计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2019.20180760
56(1):69-89,2019

边缘计算:现状与展望

施巍松[1] 张星洲[2,3] 王一帆[2,3] 张庆阳[4]
[1](韦恩州立大学计算机科学系 美国密歇根州底特律 48202)
[2](中国科学院计算技术研究所 北京 100190)
[3](中国科学院大学 北京 100190)
[4](安徽大学计算机科学与技术学院 合肥 230601)
(weisong@wayne.edu)

Edge Computing: State-of-the-Art and Future Directions

Shi Weisong[1], Zhang Xingzhou[2,3], Wang Yifan[2,3], and Zhang Qingyang[4]
[1](*Department of Computer Science, Wayne State University, Detroit, MI, USA* 48202)
[2](*Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100190)
[3](*University of Chinese Academy of Sciences, Beijing* 100190)
[4](*School of Computer Science and Technology, Anhui University, Hefei* 230601)

Abstract With the burgeoning of the Internet of everything, the amount of data generated by edge devices increases dramatically, resulting in higher network bandwidth requirements. In the meanwhile, the emergence of novel applications calls for the lower latency of the network. It is an unprecedented challenge to guarantee the quality of service while dealing with a massive amount of data for cloud computing, which has pushed the horizon of edge computing. Edge computing calls for processing the data at the edge of the network and develops rapidly from 2014 as it has the potential to reduce latency and bandwidth charges, address the limitation of computing capability of cloud data center, increase availability as well as protect data privacy and security. This paper mainly discusses three questions about edge computing: where does it come from, what is the current status and where is it going? This paper first sorts out the development process of edge computing and divides it into three periods: technology preparation period, rapid growth period and steady development period. This paper then summarizes seven essential technologies that drive the rapid development of edge computing. After that, six typical applications that have been widely used in edge computing are illustrated. Finally, this paper proposes six open problems that need to be solved urgently in future development.

Key words edge computing; cloud computing; Internet of everything; function cache; edge intelligence

摘 要 随着万物互联时代的到来,网络边缘设备产生的数据量快速增加,带来了更高的数据传输带宽需求,同时,新型应用也对数据处理的实时性提出了更高要求,传统云计算模型已经无法有效应对,因此,边缘计算应运而生.边缘计算的基本理念是将计算任务在接近数据源的计算资源上运行,可以有效减小计算系统的延迟,减少数据传输带宽,缓解云计算中心压力,提高可用性,并能够保护数据安全和隐私.得益于这些优势,边缘计算从2014年以来迅速发展.旨在探讨3个问题:边缘计算从哪里来、它的现状如何、它要到哪里去.围绕这3个问题,首先梳理了边缘计算的发展历程,将其归纳为技术储备期、快速增长期和稳健发展期3个阶段,并列举了不同阶段的典型事件.随后,总结了推动边缘计算迅速发展

收稿日期:2018-11-08;修回日期:2018-12-03

的7项关键技术,并结合已经广泛采用边缘计算的6类典型应用进行了说明.最后,提出了边缘计算在未来发展中需要紧迫解决的6类问题.

关键词 边缘计算;云计算;万物互联;功能缓存;边缘智能

中图法分类号 TP391

近年来,随着万物互联时代的快速到来和无线网络的普及,网络边缘的设备数量和产生的数据都快速增长.根据IDC预测[1],到2020年全球数据总量将大于40泽字节(zettabyte, ZB),而物联网产生数据的45%都将在网络边缘处理.在这种情形下,以云计算模型为核心的集中式处理模式将无法高效处理边缘设备产生的数据.集中式处理模型将所有数据通过网络传输到云计算中心,利用云计算中心超强的计算能力来集中式解决计算和存储问题,这使得云服务能够创造出较高的经济效益.但是在万物互联的背景下,传统云计算有4个不足:

1)实时性不够.万物互联场景下应用对于实时性的要求极高.传统云计算模型下,应用将数据传送到云计算中心,再请求数据处理结果,增大了系统延迟.以无人驾驶汽车应用为例,高速行驶的汽车需要毫秒级的反应时间,一旦由于网络问题而加大系统延迟,将会造成严重后果.

2)带宽不足.边缘设备实时产生大量数据,将全部数据传输至云端造成了网络带宽的很大压力.例如,波音787每秒产生的数据超过5GB[2],但飞机与卫星之间的带宽不足以支持实时传输.

3)能耗较大.数据中心消耗了极多的能源,根据Sverdlik的研究[3],到2020年美国所有数据中心能耗将增长4%,达到730亿千瓦时,我国数据中心所消耗的电能也已经超过了匈牙利和希腊两国用电总和.随着用户应用程序越来越多,处理的数据量越来越大,能耗将会成为限制云计算中心发展的瓶颈.

4)不利于数据安全和隐私.万物互联中的数据与用户生活联系极为紧密,例如,许多家庭安装室内智能网络摄像头,视频数据传输到云端,会增加泄露用户隐私的风险.随着欧盟"通用数据保护条例"(GDPR)[4]的生效,数据安全和隐私问题对于云计算公司来说变得更加重要.

为了解决以上问题,面向边缘设备所产生海量数据计算的边缘计算模型应运而生.边缘计算是在网络边缘执行计算的一种新型计算模型[5-6],边缘计算操作的对象包括来自于云服务的下行数据和来自于万物互联服务的上行数据,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,是一个连续统(continuum).边缘计算模型和云计算模型并不是取代的关系,而是相辅相成的关系,边缘计算需要云计算中心强大的计算能力和海量存储的支持,而云计算中心也需要边缘计算中边缘设备对海量数据及隐私数据的处理.

边缘计算模型具有3个明显的优点:

1)在网络边缘处理大量临时数据,不再全部上传云端,这极大地减轻了网络带宽和数据中心功耗的压力;

2)在靠近数据生产者处做数据处理,不需要通过网络请求云计算中心的响应,大大减少了系统延迟,增强了服务响应能力;

3)边缘计算将用户隐私数据不再上传,而是存储在网络边缘设备上,减少了网络数据泄露的风险,保护了用户数据安全和隐私.

得益于这些优势,边缘计算近年来得到了迅速发展,本文首先梳理了边缘计算的发展历程,将其归纳为技术储备期、快速增长期和稳健发展期3个阶段,并列举了不同阶段的典型事件.随后,本文总结了推动边缘计算迅速发展的7项关键技术,即网络、隔离技术、体系结构、边缘操作系统、算法执行框架、数据处理平台以及安全和隐私.然后提出广泛采用边缘计算的6类典型应用:公共安全中实时数据处理、智能网联车和自动驾驶、虚拟现实、工业物联网、智能家居和智慧城市.最后,本文提出了边缘计算在未来发展中需要紧迫解决的6类问题.

1 边缘计算的发展历程

本文在谷歌学术上以"edge computing"为关键词进行搜索每年的文章数量,结果如图1所示.可以看到,2015年以前,边缘计算处于原始技术积累阶段;2015—2017年,边缘计算开始被业内熟知,与之相关的论文增长了10余倍,得到了飞速发展;2018年边缘计算开始稳健发展(其中2018年的论文数量根据2018年前9个月的数据推算而来,实际数据有出入).本文依据这一分析结果和对行业发展趋势的

观察将边缘计算的发展分为3个阶段:技术储备期、快速增长期和稳定发展期.图2列举了边缘计算发展中的典型事件(粗体字为中国对边缘计算发展的贡献).

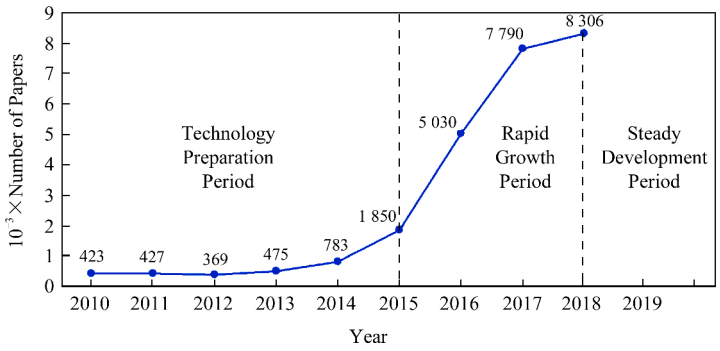

Fig. 1 Number of papers retrieved by "edge computing" on Google Scholar

图1 谷歌学术上以"edge computing"为关键词搜索到的文章数量

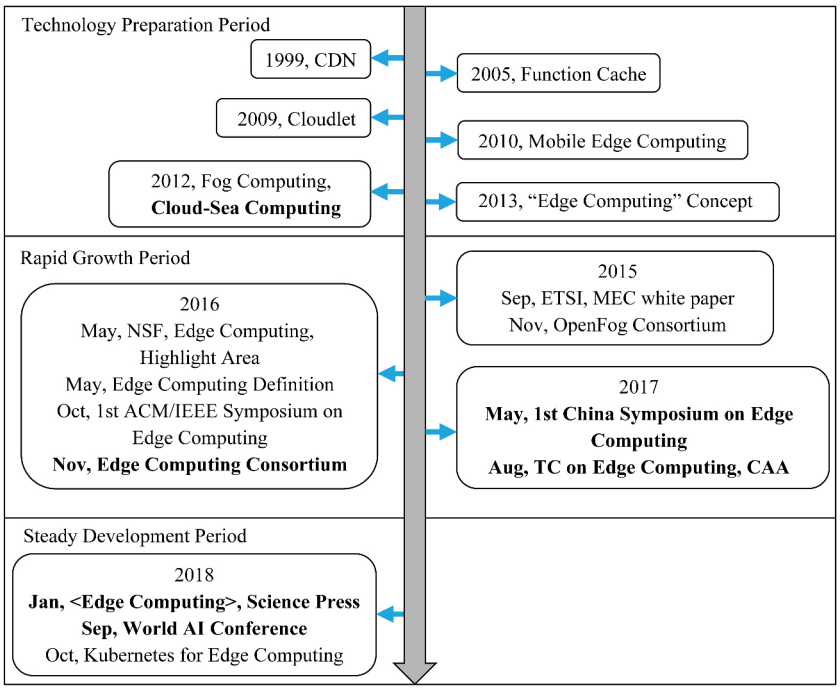

Fig. 2 Development states of edge computing and the typical events at each stage

图2 边缘计算的发展历程及典型事件

1.1 技术储备期

在此阶段,边缘计算历经"蛰伏—提出—定义—推广"等发展过程.边缘计算最早可以追溯至1998年阿卡迈(Akamai)公司提出的内容分发网络[7](content delivery network, CDN),CDN是一种基于互联网的缓存网络,依靠部署在各地的缓存服务器,通过中心平台的负载均衡、内容分发、调度等功能模块,将用户的访问指向距离最近的缓存服务器上,以此降低网络拥塞,提高用户访问响应速度和命中率.CDN强调内容(数据)的备份和缓存,而边缘计算的基本思想则是功能缓存(function cache).2005年美国韦恩州立大学施巍松教授的团队就已提出功能缓存的概念,并将其用在个性化的邮箱管理服务中,以节省延迟和带宽[8].2009年Satyanarayanan等人提出了Cloudlet[9]的概念,Cloudlet是一个可信且资源丰富的主机,部署在网络边缘,与互联网连接,可以被移动设备访问,为其提供服务,Cloudlet可以像云一样为用户提供服务,又被称为"小朵云".此时的边缘计算强调下行,即将云服务器上的功能下行至边缘服务器,以减少带宽和时延.

随后,在万物互联的背景下,边缘数据迎来了爆发性增长,为了解决面向数据传输、计算和存储过程

中的计算负载和数据传输带宽的问题,研究者开始探索在靠近数据生产者的边缘增加数据处理的功能,即万物互联服务功能的上行.具有代表性的是移动边缘计算(mobile edge computing, MEC)、雾计算(fog computing)和海云计算(cloud-sea computing).

移动边缘计算[10]是指在接近移动用户的无线接入网范围内,提供信息技术服务和云计算能力的一种新的网络结构,并已成为一种标准化、规范化的技术.由于移动边缘计算位于无线接入网内,并接近移动用户,因此可以实现较低延时、较高带宽来提高服务质量和用户体验.移动边缘计算强调在云计算中心与边缘计算设备之间建立边缘服务器,在边缘服务器上完成终端数据的计算任务,但移动边缘终端设备基本被认为不具有计算能力,而边缘计算模型中的终端设备具有较强的计算能力,因此移动边缘计算类似一种边缘计算服务器的架构和层次,作为边缘计算模型的一部分.思科公司于2012年提出了雾计算[11],并将雾计算定义为迁移云计算中心任务到网络边缘设备执行的一种高度虚拟化计算平台.它通过减少云计算中心和移动用户之间的通信次数,以缓解主干链路的带宽负载和能耗压力.雾计算和边缘计算具有很大的相似性,但是雾计算关注基础设施之间的分布式资源共享问题,而边缘计算除了关注基础设施之外,也关注边缘设备,包括计算、网络和存储资源的管理,以及边端、边边和边云之间的合作.与此同时,2012年,中国科学院启动了战略性先导研究专项,称之为下一代信息与通信技术倡议(Next Generation Information and Communication Tech-nology initiative, NICT倡议),其主旨是开展"海云计算系统项目"的研究[12],其核心是通过"云计算"系统与"海计算"系统的协同与集成,增强传统云计算能力,其中,"海"端指由人类本身、物理世界的设备和子系统组成的终端.与边缘计算相比,海云计算关注"海"和"云"这两端,而边缘计算关注从"海"到"云"数据路径之间的任意计算、存储和网络资源.

2013年,美国太平洋西北国家实验室的Ryan LaMothe在一个2页纸的内部报告中提出"edge computing"一词,这是现代"edge computing"的首次提出[13].此时,边缘计算的涵义已经既有云服务功能的下行,还有万物互联服务的上行.

1.2 快速增长期

2015—2017年为边缘计算快速增长期,在这段时间内,由于边缘计算满足万物互联的需求,引起了国内外学术界和产业界的密切关注.

在政府层面上,2016年5月,美国自然科学基金委(National Science Foundation, NSF)在计算机系统研究中将边缘计算替换云计算,列为突出领域(highlight area);8月,NSF和英特尔专门讨论针对无线边缘网络上的信息中心网络(NSF/Intel partnership on ICN in Wireless Edge Networks, ICN-WEN)[14];10月,NSF举办边缘计算重大挑战研讨会(NSF Workshop on Grand Challenges in Edge Computing)[15],会议针对3个议题展开研究:边缘计算未来5～10年的发展目标,达成目标所带来的挑战,学术界、工业界和政府应该如何协同合作来应对挑战.这标志着边缘计算的发展已经在美国政府层面上引起了重视.

在学术界,2016年5月,美国韦恩州立大学施巍松教授团队给出了边缘计算的一个正式定义[6]:边缘计算是指在网络边缘执行计算的一种新型计算模型,边缘计算操作的对象包括来自于云服务的下行数据和来自于万物互联服务的上行数据,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,是一个连续统.并发表了"Edge Computing: Vision and Challenges"一文,第1次指出了边缘计算所面临的挑战[6],该文在2018年底被他引650次.同年10月,ACM和IEEE开始联合举办边缘计算顶级会议(ACM/IEEE Symposium on Edge Computing, SEC)[16],这是全球首个以边缘计算为主题的科研学术会议.自此之后,ICDCS, INFOCOM, MiddleWare, WWW等重要国际会议也开始增加边缘计算的分会(track)或者专题研讨会(workshop).

工业界也在努力推动边缘计算的发展,2015年9月,欧洲电信标准化协会(ETSI)发表关于移动边缘计算的白皮书[10],并在2017年3月将移动边缘计算行业规范工作组正式更名为多接入边缘计算(multi-access edge computing, MEC)[17],致力于更好地满足边缘计算的应用需求和相关标准制定.2015年11月,思科、ARM、戴尔、英特尔、微软和普林斯顿大学联合成立了OpenFog联盟[18],主要致力于Fog Reference Architecture的编写.为了推进和应用场景在边缘的结合,该组织于2018年12月并入了工业互联网联盟.

国内边缘计算的发展速度和世界几乎同步,特别是从智能制造的角度.2016年11月,华为技术有限公司、中国科学院沈阳自动化研究所、中国信息通信研究院、英特尔、ARM等在北京成立了边缘计算

产业联盟(edge computing consortium)[19],致力于推动"政产学研用"各方产业资源合作,引领边缘计算产业的健康可持续发展.2017 年 5 月首届中国边缘计算技术研讨会在合肥开幕,同年 8 月中国自动化学会边缘计算专委会成立,标志着边缘计算的发展已经得到了专业学会的认可和推动.

1.3 稳健发展期

2018 年是边缘计算发展过程中的重要节点,尽管此前业内已经对边缘计算报以了很大期望,而 2018 年边缘计算被推向前台,开始被大众熟知.这一阶段,边缘计算的参与者范围扩大很快,如表 1 所示,参与者已经基本涵盖了计算机领域的方方面面,本文将它们分为 6 类:云计算公司、硬件厂商、CDN 公司、通信运营商、科研机构和产业联盟/开源社区,并在表 1 中列举它们近 2 年在边缘计算领域的事件.

Table 1 Key Players of Edge Computing and the Current Events

表 1 边缘计算主要参与者及其近期事件

Key Players	Data	Companies/ Organizations	Events
Cloud Companies	2017	Amazon	"Greengrass" software was published to support edge machine learning[20].
	2017	Google	"Cloud IoT Core" service was published to manage edge devices[21].
	2018	Google	Edge TPU was published to run inference at the edge[22].
	2018	Microsoft	"Azure IoT Edge" service was published in the Microsoft Build conference[23].
	2018-03	Alibaba	IoT edge computing product "Link Edge" was published[24].
	2018	Baidu	"Intelligence Edge" solution was published[25].
Hardware Providers	2018	ARM	"Trillium" drived machine learning capabilities for edge devices[26].
	2018	AMD	EPYC 3000 and Ryzen V1000 was published to target the edge scenario[27].
	2017	Cisco	"Kinetic" and "Jasper" was published in Cisco Live conference[28].
	2018-02	Intel	Next-generation Xeon D processor for edge computing environments was published[29].
	2018	VMware	Internet of Things strategy with new edge computing solution was published[30].
	2017	Huawei	"EC-IOT" solution for edge computing was published[31].
CDN Companies	2018	Akamai	IoT Edge Connect solution was drived[32].
	2017	CloudFlare	"CloudFlare Workers" opened edge computing by micro-services[33].
	2018	Limelight	Enhanced version of "EdgePrism" OS allowed operation on the edge[34].
	2017	ChinaNetCenter	Edge computing micro-services and edge IaaS and PaaS services was launched[35].
Communication Carriers	2018-10	China Mobile	Open Computing Laboratory for Edge Computing was established[36].
	2018-06	China Unicom	Edge cloud ecology partner conference was held[37].
	2018-02	AT&T	Edge computing test zone was established in Palo Alto[38].
	2018-01	Deutsche Telekom	Edge computing service MobileedgeX was established[39].
Research Institutions	2018-01	China Academy of Information and Communications Technology	"Internet of Things Edge Computing" international standard was led on ITU-T SG20[40].
	2017-08	Chinese Association of Automation	Edge Computing Technical Committee was established[41].
	2018-10	China Institute of Communications	"2018 Edge Computing Technology Summit" was held[42].
Consortiums/ Open Source Communities	2016	Edge Computing Consortium	The Third ACM/IEEE Symposium on Edge Computing was established in 2016 and sponsored in 2018[19].
	2017-12	Avnu Alliance	Avnu Alliance was collaborated with ECC to promote the development of edge computing[43].
	2018	Automotive Edge Computing Consortium (AECC)	AECC was established in 2018 to drive the network and computing infrastructure needs of automotive big data[44].
	2018	Edgecross Consortium	Edgecross Consortium was established in 2017, and launched edgecross to provide basic edge computing software services in 2018[45].
	2018-10	CNCF, Eclipse Foundation	CNCF and Eclipse Foundation pushed Kubernetes to the edge[46].

边缘计算在稳健发展期有4个重要事件,2018年1月全球首部边缘计算专业书籍《边缘计算》出版[47],它从边缘计算的需求与意义、系统、应用、平台等多个角度对边缘计算进行了阐述.2018年9月17日在上海召开的世界人工智能大会,以"边缘计算,智能未来"为主题举办了边缘智能主题论坛[48],这是中国从政府层面上对边缘计算的发展进行了支持和探讨.2018年8月两年一度的全国计算机体系结构学术年会以"由云到端的智能架构"为主题[49],由此可见,学术界的研究焦点已经由云计算开始逐渐转向边缘计算.同时,边缘计算也得到了技术社区的大力支持,具有代表性的是:2018年10月CNCF基金会和Eclipse基金会展开合作,将把在超大规模云计算环境中已被普遍使用的Kubernetes,带入到物联网边缘计算场景中.新成立的Kubernetes物联网边缘工作组将采用运行容器的理念并扩展到边缘,促进Kubernetes在边缘环境中的适用[46].

本文相信,经过前期的技术储备和最近几年的快速增长,边缘计算将成为学术界和产业界的热门话题,实现学术界与工业界的融合,加快产品落地,便利大众生活,步入稳健发展时期.

2 支持边缘计算的核心技术

计算模型的创新带来的是技术的升级换代,而边缘计算的迅速发展也得益于技术的进步.本节总结了推动边缘计算发展的7项核心技术,它们包括网络、隔离技术、体系结构、边缘操作系统、算法执行框架、数据处理平台以及安全和隐私.

2.1 网 络

边缘计算将计算推至靠近数据源的位置,甚至于将整个计算部署于从数据源到云计算中心的传输路径上的节点,这样的计算部署对现有的网络结构提出了3个新的要求:

1) 服务发现.在边缘计算中,由于计算服务请求者的动态性,计算服务请求者如何知道周边的服务,将是边缘计算在网络层面中的一个核心问题.传统的基于DNS的服务发现机制[50],主要应对服务静态或者服务地址变化较慢的场景下.当服务变化时,DNS的服务器通常需要一定的时间以完成域名服务的同步,在此期间会造成一定的网络抖动,因此并不适合大范围、动态性的边缘计算场景.

2) 快速配置.在边缘计算中,由于用户和计算设备的动态性的增加,如智能网联车[51],以及计算设备由于用户开关造成的动态注册和撤销[52],服务通常也需要跟着进行迁移,而由此将会导致大量的突发网络流量.与云计算中心不同,广域网的网络情况更为复杂,带宽可能存在一定的限制.因此,如何从设备层支持服务的快速配置,是边缘计算中的一个核心问题.

3) 负载均衡.边缘计算中,边缘设备产生大量的数据,同时边缘服务器提供了大量的服务.因此,根据边缘服务器以及网络状况,如何动态地对这些数据进行调度至合适的计算服务提供者,将是边缘计算中的核心问题.

针对以上3个问题,一种最简单的方法是,在所有的中间节点上均部署所有的计算服务,然而这将导致大量的冗余,同时也对边缘计算设备提出了较高的要求.因此,我们以"建立一条从边缘到云的计算路径"为例来说,首当其冲面对的就是如何寻找服务,以完成计算路径的建立.命名数据网络(named data networking, NDN)[53]是一种将数据和服务进行命名和寻址,以P2P和中心化方式相结合进行自组织的一种数据网络.而计算链路的建立,在一定程度上也是数据的关联建立,即数据应该从源到云的传输关系.因此,将NDN引入边缘计算中,通过其建立计算服务的命名并关联数据的流动,从而可以很好地解决计算链路中服务发现的问题.

而随着边缘计算的兴起,尤其是用户移动的情况下,如车载网络,计算服务的迁移相较于基于云计算的模式更为频繁,与之同时也会引起大量的数据迁移,从而对网络层面提供了动态性的需求.软件定义网络(software defined networking, SDN)[54-55],于2006年诞生于美国GENI项目资助的斯坦福大学Clean Slate课题,是一种控制面和数据面分离的可编程网络,以及简单网络管理.由于控制面和数据面分离这一特性,网络管理者可以较为快速地进行路由器、交换器的配置,减少网络抖动性,以支持快速的流量迁移,因此可以很好地支持计算服务和数据的迁移.同时,结合NDN和SDN,可以较好地对网络及其上的服务进行组织,并进行管理,从而可以初步实现计算链路的建立和管理问题.

2.2 隔离技术

隔离技术是支撑边缘计算稳健发展的重要研究技术,边缘设备需要通过有效的隔离技术来保证服务的可靠性和服务质量.隔离技术需要考虑2方面:1)计算资源的隔离,即应用程序间不能相互干扰;2)数据的隔离,即不同应用程序应具有不同的访问

权限.在云计算场景下,由于某一应用程序的崩溃可能带来整个系统的不稳定,造成严重的后果,而在边缘计算下,这一情况变得更加复杂.例如在自动驾驶操作系统中,既需要支持车载娱乐满足用户需求,又需要同时运行自动驾驶任务满足汽车本身驾驶需求,此时,如果车载娱乐的任务干扰了自动驾驶任务,或者影响了整个操作系统的性能,将会引起严重后果,对生命财产安全造成直接损失.隔离技术同时需要考虑第三方程序对用户隐私数据的访问权限问题,例如,车载娱乐程序不应该被允许访问汽车控制总线数据等.目前在云计算场景下主要使用 VM 虚拟机和 Docker 容器技术等方式保证资源隔离.边缘计算可汲取云计算发展的经验,研究适合边缘计算场景下的隔离技术.

在云平台上普遍应用的 Docker 技术可以实现应用在基于 OS 级虚拟化的隔离环境中运行,Docker 的存储驱动程序采用容器内分层镜像的结构,使得应用程序可以作为一个容器快速打包和发布,从而保证了应用程序间的隔离性.Li 等人建立了一个基于 Docker 迁移的有效服务切换系统[56],利用 Docker 的分层文件系统支持,提出了一种适合边缘计算的高效容器迁移策略,以减少包括文件系统、二进制内存映象、检查点在内的数据传输的开销.Ha 等人提出了一种 VM 切换技术[57],实现虚拟机 VM 的计算任务迁移,支持快速和透明的资源放置,保证将 VM 虚拟机封装在安全性和可管理行要求较高的应用中.这种多功能原语还提供了动态迁移的功能,对边缘端进行了优化.这种基于 VM 的隔离技术提高了应用程序的抗干扰性,增加了边缘计算系统的可用性.

2.3 体系结构

无论是如高性能计算一类传统的计算场景,还是如边缘计算一类的新兴计算场景,未来的体系结构应该是通用处理器和异构计算硬件并存的模式[58].异构硬件牺牲了部分通用计算能力,使用专用加速单元减小了某一类或多类负载的执行时间,并且显著提高了性能功耗比[59-61].边缘计算平台通常针对某一类特定的计算场景设计,处理的负载类型较为固定,故目前有很多前沿工作针对特定的计算场景设计边缘计算平台的体系结构.

ShiDianNao[62]首次提出了将人工智能处理器放置在靠近图像传感器的位置,处理器直接从传感器读取数据,避免图像数据在 DRAM 中的存取带来的能耗开销;同时通过共享卷积神经网络(convolutional neural networks, CNNs)权值的方法,将模型完整放置在 SRAM 中,避免权值数据在 DRAM 中的存取带来的能耗开销;由于计算能效地大幅度提升(60 倍),使其可以被应用于移动端设备.EIE[63]是一个用于稀疏神经网络的高效推理引擎,其通过稀疏矩阵的并行化以及权值共享的方法加速稀疏神经网络在移动设备的执行能效.Phi-Stack[64]则提出了针对边缘计算的一整套技术栈,其中针对物联网设备设计的 PhiPU,使用异构多核的结构并行处理深度学习任务和普通的计算任务(实时操作系统).In-Situ AI[65]是一个用于物联网场景中深度学习应用的自动增量计算框架和架构,其通过数据诊断,选择最小数据移动的计算模式,将深度学习任务部署到物联网计算节点.除了专用计算硬件的设计,还有一类工作探索 FPGA 在边缘计算场景中的应用.ESE[66]通过 FPGA 提高了稀疏长短时记忆网络(long short term memory network, LSTM)在移动设备上的执行能效,用于加速语音识别应用.其通过负载平衡感知的方法对 LSTM 进行剪枝压缩,并保证硬件的高利用率,同时在多个硬件计算单元中调度 LSTM 数据流;其使用 Xilinx XCKU060 FPGA 进行硬件设计实现,与 CPU 和 GPU 相比,其分别实现了 40 倍和 11.5 倍的能效提升.Biookaghazadeh 等人通过对比 FPGA 和 GPU 在运行特定负载时吞吐量敏感性、结构适应性和计算能效等指标,表明 FPGA 更加适合边缘计算场景[67].

针对边缘计算的计算系统结构设计仍然是一个新兴的领域,仍然具有很多挑战亟待解决,例如如何高效地管理边缘计算异构硬件、如何对这类的系统结构进行公平及全面的评测等.在第三届边缘计算会议(SEC 2018)上首次设立了针对边缘计算体系结构的 Workshop:ArchEdge,鼓励学术界和工业界对此领域进行探讨和交流.

2.4 边缘操作系统

边缘计算操作系统向下需要管理异构的计算资源,向上需要处理大量的异构数据以及多用的应用负载,其需要负责将复杂的计算任务在边缘计算节点上部署、调度及迁移,从而保证计算任务的可靠性以及资源的最大化利用.与传统的物联网设备上的实时操作系统 Contiki[68]和 FreeRTOS[69]不同,边缘计算操作系统更倾向于对数据、计算任务和计算资源的管理框架.

机器人操作系统(robot operating system, ROS)[70]最开始被设计用于异构机器人机群的消息通信管理,

现逐渐发展成一套开源的机器人开发及管理工具，提供硬件抽象和驱动、消息通信标准、软件包管理等一系列工具，被广泛应用于工业机器人、自动驾驶车辆即无人机等边缘计算场景. 为解决 ROS 中的性能问题，社区在 2015 年推出 ROS2.0[71]，其核心为引入数据分发服务(data distribution service, DDS)，解决 ROS 对主节点(master node)性能依赖问题，同时 DDS 提供共享内存机制提高节点间的通信效率. $EdgeOS_H$ 则是针对智能家居设计的边缘操作系统[52]，其部署于家庭的边缘网关中，通过 3 层功能抽象连接上层应用和下层智能家居硬件，其提出面向多样的边缘计算任务，服务管理层应具有差异性(differentiation)、可扩展性(extensibility)、隔离性(isolation)和可靠性(reliability)的需求. Phi-Stack 中提出了面向智能家居设备的边缘操作系统 PhiOS[64]，其引入轻量级的 REST 引擎和 LUA 解释器，帮助用户在家庭边缘设备上部署计算任务. OpenVDAP[72]是针对汽车场景设计的数据分析平台，其提出了面向网联车场景的边缘操作系统 $EdgeOS_v$. 该操作系统中提供了任务弹性管理、数据共享以及安全和隐私保护等功能.

根据目前的研究现状，ROS 以及基于 ROS 实现的操作系统有可能会成为边缘计算场景的典型操作系统，但其仍然需要经过在各种真实计算场景下部署的评测和检验.

2.5 算法执行框架

随着人工智能的快速发展，边缘设备需要执行越来越多的智能算法任务，例如家庭语音助手需要进行自然语言理解、智能驾驶汽车需要对街道目标检测和识别、手持翻译设备需要翻译实时语音信息等. 在这些任务中，机器学习尤其是深度学习算法占有很大的比重，使硬件设备更好地执行以深度学习算法为代表的智能任务是研究的焦点，也是实现边缘智能的必要条件. 而设计面向边缘计算场景下的高效的算法执行框架是一个重要的方法. 目前有许多针对机器学习算法特性而设计的执行框架，例如谷歌于 2016 年发布的 TensorFlow[73]、依赖开源社区力量发展的 Caffe 等[74]，但是这些框架更多地运行在云数据中心，它们不能直接应用于边缘设备. 如表 2 所示，云数据中心和边缘设备对算法执行框架的需求有较大的区别. 在云数据中心，算法执行框架更多地执行模型训练的任务，它们的输入是大规模的批量数据集，关注的是训练时的迭代速度、收敛率和框架的可扩展性等. 而边缘设备更多地执行预测任务，输入的是实时的小规模数据，由于边缘设备计算资源和存储资源的相对受限性，它们更关注算法执行框架预测时的速度、内存占用量和能效.

Table 2 Comparison of Frameworks on Cloud and Edge
表 2 云数据中心和边缘设备的算法执行框架比较

Factors	Cloud Servers	Edge Devices
Input	Large-scale, patch	Small-scale, real-time
Task	Train, inference	Inference
Concerns	Training Speed Convergence Rate Scalability	Inference Latency Memory Resource Usage Energy Efficiency

为了更好地支持边缘设备执行智能任务，一些专门针对边缘设备的算法执行框架应运而生. 2017 年，谷歌发布了用于移动设备和嵌入式设备的轻量级解决方案 TensorFlow Lite[75]，它通过优化移动应用程序的内核、预先激活和量化内核等方法来减少执行预测任务时的延迟和内存占有量. Caffe2[76]是 Caffe 的更高级版本，它是一个轻量级的执行框架，增加了对移动端的支持. 此外，PyTorch[77]和 MXNet 等[78]主流的机器学习算法执行框架也都开始提供在边缘设备上的部署方式.

Zhang 等人[79]对 TensorFlow, Caffe2, MXNet, PyTorch 和 TensorFlow Lite 等在不同的边缘设备(MacBook Pro, Intel FogNode, NVIDIA Jetson TX2, Raspberry Pi 3 Model B+, Huawei Nexus 6P)上的性能从延迟、内存占用量和能效等方面进行了对比和分析，最后发现没有一款框架能够在所有维度都取得最好的表现，因此执行框架的性能提升空间比较大. 开展针对轻量级的、高效的、可扩展性强的边缘设备算法执行框架的研究十分重要，也是实现边缘智能的重要步骤.

2.6 数据处理平台

边缘计算场景下，边缘设备时刻产生海量数据，数据的来源和类型具有多样化的特征，这些数据包括环境传感器采集的时间序列数据、摄像头采集的图片视频数据、车载 LiDAR 的点云数据等，数据大多具有时空属性. 构建一个针对边缘数据进行管理、分析和共享的平台十分重要.

以智能网联车场景为例，车辆逐渐演变成一个移动的计算平台，越来越多的车载应用也被开发出来，车辆的各类数据也比较多. 由 Zhang 等人提出的 OpenVDAP[72]是一个开放的汽车数据分析平台，如图 3 所示，OpenVDAP 分成 4 部分，分别是异构

计算平台(VCU)、操作系统($EdgeOS_v$)、驾驶数据收集器(DDI)和应用程序库(libvdap),汽车可安装部署该平台,从而完成车载应用的计算,并且实现车与云、车与车、车与路边计算单元的通信,从而保证了车载应用服务质量和用户体验.因此,在边缘计算不同的应用场景下,如何有效地管理数据、提供数据分析服务,保证一定的用户体验是一个重要的研究问题.

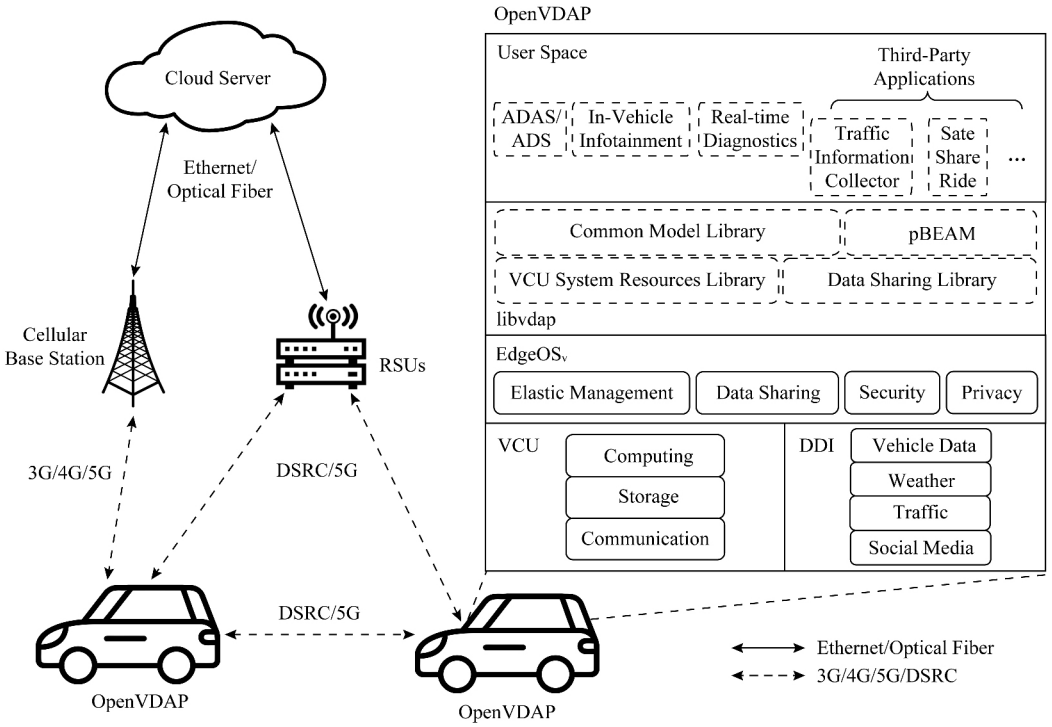

Fig. 3 The architecture of OpenVDAP

图3 OpenVDAP架构图

2.7 安全和隐私

虽然边缘计算将计算推至靠近用户的地方,避免了数据上传到云端,降低了隐私数据泄露的可能性[80].但是,相较于云计算中心,边缘计算设备通常处于靠近用户侧,或者传输路径上,具有更高的潜在可能被攻击者入侵,因此,边缘计算节点自身的安全性仍然是一个不可忽略的问题.边缘计算节点的分布式和异构型也决定了其难以进行统一的管理,从而导致一系列新的安全问题和隐私泄露等问题.作为信息系统的一种计算模式,边缘计算也存在信息系统普遍存在的共性安全问题,包括:应用安全、网络安全、信息安全和系统安全等.

在边缘计算的环境下,通常仍然可以采用传统安全方案来进行防护,如通过基于密码学的方案来进行信息安全的保护、通过访问控制策略来对越权访问等进行防护.但是需要注意的是,通常需要对传统方案进行一定的修改,以适应边缘计算的环境.同时,近些年也有一些新兴的安全技术(如硬件协助的可信执行环境)可以使用到边缘计算中,以增强边缘计算的安全性.此外,使用机器学习来增强系统的安全防护也是一个较好的方案.

可信执行环境(trusted execution environment, TEE)[81-82]是指在设备上一个独立于不可信操作系统而存在的可信的、隔离的、独立的执行环境,为不可信环境中的隐私数据和敏感计算提供了一个安全而机密的空间,而TEE的安全性通常通过硬件相关的机制来保障.常见的TEE包括Intel软件防护扩展[83-85]、Intel管理引擎[86]、x86系统管理模式[87]、AMD内存加密技术[88]、AMD平台安全处理器[89]和ARM TrustZone技术[90].通过将应用运行于可信执行环境中,并且将使用到的外部存储进行加解密[91],边缘计算节点的应用,可以在边缘计算节点被攻破时,仍然可以保证应用及数据的安全性.

3 边缘计算的典型应用

得益于第2节的7项核心技术的发展,边缘计算在许多应用场景下取得了好的效果.本节给出已经

基于边缘计算模型设计的6个成功典型应用,通过这些应用来发现边缘计算的研究机遇和挑战,并探讨更多的应用场景.

3.1 公共安全中实时数据处理

公共安全从社会的方方面面,如消防、出行,影响着广大民众的生活.随着智慧城市和平安城市的建设,大量传感器被安装到城市的各个角落,提升公共安全.例如武汉的"雪亮工程"建设,计划到2019年6月底,全市公共安全视频监控总量将达到150万个.得益于"雪亮工程"的建设,全市刑事有效警情同比下降27.2%,并为群众查找走失老人小孩、追回遗失贵重物品等服务1万余次[92].随着共享经济的兴起,各种共享经济产品落地并得到发展,如滴滴、Uber和共享单车.然而,这些产品同时也存在大量的公共安全事件.例如顺风车司机对乘客进行骚扰,甚至发生刑事案件.因此,2018年9月受顺风车安全事件的影响,滴滴已经临时下线顺风车业务并进行整改,首当其冲的是在司机端加入服务时间段的自动录音功能.然而,想要进一步提升安全性,最终还是得依赖于视频等技术,然而这将导致大量的带宽需求.按照Uber 2017年的使用情况[93](45 787次/分钟),假设将每次驾乘的视频发送至云端(每次20分钟),每天云端将新增9.23 PB的视频数据.边缘计算作为近数据源计算,可以大量地降低数据带宽,将可以用来解决公共安全领域视频数据处理的问题[94].

虽然当前城市中部署了大量的IP摄像头,但是大部分摄像头都不具备前置的计算功能,而需要将数据传输至数据中心进行处理,或者需要人工的方式来进行数据筛选.Sun等人提出了一种基于边缘计算的视频有用性检测系统[95],其可以通过在前端或者靠近视频源的位置,对视频内容进行判断,从而检测摄像头故障、内容错误以及根据内容对视频质量进行动态调整.Zhang等人受启发于安珀警报系统,基于边缘计算技术,开发了安珀警报助手(Amber alert assistant, A3)[96],其可以自动化地在边缘设备上部署视频分析程序,并与附近的边缘设备协同实时地对视频进行处理,同时和周边摄像头进行联动,以完成绑匪车辆的实时追踪.

针对滴滴等共享车辆服务近年发生的危害公共安全的事件,Liu等人提出了SafeShareRide[97]系统,其会在两者情况下触发视频报警功能——司机驾驶行为异常,如偏离轨道和车内发生争吵或者口头呼救.SafeShareRide系统通过将用户手机作为边缘端,实时地监控车内情况和司机情况,做到数据的预先处理,避免了安全时间段内的视频上传,从而大量地降低了流量的损耗.

以上工作主要针对系统的有效性,更多地关注民众安全.而保护维护公共安全的人员,如警察、消防员等的安全,也至关重要.Wu等人提出了一种适用于消防系统的边缘计算系统[98].其通过在救火车上部署边缘服务器,接受消防员配备的红外摄像头数据和各种传感器数据(如室内定位系统),实时地处理获得消防员位置信息和周边情况,并可视化地展现给现场指挥,同时也推送给远程控制中心,以保障消防员的人身安全.

3.2 智能网联车和自动驾驶

随着机器视觉、深度学习和传感器等技术的发展,汽车的功能不再局限于传统的出行和运输工具,而是逐渐变为一个智能的、互联的计算系统,我们称这样新型的汽车为智能网联车(connected and autonomous vehicles, CAVs).智能网联车的出现催生出了一系列新的应用场景,例如自动驾驶[99]、车联网[100]以及智能交通[101].Intel在2016年的报告指出[102],一辆自动驾驶车辆一天产生的数据为4 TB,这些数据无法全部上传至云端处理,需要在边缘节点(汽车)中存储和计算.

自动驾驶计算场景无疑是目前最热的研究方向之一,围绕此场景有经典的自动驾驶算法评测数据集KITTI[99,103],还有针对不同自动驾驶阶段的经典的视觉算法[104-106].在工业界目前有许多针对CAVs场景推出的计算平台,例如NVIDIA DRIVE PX2[107]和Xilinx Zynq UltraScale+ZCU106[108].同时,学术界有许多前沿工作也开始探索CAVs场景下的边缘计算平台的系统设计.Liu等人将自动驾驶分为传感(sensing)、感知(perception)和决策(decision-making)3个处理阶段,并比较3个阶段在不同异构硬件上的执行效果,由此总结除了自动驾驶任务与执行硬件之间的匹配规则[109].Lin等人对比了感知阶段3个核心应用,即定位(localization)、识别(detection)和追踪(tracking)在GPUs,FPGAs和ASICs不同组合运行的时延和功耗,指导研究人员设计端到端的自动驾驶计算平台[110].除了硬件系统结构设计,还有一类研究推出完整的软件栈帮助研究人员实现自动驾驶系统,例如百度的Apollo[111]和日本早稻田大学的Autoware[112].

如2.6节所述,OpenVDAP是一个开放的车载数据分析平台,其提供了车载计算平台、操作系统、函数库等全栈的车载数据计算服务.除了自动驾驶,

OpenVDAP 中还总结了 3 类智能网联车应用中的典型计算场景,分别是实时诊断、车载娱乐和第三方应用. 前 2 个计算场景目前主要被工业界所关注,而学术界有很多在车载第三方应用中使用边缘计算技术的研究工作,例如利用车上设备实时检测异常驾驶行为[113],根据司机行为判断司机身份的 PreDriveID[114],通过分析车辆行驶行为数据、车内音频数据和手机摄像头数据保证出租车内司机和乘客安全的 SafeShareRide 等[97].

3.3 虚拟现实

虚拟现实(virtual reality, VR)和增强现实(augment reality, AR)技术的出现彻底改变了用户与虚拟世界的交互方式. 为保证用户体验,VR/AR 的图片渲染需要具有很强的实时性. 研究表明:将 VR/AR 的计算任务卸载到边缘服务器或移动设备,可以降低平均处理时延[115]. MUVR[116]是一个在边缘服务器上支持多用户 VR 程序的处理框架,其将 VR 图像渲染卸载到边缘服务器,并尝试重用用户之前的 VR 图像帧,以降低边缘服务器的计算和通信负担. Furion[117]是一个移动端 VR 框架,其将 VR 负载分为前景交互和背景环境 2 种,前景交互依然在云端处理,而背景环境渲染卸载到移动端处理,由此实现移动设备上的高质量的 VR 应用. Ha 等人设计了一个基于 VR 与边缘计算的可穿戴认知助手,Google Glass 用于数据收集和接受及显示 VR 图像;图片渲染、人脸识别等计算任务在 Cloudlet(边缘节点)中执行,有效地解决了可穿戴设备电池容量以及处理能力有限的问题[118].

3.4 工业物联网

工业互联网是机器、计算机和人员使用业务转型所取得的先进的数据分析成果来实现智能化的工业操作. 但是在工业物联网领域的应用实践中,对于工业实时控制及边缘设备安全隐私的要求较高,并且产生的数据需要本地化处理,因此将边缘计算应用于工业物联网成为了行业发展的方向. 2018 年,工业互联网联盟(IIC)正式发布了《工业物联网边缘计算介绍》白皮书[119],旨在阐述边缘计算对于工业物联网应用的价值,并总结了工业互联网边缘计算模型的独特性和从云到边缘计算的关键驱动力.

边缘计算应用于工业物联网有 3 个优势:

1) 改善性能,工业生产中常见的报警、分析等应用靠近数据生产者的地方处理和决策会更快,通过减少与云数据中心的通信可以增加边缘处理的弹性.

2) 保证数据安全和隐私,可以避免数据传输到共享数据中心后数据暴露等带来的安全隐私问题.

3) 减少操作成本,通过在边缘做计算处理,可以减少边缘设备和数据中心的数据传输量和带宽,从而减少了工业生产中由网络、云数据中心计算和存储带来的成本.

Chen 等人用边缘计算技术对薄膜壁焊接的工业级机器人系统做优化[120],设计了物理资源-边缘-云的架构,实验结果表明:该系统比基于云计算的传统系统实时性更好,并且最多可节省 883.38 Kbps 的带宽,满足了工业级产品的需求.

3.5 智能家居

随着物联网技术的发展,智能家居系统得到进一步的发展,其利用大量的物联网设备(如温湿度传感器、安防系统、照明系统)实时监控家庭内部状态,接受外部控制命令并最终完成对家居环境的调控,以提升家居安全性、便利性、舒适性. Berg Insight 的调查报告显示[121],欧美和北美洲的智能家居数据将在 2019 年达到 6 800 万. 然而,随着智能家居设备的越来越多,且这些设备通常都是异构的[122],如何管理这些异构设备将会是一个亟待解决的问题[123],如设备的命名、数据的命名以及设备的智能化联动. 并且,由于家庭数据的隐私性,用户并不总是愿意将数据上传至云端进行处理,尤其是一些家庭内部视频数据. 而边缘计算可以将计算(家庭数据处理)推送至家庭内部网关,减少家庭数据的外流,从而降低数据外泄的可能性,提升系统的隐私性.

工业界的一些企业已经注意到这一点,例如亚马逊的 Echo、三星的 SmartThings 和谷歌的 Google Home,均可作为智能家居的控制中心. 然而,这些设备,仍然需要一些额外的网络服务,如各种识别服务,不能完全依靠自身去处理数据,从而导致仍存在一定的隐私泄露隐患. 微软和苹果分别提出了 HomeOS 和 HomeKit,其作为智能家居的框架,可以方便用户对设备进行控制,但是仍然缺少一些具体的工作. 开源社区也建立并维护了多个智能家居系统[124],在表 3 中列举出其中功能、文档较为完备的 3 个系统并进行对比.

与此同时,学术界也有大量的学者根据边缘计算的思想在建设智能家居系统. 曹杰等人提出了一个适用于智能家居的边缘计算操作系统(edge operating system for home, $EdgeOS_H$)[52]. 受启发于边缘计算,作者在家庭中设置边缘服务器,并提出了 $EdgeOS_H$ 的工作,利用 $EdgeOS_H$ 在网络边缘侧对家庭数据进行处理. $EdgeOS_H$ 包含多个模块:通信模块负责智能

家居设备互联的,其适配多种智能家居中常用的协议;数据管理模块管理所有家庭数据,对数据进行融合和处理;自管理模块提供设备的管理以及智能家居服务间的管理,以期提供智能化的家居环境.作者认为命名(naming)和编程接口是智能家居发展中的几个较为关键的问题.因此,作者提供了编程接口以方便开发者在其上进行开发.同时,命名服务和其他模块进行合作,对资源进行统一的命名,提供方便地管理.

Table 3 The Comparison of Smart Home Systems

表 3 开源智能家居系统比较

Smart Home Systems	Data Abstraction	Device Abstraction	Automation
Home Assistant[125]	No	Yes	Rules, scripts
openHAB[126]	No	Yes	Rules, scripts
Freedomotic[127]	No	No	Rules, scripts

与 $EdgeOS_H$ 一样,中国科学院计算技术研究所的徐志伟研究员团队也一样认为,编程接口在智能家居等物联网设备中的应用较为重要.该团队拓展 RESTful 设计风格,将其引入物联网设备中[128].通过 RESTful 风格的接口,即使是外部用户也可以方便地访问智能家居设备,从而拉近了智能家居系统和传统网络的距离.同时在智能家居边缘侧,其利用非侵入式负荷监测(non-intrusive load monitoring, NILM)技术[129-130],关注于家庭的用电状况,并分析用电情况,提供更高效的节能方案.

3.6 智慧城市

智慧城市是利用先进的信息技术,实现城市智慧式的管理和运行.2016 年阿里云提出了"城市大脑"的概念[131],实质是利用城市的数据资源来更好地管理城市,2017 年 10 月 Alphabet 旗下城市创新部门 Sidewalk Labs 建造名为 Quayside 的高科技新区[132],并希望该智慧城市项目能够成为全球可持续和互联城市的典范.然而,智慧城市的建设所依赖的数据具有来源多样化和异构性的特点,同时涉及城市居民隐私和安全的问题,因此应用边缘计算模型,将数据在网络边缘处理是一个很好的解决方案.

边缘计算在智慧城市的建设中有丰富的应用场景.在城市路面检测中,在道路两侧路灯上安装传感器收集城市路面信息,检测空气质量、光照强度、噪音水平等环境数据,当路灯发生故障时能够及时反馈至维护人员.在智能交通中,边缘服务器上通过运行智能交通控制系统来实时获取和分析数据,根据实时路况来控制交通信息灯,以减轻路面车辆拥堵等.在无人驾驶中,如果将传感器数据上传到云计算中心将会增加实时处理难度,并且受到网络制约,因此无人驾驶主要依赖车内计算单元来识别交通信号和障碍物,并且规划路径.$EdgeOS_c$[47] 是一种基于边缘计算的面向智慧城市的系统级操作系统,它分为 3 个部分,底层的数据感知层、中间的网络互联层和顶层数据应用管理层.该操作系统可以有效管理智慧城市中的多来源数据,提高了数据共享的范围和深度,以实现智慧城市中数据价值的最大化.

4 边缘计算面临的紧迫问题

目前边缘计算已经得到了各行各业的广泛重视,并且在很多应用场景下开花结果.根据边缘计算领域特定的特点,本文认为 6 个方向是未来几年迫切需要解决的问题:编程模型、软硬件选型、基准程序与标准、动态调度、与垂直行业的紧密结合以及边缘节点的落地.

4.1 编程模型

编程模型可以使开发者快速上手开发应用产品,从而快速推动领域的发展.在云计算场景中,用户程序在目标平台上编写和编译,然后运行到云服务器,基础设施对于用户是透明的.例如亚马逊基于此编程模型推出的 Lambda 计算服务[133],可使用户无需预配置或者管理服务器即可运行代码,极大地方便了用户的使用.然而,边缘计算模型与云计算模型存在较大的区别,从功能角度讲,边缘计算是一种分布式的计算系统,具有弹性管理、协同执行和环境异构的特点,如图 4 所示:

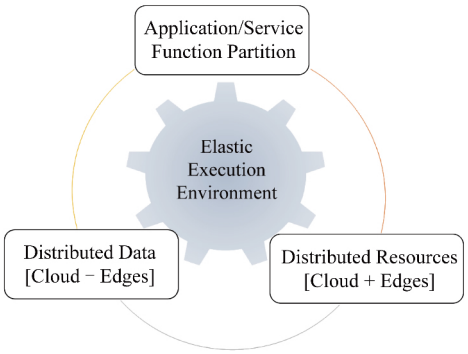

Fig. 4 The edge computing model

图 4 边缘计算模型

从图 4 可知,边缘计算包含 3 个关键内容:

1) 应用程序/服务功能可分割.边缘计算中的

一个任务可以分成若干个子任务并且任务功能可以迁移到不同的边缘设备去执行.任务可分割包括仅能分割其自身或将一个任务分割成子任务,任务的执行需要满足可迁移性,即任务可迁移是实现在边缘设备上进行数据处理的必要条件.

2)数据可分布.数据可分布既是边缘计算的特征也是边缘计算模型对待处理数据集合的要求.边缘数据的可分布性是针对不同数据源而言的,不同数据源来源于数据生产者所产生的大量数据.

3)资源可分布.边缘计算模型中的数据具有一定的可分布性,从而要求处理数据所需要的计算、存储和通信资源也具有可分布性.只有当边缘计算系统具备数据处理和计算所需要的资源,边缘设备才可以对数据进行处理.

因此,传统的编程模型并不适合边缘计算.边缘计算中的设备大多是异构计算平台,每个设备上的运行时环境、数据也不相同,且边缘设备的资源相对受限,在边缘计算场景下部署用户应用程序会有较大的困难.Li 等人[134]针对边缘设备资源受限的特性设计了一种轻量级的编程语言 EveryLite,该工作将计算迁移任务中主体为接口调用的、时间和空间复杂度受限的计算任务称为微任务(micro task),EveryLite 能够在物端设备上处理边缘计算场景中微任务,经过实验对比可以发现 EveryLite 的执行时间分别比 JerryScript 和 Lua 低 77%和 74%,编译后内存占用量分别是 JerryScript 和 Lua 的 18.9%和 1.4%.因此,针对边缘计算场景下的编程模型的研究具有非常大的空间,也十分紧迫.

4.2 软硬件选型

边缘计算系统具有碎片化和异构性的特点.在硬件层面上,有 CPU,GPU,FPGA,ASIC 等各类计算单元,即便是基于同一类计算单元,也有不同的整机产品,例如基于英伟达 GPU 的边缘硬件产品,既有计算能力较强的 DRIVE PX2,又有计算能力较弱的 Jetson TX2;在软件系统上,针对深度学习应用,有 TensorFlow,Caffe,PyTorch 等各类框架.不同的软硬件及其组合有各自擅长的应用场景,这带来了一个问题:开发者不知道如何选用合适的软硬件产品以满足自身应用的需求.

在软硬件选型时,既要对自身应用的计算特性做深入了解,从而找到计算能力满足应用需求的硬件产品,又要找到合适的软件框架进行开发,同时还要考虑到硬件的功耗和成本在可接受范围内.因此,设计并实现一套能够帮助用户对边缘计算平台进行性能、功耗分析并提供软硬件选型参考的工具十分重要.

4.3 基准程序和标准

随着边缘计算的发展,学术界和工业界开始推出越来越多的针对不同边缘计算场景设计的硬件或软件系统平台,那么我们会面临一个紧迫的问题,即如何对这些系统平台进行全面并公平的评测.传统的计算场景都有经典基准测试集(benchmark),例如并行计算场景中的 PARSEC[135]、高性能计算场景中的 HPCC[136]、大数据计算场景中的 BigDataBench[137].

由于边缘计算仍然是较新的计算场景,业界仍然没有一个比较权威的用于评测系统性能的 Benchmark 出现,但是学术界已经开始有了一些探索工作.SD-VBS[138]和 MEVBench[139]均是针对移动端设备评测基于机器视觉负载的基准测试集.SD-VBS 选取了 28 个机器视觉核心负载,并提供了 C 和 Matlab 的实现;MEVBench 则提供了一些列特征提取、特征分类、物体检测和物体追踪相关的视觉算法负责,并提供单线程核多线程的 C++实现.SLAMBench[140]是一个针对移动端机器人计算系统设计的基准测试集,其使用 RGB-D SLAM 作为评测负载,并且针对不同异构硬件提供 C++,OpenMP,OpenCL 和 CUDA 版本的实现.CAVBench[141]是第 1 个针对智能网联车边缘计算系统设计的基准测试集,其选择 6 个智能网联车上的典型应用作为评测负责,并提供标准的输入数据集和应用-系统匹配指标.

由于边缘计算场景覆盖面广,短期来看不会出现一个统一的基准测试集可以适应所有场景下的边缘计算平台,而是针对每一类计算场景会出现一个经典的基准测试集,之后各个基准测试集互相融合借鉴,找出边缘计算场景下的若干类核心负载,最终形成边缘计算场景中的经典基准测试集.

4.4 动态调度

在云计算场景下,任务调度的一般策略是将计算密集型任务迁移到资源充足的计算节点上执行.但是在边缘计算场景下,边缘设备产生的海量数据无法通过现有的带宽资源传输到云计算中心进行集中式计算,且不同边缘设备的计算、存储能力均不相同,因此,边缘计算系统需要根据任务类型和边缘设备的计算能力进行动态调度.调度包括 2 个层面:1)云计算中心和边缘设备之前的调度;2)边缘设备之间的调度.

云计算中心与边缘设备间的调度分为 2 种方式:自下而上和自上而下. 自下而上是在网络边缘处将边缘设备采集或者产生的数据进行部分或者全部的预处理,过滤无用数据,以此降低传输带宽;自上而下是指将云计算中心所执行的复杂计算任务进行分割,然后分配给边缘设备执行,以此充分利用边缘设备的计算资源,减少整个计算系统的延迟和能耗. 2017 年,Kang 等人[142]设计了一个轻量级的调度器 Neurosurgeon,它可以将深度神经网络不同层的计算任务在移动设备和数据中心间自动分配,使得移动设备功耗最多降低了 94.7%,系统延迟最多加快了 40.7 倍,并且数据中心的吞吐量最多增加了 6.7 倍. 边缘设备间也需要动态调度. 边缘设备的计算、存储能力本身是不同的,并且会随着时间的变化而变化,而它们承担的任务类型也是不一样的,因此需要动态调度边缘设备上的任务,提高整体系统性能,防止出现计算任务调度到一个系统任务过载情况下的设备. Zhang 等人[143]针对延迟敏感性的社会感知任务设计了一个边缘任务调度框架 CoGTA,实验证明该框架可以满足应用和边缘设备的需求.

综上所述,动态调度的目标是为应用程序调度边缘设备上的计算资源,以实现数据传输开销最小化和应用程序执行性能的最大化. 设计调度程序时应该考虑:任务是否可拆分可调度、调度应该采取什么策略、哪些任务需要调度等. 动态调度需要在边缘设备能耗、计算延时、传输数据量、带宽等指标之间寻找最优平衡. 根据目前的工作,如何设计和实现一种有效降低边缘设备任务执行延迟的动态调度策略是一个急需解决的问题.

4.5 和垂直行业紧密合作

在云计算场景下,不同行业的用户都可将数据传送至云计算中心,然后交由计算机从业人员进行数据的存储、管理和分析. 云计算中心将数据抽象并提供访问接口给用户,这种模式下计算机从业人员与用户行业解耦和,他们更专注数据本身,不需对用户行业领域内知识做太多了解.

但是在边缘计算的场景下,边缘设备更贴近数据生产者,与垂直行业的关系更为密切,设计与实现边缘计算系统需要大量的领域专业知识. 另一方面,垂直行业迫切需要利用边缘计算技术提高自身的竞争力,却面临计算机专业技术不足的问题. 因此计算机从业人员必须与垂直行业紧密合作,才能更好地完成任务,设计出下沉可用的计算系统. 在与垂直行业进行合作时,需要着重解决 3 个问题:

1) 减少与行业标准间的隔阂. 在不同行业内部有经过多年积累的经验与标准,在边缘计算系统的设计中,需要与行业标准靠近,减少隔阂. 例如,在针对自动驾驶汽车的研究中,自动驾驶任务的完成需要使用到智能算法、嵌入式操作系统、车载计算硬件等各类计算机领域知识,这对于计算机从业人员而言是一个机遇,因此许多互联网公司投入资源进行研究. 然而,若想研制符合行业标准的汽车,仅应用计算机领域知识是完全不够的,还需要对汽车领域专业知识有较好的理解,例如汽车动力系统、控制系统等,这就需要与传统汽车厂商进行紧密合作. 同样,在智能制造、工业物联网等领域,同样需要设计下沉到领域内部、符合行业标准的边缘计算系统.

2) 完善数据保护和访问机制. 在边缘计算中,需要与行业结合,在实现数据隐私保护的前提下设计统一、易用的数据共享和访问机制. 由于不同行业具有的特殊性,许多行业不希望将数据上传至公有云,例如医院、公安机构等. 而边缘计算的一大优势是数据存放在靠近数据生产者的边缘设备上,从而保证了数据隐私. 但是这也导致了数据存储空间的多样性,不利于数据共享和访问. 在传统云计算中,数据传输到云端,然后通过统一接口进行访问,极大地方便了用户的使用. 边缘计算需要借助这种优势来设计数据防护和访问机制.

3) 提高互操作性. 边缘计算系统的设计需要易于结合行业内现有的系统,考虑到行业现状并进行利用,不要与现实脱节. 例如在视频监控系统中,除了近些年出现的智能计算功能的摄像头,现实中仍然有大量的非智能摄像头,其每天仍然在采集大量的视频数据,并将数据传输至数据中心. 学术界设计了 A3[113]系统,它利用了商店或者加油站中已有的计算设备. 然而实际情况下,摄像头周边并不存在计算设备. 因此,在边缘计算的研究中需要首先考虑如何部署在非智能的摄像头附近部署边缘计算设备. 在目前的解决方案中,多是采用建立更多的数据中心或 AI 一体机来进行处理,或者采用一些移动的设备,如各种单兵作战设备,来进行数据的采集. 前者耗费巨大,且从本质来说,仍然是云计算的模式;后者通常使用于移动情况下,仅作为临时的计算中心,无法和云端进行交互. 在视频监控领域,Luo 等人提出了一个尚属于前期探讨的 EdgeBox 方案[144],其同时具备计算能力和通信能力,可以作为中间件插入到摄像头和数据中心之间,完成数据的预处理. 因此,如何与垂直行业紧密合作,设计出下沉可用的

边缘计算系统,实现计算机与不同行业间的双赢是边缘计算面临的一个紧迫问题.

4.6 边缘节点落地问题

边缘计算的发展引起了工业界的广泛关注,但是在实际边缘节点的落地部署过程中,也涌现出一些急需解决的问题,例如应该如何建立适用于边缘计算的商业模式、如何选择参与计算的边缘节点和边缘计算数据、如何保证边缘节点的可靠性等.

1)新型商业模式. 在云计算场景下,云计算公司是计算服务的提供者,它们收集、存储、管理数据并且负责软硬件、基础设施的建设和维护,用户付费购买服务,不需要关注计算节点本身的成本,也无需关注服务质量的升级换代过程. 这种商业模式为用户使用云服务带来了便利,也让云计算公司具备盈利能力,从而更好地提高服务质量.

而在边缘计算场景下,边缘节点分布在靠近数据生产者的位置,在地理位置上具有较强的离散性,这使得边缘节点的统一性维护变得困难,同时也给软硬件升级带来了难度. 例如提供安全服务的摄像头,在使用过程中需要进行软硬件的升级,软件的升级可以通过网络统一进行,而硬件的升级需要亲临现场. 依赖于服务提供者去为每一个边缘节点(摄像头)进行硬件的升级和维护会带来巨大的成本开销,而服务的使用者一般不关注也不熟悉硬件设备的维护工作. 又如,在 CDN 服务的应用中,需要考虑 CDN 服务器是以家庭为单位还是以园区为单位配置,不同的配置方式会带来成本的变化,也为服务质量的稳定性增加了不确定因素,而维护 CDN 所需的开销,需要考虑支付者是服务提供者还是使用者.

因此工业界需要寻求一种或多种新的商业模式来明确边缘计算服务的提供者和使用者各自应该承担什么责任,例如谁来支付边缘节点建立和维护所需的费用、谁来主导软硬件升级的过程等.

2) 边缘节点的选择. 边缘计算是一个连续统,边缘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[5]. 在实际应用中,用户可以选择云到端整个链路上任意的边缘节点来降低延迟和带宽. 由于边缘节点的计算能力、网络带宽的差异性,不同边缘节点的选择会导致计算延迟差异很大. 现有的基础设施可以用作边缘节点,例如使用手持设备访问进行通信时,首先连接运营商基站,然后访问主干网络. 这种以现有基础设施当做边缘节点的方式会加大延迟,如果手持设备能够绕过基站,直接访问主干网络的边缘节点,将会降低延迟. 因此,如何选择合适的边缘节点以降低通信延迟和计算开销是一个重要的问题. 在此过程中,需要考虑现有的基础设施如何与边缘节点融合,边缘计算技术会不会构建一个新兴的生态环境,给现有的基础设施发生革命性的变化?

3) 边缘数据选择. 边缘节点众多,产生的数据数量和类型也众多,这些数据间互有交集,针对一个问题往往有多个可供选择的解决方案. 例如在路况实时监控应用中,既可以利用车上摄像头获得数据,也可以利用交通信号灯的实时数据统计,还可以利用路边计算单元进行车速计算. 因此如何为特定应用合理地选择不同数据源的数据,以最大程度地降低延迟和带宽,提高服务的可用性是一个重要问题.

4) 边缘节点的可靠性. 边缘计算中的数据存储和计算任务大多数依赖于边缘节点,不像云计算中心有稳定的基础设施保护,许多边缘节点暴露于自然环境下,保证边缘节点的可靠性非常重要. 例如,基于计算机视觉的公共安全解决方案需要依赖智能摄像头进行存储和计算,然而在极端天气条件下,摄像头容易在物理上收到损害,例如暴风天气会改变摄像头的角度,暴雪天气会影响摄像头的视觉范围,在此类场景中,需要借助基础设施的配合来保证边缘节点的物理可靠性. 同时,边缘数据有时空特性,从而导致数据有较强的唯一性和不可恢复性,需要设计合理的多重备份机制来保证边缘节点的数据可靠性. 因此,如何借助基础设施来保障边缘计算节点的物理可靠性和数据可靠性是一个重要的研究课题.

在边缘节点落地过程中,已经有了不少尝试,例如联通提出了建设边缘云,其规划至 2020 年建设 6 000～7 000 个边缘节点,将高带宽、低时延、本地化业务下沉到网络边缘,进一步提高网络效率、增强服务能力. 因此针对如何选择边缘节点,处理好边缘节点与现有基础设施的关系,保证边缘节点的可靠性的研究非常紧迫.

5 总　　结

边缘计算经过近几年的技术储备,已经得到了来自国内外政府、学术界和工业界的广泛重视和一致认可,现在到了全面开花结果并带来经济效益的时候. 得益于网络、隔离技术、体系结构、操作系统、算法执行框架、数据处理平台以及安全隐私这 7 个关键技术的快速发展,边缘计算技术已经走向成熟,并在许多应用场景下发挥作用.

本文总结了边缘计算的6个典型应用场景,发现边缘计算目前已经在公共安全中的实时数据处理和自动驾驶中取得了成功的经验,并迅速扩展到虚拟现实、智能家居等场景,本文相信在未来几年我们将会看到边缘计算在工业物联网和智慧城市等领域取得更多成功的例子,为产业升级提供助力.

为了能够在更多应用场景中取得成功,本文还提出了边缘计算现在发展面临的紧迫问题,包括编程模型、软硬件选型、基准程序与标准、动态调度、与垂直行业的紧密结合以及边缘节点的落地问题.边缘计算具有与垂直行业结合紧密的特点,如果能够相互配合,设计出下沉可用的边缘计算系统,将会实现计算机行业与各行各业间的双赢.

边缘计算自正式提出以来不过短短几年,就已经取得了爆发性的增长,本文相信:按照这个趋势继续进行下去,边缘计算将产生更大的外溢效果,成为各行各业的粘合剂和智能产业发展的催化剂,促进整个工业体系的升级转型.

致谢 作者感谢《计算机研究与发展》编辑部的邀请来为庆刊60周年撰写这篇论文.感谢腾讯公司黄世飞与作者就边缘计算落地问题的讨论!

参 考 文 献

[1] Turner V, Gantz J F, Reinsel D, et al. The digital universe of opportunities: Rich data and the increasing value of the Internet of things [EB/OL]. [2018-11-26]. https://www.emc.com/leadership/digital-universe/2014iview/index.htm

[2] Finnegan M. Boeing 787s to create half a terabyte of data per flight [EB/OL]. [2016-12-03]. https://www.computerworlduk.com/data/boeing-787s-create-half-terabyte-of-data-per-flight-says-virgin-atlantic-3433595/

[3] Sverdlik Y. Here's how much energy all US data centers consume [EB/OL]. [2016-12-03]. https://www.datacenterknowledge.com/archives/2016/06/27/heres-how-much-energy-all-us-data-centers-consume

[4] Voigt P, Von dem Bussche A. The EU General Data Protection Regulation (GDPR)[M]. Berlin: Springer, 2017

[5] Shi Weisong, Sun Hui, Cao Jie, et. al. Edge computing—An emerging computing model for the Internet of everything era [J]. Journal of Computer Research and Development, 2017, 54(5): 907-924 (in Chinese)
(施巍松, 孙辉, 曹杰, 等. 边缘计算: 万物互联时代新型计算模型[J]. 计算机研究与发展, 2017, 54(5): 907-924)

[6] Shi Weisong, Cao Jie, Zhang Quan, et al. Edge computing: Vision and challenges [J]. IEEE Internet of Things Journal, 2016, 3(5): 637-646

[7] Pallis G, Vakali A. Insight and perspectives for content delivery networks [J]. Communications of the ACM, 2006, 49(1): 101-106

[8] Ravi J, Shi Weisong, Xu Chengzhong. Personalized email management at network edges [J]. IEEE Internet Computing, 2005, 9(2): 54-60

[9] Satyanarayanan M, Bahl V, Caceres R, et al. The case for VM-based cloudlets in mobile computing [J]. IEEE Pervasive Computing, 2009, 8(4): 14-23

[10] Hu Y C, Patel M, Sabella D, et al. Mobile edge computing—A key technology towards 5G [J]. ETSI White Paper, 2015, 11(11): 1-16

[11] Bonomi F, Milito R, Zhu J, et al. Fog computing and its role in the Internet of things [C] //Proc of the 1st Edition of the MCC Workshop on Mobile Cloud Computing. New York: ACM, 2012: 13-16

[12] Xu Zhiwei. Cloud-sea computing systems: Towards thousand-fold improvement in performance per watt for the coming zettabyte era [J]. Journal of Computer Science and Technology, 2014, 29(2): 177-181

[13] Ryan LaMothe. Edge Computing [R]. Richland, WA: Pacific Northwest National Laboratory, 2013

[14] NSF. NSF/Intel partnership on ICN in wireless edge networks, ICN-WEN [EB/OL]. [2018-11-05]. https://www.nsf.gov/funding/pgm_summ.jsp?pims_id=505310

[15] NSF. NSF Workshop Report on Grand Challenges in Edge Computing [EB/OL]. [2018-12-18]. http://iot.eng.wayne.edu/edge/NSF%20Edge%20Workshop%20Report.pdf

[16] IEEE, ACM. The First IEEE/ACM Symposium on Edge Computing [EB/OL]. [2018-11-05]. http://acm-ieee-sec.org/2016/

[17] ETSI. Multi-access edge computing (MEC) [EB/OL]. [2018-11-05]. https://www.etsi.org/technologies-clusters/technologies/multi-access-edge-computing

[18] OpenFog Consortium. OpenFog [EB/OL]. [2018-11-04]. https://www.openfogconsortium.org/

[19] ECC. Edge computing consortium [EB/OL]. [2018-11-03]. http://www.ecconsortium.org/

[20] Amazon. AWS Greengrass [EB/OL]. [2018-11-05]. https://aws.amazon.com/greengrass/

[21] Google. Cloud IoT core [EB/OL]. [2018-11-05]. https://cloud.google.com/iot-core/

[22] Google. Edge TPU: Google's purpose-built ASIC designed to run inference at the edge [EB/OL]. [2018-11-27]. https://cloud.google.com/edge-tpu/

[23] Microsoft. Azure IoT edge [EB/OL]. [2018-11-05]. https://azure.microsoft.com/en-us/services/iot-edge/

[24] Alibaba. Aliyun IoT [EB/OL]. [2018-11-05]. https://iot.aliyun.com/products/linkedge

[25] Baidu. IoT intelligent edge [EB/OL]. [2018-11-05]. https://cloud.baidu.com/solution/iot/intelligent_edge.html

[26] ARM. Arm's project trillium [EB/OL]. [2018-11-05]. https://www.arm.com/products/silicon-ip-cpu/machine-learning/project-trillium

[27] AMD. AMD EPYC™ embedded processors[EB/OL]. [2018-11-05]. https://www.amd.com/en/products/embedded-epyc

[28] Cisco. Cisco announces kinetic IoT operations platform [EB/OL]. [2018-11-05]. https://www.zdnet.com/article/cisco-announces-kinetic-iot-operations-platform/

[29] Intel. Intel® Xeon® D-2100 processor [EB/OL]. [2018-11-05]. https://www.intel.com/content/www/us/en/products/processors/xeon/d-processors.html

[30] VMware. VMware extends Internet of things strategy with new edge computing solutions [EB/OL]. [2018-11-05]. https://www.vmware.com/company/news/releases/vmw-newsfeed.VMware-Extends-Internet-of-Things-Strategy-with-New-Edge-Computing-Solutions.1396703.html

[31] Huawei. EC-IoT solution [EB/OL]. [2018-11-05]. https://e.huawei.com/en/solutions/technical/sdn/agile-iot

[32] Akamai. IoT edge connect [EB/OL]. [2018-11-05]. https://www.akamai.com/us/en/products/web-performance/iot-edge-connect.jsp

[33] CloudFlare. CloudFlare workers [EB/OL]. [2018-11-05]. https://www.cloudflare.com/products/cloudflare-workers/

[34] Limelight. EdgePrism [EB/OL]. [2018-11-05]. https://www.limelight.com/orchestrate-platform/advanced-content-delivery/

[35] Li Dong. Edge computing has become a new track in the cloud market [EB/OL]. [2018-11-05]. http://epaper.zqrb.cn/html/2018-04/19/content_281057.htm?div=-1 (in Chinese)
(李东. 边缘计算已成千亿云市场新赛道[EB/OL]. [2018-11-05]. http://epaper.zqrb.cn/html/2018-04/19/content_281057.htm?div=-1)

[36] People's Daily Online. China Mobile established the Open Computing Laboratory for Edge Computing [EB/OL]. [2018-11-05]. http://gz.people.com.cn/n2/2018/1102/c372080-32235554.html (in Chinese)
(人民网. 中国移动宣布成立中国移动边缘计算开放实验室 [EB/OL]. [2018-11-05]. http://gz.people.com.cn/n2/2018/1102/c372080-32235554.html)

[37] China Economic Weekly. China Unicom's edge cloud ecology partner conference was held in Beijing [EB/OL]. [2018-11-05]. http://www.ceweekly.cn/2018/0613/227246.shtml (in Chinese)
(经济网. 2018 中国联通边缘云生态合作伙伴大会在京举行 [EB/OL]. [2018-11-05]. http://www.ceweekly.cn/2018/0613/227246.shtml)

[38] AT&T. Edge computing test zone: What we've learned and what's ahead [EB/OL]. [2018-11-05]. https://about.att.com/story/2018/edge_update.html

[39] MobiledgeX, Inc.. MobiledgeX [EB/OL]. [2018-11-05]. https://mobiledgex.com/

[40] CAICT. China Academy of Information and Communications Technology leads the ITU-T SG20 "Internet of Things Edge Computing" International Standard [EB/OL]. [2018-11-05]. http://www.caict.ac.cn/xwdt/ynxw/201804/t20180426_157876.htm (in Chinese)
(中国信通院. 中国信通院牵头推进 ITU-T SG20"物联网边缘计算"国际标准[EB/OL]. [2018-11-05]. http://www.caict.ac.cn/xwdt/ynxw/201804/t20180426_157876.htm)

[41] ECC. China Automation Association Edge Computing Professional Committee was formally established [EB/OL]. [2018-11-05]. http://www.ecconsortium.org/Lists/show/id/235.html (in Chinese)
(边缘计算产业联盟. 中国自动化学会边缘计算专业委员会正式成立[EB/OL]. [2018-11-05]. http://www.ecconsortium.org/Lists/show/id/235.html)

[42] CIC. China Institute of Communications and China Mobile hosted "2018 Edge Computing Technology Summit" [EB/OL]. [2018-11-05]. http://www.china-cic.cn/Detail/24/1219/1219 (in Chinese)
(中国通信学会. 中国通信学会联合中国移动举办"2018 边缘计算技术峰会"[EB/OL]. [2018-11-05]. http://www.china-cic.cn/Detail/24/1219/1219)

[43] Avnu A. Avnu Alliance [EB/OL]. [2018-11-05]. https://avnu.org/

[44] AECC. A consortium for driving the network and computing infrastructure needs of automotive big data [EB/OL]. [2018-11-05]. https://aecc.org/

[45] Edgecross C. Edgecross consortium [EB/OL]. [2018-11-05]. https://www.edgecross.org/en/

[46] CNCF. Eclipse foundation push Kubernetes to the edge [EB/OL]. [2018-11-05]. https://www.sdxcentral.com/articles/news/cncf-eclipse-foundation-push-kubernetes-to-the-edge/2018/09/

[47] Shi Weisong, Liu Fang, Sun Hui, et al. Edge Computing [M]. Beijing: Science Press, 2018 (in Chinese)
(施巍松, 刘芳, 孙辉, 等. 边缘计算[M]. 北京: 科学出版社, 2018)

[48] WAIC. World Artificial Intelligence Conference 2018 [EB/OL]. [2018-11-05]. http://www.waic2018.com/

[49] CCF. Advanced Computer Architecture 2018 [EB/OL]. [2018-11-05]. http://aca2018.tcarch.org/

[50] Cheshire S, Krochmal M. DNS-based service discovery [EB/OL]. [2018-11-26]. https://tools.ietf.org/html/rfc6763

[51] Lu N, Cheng N, Zhang N, et al. Connected vehicles: Solutions and challenges [J]. IEEE Internet of Things Journal, 2014, 1(4): 289-299

[52] Cao Jie, Xu Lanyu, Abdallah R, et al. EdgeOS_H: A home operating system for Internet of everything [C] //Proc of the 37th IEEE Int Conf on Distributed Computing Systems (ICDCS 2017). Piscataway, NJ: IEEE, 2017: 1756-1764

[53] Zhang L, Afanasyev A, Burke J, et al. Named data networking [J]. ACM SIGCOMM Computer Communication Review, 2014, 44(3): 66-73

[54] McKeown N. Software-defined networking [J]. INFOCOM Keynote Talk, 2009, 17(2): 30-32

[55] McKeown N, Anderson T, Balakrishnan H, et al. OpenFlow: Enabling innovation in campus networks [J]. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 69-74

[56] Ma L, Yi S, Li Q. Efficient service handoff across edge servers via Docker container migration [C] //Proc of the 2nd ACM/IEEE Symp on Edge Computing (SEC). New York: ACM, 2017: 11:1-11:13

[57] Ha K, Abe Y, Eiszler T, et al. You can teach elephants to dance: Agile VM handoff for edge computing [C] //Proc of the 2nd ACM/IEEE Symp on Edge Computing (SEC). New York: ACM, 2017: 12:1-12:14

[58] Chung E S, Milder P A, Hoe J C, et al. Single-chip heterogeneous computing: Does the future include custom logic, FPGAs, and GPGPUs? [C] //Proc of the 43rd Annual IEEE/ACM Int Symp on Microarchitecture (MICRO 2010). Piscataway, NJ: IEEE, 2010: 225-236

[59] Nurvitadhi E, Sim J, Sheffield D, et al. Accelerating recurrent neural networks in analytics servers: Comparison of FPGA, CPU, GPU, and ASIC [C] //Proc of the 26th Int Conf on Field Programmable Logic and Applications (FPL2016). Piscataway, NJ: IEEE, 2016: 1-4

[60] Nurvitadhi E, Sheffield D, Sim J, et al. Accelerating binarized neural networks: Comparison of FPGA, CPU, GPU, and ASIC [C] //Proc of 2016 Int Conf on Field-Programmable Technology (FPT). Piscataway, NJ: IEEE, 2016: 77-84

[61] Lin S C, Zhang Y, Hsu C H, et al. The architectural implications of autonomous driving: Constraints and acceleration [C] //Proc of the 23rd Int Conf on Architectural Support for Programming Languages and Operating Systems. New York: ACM, 2018: 751-766

[62] Du Z, Fasthuber R, Chen T, et al. ShiDianNao: Shifting vision processing closer to the sensor [C] //Proc of ACM SIGARCH Computer Architecture News. New York: ACM, 2015, 43(3): 92-104

[63] Han S, Liu X, Mao H, et al. EIE: Efficient inference engine on compressed deep neural network [C] //Proc of the 43rd Annual ACM/IEEE Int Symp on Computer Architecture (ISCA 2016). Piscataway, NJ: IEEE, 2016: 243-254

[64] Xu Z, Peng X, Zhang L, et al. The Φ-stack for smart Web of things [C] //Proc of the Workshop on Smart Internet of Things (SmartIoT'17). New York: ACM, 2017: 10:1-10:6

[65] Song M, Zhong K, Zhang J, et al. In-Situ AI: Towards autonomous and incremental deep learning for IoT systems [C] //Proc of 2018 IEEE Int Symp on High Performance Computer Architecture (HPCA). Piscataway, NJ: IEEE, 2018: 92-103

[66] Han S, Kang J, Mao H, et al. ESE: Efficient speech recognition engine with sparse LSTM on FPGA [C] //Proc of the 2017 ACM/SIGDA Int Symp on Field-Programmable Gate Arrays. New York: ACM, 2017: 75-84

[67] Biookaghazadeh S, Ren F, Zhao M. Are FPGAs suitable for edge computing? [J]. arXiv preprint arXiv: 1804. 06404, 2018

[68] Dunkels A, Gronvall B, Voigt T. Contiki—A lightweight and flexible operating system for tiny networked sensors [C] //Proc of the 29th Annual IEEE Int Conf on Local Computer Networks. Piscataway, NJ: IEEE, 2004: 455-462

[69] FreeRTOS. The FreeRTOS™ kernel [EB/OL]. [2018-11-05]. http://www.freertos.org/

[70] Quigley M, Conley K, Gerkey B, et al. ROS: An open-source robot operating system [EB/OL]. [2018-11-26]. http://www.willowgarage.com/sites/default/files/icraoss09-ROS.pdf

[71] Maruyama Y, Kato S, Azumi T. Exploring the performance of ROS2 [C] //Proc of the 13th Int Conf on Embedded Software. New York: ACM, 2016: 5:1-5:10

[72] Zhang Q, Wang Y, Zhang X, et al. OpenVDAP: An open vehicular data analytics platform for CAVs [C] //Proc of the 38th IEEE Int Conf on Distributed Computing Systems (ICDCS 2018). Piscataway, NJ: IEEE, 2018: 1310-1320

[73] Abadi M, Barham P, Chen J, et al. TensorFlow: A system for large-scale machine learning [C] //Proc of OSDI 2016. Berkeley, CA: USENIX Association, 2016: 265-283

[74] Jia Y, Shelhamer E, Donahue J, et al. Caffe: Convolutional architecture for fast feature embedding [C] //Proc of the 22nd ACM Int Conf on Multimedia. New York: ACM, 2014: 675-678

[75] Google. TensorFlow Lite Developer Guide [OL]. [2018-11-04]. https://www.tensorflow.org/lite/devguide

[76] Facebook. Caffe2 [EB/OL]. [2018-11-05]. https://caffe2.ai/

[77] Ketkar N. Introduction to PyTorch [M] //Deep Learning with Python. Berkeley, CA: Apress, 2017: 195-208

[78] Chen T, Li M, Li Y, et al. MXNet: A flexible and efficient machine learning library for heterogeneous distributed systems [J]. arXiv preprint arXiv: 1512. 01274, 2015

[79] Zhang X, Wang Y, Shi W. pCAMP: Performance comparison of machine learning packages on the edges [C/OL] //Proc of USENIX Workshop on Hot Topics in Edge Computing (HotEdge'18). Berkeley, CA: USENIX Association, 2018 [2018-12-19]. https://www.usenix.org/conference/hotedge18/presentation/zhang

[80] Yi S, Qin Z, Li Q. Security and privacy issues of fog computing: A survey [C] //Proc of Int Conf on Wireless Algorithms, Systems, and Applications. Cham, Switzerland: Springer, 2015: 685-695

[81] Sabt M, Achemlal M, Bouabdallah A. Trusted execution environment: What it is, and what it is not [C] //Proc of the 14th IEEE Int Conf on Trust, Security and Privacy in Computing and Communications. Piscataway, NJ: IEEE, 2015: 57-64

[82] Ning Z, Zhang F, Shi W, et al. Position paper: Challenges towards securing hardware-assisted execution environments [C] //Proc of the Hardware and Architectural Support for Security and Privacy. New York: ACM, 2017: 6:1-6:8

[83] Anati I, Gueron S, Johnson S, et al. Innovative technology for CPU based attestation and sealing [C] //Proc of the 2nd Int Workshop on Hardware and Architectural Support for Security and Privacy. New York: ACM, 2013: 13:1-13:7

[84] Hoekstra M, Lal R, Pappachan P, et al. Using innovative instructions to create trustworthy software solutions [C] //Proc of the 2nd Int Workshop on Hardware and Architectural Support for Security and Privacy. New York: ACM, 2013: 11

[85] Mckeen F, Alexandrovich I, Berenzon A, et al. Innovative instructions and software model for isolated execution [C] //Proc of the 2nd Int Workshop on Hardware and Architectural Support for Security and Privacy. New York: ACM, 2013: 10

[86] Ruan Xiaoyu. Platform Embedded Security Technology Revealed: Safeguarding the Future of Computing with Intel Embedded Security and Management Engine [M]. New York: Apress, 2014

[87] Intel. Intel® 64 and IA-32 architectures software developer manuals [OL]. [2018-05-18]. https://software.intel.com/en-us/articles/intel-sdm

[88] Kaplan D, Powell J, Woller T. AMD memory encryption [OL]. [2016-06-19]. http://amd-dev.wpengine.netdna-cdn.com/wordpress/media/2013/12/AMD_Memory_Encryption_Whitepaper_v7-Public.pdf

[89] AMD. AMD secure technology [OL]. [2017-06-18]. http://www.amd.com/en-us/innovations/software-technologies/security

[90] ARM. Security technology-building a secure system using TrustZone [OL]. [2015-10-07]. http://infocenter.arm.com/help/topic/com.arm.doc.prd29-genc-009492c/PRD29-GENC-009492C_trustzone_security_whitepaper.pdf

[91] Wang Y, Liu L, Su C, et al. CryptSQLite: Protecting data confidentiality of SQLite with Intel SGX [C] //Proc of 2017 Int Conf on Networking and Network Applications (NaNA). Piscataway, NJ: IEEE, 2017: 303-308

[92] Xinhua Net. Wuhan plans to promote "Snow Project", which will increase the number of video surveillance probes to 1.5 million [EB/OL]. [2018-10-15]. http://m.xinhuanet.com/hb/2018-07/18/c_1123143898.htm (in Chinese)
(新华社. 武汉推进"雪亮工程"视频监控探头将增至150万个[EB/OL]. [2018-10-15]. http://m.xinhuanet.com/hb/2018-07/18/c_1123143898.htm)

[93] Domo Inc. Data never sleeps 5.0 [OL]. [2018-10-13]. https://www.domo.com/learn/data-never-sleeps-5

[94] Zhang Q, Yu Z, Shi W, et al. Demo abstract: EVAPS: Edge video analysis for public safety [C] //Proc of the 1st IEEE/ACM Symp on Edge Computing (SEC). Piscataway, NJ: IEEE, 2016: 121-122

[95] Sun H, Liang X, Shi W. VU: Video usefulness and its application in large-scale video surveillance systems: An early experience [C] //Proc of the Workshop on Smart Internet of Things. New York: ACM, 2017: 6:1-6:6

[96] Zhang Q, Zhang Q, Shi W, et al. Distributed collaborative execution on the edges and its application to AMBER Alerts [J]. IEEE Internet of Things Journal, 2018, 5(5): 3580-3593

[97] Liu L, Zhang X, Qiao M, et al. SafeShareRide: Edge-based attack detection in ridesharing services [C] //Proc of the 3rd IEEE/ACM Symp on Edge Computing (SEC). Piscataway, NJ: IEEE, 2018: 17-29

[98] Wu X, Dunne R, Zhang Q, et al. Edge computing enabled smart firefighting: Opportunities and challenges [C] //Proc of the 5th ACM/IEEE Workshop on Hot Topics in Web Systems and Technologies. New York: ACM, 2017: 11:1-11:6

[99] Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? the kitti vision benchmark suite [C] //Proc of 2012 IEEE Conf on Computer Vision and Pattern Recognition (CVPR). Piscataway, NJ: IEEE, 2012: 3354-3361

[100] Gerla M, Lee E K, Pau G, et al. Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds [C] //Proc of 2014 IEEE World Forum on Internet of Things (WF-IoT). Piscataway, NJ: IEEE, 2014: 241-246

[101] Dimitrakopoulos G, Demestichas P. Intelligent transportation systems [J]. IEEE Vehicular Technology Magazine, 2010, 5(1): 77-84

[102] Intel. Data is the new oil in the future of automated driving [OL]. [2018-11-04]. https://newsroom.intel.com/editorials/krzanich-the-future-of-automated-driving/

[103] Geiger A, Lenz P, Stiller C, et al. Vision meets robotics: The KITTI dataset [J]. The International Journal of Robotics Research, 2013, 32(11): 1231-1237

[104] Mur-Artal R, Tardós J D. Orb-slam2: An open-source slam system for monocular, stereo, and RGB-D cameras [J]. IEEE Transactions on Robotics, 2017, 33(5): 1255-1262

[105] Liu W, Anguelov D, Erhan D, et al. SSD: Single shot multibox detector [C] //Proc of European Conf on Computer Vision. Cham, Switzerland: Springer, 2016: 21-37

[106] Xiang Y, Alahi A, Savarese S. Learning to track: Online multi-object tracking by decision making [C] //Proc of 2015 IEEE Int Conf on Computer Vision (ICCV). Piscataway, NJ: IEEE, 2015: 4705-4713

[107] NVIDIA DRIVE PX2. NVIDIA DRIVE: Scalable AI platform for autonomous driving [EB/OL]. [2018-11-04]. https://www.nvidia.com/en-us/self-driving-cars/drive-platform

[108] Xilinx. Xilinx Zynq UltraScale + MPSoC ZCU106 Evaluation Kit [EB/OL]. [2018-11-04]. https://www.xilinx.com/products/boards-and-kits/zcu106.html

[109] Liu S, Tang J, Zhang Z, et al. Computer architectures for autonomous driving [J]. Computer, 2017, 50(8): 18-25

[110] Lin S C, Zhang Y, Hsu C H, et al. The architectural implications of autonomous driving: Constraints and acceleration [C] //Proc of the 23rd Int Conf on Architectural Support for Programming Languages and Operating Systems. New York: ACM, 2018: 751-766

[111] Baidu. Apollo open platform [EB/OL]. [2018-11-05]. http://apollo.auto/index.html

[112] Kato S, Takeuchi E, Ishiguro Y, et al. An open approach to autonomous vehicles [J]. IEEE Micro, 2015, 35(6): 60-68

[113] Chhabra R, Verma S, Krishna C R. A survey on driver behavior detection techniques for intelligent transportation systems [C] //Proc of the 7th Int Conf on Cloud Computing, Data Science & Engineering-Confluence. Piscataway, NJ: IEEE, 2017: 36-41

[114] Kar G, Jain S, Gruteser M, et al. PredriveID: Pre-trip driver identification from in-vehicle data [C] //Proc of the 2nd ACM/IEEE Symp on Edge Computing (SEC). New York: ACM, 2017: 2:1-2:12

[115] Satyanarayanan M. The emergence of edge computing [J]. Computer, 2017, 50(1): 30-39

[116] Li Y, Gao W. MUVR: Supporting multi-user mobile virtual reality with resource constrained edge cloud [C] //Proc of the 3rd IEEE/ACM Symp on Edge Computing (SEC). Piscataway, NJ: IEEE, 2018: 1-16

[117] Lai Z, Hu Y C, Cui Y, et al. Furion: Engineering high-quality immersive virtual reality on today's mobile devices [C] //Proc of the 23rd Annual Int Conf on Mobile Computing and Networking. New York: ACM, 2017: 409-421

[118] Ha K, Chen Z, Hu W, et al. Towards wearable cognitive assistance [C] //Proc of the 12th Annual Int Conf on Mobile Systems, Applications, and Services. New York: ACM, 2014: 68-81

[119] Edge Computing Task Group. Introduction to edge computing in IIoT [OL]. [2018-11-03]. https://www.iiconsortium.org/pdf/Introduction_to_Edge_Computing_in_IIoT_2018-06-18.pdf

[120] Chen Y, Feng Q, Shi W. An industrial robot system based on edge computing: An early experience [C/OL] //Proc of USENIX Workshop on Hot Topics in Edge Computing (HotEdge'18). Berkeley, CA: USENIX Association, 2018 [2018-12-19]. https://www.usenix.org/conference/hotedge18/presentation/chen

[121] Kurkinen L. Smart homes and home automation: 3rd edition [R]. Sweden: Berg Insights, 2015 [2018-12-18]. https://ec.europa.eu/research/innovation-union/pdf/active-healthy-ageing/berg_smart_homes.pdf

[122] Peng Xiaohui, Zhang Xingzhou, Wang Yifan, et al. Web enabled things computing system [J]. Journal of Computer Research and Development, 2018, 55(3): 572-584 (in Chinese)
(彭晓晖, 张星洲, 王一帆, 等. Web 使能的物端计算系统 [J]. 计算机研究与发展, 2018, 55(3): 572-584)

[123] Abdallah R, Xu L, Shi W. Lessons and experiences of a DIY smart home [C] //Proc of the Workshop on Smart Internet of Things. New York: ACM, 2017: 4:1-4:6

[124] Cao J, Xu L, Raef A, et al. An OS for Internet of everything: Early experience from a smart home prototype [J]. ZTE Communications, 2017, 15(4): 12-22

[125] Home A, Home assistant [EB/OL]. [2018-11-04]. https://home-assistant.io/

[126] OpenHAB Community and the openHAB Foundation. OpenHAB [EB/OL]. [2018-11-04]. http://www.openhab.org/

[127] Freedomotic. Freedomotic: Open IoT framework [EB/OL]. [2018-11-04]. http://www.freedomotic.com/

[128] Xu Zhiwei, Chao Lu, Peng Xiaohui, et al. T-REST: An open-enabled architectural style for the Internet of things [J/OL]. IEEE Internet of Things Journal, 2018 [2018-12-09]. https://ieeexplore.ieee.org/document/8491289

[129] Wang Y, Zhang X, Chao L, et al. PowerAnalyzer: An energy-aware power monitor system aiming at energy-saving [C] //Proc of the 8th Int Green and Sustainable Computing Conf (IGSC 2017). Piscataway, NJ: IEEE, 2017: 1-8

[130] Zhang X, Wang Y, Chao L, et al. IEHouse: A non-intrusive household appliance state recognition system [C] //Proc of 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI). Piscataway, NJ: IEEE, 2017: 1-8

[131] Alibaba. ET city brain [EB/OL]. [2018-11-03]. https://et.aliyun.com/brain/city

[132] Alphabet's Sidewalk Labs. Welcome to sidewalk Toronto [OL]. [2018-11-04]. https://sidewalktoronto.ca/

[133] Amazon. AWS Lambda [EB/OL]. [2018-11-04]. https://aws.amazon.com/cn/lambda/

[134] Li Z, Peng X, Chao L, et al. EveryLite: A lightweight scripting language for micro tasks in IoT systems [C] //Proc of the 1st ACM/IEEE Workshop on Hot Topics on Web of Things. (HotWoT). Piscataway, NJ: IEEE, 2018: 381-386

[135] Bienia C, Kumar S, Singh J P, et al. The PARSEC benchmark suite: Characterization and architectural implications [C] //Proc of the 17th Int Conf on Parallel Architectures and Compilation Techniques. New York: ACM, 2008: 72-81

[136] Dongarra J, Luszczek P. Introduction to the HPC challenge benchmark suite, ICL-UT-05-01 [R/OL]. [2018-12-19]. http://www.icl.utk.edu/~luszczek/pubs/hpcc-challenge-intro.pdf

[137] Wang L, Zhan J, Luo C, et al. BigDataBench: A big data benchmark suite from Internet services [C] //Proc of the 20th IEEE Int Symp on High Performance Computer Architecture (HPCA 2014). Piscataway, NJ: IEEE, 2014: 488-499

[138] Venkata S K, Ahn I, Jeon D, et al. SD-VBS: The San Diego vision benchmark suite [C] //Proc of IEEE Int Symp on Workload Characterization (IISWC 2009). Piscataway, NJ: IEEE, 2009: 55-64

[139] Clemons J, Zhu H, Savarese S, et al. MEVBench: A mobile computer vision benchmarking suite [C] //Proc of IEEE Int Symp on Workload Characterization (IISWC2011). Piscataway, NJ: IEEE, 2011: 91-102

[140] Nardi L, Bodin B, Zia M Z, et al. Introducing SLAMBench, a performance and accuracy benchmarking methodology for SLAM [C] //Proc of 2015 IEEE Int Conf on Robotics and Automation (ICRA). Piscataway, NJ: IEEE, 2015: 5783-5790

[141] Wang Y, Liu S, Wu X, et al. CAVBench: A benchmark suite for connected and autonomous vehicles [C] //Proc of the 3rd IEEE/ACM Symp on Edge Computing (SEC). Piscataway, NJ: IEEE, 2018: 30-42

[142] Kang Y, Hauswald J, Gao C, et al. Neurosurgeon: Collaborative intelligence between the cloud and mobile edge [J]. ACM SIGPLAN Notices, 2017, 52(4): 615-629

[143] Zhang Y, Ma Y, Zheng C, et al. Cooperative-competitive task allocation in edge computing for delay-sensitive social sensing [C] //Proc of the 3rd IEEE/ACM Symp on Edge Computing (SEC). Piscataway, NJ: IEEE, 2018: 243-259

[144] Luo B, Tan S, Yu Z, et al. EdgeBox: Live edge video analytics for near real-time event detection [C] //Proc of the 3rd IEEE/ACM Symp on Edge Computing (SEC). Piscataway, NJ: IEEE, 2018: 347-348

Shi Weisong, born in 1974. Professor, PhD supervisor. IEEE Fellow, Charles H. Gershenson Distinguished Faculty Fellow, ACM Distinguished Scientist. His main research interests include edge computing, computer systems, and energy-efficiency.

Zhang Xingzhou, born in 1992. PhD candidate at the Institute of Computing Technology, Chinese Academy of Sciences. Student member of CCF. His main research interests include edge computing, machine learning, and computer systems.

Wang Yifan, born in 1992. PhD candidate at the Institute of Computing Technology, Chinese Academy of Sciences. Student member of CCF. His main research interests include edge computing, computer systems, performance evaluation and autonoumous vehicles.

Zhang Qingyang, born in 1992. PhD candidate at Anhui University. Student member of CCF. His main research interests include edge computing, computer systems, and security.