**一、open vswitch简介**

[](http://img1.sdnlab.com/wp-content/uploads/2015/12/OVS%E5%88%9D%E7%BA%A7%E6%95%99%E7%A8%8B%EF%BC%9A%E4%BD%BF%E7%94%A8open%20vswitch%E6%9E%84%E5%BB%BA%E8%99%9A%E6%8B%9F%E7%BD%91%E7%BB%9C%20%E5%9B%BE1.jpg)  
Open vSwitch是一个高质量的、多层虚拟交换机，使用开源Apache2.0许可协议，由Nicira Networks开发，主要实现代码为可移植的C代码。它的目的是让大规模网络自动化可以通过编程扩展,同时仍然支持标准的管理接口和协议（例如NetFlow, sFlow, SPAN, RSPAN, CLI, LACP, 802.1ag）。此外,它被设计位支持跨越多个物理服务器的分布式环境，类似于VMware的vNetwork分布式vswitch或Cisco Nexus 1000 V。Open vSwitch支持多种linux 虚拟化技术，包括Xen/XenServer， KVM和irtualBox。当前最新代码包主要包括以下模块和特性：

* ovs-vswitchd 主要模块，实现switch的daemon，包括一个支持流交换的Linux内核模块；
* ovsdb-server 轻量级数据库服务器，提供ovs-vswitchd获取配置信息；
* ovs-brcompatd 让ovs-vswitch替换Linuxbridge，包括获取bridge ioctls的Linux内核模块；
* ovs-dpctl 用来配置switch内核模块；

一些Scripts and specs 辅助OVS安装在Citrix XenServer上，作为默认switch；

* ovs-vsctl 查询和更新ovs-vswitchd的配置；
* ovs-appctl 发送命令消息，运行相关daemon；
* ovsdbmonitor GUI工具，可以远程获取OVS数据库和OpenFlow的流表。

此外，OVS也提供了支持OpenFlow的特性实现，包括

* ovs-openflowd：一个简单的OpenFlow交换机；
* ovs-controller：一个简单的OpenFlow控制器；
* ovs-ofctl 查询和控制OpenFlow交换机和控制器；
* ovs-pki ：OpenFlow交换机创建和管理公钥框架；
* ovs-tcpundump：tcpdump的补丁，解析OpenFlow的消息；

内核模块实现了多个“数据路径”（类似于网桥），每个都可以有多个“vports”（类似于桥内的端口）。每个数据路径也通过关联一下流表（flow table）来设置操作，而这些流表中的流都是用户空间在报文头和元数据的基础上映射的关键信息，一般的操作都是将数据包转发到另一个vport。当一个数据包到达一个vport，内核模块所做的处理是提取其流的关键信息并在流表中查找这些关键信息。当有一个匹配的流时它执行对应的操作。如果没有匹配，它会将数据包送到用户空间的处理队列中（作为处理的一部分，用户空间可能会设置一个流用于以后碰到相同类型的数据包可以在内核中执行操作）。

**二、open vswitch常用操作**

以下操作都需要root权限运行，在所有命令中br0表示网桥名称，eth0为网卡名称。

添加网桥：

#ovs-vsctl add-br br0

列出open vswitch中的所有网桥：

#ovs-vsctl list-br

判断网桥是否存在

#ovs-vsctl br-exists br0

将物理网卡挂接到网桥：

#ovs-vsctl add-port br0 eth0

列出网桥中的所有端口：

#ovs-vsctl list-ports br0

列出所有挂接到网卡的网桥：

#ovs-vsctl port-to-br eth0

查看open vswitch的网络状态：

#ovs-vsctl show

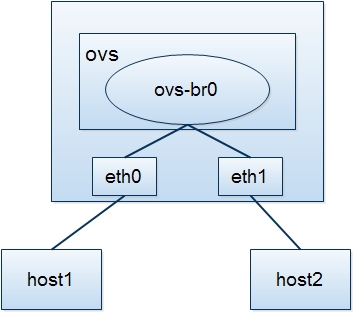
删除网桥上已经挂接的网口：

#vs-vsctl del-port br0 eth0

删除网桥：

#ovs-vsctl del-br br0

**三、使用open vswitch构建虚拟网络**

1、构建物理机和物理机相互连接的网络  
在安装open vswitch的主机上有两块网卡，分别为eth0、eth1，把这两块网卡挂接到open vswitch的网桥上，然后有两台物理机host1、host2分别连接到eth0和eth1上，实现这两台物理机的通信。构建结果图如下：  
[](http://img1.sdnlab.com/wp-content/uploads/2015/12/OVS%E5%88%9D%E7%BA%A7%E6%95%99%E7%A8%8B%EF%BC%9A%E4%BD%BF%E7%94%A8open%20vswitch%E6%9E%84%E5%BB%BA%E8%99%9A%E6%8B%9F%E7%BD%91%E7%BB%9C%20%E5%9B%BE2.jpeg)

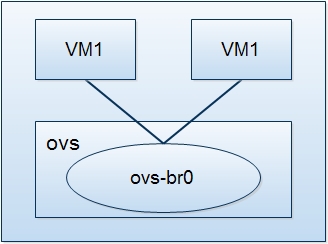
执行以下命令：

#ovs-vsctl add-br br0 //建立一个名为br0的open vswitch网桥

#ovs-vsctl add-port br0 eth0 //把eth0挂接到br0中

#ovs-vsctl add-port br0 eth1 //把eth1挂接到br0中

**2、构建虚拟机与虚拟机相连的网络**

在安装open vswitch的主机上安装两个虚拟机，把两个虚拟机的网卡都挂接在open vswitch的网桥上，实现两台虚拟机的通信，构建结果图如下：  
[](http://img1.sdnlab.com/wp-content/uploads/2015/12/OVS%E5%88%9D%E7%BA%A7%E6%95%99%E7%A8%8B%EF%BC%9A%E4%BD%BF%E7%94%A8open%20vswitch%E6%9E%84%E5%BB%BA%E8%99%9A%E6%8B%9F%E7%BD%91%E7%BB%9C%20%E5%9B%BE3.jpeg)  
执行以下命令：

# ovs-vsctl add-br br0 //建立一个名为br0的open vswitch网桥

如果使用vbox或virt-manager把bridge设置为br0即可，如果使用cli kvm则先创建两个文件，用于虚拟网卡的添加于删除。假设这两个文件分别为/etc/ovs-ifup和/etc/ovs-ifdown，则向这两个文件中写入以下内容  
/etc/ovs-ifup

#!/bin/sh

switch='br0'

/sbin/ifconfig $1 0.0.0.0 up

ovs-vsctl add-port ${switch} $1

/etc/ovs-ifdown

#!/bin/sh

switch='br0'

/sbin/ifconfig $1 0.0.0.0 down

ovs-vsctl del-port ${switch} $1

使用以下命令建立虚拟机

kvm -m 512 -net nic,macaddr=00:11:22:33:44:55-net \

tap,script=/etc/ovs-ifup,downscript=/etc/ovs-ifdown-drive\

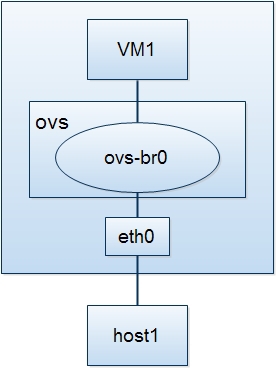
file=/path/to/disk-image,boot=on

kvm -m 512 -net nic,macaddr=11:22:33:44:55:66-net \

tap,script=/etc/ovs-ifup,downscript=/etc/ovs-ifdown-drive\

file=/path/to/disk-image,boot=on

**3、构建虚拟机与物理机相连的网络**

在装有open vswitch的主机上有一个物理网卡eth0，一台主机通过网线和eth0相连，在open vswitch的主机上还装有一台虚拟机，把此虚拟机和连接到eth0的主机挂接到同一个网桥上，实现两者之间的通信，构建结果图如下：  
[](http://img1.sdnlab.com/wp-content/uploads/2015/12/OVS%E5%88%9D%E7%BA%A7%E6%95%99%E7%A8%8B%EF%BC%9A%E4%BD%BF%E7%94%A8open%20vswitch%E6%9E%84%E5%BB%BA%E8%99%9A%E6%8B%9F%E7%BD%91%E7%BB%9C%20%E5%9B%BE4.jpeg)  
执行命令：

# ovs-vsctl add-br br0 //建立一个名为br0的open vswitch网桥

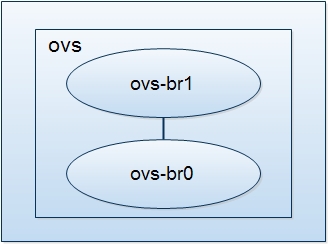
# ovs-vsctl add-port br0 eth0 //把eth0挂接到br0中

# kvm -m 512 -net nic,macaddr=00:11:22:33:44:55-net \

tap,script=/etc/ovs-ifup,downscript=/etc/ovs-ifdown-drive\

file=/path/to/disk-image,boot=on //ovs-ifup和ovs-ifdown和上一节中相同

**4、构建网桥和网桥相连的网络**

以上操作都是将多个主机（物理机或虚拟机）连接到同一个网桥上，实现它们之间的通信，但是要构建复杂的网络，就需要多个网桥，在装有open vswitch的主机上建立两个网桥，实现它们之间的连接，构建结果如下：  
[](http://img1.sdnlab.com/wp-content/uploads/2015/12/OVS%E5%88%9D%E7%BA%A7%E6%95%99%E7%A8%8B%EF%BC%9A%E4%BD%BF%E7%94%A8open%20vswitch%E6%9E%84%E5%BB%BA%E8%99%9A%E6%8B%9F%E7%BD%91%E7%BB%9C%20%E5%9B%BE5.jpeg)  
执行命令：

ovs-vsctl add-br br 添加一个名为br0的网桥

ovs-vsctl add-br br1 //添加一个名为br0的网桥

ovs-vsctl add-port br0 patch-to-br1 //为br0添加一个虚拟端口

ovs-vsctl set interface patch-to-br1type=patch //把patch-to-br1的类型设置为patch

ovs-vsctl set interface patch-to-br1 options:peer=patch-to-br0 把对端网桥和此网桥连接的端口名称设置为patch-to-br0

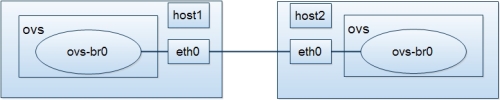
ovs-vsctl add-port br1 patch-to-br0 //为br0添加一个虚拟端口

ovs-vsctl set interface patch-to-br0type=patch //把patch-to-br0的类型设置为patch

ovs-vsctl set interface patch-to-br0options:peer=patch-to-br1 //把对端网桥和此网桥连接的端口名称设置为patch-to-br1

ovs-vsctl set interface patch-to-br0type=patch 和ovs-vsctl set interface patch-to-br0 options:peer=patch-to-br1是对ovs-database的操作，有有兴趣的同学可以参考ovs-vswitchd.conf.db.5

**5、在不同的主机之间构建网桥之间的连接**

在两台机器上分别安装上open vswitch并创建网桥，分别为两个网桥添加物理网卡，然后通过网线连接两个网桥，实现两个网桥之间的互通。构建结果图如下：  
[](http://img1.sdnlab.com/wp-content/uploads/2015/12/OVS%E5%88%9D%E7%BA%A7%E6%95%99%E7%A8%8B%EF%BC%9A%E4%BD%BF%E7%94%A8open%20vswitch%E6%9E%84%E5%BB%BA%E8%99%9A%E6%8B%9F%E7%BD%91%E7%BB%9C%20%E5%9B%BE6.jpeg)  
执行命令：  
host1

#ovs-vsctl add-br br0 //添加名为br0的网桥

#ovs-vsctl add-port br0 eth0 //把eth0挂接到br0上

host2

#ovs-vsctl add-br br0 //添加名为br0的网桥

#ovs-vsctl add-port br0 eth0 //把eth0挂接到br0上

然后使用网线把host1的eth0和host2的eth0相连即可。

使用上边五种方法的组合就可以构建出各种复杂的网络，为各种实验提供网络的支持。

文章摘自 SDN QQ群#兰大-小学生，文章来源他个人SDN技术博客：http://lamoop.com/post/2013-11-18/40060149263