

# 基于边缘云和移动辅助设备的计算卸载优化方案

方加娟<sup>1,2</sup> 李凯<sup>1</sup>

<sup>1</sup>(郑州职业技术学院软件工程系 河南 郑州 450121)

<sup>2</sup>(南京理工大学计算机科学与工程学院 江苏 南京 210094)

**摘要** 针对移动边缘云和移动辅助设备联合计算卸载和资源分配的优化问题,提出一个联合计算卸载方案。计算用户设备(Computing User Equipment, CUE)通过为辅助用户设备(Helper User Equipment, HUE)提供带宽激励的方式,将计算卸载到边缘云和辅助用户设备,并通过优化本地、辅助用户设备和边缘云中的资源分配,实现对带宽和计算资源及时间的实际约束的最小加权能耗。为了解决非凸优化问题,采用几何规划(Geometric Programming, GP)和瓶颈匹配(Bottleneck Matching, BM)方法求近似凸问题的最优解。数值研究表明,与云卸载策略相比,在卸载延迟没有明显增加的情况下,该方案的总计算时间可以减少 35%~40%。

**关键词** 移动边缘计算 计算卸载 资源分配 优化

中图分类号 TP393 文献标志码 A DOI:10.3969/j.issn.1000-386x.2020.12.002

## OPTIMAL SCHEME OF COMPUTATION OFFLOADING BASED ON EDGE CLOUD AND MOBILE AUXILIARY EQUIPMENT

Fang Jiajuan<sup>1,2</sup> Li Kai<sup>1</sup>

<sup>1</sup>(Department of Software Engineering, Zhengzhou Technical College, Zhengzhou 450121, Henan, China)

<sup>2</sup>(College of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, Jiangsu, China)

**Abstract** To solve the optimization problem of joint computation offloading and resource allocation for mobile edge cloud and mobile auxiliary equipment, this paper proposes a joint computation offloading scheme. The computing user equipment (CUE) offloaded computing to the edge cloud and the auxiliary user equipment by providing bandwidth incentives for the helper user equipment (HUE). The method aimed to optimize the computation load assignments to local processors in the mobile users, mobile peers and the edge cloud jointly with the resource allocation to achieve the minimum weighted energy consumption subject to practical constraints on the bandwidth and computing resources and time. In order to solve the non-convex optimization problem, geometric programming (GP) and bottleneck matching (BM) method were used to solve the optimal solution of the approximate convex problem. Numerical studies show that the total computation time can be reduced by 35%~40% compared with the only cloud offloading strategy without a significant increase in the offload delay time.

**Keywords** Mobile edge computing Computation offloading Resource allocation Optimization

## 0 引言

近年来,全球移动设备的数量急剧增加,随着移动设备的普及,覆盖不同领域的新的应用程序如人脸识别、增强现实、自然语言处理和交互式游戏相继出现。

这些应用具有密集的计算需求。为了满足现实计算需求,移动边缘计算(Mobile Edge Computing, MEC)的概念被研究人员提了出来<sup>[1-2]</sup>。

移动边缘计算<sup>[3]</sup>是指将移动网络与互联网有效地结合,利用在移动网络边缘部署的计算、存储和处理功能,为用户提供高带宽和超低时延的网络服务解决方

案。计算卸载<sup>[4]</sup>作为 MEC 架构中的一项关键技术,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。基于 MEC 的计算卸载技术可以提高移动设备的性能,通过节省移动设备的能耗来延长设备的工作时间。

尽管计算卸载技术可以很好地提高移动设备端的能力,但仍存在诸多问题,计算卸载效率问题是其中之一。在具有大量计算密集型任务的 CUE 的延迟敏感系统中,当多个移动用户同步通过无线信道进行计算卸载时,设备之间势必会出现相互干扰,从而降低数据传输速率,导致数据传输时间的延长。因此,在计算资源有限的情况下,将所有 CUE 的任务卸载到云端并不总是明智的<sup>[5]</sup>。如今,随着移动设备的性能稳步提高,许多移动设备并没有充分利用它们的处理器,因此将任务卸载到这些临近闲置移动设备是一个诱人的选择。为了平衡功耗与服务响应时间,移动设备需要合理选择卸载至云端和移动终端处理的应用部分。柳兴等<sup>[6]</sup>通过遗传算法来搜索移动终端和云端计算资源联合执行应用的全局最优解,主要研究了迁移任务的选择问题。张文柱等<sup>[7]</sup>从移动终端硬件出发,在能耗方面进行了优化处理。曹宾等<sup>[8]</sup>提出一种利用临近闲置移动终端作为卸载目标的分布式博弈算法,给出一种最优结果。Ti 等<sup>[9]</sup>给出了一种在移动用户、移动对等点设备和云端之间进行联合计算卸载的非中心式博弈算法,通过多用户的设置制定了资源分配的优化方案,有效提高了计算卸载效率问题。

为了尽可能缩短 CUE 任务的完成时间,本文在文献<sup>[9]</sup>研究的基础上做进一步的探索,主要考虑以下几点:识别具有空闲计算资源的辅助用户设备 HUE;决策对等点和云端之间的计算卸载平衡;云计算资源的分配问题。由于 HUE 不需要消耗它们有限的能量来为其他人计算,引入了带宽激励的概念,探讨交换带宽计算活动的好处。

## 1 移动边缘计算

MEC 是由欧洲电信标准协会 ETSI 提出的基于 5G 演进架构的一项新技术。移动边缘计算借助边缘计算中心在无线接入网(RAN)内提供 IT 服务环境、云计算和存储功能,其目标是减少网络延迟,确保高效的网络运营和服务交付,并提高用户体验。

### 1.1 MEC 基本架构

移动边缘计算的关键要素是集成在 RAN 元素上的移动边缘计算 IT 应用服务器。MEC 的基本框架如

图 1 所示,该框架从一个比较宏观的层次出发,对 MEC 架构下不同的功能实体进行了网络层、移动边缘主机层和移动边缘系统层的划分。其中 MEC 主机层包含主机(ME host)和相应的主机层管理实体(ME host-level management entity),ME 主机又可以进一步划分为 ME 平台(ME platform)、ME 应用(ME application)和虚拟化基础设施(Virtualization Infrastructure)。网络层主要包含 3GPP 蜂窝网络、本地网络和外部网络等相关的外部实体,该层主要表示 MEC 工作系统与局域网、蜂窝移动网或者外部网络的接入情况。最上层是 ME 系统层的管理实体,负责承载应用程序及 MEC 系统的全局掌控。

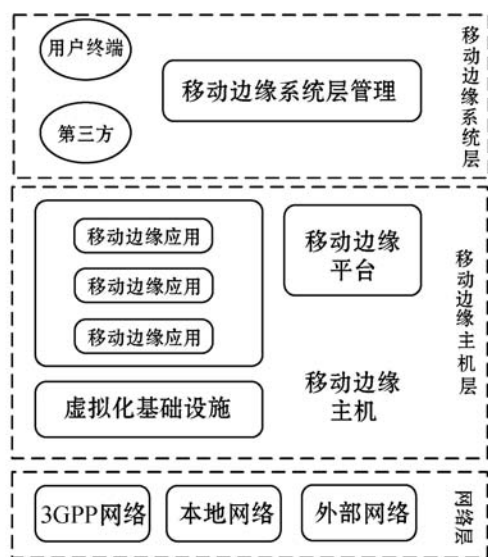


图 1 MEC 架构示意图

### 1.2 MEC 计算卸载技术

MEC 计算卸载技术不仅解决了移动设备在计算性能、资源存储和能效方面的不足,还减轻了核心网的压力,降低了因传输带来的时延。计算卸载主要包含资源分配和卸载决策两个方面。资源分配主要研究将资源卸载到哪里的问题;卸载决策则研究的是用户终端如何卸载和卸载内容的问题。图 2 为计算卸载决策和资源分配示意图。其中:(a)为 CUE 卸载决策的 3 种方式,分别是本地执行、完全卸载和部分卸载,具体决策结果可通过 CUE 能量消耗和完成计算任务时延决定;(b)为计算卸载的资源分配方案,该方案采用 MDP 解决了在 SCeNB 中的虚拟机(Virtual Machine, VM)分配问题。CUE1 通过创建 VM,将计算任务全部卸载到 SCeNB1 上。对于 CUE2,考虑到从 SCeNB1 到其他 SCeNB 的时延问题,CUE2 在时延更低的 SCeNB3 上进行计算卸载。但是,如果考虑 VM 的迁移成本时,一般会在 MEC 计算资源充足的情况下,优先选择近距离的 MEC 中进行卸载。

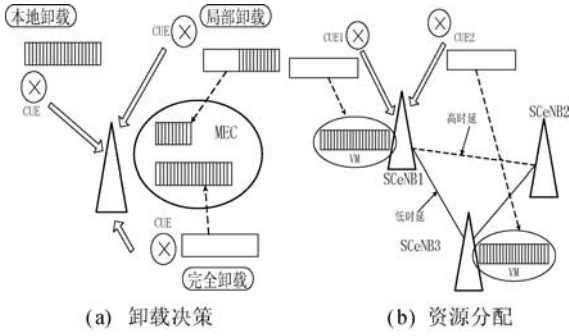


图2 计算卸载决策和资源分配示意图

## 2 计算卸载方案

考虑一个与边缘云关联的基站,可以服务  $N$  个 CUE,每个 CUE 都有一个计算密集型任务要执行。同时,附近还存在  $M$  个带有空闲处理器的 HUE。为了激励 HUE 辅助 CUE 进行计算卸载,CUE 需要分出一部分可用带宽给 HUE,因此,假设每个 HUE 至多可以帮助一个 CUE,不考虑用户的移动性和切换的情况下,每个 CUE 有两个选择:

(1) 将其任务卸载到边缘云上,这时 CUE 的全带宽可用于卸载,因此卸载延迟被最小化,但是对任务应用的计算能力降低。

(2) 将部分带宽提供给一个 HUE,然后将部分任务卸载到云,另外部分则卸载到该 HUE。此时由于只有部分带宽可供卸载导致卸载延迟会增加,但是具备更多的应用计算能力。

### 2.1 通信模型

每个 CUE 都具备一定的带宽  $B$  (以赫兹为单位),假设  $\alpha_{i,j} \in (0, 1)$  是 HUE  $j$  协助 CUE  $i$  所需的带宽份额。当 CUE  $i$  卸载到 HUE  $j$  和利用基站卸载到边缘云时,CUE  $i$  到 HUE  $j$  和基站的遍历瑞利衰落信道容量分别为:

$$R_{i,j} = (1 - \alpha_{i,j}) B \exp\left(\frac{(1 - \alpha_{i,j}) N_0 B}{P_i g_{i,j}}\right) \times E_1\left(\frac{(1 - \alpha_{i,j}) N_0 B}{P_i g_{i,j}}\right) \log_2 e \quad (1)$$

$$R_{i,MEC} = (1 - \alpha_{i,j}) B \exp\left(\frac{(1 - \alpha_{i,j}) N_0 B}{P_i g_{i,MEC}}\right) \times E_1\left(\frac{(1 - \alpha_{i,j}) N_0 B}{P_i g_{i,MEC}}\right) \log_2 e \quad (2)$$

式中,  $E_1(x) = \int_1^{\infty} t^{-1} e^{-xt} dt$  是指数积分; $g_{i,j}$  和  $g_{i,MEC}$  分别是 CUE  $i$  到 HUE  $j$  和基站的规模信道增益; $P_i$  是 CUE  $i$  的发射功率; $N_0$  是噪声谱密度。

在 CUE  $i$  所提供的带宽上,HUE  $j$  达到了其预期接收器的比特率,其规模增益为  $g_j$ :

$$R_{i,j} = \alpha_{i,j} B \exp\left(\frac{\alpha_{i,j} N_0 B}{P_j g_j}\right) E_1\left(\frac{\alpha_{i,j} N_0 B}{P_j g_j}\right) \log_2 e \quad (3)$$

在只有云的卸载中,CUE  $i$  保留了它的全部带宽。因此,它对基站的比特率是  $R'_{i,MEC}$ ,由式(2)在  $\alpha_{i,j} = 0$  时给出。

### 2.2 计算模型:云-HUE 卸载

CUE  $i$  有一个计算任务  $\phi_i = (b_i, \beta_i)$ ,其中  $b_i$  和  $\beta_i$  分别是任务总量(以 bit 为单位)和计算一个单元任务所需的处理器周期。假设 CUE  $i$  将  $b_i^j$  任务量卸载到 HUE  $j$ ,然后将  $b_i^{MEC}$  任务量卸载到云。其余的  $b_i - b_i^j - b_i^{MEC}$  任务量是在 CUE 本地计算的。设  $f_i$  为 CUE  $i$  的处理能力。CUE  $i$  的  $\phi_i$  在本地计算所需要的时间是:

$$T_i^j = \frac{\beta_i (b_i - b_i^j - b_i^{MEC})}{f_i} \quad (4)$$

式中: $b_i^j$  表示卸载到 HUE  $j$  上的任务量; $b_i^{MEC}$  表示卸载到云上的任务量。两者的卸载延迟分别为:

$$\tau_{i,j} = \frac{b_i^j}{R_{i,j}} \quad (5)$$

$$\tau_{i,MEC} = \frac{b_i^{MEC}}{R_{i,MEC}} \quad (6)$$

假设  $f_j$  为 HUE  $j$  的处理能力, $f_i$  表示云在 CUE  $i$  的处理能力,当云总处理能力为  $F$  时,则  $\sum_{i \in C} F_i \leq F$ 。

HUE  $j$  和云的  $\phi_i$  的计算时间为:

$$T_j^i = \frac{\beta_i b_i^j}{f_j} \quad (7)$$

$$T_{MEC}^i = \frac{\beta_i b_i^{MEC}}{F_i} \quad (8)$$

由于 CUE  $i$  顺序卸载到 HUE  $j$  和云上,因此在 HUE  $j$  和云处的总体完成时间分别为  $\tau_{i,j} + T_j^i$  和  $\tau_{i,j} + \tau_{i,MEC} + T_{MEC}^i$ 。因此,任务  $\phi_i$  的总体完成时间为:

$$T_{i,j} = \max(T_j^i, \tau_{i,j} + T_j^i, \tau_{i,j} + \tau_{i,MEC} + T_{MEC}^i) \quad (9)$$

本文忽略了发送计算结果所花费的时间,因为相对于输入数据,输出数据的大小往往很小。

### 2.3 计算模型:仅云卸载

假设现在一个 CUE  $k$  ( $k \in C, k \neq i$ ) 只将其任务的一部分卸载到云上。总的完成时间为:

$$T_k = \max(T_k, \tau_k + T_{MEC}^k) \quad (10)$$

式中: $\tau_k = b_k^{MEC} / R'_{k,MEC}$  和  $T_k = \beta_k (b_k - b_k^{MEC}) / f_k$  分别是卸载延迟和 CUE  $k$  的本地计算时间。

### 3 资源分配和解决方案

让  $\pi$  表示所有用户的一个集合分区,其中每个子集都有一个 CUE,最多有一个 HUE, $\Pi$  表示所有这些可能分区的集合。例如,当  $C = \{1,2\}$  和  $H = \{1\}$  时,有三个分区:

$$\Pi = \{\{1,1\}, \{2\}, \{1\}, \{2,1\}, \{1\}, \{2\}\} \quad (11)$$

在每个子集中,第一项和第二项分别是 CUE 和 HUE,假设  $\rho_\delta$  和  $\zeta_\delta$  分别表示  $\delta$  集合中具有基数 1 和 2 的子集,例如  $\delta = \{\{1\}, \{2,1\}\}$  表示 CUE1 只卸载到云上,而 CUE2 卸载到云和 HUE1 上,  $\rho_\delta = \{1\}$ ,  $\zeta_\delta = \{2,1\}$ 。然后,决定哪个 HUE 与每个 CUE 配对,哪个任务共享卸载到配对的 HUE 和云,以及如何在用户之间划分云资源的问题可以表示为:

$$\begin{aligned} \min_{\delta \in \Pi, \mathbf{F}, \mathbf{b}_\delta, \alpha_{\zeta_\delta}} \quad & \max(\max_{\{i,j\} \in \zeta_\delta} T_{i,j}, \max_{k \in \rho_\delta} T_k) \\ \text{s. t.} \quad & \sum_{i=1}^N F_i \leq \mathbf{F}, R_{j,i} \geq R_{j,th} \quad \forall \{i,j\} \in \zeta_\delta \end{aligned} \quad (12)$$

式中:  $R_{j,th}$  表示 HUE  $j$  在 CUE  $i$  带宽上所需的比特率;  $\mathbf{b}_\delta$  是  $b_i^j$ 、 $b_i^{\text{MEC}}$  和  $b_k^{\text{MEC}}$  的向量;  $\alpha_{\zeta_\delta}$  和  $\mathbf{F}$  分别表示  $\alpha_{i,j}$  和  $F_i$  的向量。

由于  $\alpha_{i,j} < 1$ , 所以

$$\begin{aligned} R_{j,th} < R_{j,th}^{\max} = \\ & \text{Bexp}\left(\frac{N_0 B}{P_j g_j}\right) E_1\left(\frac{(1 - \alpha_{i,j}) N_0 B}{P_j g_j}\right) \log_2 e \end{aligned} \quad (13)$$

如果 HUE 的速率请求  $R_{j,th}$  等于或者大于  $R_{j,th}^{\max}$ , 那么 CUE 不能够满足 HUE 的要求,因此这样的 HUE 不属于式(12)。

#### 3.1 HUE 固定分配

假定 HUE 到 CUE 的分配是固定的,并且优化扩展到任务共享以卸载和用户之间的云资源分区。

##### 3.1.1 最佳解决方案

对于给定的 HUE 分配  $\delta$ , 式(12)可以变为:

$$\begin{aligned} \min_{\mathbf{F}, \mathbf{b}_\delta, \alpha_{\zeta_\delta}} \quad & \max(\max_{\{i,j\} \in \zeta_\delta} T_{i,j}, \max_{k \in \rho_\delta} T_k) \\ \text{s. t.} \quad & \sum_{i=1}^N F_i \leq \mathbf{F}, R_{j,i} \geq R_{j,th} \quad \forall \{i,j\} \in \zeta_\delta \end{aligned} \quad (14)$$

式中:  $\zeta_\delta$  表示  $\delta$  集合中基数为 2 的子集。

随着  $\alpha_{i,j}$  的增加,  $T_{i,j}$  和  $R_{j,i}$  也逐渐变大,当获取最佳  $\alpha_{i,j}^*$  时,  $R_{j,i} = R_{j,th}$ 。然后,在式(14)中通过删除第二个约束并引入松弛变量:

$$V = \max(\max_{\{i,j\} \in \zeta_\delta} T_{i,j}, \max_{k \in \rho_\delta} T_k) \quad (15)$$

可以获得:

$$\begin{aligned} \min_{\mathbf{F}, \mathbf{b}_\delta} \quad & V \\ \text{s. t.} \quad & T_i^j \leq V \quad \forall \{i,j\} \in \zeta_\delta \\ & \tau_{i,j} + T_j^i \leq V \quad \forall \{i,j\} \in \zeta_\delta \\ & \tau_{i,j} + \tau_{i,\text{MEC}} + T_{\text{MEC}}^i \leq V \quad \forall \{i,j\} \in \zeta_\delta \\ & T_k \leq V \quad k \in \rho_\delta \\ & T_{\text{MEC}}^k + \tau_k \leq V \quad k \in \rho_\delta \\ & \sum_{i=1}^N F_i \leq \mathbf{F} \end{aligned} \quad (16)$$

式中:  $\tau_{i,j}$  是  $\alpha_{i,j}^*$  的函数。虽然式(16)是非凸的,但可以通过单一凝聚方法将其转换为几何规划(Geometric Programming, GP)问题。分子中具有多项式的分数约束和分母中的单项式可以转换为凸函数。并且,如果约束是对数的比例,则分母可以近似为单项式。利用下面不等式可以转换约束条件:

$$f(x) = \sum_{\gamma} f_{\gamma}(x) \geq \hat{f}(x) = \prod_{\gamma} \left[ \frac{f_{\gamma}(x)}{\delta_{\gamma}} \right]^{\delta_{\gamma}} \quad (17)$$

式中:  $\delta_{\gamma} > 0$ ,  $\sum_{\gamma} \delta_{\gamma} = 1$ ;  $f(x)$  表示多项式;  $f_{\gamma}(x)$  表示单项式。

本文应用迭代技术来最优求解式(16),对于迭代  $t$ , 使用式(17)和式(18)将其中的第一约束转换为单项式:

$$\begin{aligned} \beta_i b_i \left( \frac{V(t) f_i}{\lambda_1(t)} \right)^{-\lambda_1(t)} \left( \frac{\beta_i b_i^j(t)}{\lambda_2(t)} \right)^{-\lambda_2(t)} \times \\ \left( \frac{\beta_i b_i^{\text{MEC}}(t)}{\lambda_3(t)} \right)^{-\lambda_3(t)} \leq 1 \end{aligned} \quad (18)$$

式中:  $\lambda_1(t)$ 、 $\lambda_2(t)$ 、 $\lambda_3(t)$  为:

$$\begin{aligned} \lambda_1(t) &= \frac{V(t-1) f_i}{V(t-1) f_i + \beta_i b_i^j(t-1) + \beta_i b_i^{\text{MEC}}(t-1)} \\ \lambda_2(t) &= \frac{\beta_i b_i^j(t-1)}{V(t-1) f_i + \beta_i b_i^j(t-1) + \beta_i b_i^{\text{MEC}}(t-1)} \\ \lambda_3(t) &= \frac{\beta_i b_i^{\text{MEC}}(t-1)}{V(t-1) f_i + \beta_i b_i^j(t-1) + \beta_i b_i^{\text{MEC}}(t-1)} \end{aligned} \quad (19)$$

同样,对于每次迭代  $t$ , 第四约束条件可以通过式(17)和式(18)将其转换为单项式:

$$\beta_k b_k \left( \frac{V(t) f_k}{\gamma_1(t)} \right)^{-\gamma_1(t)} \left( \frac{\beta_k b_k^{\text{MEC}}(t)}{\gamma_2(t)} \right)^{-\gamma_2(t)} \leq 1 \quad (20)$$

式中:  $\gamma_1(t)$ 、 $\gamma_2(t)$  为:

$$\begin{aligned} \gamma_1(t) &= \frac{V(t-1) f_k}{V(t-1) f_k + \beta_k b_k^{\text{MEC}}(t-1)} \\ \gamma_2(t) &= \frac{\beta_k b_k^{\text{MEC}}(t-1)}{V(t-1) f_k + \beta_k b_k^{\text{MEC}}(t-1)} \end{aligned} \quad (21)$$

总而言之,迭代  $t$  要解决的整体优化问题是:

$$\min_{F(t), b_\delta(t)} V(t) \quad (22)$$

s. t. 式(18)

$$\frac{b_i^j(t)}{R_{i,j}} + \frac{\beta_i b_i^j(t)}{f_i} \leq V(t) \quad \forall \{i,j\} \in \zeta_\delta$$

$$\frac{b_i^j(t)}{R_{i,j}} + \frac{b_i^{\text{MEC}}(t)}{R_{i,\text{MEC}}} + \frac{\beta_i b_i^{\text{MEC}}(t)}{F_i(t)} \leq V(t) \quad \forall \{i,j\} \in \zeta_\delta$$

$$\frac{b_k^{\text{MEC}}(t)}{R'_{k,\text{MEC}}} + \frac{\beta_k b_k^{\text{MEC}}(t)}{F_k(t)} \leq V(t) \quad k \in \rho_\delta$$

$$\text{式(20)} \quad k \in \rho_\delta$$

$$\sum_{i=1}^N F_i(t) \leq F$$

当  $|V(t) - V(t-1)| < \varepsilon, 0 \leq \varepsilon < 1$  时迭代终止。

**算法 1** 基于 GP 的固定 HUE 分配算法

1. 令  $t=1$ , 初始化  $V(t), F_i(t), b_i^j(t), b_i^{\text{MEC}}(t), b_k^{\text{MEC}}(t), \forall \{i,j\} \in \zeta_\delta, k \in \rho_\delta$ , 保持式(16)的可行性;
2. while true do
  - t = t + 1;
  - 计算  $\lambda_1(t), \lambda_2(t), \lambda_3(t)$ ;
  - 利用 GGPLAB 求解式(22), 寻找  $V(t), F_i(t), b_i^j(t), b_i^{\text{MEC}}(t), b_k^{\text{MEC}}(t)$  的最优解;
  - if  $|V(t) - V(t-1)| \leq \varepsilon$  then
    - Break;
  - end if
- end while

### 3.1.2 次优云资源分配

在式(12)中, 云资源的最佳分配取决于 HUE 分配。制定了一个独立于这些任务的有效的次优分配。为此, 考虑每个 CUE 仅卸载到云的情况, 即:

$$\min_{F_i, b_i^{\text{MEC}}} \max_i T_i \quad \text{s. t.} \quad \sum_{i=1}^N F_i \leq F \quad (23)$$

上述优化问题类似于式(16), 因此可以使用基于 GP 的算法最佳地求解。通过求解式(23)得到的  $F_i$  值就是所寻求的次优分配。

### 3.2 云资源固定分配

对于给定的云资源分配, 可以简化为最小化每个 CUE 的完成时间, 即对于  $\{i,j\} \in \zeta_\delta$  和  $k \in \rho_\delta$ , 分别得到  $b_i^j, b_i^{\text{MEC}}$  和  $b_k^{\text{MEC}}$  的时间  $T_{i,j}, T_k$  的最小化值。这些优化可以表示为:

$$\min_{b_i^j, b_i^{\text{MEC}}} V_1 \quad (24)$$

$$\text{s. t.} \quad \frac{\beta_i (b_i - b_i^j - b_i^{\text{MEC}})}{f_i} \leq V$$

$$\frac{b_i^j}{R_{i,j}} + \frac{\beta_i b_i^j}{f_j} \leq V$$

$$\frac{b_i^j}{R_{i,j}} + \frac{b_i^{\text{MEC}}}{R_{i,\text{MEC}}} + \frac{\beta_i b_i^{\text{MEC}}}{F_i} \leq V$$

和

$$\min_{b_k^{\text{MEC}}} V_2 \quad (25)$$

$$\text{s. t.} \quad \frac{\beta_k (b_k - b_k^{\text{MEC}})}{f_k} \leq V$$

$$\frac{b_k^{\text{MEC}}}{R'_{k,\text{MEC}}} + \frac{\beta_k b_k^{\text{MEC}}}{F_k} \leq V$$

式中:  $V_1$  和  $V_2$  为松弛变量。式(24)中的优化是一个线性规划问题, 可以用变量和比特数的多项式来最优求解。令  $B_i^{\text{MEC},j}$  和  $B_i^j$  分别是卸载到云和 HUE  $j$  的比特数的后续解。式(24)中的云-HUE 卸载的完成时间可以表示为:

$$\Gamma_{i,j} = \frac{\beta_i (b_i - B_i^j - B_k^{\text{MEC},j})}{F_i} \quad (26)$$

式(25)中的第一和第二约束分别随 MEC 的增加而减小。因此, 当两个约束都满足相等条件时, 就可以得到卸载到云中的最佳比特数:

$$B_k^{\text{MEC}} = \frac{\beta_k b_k R'_{k,\text{MEC}} F'_k}{\beta_k R'_{k,\text{MEC}} (f_k + F'_i) + f_k F'_i} \quad (27)$$

CUE  $k$  仅云卸载的完成时间为:

$$\Gamma_k = \frac{\beta_k (b_k - B_k^{\text{MEC}})}{F'_i} \quad (28)$$

在解决式(12)中低复杂度的 HUE 分配时, 可以将  $F_i$  赋值为式(23)的解  $F'_i$ , 以及将  $b_i^j, b_i^{\text{MEC}}, b_k^{\text{MEC}}$  赋值为式(24)、式(25)的解  $B_i^j, B_i^{\text{MEC}}, B_k^{\text{MEC}}$ 。

### 3.3 HUE 最佳分配

式(12)的最优解可以通过搜索 3.1.1 节中描述的所有可能的 HUE 分配以及对每个 HUE 分配的优化来获得。但是, 这需要搜索  $(M+N)!/M!$  个 HUE 分配。或者, 从 3.1.2 节和 3.2 节中导出的用于云资源分配和卸载位数的次优解中, 因此 HUE 分配问题可以简化为:

$$\min_{\delta \in \Pi} \max(\max_{\{i,j\} \in \zeta_\delta} \Gamma_{i,j}, \max_{k \in \rho_\delta} \Gamma_k) \quad (29)$$

并通过一个二分图匹配算法对其进行优化求解。

给定一个二分图  $G$ , 包含顶点集  $v$  和边集  $\eta$ 。当  $v = v_1 + v_2, \eta$  中的每个边都可以将  $v_1$  中的一个顶点连接到  $v_2$  中的一个顶点。图 1(a) 为二分图示意图, 其中:  $v_1 = \{v_1^1, v_1^2\}, v_2 = \{v_2^1, v_2^2\}$  是两个顶点集, 边集为:

$$\eta = \{(v_1^1, v_2^1), (v_1^1, v_2^2), (v_1^2, v_2^1), (v_1^2, v_2^2)\} \quad (30)$$

$G$  中的匹配是  $\eta$  的一个子集, 这样每个顶点  $v \in v$  最多与匹配的一个边相关联。  $G$  中最大匹配  $M^*$  包含

最大可能的边数。对于图 1(a) 中的二部图,两个可能的最大匹配是  $\{(v_1^1, v_2^1), (v_1^2, v_2^2)\}$  和  $\{(v_1^1, v_2^2), (v_1^2, v_2^1)\}$ 。

对于式(29),网络被表示为一个加权二分图,每个 CUE  $i$  和 HUE  $j$  分别表示归属于不同子集的顶点  $v_1^i \in v_1$  和  $v_2^j \in v_2$ ,边  $(v_1^i, v_2^j)$  的权重可以表示为:

$$W_{(v_1^i, v_2^j)} = \Gamma_{ij} \quad (31)$$

此图的最大匹配对应于 CUE 和 HUE 之间的配对。为了包含仅云卸载选项,将  $n$  个虚拟顶点添加到  $v_2$  中,其中第  $i$  个虚拟顶点表示 CUE  $i$  的仅云卸载选项。顶点  $v_1^i$  和  $v_2^{M+i}$  之间的边权重根据仅云卸载的完成时间进行分配:

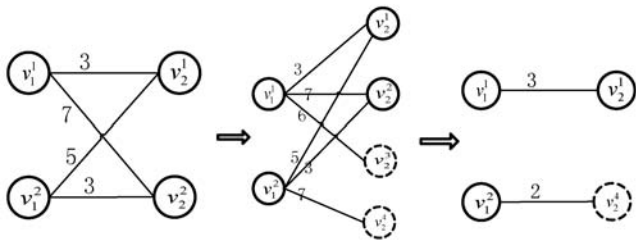
$$W_{(v_1^i, v_2^{M+i})} = \Gamma_i \quad (32)$$

式(32)中的 HUE 选择问题可以表示为所定义的图的瓶颈匹配(Bottleneck Matching, BM)问题,即最大边缘权重尽可能小的最大匹配问题:

$$\min_{\phi \in \Phi} \max_{(v_1^i, v_2^j) \in \phi} W_{(v_1^i, v_2^j)} \quad (33)$$

式中:  $\Phi$  包含所有可能的最大匹配,并且构造的二分图具有  $MN + N$  个边和  $2N + M$  个顶点。因此,可以采用 BM 算法,在复杂度为  $\Omega(\max(N^2 \sqrt{M}, M^2 \sqrt{N}))$  的情况下,最优地解决 HUE 分配问题。如果顶点  $v_1^i$  与图中虚拟顶点  $v_2^{M+i}$  配对时,CUE  $i$  只卸载到云上。

图 3 给出了一个包含有 CUE  $\{1, 2\}$  和 HUE  $\{1, 2\}$  网络的图形构造和 BM 输出。有、无 HUE 辅助时的完成时间分别是:  $\Gamma_1 = 6, \Gamma_2 = 2, \Gamma_{1,1} = \Gamma_{2,2} = 3, \Gamma_{1,2} = 7, \Gamma_{2,1} = 5$ 。顶点集  $\{v_1^1, v_1^2\}$  和  $\{v_2^1, v_2^2\}$  分别对应 CUE 和 HUE。虚拟顶点  $v_2^3$  和  $v_2^4$  分别对应 CUE1 和 CUE2 仅卸载到云端的操作。BM 的结果输出是  $\{(v_1^1, v_2^1), (v_1^2, v_2^4)\}$ ,因此 CUE1 卸载到 HUE1 和云端,而 CUE 2 卸载到云端。



(a) 图形网络 (b) 带有虚拟顶点的图像 (c) BM 输出

图 3 图形构造和 BM 输出

## 4 实验结果与分析

对于随后的评估,CUE 和 HUE 均匀分布在半径为

50 m 的圆形区域上,中心有云相关的 BS。HUE 本身的传输直接指向 BS 本身。每个 HUE 的速率阈值等于  $0.3R_{j,\text{th}}^{\text{max}}$ 。表 1 为基于文献[10]的其余参数。实验中的对比数据是 300 次实验的平均结果。

表 1 模拟参数

参数	值
$f_i, f_j$	[0.5, 1.5] GHz
$\beta_i$	[500, 1 500] bit
$b_i$	[100, 500] Kb
$P_i, P_j$	200 mW
$B$	1 MHz
$N_0$	-147 dBm/Hz
$\varepsilon$	$10^{-5}$

本文主要对比 4 种不同的卸载方案:

(1) “HUE-云 BM”方案,卸载量由式(24)、式(25)给出,HUE 分配通过 BM 算法获得。

(2) “HUE-云 BM-GP”,首先获得卸载量和 HUE 分配,然后通过算法 1 求解式(16),更新卸载量和云资源分配。

(3) “HUE-云 随机选择”,其中卸载量是式(24)、式(25)的解决方案,每个 CUE 随机分配一个 HUE。这个基线的复杂性是  $\Omega(\min(N, M))$ 。

(4) “云”,仅卸载到云,这是任务计算的传统选择。

图 4 - 图 5 为不同策略的完成时间对比,其中云的计算能力在 4 ~ 20 GHz 之间变化,图 4 中有 30 个 CUE 和 50 个 HUE 存在网络中,图 5 则将网络中的 CUE 和 HUE 的数量分别增加到 70 和 100。可以发现,卸载到云和 HUE 的具有极大的优势,即便 HUE 是随机分配的。算法 1 对计算结果的提升度很小,简单的“HUE-云 BM”方案即可达到理想的结果。

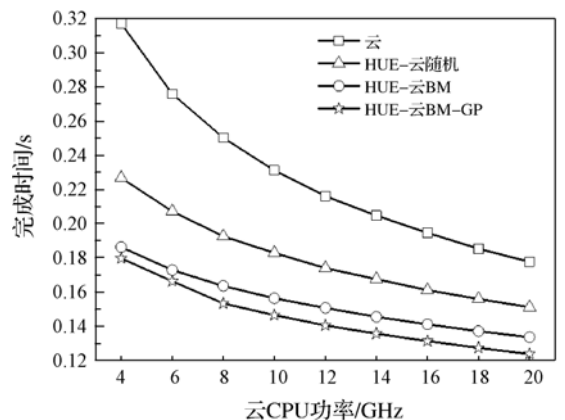


图 4 不同卸载方案的完成时间对比(CUE = 30)

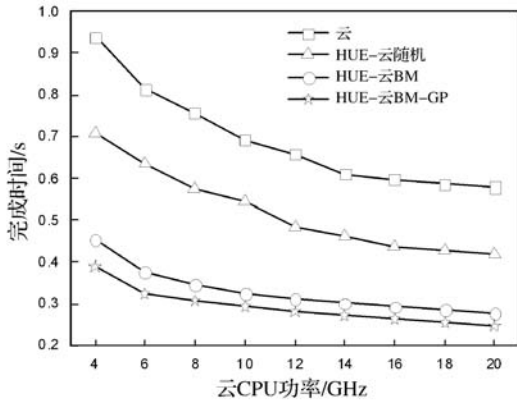


图5 不同卸载方案的完成时间对比(CUE = 70)

在图6中,通过详细检查特定CUE任务的卸载延迟和计算时间,进一步了解不同方案的性能。通过分析不同方案在本地、HUE与云的计算时间和卸载时间发现,计算时间在不同的处理器之间很平衡,并且卸载延迟时间在总时间中所占比例不高。

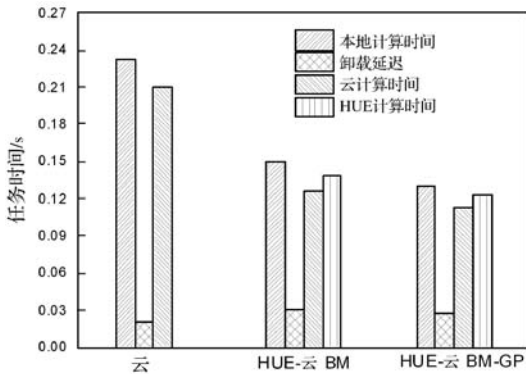


图6 不同卸载方案的计算时间和卸载延迟对比

## 5 结语

本文提出一个将计算卸载到边缘云和移动对等端的通用框架。该设计旨在保持应用程序延迟需求的同时,最大限度地减少总计算时间。数值结果表明,通过带宽激励,将计算卸载到边缘云和移动对等端的方案是可行的。与将计算卸载到边缘云方案相比,该方案具有较大的性能增益,在卸载延迟时间没有明显增加的情况下,总计算时间可以减少35%~40%。

## 参考文献

- [1] Abbas N, Zhang Y, Taherkordi A, et al. Mobile edge computing: A survey [J]. IEEE Internet of Things Journal, 2018, 5(1): 450-465.
- [2] Tran T X, Pompili D. Joint task offloading and resource allocation for multi-server mobile-edge computing networks [J]. IEEE Transactions on Vehicular Technology, 2019, 68(1): 856-868.
- [3] Mach P, Becvar Z. Mobile edge computing: A survey on ar-

chitecture and computation offloading [J]. IEEE Communications Surveys & Tutorials, 2017, 19(3): 1628-1656.

- [4] 于博文, 蒲凌君, 谢玉婷, 等. 移动边缘计算任务卸载和基站关联协同决策问题研究 [J]. 计算机研究与发展, 2018, 55(3): 537-550.
- [5] 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.
- [6] 柳兴, 李建彬, 杨震, 等. 移动云计算中的一种任务联合执行策略 [J]. 计算机学报, 2017, 40(2): 364-377.
- [7] 张文柱, 曹菲菲, 周雪婷. 移动云环境下高能效的移动终端计算迁移策略 [J]. 西安电子科技大学学报(自然科学版), 2017, 44(3): 175-180.
- [8] 曹宾, 梁裕丞, 罗雷, 等. ad hoc 云环境中分布式博弈卸载策略 [J]. 通信学报, 2017, 38(11): 24-34.
- [9] Ti N T, Le L B. Computation offloading leveraging computing resources from edge cloud and mobile peers [C]//2017 IEEE International Conference on Communications (ICC), 2017.
- [10] Du J, Zhao L, Feng J, et al. Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee [J]. IEEE Transactions on Communications, 2018, 66(4): 1594-1608.

(上接第5页)

系统功能测试阶段已过,处于程序优化阶段,但基本功能已经实现。今后将继续优化该系统的两大功能,进一步完善 RoboDK 相关拓展功能<sup>[8-9]</sup>。

## 参考文献

- [1] 邓华健. 机器人离线编程系统的开发及其应用 [D]. 广州: 广东工业大学, 2017.
- [2] 李双双. 工业机器人建模、运动仿真与轨迹优化 [D]. 呼和浩特: 内蒙古大学, 2012.
- [3] 俊杰. 基于无线网络的机器人远程控制系统集成平台研究 [D]. 北京: 北京交通大学, 2012.
- [4] 罗瑜. 远程监控系统中网络通信的研究与实现 [D]. 武汉: 武汉理工大学, 2006.
- [5] 魏学舟, 刘涛. 基于 Snap7 的 PLC 上位机监控软件开发 [J]. 设备管理与维修, 2018(14): 129-131.
- [6] 李华, 安磊. 利用 Python Snap7 实现上位机与西门子 PLC 的通信 [J]. 数字化用户, 2018, 24(28): 20.
- [7] 王博, 黎柏春, 杨建宇, 等. 智能制造系统的 6R 工业机器人仿真和监控平台 [J]. 哈尔滨工程大学学报, 2019, 40(2): 365-373.
- [8] 刘和彬. 基于 VR 的工业机器人任务仿真与实时监控研究 [D]. 沈阳: 东北大学, 2013.
- [9] Rossmann J, Hempe N, Emde M, et al. A real-time optical sensor simulation framework for development and testing of industrial and mobile robot applications [C]//ROBOTIK 2012, 7th German Conference on Robotics, 2012.