移动边缘计算卸载技术综述

谢人超，廉晓飞，贾庆民，黄韬，刘韵洁
（北京邮电大学网络与交换技术国家重点实验室，北京 100876）

摘 要：移动边缘计算(MEC，mobile edge computing)中计算卸载技术即将移动终端的计算任务卸载到边缘网络，解决了设备在资源存储、计算性能以及能效等方面存在的不足。同时相比于云计算中的计算卸载，MEC 解决了网络资源的占用、高时延和额外网络负载等问题。首先介绍了 MEC 的网络架构及其部署方案，并对不同的部署方案做了分析和对比；然后从卸载决策、资源分配和系统实现这 3 个方面对 MEC 计算卸载关键技术进行了研究；通过对 5G 环境及其 MEC 部署方案的分析提出了两种计算卸载优化方案，总结归纳了目前 MEC 中计算卸载技术面临的移动性管理、干扰管理以及安全性等方面的核心挑战。

关键词：MEC；MEC 部署方案；计算卸载；计算卸载决策；资源分配；计算卸载系统
中图分类号：TP393
文献标识码：A
doi: 10.11959/j.issn.1000-436x.2018215

Survey on computation offloading in mobile edge computing

XIE Renchao, LIAN Xiaofei, JIA Qingmin, HUANG Tao, LIU Yunjie
Stat Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

Abstract: Computation offloading in mobile edge computing would transfer the resource intensive computational tasks to the edge network. It can not only solve the shortage of mobile user equipment in resource storage, computation performance and energy efficiency, but also deal with the problem of resource occupation, high latency and network load compared to cloud computing. Firstly the architecture of MEC was introduce and a comparative analysis was made according to various deployment schemes. Then the key technologies of computation offloading was studied from three aspects of decision on computation offloading, allocation of computing resource within MEC and system implement of MEC. Based on the analysis of MEC deployment scheme in 5G, two optimization schemes on computation offloading was proposed in 5G MEC. Finally, the current challenges in the mobility management was summarized, interference management and security of computation offloading in MEC.

Key words: MEC, MEC deployment scheme, computation offloading, decision on computation offloading, resource allocation, computation offloading system

1 引言

随着移动通信技术的发展和智能终端的普及，各种网络服务和应用不断涌现，用户对网络服务质量、请求时延等网络性能的要求越来越高。尽管新的移动设备的中央处理单元（CPU, central process unit）的处理能力越来越强大，但依然无法在短时间内处理巨大的应用程序[1-3]。此外，本地处理这些应用也面临另一个问题，即电池电量的快速消耗和自身损耗。这些问题严重影响了应用程序在用户设备上的运行效率和用户体验。为了解决以上问题，业界提出了移动边缘计算和计算卸载技术。

收稿日期：2017−12−25；**修回日期**：2018−07−04
基金项目：中央高校基本科研业务费专项基金资助项目（No.2018PTB-00-03）；国家自然科学基金资助项目（No.61501042）
Foundation Items: The Fundamental Research Funds for the Central Universities (No.2018PTB-00-03), The National Natural Science Foundation of China (No. 61501042)

移动边缘计算[4]是指在移动网络边缘部署计算和存储资源，为移动网络提供 IT 服务环境和云计算能力，从而为用户提供超低时延和高带宽的网络服务解决方案。作为 MEC 中关键技术之一，计算卸载[5-6]是指终端设备将部分或全部计算任务交给云计算环境处理的技术，以解决移动设备在资源存储、计算性能以及能效等方面存在的不足。计算卸载技术主要包括卸载决策、资源分配和卸载系统实现这三方面。其中，卸载决策主要解决的是移动终端决定如何卸载、卸载多少以及卸载什么的问题；资源分配则重点解决终端在实现卸载后如何分配资源的问题；对于卸载系统的实现，则侧重于移动用户迁移过程中的实现方案。

得益于全球各国对 MEC 网络体系架构研究的支持，MEC 计算卸载技术成为移动网络研究的热点之一。文献[7]从计算、缓存和通信等多角度出发，介绍了当前 MEC 技术进展。文献[4]对 MEC 的网络架构以及现有的MEC计算卸载方案进行了介绍。文献[8]详细总结了 MEC 的关键技术以及研究进展。虽然相关研究取得了一定的成果[7-11]，但绝大多数都是从 MEC 整体架构或实现算法方面出发，在 MEC 计算卸载技术的系统实现和方案对比上总结得不够完善，缺少对 MEC 理论进行相关梳理和兼容当前网络的 MEC 实施方案，也没有详细介绍 MEC 中计算卸载技术面临的问题和挑战。因此，从众多研究成果中及时地总结出 MEC 中计算卸载技术之间的差异以及各自的优势与不足，为未来研究方向提供理论基础和研究方向具有重要意义。本文对MEC网络架构和部署以及MEC计算卸载方案进行了详细的论述，通过对相关的研究成果的分析和比较，提出了 5G 环境下的 MEC 计算卸载方案。

2　MEC 网络架构和部署

2014 年，欧洲电信标准化协会（ETSI, European Telecommunications Standards Institute）为了将边缘计算融合进移动网络的架构，提出了移动边缘计算[12]。其中，“M”是英文单词“mobile”的缩写，MEC 特指移动网络中的边缘计算，但随着研究推进，ETSI 将“M”定义为“multi-access”，旨在将边缘计算拓展到 Wi-Fi 等非 3GPP 场景下，MEC 的定义逐渐过渡为“多接入边缘计算”（MEC, multi-access edge computing）[9]。由于业界对 MEC 的研究重点仍是移动网络，因此现在业界还是以“移动边缘计算”称之。

MEC 可以被看作是运行在移动网络边缘的云服务平台，通过将部分业务处理和资源调度的功能部署到云服务平台上来实现服务性能和用户体验的提升。MEC 将原本位于云数据中心的服务和功能“下沉”到移动网络的边缘，通过在移动网络边缘部署计算、存储、网络和通信等资源，不仅减少了网络操作，而且降低了服务交付时延，提升用户服务体验。同时，大幅增长的网络数据，对回传链路和移动核心网造成了巨大的链路负载，MEC 在网络边缘部署服务器后，可以在边缘对用户进行响应，降低了对回传网和核心网的带宽要求。本节将从网络架构和部署方案两方面对 MEC 进行详细介绍。

2.1　MEC 网络架构

如图 1 所示，MEC 服务平台主要由 MEC 基础设施和 MEC 应用平台、应用管理系统三层逻辑实体组成[13]。

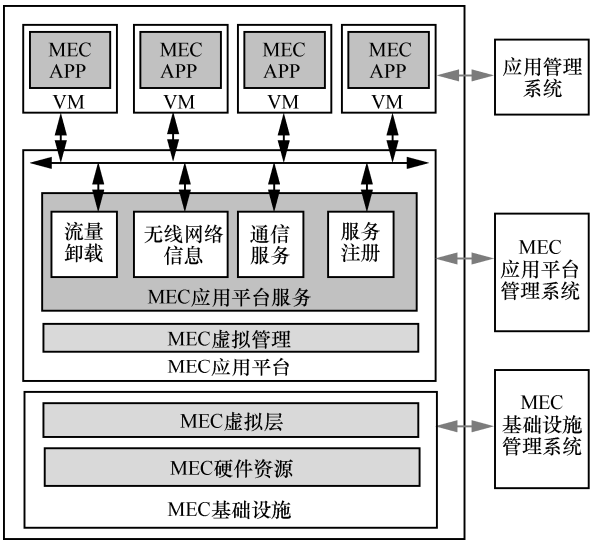


图 1　MEC 服务平台三层逻辑实体

MEC 基础设施由基于网络功能虚拟化的硬件资源和虚拟化层组成。其中，硬件资源主要提供底层的计算、存储以及控制功能；而硬件虚拟化组件（包括基于 Openstack 的虚拟操作系统、KVM 等）则主要完成计算处理、缓存、虚拟交换及相应的管理功能。

MEC 应用平台承载业务的对外接口适配功能，通过 API 完成和 eNodeB 及上层应用层之间的接口协议封装，主要提供流量卸载、无线网络信息服务、通信服务以及应用与服务的注册等功能，具备相应的底层数据分组解析、内容路由选择、上层应用注

册管理、无线信息交互等基础功能。基于网络功能虚拟化的虚拟机（VM ，virtual machine）则将 MEC 功能组件层封装的基础功能进一步组合形成虚拟应用，包括无线缓存、本地内容转发、增强现实、业务优化等，并通过 API 和第三方应用 APP 实现对接。

基于 MEC 逻辑实体，文献[9]提出了 MEC 基本参考框架。如图 2 所示，该架构主要由移动边缘系统层与移动边缘服务器层组成。前者主要由用户应用程序生命周期管理模块（LCM，lifecycle management）、操作支持系统（OSS，operation support system）和移动边缘编排器(MEO，mobile edge orchestrator)组成，主要用于管理应用程序的生命周期、应用规则、服务授权以及流量规则等。后者则由移动边缘平台管理器、虚拟化基础设施管理器和 MEC 服务器组成，主要负责虚拟化计算存储资源的分配、管理和发布。

位于 UE 中应用程序（UE App）和面向客户服务（CFS, customer facing service）的服务门户通过移动边缘系统层（mobile edge system level）与 MEC 系统进行交互。首先，通过 LCM 调用请求，如 MEC 系统内的 UE 应用程序的启动、终止或重新定位到 OSS。然后，OSS 决定是否授予请求。授权的请求被转发到 MEO，MEO 根据应用需求（如等待时间）将虚拟化的 MEC 资源分配给将启动的应用。MEO 与 OSS 之间通过 Mm1 参考点来触发应用程序的实例化和终止。MEO 与虚拟化基础设施管理器之间通过 Mm4 参考点来管理虚拟化资源和 VM，同时维持可用资源的状态信息。

2.2 MEC 部署方案

MEC 的服务是由拥有计算和存储功能的 MEC 服务器提供的，MEC 服务器在网络中如何部署是首先要考虑的问题。在 4G 网络中部署边缘服务器有多种方案，本文基于文献[4]中的部署方式分析 4G 网络中的部署方案，并在 4.1 节提出 5G 部署方案，表 1 列出了各部署方案间的对比分析。

1) SCC 部署方案

SCC（small cell cloud）的核心思想是通过额外的计算和存储能力来增强小基站（如微蜂窝、微微蜂窝或毫微微蜂窝）的功能，通过提供云端化的 SCeNB（small cells eNodeB）实现边缘计算[14-15]。

为了将 SCC 概念整合到移动网络架构中，该部署方案引入了小基站管理器（SCM, small cell manager）[16]，用于负责 SCeNB 计算和存储资源的管理。关于 SCC 架构的一个重要方面是如何部署 SCM。

如图 3 所示，SCC 部署根据 SCM 部署方式的不同，可分为 2 种情况。一种是集中式 SCM 部署方案如图 3 (a)中所示，SCM 位于无线接入网（RAN，radio access network），靠近 SCeNB 的集群，或作为对 MME 的扩展部署在核心网（CN, core network，）[16-17]。另一种是分布式 SCM 部署方案如图 3(b)所示，本地小基站管理器（L-SCM，local small cell manager）和虚拟本地小基站管理器（VL-SCM，virtual Local small cell manager）管理附近的 SCeNB

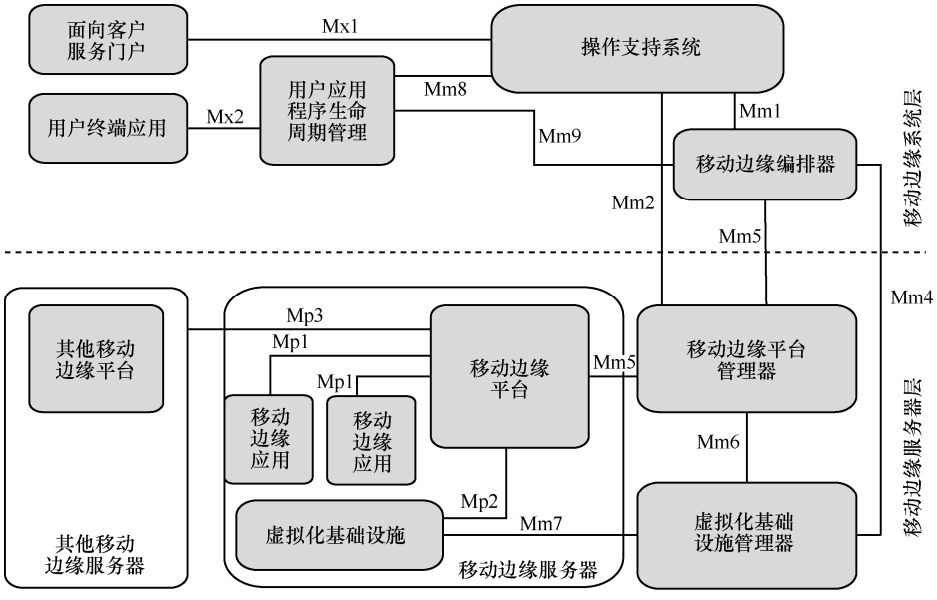


图 2 MEC 基本架构

表 1　　边缘服务器部署方案对比

方案	服务器部署位置	控制实体	控制实体部署位置	部署方式	优缺点
小基站云(SCC)	SCeNBs	SCM	1) 部署在靠近 RAN 或作为 MME 的扩展部署在 CN 2) 分层式部署本地 SCM（L-SCM）和位于 CN 的 SCM（R-SCM）	分布式部署	优点：靠近网络边缘，有较低的时延 缺点：1）安装成本高；2）在边缘部署会引入鉴权和认证及安全等问题
移动微型云(MCC)	eNodes	无	—	分布式部署	优点：1）靠近网络边缘，减小终端时延；2）MMC 服务器互连，在 VM 迁移时能保证业务连续性 缺点：1）无集中式控制实体会增大信令开销；2）边缘部署会引入鉴权和认证以及安全问题
快速移动私人云(FMPC)	接近 RAN 的运营商云（cloud）	MC	在 SDN 传输网络中分布式部署	分布式部署	优点：1）有较低时延；2)引入 SDN，信令开销小；3）减小基站压力 缺点：引入鉴权和认证及安全问题
漫游云(FMC)	CN 侧	FMCC	在分布式 CN 后以集中方式部署	分布式部署	优点：网络接入的鉴权认证和安全问题得到解决 缺点：1)有相对较高的时延；2）占用核心网资源
CONCERT	eNode B 或者 CN	Conductor	在控制平面以集中或分层方式部署	分层式部署	优点：1）分层地放置资源可以灵活和弹性地管理网络和云服务；2）能更好地实现负载均衡

群集的计算和存储资源，而位于 CN 的远程小基站管理器（R-SCM，remote small cell manager）集成在 MME 的功能中，管理连接到 CN 的所有 SCeNB 的资源[18]。

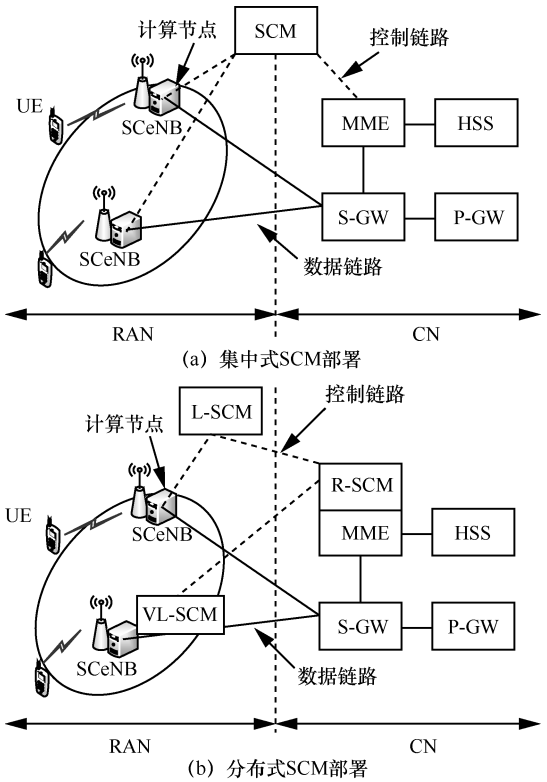


图 3　SCC 部署方案

2) MMC 部署方案

MMC (mobile micro cloud)[19]部署的核心思想也是将 MMC 部署在基站，以降低用户的访问时延，与 SCC 部署方案不同的是，MMC 部署没有引入任何控制实体。如图 4 所示，在 MMC 中，控制功能直接扩展在 MMC 服务器上，和 SCC 以分层方式部署在 SCeNB 的 VL-SCM 一样，直接在 MMC 服务器上扩展。各个 MMC 之间是互连的，在 VM 迁移时能够更好地保证业务的连续性。但是由于没有集中式控制实体会引入信令开销过大的问题。

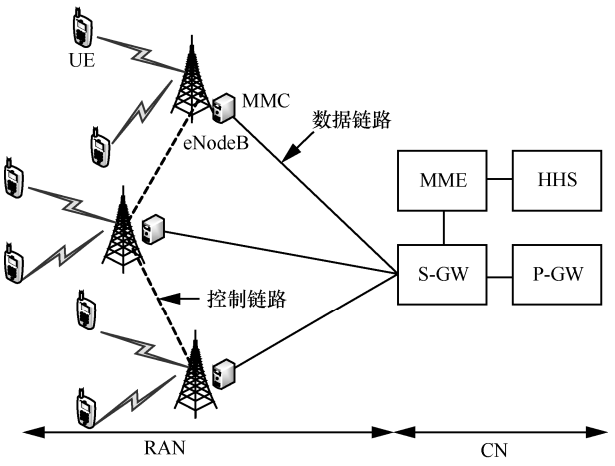


图 4　MMC 部署方案

3) FMPC 部署方案

如图 5 所示，FMPC（fast move personal cloud）[20]通过 SDN（software defined network）和 NFV（network functions virtualisation）技术将云服务与移动网络进行融合，相比于 SCC 和 MMC 部署方案，FMPC 的云服务资源不是直接部署在接入节点 eNodeB 或 SCeNB, 而是部署在接近 RAN 侧的运营商云（cloud），与 SCC 类似，FMPC 也是分布式部署，并引入了一个新的控制实体 MC (MobiScud control)，与移动网络、SDN 交换机和主云进行通信。

MC 具有两种功能：一是监控移动网络网元之间控制平面的信令消息，以便了解 UE 的动态，如用户切换等；二是在支持 SDN 的传输网络内编排和转发数据业务，以便于应用卸载和 VM 迁移。

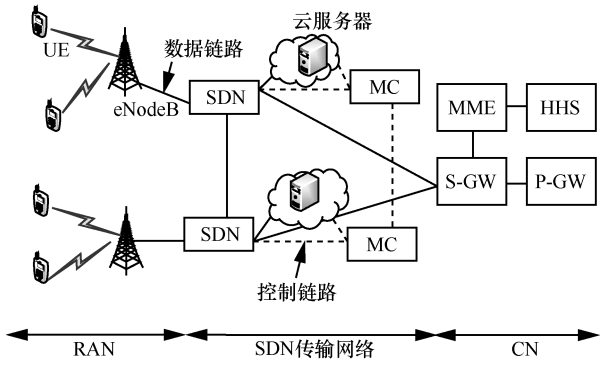


图 5 FMPC 部署方案

4) FMC 部署方案

FMC(follow me cloud)[21-22]部署如图 6 所示，其关键思想是通过在分布式的数据中心（DC）部署服务器来提供边缘服务。与 SCC、MMC 和 FMPC 相比，FMC 的计算存储资源部署在 CN，离 UE 更远。与 SCC 和 FMPC 一样，FMC 也引入了新的控制实体 FMCC(FMC control)，FMCC 既可以是在现有网络节点并置的功能实体，也可以是在 DC 上运行的软件，主要用于管理 DC 的计算和存储资源，并决定将哪一个 DC 关联到 UE。FMCC 可以集中部署，也可以分层部署。

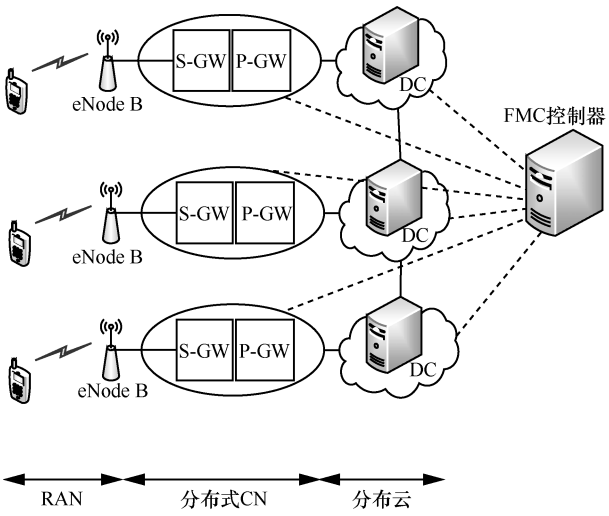


图 6 FMC 部署方案

5) CONCERT 部署方案

文献[23]提出了融合云和蜂窝系统的概念，缩写为 CONCERT。CONCERT 利用 NFV 和 SDN 技术，将计算和存储资源虚拟化，整个架构划分为控制平面和数据平面。其中，控制平台由 conductor 组成，是管理 CONCERT 架构的通信、计算和存储资源的控制实体。conductor 部署既可以是集中式的，也可以是分层式的，相比于 SCC 和 FMC，具有更好的可扩展性。数据平面由 eNode B、SDN 交换机和计算资源的无线接口设备组成，如图 7 所示。在该方案中，计算资源主要用于基带处理和应用程序处理（如用于卸载应用）。在该方案中可根据资源占用状况动态选择本地服务器或中心服务器。例如，当本地计算资源充足时，可以选择在本地服务器卸载应用程序，否则在资源充足的中心服务器进行处理。这种在网络中分层地放置资源可以实现网络和云服务的弹性管理。

综上所述，MEC 服务器部署的选择取决于多种因素，包括可扩展性、物理部署约束、性能指标（如延时）等。此外，部署 MEC 服务器还需要考虑延时、安装成本和服务质量（QoS, quality of service）之间的权衡。例如，对于仅需要低计算能力的 UE 而言，可以由位于 eNode B 本地 MEC 服务器来服务，而对于高要求的应用，则需要离 UE 更远的更强大的 MEC 服务器来服务。

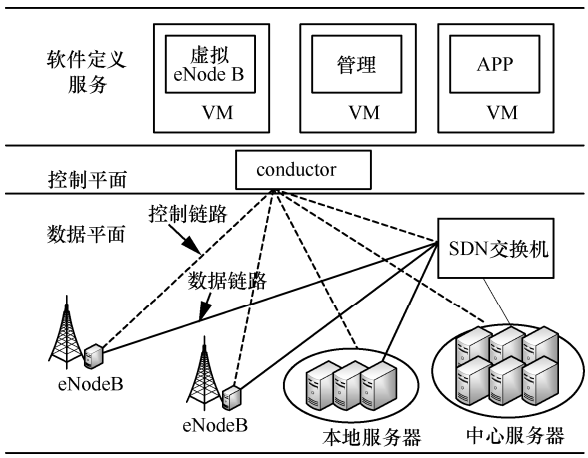


图 7 CONCERT 部署方案

3 MEC 计算卸载技术研究

MEC 计算卸载技术是指受资源约束的设备完全或部分地将计算密集型任务卸载到资源充足的云环境中，主要解决了移动设备在资源存储、计算性能以及能效等方面存在的不足。MEC 计算卸载技术不仅减轻了核心网的压力，而且降低了因传输带来的时延。如出于安全考虑的视频监控系统，传统

的监控系统由于设备处理能力有限，只能通过视频监控系统捕获各种信息，然后将视频送到云端的监控服务器上，从这些视频流中提取有价值的信息，而这种方式会传输巨大的视频数据，不仅加重了核心网的流量负载，而且存在较高的延时。而 MEC 卸载技术可以直接在离监控设备很近的 MEC 服务器上直接进行数据分析，这不仅减轻了网络压力，而且解决了监视系统的能耗等瓶颈问题。同时，物联网技术的发展也需要计算卸载技术的支撑，由于物联网设备的资源通常是有限的，若要达到万物互连的场景，就需要在终端设备受限的情况下,需要将复杂的计算任务卸载到边缘服务器。计算卸载技术不仅有助于物联网的发展，而且能降低终端设备的互连准入标准。此外，计算卸载技术也促进了零延时容忍新兴技术的发展。例如，在车联网服务、自动驾驶等领域，车辆需要通过实时感知道路状况、障碍物、周围车辆的行驶信息等，这些信息可通过 MEC 计算卸载技术实现快速计算和传输，从而预测下一步该如何行驶。

计算卸载作为 MEC 的关键技术，目前已有很多相关研究成果，主要包含卸载决策和资源分配两个问题，其中，卸载决策研究的是用户终端要不要卸载、卸载多少和卸载什么的问题。资源分配则是研究将资源卸载到哪里的问题。本节将详细介绍计算卸载当中的卸载决策和资源分配问题，并对目前在工程上实现的卸载系统进行分析。

3.1　卸载决策

卸载决策是指 UE 决定是否卸载、卸载多少以及卸载什么的问题。在卸载系统中，UE 一般由代码解析器、系统解析器和决策引擎组成，其执行卸载决策分为 3 个步骤：首先，代码解析器确定什么可以卸载，具体卸载内容取决于应用程序类型和代码数据分区；然后，系统解析器负责监控各种参数，如可用带宽、要卸载的数据大小或执行本地应用程序所耗费的能量；最后决策引擎确定是否卸载。

如图 8 所示，UE 卸载决策结果分为本地执行、完全卸载和部分卸载 3 种情况。具体决策结果由 UE 能量消耗和完成计算任务时延决定。卸载决策目标主要分为降低时延、降低能量消以及权衡时延与能量 3 方面。本节将从优化目标角度来分别分析目前的卸载决策方案问题，详细的研究进展在表 2 列出。

1) 以降低时延为目标的卸载决策

如果在本地执行应用任务，所耗费的时间为应用执行任务的时间，而如果将任务卸载到 MEC，所耗费的时间将涉及 3 个部分：将需要卸载的数据传送到 MEC 的时间、在 MEC 处理任务的时间和接收

表 2　　MEC 中卸载决策方案总结归类

优化目标	卸载类型	相关文献	关键研究点
以降低时延为目标	全部卸载	文献[24-26]	• 以降低时延为目标的卸载决策方案 • 分析时延模型 • 多用户的计算卸载问题，建模为 NP-hard 问题 • 基于 Lyapunov 优化的动态卸载算法 • 联合优化通信资源和计算资源分配 • 采用 Stackelberg 博弈论的方法优化多用户卸载方案
	部分卸载	文献[27-28]	• 研究卸载内容之间的依赖关系 • 与计算任务全部卸载的时延相对比 • 采用启发式算法
以降低能量消耗为目标	全部卸载	文献[29-31]	• 在保证时延的要求下，以降低能耗为目标的卸载决策方案 • 能量模型的优化方案 • 相关参数的对比分析，如信道链路状况、CPU、系统容量等 • 在线学习方案和预先计算的离线策略 • 采用人工鱼群算法、约束性马尔可夫链方法
	部分卸载	文献[32-33]	• 研究卸载内容之间的依赖关系 • 与计算任务全部卸载的能耗相对比 • 基于阈值结构的最优卸载方案 • 采用凸优化的方法
权衡能耗和时延为目标	全部卸载	文献[35-37]	• 以权衡能量和时延为目标的卸载决策方案 • 能量和时延的权衡模型建立 • 采用基于 Lyapunov 的优化算法、在线学习策略
	部分卸载	文献[34]	• 研究卸载内容之间的依赖关系 • 与计算任务全部卸载的时延和能耗均衡方案对比

从 MEC 返回数据的时间。因此，将计算任务卸载到 MEC 所产生的时延直接影响用户的 QoS，为了保证 QoS，出现了大量以降低时延为目标的研究，其中涉及不同的优化算法和应用场景。

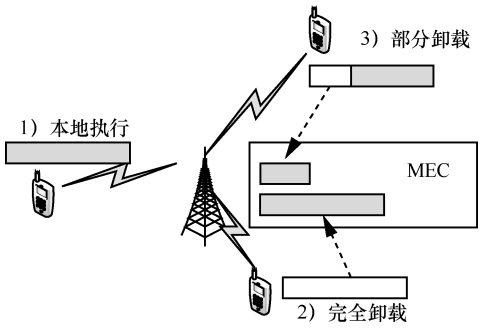


图 8 卸载决策

文献[24]以优化卸载过程中的时延为目标，设计了图 9 的计算卸载模型。该模型决定了在每一个时隙内，是否将缓冲任务卸载到 MEC 服务器。在此模型中，卸载决策主要由缓冲任务的队列状态、本地处理单元和传送单元 3 个部分来完成。其中，MEC 服务器会给传送单元返回信道状态信息（CSI ，channel state information），包含应用缓冲队列的状态、UE 与 MEC 计算消耗的能量以及 UE 与 MEC 之间的信道状态等。最后计算卸载策略根据优化目标做出是否卸载的决定。作者使用马尔可夫决策过程(MDP ，Markov decision process)对每个任务的平均时延和设备的平均功耗进行分析，并使用一维搜索算法找到最优随机计算卸载策略。

作者将其提出的算法方案与本地执行，完全卸载到 MEC 和贪婪的卸载策略做了对比，仿真结果表明，提出的方案与全部在本地执行相比，时延减少了 80%，与在远程云端执行相比，时延减少了约 44%。尽管作者所设计的方案在性能方面带来显著提升，但是此卸载模型也存在缺点，比如 UE 需要 MEC 服务器根据 CSI 提供的反馈决定是否卸载以及卸载多少计算任务，这种机制引入了过多的信令开销。

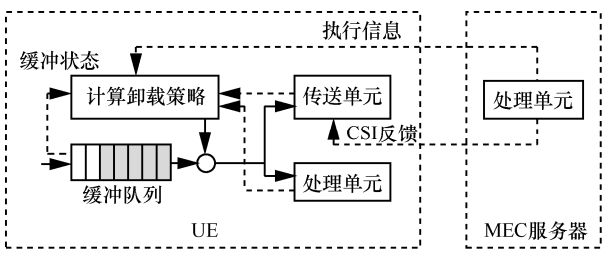


图 9 计算卸载模型

为了降低时延，在文献[25]提出的方案中，优化目标包含了执行时延和执行故障优化两部分。对这两个方面的优化，不仅能使任务时延最小化，还能保证故障率最低，降低了卸载失败的风险。作者提出采用动态电压频率调整和功率控制的技术分别优化计算执行过程和计算卸载的数据传送。基于这个模型，提出了一种基于 Lyapunov 优化的动态卸载（LODCO, low-complexity Lyapunov optimization based dynamic computation offloading）算法。LODCO 算法会在每个时隙中进行卸载决定，然后为 UE 分配 CPU 周期（在本地执行）或分配传输功率（卸载到 MEC），结果表明能将运行时间缩短 64%。此外，文献[26]提出的以降低时延为目标的最优卸载方案，考虑了 MEC 的计算资源有限的情况，如何进行卸载决策和资源分配的问题，提出了分层的 MEC 部署架构，采用 Stackelberg 博弈论的方法解决了多用户卸载方案。

文献[27-28]也提出了以优化时延为目标的在线任务卸载算法。文献[27]对于顺序任务，即线性拓扑任务图，找到最优的任务卸载到边缘云，而对于并发任务，则采用负载均衡启发式算法将任务卸载到边缘云中，以使 UE 和 MEC 服务器之间的并行最大化，达到最小的时延。文献[28]针对部分卸载模型，提出了任务之间的依赖关系对卸载决策的影响，并采用多项式时间算法来解决卸载决策的最优方案。

2) 以降低能量消耗为目标的卸载决策

将计算卸载到 MEC 服务器消耗的能量主要由两部分组成，一是将卸载数据传送到 MEC 的传送能量，二是接收 MEC 返回的数据所消耗的能量。

文献[29]提出的能量优化模型中的能量优化不是某个时刻的优化过程，而是一个持续的优化过程。资源分配方案也不只包含无线资源的分配，还有计算资源的联合分配，主要取决于信道状态、终端发出的任务队列状态等。做出的决策结果有 3 种情况：在 UE 处理计算任务、将计算任务卸载到基站（文献[29]中假设基站有计算存储等能力）或无效状态。文献[29]以在满足应用时延的同时优化 UE 处的能量消耗为目标，提出了两种资源分配方案。第一种策略基于在线学习，网络状态动态的调整以适应 UE 的任务要求。第二种策略是预先计算的离线策略，需要每个时隙的数据速率，无线信道状况的信息支持。信息由 UE 在 k 时刻发给基站的持续

性状态信息 $s_k=(b_k+x_k)$ 表示，b_k 为缓存的状态信息，x_k 为信道状态信息。最后提出两种方案下的能量模型，由 3 种不同的决策结果决定能量消耗。此模型由 MDP 来分析解决，结果表明预计的离线策略在低负载的情况下能量消耗优于在线策略高达 50%。文献[30]在离线策略的基础上设计了两种动态离线策略用于卸载，确定性离线策略和随机离线策略。实验数据也表明在节能方面有高达 78% 的提升。

文献[31]提出了在保证时延的情况下对能量进行优化的卸载方案。该方案同时考虑了前传网络和回传网络的链路状况，采用人工鱼群算法进行全局优化。作者在验证卸载方案的实验中，设计场景由 20 个移动终端、5 个微基站和 3 个毫微微基站（MEC 服务器部署在毫微微基站）组成，通过与随机卸载方案、本地卸载方案以及转化方案进行对比，基于人工鱼群算法的卸载方案节约大约 30%的能耗，但是该方案的不足之处是算法复杂度过高。

相比于以前文献中提出的全部卸载方案，文献[32]是在多 UE 的情况下，提出部分卸载方案。但是这种方案需要应用程序的支持，如果应用程序不可分割或分割后的各个部分存在紧密联系，则不能采用部分卸载决策的策略。文献[32]采用 TDMA 系统划分时隙的概念，在每个时隙内，UE 根据信道质量、本地计算能量消耗以及 UE 之间的公平性将其数据卸载到 MEC。基于满足应用时延的同时优化能量消耗的目标做出决策，提出了基于阈值的最优资源分配策略，最优分配策略为每个 UE 做出卸载决策，如果 UE 具有高于给定阈值的优先级，则 UE 将计算任务完全卸载到 MEC，相反，如果 UE 具有比阈值更低的优先级，则仅卸载少量计算以满足时延约束。给那些不能满足应用时延约束的 UE 更高的优先级，将计算任务在本地执行。鉴于通信和计算资源的最优联合分配具有较高的复杂度，因此提出了一种次优分配算法，该算法将通信和计算资源分配分离。仿真结果表明，与最优分配相比，这种简化使能量消耗更高，增加了 20%的能耗，但降低了算法复杂度。文献[33]对该卸载方案进行了拓展，基于 OFDMA 系统实现的卸载方案能够比在 TDMA 里实现的方案在能耗方面降低了 90%。

3) 权衡能耗和时延为目标的卸载决策

在执行复杂的计算任务时，如人脸识别系统，实时视频系统，车联网等，能耗和时延都直接影响 QoS，因此如何在执行卸载任务的时候综合考虑能耗和时延是进行卸载决策的重要考虑因素。

文献[34]提出了部分卸载决策的能耗和执行时延之间的权衡分析。卸载的过程中考虑以下几个参数：要处理的总数据量、UE 和 MEC 的计算能力，在 UE 和 SCeNB（使 UE 和 MEC 连接的中间基站）之间的信道状态以及 UE 的能耗。作者提出了一个动态调度机制，允许用户根据任务的计算队列和无线信道状态进行卸载决策。通过凸优化方法解决该优化问题。仿真结果表明，UE 的能耗随总执行时间的增加而减少。此外，作者表明如果通信信道质量很差，则需要耗费大量能源来卸载任务是得不偿失的，在这种情况下优先选择本地处理。如果信道质量良好，则卸载一部分到 MEC 能获得较小的能耗和延时。如果信道质量高，且 MEC 的计算存储资源充足的情况下可以进行全部卸载。

文献[35-37]也提出了能量和时延的权衡优化的卸载方案。其中，文献[35]通过基于 Lyapunov 的优算法和在线学习算法来解决权衡优化问题。文献[36]将问题建模为线性规划问题并提出了两种优化算法包含两阶段优化算法和迭代改进算法。文献[37]提出了多目标优化方案，并采用标量化方案和内点法来解决多目标优化问题。

如表 2 所列，本节从优化目标的角度对目前的卸载决策方案进行了详细的分析。在各个方案中，大多数的计算卸载决策方案的目标是在满足卸载应用程序可接收时延的同时最小化 UE 处的能量消耗或根据不同应用的需求在两个优化目标之间做出权衡。这些研究根据实际的计算卸载应用场景，如车联网、职能视频卸载等抽象出具体的数学模型，采取不同的优化策略，通过仿真的方式验证优化结果。

3.2　计算资源分配

一旦完成了卸载决策，接下来就要考虑合理的资源分配的问题，即卸载在哪里的问题。如果 UE 的计算任务是不可分割的或可以分割但分割的部分存在联系，这种情况下卸载任务就需要卸载到同一个 MEC 服务器。而对于可以分割但分割的部分不存在联系的计算任务，则可以将其卸载到多个 MEC 服务器。如表 3 所示，目前资源分配节点主要分为单节点分配和多节点分配。

表 3　　MEC 中资源分配方案总结归类

节点数量	优化目标	参考文献	计算卸载方案	仿真优化结果
单节点	时延	文献[38]	通过动态考虑整个迁移过程来最优化 VM 分配方案	减少 46%的时延以及减小 80%的迁移成本
		文献[39]	提出了 MEC 在满足应用程序时延要求的同时，提供服务的应用程序数量最大化为目标的卸载方案，通过优先级来分配计算节点	减小 25%的时延
		文献[26]	考虑了 MEC 的计算资源有限的情况，如何进行资源分配的问题。提出了分层的 MEC 部署架构	—
	时延和能耗	文献[40]	通过分配索引策略来让 UE 选择合适的 MEC 服务器，解决了高复杂度和高通信开销问题	索引策略和普通策略相比减小了 7%的能耗
	能耗	文献[43]	提出了合作式缓存和卸载方案，MEC 服务器联合起来为 UE 执行计算和缓存任务	—
		文献[44]	采用深度学习的方法分配资源，提出了动态的卸载方案	减小了 50%的能耗
多节点	时延	文献[45]	多用户卸载，将问题建模为 NP-hard 问题，并采用背包模型去优化整个资源分配和负载均衡的问题	相比于全部在 UE 执行任务减少了 70%的延时，比在 CC 执行减小了 58%的时延
		文献[46]	提出了干扰管理的方案，在最小化干扰的条件下进行通信资源分配、计算资源分配的方案	减小了 40%的时延
	能耗和时延	文献[41]	提出了通过优化时延和基站能耗的集群选择策略，并对比分析不同的回传技术和网络拓扑的影响	—
		文献[42]	提出 3 种不同的云集群选择策略，分别以优化时延、优化集群总能耗和优化集群中每个 SCeNB 的能耗为目标	减少了 22%的时延，降低了 61%的能耗

1) 单一节点的计算资源分配

文献[38]以时延最小为目标，同时考虑通信、计算资源重载和 VM 迁移的能耗问题，提出了如图 10 所示的计算卸载的资源分配方案。此模型采用 MDP 解决了在 SCeNB 中的 VM 分配问题。该方案中，影响 VM 分配方案的主要因素是 VM 迁移的能耗和链路状态信息。如图 10 所示，UE1 将计算任务全部卸载到 SCeNB1，创建了 VM。而对于 UE2，考虑到从 SCeNB1 到其他 SCeNB 的时延，UE2 选择了时延更低的 SCeNB3。但考虑到 VM 的迁移成本，在 MEC 计算资源充足的情况下，VM 需要优先在距离用户更近的 MEC 中进行。为了考虑迁移成本对迁移方案的影响，作者在仿真验证中研究了迁移成本的影响，实验结果也表明 VM 在 SCeNB 中的分配比例会随着迁移成本的增加而增加。迁移成本较低时，终端用户会选择信道状态较好的 SCeNB 进行迁移。提出的资源分配方案在两个微基站，两个用户的情况下能减少 46%的时延和减少 80%的迁移成本。

文献[39]提出 MEC 在满足应用程序时延要求的同时，使提供服务的应用程序数量最大化的卸载方案。在这种方案中，单个应用程序的放置取决于应用程序的优先级（有较高时延要求的有更高的优先级）和计算资源的可用性。卸载的应用程序首先被传送到 MEC 的本地调度程序，调度程序检查是否存在足够可用资源的计算节点，如果存在并且具有足够的计算资源，则在该节点分配 VM，然后在该 MEC 节点处理该应用程序，最后将结果返回给 UE，但是，如果 MEC 服务器提供的计算能力不足，则调度程序将应用程序委托给中心云（CC，central cloud）进行处理。为了最大化处理的应用程序的数量同时满足时延要求，文献[39]提出了基于优先级的卸载策略，为每个优先级级别定义了几个缓冲区阈值，如果缓冲区已满，应用程序将被传送到 CC 处理，通过递归算法找到缓冲区阈值的最佳大小。基于优先级的卸载策略能减小 25%的时延。

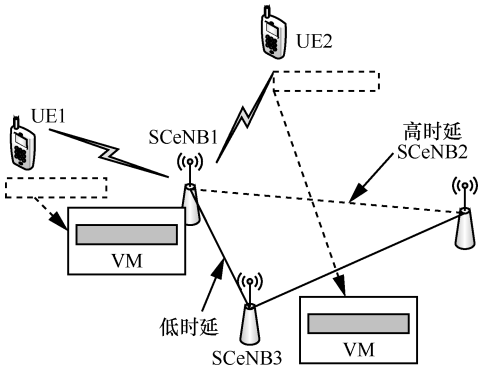


图 10　单个节点下的 VM 分配

文献[40]考虑了 UE 密集的热点地区的服务场景，结合时延和能耗问题，提出了一个等价离散

MDP 框架的最优策略。但是随着 MEC 服务器数量的增加，这种方法导致通信开销和复杂度较高，因此通过为应用程序分配索引策略来解决这个问题，每个基站根据计算资源的状态计算自己的索引策略，索引策略由基站广播，从而 UE 根据广播内容选择最合适的 MEC 服务器。实验结果表明索引策略和普通策略相比减小了 7%的能耗。文献[26]考虑了在 MEC 的计算资源有限的情况下，如何实现卸载决策和资源分配，采用 Stackelberg 博弈论的方法解决多用户卸载方案。

单一节点的资源分配虽然实现了资源分配，但无法实现网络间的资源互补，容易产生负载失衡问题，因此，考虑在多节点间卸载计算资源成为提升卸载性能的主要途径。

2) 多节点计算资源分配

计算节点的选择不仅对时延有显著影响，对计算节点的能耗也有很大影响。文献[41]的主要目标是分析集群大小（即执行计算的 SCeNB 数量）对卸载应用程序的时延和 SCeNB 的能耗影响。作者提出了选择不同集群的动态优化算法，并且对不同的回程网络（环形网络、树网络等）和技术（光纤技术、微波技术、LTE 技术等）进行分析。作者以优化时延和基站能耗为目标，时延方面主要有 3 个组成部分，包含 UE 传送到 SCeNB 的时延、在 SCeNB 处理任务的时延以及从 SCeNB 传到 UE 的时延。UE 和 SCeNB 之间的传送时延主要取决于信道质量和传送的数据量。在 SCeNB 的计算时延主要取决于不同的集群数量、计算任务的数据量、计算能力等。能量方面主要取决于服务的集群数量、网络拓扑、回传技术等。仿真结果表明全网型拓扑结合光纤或微波连接在执行任务的时延方面是最有优势的。在光纤连接的环形拓扑能形成最低的能耗。另外，该研究还表明 SCeNB 数量的增加并不总是缩短时延，相反，如果过多的 SCeNB 处理卸载的应用程序，可能导致传输时延比计算时延更长。而且随着集群的增加也会导致能耗的增加，因此，选择适当的 SCeNB 的集群在系统性能中起着关键作用。

文献[42]也提出一种考虑计算节点的执行时延和 SCeNB 集群能耗的最佳组合问题，并提出了 3 种不同的选择策略。第一种策略为选择 SCeNB 是以最小化时延为目标，由于系统模型中的所有 SCeNB 被假定为一跳，所以基本上所有的 SCeNB 都包含在计算中，导致执行时延减少高达 22%，这是由于计算时延远大于传输时延导致总体时延的减小。第二种策略为最小化集群的总能耗，优先选取一个 SCeNB 来计算，抑制相邻的 SCeNB 的计算，这种策略可降低 61%的功耗，但是会导致负载不均衡。第三种策略是使集群中的每个 SCeNB 的能耗最小化，这样能解决第二种负载不均衡的问题，其主要的影响因子是集群的大小、时延、能耗和负载分布等。文献[43]提出了合作式缓存和卸载方案，MEC 服务器联合起来为 UE 执行计算和缓存任务。资源分配方案是以最大化资源利用率为目标。文献[44]采用深度学习的方法分配资源，提出了动态的卸载方案。文献[45]考虑了计算卸载时多节点间的负载均衡问题，并采用背包模型对提出的资源分配方案进行仿真验证。

如果将很多接入设备的应用同时卸载到 MEC 服务器，会产生严重的干扰问题，因此，如何在保证 QoS 的前提下进行资源的合理分配尤为关键。文献[46]提出在计算卸载时考虑干扰管理的方案，优化了计算卸载决策、物理资源块（PRB, physical resource block）分配和 MEC 计算资源分配。作者通过计算任务的数据量以及 MEC 服务器的服务能力等做出卸载决策，采用改进的图着色方法为 UE 分配 PRB，通过最小化时延来进行计算资源的分配。在本方案中，随着 UE 数量的增多，进行卸载的 UE 并不会一直增多，这是因为有更多的计算任务卸载到 MEC 服务器的时候，会导致速率下降，产生严重干扰，因此卸载的 UE 数量会减少。

资源分配作为 MEC 计算卸载技术的关键研究点，涉及将计算任务卸载到哪里的问题，上述列举的研究成果主要从网络环境、MEC 部署、传输技术等方面研究了针对卸载任务的计算资源分配问题。表 3 对 MEC 中资源分配方案进行了总结归类。可以看出，这些研究主要致力于平衡计算资源和通信资源，以达到最小化时延、能耗以及提高网络整体性能。但是大部分研究都忽略了 UE 的移动性，如果 UE 是移动的，为了保持业务的连续性，需要重新考虑资源分配的方案。本文将会在第 4 节提出考虑移动性的卸载方案。

3.3 卸载系统

现有的计算卸载一般按照划分粒度进行分类，主要分为基于进程或功能函数进行划分的细粒度计算卸载和基于应用程序或 VM 划分的粗粒度计算卸载。本节将以 MAUI 卸载系统和 Cloudlet 卸载系

统为例，分别对两种粒度的卸载系统进行介绍。

1) MAUI

MAUI[47]是以动态方式实现迁移的基于代理的计算卸载系统，属于细粒度计算卸载下的一个实例。系统架构如图 11 所示。MAUI 卸载系统以减小客户端消耗能量和延时为目的，绕过了终端设备的限制，通过远程服务器执行计算功能。

MAUI 提供一个编程环境，开发人员可以通过编写代码决定应用程序的哪些方法可以卸载到远端服务器。每次调用程序方法时，如果远程服务器可用，MAUI 系统会通过其优化框架决定是否卸载该方法。完成卸载决策后，MAUI 系统记录分析信息，用于更好地预测未来的调用是否应该卸载。

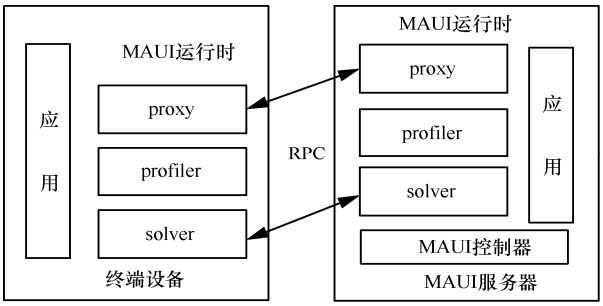


图 11 MAUI 计算卸载系统

MAUI 通过应用程序来确定卸载计算任务的成本，如远程执行计算任务需要传输的能耗，以及卸载所带来的优势（如由于卸载而节省的 CPU 周期数）。此外，MAUI 会不断检测网络连接，获取其带宽和时延信息，通过以上信息决定哪些方法应该被卸载到边缘服务器上，哪些应该继续在智能终端上本地执行。

终端设备包含 slover、proxy 和 profiler 这 3 个组件。solver 负责提供卸载决策引擎的接口，proxy 负责执行卸载过程中数据的传输与控制。profiler 用来监测应用程序并收集应用程序数据，如能量和传输要求等测量结果。

服务器端包含 slover、proxy、profiler 和 MAUI 控制器 4 个组件。其中，slover 和 proxy 执行与其客户端相似的角色，proxy 周期性地优化线性规划的决策引擎，相比于客户端，又增加了一个 MAUI 控制器组件，负责处理传入请求的身份验证和资源分配等。

除了 MAUI 之外，还有很多细粒度计算卸载系统，如实现集群并发式处理数据的 misco 系统[48]和实现动态迁移的 comet 计算卸载系统[49]。细粒度的计算卸载系统由于程序划分、迁移决策等会导致额外的能量开销，也会增加程序员的负担。

2) cloudlet

cloudlet[50]是基于动态 VM 合成技术的计算卸载系统，是粗粒度卸载的实现实例，由卡耐基梅隆大学提出，整个系统实现了 MEC 的重要功能，如快速配置（rapid provisioning）、虚拟机迁移（VM hand-off）和 cloudlet 发现（cloudlet discovery）等。快速配置指的是实现灵活的虚拟机快速配置。由于移动终端具有移动性，cloudlet 与移动终端的连接是高度动态化的，用户的接入和离开都会导致对 cloudlet 所能提供功能的需求发生变化，因此 cloudlet 必须实现灵活的快速配置。虚拟机迁移指的是为了维持网络连通性和服务的正常工作，cloudlet 需要解决用户移动性的问题。用户在移动过程中，可能超出原 cloudlet 的覆盖范围而进入其他微云的服务范围，这种移动将会造成上层应用的中断，严重影响用户体验，因此，cloudlet 必须在用户的切换过程中无缝完成服务的迁移。cloudlet 发现用于发现和选择合适的微云。cloudlet 是地理上分布式的小型数据中心，在 cloudlet 开始配置之前，移动终端需要发现其周围可供连接的 cloudlet，然后根据某些原则（如地理临近性或者网络状况信息）选择合适的 cloudlet 并进行连接。

cloudlet 主要实现过程如图 12 所示，首先，移动设备发现并准备启用 cloudlet，发送一个 VM overlay（launch VM 和 base VM 产生的二进制差异）到有 base VM 的 cloudlet 上，然后基于 base VM 和 VM overlay 创建 launch VM，配置虚拟机实例准备为卸载的应用进行服务，当任务执行完毕后，将执行结果返回给 UE，并且释放 VM。

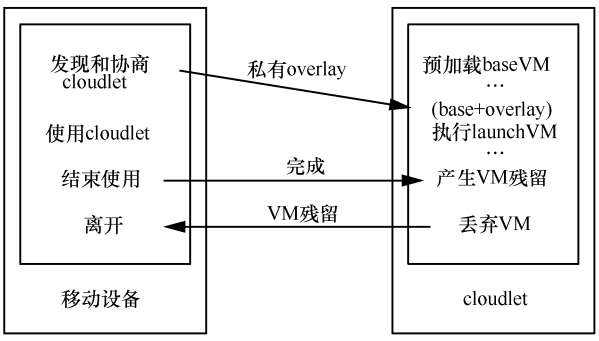


图 12 cloudlet 计算卸载系统

其他计算卸载系统如 CloneCloud[51]、Tango[52]也是基于 VM 的粗粒度计算卸载系统。CloneCloud 为了优化效率针对不同的应用设计了不同的迁移算法。Tango 则是在移动端和云端同时执行计算任务，

保留最快的结果，以此来克服无线网络的抖动问题。

4　面向 5G 的 MEC 计算卸载方案

前面 2 节讲述了 MEC 的架构和部署方案以及计算卸载技术，但主要都是针对 4G 网络。然而，5G 网络在根本架构上做出了革新，为了将 MEC 更好地融合 5G 网络，本节将分析在 5G 移动网的环境下部署 MEC 方案并提出两种卸载方案。

4.1　5G 环境下的 MEC 部署方案

5G 由接入平面、控制平面和转发平面 3 种功能平面组成[53-54]。其中，接入平面引入了多站点协作、多连接机制和多制式融合技术，构建出更灵活的接入网拓扑；控制平面基于可重构的集中网络控制功能，提供按需的接入、移动性和会话管理，支持精细化资源管控和全面能力开放；转发平面具备分布式的数据转发和处理功能，提供动态的锚点设置，以及更丰富的业务链处理能力。从整体逻辑架构来看，5G 网络采用模块化功能设计模式，并通过“功能组件”的组合，构建满足不同应用场景需求的专用逻辑网络[55]。对应上述三层逻辑划分，图 13 是 3GPP 5G 标准中给出的 5G 系统架构。控制面被分为接入管理功能（AMF）和会话管理功能（SMF）：单一的 AMF 负责终端的移动性和接入管理；SMF 负责对话管理功能，可以配置多个。AMF 和 SMF 是控制面的两个主要节点，配合两个主要节点还有统一数据管理（UDM）、认证服务器功能（AUSF）、策略管控功能（PCF），以执行用户数据管理、鉴权、策略控制等。另外还有网络开放功能（NEF）和网元存储功能（NRF）这两个平台支持功能节点，用于帮助导出网络数据，以及帮助其他节点发现网络服务。

MEC 在 5G 网络中部署的架构如图 14 所示，MEC 位于核心网与接入网融合的部分，通过 NEF 接入 5G 网络。5G 核心网络选择靠近 UE 的 UPF，通过 N6 接口执行 UPF 到本地数据网络的流量卸载。用户请求通过 UPF 到达 MEC，在 PCF 的管控下，MEC 为用户提供各种各样的缓存、计算和网络服务。在 5G 架构下的 MEC 部署中，能解决会话和业务连续性问题、QoS 和计费问题和对 MEC 本地网络的支持问题。

MEC 在 5G 网络下的具体部署方式上也非常灵活，根据运营商的部署，既可以选择集中部署，与

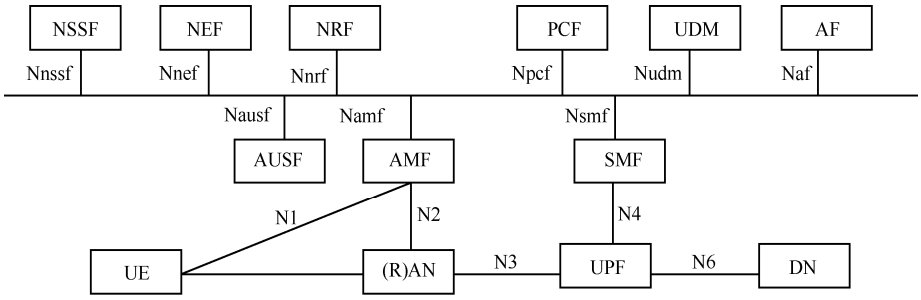


图 13　5G 系统架构

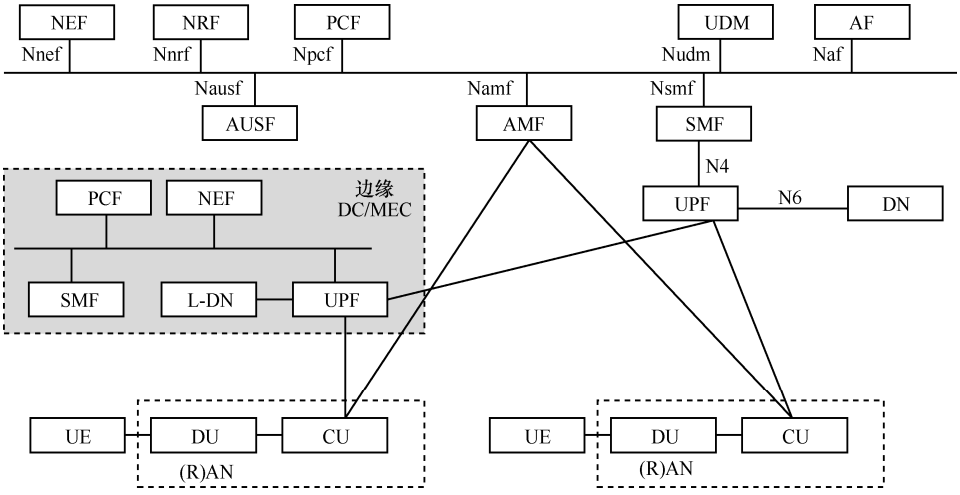


图 14　MEC 在 5G 架构下的部署

用户面设备耦合，提供增强型网关功能，也可以分布式地部署在不同位置，通过集中调度实现服务能力。这种在网络中分层地放置资源可以使网络管理更加灵活和开放。

4.2 MEC 卸载方案优化

1) MEC 协作式卸载方案设计

计算卸载的计算资源分配问题已经有很多研究，包括单节点和多节点的计算资源分配。但即使在多节点之间的计算资源分配中，大部分研究也只是致力于研究节点之间的合理资源分配，在实现计算卸载时，在各节点之间没有设置一致性目标来优化整个卸载方案，为此，本文提出以一致性目标来优化整个协作式卸载方案。

本方案在 5G 网络环境下以分层的方式部署 MEC 服务器，各个 MEC 之间协作为 UE 提供卸载服务。由于 MEC 的资源有限，因此，不仅设计了 MEC 之间可以协同服务，MEC 和 MCC 也可以协同合作。这样可以保证负载均衡的条件下实现能量和时延的最小化，具体的协作方案如图 15 所示。首先，SCeNB 尝试为附着在其上的 UE 服务，因为这会使通信延时最小，在图 15 中表现为 SCeNB1 将计算资源分配给 UE1。当 SCeNB 不能处理应用时，有两种解决方案：一种是转发给同一个集群中的所有 SCeNB，在图 15 中表现为 UE2 的计算在 SCeNB2 和 SCeNB3 处完成；另外一种是发送到 MCC 中，使 MEC 和 MCC 共同完成，在图 15 中表现为 UE3 的计算在 MCC 完成。第一种方式 MEC 之间的协作会引入的能耗消耗过多的问题，第二种方式将计算任务卸载到 MCC 会导致时延，如何权衡能耗和时延，使优化更加完善是本文方案的设计目标。

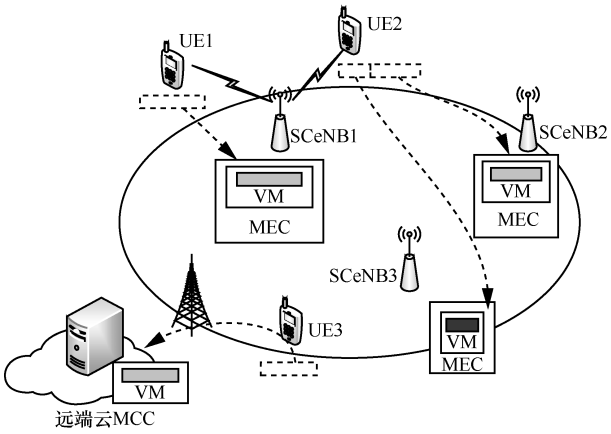


图 15 MEC 协作式卸载方案

基于提出的方案策略可以采用一致性算法来解决 MEC 的资源协同问题。SCeNB 集群中的每一个 SCeNB 都以相同的目标，通过一次次迭代找到全局最优的卸载方案。

2) 以支持业务连续性为主要目标的方案设计

分布式 MEC 服务器之间的协同服务能够尽可能协调网络资源，减小服务时延，提升用户的 QoE，但也存在一定的问题，比如如何保持业务的连续性等问题[56-57]，若计算任务是几个 MEC 服务器协同完成的，当用户移动时就需要考虑涉及 VM 的迁移等来保持业务的连续性，因此需要移动性管理技术来支撑。之前的研究主要致力于研究卸载决策和计算资源分配上，很少研究考虑到移动性带来的问题。本方案主要考虑用户移动时保证业务连续性的情况下来提升系统性能。

本方案的设计主要以在可容忍时延范围内优化能量为目标来决定是否进行 VM 迁移以及迁移之后的资源分配问题。VM 迁移消耗的能量消耗与 MEC 之间协作得到的能量减小收益来进行均衡，采用 MDP 算法来进行优化均衡。

5 问题与挑战

在边缘计算网络环境中进行计算卸载的任务，不仅能减少移动端的计算压力和能耗，还能降低传输时延。但在移动性管理、安全性、干扰管理等方面依然面临很多问题与挑战。

5.1 移动性管理

在传统的蜂窝网络中，用户在 eNode B/SCeNB 之间移动时，为保证服务的连续性，都有严格的切换流程。类似地，如果将 UE 的计算任务卸载到 MEC，如何保证服务的连续性是 MEC 计算卸载的挑战之一。在应用计算卸载技术的前提下，UE 的切换可以通过 VM 迁移来保证服务的连续性。VM 迁移即在当前计算节点处运行的 VM 被迁移到另一个更合适的计算节点[58-61]。VM 迁移的工作大部分都只考虑单个计算节点对每个 UE 进行计算的场景。当应用程序被卸载到多个计算节点时，如何有效地处理虚拟机迁移过程成为保证 QoS 的一大挑战。而且，虚拟机迁移给回程链路造成了很大的负担，并导致很高的延时。因此，实现能够以毫秒为单位快速迁移虚拟机的技术十分必要。此外，由于计算节点之间的通信限制，更现实的挑战是如何实现预先迁移计算任务

（例如基于某些预测技术），以便用户觉察不到服务的中断，从而提升用户体验。

在移动性管理中，为了完成相应任务的迁移，并满足相应的时延需求、安全等各方面的要求，需要对低时延技术、路径预测技术等加以考量，在保持业务连续性的同时达到绿色节能通信。

1) 低延时的移动性管理

物联网和车联网等低延时应用需要具有非常高的可靠性和非常低的端到端时延（毫秒级）的通信。为了支持非常低的时延，当用户从一个 MEC 区域移动到另一个区域时，虚拟机和数据会进行迁移。迁移过程可能对应用程序时延产生负面影响，如源 MEC 结束服务和目的 MEC 开启服务过程中时间相对较长，导致用户需要等待更久。为了支持这类低延时应用，MEC 系统需要用时更短的迁移。因此可以考虑在回程链路选用时延更小的高速通路，对传输数据进行压缩，简化虚拟机复原流程等。

2) 路径预测技术对移动性管理技术的支撑

移动性管理的关键是进行虚拟机和数据的迁移，传统的 MEC 迁移方案只有在移交时才会将计算任务交给另一台服务器，这种突发地传输大量的数据迁移机制会带来很高的时延并增加 MEC 网络负载。针对这个问题，有效的解决方案是在 MEC 为用户提供服务期间，利用用户轨迹的统计信息预测用户将要到达的下一个 MEC 区域，从而提前将数据传输至新的 MEC。但这一技术主要存在两个挑战：第一个挑战在于轨迹预测，准确的预测可以实现 MEC 服务器之间的无缝切换，并减少预取冗余。但要实现准确预测需要精确的建模和高复杂度的机器学习技术。第二个挑战在于如何选择预先传输的计算数据。因为预测的 MEC 并不是一定准确，所以将所有数据传输到预测的 MEC 可能会造成浪费，如何在传输的数据量和预测的准确性之间做决策也是一个必须考虑的问题。

移动性管理技术是解决新型业务更好适用于边缘计算网络的关键技术支撑，目前，ETSI 和各大厂商也在逐步解决移动性问题，相信随着研究的不断深入，移动性问题会得到全方位解决。

5.2　安全性

安全性在云计算卸载中是需要重要考虑的技术难点[62-64]。由于 MEC 是分布式部署，单点的防护能力减弱，特别是物理安全，单点突破可能导致全局突破。而多租户的形式会导致恶意用户潜入网内，利用云平台漏洞攻击网络。此外，由于软件是开源的，对代码的深入研究更容易找到脆弱点，更便于模拟攻击。卸载到云端的数据也很容易被攻击或篡改。因此设计合理的安全措施显得十分重要。另外，由于计算任务被卸载到边缘网络中，面临更加复杂的网络环境，原本用于云计算的许多安全解决方案也不再适用于边缘计算的计算卸载。

MEC 中计算卸载面临的安全问题分布在各个层级，主要包括边缘节点安全、网络安全、数据安全应用安全、安全态势感知、安全管理与编排、身份认证管理等[63]。

边缘节点安全即在边缘网络处提供安全的节点、软件加固和安全与可靠的远程升级服务，防止用户的恶意卸载行为，解决最基本的受信问题。网络安全需要保证包括防火墙、入侵检测系统、DDoS 防护、VPN/TLS 等功能，也包括一些传输协议的安全功能重用（例如 REST 协议的安全功能）。数据安全即对卸载到边缘网络中的数据进行信任处理，同时也需要对数据的访问控制进行加强，数据安全包含数据加密、数据隔离和销毁、数据防篡改、隐私保护（数据脱敏）、数据访问控制和数据防泄漏等。其中，数据加密包含数据在传输、存储和计算时的加密；另外，边缘计算的数据防泄漏也与传统的数据防泄漏有所不同，因为边缘计算的设备往往是分布式部署，需要特别考虑这些设备被盗以后，相关的数据即使被获得也不会泄漏。应用安全需要设置白名单、应用安全审计、恶意卸载内容防范等。安全态势感知、安全管理与编排即需要采用主动积极的安全防御措施，包括基于大数据的态势感知和高级威胁检测，以及统一的全网安全策略执行和主动防护，从而更加快速响应和防护。再结合完善的运维监控和应急响应机制，则能够最大限度保障边缘计算系统的安全、可用、可信。身份认证信任管理即网络的各个层级中涉及的实体需要身份认证，一些研究者提出可以通过限制共享信息来确保身份验证密钥的安全交换，完成验证过程。海量的设备接入使传统的集中式安全认证面临巨大的性能压力，特别是在设备集中上线时认证系统往往不堪重负。在必要的时候，

去中心化、分布式的认证方式和证书管理成为新的技术选择。

由于边缘计算中的计算卸载可以理解为是云计算中计算卸载的迁移，很多科学研究问题往往可以借鉴云计算中较为成熟的解决方案，但边缘网络中的安全性问题由于其特殊性不能完全借鉴云计算中的方案，首先由于边缘计算由于其分布式部署导致面临更加复杂的网络环境与不同层次的网络实体交互也使其认证问题具有挑战性。当然，边缘计算中的计算卸载的安全解决方案可以从云计算的安全解决方案中得到灵感，但毋庸置疑的是，边缘计算必须对这些方案实现新的扩展和延伸，以确保边缘计算特有的安全问题可以得到解决。

5.3 干扰管理

干扰问题[65-67]也是计算卸载中亟待解决的关键问题之一，如果将很多接入设备的应用同时卸载到 MEC 服务器，会产生严重的干扰问题，如何在保证 QoS 的前提下实现资源的合理分配同时解决干扰问题是 MEC 计算卸载面临的关键挑战之一。

干扰管理具有多种多样的实现方式，与资源管理紧密相连，这是因为干扰的本质是资源的冲突使用，网络资源分配的不理想是产生干扰的根本原因。此外，由于移动边缘计算网络是分布式部署，海量终端的卸载处理请求以及复杂的网络环境降低了资源使用率。因此，有效资源分配作为干扰管理的重要手段，一方面可以通过合理利用网络资源，增加网络容量，另一方面可以通过干扰管理修正资源分配策略，促进网络容量的提升。尽管如此，干扰管理依然面临巨大的挑战：

1) MEC 的分部署方式导致干扰调度不均匀

在 MEC 网络中，MEC 服务器的部署具有随机性，其分布与覆盖情况无法预期，这就可能导致 MEC 服务器分配不均匀，MEC 服务器部署的随机分布将导致网络中不同区域的干扰分布不均。结合位置信息和卸载请求预测智能处理干扰问题是未来 MEC 计算卸载干扰管理的重要技术点之一。

2) 计算资源和网络资源的分配方案

资源管理是解决干扰问题的核心，因此如何根据 MEC 网络环境以及终端的卸载请求，做出合理的资源分配是解决干扰问题的途径之一。文献[46]提出了在考虑干扰的情况下的计算卸载资源分配方案，通过联合优化计算卸载决策，PRB 分配和 MEC 计算资源分配来提升系统性能。由于 MEC 的分布式部署和计算任务的海量卸载，MEC 中计算卸载技术解决干扰问题的方式不同于传统网络。因此，合理的资源分配方案将会成为解决干扰问题的技术选择。

5.4 总结

边缘计算在移动网络边缘提供计算、存储和网络资源，可以极大地降低处理时延，终端在卸载计算任务的同时也满足了绿色通信的要求，提升了服务质量。然而，由于终端卸载任务后可能会发生移动，为满足 MEC 处理卸载的计算任务时保持业务的连续性，解决移动性管理问题成为移动边缘计算中计算卸载技术的重要挑战之一。此外，由于 MEC 的分布式部署环境，使原本适用于云计算的安全管理机制已经不再适用于 MEC，因此为了保持安全通信需要从各个层级解决安全问题，比如确保边缘节点安全、网络安全、数据安全、应用安全、安全态势感知、安全管理编排和身份认证感知等。同时，由于大规模终端计算任务的卸载，使干扰问题不可避免，这就需要通过有效的干扰管理机制来解决，目前，学术界有很多关于干扰管理的方案，主要涉及计算和网络资源的联合分配问题，合理的资源分配能够控制干扰问题的产生，实现高效通信。在实现 MEC 大规模应用之前，移动边缘计算以及各种边缘计算卸载技术的解决方案在移动性管理、安全、干扰管理、QoE 保障等方面仍需要进一步研究和发展。

6 结束语

近年来，MEC 研究受到国内外的广泛关注，已成为移动网络研究的重点内容。作为 MEC 关键技术之一，计算卸载解决了移动设备在资源存储、计算性能以及能效等方面存在的不足。本文重点对 MEC 的网络架构和部署方案，以及 MEC 中计算卸载研究进展进行了分析和总结。通过对不同 MEC 卸载方案的分析和对比，提出了基于 5G 的 MEC 卸载方案，并对 MEC 计算卸载面临的问题和挑战进行了总结归纳。通过综述该领域的已有研究成果，探讨分析研究目标和方法，总结研究思路，从而为相关领域的研究人员提供参考和帮助。

参考文献：

[1] DINH H T, LEE C, NIYATO D, et al. A survey of mobile cloud computing: architecture, applications and approaches[J]. Wireless Communications & Mobile Computing, 2013, 13(18):1587-1611.

[2] KHAN A U R, OTHMAN M, MADANI S A, et al. A survey of mobile cloud computing application models[J]. IEEE Communications Surveys & Tutorials, 2014, 16(1):393-413.

[3] WANG Y, CHEN I R, WANG D C. A survey of mobile cloud computing Applications: perspectives and challenges[J]. Wireless Personal Communications, 2015, 80(4):1607-1623.

[4] MACH P, BECVAR Z. Mobile edge computing: a survey on architecture and computation offloading[J]. IEEE Communications Surveys & Tutorials, 2017, PP(99):1-1.

[5] FLORES H, HUI P, TARKOMA S, et al. Mobile code offloading: from concept to practice and beyond[J]. IEEE Communications Magazine, 2015, 53(3):80-88.

[6] JIAO L, FRIEDMAN R, FU X, et al. Cloud-based computation offloading for mobile devices: State of the art, challenges and opportunities[C]// Future Network and Mobile Summit. 2013:1-11.

[7] WANG S, ZHANG X, ZHANG Y, et al. A survey on mobile edge networks: convergence of computing, caching and communications[J]. IEEE Access, 2017, PP(99):1-1.

[8] MAO Y, YOU C, ZHANG J, et al. A survey on mobile edge computing: the communication perspective[J]. IEEE Communications Surveys & Tutorials, 2017, PP(99):1-1.

[9] TALEB T, SAMDANIS K, MADA B, et al. On multi-access edge computing: a survey of the emerging 5G network edge architecture & orchestration[J]. IEEE Communications Surveys & Tutorials, 2017, PP(99): 1.

[10] YU Y. Mobile edge computing towards 5G: vision, recent progress, and open challenges[J]. China Communication, 2016, 13(S2):89-99.

[11] AHMED A, AHMED E, A survey on mobile edge computing[C]// 2016 10th International Conference on Intelligent Systems and Control (ISCO). 2016: 1-8.

[12] 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.

[13] WIKIPEDIA C. Mobile edge computing[J]. Wikipedia, The Free Encyclopedia. 2017

[14] CHOCHLIOUROS I P, GIANNOULAKIS I, KOURTIS T, et al. A model for an innovative 5G- oriented, architecture, based on small cells coordination for multi-tenancy and edge services[C]// IFIP International Conference on Artificial Intelligence Applications and Innovations. 2016: 666-675.

[15] GIANNOULAKIS I, KAFETZAKIS E, TRAJKOVSKA I, et al. The emergence of operator – neutral small cells as a strong case for cloud computing at the mobile edge[J]. Transactions on Emerging Telecommunications Technologies, 2016, 27(9):1152-1159.

[16] LOBILLO F, BECVAR Z, PUENTE M A, et al. An architecture for mobile computation offloading on cloud-enabled LTE small cells[C]// Wireless Communications and NETWORKING Conference Workshops. 2014:1-6.

[17] PUENTE M A, BECVAR Z, ROHLIK M, et al. A seamless integration of computationally-enhanced base stations into mobile networks towards 5G[C]// Vehicular Technology Conference. 2015:1-5.

[18] BECVAR Z, ROHLIK P, VONDRA M, et al. Distributed architecture of 5G mobile networks for efficient computation management in mobile edge computing[J]. Chapter in 5G Radio Access Network (RAN)—Centralized RAN, Cloud-RAN and Virtualization of Small Cells, 2017

[19] WANG S, TU G H, GANTI R, et al. Mobile micro-cloud: Application classification, mapping, and deployment[C]//Proc Annu Fall Meeting ITA (AMITA). 2013: 1-7.

[20] WANG K, SHEN M, CHO J, et al. MobiScud: a fast moving personal cloud in the mobile network[C]// The Workshop on All Things Cellular: Operations, Applications and Challenges. 2015:19-24.

[21] TALEB T, KSENTINI A, FRANGOUDIS P. Follow-me cloud: when cloud services follow mobile users[J]. IEEE Transactions on Cloud Computing, 2016, PP(99):1-1.

[22] AISSIOUI A, KSENTINI A, GUEROUI A. An efficient elastic distributed SDN controller for follow-me cloud[C]//International Conference on Wireless and Mobile Computing, Networking and Communications. 2015:876-881.

[23] LIU J, ZHAO T, ZHOU S, et al. CONCERT: a cloud-based architecture for next-generation cellular systems[J]. IEEE Wireless Communications, 2014, 21(6):14-22.

[24] LIU J, MAO Y, ZHANG J, et al. Delay-optimal computation task scheduling for mobile-edge computing Systems[C]//IEEE International Symposium on Information Theory. 2016:1451-1455.

[25] MAO Y, ZHANG J, LETAIEF K B. Dynamic computation offloading for mobile-edge computing with energy harvesting devices[J]. IEEE Journal on Selected Areas in Communications, 2016, 34(12): 3590-3605.

[26] ZHANG K, MAO Y, LENG S, et al. Optimal delay constrained offloading for vehicular edge computing networks[C]// IEEE International Conference on Communications. 2017:1-6.

[27] JIA M, CAO J, YANG L. Heuristic offloading of concurrent tasks for computation-intensive applications in mobile cloud computing[C]// Computer Communications Workshops. 2014:352-357.

[28] KAO Y H, KRISHNAMACHARI B, RA M R, et al. Hermes: latency optimal task assignment for resource-constrained mobile computing[C]//IEEE Conference on Computer Communications. 2015: 1894-1902.

[29] KAMOUN M, LABIDI W, SARKISS M. Joint resource allocation and offloading strategies in cloud enabled cellular networks[C]// IEEE International Conference on Communications. 2015:5529-5534.

[30] LABIDI W, SARKISS M, KAMOUN M. Energy-optimal resource scheduling and computation offloading in small cell networks[C]// International Conference on Telecommunications. 2015:313-318.

[31] ZHANG H, GUO J, YANG L, et al. Computation offloading consid-

ering fronthaul and backhaul in small-cell networks integrated with MEC[C]//2017 IEEE Conference on Computer Communications Workshops. 2017: 115-120.

[32] YOU C, HUANG K. Multiuser resource allocation for mobile-edge computation offloading[C]//Global Communications Conference.2017:1-6.

[33] YOU C, HUANG K, CHAE H, et al. Energy-efficient resource allocation for mobile-edge computation offloading[J]. IEEE Transactions on Wireless Communications, 2017, 16(3):1397-1411.

[34] MUÑOZ O, PASCUAL-ISERTE A, VIDAL J. Optimization of radio and computational resources for energy efficiency in latency-constrained appli-cation offloading[J]. IEEE Transactions on Vehicular Technology, 2015, 64(10):4738-4755.

[35] NAN Y, LI W, BAO W, et al. Adaptive energy-aware computation offloading for cloud of things systems[C]//IEEE Access, 2017(5): 23947-23957.

[36] WANG W, ZHOU W. Computational offloading with delay and capacity constraints in mobile edge[C]// IEEE International Conference on Communications. 2017:1-6.

[37] LIU L Q, CHANG Z, GUO X J, et al. Multi-objective optimization for computation offloading in mobile-edge computing[C]//2017 IEEE Symposium on Computers and Communications (ISCC). 2017: 832-837.

[38] VALERIO V D, LO P F. Optimal virtual machines allocation in mobile femto-cloud computing: an MDP approach[C]// Wireless Communications and NEtworking Conference Workshops. 2014:7-11.

[39] ZHAO T, ZHOU S, GUO X, et al. A cooperative scheduling scheme of local cloud and internet cloud for delay-aware mobile cloud computing[C]// IEEE GLOBECOM Workshops. 2015:1-6.

[40] GUO X, SINGH R, ZHAO T, et al. An index based task assignment policy for achieving optimal power-delay tradeoff in edge cloud systems[C]//IEEE International Conference on Communications. 2016: 1-7.

[41] OUEIS J, CALVANESE S E, DE D A, et al. On the impact of backhaul network on distributed cloud computing[C]// Wireless Communications and Networking Conference Workshops. 2014:12-17.

[42] OUEIS J, STRINATI E C, BARBAROSSA S. Small cell clustering for efficient distributed cloud computing[C]// International Symposium on Personal, Indoor, and Mobile Radio Communication. 2015: 1474-1479.

[43] NDIKUMANA A, ULLAH S, LEANH T, et al. Collaborative cache allocation and computation offloading in mobile edge computing[C]//2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS). 2017: 366-369.

[44] XU J, CHEN L, REN S. Online learning for offloading and autoscaling in energy harvesting mobile edge computing[J]. IEEE Transactions on Cognitive Communications & Networking, 2017, PP(99):1-1.

[45] KETYKÓ I, KECSKÉS L, NEMES C, et al. Multi-user computation offloading as Multiple Knapsack Problem for 5G Mobile Edge Computing[C]// European Conference on Networks and Communications. 2016: 225-229.

[46] WANG C, YU F R, LIANG C, et al. Joint computation offloading and Interference management in wireless cellular networks with mobile edge computing[J]. IEEE Transactions on Vehicular Technology, 2017, PP(99):1-1.

[47] CUERVO E, BALASUBRAMANIAN A, CHO D K, et al. MAUI:making smartphones last longer with code offload[C]// International Conference on Mobile Systems, Applications and Services. 2010:49-62.

[48] KOSTA S, AUCINAS A, HUI P, et al. ThinkAir: dynamic resource allocation and parallel execution in the cloud for mobile code offloading[C]// INFOCOM, 2012 Proceedings IEEE. 2012:945-953.

[49] GORDON M S, JAMSHIDI D A, MAHLKE S, et al. COMET: code offload by Migrating Execution Transparently[C]// Usenix Conference on Operating Systems Design and Implementation. 2012:93-106.

[50] PANG Z, SUN L, WANG Z, et al. A survey of cloudlet based mobile computing[C]// International Conference on Cloud Computing and Big Data. 2016: 268-275.

[51] CHUN B G, MANIATIS P. Augmented smartphone applications through clone cloud execution[C]// Conference on Hot Topics in Operating Systems. 2009:8.

[52] GORDON M S, HONG D K, CHEN P M, et al. Accelerating mobile applications through flip-flop replication[C]// International Conference on Mobile Systems, Applications and Services. 2015:137-150.

[53] 张平, 陶运铮, 张治. 5G 若干关键技术评述[J]. 通信学报, 2016, 37(7):15-29.
ZHANG P, TAO Y Z, ZHANG Z, Survey of several key technologies for 5G[J], Journal on Communications, 2016,37(7):15-29.

[54] 黄韬, 刘江, 霍如,等. 未来网络体系架构研究综述[J]. 通信学报, 2014, 35(8):184-197.
HUANG T, LIU J, HUO R, et al. Survey of research on future network architectures[J], Journal on Communications, 2014, 35(8):184-197

[55] 刘韵洁, 黄韬, 张娇,等. 服务定制网络[J]. 通信学报, 2014, 35(12):1-9.
LIU Y J, HUANG T, ZHANG J, et al. Service customized networking[J]. Journal on Communications, 2014, 35(12):1-9.

[56] MACH P, BECVAR Z. Cloud-aware power control for cloud-enabled small cells[C]// GLOBECOM Workshops. 2015:1038-1043.

[57] MACH P, BECVAR Z. Cloud‐aware power control for real‐time application offloading in mobile edge computing[J]. Transactions on Emerging Tele-communications Technologies, 2016, 27(5):648-661.

[58] WANG S, URGAONKAR R, HE T, et al. Mobility-induced service migration in mobile micro-clouds[C]// Military Communications Conference. 2014:835-840.

[59] WANG S, URGAONKAR R, ZAFER M, et al. Dynamic service migration in mobile edge-clouds[C]//IFIP Networking Conference. 2015: 1-9.

[60] NADEMBEGA A, HAFID A S, BRISEBOIS R. Mobility prediction model-based service migration procedure for follow me cloud to support QoS and QoE[C]// IEEE International Conference on Communications. 2016:1-6.

[61] WANG S, URGAONKAR R, HE T, et al. Dynamic service placement for mobile micro-clouds with predicted future costs[J]. IEEE Transactions on Parallel & Distributed Systems, 2017, 28(4):1002-1016.

[62] SHIBIN D, KATHRINE G J W. A comprehensive overview on secure offloading in mobile cloud computing[C]//2017 4th International Conference on Electronics and Communication Systems (ICECS), 2017: 121-124.

[63] 边缘计算产业联盟, 工业互联网产业联盟. 边缘计算参考架构 2.0 [R]. 北京: 工业互联网产业联盟，2017.
ECC, AII. The Architecture of Edge Computing 2.0 [R]. Beijing: AII, 2017.

[64] YANG W, FUNG C. A survey on security in network functions virtualization[C]//Netsoft Conference and Workshops. 2016:15-19.

[65] LI C, ZHANG J, HAENGGI M, et al. User-centric intercell interference nulling for downlink small cell networks[J]. IEEE Transactions on Communications, 2014, 63(4):1419-1431.

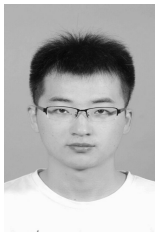
[66] HUANG G, LI J. Interference mitigation for femtocell networks via adaptive frequency reuse[J]. IEEE Transactions on Vehicular Technology, 2016, 65(4):2413-2423.

[67] BU S, YU F R, SENARATH G. Interference-aware energy-efficient resource allocation for heterogeneous networks with incomplete channel state in-formation[C]//IEEE International Conference on Communications. 2013: 6081-6085.

[作者简介]

谢人超（1984–），男，福建南平人，博士，北京邮电大学副教授、硕士生导师，主要研究方向为信息中心网络、移动网络内容分发技术和移动边缘计算等。

廉晓飞（1992–），女，天津人，北京邮电大学硕士生，主要研究方向为 5G 网络、移动边缘计算等。



贾庆民（1990–），男，山东泰安人，北京邮电大学博士生，主要研究方向为新型网络体系架构、内容分发和移动边缘计算等。

黄韬（1980–），男，重庆人，博士，北京邮电大学教授、博士生导师，主要研究方向为新型网络体系架构、内容分发网络、软件定义网络等。

刘韵洁（1943–），男，山东烟台人，中国工程院院士，北京邮电大学教授、博士生导师，主要研究方向为未来网络体系架构。