IEEE.org | IEEE Xplore数字图书馆 | IEEE-SA | IEEE Spectrum | 更多网站

购物车(0)|创建帐号|个人登录

访问提供者:

Hytera Communication Corp.Ltd 登出

浏览 我的设置 文件柜 得到帮助

Advertisement

充分 文字视图

Journals & Magazines > IEEE Transactions on Network ... > Volume: 14 Issue: 3 IEEE网络交易... > 卷:14期:3

T-NOVA:用于NFV基础设施的开源MANO堆栈

18 (S)

Michail-Alexandros Kourtis ; Michael J. McGrath; Georgios Gardikis; Geo... 查看所有作者

Export to

Collabrated

管理内容提 醒 添加到引文

提醒

抽象

文件部分

PDF

-世。 介绍

T-NOVA整体观 II. 点

III. 资源发现,选 择,匹配

IV. 服务映射

V. 服务部署

显示完整大纲 ▼

作者

数据

参考

引文

关键词

度量

更喜欢这个

下载

摘要: 与网络功能虚拟化(NFV)相关的主要挑战之一是服务生命周期的自动化管理。 在本文中,我们提出了一个... 查看更多

元数据

与网络功能虚拟化(NFV)相关的主要挑战之一是服务生命周期的自动化管理。在本文 中,我们提供了一个完整的基于软件的管理和编排(MANO)堆栈,它与OpenStack和 OpenDaylight控制器一起运行,并具有内置功能,可以自动化NFV服务生命周期的关键 阶段,即资源发现和匹配,服务映射,服务部署和监控。MANO堆栈由欧盟FP7项目T-NOVA实施,其组件作为开源软件发布。详细介绍了在T-NOVA范围内开发的服务映射和 服务部署解决方案。作为概念验证,我们评估虚拟化流量分类器网络功能的性能,

发表于: IEEE网络和服务管理交易 (卷:14, Issue:3, 2017年9月)

页数: 586 - 602 INSPEC登录号: 17153743

出版日期: 2017年7月31日 **DOI:** 10.1109 / TNSM.2017.2733620

ISSN信息: 出版商: IEEE

资助机构:

Citation Map

1. B. Han, V. Gopalakrishnan, L. Ji, S. Lee, "Network function virtualization: Challenges and opportunities for innovations", IEEE Commun. Mag., vol. 53, no. 2, pp. 90-97, Feb. 2015.

Full Text: PDF (208KB) Show Context View Article Google Scholar

2. R. Guerzoni, "Network functions virtualisation: An introduction benefits enablers challenges and call for action introductory white paper", Proc. SDN OpenFlow World Congr., pp. 1-16, Jun. 2012.

Show Context Google Scholar

3. J. Wu, Z. Zhang, Y. Hong, Y. Wen, "Cloud radio access network (C-RAN): A

更喜欢这个 云计算环境中的软件定义网络(SDN)和分布式拒绝服务(DDoS)攻击:调查,一些研究问题和挑战 IEEE通信调查和教程 发布时间: 2016年 事件支持视觉云计算利用软件定义网络 IEEE视频技术电路和系统交易 发布时间:2017年

查看更多



primer", IEEE Netw., vol. 29, no. 1, pp. 35-41, Jan./Feb. 2015. Show Context View Article Full Text: PDF (255KB) Google Scholar

4. C-RAN: The Road Towards Green RAN. White Paper. Version 2.5, Beijing, China, Oct. 2011.

Show Context Google Scholar

5. ETSI GS NFV 002 V1.2.1: Network Functions Virtualisation (NFV); Architectural Framework, Sophia Antipolis, France, Dec. 2014, [online] Available: http://www.etsi.org/deliver/etsigs/NFV/001099/002/01.02.0160/gsNFV002v010 201p.pdf.

Show Context Google Scholar

6. ETSI GS NFV 003 V1.2.1: Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV, Sophia Antipolis, France: ETSI Ind. Specification Group (ISG) NFV, Dec. 2014, [online] Available:

http://www.etsi.org/deliver/etsigs/NFV/001099/003/01.02.0160/gsNFV003v010 201p.pdf.

Show Context Google Scholar

- 7. S. Oechsner, A. Ripke, "Flexible support of VNF placement functions in OpenStack", Proc. 1st IEEE Conf. Netw. Softwarization (NetSoft), pp. 1-6, 2015. Show Context View Article Full Text: PDF (223KB) Google Scholar
- 8. K. Giotis, Y. Kryftis, V. Maglaris, "Policy-based orchestration of NFV services in software-defined networks", Proc. 1st IEEE Conf. Netw. Softwarization (NetSoft), pp. 1-5, 2015.

Show Context View Article Full Text: PDF (358KB) Google Scholar

9. M. Yoshida, W. Shen, T. Kawabata, K. Minato, W. Imajuku, "MORSA: A multiobjective resource scheduling algorithm for NFV infrastructure", Proc. 16th Asia -Pac. Netw. Oper. Manag. Symp. (APNOMS), pp. 1-6, 2014.

Show Context View Article Full Text: PDF (342KB) Google Scholar

- 10. Aug. 2017, [online] Available: http://www.t-nova.eu/. **Show Context**
- 11. Aug. 2017, [online] Available: https://github.com/T-NOVA. **Show Context**
- 12. G. Xilouris et al., "T-NOVA: A marketplace for virtualized network functions", Proc. Eur. Conf. Netw. Commun. (EuCNC), pp. 1-5, Jun. 2014. Show Context View Article Full Text: PDF (1053KB) Google Scholar
- 13. Aug. 2017, [online] Available:

https://osm.etsi.org/wikipub/index.php/Main_Page.

Show Context

- 14. Open Network Automation Platform (ONAP), Aug. 2017, [online] Available: https://www.onap.org. Google Scholar
- 15. Aug. 2017, [online] Available: http://openbaton.github.io/. **Show Context**
- 16. Aug. 2017, [online] Available: https://www.open-o.org/. **Show Context**
- 17. Aug. 2017, [online] Available:

http://about.att.com/content/dam/snrdocs/ecomp.pdf. **Show Context**

Aug. 2017, [online] Available: https://www.openstack.org/.
 Show Context

19. Aug. 2017, [online] Available: https://www.opendaylight.org/.

20. Aug. 2017, [online] Available: http://occi-wg.org/. Show Context

21. S. Bradner, J. McQuaid, Benchmarking methodology for network interconnect devices, Fremont, CA, USA:Internet Eng. Task Force, Mar. 1999. Show Context CrossRef Google Scholar

22. Aug. 2017, [online] Available:

https://www.kernel.org/doc/Documentation/vm/hugetlbpage.txt. Show Context

23. J. R. Quinlan, C4.5: Programs for Machine Learning, San Francisco, CA, USA:Morgan Kaufmann, 1993.

Show Context Google Scholar

24. V. Riccobene, M. J. McGrath, M.-A. Kourtis, G. Xilouris, H. Koumaras, "Automated generation of VNF deployment rules using infrastructure affinity characterization", *Proc. 2nd IEEE Conf. Netw. Softwarization (NetSoft)*, pp. 226-233, Jun. 2016.

Show Context View Article Full Text: PDF (3034KB) Google Scholar

- **25.** Heat Orchestration Template (HOT) Guide, 2015, [online] Available: http://docs.openstack.org/developer/heat/template_guide/hot_guide.html. Show Context Google Scholar
- **26.** H. Ballani, P. Costa, T. Karagiannis, A. Rowstron, "Towards predictable datacenter networks", *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. 242-253, 2011.

Show Context Access at ACM Google Scholar

- **27.** J. Lee et al., "CloudMirror: Application-aware bandwidth reservations in the cloud", *Proc. 5th USENIX Workshop Hot Topics Cloud Comput.*, pp. 1-6, 2013. Show Context Google Scholar
- **28.** C. Guo et al., "SecondNet: A data center network virtualization architecture with bandwidth guarantees", *Proc. ACM Conf. Emerg. Netw. Exp. Technol. (CoNEXT)*, pp. 1-12, 2010.

Show Context Access at ACM Google Scholar

29. A. Gember, R. Grandl, A. Anand, T. Benson, A. Akella, "Stratos: Virtual middleboxes as first-class entities", 2012.

Show Context Google Scholar

- 30. S. Oechsner, A. Ripke, "Flexible support of VNF placement functions in OpenStack", *Proc. 1st IEEE Conf. Netw. Softwarization (NetSoft)*, pp. 1-6, 2015. Show Context View Article Full Text: PDF (223KB) Google Scholar
- **31.** J. Lee et al., "CloudMirror: Application-aware bandwidth reservations in the cloud", *Proc. 5th USENIX Workshop Hot Topics Cloud Comput.*, pp. 1-6, 2013. Show Context Google Scholar
- **32.** A. Fischer, J. F. Botero, M. T. Beck, H. De Meer, X. Hesselbach, "Virtual network embedding: A survey", *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 1888-1906, 4th Quart. 2013.

Show Context View Article Full Text: PDF (5132KB) Google Scholar

33. R. Riggio, T. Rasheed, R. Narayanan, "Virtual network functions orchestration

in enterprise WLANs", *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manag. (IM)*, pp. 1220-1225, 2015.

Show Context View Article Full Text: PDF (1219KB) Google Scholar

34. R. Guerzoni et al., "A novel approach to virtual networks embedding for SDN management and orchestration", *Proc. Netw. Oper. Manag. Symp. (NOMS)*, pp. 1-7, 2014.

Show Context Google Scholar

35. S. Mehraghdam, M. Keller, H. Karl, "Specifying and placing chains of virtual network functions", *Proc. IEEE 3rd Int. Conf. Cloud Netw. (CloudNet)*, pp. 7-13, 2014.

Show Context Google Scholar

36. V. Abedifar, M. Eshghi, S. Mirjalili, S. M. Mirjalili, "An optimized virtual network mapping using PSO in cloud computing", *Proc. 21st Iran. Conf. Elect. Eng. (ICEE)*, pp. 1-6, 2013.

Show Context View Article Full Text: PDF (763KB) Google Scholar

37. R. Mijumbi et al., "Network function virtualization: State-of-the-art and research challenges", *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 236-262, 1st Quart. 2016.

Show Context Google Scholar

- **38.** D. Dietrich, A. Abujoda, P. Papadimitriou, "Network service embedding across multiple providers with nestor", *Proc. IFIP Netw.*, pp. 1-9, May 2015. Show Context View Article Full Text: PDF (234KB) Google Scholar
- **39.** D. Dietrich, A. Rizk, P. Papadimitriou, "Multi-provider virtual network embedding with limited information disclosure", *IEEE Trans. Netw. Service Manag.*, vol. 12, no. 2, pp. 188-201, Jun. 2015.

Show Context View Article Full Text: PDF (1807KB) Google Scholar

40. *Virtual Network Mapping Problem Instances*, 2015, [online] Available: https://www.ac.tuwien.ac.at/research/problem-instances/.

Show Context Google Scholar

41. J. Inführ, G. R. Raidl, Introducing the Virtual Network Mapping Problem With Delay Routing and Location Constraints, Heidelberg, Germany:Springer, pp. 105-117, 2011.

Google Scholar

42. *Monasca Monitoring-As-a Service for OpenStack,* Aug. 2017, [online] Available: https://wiki.openstack.org/wiki/Monasca.

Show Context Google Scholar

43. *Gnocchi: TDBaaS for OpenStack,* Aug. 2017, [online] Available: https://wiki.openstack.org/wiki/Gnocchi.

Show Context Google Scholar

44. *OPNFV Doctor Project (Fault Management)*, Aug. 2017, [online] Available: https://wiki.opnfv.org/doctor.

Show Context Google Scholar

45. *OPNFV Prediction Project (Data Collection for Failure Prediction),* Aug. 2017, [online] Available: https://wiki.opnfv.org/prediction.

Show Context Google Scholar

46. *OpenStack Telemetry API v2 (CURRENT),* Aug. 2017, [online] Available: http://developer.OpenStack.org/api-ref-telemetry-v2.html.

Show Context Google Scholar

47. *Collectd the System Statistics Collection Daemon*, Aug. 2017, [online] Available: https://collectd.org/.

Show Context Google Scholar

48. *InfluxDB:* An Open-Source Distributed Time Series Database With No External Dependencies, Aug. 2017, [online] Available: http://influxdb.com/.

Show Context Google Scholar

49. Grafana: An Open Source Feature Rich Metrics Dashboard and Graph Editor for Graphite InfluxDB & OpenTSDB, Aug. 2017, [online] Available: http://grafana.org/.

Show Context Google Scholar

- **50.** Aug. 2017, [online] Available: https://github.com/httperf/httperf. Show Context
- **51.** Aug. 2017, [online] Available: https://github.com/T-NOVA/vim-monitoring. Show Context
- **52.** Aug. 2017, [online] Available: https://wiki.opnfv.org/yardstick. Show Context
- **53.** *PCI Special Interest Group*, Aug. 2017, [online] Available: http://www.pcisig.com/home.

Show Context Google Scholar

54. L. Deri, M. Martinelli, T. Bujlow, A. Cardigliano, "nDPI: Open-source high-speed deep packet inspection", *Proc. Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, pp. 617-622, Aug. 2014.

Show Context Google Scholar

55. A. Cardigliano, L. Deri, J. Gasparakis, F. Fusco, "vPF_RING: Towards wirespeed network monitoring using virtual machines", *Proc. IMC*, pp. 533-548, Nov. 2011.

Show Context Access at ACM Google Scholar

- **56.** Aug. 2017, [online] Available: https://www.docker.com/. Show Context
- **57.** *DPDK: Data Plane Development Kit*, Aug. 2015, [online] Available: http://dpdk.org/.

Show Context Google Scholar

58. K. Salah, A. Qahtan, "Implementation and experimental performance evaluation of a hybrid interrupt-handling scheme", *Comput. Commun.*, vol. 32, no. 1, pp. 179-188, Jan. 2009.

Show Context CrossRef Google Scholar

59. *PktGen Pktgen Version 2.7.7 Using DPDK-1.7.1*, Aug. 2017, [online] Available: https://github.com/Pktgen/Pktgen-DPDK/.

Show Context Google Scholar

广告





网络管理和服务供应领域的许多最新发展主要集中在网络功能虚拟化(NFV)上。 NFV为网络功能的实施和管理引入了一种新方法。特别是,NFV的主要关注点是将 物理设备与运行在它们上面的网络功能分离开来;换句话说,从硬件到基于软件的网络功能的转变[1],[2]。这种模式得到了电信服务提供商(TSP)的极大关注,因为它有可能显着降低运营支出(OPEX)和资本支出(CAPEX)[3],[4]以及通过新颖的增值软件网络产品拓宽Telco的服务组合。

虚拟化网络功能(VNF)形式的软件设备在标准高容量(SHV)服务器上运行,该服务器整合了公共基础设施上各种网络设备的操作和管理。VNF提供了在网络中的不同位置动态启动,重新配置甚至重新分配的能力,而无需安装新硬件。NFV旨在提供更加开放和可扩展的网络环境,新网络服务的部署变得更加容易和快捷。

然而,为了实现这些期望,必须实现各种基本发展,其中许多仍处于实施和不断发展的状态。NFV的部署及其在电信级系统中的应用仍然包含[5]中确定的各种未决问题。NFV将对当前的网络管理系统提出重大挑战,并且需要比当前部署的系统更高级别的复杂性。这种转变的关键特征之一是管理层,它必须能够支持启用NFV的系统的独特功能。类似的工作已经解决了管理层,如管理和编排系统(MANO)[6]。MANO的主要目标是协调NFV基础设施(NFVI)资源并将其有效映射到各种VNF。反过来,VNF可以互连成链,以实现更复杂的网络服务(NS)。NFV NS可被视为传统电信连接服务的演进,因为它通过动态插入网络流量路径的VNF链进行扩充。

MANO系统要解决的第一个关键挑战是自动部署。网络功能不再绑定到物理机器;相反,它们驻留在共享资源环境中。因此,在部署(VNF"布局")决策时会出现各种考虑因素,例如性能优化,拓扑,弹性等[7]。大多数当前的MANO方法都基于对托管VNF工作负载的节点使用预先计划的配置[8]。这可以限制VNF性能,因为主机节点的特征不一定匹配工作负载的特性(例如,数据平面,控制平面,重处理等)。随着托管计算基础架构变得更加异构,各种附加功能支持硬件加速(即PCIe,GPU),可以支持各种系统增强和功能,这大大增加了VNF放置决策的复杂性。部署时自动放置智能会增加MANO编排服务的复杂性。但是,它可以实现更优化的资源分配和更高的VNF性能。在[9]中报告了许多MANO VNF调度策略的方法。然而,他们通常采用抽象的方法在NFVI中放置VNF,以实现特定目标,例如减少数据中心内部流量[9]。这些研究未必解决将资源映射到特定特征和单个VNF工作负载需求的特定问题。

在本文中,我们提出了一种切实可行的方法来满足这一要求,共同解决资源发现和服务映射的挑战,同时确定在部署时分配给VNF的资源数量和类型。在这方面,我们生成一个原型模型,该模型表示与特定性能水平相关的资源分配。此外,还生成了特定规则,Orchestrator可以利用这些规则在诸如OpenStack之类的云环境中向VNF部署请求添加智能。

此外,服务供应和业务流程的关键要素是监控。监控不仅涵盖NFVI资源,还涵盖VNF服务的状态。全面监控还有助于服务映射和计费/收费。它还可以实现故障恢复和可用性,这也是关键问题,因为VNF的故障可能会影响整个网络服务。出于所有上述原因,为NFV开发集成监控框架,从物理基础架构资源收集指标以及特定虚拟化服务指标对于任何NFV基础架构都是至关重要的。有效的NFV监测框架应使所有管理实体全面了解已部署服务的状态和性能以及基础设施,以便后者能够及时做出适当的决策。在此背景下,我们还提出了一个集成的NFV监控解决方案,作为MANO堆栈的关键组件。

总之,本文提出了一种新的NFO基础设施MANO解决方案,目前由EC FP7项目T-NOVA [10]开发,并作为开源贡献发布[11]。MANO堆栈围绕一个新的NFV Orchestrator平台("TeNOR")构建,其当前实现的模块支持管理 - 以自动方式 - 管理NFV服务生命周期的四个关键阶段,即资源发现,服务映射,服务部署和监控。

在本文的其余部分安排如下。我们最初描述了整个T-NOVA系统架构,它涵盖了NFV系统的所有层,也超出了MANO范围。随后,我们提供MANO堆栈中每个组件的详细描述和评估。最后,我们提出了概念验证部署和虚拟化流量分类器VNF的评估,旨在展示加速技术在启用NFV的环境中的优势。

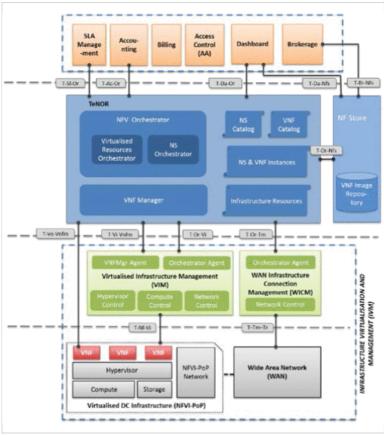
第二节

T-NOVA整体观点

A. T-NOVA系统架构

T-NOVA系统架构[12]从通用ETSI NFV ISG架构模型[5]继承了其大部分概念,并使用特定的附加功能对其进行了扩展。T-NOVA架构包含四个关键架构层(如图1所示):

- 所述NFV基础设施(NFVI)层由在其上的网络服务被部署在物理和虚拟节点 (高容量服务器,虚拟机(VM),存储系统,交换机,路由器等)的;
- 该NFVI基础设施(NFVI)管理层包括基础设施管理实体:虚拟化基础架构管理(VIM)和广域网基础设施连接管理(WICM)。T-NOVA采用OpenStack [18]云操作系统来控制计算和数据中心资产,OpenDaylight [13]用于控制网络基础设施(大部分是基于SDN的);
- 该业务流程层是基于T-NOVA男高音Orchestrator中还包括一个"网络功能来存储",这是所有出版VNFs库。Orchestrator以及NFVI管理层包括T-NOVAMANO堆栈;
- 最后, Marketplace层包含所有面向客户的界面和模块,这些界面和模块有助于多角色参与并实现与业务相关的功能。



图。1。 T-NOVA架构。

T-NOVA在NFV框架中引入了市场的概念。Marketplace的目的是促进VNF服务产品的推广,促进商业活动以及与T-NOVA系统交互的各种业务利益相关者之间的无缝互动。T-NOVA市场为底层MANO堆栈提供直观的界面,允许Telcos和VNF开发人员发布和描述服务产品,以及使客户能够浏览和选择服务,部署和管理/监控它们。有关T-NOVA市场的更深入描述可以在[12]中找到。

T-NOVA Orchestrator("TeNOR")是T-NOVA架构框架的核心组件。其主要目标是通过分布式和虚拟化网络/ IT基础架构解决NS和VNF的生命周期管理操作。

目前,ETSI NFV MANO Architecture [5]的各种实现趋势。最值得注意的是OSM [13]和ONAP o。

第一个是开源项目,为ETSI的NFV ISG标准化下的MANO提供了参考架构的实际实现。与TeNOR相关,两个平台都提供端到端服务,但与架构方面存在一些差异 - 例如,OSM在Resource Orchestrator和Service Orchestrator之间有明确的架构分割。将它们分开的另一个方面是它们的起源;T-NOVA使用通用接口将其组件从头开始实现为微服务。这使得能够在不损失功能的情况下自由地开发或替换任何TeNOR组件,这在OSM中是不可能的。此外,TeNOR还有一个特定的模块,Gatekeeper,它为组件之间的接口提供了通用的安全机制,而OSM依赖于每个组件的安全功能。最后,这两个平台之间的最大区别是TeNOR还在VNF和NS数据模型中解决了NFV的业务问题,而OSM仅关注NS的运营。考虑到这一点,OSM GUI和T-NOVA

Marketplace也在图形界面中共享一些功能,例如,他们使用类似的方法来处理板载 VNF并组成NSD。

后者是Open-O [16]和ECOMP ¹ [17]的代码库合并的后代。, 更加注重NFV和SDN 之间的差距,提供端到端服务编排和自动化。从T-NOVA的角度来看,集成平台可被 视为实现虚拟网络功能的实时,策略驱动的软件自动化的方法,这将使软件,网络, IT和云提供商和开发人员能够快速创建新的服务。为了比较ONAP想要的那么广泛的 东西,我们需要将ONAP分解为两个主要框架。一个是设计时框架,主要由ECOMP 支持,用于网络服务,策略和VNF的设计和组合。与T-NOVA相比,这是由T-NOVA 市场处理的,它超越了业务流程讨论。ONAP的另一个框架是运行时执行框架,它基 于Open-O,实现像TeNOR一样的ONAP的所有编排功能。在此背景下,Open-O和 ONAP经过优化,可提供开放式管理平台,用于基于虚拟化网络和基础架构资源以及 软件应用定义,运营和管理各种产品和服务,而T-NOVA主要关注VNFaaS环境和解 决VNFaaS特定的要求和挑战。因此,TeNOR不会在基本生命周期操作之外实现任 何类型的Service Orchestration。此外,TeNOR仅支持基于Openstack的VIM,但也 可以轻松扩展以支持其他VIM技术。基于虚拟化网络和基础架构资源以及软件应用 程序运营和管理各种产品和服务,而T-NOVA主要关注VNFaaS环境并解决VNFaaS 特定的要求和挑战。因此,TeNOR不会在基本生命周期操作之外实现任何类型的 Service Orchestration。此外,TeNOR仅支持基于Openstack的VIM,但也可以轻松 扩展以支持其他VIM技术。基于虚拟化网络和基础架构资源以及软件应用程序运营 和管理各种产品和服务,而T-NOVA主要关注VNFaaS环境并解决VNFaaS特定的要 求和挑战。因此,TeNOR不会在基本生命周期操作之外实现任何类型的Service Orchestration。此外,TeNOR仅支持基于Openstack的VIM,但也可以轻松扩展以 支持其他VIM技术。

其他方法包括OpenBaton [15]这是Fraunhofer Fokus和TUB之间为实施和符合ETSI标准的MANO解决方案的协同作用的结果。与TeNOR相比,两种平台都具有以下共同特征:自动扩展,虚拟网络服务的多PoP部署,VNF和服务存储目录以及功能仪表板。然而,在T-NOVA中,协调器仪表板功能更丰富,并且包括会计,计费和经纪等丰富的市场。此外,T-NOVA Orchestrator支持POP之间的传输网络管理,并支持高级服务功能链接功能(通过NetFloc)。Open Baton默认SDN管理到不受协调器本身影响的外部模块。打开Baton的外部模块Network-Slicing-Engine执行网络切片并确保QoS。

由于每种解决方案的成熟度水平不同,因此无法与上述业务流程解决方案进行全面比较。但是,TeNOR的可区分特征包括微服务架构,多弹出/多管理域支持以及对模块化服务映射和放置算法的支持。

TeNOR与市场进行交互,市场是负责建立T-NOVA业务和运营管理的外部终端。除了市场,TeNOR还与VIM连接,用于管理数据中心网络/IT基础架构资源,以及用于WAN元件连接管理的WAN基础设施连接管理(WICM)。最后,TeNOR与VNF本身进行交互,以提供适当的生命周期管理。

为了支持NFV生命周期管理操作,定义了以下类别,其中大部分都继承自ETSI NFV ISG建议:

- NS目录:代表所有已加入的NS(已部署的NS)的存储库,以支持NS生命周期管理;
- NS描述符(NSD): 包含服务描述,包括服务水平协议,部署风格,对虚拟链路(VL)和组成VNF的引用,虚拟网络功能转发图(VNFFG);
- 虚链路描述符(VLD): 包含组成服务的虚拟网络链路的描述(互连VNF);
- VNF转发图描述符(VNFFG):包含NS组成的VNF,以及它们在网络连接方面的部署;
- VNF目录: 代表所有已加入的VNF的存储库, 以支持其生命周期管理:
- VNF描述符(VNFD):包含VNF描述,包括基于虚拟网络功能组件 (VNFC)的内部分解,部署风格和对虚拟链路(VLD)的引用;
- 位于NFVI层的VM的软件映像;
- NS和VNF实例:表示所有实例化的NS和VNF的存储库,可以在生命周期管理操作期间更新/发布;

 基础架构资源:表示VIM跨运营商的基础架构域抽象的可用/保留/分配的 NFVI资源的存储库。此外,它还包括WAN段中可用/预留/分配的资源。

B. TeNOR - T-NOVA NFV Orchestrator

网络服务生命周期供应由T-NOVA编排平台TeNOR管理,其NFV Orchestrator功能模块分为两个主要核心子模块: Network Service Orchestrator(NSO)和Virtualized Resource Orchestrator(VRO),如图2所示。

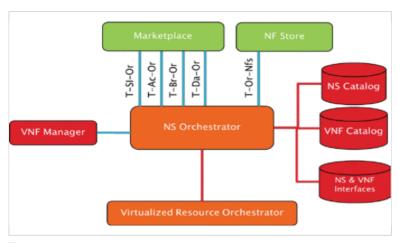


图2。 TeNOR NS Orchestrator (内部和外部)交互。

国家统计局的职责是管理NS生命周期和相关程序,包括:

NSes 和VNF on-boarding: 网络服务部署模板的管理,也称为NS描述符和VNF包,以及NSs实例拓扑(例如,创建,更新,查询,删除VNF转发图)。NS的登记包括在NS目录中注册,因此确保存储所有模板(NSD)。

NS实例化:根据在入站的NS和VNF部署模板中捕获的触发器和操作,触发NS和VNF实例的实例化。此外,管理VNF实例化,与虚拟化网络功能管理器(VNFM)协调以及验证来自VNFM的NFVI资源请求,因为它们可能影响NS,例如,缩放过程。

NS更新: 支持不同复杂度的NS配置更改,例如更改VNF间连接或组成VNF;

NS 监视和测量NS性能以及每个服务实例的获取度量的相关性。数据从IVM层(与互连网络功能的虚拟网络链路相关的性能指标)和VNFM(与VNF相关的聚合性能指标;

NS扩展:根据每个实例和每个服务的自动扩展策略增加或减少NS的容量。NS扩展可以暗示特定VNF容量的增加/减少,新/旧VNF实例的创建/终止和/或网络功能之间的连接链路数量的增加/减少;

NS終止:通过删除关联的VNF和关联的连接链接以及虚拟化基础架构资源来释放特定的NS实例。

VRO负责管理底层物理计算和网络资源。在此上下文中,VRO与以下各项接口: (i) 管理每个NFVI的存在点(NFVI-PoP)中部署VNF的资源的虚拟化基础架构管 理器;(ii) 管理WAN资源分配的WAN基础设施管理器(WIM)。VRO还通过基础 架构存储库跟踪所有底层基础架构。在NS部署请求的情况下,VRO负责根据每个NS 施加的要求分配所需的资源。

从外部角度来看,国家统计办公室与市场的关键业务和业务管理目的如下:

- 关于NSes (通过T-Da-Or接口) 交换供应信息 (例如,请求,修改/更新,确认);
- 向Orchestrator提供有关每个NS实例SLA协议的信息。反过来,Orchestrator将与SLA相关的指标发送到Marketplace (通过T-Sl-Or界面);
- 向VNF和NS交付关于VNF和NS的使用会计信息(通过T-Ac-Or界面);

• 向Orchestrator提供有关NS组成的信息。Orchestrator在可用的VNF上发送市场信息(通过T-Br-Or接口)。

在国内,国家统计局有以下沟通要点:

- NS目录: 收集有关NS (NSD) 的信息,包括组成VNF,互连网络链路 (VLD) 和网络拓扑信息 (VNFFGD);
- VNF目录: 在入职程序中存储虚拟网络功能描述符文档;
- NS和VNF实例:存储有关NS实例状态的信息;
- VRO:在数据中心范围内(例如,计算,存储和网络)和/或在传输网络段上交换与虚拟化资源和/或连接相关的管理动作;
- VNFM: 交换与VNF相关的生命周期管理操作。

接下来的部分将详细讨论T-NOVA MANO堆栈的特定组件,这些组件可以管理NFV 服务生命周期的关键阶段。

第三节

资源发现,选择,匹配

IT云计算环境旨在最大限度地提高计算资源的利用率(效率),并采用高度自动化的操作来安排和管理工作负载。为了实现这些目标,需要显着的灵活性,这是通过抽象底层基础设施环境来实现的。通过隐藏底层计算,存储和网络资源的异构性,可以更轻松地放置或迁移工作负载。因此,应用程序性能通常在尽力而为的基础上提供。

相比之下,Telco云应用程序具有一组独特的特性,例如低延迟,高数据包吞吐量,低抖动,网络设置和拆卸限制的要求。此外,这些工作负载可能具有SLA中定义的关键性能指标(KPI),这些指标必须在运营环境中得到满足。托管网络工作负载的底层基础架构在帮助实现这些KPI方面发挥着重要作用。在NFV环境中尤其如此,其目标是在虚拟化基础架构环境中实现近线速率性能。

这些要求导致了许多必须在Telco云环境中解决的关键挑战。首先,云中可用的资源 必须是可发现的,并且其相关描述存储在可访问的存储库中。其次,有必要理解和映 射工作负载的特征(例如,数据平面工作负载与控制平面工作负载)及其与资源类型 的关系(例如,Open vSwitch(OVS),SR-IOV,支持DPDK的网络接口卡)和分 配的资源数量(例如,vCPU的数量,RAM的数量等)。这对于防止过度分配对工作 负载性能没有实际好处的资源是必要的。在分配可用多个选项的资源时(例如、VM 的网络连接类型),重要的是分配资源,为工作负载性能提供可量化的好处。这使得 Telco服务提供商可以通过确保将工作负载部署在可从特定可用资源中受益的平台上 来更好地利用其基础架构,并改善工作负载整合。最后,平台配置也是一个重要的考 虑因素。例如,使用PCIe感知非统一内存访问(NUMA)固定部署具有高数据包吞 吐量特性的工作负载时,可能会对VNF性能产生重大影响。NUMA是一种多处理存 储器设计,为每个处理器提供单独的存储器,使处理器能够绑定到存储器资源并更快 地访问它。这使得Telco服务提供商可以通过确保将工作负载部署在可从特定可用资 源中受益的平台上来更好地利用其基础架构,并改善工作负载整合。最后,平台配置 也是一个重要的考虑因素。例如,使用PCIe感知非统一内存访问(NUMA)固定部 署具有高数据包吞吐量特性的工作负载时,可能会对VNF性能产生重大影响。 NUMA是一种多处理存储器设计,为每个处理器提供单独的存储器,使处理器能够 绑定到存储器资源并更快地访问它。这使得Telco服务提供商可以通过确保将工作负 载部署在可从特定可用资源中受益的平台上来更好地利用其基础架构,并改善工作负 载整合。最后,平台配置也是一个重要的考虑因素。例如,使用PCIe感知非统一内 存访问(NUMA)固定部署具有高数据包吞吐量特性的工作负载时,可能会对VNF 性能产生重大影响。NUMA是一种多处理存储器设计,为每个处理器提供单独的存 储器,使处理器能够绑定到存储器资源并更快地访问它。最后,平台配置也是一个重 要的考虑因素。例如,使用PCIe感知非统一内存访问(NUMA)固定部署具有高数 据包吞吐量特性的工作负载时,可能会对VNF性能产生重大影响。NUMA是一种多 处理存储器设计,为每个处理器提供单独的存储器,使处理器能够绑定到存储器资源 并更快地访问它。最后,平台配置也是一个重要的考虑因素。例如,使用PCIe感知 非统一内存访问(NUMA)固定部署具有高数据包吞吐量特性的工作负载时,可能 会对VNF性能产生重大影响。NUMA是一种多处理存储器设计,为每个处理器提供 单独的存储器,使处理器能够绑定到存储器资源并更快地访问它。

A. NFVI资源发现和意识

为了使VNF实现与对应的专用硬件实现接近或类似的性能,在NFVI环境中在硬件和软件方面适当地利用平台特征是至关重要的。T-NOVA通过开发基础设施储存库子系统来实现这一目标,该子系统是该项目开发的TeNOR的一部分。存储库子系统的设计解决了从NFVI / VIM层内的源(即T-NOVA系统内的云基础架构和数据中心网络环境)吸收基础架构相关信息的挑战。该子系统包括许多关键元素,包括数据模型,资源信息存储库和信息存储库的访问机制。

增强的平台感知(EPA)支持NFVI中可用资源与VNF的细粒度匹配,以便更好地优化所提供的网络服务。要启用EPA,必须识别和公开可用的平台功能。如图3所示包含在NFVI中的计算节点上运行的EPA代理的子系统收集并报告所需的平台信息。当EPA代理提供平台更新时,通知消息将通过特定的侦听器服务发送到控制器。在收到来自EPA侦听器服务的消息后,由EPA代理发送到存储目录的数据文件由控制器处理并用于更新中央存储库数据库。侦听器服务还用于拦截和管道OpenStack中与基础架构相关的消息,并通过控制器更新存储库数据库。基础结构存储库数据库实现为图形数据库。数据库以存储为链接的节点之间的语义相关连接的形式提供层次关系。图形的使用方便地映射到NFVI内的计算,存储和网络元素的分层结构。开放式云计算接口(OCCI)[20]兼容中间件层为存储在图数据库中的资源信息提供公共接口。此外,中间件API为OpenDaylight通过其REST API提供的物理网络信息提供了一个抽象的单一访问点。中间件层还具有图形数据库,用于在TeNOR的控制下为每个NFV-PoP存储OpenStack服务的端点信息。

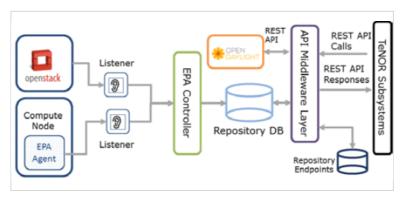


图3。 T-NOVA资源库子系统。

B.使用基础设施资源表征VNF亲和力

对于任何VNF类型的工作负载,有必要构建VNF对计算资源分配和平台特定功能的 亲和力的图片。将电信公司的业绩方面置于语境中也很重要,因为它们本质上是多方面的。确定了数据包大小和网络连接类型等关键影响因素以及确定性能的关键指标,如数据包吞吐量等。调查了各种部署配置,包括关键网络技术(虚拟化数据包交换和数据包加速),存储和计算(核心钉扎,NUMA钉扎,异构计算资源BIOS配置等)。

例如,在利用不同类型的资源时,在虚拟化流量分类器部署上研究了噪声邻居的影响。嘈杂的邻居是云计算环境中常用的术语,用于描述共同租户工作负载,这会通过大量利用特定资源(如网络带宽,CPU,内存等)对其他工作负载的性能产生负面影响。因此,噪声很大邻居效应会导致其他工作负载(如共享云基础架构的VNF)遭受不可预测的性能影响。为了研究不同配置对虚拟化流量分类器VNF性能的影响,两个测试用例都遵循RFC 2544 [21]提出的标准基准测试方法。测量网络吞吐量。测试用例使用七种不同的数据包大小(64,128,256,512,1024,1280和1518字节),测试持续时间为60秒,用于吞吐量测量。在第一个测试案例中,部署不包括噪声邻居,而在第二个测试案例中,吞吐量是在存在噪声邻居的情况下测量的,这些邻居会为托管VNF的节点的CPU增加额外开销。图4显示了使用1280字节的数据包大小获得的吞吐量结果。在这种情况下,3.5Gbps的吞吐率对应于每秒~270万帧。

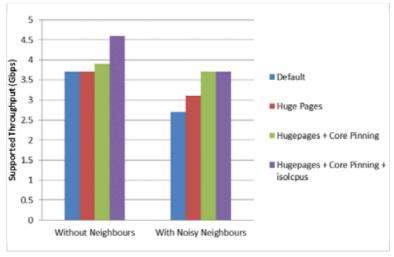


图4。 具有不同计算配置的VNF的数据包吞吐量性能。

在第一个测试用例中,结果显示大页面(内存,例如1GB)[22]对性能没有可测量的影响,而核心固定提高了性能,特别是与"isolcpus"Grub选项一起使用时,性能提高了24%。isolcpus选项提供与一般内核对称多处理(SMP)平衡和调度程序算法的CPU隔离。这具有将核心与用户空间任务隔离的效果。在第二个测试用例(带有噪声的邻居)中,默认配置的基线性能为2.7Gbps,低于第一个测试用例场景。吞吐量降低的原因是CPU处理来自附加VM的请求的额外计算开销。在此测试案例中,巨大页面对性能的影响清晰可见(吞吐量提高了14%)。

C.发现VNF的最佳资源分配

如前所述,一旦基础设施资源可被发现并已被表征,最后一个要素就是工作负载表 征。表征过程的目标是匹配适当的资源类型和数量,以便为给定部署实现所需的 KPI。当前在云计算环境中部署诸如VNF之类的工作负载的方法基于静态预定义的资 源分配。这些分配通常以抽象方式定义,即,资源请求通常简单地定义为vCPU,存 储器,存储和网络连接的数量,而不参考这些资源类型的特定特征或属性。这些特性 可能包括通过CPU支持特定指令集(如AES-NI)或网卡中的智能卸载支持等。这会 导致资源过度分配,其次与每种类型的资源可用资源类型不匹配。工作量。为了解决 这个问题,开发了一个VNF工作负载表征框架,该框架可用于查询定义的一组配置 变量和相关值范围的所有潜在配置排列。该框架收集每个排列的度量数据,并使用机 器学习环境处理数据以生成一组部署规则。工作负载表征和部署规则生成的通用过程 如下所示 这导致资源的过度分配,其次与每种类型的工作负载有效可用的资源类型 不匹配。为了解决这个问题,开发了一个VNF工作负载表征框架,该框架可用于查 询定义的一组配置变量和相关值范围的所有潜在配置排列。该框架收集每个排列的度 量数据,并使用机器学习环境处理数据以生成一组部署规则。工作负载表征和部署规 则生成的通用过程如下所示 这导致资源的过度分配,其次与每种类型的工作负载有 效可用的资源类型不匹配。为了解决这个问题,开发了一个VNF工作负载表征框 架,该框架可用于查询定义的一组配置变量和相关值范围的所有潜在配置排列。该框 架收集每个排列的度量数据,并使用机器学习环境处理数据以生成一组部署规则。工 作负载表征和部署规则生成的通用过程如下所示 开发了VNF工作负载表征框架,该 框架可用于查询定义的一组配置变量和相关值范围的所有潜在配置排列。该框架收集 每个排列的度量数据,并使用机器学习环境处理数据以生成一组部署规则。工作负载 表征和部署规则生成的通用过程如下所示 开发了VNF工作负载表征框架,该框架可 用于查询定义的一组配置变量和相关值范围的所有潜在配置排列。该框架收集每个排 列的度量数据,并使用机器学习环境处理数据以生成一组部署规则。工作负载表征和 部署规则生成的通用过程如下所示图5。为了支持该过程,开发了一个框架,该框架 将定义感兴趣的资源和数量范围的配置作为其输入,例如,从1到10个vCPU,从1到 8GB的RAM等。配置文件然后转换配置选项。到目标虚拟化环境的一组部署模板, 例如OpenStack。使用部署模板自动部署测试中的工作负载。针对每种配置自动执行 测试用例,例如,网络吞吐量的测量(RFC2544)并且收集和存储相应的遥测数 据。测试案例侧重于VNF绩效指标的量化。当部署了所有部署模板后,收集的数据 将被聚合并由机器学习算法(例如,C4.5)使用[23])生成一个模型,该模型将VNF 的性能指标(例如,吞吐量)与不同的资源分配相关联。然后可以以部署规则的形式 表达这些关系(例如,通过分配vCPU = 4, RAM = 8GB, 网络连接= SR-IOV等, 可以实现吞吐量=5Gbps)。关于方法和应用的其他细节可以在[24]中找到。

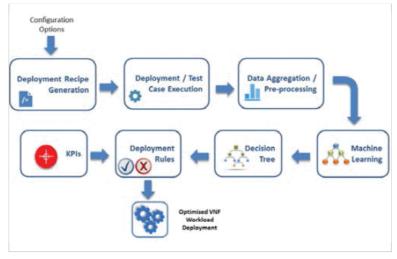


图5。 VNF部署规则生成过程。

可以利用框架自动生成的规则集来生成部署配方(例如,针对OpenStack的Heat Orchestration Templates(HOT)[25],其中参数是根据正在部署的工作负载或服务的目标KPI导出的。图6显示了用于实现Heat模板自动生成的HOT语法。根据工作负载目标KPI,创建了具有通用参数的Heat模板,该模板在运行时被替换为通过应用框架生成的规则计算的值。

```
port:
type: OS::Neutron::Port
properties:
 network: { get_param: network }
 binding:vnic_type: #vnic_type
# allowed values: [normal, direct]
 fixed ips:
      subnet: { get param: subnet }
server_flavor:
type: OS::Nova::Flavor
properties:
 disk: #disk
                 # allowed values: [1024, 2048, ...]
                # allowed values: [1024, 2048, ...]
 ram: #ram
                    # allowed values: [1, 2, 3, ...]
 vcpus: #vcpus
  extra_specs: {
      "hw:cpu_policy": "#core_pinning_enabled",
          # allowed values: [shared, dedicated]
      "hw:cpu_threads_policy": "#core_pinning_mode",
          # allowed values: [isolate, prefer]
      "hw:mem_page_size": "#hugepages"
          # allowed values: [small, large]
                 }
```

图6。 从提交给Heat之前操作的通用热模板中提取代码。

总体而言,VNF部署的资源发现,工作负载和技术特性,自动部署规则和配方生成在T-NOVA系统中提供了独特的功能。

第四节

服务映射

术语服务映射(SM)是指将虚拟化基础设施资源动态映射到NSes的问题,尊重由以下因素构成的约束: (i) 网络基础设施资源的当前可用性, (ii) 服务所需资源的类型和数量待制图和(iii) SLA特定需求。在过去的这些年中,这个问题引起了越来越多的关注。文献中描述的早期研究方法,如[26] - [27] [28],已经提出了云平台实

现,允许NSes任意集成到VM中,而不考虑服务链的功能。其他研究工作[29],[30] 为数据中心(DC)网络内的NS链映射提供启发式方法,旨在最大化KPI,如机架间流量。文献中的几种方法致力于服务映射建模的主题。其中,[31]开发了一种基于图形的技术,称为租户应用图(TAG),以准确捕获正在部署的虚拟机的带宽需求,避免过度配置与先前方法相关的低效率。同样的工作提出了针对SM的最小割和背包算法,其驱动原理是最大化由边缘链接的VM的共址以及严格的链路带宽要求。

在多个DC上部署NSes,即在DC间网络上映射NS链最终能够实现NS的广域部署。这种类型的问题通常与虚拟网络嵌入问题相比[32],这引发了一些服务映射公式。然而,由于不同的NS类型,中间盒运营商所采取的策略以及由一些NS引起的流量变化,所提出的各种嵌入算法不能直接应用于服务链。在高级服务映射算法中,[33]提出了基于成本度量优化(针对节点/链路负载平衡)和最短路径计算的NS节点和虚拟链路映射。Guerzoni 等。[34]提出了一种基于基础设施和服务的无向图建模的整数线性规划(ILP)算法。有趣的是,作者实现并讨论了所谓的"前瞻"属性,即同时映射NS束。类似地,在[35]中提出了混合ILP策略,其旨在最小化路径延迟,使用的资源的数量和最大化剩余的节点/链路资源可用性。还提出了软计算技术,例如[36]中概述的方法。其他类似于上述方法的SM方法可以在[8],[32]和[37]中找到。

T-NOVA中解决的服务映射问题集中在基于组成网络基础设施(NI)的互连DC的组成每个NS请求的VNF的优化技术的分配。对NS的请求动态地到达系统:它们不是事先已知的,并且SM算法必须为每个请求识别最佳分配,或者根据需要丢弃它。对于任何在线优化问题,在求解每个映射请求时优化的目标函数必须隐式地建模真实的总体目标函数,即最大化所接受的请求的数量。

在本文中,我们提出了一种基于ILP的SM问题方法,特别是将VNF分配给NFV-PoP。在这方面,拟议的ILP的目标是最大化已接受的NS请求。特定的SM方法已经实施并集成到T-NOVA协调器中。我们的SM方法不同于[38]和[39]中提出的基于ILP的方法。具体而言,在我们的系统中,服务映射模块完全由基础设施提供商执行,并且优化目标根据其策略(即,请求接受的最大化以及因此收入)而定制。在[38]和[39],向PoP分配VNF由经纪人执行,主要目的是最大限度地减少客户的支出。T-NOVA研究了三种SM方法,两种基于ILP,一种基于强化学习。本文将重点介绍目前完全集成在T-NOVA系统中的ILP方法。这种选择主要取决于基于ILP模型的解决方案比基于定制启发式的解决方案更灵活的事实,特别是如果T-NOVA将在未来的项目中进一步阐述。我们的目标有三个:(a)让模型能够尽可能多地考虑资源和种类的资源限制,(b)考虑多个目标函数,以及(c)提供可扩展的解决方案,即使非使用商业解算器。

A.问题建模

T-NOVA中解决的SM问题在图7(a)中概述,其示出了由两个VNF组成的NS,由四个互连的DC组成的NI及其对应的图。图7(b)显示了相应SM问题的可能解决方案。VNF $_1$ 已分配给DC $_1$,VNF $_2$ 已分配给DC $_4$,并且连接VNF $_1$ 和VNF $_2$ 的电弧已分配到从DC $_1$ 到DC $_4$ 到DC $_3$ 的蓝色路径。

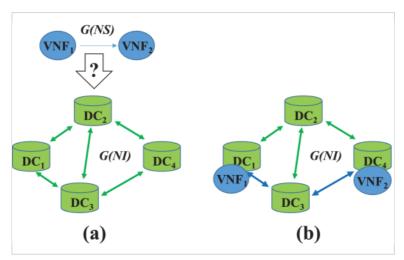


图7。 SM问题(a)及其解决方案(b)的说明性示例。

如图7所示,每个NS被建模为有向图G (N小号) = 其中集盆中的每个顶点V,说 H,代表一个VNF,每个弧 一个说 (h , k表示连接两个VNF的链接。类似地,NI被建模为有向图G (N一世) = (\mathfrak{p} 中群 \mathfrak{p}) 据都说 p , 在集合中 V一代表一个PoP,每个弧都在 一一 说 (p , q表示两个PoP之间的网络连接。每个NS,一套P 路径关联连接所有VNF对(每个路径 $\pi \in P$ 是图中的一系列弧 G (N小号

基于上述建模,以下各节概述了在T-NOVA系统中实施的基于ILP的映射方法。在此之前,下一节将简要介绍根据T-NOVA架构管理SM模块功能的参考架构和顺序图。

B.参考架构和SM序列图

图8显示了管理SM操作的基本顺序图。已经基于RESTful Web服务开发了用于托管SM算法的SM微服务。微服务主要与Infrastructure Repository和VNF / NS Service Catalogs连接,后者是在每个映射时间填充前面段落中详述的基于图形的SM模型的数据库源。

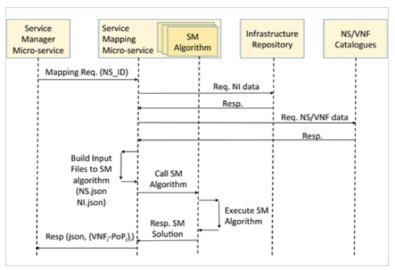


图8。 参考SM序列图。

每次必须映射NS时,都会向SM微服务发出请求,SM微服务又向基础架构存储库和服务目录查询构建和解决SM问题所需的输入信息。

C.基于ILP的网络服务映射

在本节中,我们将讨论在T-NOVA中提出和开发的基于ILP的服务映射方法。为了便于网络服务映射,我们将问题分解为两个级别:(i)将VNF分配给DC(以下称为 \hat{g} 一级SM)和(ii)将VNF放置到所选DC中的服务器上,如以及将NF图形边缘分配到NI内的物理路径上。这种方法显着降低了服务映射问题的复杂性。

在下文中,我们提出了针对一级服务SM问题的ILP公式。对于第二级,T-NOVA直接依赖于OpenStack原语。我们使用二进制变量 \mathbf{y}_p^H 表达VNF的分配 H去DC p,而二元变量 $X_p^h f_q^k$ 表明是否链接(h,在图中 G (N小号) = 过被映射到图中DC之间的路径 G (N一世) = (\mathbf{w} 使開紙接 $(p)_{r}$, q)

第一级映射的ILP公式如下:

Problem (基于 ILP 的服务映射)

中号我Ñ我中号我žë

$$\begin{array}{ll} \alpha & \& \operatorname{Sigma}_{p} \mathbf{y}_{p}^{H} + \beta \sum_{\pi \in P} \sum_{\substack{\mathbf{h} \in \mathbb{N} \\ \mathbf{h} \in V_{p} \in V}} \sum_{\Xi} \sum_{\pi \in P} \sum_{\substack{\mathbf{h} \in \mathbb{N} \\ \mathbf{h} \in V_{p} \notin \Psi_{q}}} \sum_{\mathbf{h} \in \mathcal{H}_{p}} \sum_{\mathbf{h} \in \mathcal{H}_$$

Subjectto

$$\sum_{p \in V} \mathbf{y}_p^H = 1 \qquad \forall \, \mathbf{h} \in V \qquad (2)$$

$$\Sigma$$
 Σ $\delta_{p,q} N_{p,q}^{h,k} \le \Delta_{\pi}$ ∀ $\pi \in P$ (4)

$$\sum_{\substack{p \in V \\ p \in V}} R R_H^{\mathsf{T}} \dot{y}_p^H \leq [R \stackrel{\mathsf{H}}{p} \forall p \in \mathring{\mathsf{N}}^{-\frac{\mathsf{H}}{p}} \quad \forall \ \mathsf{IR} \in \mathring{\mathsf{N}} \check{\mathsf{T}} \tag{6}$$

$$\mathbf{y}_{p}^{H} \in \{0, 1\} \quad \forall \, \mathbf{h} \in V, \quad \forall \, p \in V^{-\square}$$
 (7)

查看来源 🕝

目标函数(1)旨在隐式地建模实际服务映射目标,其是接受的NS请求的数量的最大化。它是三个分量的加权和:(i)向DC分配VNF的成本,(ii)总延迟和(iii)整体资源链路使用,通过将VNF之间的链路分配给DC之间的路径而得到。特别是,给定服务请求和网络基础设施,为每个VNFH组成服务,并为每个DC p 构成网络基础设施,我们引入虚拟成本(将VNF映射到DC)(p 用于加权分配 成本H至 p 在接受请求的最大化方面。加权参数,并根据实验活动的结果进行了调整。

约束(2)确保每个VNFH被精确映射到一个DC。约束(3)确保对于给定的一对 VNFH和 k 分配给DC p 和 q 分别在网络基础设施图中有一条路径 G (N 连接 至 q 边缘(h , 记被映射。约束(4)基于每个路径的延迟阈值强加SLA的可满足性P 组。约束(5)对每个链路资源类型强加了DC间连接的链路资源限制T在资源类型集中 L T.。约束(6)对每个节点资源类型强加DC的节点资源限制T在资源类型集中 \tilde{h} 下。约束(7)和(8)为变量强制执行二进制域。

我们的ILP公式包含多项式数量的变量和约束,因此适合通用求解器进行优化。因此,我们对该方法进行了计算评估。为此任务构建了一个合成模拟器,其工作原理如下。首先,模拟器填充了网络基础设施,并且包括在数据集中的网络搜索引擎[40]中描述的错误!找不到参考源。,映射算法文献中的流行参考。该数据集由210个基本实例组成,分为7类不断增加的网络,每个网络包含30个实例。任何实例都包含描述NI的图形和描述NSes的一些较小的图形。NS图表属于有限数量的拓扑,代表不同的服务类型。NI(分别为NS)图节点用表示可用(分别是必需的)节点和链接资源的数据进行注释。此外,对于每个NS节点,给出了与NI节点的兼容性向量,指示由于NS所需的特定资源,哪些映射是可行的。我们将NS图中的每个弧(并且只有它们)视为关键路径,其延迟约束必须得到遵守。

我们将NS请求的到达建模为泊松²过程,其平均到达间隔时间表示为 λ 。我们还假设NS分配请求的持续时间是随机的,遵循正态分布,其均值表示为 μ 其标准差表示为 σ 。模拟运行包括选择特定的基本实例(即NI和相应的可能NS池),然后(a)迭代和随机选择一个NS,到达间隔时间和相应分配请求的持续时间(b)要求映射算法为NI找到合适的绘制NS分配,(c)最终根据算法提供的映射或由于过期服务更新NI资源。一旦到达时间的总和超过给定的时间范围,模拟就结束 τ ,表示模拟的时间长度。

1) 调整模型参数:

在初步测试中,我们尝试评估服务映射算法的行为作为相应模型更改的参数。我们只考虑NI跨越20个NFV-PoP的实例, 3 而NS的数量设置为40.我们考虑了8个配置,每个模型参数的每个可能选择值1.0或值0.0 α , β 和 γ 。我们考虑给定的基本实例所有NS分配请求按照它们在实例文件中出现的顺序到达,具有可忽略的固定到达间隔时间,并且持续时间等于 τ 。也就是说,所有NS分配请求一个接一个地到达,它们被拒绝或分配,而在第二种情况下,分配的资源永远不会被释放。我们总是设置模拟长度 $\tau=168$ 小时(即一周),平均分配请求持续时间 $\mu=24$ 小时,和相应的标准差 $\sigma=2$ 小时。在所有配置中始终接受前20个分配请求。在接下来的20个中我们注册了从那里移动 $\beta=0.0$ 至 $\beta=1.0$ 大大提高了录取率;同样适用于移动 $\gamma=0.0$ 至 $\gamma=1.0$ 。搬离 $\alpha=0.0$ 至 $\alpha=1.0$ 与设置结合使用时仍有一些影响 $\beta=1.0$ 和 $\gamma=1.0$ 。这符合我们的建模目标,其中客观术语与之相关 β 和 γ 直接影响分配的可

行性,而那与之相关 α 仅用于多样化,因此在PoP之间平衡。总的来说,将所有参数设置为1.0会产生最佳结果。

2) 评估求解器随NI大小增加的可扩展性:

首先,我们验证了该方法的计算效率足以嵌入TeNOR的服务映射模块中。我们考虑使用开源求解器GNU GLPK或IBM的商业CPLEX求解器。每个求解器运行的时间限制为60秒。模拟使用多达100个NI节点的实例运行。

我们分析了两种情况:温和和高平均NI负载,在前一种情况下设置 $\lambda = \mu$ (0.50 *(L/升),在后一种情况下 $\lambda = \mu$ (0.75 *(L/升),其中L是NI中的总体可用CPU资源,并且 于是相应池中每个NS所需的平均CPU资源量。也就是说,如果CPU是唯一的稀缺资源,并且NS节点碎片是可能的,那么在温和(分别为高)平均NI负载情况下,我们预计总共会分配大约50%(分别为75%)的总CPU资源。我们考虑了两个性能指标:接受的NS分配请求的百分比,以及每个分配请求的平均计算时间。

我们首先观察到计算时间不是一个关键问题:商业解算器CPLEX永远不会超过时间限制,即使对于大型NI,平均响应时间也低至0.3秒。开源求解器GLPK产生的计算时间比CPLEX的计算时间大一个数量级;这是预期的,给出了文献中的基准测试结果。不过,平均查询时间总是小于3秒。分配器拒绝几乎总是由于求解器检测到不可行性而产生(在0.3%的运行中平均观察到超时,并且仅针对GLPK);报告平均拒绝比分配更快。总之,两个求解器在计算时间方面都可以很好地扩展:使用CPLEX或GLPK嵌入可能会产生性能瓶颈不是服务映射算法的系统。

同时,GLPK和CPLEX提供几乎相同的结果,即使GLPK在超时中更频繁地发生(0.3%的运行)。为了在需要非常快速的响应时进一步检查求解器的效率,我们通过在计算时间内设置3s而不是6os的时间限制来重复测试。减少时限并没有产生任何明显的低效率。因此,可以在相对较短的时间尺度上生成有效映射。

同样在接受率方面,嵌入CPLEX或GLPK,即使施加严格的时间限制,也不会对系统的整体性能产生重大影响。

3)解决方案可扩展性作为数据中心负载增加:

其次,我们分析了基于ILP的方法的性能,作为平均DC负载变化。特别是,我们将测试限制在NI包含20个节点的实例中。模拟考虑了NI节点的平均CPU负载(所请求的CPU资源与可用CPU资源的比率),表示为 δ ,范围从0.1到1.2,考虑每个步骤0.1点,并设置 $\lambda = \mu/(\delta^* (L \mathfrak{HF})$ 下点。

随着平均负荷的增加,接受率几乎呈线性地从100%下降到50%,即使在高应力水平下也具有稳健的系统稳定性。平均负载似乎对求解器的计算时间要求影响很小。在两个求解器之间的映射方面,时间限制的减少不会产生任何显着的低效率。

第五节

服务部署

供应和实例化代表了T-NOVA系统内的主要生命周期管理活动,包括网络服务和组成它们的VNF。一旦完成了服务映射(如第IV节所述),TeNOR系统就可以看到必须提供服务和功能的PoP。除了实例化之外,生命周期管理还包括终止VNF。图9显示了从TeNOR角度看NS Provisioning的序列图,包括工作流的高级交互。

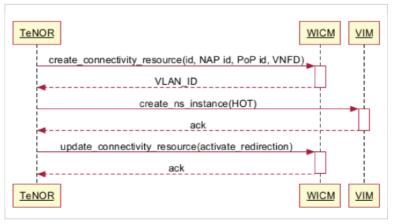


图9。

高级TeNOR NS配置序列图。

从一般角度来看,NS Provisioning由以下阶段组成(参见图1和2,了解T-NOVA架构组件):

- 如果需要,TeNOR通过发送所需信息(例如标识符,网络接入点或描述符) 在WICM中创建连接资源。它接收所创建的入口和出口网络点的相应VLAN标识符。
- TeNOR启动供应过程,该过程通过发送到VIM的HEAT [25]编排模板实现。 VIM负责实例化模板中描述的不同组件。
- TeNOR更新创建的连接资源(即WICM),以使流量重定向适应前一次调用 创建的实例(VM和虚拟网络)。

此过程实现了端到端服务供应的完全自动化,包括一个或多个NFVI-PoP以及由WICM管理的传输网络。

A. NS创建过程

创建一个NS实例,即create_ns_instance(HOT),如图9所示,需要对TeNOR内的各种微服务采取具体行动。首先,值得注意的是,配置一个NS可以理解组成它的所有VNF的部署,因此它可能成为一个复杂的自动化过程。本质上,前一工作流程的步骤2中的活动可以总结如下: (i)每当新的NS实例化请求从市场到达TeNOR时,NS Provisioning微服务从NS描述符中检索信息; (ii)使用该信息,服务映射返回可行PoP的有序列表,其中分配组成该服务的VNF; (iii)利用PoP和PoP间连接信息,启动一个循环,然后将属于NS的每个VNF转发到相应的VNF管理器,谁负责在VIM层部署VNF(即创建HEAT模板并与相应的VIM服务进行通信); (iv)最后,在每次成功的NS实例化请求时,创建一个SLA并订阅必要的参数进行监控(参见第VII节)。

B. NS供应的绩效评估

在评估T-NOVA内NS Provisioning的性能方面,要评估的最重要指标是整个NS供应时间。这是从收到请求的总时间,完成处理所需的时间,服务映射时间和NFVI中的VNF实例化。因此,第一个分析时间是映射算法,其计算NFVI中的最佳可用位置。然后,VNF实例化时间,其包括解析来自VNFD的信息,生成HEAT模板,以及将请求发送到OpenStack Heat服务以实例化NFVI中的VM。

图10中所示的图表描述了网络服务的多个(同时,1,2,5,10,15和20个请求)实例化所需的时间(仅包含一个需要2vCPU,2GB RAM和20GB磁盘空间的虚拟机))。映射时间是映射算法定义资源分配所需的总时间。实例化时间从NS配置知道在何处分配VNF直到实例在OpenStack中可用时开始。这一次包括租户和用户创建,Heat模板的创建以及从Glance下载VNF图像。

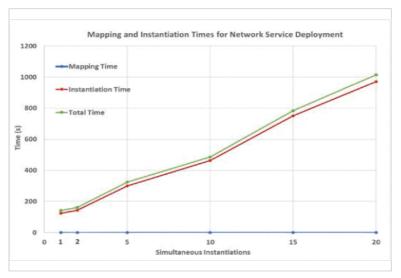


图10。 多个网络服务实例化的映射和实例化时间。

我们可以看到,映射时间在所有同时实例化中变化不大,因为REST调用/对Mapping 模块的响应需要不到100ms。但是,由于协调器向OpenStack发送唯一请求以部署网

络资源和每个实例的VNF映像,因此实例化时间随着同时实例化的数量逐渐增加而增加。在下一步中,OpenStack同时部署每个实例,导致对OpenStack的要求苛刻,以便同时分配资源。OpenStack分配操作成为瓶颈,因此给定5个同时实例化请求,部署所需的最后时间加倍。另外,关于上述图的统计显着性,计算相应时间序列的标准偏差值。结果为:映射时间为0.01544,实例化时间为341.4665,总时间为351.334。

第六节

基础设施和服务监控

在NFV环境中,适当的基础设施和服务监控至关重要,原因有很多: i)维护NFVI状态和资源的综合图,ii)实现适当的服务映射;iii)促进会计/计费和SLA管理以及iv)在故障情景中触发(主动或被动)动作。

三十多年来,IT和网络监控一直是活跃研究领域的一个领域。结果,可以获得各种各样的工具和概念,这些工具和概念可以被利用和/或适用于NFV环境。但是,并非所有这些都可以按原样使用。专门针对NFV环境量身定制的监控框架应满足特定要求,例如:

- 与VIM控制模块连接(例如, OpenStack, OpenDaylight等)
- 与VNF和/或VNFM连接以检索特定于VNF(应用程序)的度量标准
- 可扩展性。可扩展性要求至关重要,因为在大规模NFV环境中,MANO堆栈可能会因为NFV基础设施和服务的所有部分而每秒进行数百万次测量而过载。监控框架需要能够过滤掉不必要的数据并识别要传达给高级协调实体的特定事件(以及,如果需要,警报)。

为了满足这些要求,在T-NOVA中,我们采用了设计和开发专门针对运行NFV环境需求的轻量级监控框架的方法。在设计和实施方面,我们研究并利用了现有最先进的虚拟化环境监测和支持框架的一些特征和概念,如Monasca [42]和Gnocchi [43]。我们还研究了有关NFV监测的新兴OPNFV项目,如Doctor [44]和Prediction [45]。

我们的方法基于VIM层内的专用监控实体VIM监控管理器($VIM\ MM$),其功能架构如图11所示。

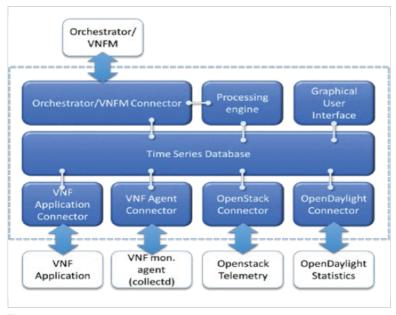


图11。 T-NOVA VIM监控管理器的功能架构。

VIM MM收集, 汇总和处理来自各种来源的数据, 即:

• 物理和虚拟基础架构,包括计算节点,存储服务器以及网络交换机和路由器。 为此,VIM MM直接与基础设施控制器(OpenStack和OpenDaylight)连接, 轮询其监控API(分别为Ceilometer / Telemetry API [46]和Statistics)。

- 运行VNF的VM。与Ceilometer本身提供的有限指标集相比,此处的目的是增强系统的监控功能。为此,我们在每个VM中引入一个监视代理程序。该代理基于流行的收集核心模块[47]可以以最小的开销直接安装。与OpenStack的Ceilometer数据收集服务相比,使用收集的代理可以访问来宾操作系统中更广泛的指标集,例如详细的内存和网络使用指标,负载信息和流程统计信息。它还以更高的分辨率收集指标,从而缩短监控系统的响应时间。代理可以预先安装在VNF映像中,也可以在部署时通过Heat模板安装;这两个选项都受支持。然而,在不希望在VNF中包含监视代理的情况下(例如,VNF开发者不允许/不支持),系统也可以在无代理模式下工作,仅依赖于来自Ceilometer的数据。
- VNF应用程序本身。VNF有望提供特定于应用程序的指标,例如会话数,流数,响应延迟等。为了便于将这些指标暴露给监控系统,我们创建了一个可供VNF使用的轻量级SDK开发人员将指标直接传达给VIM MM(除了与VNFM交换的信息)。

所有收集的度量标准都存储在时间序列数据库中,该数据库比标准关系数据库 (DB) 更适合于时间序列数据。为此,我们选择了InfluxDB [48]。通过将所有数据 聚合到高性能数据库并依赖于定期提要,我们可以简化工作流程,减少组件间信令,从而消除对消息队列的需求,消息队列通常用于监视框架。

后端处理引擎实时处理数据,以生成事件和警报。事件/警报通过预定义的阈值(由 Orchestrator设置)或动态检测与"正常"VNF操作的偏差生成。这是使用异常检测算 法实现的。异常检测功能仍在开发中。

所有信息都通过基于REST的API公开给Orchestrator,该API支持以下操作:

- 标准GET请求按需检索特定指标或指标组。
- 订阅定期"推送"特定指标的操作。
- 使用预定义阈值订阅警报。

在所有情况下,传达或用作警报阈值的值可以指瞬时样本或统计聚合(最小值,最大值,平均值,中位数等),除了程序界面(API)之外,度量还可以直观地显示,基于Web的图形用户界面(GUI),基于Grafana [49]。GUI的屏幕截图如图12所示。在这种情况下,将显示来自虚拟流量分类器(vTC)VNF的信息(请参阅下一节)。以下指标已集成到单个视图中:

- 从OpenStack Ceilometer检索的VNF CPU利用率
- 客户操作系统通过收集的代理报告的VNF内存使用情况和网络流量(累计数据包计数)。
- vTC服务报告的特定于VNF的指标(由vTC检测到的不同应用程序的数据包速率,例如Skype, Bittorrent, Dropbox, Google, Viber等)。



图12。 GUI屏幕截图汇总各种来源的指标以监控vTC VNF。

我们还评估了VIM MM的北向API的可扩展性和行为。为了模拟与Orchestrator的频繁数据交换,我们使用了许多并发GET请求,从vTC VNF指定单个指标(CPU负载)。我们使用httperf软件[50]以各种速率生成合成的HTTP GET请求,并测量收到的响应速率。然后,我们重复该过程,这次直接轮询Ceilometer以获得相同的度量。

这两组测量在具有类似硬件功能的平台上进行。结果如图13所示。

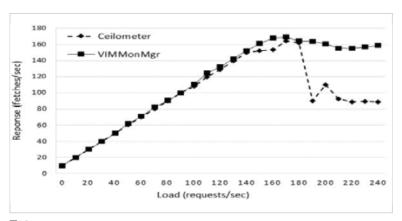


图13。 VIM Monitoring Manager性能。

从所获得的结果可以清楚地看出,VIM MM可以暴露具有与天然云量计相当的性能水平的度量。在过载情况下(对于给定的硬件配置,超过160个请求/秒),它似乎也表现出更好的稳定性。应该注意的是,这个相对较低的饱和点是由于平台的硬件资源有限;在任何情况下,结果都与Ceilometer相比并不是绝对的。

T-NOVA VIM监控框架的所有组件以及相应的文档已作为T-NOVA MANO堆栈的一部分在GPL许可下作为开源[51]发布。已发布的版本已集成在Docker容器中,以便于在异构环境中进行部署。

第七节

使用案例:部署硬件加速的虚拟流量分类器(vTC)

A. vTC架构和加速技术

在最后一节中,我们将T-NOVA MANO堆栈的指示性用例视为包含虚拟流量分类器(vTC)的NFV服务的部署,该虚拟流量分类器利用NFVI中通常可用的特定硬件加速功能。vTC也是在T-NOVA项目的背景下开发的,目前正被用作OPNFV Yardstick [52]项目的一部分,该项目侧重于NFV工作负荷表征。

vTC旨在分析实时网络流量,识别特定应用程序并根据应用程序优先级策略确定每个流量的优先级。由于此操作会产生大量工作负载,因此强烈建议使用特定的硬件加速功能来实现高效的流量处理。在以下段落中,我们简要描述了所采用的技术。

已开发vTC的Linux网络堆栈通常用作云网络解决方案的基础。其主要目标是为全功能操作系统提供通用网络堆栈,而不是专为高数据包吞吐量性能而设计的堆栈。因此,标准Linux网络堆栈无法扩展到软件网络设备通常所需的性能级别。

由于I/O性能在云基础架构中至关重要,因此需要进行虚拟化优化,以便最大限度地利用计算机系统资源。单根I/O虚拟化(SR-IOV)[53]是PCI-SIG发布的规范,它定义了减少虚拟机管理程序与VM交互的硬件增强,以提高其数据处理性能。支持SR-IOV的设备能够产生PCI功能的各种"轻"实例,称为虚拟功能(VF)。每个VF都可以固定到VM,允许直接访问其物理资源。利用此功能,多个VM可以共享相同的物理PCI设备资源,从而实现更高的性能,而不会产生大量额外的CPU开销。由于SR-IOV提供对系统硬件资源的直接访问和控制,因此可以有效分配低级网络资源。

vTC VNF用作T-NOVA中的用例,用于演示SR-IOV规范和数据包处理加速框架 DPDK提供的性能增强。vTC包括两个虚拟网络功能组件(VNFC),即流量检查引擎和分类和转发功能。两个VNFC在各自的VM中实现。建议的流量分类解决方案基于深度包检测(DPI)方法,该方法用于分析来自流的少量初始包以识别流类型。在流识别步骤之后,不再检查分组。流量分类器遵循基于分组的每流状态(PBFS)以跟踪相应的流。此方法使用表来基于5元组跟踪每个会话(源地址,为每个流维护的目标地址,源端口,目标端口和传输协议。vTC VNF的体系结构概述如下所示图 14。

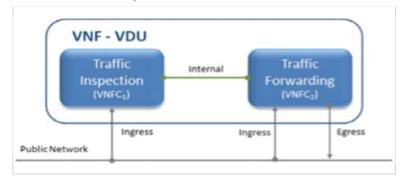


图14。 虚拟流量分类器VNF组件体系结构。

vTC利用各种技术来提供稳定且高性能的VNF。vTC实现基于开源nDPI库[54]。使用多种技术实现分组捕获机制,以便研究性能和模块性之间的相应折衷。评估的数据包处理/转发技术是:

PF_RING [55]: PF_RING是一组库驱动程序和内核模块,可实现高吞吐量,数据包捕获和采样。对于vTC,使用PF_RING内核模块库,它通过LINUX NAPI轮询数据包。数据包从内核复制到PF_RING缓冲区,然后使用nDPI库进行分析。

Docker [56]: Docker是Linux容器的一种形式,它提供了一个独立的执行环境,它基于共享主机操作系统的内核提供隔离的CPU,内存,块I/O和网络资源。为了研究容器技术的优缺点,vTC也是作为独立的容器应用而开发的。还使用PF_RING和nDPI执行流量的转发和检查,并且相应地对它们进行修改以在容器化环境中正常运行。

DPDK.为了在网络和计算领域充分利用系统资源,同时增强和促进密集网络应用的实施,英特尔开发了数据平面开发套件(Intel®DPDK)[57]。DPDK包含一组库,这些库通过访问系统的网络接口卡(NIC)来支持网络功能的高效实现。DPDK为网络功能开发人员提供了一套用于构建高速数据平面应用程序的工具。DPDK在轮询模式下进行数据包处理,而不是默认的中断模式。轮询模式操作采用忙等待技术,不断检查网络接口的状态变化。这减轻了数据包处理的中断,因为它绕过了内核,有效地消耗了CPU周期,从而导致数据包吞吐量增加[58]。使用DPDK网络数据包入口和出口与标准Linux内核网络堆栈相比更快,因为用户空间支持应用程序,从而绕过内核网络堆栈瓶颈。已经实现了启用DPDK的vTC版本,以便通过绕过内核空间来优化对检查和转发的流量的数据包处理和处理。分析和转发功能完全在用户空间中执行,这增强了vTC性能。

B.使用T-NOVA MANO组件进行验证和评估

作为验证测试,我们使用前面部分中描述的T-NOVA MANO组件来发现vTC的资源,分配适当的资源,部署服务并在负载条件下对其进行监控。在每次迭代期间,正确检测vTC的硬件加速功能(通过VNF描述符(VNFD)文档中的相应条目),并正确匹配具有这些特定功能的计算节点。

在物理和虚拟化环境中进行了vD的DPDK,PF_RING和Docker版本之间的一系列比较测试。使用开源PktGen流量生成器[59]生成测试流量,该流量生成器可以生成高达10Gbps的网络流量。收集的结果说明了基于线速扩展网络流量负载的每种实现的行为,线路速率高达10Gbps。每隔一秒从VNF收集流量统计数据,并进行后处理以进行性能评估。

图15中的结果显示了当使用DPDK时vTC的性能明显改善。物理和虚拟化解决方案之间的性能差距表明需要进一步优化才能实现线路速率。此外,很明显,尽管使用了SR-IOV,网络内核堆栈仍然是数据包处理路径的瓶颈。

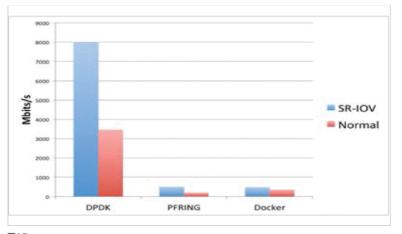


图15。 vTC VNF在各种设置环境中的比较结果。

如图14所示,SR-IOV / DPDK版本实现了物理DPDK测试数据包传输性能的大约81%。PF_RING内核版本在500Mbps时显示饱和效果,与DPDK版本相比,吞吐量降低了87.5%。结果还表明,与在非虚拟化硬件上执行的相应测试相比,DPDK在虚拟化场景中的性能降低了大约19%。这种性能下降是虚拟化环境中关于分组处理开销的额外管理程序开销的结果。

第八节 结论

我们介绍了由T-NOVA项目开发并作为开源项目发布的NFV MANO堆栈的组件 [11]。当前版本的T-NOVA MANO系统有助于关键NFV服务生命周期步骤的自动 化,特别是资源发现和匹配,服务映射,服务部署和监控。特别重要的是系统内发现,映射和利用硬件平台特定功能以优化性能的能力,如vTC评估活动中所示。完整的MANO堆栈在面向服务的体系结构下以完全模块化的方式设计,因此,如果需要,本文中描述的每个组件也可以单独使用。

要实现的其他功能包括服务自动扩展(扩展输入/输出),SLA监控,全自动流量转向/服务功能链(SFC)和异常检测。

 作者
 ン

 数据
 ン

 参考
 ^

引文图

1. B. Han , V。Gopalakrishnan , L。Ji , S。Lee , "网络功能虚拟化:创新的挑战和机遇" , *IEEE Commun。MAG。* , 第一卷 53 , 不。2 , pp.90-97 , 2015年2 月。

显示上下文 查看文章 全文: PDF (208KB) Google学术搜索

2. R. Guerzoni ,"网络功能虚拟化:介绍有益于推动者挑战并呼吁采取行动入门白皮书" ,*Proc。SDN OpenFlow World Congr。* ,2012年6月1日至16日。显示上下文 Google学术搜索

3. J. Wu , Z。Zhang , Y。Hong , Y。Wen , "Cloud radio access network (C-RAN) : a primer" , *IEEE Netw。*,第一卷 29 , 不。1 , pp.35-41 , 1月/2月。2015年

显示上下文 查看文章 全文: PDF (255KB) Google学术搜索

4. C-RAN: 走向绿色RAN的道路。白皮书。版本2.5,北京,中国,2011年10月。显示上下文 Google学术搜索

5. ETSI GS NFV 002 V1.2.1: 网络功能虚拟化 (NFV); 建筑框架, 索菲亚安提波利斯, 法国, 2014年12月, [在线]可用:

 $\label{eq:http://www.etsi.org/deliver/etsigs/NFV/001099/002/01.02.0160/gsNFV002 v010201p.pdf \end{substitute} $$v010201p.pdf $$$

显示上下文 Google学术搜索

6. ETSI GS NFV 003 V1.2.1: 网络功能虚拟化 (NFV); 法国索菲亚安提波利斯NFV 主要概念术语: ETSI Ind。规范组 (ISG) NFV, 2014年12月, [在线]可用: http://www.etsi.org/deliver/etsigs/NFV/001099/003/01.02.0160/gsNFV003v010201p.pdf。显示上下文 Google学术搜索

7. S. Oechsner, A。Ripke, "在OpenStack中灵活支持VNF放置功能", *Proc。第一届IEEE会议 网络。软化(NetSoft)*,第1-6页,2015年。显示上下文 查看文章 全文:PDF (223KB) Google学术搜索

8. K. Giotis, Y. Kryftis, V. Maglaris, "基于策略的软件定义网络中NFV服务的编排", *Proc. 第一届IEEE会议 网络。软化(NetSoft)*,第1-5页,2015年。显示上下文 查看文章 全文:PDF (358KB) Google学术搜索

9. M. Yoshida, W. Shen, T. Kawabata, K. Minato, W. Imajuku, "MORSA:用于NFV基础设施的多目标资源调度算法", *Proc。第16届亚太地区。网络。歌剧院。MANAG。SYMP。(APNOMS)*,第1-6页,2014年。显示上下文 查看文章 全文:PDF (342KB) Google学术搜索

10. 2017年8月, [在线]可用: http://www.t-nova.eu/。显示上下文

11. 2017年8月,[在线]可用:https://github.com/T-NOVA。显示上下文

12. G. Xilouris等人,"T-NOVA:虚拟化网络功能的市场",*Proc。欧元。CONF。网络。COMMUN。(EuCNC*),第1-5页,2014年6月。显示上下文 查看文章 全文:PDF (1053KB) Google学术搜索

13.2017年8月, [在线]可用:

https://osm.etsi.org/wikipub/index.php/Main_Page。 显示上下文

14. 开放式网络自动化平台(ONAP), 2017年8月, [在线]可用: https://www.onap.org。 谷歌学术

15. 2017年8月,[在线]可用:http://openbaton.github.io/。显示上下文

16. 2017年8月,[在线]可用:https://www.open-o.org/。显示上下文

17.2017年8月, [在线]可用:

http://about.att.com/content/dam/snrdocs/ecomp.pdf。 显示上下文

18. 2017年8月,[在线]可用:https://www.openstack.org/。 显示上下文

19. 2017年8月, [在线]可用: https://www.opendaylight.org/。

20. 2017年8月,[在线]可用:http://occi-wg.org/。 显示上下文

21. S. Bradner , J。McQuaid , 网络互连设备的基准测试方法 , 美国加利福尼亚州 弗里蒙特: Internet Eng。特遣部队 , 1999年3月。 显示上下文 CrossRef 谷歌学术

22.2017年8月, [在线]可用:

https://www.kernel.org/doc/Documentation/vm/hugetlbpage.txt。 显示上下文

23. JR Quinlan , C4.5:机器学习计划,美国加利福尼亚州旧金山:Morgan Kaufmann , 1993。

显示上下文 Google学术搜索

24. V. Riccobene, MJ McGrath, M.-A。Kourtis, G。Xilouris, H。
Koumaras, "使用基础设施亲和性表征自动生成VNF部署规则", *Proc。第二届IEEE会议 网络。Softwarization(NetSoft)*, pp.226-233, 2016年6月。显示上下文 查看文章 全文: PDF (3034KB) Google学术搜索

25. Heat Orchestration Template (HOT) 指南 , 2015 , [在线]可用:http://docs.openstack.org/developer/heat/template_guide/hot_guide.html

显示上下文 Google学术搜索

26. H. Ballani , P. Costa , T. Karagiannis , A. Rowstron , "迈向可预测的数据中心网络" , *ACM SIGCOMM Comput。COMMUN。Rev.* , vol. 41 , 不。4 , pp.242-253,2011。

在ACM上 显示上下文 访问 谷歌学术

27. J. Lee等人,"CloudMirror:应用感知带宽在云中保留",*PROC。第五届 USENIX研讨会热门话题云计算。*,第1-6页,2013年。显示上下文 Google学术搜索

28. C. Guo等人,"Second Net:带有带宽保证的数据中心网络虚拟化架构", Proc。ACM会议 EMERG。网络。进出口。TECHNOL。(CONEXT),第1-12页, 2010年。

在ACM上 显示上下文 访问 谷歌学术

29. A. Gember , R。Grandl , A。Anand , T。Benson , A。Akella , "Stratos : Virtual middleboxes as first-class entities" , 2012。 显示上下文 Google学术搜索

30. S. Oechsner, A. Ripke, "在OpenStack中灵活支持VNF放置功能", *Proc.* 第一届IEEE会议 网络。软化(NetSoft),第1-6页,2015年。显示上下文 查看文章 全文:PDF (223KB) Google学术搜索

31. J. Lee等人,"CloudMirror:云中的应用程序感知带宽预留",*Proc。第五届 USENIX研讨会热门话题云计算。*,第1-6页,2013年。显示上下文 Google学术搜索

32. A. Fischer, JF Botero, MT Beck, H. De Meer, X. Hesselbach, "虚拟网络嵌入:调查", *IEEE Commun。调查结果。*,第一卷 15,不。4,pp.1888-1906,4th Quart。2013。

显示上下文 查看文章 全文: PDF (5132KB) Google学术搜索

33. R. Riggio , T。Rasheed , R。Narayanan , "企业WLAN中的虚拟网络功能协调" , *Proc。IFIP / IEEE Int。SYMP。INTEGR。网络。MANAG。(IM)* , 第 1220-1225页 , 2015年。

显示上下文 查看文章 全文: PDF (1219KB) Google学术搜索

34. R. Guerzoni等人,"用于SDN管理和编排的虚拟网络嵌入的新方法",*Proc。 网络。歌剧院。MANAG。SYMP。(NOMS)*,第1-7页,2014年。 显示上下文 Google学术搜索

35. S. Mehraghdam , M。Keller , H。Karl , "指定和放置虚拟网络功能链" , *Proc。IEEE 3rd Int。CONF。云网。(CloudNet)*,第7-13页,2014年。显示上下文 Google学术搜索

36. V. Abedifar, M。Eshghi, S。Mirjalili, SM Mirjalili, "使用PSO在云计算中优化的虚拟网络映射", *Proc。21伊朗。CONF。选。工程。(ICEE)*, 第1-6页, 2013年。

显示上下文 查看文章 全文: PDF (763KB) Google学术搜索

37. R. Mijumbi等人,"网络功能虚拟化:最先进和研究挑战",*IEEE Commun。调查结果。*,第一卷 18,不。1,pp.236-262,1st Quart。2016年显示上下文 Google学术搜索

38。 D. Dietrich , A。Abujoda , P。Papadimitriou , "使用nestor嵌入多个提供商的网络服务" , *Proc。IFIP网。* ,第1-9页 , 2015年5月。显示上下文 查看文章 全文:PDF (234KB) Google学术搜索

39。 D. Dietrich, A。Rizk, P。Papadimitriou, "多供应商虚拟网络嵌入与有限信息披露", *IEEE Trans。网络。服务管理。*,第一卷 12,不。2,pp.188-201,2015年6月。

显示上下文 查看文章 全文: PDF (1807KB) Google学术搜索

40。 虚拟网络映射问题实例, 2015, [在线]可用:

https://www.ac.tuwien.ac.at/research/problem-instances/。 显示上下文 Google学术搜索

41. J.Inführ, GR Raidl, 引入虚拟网络映射问题延迟路由和地点的限制,德国海德堡:施普林格,页105-117,2011。 谷歌学术

42. *Monasca Monitoring-As-a Service for OpenStack* , 2017年8月,[在线]可用:https://wiki.openstack.org/wiki/Monasca。显示上下文 Google学术搜索

43. Gnocchi: OpenStack的TDBaaS, 2017年8月, [在线]可用:

 $https:/\!/wiki.openstack.org/wiki/Gnocchi_{\circ}$

显示上下文 Google学术搜索

44. OPNFV博士项目(故障管理), 2017年8月, [在线]可用:

https://wiki.opnfv.org/doctor。 显示上下文 Google学术搜索

45. *OPNFV预测项目(故障预测数据收集)*,2017年8月,[在线]可用:

https://ieeexplore.ieee.org/document/7997799/references#references

| T-NOVA: An Open-Source MANO Stack for NFV Infrastructures - IEE |
|---|
| nttps://wiki.opmv.org/preaiction。 显示上下文 Google学术搜索 |
| <u> </u> |
| 46. OpenStack遥测API v2(CURRENT),2017年8月,[在线]可用: |
| http://developer.OpenStack.org/api-ref-telemetry-v2.html。 |
| 显示上下文 Google学术搜索 |
| 47. <i>收集系统统计收集守护进程</i> ,2017年8月,[在线]可用: |
| https://collectd.org/。 |
| ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・ |
| |
| 48. InfluxDB:一个没有外部依赖关系的开源分布式时间序列数据库,2017年8月 |
| [在线]可用:http://influxdb.com/。 |
| 显示上下文 Google学术搜索 |
| 49. Grafana:Graphite InfluxDB和OpenTSDB的开源特性丰富度量仪表板和图册 |
| 438 , 2017年8月 , [在线]可用:http://grafana.org/。 |
| 显示上下文 Google学术搜索 |
| |
| 50。 2017年8月,[在线]可用:https://github.com/httperf/httperf。 |
| 显示上下文 |
| 51. 2017年8月,[在线]可用:https://github.com/T-NOVA/vim-monitoring |
| SELECT/中でリートにはSELECTION - MEDICAL M |
| 252.7% |
| 52. 2017年8月,[在线]可用:https://wiki.opnfv.org/yardstick。 |
| 显示上下文 |
| |
| 53. <i>PCI特别兴趣小组</i> ,2017年8月,[在线]可用: |
| http://www.pcisig.com/home. |
| 显示上下文 Google学术搜索 |
| 54. L. Deri , M。Martinelli , T。Bujlow , A。Cardigliano , "nDPI : Open- |
| source high-speed deep packet inspection", <i>Proc。诠释。无线通讯。移</i> 动 |
| 算机。CONF。(IWCMC),第617-622页,2014年8月。 |
| 显示上下文 Google学术搜索 |
| |
| 55. A. Cardigliano , L。Deri , J。Gasparakis , F。Fusco , "vPF_RING : 使用 |
| 拟机进行线速网络监控", <i>Proc。IMC</i> ,pp.533-548,2011年11月。 在ACM上 显示上下文 访问 谷歌学术 |
| 任ACM工业小工下文切问 台歌子小 |
| 56. 2017年8月,[在线]可用:https://www.docker.com/。 |
| 显示上下文 |
| |
| 57. DPDK:数据平面开发工具包,2015年8月,[在线]可用:http://dpdk.org |
| 显示上下文 Google学术搜索 |
| 58. K. Salah , A。Qahtan , "混合中断处理方案的实施和实验性能评估" , |
| Comput。COMMUN。, 第一卷 32,不。1, pp.179-188, 2009年1月。 |
| 显示上下文 CrossRef 谷歌学术 |
| |
| 59. PktGen Pktgen版本2.7.7使用DPDK-1.7.1,2017年8月,[在线]可用: |
| https://github.com/Pktgen/Pktgen-DPDK/。 |
| 显示上下文 Google学术搜索 |

引文 关键词 度量



A not-for-profit organization, IEEE is the world's largest technical professional organization dedicated to advancing technology for the benefit of humanity. © Copyright 2019 IEEE - All rights reserved. Use of this web site signifies your agreement to the terms and conditions.

US & Canada: +1 800 678 4333 Worldwide: +1 732 981 0060

| IEEE帐户 | 购买细节 | 档案信息 | 需要帮忙? |
|------------|-----------|---------|------------------------------|
| » 更改用户名/密码 | »付款方式 | » 通讯首选项 | » 美国和加拿大: +1 800 678 4333 |
| » 更新地址 | » 订单历史 | »职业与教育 | » 全球: +1 732 981 0060 |
| | » 查看购买的文档 | »技术兴趣 | » 联系与支持 |

关于IEEE Xplore | 联系我们 | 救命 | 无障碍 | 使用条款 | 非歧视政策 | 网站地图 | 隐私和选择退出Cookie

作为一个非营利组织,IEEE是世界上最大的技术专业组织,致力于为人类的利益推进技术。 ©版权所有2019 IEEE - 保留所有权利。使用本网站即表示您同意这些条款和条件。