Enhance KVM for Intel® Virtualization Technology for Connectivity

YaoZu (Eddie) Dong

Eddie.dong@intel.com

Intel Open Source Technology Center

Agenda

Intel® Virtualization Technology for Connectivity (VT-c)

Virtio-net overview

VMDq enhancement

SR-IOV

Summary

What is VT-c

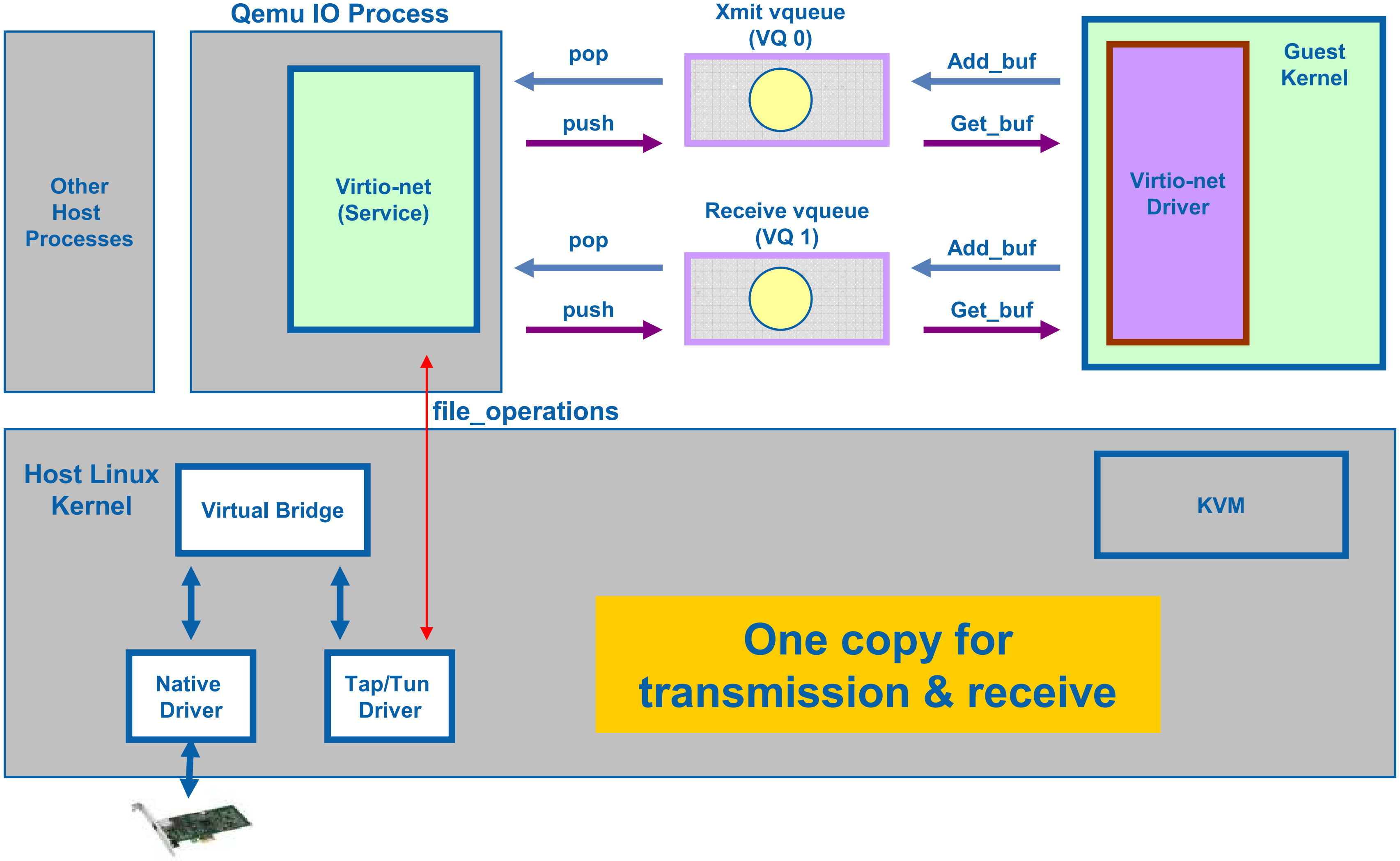
VMDq

- Multiple queue pairs for partitioning
- Filters a specific VM's unicast packets into individual receive queues
  - Such as MAC filtering, VLAN filtering
- Ensures transmit fairness between VMs
  - Prevents head-of-line blocking

SR-IOV

- PCI SIG IO virtualization technology, providing multiple virtual functions (VFs) to partition among VMs

Virtio-net Architecture



Qemu IO Process

Other Host Processes

Virtio-net (Service)

Xmit vqueue (VQ 0)

pop

push

Add_buf

Get_buf

Receive vqueue (VQ 1)

pop

push

Add_buf

Get_buf

Guest Kernel

Virtio-net Driver

file_operations

Host Linux Kernel

Virtual Bridge

Native Driver

Tap/Tun Driver

KVM

One copy for transmission & receive

vringfd (WIP by Rusty)

A separate char device used for vring based user/kernel communication

- File_operations: for user access
- Vring_ops: to manipulate the vring
  - Needs_poll: data ready
  - Pull (like pop in user level BE service)
  - Push (like push in user level BE service)

Tun device enhancement with vring

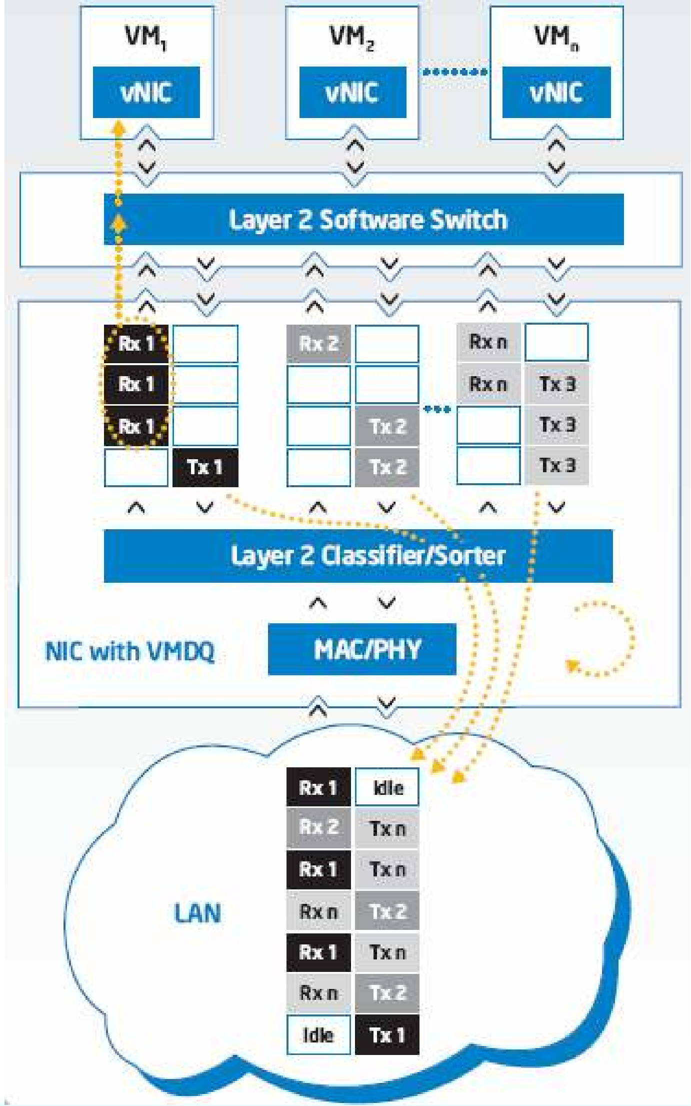
- Xmit can directly take user buffer (after pined) for xmit
- Transmission becomes zero copy now

What is VMDq

An Integral Part of Intel® Virtualization Technology for Connectivity, or VT-c

HW L2 classifier/sorter places packet to the destination VM's queue based on MAC address and VLAN tags

http://www.intel.com/technology/platform-technology/virtualization/VMDq_whitepaper.pdf



VMDq Enhancement

Using HW pre-sorting mechanism to avoid receive side copy

- Renato J Santos proposed a network enhancement in Linux to support VMDq in Xen
  - A network driver can take skbs from outside
  - A new API vmq_netif_rx is used to replace netif_rx to bypass bridge
- A kernel module, say VMDq agency, to receive pre-sorted packets with 0 copy

Reusing vringfd for kernel side vring operation and avoid transmission side copy

Packets to default queue still go to bridge

- Multiple queue guest network driver

Build

SR-IOV Specification

Start with a single function device

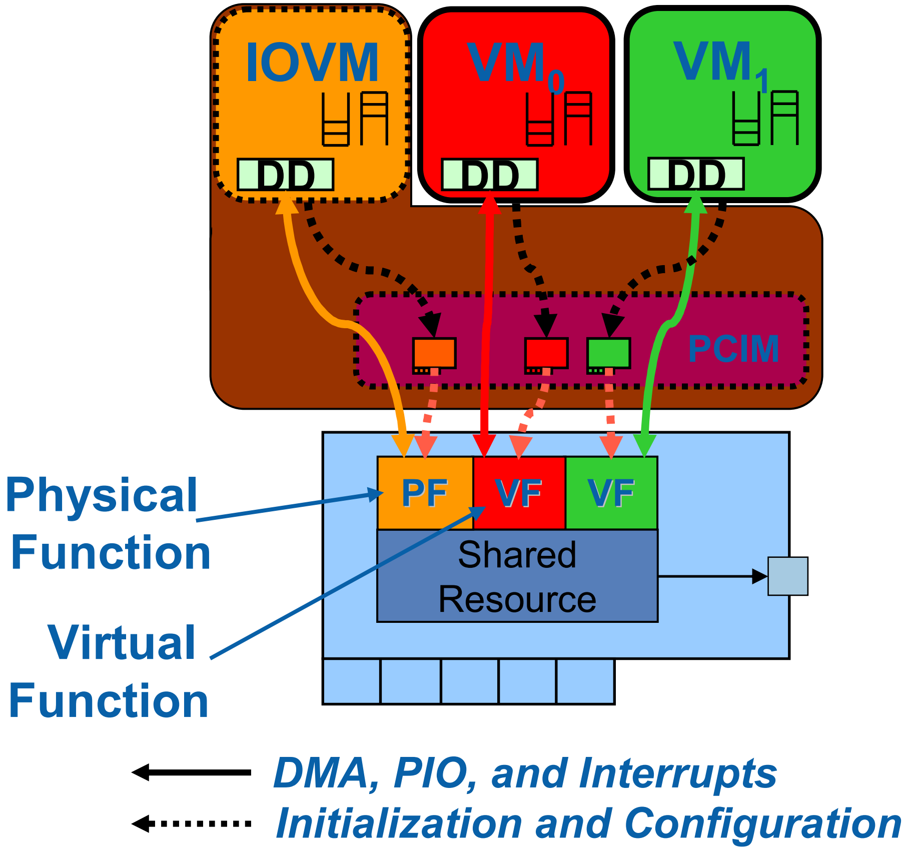
- HW under the control of privileged SW
- Includes an SR-IOV Extended Capability
- Physical Function (PF)

Replicate the resources needed by a VM

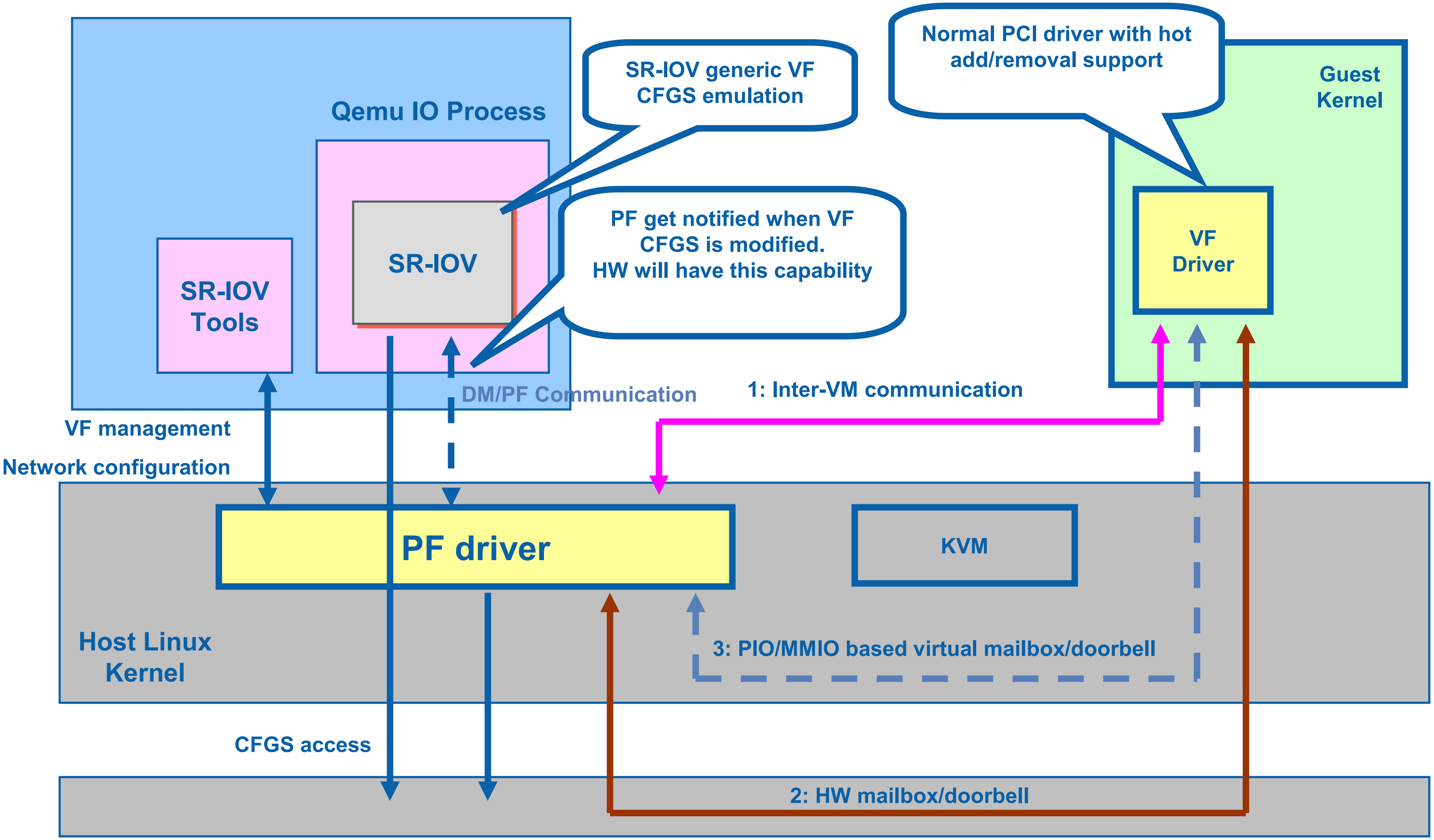
- MMIO for direct communication
- RID to tag DMA traffic
- Minimal configuration space
- Virtual Function (VF)

Introduces PCI Manager (PCIM)

- Conceptual SW entity
- Completes the configuration model
- Translates VF into a full function
- Configures SR-IOV resources



SR-IOV Virtio-net architecture


Qemu IO Process
SR-IOV
SR-IOV Tools
SR-IOV generic VF CFGS emulation
PF get notified when VF CFGS is modified. HW will have this capability
Normal PCI driver with hot add/removal support
Guest Kernel
VF Driver
DM/PF Communication
1: Inter-VM communication
VF management
Network configuration
PF driver
KVM
Host Linux Kernel
3: PIO/MMIO based virtual mailbox/doorbell
CFGS access
2: HW mailbox/doorbell

Summary

VT-c brings significant network performance boost with minimal CPU use

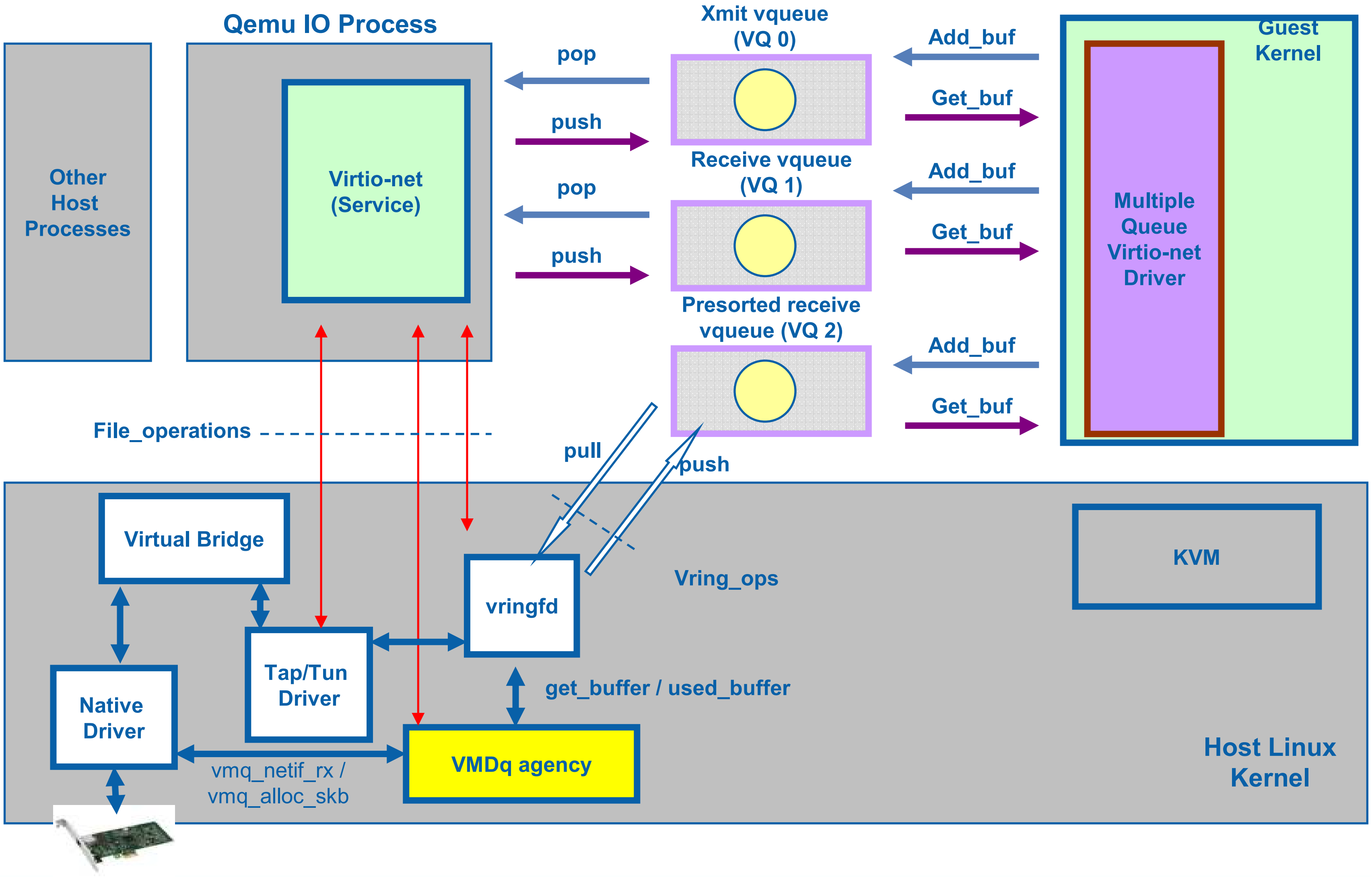
Many tasks ahead to push changes to upstream Linux

Your participation is very welcome!!!

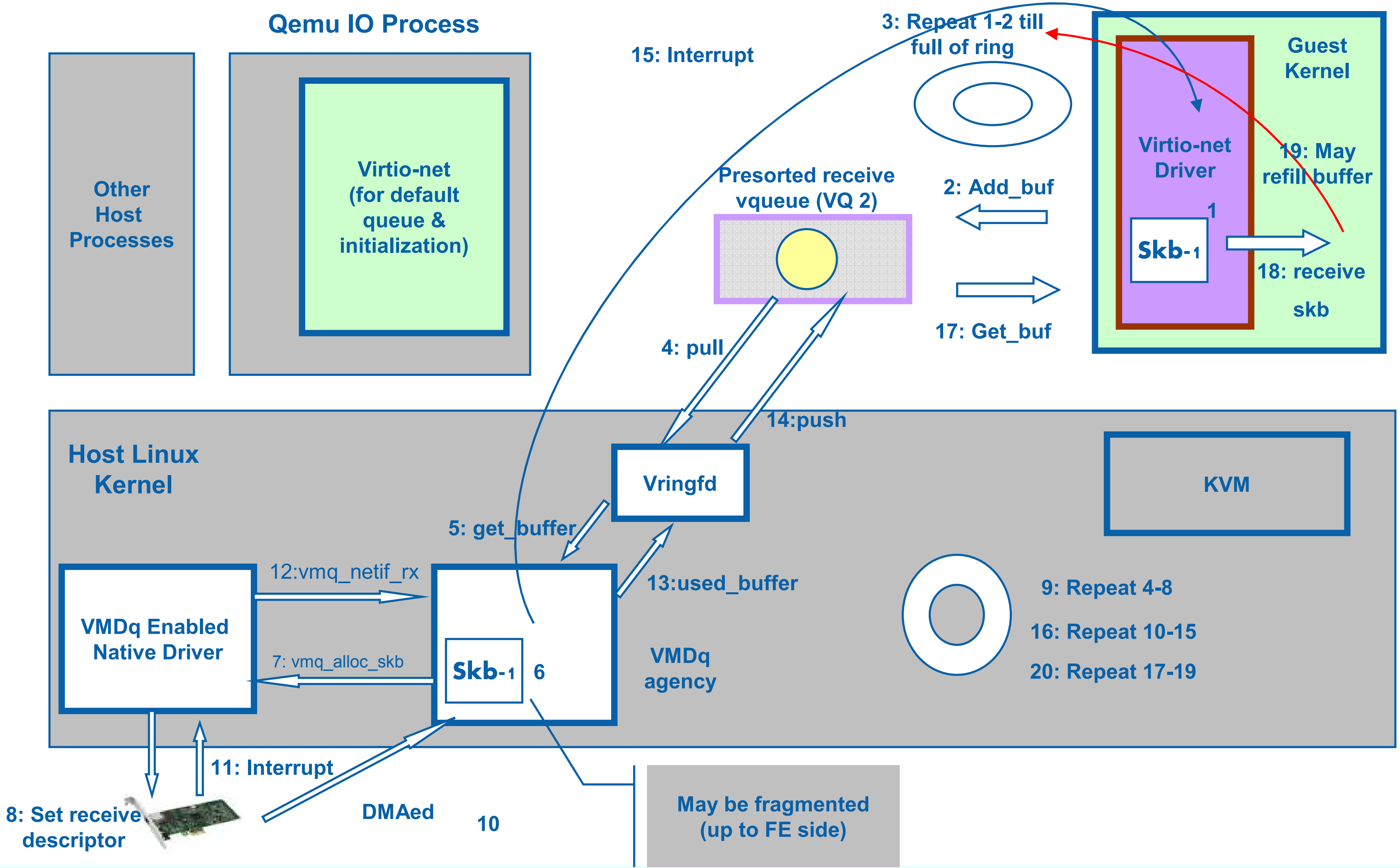
- Discuss details at BOFs?

Backup

Virtio-net with VMDq



Pre-sorted Packet Receiving



SR-IOV VF/PF Communication Channel

Inter-VM APIs → PV VF driver

- Depends on VMM, Guest OS, and even OSVs
  - There is no Windows Inter-VM APIs in upstream, no standard release yet.

Guest hardware → VMM independent VF driver

- Real Hardware mailbox/doorbell – No SR-PCIM involvement
  - Good Performance, but IHVs may not implement.
- Virtual mailbox/doorbell – Need SR-PCIM support
  - Virtual BAR (PIO or MMIO)
  - Need SR-IOV standard

VF/PF driver pair's decision to use whatever mechanism,
but suggest using guest hardware

PCI Device Instance of VF in Host?

Created VF instance

- Pros: Easy for assignment
- Cons: Confuse to other pci modules, invasive change, need community decision on how to change → Need long time.
- Could be a long term solution.

No create VF instance

- Need access path for Qemu to R/W VF CFGS
- Modifications are mostly in Qemu side

Legal Information

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. 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.

Intel may make changes to specifications, product descriptions, and plans at any time, without notice.

All dates provided are subject to change without notice.

Intel is a trademark of Intel Corporation in the U.S. and other countries.

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

intel®

Leap ahead™