**移动边缘计算中计算卸载技术研究综述**

**Abstract** Recently, with the popularization of mobile smart devices and the development of wireless communication technologies such as 5G, edge computing is proposed as a novel and promising computing mode, which is regarded as an extension of traditional cloud computing. The basic idea of edge computing is to transferm the computing tasks generated on mobile devices from offloading to remote clouds to offloading to the edge of the network, to meet the low-latency requirements of computing intensive applications such as real online game and augmented reality. The offloading problem of computing tasks in edge computing is an important issue that studies whether computing tasks should be performed locally or offloaded to edge nodes or remote clouds, since it has a big impact on task completion delay and energy consumption of devices. Firstly, this paper expounds the basic concept of edge computing and introduces the architecture of edge computing. Then, the related research work of computing offload in edge computing is expounded. Then, the practical application scenarios of computing offload technology are introduced in detail. Finally, the challenges faced by offload technology and the future research direction are pointed out.

**Keywords** mobile edge computing, computation offloading, resource allocation, intelligence computation offloading

# 1引言

随着物联网的发展和5G网络的日益普及，各种终端智能设备呈现爆炸式的增长，据IMT（International Mobile Telecommunications）2020（5G）推进组对终端设备连接数增长趋势的预测，到2030年全球物联网设备连接数量将达到1000亿左右，这些分布广泛且数量众多的设备产生的数据也越来越多，数据总量每年都呈指数形式的快速增长。尤其是随着人工智能、物联网技术和5G技术的发展，产生了各种新型应用，很多部署在终端设备上的计算密集型和时延敏感型的应用，对网络的时延和带宽提出了更高的要求，比如自动驾驶、增强现实、虚拟现实和在线游戏等新型应用，要求更低的时延、更快的速度、更高的带宽和更大的存储能力。虽然现有技术已经发展得很快，但终端设备的计算和存储能力还是无法处理和存储如此庞大的数据，难以满足这些应用的需求。另外，由于电池技术的限制，移动设备的能量有限，执行高能耗任务也是一个很大的挑战。为了解决这些问题，移动云计算（Mobile Cloud Computing，MCC）应运而生。移动云计算集合了云计算和移动互联网的优势特点[1]。所有复杂的计算都可以在云中执行，因此移动设备得以简化，不需要很复杂的配置。在MCC中，终端设备可以将低时延高能耗的应用任务，卸载到云服务器中进行计算处理，以此来增强终端设备的计算和存储能力，减少设备的能耗，缩短任务执行时间，给用户带来很好的体验。然而，虽然将设备任务卸载到云服务器，设备的性能得到了很大的提升，但是MCC模式也带来了新的问题，由于云服务器离终端设备比较远，导致任务执行时延可能很高，而且对于像自动驾驶这类对时延要求精确到毫秒级的应用来说，是无法满足其需求的。因此，MCC难以满足时延敏感型应用的需求。并且MCC采用的是集中式处理模式，大量终端设备产生的海量数据都传送到云服务器进行处理，不仅会给传输网路带来很大的负担，而且也会造成数据隐私的泄露，产生数据安全性问题。基于此业内提出了移动边缘计算(Mobile Edge Computation, MEC)。

移动边缘计算能有效解决时延长、能耗高和数据不安全等问题。尤其是计算卸载技术作为MEC的关键技术更是成为该领域的研究热点。MEC计算卸载技术允许终端设备将计算密集型任务卸载至MEC服务器执行，借助高计算性能MEC服务器实现任务执行时延降低的目的[2]。同时，将终端设备的任务卸载到边缘服务器，也可以有效降低设备能耗。因此，MEC计算卸载技术能有效解决云计算带来的问题的同时也能有效缓解终端设备资源受限与对需求有高要求的任务间的突出矛盾。虽然目前已有大量MEC相关领域的研究文献和成果，但很少有对MEC领域相关理论进行系统的梳理，并且对于MEC计算卸载技术的现状和面临的问题挑战介绍得也不够详细。因此本文对MEC相关理论、卸载技术和应用场景进行了系统的梳理，并对计算卸载技术研究现状和面临的挑战进行了详细的分析和总结。

# 2 移动边缘计算

在本节将详细介绍MEC的基本概念和架构。首先介绍了欧洲电信标准化协会(European Telecommunications Standards Institute, ETSI)对MEC的定义，再对比了MEC和云计算的区别，最后介绍MEC的三层基本架构。

## 2.1 MEC概念

边缘计算是在网络边缘执行计算的一种新型计算模型[3]，在无线接入网络的边缘侧，就近为终端设备提供边缘服务。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服务器对海量数据及隐私数据的处理[4]。而移动边缘计算是由欧洲电信标准化协会（European Telecommunications Standards Institute，ETSI）于2014年提出，是指将云服务器的部分云计算能力下沉到用户侧，在网络边缘部署计算和存储资源，为用户提供边缘服务，由于更加地靠近边缘设备，从而能为用户提供超低时延和高带宽，给用户更佳的网络体验。

与云计算相比，移动边缘计算具有很大的优势。第一，在无线接入网侧部署MEC服务器，终端设备可以将计算任务卸载至边缘服务器，不必经过核心网将任务卸载到云服务器，从而达到减少时延的目的。第二，设备产生的大量数据卸载到边缘服务器进行处理，极大地缓解了核心网络和云服务器的压力，也有效地保护了用户数据的隐私。第三，终端设备的能耗主要包括执行任务的能耗和传输任务的能耗。由于电池技术和终端设备尺寸的限制，导致设备能量很低，不能执行高能耗的任务。将设备上的任务卸载到边缘服务器可以有效地降低设备的能耗，延长设备使用的周期。表1列出了MEC与云计算系统的对比分析。

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 | 计算量高但对延迟要求不高的应用 |

## 2.2 MEC架构

MEC架构由三层基本架构组成，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。

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[5].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。

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。

# 3计算卸载的研究

在本小节中，详细介绍了计算卸载相关内容。首先，详细阐述计算卸载的基本概念，基于流程图详述计算卸载的过程，然后分别从性能指标、能耗指标和花费指标介绍计算卸载的优化目标及约束。最后，将计算卸载方案按照分成基于启发式的传统算法和基于在线学习智能卸载方案，详细介绍实现相关方案所用的技术和解决思路。

## 3.1 计算卸载的基本概念

计算卸载技术[6]作为MEC的关键技术之一，是指通过合理的卸载决策和资源分配策略将终端设备上运行的任务卸载到边缘服务器，利用服务器充足的计算和存储资源完成任务的执行，减少任务完成时延和设备的能耗，提高设备性能。计算卸载技术在云计算中已经被 运用，与移动边缘计算中的计算卸载技术上唯一的不同就是卸载的目的地不同。文献[7]将计算卸载流程大致分为六个阶段，包括寻找可用的 MEC 计算节点、程序分割、卸载决策、程序传输、执行计算和计算结果返回， 如图1所示。

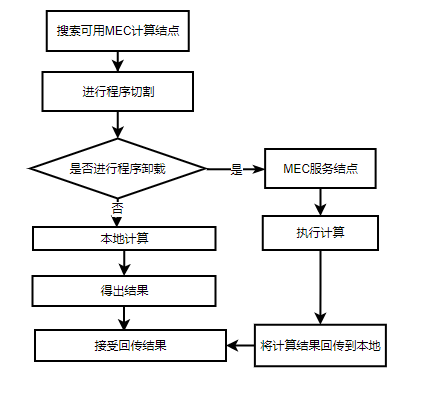


图1 卸载流程

计算卸载技术主要包含卸载决策和资源分配两个问题。其中卸载决策主要解决的是设备上的计算任务是否需要卸载，按照何种策略进行卸载，是完全卸载还是部分卸载等问题。资源分配所要解决的是如何对计算、网络和带宽等资源合理分配的问题，比如在单服务器的场景下，多个任务卸载到服务器上，如何有效合理地分配计算和存储资源，才能最大程度地降低总任务的时延和能耗。

在卸载系统中的设备一般由代码解析器、系统解析器和决策引擎三部分组成。其中代码解析器用于确定哪部分任务可以卸载，系统解析器用于监控卸载数据大小、执行任务的能耗和可用的带宽等各种参数，卸载引擎用于决定任务是否可以卸载。

## 3.2 计算卸载的优化目标及约束

将用户、运营方等方面的需求建模为优化目标与约束，进而寻求问题的解决方案非常重要。文献[8]整理了280项雾计算下的优化指标，包括时延、能耗、收益与花费、设备相关和其他5大类。本节从性能、能耗、花费3个方面介绍边缘计算下优化目标和约束的建模与典型优化方法。

## 3.2.1 性能指标

**时延** 服务时延或反应时间，是指应用提交请求和收到回应间的耗时。时延是边缘计算资源调度优化的重要指标。文献[8]将时延分为计算节点上耗时、节点间传输耗时及计算节点间迁移时间共6类。针对边缘计算环境，文献[9]给出了时延解析模型，任务的服务时延由不同计算节点的处理时延按分配概率加权计算得到。文献[10]考虑架构中的网关数量以及缓冲区占用率等情况，定义了多个时延计算方程。除理论模型外，文献[11]对雾计算时延进行了实验统计分析，统计了不同应用、http 需求等在不同雾计算设备、不同环境下的时延，发现 II 型广义极值分布能较好地模拟时延，指出小型计算任务时延取决于通讯时间且不同环境对时延的影响最高可达11倍。此外，针对时延的稳定性实验表明，时延敏感型任务应将根据环境变化而调整分配方案。目前，以最小化时延为目标的算法很多，分布式优化策略、计算任务卸载、任务调度等方法或模型都旨在降低时延进而提升系统性能。

**截止时间** 除最小化时延外，任务的截止时间可表示任务的紧迫程度。实时系统给出硬期限和软期限的定义。 不同任务的时延敏感度不同，一些任务未能在期限前完成则会出现严重后果，于是被定义为硬期限约束任务，否则为软期限约束任务。若完成时间大于期限，则存在拖期，即完成时间和期限之差。此外，时间相关效率函数也被用来描述任务完成时间的要求，譬如用阶跃函数刻画硬约束，用衰减指数函数作为延时函数，任务完成时间越晚则其完成效用越低。文献[12]提出在线调度方法，最大化满足截止时间的任务个数。文献[13]将用户体验转化为任务截止 时间并作为问题的约束，优化获得满足条件的可行解。文献[14]在定义适配值时对超过截止时间的应用添加惩罚项。文献[15]将工作流中的子任务按期限类型分为硬期限和软期限约束任务，分别考虑其合法性和拖期大小。文献[11]用概率分布函数模拟任务时延，对任务完成的时间函数与设定时间相关效率函数的积分作为任务完成效率，并最大化任务分配问题的任务完成效率。目前，根据任务特征及边缘计算场景需求选择合适的指标，设计特定优化算法，是截止时间相关优化问题的重点。

**用户体验质量** 相比于客观的时延等性能，QoE的定义依赖于用户、环境和服务，较为主观。在国际电信联盟给出的定义基础上，文献[16]将QoE定义为用户在一定的客观环境中对所使用的服务或者业务对整体的认可程度。常见的QoE优化问题要求提高用户对音频质量、视频清晰度的满意程度等。随着移动端视频需求的增大，对QoE的量化、建模及优化的研究日益增多。边缘计算作为降低即时应用时延的解决方案之一，可提高用户满意度，通过部署雾节点提供雾计算服务，因而考虑QoE的边缘资源分配和任务调度策略可为用户提供更优质的服务。为了优化用户玩交互游戏的体验，文献[17]综合反应时间、网络拥塞和服务覆盖度作为用户的QoE，并设计Cloud-Fog系统，从而降低时延，提升终端用户的覆盖度。文献[18]利用QoE指标估计和指导下 一步的资源分配。另外，文献[19]将 QoE 设为边缘计算资源配置算法的唯一优化目标，根据用户对应用的期待和边缘计算资源状态，匹配“特定应用-计算节点”，从而优化整体满意度。目前，如何与具体问题结合来量化QoE并设计优化算法，满足用户需求并提升满意度，是重点的研究工作。

## 3.2.2 能耗指标

能耗是数据中心的主要开销之一，包括计算机器、制冷散热设备耗电等。将计算设备移至网络边缘，保证资源有限的电池供电设备正常运行非常重要。文献[20]探讨了边缘计算是否能节省云计算能耗，讨论 了下载、更新、预加载数据量等应用以及不同场景下用电量的变化。研究不同情况下的资源分配和任务 调度对降低能耗很有意义。

**电池电量** 边缘计算环境下，移动终端设备的电池电量约束问题亟待研究，其中耗电量分为监测、计算、通讯 和执行4部分。监测耗电量与其数据包大小、时长相关；通讯分为上传和接收2部分，上传能耗与数据量及传输距离正相关，接收能耗与数据量正相关；计算能耗取决于具体硬件参数；执行能耗与具体执行应用和次数正相关。文献[21]在应用截止时间约束下，分配和排序任务最小化物联网设备总能耗。此外，文献[22]将系统的生存周期定义为所有传感器中最先耗尽电量的传感器的工作时长，进而延长系统生存周期成为一个优化目标。目前，如何有效的节约移动端电量，并维持系统稳定，是研究工作的重点。

**其他能耗** 文献[20]给出了路由器等多用户访问的网络设备和家用服务器等计算设备的能耗模型，其中网络设备主要依据数据流量计算能耗，服务器能耗则基于时间，同时指出微型服务器接入特定类型网络、活跃时间呈一定比例、下载活动占较大比例的应用等情形下边缘计算可节省云计算中心的能耗。文献[23]将边缘计算节点能耗建模为关于计算量的单增严格凸函数，并用二次函数刻画其特性，用服务器开关状态以及处理器CPU频率的一次函数刻画云计算节点能耗。文献[24]将能耗分为空闲时和占用时能耗2类，可以通过对时间积分计算。目前，针对边缘计算和云计算节点的能耗优化，包括总能耗的优化、性能和能耗的多目标均衡优化。

## 3.2.3 花费指标

经济指标是用户选择资源以及运营商提供解决方案时需要考虑的重要因素。云计算的广泛应用一定程度上取决于其成功的“pay-as-you-go”商业模式。可见，服务的合理定价和对商业成本的优化是边缘计算亟待解决的重要问题。资源调度优化的花费主要集中于计算、存储和网络等资源成本。文献[25]在考虑性能、虚拟机等约束下，将虚拟机的配置成本和数据通讯的花费之和作为优化目标， 获得最优的用户基站选择、虚拟机-基站的配对等解决方案。

文献[26]考虑城域网中的经济成本问题，将应用分为只需计算、只需存储以及存储与计算等类别，以计算节点计算、存储、链路容量和时延为约束，最小化支持网络拓扑结构下的经济成本，包括计算、存储、城域网带宽和核心计算节点上下游各项的成本。同时，参考谷歌云等的价格模型给出了不同计算节点的计算、存储的价格，价格差异将导致不同应用的资源选择方案不同。目前，如何将定价模型和实际应用结合，根据边缘计算特点定义花费和建立优化模型，是研究工作的重点。

## 3.3 计算卸载方案研究

在本节介绍计算卸载技术相关研究，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御载方案。本文提出这种分类依据，可以让读者更清晰地了解计算御载方案异同点、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。

## 3.3.1 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研究工作调研，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。

**最小化延迟时间** Liu等人[27]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。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路。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。 Mao等人[28]提出了基于具有绿色能源收集的MEC系统的计算卸载策略。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。

**最小化能耗** You等人[29]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御载。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[30]在多移动终端设备的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(Reformulation-Linearization-Technique-basedBranch-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。与上述方案不同，Lyu[31]等人提出了一个新的云、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。中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。然后设计了一个选择性卸载方案米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，优点是可以使得设备根据自身实际情况选择是否卸载。

**权衡延迟时间和能耗** 与传统的卸载方案不同，Chen等人[32]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[32]不同，Dinh等人[33]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。

## 3.3.2 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, IRL)、联邦学为(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。其优化算法的主要日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[34]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御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[35]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性能。

Zhang等人[36]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任务迁移策略，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。

基于在线学习智能计算卸载方案，其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可以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。

# 4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。

## 4.1 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。

## 4.2 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问题。 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。

## 4.3 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。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。

## 4.4 智慧医疗

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，降低延识从而快速响应时间。

## 4.5 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。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。

## 4.6 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中计算御载技术为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。

## 4.7 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。

# 5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，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。

1. 不同设备任务相关性方面

现有的关于MEC的研究都仅考虑了单个设备上任务的卸载情况。对于不同设备之间任务具有依赖性的研究非常少。实际上，在不同设备上执行的任务通常也是具有相关依赖性的，一个设备上子任务的输入，依赖于另一个设备上的子任务的输出。像在智能家居环境中，温度感应器通过感应室内温度获得温度值，根据这个温度值实现对其他智能设备的控制，比如可以根据这个温度值调节空调的温度。基于不同设备之间任务也是具有依赖关系的，文献[37]考虑了MEC系统环境下两个不同设备之间的任务依赖性，一个设备上的任务输入需要另一个设备上的最终任务输出，将任务的依赖关系建模为顺序图，基于椭球法和单爬策咯获得最佳卸载策略，使设备总的时延和能耗加权和最小。文献[38]在文献[37]的基础上，扩展到多个设备，提出了一种降低复杂度的Gbbs采样算法来获得最优的卸载决策。但这两个文献都是将任务的依赖关系建模为顺序图，不够一般化，可以考虑更细粒度的任务依赖关系，将任务之间的依赖关系建模为有向无环图。考虑不同设备上任务的依赖关系问题的复杂度要高于仅考虑单设备任务依赖关系。其中一个难点，需要考虑两个设备上具有依赖关系任务的位置，位置不同对系统总体性能的影响是不同的。针对此难点，需要一一考虑不同设备具有依赖关系任务的每一个卸载位置关系。另一个难点，如果两个设备的任务依赖关系都分别建模为有向无环图，那么将会增加问题的复杂度。针对此问题可以在任务执行卸载前考虑运用剪枝或聚类技术缩小图的规模，来降低复杂度。

1. 移动性方面

当移动设备在部署有MEC服务器的基站覆盖范围内移动时，不需要服务间的切换，但当移动设备从一个MEC服务器覆盖范围内移动到另一个MEC服务器覆盖范围内时，会出现服务的不连续性，影响用户的体验。如何在不同EC服务器覆盖范围之间进行服务相关业务的快速切换，保证EC服务的连续性，使服务质量保持稳定，是MEC计算卸载技术面临的重要挑战。日前针对此问题大多都是通过任务迁移技术进行解决，因为设备在不同服务范围内的移动，涉及到设备相关状态信息和服务相关业务的切换，需要将状态信息和服务业务进行快速迁移，以此来保证服务的连续性。如果能够对设备移动的方向进行提前预测，做到预先任务迁移，快速在服务器之间进行切换，让用户察觉不到服务的中断，这将更能提升用户的体验，但如何做到精准预测，又是需要解决的一个难点。可以通过与机器学习、深度学习相结合来解决设备行为预测问题。文献[39]考虑了MEC与深度学习相结合，基于LSTM预测模型对用户移动性进行预测。移动性管理是移动边缘计算的重要挑战，需要该领域的研究学者深入研究寻找相应的解决策略。

1. 安全性方面

由于MEC服务器在移动边缘网络环境中是分布式部署的，导致MEC服务器的安全防御是很薄弱的，很容易被攻击。并且随着联网设备的增多，大量的数据被卸载到边缘服务器，有些数据是非常重要的，如何保障这些数据的安全，防止这些隐私数据被泄露窃取，是需要解决的重要问题。针对此问题可以对访问服务器的设备进行身份的验证，防止非法设备攻击服务器上的应用程序和数据，造成隐私的泄露。另一方面也需要对服务器上的应用程序进行验证，防止设备被恶意攻击。设备任务被卸载到边缘服务器，需要通过通信链路在复杂的网络环境中传输，面对如此复杂的网络环境，以往针对云计算的许多安全防范措施，可能不适用于移动边缘网络环境，因此如何在这复杂多变的边缘网络环境中保障通信链路的安全，也是MEC计算卸载技术所要面临的重要挑战。针对此问题可以考虑将以往的一些安全措施和MEC技术相结合，针对边缘网络环境来部署安全措施，以此来保证移动边缘网络环境的安全。在如今信息化时代，数据的安全是非常重要的，因此需要相关研究者进行更深入的研究，解决移动边缘网络环境中的数据安全性管理方面的问题。

1. 干扰性方面

当MEC服务器同时接收到多个不同设备上的卸载任务时，任务与任务之间将会存在网络、计算和存储资源的竞争问题，竞争导致的资源的分配不均是产生干扰的根本原因。由于不同设备卸载的任务大小是不一样的，所需的网络、计算和存储资源也是不同的，如何在这些任务之间进行资源的合理分配，通过何种有效资源分配策略使得设备之间的干扰降到最低，是需要解决的一个难点，也是计算卸载技术面临的重要挑战。目前针对此问题的一个重要解决手段是通过考虑寻找一种合理有效的资源分配策略，充分地利用网络、计算和存储资源，来减少设备之间的竞争，解决不同任务之间的干扰问题。文献[40]共同考虑了计算决策和干扰管理，通过联合优化计算卸载决策、物理资源块分配和MEC计算资源分配问题获取最佳卸载决策来提高移动边缘计算网络的性能。文献[41]考虑了非正交多址访问系统下节能资源分配问题，通过优化子信道分配和功率分配来最大化非正交多址访问系统的能效性能。

# 6总结

结束语近年来，随着日益增长的低延迟、计算密集型应用的涌现，越来越多的研究人员加入到对边缘计算的研究中，边缘计算将掀起网络时代的新浪潮。计算卸载是边缘计算中的核心技术，其不单指表面上的“卸载”，还囊括了计算任务调度、计算和网络资源分配、服务缓存与管理、云边端协同等多方面的技术研究。

本文阐述了边缘计算的概念与结构，介绍了边缘计算系统中计算卸载的优化目标及约束，重点梳理了边缘计算系统中计算卸载问题的研究现状，罗列了目前计算卸载技术的应用场景，并指出了目前计算卸载技术面临的挑战。希望本文能为相关领域的专家学者的研究工作提供一定的参考与帮助。

参考文献

[1] LINTHICUM D S. Connecting fog and cloud computing[J]. IEEE Cloud Computing，2017，4（2）：18-20.

[2] SHI W S，CAO J，ZHANG Q，et al. Edge computing：vision and challenges[J].IEEE Internet of Things Journal，2016， 3（5）：637-646.

[3] Wang T, Luo II, 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

[4] 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

[5] ETSI.ETSI GS MEC 003:Mobile Edge Computing (MEC). Framework and Reference Architecture V1.1.1.Nice:European Telecommunication Standard,2016

[6] FLORES H，HUI P，TARKOMA S，et al. Mobile code offloading：from concept to practice and beyond

[7] 董思岐，李海龙，胡磊，等 . 移动边缘计算中的计算卸载策略研究综述[J]. 计算机科学，2019，46（11）：32-40.

[8] Brogi A, Forti S, Guerrero C, et al. How to Place Your Apps in the Fog: State of the Art and Open Challenges[J]. Software: Practice and Experience (S0167-739X), 2019, 1(1): 1-8.

[9] Yousefpour A, Ishigaki G, Gour R, et al. On Reducing IoT Service Delay Via Fog Offloading[J]. IEEE Internet of Things Journal (S2327-4662), 2018, 5(2): 998-1010.

[10] Desikan K E S, Srinivasan M, Murthy C S R. A Novel Distributed Latency-aware Data Processing in Fog Computing-enabled IoT Networks[C]//Proc. of the ACM Workshop on Distributed Information Processing in Wireless Networks. Chennai, India: ACM, 2017: 1-6.

[11] Gorlatova M, Inaltekin H, Chiang M. Characterizing Task Completion Latencies in Fog Computing[J].Computer Networks (S1389-1286), 2020, 181:107526.

[12] Meng J, Tan H, Li X Y, et al. Online Deadline-aware Task Dispatching and Scheduling in Edge Computing[J]. IEEE Transactions on Parallel and Distributed Systems (S1045-9219), 2020, 31(6): 1270-1286.

[13] Ye D, Wu M, Tang S, et al. Scalable Fog Computing with Service Offloading in Bus Networks[C]//3rd International Conference on Cyber Security and Cloud Computing. Beijing, China: IEEE, 2016: 247-251.

[14] Verba N, Chao K M, Lewandowski J, et al. Modeling Industry 4.0 based Fog Computing Environments for Application Analysis and Deployment[J]. Future Generation Computer Systems (S0167-739X), 2019, 91(1): 48-60.

[15] Wu C, Wang L. A Deadline-aware Estimation of Distribution Algorithm for Resource Scheduling in Fog Computing Systems[C]//IEEE Congress on Evolutionary Computation. Wellington, New Zealand: IEEE, 2019: 660-666.

[16] 林闯, 胡杰, 孔祥震. 用户体验质量(QoE)的模型与评价方法综述[J]. 计算机学报, 2012, 35(1): 1-15. Lin Chuang, Hu Jie, Kong Xiangzhen. Survey on Models and Evaluation of Quality of Experience[J]. Chinese Journal of Computers, 2012, 35(1): 1-15.

[17] Lin Y, Shen H. Cloud Fog: Towards high quality of experience in cloud gaming[C]//44th International Conference on Parallel Processing. Beijing, China: IEEE, 2015: 500-509.

[18] Aazam M, St-Hilaire M, Lung C, et al. MeFoRE: QoE based Resource Estimation at Fog to Enhance QoS in IoT[C]//23rd International Conference on Telecommunications. Thessaloniki, Greece: IEEE, 2016: 1-5.

[19] Buyya R., Srirama S. N. 雾计算与边缘计算: 原理和范式[M]. NJ, USA: Wiley, 2019: 31-50.

[20] Jalali F, Hinton K, Ayre R, et al. Fog Computing May Help to Save Energy in Cloud Computing[J]. IEEE Journal on Selected Areas in Communications (S0733-8716), 2016, 34(5): 1728-1739.

[21] Li W, Delicato F C, Zomaya A Y. Adaptive Energy-efficient Scheduling for Hierarchical Wireless Sensor Networks[J]. ACM Transactions on Sensor Networks (S1550-4859), 2013, 9(3): 1-34.

[22] Dietrich I, Dressler F. On the Lifetime of Wireless Sensor Networks[J]. ACM Transactions on Sensor Networks (S1550-4859), 2009, 5(1): 1-39.

[23] Deng R, Lu R, Lai C, et al. Optimal Workload Allocation in Fog-cloud Computing Towards Balanced Delay and Power Consumption[J]. IEEE Internet of Things Journal (S2327-4662), 2016, 1(1): 1-1.

[24] Wan J, Chen B, Wang S, et al. Fog Computing for Energy-aware Load Balancing and Scheduling in Smart Factory[J]. IEEE Transactions on Industrial Informatics (S1551-3203), 2018, 14(10): 4548-4556.

[25] Gu L, Zeng D, Guo S, et al. Cost Efficient Resource Management in Fog Computing Supported Medical Cyber-Physical System[J]. IEEE Transactions on Emerging Topics in Computing (S2168-6750), 2017, 5(1): 108-119.

[26] Sturzinger E, Tornatore M, Mukherjee B. Application- aware Resource Provisioning in a Heterogeneous Internet of Things[C]//International Conference on Optical Network Design and Modeling. Budapest: IEEE, 2017: 1-6.

[27] 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

[28] Mao Y, Zhang J, Letaief K B. Dynamic computation offloading for mobile-edge computing with energy harvesting devices. IEEE Journal on Select Areas in Communications, 2016, 34(12):

3590-3605

[29] You C, IIuang K. Multiuser resource allocation for mobile-edge computation offloading//Proceedings of the 2016 IEEE Global Communications Conference. Washington, USA, 2016: 1-6

[30] Zhao P, Tian II, Qin C, et al. Energy-saving offloading by jointly allocating radio and computational resources for mobile edge computing. IEEE Access,2017,5:11255-11268

[31] Lyu X, Tian II, Jiang L, et al. Selective offloading in mobile edge computing for the green Internet of Things.IEEENetwork,2018,32(1):54-60

[32] 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

[33] 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

[34] Alam M G R, Ilassan 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

[35] Qiu X, Liu L, Chen W, et al. Online deep reinforcement learning for computation offloading in blockchain-empowered mobile edge computing. IEEE Transactions on VehicularTechnology,2019,68(8):8050-8062

[36] Zhang C, Zheng Z. Task migration for mobile edge computing using deep reinforcement learning. Future Generation ComputerSystems,2019,96:111-118

[37] YAN J, BI S, Zhang Y, Optimal offloading and resource allocation in mobile-edge computing with inter-user task dependency[C]. IEEE Global Communications Conference,2018

[38] YAN J, BI S, Zhang Y, Optimal offloading and resource allocation in mobile edge computing with inter user task dependency[J]. IEEE Transactions on Wireless Communications,2020,19(1): 235-250

[39] 黄祥岳.移动边缘计算缓存优化与用户移动性预测研究D].杭州：浙江大学，2019.

[40] Wang C, Liang C L，et al. Joint computation offloading and interference management in wireless cellular networks with mobile edge computing[J]. IEEE Trans on Vehicular Technology,2017,66(8):743-7445

[41] Fang F, Zhang H, Cheng J, et al. Energy efficient resource allocation for downlink non-orthogonal multiple access network[J]. IEEE Trans on Communications,2016,64(9):3772-3732