分享你的Crypto资源：基于DPDK的Virtio-Crypto运算资源虚拟化方案

SDNLAB君 · 19-04-17

> **作者简介：**Zhang, Roy Fan Intel软件工程师，主要从事DPDK开发与虚拟化方面的工作。

本文转载自DPDK与SPSK开源社区

为什么要Virtio-Crypto?

随着近年来互联网，特别是移动互联网的高速发展，用户对数据安全的要求也越来越高。越来越多的网络信息流都被做了加密处理，来防止诸如泄密，仿冒，和重播等类型的网络攻击。Google目前已实现全站HTTPS加密，并在其安全性报告中指出截止2018年7月，超过70%的Chrome数据都已实现加密。

而密保工作如IPSec则需要大量的计算机资源来进行。一个最基本的AES块加密工作也需要数十个回合的查表，移位，线性变换等操作。当今的网络流吞吐量的急剧增加导致加密的工作密度呈几何数量级的增长，这个增长刺激了整个加密加速器技术的发展。Intel最新的采用 C620 系列芯片组 (Purley) 的至强处理器可使用其集成的Intel QuickAssit Technology (QAT)芯片实现超过130Gbps的IPSec数据面处理能力。

与此同时，云计算也有着飞速的增长。云用户对网络安全的需求也随之增加。为了让云用户能享受到加密加速技术，大部分市面上的加密加速器，包括QAT，都使用了Single-root Input/Output (SRIOV) Virtual Function (VF)技术，使得多台VM/容器能共享一个QAT资源，并几乎没有任何性能损失，如图1所示。

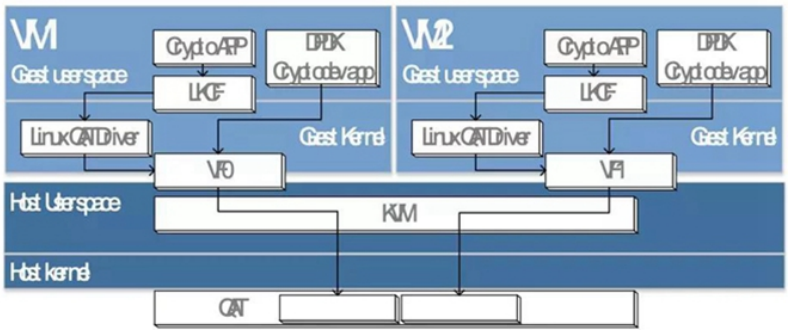


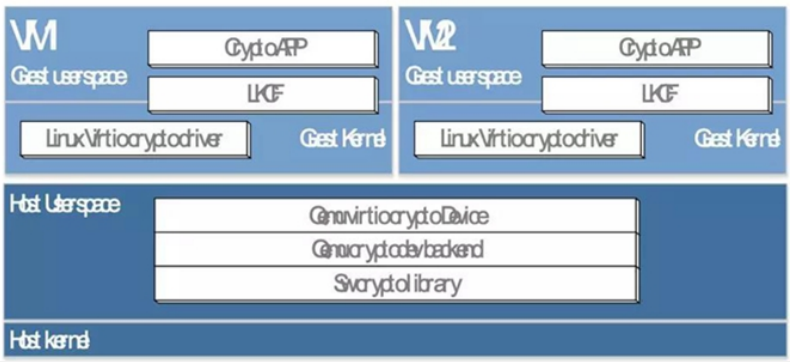
图1：QAT的SR-IOV在VM中的使用

1.首先，VM需要安装对应的VF驱动，与硬件有强耦合性。
2.同时，一个PF的VF的数量并非无限，其最大数量限制了能共享该PF的VM/容器 的数量。
3.再次，一个PF的最大吞吐量即为一个VF的最大吞吐量，用户无法获得更高的性能。
4.最后，VF目前并不具备限流功能，更没有QoS加持，而PF的最大吞吐量有限，其中某个VF的吞吐量加大则会制约其他VF，这个限制使得加密加速器的VF技术无法普及在公有云中。

所以，云用户所需要的加密加速器的虚拟化技术需要有和实机硬件无关，可拓展等特点。同时后端（物理机端）能实现聚合多个加速器资源和完成限流，QoS等工作。而这一切的答案就是Virtio-Crypto。

什么是Virtio-Crypto?

Virtio-Crypto是Virtio标准所支持的虚拟设备之一，由前端驱动和后端设备组成。前端一般为虚拟机和容器可访问的Virtio-Crypto设备驱动，而后端则是由物理服务器上的程序如Qemu所模拟的Vhost-Crypto设备。若虚拟机或者容器应用通过访问虚拟Virtio-Crypto设备做加密或验证等操作时，前端的Virtio-Crypto驱动会将其传入到后端并交由物理服务器的Crypto资源来处理。前端的Virtio-Crypto驱动已被整合到Linux内核中，而在后端也已有Qemu的软件Virtio-Crypto设备实现。



但是，这套方案也有有缺点

1.前端LKCF拖慢性能。因为前端的Virtio-Crypto驱动在内核的Linux Kernel Crypto Framework (LKCF)中，每次加密需要将数据进行2次拷贝，需要若干Callback调用，且不支持Burst操作。这些问题都严重拖慢了前端性能。

2.后端Qemu的Virtio-Crypto设备不支持硬件加速。

因此，目前的这套Virito-Crypto方案的效率太低，为了解决这个问题，我们提出了基于DPDK的Virtio-Crypto前后端的完整解决方案。

基于DPDK的Vhost-User-Crypto后端解决方案

2018年初Qemu被加入了Vhost-User的Crypto Proxy支持,使得除了Qemu之外其他的用户端Vhost-User应用来加速Crypto运算成为了可能。同年5月基于DPDK的Vhost User Crypto面世，成为了第一个支持Crypto设备的Vhost User加速方案。

Vhost Crypto作为DPDK Vhost 库的扩充，提供了几个新的API，并尽力隐藏了许多Vhost及Cryptodev实现的细节。我们将通过一个实例来看看他们怎么工作的。

首先，我们需要初始化Host的Cryptodev资源，包括设备，队列，和创建session mempool等。因为Virtio的特殊性，Host应该使用加密性能更强的Lookaside设备，如QAT。使用软件Cryptodev如AESNI-MB或者OPENSSL的话，因为VIRTIO数据传输的开销其总体性能将弱于VM上直接用软件加密引擎。

对于如何初始化Host的Cryptodev资源请参见本公众号以前的文章，或者是Cryptodev Programmer’ s

https://doc.dpdk.org/guides/prog_guide/cryptodev_lib.html)。

然后，和配置其他Vhost Application一样，我们需要创建一个vhost_device_ops实例，用来注册Socket连接在创建和销毁时的回调函数，分别为new_device和destroy_device。

在new_device()函数中，我们需要做一下几件事。

```Java
static int
new_device(int vid) {
...
/*    首先，我们需要调用rte_vhost_crypto_cr
化Vhost Crypto实例。其中vid为Virtio Device
化好的DPDKCryptodev设备号，crypto_sess_pool
用来创建和初始化处理VM加解密任务的session。
在Virtio Crypto标准中，session的信息交换为s
Cryptodev中session则为一段具体的数据。因此Vh
维护一个用来映射virtio crypto session_id和D
的关系的表，并自动处理VM发送的创建及删除sessi
干预，只需在创建实例时提供Cryptodev ID和sess
*/
rte_vhost_crypto_create(vid,
cryptodev_id,
crypto_sess_pool,
crypto_sess_priv_pool,
rte_lcore_to_socket_id(lcore_id);
...
/*    然后，我们需要告诉Vhost Crypto是否开启
让函数避免在处理virtio cryptodata request时
request的物理地址进行VM->Host的转换。开启zer
但该方法有一定的局限：要求HOST系统开启IOMMU，
持Host端使用AESNI-MB PMD，以及不支持该数据内
LKCF对每个设备初始化后会进行可行性检测，其中就
正确性检测。开启zero copy会使VM上的设备驱动无
失败，因此需要HOST端在等待LKCF检测完成之后再开
rte_vhost_crypto_set_zero_copy(vid,zero_c
...
}

Vhost Crypto用户应用的数据面处理也非常简单。
的代码片段。
static int
```

```
27      intcallfds[MAX_NB_VIRTIO_CRYPTO_DEVICE
28      structrte_crypto_op *ops[burst_size];
29  ...
30      /*  首先分配一个burst的DPDK Crypto Oper
31      rte_crypto_op_bulk_alloc(cop_pool,RTE_
32  ops,
33              burst_size);
34
        /*  实际处理VM Crypto Data Request的无限
35      for(;;) {
36          uint16_tfetched;

37  //      获得能完整enqueue到Cryptodev中operati
38  nb_ops =RTE_MIN(burst_size, NB_DESCRIPTORS
39
40  /*      从virtio queue中获取最大nb_ops个virti
41  DPDK Crypto Operation，并写入ops中，最后返回
42  fetched = rte_vhost_crypto_fetch_requests(
43  ops,nb_ops);

    /*      将转换好的ops enqueue到Cryptodev的que
           inflight+= rte_cryptodev_enqueue_bu
    queue_id,
              ops,fetched);
44  /*      一个小贴士：DPDK Cryptodev的queue并非
45  和，以上操作将不能成功enqueue到fetched个oper
46  operation必须妥善保存，以避免VM的crypto requ
47  下初始Cryptodev Queue时指定NB_DESCRIPTORS (
48  rte_vhost_crypto_fetch_requests () 时用其与
49  nb_ops大小，以避免fetched太多而不能enqueue的
50  以用来进行更复杂的SLA相关操作，如QoS等 */
51
    /*      将DPDK Cryptodev处理完的Crypto Operat
    fetched= rte_cryptodev_dequeue_burst(crypt
52          ops_dequeue, RTE_MIN(burst_size, in
53
54  /*      更新inflight */
55  inflight-= fetched;
56  /*      因为Cryptodev操作的异步性，dequeue的o
57  的，而且也不一定属于同一个Virtio Device。因此
58  buffer写回（zero copy未开启）,并在callfds中
    所有Virtio Device ID */
           rte_vhost_crypto_finalize_requests(
    fetched,callfds, &nb_callfds);

           /*  最后，通知每一个Virtio Device，加
           for(i = 0; i < nb_callfds; i++)
               eventfd_write(callfds[i],(event
    }
```

Vhost Crypto的API具有易用，前后端解耦，扩展性高等优点。使用ZERO COPY也能大幅提升性能。可以很容易地整合到复杂的虚拟Crypto资源服务的应用中。然而，由于Virtio Crypto的LINUX驱动及QEMU PROXY的局限性，目前仅支持AES-CBC和SHA1-MAC算法。随着未来支持算法的增加，Vhost-Crypto也会不断发展壮大。

基于DPDK的Virtio-User-Crypto前端轮询驱动（PMD）

为解决LKCF的性能拖慢问题，我们在DPDK的Cryptodev Framework中加入了基于Virtio User的Virtio-Crypto PMD。关于DPDK Cryptodev Framework的介绍已在曾经的公众号文章中详细介绍过，这里就不在赘述了。该PMD和其他的DPDK Crypto PMD共享相同的控制面和数据面API，区别是其工作在虚拟机或者容器中。

要使用它，我们首先需要将Vhost Crypto应用启动并使其创建UNIX socket file，并将其有Qemu传递给VM。QEMU命令范例如下。

```
                                                    Java
1 qemu-system-x86_64 \
2 [...] \
3     -chardevsocket,id=charcrypto0,path=/pat
4     -objectcryptodev-vhost-user,id=cryptode
5 \
6     -devicevirtio-crypto-pci,id=crypto0,cry
  [...]
```

在VM中，该设备将被Linux Kernel默认绑定为一个Virtio Crypto设备。我们需要将QEMU传递上来的virtio设备绑定为UIO-PCI-GENERIC驱动。假设该设备拥有0000:00:04.0的PCI地址，我们可通过如下命令来将其改绑定。

```
                                                    Java
```

```
pci/unbindecho "1af4 1054" >
/sys/bus/pci/drivers/uio_pci_generic/new_i
```

然后，我们就能将其像其他Cryptodev PMD一样在DPDK中使用，并能享受HOST端的加速了。

本文链接：https://www.sdnlab.com/23140.html

分享到：　　0　　0

相关文章

DPDK内存篇（二）：深入学习IOVA

DPDK内存篇（一）：基本概念

2019 Intel® 网络技术研讨会免费报名进行中，9月北京举行

SDNLAB君　发表于19-04-17　　0　　0

DPDK数据流过滤规则例程解析—— 网卡流处理功能窥探

DPDK对于Queue的配置

FD.io——助你创新更高效、更灵活的报文处理方案

0条评论

有话不说，憋着多难受啊!

请登录后才可以评论

评论

热度资讯

“互联网之父”确诊新冠肺炎；…

BAT引领中国公共云市场 AWS和微软主导亚太…

今天，华为云崩了

4月10日，据微博多位网友反映，华为云登录…

反目成仇！WeWork起诉软银; …

反目成仇！WeWork起诉软银，要求完成30亿…

直播预告：Smartnic带来的数…

分享嘉宾：唐杰 Xilinx系统架构师

直播预告：云网演进之Scale O…

一期一会第二场直播活动本周四上线，分享…

最新评论

☆宇宙浪子☆

七大要素解读: 1.硬件生命周期:用户硬件开支下降又是专用设备，品牌商和白盒制造商双杀。 2.托管服务合同的续订:DI…

SDNLAB君 发表于19-04-17

0 0

春风十里吹不动你

文件可以再发一下吗，失效了

衣依儿

@飘絮 谢谢

飘絮

链接已经更新哈。

衣依儿

文末的连接失效了，SDNLAB君能帮忙再发一份吗?

caozhongbo

有没有sd-wan的实战培训?

cctvghost

支持FlexE，dedicated queue么?

Soul-Mates

工具在哪里？现在能用吗

insurgent

作为一个网络工程师，面对SDN这个历史性节点，想越过ODL不太现实。但是想了解、学习ODL，确实又一时无法入门，难得其…

走路去塔尖

一直在作SDN方面的工作，最近工作需要开始接触ODL控制器，庞大的项目系统让我眼花缭乱，希望通过耿老师的课程慢慢学…

有奖投稿

职位发布

SDNLAB君 发表于19-04-17

0 0