What’s New in Virtio 1.1

Jason Wang, Red Hat
Tiwei Bie, Intel
What’s Virtio

- Invented by Rusty Russell for easy mechanism to provide virtual devices to guests
  - Net, block, SCSI, GPU, ...

- Standard driver means compatibility across hypervisors and operating systems

“The purpose of VIRTIO is to ensure that virtual environments and guests have a straightforward, efficient, standard, and extensible mechanism for virtual devices, rather than boutique per-environment or per-OS mechanisms.”
Virtio architecture

Device types
- net
- SCSI
- ...

Core device model
- Virtqueues
- Features bits
- Config spaces
- PCI
- MMIO
- CCW

Transport
Why for virtio 1.1?

Virtqueues

Device types
- net
- SCSI
- ...
- Virtqueues
- Features bits
- Config spaces
- PCI
- MMIO
- CCW

Core device model

Transport

The need for speed:

40G+ Ethernet card, NFV, Hardware virtio implementation
Packed Virtqueue
### Split virtqueues (virtio 1.0)

#### Available Ring
- **flag**: N
- **avail_idx**: ...

#### Descriptor Ring
- **address**: 0x80004200
- **len**: 0x80
- **flag**: RW
- **next**: NIL

#### Used Ring
- **flag**: M
- **used_idx**: ...

---

<table>
<thead>
<tr>
<th>Available Ring</th>
<th>Descriptor Ring</th>
<th>Used Ring</th>
</tr>
</thead>
<tbody>
<tr>
<td>flag</td>
<td>avail_idx</td>
<td>address</td>
</tr>
<tr>
<td>N</td>
<td>...</td>
<td>0x80004200</td>
</tr>
</tbody>
</table>

N: Available
M: Used

---

2 0x40
Issues of split virtqueues

- **For software backends**
  - Bad cache utilization, several cache misses per request
    - metadata is scattered into several places
    - descriptor chain is not contiguous in memory
    - cache contention in many places

- **For hardware implementation**
  - several PCI transactions per descriptor
Packed virtqueue

Descriptor Ring

- offset
  - 0x00
  - 0x08

- address
- len
- id
- flags

- next
- write
- reserved
- indirect
- reserved
- avail
- reserved
- used

0    1    3    7    15

Compact, better cache utilization, hardware friendly
packed virtqueues VS split virtqueues

64 Byte Performance

http://dpdk.org/ml/archives/dev/2018-April/095470.html By Jens Freimann
In order completion
In order completion

- OOO
  - Descriptor may be completed out of order
    - Zerocopy (priority queue in SW/HW)
  - Packed ring: compatibility
  - Complex driver

- In order device
  - Batch notification
  - Simpler device and driver
  - Optimization is easy

If VIRTIO_F_IN_ORDER has been negotiated, a device MUST use buffers in the same order in which they have been available.
In order device performance

Packet Per Second (Mpps)

Out of order

In order

http://dpdk.org/ml/archives/dev/2018-June/103665.html By Marvin Liu
Hardware implementation of virtio
Hardware virtio SRIOV

• **Hardware SRIOV support**
  - Virtio PF to generate multiple virtio VFs
  - VFs can serve difference VM/Container

• **Accelerate NFV**
  - Virtio-net driver only
vDPA

- **Vhost DataPath Acceleration**
  - datapath offloaded to hardware
- **control path**
  - Device specific through vhost
- **Advantages**
  - Live migration support
  - No #VFs limitation
  - Not tied to PCIE/SRIOV
- **Transparent acceleration for cloud users**
More coming

- **Backup feature**
  - Allow virito-net as a standby device for VF
  - Transparent bonding/acceleration of virtio-net

- **Flag for hardware backend**
  - IO barriers / DMA

- **Notification Data**
  - Piggyback the avail index when writing doorbell
  - Prefetching descriptors for saving PCIE bandwidth
  - Less pressure on the descriptor ring

- .......
Switch to virtio1.1

• Compatibility is kept
  – All extensions are added as new features.
  – Compatibility is achieved by features negotiation.
  – New drivers & devices will be fully compatible with legacy drivers & devices.

• Faster if using new driver and device

• Enjoy!
Summary

- **Virtio 1.1 is for**
  - performance
  - hardware implementation

- **Join the party and improve virtio**
  - open an issue: [https://github.com/oasis-tcs/virtio-spec/issues](https://github.com/oasis-tcs/virtio-spec/issues)
  - clone the spec from: [https://github.com/oasis-tcs/virtio-spec.git](https://github.com/oasis-tcs/virtio-spec.git)
  - send a patch to virtio-dev@lists.oasis-open.org
  - Biweekly dpdk/virtio conf call, contact me if you want to attend!
  - propose your novel ideas!
Thanks!
APPENDIX: Standardization of Virtio

• **Standardized since 1.0**
  - Community to OASIS Committee
  - Formal process, formal document (Virtio Specification)
  - People:
    • chair: Michael S. Tsirskin
    • editors from different vendors
    • contributors from Intel, IBM, Huawei, Red Hat ...

• **Specification contains:**
  - Basic facilities of a Virtio device
    • Status, feature bits, configuration space, virtqueues
  - Virtio transport options
    • PCI, Channel IO, MMIO
  - Device Types