Virtio网络的演化之路

Xia, Chenbo DPDK与SPDK开源社区 11月13日

作为一个开放的标准接口，virtio一直在云计算与虚拟化中扮演着重要的角色。而virtio网络接口，作为virtio标准支持下最复杂的接口之一，在虚拟机/容器网络加速、混合云加速中一直扮演着重要角色。本文将在读者对virtio标准与虚拟化有一定了解的前提下，介绍virtio网络架构从创造之初到如今的演化之路。

1.virtio-net驱动与设备:
最原始的virtio网络

Virtio网络设备是一种虚拟的以太网卡，支持多队列的网络包收发。熟悉virtio的读者应该知道，在virtio的架构中有前后端之分。在virtio 网络中，所谓的前端即是虚拟机中的virtio-net网卡驱动。而后端的实现多种多样，后端的变化往往标志着virtio网络的演化。图一中的后端即是QEMU的实现版本，也是最原始的virtio-net后端（设备）。virtio标准将其对于队列的抽象称为Virtqueue。Vring即是对Virtqueue的具体实现。一个Virtqueue由一个Available Ring和Used Ring组成。前者用于前端向后端发送数据，而后者反之。而在virtio网络中的TX/RX Queue均由一个Virtqueue实现。所有的I/O通信架构都有数据平面与控制平面之分。而对于virtio来说，通过PCI传输协议实现的virtio控制平面正是为了确保Vring能够用于前

实现的virtio控制平面正是为了确保Vring能够用于前后端正常通信，并且配置好自定义的设备特性。而数据平面正是使用这些通过共享内存实现的Vring来实现虚拟机与主机之间的通信。举例来说，当virtio-net驱动发送网络数据包时，会将数据放置于Available Ring中之后，会触发一次通知（Notification）。这时QEMU会接管控制，将此网络包传递到TAP设备。接着QEMU将数据放于Used Ring中，并发出一次通知，这次通知会触发虚拟中断的注入。虚拟机收到这个中断后，就会到Used Ring中取得后端已经放置的数据。至此一次发送操作就完成了。接收网络数据包的行为也是类似，只不过这次virtio-net驱动是将空的buffer放置于队列之中，以便后端将收到的数据填充完成而已。

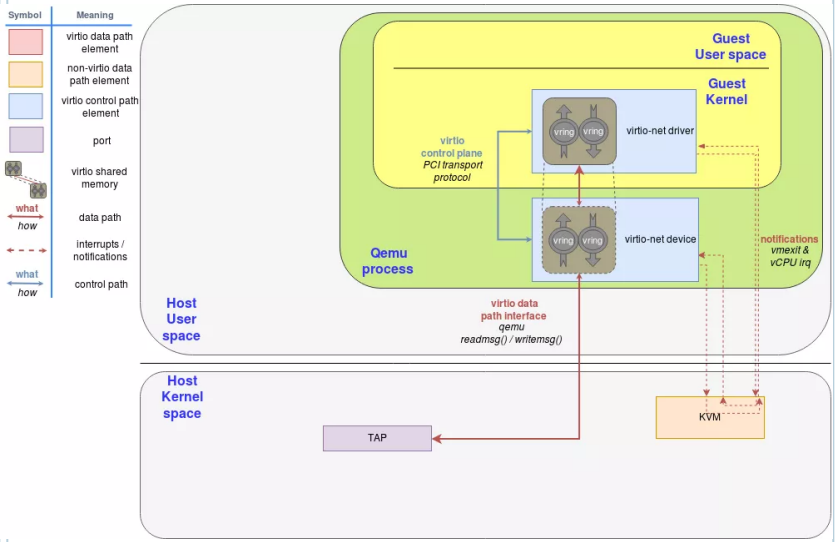


图 1 virtio驱动与设备

2.vhost-net：
处于内核态的后端

QEMU实现的virtio网络后端带来的网络性能并不如意，究其原因是因为频繁的上下文切换，低效的数据拷贝、线程间同步等。于是，内核实现了一个新的virtio网络后端驱动，名为vhost-net。

与之而来的是一套新的vhost协议。vhost协议可以将允许VMM将virtio的数据面offload到另一个组件上，而这个组件正是vhost-net。在这套实现中，QEMU和vhost-net内核驱动使用ioctl来交换vhost消息，并且用eventfd来实现前后端的通知。当vhost-net内核驱动加载后，它会暴露一个字符设备在/dev/vhost-net。而QEMU会打开并初始化这个字符设备，并调用ioctl来与vhost-net进行控制面通信，其内容包含virtio的特性协商，将虚拟机内存映射传递给vhost-net等。对比最原始的virtio网络实现，控制平面在原有的基础上转变为vhost协议定义的ioctl操作（对于前端而言仍是通过PCI传输层协议暴露的接口），基于共享内存实现的Vring转变为virtio-net与vhost-net共享，数据平面的另一方转变为vhost-net，并且前后端通知方式也转为基于eventfd的实现。

如图2所示，可以注意到，vhost-net仍然通过读写TAP设备来与外界进行数据包交换。而读到这里的读者不禁要问，那虚拟机是如何与本机上的其他虚拟机与外界的主机通信的呢？答案就是通过类似Open vSwitch（OVS)之类的软件交换机实现的。OVS相关

的介绍这里就不再赘述。

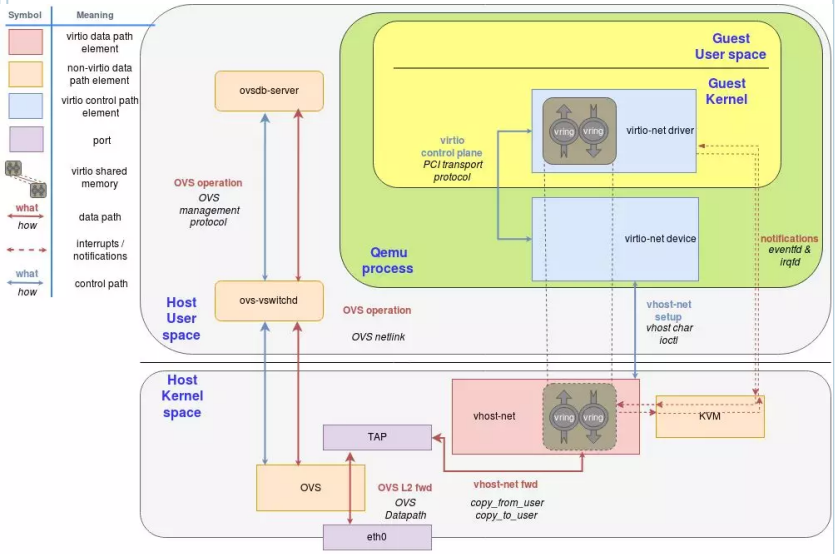


图 2 Vhost-net为后端的virtio网络架构

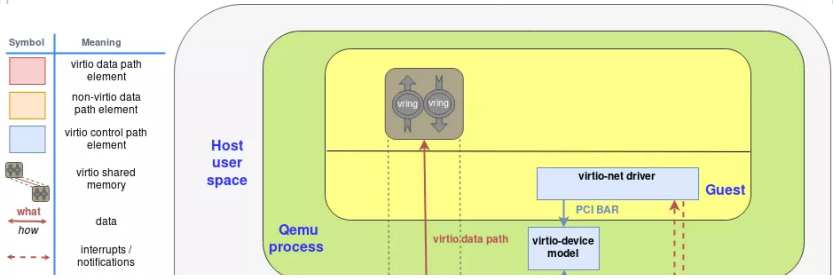
3.vhost-user:
使用DPDK加速的后端

DPDK社区一直致力于加速数据中心的网络数据平面，而virtio网络作为当今云环境下数据平面必不可少的一环，自然是DPDK优化的方向。而vhost-user就是结合DPDK的各方面优化技术得到的用户态virtio网络后端。这些优化技术包括：处理器亲和性，巨页的使用，轮询模式驱动等。除了vhost-user，DPDK还有自己的virtio PMD作为高性能的前端，本文将以vhost-user作为重点介绍。

基于vhost协议，DPDK设计了一套新的用户态协议，名为vhost-user协议，这套协议允许qemu将virtio设备的网络包处理offload到任何DPDK应用中（例如OVS-DPDK）。vhost-user协议和vhost协议最大的区别其实就是通信信道的区别。Vhost协议通过对vhost-net字符设备进行ioctl实现，而vhost-user协议则通过unix socket进行实现。通过这个unix socket，vhost-user协议允许QEMU通过以下重要的操作来配置数据平面的offload：

1. 特性协商：virtio的特性与vhost-user新定义的特性都可以通过类似的方式协商，而所谓协商的具体实现就是QEMU接收vhost-user的特性，与自己支持的特性取交集。
2. 内存区域配置：QEMU配置好内存映射区域，vhost-user使用mmap接口来映射它们。
3. Vring配置：QEMU将Virtqueue的个数与地址发送给vhost-user，以便vhost-user访问。
4. 通知配置：vhost-user仍然使用eventfd来实现前后端通知。

基于DPDK的Open vSwitch(OVS-DPDK)一直以来就对vhost-user提供了支持，读者可以通过在OVS-DPDK上创建vhost-user端口来使用这种高效的用户态后端。



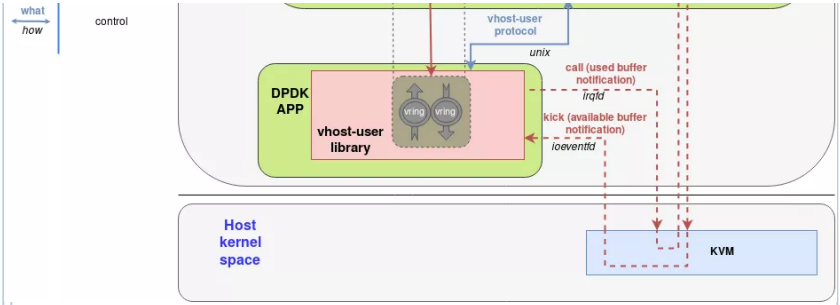


图 3 DPDK vhost-user架构

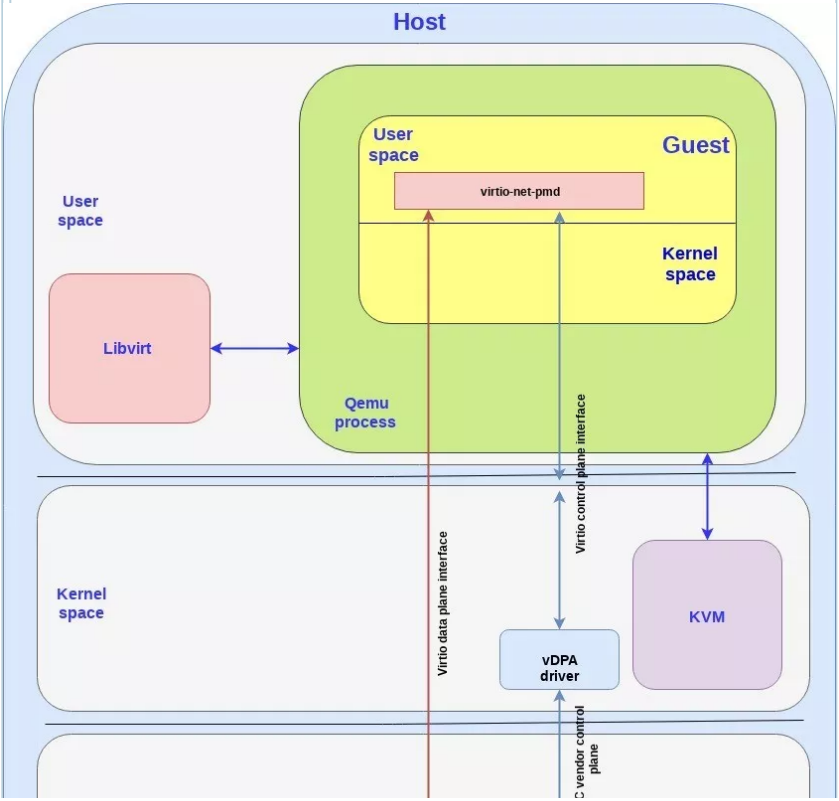
4.vDPA:
使用硬件加速数据面

Virtio作为一种半虚拟化的解决方案，其性能一直不如设备的pass-through，即将物理设备（通常是网卡的VF）直接分配给虚拟机，其优点在于数据平面是在虚拟机与硬件之间直通的，几乎不需要主机的干预。而virtio的发展，虽然带来了性能的提升，可终究无法达到pass-through的I/O性能，始终需要主机（主要是软件交换机）的干预。

vDPA(vhost Data Path Acceleration)即是让virtio数据平面不需主机干预的解决方案。从图中可以看到virtio的控制平面仍需要vDPA driver进行传递，也就是说QEMU，或者虚拟机仍然使用原先的控制平面协议作为接口，而这些控制信息被传递到硬件中，硬件会通过这些信息配置好数据平面。而数据平面上，经过配置后的数据平面可以在虚拟机和网卡之间直通。鉴于现在后端的数据处理其实完全在硬件中，原先的前后端通知方式也可以几乎完全规避主机的干预，以中断为

例，原先中断必须由主机处理，主机通过软件交换机得知中断的目的地之后，将虚拟中断注入到虚拟机中，而在vDPA中，网卡可以直接将中断发送到虚拟机中。总体来看，vDPA的数据平面与SR-IOV设备直通的数据平面非常接近，并且在性能数据上也能达到后者的水准。更重要的是vDPA框架保有virtio这套标准的接口，使云服务提供商在不改变virtio接口的前提下，得到更高的性能。

需要注意的是，vDPA框架中利用到的硬件必须至少支持virtio ring的标准，否则可想而知，硬件是无法与前端进行正确通信的。另外，原先软件交换机提供的交换功能，也转而在硬件中实现。



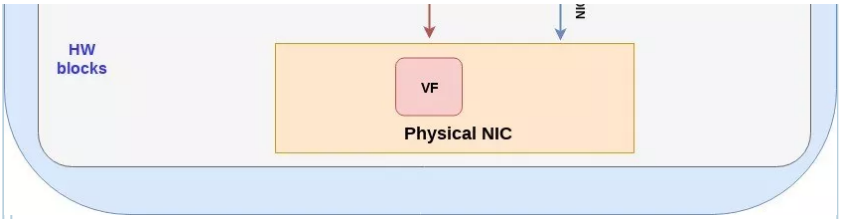


图 4 vDPA架构

5.总结

纵观virtio网络的发展，控制平面由最原始的virtio到vhost-net协议，再到vhost-user协议，逐步得到了完善与扩充。而数据平面上，从原先集成在QEMU中或内核模块的中，到集成了DPDK数据平面优化技术的vhost-user，最终到使用硬件加速数据平面。在保留virtio这种标准接口的前提下，达到了SR-IOV设备直通的网络性能。

转载须知

DPDK与SPDK开源社区公众号文章转载声明

推荐阅读

Virtio-PMD的路径选择与用法

本文所有图片均引用自Redhat官网


DPDK
SPDK
DPDK与SPDK开源社区
长按二维码关注 获取最新资讯

dpdkchina

扫一扫上面的二维码图案，加我微信

文章已于2019-11-13修改