Topic: Accelerate virtio/vhost using DPDK in NFV/Cloud Environment
Company: Intel
Title: Software Engineer
Name: Xie, huawei; Tan, Jianfeng
Agenda

• virtio/vhost background
• virtio in NFV/Cloud (challenges, solutions)
  1. virtio PMD optimization
  2. vhost TSO
  3. vhost reconnect
  4. VM2VM fastpath
• virtio in container
Virtio is the defacto para-virtualization standard for communicating with Virtual Machines (VM) efficiently.

Vhost is the KVM backend for Virtio, supplying packets to a Virtio Frontend.

Packet Flow
A bridge/virtual switch, switches packets to the backend (vhost) and these are forwarded to the frontend (virtio) in the Guest.
virtio in NFV/Cloud

North2South Perf
- virtio PMD optimization
- vhost AVX, delayed copy
- vhost TSO

East2West Perf
- VM2VM

Stability
- vhost reconnect
DPDK virtio development journey

- **2013**: kni vhost
- **2014**: xenvirt
- **2015**: multiple queue
- **2016**: mergeable
- **2017**: virtio 1.0

**vhost lib**
- **2015**: live migration
- **2016**: virtio ring layout

**vhost user**
- **2015**: vhost PMD
- **2016**: vhost in container

**vhost cuse**
- **2016**: virtio reconnect
- **2017**: vhost delayed copy

**vhost PMD**
- **2013**: zero copy
virtio in NFV/Cloud

North2South Perf
- virtio PMD optimization
- vhost AVX, delayed copy
- vhost TSO

East2West Perf: VM2VM

Stability
- vhost reconnect
normal virtio process

AVAIL

head idx

Free Descriptor List

guest mbuf

head

host mbuf

Used

‘virtio’

‘vhost’
RX ring layout optimization
TX ring layout optimization

- **Avail Ring**: 0, 1, ..., 127, 128, 129, ..., 255
- **Desc Ring for virtio_net_hdr**: 128, 129, ..., 255, 128, 129, ..., 255
- **Desc Ring for tx dat**: 0, 1, ..., 127, 0, 1, ..., 127
ring layout opt. and vectorization

TXonly

normal tx  simple tx

2.30X

RXonly

rx  vec rx

2.64X
new ring layout ?
virtio in NFV/Cloud

North2South Perf
virtio PMD and optimization
vhost AVX, delayed copy
vhost TSO

East2West Perf VM2VM

Stability
vhost reconnect

Virtualization Infrastructure

VIRTIO PORT
VHOST PORT
PHYSICAL PORT
DPDK accelerated vNFs

PACKET PROCESSING PIPELINE
DPDK based Virtual Switch

KVM
Commodity X86 Server
vhost TSO in VM2VM

```
.ol_flags = PKT_TX_TCP_CKSUM | PKT_TX_TCP_SEG
.tso_segsz = .gso_size
.l4_len
```
vhost TSO performance
virtio in NFV/Cloud

North2South Perf
- virtio PMD and optimization
- vhost AVX, delayed copy
- vhost TSO

East2West Perf: VM2VM

Stability
- vhost reconnect
vhost reconnect

QEMU SERVER

- connect
- SET_MEM_TABLE
- SET_VRING_NBA
- SET_VRING_FD

VHOST CLIENT

- ‘disconnect’
- crash happens!
- restart vhost

- connect
- SET_MEM_TABLE
- SET_VRING_NBA
- SET_VRING_FD

vq->last_used_idx ?

vq->last_used_idx = vq->used->idx !
virtio in NFV/Cloud

North2South Perf
virtio PMD and optimization
vhost AVX, delayed copy
vhost TSO

East2West Perf: VM2VM

Stability
vhost reconnect

Virtualization Infrastructure

Teleco
VNF1
VNF2
VNF...

Cloud
Tenant
APP
IP STACK
Tenant
APP
IP STACK

VIRTIO PORT
VHOST PORT
PHYSICAL PORT
DPDK accelerated vNFs

PACKET PROCESSING PIPELINE
DPDK based Virtual Switch

KVM
Commodity X86 Server

DATA PLANE DEVELOPMENT KIT
VM2VM fastpath (WIP)

VM2VM fastpath (WIP)
VM2VM fastpath (WIP)
Future work

- new ISA
- vhost delayed copy
- vhost AVX
- vhost FPGA
virtio for container - Motivations

• Requirements for Container-based NFV
  – high throughput
  – low latency
virtio for container - Status quo

PF/VF

VM

Container

VIRTIO
virtio for container - Architecture

• A new IPC in essence
  – Kernel-bypass
  – Well defined msg format
  – Cache friendly
• Virtio in Container vs VM
  – Device emulation
  – Address translation
Virtio for container – Addr trans

VM

GPA: Guest Physical Address

<table>
<thead>
<tr>
<th>GPA</th>
<th>LEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPA 0</td>
<td>Len 0</td>
</tr>
<tr>
<td>GPA 1</td>
<td>Len 1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>GPA n</td>
<td>Len n</td>
</tr>
</tbody>
</table>

Container

FVA: Frontend Virtual Address

<table>
<thead>
<tr>
<th>FVA</th>
<th>LEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>FVA 0</td>
<td>Len 0</td>
</tr>
<tr>
<td>FVA 1</td>
<td>Len 1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>FVA n</td>
<td>Len n</td>
</tr>
</tbody>
</table>
Virtio/Container - Performance

**CPU:** Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz with HT disabled

**Disclaimer:** prototyping result, subject to change with different system configurations

<table>
<thead>
<tr>
<th>Test with various packet size</th>
<th>Kernel forwarding (Mpps)</th>
<th>DPDK Pcap Test (Mpps)</th>
<th>Container with virtio (Mpps)</th>
<th>VM with virtio (Mpps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>64 Bytes</td>
<td>0.933</td>
<td>0.699</td>
<td>9.8</td>
<td>9.5</td>
</tr>
<tr>
<td>256 Bytes</td>
<td>0.909</td>
<td>0.673</td>
<td>4.5</td>
<td>4.5</td>
</tr>
<tr>
<td>512 Bytes</td>
<td>0.821</td>
<td>0.655</td>
<td>2.3</td>
<td>2.3</td>
</tr>
<tr>
<td>1024 Bytes</td>
<td>0.819</td>
<td>0.623</td>
<td>1.2</td>
<td>1.2</td>
</tr>
</tbody>
</table>

CPU: Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz with HT disabled

Disclaimer: prototyping result, subject to change with different system configurations
**virtio/vhost background**

**Virtio** is the KVM standard for communicating with Virtual Machines (VM) efficiently.

**Vhost** is the KVM backend for Virtio, supplying packets to a Virtio Frontend.

**Packet Flow**
A virtual switch, switches packets to the backend (vhost) and these are forwarded to the frontend (virtio) in the Guest.

Existing kernel space components.
DPDK virtio development journey
virtio optimization: ring layout and vectorization

RX fixed AVAIL ring

<table>
<thead>
<tr>
<th>AVAIL Ring</th>
<th>flags</th>
<th>avail_idx</th>
<th>head_idx</th>
<th>head_idx</th>
<th>......</th>
<th>head_idx</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESC Ring</td>
<td>desc</td>
<td>desc</td>
<td>......</td>
<td>desc</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mbuf</td>
<td>head_room</td>
<td>data area</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DATA PLANE DEVELOPMENT KIT
virtio optimization: ring layout and vectorization

TX fixed AVAIL ring

<table>
<thead>
<tr>
<th>flags</th>
<th>avail_idx</th>
<th>head_idx</th>
<th>head_idx</th>
<th>.....</th>
<th>head_idx</th>
<th>head_idx</th>
<th>head_idx</th>
<th>.....</th>
<th>head_idx</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>127</td>
<td>128</td>
<td>129</td>
<td>255</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DESC Ring

<table>
<thead>
<tr>
<th>desc</th>
<th>desc</th>
<th>.....</th>
<th>desc</th>
<th>desc</th>
<th>desc</th>
<th>desc</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>129</td>
<td>.....</td>
<td>255</td>
<td>128</td>
<td>129</td>
<td>255</td>
</tr>
<tr>
<td>desc</td>
<td>desc</td>
<td>.....</td>
<td>desc</td>
<td>desc</td>
<td>desc</td>
<td>desc</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>.....</td>
<td>127</td>
<td>0</td>
<td>1</td>
<td>127</td>
</tr>
</tbody>
</table>

mbuf

<table>
<thead>
<tr>
<th>head_room</th>
<th>data area</th>
</tr>
</thead>
</table>

DATA PLANE DEVELOPMENT KIT
VM2VM fastpath (WIP)