第 46 卷　第 3 期　　计　算　机　工　程　　2020 年 3 月
Vol. 46　No. 3　　Computer Engineering　　March 2020

·热点与综述·　　文章编号：1000-3428(2020)03-0018-06　　文献标志码：A　　中图分类号：TP391

# 面向优先级任务的移动边缘计算资源分配方法

董思岐[1]，吴嘉慧[2]，李海龙[1]，屈毓锛[1]，胡　磊[1]

（1. 火箭军工程大学 作战保障学院，西安 710025；2. 中国电子科技集团公司第二十八研究所，南京 210007）

**摘　要：**目前移动边缘计算中的资源分配方法，多数按照任务请求计算卸载的时间顺序分配计算资源，未考虑实际应用中任务存在优先级的问题。针对此类情况下的计算需求，提出一种面向优先级任务的资源分配方法。根据任务平均处理价值赋予其相应的优先级，对不同优先级的任务进行计算资源加权分配，在保证高优先级任务获取充足计算资源的同时，减少完成所有任务计算的总时间及能耗，从而提高服务质量。仿真结果表明，与平均分配、按任务数据量分配和本地计算方法相比，该方法的计算时延分别降低 83.76%、15.05% 和 99.42%，能耗分别降低 84.78%、17.37% 和 87.69%。

**关键词：**移动边缘计算；优先级任务；加权分配；资源分配；服务质量

**开放科学（资源服务）标志码（OSID）：**

**中文引用格式：**董思岐，吴嘉慧，李海龙，等. 面向优先级任务的移动边缘计算资源分配方法[J]. 计算机工程，2020，46(3)：18-23.

**英文引用格式：**DONG Siqi，WU Jiahui，LI Hailong，et al. Resource allocation method for priority task in mobile edge computing[J]. Computer Engineering，2020，46(3)：18-23.

# Resource Allocation Method for Priority Task in Mobile Edge Computing

DONG Siqi[1]，WU Jiahui[2]，LI Hailong[1]，QU Yubeng[1]，HU Lei[1]

（1. Combat Support College，Rocket Force University of Engineering，Xi'an 710025，China；
2. The 28th Research Institute of China Electronics Technology Group Corporation，Nanjing 210007，China）

**【Abstract】**In mobile edge computing，most of existing resource allocation methods allocate computing resources according to the time that tasks request computation offloading，without considering task priority in practical applications. To meet computing requirements in such cases，this paper proposes a resource allocation method for priority tasks. This method assigns priority to each task based on their average processing value，and implements weighted allocation of computing resources for tasks of different priorities. It ensures high-priority tasks obtain sufficient computing resources while the total time and energy consumption for completing all tasks are reduced. Thus the Quality of Service（QoS）is improved. Simulation results show that，compared with the method of evenly allocating computing resources，the method of allocating resources according to the amount of task data and the method of placing all tasks on mobile terminals，the calculation delay of this method is reduced by 83.76%，15.05% and 99.42% respectively，and the energy consumption is reduced by 84.78%，17.37% and 87.69% respectively.

**【Key words】**mobile edge computing；priority task；weighted allocation；resource allocation；Quality of Service（QoS）

**DOI：**10.19678/j.issn.1000-3428.0054420

## 0　概述

移动互联网的发展使得移动终端（如手机、平板电脑等）逐渐取代台式设备，成为人们日常处理信息的主要工具。目前，移动终端的通信计算量呈爆炸式增长的趋势，但由于硬件设备条件约束，其在处理大量计算业务时会导致较高的计算时延，并且加速电池的能量消耗，降低用户的体验质量（Quality of Experience，QoE）。传统方法将移动端需要进行计算的任务部署至云服务器，解决了移动设备计算资源有限导致计算时延过高的问题，但是将任务全部

**基金项目：**国家自然科学基金青年科学基金项目（61702525）。

**作者简介：**董思岐（1995—），女，硕士研究生，主研方向为移动边缘计算；吴嘉慧，工程师、硕士；李海龙，副教授、博士；屈毓锛，讲师、博士；胡　磊，硕士研究生。

**收稿日期：**2019-04-03　　**修回日期：**2019-05-06　　**E-mail：**dsq301617@163.com

传输至距移动端较远的云服务器进行计算,会面临网络负荷和通信链路传输时延增加,以及通信链路带宽有限的问题。对此,研究者提出了移动边缘计算的概念[1]。移动边缘计算整合了距用户移动端较近的边缘网络中的计算资源,能够降低网络负荷和通信链路的传输时延,提高任务分发能力同时优化终端用户的使用体验。但移动边缘网络中的计算资源有限,因此需要设计合理的计算资源分配策略,以降低计算时延和能量消耗。

目前关于计算资源分配的研究,多数按照任务请求计算卸载的时间顺序为其分配计算资源,而在许多实际应用中,任务计算处理的紧迫程度或重要性具有一定区分度,仅依靠任务请求计算卸载的时间顺序对其进行资源分配是不合理的,结合实际需要划分任务的优先级并据此进行计算资源分配,能够更好地满足任务的计算需求。因此,本文提出面向优先级任务的边缘计算资源分配方法,根据任务优先级通过加权法分配相应的计算资源,在保障高优先级任务获取计算资源的同时,减少系统任务的计算时间及能量消耗。

## 1　相关研究

将计算任务卸载至边缘服务器进行计算,需要通过通信链路将数据传输至边缘服务器端,边缘服务器利用计算资源对任务进行计算,计算完成后再将计算结果传输至移动端。整个过程中涉及的资源主要为通信链路的通信资源及边缘服务器的计算资源。文献[2]提出了基于半定松弛和随机化映射的启发式算法,将最小化时延和能耗问题公式化为非凸约束二次规划问题。实验结果表明,该算法经过少量随机性迭代即可达到最优性能。文献[3]提出利用迭代算法查找上行链路发送的比特数量的最优值,其对文献[2]的卸载策略做了进一步的讨论,论证了信道质量好时执行计算卸载的效率更高。文献[4]通过设置存放计算业务的缓冲区,在缓冲区稳定性约束的基础上构建功耗最小化模型,并利用 Lyapunov 算法进行传输功率和带宽分配优化。文献[5]提出整合上行链路的传输功率与边缘计算服务器处的计算资源分配,利用凸优化算法实现计算资源分配,设计一种启发式算法解决任务卸载问题。文献[6]提出了基于效益函数模型的资源优化方法增加系统收益,并对多种研究场景进行了讨论分析。文献[7]通过对任务数据进行压缩,构建在通信资源充足而计算资源有限场景下的分段优化模型,达到减少移动端计算时延的目的。文献[8]在增强现实任务的场景下建立凸优化资源求解数学模型,研究在不同时延约束条件下节约能耗的情况。文献[9]通过对博弈算法和匈牙利算法的相互迭代解决资源分配的优化问题,从而降低了计算能耗及时延。文献[10]提出一种能耗感知的卸载方案,将电池剩余电量引入到能量消耗和时间延迟的加权因子中,通过迭代搜索算法优化通信和计算资源的分配。文献[11]构建了移动端的待处理数据具有异构性的能量消耗模型,根据任务的截止时间在调度阶段设定卸载时间间隔。文献[12]通过设计分布式计算卸载算法,对计算时延和能耗指标进行量化,达到了更低的计算时间开销。文献[13]通过二分法寻找需要进行计算卸载的移动设备的最优发射功率和匹配计算资源,提高了用户的任务卸载量,有效减少了系统开销。

由上述研究可以看出,在资源分配方面,研究者主要通过优化移动端的发射功率以及合理分配通信带宽,或结合优化边缘服务器端的计算资源来降低时延及能耗。此外,在移动边缘计算的研究场景中没有区分任务之间的优先级,默认任务之间为平等关系。文献[14]考虑了任务间的优先级关系,在云计算场景下构建符合用户优先级的适应度函数,通过引入重优化判断准则优化了粒子群算法,从而得到全局最优解。

区别于云计算场景下基于任务优先级的资源分配算法、对任务进行部分卸载计算及定义任务优先级的方式,面向移动边缘计算场景,本文提出针对边缘服务器资源的加权分配策略,以任务平均价值量作为任务优先级区分标准,将任务全部卸载到边缘服务器进行计算。

## 2　场景分析与系统建模

### 2.1　场景分析

本文策略针对的移动端(任务)与服务器多对一场景如图 1 所示,其中有多个移动设备终端和一台边缘计算服务器,每个设备终端处包含一个需要进行计算处理的任务。各个移动终端上待处理的任务根据其平均计算价值量划分优先级高低。由于移动终端计算能力有限,若在移动终端进行处理,可能会导致计算时间过长超出任务的处理时间限制,或本地资源不足以支持完成计算。因此,需要将待处理任务卸载至边缘计算服务器上进行计算。通过无线通信链路将任务传输至边缘计算服务器,边缘计算服务器利用自身计算资源完成计算任务后,再由无线通信链路将计算结果传回至移动设备终端。考虑到通信链路带宽及通信资源有限,边缘计算服务器的计算资源也是有限的,本文对边缘服务器端的计算资源优化分配方法进行分析,以期在存在资源和任务优先级约束的条件下降低计算时延与能量消耗。

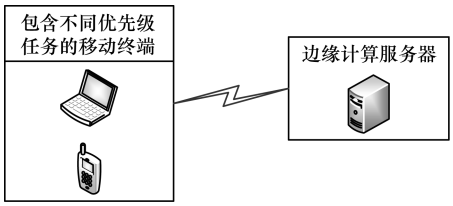


**图 1 资源分配场景**

**Fig. 1 Resource allocation scenario**

## 2.2 系统建模

基于上述场景分析及任务计算流程,将系统建模分为以下 3 个部分:

1)任务平均优先级建模。首先需要对任务根据其要处理的平均价值量进行任务的优先级划分。

2)对任务在通信链路上的传输时延进行建模。传输时延为任务数据由移动端发送至边缘服务器过程中消耗的时间与边缘服务器将计算结果回传至移动端消耗的时间之和。由于边缘服务器的发射功率较大,因此在计算时间延迟时可以忽略将数据由边缘服务器回传至移动端的时间消耗,只考虑将任务数据上传至边缘服务器的时间消耗。

3)对任务在边缘服务器端的计算时间进行建模。通过对优先级不同的任务进行加权处理,分配相应的计算资源,服务器同时处理所需计算的任务,完成后即将计算结果发送回对应的移动端,取边缘服务器进行任务计算所需的最长计算时间作为计算时间。

由移动端及边缘服务器构成的系统任务处理流程如图 2 所示。

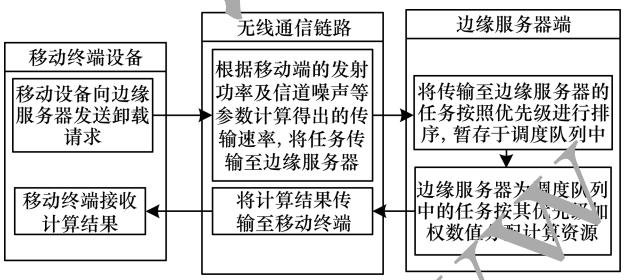


**图 2 任务处理流程**

**Fig. 2 Task processing flow**

### 2.2.1 任务优先级加权设定

设定在研究场景中有 $k$ 个移动终端,每个移动终端上存在一个具有优先级的待处理任务,即共有 $k$ 个待处理的任务,用 $T=\{T_1,T_2,\cdots,T_n,\cdots,T_k\}$ 表示待处理的任务,其中 $0<n\leqslant k, n\in\mathbb{N}^+$,对于第 $n$ 个移动终端的任务 $T_n$,具体信息的参数表示为 $T_n=\{T_n^{\mathrm{ID}},T_n^{\mathrm{Datasize}},T_n^{\mathrm{Deadline}},T_n^{\mathrm{Trans}},T_n^{\mathrm{Value}}\}$,其中:$T_n^{\mathrm{ID}}$ 为任务的标识编号;$T_n^{\mathrm{Datasize}}$ 为任务的数据量(以 bit 为单位);$T_n^{\mathrm{Deadline}}$ 为任务的最迟截止处理时间;$T_n^{\mathrm{Trans}}$ 为任务进行计算卸载时需要请求的通信链路传输带宽(以 bit/s 为单位);$T_n^{\mathrm{Value}}$ 为任务的计算价值量,表示任务所具有的重要程度,该值越大表明任务 $T_n$ 的重要程度越高。因此,任务的优先级 $P$ 可以表示为 $P_n=T_n^{\mathrm{Value}}/T_n^{\mathrm{Datasize}}$,该式表明,在单位时间内需要处理的平均计算价值量越大,任务的优先级越高[15]。

### 2.2.2 数据传输时间设定

设无线通信链路传输数据的带宽为 $B$,为保证数据在传输过程中不受通信链路带宽有限的影响,设置同一时刻传输的数据所请求的传输带宽不超过通信链路的最大带宽,即 $\sum_{n=1}^{k}T_n^{\mathrm{Trans}}\leqslant B$。在任务数据的传输过程中需要通过无线信道进行通信传输,由香农定理可知,在有限带宽、具有噪声干扰的信道环境下[16],数据传输速率极限值 $V_{\max}$ 可表示为:

$$V_{\max}=B\mathrm{lb}(1+S/N_0) \tag{1}$$

其中,$S/N_0$ 为信噪比,$S$ 为信道内信号传输功率,$N_0$ 为信道内噪声功率。任务 $T_n$ 在通信信道中由移动设备终端传输至边缘计算服务器的传输速率可表示为:

$$V_n=B\mathrm{lb}\left(1+\frac{\beta M_{pn}}{N_0}\right) \tag{2}$$

其中:$\beta$ 为任务 $T_n$ 所处的第 $n$ 个移动终端与边缘服务器间的信道增益,信道增益描述的是信道本身的衰减及衰落特性[17],$\beta$ 为一个随机的独立同分布变量;$M_{pn}$ 为第 $n$ 个任务所在的移动终端将任务 $n$ 发送至边缘服务器时提供的发射功率。设定 $M_p$ 为移动终端所能提供的最大的发射功率,有 $M_{pn}\leqslant M_p$。因此,数据由移动端发送至边缘服务器过程中需要的传输时间可以记为 $t_{\mathrm{trans}}=T_n^{\mathrm{Datasize}}/V_n$。

### 2.2.3 边缘服务器端计算时间建模

边缘服务器端的具体参数设定如下:边缘网络中有 1 台服务器,用 $S_1$ 表示。对于 $S_1$ 服务器的相关参数,用集合 $S_1=\{S_{\mathrm{W}},S_{\mathrm{C}}\}$ 表示具体的参数配置及服务器状态,其中:$S_{\mathrm{W}}$ 表示边缘服务器的工作状态,$S_{\mathrm{W}}=1$ 代表服务器正忙,$S_{\mathrm{W}}=0$ 代表服务器处于空闲状态;$S_{\mathrm{C}}$ 表示边缘服务器的计算能力,即单位时间内边缘服务器可提供的计算资源量(以 bit/s 为单位);$S_{\mathrm{P}}$ 表示边缘服务器的计算功率。

任务的期望完成时间定义为:将服务器的计算资源全部分配给该任务时,对该任务进行计算所需要的时间。因此,任务 $T_n$ 的期望完成时间 $T_{\mathrm{EC}}$ 可以表示为 $T_{\mathrm{EC}}=T_n^{\mathrm{Datasize}}/S_{\mathrm{C}}$。设在进行计算时服务器分配给任务 $T_n$ 的计算资源为 $S'_{\mathrm{C}n}$,那么边缘服务器完成对任务 $T_n$ 的计算所需要的时间 $T_{\mathrm{C}n}$ 可表示为 $T_{\mathrm{C}n}=T_n^{\mathrm{Datasize}}/S'_{\mathrm{C}n}$,边缘服务器在同一时刻开始对任务进行计算处理,完成任务的计算后将计算结果传输回相应的移动终端,因此,取计算时间最长的任务计算值作为计算时延,记为 $T_{\mathrm{C}}$。在本文设定的场景中,移动

终端单位时间内的计算请求量小于服务器可提供的计算资源量。

## 3　边缘服务器计算资源分配策略

### 3.1　基于优先级的资源分配方式

边缘服务器的计算资源有限,需要根据任务的优先级进行计算资源分配,分配过程如图 3 所示。

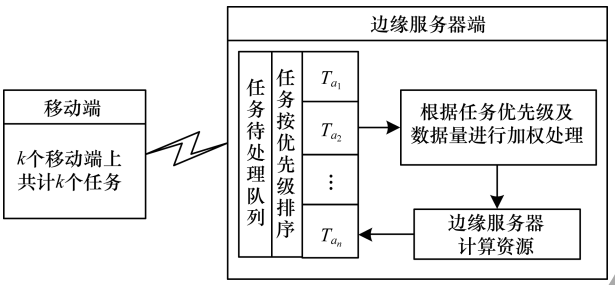


**图 3　资源分配过程**

**Fig. 3　Resource allocation process**

在待处理任务通过通信链路传输至边缘服务器端后,首先对任务的优先级排序,形成优先级序列 $P_{n\gamma}$,其中,$n$ 对应待处理任务 $T_n$,$\gamma$ 对应所有任务 $T_n$ 进行优先级排序后的得到的次序数值($\gamma \subset \mathbb{N}, 1 \leqslant \gamma \leqslant k$),$\gamma = 1$ 表示对应的任务在所有任务中的优先级最高,$\gamma = k$ 表示对应的任务在所有任务中的优先级最低。设定单位时间内服务器的总的计算资源为 $S_C$,为优先级序列 $\{P_{a_1}, P_{a_2}, \cdots, P_{a_x n}\}$ 所对应的任务 $T_{a_1}, T_{a_2}, \cdots, T_{a_x}$ 依次分配数值为 $S'_{Ca_1}, S'_{Ca_2}, \cdots, S'_{Ca_x}$ 的计算资源。其中 $a_1, a_2, \cdots, a_x$ 为互不相等的正整数,且 $\forall a_x, x \in [1,k]$,都有 $a_x \in [1,k]$,边缘服务器进行计算时任务所分配到的计算资源 $S'_{Cn}$ 与边缘服务器的总的计算资源 $S_R$ 之间存在如下关系:

$$\begin{cases} S'_{Ca_1} = \lambda_1 S_C \\ S'_{Ca_2} = \lambda_2 S_C \\ \quad \vdots \\ S'_{Ca_n} = \lambda_k S_C \end{cases} \tag{3}$$

由式(3)可知,$\lambda$ 系数影响对应优先级任务所能获取到的计算资源量,因此,$\lambda$ 需要体现任务的优先级指标作为资源分配的加权系数。假定 $\lambda_1, \lambda_2, \cdots, \lambda_k$ 之间满足以下关系:

$$\begin{cases} 0 \leqslant \lambda_k < \cdots < \lambda_2 < \lambda_1 \leqslant 1 \\ \lambda_1 + \lambda_2 + \cdots + \lambda_k = 1 \\ \lambda_n = \left(P_{a_x n} / \sum_{n=1}^{k} P_{a_x n}\right) \times \left(T_{Ba_x} / S_C\right) \times \left(T_{Ba_x} / \sum_{n=1}^{k} T_{Ba_x}\right) \times \delta \end{cases} \tag{4}$$

其中,$\delta$ 为 $\lambda_n$ 的归一化系数,$n \in [1,k]$ 且 $n \in \mathbb{N}^+$。

忽略任务的排序时间及调度任务时间,将边缘服务器对任务进行计算所消耗的时间 $T_C$ 表示为:

$$T_C = \max\{T_{C1}, T_{C2}, \cdots, T_{Ck}\} \tag{5}$$

在进行计算的整个过程中,每个任务间的计算是相互独立的,可以将计算事件视为独立离散事件,将最小化时延转化为约束条件下的单目标组合优化性质的问题,因此时间延迟的数学模型可表示为:

$$\min\{T_C\}$$

$$\text{s.t.} \begin{cases} S'_{Ca_x} = \lambda_k S_C \\ 0 \leqslant \lambda_k < \cdots < \lambda_2 < \lambda_1 \leqslant 1 \\ \lambda_1 + \lambda_2 + \cdots + \lambda_k = 1 \\ \lambda_n = \left(P_{a_x n} / \sum_{n=1}^{k} P_{a_x n}\right) \times \left(T_{a_x}^{\text{Datasize}} / S_C\right) \times \\ \qquad \left(T_{a_x}^{\text{Datasize}} / \sum_{n=1}^{k} T_{a_x}^{\text{Datasize}}\right) \times \delta \\ \sum_{n=1}^{k} T_n^{\text{Trans}} \leqslant B \\ n \in [1,k], n \in \mathbb{N}^+, a_x \in [1,k] \end{cases} \tag{6}$$

### 3.2　加权算法描述

本文算法将最小化时延问题转化为加权线性规划问题,首先通过分治思想将时间延迟及能量消耗问题分解为各部分建模模块消耗的时间及能耗的子问题进行求解,结合任务优先级、服务器计算能力及计算资源量的约束,通过线性规划方法为不同任务分配相应加权值的计算资源,降低处理任务计算能耗的时间延迟。

面向优先级任务的加权分配算法执行步骤如下:1)初始化时输入任务的优先级和任务数据数据量等参数;2)根据相关参数及归一化系数设定加权值;3)利用加权值将计算资源分配给各个任务进行计算。算法描述如下:

**输入**　$T_n, S_C, S_W$

**输出**　计算结果,计算时延,系统能耗

1)任务传输至边缘服务器端后,将其置于调度队列中,以任务优先级为关键字对任务进行排序。

2)根据任务优先级及任务的期望完成时间,以及边缘服务器的计算能力,按式(4)计算加权值。

3)检测边缘服务器的工作状态 $S_W$,若 $S_W = 1$,则将任务置于调度队列中等待;若 $S_W = 0$,则按加权数值进行进行资源分配并开始计算,计算完成后将计算结果回传至各任务所对应的移动端。

4)取 $k$ 个任务中计算时间最大值作为计算时延。

忽略任务进行优先级排序及资源分配所需时间,由移动端与边缘服务器组成的系统完成计算所

消耗的总时间 $t$ 可记为：

$$t = t_{\text{trans}} + T_{\text{C}} \tag{7}$$

设定各个移动终端的发射功率及计算功率相同，则由移动端与边缘服务器组成的系统完成计算所消耗的总能量 $E_{\text{total}}$ 可记为：

$$E_{\text{total}} = M_{pn} \times t_{\text{trans}} + S_P \times T_{\text{C}} \tag{8}$$

## 4 仿真实验

利用 Matlab 仿真平台对本文方法作实验验证，并与以下 3 种方法进行对比：将任务全部置于本地移动端进行计算的方法（本地计算方法），将任务卸载至边缘服务器进行计算时平均分配计算资源的方法[18]（平均分配方法），按照任务数据量大小分配计算资源的方法[19]（按数据量分配方法）。对比的性能指标为计算时延和能耗：将任务置于本地移动设备端进行处理的时延为移动端的计算时延；将任务卸载至边缘服务器进行处理的时延为边缘服务器计算时延与将数据传输至服务器的上传时延之和；将任务置于本地进行处理的能耗为移动端的计算功率与计算时延之积。将任务卸载至边缘服务器进行处理时，由移动端及边缘服务器构成的整体系统的能耗为移动端的待机功率与时延之积、移动端的发射功率与数据传输时延之积、边缘服务器的计算时间与计算功率之积三部分总和。具体数据范围及参数设定如表 1 所示。

表 1 仿真参数设定

Fig. 1 Simulation parameter setting

| 仿真参数 | 参数设定 |
|---|---|
| 移动端个数范围 | [10,100] |
| 任务价值量范围 | $[10^8, 10^9]$ |
| 上行链路带宽/Hz | $10^8$ |
| 移动端传输功率/W | 0.25 |
| 边缘服务器计算能力/(bit·s$^{-1}$) | $10^9$ |
| 移动终端计算能力/(bit·s$^{-1}$) | $10^7$ |
| 任务数据量/bit | $[10^6, 10^7]$ |
| 边缘服务器功率/W | 200 |
| 移动端计算功率/W | 0.25 |

本文研究系统中移动端的任务数量范围为 10 个～100 个时，移动端及由移动端和边缘服务器构成的系统所用的时延及能耗的数值。通过本文方法与本地计算及资源平均分配算法的对比，论证本文算法的优越性。根据经验值给出具体的仿真参数范围[8-15]，实验中使用的数据值由 Matlab 在表 1 范围内生成随机数作为具体参数。给定相同的数据时，移动端与边缘服务器的计算时间如图 4 所示。

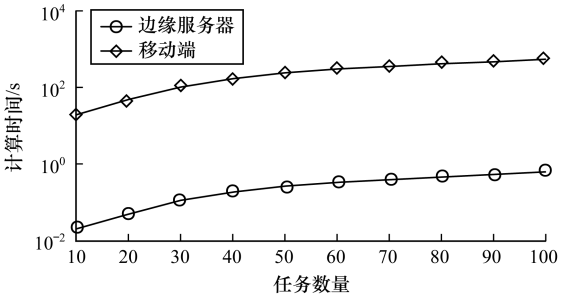


图 4 移动端与边缘服务器的计算时间

Fig. 4 Calculation time of mobile terminal and edge server

由图 4 可以看出，在相同的数据量下，将任务置于边缘服务器进行计算所需要的时间远低于在移动端本地进行计算消耗的时间，因此，有必要将任务卸载到边缘服务器进行计算。为进一步降低计算时延，将边缘服务器端的计算资源根据任务的优先级进行合理分配。仿真结果表明：本文算法与平均分配方法相比降低了 83.76% 的时间消耗，与平均分配方法相比降低了15.05% 的时间消耗，与本地计算方法相比降低了 99.42% 的时间消耗。为便于直观观察不同方法间的性能差异，采用对数纵坐标显示消耗时间的值，4 种方法所需时延如图 5(a)所示，为进一步显示本文方法与按数据量分配方法的对比，将图 5(a)中的虚线框部分放大表示为图 5(b)。

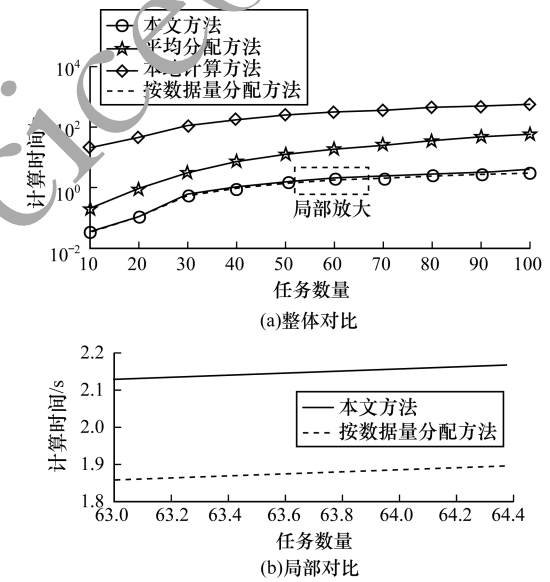


图 5 4 种方法的计算时间对比

Fig. 5 Comparison of calculation time of four methods

在对比计算时延的基础上，对未进行计算卸载时移动端的能量消耗情况与进行计算卸载时由移动端与边缘服务器构成的系统的能量消耗进行对比，如图 6 所示。仿真实验结果表明：与本地计算方法相比，本文方法能够节约 87.69% 的计算能耗；与平均分配方法相比，本文方法能够节约 84.87% 的能量

消耗,与按任务数据量分配方法相比,本文方法能够降低 17.37% 的能量消耗。

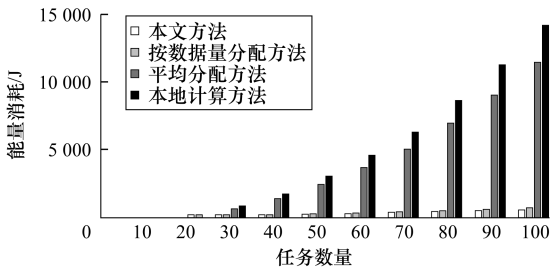


**图 6　4 种方法的能耗对比**

**Fig. 6　Comparison of energy consumption of four methods**

## 5　结束语

为解决移动端进行任务计算时时延及能耗高,以及目前研究中利用边缘计算技术处理任务时较少考虑任务优先级的问题,本文提出一种资源加权分配方法,将有限的计算资源在约束条件下合理分配给具有不同优先级的任务。实验结果表明,该方法可以有效降低计算时延与系统能耗。为便于计算,本文设定数据在同一时刻对边缘服务器进行计算卸载请求,数据传输过程中速率恒定,未计算数据的回传时间。下一步将结合实际情况设定任务在不同时刻动态向边缘服务器进行计算卸载请求,同时在数据传输过程中加入干扰因子使传输速率更符合实际传输情景。

## 参考文献


[1] SATYANARAYANAN M. The emergence of edge computing[J]. Computer,2017,50(1):30-39.

[2] CHEN M H, LIANG B, DONG M. A semidefinite relaxation approach to mobile cloud offloading with computing access point [C]//Proceedings of IEEE International Workshop on Signal Processing Advances in Wireless Communications. Washington D. C., USA: IEEE Press,2015:186-190.

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

[4] MAO Y Y, ZHANG J, SONG S H, et al. Power-delay tradeoff in multi-user mobile-edge computing systems[C]// Proceedings of IEEE Global Communications Conference. Washington D.C.,USA:IEEE Press,2016:1-6.

[5] TRAN T X, POMPILI D. Joint task offloading and resource allocation for muliti-server mobile edge computing networks[J]. IEEE Transactions on Vehicular Technology,2019,68(1):856-868.

[6] YE Dongdong. Research on resource optimization in mobile edge computing environment[D]. Guangzhou: Guangdong University of Technology,2018. (in Chinese)
叶东东. 移动边缘计算环境下的资源优化研究[D]. 广州:广东工业大学,2018.

[7] REN Jinke, YU Guanding, CAI Yunlong, et al. Latency optimization for resource allocation in mobile-edge computation offloading [J]. IEEE Transactions on Wireless Communications,2018,17(8):5506-5519.

[8] YU Yun, LIAN Xiaocan, ZHU Yuhang, et al. Resource allocation optimization method for augment reality applications based on mobile edge computing [J]. Journal of Computer Applications,2019,39(1):22-25. (in Chinese)
余韵,连晓灿,朱宇航,等. 增强现实场景下移动边缘计算资源分配优化方法研究[J]. 计算机应用,2019,39(1):22-25.

[9] ZHANG Jing, XIA Weiwei, YAN Feng, et al. Joint computation offloading and resource allocation optimization in heterogeneous networks with mobile edge computing[J]. IEEE Access,2018,6:19324-19337.

[10] ZHANG Jiao, HU Xiping, NING Zhaolong, et al. Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks [J]. IEEE Internet of Things Journal,2018,5(4):2633-2645.

[11] YOU Changsheng, ZENG You, ZHANG Rui, et al. Asynchronous mobile-edge computation offloading: energy-efficient resource management[J]. IEEE Transactions on Wireless Communications,2018,17(11):7590-7605.

[12] CHEN Xu,JIAO Lei,LI Wangzhong,et al. Efficient multi-user computation offloading for mobile-edge cloud computing [J]. IEEE/ACM Transactions on Networking, 2016,24(5):2795-2808.

[13] PHAM Q V, LEANH T, TRAN N H, et al. Decentralized computation offloading and resource allocation for mobile-edge computing: a matching game approach[J]. IEEE Access, 2018,6:75868-75885.

[14] PU Xun,DU Jia,LU Xianliang. Task scheduling policy for cloud computing based on user priority level[J]. Computer Engineering,2013,39(8):64-68. (in Chinese)
蒲汛,杜嘉,卢显良. 基于用户优先级的云计算任务调度策略[J]. 计算机工程,2013,39(8):64-68.

[15] LU Wenxing, LI Guangzhi. Task scheduling strategy based on priority and bandwidth constraint in cloud computing[J]. Chinese Journal of Management Science, 2016,24(S1):68-73. (in Chinese)
陆文星,李光智. 云计算下基于优先级和带宽约束的任务调度策略[J]. 中国管理科学,2016,24(S1):68-73.

[16] SHEN Lianfeng, YE Zhihui. Information theory and coding[M]. Beijing: Science Press,2007. (in Chinese)
沈连丰,叶芝慧. 信息论与编码[M]. 北京:科学出版社,2007.

[17] SONG Hailong, ZHANG Shuzhen. Gaussian transmission channel optimization algorithm based on superposition coding and multi-user scheduling[J]. Journal of Computer Applications,2015,35(6):1537-1540,1584. (in Chinese)
宋海龙,张书真. 基于叠加编码及多用户调度的高斯传输信道优化算法[J]. 计算机应用,2015,35(6):1537-1540,1584.

[18] SONG Hu. Research on cloud computing management mechanism for user service demand [D]. Hefei: University of Science and Technology of China, 2013. (in Chinese)
宋浒. 面向用户服务需求的云计算管理机制研究[D]. 合肥:中国科学技术大学,2013.

[19] YANG Ming,LIU Yuan'an,MA Xiaolei,et al. Grid resource allocation-pricing based on weighted average[J]. Journal of Beijing University of Posts and Telecommunications, 2009, 32(6):9-13,18. (in Chinese)
杨明,刘元安,马晓雷,等. 基于加权平均的网格资源分配与定价[J]. 北京邮电大学学报,2009,32(6):9-13,18.


编辑　金胡考