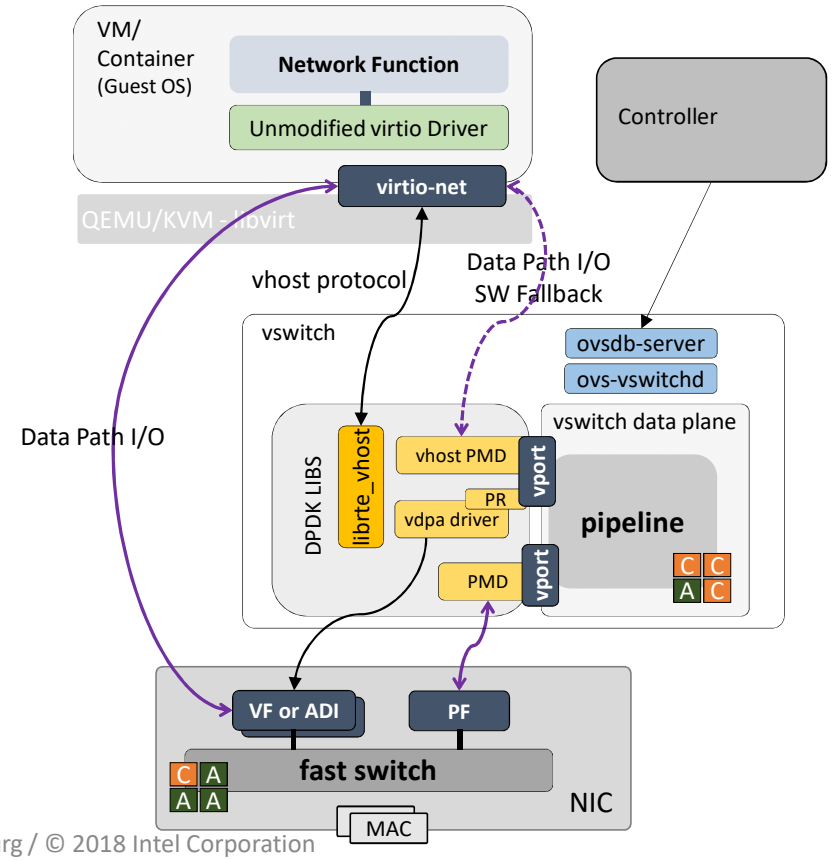


Virtio Direct IOV by VDPA

https://dpdksummit.com/Archive/pdf/2017Asia/DPDK-China2017-LiangWang-A-Better-Virtio-towards-NFV-Cloud.pdf
https://www.linux-kvm.org/images/8/87/KVM17vDPA-v4_0.pdf
https://www.dpdk.org/wp-content/uploads/sites/35/2018/09/xiao-wang-DPDK_Summit18_vDPA_for_vhost_acceleration-v4.pptx

HOW IT WORKS IN DPDK

- Add selective data path support in vhost-user library
- Associate device driver with vhost socket
- Attach with vswitch as port representor
- Being capable fallback to S/W data path



HUM, WHY NOT JUST PASS-THRU VIRTIO

- VDPA is Not Device Pass-thru
  - CANNOT live migrate from S/W backend to H/W and vice versa
  - CANNOT support stock VM using 0.95 w/ PIO
  - Any SPEC. change MAY impact more for large size H/W surface
  - Host endpoint CANNOT have more features than Guest endpoint
  - Lock H/W interface on VIRTIO is NOT Win-Win with IHVs
- VDPA is Not Vendor Specific
  - Min prerequisite is following SPEC. defined vring layout & operation
  - Optimal choice is supporting SPEC. defined doorbell
  - Even H/W specific design follows backend design interface

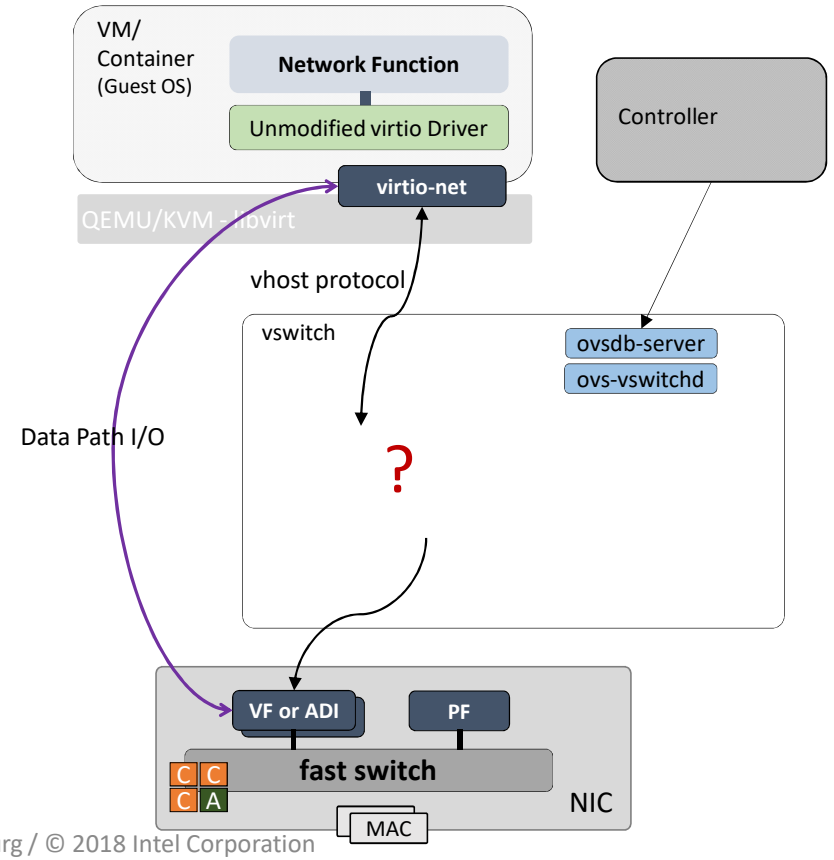
Data Path effort is not in today's agenda

Focus on the way to construct data path

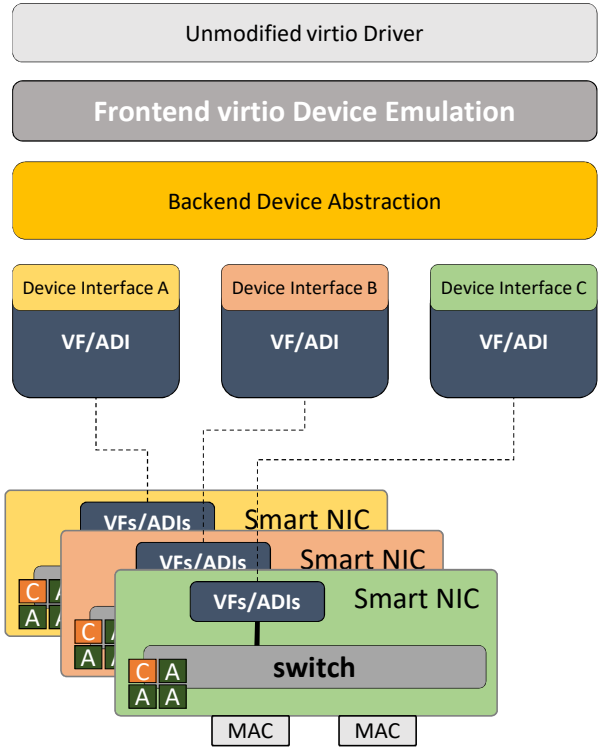
PROBLEM & PROPOSAL

WHAT ABOUT DATA PLANE SEPARATION

- Entire data plane separate to discrete system on card
- VMM emulates machine but no longer supply I/O
- Needs intermediate to connect vhost CMD/MSG and device driver

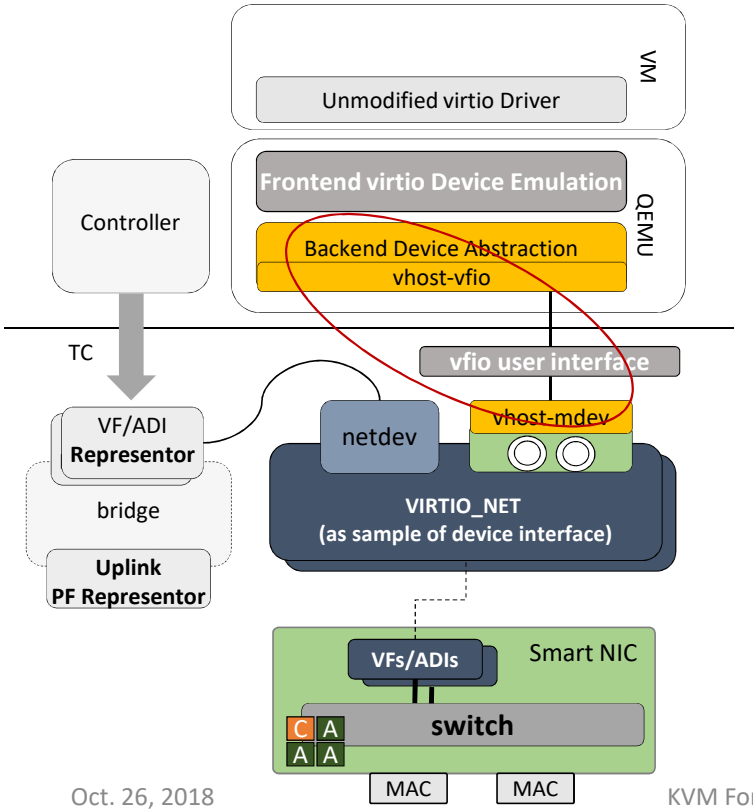


REVISIT VDPA DEVICE ABSTRACTION



- Different Device Interface
  - Not always exact virtio like
  - Needs backend device abstraction
- vhost + vfio = vhost-vfio
  - vhost: natural backend abstraction
  - vfio: unified device control for QEMU
- Consolidate direct virtio around vhost
  - Remains decoupling frontend/backend
  - Remains diverse backend adaption

MDEV BASED H/W VHOST BACKEND

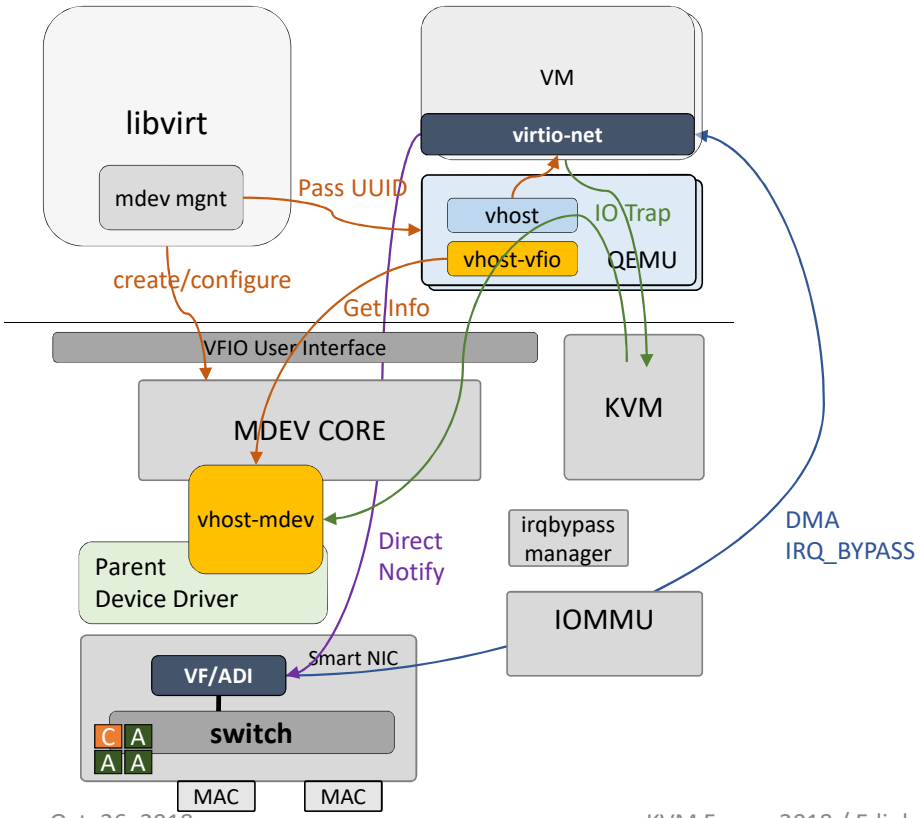


- QEMU: vfio-pci w/ quirks
  - Good for almost similar virtio device interface

  But what about
  - Other device interface
  - Control separation on the host
  - Host features beyond guest virtio device
- Kernel: vhost message over vhost-vfio
  - Reducing virtio spec. change impact
  - Independent with virtio emulation device
  - Avoid introducing IHV drivers in QEMU
  - Max flexibility on the hardware design

DEEP DIVE

INTRODUCING NEW NET CLIENT



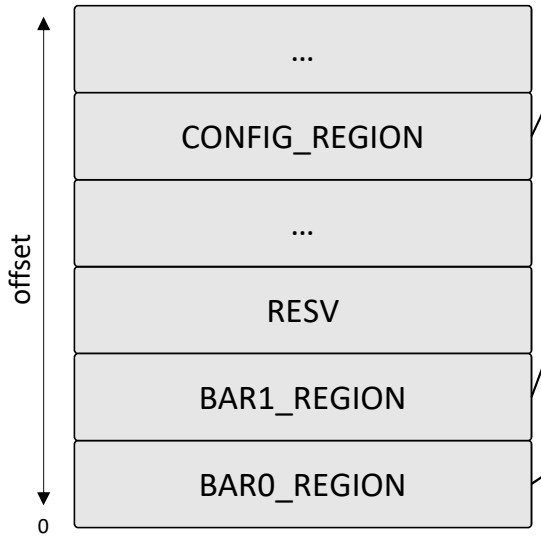

```
# Query the number of available mdev instances
$ cat /sys/class/mdev_bus/0000:84:00.3/.../available_instances

# Create a mdev instance
$ echo $UUID > /sys/class/mdev_bus/0000:84:00.3/.../create

# Launch QEMU with a virtio-net device
$ qemu-system-x86_64 -cpu host -enable-kvm \
    <snip>  \
    -netdev type=vhost-vfio,sysfsdev=/sys/bus/mdev/devices/$UUID,id=net0 \
    -device virtio-net-pci,netdev=net0,page-per-vq=on \
```

- Device Setup
  - Generate instance by mdev sysfs
  - Pass UUID to net client vhost-vfio
- I/O Trap & Emulation
  - CSR, PIO/MMIO BAR, Doorbell Relay
- Direct Mapped Doorbell
  - Page aligned per queue doorbell
- DMAR/INTR
  - Direct DMA ring/packet buffer
  - Interrupt notification

VHOST-MDEV DEVICE LAYOUT



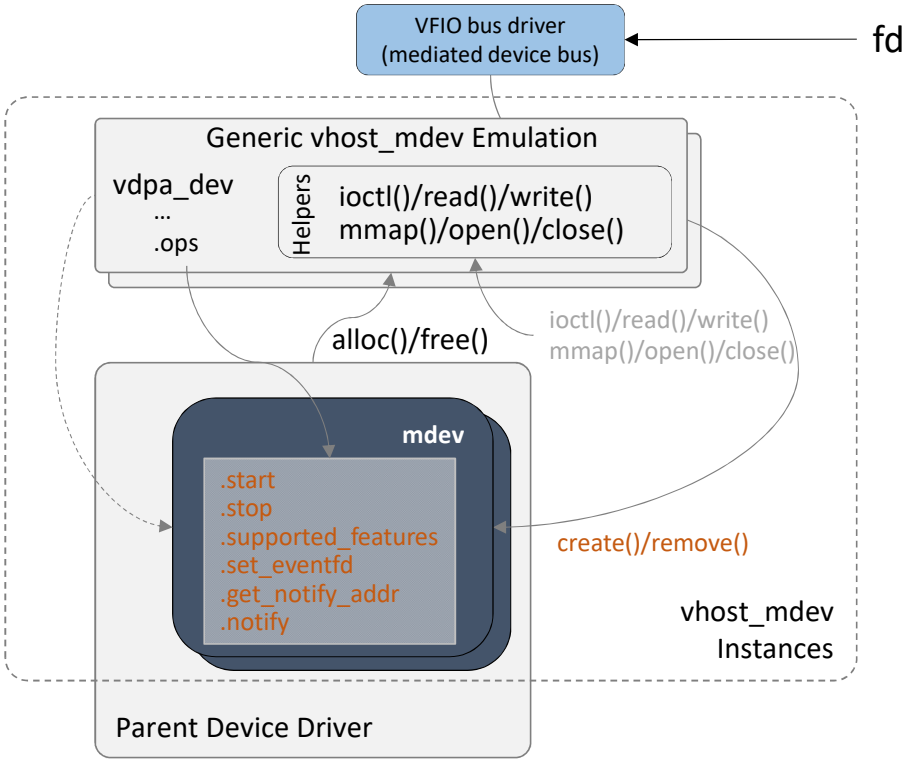
- Configure Space
  - ID to identify device type
  - Others are dummy
- BAR1 (mmap-able)
  - Mmap-able, direct notify
  - Write-able, notify relay
- BAR0
  - main control interface
  - vhost message by read/write
  - 'request' remains the same as vhost-kernel ioctl cmds

```
struct vhost_vfio_op {
        __u64 request;
        __u32 flags;
        /* Flag values: */
#define VHOST_VFIO_NEED_REPLY 0x1 /* Whether need reply */
        __u32 size;
        union {
                __u64 u64;
                struct vhost_vring_state state;
                struct vhost_vring_addr addr;
                struct vhost_memory memory;
        } payload;
};
```

Message Format

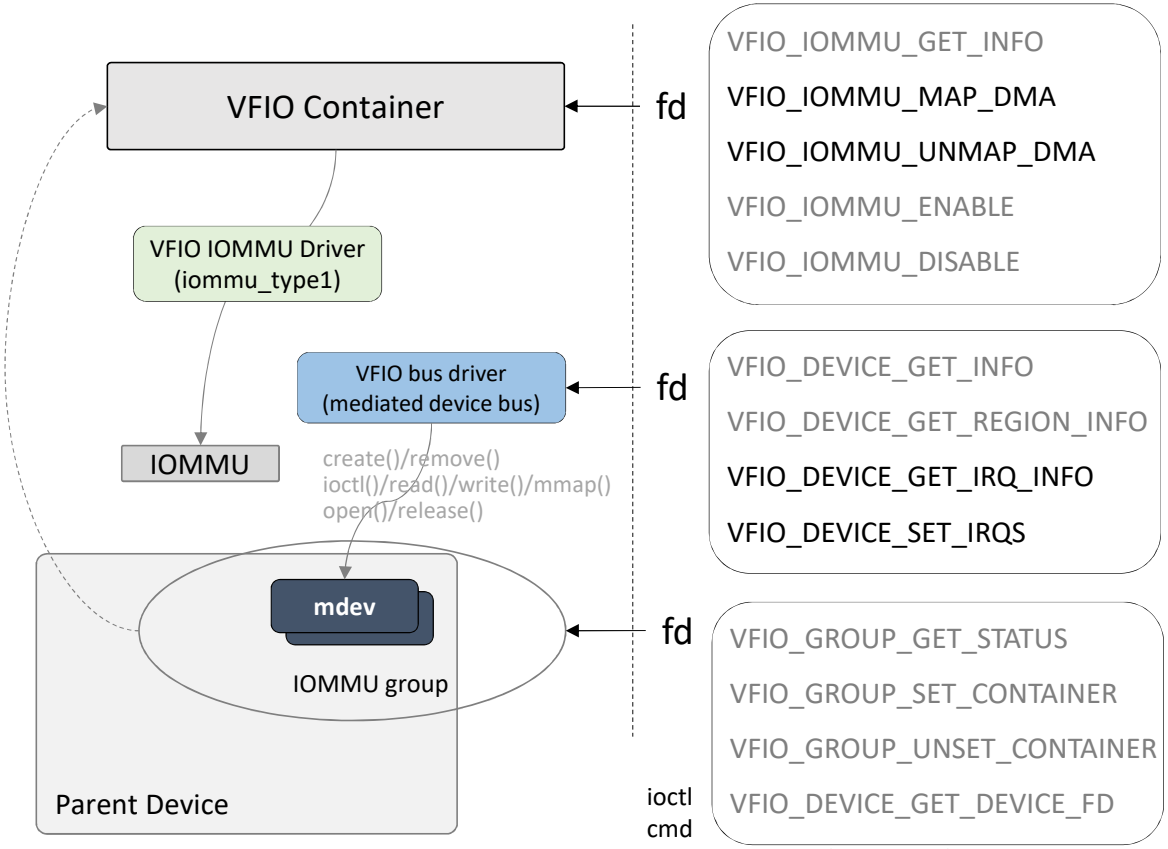
- Each message will be written to or read from BAR0_REGION at offset 0
- Set: write Req.
- Get: write Req. followed by read Resp.

GENERIC VHOST DEVICE EMULATION



- Needs to impl mdev parent ops$_{(.create/.remove)}$
- mdev parent calls alloc()/free() to populate mdev instances
- Other mdev ops$_{(.read/.write/.ioctl/.mmap/.open/.release)}$ have been provided by vhost helpers
- vdpa device ops$_{(.start/.../.notify)}$ is called by the generic vhost mdev emulation

DMA & Interrupt



DMA

- VFIO DMA ioctl API is used to remap memory
- vfio/mdev: IOMMU aware mediated device
  https://lkml.org/lkml/2018/10/12/225

Interrupt

- VFIO interrupt ioctl API is used to setup device interrupts
- VFIO IRQ eventfd associate with IRQFD
- IRQ Bypass Manager to connect Prod/Cons
- VFIO_PCI_MSIX_IRQ_INDEX in the 1st

Status & Plan

- Key Developers
  - TIWEI BIE (tiwei.bie AT intel.com)
  - XIAO WANG (xiao.w.wang AT intel.com)
- Kernel RFC Patch
  https://lwn.net/Articles/750770/
- QEMU RFC Patch
  https://patchwork.ozlabs.org/patch/984764/
  https://patchwork.ozlabs.org/patch/984765/
- Next Step: a Sample Parent Device Driver to use mdev-vhost
  - virtio-net device to support mdev-vhost
- Welcome to DPDK-Virtio Monthly Meeting

Legal Notices & Disclaimers

- Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at intel.com.
- Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit http://www.intel.com/performance.
- Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit http://www.intel.com/performance.
- Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.
- No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.
- Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.
-

Questions?

Thank You