网络虚拟化

> 本文首发于我的公众号 **Linux云计算网络（id: cloud_dev）** ，专注于干货分享，号内有 **10T** 书籍和视频资源，后
> 取，欢迎大家关注，二维码文末可以扫。

CONTENTS ×

前言

网络虚拟化相对计算、存储虚拟化来说是比较抽象的，以我们在学校书本上学的那点网络知识来理解网络虚拟化可

在我们的印象中，网络就是由各种网络设备（如交换机、路由器）相连组成的一个网状结构，世界上的任何两个人
接。

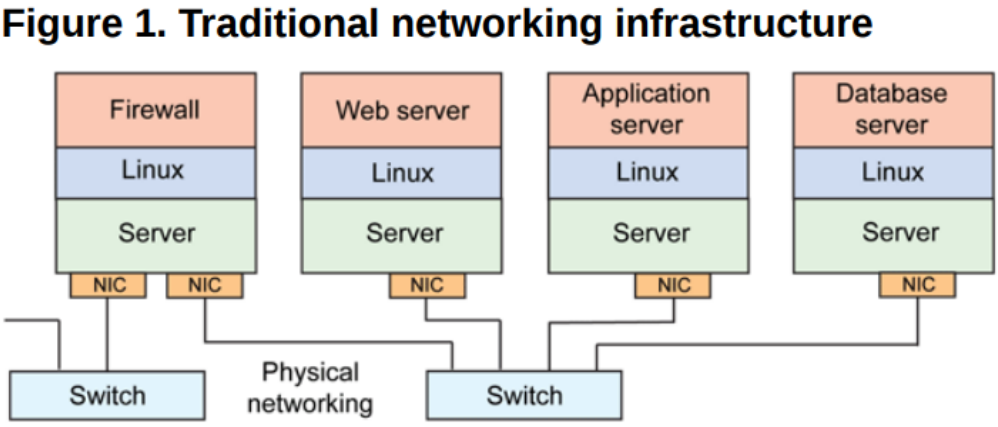
带着这样一种思路去理解网络虚拟化可能会感觉云里雾里——这样一个庞大的网络如何实现虚拟化？

其实，网络虚拟化更多关注的是数据中心网络、主机网络这样比较「细粒度」的网络，所谓细粒度，是相对来说的，是深入到某一台物理主机之上的网络结构来谈的。

如果把传统的网络看作「宏观网络」的话，那网络虚拟化关注的就是「微观网络」。网络虚拟化的目的，是要节省物理主机的网卡设备资源。从资源这个角度去理解，可能会比较好理解一点。

传统网络架构

在传统网络环境中，一台物理主机包含一个或多个网卡（NIC），要实现与其他物理主机之间的通信，需要通过自身的 NIC 连接到外部的网络设施，如交换机上，如下图所示。



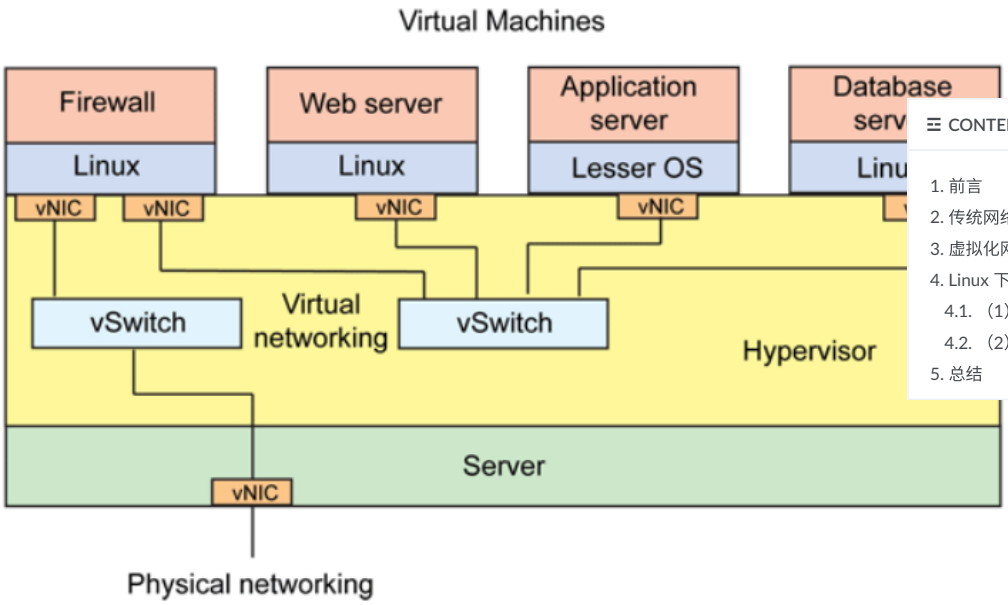
这种架构下，为了对应用进行隔离，往往是将一个应用部署在一台物理设备上，这样会存在两个问题，1）是某些应用大部分情况可能处于空闲状态，2）是当应用增多的时候，只能通过增加物理设备来解决扩展性问题。不管怎么样，这种架构都会对物理资源造成极大的浪费。

虚拟化网络架构

为了解决这个问题，可以借助虚拟化技术对一台物理资源进行抽象，将一张物理网卡虚拟成多张虚拟网卡（vNIC），通过虚拟机来隔离不同的应用。

这样对于上面的问题 1），可以利用虚拟化层 Hypervisor 的调度技术，将资源从空闲的应用上调度到繁忙的应用上，达到资源的合理利用；针对问题 2），可以根据物理设备的资源使用情况进行横向扩容，除非设备资源已经用尽，否则没有必要新增设备。这种架构如下所示。

Figure 2. Virtualized networking infrastructure

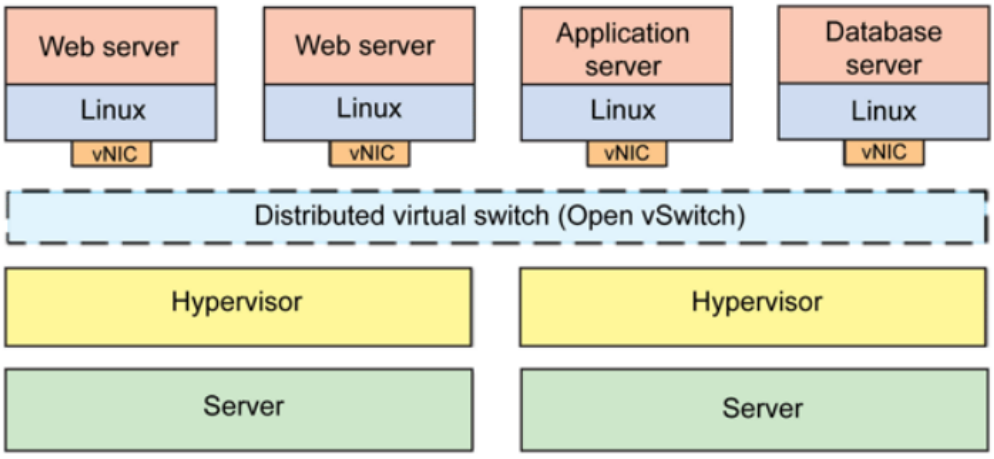


其中虚拟机与虚拟机之间的通信，由虚拟交换机完成，虚拟网卡和虚拟交换机之间的链路也是虚拟的链路，整个主机内部构成了一个虚拟的网络，如果虚拟机之间涉及到三层的网络包转发，则又由另外一个角色——虚拟路由器来完成。

一般，这一整套虚拟网络的模块都可以独立出去，由第三方来完成，如其中比较出名的一个解决方案就是 Open vSwitch（OVS）。

OVS 的优势在于它基于 SDN 的设计原则，方便虚拟机集群的控制与管理，另外就是它分布式的特性，可以「透明」地实现跨主机之间的虚拟机通信，如下是跨主机启用 OVS 通信的图示。

Figure 3. The distributed virtual switch



总结下来，网络虚拟化主要解决的是虚拟机构成的网络通信问题，完成的是各种网络设备的虚拟化，如网卡、交换设备、路由设备等。

Linux 下网络设备虚拟化的几种形式

为了完成虚拟机在同主机和跨主机之间的通信，需要借助某种“桥梁”来完成用户态到内核态（Guest 到 Host）的数据传输，这种桥梁的角色就是由虚拟的网络设备来完成，上面介绍了一个第三方的开源方案——OVS，它其实是一个融合了各种虚拟网络设备的集大成者，是一个产品级的解决方案。

但 Linux 本身由于虚拟化技术的演进，也集成了一些虚拟网络设备的解决方案，主要有以下几种：

（1）TAP/TUN/VETH

TAP/TUN 是 Linux 内核实现的一对虚拟网络设备，TAP 工作在二层，TUN 工作在三层。Linux 内核通过 TAP/TUN 设备向绑定该设备的用户空间程序发送数据，反之，用户空间程序也可以像操作物理网络设备那样，向 TAP/TUN 设备发送数据。

基于 TAP 驱动，即可实现虚拟机 vNIC 的功能，虚拟机的每个 vNIC 都与一个 TAP 设备相连，vNIC 之于 TAP 就如同 NIC 之于 eth。

当一个 TAP 设备被创建时，在 Linux 设备文件目录下会生成一个对应的字符设备文件，用户程序可以像打开一个普通文件一样对这个文件进行读写。

如，当对这个 TAP 文件执行 write 操作时，相当于 TAP 设备收到了数据，并请求内核接受它，内核收到数据后将根据网络配置进行后续处理，处理过程类似于普通物理网卡从外界收到数据。当用户程序执行 read 请求时，相当于向内核查询 TAP 设备是否有数据要发送，有的话则发送，从而完成 TAP 设备的数据发送。

TUN 则属于网络中三层的概念，数据收发过程和 TAP 是类似的，只不过它要指定一段 IPv4 地址或 IPv6 地址，并描述其相关的配置信息，其数据处理过程也是类似于普通物理网卡收到三层 IP 报文数据。

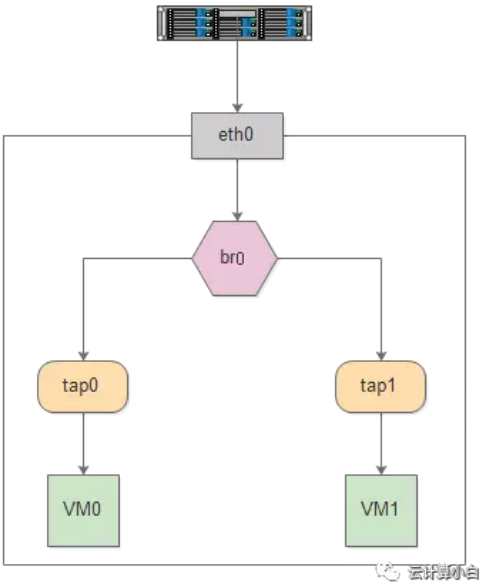
VETH 设备总是成对出现，一端连着内核协议栈，另一端连着另一个设备，一个设备收到内核发送的数据后，会发送
设备通常用于容器中两个 namespace 之间的通信。

（2）Bridge

Bridge 也是 Linux 内核实现的一个工作在二层的虚拟网络设备，但不同于 TAP/TUN 这种单端口的设备，Bridge 实现
虚拟交换机，具备和物理交换机类似的功能。

Bridge 可以绑定其他 Linux 网络设备作为从设备，并将这些从设备虚拟化为端口，当一个从设备被绑定到 Bridge 上
的交换机端口上插入了一根连有终端的网线。

如下图所示，Bridge 设备 br0 绑定了实际设备 eth0 和 虚拟设备设备 tap0/tap1，当这些从设备接收到数据时，会发送给 br0 ，br0 会根据 MAC 地址与端口的映射关系进行转发。



因为 Bridge 工作在二层，所以绑定到它上面的从设备 eth0、tap0、tap1 均不需要设 IP，但是需要为 br0 设置 IP，因为对于上层路由器来说，这些设备位于同一个子网，需要一个统一的 IP 将其加入路由表中。

这里有人可能会有疑问，Bridge 不是工作在二层吗，为什么会有 IP 的说法？其实 Bridge 虽然工作在二层，但它只是 Linux 网络设备抽象的一种，能设 IP 也不足为奇。

对于实际设备 eth0 来说，本来它是有自己的 IP 的，但是绑定到 br0 之后，其 IP 就失效了，就和 br0 共享一个 IP 网段了，在设路由表的时候，就需要将 br0 设为目标网段的地址。

总结

1. 传统网络架构到虚拟化的网络架构，可以看作是宏观网络到微观网络的过渡
2. TAP/TUN/VETH、Bridge 这些虚拟的网络设备是 Linux 为了实现网络虚拟化而实现的网络设备模块，很多的云开源项目的网络功能都是基于这些技术做的，比如 Neutron、Docker network 等。
3. OVS 是一个开源的成熟的产品级分布式虚拟交换机，基于 SDN 的思想，被大量应用在生产环境中。

PSS：

我的博客即将搬运同步至腾讯云+社区，邀请大家一同入驻：https://cloud.tencent.com/developer/support-plan

---

CONTENTS

分类: 云计算, 虚拟化
标签: 云计算, 虚拟化

« 上一篇： 内存虚拟化
» 下一篇： 从 Bridge 到 OVS，探索虚拟交换机

posted @ 2017-12-14 12:51 CloudDeveloper 阅读(11820) 评论(4) 编辑 收藏

评论列表

#1楼 2017-12-14 15:00 myg

写的不错，如能针对每种 虚拟化网络设备 举些例子那就更好了。

支持(0) 反对(0)

#2楼 楼主 2017-12-14 16:03 CloudDeveloper

@ myg
好主意！

支持(0) 反对(0)

#3楼 2017-12-15 14:03 myg

特别是如能针对 OVS 是如何基于 SDN 的设计的 这 块 ，加以说明 就较好了

支持(0) 反对(0)

#4楼 楼主 2017-12-16 09:31 CloudDeveloper

@ myg
后续会有的，可以关注我的公众号

支持(0) 反对(0)