

SR-IOV 技术在 OpenStack 中的应用^①

张 驰¹, 张 傲²

¹(武汉邮电科学研究院, 武汉 430074)

²(烽火信息集成技术有限公司, 武汉 430074)

摘 要: 在 OpenStack 云平台中, 一台物理服务器上可能同时运行着十几台虚拟机, 这对于物理服务器的 I/O 性能要求是非常高的. 因此, I/O 虚拟化技术的效率对于整个 OpenStack 云平台的网络性能提升都有着至关重要的作用. 为了提高系统整体的网络性能, 在 OpenStack 云平台中引入 SR-IOV 技术成为了一种可选的方式. 本文通过对比实验测试了 SR-IOV 技术对于 OpenStack 云平台上网络 I/O 性能的影响. 最终对实验结果进行分析可知, 在引入 SR-IOV 技术后, OpenStack 云平台上的计算节点 I/O 虚拟化性能提升了大概 50%.

关键词: 云计算; OpenStack 云平台; I/O 虚拟化; 网络性能; SR-IOV 技术

引用格式: 张驰, 张傲. SR-IOV 技术在 OpenStack 中的应用. 计算机系统应用, 2017, 26(9): 246-252. <http://www.c-s-a.org.cn/1003-3254/5925.html>

Application of SR-IOV Technology in OpenStack

ZHANG Chi¹, ZHANG Ao²

¹(FiberHome Technologies Group, Wuhan 430074, China)

²(Wuhan Fiberhome Integration Technologies Co. Ltd., Wuhan 430074, China)

Abstract: On an OpenStack cloud platform, a server may simultaneously run more than a dozen virtual machines, which requires high system network I/O performance. Therefore, the efficiency of I/O virtualization is important for the improvement of network performance on OpenStack cloud platform. In order to improve the overall network performance of the system, introducing SR-IOV technology into the OpenStack cloud platform is an option. The influence of SR-IOV technology on network I/O performance of OpenStack cloud platform is tested by contrast experiments. Finally, the experiment results show that, after the introduction of SR-IOV technology, the I/O virtualization performance of computing nodes in OpenStack cloud platform have increased by about 50%.

Key words: cloud computing; OpenStack cloud platform; I/O virtualization; network performance; SR-IOV technology

随着云时代的来临, 各种各样的云计算平台应运而生, 例如亚马逊公司的 AWS 云平台以及微软公司的 Azure 云平台等. 而围绕着开源的优势, OpenStack 云平台在近几年得到了越来越多的个人开发者和企业的支持, 成为当今最火的云平台之一.

而在虚拟化技术中, 内存虚拟化技术和 CPU 虚拟化技术发展都已较为成熟, 但是 I/O 虚拟化技术的发展却相对缓慢, 远远没有达到业界的要求. I/O 虚拟化技术的落后不仅影响了物理服务器的 I/O 虚拟化效率,

更是影响了虚拟机的整体性能^[1]. SR-IOV(Single-Root I/O Virtualization, 单根 I/O 虚拟化)技术是一种基于物理硬件的虚拟化解方案, 可以提高物理 I/O 设备的性能与可扩展性. SR-IOV 技术允许在虚拟机之间高效共享 PCIe 设备, 由于该技术是基于硬件实现的, 可以使虚拟机获得与宿主机相似的 I/O 性能^[2].

随着云环境里虚拟机(Virtual Machine, VM)的数量增多, 云平台中虚拟机相互之间的通信, 以及虚拟机和互联网之间的通信, 导致了网络虚拟化的基本需求.

^① 收稿时间: 2016-12-20; 采用时间: 2017-01-09

在 OpenStack 云平台中,负责网络功能的组件称为 Neutron 组件. Neutron 组件具有灵活部署虚拟网络的能力,但是其性能问题一直是影响 OpenStack 大规模应用的瓶颈之一,特别是 I/O 性能问题更是其中重要的一环.由于 Neutron 组件需要在物理服务器中创建大量的虚拟设备,例如虚拟端口以及虚拟交换机等,这些虚拟的网络组件会对所在的物理服务器 CPU 造成比较大的开销,并且这些虚拟设备本身具有的—定的性能瓶颈,因此会对整个 OpenStack 云平台的网络 I/O 性能造成极大的影响.同时,在 OpenStack 环境中,一台物理服务器上必然会同时运行多台虚拟机,如果所使用的虚拟化 I/O 方案不支持将单个物理 I/O 设备同时共享给多台虚拟机,那么给云平台中的每台虚拟机都分配单独的物理 I/O 设备显然是不现实的.考虑到 SR-IOV 技术的特点以及优势,将 SR-IOV 技术引入到 OpenStack 云平台便成为了解决 OpenStack 云平台中 Neutron 组件网络 I/O 虚拟化性能较差的一种非常合适的方式.

1 OpenStack 云平台介绍

云计算平台事实上是一种以虚拟化技术为基础,以提高服务器 CPU 和网络资源利用率为目的新型服务系统和运营模式.从云计算所提供的服务层次来划分,云计算从上层到下层可以分为软件即服务(SaaS)、平台即服务(PaaS)和基础设施即服务(IaaS)^[3].OpenStack 是一个典型的 IaaS 层的云计算平台.

OpenStack 诞生于 2010 年 7 月,是由美国国家航空航天局和 RackSpace 公司合作研发而成,目前由 OpenStack 社区进行管理^[4].OpenStack 社区每隔几个月便会发布一个新的版本,以 26 个英文字母为首字母按照从 A 到 Z 的顺序依次命名新的版本,第一个版本的 OpenStack 被命名为 Austin. OpenStack 是一个开源的云操作系统,最显著的特点便是体现在开源上,任何公司和个人开发者都可以参加到该项目中,参与代码开发与测试,从而为 OpenStack 的发展做出贡献.

OpenStack 为用户提供了简单易用的 Web 操作界面和详细而全面的 API 接口.通过浏览器访问 OpenStack 的 Web 管理界面,用户可以以租户或管理员的身份进行登录.租户可以创建自己的私有网络,以及启动虚拟机实例进行操作,管理员则可以管理用户资源的配额和其他控制权限.而通过 OpenStack 提供的 API,开发

者可以轻易地对云平台进行开发和学习. OpenStack 云平台内的各组件是通过交互模式来提供云服务的,各个组件可以任意分配到不同的物理服务器上,具有非常好的灵活性与扩展性.

2 Neutron 组件与虚拟网络

在 OpenStack 中负责云平台虚拟网络的是 Neutron 组件. Neutron 组件的主要作用是在云环境下为不同租户提供建网组网、分配虚拟机 IP 地址、防火墙、负载均衡等服务. Neutron 组件将真实物理网络世界中的网络(network)、子网(subnet)和端口(port)等概念抽象化,应用到云环境的虚拟网络中,使得云平台中的虚拟机可以像物理服务器一样直接连接到虚拟网络上,从而实现在虚拟网络中的数据通信.虽然云平台中网络都是虚拟的,但是这些虚拟的网络、子网等在 OpenStack 提供的操作界面 Horizon 上都是可视化的,用户可以非常轻易地去部署完成一个虚拟私有网络.

2.1 Neutron 组件原理

云计算中所用到的网络技术与传统的网络技术原理类似,只是将原有的传统网络技术运用到了云环境下的虚拟网络环境中.对于 Neutron 组件来说,为了给 OpenStack 平台提供虚拟网络能力,事实上在底层使用到了非常多 Linux 系统中的相关技术.

图 1 所示为 Neutron 组件在 Vlan 网络环境下的原理图.图的左边是 OpenStack 平台上充当计算节点的一台物理服务器,右边是充当网络节点的物理服务器.当租户创建一台虚拟机时,Neutron 组件会相应地给该虚拟机创建一张虚拟网卡,该虚拟网卡设备在计算节点看来类似于 Linux 中的 Tap 设备.该 Tap 设备上包含了 Neutron 组件提供给该虚拟机的 IP 以及 MAC 等信息,虚拟机可以直接通过对 Tap 设备进行数据的读写,从而实现数据的通信.在图 1 中的设备 A 与设备 Q 便是分别属于 VM1 与 VM2 的两个 Tap 设备.

虚拟机的虚拟网卡 Tap 设备连接到了一个名为 qbr 的 Linux 网桥上.在 qbr 网桥上,Neutron 组件利用了 Linux 上的 Iptables 来实现虚拟机的安全组功能,因此 qbr 也称之为安全网桥.

Linux 网桥设备 qbr 通过一对 Veth 设备 qvb 与 qvo 直接连接到 OVS(Open VSwitch)软件所创建的虚拟交换机 br-int 上. OVS 是一套可以创建高性能虚拟交换机的软件,它所创建的虚拟交换机相比于传统的

Linux 网桥设备, 具有更好的性能以及支持更多的功能. 从虚拟交换机 br-ethx 方向转发过来的数据报文, 都需要在虚拟交换机 br-int 上进行 Vlan id 的转换. 同理, 从虚拟交换机 br-int 传向虚拟交换机 br-ethx 的数据报文, 也需要在虚拟交换机 br-ethx 上进行 Vlan id 的转换. 这是因为 qvo 设备所分配到的 Vlan id 是只属于云平台虚拟网络内部的, 该 Vlan id 会对应一个真正能在云平台业务网段上进行传输的 Vlan id. 这样虚拟机所产生的带有内部 Vlan id 的数据报文在通过虚拟交换机 br-ethx 后, 经过 Vlan id 的转换, 便可以在云平台的物理业务网上进行传输. 从外界业务网上发来的数据报文经过虚拟交换机 br-int 后, 所带有的 Vlan id 会被转换成内部的 Vlan id, 然后发送给对应的虚拟机, 这样便实现了虚拟网络与物理网络的对应. 这些 Vlan id 的映射都是通过虚拟交换机 br-int 与 br-ethx 中的流表来控制实现的, 而流表的规则则是由 Neutron 组件进行下发的.

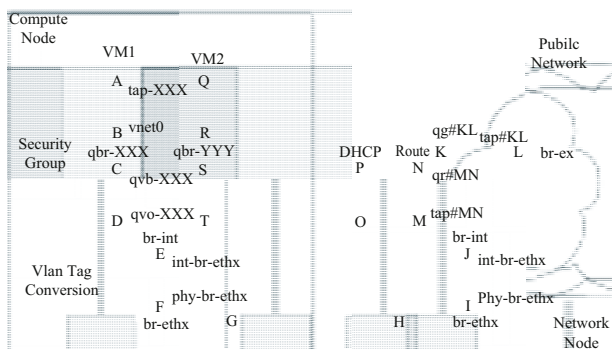


图 1 Neutron 组件原理图

2.2 虚拟机流量走向

在 OpenStack 云环境中, 如果虚拟机需要访问外网或者进行跨网段的访问, 都需要经过网络节点上虚拟路由器的配合. 而同一租户里相同网段的两台虚拟机进行互访是不需要经过网络节点的. 由于本文只涉及到计算节点, 因此下面介绍同一租户下相同网段的两台虚拟机进行互访的情况.

如图 2 中所示, 虚拟机 1 存在于计算节点 1 上, 虚拟机 4 存在于计算节点 2 上, 并且虚拟机 1 和虚拟机 4 属于同一个租户网络的同一个子网内, 两者之间的数据通信将会经过连接到计算节点 1 与计算节点 2 的物理交换机上进行传输. 当虚拟机 1 想发送一个报文给位于不同计算节点上的虚拟机 4 时, 首先会发送一个

ARP 广播报文来确定虚拟机 4 的 MAC 地址. 该 ARP 广播报文会通过 Tap 设备以及 qbr 网桥, 然后被计算节点 1 上的虚拟交换机 br-int 转发到所有与 br-int 相连的接口上. 当广播报文经过计算节点 1 上 br-ethx 时, 会被带上一个外部 Vlan id. 需要注意的是, 同一租户的相同网络里所有的虚拟机发出与接收的报文都会带有相同的外部 Vlan id, 因此该 ARP 报文会被物理交换机转发到所有其他节点上. 当 ARP 报文到达计算节点 2 上时, 该数据报文的 Vlan id 会在虚拟交换机 br-int 上被转换成对应的内部 Vlan id, 并被广播到所有与 br-int 所连的接口上, 最后虚拟机 4 会应答该 ARP 广播. 当虚拟机 1 知道虚拟机 4 的 MAC 地址后, 就可以直接与虚拟机 4 进行数据通信了.

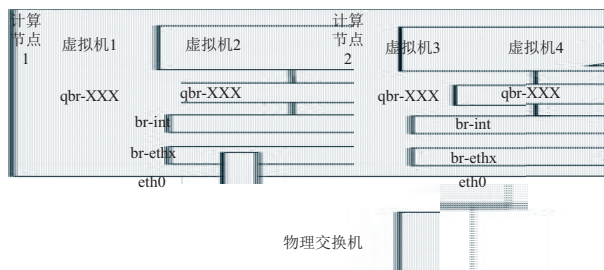


图 2 虚拟机之间互相访问

2.4 Neutron 组件性能瓶颈

通过上面的分析可以看到, 为了实现 OpenStack 中的虚拟网络, Neutron 组件在计算节点上创建了非常多的虚拟网络设备. 这其中包括虚拟机的虚拟网卡 Tap 设备、OVS 虚拟交换机 br-int 和 br-ethx 等. 利用这些虚拟网络设备, Neutron 组件不仅完成了云平台中虚拟网络的创建, 还将整个虚拟网络与真实物理网络相连通, 实现了对整个服务器集群网络资源的虚拟化. 但是另一方面, 这些虚拟网络设备的建立和运行会给云平台上的计算节点带了很大的 CPU 开销, 加上这些虚拟网络设备本身存在一定的缺陷和性能瓶颈, 会极大地减少计算节点上的网络带宽.

举个例子, 假如在一台计算节点上, 使用一般的英特尔 82599ES 万兆网卡, 那么该计算节点最大的网络 I/O 吞吐量可能只能达到 5~6 Gbps 左右. 考虑到在云计算中心, 可能拥有数以万计的物理服务器, 那么这种程度的 I/O 性能损耗显然是一种极大的浪费. 因此寻找一种更加优秀的 I/O 虚拟化方案来替代目前 Neutron 实现虚拟化网络的方式, 有可能解决 OpenStack 云平

台网络 I/O 性能瓶颈问题。

3 SR-IOV 技术与 Neutron 组件

3.1 SR-IOV 技术简介

为提高服务器里虚拟机收发报文的性能和可伸缩性,解决 I/O 虚拟化最后一公里的问题,提出了基于硬件的 SR-IOV 虚拟化解决方案。SR-IOV 规范定义了一种虚拟化 PCIe 设备的标准机制。该机制能将一个 PCIe 以太网设备虚拟成多个 PCIe 设备。用户可以将每个虚拟的 PCIe 设备直接分配给虚拟机,绕过虚拟机监控层(VMM),从而实现低延时和近线速^[5]。

SR-IOV 中具有两种功能类型:物理功能(Physical Function, PF)和虚拟功能(Virtual Function, VF),PF 用于支持 SR-IOV 的 PCI 功能,拥有完全配置或控制 PCIe 设备资源的能力。而 VF 是一种轻量级的 PCIe 功能,与 PF 相关联,可以与同一物理功能关联的其他 VF 共享一个或多个物理资源^[6]。

SR-IOV 技术允许将单个物理网络设备同时共享给多台虚拟机,这样不仅可以实现对物理网卡的共享,同时还可以使虚拟机的流量绕过 Neutron 所创建的虚拟网络设备,直接到达物理网卡上。虚拟机数据处理逻辑跳过了计算节点的虚拟网络设备,由物理网卡决定数据如何处理,从而释放了计算节点的 CPU 资源。这种方式使得虚拟机达到近线速的性能,并且不需要为每台虚拟机单独去分配独立的物理网卡端口。

3.2 在 Neutron 中引入 SR-IOV 技术

将 SR-IOV 技术引入到 OpenStack 中后,可以将虚拟机的虚拟端口与支持 SR-IOV 功能的物理网卡上所虚拟出来的虚拟功能 VF 相关联,而不再需要 Neutron 组件额外地去创建 Tap 设备、qbr 网桥以及 OVS 虚拟交换机了,这样极大地减少了计算机点上的 CPU 开销。

如图 3 所示,虚拟机 1 处于计算节点 1 上,虚拟机 4 处于计算节点 2 上,两台虚拟机属于同一租户下同一网段内。两台计算节点上均有一张带有 SR-IOV 功能的物理网卡,其中虚拟网卡功能 VF1 是分配给虚拟机 1 的,虚拟网卡功能 VF4 是分配给虚拟机 4 的。虚拟机所发出的报文会直接发送到与虚拟机相关联的 VF 上,然后由物理网卡来决定如何对数据报文进行处理。在图 3 中可以看到,从虚拟机 1 发出的报文直接发送到 VF1 上,然后此报文经过物理网卡的处理后流入到物理交换机上,通过物理交换机到达计算节点 2 上,最终

到达虚拟机 4。与传统 Neutron 组件中 Linux 网桥和 OVS 虚拟交换机方式相比,在计算节点内部没有了网桥 qbr 和虚拟交换机 br-int 与 br-ethx 等虚拟设备,虚拟机的数据报文直接通过 VF 传到了物理网卡上。

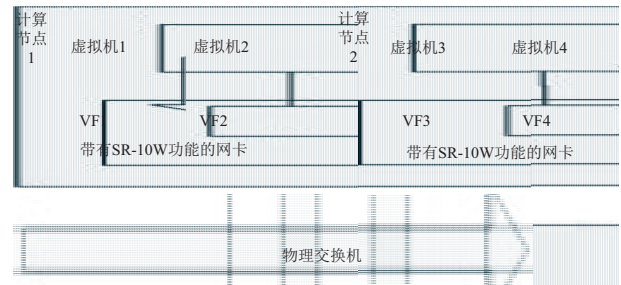


图 3 引入 SR-IOV 技术后的流量路径

4 测试与分析

在之前的章节中详细介绍了 Neutron 组件和 SR-IOV 技术的原理,在本章节,会对 SR-IOV 技术在 OpenStack 云平台上进行具体的测试,证明之前理论分析的正确性,同时得出实验结果。

4.1 测试环境介绍

在测试实验中,一共使用了三台物理服务器组成 OpenStack 环境。其中一台物理服务器用作 OpenStack 云平台的控制节点和网络节点,另外两台物理服务器用作 OpenStack 的计算节点。这三台物理服务器都运行了 Ubuntu 14.04LTS 操作系统。

在表 1 中可以看到,作为 OpenStack 云平台的控制节点/网络节点,这台物理服务器使用的是英特尔六核的处理器,其内存为 64GB。该服务器上存在有三张网卡,其中两张是博通公司 BCM5720 千兆网卡,该网卡用于 OpenStack 平台的管理网络与外网网络。还有一张是英特尔公司的 82599ES 万兆光卡,主要用于云上虚拟机的业务网络。计算节点上的配置信息基本与控制/网络节点类似,唯一的区别是控制/网络节点的英特尔 82599ES 万兆网卡没有开启 SR-IOV 功能,而计算节点上是开启了该功能的。

此次测试环境中物理网络布局图如图 4 所示。测试环境一共包含了三个网络。第一个网络为 OpenStack 云平台的管理网络,其网段为 192.168.1.0/24,主要提供 OpenStack 各个组件之间的内部通信,以及 API 访问端点(Endpoint)。第二个网络为外部网络,网络段位 10.89.0.0/24,此网络是实验环境的外网网络,通过网络可以访问到外界互联网,从而可在服务器中进行软

件以及各种服务的下载. 第三个网络为虚拟机的业务网络. 该网络的作用是提供虚机在计算节点之间, 以及计

算节点和网络节点之间的通信. 该网络所连接的网卡均为支持 SR-IOV 功能的网卡, 也就是图中的 eth3 设备.

表 1 测试环境配置信息表

	控制/网络节点	计算节点1	计算节点2
系统	ubuntu14.04	ubuntu14.04	ubuntu14.04
CPU	Intel E5-2620 v3 6核	InterE5-2650 v2 4核	InterE5-2650 v2 4核
内存	64 GB	128 GB	128 GB
网卡	博通5720千兆网卡*2 Intel 82599ES万兆网卡*1	Intel I350千兆网卡*2 Intel 82599ES万兆网卡*1	Intel I350千兆网卡*2 Intel 82599ES万兆网卡*1

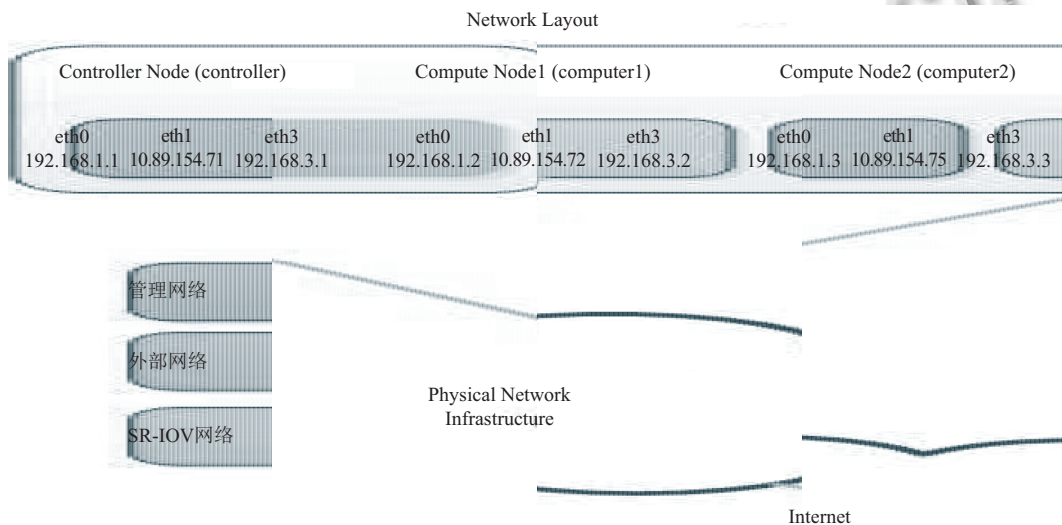


图 4 网络布局

4.2 性能测试

在性能测试中使用 Iperf 软件作为测试工具, 通过在同一租户内的两台虚拟机之间发送数据包来测试计算节点带宽. 测试实验一共分为三组场景, 分别为同一个网段同一台物理服务器上的两台虚拟机的带宽测试, 同一个网段不同物理服务器上的两台虚拟机的带宽测试, 以及不同网段同一物理服务器上的两台虚拟机的带宽测试. 每组测试都会使用 Iperf 工具测试 1 个网络连接, 3 个网络连接以及 5 个网络连接的情况.

虚拟机的 IP 以及所在节点的示意图如图 5 所示. 第一组测试为处于同一网段内同一物理节点上两台 VM 的带宽测试, 因此对比图 5 可知, 我们只需要在虚拟机 vm1 与 vm5 之间使用 Iperf 工具进行测试即可. 这种场景下, 虚拟机的流量是直接被带有 SR-IOV 功能的物理网卡进行处理, 数据包从 vm1 发出到达计算

节点 1 上网卡后便直接到达了 vm5 上. 第二组测试为处于同一网段不同物理节点两台 VM 的带宽测试, 因此只需要在 vm1 与 vm2 之间进行测试即可. 数据包从 vm1 出来后到达计算节点 1 的网卡, 然后经过物理交换机到达计算节点 2 的网卡上, 最终数据包流入到 vm2 上. 第三组测试为处于不同网段同一物理节点上两台 VM 的带宽测试. 因此只需要在 vm2 与 ins2 之间进行测试即可. 该场景下, 数据包从 vm2 发出后, 会先到达网络节点的虚拟路由上, 然后再回到计算节点 2 的网卡上, 最终到达 ins2 上.

最终的测试结果如图 6 所示. 在测试结果柱状图中纵坐标表示计算节点的网络 I/O 吞吐量, 其单位为 Gbit/sec, 横坐标表示一共进行了三组场景的测试, 每组测试实验都有三组数据, 分别为使用 Iperf 工具进行一个连接、三个连接、五个连接下的测试结果. 其中

场景一为同一个网段同一台物理服务器上的两台虚拟机的带宽测试. 场景二为同一个网段不同物理服务器上的两台虚拟机的带宽测试. 场景三为不同网段同一物理服务器上的两台虚拟机的带宽测试

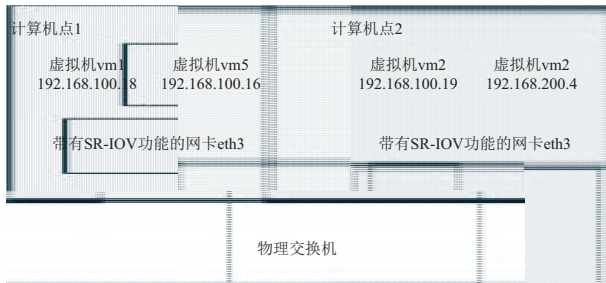


图5 虚拟机所在节点示意图

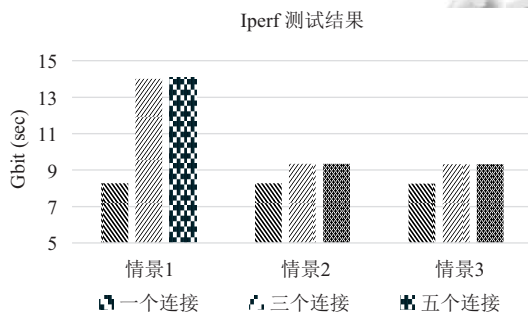


图6 Iperf工具测试结果

将此次实验的测试数据做出统计后, 与传统 Neutron 组件虚拟网桥方式的 I/O 吞吐量作对比, 如图7所示. 可以看出 SR-IOV 技术对于云平台上计算节点的 I/O 性能有巨大的提升.

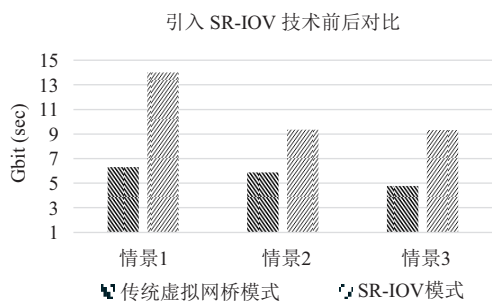


图7 网络 I/O 性能对比

4.3 测试结果分析

在此次实验中使用的是万兆以太网卡, 因此在单台物理服务器上网络带宽所能达到的理论值应该为 10Gbit/sec. 第一种测试场景是相同网段同一节点的两台虚拟机互相测试, 数据包的处理是直接在 SR-

IOV 网卡的上进行的, 虚拟机的数据包发送到该虚拟机绑定的 VF 上, 经网卡处理直接转发到了另一台虚拟机所绑定的 VF 上, 在这过程中并没有真正进行两张网卡之间的发包与收包, 而是仅仅在一张网卡上对包进行了逻辑处理, 因此在物理网卡上对数据包的处理速度接近了 14Gbit/sec, 超过了网卡的理论值 10Gbit/sec, 这是可以理解的.

而在第二种与第三种测试场景下, 数据包的路径经过了多个物理服务器节点, 在网卡上有真正的发包与收包过程, 可以看到在一个网络连接下, 带宽达到了 8.3Gbit/sec. 由于单个连接较少, 并没有达到带宽的极限值, 当将连接数增加到 3 个以及 5 个时, 可以明显看到, 网络的 I/O 吞吐量已经到达极限值约 9.4Gbit/sec, 这已经与该网卡的理论速率 10Gbit/sec 非常接近了. 基本上可以证明在此次试验中, SR-IOV 技术的引用, 的确可以让 OpenStack 云平台中虚拟机的 I/O 吞吐量达到近线速. 而在图7中可以非常清楚的看到, 在引入了 SR-IOV 技术后, OpenStack 云平台上计算节点的最大 I/O 吞吐量有了非常巨大的提升, 提升空间在 40%~60% 之间.

5 结语

本文探讨了 OpenStack 云平台中 Neutron 组件的原理以及不足, 发现将 SR-IOV 技术引入到 OpenStack 云平台是一种可以解决云平台上计算节点网络瓶颈的一种很好的方式. SR-IOV 技术引入后, 通过改变计算节点上虚拟机的访问方式, 使虚拟机不再去依赖 Neutron 组件创建的各种虚拟网桥, 而是利用物理网络硬件的 SR-IOV 功能, 使虚拟机的数据报文直接到达物理网络设备上. 经过测试, 这种方式可以使得云平台的计算节点最大吞吐量提高约 50% 左右.

在其他方面, 还存在部分问题. 第一点, 在 SR-IOV 技术引入到 OpenStack 云平台后, 用户操作界面 Horizon 暂时不能支持 SR-IOV 技术, 但可以通过命令行调用 API 去创建带有 SR-IOV 接口的虚拟机, 并不会影响正常使用. 第二点, 目前的 SR-IOV 技术存在一定的缺陷, 例如物理网卡设备最多能虚拟出来的 VF 数量有限, 但能满足当前 OpenStack 的需求. 第三点, SR-IOV 技术解决了 OpenStack 云平台中计算节点的网络 I/O 问题, 但是对网络节点的 I/O 性能瓶颈却没有影响. 想要解决此问题, 需要引入 DVR 技术或者是

DPDK 等技术才能实现.

参考文献

- 1 李超. SR-IOV 虚拟化技术的研究与优化[硕士学位论文]. 长沙: 国防科学技术大学, 2010.
- 2 杨洪波. 高性能网络虚拟化技术研究[博士学位论文]. 上海: 上海交通大学, 2012.
- 3 罗军舟, 金嘉晖, 宋爱波, 等. 云计算: 体系架构与关键技术. 通信学报, 2011, 32(7): 3-21.
- 4 陈陶, 顾双双, 柳钮滔, 等. 基于 OpenStack Juno 版的私有云平台部署及实践. 物联网技术, 2015, (6): 64-67, 69.
- 5 黄智强. 基于 SR-IOV 的高性能 I/O 虚拟化研究与优化[硕士学位论文]. 上海: 上海交通大学, 2013.
- 6 龚珊珊. 基于 SR-IOV 技术的网卡虚拟化研究与实现[硕士学位论文]. 北京: 中国舰船研究院, 2015.

www.c-s-a.org.cn

www.c-s-a.org.cn