


导航

博客园
首页
新随笔
联系
订阅 
管理

<	2020年4月						>
日	一	二	三	四	五	六	
29	30	31	1	2	3	4	
5	6	7	8	9	10	11	
12	13	14	15	16	17	18	
19	20	21	22	23	24	25	
26	27	28	29	30	1	2	
3	4	5	6	7	8	9	

公告

昵称： 钱小小
园龄： 5年5个月
粉丝： 0
关注： 1
[+加关注](#)

搜索

<input type="text"/>	<input type="button" value="找找看"/>
<input type="text"/>	<input type="button" value="谷歌搜索"/>

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

linux基础(3)
网络(3)
转载(3)
GRO(2)
linux内核(2)
nova(1)
open-falcon(1)
OpenStack(1)
iptables(1)
jsonrpc(1)
[更多](#)

【翻译】QEMU内部机制：vhost的架构

系列文章：

1. [【翻译】QEMU内部机制：宏观架构和线程模型](#)
2. [【翻译】QEMU内部机制：vhost的架构\(本文\)](#)
3. [【翻译】QEMU内部机制：顶层概览](#)
4. [【翻译】QEMU内部机制：内存](#)

原文地址：<http://blog.vmsplICE.net/2011/09/qemu-internals-vhost-architecture.html>

原文时间：2011年9月7日

作者介绍：Stefan Hajnocy来自红帽公司的虚拟化团队，负责开发和维护QEMU项目的block layer, network subsystem和tracing subsystem。

目前工作是multi-core device emulation in QEMU和host/guest file sharing using vsock，过去从事过disk image formats, storage migration和I/O performance optimization

QEMU内部机制：vhost的架构

该文章揭示了vhost机制如何在内核层面为KVM提供对virtIO设备的支持。

我本人最近在研究vhost-scsi，而且回答了许多关于ioeventfd, irqfd和vhost相关的问题，所以我认为这篇文章将对大家了解QEMU内部机制十分有用。

vhost介绍

在linux中，vhost驱动程序提供内核级别的virtIO设备模拟。在此之前，virtIO的后端驱动一般是由QEMU用户空间的进程来模拟的。

vhost在内核中实现了virtIO的后端驱动，将用户态的QEMU从virtIO的机制中剔除。

这使得设备模拟代码无需通过从用户态的系统调用就可以直接调用内核子系统的功能。

vhost-net驱动在内核态模拟了网卡相关的IO，它是最早以vhost形式实现的并且是被linux主线接纳的驱动。同

随笔分类

Apache全景分析(1)
QT(1)
安全
编程基础(27)
技术调研(19)
数据库(1)
虚拟化(13)

随笔档案

2020年2月(1)
2019年7月(14)
2019年6月(16)
2019年5月(1)
2019年4月(6)
2016年3月(2)
2015年12月(2)
2015年3月(12)
2014年12月(2)
2014年11月(9)

wxWidgets相关

wxWidgets简明教程
wxWidgets的linux安装配置
wxWidgets在线帮助

最新评论

1. Re: 【转】extern "C"的含义和用法
nm test.so可以查看符号表

--钱小小

2. Re:“云端融合”思想的自我摸索（很不靠谱）
目前，云计算技术的发展日新月异，它可以促进信息技术和数据资源充分合理利用，是信息化发展的必然趋势。下面是我对于云计算与操作系统的简单想法，其中内容不乏失实及误导之陈述，请领导批评指正。 经过将近十年的...

--钱小小

3. Re:debian包的补丁管理工具：quilt
来自ubuntu的
dh_quilt_patch命令帮助：
NAME dh_quilt_patch -
apply patches listed in
debian/patches/seriesSY
NOP...

--钱小小

时，vhost-blk和vhost-scsi项目也在开发中。

Linux内核v3.0的vhost代码在drivers/vhost/中。

被所有virtIO设备使用的通用代码位于drivers/vhost/vhost.c中，其中包含了所有设备都会用来与vm交互的vring处理函数。

vhost-net驱动的代码在drivers/vhost/net.c中。

vhost驱动模型

vhost-net驱动会在宿主机上创建/dev/vhost-net字符型设备，该设备是配置vhost-net实例的接口。

当QEMU进程以-netdev tap,vhost=on参数启动时，它会使用ioctl调用/dev/vhost-net，实现vhost-net实例的初始化。

该操作有三个作用：将QEMU进程与vhost-net实例关联起来、为协商virtIO的特性做准备、将vm的物理内存映射传递给vhost-net驱动。

在初始化过程中，vhost驱动会创建一个名为vhost-\$pid_of_qemu的内核线程，该线程称为"vhost工作线程"，它负责处理IO事件并执行设备模拟。

生产环境下qemu-kvm进程参数：

```
qemu 2726347 1 /usr/libexec/qemu-kvm -netdev  
tap,fd=40,id=hostnet0,vhost=on,vhostfd=42
```

对应的内核线程：

```
root 2726349 2 [vhost-2726347]
```

在内核态如何工作

vhost并不会模拟一个完整的virtIO PCI适配器，它仅用于处理virtqueue。

4. Re:在Linux下开发多语言软件(gettext解决方案)

附图bug修复报告:

--钱小小

阅读排行榜

1. golang类型断言的使用 (Type Assertion) (3770)
2. 进程占用过高cpu的排查(977)
3. 【转】QEMU Monitor机制实例分析(792)
4. 【转】理解qemu对设备的模拟机制(618)
5. 【翻译】QEMU内部机制:宏观架构和线程模型(402)

评论排行榜

1. 在Linux下开发多语言软件(gettext解决方案)(1)
2. debian包的补丁管理工具: quilt(1)
3. "云端融合"思想的自我摸索(很不靠谱) (1)
4. 【转】extern "C"的含义和用法(1)

在vhost机制中, 依旧会使用QEMU来进行比如说virtIO特性协商、动态迁移等操作。

也就是说vhost驱动不是一个独立的virtIO设备实现, 它借助用户态处理控制平面, 而在内核态, 它实现了数据平面的处理。

"vhost工作线程"等待virtqueue的kicks, 然后处理virtqueue中的buffers数据。

对vhost-net驱动来说, 就是从tx virtqueue中取出数据包并且把他们传输到tap设备的文件描述符中。

对tap设备的文件描述符的轮询也是由"vhost工作线程"负责的(这里指的是它使用epoll机制监听该fd), 该线程会在数据包到达tap设备时被唤醒, 并且将数据包放置在rx virtqueue中以便vm可以接收。

"vhost工作线程"的运行原理

vhost架构的一个神奇之处是它并不是只为KVM设计的。vhost为用户空间提供了接口, 而且完全不依赖于KVM内核模块。

这意味着其他用户空间代码在理论上也可以使用vhost设备, 比如libpcap(tcpdump使用的库)如果需要方便、高效的IO接口时。【什么场景?】

当vm将buffers放置到virtqueue中的事件发生时, vm会kicks宿主, 此时, 需要一种机制通知"vhost工作线程"有新的任务要做。

由于vhost与kvm模块是独立的, 他们之间不能直接进行交互。vhost实例在启动时会被分配一个eventfd文件描述符, "vhost工作线程"会监听eventfd。KVM拥有一个叫做ioeventfd的机制, 它可以将一个eventfd挂钩(hook)于某个特定vm的IO exit事件(当vm进行I/O操作时, 虚拟机通过vm exit将cpu控制权返回给VMM)。QEMU用户空间进程则为启动了virtqueue的VIRTIO_PCI_QUEUE_NOTIFY硬件寄存器访问事件注册了一个ioeventfd。

如此一来, "vhost工作线程"就能够在vm启动virtqueue时获得kvm模块的通知了。

回程时, "vhost工作线程"通过相似的方法向vm发起中断。vhost会接受一个"irqfd"文件描述符, 并向其中写入数据以便kick vm。

KVM有一个叫做irqfd的机制用于让一个eventfd向vm发起中断。QEMU用户空间进程则为virtIO PCI设备的中断注册一个irqfd, 并且通知给vhost实例。

如此一来, "vhost工作线程"就可以向vm发起中断了。

总之，vhost实例仅能感知到vm的内存映射、一个用于启动的eventfd和一个用于回调的irqfd。

参考代码：

drivers/vhost/vhost.c - vhost设备都会使用到的通用代码

drivers/vhost/net.c - vhost-net驱动

virt/kvm/eventfd.c - ioeventfd和irqfd

QEMU用户空间的用于初始化vhost实例的代码：

hw/vhost.c - vhost设备通用的初始化代码

hw/vhost_net.c - vhost-net驱动的初始化代码

===精选评论===

评论1：

途中显示在qemu中有一个DMA访问过程，

能否说明这个DMA-Transfer是在哪里初始化的么

我理解的是：只有物理网卡的driver能够在它的RX/TX buffers中执行DMA操作，

vhost本身是不支持的。

vhost与物理网卡的driver是通过sockets通信的么？

如此，RX/TX buffers的传输使用的是一个普通的memcpy吗？

回复1：

关于DMA和内存拷贝：

vhost-net支持zero-copy的传输方式。也就是说通过映射vm的RAM以便实现物理网卡直接从它进行DMA。

在接收路径中，仍需要一个从宿主内核态socket buffers到vm RAM(该RAM由QEMU用户空间进程映射至此)的内存拷贝过程。

可以参考drivers/vhost/net.c的handle_tx()和handle_rx()函数。

关于vhost与物理网卡的sockets通信：

vhost-net不会直接与物理网卡驱动通信，它只和tun设备(tap或macvtap)驱动打交道。

tap设备一般会放置于网桥中，以便将数据传输给物理网卡。

vhost-net使用内核态socket接口结构体，但是仅能与tun驱动实例协同工作。它会拒绝使用一个常规的socket文件描述符。

可以参考drivers/vhost/net.c:get_tap_socket()

请注意：tun驱动的socket是不会以socket文件描述符的形式暴露给用户空间的，如同用户空间的AF_PACKET sockets一样，它仅在内核中使用。

原文如下：

QEMU Internals: vhost architecture

This post explains how vhost provides in-kernel virtio devices for KVM. I have been hacking on vhost-scsi and have answered questions about ioeventfd, irqfd, and vhost recently, so I thought this would be a useful QEMU Internals post.

Vhost overview

The vhost drivers in Linux provide in-kernel virtio device emulation. Normally the QEMU userspace process emulates I/O accesses from the guest. Vhost puts virtio emulation code into the kernel, taking QEMU userspace out of the picture. This allows device emulation code to directly call into kernel subsystems instead of performing system calls from userspace.

The vhost-net driver emulates the virtio-net network card in the host kernel. Vhost-net is the oldest vhost device and the only one which is available in mainline Linux. Experimental vhost-blk and vhost-scsi devices have also been developed.

In Linux 3.0 the vhost code lives in `drivers/vhost/`. Common code that is used by all devices is in `drivers/vhost/vhost.c`. This includes the virtio vring access functions which all virtio devices need in order to communicate with the guest. The vhost-net code lives in `drivers/vhost/net.c`.

The vhost driver model

The vhost-net driver creates a `/dev/vhost-net` character device on the host. This character device serves as the interface for configuring the vhost-net instance.

When QEMU is launched with `-netdev tap,vhost=on` it opens `/dev/vhost-net` and initializes the vhost-net instance with several `ioctl(2)` calls. These are necessary to associate the QEMU process with the vhost-net instance, prepare for virtio feature negotiation, and pass the guest physical memory mapping to the vhost-net driver.

During initialization the vhost driver creates a kernel thread called vhost-\$pid, where \$pid is the QEMU process pid. This thread is called the "vhost worker thread". The job of the worker thread is to handle I/O events and perform the device emulation.

In-kernel virtio emulation

Vhost does not emulate a complete virtio PCI adapter. Instead it restricts itself to virtqueue operations only. QEMU is still used to perform virtio feature negotiation and live migration, for example. This means a vhost driver is not a self-contained virtio device implementation, it depends on userspace to handle the control plane while the data plane is done in-kernel.

The vhost worker thread waits for virtqueue kicks and then handles buffers that have been placed on the virtqueue. In vhost-net this means taking packets from the tx virtqueue and transmitting them over the tap file descriptor.

File descriptor polling is also done by the vhost worker thread. In vhost-net the worker thread wakes up when packets come in over the tap file descriptor and it places them into the rx virtqueue so the guest can receive them.

Vhost as a userspace interface

One surprising aspect of the vhost architecture is that it is not tied to KVM in any way. Vhost is a userspace interface and has no dependency on the KVM kernel module. This means other userspace code, like libpcap, could in theory use vhost devices if they find them convenient high-performance I/O interfaces.

When a guest kicks the host because it has placed buffers onto a virtqueue, there needs to be a way to signal the vhost worker thread that there is work to do. Since vhost does not depend on the KVM kernel module they cannot communicate directly. Instead vhost instances are set up with an eventfd file descriptor which the vhost worker thread watches for activity. The KVM kernel module has a feature known as ioeventfd for taking an eventfd and hooking it up to a particular guest I/O exit. QEMU userspace registers an ioeventfd for the VIRTIO_PCI_QUEUE_NOTIFY hardware register access which kicks the virtqueue. This is how the vhost worker thread gets notified by the KVM kernel module when the guest kicks the virtqueue.

On the return trip from the vhost worker thread to interrupting the guest a similar approach is used. Vhost takes a "call" file descriptor which it will write to in order to kick the guest. The KVM kernel module has a feature called irqfd which allows an eventfd to trigger guest interrupts. QEMU userspace registers an irqfd for the virtio PCI device interrupt and hands it to the vhost instance. This is how the vhost worker thread can interrupt the guest.

In the end the vhost instance only knows about the guest memory mapping, a kick eventfd, and a call eventfd.

Where to find out more

Here are the main points to begin exploring the code:

- `drivers/vhost/vhost.c` - common vhost driver code
- `drivers/vhost/net.c` - vhost-net driver
- `virt/kvm/eventfd.c` - ioeventfd and irqfd

The QEMU userspace code shows how to initialize the vhost instance:

- hw/vhost.c - common vhost initialization code
- hw/vhost_net.c - vhost-net initialization

分类: [虚拟化](#)

好文要顶

关注我

收藏该文



钱小小

关注 - 1

粉丝 - 0

[+加关注](#)

0

0

« 上一篇: [kvm热迁移](#)

» 下一篇: [【真的很先进】阿里云在2018-KVM Forum上分享的动态迁移实践](#)

posted on 2019-06-18 18:20 钱小小 阅读(252) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#) [网站首页](#)。

【推荐】超50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库

【推荐】腾讯云产品限时秒杀，爆款1核2G云服务器99元/年！

相关博文:

- [QEMU,KVM及QEMU-KVM介绍](#)
- [KVM 介绍 \(3\) : I/O 全虚拟化和准虚拟化 \[KVM I/O QEMU...](#)
- [linux下TUN/TAP虚拟网卡的使用](#)

- [理解 QEMU/KVM 和 Ceph \(1\) : QEMU-KVM 和 Ceph RB...](#)
- [浅谈ASP.NET的内部机制\(一\)](#)
- » [更多推荐...](#)

最新 IT 新闻:

- [特斯拉经营范围新增电信业务等 并正式迁入上海自贸区](#)
- [任正非最新讲话：艰苦奋斗的目的是过幸福生活](#)
- [聚美优品宣布完成私有化，正式从纽交所退市](#)
- [智能高空作业机器人公司史河科技完成3500万元Pre A+轮融资](#)
- [亚马逊下调广告营销联盟佣金费率 或重创出版商营收](#)
- » [更多新闻...](#)

历史上的今天:

2019-06-18 [kvm热迁移](#)

2019-06-18 [【转】KVM热迁移大体流程和内存降速问题](#)

Powered by:

[博客园](#)

Copyright © 2020 钱小小

Powered by .NET Core on Kubernetes