

基于网卡虚拟化的高性能容器网络设计

臧迪¹, 杨志刚¹, 王晶¹, 姚治成², 张伟功³

(1.首都师范大学 信息工程学院, 北京 100048; 2.中国科学院计算技术研究所 先进计算机系统研究中心, 北京 100080;

3.首都师范大学 高可靠嵌入式系统技术北京市工程中心, 北京 100048)

摘要: 单根 I/O 虚拟化技术为传统数据中心提供高效的服务器整合能力和灵活的应用部署能力, 通过将多个网卡直通到虚拟机, 减少额外包复制带来的性能损失, 使得网络 I/O 具有接近主机的性能。然而, 在网络功能虚拟化场景下单独使用单根 I/O 虚拟化技术会降低传统数据中心的网络 I/O 虚拟化性能。针对网络功能虚拟化长链场景, 结合单根 I/O 虚拟化技术和软件虚拟化技术, 设计基于网卡虚拟化的高性能容器网络。通过转发模块判断网络流量的目的地址, 寻找最优的流量转发路径, 实现流量的灵活转发。利用基于脚本程序的自动化部署模块, 对每个节点业务进行支持动态增删服务的配置, 便于用户对网络进行管理和修改。实验结果表明, 在网络功能虚拟化长链场景下, 相比单根 I/O 虚拟化技术, 该网络延迟降低约 20%, 同时能够有效提高网络吞吐量, 解决数据中心的网络 I/O 虚拟化问题。

关键词: 单根 I/O 虚拟化; 网络功能虚拟化; 网卡虚拟化; 容器网络; 数据中心

开放科学(资源服务)标志码(OSID):



中文引用格式: 臧迪, 杨志刚, 王晶, 等. 基于网卡虚拟化的高性能容器网络设计[J]. 计算机工程, 2022, 48(7): 214-219.

英文引用格式: ZANG D, YANG Z G, WANG J, et al. Design of high-performance container network based on network interface card virtualization[J]. Computer Engineering, 2022, 48(7): 214-219.

Design of High-Performance Container Network Based on Network Interface Card Virtualization

ZANG Di¹, YANG Zhigang¹, WANG Jing¹, YAO Zhicheng², ZHANG Weigong³

(1. Information Engineering College, Capital Normal University, Beijing 100048, China;

2. Center for Advanced Computer Systems, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China;

3. Beijing Engineering Research Center of High Reliable Embedded System, Capital Normal University, Beijing 100048, China)

[Abstract] Single Root I/O Virtualization (SR-IOV) technology provides traditional data centers with efficient server integration and flexible application deployment capabilities. To reduce the performance loss caused by additional packet replication, SR-IOV technology connects multiple network interface cards directly to the virtual machine, bringing the network I/O performance closer to that of the host. However, SR-IOV technology in the Network Function Virtualization (NFV) scenario degrades the performance of traditional data center virtualization. For the combined use of SR-IOV and software virtualization technology in such scenarios, a high-performance container network based on network interface card virtualization is designed. A forwarding module is used to determine the destination address, find the optimal forwarding path, and ensure the flexible forwarding of network traffic. A fully automatic deployment module based on a script program is used to support dynamic addition and deletion services configuration for the business of each node, facilitate users to manage and modify the network, and improve the module's adaptability. Experimental results show that the network's delay in the NFV long chain scenario is about 20% lower than that of SR-IOV technology, as the former improves throughput. This solves network I/O virtualization issues in data centers.

[Key words] Single Root I/O Virtualization (SR-IOV); Network Function Virtualization (NFV); network interface card virtualization; container network; data center

DOI: 10.19678/j.issn.1000-3428.0061999

基金项目: 国家自然科学基金(62076168)。

作者简介: 臧迪(1995—), 男, 硕士, 主研方向为计算机体系结构、智能芯片; 杨志刚(通信作者), 副教授、硕士; 王晶, 教授、博士; 姚治成, 工程师、硕士; 张伟功, 研究员、博士。

收稿日期: 2021-07-07 **修回日期:** 2021-09-26 **E-mail:** zhg_young@cnu.edu.cn

0 概述

云计算具有资源利用率高、灵活和部署便利等特点,因此,越来越多的企业逐渐从传统模式过渡到云计算模式。然而,云计算需要数据中心做支撑,当数据包通过互联网到达数据中心时,数据中心会采用防火墙、深度报文检测、入侵检测、网络地址转换等^[1]技术对数据包进行处理。网络中间件与Web业务紧密耦合,能够满足运营商电信级的业务要求。随着云平台规模的不断扩大,专用中间件的部署需求不断增加,导致传统软硬一体化的封闭式结构功能单一、不灵活、价格昂贵的缺点逐渐显现。然而,软件定义网络(Software Define Network, SDN)结合软硬解耦原理对网络功能进行虚拟化。这种网络功能虚拟化(Network Function Virtualization, NFV)使得网络架构更灵活、功能更丰富、价格更低廉,推动云计算平台的发展,加快了容器技术、虚拟交换机技术和网卡虚拟化技术的成熟和推广。

NFV技术实现了软硬件解耦,并且具有灵活的网络部署和动态扩展的特点。传统的NFV平台主要依靠虚拟机技术,基于虚拟机的NFV架构能够将空间资源隔离,具有较优的灵活性和按需分配性,但是虚拟机带来的资源消耗较大^[2]。例如,在NFV开放平台(OPNFV)^[3]框架下,使用OpenStack虚拟机技术与OpenDaylight构建虚拟化基础设施管理器,产生的资源消耗较大。

容器技术是一种基于内核命名空间的资源隔离技术,近年来,作为下一代虚拟化技术进入快速普及阶段。由于容器技术没有硬件损失,因此具有接近裸机的性能。此外,容器技术的启动速度可以达到毫秒级,因此在资源利用率和灵活部署方面有很大优势。采用容器技术构建虚拟网络功能能够有效解决资源开销大的问题^[4-5]。基于容器技术构建虚拟网络功能通常采用Linux自带的桥接模式、overlay网络或者虚拟交换机部署^[6-7]。这种模式虽然可以实现流量转发,但是在跨主机情况下,流量经过网卡需要两次转发,导致延迟敏感型应用产生较大的性能损失,并且CPU的负载较大,导致带宽下降。

本文提出目标感知的容器网络用于NFV长链场景。结合硬件虚拟化技术和软件模拟技术判断流量的目的地址,寻找最优的流量转发路径,优化NFV场景下经过每个业务节点的网络延迟和吞吐量,同时利用基于脚本程序的自动化部署模块对每个节点业务进行配置,便于用户对网络进行管理和修改。

1 相关工作

在NFV部署中,NFV服务链的优化技术和网络I/O虚拟化技术是影响网络性能最直接的因素。

在NFV服务链的优化技术方面,文献[8]将整条服务链的网络功能拆分成若干个子模块,消除原有服务链中的冗余模块,加快服务链的处理速度。文献[9]通过拷贝、合并等方式对原有的串行处理且无依赖关系的网络功能进行并行化包处理,优化整条服务链的延迟和吞吐量性能。文献[10]提出IETF(Internet Engineering Task Force),在云数据中心通过优化虚拟网络功能的调度完成时间来减少延迟。

现有的网络I/O虚拟化技术分为单根I/O虚拟化(Single Root I/O Virtualization, SR-IOV)技术、软件模拟虚拟化技术和网卡直通技术。SR-IOV技术是由PCI-SIG制定的输入输出虚拟化标准^[11],通过直接I/O技术减少额外包复制带来的性能损失,达到接近主机的性能。文献[12]提出在标准X86服务器上使用SR-IOV技术部署虚拟化网络功能,以达到提高吞吐量的目的。基于软件模拟的网络I/O虚拟化技术又称虚拟交换机技术。虚拟交换机技术的工作原理与物理交换机类似,其两端分别连接着物理网卡和多块虚拟网卡,同时虚拟交换机内部会维护一张映射表,根据MAC地址寻找对应的虚拟机链路进而完成数据转发。文献[13]提出一种虚拟交换机开源软件OpenVSwitch。文献[14]对NFV场景下虚拟交换机的性能进行评估与分析。网卡直通技术使用硬件辅助虚拟化技术,将宿主机中的物理PCI设备直接分配给客户机使用,客户机以独占方式访问该宿主机的PCI/PCI-E设备^[15]。文献[16]对X86的网卡直通技术进行分析,指出物理网卡直通更容易受到“拒绝攻击”,从而影响每个服务器的性能。由于网卡直通技术只能被一个虚拟机独享,安全性也存在隐患,因此逐渐被单根虚拟化技术和软件模拟虚拟化技术所取代。

单根I/O虚拟化技术和软件模拟虚拟化技术已成为主流技术,为容器提供网桥的功能。但是单根I/O虚拟化技术在主机内部传输方面比软件模拟虚拟化技术性能差,而单根虚拟化技术在跨主机的性能要优于软件模拟虚拟化技术。在NFV长链的场景下,需要同时采用主机内部的传输和主机外部的传输,使用单一某种技术难以满足NFV长链场景的需求。针对上述问题,本文提出一种目标感知的容器网络,通过优化SR-IOV技术在NFV场景下的性能,降低NFV场景下网络的延迟并且提高网络吞吐量。

2 目标感知的容器网络

目标感知的容器网络设计方案主要是对硬件虚拟化进行改进。传统的SR-IOV^[17]技术是通过驱动把单一的物理网卡虚拟出独立的虚拟网卡,并通过驱动将网卡划分为多个地址段。SR-IOV技术主要

有虚拟功能(Virtualization Function, VF)和物理功能(Physical Function, PF)。SR-IOV 技术原理示意图如图 1 所示。

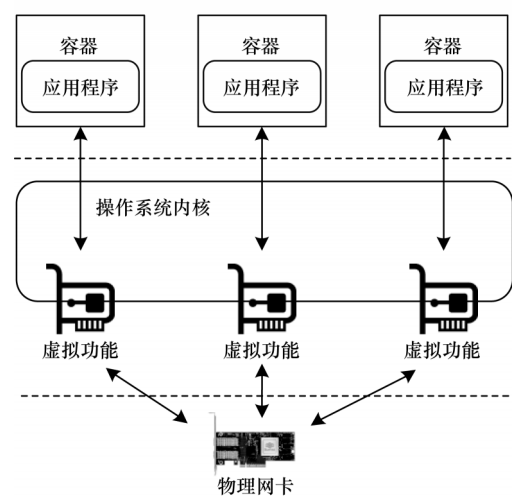


图 1 SR-IOV 技术原理示意图
Fig.1 Schematic diagram of SR-IOV technology principle

SR-IOV 技术主要分为 3 层,最底层是支持 SR-IOV 技术的硬件网卡,中间层是操作系统内核,最上层是提供某种服务的容器。SR-IOV 技术的原理是网卡通过 SR-IOV 技术虚拟出多个虚拟功能,虚拟出的 VF 可以跨过操作系统内核直接分配给提供某种服务的容器,但是在 NFV 级联的场景下,单独使用 SR-IOV 技术会导致其性能下降,因此针对 NFV 场景,本文设计目标感知的容器网络。

在 NFV 场景下需要将各容器相互级联,但是对于图 1 的设计,单独使用 SR-IOV 技术的网络性能会出现瓶颈。因此,为解决 SR-IOV 技术在 NFV 场景下的性能瓶颈问题,本文提出目标感知的容器网络,其结构如图 2 所示。

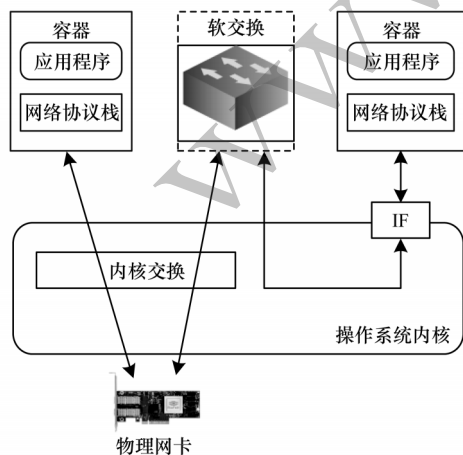


图 2 目标感知的容器网络结构
Fig.2 Structure of container network for target perception

该网络结构的最顶层是使用容器技术把每个 NFV 独立应用分别放在不同的容器中,使得容器之间的应用相互隔离,如果需要某种应用,可以直接使用特定功能的容器镜像,以便于维护,例如容器应用支持防火墙、深度包检测、网络地址转换和负载均衡等功能。本文网络利用 SR-IOV 技术跨过操作系统内核,通过软交换实现容器与容器之间的交互,因此在网卡流量发送到主机时需要判断流量发送位置。当网卡跨过操作系统内核到容器的网络中间件时,目标感知的容器网络通过转发模块判断目标的地址,通过目的地址判断网络流量的方向,之后提交给最顶层。

2.1 转发模块

目标感知的容器网络结合网络地址转换(NAT)和 OpenFlow^[18-19]对转发模块进行设计,主要依据软件虚拟技术实现流量转发。OpenVSwitch 是目前主流的软件模拟技术,它提供 OpenFlow 协议并且 OpenVSwitch 的转发性能比 Linux 下的网桥模式较优。因此,转发模块使用基于 NAT 和 OpenVSwitch 的网络设计实现流量的转发。当前转发分为单根虚拟化的虚拟网卡和 OpenVSwitch 的虚拟网卡 2 个方向。流量转发会判断发送的目的地址是否在 NFV 长链中,如果在 NFV 长链中,转发模块将流量提交到上层并对数据包进行处理,例如深度包检测或者负载均衡等服务,如果不在 NFV 长链中,转发模块就会丢弃该流量。转发模块还可以根据用户输入的 OpenFlow 语法更改路径,使用户调整转发路径,实现灵活转发,即可以对容器的流量进行灵活规划。

2.2 自动化部署模块

本文网络在原有的基础上增加了软件交换技术和转发模块配置,因此提高了部署的复杂性。为便于部署,本文使用基于脚本程序的自动化部署方案,无需手动干预,并且对每个节点业务进行配置。配置包括整体网络框架配置、转发模块配置、网络中间件设计以及服务节点设计。自动化部署序列图如图 3 所示。

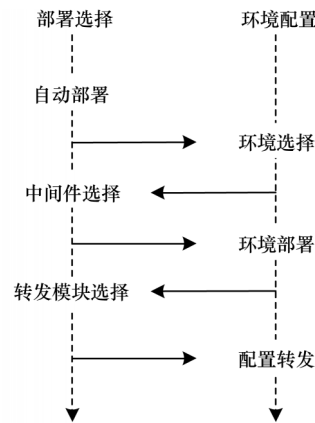


图 3 自动化部署序列图
Fig.3 Sequence diagram of automatic deployment

自动化部署模块主要分为3个部分:1)部署选择,主要是配置容器环境以及虚拟交换机和单根虚拟化;2)部署的环境配置,当用户启动自动化部署模块时,自动化部署会让用户选择自动化部署模式,进而选择中间件,例如容器提供的服务(防火墙服务、深度包检测服务、负载均衡服务、网页服务等);3)对转发模块进行配置,转发模块分为使用网络地址转换和使用OpenFlow协议自定义转换。转发模块配置完成之后,整体网络实现自动化部署。

自动化部署模块的整体架构如图4所示。从图4可以看出,最底层使用SR-IOV技术生成虚拟网卡,应用层使用Docker容器技术,系统内核层中的软交换使用OpenVSwitch。自动化部署模块能够自动配置转发模块规则以及整体的网络拓扑。

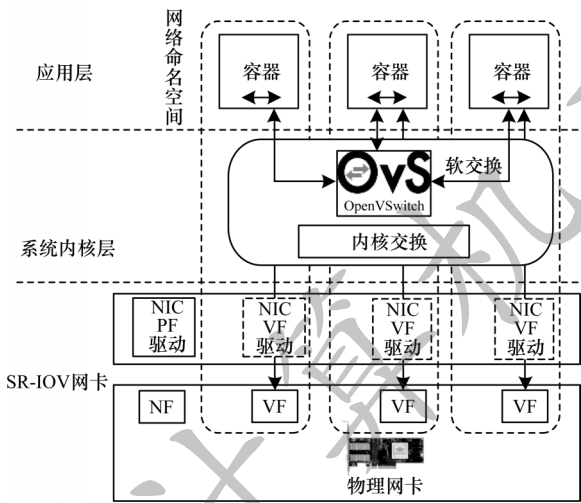


图4 自动化部署模块的整体架构

Fig.4 Overall architecture of automatic deployment module

3 实验与结果分析

3.1 实验环境与配置

本文使用两个测试工具,分别是对网络性能进行测试的Netperf^[20]和对请求数进行测试的Apache Benchmark,测试的内容为长链NFV的网络延迟带宽和每秒请求数。

Web服务器软件Apache2网页应用和Haproxy^[21]负载均衡应用主要是用于测试负载。Haproxy具有较高的可用性和负载均衡能力,适用于负载大的Web站点。

本文选择Docker容器^[22-23],在容器中运行Netperf网络性能评测工具,主要针对TCP或UDP的传输,针对不同应用进行不同模式的网络性能测试。软件虚拟化使用开源OpenVSwitch,硬件虚拟化使用SR-IOV技术。本文实验的硬件环境采用型号Intel E5620 X2和主频2.4 GHz的CPU,内存16 GB,内存频率1 333 MHz,型号82599和带宽为10 Gb/s的网卡。

本文实验的基本环境是使用2台服务器。每台服务器有2个主频2.4 GHz的E 5620处理器,使用Intel 82599网卡,选择Centos系统。

3.2 实验设计

本文配置了SR-IOV和OpenVSwitch的实验环境,并在容器中运行Netperf应用、防火墙、Apache2网页和Haproxy负载均衡应用。目标感知的容器网络拓扑结构如图5所示。

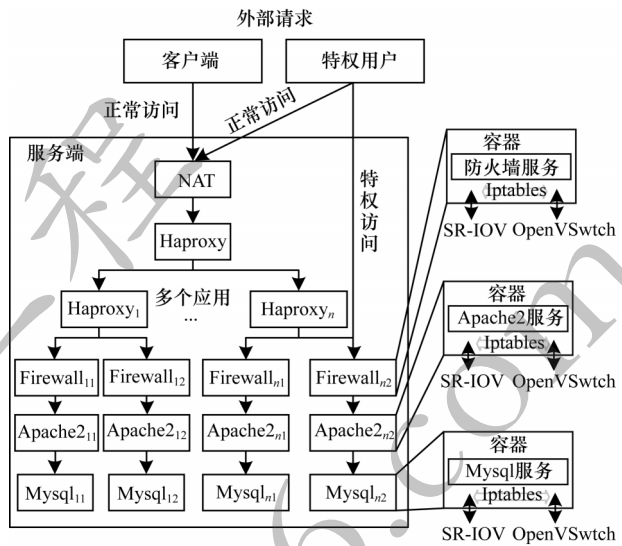


图5 目标感知的容器网络拓扑结构

Fig.5 Topology structure of container network for target perception

在NFV长链下,如果外部请求需要访问Apache2网页,经过NAT、Haproxy、防火墙等一系列长链服务才能够请求成功。在这种情景下,如果单独使用SR-IOV会存在一定的不足,但是目标感知的容器网络可以解决这个问题。

当外部只访问NFV长链中的某一个应用时,本文网络能够解决OpenVSwitch经过操作系统内核后性能降低的问题。

本文网络主要使用内网转发技术,在容器中增加两个网卡,外部提供访问的网卡使用SR-IOV网卡,内部使用OpenVSwitch网卡,通过内网转发将OpenVSwitch网卡相连接。从图5可以看出,当外部访问时,在整个长链NFV的请求中选择一条最优的路径。当外部请求NFV长链中的某一项服务时,可以抽象为图5的网络拓扑结构,这种情况通常出现在排查网络故障和内部数据维护的时候。

3.3 实验结果

首先搭建SR-IOV与OpenVSwitch的环境,分别在SR-IOV和OpenVSwitch环境下部署不同应用,使用Netperf和Apache Benchmark压力测试工具进行精确测试并分析SR-IOV的性能;然后部署SR-IOV与OpenVSwitch混合策略,与未优化前的数据进行对比,在NFV长链下,对目标感知容器网络技术与

单独的 SR-IOV 与 OpenVSwitch 技术的性能进行对比。

在对外提供服务的场景中,网络需要访问 NFV 长链中的所有节点,以保证服务的安全性和完整性。

3.3.1 延迟测试

本文使用 Netperf 测试工具分别测试 SR-IOV 网络、OpenVSwitch 网络和本文网络的延迟。每个实验测试次数为 10 次,最终结果为 10 次的平均值,测试结果如图 6 所示。

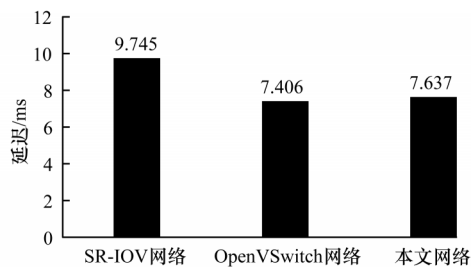


图 6 NFV 长链下不同网络的 Netperf 延迟测试结果对比

Fig.6 Netperf delay test results comparison among different networks in NFV long chain

从图 6 可以看出,在 NFV 长链下,单独使用 SR-IOV 技术的网络延迟为 9.745 ms,使用 OpenVSwitch 技术的网络延迟为 7.406 ms,使用本文技术得到的网络延迟为 7.637 ms。因此,本文网络性能比单独的 SR-IOV 性能提高 21.6%。在 NFV 长链的应用场景中,OpenVSwitch 技术的延迟最优,本文网络与 OpenVSwitch 持平。

3.3.2 NFV 长链下每秒请求数测试

本文在 NFV 长链应用场景下,使用 Apache Benchmark 对 SR-IOV、OpenVSwitch 和本文网络进行请求数测试,测试结果如图 7 所示。

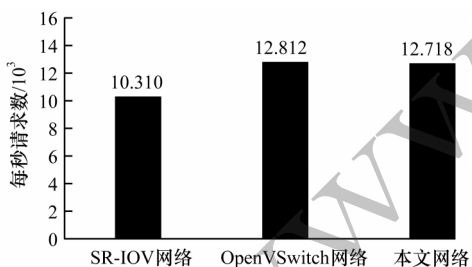


图 7 不同网络的每秒请求数测试结果

Fig.7 Test results of requests per second for different networks

从图 7 可以看出,在 Apache Benchmark 的负载下,当本文访问 NFV 长链服务时,其请求数是每秒 12 718 个,OpenVSwitch 网络的每秒请求数 12 812 个,SR-IOV 网络的每秒请求数 10 310 个。在 NFV 长链的应用场景下,SR-IOV 的每秒请求数是最差的,本文网络的每秒请求数与 OpenVSwitch 持平,本文网络比 SR-IOV 性能提高了 23.35%。

3.3.3 某一项服务延迟测试

在 Netperf 负载下,不同网络访问 NFV 长链中某一项服务的延迟对比如图 8 所示。

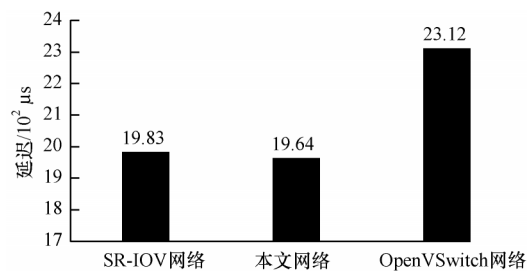


图 8 不同网络访问 NFV 长链某一项服务的延迟对比

Fig.8 Comparison of delay of accessing a service in NFV long chain by different networks

从图 8 可以看出,当访问 NFV 长链某一项服务时,SR-IOV 的延迟为 1 983 μ s,OpenVSwitch 的延迟为 2 312 μ s,本文网络的延迟为 1 964 μ s。因此,当访问单独服务时,OpenVSwitch 的性能最差,本文网络的性能与 SR-IOV 的性能持平。

3.3.4 某一项服务的请求数测试

本文实验对 NFV 长链中单独 Apache2 服务进行请求数测试,测试结果如图 9 所示。

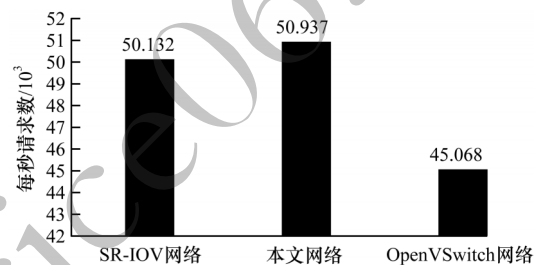


图 9 不同网络的 Apache2 服务请求数

Fig.9 The number of Apache2 service requests of different networks

从图 9 可以看出,在 Apache Benchmark 负载下,本文网络访问单独服务的请求数是每秒 50 937 个,OpenVSwitch 的每秒请求数是 45 068 个,SR-IOV 的每秒请求数是 50 132 个。本文在访问单独服务时与 SR-IOV 的结果持平。

综上可知,本文网络在长链 NFV 下的性能优于单独使用 SR-IOV 和 OpenVSwitch 的性能,并且综合 OpenVSwitch 和 SR-IOV 的优点。在长链 NFV 中,本文网络比 SR-IOV 的性能提高约 20%,在直通访问时,本文网络比软件模拟网络性能提高 15%。

4 结束语

针对网络功能虚拟化场景下网络 I/O 性能瓶颈,本文设计一种目标感知的高性能容器网络,通过转发模块和基于脚本程序的自动化部署模块,寻找最优的流量转发路径,并对每个节点进行业务配置,实现容器网络流量的灵活转发,便于用户对网络进行

管理和修改。实验结果表明,本文设计的容器网络能够有效提高NFV长链场景下的网络性能,为解决数据中心网络I/O虚拟化问题提供新思路。下一步将在单根I/O虚拟化技术的基础上增加OpenFlow流量转发规则,以提高平台流量转发的效率。

参考文献

- [1] PHAM C, TRAN N H, REN S L, et al. Traffic-aware and energy-efficient VNF placement for service chaining: joint sampling and matching approach[J]. *IEEE Transactions on Services Computing*, 2020, 13(1): 172-185.
- [2] MECHTRI M, GHRIBI C, SOUALAH O, et al. NFV orchestration framework addressing SFC challenges[J]. *IEEE Communications Magazine*, 2017, 55(6): 16-23.
- [3] Mark Beierl. OPNFV[EB/OL]. [2021-06-05]. <http://www.creader.com/news/20011219/200112190019.html>.
- [4] 徐江,孟相如,韩晓阳,等. 博弈论组合赋权的虚拟网络映射算法[J]. *小型微型计算机系统*, 2020, 41(7): 1464-1469.
XU J, MENG X R, HAN X Y, et al. Virtual network embedding algorithm based on game theory combination weighting[J]. *Journal of Chinese Computer Systems*, 2020, 41(7): 1464-1469. (in Chinese)
- [5] 黄强,李宁. 5G边缘计算演进[J]. *邮电设计技术*, 2018(11): 68-73.
HUANG Q, LI N. MEC in 5G evolution[J]. *Designing Techniques of Posts and Telecommunications*, 2018(11): 68-73. (in Chinese)
- [6] VIZARRETA P, CONDOLUCI M, MACHUCA C M, et al. QoS-driven function placement reducing expenditures in NFV deployments[C]//*Proceedings of IEEE International Conference on Communications*. Washington D. C., USA: IEEE Press, 2017: 1-7.
- [7] BEN JEMAA F, PUJOLLE G, PARIENTE M. QoS-aware VNF placement optimization in edge-central carrier cloud architecture [C]//*Proceedings of IEEE Global Communications Conference*. Washington D. C., USA: IEEE Press, 2016: 1-7.
- [8] BREMLER-BARR A, HARCHOL Y, HAY D. OpenBox: a software-defined framework for developing, deploying, and managing network functions[C]//*Proceedings of ACM SIGCOMM Conference*. New York, USA: ACM Press, 2016: 511-524.
- [9] SUN C, BI J, ZHENG Z L, et al. NFP: enabling network function parallelism in NFV [C]//*Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. New York, USA: ACM Press, 2017: 1-10.
- [10] GARG P, WANG Y S. NVGRE: network virtualization using generic routing encapsulation[EB/OL]. [2021-06-05]. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.697.7171&rep=rep1&type=pdf>.
- [11] Intel Company. Single root I/O virtualization and sharing 1.0 S-pecification[EB/OL]. [2021-06-05]. https://composter.com.ua/documents/sr-iov1_1_20Jan10_cb.pdf.
- [12] LEIVADEAS A, FALKNER M, PITAEV N. Analyzing service chaining of virtualized network functions with SR-IOV[C]//*Proceedings of the 21st International Conference on High Performance Switching and Routing*. Washington D. C., USA: IEEE Press, 2020: 1-6.
- [13] Linux Foundation. OpenVSwitch, an open virtual switch [EB/OL]. [2021-06-15]. <https://www.openvswitch.org/>.
- [14] ZHANG T Z, LINGUAGLOSSA L, GIACCONE P, et al. Performance benchmarking of state-of-the-art software switches for NFV[J]. *Computer Networks*, 2021, 188: 1-10.
- [15] ABRAMSON D, JACKSON J, MUTHRASANALLUR S, et al. Intel virtualization technology for directed I/O[J]. *Intel Technology Journal*, 2006, 10(3): 1-10.
- [16] RICHTER A, HERBER C, RAUCHFUSS H, et al. Performance isolation exposure in virtualized platforms with PCI passthrough I/O sharing [C]//*Proceedings of International Conference on Architecture of Computing Systems*. Berlin, Germany: Springer, 2014: 171-182.
- [17] DONG Y Z, YANG X W, LI J H, et al. High performance network virtualization with SR-IOV[J]. *Journal of Parallel and Distributed Computing*, 2012, 72(11): 1471-1480.
- [18] 张懿,禹忠,王军选. 基于OpenFlow的控制部署可靠性方案设计[J]. *计算机工程*, 2018, 44(8): 74-78.
ZHANG Y, YU Z, WANG J X. Design of controller deployment reliability scheme based on OpenFlow[J]. *Computer Engineering*, 2018, 44(8): 74-78. (in Chinese)
- [19] 刘海客,李集林,尤启迪,等. 一种OpenFlow网络的动态负载均衡方法[J]. *计算机工程*, 2016, 42(8): 85-90.
LIU H K, LI J L, YOU Q D, et al. A dynamic load balancing method for OpenFlow network[J]. *Computer Engineering*, 2016, 42(8): 85-90. (in Chinese)
- [20] Hewlett-Packard Company. Netperf Manual [EB/OL]. [2021-06-15]. <http://www.cs.kent.edu/~farrell/dist/ref/Netperf-f.html>.
- [21] Willy Tarreau. Haproxy[EB/OL]. [2021-06-15]. <https://www.haproxy.com/>.
- [22] 杨鹏,马志程,彭博,等. 集成Docker容器的OpenStack云平台性能研究[J]. *计算机工程*, 2017, 43(8): 26-31.
YANG P, MA Z C, PENG B, et al. Performance research of OpenStack cloud platform integrated with Docker container[J]. *Computer Engineering*, 2017, 43(8): 26-31. (in Chinese)
- [23] JAMS T. The Docker book[M]. Cambridge USA: MIT Press, 2016.

编辑 薛晋栋