

边缘计算：平台、应用与挑战

赵梓铭 刘芳 蔡志平 肖侗

(国防科学技术大学 计算机学院, 长沙 中国 410073)

Edge Computing: Platforms, Applications and Challenges

Zhao Ziming, Liu Fang, Cai Zhongping, Xiao Nong

(College of Computer, National University of Defense Technology, Changsha 410073)

Abstract With the trend of Internet of Everything, the number of end devices (e.g. smartphones, smartglasses) has increased rapidly which makes the data produced by these devices have grown at rates far more than the growth rate of network bandwidth. At the same time, the emergence of novel applications demands for lower latency of the network, such as augmented reality and manless driving. Edge Computing integrates any computing, storage and network resources at the edge of the network into a unified platform that provides services for users. This new computing model gets around the bottleneck that is caused by network bandwidth and latency, has received widespread attention in both industrial and academic. In this survey, we first introduce the concepts of edge computing, and provide the definition of it. To help the readers to better understand the characteristics of edge computing, we contrast it with cloud computing. We then analysis the two representative instances of edge computing platform and make a systematic comparison of them. After that, we enumerate some typical applications which base on the edge computing to describe the advantages of edge computing in mobile or Internet of Things applications. Finally, the paper lays out several grand challenges of edge computing.

Key words Edge Computing; Fog Computing; Mobile Computing; Internet of Things; Cloud Computing;

摘要 随着万物联网的趋势不断加深, 智能手机、智能眼镜等端设备的数量不断增加, 使数据的增长速度远远超过了网络带宽的增速; 同时, 增强现实、无人驾驶等众多新应用的出现对延迟提出了更高的要求。边缘计算将网络边缘上的计算、网络与存储资源组成统一的平台为用户提供服务, 使数据在源头附近就能得到及时有效的处理。这种模式不同于云计算要将所有数据传输到数据中心, 绕过了网络带宽与延迟的瓶颈, 引起了广泛的关注。本文首先介绍边缘计算的概念, 并给出边缘计算的定义; 随后, 比较了当前比较有代表性的两个边缘计算平台, 并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; 最后阐述了当前边缘计算面临的挑战。

关键词 边缘计算; 雾计算; 移动计算; 物联网; 云计算

中图法分类号 TP393

近年来, 大数据、云计算、智能技术的快速发展, 了新的要求。
给互联网产业带来了深刻的变革, 也对计算模式提出 大数据时代下每天产生的数据量急增, 而物联网

收稿日期: ; **修回日期:**

基金项目: 国家自然科学基金项目 (61433019,61332003,61379145); 国家重点研发计划项目 (2016YFB1000300)

This work is supported by the National Natural Science Foundation of China (61433019,61332003,61379145), the National Key Research and Development Program of China (2016YFB1000300).

通信作者: 刘芳 (liufang@nudt.edu.cn)

等应用背景下的数据在地理上分散,并且对响应时间和安全性提出了更高的要求。云计算虽然为大数据处理提供了高效的计算平台,但是目前网络带宽的增长速度远远赶不上数据的增长速度,网络带宽成本的下降速度要比CPU、内存这些硬件资源成本的下降速度慢很多^[1],同时复杂的网络环境让网络延迟很难有突破性提升,因此传统云计算模式需要解决带宽和延迟这两大瓶颈^[2]。

在这种应用背景下,边缘计算(Edge Computing)应运而生,并在近两年得到了研究者的广泛关注。边缘计算中的边缘(edge)指的是网络边缘上的计算和存储资源,这里的网络边缘与数据中心相对,无论是从地理距离还是网络距离上来看都更贴近用户。边缘计算则是利用这些资源在网络边缘为用户提供服务的技術,使应用可以在数据源附近处理数据。如果从仿生的角度来理解边缘计算,我们可以做这样的类比:云计算相当于人的大脑,边缘计算相当于人的神经末端。当针刺到手时总是下意识的收手,然后大脑才会意识到针刺到了手,因为将手收回的过程是由神经末端直接处理的非条件反射。这种非条件反射加快人的反应速度,避免受到更大的伤害,同时让大脑专注于处理高级智慧。未来是万物联网的时代,思科预计2020年将有500亿的设备接入互联网^[3],我们不可能让云计算成为每个设备的“大脑”,而边缘计算就是让设备拥有自己的“大脑”。

相比于云计算,边缘计算可以更好的支持移动计算与物联网应用,具有以下明显的优点:

1) 极大缓解网络带宽与数据中心压力:思科在2015-2020年的全球云指数^[4]中指出,随着物联网的发展,2020年全球的设备将会产生600ZB的数据,但其中只有10%是关键数据,其余90%都是临时数据无需长期存储。边缘计算可以充分利用这个特点,在网络边缘处理大量临时数据,从而减轻网络带宽与数据中心的压力。

2) 增强服务的响应能力:移动设备在计算、存储和电量等资源上的匮乏是其固有的缺陷,云计算可以为移动设备提供服务来弥补这些缺陷,但是网络传输速度受限于通信技术的发展,复杂网络环境中更存在链接和路由不稳定等问题,这些因素造成的延迟过高、抖动过强、数据传输速度过慢等问题严重影响了云服务的响应能力^[5]。而边缘计算在用户附近提供服务,近距离服务保证了较低的网络延迟,简单的路由也减少了网络的抖动,千兆无线技术的普及为网络传输速度提供了保证,这些都使边缘服务比云服务有更强的响应能力^[6]。

3) 保护隐私数据,提升数据安全性:物联网应

用中数据的安全性一直是关键问题,调查显示约有78%的用户担心他们的物联网数据在未授权的情况下被第三方使用^[7]。云计算模式下所有的数据与应用都在数据中心,用户很难对数据的访问与使用进行细粒度的控制。而边缘计算则为关键性隐私数据的存储与使用提供了基础设施,将隐私数据的操作限制在防火墙内,提升数据的安全性。

边缘计算因为其突出的优点,满足未来万物联网的需求,从2016年开始迅速升温,引起国内外的密切关注。ACM和IEEE从2016年开始联合举办边缘计算的顶级会议SEC(IEEE/ACM Symposium on Edge Computing),一些重要国际会议也都开始举办边缘计算的Workshop,例如2017年的ICDCS、MiddleWare等。

本文总结了边缘计算的相关概念,对比分析了边缘计算的相关平台,介绍了相关应用,并综述了边缘计算的发展趋势和面临挑战,对于广大研究者有很好的参考价值。

1 边缘计算概念

1.1 边缘计算定义

边缘计算目前还没有一个严格的统一的定义,不同研究者从各自的视角来描述和理解边缘计算。美国卡内基梅隆大学的Mahadev Satyanarayanan教授^[8]把边缘计算描述为:“边缘计算是一种新的计算模式,这种模式将计算与存储资源(例如:Cloudlet、微型数据中心或雾节点等)部署在更贴近移动设备或传感器的网络边缘。”美国韦恩州立大学的施巍松教授^[9]把边缘计算定义为:“边缘计算是指在网络边缘执行计算的一种新型计算模式,边缘计算中边缘的下行数据表示云服务,上行数据表示万物互联服务,而边缘计算的边缘是指从数据源到云计算中心路径之间的任意计算和网络资源。”

这些定义都强调边缘计算是一种新型计算模式,它的核心理念是“计算应该更靠近数据的源头,可以更贴近用户”。这里“贴近”一词包含多种含义。首先可以表示网络距离近,这样由于网络规模的缩小带宽、延迟、抖动这些不稳定的因素都易于控制与改进。还可以表示为空间距离近,这意味着边缘计算资源与用户处在同一个情景之中(如位置),根据这些情景信息可以为用户提供个性化的服务(如基于位置信息的服务)。空间距离与网络距离有可能并没有关联,但应用可以根据自己的需要来选择合适的计算节点。

网络边缘的资源主要包括移动手机、个人电脑等用户终端,WIFI接入点、蜂窝网络基站与路由器等基础设施,摄像头、机顶盒等嵌入式设备,Cloudlet、

Micro Data Center 等小型计算中心等。这些资源数量众多,相互独立,分散在用户周围,我们称之为边缘节点,边缘计算就是要把这些独立分散的资源统一,为用户提供服务。

综上所述,我们把边缘计算定义为:“边缘计算是一种新的计算模式,将地理距离或网络距离上与用户临近的资源统一起来,为应用提供计算、存储和网络服务。”

1.2 边缘计算、云计算、雾计算

边缘计算是一种新型的计算模式,从边缘计算的定义可以看出,边缘计算并不是为了取代云计算,而是对云计算的补充,为移动计算、物联网等提供更好的计算平台。边缘计算可以在保证低延迟的情况下为用户提供丰富的服务,克服移动设备资源受限的缺陷;同时也减少了需要传输到云端的数据量,缓解了网络带宽与数据中心的压力。目前,移动应用越来越复杂,接入互联网的设备越来越多,边缘计算的出现可以很好的应对这些趋势。但并不是所有服务都适合

部署在网络边缘,很多需要全局数据支持的服务依然离不开云计算,例如电子商务应用,用户对自己购物车的操作都可以在边缘节点上进行,以达到最快的响应时间,而商品推荐等服务则更适合在云中进行,因为它需要全局数据的支持。边缘计算的架构是“端设备—边缘—云”三层模型,三层都可以为应用提供资源与服务,应用可以选择最优的配置方案。我们将边缘计算与云计算的对比总结为表 1。

雾计算^[11] (Fog Computing) 是另一个与边缘计算相关的概念,它由思科公司在 2012 年提出,核心理念也是拉近用户与服务之间的距离。雾计算希望在路由器、交换机这些网络节点上运行应用,使数据源到数据中心整条链路都可以为用户提供计算、存储与网络服务,成为数据处理的“流水线”,而不仅仅是“数据管道”。雾计算的主要动机是为物联网提供可伸缩的基础设施,而非对移动应用的增强(如加快响应速度,感知情景信息等),例如智能家居就并非雾计算关注的重点。

Table 1 Comparison of Cloud Computing and Edge Computing

表 1 边缘计算与云计算比较

	边缘计算	云计算
目标应用	移动应用或物联网应用	一般互联网应用
服务器节点的位置	更贴近用户,位于数据源到数据中心的传输路径上 (如网关、WiFi 接入点或蜂窝网站等)	数据中心
客户端与服务器的通信网络	无线局域网, 4G/5G 等	广域网
可服务的设备(用户)数量	数十亿计	数百万计
提供的服务类型	基于本地信息的服务	基于全局信息的服务

2 边缘计算平台

边缘计算利用数据传输路径上的计算、存储与网络资源为用户提供服务,这些资源数量众多且在空间上分散,边缘计算平台将对这些资源进行统一的控制与管理,使开发者可以快速的开发与部署应用,成为边缘计算的基础设施。目前关于边缘计算平台的研究有很多,ParaDrop、Cloudlet 是其中比较有代表性的两个项目,其中从 Cloudlet 还演化出了 Open Edge Computing 联盟^[35]。

2.1 ParaDrop

ParaDrop^[12, 13] 是威斯康星大学麦迪逊分校 WiNGS 实验室的研究项目,无线网关可以在 ParaDrop 的支持下扩展为边缘计算平台,可以像普通服务器一样运行应用。ParaDrop 适用于物联网应用,例如智能电网(Smart Grid)、车联网(Connected Vehicles)、无线传感执行网络(Wireless Sensor and

Actuator Network) 等,可以作为物联网的智能网关平台。在物联网应用中,传感器数据都会汇集到物联网网关中,再传输到云中进行分析。而 ParaDrop 则在物联网网关中植入单片机使其具备通用计算能力,并通过软件技术使得部署在云端的应用与服务都可以迁移到网关,开发者可以动态定制网关上运行的应用。

ParaDrop 的整体结构如图 1 所示,ParaDrop 使用容器技术来隔离不同应用的运行环境,因此一个网关上可以运行多个租户的应用。网关上所有应用的安装、运行与撤销都由云端的后台服务控制,并对外提供一组 API,开发者通过 API 来控制资源的利用及监控资源的状态。用户通过 Web 页面与应用进行交互,ParaDrop 将 Web 的服务与数据分离,Web 服务由云端的后台服务提供,而传感器采集的原始数据则都存储在网关上,用户可以对云端访问的数据进行控制,保护了用户的数据隐私。

ParaDrop 的优势主要有:敏感数据可以在本地处理,不用上传云端,保护了用户隐私;WIFI 接入点

距离数据源只有一跳，具有低且稳定网络延迟，在 WIFI 接入点上运行的任务有更短的响应时间；减少传输到互联网上的数据量，只有被用户请求的数据才会通过互联网传输到用户设备；网关可以通过无线电信号获取一些位置信息，如设备之间的距离，设备的

具体位置等，利用这些信息可以提供位置感知的服务；遇到特殊情况，无法连接互联网时，应用的部分服务依然可以使用。目前，ParaDrop 得到了很好的发展，软件系统已经全部开源，支持 ParaDrop 的硬件设备也已经准备对外销。

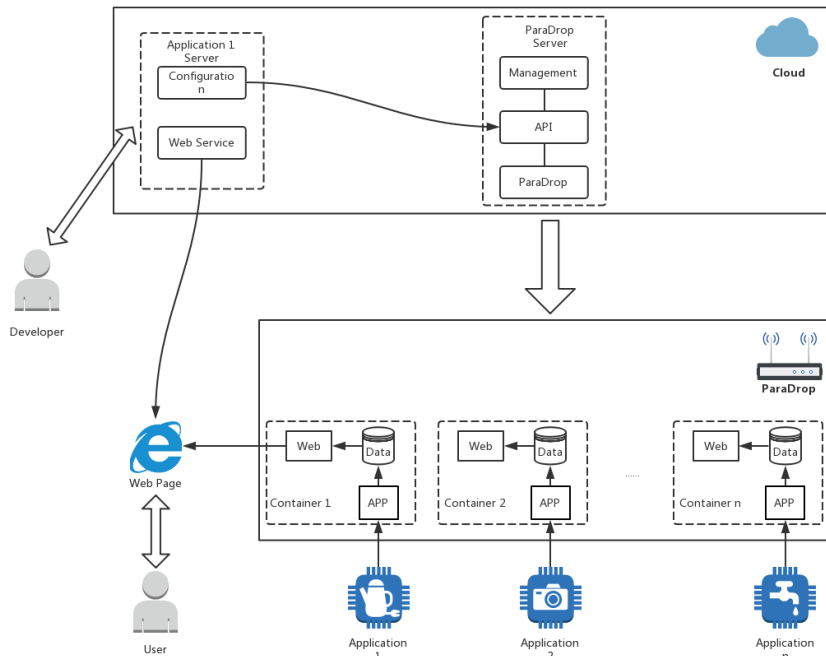


Fig.1 The full ParaDrop platform

图 1 ParaDrop 平台结构图

2.2 Cloudlet

2009 年卡内基梅隆大学提出 Cloudlet^[10]的概念，Cloudlet 是一个可信且资源丰富的主机或机群，它部署在网络边缘与互联网连接并可以被周围的移动设备所访问，为设备提供服务。Cloudlet 将原先移动计算的两层架构“移动设备—云”变为三层架构“移动设备—Cloudlet—云”，Cloudlet 也可以像云一样为用户提供服务，所以它又被称为“小云”(Data Center in a Box)。虽然 Cloudlet 项目不是以边缘计算的名义提出并运行，但它架构和理念契合边缘计算的理念和思想，可以被用来构建边缘计算平台。

Cloudlet 主要用来支持移动计算中的游牧服务^[14](Cyber Foraging)，游牧服务是解决移动设备计算资源不足的重要手段，通过游牧服务移动设备可以将繁重的计算任务卸载到其他资源上。云计算一直是充当这类资源的最佳角色，而 Cloudlet 的出现为用户提供了新的选择。Cloudlet 的软件栈分为三层，第一层由操作系统和 Cache 组成，其中 Cache 主要是对云中的数据进行缓存，第二层是虚拟化层，将资源虚拟化，并通过统一的平台 OpenStack++^[15]对资源进行管理。

第三层是虚拟机实例，移动设备卸载的应用都在虚拟机中运行，这样可以弥补移动设备与 Cloudlet 应用运行环境（操作系统、函数库等）的差异。

与云不同，Cloudlet 部署在网络边缘，只服务附近的用户，但 Cloudlet 也支持应用的移动性，设备可以随着移动切换到最近的 Cloudlet。如图 2 所示，Cloudlet 对应用移动性的支持主要依赖三个关键步骤：

- 1) Cloudlet 资源发现 (Cloudlet Discovery)^[16]: 移动中的移动设备可以快速发现周围可用的 Cloudlet，并选择最合适的作为卸载任务的载体。
- 2) 虚拟机配给 (VM Provisioning)^[16]: 在选定的 Cloudlet 上启动运行应用的虚拟机，并配置运行环境。
- 3) 资源切换 (VM Handoff)^[17]: 将运行应用的虚拟机迁移到另一个 Cloudlet 上。

动态虚拟机合成 (Dynamic VM Synthesis)^[10]是 Cloudlet 支持移动性的关键技术，可以将虚拟机镜像拆分为基底 (Base) 与覆盖层 (Overlay)，基底与覆

盖层也可以重新组合为虚拟机镜像。基底包含虚拟机的操作系统、函数库等基础软件,这一部分在虚拟机镜像之间都是重复的,且占用空间大;而覆盖层是一个很小的二进制增量文件,只包含用户在原始虚拟机上的一些定制信息,占用空间小。在虚拟机配置和资源切换时,使用动态虚拟机合成技术可以只传输轻量的覆盖层,减少了数据传输量,加快了虚拟机配置和资源切换的速度,保证了应用在 Cloudlet 中能得到及时的资源供给。

Cloudlet 的主要优势有:对应用开发者没有任何约束,现有程序基本不需要修改就能在 Cloudlet 中运

行;加快了很多复杂移动应用的响应速度。随着研究不断完善,Cloudlet 在认知辅助系统(cognitive assistance system)^[18]、众包(Crowdsourcing)^[19]、敌对环境(Hostile Environments)^[20]等方面都有很好的应用。为了推动 Cloudlet 的发展,CMU 联合 Intel、Huawei 等公司建立了 Open Edge Computing 联盟,为基于 Cloudlet 的边缘计算平台制定标准化 API。目前,该联盟正在将 OpenStack 扩展到边缘计算平台,使分散的 Cloudlet 可以通过标准的 OpenStack API 进行控制和管理。

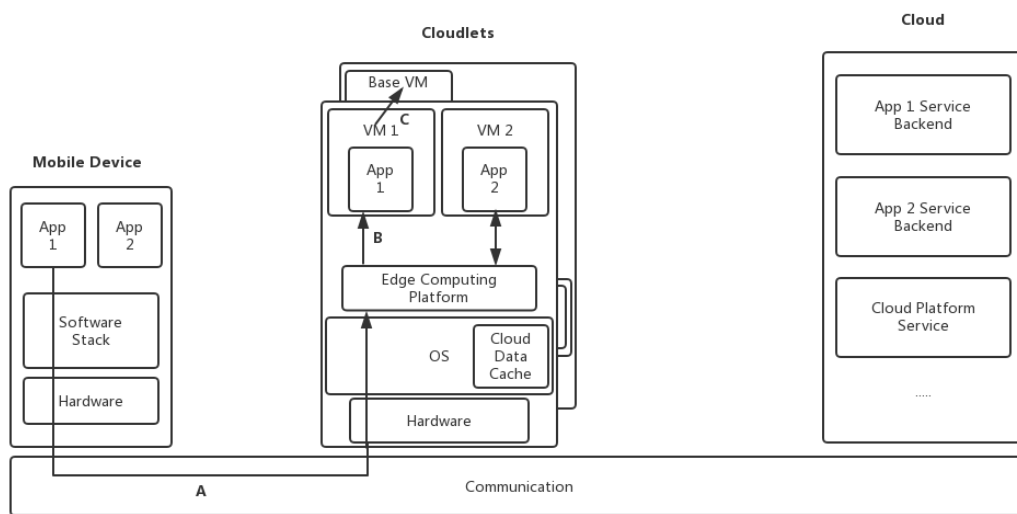


Fig.2 Cloudlet Component Overview and Functions that support application mobility. A:Cloudlet Discovery, B: VM Provisioning, C: VM Handoff

图 2: Cloudlet 组件总揽及应用移动性机制。A: Cloudlet 资源发现, B: 虚拟机配置, C: 资源切换

3.3 小结

从应用领域、服务移动性、服务状态、虚拟化技术这四个方面对边缘计算平台进行比较,并总结为表 2。

1) 应用领域:虽然这两个平台都可以在网络边缘为用户提供服务,但是他们在设计时所针对的应用领域存在差异。Cloudlet 针对延迟敏感的移动应用,而 ParaDrop 则针对物联网应用。

2) 服务的移动性:应用领域的不同导致这两个平台对服务移动性的支持不同。Cloudlet 是为移动应用的后台服务提供临时的部署点,为了保证低且稳定的网络延迟,设备的移动会使后台服务也要移动到就近的 Cloudlet;就近服务的特性使 Cloudlet 对移动性的支持特别困难,需要资源发现、虚拟机配置、资源切换这三步结合,同时还要保证实时性。而在物联网

应用中,大多数应用的流程是传感器采集原始数据汇集到无线网关进行初步处理,处理结果上传到云端进行进一步的分析;传感器与无线网关的连接关系一般保持不变,因此不需要考虑服务移动性问题。

3) 服务状态:移动应用虽然能将后台服务部署在 Cloudlet 上,但这种部署只有短暂的一段时间,Cloudlet 不会长久保存服务的状态信息,重要的信息都要传输到云端保存,一旦应用离开 Cloudlet 的服务范围,这些数据会被清除。而 ParaDrop 则直接存储传感器的原始数据,web 服务需要的数据都需要从本地获取。

4) 虚拟化技术:虚拟化技术方便了资源的管理,是边缘计算平台的必然选择。Cloudlet 使用虚拟机来虚拟化资源,而 ParaDrop 则使用容器。主要的原因是移动应用后台服务的执行环境多种多样,无论是基

于 Windows 或是 Linux 系统的后台服务都应该可以快速迁移到同一个 Cloudlet 上,虚拟机是对物理机器的虚拟化,可以很好的应对执行环境的变化,容器则依赖特定的操作系统。而物联网应用的后台服务一般不需要灵活的执行环境,边缘计算平台的执行环境与云端保持一致,因此使用容器就可以满足需求,同时还具备占用资源少、启动快等优点。

Table 2 Comparison of Cloudlet and ParaDrop

表 2 Cloudlet 与 ParaDrop 比较

	Cloudlet	ParaDrop
应用领域	移动应用	物联网应用
服务移动性	支持	不支持
服务状态	软状态	硬状态
虚拟化技术	虚拟机	容器

3 应用案例

边缘计算在数据源附近提供服务,使其可以在很多移动应用和物联网应用上发挥出巨大优势,本节将列举一些典型的应用案例,结合这些案例可以帮助我们理解边缘计算的优势。

3.1 增强现实

增强现实技术将现实世界的场景与虚拟信息高度集成,生成被人类感官所感知的信息,来达到超越现实的感官体验。增强现实技术可以使用在智能手机、平板电脑与智能眼镜等移动设备上,来支持新的应用与服务,如虚拟游戏、3D 观影等。增强现实技术需要对视频、图像数据进行处理,这些任务复杂度高,而需要与用户进行互动的特点又对实时性有了很高的要求。

CMU 与 Intel 实验室在 2014 年开发了一个基于增强现实技术的认知辅助系统 (cognitive assistance system) [18],通过谷歌眼镜来增强某些病人的认知能力。实现系统需要解决的关键问题是如何将处理任务的延迟控制在几十毫秒,让感知缺陷的病人也拥有正常人一样的反应速度。考虑到重量、大小、续航等因素,可穿戴设备的计算能力很差,处理任务的时间是一般服务器的数倍,直接使用设备内部的资源进行计算是不可行的。将应用部署到云中可以加快任务的处理速度,但端设备到云端的网络延迟很高且极不稳定,很可能成为整个系统的瓶颈。为了解决这个问题,系统使用了边缘计算技术,将延迟敏感的计算任务卸载到附近的 Cloudlet 来降低任务的处理延迟。同时为了保证系统在无法连接网络时依然可以使用,系统也支持通过蓝牙等通信方式将任务卸载到附近的个人

设备 (如随身携带的笔记本、平板电脑等)。

网络的延迟与设备性能、能耗的瓶颈是很多移动应用都会遇到的问题,而边缘计算可以帮助移动应用突破这些瓶颈,让应用具有更快的响应速度,使用更复杂的算法。

3.2 深度学习

美国里海大学与 IBM 提出了一个基于深度学习的自适应对象识别框架 DeepCham^[21],该框架适用于移动设备,将边缘计算节点作为 master 来控制附近的移动设备训练深度模型。DeepCham 采用众包的思想,深度模型训练所使用的数据集与数据的标记都是由参与者提供,这种方式使其可以获取大量有标记的数据集,进行有监督学习。

在一个特定视域 (图像的光线、背景、视角等) 内采集图像用来训练深度模型,得到的深度模型对该视域内对象的识别准确率更高。DeepCham 充分利用这一点,在同一个边缘计算节点周围采集数据来训练模型,这使模型很好的适应周围的视域,从而使 DeepCham 可以自适应视域的转换。

适用于特定领域的深度学习模型要比通用模型更好训练,也有更高的准确性。边缘计算模式可以为深度学习收集大量特定领域的学习数据,训练更加个性化的深度模型。

3.3 网站性能优化

网站性能优化 (Web Performance Optimization) 是用来提高用户浏览器的网站加载和显示速度的技术,随着用户体验的重要性不断增强和用户速度的需求日益增长,网站性能优化行业得到快速发展,很多互联网公司都对外提供网站性能优化的服务与工具,如雅虎的 YSlow^[41]与谷歌的 PageSpeed Tools^[42]。在用户请求网页的过程中,80%~90%的响应时间都发生在前端 (下载组件、页面的渲染与执行等),在网络边缘上的优化才是提高网站性能的关键^[33]。传统的网站优化方案是在 Web 服务器上利用固定规则优化网站页面,再通过内容分发网络加速传输。这种方法没有充分利用边缘网络资源,优化方法对所有用户都是一样的,没有考虑到用户的网络状态情况;内容分发网络虽然是在边缘网络上的优化技术,但是它只能加快组件的下载速度,页面渲染、执行的速度依然取决于设备的计算能力。

为了充分利用边缘网络资源,日本电报电话公司 (NTT) 设计了一种基于边缘计算的网站加速平台 EAWP (Edge Accelerated Web Platform) ^[34],为 Web 应用开发者提供情景感知的网站优化服务与工具。这个方案中,边缘服务器与 WIFI 接入点、蜂窝网基站等通信设施结合,可以获取用户接入网的状态信息来

对网站进行优化。比如,当发现用户所在的边缘网络出现拥塞时,边缘服务器可以降低页面质量(如使用低分辨率图片)来优化访问的响应时间。同时,为了弥补移动设备计算能力的不足,平台支持将页面内容的执行、渲染等复杂的工作卸载到边缘服务器中执行。EAWP 为其他支持 HTML 标准的 Web 引擎提供等效的运行环境,现有的应用程序可以不需要更改直接在平台上运行。

3.4 智慧城市

智慧城市是一种现代化城市模型,运用信息技术与物联网技术对城市资源做出智能化、的管理。智慧城市在近几年得到了快速发展,IBM、Inetl 和 Google 等公司都开始将他们的产品与服务整合到智慧城市的框架中。智慧城市系统要随时感测、分析、整合城市的各项关键信息,会产生大量的原始数据,一座一百万人的城市,平均每天会产生 200PB 的数据^[4]。同时,这些数据在地理上广泛分布,且大部分数据存储在本地,这为数据的查找与分析带来了极大的困难。如果没有一种高效的解决方案,很容易使城域网被大量的数据堵塞。

Tang^[22]等人提出了一种以智慧城市为背景的大数据分析框架,对处理在地理上广泛分布的数据有很好的效果。析框架分为四层,第一层是传感器网络,由分散在城市中的传感器构成,昼夜不停的生成大量原始数据;第二层由边缘节点组成,每个边缘节点都要控制本地的一组传感器,边缘节点可以根据预先设定的模式分析和处理传感器数据,还可以控制执行器处理任务;第三层由中间计算节点组成,每个中间节点要控制一组边缘节点,将边缘节点上传的信息与时空信息相结合来识别一些潜在的突发事件,当突发事件发生时,中间节点还要控制下层设备做出应急响应;第四层是云计算中心,对全市的状态进行监控并进行中心控制,在这一层进行长期的、全市范围的行为分析。

这个分析框架使用了边缘计算技术,第二、三层构成了边缘计算平台。边缘计算平台充分利用了数据传输路径上的计算设备,将众多互不相关的轻量级任务分配到各个节点,使得任务可以并行执行;同时,原始数据在这两层加工后,已被精炼化,在核心网络上传的数据量大大减小。边缘计算技术保证了分析框架的高效运行,减少了需要上传到云中的数据量,是整个框架的高效运行的关键。

3.5 车联网

车联网将汽车接入开放的网络,车辆可以将自己的状态信息(如油耗、里程等)通过网络传到云端进行分析,车辆间也可以自由交换天气、路况、行人等

信息,并进行实时的互动。

韦恩州立大学在 GENI Racks 上构建了一个边缘计算平台^[23],并在上面部署了实时 3D 校园地图、车量状态检测、车联网仿真三个应用。3D 校园地图通过将校园内监控录像与行驶车辆的录像数据融合,通过处理后可以增强为实时 3D 地图,校园安保人员可以无缝的监控校园状态;车量状态检测可以实时记录车辆的引擎转速、里程、油耗等状态,并对数据进行分析,从而检测车辆的性能,发现车辆的故障;车联网仿真将众多的车辆状态信息汇总,利用这些真实的交通信息可以进行车联网应用的仿真实验。

这些应用都会产生大量的传感器数据,很多数据都需要进行实时处理,而边缘计算可以在数据源附近对数据进行处理,减少了不必要的网络传输,并提高了应用的响应速度。

4 边缘计算面临的挑战

目前,关于边缘计算的研究才刚刚起步,虽然已经取得了一定成果,但从实际应用来说,还存在很多问题需要研究,下面对其中的几个主要问题进行分析。

4.1 多主体的资源管理

边缘计算资源分散在数据的传输路径上,被不同的主体所管理和控制,比如用户控制终端设备、网络运营商控制通信基站、网络基础设施提供商控制路由器、应用服务供应商控制边缘服务器与内容传输网络。云计算中的资源都是集中式的管理,因此云计算的资源管理方式并不适用管理边缘计算分散的资源,而目前关于边缘计算的研究^[24, 25]也主要集中在对单一主体资源的管理和控制,还未涉及多主体资源的管理。一种比较直观的解决方式是各个主体对资源自我管理,然后通过中间服务(broker service)^[26]来进行资源供给。但这种方式只能提供基本的功能,如果要满足使用者的特殊需求(如自动供给),中介层则需要自己实现部分 IaaS 平台功能,这需要依赖各个主体提供的 API,只要有一个主体提供的 API 不够灵活,就很难实现。因此实现灵活的多主体资源管理是一个十分富有挑战性的问题。

4.2 应用的移动管理

边缘计算依靠资源在地理上广泛分布的特点来支持应用的移动性,一个边缘计算节点只服务周围的用户。应用的移动就会造成服务节点的切换。而云计算对应用移动性的支持则是“服务器位置固定,数据通过网络传输到服务器”,所以在边缘计算中应用的移动管理也是一种新模式,主要涉及以下两个问题:

资源发现：应用在移动的过程中需要快速发现周围可以利用的资源，并选择最合适的资源。当前虽然也有很多成熟的资源发现技术，在云监控（Cloud Monitor）^[27,28]与云中介（Service Brokerage）^[29]中被广泛运用，但边缘计算的资源发现需要适应异构的资源环境，还需要保证资源发现的速度，才能使应用不间断的为用户提供服务。

资源切换：用户移动时，移动应用使用的计算资源可能会在多个设备间切换，而资源切换要将服务程序的运行现场迁移。热迁移技术可以解决这个问题，但是传统热迁移技术的目标是 minimized 停机时间，而资源切换需要最小化总迁移时间，因为在迁移的过程中用户要忍受升高的延迟。另外，传统的虚拟机迁移是在数据中心的内部进行，设备的计算能力与网络带宽比较固定，而边缘计算资源的异构性与网络的多样性，需要迁移过程自适应设备计算能力与网络带宽的变化。所以，边缘计算需要一套自适应的快速热迁移方案，来满足移动应用资源切换的需求。

4.3 虚拟化技术

为了方便资源的有效管理，边缘计算需要虚拟化技术的支持，为系统选择合适的虚拟化技术是边缘计算的一个研究热点。边缘计算对虚拟化技术的要求体现在如下几个方面，首先，边缘计算资源是一种基础设施，要尽可能的保持通用性，所以虚拟化技术应该实现最小化对应用程序运行时环境的约束，不应强制应用使用特定的操作系统、函数库等；其次，边缘计算资源的能力有限，不能像计算中心一样为应用提供充足的资源，虚拟化技术应最大化资源利用率，使有限的资源在同一时间内满足更多的请求；最后，有些边缘计算资源在处理用户任务的同时还要对外提供其他服务，虚拟化技术应将不同的任务彻底隔离，一个应用的崩溃，内存溢出，高 CPU 占用不会对其他的任务造成影响。例如在移动边缘计算^[30]中，基站能够处理用户的任务，但是这些任务不能影响基站最基本的无线接入功能。这三点可能会出现冲突，系统要根据自己的需求在这之间做出权衡。目前，新型的虚拟化技术层出不穷，其中有很多打破了虚拟机和容器的规则与界线，将二者充分融合，同时具备二者的优势，如 LXD^[36]、Hyper^[37]与 Rancher OS^[38]等。所以，不拘泥于虚拟化技术现有的规则与界线，设计适应边缘计算特点的虚拟化技术也是一大挑战。

4.4 数据分析

数据分析的数据量越大，往往提取出的价值信息就越多。但是收集数据需要时间，价值信息往往也具有时效性，没有人会关注昨天的天气预报。边缘计算使数据可以在汇集的过程中被处理与分析，很多数据

如果被过早的分析，可能会丢失很多有价值的信息，所以如何权衡提取信息的价值量与时效性是一个关键性问题。

边缘计算利用的计算节点数量众多，但节点的计算资源有限，很多都是单片机或片上系统，例如 Intel 小型蜂窝基站上配备 T3K 片上系统（system-on-chip），拥有 4 核 ARM 处理器和 2GB 的内存；而目前流行的 Hadoop、Spark 等数据分析模型利用的计算资源特点却是“数量相对较少，但资源十分丰富”，高效运行的 Spark 需要 8 核 CPU 与 8GB 内存的计算节点支持，因此 Hadoop、Spark 等数据分析框架不适应边缘计算的资源环境。现有很多计算框架适用于资源有限的计算节点，如实时流数据处理框架 Apache Edgent^[39]，深度学习框架 TensorFlow^[40]等，但仍不成熟，有很多问题需要解决。例如 Apache Edgent 只支持类似过滤，聚集这样的简单操作，而 TensorFlow 目前的版本只能用于单 PC 或单移动设备上的计算。

4.5 编程模型

边缘计算资源动态、异构与分散的特性使应用程序的开发十分困难，为减少应用的开发难度，需要可以适应边缘计算资源的编程模型。Hong^[31]等人提出了一个边缘计算编程模型，针对地理空间分布的延迟敏感的大规模应用，该模型适应分散、异构的资源环境，并使程序可以根据负载动态伸缩。但是该模型假设资源之间的网络拓扑必须是树状的，无法适应边缘计算资源的动态性。Sajjad^[32]等人研究了流处理应用的编程模型，该模型利用空间上分散的计算资源处理数据，将任务区分为本地任务和全局任务，本地任务可以在更靠近数据源的计算节点上执行，从而减少应用在网络上传输的数据量。

5 总结与展望

边缘计算利用数据传输路径上的资源为用户提供服务，作为一种新型的计算模式，边缘计算在很多应用领域都具有巨大的潜力，并对未来万物联网的趋势有着巨大的推动作用。本文介绍了边缘计算的优势，并从计算资源、数据以及与云计算的关系这三方面出发对边缘计算做出定义，也对边缘计算的相关概念进行了分析与比较；随后，介绍了边缘计算的基础设施——边缘计算平台，主要从 Cloudlet 与 ParaDrop 这两个具有代表性的平台入手，从应用领域、服务移动性、服务状态、虚拟化技术这四个方面进行了分析与比较；接着，通过列举代表性应用，分别展示出边缘计算在物联网应用和移动应用上的优势；最后，梳

理了当前边缘计算所面临的一些挑战。

致谢

本文受到国家自然科学基金与国家重点研发计划资助。同时,美国韦恩州立大学的施巍松教授对论文进行审阅并提出一些宝贵的建议。在此一并致以衷心的感谢。

参考文献

- [1] Gray J. Distributed Computing Economics[J]. Queue, 2004, 6(3):63-68.
- [2] Armbrust, Michael, Fox, et al. Above the Clouds: A Berkeley View of Cloud Computing[J]. Eecs Department University of California Berkeley, 2009, 53(4):50-58..
- [3] Evans D. The internet of things how the next evolution of the internet is changing every thing. CISCO White papers, 2011
- [4] Cisco Visual Networking. Cisco global cloud index: Forecast and methodology 2015-2020, CISCO White paper, 2015.
- [5] Ha K, Pillai P, Lewis G, et al. The Impact of Mobile Multimedia Applications on Data Center Consolidation[C]// IEEE International Conference on Cloud Engineering. IEEE, 2012:166-176.
- [6] Hu W, Gao Y, Ha K, et al. Quantifying the Impact of Edge Computing on Mobile Applications[C]// ACM Sigops Asia-Pacific Workshop on Systems. ACM, 2016:5.
- [7] Groopman J, Etlinger S. Consumer Perceptions of Privacy in the Internet of Things: What Brands Can Learn from a Concerned Citizenry. Technical report, Altimeter Group, June 2015.
- [8] Satyanarayanan M. The Emergence of Edge Computing[J]. Computer, 2017, 50(1): 30-39.
- [9] Shi W, Cao J, Zhang Q, et al. Edge computing: Vision and challenges[J]. IEEE Internet of Things Journal, 2016, 3(5): 637-646.
- [10] Satyanarayanan M, Bahl P, Caceres R, et al. The Case for VM-Based Cloudlets in Mobile Computing.[J]. IEEE Pervasive Computing, 2009, 8(4):14-23.
- [11] Bonomi F, Milito R, Zhu J, et al. Fog computing and its role in the internet of things[C]//Proceedings of the first edition of the MCC workshop on Mobile cloud computing. ACM, 2012: 13-16.
- [12] Liu P, Willis D, Banerjee S. ParaDrop: Enabling Lightweight Multi-tenancy at the Network's Extreme Edge[C]//Edge Computing (SEC), IEEE/ACM Symposium on. IEEE, 2016: 1-13.
- [13] Willis D, Dasgupta A, Banerjee S. Paradrop: a multi-tenant platform to dynamically install third party services on wireless gateways[C]//Proceedings of the 9th ACM workshop on Mobility in the evolving internet architecture. ACM, 2014: 43-48.
- [14] Satyanarayanan M. A brief history of cloud offload: A personal journey from odyssey through cyber foraging to cloudlets[J]. GetMobile: Mobile Computing and Communications, 2015, 18(4): 19-23.
- [15] Ha K, Satyanarayanan M. Openstack++ for cloudlet deployment[J]. School of Computer Science Carnegie Mellon University Pittsburgh, 2015.
- [16] Ha K, Pillai P, Richter W, et al. Just-in-time provisioning for cyber foraging[C]//Proceeding of the 11th annual international conference on Mobile systems, applications, and services. ACM, 2013: 153-166.
- [17] Ha K, Abe Y, Chen Z, et al. Adaptive vm handoff across cloudlets. Technical Report CMU-CS-15-113, CMU School of Computer Science, 2015.
- [18] Ha K, Chen Z, Hu W, et al. Towards wearable cognitive assistance[C]//Proceedings of the 12th annual international conference on Mobile systems, applications, and services. ACM, 2014: 68-81.
- [19] Simoens P, Xiao Y, Pillai P, et al. Scalable crowd-sourcing of video from mobile devices[C]//Proceeding of the 11th annual international conference on Mobile systems, applications, and services. ACM, 2013: 139-152.
- [20] Satyanarayanan M, Lewis G, Morris E, et al. The role of cloudlets in hostile environments[J]. IEEE Pervasive Computing, 2013, 12(4): 40-49.
- [21] Li D, Salonidis T, Desai N V, et al. DeepCham: Collaborative Edge-Mediated Adaptive Deep Learning for Mobile Object Recognition[C]//Edge Computing (SEC), IEEE/ACM Symposium on. IEEE, 2016: 64-76.
- [22] Tang B, Chen Z, Hefferman G, et al. A hierarchical distributed fog computing architecture for big data analysis in smart cities[C]//Proceedings of the ASE BigData & SocialInformatics 2015. ACM, 2015: 28.
- [23] Gosain A, Berman M, Brinn M, et al. Enabling Campus Edge Computing Using GENI Racks and Mobile Resources[C]//Edge Computing (SEC), IEEE/ACM Symposium on. IEEE, 2016: 41-50.
- [24] Bhardwaj K, Shih M W, Agarwal P, et al. Fast, scalable and secure onloading of edge functions using AirBox[C]//Edge Computing (SEC), IEEE/ACM Symposium on. IEEE, 2016: 14-27.
- [25] Nastic S, Truong H L, Dustdar S. A Middleware Infrastructure for Utility-based Provisioning of IoT Cloud Systems[C]//Edge Computing (SEC), IEEE/ACM Symposium on. IEEE, 2016: 28-40.
- [26] Loreto S, Mecklin T, Opsenica M, et al. Service broker architecture: Location business case and mashups[J]. IEEE Communications Magazine, 2009, 47(4):97-103.
- [27] Chaves S A D, Uriarte R B, Westphal C B. Toward an architecture for monitoring private clouds[J]. IEEE Communications Magazine, 2011, 49(12):130-137.
- [28] Povedano-Molina J, Lopez-Vega J M, Lopez-Soler J M, et al. DARGOS: A highly adaptable and scalable monitoring architecture for multi-tenant Clouds[J]. Future Generation Computer Systems, 2013, 29(8):2041-2056.
- [29] Grozev N, Buyya R. Inter-Cloud architectures and application brokering: taxonomy and survey[J]. Software Practice & Experience, 2014, 44(3):369-390.
- [30] Ahmed A, Ahmed E. A Survey on Mobile Edge Computing[C]// IEEE International Conference on Intelligent Systems and Control. IEEE, 2016.
- [31] Hong K, Lillethun D, Ramachandran U, et al. Mobile fog:a programming

- model for large-scale applications on the internet of things[C]// ACM SIGCOMM Workshop on Mobile Cloud Computing. ACM, 2013:15-20.
- [32] Sajjad H P, Danniswara K, Al-Shishtawy A, et al. SpanEdge: Towards Unifying Stream Processing over Central and Near-the-Edge Data Centers[C]//Edge Computing (SEC), IEEE/ACM Symposium on. IEEE, 2016: 168-178.
- [33] Hou Z W, Zhai H X, Gao G H. A Study on Web Services Performance Optimization[J]. Proceedings of the International Symposium on Electronic Commerc, 2010, 8(1):84 - 93.
- [34] Takahashi N, Tanaka H, Kawamura R. Analysis of Process Assignment in Multi-tier mobile Cloud Computing and Application to Edge Accelerated Web Browsing[C]// IEEE International Conference on Mobile Cloud Computing, Services, and Engineering. IEEE, 2015:233-234.
- [35] Open Edge Computing [EB/OL]. <http://openedgecomputing.org/>. 2017
- [36] LXD [EB/OL]. <https://linuxcontainers.org/lxd/>. 2017
- [37] Hyper [EB/OL]. <https://hypercontainer.io/>. 2017
- [38] Rancher OS [EB/OL]. <http://rancher.com/rancher-os/>. 2017
- [39] Apache Edgent [EB/OL]. <https://edgent.apache.org/>. 2017
- [40] TensorFlow [EB/OL]. <https://www.tensorflow.org/>. 2017
- [41] YSlow. <http://yslow.org/>. 2017
- [42] PageSpeed Tools[EB/OL]. <https://developers.google.com/speed/pagespeed/>. 2017