



移动边缘计算时延与能耗联合优化方法

张先超^{1,2*}, 任天时³, 赵耀³, 樊锐⁴

(1. 东南大学移动通信国家重点实验室 南京 210096; 2. 嘉兴学院浙江省医学电子与数字健康重点实验室 浙江 嘉兴 314001;
3. 北京理工大学信息与电子学院 北京 海淀区 100081; 4. 中国电子科学研究院 北京 石景山区 100041)

【摘要】针对移动边缘计算中时延与能耗是关键性能指标,且相互制约的问题,研究了通过在边缘与终端之间进行任务分配,对时延与能耗进行联合优化。首先,建立了能耗与时延联合优化的 0-1 整数规划模型;其次,设计了对任务进行分配的分支定界算法。仿真结果表明,该方法能够有效降低移动边缘计算能耗与时延。

关键词 分支定界方法; 能耗; 联合优化; 移动边缘计算; 时延
中图分类号 TN92 文献标志码 A doi:10.12178/1001-0548.2021244

Joint Optimization Method of Energy Consumption and Time Delay for Mobile Edge Computing

ZHANG Xianchao^{1,2*}, REN Tianshi³, ZHAO Yao³, and FAN Rui⁴

(1. National Mobile Communications Research Laboratory, Southeast University Nanjing 210096;
2. Key Laboratory of Medical Electronics and Digital Health of Zhejiang Province, Jiaying University Jiaying Zhejiang 314001;
3. School of Information and Electronics, Beijing Institute of Technology Haidian Beijing 100081;
4. China Academic of Electronics and Information Technology Shijingshan Beijing 100041)

Abstract In mobile edge computing, time delay and power consumption are key performance indicators, and they are mutually restricted. This paper studies the joint optimization of time delay and power consumption by distributing tasks between edge and terminal. Firstly, the 0-1 integer programming model for joint optimization of energy consumption and time delay is established on the basis of describing the research problem. Secondly, a branch and bound algorithm for task assignment is designed. Finally, the simulation show that the proposed method can effectively reduce the energy consumption and time delay of mobile edge computing.

Key words branch-and-bound method; energy consumption; joint optimization; mobile edge computing; time delay

移动互联网的广泛应用,使得用户对数据速率和服务质量 (quality of service, QoS) 的需求呈指数增长。尽管移动终端设备的中央处理器性能不断提升,但仍无法应对处理任务的急剧增长,移动终端设备计算能力仍显不足。并且,移动终端设备的计算需要大量能耗,降低了设备的续航时间。这些问题推动了移动云计算概念的出现和发展^[1]。

近年来物联网技术的发展与普及,导致移动云计算的缺点逐渐暴露。首先,云计算无法满足新兴应用场景对更低网络时延的需求^[2];其次,接入设备数量的急剧增加将导致网络数据传输量呈爆炸性

增长趋势,采用云计算汇聚的网络流量会使核心网不堪重负^[3]。为此,欧洲电信标准化协会 (European telecommunications standards institute, ETSI) 于 2014 年提出了移动边缘计算 (mobile edge computing, MEC) 的概念,给出定义^[4]:在无线接入网 (radio access network, RAN) 内和靠近移动用户的移动网络边缘,MEC 能够提供 IT 服务环境和云计算的能力。2017 年,ETSI 将 MEC 的概念延伸至 Wi-Fi、车联网等接入网络,并将其更名为多接入边缘计算 (multi-access edge computing),简写仍为 MEC^[5]。

移动边缘计算由移动边缘服务器、基站、终端

收稿日期:2021-09-06;修回日期:2022-03-15

基金项目:国家自然科学基金重点项目 (61941104, U19B2015)

作者简介:张先超 (1984-),男,博士,教授,主要从事无线网络、人工智能等方面的研究。

*通信作者:张先超, E-mail: xianchao.zhang@buaa.edu.cn

用户、核心网、云等构成,其中,移动边缘服务器部署在基站附近,为该基站小区内的用户提供计算资源。通过直接向移动边缘服务器寻求服务,用户可以享受低时延、高能效的体验。当移动边缘服务器的计算能力不足以支撑用户需求时,位于核心网的云计算(mobile cloud computing, MCC)才会进一步服务于用户的计算^[6-7]。相比于 MCC, MEC 有更低的时延、更低的移动设备能耗^[8],更好的上下文感知能力^[9-11]和更强的隐私性与安全性^[12]。MEC 技术能够有效应用的关键是计算任务在边缘服务器与终端之间有效分配^[13]。计算任务分配就是将任务分配给合适的任务处理设备,包括本地处理器、临近的 IoT 设备处理器、MEC 服务器、云服务器等,同时,明确应用中任务的依赖关系,给出任务处理的先后顺序。

5G 的商用投入和 6G 高速发展增强了医疗急救、灾害救援等应急场景的处置能力。这些应急场景对现场信息处理有更高的要求。然而,用户端的计算能力通常难以满足,如移动急救车辆上的计算能力无法满足车辆上信息处理的需要,采用边缘计算是事宜的解决方案。

在应急情景下,单个小区中需要对多达数十个用户的计算任务进行合理有效的分配,以满足其低时延、低能耗的应用需求。文献 [14] 研究了带有能量收集设备的绿色 MEC 系统,并给出了基于李亚普诺夫优化的动态计算任务分配策略,以最小化执行延迟和任务失败概率为目标,能够接近最优来解决任务分配问题。文献 [15] 将最优分配表征为在计算延迟约束下,最小化加权能耗和的凸优化问题,证明了对于分配优先级函数,最优策略具有阈值特征,优先级高于和低于给定阈值的用户将分别执行完整分配和最小值分配。文献 [16] 采用了博弈论的方法来实现有效的分布式计算任务分配,将移动设备用户之间的分布式计算卸载决策问题表述为一个多用户计算卸载博弈,设计了一种可以实现纳什均衡的分布式计算任务分配算法。文献 [17] 设计了一种移动边缘环境下对多用户时延与移动边缘计算服务器资源分配均衡性进行联合优化的计算卸载方法,构造了联合优化平均卸载时延与资源分配均衡度的目标函数,从而有效地减小多用户的平均卸载时延,同时平衡各移动边缘计算服务器的工作负荷。

本文针对 5G 场景下超可靠低时延通信(ultra-reliable and low-latency communications, uRLLC) 典

型应用场景中单小区多用户的移动边缘计算问题,研究在边缘服务器与移动用户终端之间进行计算任务分配,实现时延和能耗联合优化。

1 问题描述

设有一个包含 gNB 节点和 N 个终端用户设备的无线接入网络, gNB 节点上部署着 MEC 服务器和 MEC 任务管理器,如图 1 所示。gNB 节点可以控制通信流量, MEC 服务器负责对计算任务的处理, MEC 任务管理器模块与 MEC 服务器在相同位置部署,负责 MEC 系统中计算任务分配等。用 $Z = \{0, 1, \dots, n, \dots, N\}$ 表示 N 个终端用户设备。每个终端用户设备都要完成大量计算任务,任务不能被分割,全部在终端 CPU 处理,或全部通过无线信道将任务分配到 MEC 服务器处理。MEC 任务管理器获取终端用户的状态、需要分配的任务以及 MEC 服务器的可用资源,通过考虑能耗与时延等要求,确定所有终端用户的任务分配策略,并将任务分配策略反馈给终端用户设备和基站。

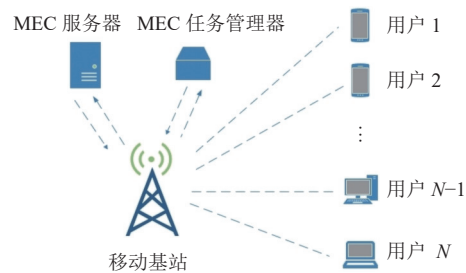


图 1 多用户移动边缘计算系统构成

第 n ($1 \leq n \leq N$) 个终端用户的计算任务,表征为 $A_n \sim (L_n, \tau_n^d, X_n)$, 其中, L_n 表示任务数据大小(单位: bit), τ_n^d 表示第 n 个终端用户的最大要求时延(单位: s), F 是终端用户的总功率上限, X_n 表示单位数据计算量(以 CPU 周期/bit 为单位), X_n 的与 L_n 呈正相关^[18-19]。设任务 A_n 在终端用户上处理,需要的时间和能耗分别是 t_n^l 和 E_n^l , 综合代价为 $C_n^l = g_n^l(t_n^l, E_n^l)$, 在 MEC 服务器上处理,需要的时间和能耗分别是 t_n^o 和 E_n^o , 综合代价为 $C_n^o = g_n^o(t_n^o, E_n^o)$ 。用 $\alpha_n \in \{0, 1\}$ 表征 A_n 在终端用户或 MEC 服务器上处理,当 $\alpha_n = 1$ 时, A_n 在 MEC 服务器上处理,当 $\alpha_n = 0$ 时, A_n 在用户终端上处理。 $\alpha_1, \alpha_2, \dots, \alpha_n, \dots, \alpha_N$ 构成的向量表征为 $Z = \{\alpha_1, \alpha_2, \dots, \alpha_n, \dots, \alpha_N\}$ 。那么, N 个终端用户的计算总代价为:

$$C_{\text{all}} = \sum_{n=1}^N [(1 - \alpha_n) C_n^l + \alpha_n C_n^o] \quad (1)$$

通过求解 \mathbf{Z} , 使得该 MEC 系统的 C_{all} 最小。

2 联合优化模型

若任务 A_n 选择在用户终端处理, 终端设备的计算时延与能耗为:

$$t_n^l = \frac{L_n X_n}{f_n} \quad (2)$$

$$E_n^l = \kappa L_n X_n f_n^2 \quad (3)$$

式中, f_n 表示用户设备 n 的 CPU 频率 (以 CPU 周期 s 为单位); κf_n^2 表示 CPU 一个周期的能耗。

用时延和能耗的加权和, 表征任务 A_n 在用户终端处理所需的代价:

$$C_n^l = g_n^l(t_n^l, E_n^l) = I_n^l t_n^l + I_n^e E_n^l \quad (4)$$

式中, I_n^l 是任务 A_n 的时延成本的权重; I_n^e 是能耗成本的权重, $0 \leq I_n^l \leq 1$, $0 \leq I_n^e \leq 1$ 且 $I_n^l + I_n^e = 1$ 。 I_n^l 与 I_n^e 对于不同类型的任务可能是不同的, 为简单起见, 认为每个任务的权重在整个计算过程中保持不变。

若任务 A_n 分配至 MEC 服务器进行处理, 需要将任务 A_n 从用户终端传输到 MEC 服务器进行处理, 处理结果由 MEC 服务器传输回本地。

由于处理结果传回用户设备的数据量通常较小, 且 MEC 服务器的传输功率很大, 因此可以忽略处理结果由 MEC 服务器传输回本地的时延, 则任务 A_n 由用户终端传输至 MEC 服务器与在 MEC 服务器处理的时延以及移动设备能耗分别为:

$$t_n^o = t_n^{\text{ul}} + t_n^s = \frac{L_n}{R^{\text{ul}}} + \frac{L_n X_n}{f_s} \quad (5)$$

$$E_n^o = P^{\text{ul}} t_n^{\text{ul}} + P^i t_n^s \quad (6)$$

式中, R^{ul} 是数据从用户终端传输至 MEC 服务器的速率; f_s 表示为该用户分配的虚拟 CPU 频率 (以 CPU 周期/ s 为单位); P^{ul} 指移动设备传输数据的功率; P^i 指移动设备空闲时的功率。

考虑只有一个移动基站的系统, 忽略其他小区的通信干扰, 那么, 用户设备的上传数据速率为:

$$R^{\text{ul}} = W \lg \left(1 + \frac{P_n h_n}{W N_0} \right) \quad (7)$$

式中, W 是无线信道的带宽; P_n 是无线信道中第 n 个用户设备的传输功率; h_n 是无线信道中第 n 个用户设备的信道增益; N_0 为噪声功率。

同样, 用时延和能耗的加权和, 表征任务分配

至 MEC 服务器处理所需代价:

$$C_n^o = g_n^o(t_n^o, E_n^o) = I_n^l t_n^o + I_n^e E_n^o \quad (8)$$

式中, I_n^l 指任务 A_n 的时延成本的权重; I_n^e 是能耗成本的权重, $0 \leq I_n^l \leq 1$, $0 \leq I_n^e \leq 1$, $I_n^l + I_n^e = 1$ 。同样, 权重在任务处理过程中保持不变。

因此, 综合本地计算模型和 MEC 服务器的计算模型, N 个任务分配所需代价的和可由式(1)得到。

根据上述分析, 建立问题的 0-1 整数规划模型:

$$C_{\text{all}} = \min \sum_{n=1}^N [(1 - \alpha_n) C_n^l + \alpha_n C_n^o] \quad (9a)$$

$$\text{s.t.} \quad (1 - \alpha_n) T_n^l + \alpha_n T_n^o \leq \tau_n^d \quad \forall n \in N \quad (9b)$$

$$\sum_{n=1}^N \alpha_n f_n \leq F \quad \forall n \in N \quad (9c)$$

$$\alpha_n \in \{0, 1\} \quad \forall n \in N \quad (9d)$$

式(9a)是优化的目标, 使得终端用户的时延和能耗的代价和最小; 式(9b)是终端用户时延约束; 式(9c)是终端用户的功率约束; 式(9d)中 α_n 是求解变量, 表示任务 A_n 在终端用户或在 MEC 服务器处理。

3 分支定界算法

分支定界算法是求解整数规划问题的常用算法, 分支把区域逐次分割成越来越多的小区域, 定界在这些小区域内, 确定目标函数的上界和下界^[20]。

针对模型式(9), 设计如下分支定界算法。

算法 1: 时延与能耗联合优化分支定界算法

1) 初始化全局参数, 全局上界 $U = \infty$, 全局下界 $L = -\infty$, 全局最优值 $C^* \leftarrow \emptyset$, 问题最优解 $Z^* \leftarrow \emptyset$, 初始化松弛线性规划问题队列 Q

2) 初始化 N 个用户的参数, 计算用户本地计算时延 T_n^l , 本地计算能耗 E_n^l , MEC 服务器计算时延 T_n^o , MEC 服务器计算能耗 E_n^o

3) 计算初始整数规划问题求解所需参数

4) 取第一个用户的任务分配决策 $\alpha_1 = 0$ 和 $\alpha_1 = 1$, 分作两个问题

5) 将两个问题分别松弛求解, 得到解 Z_1, Z_2 和目标函数值 $C = \min(Z_1, Z_2)$

6) 全局下界更新 $L \leftarrow C$

7) 若两问题均无解, 则算法失败, 停机

8) 若解 $\alpha_n \in \{0, 1\}$, $n = 1, \dots, N$, 则 $Z^* \leftarrow Z$, 算法结束, 停机

9) 否则, 将解、目标函数值与约束参数放入队列 Q

10) while Q 不为空 do

11) 以广度搜索取出当前问题

12) if $C > U$ do

13) continue

14) if 解 $\alpha_n \in \{0, 1\}, n = 1, \dots, N$ do

15) if $U > C$ do

16) $U \leftarrow C$

17) end if

18) if $C^* > C$ do

19) $C^* \leftarrow C, Z^* \leftarrow Z$

20) end if

21) else

22) 寻找解中第一个不为 0 或 1 的分量, 取其下标 idx , 带入其已确定的节点值, 构建两个新的线性规划问题 r_1 和 r_2

23) if r_1 计算成功 and r_2 计算失败

24) 将 r_2 的解与约束参数放入队列 Q

25) elif r_2 计算成功 and r_1 计算失败

26) 将 r_1 的解与约束参数放入队列 Q

27) elif r_2 计算成功 and r_1 计算成功

28) 首先将目标函数值 C 小的问题加入队列 Q

29) end if

30) end if

31) end while

区别于随机算法, 该算法按照广度优先的方式对状态空间树进行搜索, 通过不断地剪枝操作寻找最优解的子节点, 最终获得模型式 (9) 的确定性解。具体来说, 该算法在获取所有用户参数之后, 将 0-1 整数规划问题的解松弛为 $[0,1]$ 之间的连续变量, 根据松弛线性规划问题的解是否为整数, 来判断下一步继续分支还是停止计算。该算法通过不断地松弛求解, 得到原问题的最优解, 且其在数十个用户的小规模问题下可以适用。

本文算法的复杂度主要取决于对松弛线性规划问题的求解。在使用分支定界法的过程中, 每进行一次分支, 就会产生两个松弛线性规划问题, 即最多会产生 2^N 个松弛线性规划问题, 因此在最差的情况下, 本文算法的复杂度为 $O(2^N)$ 。在进行分支定界法的剪枝操作后, 当用户数为 150 个时, 计算时间可以控制在 0.1 s, 因此在几十个用户规模的问题下该算法可以适用。

具体来说, 算法的 1)~3) 行为初始化全局参

数, 将上界和下界均设为极值, 将最优解与最优目标函数值设为空, 再使用环境模型计算出分支定界算法所需要的参数; 算法的 4)~8) 行进行初次的分支, 将第一个用户的决策分为两支后对分支后的两个问题松弛计算。判断该线性规划问题的解, 若解均为 0 或 1, 则算法结束, 找到最优目标函数值; 若无解, 则算法失败; 否则开始分支。算法的 8)~29) 行通过不断将不能求解的子问题和目标函数值已经大于问题上界的节点剪枝, 同时对该问题下没有剪枝的叶子节点分支, 直到得到解均为整数 0 或 1, 算法结束。

4 仿真分析

设一个带宽为 $W = 50$ MHz 的 gNB 小区, gNB 基站部署有 MEC 服务器, 用户终端设备随机分布在距离 gNB 基站 200 m 的范围内。MEC 服务器的计算容量为 $F = 10$ GHz, 每个用户设备的 CPU 频率为 $f_n^i = 2$ GHz, 用户设备的传输功率和空闲功率设置为 $P_n = 500$ mW 和 $P_n^i = 100$ mW。需要计算分配的数据 B_n (单位: Bytes) 满足 (300, 500) 之间的均匀分布, CPU 周期 D_n (单位: Megacycles) 满足 (900, 1 000) 之间的均匀分布, 时延要求为 2 s。每个用户对时延和能耗分配的权重被设置为 $I_n^t = I_n^e = 0.5$ 。平均信道增益 h_n 遵循自由空间路径损耗模型:

$$h_i = A_d \left(\frac{3 \times 10^8}{4\pi f_c d_i} \right)^{d_e} \quad (10)$$

式中, $A_d = 4.11$ 表示天线增益; $f_c = 915$ MHz 表示载波频率; $d_e = 2.8$ 表示路径损耗指数。

根据式 (7), 可以计算得到用户设备的数据上传速率 R^u 。设置用户设备数分别为 4、6、8、10、12、14、16 个, 根据分支定界法计算该模型的最优解 $Z = \{\alpha_1, \alpha_2, \dots, \alpha_n, \dots, \alpha_N\}$, 如表 1 所示。

表 1 不同用户设备数下的最优解

用户数	最优解
4	[1 1 1 1]
6	[1 1 0 1 1 1]
8	[0 1 1 1 0 1 1 0]
10	[0 1 0 0 1 1 1 0 1 0]
12	[0 0 1 1 1 0 0 0 1 0 1 0]
14	[0 0 1 0 0 1 0 0 0 0 1 1 0 1]
16	[0 0 0 0 0 1 1 0 1 0 0 0 1 0 0 1]

图 2 是不同用户数的代价曲线。从图中可以看出, 本文优化方法的代价小于全部在用户终端处理

或全部在 MEC 服务器处理, 且随着用户数的增加, 本文的优化方法优势更加明显。同时, 本文将实验结果与能有效地帮助移动设备实现 MEC 系统的节能运行的最大节能优先算法^[21](select maximum saved energy first, SMSEF) 进行对比, 本文算法利用贪婪选择解决优化问题, 为 MEC 计算资源分配等问题提供解决方案, 相较于传统方法节能效果更好, 在用户数量增大时, 本文方法的优势也逐渐增大。

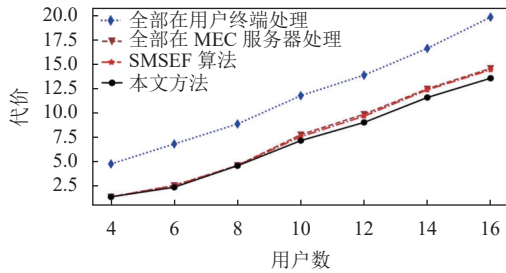


图 2 代价与用户设备数的关系

固定用户设备数目为 8 个, 改变 MEC 服务器的计算容量为 6、8、10、12、14 GHz, 代价曲线如图 3 所示。由于用户终端处理不需要 MEC 服务器, 所以随着 MEC 服务器计算容量地增加, 用户终端所需代价几乎不变。当 MEC 服务器计算容量足够大时, 全部在 MEC 服务器处理和本文的优化方法得到的结果相接近。同时, 相比于最大节能优先算法, 本文方法在 MEC 服务器计算容量较小时所需的代价更小。

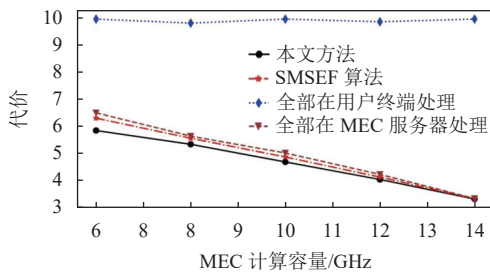


图 3 用户设备数为 8 时代价与 MEC 服务器计算容量关系

不失一般性, 改变用户设备的 CPU 频率 f_n^c 为 (1,3) 之间的均匀分布, 使得用户设备的 CPU 频率各不相同, 设置用户设备数目分别为 4、6、8、10、12、14、16 个, 其代价曲线如图 4 所示。可以看出, 随着用户数目的增大, 本文的优化方法相较于全部在用户终端处理、全部在 MEC 服务器处理和 SMSEF 算法, 能取得较低的代价, 且趋势与固定用户设备 CPU 频率时相同。

提升 MEC 服务器的计算容量为 100 GHz, 并

增加用户数量为 50、100、150、200、250、300 个, 如图 5 所示。将本文的优化方法的代价, 与全部在用户终端处理、全部迁移到 MEC 服务器处理、随机调度和循环调度进行比较 (随机调度指随机决定任务的分配方式, 循环调度指采用循环的方式将任务依次分配到用户设备或 MEC 服务器进行处理)。可以看出, 随着用户数量的增加, 本文方法相较于其他方法的优势更加显著。

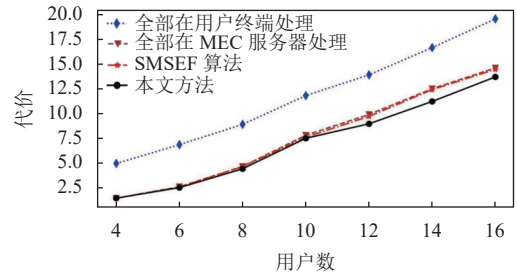


图 4 不同 CPU 频率下代价与用户设备数的关系

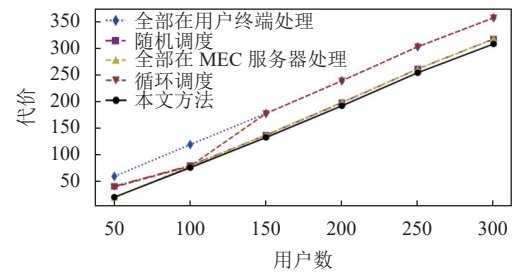


图 5 扩大 MEC 服务器容量时代价与用户设备数关系

随着用户规模的增大, 本文方法进行任务分配决策的时间也在增加, 如图 6 所示。当用户数小于 150 时, 计算时间可以控制在 0.1 s; 当用户数为 300 时, 计算时间在 0.4 s 以内。在不超过 300 个用户数的情况下, 决策时间可接受。随着用户数的大规模增加, 需要能够快速决策的智能优化方法来满足要求。

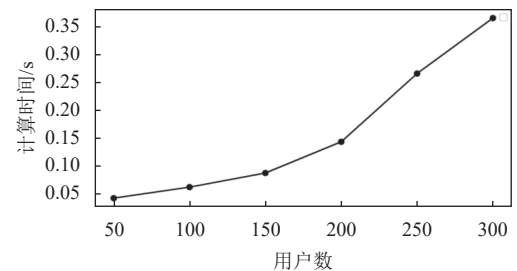


图 6 分支定界法计算时间与用户数的关系

5 结束语

本文针对移动边缘计算中低时延与低能耗两大

要求,研究了时延与能耗联合优化方法。通过对优化问题的研究,建立了0-1整数规划模型,采用分支定界算法对模型予以求解,通过仿真,验证了本文的优化方法的优越性和适用性。本文方法不仅能够和5G技术协同解决VR/AR应用的不足,还能够应用于联合作战、生命健康、智能制造等多个领域。为了适用超大规模用户终端的需求,未来还将研究智能优化方法,进一步提升移动边缘计算任务分配的效率与效果。

参 考 文 献

- [1] ZHANG Q, CHENG L, BOUTABA R. Cloud computing: State-of-the art and research challenges[J]. *Internet Services Application*, 2010, 1(1): 7-18.
- [2] LI A, YANG X, KANDULA S. Cloud Computing: Comparing public cloud providers[C]//The 10th ACM SIGCOMM Internet Measurement Conference (IMC) Melbourne: [s.n.], 2010: 1-14.
- [3] SHAFI M, MOLISCH A. F, SMITH P. J. 5G: A tutorial overview of standards, trials, challenges, deployment, and practice[J]. *IEEE Journal on Selected Areas in Communications*, 2017, 35(6): 1201-1221.
- [4] PATEL M, NAUGHTON B, CHAN C. Mobile-Edge computing—introductory technical white paper[R]. Sophia Antipolis: Mobile-Edge Computing Industry Initiative, 2014.
- [5] REZNIK A, ARORA R, CANNON M. Developing software for multi-access edge computing[R]. Sophia Antipolis: ETSI White Paper, 2017.
- [6] LEI L, ZHONG Z, ZHENG K, et al. Challenges on wireless heterogeneous networks for mobile cloud computing[J]. *IEEE Wireless Communication*, 2013, 20(3): 34-44.
- [7] BONOMI F, MILITO R, ZHU J, et al. Fog computing and its role in the Internet of Things[C]//Proceeding of MCC Workshop Mobile Cloud Computing. Helsinki: [s.n.], 2012: 13-16.
- [8] HRISHIKESH J, KANGWOO L, WOO S, et al. Powering the internet of things[C]//2014 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED). La Jolla, CA: IEEE, 2014: 375-380.
- [9] SCHILIT W N. A system architecture for context-aware mobile computing[D]. New York: Columbia University, 1995.
- [10] PERERA C, ZASLAVSKY A, CHRISTEN P. Context aware computing for the internet of things: A survey[J]. *IEEE Communications Surveys & Tutorials*, 2014, 16(1): 414-454.
- [11] NUNNA S, KOUSARIDAS A, IBRAHIM M. Enabling real-time context-aware collaboration through 5G and mobile edge computing[C]//IEEE International Conference Information Technology New Generation. (ITNG). Las Vegas: IEEE, 2015: 601-605.
- [12] SUO H, LIU Z, WAN J, et al. Security and privacy in mobile cloud computing[C]//Proceeding of IEEE International Wireless Communications and Mobile Computing Conference(IWCMC). Cagliari: IEEE, 2013: 655-659.
- [13] 黄永明, 郑冲, 张征明, 等. 大规模无线网络移动边缘计算和缓存研究[J]. *通信学报*, 2021, 42(4): 44-61.
HUANG Y M, ZHENG C, ZHANG Z M, et al. Research on mobile edge computing and caching in massive wireless communication network[J]. *Journal on Communications*, 2021, 42(4): 44-61.
- [14] MAO Y, ZHANG J. Dynamic computation offloading for mobile-edge computing with energy harvesting devices[J]. *IEEE Journal on Selected Areas in Communications*, 2016, 34(12): 3590-3605.
- [15] YOU C, HUANG K, CHAE H. Energy-Efficient resource allocation for mobile-edge computation offloading[J]. *IEEE Transactions on Wireless Communications*, 2017, 16(3): 1397-1411.
- [16] CHEN X, JIAO L, LI W Z, et al. Efficient multi-user computation offloading for mobile-edge cloud computing[J]. *IEEE-ACM Transactions on Networking*, 2016, 24(5): 2827-2840.
- [17] 张文柱, 曹琲琲, 余静华. 移动边缘计算中一种多用户计算卸载方法[J]. *西安电子科技大学学报*, 2020, 47(6): 131-138.
ZHANG W Z, CAO B B, YU J H. Multi-User computation offloading approach for mobile edge computing[J]. *Journal of Xidian University*, 2020, 47(6): 131-138.
- [18] MAO Y, YOU C, ZHANG J, et al. A Survey on mobile edge computing: The communication perspective[J]. *IEEE communications surveys & tutorials*, 2017, 19(4): 2322-2358.
- [19] LI J, GAO H, LV T, et al. Deep reinforcement learning based computation offloading and resource allocation for MEC[C]//IEEE Wireless Communications and Networking Conference. Barcelona: IEEE, 2018: 1-6.
- [20] 杨永健. 求全局最优化的几种确定性算法[D]. 上海: 上海大学, 2005.
YANG Y J. Some deterministic algorithms for global optimization[D]. Shanghai: Shanghai University, 2005.
- [21] WEI F, CHEN S, ZOU W. A greedy algorithm for task offloading in mobile edge computing system[C]//2018 IEEE 18th International Conference on Communication Technology (ICCT). [S.l.]: IEEE, 2018: 149-157.

编辑 税红