

基于边缘计算平台的分析与研究

林 博,张惠民

(陆军装甲兵学院 信息通信系 北京 100072)

摘 要 边缘计算作为万物互联时代的关键技术,具有广泛的应用场景。文章首先分析了边缘计算平台在推广中面临的问题,随后从架构出发分析了典型边缘计算平台,并列举了边缘计算应用场景的需求参数,最后提出了一种边缘计算平台分类模型。

关键词 边缘计算;雾计算;Cloudlet;EdgeX Foundry;MEC

中图分类号 TP391 **文献标识码** A

DOI:10.19414/j.cnki.1005-1228.2019.04.007

Analysis and Research Based on Edge Computing Platform

LIN Bo,ZHANG Hui-min

(Department of Information and Communication, Academy of Army Armored Force, Beijing 100072, China)

Abstract : Edge computing is a important technology in the era of IoT and has a wide range of application scenarios. Firstly, This paper analyzes the problems faced by the edge computing platform in the promotion; then analyzes the typical edge computing platform from the architecture; and enumerates the demand parameters of the edge computing application scenario. Finally, the classification model of the edge computing platform is proposed.

Key words : edge computing; fog calculation; Cloudlet; EdgeX Foundry; MEC

物联网应用发展带来终端设备的指数型增加,据思科网络指数预估^[1],到 2022 年,网络设备连接量将达到 500 亿,其中物联网终端占比将达到 51%。由于物联网终端的资源局限,往往需要借助远端的云计算资源为用户提供服务,如果将终端所有数据传输到云中心统一处理后再返回终端,势必给网络链路和数据中心带来极大的压力,也极易导致云中心过载而拒绝服务(DoS),影响终端用户体验。因此,研究者们根据分

级分层计算的理念和应用实践,逐步提出了边缘计算模式,通过在用户邻近区域提供计算服务,有效减少网络和云计算中心资源压力。边缘计算并不是为了取代云计算,而是对云计算的拓展,为物联网提供更好的计算平台。边缘计算的构架是“端设备—边缘—云”三层模型,三层均可以为应用提供资源与服务,其结构如图 1 所示。

刘等^[2]阐述了边缘计算的定义,通过一些应用实例来分析边缘计算在移动应用和物联网应用上的优势,指出边缘计算的发展方向。吕等^[3]研究了边缘计算标准化的进展,分析了一些边缘计算应用案例。温^[4]研究了当前的多个边缘计算平台,提出了移动边缘计算和 Cloudlet 集成的改进架构,增强边缘计算平台的适应性。然而,应用场景的多样性导致了边缘计算缺乏统一标准、平台种类繁多,限制了边缘计算的推广应用。基于此,本文分析提出了一组边缘应用参数,构建了边缘计算的分类

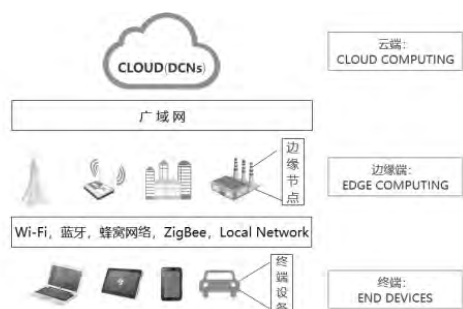


图 1 边缘计算逻辑结构图

收稿日期 2019-02-22

作者简介 林博(1996-)男,四川资中人,硕士研究生,主要研究方向 边缘计算 张惠民(1973-)男,福建上杭人,教授,博士,主要研究方向:云计算、大数据、深度学习、物联网。

模型。

1 典型边缘计算平台

边缘计算整合网络链路中广泛分布且数量众多的计算、存储资源为用户提供服务,通过边缘计算平台开发者可以快速发布和部署应用。目前,边缘计算平台数目众多,移动边缘计算(MEC)、雾计算、EdgeX Foundry 和 Cloudlet Computing(CC)是具有代表性的平台。

1.1 移动边缘计算

2014 年欧洲电信标准协会提出移动边缘计算(Mobile Edge Computing, MEC)的概念^[5], MEC 将网络控制、计算和存储能力带到网络边缘,有效减少用户延迟并拓展云计算的环境感知能力。MEC 节点通常位于大型基站或无线网络控制器处,与用户处于同一无线区域网(Radio Area Network, RAN)。移动边缘计算内部框架是“网络—主机—系统”三层结构,如图 2 所示。网络层代表移动边缘计算支持的连接方式,由移动蜂窝网络、本地网络和外部网络相关硬件组成。移动边缘主机层包含移动边缘主机和移动边缘层管理设备,其中移动边缘主机内部可以进一步划分为 ME 平台、ME 应用和虚拟基础设施。系统层是 MEC 最重要的一层,由运营商网络内的 ME 主机和管理 ME 应用必需的 ME 组建成,负责管控 MEC 的任务分配和系统运行。



图 2 移动边缘计算平台框架图

1.2 雾计算

雾计算是美国思科公司的研究成果,“雾计算是一个高度虚拟化的平台,可在终端设备和传统云计算数据中心之间提供计算、存储和网络服务,但通常位于网络边缘”^[6],雾计算实现了不同通信协议层和不同通信协议设备间的通信。因此,雾计算支持各种不同类型的基础设施,包括但不限于路由器、交换机、无线控制器、网关等设备。但是,丰富的设备支持导致雾计算平台的调度、运维和部署的难度增加。如图 3 所示,雾计算软件栈分为三层:抽象层、编排层、服务层。抽象层向上层输出标准的抽象层 API,隐藏不同设备的异构性,降低

应用开发难度。编排层利用抽象层 API 构建雾计算资源池,提供虚拟化基础设施,接收服务层的资源请求。服务层是虚拟机实例,用户请求的服务均在虚拟机中运行,兼容用户的可执行环境。



图 3 雾计算软件结构图

1.3 EdgeX Foundry

EdgeX Foundry 是一个由 Linux 基金会主持的开源项目^[7],其目的是为边缘计算的开发构建一个通用开放的平台,已获得 Dell、百度、Intel 等 50 余家企业的支持,EdgeX Foundry 兼容多种操作系统,支持多种硬件架构,支持不同协议设备间的通信。如图 4 所示,EdgeX Foundry 基于微服务架构设计有效提高应用与服务开发效率,其微服务被划分为四个服务层和两个基础系统服务,EdgeX Foundry 的四个服务层包含核心服务层、支持服务层、导出服务层和设备服务层;系统服务包含安全服务和管理服务。基于微服务可裁剪特性,从实现在低性能设备运行 EdgeX Foundry 服务。因此,EdgeX Foundry 支持嵌入式 PC、集线器、网关、路由器和本地服务器等异构设备。EdgeX Foundry 支持容器化部署,有效提高了平台运行效率。

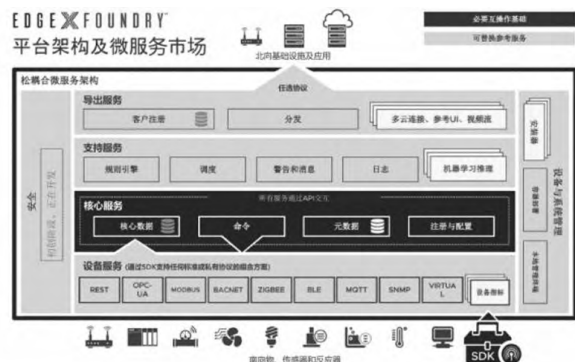


图 4 EdgeX Foundry 微服务架构图

1.4 Cloudlet Computing 平台

Cloudlet 于 2009 年由卡内基梅隆大学教授 Satya-narayanan M 提出^[8],Cloudlet 是一个位于边缘设备附

近,与互联网保持高速连接,且资源丰富的微型数据中心(Data Center in a box),向直接连接的局域网用户提供计算服务。Cloudlet 将用户数据直接保存在云端,用户即使移动到其他 Cloudlet 局域网中,也能通过云端备份数据快速为该用户重建应用服务。Cloudlet 通过高带宽的一跳网络向用户提供服务,最大程度降低延迟。如图 5 所示,Cloudlet 软件架构分为三层:第一层由用户数据缓存器组成,主要用于接收用户云端数据;第二层由 Cloudlet 边缘计算平台组成,实现主机资源虚拟化和编排管理应用虚拟机;第三层由虚拟机组成,运行用户卸载至边缘节点的应用,通过虚拟机弥补与用户可执行环境的差异。

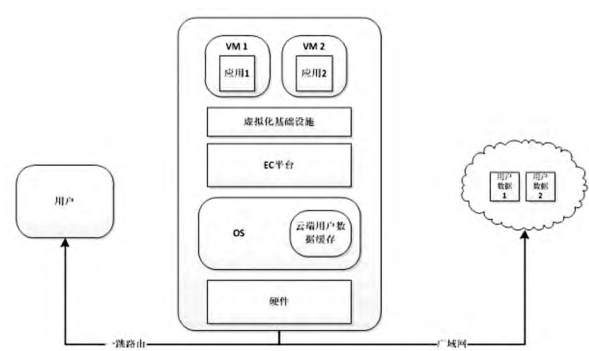


图 5 Cloudlet 平台结构图

2 边缘计算应用场景分析

边缘计算系统对众多边缘设备和边缘应用提供支持,但是不同的边缘计算系统具有不同的需求和均衡,决定了适合该应用场景的最佳边缘计算系统。本节根据边缘计算应用场景需求提出了一组参数,并分析了典型边缘计算平台在这组参数下的特点和性能(见表 1)。

邻近度:边缘层与设备层之间的邻近度包含有两层含义。第一,逻辑上的邻近度,代表边缘层基础设施与终端设备之间的路由跳数,跳数越多代表路由中遭遇拥塞的机会越大,延迟增加的可能性也越大。第二,指物理上的邻近度,物理上的邻近度取决于终端设备与边缘层的物理距离和边缘计算设备的性能。如果在单个边缘计算节点的一跳服务范围内,存在大量终端设备且已超过单个边缘计算节点的服务容量,将导致边缘计算设备拒绝服务,带来大量服务延迟。

接入方式:设备层到边缘层的连接方式有多种选择,如 Wi-Fi、蓝牙、蜂窝网络和有线连接等。接入方式在很多方面都占有重要地位,接入方式决定了终端设备到边缘计算层的带宽、延迟、连接范围以及对连接的设备类型。

表 1 边缘计算平台对比表

平台需求	移动边缘计算	雾计算	EdgeX Foundry	朵云计算
节点设备	服务器	路由器、交换机、网关等	定制设备、路由器、网关、交换机	微型数据中心、集群
节点位置	无线控制器 / 基站	终端设备到云的任何设备	主要位于网络接入点,部分存在于终端设备于云间	位于网络接入点
环境感知能力	高	中	中	低
邻近度	一跳	一跳或多跳	一跳	一跳
接入方式	蜂窝网络	蓝牙、WiFi、蜂窝网络	蓝牙、WiFi、蜂窝网络	WiFi
能耗	高	低	低	低
计算时延	低	高	低	低
部署与运维	一般	一般	简单	一般

环境感知:环境感知是将网络的参数和周围设备的位置信息暴露给附近的边缘节点,这是物联网应用的一个关键参数。移动边缘计算服务器通过放置在无线网络控制器附近,实现检测网络的链路状况、负载以及网络带宽,接收终端设备的位置信息,因此移动边缘计算的感知能力极强。

功耗:如果终端设备处于资源受限的环境中,功耗将是边缘计算的一个重要参数。蜂窝网络带来的能量

消耗远高于 Wi-Fi 的能量消耗。因此,移动边缘计算的能耗高于 Cloudlet。雾计算和 EdgeX Foundry 均拥有多种接入方式,可以根据需求灵活选择,实现最优的能耗表现。

计算时延和服务能力:计算时延和服务能力直接取决边缘计算节点的计算能力和资源调度策略。在这一方面,移动边缘计算和 Cloudlet 采用资源丰富的专用主机作为服务节点,具有显著优势,而雾计算节点主

要由传统设备组成,因此计算性能较差。EdgeX Foundry 通过容器和 GO 语言的轻量级特性,兼容不同性能设备,并按需提供计算服务。

部署与运维:在边缘计算平台的管理和监督中,每个平台均遵循分层管理方法,通常监督节点作为代理,负责与底层节点通信并收集底层节点可提供的资源和状态。然而,不同边缘计算平台在代理设置和分层的界定均有不同。雾计算平台中由于节点的异构和多样性,需要设计多个调度层,EdgeX Foundry 通过容器技术实现异构节点的统一管理;后两者通过构建统一的编排层即可实现对于设备和任务的监督和管理。

3 边缘计算选择模型

在本小节中,我们基于前一节中定义的多元应用场景参数提出了一个边缘计算选择模型图,如图 6 所示,直观反应不同参数与边缘计算系统间的耦合关系。在实际应用中,分析目标应用场景需求,将其对应到分类模型参数中,帮助用户快速选择目标应用场景的最佳边缘计算系统。

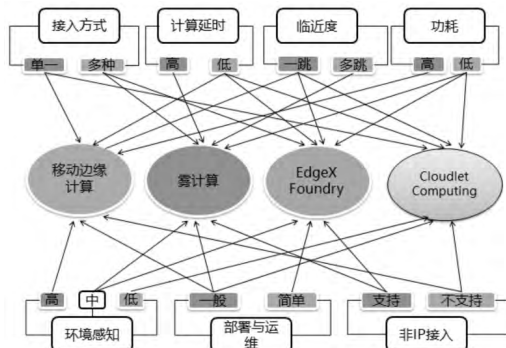


图 6 边缘计算分类模型图

例如,选取适用于远程医疗监护的边缘计算平台,分析在远程医疗监护应用中的需求,远程医疗监护主要便于医护人员即使掌握患者的生理状况,应对患者的突发状况,并实现远程诊疗服务。对于此应用场景,我们可以将其分为三个部分:患者数据采集端、数据中

心、监护中心。患者数据采集端是该应用场景的主要部分,其包含了以下功能:第一,采集和存储不同患者的生理状况数据;第二,对采集到的数据预处理,检测关键生理状况数据即时向患者家属和监护中心报警,并将处理后的数据传输到数据中心;第三,提供流媒体服务能力。数据中心用于长期存储患者数据,并即时推送到监护中心。监护中心具有管理所有患者数据和提供视频服务能力。

分析该应用的需求,首先采集数据将面临多种不同设备的接入,因此对于接入方式和对于非 IP 设备的支持应当作为第一选择,其次需要具有良好的网络环境感知能力,第三需要对患者的生理状况数据在边缘端预处理和长期存储;第四,对于流媒体传输服务需要良好的计算性能和较低的延迟需求;最后,在实现应用的功能需求时,降低功耗和部署运维难度也必不可少。根据边缘计算平台的特点和应用场景需求分析:第一,在接入方式与非 IP 设备接入需求中,EdgeX Foundry 和雾计算均支持多种接入方式和非 IP 设备的通信能力,但 Cloudlet 和移动边缘计算平台接入方式较少,不符合改应用场景需求,因此暂不考虑 Cloudlet 和移动边缘计算平台;第二,在网络环境感知中,移动边缘计算感知能力最强,EdgeX Foundry 和雾计算略低,Cloudlet 环境感知能力最差,移动边缘计算平台为最佳选择,而 EdgeX Foundry 和雾计算作为备选平台;第三,数据预处理、存储和应用延时需求与计算延时和服务能力相关,其中雾计算计算和存储能力差,与其他平台存在较大差距,因此优先考虑 EdgeX Foundry;第四,在功耗方面,雾计算和 EdgeX Foundry 平台均可以调整设备实现最优的功耗,而移动边缘计算和 Cloudlet 平台属于高功耗平台;最后,在部署运维和邻近度方面,EdgeX Foundry 均优于雾计算平台,需求对比见表 2。综合上述分析,EdgeX Foundry 相较于雾计算、移动边缘计算和 Cloudlet 更加契合远程医疗应用场景。

表 2 远程医疗监护需求表

	接入方式	非 IP	环境感知	计算延迟	邻近度	功耗	部署与运维
远程医疗监护应用	多	支持	高	低	一跳	低	简单
MEC	少	不支持	高	低	一跳	高	一般
Cloudlet	少	不支持	低	低	一跳	高	一般
EdgeX Foundry	多	支持	中	低	一跳	低	简单
fog	多	支持	中	高	多跳	低	一般
最佳平台	EdgeX Foundry、fog	EdgeX Foundry、fog	MEC	MEC、Cloudlet、EdgeX Foundry	MEC、Cloudlet、EdgeX Foundry	EdgeX Foundry、fog	EdgeX Foundry

(下转第 47 页)

共同制订专业人才培养方案,联合培养人才、组织教师进行企业锻炼、共建校内外实训与实习基地、联合开展技术合作等多种合作形式,增强高职院校办学活力、满足企业人才需求。还要不断促进高等职业教育与职业培训的衔接,强化培养学生职业能力与岗位能力,切实增强高职院校办学吸引力。

4 结束语

专业建设是高职职业院校发展的核心,是高职院校服务于区域经济发展的有效载体。只有牢牢抓住专业建设,才能更好的服务于区域经济和社会发展。高职专业建设存在专业结构不合理、专业内涵建设有待加强、专业设置与区域经济社会发展不匹配等问题,导致

高职人才供给与企业需要仍存在差距,迫切需要提出对策,提高供给端的质量与效益。

参考文献:

- [1] 陈春艳. 供给侧改革背景下高职专业建设研究 [J]. 江苏教育, 2016(52):28-29.
- [2] 姜玉凤,陈跃.“供给侧改革”背景下专业建设的探索——以无机非金属材料工程专业为例[J].湖北理工学院学报,2017(6):67-70.
- [3] 乐艳, 陈德林. 我国高职专业建设的现状、特征与对策研究——以 X 职业学院为例[J].湖北科技学院学报, 2018(2): 126-129.
- [4] 樊星.高职院校专业建设的对策研究[D].西安建筑科技大学, 2017.

(上接第 24 页)

4 结束语

雾计算,移动边缘计算,Cloudlet computing 和 EdgeX Foundry 是目前广受关注的四种边缘计算平台,这四种计算平台都具有自己的适用范围,本文通过研究物联网应用的特点,提出了不同的分类特性,对边缘计算的特征予以区别,利于边缘计算系统的推广。然而,边缘计算还在不断发展,未来边缘计算的特征和应用场景将更加复杂,因此边缘计算的分类势必更加复杂。

参考文献:

- [1] Cisco. 思科移动产业 VNI 预测报告[R].2018.
- [2] 赵梓铭,刘芳,蔡志平,等.边缘计算:平台、应用与挑战[J].计算机研究与发展, 2018, 55(2): 327-337.
- [3] 吕华章,陈丹,范斌,等.边缘计算标准化进展与案例分析[J].计算机研究与发展, 2018, 55(3): 487-3351.

- [4] 温华斌. 基于 Cloudlet 三层结构模型的移动协同计算平台的研究与实现[D]. 哈尔滨工业大学, 2015.
- [5] Garcia-Lopez,P., Montresor, A., Epema, D., Datta,A.,Higashino, T.,Iamnitchi,A.,Barcellos,M., Felber,P.,Riviere, E.: Edge-centric computing: vision and challenges. ACM SIGCOMM Comput. Commun. Rev. 45(5), 37-42 (2015)
- [6] Y. Jararweh, A. Doulat, O. AlQudah, E. Ahmed, M. Al-Ayyoub, and E. Benkhelifa, “The future of mobile cloud computing: integrating cloudlets and mobile edge computing,” in Telecommunications (ICT), 2016 23rd International Conference on. IEEE, 2016, pp. 1-5.
- [7] P.J. Escamilla-Ambrosio, A. Rodríguez-Mota, et al. Distributing Computing in the Internet of Things: Cloud, Fog and Edge Computing Overview, Studies in Computational Intelligence, 2018, 87-115
- [8] Satyanarayanan M, Lewis G, Morris E, et al. The role of cloudlets in hostile environments [J]. Pervasive Computing, IEEE, 2013, 12(4): 40-49.