

# 移动边缘计算中计算卸载方案研究综述

张依琳<sup>1),3)</sup> 梁玉珠<sup>1)</sup> 尹沐君<sup>3)</sup> 全韩或<sup>3)</sup> 王 田<sup>1),2),3)</sup> 贾维嘉<sup>1),2)</sup>

<sup>1)</sup>(北京师范大学人工智能与未来网络研究院 广东 珠海 519000)

<sup>2)</sup>(北京师范大学-香港浸会大学联合国际学院 广东省人工智能与多模态数据处理重点实验室 广东 珠海 519000)

<sup>3)</sup>(华侨大学计算机科学与技术学院 福建 厦门 361021)

**摘 要** 在移动边缘计算中通过将终端设备的计算任务卸载到边缘服务器,可以利用边缘服务器资源解决终端设备计算能力不足的问题,同时满足移动应用程序对低延迟的需求.因此,计算卸载备受关注并成为移动边缘计算的关键技术之一.本文对移动边缘计算的计算卸载研究进展进行深度调研.首先,总结归纳出两类计算卸载方法——基于启发式算法的传统方法和基于在线学习的智能方法;从最小化延迟时间、最小化能耗、权衡时间和能耗三个不同优化目标对基于启发式算法的传统计算卸载进行分析对比;梳理了基于在线学习智能计算卸载采用的底层人工智能技术;然后介绍了边缘服务器资源分配方案和新兴的移动边缘计算应用场景;最后分析计算卸载方案存在的问题并展望移动边缘计算的计算卸载研究的未来方向,为后续研究工作指明方向.

**关键词** 移动边缘计算;计算卸载;智能计算卸载;边缘服务器;资源分配

**中图法分类号** TP301 **DOI号** 10.11897/SP.J.1016.2021.02406

## Survey on the Methods of Computation Offloading in Mobile Edge Computing

ZHANG Yi-Lin<sup>1),3)</sup> LIANG Yu-Zhu<sup>1)</sup> YIN Mu-Jun<sup>3)</sup> QUAN Han-Yu<sup>3)</sup> WANG Tian<sup>1),2),3)</sup> JIA Wei-Jia<sup>1),2)</sup>

<sup>1)</sup>(Institute of Artificial Intelligence and Future Networks, Beijing Normal University, Zhuhai, Guangdong 519000)

<sup>2)</sup>(Guangdong Key Lab of AI and Multi-Modal Data Processing,

Beijing Normal University-Hong Kong Baptist University United International College, Zhuhai, Guangdong 519000)

<sup>3)</sup>(College of Computer Science and Technology, Huaqiao University, Xiamen, Fujian 361021)

**Abstract** With the rapid development of the Internet of Things (IoT), cloud computing, and big data, data manifests explosive growth. Traditional cloud computing uploads massive data to cloud servers. Due to the long distance between cloud servers and mobile devices, traditional mobile cloud computing suffers from high energy consumption and network delay, which limits the development of mobile applications. To overcome this limitation, Mobile Edge Computing (MEC), a novel networking and computing paradigm, is proposed and becoming more and more prevalent. In MEC, the computation tasks are offloaded from the resource-limited mobile devices to the powerful network edges, which can leverage the computing resources of the network edges to perform the computation tasks while providing quite low latency as most mobile applications ask. Therefore, computation offloading has become one of the essential technologies of MEC and gained a lot of attention in both the academic community and industrial world. In this paper, we conduct a deep survey of the state-of-the-art works of computation offloading in MEC. First, we divide the existing computation offloading schemes into two categories: the traditional computation offloading based on heuristic algorithms and the intelligent computation offloading based on online

收稿日期:2020-07-11;在线发布日期:2021-03-10. 本课题得到国家自然科学基金重点项目(61532013)、国家自然科学基金项目(62172046)、福建省自然科学基金杰出青年项目(2020J06023)、UIC 科研启动经费(R72021202)、福建省自然科学基金项目(2020J05059)、华侨大学科研基金项目(605-50Y19028)、广东省教育厅普通高校重点领域专项项目(2021ZDZX1063)和珠海市产学研合作项目(ZH22017001210133PWC)资助. 张依琳,硕士研究生,主要研究领域为移动边缘计算. E-mail: erinzyl@163.com. 梁玉珠,博士研究生,主要研究方向为物联网. 尹沐君,硕士研究生,主要研究方向为移动边缘计算. 全韩或,博士,讲师,主要研究方向为网络安全. 王 田(通信作者),博士,教授,中国计算机学会(CCF)高级会员,主要研究领域为边缘计算、物联网. 贾维嘉,博士,教授,IEEE Fellow,主要研究领域为智慧城市和分布式系统.

learning. We compare these two different methods in detail and analyze the advantages and disadvantages of the existing computation offloading methods. In general, the intelligent computation offloading methods are the mainstream of research directions in the future which outperforms the traditional computation offloading in terms of privacy data and user mobility. We analyze the traditional computation offloading schemes according to their optimization objectives, which include minimizing network delay, minimizing energy consumption, and optimizing the trade-off between network delay and energy consumption. These schemes are based on heuristic algorithm based on an optimization objective, and then design a heuristic algorithm to approach the optimal solutions. We also analyze the intelligent computation offloading schemes based on the underlying Artificial Intelligence (AI) technologies. These schemes based on online learning not only can solve the problem of high delay and energy consumption but also address the problem of data security and privacy, which is not considered in the traditional methods. More importantly, with AI technologies, most intelligent computation offloading does not take the network delay or the energy consumption as a single optimization objective but considers the overall performance of offloading. Then, we introduce the resource allocation schemes of the edge servers, which is an important process after the concrete computation tasks are offloaded to the edge servers. We also present several emerging application scenarios such as Internet of Things (IoT), Internet of Vehicles (IoV), Blockchain, Unmanned Aerial Vehicle (UAV), Virtual Reality (VR), and Augmented Reality (AR). Finally, we conclude technological challenges which include mobility of end devices, edge servers, security, user privacy data, and service heterogeneity and prospect future directions about computation offloading which could point out the direction for the follow-up research. Though this area of research is young, there is much room for improvement. We believe that more studies about computation offloading in mobile edge computing will bring more opportunities soon.

**Keywords** mobile edge computing; computation offloading; intelligence computation offloading; edge server; resource allocation

## 1 引言

近年来,随着物联网、云计算、大数据等技术的高速发展,数据规模呈爆炸式增长<sup>[1]</sup>.传统云计算模式需要上传海量级数据到云服务器,但由于云服务器距离终端设备较远,传输过程中的传输速率、能量损耗、响应延迟、网络干扰、数据安全等问题都难以避免<sup>[2]</sup>.尽管云计算拥有强大的计算能力,可以解决终端设备无法完成大量计算以及设备电池损耗的问题,但是随着智能终端、新型网络应用和服务日新月异的发展,用户更加追求流畅的体验感,对数据传输速率、低延迟和服务质量的要求变高,这就使得云计算难以满足许多技术和场景的需求<sup>[3]</sup>.

并且随着 5G 的发展,终端设备和远程云服务器之间的数据交换可能会导致回程网络瘫痪,仅仅依靠云计算模式难以实现毫秒级的计算和通信延

迟.为了解决因云计算数据中心离终端设备较远带来的延迟和能耗问题,学者们提出将云的功能向网络边缘端转移.在接近终端移动设备的网络边缘端,移动边缘计算(Mobile Edge Computing, MEC)作为提供信息技术服务且具有计算能力的一种新型网络结构和计算范式出现了<sup>[4]</sup>.

MEC并不是取代云计算,而是作为云计算的延伸<sup>[5-7]</sup>.与云计算不同,MEC中终端设备的计算任务卸载到距离终端设备更近的边缘服务器上运行,并且边缘服务器可以提供计算、内容缓存等功能.分布在网络边缘上的服务器(也称计算节点、边缘节点)可以减轻终端设备计算压力,降低与云计算的集中式数据中心交互的频率,还可以显著减少消息交换中的等待时间<sup>[8-9]</sup>.由于边缘服务器有一定的存储空间和计算能力且距离终端设备更近,计算密集型或对延迟敏感的移动终端设备可以把计算任务卸载到边缘服务器进行计算.因此,计算卸载作为 MEC 中

的关键技术之一,备受关注.

本文对近年的云计算、移动边缘计算和计算卸载进行了大量的调研.图 1 为近三年来 Google 上关于云计算、MEC 的搜索热度趋势图,搜索热度用 0~100 分表示,可见学者对云计算关注度较多,而对 MEC 的关注度较平稳.图 2 是国内外已发表的论文的数量趋势图,其中实线表示的是 Google 学术检索的论文,虚线表示的是国内期刊数据库检索的论文.近年来对于云计算的研究下降趋势比较明显,侧面反映出云计算发展过程存在问题;有关 MEC 和计算卸载的研究处于较平缓的趋势. MEC 可以解决云计算模式下延迟和能耗问题,为了保证 MEC 的性能,计算卸载方案也是亟待解决的问题.

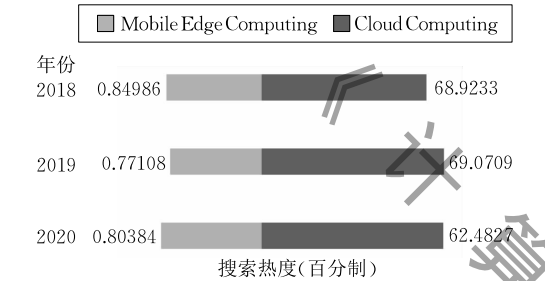


图 1 Google 搜索热度

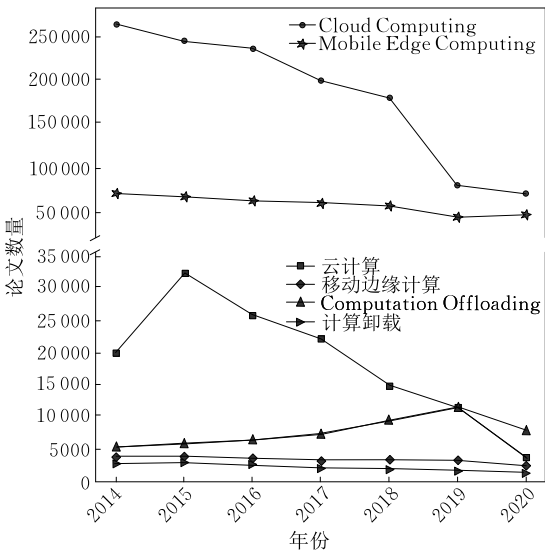


图 2 国内外相关研究趋势图

当某计算卸载方案确定后,终端设备或终端用户按照设计的方案将任务卸载到边缘服务器,然而边缘服务器的资源是有限的,如何针对有限的 MEC 服务器资源进行分配也是一个关键问题.

MEC 中计算卸载方案和边缘服务器资源分配已然逐渐成为研究热点,文献[10]对 MEC 展开了全面的综述,介绍了无线电和计算资源管理.文献[11]讨论了 MEC 的标准化和新进展,并将计算卸载的

研究按照三个方向划分并进行介绍:计算决策、MEC 计算资源分配、移动性管理.虽然这些相关研究取得了一定的成果,但由于调研时间较早,相关计算卸载方案在当时有较好的性能.例如文献[12]提出了一种调度器 LEO,在不改变推理精度的情况下,处理卸载任务和调度,最大化传感器应用程序的工作负载,其能源效率最高可以提高 19%;文献[13]提出了一个细粒度感知能量的卸载系统,最大化代码卸载的能量效益、节省能耗.但是文献[12]卸载属于 Device-to-Cloud(D2C)卸载,即终端设备卸载任务至云服务器,云服务器的计算能力强于 MEC 服务器且云服务器距离终端设备较远可能导致回传延迟较大,增加了用户的等待时间,不适用于 MEC 场景<sup>[14]</sup>;文献[13]提出的方案不具有足够的能量效率应用在连续的传感和 MEC 场景<sup>[15]</sup>.随着学者们的深入研究,近年来提出并设计许多性能更优的方案,方案在 3.3 节基于在线学习智能计算卸载方案中进行介绍.文献[16]对计算卸载技术展开综述,介绍了计算卸载、能耗最小化、服务质量(Quality of Service, QoS)、服务体验质量(Quality of Experience, QoE)等方面,并从云边博弈与协作、启发式计算卸载两个方面对计算卸载技术方案分类介绍.文献[17]针对移动边缘网络中 MEC 架构、计算卸载、边缘缓存、服务编排等方面的问题展开综述,按照完全卸载和部分卸载分别介绍了卸载方案算法;而文献[18]针对基于机器学习的方案对卸载方案进行综述,将方案分为基于强化学习的机制、基于监督学习的机制、基于非监督学习的机制三类进行介绍.与文献[11,16-18]不同,本文从技术上将方案分为基于启发式算法的传统计算卸载方案和基于在线学习的智能计算卸载方案,从三个目标来介绍传统计算卸载方案,并介绍最新基于人工智能技术的保护用户隐私数据的计算卸载方案,同时提出了计算卸载技术面临的挑战和未来研究方向.

综上所述,虽然国内外专家和学者对计算卸载方案进行了一定的研究,但缺乏从 MEC 角度进行系统化的理论梳理.本文详细介绍了目前有关 MEC 计算卸载方案的相关算法设计思路和效果,并总结了有关计算卸载方案的最新研究进展成果.通过大量的调研,从全新的角度并按照计算卸载技术的算法设计方式对计算卸载方案分类综述.本文更进一步地详细介绍了目前 MEC 中计算卸载技术的应用场景、面临的挑战和未来研究方向,对该领域的未来研究具有重要意义.

本文第 2 节介绍 MEC 的基本概念和架构;第 3

节介绍两类计算卸载方案——基于启发式算法的传统计算卸载方案和基于在线学习智能计算卸载方案;第4节介绍计算卸载方案确定后随之带来的 MEC 服务器资源分配问题;第5节介绍计算卸载技术的应用场景;第6节提出 MEC 中计算卸载技术面临的挑战;第7节探讨计算卸载的未来研究方向;在第8节对整篇文章进行总结。

## 2 MEC 基本概念和架构

在本节详细介绍 MEC 的背景、基本概念和架构。首先介绍了欧洲电信标准化协会 (European Telecommunications Standards Institute, ETSI) 对 MEC 的定义;再从服务器硬件、服务器部署位置、系统管理、延迟等方面对 MEC 和云计算进行对比;为了将 MEC 与其他近端云计算模式概念进行对比,按照时间顺序介绍三种近端云计算模式的概念;最后介绍了 ETSI 提出的 MEC 的系统架构。

### 2.1 MEC 基本概念

2014 年,ETSI 提出 MEC 的概念,并被定义为“在移动网络边缘提供 IT 服务环境且具有云计算能力的新平台”<sup>[19]</sup>。2016 年,ETSI 把 MEC 扩展为 Multi-Access Edge Computing,即一种多通路边缘计算,将访问通路从移动通信网络进一步延伸至其他接入网络(Wi-Fi、有线等)。MEC 的定义修改主要是侧重商业利益中访问通路变化,其针对移动通信场景的性质并没有改变,同时,我们调研中发现大多学者在研究计算卸载方案时都是围绕 Mobile Edge Computing 展开,所以本文仍采用 Mobile Edge Computing 的说法(后简称 MEC)<sup>[5,10-11]</sup>。

MEC 具有一定的计算能力、可以提供低延迟的服务并且支持移动性<sup>[20-21]</sup>。并且,MEC 可以持续地管理和控制终端用户中的信息<sup>[22]</sup>。在服务器分布上,MEC 服务器主要是部署在靠近终端设备的网络边缘的基站中。通过将终端设备的任务卸载到边缘服务器执行,越来越多的新兴移动应用程序将从 MEC 中受益<sup>[23]</sup>。

MEC 支持 3G/4G/5G、有线网络和 WiFi 无线等场景,其潜在应用主要包括:计算卸载、内容缓存、服务内容优化、协作计算等<sup>[24]</sup>。MEC 可以为工业、娱乐、个人计算和具有计算和存储能力的其他应用提供优势<sup>[25]</sup>。

### 2.2 MEC 与云计算模式对比

与云计算相比,MEC 可以视为云计算的延伸,

因为 MEC 可以将云计算模式延伸到网络边缘,以弥补云计算中数据存储的安全性不足和服务的高延迟的缺陷<sup>[26]</sup>。MEC 和云计算系统在计算服务上,与终端用户的距离和延迟性等方面存在显著差异。与云计算相比,MEC 的优点是:延迟更低,能够节省移动设备能耗,并保护移动应用程序的隐私和增强安全性。

在表 1 中,从服务器硬件和位置、部署方式等方面对 MEC 与云计算进行对比。MEC 的服务器硬件设备较终端设备而言较大,而相对于作为大型数据中心的云计算而言,其资源较少。在 MEC 中,服务器通常是运营商部署在距离终端用户较近的位置,例如小型机房、基站等;但云计算通常由 IT 公司内部部门部署在独立大型的建筑里。MEC 适合计算量高、对延迟敏感的应用,其距离终端用户或终端设备比较近,延迟低。而云计算计算能力相比于 MEC 要高,但其距离终端用户或终端设备较远,延迟较高,不适合对延迟敏感的应用。

表 1 MEC 与云计算系统对比

	MEC	云计算
服务器硬件	资源适中的小型数据中心	大型数据中心
服务器位置	在小型机房和基站距离终端用户近	在大规模的 建筑里距离终端用户远
部署	由运营商进行部署	由 IT 公司进行部署
系统管理	分层控制(集中/分布式)	集中控制
延迟	少于几十毫秒	大于一百毫秒
应用程序	计算量高的应用和要求低延迟应用,例如自动驾驶	计算量高但对延迟要求不高的应用,例如在线社交网络

### 2.3 三种近端云计算模式

近年来,学者们和一些组织机构陆续提出了一些近端云计算模式。2005 年,Zhang 等人<sup>[27]</sup>提出透明计算的概念,通过距离终端较近的服务器部署终端所需要的操作系统和应用,终端向服务器请求服务就不需要考虑其自身的存储空间和计算能力等问题。

2011 年,Cisco 公司提出一个新的研究领域,称为雾计算(Fog Computing)<sup>[28-30]</sup>。他们把雾计算定义为一个高度虚拟化的平台,可以在终端设备和云计算数据中心之间提供服务,但不完全处于网络边缘<sup>[31]</sup>。雾计算的服务器可以连续部署在终端和云计算数据中心之间。

2012 年,学术界提出了露计算。Wang 等人<sup>[32]</sup>进一步提出了关于露计算的“云-露”架构,为了使 Web 站点可以在无网络连接的情况下被访问。露服务器部署在用户电脑的 Web 服务器上。

上述三种近端云计算模式针对不同的应用场景,与 MEC 的设计方式和服务器部署位置有所差别. MEC 主要针对移动用户,将服务器部署在靠近设备终端,以减少网络延迟等.

2.4 MEC 架构

图 3 展示了 MEC 基本三层架构,即云-边-端三层结构,分别是云层(云计算)、边缘层(MEC 服务器)、终端层(移动终端设备). 在 MEC 中,移动终端设备不可以与服务器直接通信,需要通过终端层的基站或无线接入点与 MEC 服务器进行通信. MEC 服务器部署在更靠近终端设备的边缘层,可以提供计算和缓存服务,避免了终端设备所有任务都向远程云端请求服务造成的延迟和能耗问题.

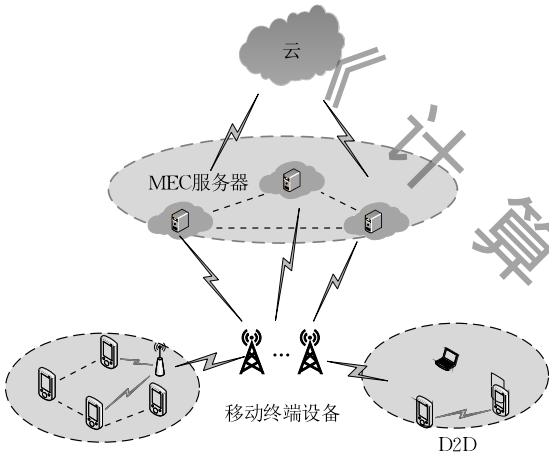


图 3 MEC 基本三层架构

如图 4,ETSI 提出了 MEC 架构结构,其由功能元素和允许它们之间相互作用的参考点组成<sup>[33]</sup>. 该 MEC 架构主要由终端、移动边缘系统层管理、移动边缘服务层管理、移动边缘服务器四部分组成.

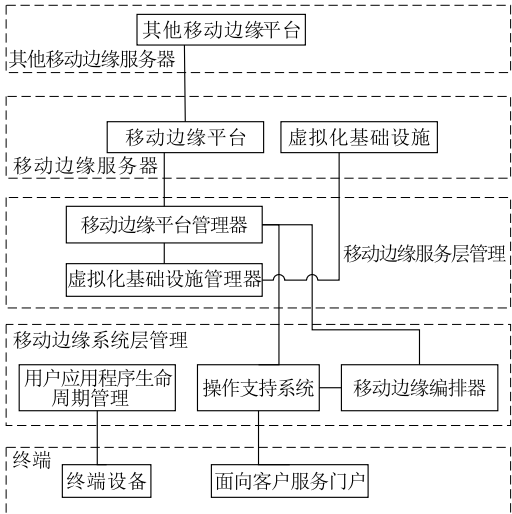


图 4 MEC 架构

移动边缘服务层管理由移动边缘平台管理器和虚拟化基础设施管理器组成. 移动边缘平台管理器连接移动边缘服务器的边缘平台,所有的移动边缘服务器通过移动边缘平台相连;虚拟化基础设施管理器负责分配、管理和释放位于移动边缘服务器内的虚拟化计算和存储资源. 而移动边缘系统层管理主要功能是管理移动终端设备用的应用程序生命周期、操作支持系统、移动边缘编排器,其中操作支持系统与移动终端设备的面向客户服务门户相连.

随着 MEC 架构越发成熟,将 MEC 架构应用于物联网系统,可以解决传统物联网系统中安全策略无法有效解决的内部攻击问题<sup>[34-35]</sup>.

3 计算卸载的研究

在本节介绍计算卸载技术相关研究. 首先,介绍了计算卸载的基本概念、流程以及计算卸载的三种情况. 然后,通过对近几年计算卸载方案的研究工作进行对比分析,按照算法的设计思路和应用技术的不同,将计算卸载方案分为两类介绍——基于启发式算法的传统计算卸载方案和基于在线学习智能计算卸载方案. 本文提出这种分类依据,可以让读者更清晰地了解计算卸载方案异同点、优缺点和最前沿的计算卸载解决方案. 基于启发式算法的传统计算卸载方案主要是先确定一个优化目标,再通过设计启发式算法来接近最优解. 针对基于启发式算法的传统计算卸载方案按照方案的不同优化目标,从最小化延迟时间、最小化能耗、权衡延迟时间和能耗这三个方向介绍传统的计算卸载方案. 随着人工智能技术的不断发展,相关技术应用到计算卸载方案中,使得整个系统有较好的性能.

与基于启发式算法的传统计算卸载方案分类介绍方法不同,没有将基于在线学习的智能卸载算法按照延时、能耗的维度划分,因为大部分智能卸载方案不以延时或能耗为单一优化目标,而是综合考虑智能卸载的性能. 基于在线学习的智能卸载方案主要是通过人工智能相关技术来训练卸载模型从而设计卸载方案,同时应用联邦学习模型可以解决基于启发式算法的传统计算卸载方案中存在的隐私数据及用户移动性等问题. 本文从应用的主要人工智能技术介绍基于在线学习的智能卸载方案,向读者介绍最前沿的计算卸载的研究方案.

3.1 计算卸载的基本概念

在 MEC 中,计算卸载是指计算任务从终端设

备卸载到边缘端或云端. 由于边缘层服务器资源和计算能力有限, 计算复杂的任务应卸载到云层进行处理. 利用 MEC 服务器的资源可以减少终端设备自身计算负担, 将终端设备的计算任务卸载到边缘层可以节省其自身能耗、加快计算速度. 计算卸载有三个重要部分: 卸载决策、卸载的任务量以及哪些计算任务应该卸载<sup>[36]</sup>. 如图 5 所示, 计算卸载第一步需要判断本地设备的请求服务是否合法, 若不合法则在本地进行计算, 只有当服务是合法的, 才会进行下一步判断, 即 MEC 服务器是否还有可用的资源<sup>[37]</sup>. 若边缘层有可用的资源则根据设计的算法进行资源分配、更新资源库、计算卸载; 若没有可用资源则只能在本地计算.

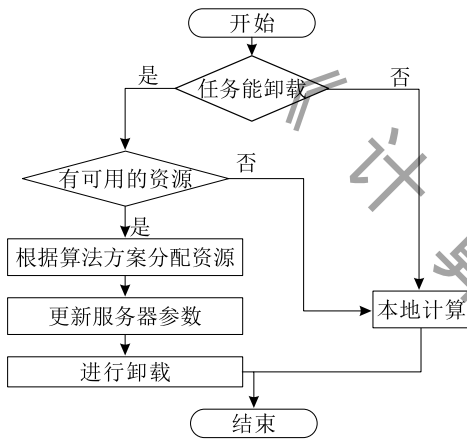


图 5 计算卸载的流程图

如图 6, 按照任务的卸载情况, 计算卸载可分为以下三种情况:

- (1) 完全本地执行 (无需卸载至服务器). 全部计算任务由本地设备完成;
- (2) 完全卸载. 全部计算任务由边缘服务器或云服务器处理;
- (3) 部分卸载. 部分计算任务由本地设备进行处理, 剩下的任务卸载到边缘层和云层, 由边缘服务器和云服务器进行处理.

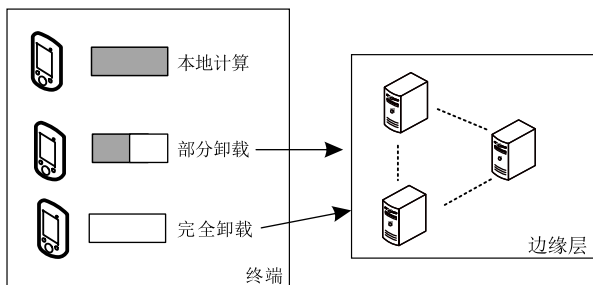


图 6 计算卸载的三种情况

### 3.2 基于启发式算法的传统计算卸载方案

通过对基于启发式算法的传统计算卸载方案的

研究工作调研, 笔者发现此类研究方案主要是从三个不同的优化目标开展研究工作: (1) 最小化延迟时间; (2) 最小化能耗; (3) 权衡延迟时间和能耗. 本文将基于启发式算法的传统计算卸载方案分为以上三类, 对目标相同的计算卸载方案进行对比, 分析方案的异同点和优缺点. 相关工作针对不同的 MEC 应用场景要求 (例如单移动终端设备、多移动终端设备) 设置不同的优化目标, 从而在某一场景下得到最优计算卸载方案.

#### 3.2.1 最小化延迟时间

Liu 等人<sup>[38]</sup>将计算卸载问题转化为马尔可夫决策过程, 根据传输单元的传输状态、设备本地处理单元的执行状态和任务缓冲区的队列状态来调度计算任务, 方案图如图 7. 作者通过分析任务的平均延迟时间和移动设备的平均能耗, 旨在功率限制的条件下最小化延迟时间, 由一维搜索算法找到最佳的计算卸载策略. 作者提供仿真结果已经证明与基准策略相比, 所提出的最佳随机任务调度策略有更短的平均执行延迟.

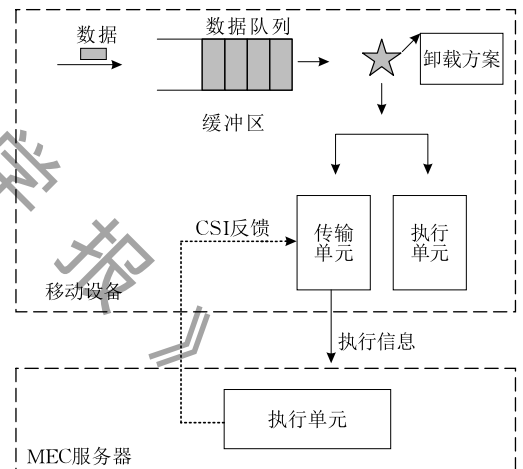


图 7 卸载系统方案图

该卸载模型的优点是通过将任务卸载到边缘端执行, 大大减少了本地设备的执行时间, 但是其缺点是终端设备依赖 MEC 服务器回传的信道状态信息 (Channel State Information, CSI) 反馈从而根据反馈情况做出卸载决定, 如果信道状态不佳会导致终端设备无法获取或等待较长时间才获取到是否可以卸载的信息, 从而增加额外的通信成本和延迟, 并且文章中没有讨论和分析其方案的复杂度和造成的额外通信成本<sup>[11]</sup>.

Mao 等人<sup>[39]</sup>提出了基于具有绿色能源收集的 MEC 系统<sup>[40]</sup>的计算卸载策略. 作者以延迟时间和任务失败的成本作为性能指标, 提出了基于 Lyapunov



优化的动态的在线计算卸载算法,该算法复杂度较低且在卸载时考虑了中央处理器(Central Processing Unit,CPU)周期频率以及传输过程中损耗能量.该算法的优点是:计算卸载方案单纯取决于系统状态,而不需要任务、无线信道和能量收集进程等信息,算法实现了资源的实时分配,降低延迟,但是由于数据是完全卸载的,其适应性很弱.

3.2.2 最小化能耗

在上一小节中,介绍了以最小化延迟时间为目标的计算卸载方案,然而某些应用场景中的系统在可接受的延迟范围内更注重能耗问题,接下来介绍以最小化能耗为目标的计算卸载方案.

You 等人<sup>[41]</sup>考虑了 MEC 中多个终端设备卸载到单边缘服务器的资源分配问题,并将问题转化为凸优化问题,在延迟时间的约束下最小化移动设备能量消耗.作者基于无限边缘服务器计算能力、有限边缘服务器计算能力两种情况,首先证明最优解存在在一个阈值,再根据用户的信道增益和本地计算能量消耗为终端设备产生优先级,使优先级高于阈值的终端设备进行完全卸载,低于阈值的终端设备部分卸载.其中计算阈值需要迭代计算,为了降低复杂度,作者提出了次优资源分配算法,结果表明使用该算法性能接近最优.该算法通过考虑用户的卸载任务的优先级来决定在边缘层处理的顺序,是一个很好的方案,但是也应该考虑队列等待导致的延迟问题.

同样是最小化设备的能耗,Zhao 等人<sup>[42]</sup>在移动终端设备的 MEC 系统中,为了使智能移动设备的能耗最小化,优化了卸载选择、无线电资源分配和计算资源分配问题.将能量消耗最小化问题定义为受特定应用约束的混合非线性规划问题(Mixed Integer Nonlinear Program,MINLP).为了解决这个问题,作者提出了一种基于线性化技术的分支定界(Reformulation-Linearization-Technique-based Branch-and-Bound,RLTBB)方法.考虑到该方法的复杂性,作者设计基于 Gini 系数的贪婪启发式算法(Gini Coefficient-based Greedy Heuristic,GCGH),通过将问题转化为凸问题来解决多项式复杂度的 MINLP 问题.仿真结果表明 RLTBB 和 GCGH 在进行计算卸载时具有显著的节能效果.

经典的三节点 MEC 系统由一个终端节点、一个辅助节点和一个边缘节点组成.Cao 等人<sup>[43]</sup>提出了经典三节点 MEC 系统中联合计算和通信协作的方法.针对有限长度块上的用户延迟时间约束计算,

作者提出了一种实现联合计算和通信协作的协议.在该设置下,联合优化了终端节点和辅助节点的任务划分和时间分配,以及卸载的传输功率和本地计算设备的 CPU 频率,从而在终端设备的延迟时间约束下最小化总能耗.作者通过凸优化方法得到最优解.数值结果表明,与其他方案相比,该方法显著提高了用户的计算能力和能源效率.但是缺点是只考虑了经典的三节点 MEC 系统,在实际场景下,远不止三个节点.

与上述方案不同,Lyu 等人<sup>[44]</sup>提出了一个新的云、边缘云和物联网设备的架构,并提出了一个允许轻量级的请求的框架来解决提出的可伸缩性问题.由于没有设备之间的协调,通过将延迟需求封装在卸载请求中,该框架可以在终端物联网设备和计算服务器上分别运行,如图 8 所示.然后设计了一个选择性卸载方案来最小化设备的能量消耗,通过允许设备自指定或自拒绝卸载,可以进一步降低信令开销.仿真结果表明,提出的选择性卸载方案能够满足不同业务的延迟需求,降低终端物联网设备的能耗.该方案优点是可以使得设备根据自身实际情况选择是否卸载.

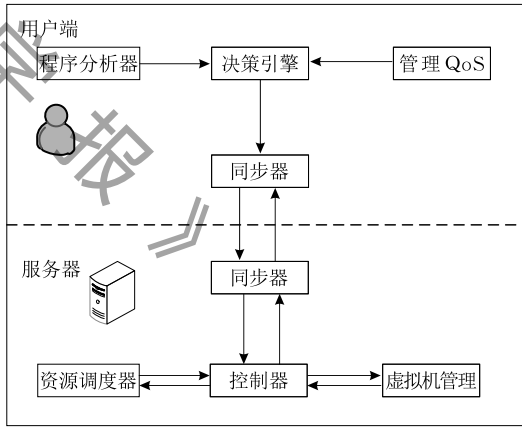


图 8 MEC 计算卸载框架图

MEC 中网络大多是采用蜂窝网络与骨干网相结合的技术,存在访问方式单一、拥塞高等缺点.鉴于加强网络集成和融合领域的研究以支持 MEC 向 5G 发展的重要性,授权集成光纤-无线(Fiber-Wireless,FiWi)接入网具有 MEC 能力成为可能<sup>[45]</sup>.Guo 等人<sup>[46]</sup>引入了混合光纤-无线网络,并提出了一种采用光纤-无线接入网的架构.作者首先提出了一个在满足移动设备计算延迟时间约束下的云边协同计算最优解(Cloud-MEC Optimal Collaborative computation Offloading,CMOCO)问题,并证

明了该问题是 NP 难的. 作者提出了三种方案: 博弈论协同计算卸载方案、最优枚举协同计算卸载方案、协同计算卸载方案. 通过实验表明, 三种方案均能有效地降低移动设备的能耗. 随着迭代次数的增加, 博弈论协同计算卸载方案总是能保持总能耗最小, 并收敛于一个均衡, 实验结果表明该方案收敛速度快且稳定性好.

### 3.2.3 权衡延迟时间和能耗

在上文分别介绍了以最小化延迟和最小化能耗为优化目标的卸载方案, 但在某些应用场景下, 对延迟和能耗都有一定的要求, 需要综合考虑延迟时间和能耗, 故将权衡延迟时间和能耗作为优化目标.

与传统的卸载方案<sup>[12-13]</sup>不同, Chen 等人<sup>[47]</sup>针对多信道无线干扰环境下 MEC 多用户计算卸载问题展开研究. 作者首先证明求解计算卸载方案最优解是 NP 难的问题, 所以根据分布式系统特点结合博弈论设计卸载策略. 作者首先证明提出的博弈模型存在纳什均衡并设计了基于分布式的计算卸载算法. 其次, 作者考虑了多信道多用户计算卸载场景. 数值结果表明, 该算法可以实现计算分流功能, 并且随着用户数量的增加具有良好的扩展性, 且能耗节省 40%. 该方案采取了经典的博弈论求解问题, 算法适应度高, 在不同场景下均能保持性能.

与文献<sup>[47]</sup>不同, Dinh 等人<sup>[48]</sup>提出从一个移动设备卸载任务到多个 MEC 服务器的优化框架, 目标是通过联合计算卸载和移动设备的 CPU 频率来最小化总任务执行延迟和移动设备能耗. 文章考虑了两种情况, 固定 CPU 频率和弹性 CPU 频率, 作者证明该问题是 NP 难的. 对于固定 CPU 频率, 作者提出了基于线性松弛的方法和基于半正定松弛 (Semidefinite Relaxation, SDR) 的方法, 对于弹性 CPU 频率, 作者提出了基于穷举搜索的方法和基于 SDR 的方法. 仿真结果表明, 基于 SDR 的算法性能接近最优. 该方案在考虑多个边缘层服务器和弹性 CPU 频率的情况下, 在能量消耗和任务执行延迟方面都有明显的性能改进. 最后, 作者还证明了移动设备的 CPU 范围会对计算卸载产生影响. 但该方案缺点是只考虑了单个移动设备进行计算卸载的情况.

为了最小化延迟的同时节省终端设备的能耗, Chen 等人<sup>[49]</sup>利用软件定义网络 (Software Defined Network, SDN) 的思想, 研究在超密集网络 (Ultra-Dense Network, UDN) 中 MEC 的计算卸载问题. 作者将计算卸载问题表示为 NP 难的混合整数非线性

规划问题. 为了解决上述问题, 作者将这个优化问题转化为两个子问题, 即任务卸载和资源分配, 并提出了一种有效的软件定义任务卸载 (Software Defined Task Offloading, SDTO) 的计算卸载方案. 实验结果表明, 与随机计算任务卸载方案、均匀计算任务卸载方案相比, 作者提出的方案可以减少 20% 的任务持续时间, 节省 30% 的能耗.

Ning 等人<sup>[50]</sup>从单终端设备计算卸载的问题开始研究, 由于单设备的 MEC 资源不受限制, 它可以用分支定界算法求解. 随后, 作者考虑移动终端设备之间的资源竞争, 将多终端设备计算卸载问题转化为混合整数线性规划 (Mixed Integer Linear Programming, MILP) 问题, 设计了一种迭代启发式 MEC 资源分配算法 (Iterative Heuristic MEC Resource Allocation, IHRA). 通过仿真实验, 结果表明该算法在执行延迟和卸载效率方面都优于对比方案.

上文介绍了考虑不同优化目标的计算卸载的相关策略, 现有的许多工作也都集中在研究计算卸载方案, 但是服务缓存是 MEC 的一个同样重要的主题, 但其很少受到关注. 服务缓存是指在边缘服务器中缓存应用服务及其相关的数据库, 从而使得边缘服务器能够执行相应的计算任务. 由于只能同时在资源有限的边缘服务器中缓存少量应用程序服务, 因此必须选择要缓存的服务, 以最大程度地提高移动边缘计算性能.

在这样的背景下, Xu 等人<sup>[51]</sup>研究了密集蜂窝网络中研究较少的 MEC 动态服务缓存问题, 提出了一种针对 MEC 高效率的在线服务缓存算法 (Online service caching for mobile Edge computing, OREO), 该算法联合优化计算卸载和动态服务缓存问题. 图 9 展示了作者提出的 MEC 联合计算卸载和边缘缓存框架. 该算法是基于 Lyapunov 优化和 Gibbs 采样开发的, 可以在线运行而无需将来的信息, 并且可以实现接近最佳的性能. 仿真结果表明, OREO 算法可以减少移动设备的计算延时, 与此同时保持系统较低的能耗. 该方案优点是将服务缓存和计算卸载同时考虑, 能够提高 MEC 系统性能. 超高密度异构网络 (Heterogeneous Networks, HetNets) 的部署被认为是下一代无线网络智能应用的基础. 在超高密度的 HetNets (例如宏蜂窝和小蜂窝中的小规模异构边缘服务器) 中对于延迟关键型智能应用, 降低下行链路和上行链路的通信时延应该引起更多的关注.



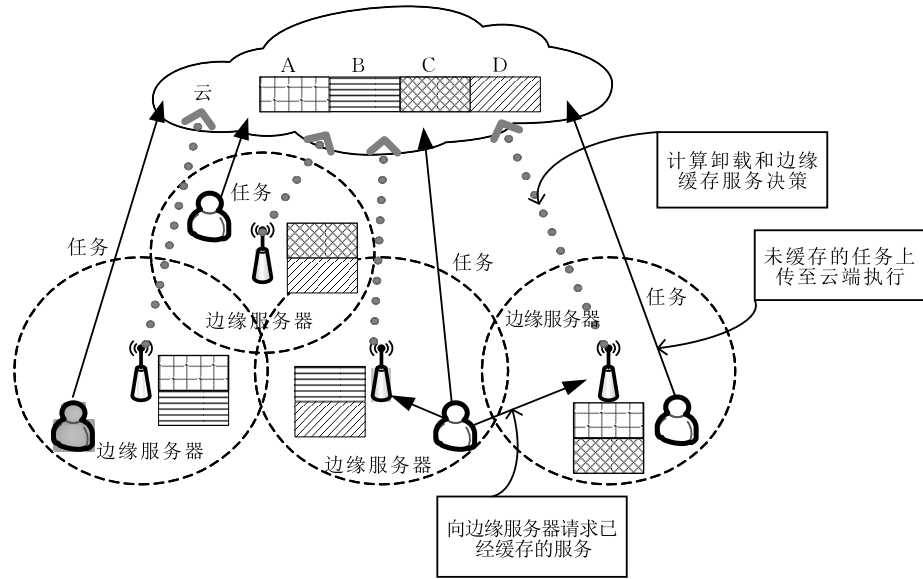


图 9 MEC 联合计算卸载和边缘缓存框架图

Zheng 等人<sup>[52]</sup>提出联合下行链路和上行链路的边缘计算卸载以及通信和计算资源分配,然后将计算卸载优化问题转化为最小化任务的总延迟时间,同时节约移动设备的能耗。作者提出具有资源分配的联合下行链路和上行链路计算卸载问题是一个难以处理的混合二进制整数规划问题。将规划问题转化为资源分配和计算卸载两个问题,提出了联合合卸载算法。数值结果验证了该算法在时延和系统节能方面的有效性。表 2 中总结并对比了基于启发式算法的传统计算卸载方案。笔者将研究按照算法目标不同进行分类,主要有三类目标:最小化延迟时间、最小化能耗或者权衡延迟时间和能耗。学者们大部分都是将计算卸载这个 NP 难的问题,按照一定的假设情况,转换成可解的问题模型后设置自己要优化的目标,然后求解问题,最后通过仿真实验来表明

提出的算法的优越性。许多学者将问题转化成马尔可夫决策、混合整数线性规划等经典问题求解,在学者所研究移动边缘计算的场景下都有很好的效果。学者们在考虑完全卸载和部分卸载方案时,根据实际应用场景下的任务是否可拆分来进行,若计算任务是可以拆分的则考虑部分卸载,若计算任务不可以拆分则考虑完全卸载。在实际应用场景下,多边缘设备情况比较符合现实,学者在研究了单个边缘设备情况后,扩展研究更符合真实场景下的多边缘设备计算卸载方案。学者们提出了新颖的基于启发式算法的计算卸载方案,每个方案在各自的场景下,都达到了各自的优化目标。但是,对于大规模的 MEC 系统或优化问题是一个 NP 难的问题的情况,用优化方法来设计 MEC 的计算卸载决策可能需要较长的时间。

表 2 基于启发式算法的计算卸载方案对比

工作	期刊/会议	目标	卸载方式	解决方法和贡献	场景	效果
Liu et al. 2016 <sup>[38]</sup>	IEEE International Symposium on Information Theory	降低延迟	完全卸载	提出一维搜索算法寻找最优任务卸载策略	单 ED	缩短 80%
Mao et al. 2016 <sup>[39]</sup>	IEEE Journal on Selected Areas in Communications	降低延迟	完全卸载	基于 Lyapunov 优化的低复杂度算法,得到有效的卸载策略	单 ED	缩短 64%
You et al. 2016 <sup>[41]</sup>	IEEE Global Communications Conference (GLOBECOM)	降低能耗	部分卸载	凸优化的分布式迭代算法来计算用户卸载的优先级	多 ED	N/A
Zhao et al. 2017 <sup>[42]</sup>	IEEE Access	降低能耗	部分卸载	作者设计一种基于 Gini 系数的贪婪启发式算法,用凸问题来解决多项式复杂度的 MINLP 问题	多 ED	N/A
Cao et al. 2018 <sup>[43]</sup>	16th International Symposium on Modeling and Optimization in Mobile	降低能耗	部分卸载	提出了一种实现联合计算和通信协作的四槽协议,用凸优化方法求解	单 ED	N/A
Lyu et al. 2018 <sup>[44]</sup>	IEEE Network	降低能耗	部分卸载	开发出轻量级的请求和允许框架,提出选择性卸载方案	多 ED	N/A

(续 表)						
工作	期刊/会议	目标	卸载方式	解决方法和贡献	场景	效果
Guo et al. 2018 <sup>[46]</sup>	IEEE Transactions on Vehicular Technology	降低能耗	部分卸载	提出并采用 FiWi 接入网的架构,用近似协同和博弈论协同计算卸载方案来解决云边协同卸载问题	多 ED	N/A
Chen et al. 2016 <sup>[47]</sup>	IEEE/ACM Transactions on Networking	权衡延迟和能耗	部分卸载	采用博弈论方法解决分布式的高效计算卸载,设计可达到纳什均衡的计算卸载算法	多 ED	能耗节省 40%
Dinh et al. 2017 <sup>[48]</sup>	IEEE Transaction on Communications	权衡延迟和能耗	完全卸载	提出了两种基于 SDR 的方法用于固定和弹性 CPU 频率	单 ED	N/A
Chen et al. 2018 <sup>[49]</sup>	IEEE Journal on Selected Areas in Communications	权衡延迟和能耗	部分卸载	采用混合整数非线性规划问题,提出高效 SDTO 算法卸载方案	多 ED	延迟缩短 20% 能耗节省 30%
Ning et al. 2018 <sup>[50]</sup>	IEEE Internet of Things Journal	权衡延迟和能耗	部分卸载	设计了一种迭代启发式 MEC 资源分配 IHRA 算法来动态地进行计算卸载	单/多 ED	延迟缩短 30% 能耗 N/A
Xu et al. 2018 <sup>[51]</sup>	IEEE INFOCOM 2018-IEEE Conference on Computer Communications	权衡延迟和能耗	部分卸载	提出一种基于 Lyapunov 优化和 Gibbs 采样开发的针对 MEC 高效率的在线服务缓存算法	多 ED	N/A
Zheng et al. 2019 <sup>[52]</sup>	Mobile Networks and Applications	权衡延迟和能耗	部分卸载	联合下行链路和上行链路的计算卸载,提出了一种高效联合卸载算法	多 ED	延迟分别缩短 47.3%、28.9% 能耗分别减少 47.2%、33.7%

### 3.3 基于在线学习智能计算卸载方案

在上一节,介绍了基于启发式算法的计算卸载方案,在本节将给大家介绍智能的新型计算卸载方案.在 MEC 中,由于边缘服务器的资源是有限的,终端设备向边缘服务器卸载的任务也是有限的,新颖智能卸载方案结合时下可靠的人工智能技术,例如深度学习、Q-learning、深度强化学习(Deep Reinforcement Learning, DRL)、联邦学习(Federated Learning, FL)等,可以更好地保证计算卸载服务的性能.将人工智能相关技术运用到计算卸载问题中,可以使其更智能化,从而减少卸载过程中能量损耗、降低延迟时间.同时通过利用小规模 MEC 系统进行神经网络的训练再得到大规模 MEC 系统的策略采样,可以解决基于启发式算法的传统计算卸载方案不适用于大规模的 MEC 系统的情况.没有将基于在线学习的智能卸载算法按照延时、能耗的维度划分,是因为大部分智能卸载方案不以延时或能耗为单一优化目标,而是综合考虑智能卸载的性能.其优化算法的主要目的是在保证用户隐私前提的条件下实现高效率的计算卸载策略的设计,同时也能够满足实际场景的延时要求和能耗约束.

针对目前高延迟和高能耗对实时移动物联网应用产生的负面影响,Alam 等人<sup>[53]</sup>提出了在接近用户终端网络的 MEC 中的计算卸载方案.移动设备的移动性、异构性和地理分布是 MEC 计算卸载方面的几个挑战.为了处理海量移动设备对计算资源的需求,作者提出了一种基于深度 Q-learning 技术的自主管理框架,可通过马尔可夫决策过程建模和

深度强化学习求解问题.仿真实验考虑了资源需求强度和终端用户设备的移动性,实验表明所提出的基于自主深度 Q-learning 学习的方法通过最小化服务计算的延迟显著提高了计算卸载的性能.作者还研究了不同计算卸载方案导致的总能耗,实验表明所提出的计算卸载方案是节能的.

将计算密集型任务卸载到边缘是支持区块链技术的 MEC 的一种可行的解决方案.然而,基于启发式算法的传统计算卸载方案无法根据不断变化的环境调整策略,无法实现长期效果.此外,现有的基于深度强化学习的卸载方法存在高维度空间收敛速度慢的问题.Qiu 等人<sup>[54]</sup>提出了一种新的基于 DRL 的在线计算卸载方案,其中同时考虑了区块链数据挖掘任务和数据处理任务.首先,将在线卸载问题表示为马尔可夫决策过程,然后,为了最大限度地提高长期卸载性能,作者利用深度强化学习来适应高度动态的环境并解决计算复杂度问题.此外,作者还将自适应遗传算法引入到深度强化学习的探索中,有效地避免了无用的探索,在不降低性能的情况下加快了收敛速度.作者从传输时间和能耗方面评估了卸载方案的性能,实验结果表明,该方案在多种情况下有较好的鲁棒性.

为了解决 MEC 系统中的动态变化,Wang 等人<sup>[55]</sup>提出了使用 FL 结合的 DRL 来优化 MEC 系统中的缓存和计算卸载方案,如图 10 所示.终端设备可以选择将计算任务卸载到边缘节点,也可以选择本地执行.作者考虑到 MEC 环境中的大型状态和操作空间,将采用双深度 Q-Network(Double

Deep Q-Network, DDQN)方法. 为了保护用户的隐私, 作者提出了基于 FL 的方法, 利用终端用户上以往的数据进行训练. 对于边缘服务器缓存, DRL 的智能体(agent)决定服务器是否缓存文件, 以及缓存时要替换的文件. 有关边缘内容缓存, Yu 等人<sup>[56]</sup>提出了一种基于联合学习的主动内容缓存(Proactive Content Caching, FPCC)方案, 该方案不需要集中收集用户的数据来进行培训. FPCC 基于分层体系结构, 其中服务器使用联合平均汇总用户的更新, 每个用户使用堆叠式自动编码器上的混合过滤对其本地数据进行训练. 实验结果表明, 在不收集用户私人数据的情况下, 该方案在缓存效率方面仍然优于其他基于学习的缓存算法.

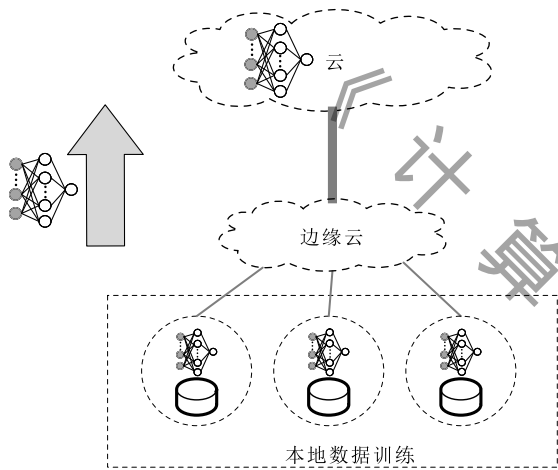


图 10 基于联邦学习的 DRL 训练模型

Ren 等人<sup>[57]</sup>认为计算卸载涉及联合和复杂的资源管理, 并且应该实时根据动态工作负载和无线电环境来确定. 作者使用部署在多个边缘节点上的多个 DRL 的 agent 来指示物联网设备上的卸载软件. 另一方面, 为了使基于 DRL 的卸载方案可行并进一步降低物联网设备与边缘节点之间的传输成本, FL 用

于训练分布式 DRL 的 agent. 实验结果表明了计算卸载方案和联合学习在动态物联网系统中的有效性.

Qian 等人<sup>[58]</sup>介绍了一种隐私感知服务放置方案, 考虑到边缘云中的资源约束, 在边缘服务器上部署用户首选的服务. 系统模型由服务各种移动设备的移动边缘云组成. 首先, 根据诸如服务请求次数之类的信息以及其他用户上下文信息(例如年龄和位置)构建用户的偏好模型. 但是, 由于这可能涉及敏感的个人信息, 因此提出了一种基于 FL 的方法来训练偏好模型, 同时将用户的数据保留在其个人设备上. 然后提出了一个优化问题, 其目标是根据用户的喜好, 在不受存储容量, 计算能力, 上行链路和下载带宽约束的情况下, 最大限度地提高边缘需求的服务数量. 然后使用贪婪算法求解优化问题, 添加最改进目标函数的服务, 直到满足资源约束为止. 该方案在边缘云上处理的服务请求数量方面, 超过了已有的计算卸载方案.

Zhang 等人<sup>[59]</sup>提出 MEC 的关键挑战是根据移动设备的移动性来决定最好在何时何地迁移任务. 目前的研究工作大多将此问题制定为顺序卸载模型, 并使用马尔可夫决策过程进行求解, 且假设用户的移动性模式已提前知晓. 但是, 很难预先获得用户的移动方式. 所以, 作者提出了一种基于深度 Q-Network (Deep Q-Network, DQN)的技术, 用于 MEC 系统中的计算卸载. 它可以从以前的经验中学到最佳的迁移策略, 而不必事先获取有关用户移动模式的信息. 在 MEC 系统中进行广泛的仿真实验, 实验表明该方案利用历史数据通过训练预测移动设备的移动轨迹, 可以更好地决策何时进行计算任务卸载.

Wang 等人<sup>[60]</sup>设计了基于边缘计算的动态卸载平台, 如图 11 所示, 将边缘层可执行的应用缓存在

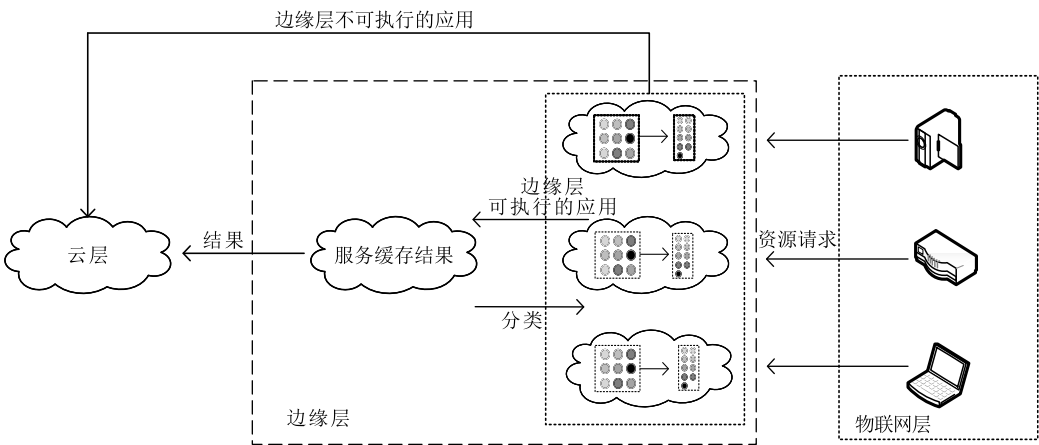


图 11 基于边缘计算的动态卸载平台

边缘层,不可执行的应用发送到云层进行计算.并提出了一种云边协同的耦合任务动态卸载算法,以确保智能动态卸载,其中所有任务都可以根据系统的实时条件在云层或边缘层卸载.作者根据应用程序的实时需求将其进一步分为四种类型:A类是计算量低且流量低的应用;B类是计算量低的高流量应用(例如视频流应用);C类是低流量应用具有较低的计算负荷(例如具有监视功能的温度模型);D类是具有高计算负荷(用于对象识别的视频处理)的高流量应用程序.作者为每种类型的应用程序均设置为合理的延迟时间,以确保系统具有更少的处理时间.文中提出了一种动态选择算法(Dynamic Switching Algorithm,DSA),首先利用 LibSVM 算法根据系统状态和应用程序需求识别不同类型的应用程序,再为每类应用设置不同的延迟时间.实验表明所提算法有较好的性能,DSA 的平均处理时间和卸载能耗分别减少了 10.3%~78.5%和 3.76%~78.6%.

Wang 等人<sup>[61]</sup>针对 MEC 中高空气球(High Altitude Balloons,HABs)网络场景展开研究,HABs

作为无线基站在空中飞行且具有较强的计算能力,用户可以将任务卸载至 HABs 进行计算处理.由于每个用户的计算任务的数据大小随时间而变化,HABs 必须动态调整用户基站关联关系、服务顺序和任务划分方案以满足用户的需求.作者提出了基于支持向量机(Support Vector Machine,SVM)的联邦学习算法,确定用户关联,优化每个用户的服务序列和任务分配.仿真实验结果表明,与传统集中式方法相比该算法的能耗和延迟加权降低 16.1%.

表 3 总结了基于在线学习智能计算卸载方案的对比,卸载方式都是部分卸载.目前智能计算卸载的研究比较少,其主要是结合深度强化学习技术、深度 Q-Learning 技术、DQN 等技术来学习终端设备和用户的历史卸载方案和卸载内容相关信息,这样在研究计算卸载方案时不用预先假设终端设备和用户的未来移动性,通过学习信息推测得到终端设备和用户的移动路径信息.当然,学习训练的结果不是百分之百的正确,正如不能保证机器学习的完全准确性一样.但这仍然是在移动边缘计算中计算卸载领域的一个重大突破.

表 3 基于在线学习智能计算卸载方案对比

工作	期刊/会议	卸载方式	主要技术	解决方法和贡献	场景
Alam et al. 2019 <sup>[53]</sup>	Future Generation Computer Systems	部分卸载	深度 Q-learning	提出了一种基于深度 Q-learning 技术的自主管理框架,可通过马尔可夫决策过程建模和深度强化学习求解	多 ED
Qiu et al. 2019 <sup>[54]</sup>	IEEE Transactions on vehicular Technology	部分卸载	DRL	一种新的基于 DRL 的在线计算卸载方案,其中同时考虑了区块链数据挖掘任务和数据处理任务	多 ED
Wang et al. 2019 <sup>[55]</sup>	IEEE Network	部分卸载	FL、DRL、DDQN	提出了使用 FL 结合的 DRL 来优化 MEC 系统中的计算卸载方案和边缘内容缓存并且采用 DDQN 方法	多 ED
Ren et al. 2019 <sup>[57]</sup>	IEEE Access	部分卸载	DRL、FL	在多个 DRL 的 agent 上指示物联网设备上的卸载软件,FL 用于训练分布式 DRL 的 agent	多 ED
Qian et al. 2019 <sup>[58]</sup>	Information Sciences	部分卸载	FL	提出了一种基于 FL 的方法来训练偏好模型,同时将用户的数据保留在其个人设备上	多 ED
Zhang et al. 2019 <sup>[59]</sup>	Future Generation Computer Systems	部分卸载	DQN	提出一种基于 DQN 的技术用于计算卸载,不必事先获取有关用户移动信息	多 ED
Wang et al. 2020 <sup>[60]</sup>	IEEE Transactions on Network Science and Engineering	部分卸载	DSA、LibSVM	提出一种 DSA 算法,利用 LibSVM 算法根据系统状态和应用程序需求对应用程序分类,再为每类应用设置不同的延迟时间	多 ED
Wang et al. 2020 <sup>[61]</sup>	IEEE Internet of Things Journal	部分卸载	SVM、FL	提出了基于 SVM 的联邦学习算法,确定用户关联,优化每个用户的服务序列和任务分配	多 ED

在利用深度学习等技术进行训练的时候,终端设备、用户数据的隐私安全也是一个需要考虑的问题.本文列举的考虑用户数据隐私和安全的方案中,都是采用了联邦学习技术,终端设备和用户不直接提供原始数据,而是提供模型要求的参数.这样,在一定程度上保护了数据的安全性.而目前也有学者质疑联邦学习本身的安全性,在这一领域也期待有进一步的研究突破.

## 4 MEC 服务器资源分配

在上文介绍了计算卸载方案,当一个方案确定后,终端用户将部分计算任务卸载到 MEC 边缘层执行.但是由于 MEC 边缘层的服务器资源有限,必须对服务器的资源进行合理分配,即用户将任务卸载到哪个边缘服务器.如何实现 MEC 服务器资源

最优分配,以最小化任务卸载到 MEC 服务器计算的等待延迟,也是计算卸载技术中的关键问题之一.若未对 MEC 服务器进行合理的资源分配,会造成资源浪费和任务等待延迟时间长等后果,从而影响计算卸载方案的效率.在本节,将介绍目前有关于 MEC 服务器资源分配和分配优化的研究工作.

Guo 等人<sup>[62]</sup>提出的分配策略是通过联合最小化移动计算任务的延迟时间(包括无线传输延迟和任务执行时间)和边缘云服务器的运行功耗来优化系统性能.为了使其易于处理,作者将问题转换为等效的离散时间马尔可夫决策过程.然后,利用经典的马尔可夫决策技术来推导算法,以便确定平稳的最佳策略.但是,这些策略在计算上过于复杂,并且涉及巨大的通信开销.因此,作者提出了一种“索引策略”,终端设备可以自主选择合适的 MEC 服务器,它可以在现有的移动网络中轻松实现,实验表明利用索引策略可以减小 7% 的能耗.

Wang 等人<sup>[63]</sup>提出并开发了边缘节点资源管理框架(Edge Node Resource Management, ENORM),如图 12.他们提出了一种新的配置和部署机制,用于连接边缘节点和云数据中心之间的通信,这样 ENORM 就可以卸载任务到边缘节点. ENORM 的开销低并且可以通过动态自动扩展机制来添加或删除资源,以有效地管理边缘节点上的工作负载.作者用延迟敏感的网络游戏验证 ENORM 的可行性,实验结果表明 ENORM 可以减少 20%~80% 之间的应用程序延迟,将边缘节点与云之间的数据传输和通信频率最多降低至 95%.

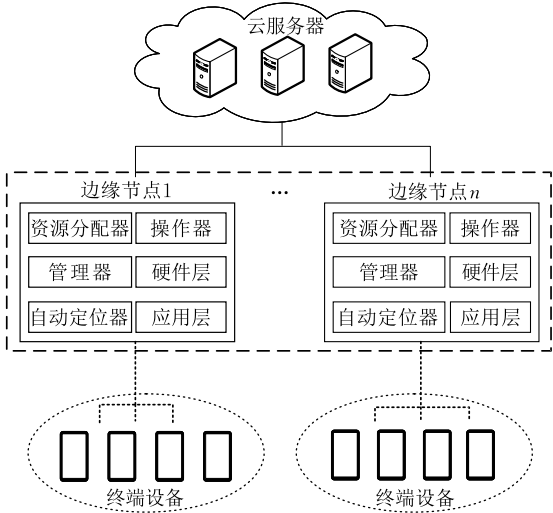


图 12 边缘节点资源管理框架

Guo 等人<sup>[64]</sup>考虑了多用户移动边缘计算系统的节能资源分配.首先,建立了两个计算效率高的模型,分别具有可忽略的边缘服务器执行时间和不可

忽略的边缘服务器执行时间,不可忽略边缘服务器执行时间的计算卸载时间线如图 13 所示.作者分别在两种模型下,通过优化分配通信和计算资源来制定总体加权和能耗最小化问题.可以忽略的边缘服务器执行持续时间的优化问题是凸的,因此作者以封闭形式获得了该问题的最优解,并利用 Johnson 算法来获得低复杂度的次优解决方案.

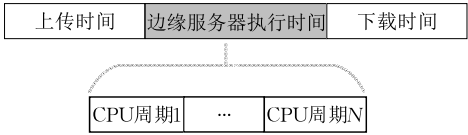


图 13 计算卸载时间线

Tran 等人<sup>[65]</sup>将问题建模为 MINLP 问题,同时考虑优化任务卸载方案、MEC 服务器的资源分配、移动设备的上行链路传输功率等问题.由于多种问题的组合性质,对于大规模网络而言,寻求符合多种问题的最佳解决方案是很困难的.为了克服上述问题,作者将原始问题分解为两部分:固定任务卸载方案的资源分配问题;最优计算卸载问题.作者利用凸优化、拟凸优化技术来解决资源优化问题,并提出了一种启发式算法,该算法在一定时间内可以达到次优解.仿真结果表明,该算法与最佳解决方案的性能接近,并且与传统方法相比,该算法显著提高了用户的卸载效率.

Zhang 等人<sup>[66]</sup>提出了一种基于 MEC 的分布式联合计算卸载和资源分配的优化方案(Joint Computation Offloading and Resource Allocation Optimization, JCORAO).作者提出了一个优化问题,给出了最优计算卸载方案、上行子信道分配、上行传输功率分配和计算资源调度.作者首先证明了该优化问题是 NP 难的,然后建立了分布式的博弈子算法,证明存在纳什均衡.作者设计资源分配算法、分布式 JCORAO 算法,通过两个子算法的相互迭代来解决优化问题.仿真结果表明,分布式 JCORAO 方案能够有效降低系统的能耗和任务完成时间,降低系统的复杂度.

边缘层资源分配是计算卸载的关键技术之一,它主要解决了有限的边缘服务器的资源分配问题和卸载任务在哪里执行的问题.在表 4 中,总结了有关资源分配的方法,学者在单边缘服务器场景下可以实现资源分配,但真实场景下多边缘服务器的情况较多.在多边缘服务器场景下,学者提出相关 MEC 框架用于资源分配和管理,通过不同的技术来解决资源优化问题,学者们提出的启发式算法和优化配置方案,均能提高系统的总体性能.

表 4 资源分配方法对比

工作	期刊/会议	解决方法和贡献	场景	效果
Guo et al. 2016 <sup>[62]</sup>	IEEE International Conference on Communications	提出了一种索引机制,终端设备可以找到合适的服务器进行计算卸载	单边缘服务器	利用索引策略减小了 7% 的能耗
Wang et al. 2017 <sup>[63]</sup>	IEEE Transactions on Services Computing	提出并开发 ENORM 框架,其通过动态自动扩展机制来添加或删除资源,以有效地管理边缘节点上的工作负载	多边缘服务器	减少 20%~80% 程序延迟,降低 95% 与云端通信频率
Guo et al. 2017 <sup>[64]</sup>	IEEE Global Communications Conference	建立了两个计算效率高的模型,分别具有可忽略和不可忽略的边缘服务器执行时间;以封闭形式获得了最优解,利用 Johnson 算法来获得低复杂度的次优解决方案	多边缘服务器	N/A
Tran et al. 2018 <sup>[65]</sup>	IEEE Transactions on Vehicular Technology	使用凸优化、拟凸优化技术来解决计算卸载和资源优化问题	多边缘服务器	N/A
Zhang et al. 2018 <sup>[66]</sup>	IEEE Access	提出了一种基于 MEC 的分布式联合计算卸载和资源分配优化方案	多边缘服务器	N/A

5 计算卸载技术应用场景

计算卸载技术作为 MEC 中关键技术之一,在众多场景均得到应用. 由于 MEC 靠近终端设备且具有一定的计算能力和存储功能的特点使其在 5G、物联网、车联网、无人机,虚拟现实/增强现实等应用领域得以应用,而其中计算卸载是一项关键任务.

5.1 物联网

学术界对物联网的定义有多种,但较为广泛接受的定义是:物联网是连接物品的互联网,而不是连接物品的网络<sup>[67]</sup>. 互联网连接各种各样物品形成物联网,可以实现物品的智能化识别和管理<sup>[68]</sup>. 物联网涉及领域广泛,如智慧城市、无线医疗、智慧电网、环境、农业等. 专注于 IoT、Machine to Machine (M2M)和工业 4.0 领域的研究机构 IoT Analytics 在 2019 年 12 月发布物联网平台报告:2020 年全球活跃的物联网设备数量将达到 100 亿台,预测 2025 年将达到 220 亿台. 这些物联网设备收集产生大量数据,若将数据全部上传至云服务中心处理会给远程云带来巨大压力而大部分物联网设备自身不具有处理数据的能力或计算能力不强,在这种情况下,将计算卸载到边缘服务器能够有效解决上述问题.

5.1.1 智慧城市

智慧城市与物联网紧密相连,智慧城市的发展离不开物联网. 物联网底层传感器收集数据,由边缘云和远程云数据中心对数据进行分析,为智慧城市发展助力. 由于传感器本身不具备分析数据的能力,通过把数据先上传至分布在靠近传感器、部署在网络边缘的边缘服务器进行初步分析和数据筛选,再上传至远程云数据中心聚合进行处理,可以降低数据传输量和带宽,延长网络寿命<sup>[69]</sup>.

5.1.2 智能设备

移动用户对移动智能设备支持的功能和服务要

求越来越高,然而随着功能或服务变得越复杂,智能设备消耗的能量就越多. 如何解决智能设备的能耗问题是十分关键的. Sucipto 等人<sup>[70]</sup>提出并实现了基于近场通信(Near Field Communication,NFC)的计算卸载框架,并设计了一种新的 NFC 的通信协议,消除了对恒定的用户交互、单向通信的约束以及对数据传输的限制. 另一种方案是 Device-to-Device (D2D)卸载,应用程序利用附近的设备或边缘服务器来提高性能和节省能耗,例如 Golkarifard 等人<sup>[71]</sup>提出了一种新型基于 D2D 的通用代码卸载系统,用于做出卸载决策.

物联网中许多智能物体,以智能手表为例,智能手表设备大小有限导致其计算能力较弱、存储空间有限、电池容量小,从而约束了其设备完成任务种类. 涉及复杂计算或功能的情况下,可以将计算卸载至边缘服务器执行. 一方面,能够扩展智能手表服务的功能,满足用户需求;另一方面,卸载至边缘服务器而不是远程云服务器,可以大幅度降低延迟和设备能耗,提升服务质量和用户对产品体验满意度.

5.1.3 智能电网

智能电网作为能源和经济发展里面的一项重要工程,保障了电力从发电厂至用户整个传输过程中的电力智能化监控和管理<sup>[72]</sup>. 智能电网由多种技术和多种基础设施构成,从过程上看,智能电网首先需要通过传感器对整个过程中涉及的重要设备进行实时监控;再通过传感器收集数据进行聚合、分析,最终对电力系统进行优化.

在上述过程中,传感器只能完成一些基本的数据过滤功能,其余任务可以先卸载到边缘服务器中进行存储和初步聚合,最终再卸载至远程云计算中心完成商业智能数据分析等任务.

5.1.4 智慧医疗

计算卸载技术也应用在智慧医疗和医疗保健的实际工作中,如 Liu 等人<sup>[73]</sup>设计了一个基于边缘计



算的用于膳食评估的食物识别系统,该系统将识别任务卸载给附近的边缘服务器或云服务器来解决延迟和能耗问题.

在新时代下,传统医疗模式逐渐走向智慧医疗模式,网络诊断能够解决地域限制、充分利用医护人员资源,为某些地区提供远程医疗服务.例如,在2020年中国抗击新型冠状病毒肺炎疫情过程中,许多医院、方舱医院都采取了智能医疗基础设施,护士通过将病人信息上传至系统,医生远程治疗,可以有效解决特殊时期医护人员紧张的问题.在医疗保健方面,移动边缘计算中计算卸载技术也有很好的应用,例如为避免中风的患者跌倒需要实时监控和反馈,这类应用数据应卸载到边缘服务器执行分析,降低延迟从而快速响应事件.

5.2 车联网

通过将车辆按照通信协议和标准用网络连接起来形成车联网,对所有车辆的动态信息进行收集分析利用,从而为运行车辆提供不同的服务.车联网对延迟的要求较高,高延迟问题会给车联网带来安全、应用和隐私数据问题<sup>[74]</sup>.车联网中行驶车辆需时刻与服务器保持连接并频繁进行数据交互,在传统的云计算服务中,云端距离车辆较远且大量的车辆作为节点接入云端会给云服务器带来通信和数据库的负载压力.而移动边缘计算可以提供实时可靠的车辆连接、通信、安全服务,通过计算卸载技术,将服务计算卸载至边缘节点进行,可以提供高效低延迟的服务质量.例如,MEC服务实现交通控制和智能停车;实时预警道路情况(前方拥堵、道路颠簸或结冰);协调车辆进行变道等.

5.3 区块链

区块链作为加密货币的基础技术已经引起了广泛的关注.但是,区块链存在巨大的可扩展性障碍,限制了其支持频繁交易服务的能力.而在MEC场景中,在网络边缘扩展分布云资源和服务存在分散管理和安全性等方面面临重大挑战.学者指出,将区块链和边缘计算集成到一个系统中,可以实现对边缘网络进行存储和计算的可靠访问和控制,从而在安全的情况下提供大规模的网络服务器、数据存储和有效性计算<sup>[75]</sup>.

将MEC应用在区块链技术上,使系统拥有大量计算资源或分布在网络边缘的存储资源,从而减轻功率受限设备的区块链存储和挖掘计算的负担.此外,边缘的链外存储和链外计算可在区块链上实现可扩展的存储和计算.

5.4 无人机

无人机是通过无线电遥控设备和程序控制的无

人飞机<sup>[76]</sup>,无论在军事还是民用领域都有广泛的应用.例如,无人侦察机、无人机航拍、无人机快递运输等.虽然无人机的应用场景广泛,但是由于技术和安全等方面的问题无人机在实际应用场景并未大规模的普及.

无人机对网络要求严格,如果采用传统基于云计算模式的数据处理系统,当数据传输至距离无人机设备地理位置较远的云数据中心处理后再返回结果,将导致高延迟从而导致服务质量下降甚至服务失败.另外,无人机在飞行或执行任务时如何避免障碍物是一个关键的安全问题.

针对无人机应用场景的难题,MEC环境提供了合适的解决方案.MEC环境提供了网络边缘端的服务,无人机可以将数据发送至边缘层先进行数据处理,若边缘端可以直接完成服务则不需要请求远程云数据中心.特别是,采用计算卸载技术将无人机采集的数据或服务数据卸载到边缘服务器上,能够有效地减少传输延迟和数据处理延迟.对于安全问题,可以利用MEC边缘节点对无人机的定位结合云计算来保障无人机飞行的安全性.

5.5 虚拟现实和增强现实

虚拟现实(Virtual Reality, VR)、增强现实(Augmented Reality, AR)通过将不同世界的景象、声音等转换到另一个世界,改变了人们的生活方式.VR通过模拟环境、感知、自然技能和传感设备等方面,形成实时动态的三维立体逼真图像,使人身临其境.AR是指通过将虚拟场景加到真实世界里,两种场景同时存在.目前一些VR/AR应用程序需要使用用户的实时位置方向等,为了保证服务质量这类应用对延迟、计算能力和带宽都有较高要求.

MEC中计算卸载技术为VR的发展提供了强有力的帮助.MEC环境下的边缘服务器可以解决VR设备计算、存储能力和电池容量不足等问题,通过将计算任务卸载到边缘服务器执行后返回结果不仅能解决上述问题,同时能够解决向云服务器请求带来的高延迟问题.利用MEC的近距离传输通道,VR系统可以摆脱有线网络的限制,推动VR系统的发展.

5.6 社交网络

计算卸载技术在社交网络中也发挥着巨大作用,以直播场景为例,直播现场有多部不同拍摄角度的摄像机.如果通过将摄像机的视频上传至云服务器处理后再传输给用户会造成严重的延迟问题,同时由于现在视频清晰度增加在传输过程中对带宽的需求提高.在MEC和5G场景下,通过计算卸载技

术,首先将视频数据传给基站附近部署边缘服务器上,进行合成和处理,形成不同分辨率的视频数据,提供给不同要求的用户,用户可以根据自身需要选择合适的分辨率频道进行观看,满足了用户的个性化需求和体验感.

## 6 计算卸载技术面临的挑战

MEC 能够显著降低移动终端设备的能耗,并且

终端设备能够实现实时应用卸载,因此备受关注.但 MEC 中计算卸载技术仍然是不成熟的技术,在将其实施到移动网络中以使其受益之前,还需要解决许多问题.在本节主要分析计算卸载技术面临的挑战,有助于理清在哪些方面是计算卸载技术需要重点解决的.将计算卸载方案技术面临的挑战整理成表 5,本节将从终端设备的移动性、边缘服务器、安全、用户隐私数据、服务异构性等方面分别介绍计算卸载方案技术面临的挑战.

表 5 计算卸载方案技术面临的挑战

技术挑战	主要问题	相关文献
终端设备的移动性	终端设备在固定 MEC 服务区域内移动导致能耗和延迟问题	/
	终端设备跨 MEC 服务区域移动时保证服务连续性和质量	[77]
边缘服务器	边缘服务器移动时如何保证服务质量	/
	边缘服务器部署模型对终端数据进行筛选	[78][79]
安全	边缘服务器安全	[80][81]
	数据安全	[82][83][84][85][86][87]
	网络安全	/
隐私保护	终端设备数据涉及隐私问题,卸载过程应保护隐私不泄露	[88][89][90][91][92]
异构性	网络环境复杂,使得卸载方案可行性不高	/
	新兴设备各式各样,卸载方案可能无法部署	/

### 6.1 终端设备的移动性

在此前研究工作中,有关终端设备的移动性主要分为三类:一是假设终端设备的移动路径;二是通过马尔可夫决策预测终端设备的移动路径;三是通过在线学习预测移动路径.终端设备移动分为在固定 MEC 服务区域内和跨 MEC 服务区域移动两种情况.

- (1)终端设备在固定 MEC 服务区域内移动
- 在固定 MEC 服务区域内,由于设备移动会导致信道质量下降,在卸载期间,传输数据所需的能量也会显著改变.与计算任务在终端设备本地执行相比,卸载到边缘服务器执行可能导致卸载过程中设备消耗的能量增加、服务期间的延迟更高等问题.

- (2)终端设备跨 MEC 服务区域移动
- 当正在进行卸载的移动设备从一个 MEC 服务区域进入到另一个 MEC 服务区域时,如何保证服务的连续性是计算卸载技术面临的挑战之一.在这种情况下,如果能够提前准确预测到移动设备的移动轨迹和路线,在跨 MEC 服务区域的某一段时间内不进行卸载数据的传输,而只在终端设备本地执行必要的计算服务或暂停计算服务.由于跨 MEC 服务区域是移动终端设备在移动过程存在的情况但跨服务区域所需要的时间较少,上述方案在理论上是可以解决存在的问题,学者可以做进一步的研究和讨论.

另一个解决方案是,在得到预测路径后,在跨越服务区域前,服务器提前将卸载任务数据传输至下一个 MEC 服务器中.例如,Ouyang 等人<sup>[77]</sup>提出了一种新颖的自适应用户管理的计算卸载方案,该方案可以优化用户的感知延迟和计算卸载的成本,并通过用户偏好进行加权.为了克服未知的终端设备未来移动信息和系统动态,将动态计算卸载问题公式化为一个关于上下文多臂赌博机 (Multi-Armed Bandit, MAB) 问题求解.

但是预测移动路径是有一定误差的,若要更精准预测路径,则需要利用机器学习等相关人工智能技术.然而,基于人工智能的方案会带来训练时间和成本等一系列新的问题.如何在计算卸载效果和路径预测中权衡最佳成本是一个挑战.

### 6.2 边缘服务器的移动性

当终端设备不再是固定不动的,而是有一定的移动路径,除了考虑跨 MEC 服务区域的问题,还需要考虑 MEC 中边缘层服务器的移动性.当边缘服务器移动时,则要考虑如何继续保持服务的连续性和满足用户对 QoS 要求.问题是:如何有效地处理边缘服务器的移动路径,将终端设备计算任务卸载到最合适的边缘服务器上执行并且满足 QoS.边缘服务器的移动会给回程带来高负载并导致高延迟,这使其不适用于实时应用程序.由于边缘服务器之

间的通信限制,边缘服务器的移动性这项工作具有挑战性. 目前有关计算卸载方案的研究工作多数仅考虑了终端设备的能耗,然而,在选择卸载方案过程中还应进一步考虑 MEC 服务器的能耗问题. 还有一个重大挑战在于如何考虑 MEC 中边缘服务器之间的回程以及在卸载过程中反映其负载和参数变化的能力. 学者可以充分利用 MEC 服务器的资源特点做进一步的研究,以增强终端设备的 QoE 并优化整个系统移动用户的性能.

6.3 终端设备的恶意数据

在物联网应用场景中,终端设备收集大量数据,其中可能包含恶意节点收集的脏数据. 这些脏数据可能会对 MEC 服务器或云服务器进行攻击,造成

服务器瘫痪. 通过在 MEC 服务器上部署相关模型,让 MEC 服务器对终端传过来的数据进行数据清洗,保留有效数据并对恶意数据进行筛选<sup>[78]</sup>. Wang 等人<sup>[79]</sup>提出了基于 MEC 的数据清洗模型,如图 14 所示,通过 SVM 建立清洗模型,并采用在线搜索的方法进行模型优化. 在传统物联网的单一场景下,终端设备的数据特征维度比较单一,终端设备的计算能力可以满足需求,但是由于 5G 和物联网的高速发展,终端设备的数据维度变得复杂使得其自己不足以支持完成数据筛选. 通过把数据筛选任务卸载至 MEC 服务器进行可以有效地解决上述问题,由于 MEC 服务器具有一定大小的存储空间,部署数据清洗模型是可行性的.

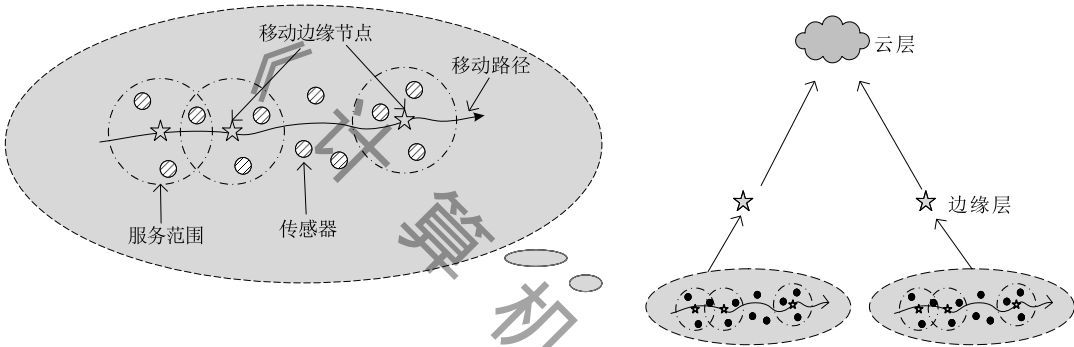


图 14 数据清洗模型

6.4 安全

安全一直是 MEC 中重要的主题,同样,在计算卸载技术中,安全也是值得研究的方向之一. MEC 中计算卸载技术现在面临诸多安全问题,例如边缘服务器安全、数据安全、网络安全等.

(1) 边缘服务器安全. 在边缘层提供服务的服务器,若受到恶意服务器攻击会造成在提供服务时会出错. 目前研究计算卸载的方案中,大多都是假设参与服务的边缘服务器都是安全且可信的,这是一个较理想的状态. 在之后的研究工作中,需要考虑实际应用情况,进一步探讨边缘服务器的安全性是有必要的. 还可以利用目前关于边缘服务器可信评估机制来保证服务器的安全性<sup>[80-81]</sup>.

(2) 数据安全. 在传统的云计算模式下,用户数据完全存储在云服务器中,并面临隐私泄露的风险<sup>[82-84]</sup>. 在数据卸载到边缘层的过程中可能被篡改,同时还会涉及隐私问题<sup>[85-87]</sup>. 数据在传输过程中可以通过加密处理,文献[86]使用的高级加密标准-里德所罗门编码(Advanced Encryption Standard-Binary Reed-Solomon, AES-BRS)编码过程. 如图 15

所示,首先将原始矩阵  $X$  转换为状态矩阵  $S$ ,如果状态矩阵  $S$  不完整,则将其填充. 其次,状态矩阵经历了几次加密,并且在每次加密之后获得了不同的状态矩阵. 将最终状态矩阵  $S$  转换为密文矩阵  $C$ . 然后针对矩阵  $C$  进行映射转换,该矩阵应用里德所罗门(Binary Reed-Solomon, BRS)算法编码. 另外,对终端设备数据的访问也是数据安全的一部分. 由于在 MEC 中,移动设备是分布式部署,若这些设备被盗之后,应考虑相关的数据即使被不法分子获得后也不会泄漏隐私数据.

(3) 网络安全. 在整个计算卸载的系统中,需考虑由于网络因素带来的安全性问题,例如防火墙系统和入侵检测系统,以免整个 MEC 计算卸载系统遭受攻击,被窃取隐私数据.

在传统云计算中,安全问题的解决方案逐步成熟,而移动边缘计算作为云计算的扩展,学者们可以借鉴传统云计算中解决相关网络安全问题的有效且成熟方案. 然而,由于 MEC 是分布式且网络环境复杂,相关方案也许并不完全适用,如何根据移动边缘计算中计算卸载技术特点设计一个有效且能够解决

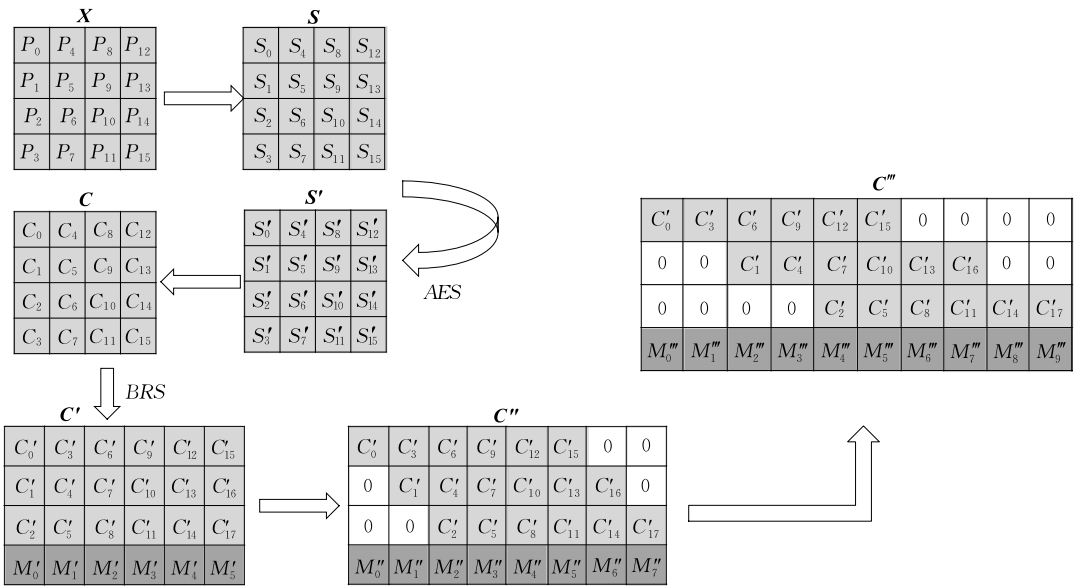


图 15 加密处理过程

安全问题的方案是一大挑战。

## 6.5 隐私保护

隐私保护问题和上一节提到的数据安全问题紧密相连。例如，在车联网中，自动驾驶需要收集每一辆车的行驶数据，但这就涉及到用户隐私问题，其中包括行驶路线、驾驶人信息等。同态加密 (Homomorphic Encryption) 可以在不解密同时对数据进行处理<sup>[88]</sup>；差分隐私 (Differential Privacy) 可以保证不泄露任意数据的隐私<sup>[89]</sup>。另外，Wang 等人<sup>[90]</sup>提出了一种数据同步更新的算法，用于确保用户数据隐私不泄露。

目前，联邦学习在符合法律法规和道德要求，基于保护数据隐私的目的，能够有效解决数据隐私和数据孤岛问题<sup>[91]</sup>。在联邦学习的实际应用场景中，由于数据的不同特征维度以及不同数据孤岛有各自的特点，学者提出将联邦学习分为三类：横向联邦学习、纵向联邦学习、联邦迁移学习<sup>[92]</sup>。基于边缘计算的联邦训练模型主要特点是：在终端上传参数时，不直接传到云服务器，而是在靠近终端设备的边缘服务器先进行边缘聚合，再上传边缘聚合后的参数至云服务器。这样可以在有效地保护隐私的同时降低通信成本，并可以减轻远程云的负担。

在 3.3 节介绍了利用联邦学习技术训练模型的计算卸载方案，但是目前相关研究工作较少，学者可以在此问题上进一步探究，设计计算卸载方案并有效地解决隐私问题。

## 6.6 异构性

移动网络环境复杂多样，在 5G 时代下，物联网发展迅速，随着各式各样的新兴应用出现，网络和设

备的异构性是一个重要问题。随着科技的发展，卸载技术方案可能会存在设备不兼容、数据类型不同等问题。目前已有的计算卸载方案都是假设相同的 MEC 服务器而进行实验、分析，但这并没有反映实际网络的异构性。当设计的卸载方案无法部署在某些新兴应用上时会使得计算卸载方案的可行性不高。所以，设计计算卸载方案时考虑实际场景下的网络 and 设备的异构性是十分关键的。如何设计卸载方案使得在不同的网络中接入不同类型的设备并保持高度兼容性是一项重大挑战。

## 7 未来研究方向

虽然计算卸载技术在近些年取得了成功，但仍有一些没有克服的问题和值得深入研究的方向。上一节，分析了计算卸载技术面临的挑战，除了在以上挑战方面进行更进一步研究外，本节提出几个有价值的未来研究方向，从而为后续研究工作指明方向。

### 7.1 新颖智能计算卸载方案

在 3.3 节，针对目前已有的新颖智能卸载方案进行对比分析，与传统基于启发式算法的计算卸载方案相比，基于在线学习的智能计算卸载方案在学习过程、模型训练中会消耗能量。人工智能逐渐从云端向边缘端迁移，智能边缘计算应运而生<sup>[93]</sup>。通过结合人工智能等相关技术来提高计算卸载的策略性能，例如利用神经网络技术，调整输入为计算模型信息、输出为相应的卸载策略。一般来说，人工智能技术中的算法越简单，在数据集较小的情况下其算法学习的效果就越好。与神经网络这类复杂模型相比，

贝叶斯、线性模型和决策树等简单模型更简单且足以处理某些应用场景中的问题. 学者在考虑智能卸载方案时, 应考虑实际问题和应用场景, 采用适当的模型用于训练. 针对没有足够的训练数据问题, 可以利用联邦学习模型方式获取多个参与方的数据进行分布式的训练, 一方面可以解决数据稀疏的问题, 另一方面还可以解决多参与方参与训练的数据隐私问题.

在 MEC 场景下计算卸载方案研究中应用人工智能技术, 将会面临新的问题和挑战: 新颖智能卸载方案在卸载过程是否能够从总体上降低能耗和时延; 加入人工智能技术后系统所增加的成本是否值得; 如何在保障卸载性能下, 降低增加相关技术实施的成本. 这一系列问题都值得学者进行探讨和研究.

目前许多有关基于在线学习智能计算卸载方案的研究围绕着联邦学习展开, 但是联邦学习自身的安全性还是有待进一步研究<sup>[94]</sup>. 类似于文献[95-97]中介绍机器学习的模型, 联邦学习在训练过程中传达模型更新的参数可能仍可以将隐私信息泄露给对手或者第三方, 这一点有待进一步研究. 目前关于联邦学习的参与者都是假设其永远连接可用的, 如果卸载过程中因为终端设备的移动导致参与训练的参与者从系统中退出, 系统性能会大大降低<sup>[98]</sup>.

7.2 智能计算卸载方案中的激励机制

在传统的计算卸载方案中, 激励机制针对云服务器和边缘服务器而设计, 为了激励云服务器运营商和边缘服务器参与计算卸载, 完成终端设备卸载的计算任务. Liu 等人<sup>[99]</sup>将传统计算卸载方案中的激励机制转化为 Stackelberg 博弈问题求解, 第一阶段由云服务器运营商设置奖励文件, 第二阶段边缘服务器根据奖励文件选择性处理计算任务.

而未来的研究方向中也应考虑智能卸载方案中针对终端设备的合理激励机制. 在智能卸载方案中涉及到终端设备参与训练的问题, 而参与方可能具有不同的计算任务, 这些任务导致不同的资源消耗, 通过设置激励机制可以吸引更多的终端设备参与模型训练. 参与方要获得更多的奖励, 而系统想尽可能控制低成本获得最佳的模型精度, 这是两个冲突矛盾的目标. 设计最优激励机制的难点是如何量化不同任务的工作量以匹配相应的奖励, 以及如何共同优化这两个冲突的目标.

7.3 计算卸载方案的可靠性

计算卸载方案的可靠性与容错性紧密相连, 方案在有故障的情况下能继续执行其预期功能的能力. 在不同应用场景下对卸载方案的要求可能不同,

车联网和智慧医疗场景下对数据速率和准确性较高, 需要实时响应, 且应用需要接近 100% 的容错率. 例如, Yang 等人<sup>[100]</sup>针对车联网场景下的任务卸载考虑了服务的可靠性、延迟和容错率.

在实际应用中, 难以保证在卸载过程中不会出现任何故障, 因此方案或系统应具备高度的可靠性以便适应时变应用场景. 计算卸载方案的可靠性是计算卸载技术领域一个重要方向, 但目前没有足够的研究论文在可靠性方面做进一步研究.

7.4 资源管理

资源管理是 MEC 中具有挑战性的研究工作, 主要是在 MEC 系统中不仅云计算节点可以进行存储和计算服务, 边缘节点和终端设备也可执行相关计算任务; 同时如何分配合适大小的资源也是一个难题. 在考虑计算卸载方案时与管理系统资源是同步进行的. 在 MEC 系统中, 终端设备将如何切分应用程序, 使得程序可以在终端设备本地、边缘层、云层同步进行; 如何管理和分配系统资源且综合考虑需求、存储和计算能力、延迟、网络带宽等因素. 在云计算中弹性资源调度是其显著特征, 同样的在 MEC 中也可以发展弹性资源调度机制来保障服务质量.

在不同的 MEC 场景中, 数据处理顺序和时间要求差异较大, 因此制定一个通用的资源管理机制或平台难度较大, 所以需要在不同场景中具体情况具体分析. 例如针对异步到达的数据序列, 可以对到达序列进行分析从而对数据进行最佳划分. 资源管理应充分利用 MEC 的优势和特点, 综合考虑终端设备的移动性、边缘服务器和云服务器的资源和配置、服务的可用性、设备能耗、网络变化等因素. 值得注意的是, MEC 资源管理系统应充分考虑在满足应用延迟的要求下, 更进一步降低延迟时间、发挥优势.

7.5 服务质量

对于用户而言, 最重要的是用户体验. MEC 可以为端到端的应用提供实时响应的服务, 保障优质的服务质量是考察计算卸载技术方案优越性的标准之一. 本文从以下三个方面探讨服务质量:

(1) 应用响应时长. 在优秀的计算卸载方案中, 用户请求服务, 系统根据不同的服务类型由不同的计算节点提供服务. 整个服务过程的延迟时间即响应时长会影响用户的体验感. 学者在设计计算卸载方案时应考虑相应场景下对延迟的要求, 如果信道的回传质量不高则更应该考虑本地执行计算任务, 否则会造成过高的传输延迟;

(2) 网络可扩展性. 随着科学技术的不断发展, 接入网络的设备越来越多, 当大量设备同时访问 MEC 服务器时, 尽管计算卸载技术可以帮助选择合适的计算节点执行服务, 仍然会产生高并发量的问题, 造成带宽瓶颈, 导致服务质量下降甚至中断. 更多的研究和论文必须关注这一问题, 以便在解决方案之间做出更好的权衡, 提高网络的可扩展性也是 MEC 发展过程中一个重要研究方向;

(3) 成本最小化. 由于部分终端设备成本较高, 服务提供商通过预算来保证服务质量. 如何降低服务提供商(云服务提供商、边缘服务提供商)的总成本同时使利润最大化是 MEC 应用发展过程中需要关注的问题.

## 7.6 定价机制

在传统云计算中, 企业用户或个人用户向云服务提供商租赁云产品和服务, 云服务商如何规定一个合理的服务定价使得用户与提供商都满意是一个关键问题<sup>[101]</sup>. 同样, 在 MEC 中也存在资源服务定价问题, 且需要保障各方参与者的利益. MEC 参与者包含云服务提供商、网络运营商、边缘云服务提供商和终端用户. 文献[102]针对 MEC 中计算和存储能力问题, 提出了一种基于 Stackelberg 博弈的计算卸载方案并设计出一种符合矿工个性化算力需求的定价策略.

MEC 中定价机制研究工作相对较少, 学者可以根据如何为不同参与者提供的服务定价、如何针对同一类参与者不同参与方分配收益、如何个性化定制定价机制等开展研究工作.

## 7.7 5G 下 MEC 计算卸载平台

在传统云计算中存在成熟的卸载平台, 2011 年, Chun 等人<sup>[103]</sup>提出在终端设备与云计算中心卸载方案 CloneCloud, 其使用静态分析和动态概要分析的组合, 以精细的粒度自动对应用程序进行分区, 将部分计算任务卸载到云. 2012 年, Kosta 等人<sup>[104]</sup>提出一款能轻松将智能手机应用程序卸载到云的框架——ThinkAir, 其利用云中智能手机虚拟化的概念, 并提供计算分流. MEC 计算卸载平台的设计可以借鉴以往优秀平台设计方案, 虽然基于云计算的卸载平台并不一定适用于 MEC, 但仍具有借鉴与参考的意义.

为了克服移动计算和 5G 应用场景下云计算的不足, MEC 应运而生. 5G 网络下, 利用边缘层服务器或基站提供的计算和存储服务可以提高终端用户的体验、节省终端设备的能耗. 例如, 在 4G LTE 网络下, 把智能手机提取图片文字的 Tesseract-OCR

代码部署到边缘云平台 Cloudlet 上, 与在 Amazon EC2-West 上执行相比, 不仅识别速度加快, 而且设备能耗降低一半以上<sup>[105]</sup>. 但是, Cloudlet 不是移动网络的固有部分, 所以将代码卸载至 Cloudlet 平台上, 终端设备的 QoS 很难像在传统云计算中心上一样得到保障.

5G 网络支持 D2D, 也就是两个对等的终端设备点对点进行直接通信而不需要基站的转发. FemtoClouds 系统就是利用这个特点, 用空闲的终端设备在网络边缘提供服务. FemtoClouds 系统中, 移动终端设备构成可以共享硬件和资源的终端集群, 终端设备可以向其他终端设备卸载计算任务<sup>[106]</sup>. 但是如何解决终端设备的集群重组构建问题有待进一步研究.

MEC 结合 5G 网络自身高带宽和低延迟的特性, 在许多场景下都可以得到应用. MEC 中计算卸载在 5G 网络下能够充分发挥其功能, 提高 5G 网络的增值服务. 后续研究工作可以致力于实现在 5G 网络下基于 MEC 的计算卸载系统平台, 使得平台可以保证终端设备的 QoS.

从文章调研的方案中, 不难发现, 所有的方案都通过仿真实验来表明其效果和性能. 学者们在计算卸载方案仿真实验表明有效性后, 可以积极寻求方案技术落地实现, 创造经济价值和社会价值. 任何技术的理论方案都要经过仿真实验并通过实际场景应用验证, 也只有在实际场景应用中发现方案技术中可能存在的潜在问题才能不断解决问题, 改进方案、发展技术.

计算卸载方案研究近几年已经较为成熟, 下一阶段可以着手于进行技术落地, 实现计算卸载平台. 现有研究从计算卸载系统的各个层次进行, 但是由于软硬件等因素的限制, 难以实现技术落地. 技术落地之后, 对许多应用领域而言是巨大的成果.

## 8 总 结

随着 5G 带来的网络变化, MEC 作为一场架构革命, 备受关注. MEC 能有效地解决客户端和服务端面临的种种问题, 尽管对 MEC 的研究有社会动力和价值意义, 但 MEC 本身仍是不成熟且未经高度验证的技术. 计算卸载是 MEC 中关键技术之一, 解决了移动设备资源不足、能耗等方面问题. 计算卸载方案起着至关重要的作用, 因为它决定了任务在哪一层进行, 是本地终端设备还是在边缘端, 亦或是两层都进行计算.



同时,在调研过程中,也发现目前大多研究建立在移动设备是固定不动这一比较理想化的情况下,若移动设备在计算卸载途中开始移动,调研的算法是否仍有效率需进一步研究和实验验证。在不同应用场景下,对延迟和能耗的要求不一样,并不是一个算法适用于所有场景,所以在讨论算法的性能和可移植性上需根据实际场景分别做讨论。

本文提出将计算卸载方案按照方案技术划分为两类:基于启发式的传统卸载方案和基于在线学习智能计算卸载方案。随着人工智能相关技术越来越成熟,其应用场景也越广泛。将人工智能相关技术运用到计算卸载方案的研究中,可以使得卸载方案更加智能、复用性更强,基于在线学习的智能计算卸载方案是计算卸载方案研究中的大势所趋。本文亦提出了目前计算卸载方案所存在的问题以及未来研究方向,从而为相关领域的研究人员提供参考和帮助。

## 参 考 文 献

- [1] Marz N, Warren J. Big Data: Principles and Best Practices of Scalable Realtime Data Systems. Greenwich, USA: Manning Publications Co., 2015
- [2] Zhao J, Liu Y, Gong Y, et al. A dual-link soft handover scheme for C/U plane split network in high-speed railway. IEEE Access, 2018, 6: 12473-12482
- [3] Chatzopoulos D, Fernandez B C, Kosta S, et al. Offloading computations to mobile devices and cloudlets via an upgraded NFC communication protocol. IEEE Transactions on Mobile Computing, 2019, 19(3): 640-653
- [4] Shi Wei-Song, Zhang Xing-Zhou, Wang Yi-Fan, et al. Edge computing: State-of-the-art and future directions. Journal of Computer Research and Development, 2019, 56(1): 69-89 (in Chinese)  
(施巍松, 张星洲, 王一帆等. 边缘计算: 现状与展望. 计算机研究与发展, 2019, 56(1): 69-89)
- [5] Zhou Yue-Zhi, Zhang Di. Near-end cloud computing: Opportunities and challenges in the post-cloud computing era. Chinese Journal of Computers, 2019, 42(4): 677-700 (in Chinese)  
(周悦芝, 张迪. 近端云计算: 后云计算时代的机遇与挑战. 计算机学报, 2019, 42(4): 677-700)
- [6] Cao Zhi-Han, Lu Yu-Cheng, Lai Si-Si, et al. Survey on sensor-cloud based on edge computing. Journal of Software, 2019, 30(Suppl. (11)): 40-50(in Chinese)  
(曹芷晗, 卢煜成, 赖思思等. 基于边缘计算的传感云研究进展. 软件学报, 2019, 30(Suppl. (11)): 40-50)
- [7] Liu Zhou-Zhou, Li Shi-Ning, Li Bin, et al. New elastic collision optimization algorithm and its application in sensor cloud resource scheduling. Journal of Zhejiang University (Engineering Science), 2018, 52(8): 6-18(in Chinese)  
(刘洲洲, 李士宁, 李彬等. 基于弹性碰撞优化算法的传感云资源调度. 浙江大学学报(工学版), 2018, 52(8): 6-18)
- [8] Ning Zhen-Yu, Zhang Feng-Wei, Shi Wei-Song. A study of using TEE on edge computing. Journal of Computer Research and Development, 2019, 56(7): 1441-1453(in Chinese)  
(宁振宇, 张锋巍, 施巍松. 基于边缘计算的可信执行环境研究. 计算机研究与发展, 2019, 56(7): 1441-1453)
- [9] Wang T, Luo H, Zheng X, et al. Crowdsourcing mechanism for trust evaluation in CPCS based on intelligent mobile edge computing. ACM Transactions on Intelligent Systems and Technology, 2019, 10(6): 1-19
- [10] Mao Y, You C, Zhang J, et al. A survey on mobile edge computing: The communication perspective. IEEE Communications Surveys & Tutorials, 2017, 19(4): 2322-2358
- [11] Mach P, Becvar Z. Mobile edge computing: A survey on architecture and computation offloading. IEEE Communications Surveys & Tutorials, 2017, 19(3): 1628-1656
- [12] Georgiev P, Lane N D, Rachuri K K, et al. LEO: Scheduling sensor inference algorithms across heterogeneous mobile processors and network resources//Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking. New York, USA, 2016: 320-333
- [13] Cuervo E, Balasubramanian A, Cho D, et al. MAUI: Making smartphones last longer with code offload//Proceedings of the 8th Annual International Conference on Mobile Systems, Applications, and Services. California, USA, 2010: 49-62
- [14] Xu D, Li T, Li Y, et al. Edge intelligence: Architectures, challenges, and applications. arXiv e-prints. arXiv:2003.12172, 2020
- [15] Lane N D, Miluzzo E, Lu H, et al. A survey of mobile phone sensing. IEEE Communications Magazine, 2010, 48(9): 140-150
- [16] Jiang C, Cheng X, Gao H, et al. Toward computation offloading in edge computing: A survey. IEEE Access, 2019, 7: 131543-131558
- [17] Zhang Kai-Yuan, Gui Xiao-Lin, Ren De-Wang, et al. Survey on computation offloading and content caching in mobile edge networks. Journal of Software, 2019, 30(8): 2491-2516(in Chinese)  
(张开元, 桂小林, 任德旺等. 移动边缘网络中计算迁移与内容缓存研究综述. 软件学报, 2019, 30(8): 2491-2516)
- [18] Shakarami A, Ghobaei-Arani M, Shahidinejad A. A survey on the computation offloading approaches in mobile edge computing: a machine learning-based perspective. Computer Networks, 2020, 182: 107496
- [19] ETSI. Mobile-edge computing introductory technical white paper. White Paper, Mobile-Edge Computing Industry Initiative, 2014: 1089-7801
- [20] Vaquero L M, Rodero-Merino L. Finding your way in the fog: Towards a comprehensive definition of fog computing. ACM SIGCOMM Computer Communication Review, 2014, 44(5): 27-32
- [21] Hong K, Lillethun D, Ramachandran U, et al. Mobile fog: A programming model for large-scale applications on the

- Internet of Things//Proceedings of the 2nd ACM SIGCOMM Workshop on Mobile Cloud Computing. New York, USA, 2013; 15-20
- [22] Wang T, Liang Y, Tian Y, et al. Solving coupling security problem for sustainable sensor-cloud systems based on fog computing. *IEEE Transactions on Sustainable Computing*, 2021, 6(1): 43-53
- [23] Li W, Chen Z, Gao X, et al. Multimodel framework for indoor localization under mobile edge computing environment. *IEEE Internet of Things Journal*, 2018, 6(3): 4844-4853
- [24] Ahmed A, Ahmed E. A survey on mobile edge computing//Proceedings of the 10th International Conference on Intelligent Systems and Control. Coimbatore, India, 2016: 1-8
- [25] Gong W, Qi L, Xu Y. Privacy-aware multidimensional mobile service quality prediction and recommendation in distributed fog environment. *Wireless Communications and Mobile Computing*, 2018; 3075849
- [26] Wang T, Lu Y, Cao Z, et al. When sensor-cloud meets mobile edge computing. *Sensors*, 2019, 19(23): 5324
- [27] Zhang Y, Zhou Y. Transparent computing, a new paradigm for pervasive computing//Proceedings of the International Conference on Ubiquitous Intelligence and Computing. Wuhan, China, 2006: 1-11
- [28] Chiang M, Zhang T. Fog and IoT: An overview of research opportunities. *IEEE Internet Things*, 2016, 3(6): 854-864
- [29] Yi S, Li C, Li Q. A survey of fog computing: Concepts, applications and issues//Proceedings of the 2015 Workshop on Mobile Big Data. Xi'an, China, 2015: 37-42
- [30] Bonomi F. Connected vehicles, the Internet of Things, and fog computing//Proceedings of the 8th ACM International Workshop on Vehicular Inter-Networking. New York, USA, 2011: 13-15
- [31] Bonomi F, Milito R, Zhu J, et al. Fog computing and its role in the Internet of Things//Proceedings of the 1st Edition of the MCC Workshop on Mobile Cloud Computing. New York, USA, 2012: 13-16
- [32] Wang Y. Cloud-dew architecture. *International Journal of Cloud Computing*, 2015, 4(3): 199-210
- [33] ETSI. ETSI GS MEC 003: Mobile Edge Computing (MEC). Framework and Reference Architecture V1.1.1. Nice: European Telecommunication Standard, 2016
- [34] Table T, Samdanis K, Mada B, et al. On multi-access edge computing: A survey of the emerging 5G network edge architecture & orchestration. *IEEE Communications Surveys & Tutorials*, 2017, 19(3): 1657-1681
- [35] Wang T, Zhang G, Liu A, et al. A secure IoT service architecture with an efficient balance dynamics based on cloud and edge computing. *IEEE Internet of Things Journal*, 2019, 6(3): 4831-4843
- [36] Zhang Y, Liu H, Jiao L, et al. To offload or not to offload: An efficient code partition algorithm for mobile cloud computing//Proceedings of the 2012 IEEE 1st International Conference on Cloud Networking. Paris, French, 2012: 80-86
- [37] Shan X, Zhi H, Li P, et al. A survey on computation offloading for mobile edge computing information//Proceedings of the 2018 IEEE 4th International Conference on Big Data Security on Cloud, IEEE International Conference on High Performance and Smart Computing and IEEE International Conference on Intelligent Data and Security. Omaha, USA, 2018; 248-251
- [38] Liu J, Mao Y, Zhang J, et al. Delay-optimal computation task scheduling for mobile-edge computing systems//Proceedings of the IEEE International Symposium on Information Theory. Barcelona, Spain, 2016: 1451-1455
- [39] Mao Y, Zhang J, Letaief K B. Dynamic computation offloading for mobile-edge computing with energy harvesting devices. *IEEE Journal on Selected Areas in Communications*, 2016, 34(12): 3590-3605
- [40] Ulukus S, Yener A, Erkip E, et al. Energy harvesting wireless communications: A review of recent advances. *IEEE Journal on Selected Areas in Communications*, 2015, 33(3): 360-381
- [41] You C, Huang K. Multiuser resource allocation for mobile-edge computation offloading//Proceedings of the 2016 IEEE Global Communications Conference. Washington, USA, 2016: 1-6
- [42] Zhao P, Tian H, Qin C, et al. Energy-saving offloading by jointly allocating radio and computational resources for mobile edge computing. *IEEE Access*, 2017, 5: 11255-11268
- [43] Cao X, Wang F, Li X, et al. Joint computation and communication cooperation for mobile edge computing//Proceedings of the 16th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks. Shanghai, China, 2018: 1-6
- [44] Lyu X, Tian H, Jiang L, et al. Selective offloading in mobile edge computing for the green Internet of Things. *IEEE Network*, 2018, 32(1): 54-60
- [45] Rimal B P, Van D P, Maier M. Mobile edge computing empowered Fiber-Wireless access networks in the 5G era. *IEEE Communications Magazine*, 2017, 55(2): 192-200
- [46] Guo H, Liu J. Collaborative computation offloading for multi-access edge computing over Fiber-Wireless networks. *IEEE Transactions on Vehicular Technology*, 2018, 67(5): 4514-4526
- [47] Chen X, Jiao L, Li W, et al. Efficient multi-user computation offloading for mobile-edge cloud computing. *IEEE/ACM Transactions on Networking*, 2016, 24(5): 2795-2808
- [48] Dinh T Q, Tang J, La Q D, et al. Offloading in mobile edge computing: Task allocation and computational frequency scaling. *IEEE Transaction on Communications*, 2017, 65(8): 3571-3584
- [49] Chen M, Hao Y. Task offloading for mobile edge computing in software defined ultra-dense network. *IEEE Journal on Selected Areas in Communications*, 2018, 36(3): 587-597
- [50] Ning Z, Dong P, Kong X, et al. A cooperative partial computation offloading scheme for mobile edge computing enabled Internet of Things. *IEEE Internet of Things Journal*, 2018, 6(3): 4804-4814

- [51] Xu J, Chen L, Zhou P. Joint service caching and task offloading for mobile edge computing in dense networks// *Proceedings of the IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. Honolulu, USA, 2018: 207-215
- [52] Zheng J, Gao L, Wang H, et al. Joint downlink and uplink edge computing offloading in ultra-dense HetNets. *Mobile Networks and Applications*, 2019, 24(5): 1452-1460
- [53] Alam M G R, Hassan M M, Uddin M Z I, et al. Autonomic computation offloading in mobile edge for IoT applications. *Future Generation Computer Systems*, 2019, 90: 149-157
- [54] Qiu X, Liu L, Chen W, et al. Online deep reinforcement learning for computation offloading in blockchain-empowered mobile edge computing. *IEEE Transactions on Vehicular Technology*, 2019, 68(8): 8050-8062
- [55] Wang X, Han Y, Wang C, et al. In-edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning. *IEEE Network*, 2019, 33(5): 156-165
- [56] Yu Z, Hu J, Min G, et al. Federated learning based proactive content caching in edge computing//*Proceedings of the 2018 IEEE Global Communications Conference*. Abu Dhabi, UAE, 2018: 1-6
- [57] Ren J, Wang H, Hou T, et al. Federated learning-based computation offloading optimization in edge computing-supported Internet of Things. *IEEE Access*, 2019, 7: 69194-69201
- [58] Qian Y, Hu L, Chen J, et al. Privacy-aware service placement for mobile edge computing via federated learning. *Information Sciences*, 2019, 505: 562-570
- [59] Zhang C, Zheng Z. Task migration for mobile edge computing using deep reinforcement learning. *Future Generation Computer Systems*, 2019, 96: 111-118
- [60] Wang T, Liang Y, Zhang Y, et al. An intelligent dynamic offloading from cloud to edge for smart IoT systems with big data. *IEEE Transactions on Network Science and Engineering*, 2020, 7(4): 2598-2607
- [61] Wang S, Chen M, Yin C, et al. Federated learning for task and resource allocation in wireless high altitude balloon networks. *IEEE Internet of Things Journal*, 2021. DOI:10.1109/JIOT.2021.3080078
- [62] Guo X, Singh R, Zhao T, et al. An index based task assignment policy for achieving optimal power-delay tradeoff in edge cloud systems//*Proceedings of the 2016 IEEE International Conference on Communications*. Kuala Lumpur, Malaysia, 2016: 1-7
- [63] Wang N, Varghese B, Matthaiou M, et al. ENORM: A framework for edge node resource management. *IEEE Transactions on Services Computing*, 2020, 13(6): 1086-1099
- [64] Guo J, Song Z, Cui Y, et al. Energy-efficient resource allocation for multi-user mobile edge computing//*Proceedings of the 2017 IEEE Global Communications Conference*. Singapore, 2017: 1-7
- [65] Tran T X, Pompili D. Joint task offloading and resource allocation for multi-server mobile-edge computing networks. *IEEE Transactions on Vehicular Technology*, 2018, 68(1): 856-868
- [66] Zhang J, Xia W, Yan F, et al. Joint computation offloading and resource allocation optimization in heterogeneous networks with mobile edge computing. *IEEE Access*, 2018, 6: 19324-19337
- [67] Shen Su-Bin, Lin Chuang. Topic foreword: Opportunities and challenges of Internet of things research. *Journal of Software*, 2014, 25(8): 1621-1624(in Chinese)  
(沈苏彬, 林闯. 专题前言: 物联网研究的机遇与挑战. *软件学报*, 2014, 25(8): 1621-1624)
- [68] Sun Qi-Bo, Liu Jie, Li Shan, et al. Internet of Things: Summarize on concepts, architecture and key technology problem. *Journal of Beijing University of Posts and Telecommunications*, 2010, 33(3): 1-9(in Chinese)  
(孙其博, 刘杰, 黎霁等. 物联网: 概念、架构与关键技术研究综述. *北京邮电大学学报*, 2010, 33(3): 1-9)
- [69] Zeng Jian-Dian, Wang Tian, Jia Wei-Jia, et al. A survey on sensor-cloud. *Journal of Computer Research and Development*, 2017, 54(5): 925-939(in Chinese)  
(曾建电, 王田, 贾维嘉等. 传感云研究综述. *计算机研究与发展*, 2017, 54(5): 925-939)
- [70] Sucipto K, Chatzopoulos D, Kosta S, et al. Keep your nice friends close, but your rich friends closer — Computation offloading using NFC//*Proceedings of the IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. Atlanta, USA, 2017: 1-9
- [71] Golkarifard M, Yang J, Huang Z, et al. Dandelion: A unified code offloading system for wearable computing. *IEEE Transactions on Mobile Computing*, 2018, 18(3): 546-559
- [72] Chen Shu-Yong, Song Shu-Fang, Li Lan-Xin, et al. Survey on smart grid technology. *Power System Technology*, 2009, 33(8): 1-7(in Chinese)  
(陈树勇, 宋书芳, 李兰欣等. 智能电网技术综述. *电网技术*, 2009, 33(8): 1-7)
- [73] Liu C, Cao Y, Luo Y, et al. A new deep learning-based food recognition system for dietary assessment on an edge computing service infrastructure. *IEEE Transactions on Services Computing*, 2017, 11(2): 249-261
- [74] Wang T, Cao Z, Wang S, et al. Privacy-enhanced data collection based on deep learning for Internet of vehicles. *IEEE Transactions on Industrial Informatics*, 2020, 16(10): 6663-6672
- [75] Yang R, Yu F R, Si P, et al. Integrated blockchain and edge computing systems: A survey, some research issues and challenges. *IEEE Communications Surveys & Tutorials*, 2019, 21(2): 1508-1532
- [76] Valavanis K P, Vachtsevanos G J. *Handbook of Unmanned Aerial Vehicles*. Dordrecht, Netherlands: Springer Publishing Company, 2015
- [77] Ouyang T, Li R, Chen X, et al. Adaptive user-managed service placement for mobile edge computing: An online

- learning approach//Proceedings of the IEEE INFOCOM 2019-IEEE Conference on Computer Communications, Paris, France, 2019; 1468-1476
- [78] Qahtan A, Tang N, Ouzzani M, et al. Pattern functional dependencies for data cleaning. *Proceedings of the VLDB Endowment*, 2020, 13(5): 684-697
- [79] Wang T, Ke H, Zheng X, et al. Big data cleaning based on mobile edge computing in industrial sensor-cloud. *IEEE Transactions on Industrial Informatics*, 2020, 16(2): 1321-1329
- [80] Qiu Lei, Jiang Wen-Xian, Li Yu-Ze, et al. Trustworthy data collection method based on edge computing and trust value. *Journal of Software*, 2019, 30(Suppl. (11)): 71-81(in Chinese) (邱磊, 蒋文贤, 李玉泽等. 基于边缘计算与信任值的可信数据收集方法. *软件学报*, 2019, 30(Suppl. (11)): 71-81)
- [81] Wang T, Luo H, Jia W, et al. MTES: An intelligent trust evaluation scheme in sensor-cloud enabled industrial Internet of Things. *IEEE Transactions on Industrial Informatics*, 2020, 16(3): 2054-2062
- [82] Domingo-Ferrer J, Farras O, Ribes-González J, et al. Privacy-preserving cloud computing on sensitive data: A survey of methods, products and challenges. *Computer Communications*, 2019, 140: 38-60
- [83] Hong J, Wen T, Guo Q, et al. Privacy protection and integrity verification of aggregate queries in cloud computing. *Cluster Computing*, 2019, 22(3): 5763-5773
- [84] Wang T, Zhou J, Chen X, et al. A three-layer privacy preserving cloud storage scheme based on computational intelligence in fog computing. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2018, 2(1): 3-12
- [85] Zhang Jia-Le, Zhao Yan-Chao, Chen Bing, et al. Survey on data security and privacy-preserving for the research of edge computing. *Journal on Communications*, 2018, 39(3): 1-21 (in Chinese) (张佳乐, 赵彦超, 陈兵等. 边缘计算数据安全与隐私保护研究综述. *通信学报*, 2018, 39(3): 1-21)
- [86] Wang T, Mei Y, Jia W, et al. Edge-based differential privacy computing for sensor — Cloud systems. *Journal of Parallel and Distributed Computing*, 2020, 136: 75-85
- [87] Zhao P, Huang H, Zhao X, et al. P<sup>3</sup>: Privacy-preserving scheme against poisoning attacks in mobile-edge computing. *IEEE Transactions on Computational Social Systems*, 2020, 7(3): 818-826
- [88] Lu R, Liang X, Li X, et al. EPPA: An efficient and privacy-preserving aggregation scheme for secure smart grid communications. *IEEE Transactions on Parallel and Distributed Systems*, 2012, 23(9): 1621-1631
- [89] Dwork C. Differential privacy: A survey of results// *Proceedings of the International Conference on Theory and Applications of Models of Computations*. Xi'an, China, 2008; 1-19
- [90] Wang T, Zhou J, Liu A, et al. Fog-based computing and storage offloading for data synchronization in IoT. *IEEE Internet of Things Journal*, 2019, 6(3): 4272-4282
- [91] Yang Qiang. AI and data privacy protection: The way to federated learning. *Journal of Information Security Research*, 2019, 5(11): 961-965(in Chinese) (杨强. AI与数据隐私保护: 联邦学习的破解之道. *信息安全研究*, 2019, 5(11): 961-965)
- [92] Pan S J, Yang Q. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 2009, 22(10): 1345-1359
- [93] Wang Zhe. The development and prospect of intelligent edge computing. *Artificial Intelligence*, 2019, (5): 18-25(in Chinese) (王哲. 智能边缘计算的发展现状和前景展望. *人工智能*, 2019, (5): 18-25)
- [94] Lim W Y B, Luong N C, Hoang D T, et al. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 2020, 22(3): 2031-2063
- [95] Ateniese G, Felici G, Mancini L V, et al. Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. *International Journal of Security*, 2015, 10(3): 137-150
- [96] Fredrikson M, Jha S, Ristenpart T. Model inversion attacks that exploit confidence information and basic countermeasures // *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. Denver, USA, 2015; 1322-1333
- [97] Tramèr F, Zhang F, Juels A, et al. Stealing machine learning models via prediction APIs// *Proceedings of the 25th USENIX Security Symposium (USENIX Security 16)*. Austin, USA, 2016; 601-618
- [98] Nishio T, Yonetani R. Client selection for federated learning with heterogeneous resources in mobile edge// *Proceedings of the IEEE International Conference on Communications*. Shanghai, China, 2019; 1-7
- [99] Liu Y, Xu C, Zhan Y, et al. Incentive mechanism for computation offloading using edge computing: A Stackelberg game approach. *Computer Networks*, 2017, 129: 399-409
- [100] Yang T, Hu Y, Gursoy M C, et al. Deep reinforcement learning based resource allocation in low latency edge computing networks// *Proceedings of the 15th International Symposium on Wireless Communication Systems*. Lisbon, Portugal, 2018; 1-5
- [101] Shi Yu-Liang, Shao Ya-Li, Zhou Zhong-Min, et al. Pricing model of privacy preserving service based on Pareto optimization. *Chinese Journal of Computers*, 2016, 39(6): 1267-1280(in Chinese) (史玉良, 邵雅丽, 周中民等. 基于帕累托最优的隐私保护服务定价模型. *计算机学报*, 2016, 39(6): 1267-1280)
- [102] Wu Yu-Xin, Cai Ting, Zhang Da-Bin. Computing power trading and pricing in mobile edge computing based on Stackelberg game. *Journal of Computer Applications*, 2020, 40(9): 2683-2690(in Chinese) (吴雨芯, 蔡婷, 张大斌. 移动边缘计算中基于 Stackelberg 博弈的算力交易与定价. *计算机应用*, 2020, 40(9): 2683-2690)

[103]

Chun B G, Ihm S, Maniatis P, et al. CloneCloud: Elastic execution between mobile device and cloud//Proceedings of the 6th Conference on Computer Systems. Salzburg, Austria, 2011; 301-314

[104]

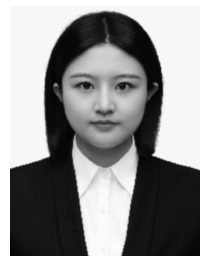
Kosta S, Aucinas A, Hui P, et al. ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading//Proceedings of the International Conference on Computer Communications. Orlando, USA, 2012; 945-953

[105]

Hu W, Gao Y, Ha K, et al. Quantifying the impact of edge computing on mobile applications//Proceedings of the 7th ACM SIGOPS Asia-Pacific Workshop on Systems. Hong Kong, China, 2016; 1-8

[106]

Habak K, Ammar M, Harras K A, et al. Femto clouds: Leveraging mobile devices to provide cloud service at the edge//Proceedings of the 2015 IEEE 8th International Conference on Cloud Computing. New York, USA, 2015; 9-16



**ZHANG Yi-Lin**, M. S. candidate. Her research interest is mobile edge computing.

**LIANG Yu-Zhu**, Ph. D. candidate. His research interest is Internet of Things.

**YIN Mu-Jun**, M. S. candidate. Her research interest is mobile edge computing.

**QUAN Han-Yu**, Ph. D. , lecturer. His research interest is network security.

**WANG Tian**, Ph. D. , professor. His research interests include edge computing and Internet of Things.

**JIA Wei-Jia**, Ph. D. , professor. His research interests include smart city and distributed systems.

Background

The traditional cloud computing which has powerful computing power and computing resources to process data centrally. Cloud computing is a centralized processing computing paradigm, which has attracted widespread attention and has been widely used. However, traditional cloud computing is no longer suitable for some scenarios with the rapid development of end devices, applications, and networks, such as IoT, IoV, and some new network applications. Therefore, MEC as a kind of new computing paradigm has been proposed in 2014. MEC is working on the edge of the network, which expands the cloud computing. With the development of the intelligence end devices, various intelligence end devices are raised with different sizes, shapes, functions. Those end devices' computing ability and storage size are limited by their size which is difficult to meet the requirement of some application scenarios and technology, such as sensors. The computation offloading is a key technology in MEC which can solve the problems of the end devices' weak computing ability and the delay of cloud computing. Summarizing the methods of computation offloading makes it easier to understand the research progress in international academic circles in the field, and to understand the application scenarios and challenges of computation offloading to propose better methods in depth.

This paper systematically summaries the existing representative solutions methods of computation offloading in

recent years and try to analyze the future direction in this field. Firstly, the background and concept of MEC are introduced in detail which is comprised of cloud computing. Second, this paper summaries the methods of computation offloading and divide them into two categories, the traditional computation offloading methods and novel intelligent computation offloading methods respectively. Third, the resource management and allocation in MEC is an important problem after a method of computation offloading is determined. Then the application scenarios and challenges are discussed in detail, which include application in IoT, IoV, blockchain, unmanned aerial vehicle, VR/AR, social networks. Finally, the future and development of computation offloading methods is prospected.

This work is supported by the Key Project of Natural Science Foundation of China under Grant Nos. 61532013 and 62172046, the Outstanding Youth Project of Fujian Province under Grant No.2020J06023, the UIC Start-Up Research Fund under Grant No. R72021202, the Natural Science Foundation of Fujian Province under Grant No. 2020J05059, the Research Fund of Huaqiao University under Grant No. 605-50Y19028, the special project of Guangdong Provincial Department of Education in key fields of colleges and universities under Grant No. 2021ZDZX1063 and the Jiont Project of Production, Teaching and Research of Zhuhai under Grant No. ZH22017001210133PWC.