



研究与开发

基于边-端协同的任务卸载资源分配联合优化算法

吴柳青, 朱晓荣

(南京邮电大学通信与信息工程学院, 江苏 南京 210023)

摘要: 为了解决移动终端资源有限的问题, 提升任务处理效率, 提出了基于边-端协同的任务卸载与资源分配算法, 以任务完成总效益最大化为目标, 以业务 QoS 保证和资源受限为约束, 形成了基于李雅普诺夫理论的最优任务卸载资源分配问题, 由于该问题是 NP 难问题, 因此提出了解将其耦为采用 KKT 条件求解的信道资源分配问题和关于任务分配的 0-1 整数规划问题。与传统算法相比, 所提算法的复杂度是 $O(mn)$, 并提高了约 20% 的效益, 降低了 15% 以上的时延。

关键词: 移动边缘计算; 任务卸载; 资源分配; 李雅普诺夫理论

中图分类号: TN929.5

文献标识码: A

doi: 10.11959/j.issn.1000-0801.2020044

Joint optimization algorithm for task offloading resource allocation based on edge-end collaboration

WU Liuqing, ZHU Xiaorong

School of Communication and Information Engineering,
Nanjing University of Posts and Telecommunications, Nanjing 210023, China

Abstract: In order to solve the problem of limited mobile terminal resources and improve task processing efficiency, a task offloading resource allocation algorithm based on edge-end collaboration was proposed. With the goal of maximizing the total benefit of tasks, the optimal task offloading resource allocation problem based on Lyapunov theory was formed by the constraint of service QoS guarantee and resource limitation. Since the problem is NP-hard, it was decoupled into a channel resource allocation problem solved by KKT condition and a 0-1 integer programming problem about task assignment. Compared with the traditional algorithm, the complexity of the proposed algorithm was $O(mn)$, the efficiency was improved by about 20%, and the delay was reduced more than 15%.

Key words: mobile edge computing, task offloading, resource allocation, Lyapunov theory

1 引言

随着高清移动视频、AR/VR、车联网、智能制造等业务的快速发展, 要求网络能够提供超低

时延、高速率、可定制化等特定能力, 这对以 5G 为代表的下一代移动通信网络提出严峻挑战。移动边缘计算 (mobile edge computing, MEC) 将云计算技术融入移动通信系统, 把移动网关功能以

及一些第三方应用和内容部署到网络边缘,形成移动边缘云。将任务卸载至边缘云处理,减少核心网压力的同时,降低任务处理时延,弥补了云计算时延较长和网络带宽成本较高的缺点,并且减少移动终端的能量消耗^[1]。但是,对于由多个 MEC 服务器组成的边缘云来说,在保证系统性能的前提下,解决是否卸载任务、将任务卸载至哪个 MEC 服务器进行处理仍是具有挑战性的问题。

针对移动边缘计算中的任务卸载和资源调度问题,国内外学者进行了深入的研究。参考文献[2]提出了一种基于边缘计算 IoT 应用程序 awaRE 工作负载分配 (AREA) 方案,以尽量减少用户应用程序的总响应时间。参考文献[3]考虑了单用户单 MEC 任务卸载的场景,研究了在多通道干扰环境中使用集中式云和 MEC 协同计算卸载的问题。参考文献[4]采用以设备为中心的方法,并设计一种资源节约型计算卸载算法,在满足用户服务质量 (quality of service, QoS) 要求的前提下,尽可能少地占用云资源。参考文献[5]研究的是单用户单 MEC 的场景,将卸载决策问题制定为多标签分类问题,开发了深度监督学习方法来最小化计算和卸载开销。但是以上文献考虑的场景比较单一,对于多用户计算卸载问题,参考文献[6]考虑了多用户单 MEC 服务器的场景,为了最小化移动终端的能耗,协同优化了卸载决策、无线资源分配和计算资源分配。参考文献[7]以分布式方式使用博弈论方法设计了一种有效的计算卸载模型,针对多用户计算卸载引起的计算卸载的 NP 难问题,通过实现多用户计算卸载博弈的纳什均衡提供了解决方案。参考文献[8]提出了边缘云任务缓存和卸载的联合优化问题,并利用基于交替迭代算法的任务缓存和卸载算法求解。当任务卸载请求超过 MEC 服务器负载时,参考文献[9]提出采用排队论的方法,制定一个基于李雅普诺夫理论的最优任务调度策略,以最大限度地减少该系统中每个任务的平均传输数据量。参考文献[10]研究了多个

MEC-BS 进行协作,将额外任务卸载到与其连接的其他 MEC-BS 来增强 MEC 计算卸载服务。参考文献[11]考虑基于正交频分复用接入 (OFDMA) 的多用户多 MEC 服务器系统,通过双层优化方法,将原始的 NP 难问题解耦为寻求功率和子载波分配以及任务卸载问题。用延迟限制来近似下层问题,并用拉格朗日方法找到最优功率和子载波分配。在上层问题中,根据条件和约束从所有卸载策略中选择关于最小总能耗目标的最优卸载策略。参考文献[12]提出的 MEC-ABR 视频传输策略考虑无线和存储资源,基于基站的流量负载,MEC 服务器决定每个基站占据的存储资源和缓存策略,但是并未考虑用户对不同业务的 QoS 需求,将系统带宽均匀分配给移动终端,这种过度简化的频谱分配方案没有考虑动态无线信道条件,使得计算卸载过程非常低效。

不同于以上参考文献,本文研究了多用户多 MEC 场景下的任务卸载资源分配问题,设计目标是有效地利用移动边缘云的计算资源,尽可能减小用户完成任务时延,提高用户满意度。为了开发一个有效的解决方案,本文共同考虑了两个嵌套的子问题,即给定卸载策略下的资源分配问题和基于优先级的任务卸载决策优化问题,主要贡献如下。

(1) 提出了多用户多 MEC 的计算卸载框架。考虑用户功率限制,根据 MEC 计算资源、信道占用情况、任务优先级等因素制定任务卸载资源调度联合优化问题。

(2) 提出了基于李雅普诺夫理论的最优任务卸载资源分配策略,以最大限度地提高用户侧的总收益。当用户有海量任务进行计算卸载时,超过各 MEC 的负载,采用排队论的方法解决队列拥塞。

(3) 由于优化问题是 NP 难问题,将最优化问题分解成两个嵌套的子问题,采用 KKT 条件来求解信道资源分配问题。基于此,提出了多用户



多 MEC 计算卸载算法 (multi-user multi-MEC task offloading assignment algorithm, MMTCAA) 求解关于任务卸载的 0-1 整数规划问题。

2 系统模型

本文研究的场景如图 1 所示, 一组密集部署的基站上均配备 MEC 服务器升级为 MEC 基站构成移动边缘云。本文考虑由 J 个 MEC 服务器组成的边缘云为 K 个用户提供计算卸载服务。在保证系统性能的条件下, 需解决是否卸载任务、将任务卸载至哪个 MEC 服务器的问题。用户只能将任务卸载至本身直接通信范围内的 MEC 服务器, 以扩展自己的计算能力并降低能耗。MEC 服务器虽然可以在不同时间接收来自不同用户的任务, 但一次只能接受一个任务的计算卸载请求。将 $J(k_i)$ 表示为向用户 k_i 提供服务的 MEC 服务器的集合。用户和 MEC 基站之间采用无线通信链路进行数据传输, 不同的无线链路采用正交信道传输, 并不会受到来自其他无线链路的干扰。根据用户的位置和任务工作量, 用户可以选择多个边缘云进行计算卸载。例如, 可以按以下方式安排任务

计算卸载, 用户 1 选择边缘云 1 进行计算卸载, 用户 2 选择边缘云 1、边缘云 2、边缘云 3 进行计算卸载, 用户 3 可选择边缘云 4 进行计算卸载。

2.1 任务模型

假设每个用户有不同优先级的任务需要本地执行或卸载至 MEC 服务器处理。本文考虑所有用户有 $M = \{m_1, m_2, m_3, \dots, m_n\}$ 个任务需要处理, 采用广泛使用的任务模型来描述任务 m_i 大小: $D_i(t)$ (bit), 即任务 m_i 的数据分组大小。任务 m_i 需处理的工作量为 $D_i(t)X_i$, 其中 X_i 表示处理 1 bit 数据量所需的 CPU 周期 (CPU cycle)。

MEC 基站提供的无线资源由其覆盖范围内的移动终端共享, 不考虑由计算卸载引起的基站间干扰和基站内干扰。假设 MECj ($j \in J(k_i)$) 拥有的频谱带宽为 $B_j(t)$, 可分为 $N_j(t)$ 个子载波, 每个子载波带宽为 $\frac{B_j(t)}{N_j(t)}$ 。根据香农公式, 子载波 n

上任务 m_i 的数据传输速率为:

$$r_{i,n,j}(t) = \frac{B_j(t)}{N_j(t)} \lg \left(1 + \frac{p_i h_{n,j}}{\sigma^2} \right) \quad (1)$$

任务 m_i 的总数据传输速率为:

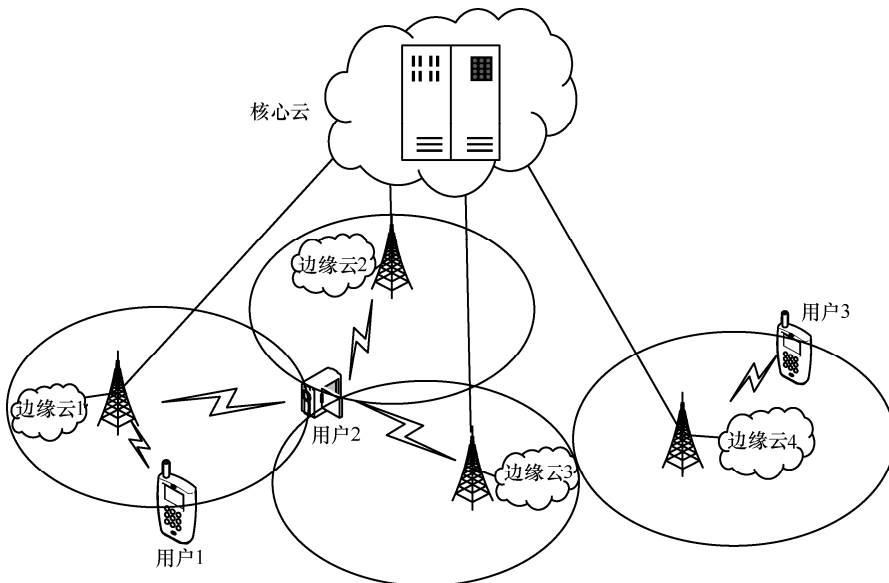


图1 多用户多 MEC 计算卸载场景

$$R_{i,j}(t) = \sum_{n=1}^{N_j(t)} \pi_{i,n}(t) r_{i,n,j}(t) = \sum_{n=1}^{N_j(t)} \frac{B_j(t)}{N_j(t)} \log \left(1 + \frac{p_i h_{n,j}}{\sigma^2} \right) \quad (2)$$

其中, $\pi_{i,n}(t)$ 是信道分配指示符, 当 $\pi_{i,n}(t)=1$ 时, 表示子载波 n 分配给任务 m_i 进行卸载; 当 $\pi_{i,n}(t)=0$ 时, 表示子载波 n 未分配给任务 m_i 进行卸载。 $h_{n,j}$ 表示用户子载波 n 的信道增益, 假设任务卸载期间用户的移动性不高, 所以 $h_{n,j}$ 是一个常数; σ^2 是信道噪声功率。

2.2 任务计算卸载模型

当用户有任务需要处理时, 有两种处理模式可选择: 本地执行或卸载至 MEC 服务器执行。本地执行不需占用通信资源, 但本地设备的 CPU 能力受限, 而卸载至 MEC 服务器处理, 尽管 MEC 服务器的 CPU 能力更强, 但卸载增加了传输时延。下面将详细讨论本地执行模型和 MEC 服务器执行模型。

2.2.1 本地执行模型

用户将任务留在本地处理时, 时延只包含任务处理时间。考虑移动用户终端的 CPU 处理能力为 f_i^l , 则本地执行时延为:

$$T_{i,j}^{le}(t) = \frac{D_i(t)X_i}{f_i^l} \quad (3)$$

2.2.2 MEC 服务器执行模型

考虑到不同 MEC 服务器的计算资源差异, 其 CPU 处理能力不一致。根据前文所述通信模型, 将任务卸载至 MEC 服务器执行时延由以下 3 部分组成: 任务卸载传输时间; 当有海量任务需要卸载至边缘云处理时, 超过 MEC 服务器负载, 任务可能需要在各 MEC 服务器排队等待, 即排队等待时间; 任务处理时间。

根据前文所述, 若任务 m_i 卸载至 MEC 服务器 j 的数据传输速率为 $R_{i,j}(t)$, 则卸载传输时间为:

$$T_{i,j}^{et}(t) = \frac{D_i(t)}{R_{i,j}(t)} \quad (4)$$

如果任务 m_i 卸载至 MEC 服务器 j 处理时, 服务器发生拥塞, 则排队等待时间为 $T_{i,j}^{ew}(t)$ 。具

体表达式将在后面讨论。

假设 MEC j 的 CPU 处理能力为 f_j^c , 任务 m_i 的处理时间为:

$$T_{i,j}^{ec}(t) = \frac{D_i(t)X_i}{f_j^c} \quad (5)$$

因此, 任务 m_i 卸载至 MEC 服务器 j 处执行的总时延为:

$$T_{i,j}(t) = T_{i,j}^{et}(t) + T_{i,j}^{ew}(t) + T_{i,j}^{ec}(t) \quad (6)$$

2.3 排队等待模型

对于任意一个 MEC 服务器来说, 任务的到达过程被建模为伯努利过程, 假设 MEC 服务器 j 的任务到达率为 λ_j 。当任务到达时, 任务首先进入具有无限容量的任务队列。在队列中等待的任务数量假设为队列状态: $Q_j(t) = \{1, 2, 3, \dots\}$ 。MEC 服务器 j 的队列 $Q_j(t)$ 更新为:

$$Q_j(t) = Q_j(t+1) - V_j(t) + A_j(t) \quad (7)$$

其中, $V_j(t)$ 表示 MEC 服务器 j 处任务的处理速度, 即在时刻 t 长度为 1 的时间内处理完成了 $V_j(t)$ 个任务。 $A_j(t)$ 表示在时刻 t 任务是否到达, $A_j(t) \in \{0, 1\}$ 。因此, 有 $\Pr\{A_j(t)=1\} = \lambda_j$ 且 $\Pr\{A_j(t)=0\} = 1 - \lambda_j$ 。

如第 2.2 节所述, 卸载至 MEC 服务器处理任务的时延包括卸载传输时延、排队等待时延和处理时延。基于 Little's Law (律特法则), 包括排队等待时延和处理时延在内的执行延迟与任务缓冲区的平均队列长度成正比。平均队列长度为:

$$\lim_{T \rightarrow \infty} \frac{1}{T} E \left[\sum_{t=0}^T Q_j(t) \right] \quad (8)$$

3 边-端协同的任务调度与资源分配

将移动用户终端和 MEC 服务器看作地位相同, 但具有不同计算开销的处理单元。移动终端通过无线接入站点接入网络, 考虑通过移动终端与 MEC 服务器之间的协同任务调度, 将一部分任务留在本地处理, 另一部分卸载至 MEC 服务器处理。为方便研究, 本文假设同一任务的业务数据



只能交给一个 MEC 服务器处理，而 MEC 服务器在任一时刻只能处理一个任务的业务数据，有新任务到达时，将被放入 MEC 服务器的任务缓存队列中等待处理。通过研究终端侧与边缘侧的协作，均衡 MEC 服务器和终端的工作负载，实现系统资源利用最大化，提出以最大化用户侧总收益为目标的联合调度策略，以提升用户服务体验。

3.1 多用户多 MEC 联合调度

本文提出的多用户多 MEC 服务器联合调度模式任务卸载如图 2 所示。假设目前小区中共有 K 个 MEC 服务器，所有 MEC 服务器的计算能力不尽相同。移动终端首先将任务卸载请求发送至 MEC 控制器上行缓存队列，MEC 控制器收到一段时间内到达的用户卸载请求后，将根据任务的业务数据大小、对计算资源的需求量以及当前网络资源和 MEC 服务器任务队列情况，分别计算任务本地处理和卸载到不同 MEC 服务器处理的时延以及系统开销。以最大化一段时间内用户的总收益为目标执行多用户多 MEC 联合调度策略，实现任务调度决策，有效利用了边缘云侧和用户侧的资源来解决两个子问题：任务是否需要卸载和任务需卸载到哪个 MEC 服务器处理。需要卸载到 MEC 服务器执行的任务由相应的移动终端通过接入站点将该业务数据上传到相应的 MEC 服务器对应的缓存队列中等待 MEC 服务器执行；需要在本地执行的任务则留在本地等待执行。

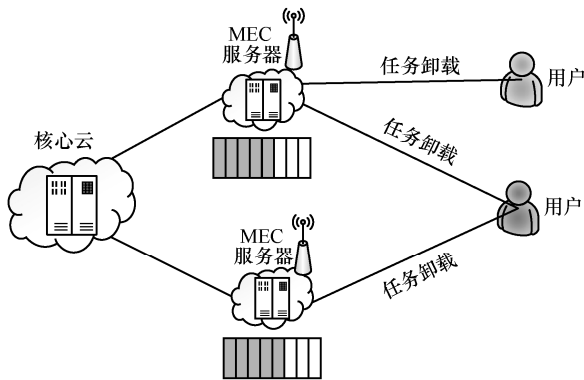


图 2 多用户多 MEC 服务器联合调度模式

3.2 优化问题

本文考虑的优化目标是最大化一段时间内的用户侧总收益。对移动终端来说，完成不同任务可产生不一样的收益，收益主要由两方面因素决定：任务自身属性（比如优先级）产生的期望收益值、任务处理过程中产生的时延损失。假设 u_i 表示任务 m_i 根据其优先级制定的期望收益。 $L(T_i)$ 表示任务 m_i 完成过程中付出的时延损失，如式 (9) 所示：

$$L(T_i) = \begin{cases} C(T_i - \rho_i), & T_i \geq \rho_i \\ 0, & T_i < \rho_i \end{cases} \quad (9)$$

其中， C 为比例系数，是常数。 ρ_i 为损耗对时延的容忍度，当时延小于 ρ_i 时，一定的时延对用户满意度不会造成影响，即不会损耗用户的收益，当时延大于 ρ_i 时，时延对用户满意度造成了影响，相应地产生了时延损耗。

任务 m_i 卸载至 MEC 服务器处理产生的收益为 $U_{i,j}^E(t) = u_i - L(T_{i,j}(t)) = u_i - L(T_{i,j}^{rel}(t) + T_{i,j}^{nw}(t) + T_{i,j}^{re}(t))$ ，任务 m_i 在本地执行产生的收益 $U_i^L(t) = u_i - L(T_i(t)) = u_i - L(T_i^l(t))$ 。由所述表达式可知，时延损失函数为线性，任务收益值与其执行时延成反比，时延越大，实际收益值越小。因此，优化问题如下：

$$P1: \max \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^T \sum_i \sum_j (U_{i,j}^E(t) s_{i,j}(t) + U_i^L(t) (1 - s_{i,j}(t))) \quad (10)$$

$$\text{s.t.} \quad \sum_j s_{i,j}(t) \leq 1, \quad \forall i \in M, \forall j \in J \quad (11)$$

$$s_{i,j}(t) \in \{0, 1\}, \quad \forall i \in M, \forall j \in J \quad (12)$$

$$\pi_{i,n}(t) \in \{0, 1\}, \quad \forall i \in M, 1 \leq n \leq N_j(t) \quad (13)$$

$$\sum_i \pi_{i,n}(t) \leq 1, \quad \forall i \in M, 1 \leq n \leq N_j(t) \quad (14)$$

$$0 \leq p_i \leq p^{\max}, \quad \forall i \in M \quad (15)$$

$$p_i t \leq E_i^{\max}, \quad \forall i \in M \quad (16)$$

$$T_{i,j}(t) s_{i,j}(t) + T_i^l(t) (1 - s_{i,j}(t)) \leq \tau_i^{\max}, \quad \forall i \in M, \forall j \in J \quad (17)$$

4 问题分析与求解

在本节中, 提出了采用李雅普诺夫方法^[13]将优化问题转化为稳态的最优化问题, 因为最优化问题是一个 NP 难问题, 将该问题分为两个嵌套的子问题: 采用 KKT 条件求解给定任务卸载分配策略下的信道资源分配问题和给定信道资源分配策略下关于任务分配的 0-1 整数规划问题。

4.1 稳态条件下问题的形成

由于每个任务的期望收益 u_i 为常数, 式 (10) 可简化为:

$$\min \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^T \sum_i \sum_j (L(T_{i,j}(t))s_{i,j}(t) + L(T_i(t))(1-s_{i,j}(t))) \quad (18)$$

由第 3 节可知, 时延损失函数 $L(\cdot)$ 是线性函数, 所以可以进一步简化为:

$$\text{P2: } \min \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^T \sum_i \sum_j (T_{i,j}(t)s_{i,j}(t) + T_i(t)(1-s_{i,j}(t))) \quad (19)$$

本文考虑的场景中有 J 个 MEC 服务器, 所以系统拥有 J 个队列, 令 $Q(t) = (Q_1(t), Q_2(t), \dots, Q_j(t), \dots, Q_J(t))$ 表示队列状态。假设各个队列的任务到达状态为伯努利过程, $Q(t)$ 根据任务到达率 λ_j 在时隙 $t \in \{0, 1, 2, \dots\}$ 上演化。为了进一步简化求解问题, 定义二次李雅普诺夫函数为:

$$F(Q(t)) \triangleq \frac{1}{2} \sum_{j=1}^J \omega_j Q_j^2(t) \quad (20)$$

ω_j 表示权重集合。不同的权重会导致不同队列在任务调度策略中所处的地位不同, 本文假设所有的 ω_j 都为 1。显然, 该李雅普诺夫函数是非负的, 当且仅当所有 $Q_j(t)$ 为 0 时, $F(Q(t))$ 等于 0。考虑到队列 j 在时刻 t 的状态为 $Q_j(t)$, 为了预测各个队列状态的变化, 定义一个时刻间的二次李雅普诺夫函数的差值的均值为李雅普诺夫漂移函数 $\Delta(Q(t))$:

$$\Delta(Q(\tau)) \triangleq E\{F(Q(t+1)) - F(Q(t)) | Q(t)\} \quad (21)$$

这种漂移是李雅普诺夫函数在一个时刻上的预期变化。

最优化问题是最大化一段时间内总收益的时间平均, 为了简化问题, 要求所有队列都是稳定的。接下来将证明队列的稳定性。

定理 1 (Lyapunov 漂移) 考虑二次李雅普诺夫函数, 假设 $E\{F(Q(0))\} < \infty$ 。假设存在常数 $\beta > 0, \varepsilon \geq 0$ 在所有时隙 $\tau \in \{0, 1, 2, \dots\}$, 所有 $Q(\tau)$ 都满足如下漂移条件:

$$\Delta(Q(t)) \leq \beta - \varepsilon \sum_{j=1}^J |Q_j(t)| \quad (22)$$

然后, 如果 $\varepsilon \geq 0$, 那么所有队列 $Q_j(t)$ 都是平均速率稳定的; 如果 $\varepsilon > 0$, 那么所有队列 $Q_j(t)$ 都非常稳定。

即满足:

$$\limsup_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{j=1}^J E\{|Q_j(\tau)|\} \leq \frac{\beta}{\varepsilon} \quad (23)$$

上述定理表明, 如果 $\varepsilon \geq 0$ 且满足漂移条件式 (22), 那么 $\Delta(Q(t)) \leq \beta$, 即所有队列都是平均速率稳定的。此外, 如果 $\varepsilon > 0$, 则所有队列都非常稳定, 时间平均预期队列长度以 $\frac{\beta}{\varepsilon}$ 为界。

假设除了想要的稳定队列 $Q(t)$ 之外, 还有一个相关的随机“惩罚”函数, 表示对时刻 t 的控制动作所引起的“惩罚”, 因为本文中简化的优化问题 P2 是最小化所有任务的总执行时延, 所以定义该函数为时延, 即 $T(t) = \sum_i \sum_j (T_{i,j}(t)s_{i,j}(t) + T_i(t)(1-s_{i,j}(t)))$ 。本文期望“惩罚”函数的时间平均值接近于目标值 T^* 。假设预期惩罚的下界受限于一个最小值 (可能是负值), 因此对于在任一时刻 t 的所有控制措施, 有:

$$E\{T(t)\} \geq T_{\min} \quad (24)$$

$T_{\min}$ 表示 $E\{T(t)\}$ 的下界, 下面进一步确定“惩罚”函数的时间平均值上界。

定理 2 (Lyapunov 最优化) 假设 $E\{T(t)\}$ 满足式 (23), 且 $E\{F(Q(0))\} < \infty$, 存在常数 $v_0 \geq 0$,



$\beta \geq 0$, $\varepsilon \geq 0$ 和 T^* , 所有 $Q(\tau)$ 在任一时刻 $\tau \in \{0, 1, 2, \dots\}$ 都满足:

$$\Delta(Q(\tau)) + v_0 E\{T(\tau) | Q(\tau)\} \leq \beta + v_0 T^* - \varepsilon \sum_{j=1}^J |Q_j(t)| \quad (25)$$

且所有队列 $Q_j(\tau)$ 都是平均速率稳定。当 $v_0 \geq 0$, $\varepsilon \geq 0$ 时, 惩罚函数的时间平均期望的上界和队列长度的上界满足:

$$\limsup_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} E\{T(\tau)\} \leq T^* + \frac{\beta}{v_0} \quad (26)$$

$$\limsup_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{j=1}^J E\{|Q_j(\tau)|\} \leq \frac{\beta + v_0(T^* - T_{\min})}{\varepsilon} \quad (27)$$

对式 (25) 进行求和, 使用伸缩和的定律得出:

$$E\{F(Q(\tau+1))\} - E\{F(Q(0))\} + v_0 \sum_{\tau=0}^{t-1} E\{T(\tau)\} \leq (\beta + v_0 T^*)t - \varepsilon \sum_{\tau=0}^{t-1} \sum_{j=1}^J E\{|Q_j(\tau)|\} \quad (28)$$

重排式 (28) 可得:

$$E\{F(Q(t))\} \leq E\{F(Q(0))\} + (\beta + v_0(T^* - T_{\min}))t \quad (29)$$

另外队列的平均速率稳定性同定理 1 中所述。

以上可理解为, 如果参数 $v_0 > 0$, 可以设计一个控制算法, 以确保在每个时刻 t 上都满足式 (25), 则惩罚函数的时间平均期望的上界满足式 (26), 因此惩罚函数的时间平均小于目标值, 或者与 T^* 相差不多, 随着 v_0 的增加, $\frac{\beta}{v_0}$ 可以任意小。但是, 队列长度的时间平均的上界随着 v_0 增大线性增加, 如式 (27) 所示。因为律特法则表明平均排队长度与平均时延成正比, 所以经常将其称为性能延迟权衡。

这个结果表明了以下控制策略: 每个时刻 t , 观察当前的 $Q(t)$ 值并采取控制动作, 根据一致的 $Q(t)$, 贪婪的最小化式 (26) 左侧的漂移加惩罚函数期望:

$$\Delta(Q(t)) + v_0 E\{T(t) | Q(t)\} \quad (30)$$

即最小化式 (31):

$$\Delta(Q(t)) + v_0 E\left\{\sum_i \sum_j (T_{i,j}(t)s_{i,j}(t) + T_i(t)(1-s_{i,j}(t))) | Q(t)\right\} \quad (31)$$

由此可得, 上文中提到的参数 v_0 表示系统对时延损失的敏感程度, 当 $v_0 = 0$ 时, 系统即不考虑时延损失, 随着 v_0 的增加, 系统对时延损失越来越敏感。

在排队模型中提到各 MEC 服务器的任务处理速度为 $V_j(t)$, $\Delta(Q(t)) = E\{F(Q(t+1)) - F(Q(t)) | Q(t)\}$,

通过计算可得 $\Delta(Q(t)) = \frac{1}{2} \sum_j (\lambda_j + 2\lambda_j Q_j(t) + E[V_j^2(t)] - 2Q_j(t)E[V_j(t)] - 2\lambda_j E[V_j(t)])$, 在某一时刻 t , MEC 服务器队列状态 $Q_j(t)$ 、任务处理速度 $V_j(t)$ 和任务到达率 λ_j 都已知, 所以 $\Delta(Q(t))$ 为常数。

因此, 当确定时延敏感参数 v_0 , 因为优化目标是 minimized 时延, 所以假设 $v_0 = 1$, 可以在每个时刻 t , 求解出 P3:

$$\text{P3: } \min \sum_i \sum_j (T_{i,j}(t)s_{i,j}(t) + T_i(t)(1-s_{i,j}(t))) \quad (32)$$

s.t. (12) ~ (19)

4.2 基于 KKT 条件的问题求解

问题 P3 是一个混合整数非线性优化问题, 为了解决 P3, 无可避免要遍历 m^n 种情况, 计算复杂度为 $O(m^n)$, 将该问题分解为以下两个子问题。

- 资源分配问题: 给定 $S'_{i,j}(t)$, 当 $S_{i,j}(t)$ 固定, P3 是一个关于 $R_{i,j}(t)$ 的凸问题。然后, 采用 KKT 条件来求解最优解 $(R^*_{i,j}(t), S'_{i,j}(t))$ 。
- 任务分配问题: 基于资源分配问题的解 $R^*_{i,j}(t)$, 任务分配问题是一个关于 $S_{i,j}(t)$ 的 0-1 整数规划问题。采用任务分配算法来获得最优解。

4.2.1 资源分配问题

定理 3 给定 $S_{i,j}(t) = S'_{i,j}(t)$, P3 是一个关于 $R_{i,j}(t)$ 的凸问题。

证明 只有当任务选择卸载至边缘云时, 才可对各个 MEC 服务器的计算资源和相应的信道

资源进行分配, 即 $S_{i,j}(t)=1$, 假设所有任务中有 l 个任务卸载至 MEC 服务器处理, 即 $S_{i,j}(t)=1$ 的个数为 l 。优化目标函数转化为:

$$f(R_{i,j}(t), S'_{i,j}(t)) = \sum_{i=1}^l \sum_{j=1}^l \left(\frac{D_i(t)}{R_{i,j}(t)} + T_{i,j}^{\text{ec}}(t) + T_{i,j}^{\text{ew}}(t) \right) \quad (33)$$

其中, $f(R_{i,j}(t), S'_{i,j}(t))$ 是关于 $R_{i,j}(t)$ 的函数。 $f(R_{i,j}(t), S'_{i,j}(t))$ 对 $R_{i,j}(t)$ 求一阶导和二阶导, 可得:

$$\frac{\partial f}{\partial R_{i,j}(t)} = \frac{-D_i(t)}{(R_{i,j}(t))^2} \quad (34)$$

$$\frac{\partial^2 f}{\partial R_{i,j}^2(t)} = \frac{2D_i(t)}{(R_{i,j}(t))^3} \quad (35)$$

由于 $f(R_{i,j}(t), S'_{i,j}(t))$ 关于 $R_{i,j}(t)$ 的二阶导均为非负, 根据参考文献[14]可知, $f(R_{i,j}(t))$ 为凸函数。只有所有约束条件为线性函数, 最优化问题为凸优化问题。根据定理 3, 可由 KKT 条件得到定理 4。

定理 4 给定 $S_{i,j}(t)=S'_{i,j}(t)$, 可获得关于 $R_{i,j}(t)$ 的最优解 $R_{i,j}^*(t)$ 。

证明 根据定理 3, 优化问题的拉格朗日函数为:

$$G(R_{i,j}(t), \mu_{i,j}) = \sum_{i=1}^l \sum_{j=1}^l \left(\frac{D_i(t)}{R_{i,j}(t)} + T_{i,j}^{\text{ec}}(t) + T_{i,j}^{\text{ew}}(t) \right) + \sum_{i=1}^l \sum_{j=1}^l \mu_{i,j} (R_{i,j}(t) - R_j^{\text{max}}(t)) \quad (36)$$

通过使用 KKT 条件, 如果 $R_{i,j}(t)$ 和 $\mu_{i,j}$ 在任意点都满足 KKT 条件, 可得到:

$$\frac{\partial f}{\partial R_{i,j}(t)} + \frac{\partial \sum_{i=1}^l \sum_{j=1}^l \mu_{i,j} (R_{i,j}(t) - R_j^{\text{max}}(t))}{\partial R_{i,j}(t)} = 0 \quad (37)$$

$$R_{i,j}(t) - R_j^{\text{max}}(t) = 0 \quad (38)$$

通过求解式 (38), 可得到最优 $R_{i,j}(t)$:

$$R_{i,j}^*(t) = R_j^{\text{max}}(t) \quad (39)$$

由此可得固定 $S_{i,j}(t)$ 的最优解:

$$f^*(R_{i,j}(t), S'_{i,j}(t)) = f(R_j^{\text{max}}(t), S'_{i,j}(t)) = \sum_{i=1}^l \sum_{j=1}^l \left(\frac{D_i(t)}{R_j^{\text{max}}(t)} + T_{i,j}^{\text{ec}}(t) + T_{i,j}^{\text{ew}}(t) \right) \quad (40)$$

4.2.2 任务分配问题

通过以上的讨论, 给定 $S_{i,j}(t)=S'_{i,j}(t)$, 可获得最优解 $R_{i,j}^*(t)$ 。基于此, 问题 P3 可转化为关于 $S_{i,j}(t)$ 的 0-1 整数规划问题。本文提出多用户多 MEC 计算卸载任务分配算法来解决这个最优化问题。首先, 重写优化问题且将其转化为 0-1 整数规划问题。为了简化问题难度, 本文优先考虑将任务卸载到 MEC 服务器处理, 求解出时延最小的 MEC 服务器 j^* , 将总时延 T_{i,j^*} 与 $(T_i + \delta)$ 进行比较, 若 $T_{i,j^*} \leq (T_i + \delta)$, 则在 MEC 服务器 j^* 处理, 反之在本地处理。

0-1 整数规划问题:

$$\text{P4: } \min \sum_i \sum_j T_{i,j}(t) s_{i,j}(t) \quad (41)$$

$$\sum_j s_{i,j}(t) \leq 1 \quad \forall i \in M, \forall j \in J \quad (42)$$

$$s_{i,j}(t) \in \{0, 1\} \quad \forall i \in M, \forall j \in J \quad (43)$$

与定理 3 证明相似, 可证明修改后的目标函数是凸函数。具体的求解算法如下所示。

算法 1 多用户多 MEC 任务卸载任务分配算法

初始化 $S = 0$;

输入 队列状态 $Q(t) = (Q_1(t), Q_2(t), \dots, Q_j(t), \dots, Q_J(t))$; MEC 服务器 CPU 处理能力 f_j^c ; 本地设备 CPU 处理能力 f_j^l ; 任务数据分组大小 $D_i(t)$

输出 卸载决策 $S = (S(0), S(1), \dots, S(k-1))$, $S(t)$ 是 $i \times j$ 的矩阵; 卸载传输速率 $R(t)$

步骤:

对 $t = 0, 1, \dots, k-1$

(1) 观察用户任务数据分组大小 $D_i(t)$, 当前卸载决策 $S(t)$, MEC 服务器队列状态 $Q(t)$;

(2) 解决问题 P3, 固定 $S(t)$, 通过 KKT 条件获得最优传输速度 $R^*(t)$

$$\text{P3: } \min \sum_i \sum_j (T_{i,j}(t) s_{i,j}(t) + T_i(t)(1 - s_{i,j}(t))) \quad (44)$$

(3) 解决问题 P4, 获得最优任务调度策略 $S^*(t)$



$$\min \sum_i \sum_j T_{i,j}(t) s_{i,j}(t) \quad (45)$$

(4) 更新 $S(t) = S^*(t)$;

(5) 对每一个任务 m_i ;

a) 计算 $T_{i,j}^*$ 和 $T_i(t)$;

b) 比较 $T_{i,j}^*$ 和 $T_i(t)$, 如果 $T_{i,j}^* \leq (T_i(t) + \delta)$,

则 $s_{i,j}(t) = 1$, 否则 $s_{i,j}(t) = 0$;

(6) 更新 $S(t)$;

(7) 更新队列状态 $Q_j(t)$:

$$Q_j(t+1) = Q_j(t) - V_j(t) + \sum_i s_{i,j}(t) \quad (46)$$

(8) 返回 S 。

5 仿真结果与性能分析

在本文中,将对所提出的多用户多 MEC 任务卸载最优化算法进行仿真估计。主要考虑以下两点:将所提的最优任务卸载方案与其他任务卸载方案进行比较;分析任务大小对任务卸载的影响。

5.1 仿真参数设置

在一次循环中,有 5 个用户共需处理 50 个任务。MEC 的数量为 5 个。共运行 100 次循环,共 5 000 个任务。任务大小为 10 kbit~1 000 Mbit。本地设备处理能力为 5~6 GHz,边缘云处理能力为 25~30 GHz,基站与用户分布在 500 m×500 m 的范围内,用户可以在 500 m 的范围内连接上基站,假设信道增益为 $127+30\log d$,其他参数设置可查阅表 1。

表 1 仿真参数设置

仿真参数	参数设置
任务数据 $D_i(t)$	10 kbit~1 000 Mbit
MEC 的 CPU 处理能力 f_j^e	25~30 GHz
本地终端 CPU 处理能力 f_i^l	1~1.5 GHz
处理 1 bit 所需 CPU 周期 X_i	1 440
任务到达率 λ_j	0.5
信道带宽 B	20 MHz
终端发射功率 P_i	0.5 W
噪声功率 σ^2	$2 \times 10^{(-13)}$ W
信道增益 $h_{j,n}$	$127+30\log d$

5.2 与其他卸载方案的比较

将本文所提最优任务调度算法与随机分配算法、全 MEC 服务器处理方案以及全本地处理方案进行比较,分析其在性能和效益以及流量上的差异,随机分配算法考虑将任务随机分配到各 MEC 服务器处理或留在本地处理,全 MEC 服务器处理方案考虑将任务卸载到最佳的 MEC 服务器执行,全本地处理方案则选择不进行任务卸载。最优任务卸载算法相较于随机分配算法、全 MEC 服务器处理和全本地处理,平均每个任务的执行时延减少了 15%、24%、33%,完成效益提高了 20%、28%、52%。具体分析结果如图 3 所示。

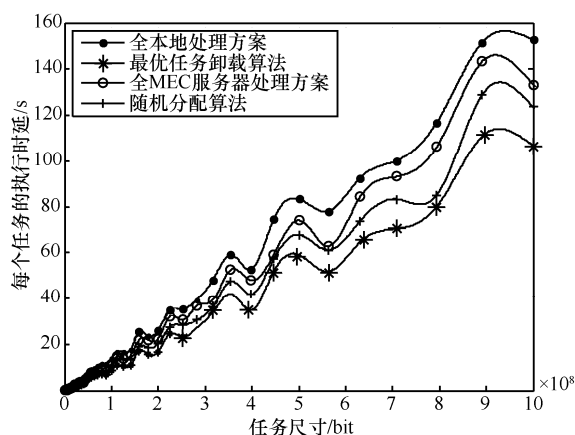


图 3 平均每个任务的时延与任务尺寸大小

最优任务调度算法卸载方案的时延最小,随机分配算法次之,全本地处理方案时延最大。随着任务尺寸越来越大,单个任务的时延随之变大,但最优任务调度算法卸载方案在时延上的恶化程度要比其他几种方案低得多。

平均每个任务的效益值与任务尺寸大小如图 4 所示。

由于本文优化的是一段时期内用户的总效益值,需对各种方案的用户效益值进行比较。图 4 中比较的是平均每个任务完成时用户可获得的效益值,由图 4 可知,最优任务调度算法卸载方案的效益值最大,随机分配算法次之,全 MEC 处理方案再次之,全本地处理的效益值最小。由于

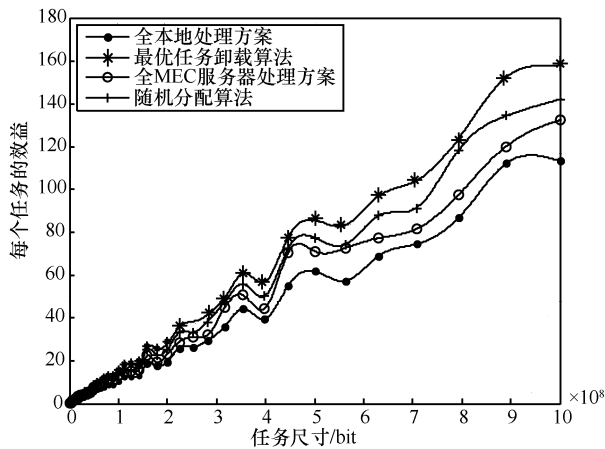


图4 平均每个任务的效益值与任务尺寸大小

本文假设的任务期望收益值与任务尺寸大小成正比,随着任务尺寸的增大,单个任务的效益值也随之增大,而且4种方案间的效益差异越来越明显。

各方案流量比较如图5所示。

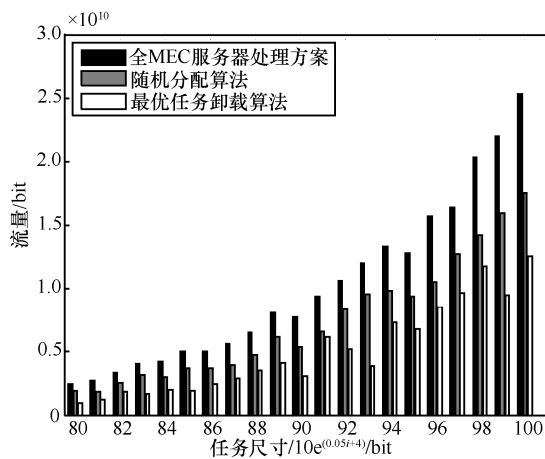


图5 各方案流量比较

因为全本地处理方案,任务不需进行卸载,所以不占用基站的无线信道资源。而最优任务调度算法卸载方案和随机分配算法以及全MEC处理方案都进行了任务卸载和任务回传,因为回传数据量小,所以本文不考虑回传。由图5可知,最优任务调度算法卸载方案比全MEC处理方案的卸载任务量减少了25%以上。可以看到,通信吞吐量随着任务尺寸增大而增大,随着MEC服务器能力的增强,越来越多的任务被卸载至MEC服务器进行处理。虽然卸载至MEC服务器处理相比

本地处理可以减少时延,但同时也增加了通信吞吐量。本文所提出的最佳方案,不仅可以实现时延最小,效用值最大,也可以节省通信吞吐量。

平均每比特数据的时延与任务尺寸大小如图6所示。

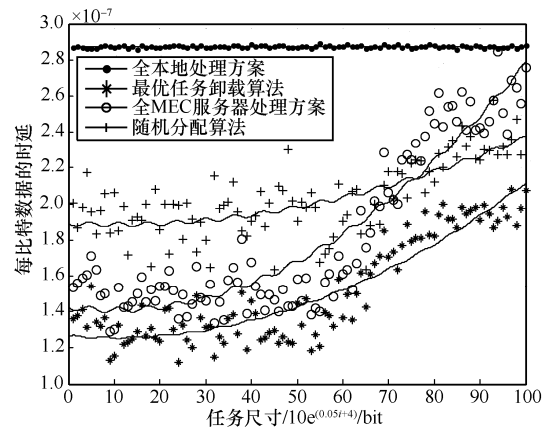


图6 平均每比特数据的时延与任务尺寸大小

由图6可知,随着任务尺寸变大,除去全本地处理方案以外的其他几个方案的平均每比特数据时延都随着任务尺寸变大而变大。随着任务尺寸变大,处理时延随之变大,当MEC服务器还未处理完上一个任务时,下一个任务已经到达,则会在MEC服务器处产生排队时延。从图6可知,当任务处理数据在10 MB以下时,全MEC处理方案仍比随机分配算法方案的时延小,而当有海量终端接入MEC服务器,且有大量数据密集型任务需要处理时,全卸载至MEC服务器进行处理不仅会造成队列拥塞,排队时延增大,用户满意度恶化,也会增加各MEC服务器承载压力。从图6可知,最优任务调度算法卸载方案可比全MEC处理方案减少15%的时延。

6 结束语

本文提出了用户效益最大化的最优任务调度策略和优化算法,以用户效益最大化为目标来分配计算资源与网络资源,确定最佳的任务调度策略。由于最优化问题是一个NP难问题,提出了采用李



雅普诺夫方法将问题转化为稳态的最优化问题,而后将该问题分为两个嵌套的子问题:资源分配问题和计算卸载决策优化问题。然后提出了多用户多MEC计算卸载任务分配算法解决计算卸载决策问题。本文所提出的算法复杂度是 $O(mn)$,远低于遍历求解的 $O(m^n)$ 。最终,求解出联合优化的资源分配策略与计算卸载决策,有效提高了用户侧的收益,降低了时延,提高了系统整体性能。

参考文献:

- [1] 谢人超, 廉晓飞, 贾庆民, 等. 移动边缘计算卸载技术综述[J]. 通信学报, 2018, 39(11): 138-155.
XIE R C, LIAN X F, JIA Q M, et al. Overview of mobile edge computing offload technology[J]. Journal of Communications, 2018, 39(11): 138-155.
- [2] FAN Q, ANSARI N. Application aware workload allocation for edge computing based IoT[J]. IEEE Internet of Things Journal, 2018, 5(3): 2146-2153.
- [3] GUO H Z, LIU J J. Collaborative computation offloading for multiaccess edge computing over fiber-wireless networks[J]. IEEE Transactions on Vehicular Technology, 2018, 67(5): 4514-4526.
- [4] CHEN X, SHI Q, YANG L. ThriftyEdge: resource-efficient edge computing for intelligent IoT applications[J]. IEEE Network, 2018, 32(1): 61-65.
- [5] CHEN M, HAO Y X. Task offloading for mobile edge computing in software defined ultra-dense network[J]. IEEE Journal on Selected Areas in Communications, 2018, 36(3): 587-597.
- [6] HAO Y X, CHEN M, HU L, et al. Energy efficient task caching and offloading for mobile edge computing[J]. IEEE Access, 2018(6): 11365-11373.
- [7] CHEN X, JIAO L, LI W, et al. Efficient multi-user computation offloading for mobile-edge cloud computing[J]. IEEE/ACM Trans. Netw., 2016, 24(5): 2795-2808.
- [8] HAO Y X, CHEN M, HU L. Energy efficient task caching and offloading for mobile edge computing[J]. IEEE Access, 2018(6): 11365-11373.
- [9] CHEN W W, WANG D, LI K Q. Multi-user multi-task computation offloading in green mobile edge cloud computing[J]. IEEE Transactions on Services Computing, 2018, 12(5): 726-738.
- [10] FAN W H, LIU Y A, TANG B H. Computation offloading based on cooperations of mobile edge computing-enabled base stations[J]. IEEE Access, 2018(6): 22622-22633.
- [11] CHENG K, TENG Y L, SUN W Q. Energy-efficient joint offloading and wireless resource allocation strategy in multi-MEC server systems[C]//Proceedings of 2018 IEEE International Conference on Communications (ICC). Piscataway: IEEE Press, 2018: 1-6.
- [12] XU X D, LIU J, TAO X. Mobile edge computing enhanced adaptive bitrate video delivery with joint cache and radio resource allocation[J]. IEEE Access, 2018(5): 16406-16415.
- [13] NEELY M. Stochastic network optimization with application to communication and queueing systems[J]. Synthesis Lectures on Communication Networks, 2010, 3(1): 211.
- [14] BOYE S, VANDENBERGHE L. Convex optimization[M]. Cambridge: Cambridge University Press, 2004.

[作者简介]



吴柳青(1995-), 女, 南京邮电大学通信与信息工程学院硕士生, 主要研究方向为移动边缘计算。



朱晓荣(1977-), 女, 博士, 南京邮电大学通信与信息工程学院教授、博士生导师, 主要研究方向为下一代无线网络、异构网络。