SW Assisted VDPA for Live Migration

XIAO WANG, Intel
Agenda

- VDPA Intro
- Device Live Migration
  - Status Quo
  - VDPA LM workflow
- SW Assisted VDPA for LM
  - Design & Impl
  - HW vs SW
- A Unified Vhost Zero-copy
- Key Takeaways
VDPA: enable data path pass-thru within Para-V

- VDPA: A Framework for virtio HW Acc
  - Build pass-thru like data path within Para-V
  - Inherit all PV advantages
- Decouple Control Path and Data Path
  - Vhost datapath in kernel/dpdk/HW
  - Transparent to guest
  - More performant with the coming virtio1.1
Device Live Migration Status Quo

- Tricky LM with VF pass-thru
  - Hacked hypervisor and guest
- Bond VF with virtio, manually or automatically
  - More or less assumption/requirement to VM
- VDPA
  - Inherit Para-Virt LM-able nature
  - Hypervisor helps to record device state
  - Zero requirement to guest kernel/userspace
  - Cross backend LM
VDPA Live Migration Workflow

Stage 1
- Set LOG_BASE
- page bitmap
- qemu
  - Src VM
- vhost/vdpa
  - VF
    - Logging
    - Queue Idx

Stage 2
- Sync
- Vhost/vdpa
  - VF
    - Logging
    - Queue Idx

Stage 3
- Sync
- Dest VM
  - qemu
  - Src VM
  - vhost/vdpa
  - VF
    - Logging
    - Queue Idx
  - GET_VRING_BASE

- Restore device state
- RARP with GUEST_ANNOUNCE
SW-Assisted VDPA for Live Migration

- SW fallback from HW
- A relay thread stands in between
- Zero-copy
- No desc.addr translation
- Page logging passingly when relay
Design & Impl

- Event driven relay
  - Epoll on guest kick and device intr
  - Dirty page logging when updating used ring
  - Batched logging of used ring
  - CPU usage increases as PPS arises

- APIs ready
  - Enable/Disable VDPA direct IO
  - Update IOMMU table for device DMA scope
  - Available ring relay for desc check
  - Used ring relay for dirty page logging
HW vs SW

Reduce bus overhead:
- Coarse-grained logging
- Ideally logging in IOMMU (long term)

Better relay perf:
- Polling mode relay
- One dedicated core
Future: A Unified Zero-copy Framework

**SW fallback for vring compatible HW**

- SW fallback from direct IO at LM stage
- As a particular zero-copy

**Unified zero-copy for generic NIC**

- Mempool as a wrapper for en/dequeue
- Minimum code change to NIC pmd
- w/o desc.addr translation
Upstreaming Status

- 18’Q2 QEMU vhost user support for VDPA [Merged]
- DPDK 18.05 VDPA framework in vhost [Merged]
- DPDK 18.05 IFCVF VDPA driver [Merged]
- Kernel VDPA (https://lwn.net/Articles/750770/) [RFC]
- DPDK 19.02 SW assisted VDPA for live migration [v1]
Key Takeaways

- VDPA combines SW Flex & HW Perf
- SW-assisted VDPA could further simplify HW design
- A generic zero-copy framework for all NICs with VDPA
Thanks!