

文章编号: 2095-4980(2018)06-1080-08

基于二次规划的任务卸载决策和资源分配方法

孟 敏

(无锡工艺职业技术学院 电子信息系, 江苏 无锡 214206)

摘 要: 主要研究移动用户均有多个独立任务的多用户移动云计算系统, 这些移动用户将任务卸载到云端时共享通信资源。如何对所有用户的任务卸载决策和通信资源分配进行联合优化, 以便使所有用户的能耗、计算量和延时降到最低是目前研究的难点。将该问题建模为 NP 难度的非凸的具有二次约束的二次规划(QCQP)问题, 提出一种高效的近似算法进行求解, 通过单独的半正定松弛(SDR)处理后, 确定二元卸载决策和通信资源最优分配。采用代表最小系统成本的性能下界作为性能基准进行仿真实验, 结果表明, 本文算法在多种参数配置下的性能均接近最优性能。

关键词: 移动云计算; 任务卸载; 资源分配; 二次规划; 近似算法; 最小系统成本

中图分类号: TP393

文献标志码: A

doi: 10.11805/TKYDA201806.1080

Research on task offloading decision and resource allocation method based on quadratic programming

MENG Min

(Department of Electronic Information, Wuxi Institute of Arts & Technology, Wuxi Jiangsu 214206, China)

Abstract: A general multi-user mobile cloud computing system where each mobile user has multiple independent tasks is considered. These mobile users share the communication resource while offloading tasks to the cloud. How to jointly optimize the task offloading decisions of all users as well as the allocation of communication resource in order to minimize the overall cost of energy, computation, and delay for all users is one of the difficult problems in current research. The optimization problem is formulated as a non-convex Quadratically Constrained Quadratic Program(QCQP), which is NP-hard, and an efficient approximate algorithm is proposed by using separable Semi-Definite Relaxation(SDR) to solve this problem, followed by recovery of the binary offloading decision and optimal allocation of the communication resource. The simulation experiments are carried out by using the lower bound of the minimum system cost as the performance benchmark, the results show that the proposed algorithm gives nearly optimal performance under various parameter settings.

Keywords: mobile cloud computing; task offloading; resource allocation; quadratic program; approximate algorithm; the minimum system cost

随着移动互联网的飞速发展和移动应用的极大丰富, 移动终端已成为人们工作生活不可或缺的一部分。移动支付、移动医疗、移动游戏及虚拟现实等日益复杂的移动应用对移动终端的计算能力、存储容量、电池容量以及安全性提出了更高的要求。为了突破移动终端计算、存储和电池等资源限制, 为移动用户提供更加丰富的应用, 将云计算引入移动环境, 移动云计算^[1-2]作为一种新的应用模式应运而生。

移动云计算技术以丰富的云资源为基础, 拓展了移动设备的性能, 提升了用户体验。为降低移动用户的能耗, 卸载其任务到云端进行处理是目前的主流方式^[3], 如何进行最优的任务卸载决策和通信资源分配是目前研究的热点^[4]。文献[5-9]对单个用户将其整个应用卸载到云端时系统能耗和系统性能之间的折衷关系进行了分析, 研究了面向单个用户的任务分割问题。其中, 文献[5]提出了 MAUI(Mobile Assistance Using Infrastructure)系统对分割后的任务进行高效处理; 文献[6]中的 Clonecloud 系统和文献[7]中的 Thinkair 系统对 MAUI 系统性能做了进一步优

收稿日期: 2018-09-14; 修回日期: 2018-10-30

基金项目: 国家自然科学基金面上项目资助(61601334)

化;文献[8]则考虑了多种无线电干扰条件下的任务分割问题。然而,单用户的任务分割会导致棘手的整数规划问题,虽然可以保证服务质量,但算法的计算量过高;文献[9]研究了单个用户对应多个独立任务和一个计算接入点的场景,通过考虑最坏情况下的任务卸载延时做出卸载决策。然而,单用户场景下的任务卸载会导致额外的开销(如移动用户和云端之间的通信延时),使得这些任务的服务质量容易受到影响^[10-11]。

为提升服务质量,本文进一步研究了多用户和云之间的交互问题。在多用户场景下,所有用户竞争一条容量有限的无线链路到达云端,因此需要联合考虑卸载决策及所有用户的通信资源共享问题,这显著增加了系统设计的难度。本文研究目标是对所有任务的卸载决策和通信资源分配进行联合优化,使所有用户的能耗、计算量及延时加权和最小化。该问题可表述为具有 NP 难度的非凸的具有二次约束的二次规划问题(QCQP)^[12]。为求解上述难题,提出一种基于半正定松弛(SDR)^[13]的高效求解算法来确定二元卸载决策和通信资源最优分配。仿真实验验证了所提算法的有效性。

1 系统模型和问题描述

1.1 多用户多任务场景下的移动云卸载

设有一个常见的云接入网络,包括1台云服务器、1个接入点(Access Point, AP)及 N 个移动用户,每个用户有 M 个独立任务,如图1所示。移动用户和接入点间进行无线连接,云和接入点间进行有线连接。每个移动用户均可在本地处理任务,或者通过接入点将部分任务卸载到云端处理。设 x_{ij} 表示用户 i 对任务 j 的二元卸载决策,则有:

$$x_{ij} = \begin{cases} 0, & \text{用户 } i \text{ 在本地处理任务 } j \\ 1, & \text{用户 } i \text{ 将任务 } j \text{ 卸载到云端} \end{cases} \quad (1)$$

1.2 本地处理的成本

设 $D_{in}(ij)$, $D_{out}(ij)$ 和 $App(ij)$ 分别表示用户 i 中任务 j 的输入数据量、输出数据量和应用类型。如果用户 i 在本地处理任务 j ,则处理能耗为 $E_{t_{ij}}$,处理时间为 $T_{t_{ij}}$ 。

1.3 远端处理的成本

如果用户 i 通过接入点(AP)将任务 j 卸载到云端,则移动用户和接入点间传输和接收数据所消耗的能量分别表示为 $E_{t_{ij}}$ 和 $E_{r_{ij}}$ 。对于移动用户和接入点间的无线连接,上行传输时间和下行传输时间分别计算为 $T_{t_{ij}} = D_{in}(ij)/c_{u_i}$ 和 $T_{r_{ij}} = D_{out}(ij)/c_{d_i}$,其中 c_{u_i} 和 c_{d_i} 分别表示分配给用户 i 进行数据传输的上行和下行带宽。其数值受上行链路容量 C_{UL} 、下行链路容量 C_{DL} 及其他用户通过接入点卸载到云端的任务数量的约束。

因为接入点需要持续将任务卸载到云端,将接入点和云端间传输数据所需时间表示为 $T_{ac_{ij}} = (D_{in}(ij) + D_{out}(ij))/R_{ac}$,云端处理时间表示为 $T_{c_{ij}} = D_{in}(ij) App(ij)/f_c$ 。假设接入点和云端间的有线传输速率 R_{ac} 以及每个用户的云端处理速率 f_c 提前确定, $T_{ac_{ij}}$ 和 $T_{c_{ij}}$ 只取决于各个任务的本身规模。用 $C_{c_{ij}}$ 表示在云端处理用户 i 中任务 j 所需要的系统效用成本。

1.4 问题描述

文中将所有用户的能耗、卸载和处理所有任务的成本及相应传输和处理延时的加权和定义为总体系统成本。目的是通过联合优化任务卸载决策 x_{ij} 和通信带宽资源分配 $r_i = (c_{u_i}, c_{d_i})$,使系统总体成本最小化,即降低移动用户的能耗,在处理用户任务时保证服务质量。该优化问题可表述为:

$$\begin{cases} \min_{\{x_{ij}\}, \{r_i\}} \sum_{i=1}^N \left[\sum_{j=1}^M \left(E_{t_{ij}} (1-x_{ij}) + E_{c_{ij}} x_{ij} \right) + \rho_i \max \{ T_{t_i}, T_{c_i} \} \right] \\ \text{约束: } \sum_{i=1}^N c_{u_i} \leq C_{UL}; \quad \sum_{i=1}^N c_{d_i} \leq C_{DL}; \quad c_{u_i}, c_{d_i} \geq 0, \forall i; \quad x_{ij} \in \{0, 1\}, \forall i, j \end{cases} \quad (2)$$

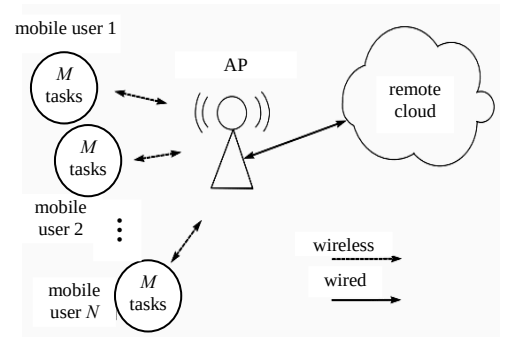


Fig.1 System model
图1 系统模型

式中: $E_{C_{ij}} \triangleq (E_{t_{ij}} + E_{r_{ij}} + \beta C_{c_{ij}})$ 为将用户 i 中任务 j 卸载到云端的能量成本, 它是传输能量和系统效用成本的加权和; β 为相对权重; $T_{L_i} \triangleq \sum_{j=1}^M T_{l_{ij}} (1 - x_{ij})$ 为移动用户 i 自己处理任务需要的处理延时; T_{C_i} 为移动用户 i 的任务在云端处理所需要的总体传输和远程处理延时; ρ_i 为系统总体成本中任务处理延时相对于能耗的权重, 根据系统对延时和能耗的性能要求, ρ_i 的值可适当调整。

式(2)中 $\sum_{i=1}^N c_{u_i} \leq C_{UL}$ 和 $\sum_{i=1}^N c_{d_i} \leq C_{DL}$ 分别表示上行和下行带宽资源约束。

2 多用户多任务卸载方案

式(2)属于混合整数非凸规划问题, 是一个 NP 难题^[14]。此外, 式(2)中的远程任务处理的总体延时 T_{C_i} 也难以准确计算。这是因为, 当某个用户卸载多个任务时, 传输时间和处理时间可能发生无法预计的重叠, 且该过程受到卸载决策、通信资源分配和任务调度次序的影响。实际上, 由于 T_{C_i} 由所有任务的上行传输时间、远程处理时间和下行传输时间构成, 可看成是多机器流水线作业调度的输出, 而多机器流水线作业调度问题目前还没有精确解^[15]。为高效求解式(2), 首先给出 T_{C_i} 的上界和下界, 然后将式(2)转化为 QCQP 问题, 最后提出基于 SDR 的近似算法来确定二元卸载决策 $\{x_{ij}\}$ 和通信资源分配方案 $\{r_i\}$ 。

2.1 远程处理延时的边界

当移动用户将多个任务卸载到云端时, 通信时间和处理时间会发生重叠, 难以准确计算总体延时 T_{C_i} 。首先确定 T_{C_i} 的上界, 并用其表示最坏情况下的延时(最大延时)。

$$T_{C_i}^U \triangleq \sum_{j=1}^M \left(\frac{D_{in}(ij)}{c_{u_i}} + \frac{D_{out}(ij)}{c_{d_i}} + T_{ac_{ij}} + T_{c_{ij}} \right) x_{ij}, \forall i \quad (3)$$

最大延时由未发生重叠的传输延时和处理延时相加而得, 因此在相同的卸载决策和资源分配条件下, 最大延时始终大于实际延时。另外, 将所有用户的卸载延时分解为多个成分, 并选择最大成分作为 T_{C_i} 的下界:

$$T_{C_i}^L \triangleq \max \{T_{u_i}, T_{d_i}, T_{uac_i}, T_{dac_i}, T_{C_i}^U\}, \forall i \quad (4)$$

式中: $T_{u_i} \triangleq \sum_{j=1}^M D_{in}(ij) x_{ij} / c_{u_i}$ 和 $T_{d_i} \triangleq \sum_{j=1}^M D_{out}(ij) x_{ij} / c_{d_i}$ 分别表示用户 i 和接入点之间的总体上行传输时间和总体下行传输时间; $T_{uac_i} \triangleq \sum_{j=1}^M D_{in}(ij) x_{ij} / R_{ac}$ 和 $T_{dac_i} \triangleq \sum_{j=1}^M D_{out}(ij) x_{ij} / R_{ac}$ 分别表示用户 i 的接入点和云端间的总体上行传输时间和总体下行传输时间; $T_{C_i}^U \triangleq \sum_{j=1}^M D_{in}(ij) App(ij) x_{ij} / f_c$ 表示用户 i 的云处理总体时间。

针对式(2)采用最大延时 $T_{C_i}^U$ 获得一个近似解, 作为系统总体成本的上界。然后, 通过类似方法, 利用 $T_{C_i}^L$ 获得系统总体成本的下界作为性能基准。

2.2 QCQP 转化和 SDR

利用 $T_{C_i}^U$ 将式(2)所示的优化问题转化为 QCQP 问题。首先, 将 $x_{ij} \in \{0, 1\}, \forall i, j$ 的整数约束重写为:

$$x_{ij} (x_{ij} - 1) = 0, \forall i, j \quad (5)$$

另外, 为式(2)中的 $\max \{T_{L_i}, T_{C_i}^U\}$ 引入辅助变量 t_i , $\sum_{j=1}^M D_{in}(ij) x_{ij} / c_{u_i}$ 引入辅助变量 D_{u_i} , $\sum_{j=1}^M D_{out}(ij) x_{ij} / c_{d_i}$ 引入辅助变量 D_{d_i} 。假设 $d_i = (D_{u_i}, D_{d_i})$ 。于是, 式(2)可等效转化为如下问题:

$$\left\{ \begin{array}{l} \min_{\{x_{ij}\}, \{r_i, d_i, t_i\}} \sum_{i=1}^N \left[\sum_{j=1}^M \left(E_{l_{ij}} (1-x_{ij}) + E_{c_{ij}} x_{ij} \right) + \rho_i t_i \right] \\ \text{约束:} \left\{ \begin{array}{l} \sum_{j=1}^M T_{l_{ij}} (1-x_{ij}) \leq t_i, \forall i; \quad D_{u_i} + D_{d_i} + \sum_{j=1}^M (T_{ac_{ij}} + T_{c_{ij}}) x_{ij} \leq t_i, \forall i \\ \sum_{j=1}^M D_{in}(ij) x_{ij} - c_{u_i} D_{u_i} \leq 0, \forall i; \quad \sum_{j=1}^M D_{out}(ij) x_{ij} - c_{d_i} D_{d_i} \leq 0, \forall i \\ \sum_{i=1}^N c_{u_i} \leq C_{UL}; \quad \sum_{i=1}^N c_{d_i} \leq C_{DL}; \quad c_{u_i}, c_{d_i} \geq 0, \forall i; \quad x_{ij} (x_{ij} - 1) = 0, \forall i, j \end{array} \right. \end{array} \right. \quad (6)$$

定义 $\mathbf{w}_i \triangleq [x_{i1}, x_{i2}, \dots, x_{iM}, c_{u_i}, D_{u_i}, c_{d_i}, D_{d_i}, t_i]^T, \forall i$, 且 $\mathbf{e}_i$ 表示第 i 个元素为 1 的 $(M+5) \times 1$ 标准单位向量。于是, 式(6)可进一步等价转化为如下的 QCQP 问题:

$$\left\{ \begin{array}{l} \min_{\{\mathbf{w}_i\}} \sum_{i=1}^N \mathbf{b}_i^T \mathbf{w}_i + \sum_{i=1}^N \sum_{j=1}^M E_{l_{ij}} \\ \text{约束:} \left\{ \begin{array}{l} \mathbf{b}_i^T \mathbf{w}_i \leq -\sum_{j=1}^M T_{l_{ij}}, \quad \mathbf{b}_i^T \mathbf{w}_i \leq 0, \quad \forall i; \quad \mathbf{w}_i^T \mathbf{A}_{u_i} \mathbf{w}_i + \mathbf{b}_{u_i}^T \mathbf{w}_i \leq 0, \forall i \\ \mathbf{w}_i^T \mathbf{A}_{d_i} \mathbf{w}_i + \mathbf{b}_{d_i}^T \mathbf{w}_i \leq 0, \forall i; \quad \sum_{i=1}^N \mathbf{b}_{u_i}^T \mathbf{w}_i = C_{UL}, \sum_{i=1}^N \mathbf{b}_{d_i}^T \mathbf{w}_i = C_{DL} \\ \mathbf{w}_i^T \text{diag}(\mathbf{e}_j) \mathbf{w}_i - \mathbf{e}_j^T \mathbf{w}_i = 0, \forall i, j; \quad \mathbf{w}_i \geq 0, \forall i \end{array} \right. \end{array} \right. \quad (7)$$

式中: $\mathbf{A}_{u_i} \triangleq \begin{bmatrix} \mathbf{0}_{M \times M} & \mathbf{0}_{M \times 2} & \mathbf{0}_{M \times 3} \\ \mathbf{0}_{2 \times M} & \mathbf{A}'_{u_i} & \mathbf{0}_{2 \times 3} \\ \mathbf{0}_{3 \times M} & \mathbf{0}_{3 \times 2} & \mathbf{0}_{3 \times 3} \end{bmatrix}; \quad \mathbf{b}_i \triangleq \left[(E_{c_{i1}} - E_{l_{i1}}), (E_{c_{i2}} - E_{l_{i2}}), \dots, (E_{c_{iM}} - E_{l_{iM}}), \mathbf{0}_{1 \times 4}, \rho_i \right]^T; \quad \mathbf{b}_{l_i} \triangleq -[T_{l_{i1}}, T_{l_{i2}}, \dots, T_{l_{iM}}, \mathbf{0}_{1 \times 4}, 1]^T;$

$\mathbf{A}_{d_i} \triangleq \begin{bmatrix} \mathbf{0}_{(M+2) \times (M+2)} & \mathbf{0}_{(M+2) \times 2} & \mathbf{0}_{(M+2) \times 1} \\ \mathbf{0}_{2 \times (M+2)} & \mathbf{A}'_{d_i} & \mathbf{0}_{2 \times 1} \\ \mathbf{0}_{1 \times (M+2)} & \mathbf{0}_{1 \times 2} & \mathbf{0} \end{bmatrix}; \quad \mathbf{b}_{c_i} \triangleq \left[(T_{ac_{i1}} + T_{c_{i1}}), (T_{ac_{i2}} + T_{c_{i2}}), \dots, (T_{ac_{iM}} + T_{c_{iM}}), 0, 1, 0, 1, -1 \right]^T; \quad \mathbf{b}_{D_i} \triangleq [\mathbf{0}_{1 \times M+2}, 1, \mathbf{0}_{1 \times 2}]^T;$

$\mathbf{b}_{d_i} \triangleq [D_{out}(i1), D_{out}(i2), \dots, D_{out}(iM), \mathbf{0}_{1 \times 5}]^T; \quad \mathbf{b}_{U_i} \triangleq [\mathbf{0}_{1 \times M}, 1, \mathbf{0}_{1 \times 4}]^T。$

式(6)和式(7)的优化约束均为一一对应。通过进一步定义 $\mathbf{z}_i \triangleq [\mathbf{w}_i^T, 1]^T$, 且删除式(7)中的常数项 $\sum_{i=1}^N \sum_{j=1}^M E_{l_{ij}}$, 可以将式(7)转化为如下问题:

$$\left\{ \begin{array}{l} \min_{\{\mathbf{z}_i\}} \sum_{i=1}^N \mathbf{z}_i^T \mathbf{G}_i \mathbf{z}_i \\ \text{约束:} \left\{ \begin{array}{l} \mathbf{z}_i^T \mathbf{G}_i \mathbf{z}_i \leq -\sum_{j=1}^M T_{l_{ij}}, \forall i; \quad \mathbf{z}_i^T \mathbf{G}_r \mathbf{z}_i \leq 0, \forall i, r = c, u, d \\ \sum_{i=1}^N \mathbf{z}_i^T \mathbf{G}_{U_i} \mathbf{z}_i \leq C_{UL}, \sum_{i=1}^N \mathbf{z}_i^T \mathbf{G}_{D_i} \mathbf{z}_i \leq C_{DL} \\ \mathbf{z}_i^T \mathbf{G}_{I_j} \mathbf{z}_i = 0, \forall i, j; \quad \mathbf{z}_i \geq 0, \forall i \end{array} \right. \end{array} \right. \quad (8)$$

式中: $\mathbf{G}_i \triangleq \begin{bmatrix} \mathbf{0}_{(M+5) \times (M+5)} & \frac{1}{2} \mathbf{b}_i \\ \frac{1}{2} \mathbf{b}_i^T & 0 \end{bmatrix}, \forall i; \quad \mathbf{G}_{l_i} \triangleq \begin{bmatrix} \mathbf{0}_{(M+5) \times (M+5)} & \frac{1}{2} \mathbf{b}_{l_i} \\ \frac{1}{2} \mathbf{b}_{l_i}^T & 0 \end{bmatrix}, \forall i; \quad \mathbf{G}_{s_i} \triangleq \begin{bmatrix} \mathbf{A}_{s_i} & \frac{1}{2} \mathbf{b}_{s_i} \\ \frac{1}{2} \mathbf{b}_{s_i}^T & 0 \end{bmatrix}, \forall i, s = u, d; \quad \mathbf{G}_{S_i} \triangleq \begin{bmatrix} \mathbf{0}_{(M+5) \times (M+5)} & \frac{1}{2} \mathbf{b}_{S_i} \\ \frac{1}{2} \mathbf{b}_{S_i}^T & 0 \end{bmatrix}, \forall i;$

$S = U, D; \quad \mathbf{G}_{I_j} \triangleq \begin{bmatrix} \text{diag}(\mathbf{e}_j) & -\frac{1}{2} \mathbf{e}_j \\ -\frac{1}{2} \mathbf{e}_j^T & 0 \end{bmatrix}, \forall j。$

式(8)属于非凸 QCQP 问题^[12], 该问题总体来说仍然属于 NP 难题。为获得近似解, 采用文献[13]中的二次优

化思路将式(8)松弛为 SDR。具体来说,定义 $\mathbf{Z}_i \triangleq \mathbf{z}_i \mathbf{z}_i^T$ 。通过去除秩约束 $\text{rank}(\mathbf{Z}_i)=1$, 可以获得如下的 SDR 问题:

$$\begin{cases} \min_{\{\mathbf{Z}_i\}} \sum_{i=1}^N \text{Tr}(\mathbf{G}_i \mathbf{Z}_i) \\ \text{约束:} \begin{cases} \text{Tr}(\mathbf{G}_{l_i}, \mathbf{Z}_i) \leq -\sum_{j=1}^M T_{l_j}, \forall i; & \text{Tr}(\mathbf{G}_{r_i}, \mathbf{Z}_i) \leq 0, \forall i, r=c, u, d \\ \sum_{i=1}^N \text{Tr}(\mathbf{G}_{u_i} \mathbf{Z}_i) \leq C_{UL}; & \sum_{i=1}^N \text{Tr}(\mathbf{G}_{d_i} \mathbf{Z}_i) \leq C_{DL} \\ \text{Tr}(\mathbf{G}_{l_j}, \mathbf{Z}_i) = 0, \forall i, j; & \mathbf{Z}_i(M+6, M+6) = 1, \forall i, \mathbf{Z}_i \geq 0, \forall i \end{cases} \end{cases} \quad (9)$$

利用文献[16]中标准的 SDR 软件 SeDuMi, 可以在多项式时间内高效获得上述 SDP 问题的最优解 $\{\mathbf{Z}_i^*\}$ 。确定 $\{\mathbf{Z}_i^*\}$ 之后, 即可利用 $\{\mathbf{Z}_i^*\}$ 获得式(2)原始问题秩为 1 的解。紧接着, 文中提出一种多用户多任务卸载(Multi-User Multi-Task Offloading, MUMTO)算法获得二元卸载决策 $\{x_{ij}\}$ 和相应的通信资源最优分配方案 $\{r_i\}$ 。

2.3 二元卸载决策和资源分配

定义 $\mathbf{x} \triangleq [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]^T$, 其中 $\mathbf{x}_i \triangleq [x_{i1}, x_{i2}, \dots, x_{iM}]$, $\forall i$ 。注意到, 只需要获得位于 $\mathbf{Z}_i^*$ 左上方的 $M \times M$ 子矩阵 $(\mathbf{Z}_i^*)$, 便可获得解 $\mathbf{x}$ 。其次, 每个 $\mathbf{Z}_i^*$ 中的每个对角元素都在 0~1 之间。定义 $\mathbf{p}_i = [p_{i1}, p_{i2}, \dots, p_{iM}] \triangleq \text{diag}(\mathbf{Z}_i^*)$, 有 $p_{ij} \in [0, 1], \forall i, j$ 。利用 $\mathbf{p}_i$ 即可确定可行决策 $\mathbf{x}_i^O$, 其中 $x_{ij}^O = \text{round}(p_{ij})$ 表示四舍五入后的结果, 整个卸载决策结果表示为 $\mathbf{x}^O = [\mathbf{x}_1^O, \mathbf{x}_2^O, \dots, \mathbf{x}_N^O]^T$ 。

确定卸载决策 $\mathbf{x}^O$ 后, 问题(2)转化为通信资源分配优化问题 $\{r_i\}$, 且:

$$\begin{cases} \min_{\{r_i\}} \left(E + \sum_{i=1}^N \rho_i \max\{T_{l_i}, T_{c_i}^U\} \right) \\ \text{约束:} \sum_{i=1}^N c_{u_i} \leq C_{UL}; \quad \sum_{i=1}^N c_{d_i} \leq C_{DL}; \quad c_{u_i}, c_{d_i} \geq 0, \forall i \end{cases} \quad (10)$$

式中 $E \triangleq \sum_{i=1}^N \sum_{j=1}^M (E_{l_{ij}}(1-x_{ij}) + E_{c_{ij}} x_{ij})$ 表示 $\{x_{ij}\}$ 给定后数值保持不变的常量。式(10)中的资源分配问题为凸问题, 利用标准的凸优化问题求解方法便可获得该问题的最优解。在实践中为了获得最优卸载决策, 需要将 $\mathbf{x}^O$ 与纯本地处理决策和纯云端处理决策做比较, 选择使式(2)总体系统成本最小化的决策作为最终的卸载决策 $\mathbf{x}^S$ 。

采用内点算法^[17]可在 $O(\sqrt{MN} \log(1/\epsilon))$ 次迭代内获得式(9)中 SDP 问题的精确度为 ϵ 的解, 且每次迭代的计算量为 $O(M^6 N^4)$ 。另外, 做出卸载决策后, 可采取随机次序对多个任务进行卸载, 获得的 T_{c_i} 小于 $T_{c_i}^U$ 。

2.4 最优解的下界

原始优化式(2)中的成本函数考虑了最坏情况下的数据传输, 并处理了式(6)中所有用户的延时。为了确定最优解的下界, 引入新的优化问题, 其中的 $T_{c_i}^L$ 可表示为:

$$\begin{cases} \min_{\{x_{ij}\}, \{r_i\}} \sum_{i=1}^N \left[\sum_{j=1}^M (E_{l_{ij}}(1-x_{ij}) + E_{c_{ij}} x_{ij}) + \rho_i \max\{T_{l_i}, T_{u_i}, T_{d_i}, T_{uac_i}, T_{dac_i}, T_{c_i}\} \right] \\ \text{约束:} \sum_{i=1}^N c_{u_i} \leq C_{UL}; \quad \sum_{i=1}^N c_{d_i} \leq C_{DL}; \quad c_{u_i}, c_{d_i} \geq 0, \forall i; \quad x_{ij}(x_{ij}-1) = 0, \forall i, j \end{cases} \quad (11)$$

式(11)中的优化问题仍然属于非凸问题, 因此可以将其表述为与式(9)类似的 SDR 问题, 鉴于篇幅所限, 此处略去。在卸载决策和通信资源分配方案相同的条件下, 该优化问题的目标函数确定的成本总是低于真实成本。因此, 它可看成是式(2)中初始优化问题最小系统成本的下界, 即最优解的下界, 将其作为性能基准来对本文算法与其他算法进行比较分析。

3 性能评估

通过 MATLAB 仿真, 在不同参数设置条件下评估 MUMTO 算法的性能。除非另外说明, 采用如下默认参数

为 $\frac{1}{730 \times 10^6}$ J/cycle。当 $APP(ij)$ 采用 X264 CBR 编码程序(1 900 周期/字节)时,计算本地时间消耗为 4.75×10^{-7} s/bit,本地处理能耗为 4.75×10^{-7} J/bit。每个任务的输入数据规模和输出数据规模分别服从 10~30 MB 和 1~3 MB 之间的均匀分布。另外,移动用户和接入点间的上行和下行传输容量为 150 Mbps(IEEE 802.11n),移动用户的传输和接收能耗均为 1.42×10^{-7} J/bit,云端分配给每个用户的各个服务器的 CPU 速率为 10×10^9 cycle/s。当任务卸载给云端时,传输速率为 15 Mbps。系统效用成本为:

$$C_{c_i} = D_{in}(i) + \alpha_1 / f_c + \alpha_2 / C_{UL} + \alpha_3 / C_{DL} \quad (12)$$

式中: $\alpha_1 = 10^{18}$ bit cycles/s; $\alpha_2 = \alpha_3 = 10^{16}$ bit $\times$ Mbps。它是关于输入数据规模、云端 CPU 速率、上行和下行链路容量的函数。同时, $\beta = 2.5 \times 10^{-7}$ J/bit。设置每个用户任务处理延时的权重为 $\rho_i = 1$ J/s。

文中选择如下算法进行比较:所有任务均由移动用户自己处理的纯本地处理算法;所有任务均卸载到云端的纯云端算法;式(11)中 SDR 问题最优解的下界。所有仿真结果均为各个任务输入和输出数据 100 次仿真的平均值。

图 2 给出了系统总体成本和系统效用成本权重 β 之间的关系。可以看到, MUMTO 算法的总体成本要低于本地处理算法和云端处理算法,接近于最优解的下界。仔细分析其原因可知,这是因为 MUMTO 算法中考虑了卸载决策及所有用户的通信资源共享问题对系统性能的影响,通过求解非凸的具有二次约束的二次规划问题对任务卸载决策和通信资源分配进行了联合优化,实现了通信资源的最优分配。

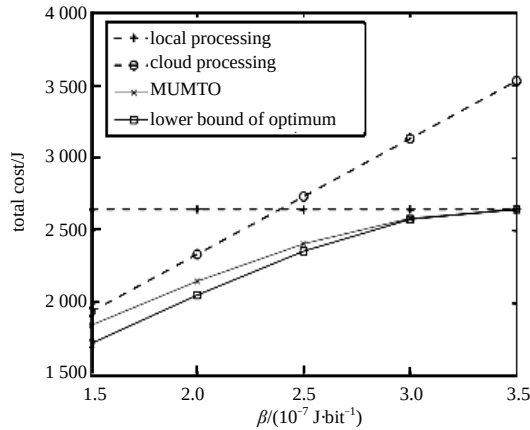


Fig.2 Total system cost vs. β
图2 系统总体成本和 β 之间的关系

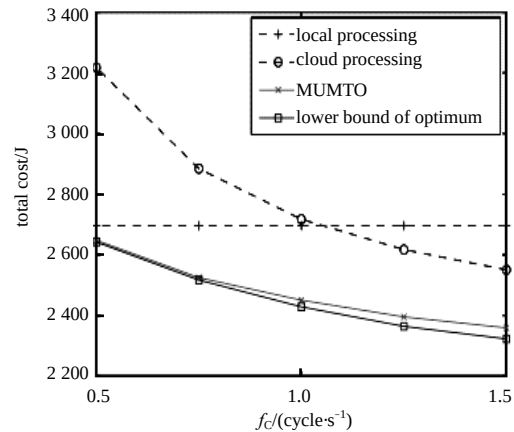


Fig.3 Total system cost vs. cloud CPU rate f_c
图3 总体系统成本和云端 CPU 速率 f_c 之间的关系

随着 β 的增加,所有任务由移动用户自己处理的概率更大。此时, MUMTO 和最优解的下界趋向于纯本地处理算法。虽然云端可增加计算容量,云端的处理时间取决于分配给每个用户的 CPU 速率 f_c 。图 3 给出了总体系统成本和 f_c 之间的关系。如预期,每个用户分配到的云端 CPU 的性能越高,系统性能越高;当单个用户的云端 CPU 太低以致无法发挥作用时, MUMTO 趋向于纯本地处理算法。

图 4 给出了延时权重 ρ 不同时系统成本的变化情况。发现 MUMTO 算法的性能接近最优,且远优于其他算法。最后,考察了 MUMTO 算法的可拓展性。图 5 和图 6 分别给出了系统成本与用户数量 N 和各个用户任务数量 M 之间的关系。可以看出,两个图中 MUMTO 算法接近于最优解下界,这充分表明了 MUMTO 算法在所有 N 和 M 个数值条件下的性能均接近最优。

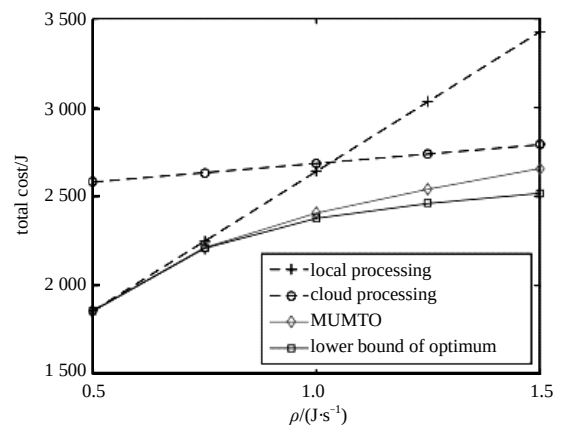


Fig.4 Total cost under different policies vs. ρ
图4 不同算法的总体成本与 ρ 之间的关系

4 结论

本文对包括多个用户和单个远端云服务器的移动云计算系统进行了研究。为使所有用户的能耗、计算成本

和延时成本的加权和最小化,将该问题建模为非凸的 QCQP 问题,并提出了基于 SDR 的近似算法来获得优化的任务卸载决策和通信资源分配方案。与最小成本下界的比较结果表明,MUMTO 算法的性能近似最优。在下一步工作中,将对任务卸载过程中可能出现的隐私泄漏问题进行分析,研究面向隐私保护的任务卸载和资源分配机制。

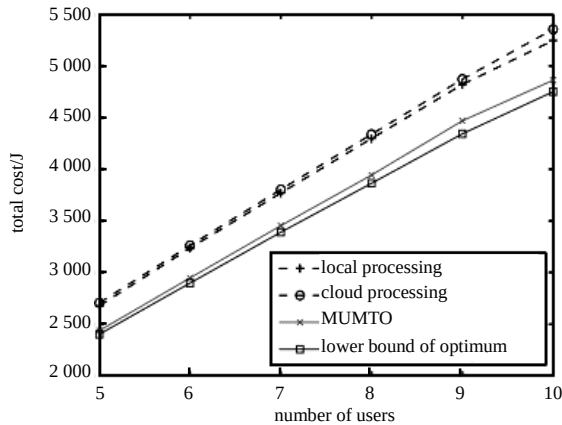


Fig.5 Total cost under different policies vs. number of users
图 5 不同算法的总体成本与用户数量之间的关系

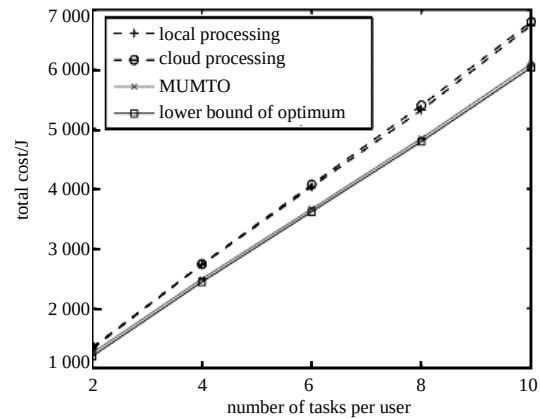


Fig.6 Total cost under different policies vs. number of tasks per user
图 6 不同算法的总体成本与各个用户任务数量之间的关系

参考文献:

- [1] 崔勇,宋健,缪葱葱,等. 移动云计算研究进展与趋势[J]. 计算机学报, 2017,40(2):273-295. (CUI Yong, SONG Jian, MIAO Congcong, et al. Mobile cloud computing research progress and trends[J]. Chinese Journal of Computers, 2017,40(2): 273-295.) DOI:10.11897/SP.J.1016.2017.00273.
- [2] 甘泉,郑均辉. 云计算中基于 QoS 的调度优化算法[J]. 太赫兹科学与电子信息学报, 2015,13(3):503-506. (GAN Quan, ZHENG Junhui. Optimized scheduling algorithm in cloud computing based on QoS[J]. Journal of Terahertz Science and Electronic Information Technology, 2015,13(3):503-506.) DOI:10.11805/TKYDA201503.0503.
- [3] LI Y, CHEN M, DAI W, et al. Energy optimization with dynamic task scheduling mobile cloud computing[J]. IEEE Systems Journal, 2017,11(1):96-105.
- [4] CARVALHO G H S, WOUNGANG I, ANPALAGAN A, et al. Intercloud and HetNet for mobile cloud computing in 5G systems: design issues, challenges, and optimization[J]. IEEE Network, 2017,31(3):80-89.
- [5] CUERVO E, BALASUBRAMANIAN A, CHO D, et al. MAUI: making smartphones last longer with code offload[C]// Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys). San Francisco, California, USA: ACM, 2010:49-62.
- [6] CHUN B G, IHM S, MANIATIS P, et al. Clonecloud: elastic execution between mobile device and cloud[C]// Proceedings of the Sixth Conference on Computer Systems. Alzburg, Austria: ACM, 2011:301-314.
- [7] KOSTA S, AUCINAS A, HUI P, et al. Thinkair: dynamic resource allocation and parallel execution in the cloud for mobile code offloading[C]// Proceedings of the IEEE INFOCOM 2012. Orlando, Florida, USA: IEEE, 2012:945-953.
- [8] EMAN MAHMOODI S, SUBBALAKSHMI K P, SAGAR V. Cloud offloading for multi-radio enabled mobile devices[C]// 2015 IEEE International Conference on Communications (ICC). London, United Kingdom: IEEE, 2015:5473-5478.
- [9] CHEN M H, LIANG B, DONG M. A semidefinite relaxation approach to mobile cloud offloading with computing access point[C]// 2015 IEEE 16th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC). Stockholm, Sweden: IEEE, 2015:186-190.
- [10] CHEN X, JIAO L, LI W, et al. Efficient multi-user computation offloading for mobile-edge cloud computing[J]. IEEE/ACM Transactions on Networking, 2016,24(5):2795-2808.
- [11] DENG S G, HUANG L T, TAHERI J, et al. Computation offloading for service workflow in mobile cloud computing[J]. IEEE Transactions on Parallel and Distributed Systems, 2015,26(12):3317-3329.
- [12] YOU C S, 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.
- [13] WANG P, SHEN C, VAN DEN HENGEL A, et al. Large-scale binary quadratic optimization using semidefinite relaxation and applications[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017,39(3):470-485.

(下转第 1104 页)