探秘DPDK Virtio的不同路径，so easy!

原创 DPDK开源社区 2017-05-26

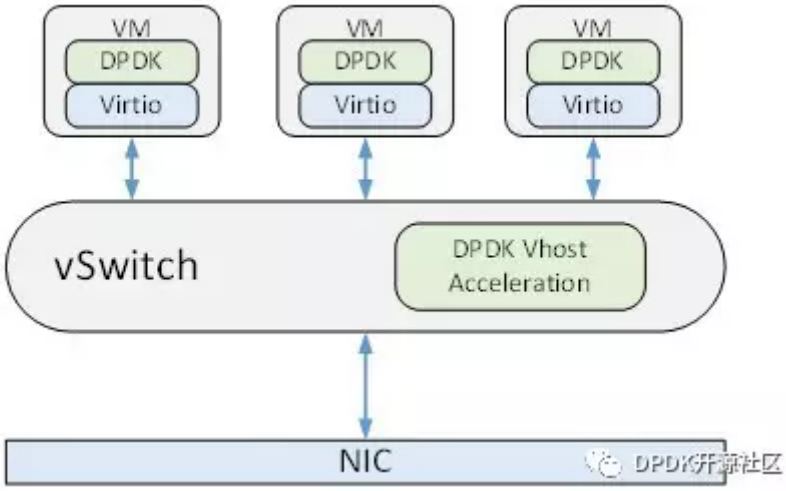
作者 姚磊

D P
D K

点击蓝字关注DPDK开源社区

什么是Vhost/Virtio

Vhost/Virtio是一种半虚拟化的设备抽象接口规范， 在Qemu和KVM中的得到了广泛的应用，在客户机操作系统中实现的前端驱动程序一般直接叫Virtio， 在宿主机实现的后端驱动程序称为Vhost。与宿主机纯软件模拟I/O相比，virtio可以获得很好的性能，在数据中心中得到了广泛的应用。Linux kernel中已经提供了相应的设备驱动，分别是virtio-net，以及vhost-net，DPDK项目为了提高数据吞吐性能，相对应的，提供了Virtio的用户态驱动virtio-pmd 和vhost的用户态实现Vhost-user,下图是一张典型的Virtio使用场景图。



Vhost/Virtio 在DPDK中收发路径配置

在DPDK的vhost/virtio 中，提供不同的Rx、Tx路径供用户使用，根据场景的不同，用户可以根据自己的功能以及性能需求，来挑选合适的路径。DPDK中，提供3条Tx、Rx路径。

Mergeable 路径

选择 Mergeable接收路径的优势在于，vhost可以将available ring中独立的mbuf组成链表来接收体量更大的数据包 。在实际应用中，这是被采用最多的路径，也是DPDK开发团队在过去几个月中，性能优化的重点方向。该路径采用的收发函数配置如下：

```
eth_dev->tx_pkt_burst = &virtio_xmit_pkts;
eth_dev->rx_pkt_burst = &virtio_recv_mergeable_pkts;
```

如想使用该路径, 需要在Vhost 和Qemu连接协商的过程中，通过VIRTIO_NET_F_MRG_RXBUF功能标志位来协商启动。Vhost-user默认支持该功能，Qemu中启用该功能的命令如下所示：（可滑动）

```
qemu-system-x86_64 -name vhost-vm1
.....
-device virtio-net-pci,mac=52:54:00:00:00:01,netdev=mynet1,mrg_rxbuf=on \
......
```

DPDK 会根据这个功能标志位，来选择相应的rx函数：（可滑动）

```
if (vtpci_with_feature(hw, VIRTIO_NET_F_MRG_RXBUF))
        eth_dev->rx_pkt_burst = &virtio_recv_mergeable_pkts;
else
        eth_dev->rx_pkt_burst = &virtio_recv_pkts;
```

不同于Vector和No-mergeable路径，rte_eth_txconf->txq_flags的值在Mergeable打开的情况下，并不会影响tx函数。

Vector

该路径利用处理器中的SIMD指令集，对数据的收发进行向量化处理，在纯IO数据包转发使用场景中，能够获得最高的性能。在DPDK中，该路径使用的收发函数如下：

```
eth_dev->tx_pkt_burst = virtio_xmit_pkts_simple;
eth_dev->rx_pkt_burst = virtio_recv_pkts_vec;
```

如想使用此收发路径，需要符合以下条件：

1) 平台处理器支持相应指令集, X86平台需要支持SSE3，DPDK中通过rte_cpu_get_flag_enabled(RTE_CPUFLAG_SSE3) 进行检查，ARM平台需要支持NEON，DPDK中通过rte_cpu_get_flag_enabled(RTE_CPUFLAG_NEON)检查。

2) RX方向的Mergeable需要关闭。DPDK会通过以下函数检查：

```
!vtpci_with_feature(hw, VIRTIO_NET_F_MRG_RXBUF)
```

Qemu中关闭该功能命令如下：（可滑动）

```
qemu-system-x86_64 -name vhost-vm1
.....
-device virtio-net-pci,mac=52:54:00:00:00:01,netdev=mynet1,mrg_rxbuf=off \
......
```

3) Offload 功能没有被启用。包括：VLAN offload, SCTP checksum offload, UDP checksum offload, TCP checksum offload。

4) rte_eth_txconf->txq_flags 需要设置为1。 例如，在DPDK提供的testpmd程序中，可以在虚拟机中通过类似如下命令进行配置Virtio设备：

```
#testpmd -c 0x3 -n 4 -- -i --txqflags=0xf01
```

从以上条件可以看出，Vector路径的功能相对有限，因而并没有成为DPDK 性能优化的重点方向。

No-mergeable 路径

No-mergeable路径在现实中较少使用，其收发路径如下：

```
eth_dev->tx_pkt_burst = &virtio_xmit_pkts;
eth_dev->rx_pkt_burst = &virtio_recv_pkts
```

如想使用该路径，需要符合如下配置：

1) RX方向Mergeable关闭

```
!vtpci_with_feature(hw, VIRTIO_NET_F_MRG_RXBUF)
```

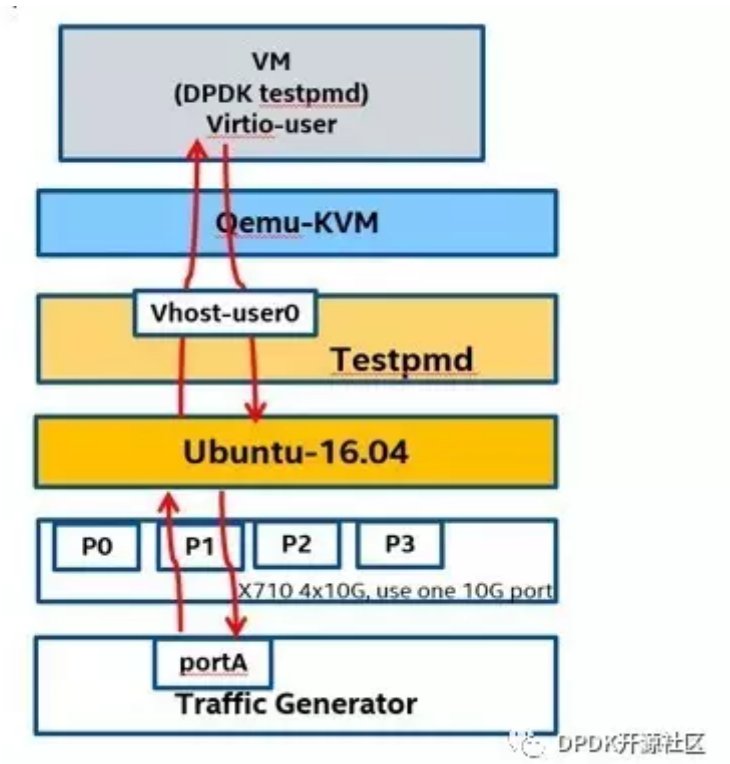
2) rte_eth_txconf->txq_flags需要设置为0，例如， 在DPDK提供的testpmd程序中，可以在虚拟机中通过类似如下命令进行配置Virtio 设备：

#testpmd -c 0x3 -n 4 -- -i --***txqflags=0xf00***

Vhost/Virtio 各路径PVP性能比较

在这部分，我们将比较一下DPDK 中vhost/virtio各收发路径 在PVP测试下的表现。PVP测试场景如下图所示，主要测试的是虚拟化环境中南北向的数据转发能力。Ixia发包器以10Gbps线速将64B数据包发送给网卡，物理机中的testpmd调用Vhost-User将数据转发进虚拟机中，虚拟机中的testpmd调用virtio-user将接收到数据转发回物理机，最终数据包回到IXIA， 数据路径为：（可滑动）

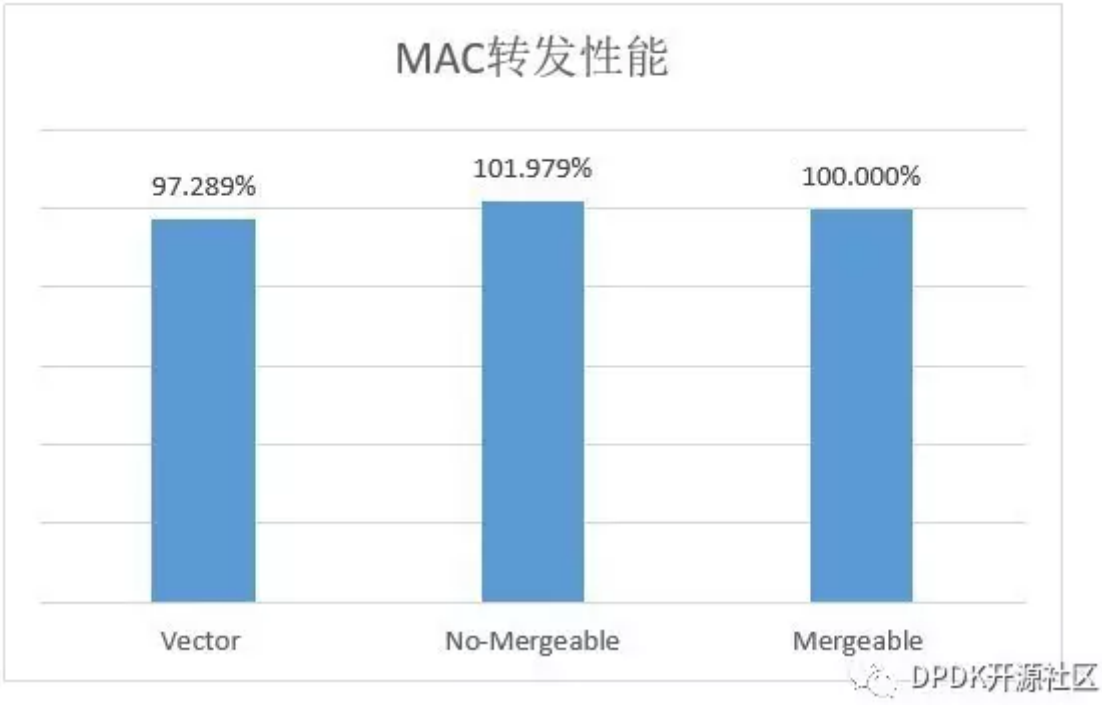
IXIA→NIC port1→Vhost-user0→Virtio-user0→NIC port1→IXIA



以DPDK 17.05 为例，在IO 转发配置下，不同路径的转发性能比较如下(以Mergeable为基准)：

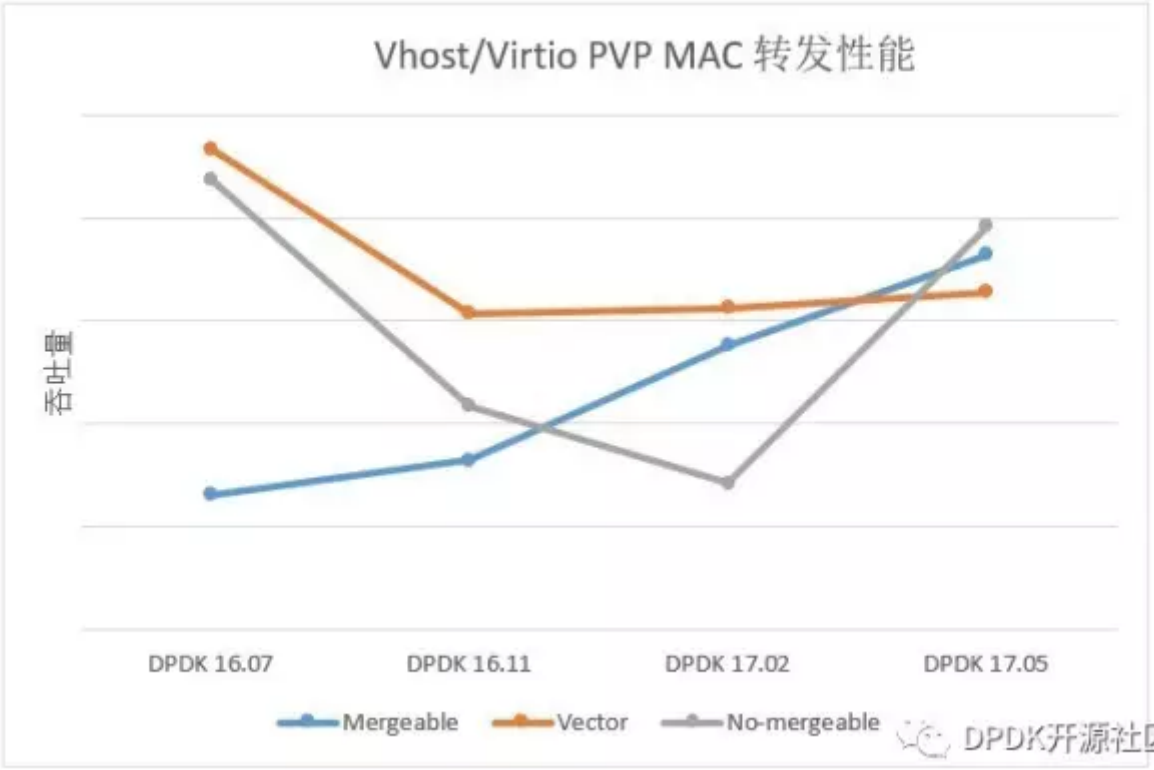
可以看到，在纯IO转发的情况下，Vector具有最好的吞吐量，比Mergeable高出了近15%。

在MAC转发配置下，不同路径的转发性能比较如下(以Mergeable为基准)：



在MAC转发的情况下，3个收发路径的性能基本相同，因为Mergeable路径可以提供更多的功能，我们推荐在此情况下，使用Mergeable 路径。

下图，是在X86平台上，DPDK16.07以来的各个版本PVP MAC转发的性能趋势，可以看到，因为Mergeable路径具有更广泛的应用场景，自16.07以来，DPDK的工程师针对Mergeable 路径，进行了很多的优化工作，此路径的PVP性能已经提升了将近20%。



Note：* 在DPDK16.11的性能下降，主要是由于添加新功能带来的性能开销，例如Vhost Xstats，Indirect descriptor table等

作者简介

姚磊，英特尔软件测试工程师， 主要负责DPDK虚拟化相关方向的测试工作。

文章精选

- 基于virtio-user的新exception path方案
- DPDK Release 17.02
- Hyperscan Release 4.4.0
- DPDK Release 16.11
- 无锁队列详细分解——Lock与Cache，到底有没有锁？
- 从计算机架构师的角度看DPDK性能

- 欢迎搭乘Hyperscan号极速列车~
- 无锁队列详细分解 — 顶层设计
- VMware Player 搭建DPDK实验平台
- Qemu/Kvm 搭建DPDK实验平台
- 技术贴：利用DPDK加速容器网络
- DPDK IP分片与重组设计实现