移动云计算中的高能效资源分配

（申请清华大学工学硕士学位论文）

培 养 单 位 ： 电子工程系

学　　科 ： 信息与通信工程

研 究 生 ： 赵 赟

指 导 教 师 ： 牛 志 升 教 授

二〇一六年四月

移动云计算中的高能效资源分配

赵赟

Energy-efficient Resource Allocation in Mobile Cloud Computing

Thesis Submitted to

Tsinghua University

in partial fulfillment of the requirement

for the degree of

Master of Science

in

Information and Communication Engineering

by

Yun Zhao

Thesis Supervisor： Professor Zhisheng Niu

April, 2016

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：

清华大学拥有在著作权法规定范围内学位论文的使用权，其中包括：（1）已获学位的研究生必须按学校规定提交学位论文，学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文；（2）为教学和科研目的，学校可以将公开的学位论文作为资料在图书馆、资料室等场所供校内师生阅读，或在校园网上供校内师生浏览部分内容。

本人保证遵守上述规定。

（保密的论文在解密后遵守此规定）

作者签名：_____ 导师签名：_____

日　　期：_____ 日　　期：_____

摘　要

随着移动通信业务量的剧增，能耗已经成为亟需解决的问题，特别是受限于电池容量的移动终端能耗（上行传输以及应用处理能耗）是处理复杂应用的主要瓶颈。移动云计算的主要功能是可以将计算任务从资源有限的移动终端通过无线网转移到云端处理，即通过无线资源和云端计算资源换取终端计算资源和能量节约。本课题旨在通过对移动云计算中的无线资源和计算资源的联合分配，在满足用户服务质量的前提下最大程度实现在移动云计算中的高能效负载转移。主要研究成果如下:

1、旨在能够减少终端能耗，为多用户在使用移动云计算时建模了无线资源和计算资源的集中式联合优化问题，并且，基于延时约束、不同移动终端的应用类型和任务量大小，为资源分配提供一种启发式的策略。数值结果表明，与在终端计算没有负载转移的制度相比，我们设计的资源分配策略在系统中有 3 个移动终端的情况下能够有效减少 40%的移动终端能耗，同时还可以满足用户的延时约束。此外，启发式算法的性能接近于搜索算法，但复杂度很低。

2、在设计的分布式资源分配策略中，云端通过调整拉格朗日乘子来调整用户的计算资源请求量。通过对偶函数的方法将原本多个用户互相耦合的优化问题解耦，用户自行决定上传任务量及请求的计算资源量，从而减少用户和基站间的信令开销，降低资源分配策略的复杂度。

3、在用经济学方法分布式分配移动云计算的资源中，我们引入云端计算资源的价格，云端可以调整计算资源的价格从而使用户购买合适的计算资源来达到用户和云端之间的纳什均衡，在该纳什均衡点，云提供商和用户都不能通过单独改变自己的策略来增加自己的收益。

4、移动云计算中的用户切换问题异于传统蜂窝网中切换问题的最大不同点在于除了无线端的切换，背后还有云端计算任务的切换。综合考虑无线端和计算端的资源，我们提出了基于延时的用户切换机制，通过切换迟滞量 HHM 在不明显引入额外附加延时的同时减少平均切换次数，降低切换开销。

关键词：移动云计算；多用户；无线资源和计算资源；高能效负载转移；移动性管理

Abstract

With the surging mobile traffic, energy consumption has been a great challenge for mobile communication. Due to the limitation of battery capacity at the mobile terminals，terminal energy consumption (energy consumption for uplink transmission and task execution) is the major concern for complicated applications. Mobile Cloud Computing (MCC) enables task offloading by transferring the computation tasks from the mobile terminals to the cloud through wireless networks, which exchanges radio and cloud computational resources for terminal computational resources and energy consumption reduction. This project focuses on realizing the energy-efficient task offloading while satisfying users' quality of service (QoS) in MCC by jointly allocating radio and computational resources. Tha main contributions of the thesis are listed as follows:

1、Aiming at reducing terminal energy consumption, we formulate a centralized radio and computational resource allocation optimization problem for multi-user MCC scenario. We design a heuristic resource allocation scheme based on the delay requirements and the amount of the computation tasks at the terminal side. Numerical results show that our designed scheme can achieve 40% terminal energy consumption reduction, with 3 mobile terminals in the system while satisfying the delay requirements of each mobile user, compared with no offloading to the cloud. Moreover, our heuristic algorithm performs really well compared with the search algorithm with low computation complexity.

2、In our distributed resource allocation scheme, the cloud side adjusts mobile terminal's computational resource request by adjusting the Lagrangian multiplier. We decompose the original multi-user coupled optimization problem using dual function. Mobile termianls can decide the amount of uploaded tasks and computational resources by themselves according to the Lagrangian multiplier. The dual decomposition method can reduce the signaling overhead between mobile users and the Base Station (BS) and the complexity to allocate resources.

3、In the economic way to allocate resources in MCC, we introduce the price of computational resources at the cloud side. The cloud provider urges mobile terminals to buy appropriate amount of computational resources to achieve the Nash Equilibrium by

adjusting the price of computational resources. At the Nash Equilibrium point, Neither the cloud provider nor each mobile user can improve his profit by solely change his own strategy.

4、The major difference in mobility management between MCC and traditional cellular networks lies in the computation task handover at the cloud side besides wireless handover. We propose a delay-based MCC handover scheme, which takes both radio and computational resources into consideration. The scheme reduces average handover number without introducing much average delay, by using handover hysteresis margin (HHM).

Key words: mobile cloud computing; multi-user; radio and computational resources; energy-efficient task offloading; mobility management

目 录

主要符号对照表

5G	第 5 代移动通信系统（The 5th Generation Mobile Communication System）
BBU	基带单元（Baseband Unit）
BS	基站（Base Station)
CPU	中央处理器 (Central Processing Unit)
C-RAN	云化无线接入网（Cloud Radio Access Network）
FDM	频分复用(Frequency Division Multiplexing)
HHM	切换迟滞量(Handover Hysteresis Margin)
IaaS	基础设施即服务 (Cloud Infrastructure as a Service)
IT	信息技术 (Information Technology)
MCC	移动云计算 (Mobile Cloud Computing)
MT	移动终端(Mobile Terminal)
PaaS	平台即服务 (Cloud Platform as a Service)
PPP	泊松点过程(Poisson Point Process)
QoE	体验质量(Quality of Experience)
QoS	服务质量(Quality of Service)
RRH	远程无线射频单元（Remote Radio Head）
RSRP	参考信号接收功率(Reference Signal Receiving Power)
SaaS	软件即服务 (Cloud Software as a Service)
SINR	信干噪比（Signal-to-Interference-plus-Noise ratio）

第 1 章　绪论

随着下一代移动通信系统（5G）以及物联网（Internet of things）的迅速发展，近年来移动云计算越来越成为研究的焦点。5G 系统目标是在 4G 系统的基础上提升 1000 倍容量，针对同样大小任务节约 90%的能耗。[1]移动云计算正是可以在尽可能保证用户延时需求的情况下为用户提供额外的计算存储能力，从而降低终端的任务处理负担进而减少终端能耗。其中，云计算、移动计算和移动云计算是构成移动云计算的关键演进阶段和相关技术。

1.1　云计算（Cloud Computing）

在分布式计算[2]以及网格计算[3]之后，一种以资源租用、应用托管、服务外包为核心的新型计算模式，即云计算，迅速成为技术研究与发展热点。[4]IT 领域按需服务的理念在云计算中真正得到了落实与践行。通过整合各种分布式的资源，云计算可以构建应对不同服务要求的应用计算处理环境，用户可以使用网络请求访问相应的服务资源，从而满足相应的定制化需求。

维基百科对“云计算”的解释是，这是一种通过互联网的，新的计算存储服务增加、使用和交付的模式。在这种模式下，动态可扩展的虚拟资源通过互联网的运行模式可以按需获取。[5]终端用户不需要了解云服务设施的具体细节和控制方式，就可以直接享受云计算服务。

其实，云计算中的”云”是对网络资源的一种抽象说法，表示用户不需要知道云端服务的详细技术架构就可以享受便利服务。具体的概念，不同专业背景的人对于云计算的理解各不相同。现在比较广受接受的定义是：云计算是一种提供按需服务、通常是虚拟化资源的计算形式。[6]云计算服务模式一般分为软件即服务 (Cloud Software as a Service, SaaS)、平台即服务 (Cloud Platform as a Service, PaaS)、基础设施即服务 (Cloud Infrastructure as a Service, IaaS)三种：[7]

1.1.1 软件即服务

软件即服务是一种软件许可与交付的模式。用户通常使用浏览器订购被集中托管的软件来接入 IaaS，也即云端通过网络为用户提供软件功能的

远程访问，用户不必专门购买软件许可。[5]云端基础设施的成本、软件的使用权、所有的托管、维护和服务支持都统一捆绑，并按需收费。

1.1.2 平台即服务

平台即服务能够提供一个开发环境让用户可以开发、运行、管理应用，从而免去了设立和维持复杂的基础设施。[8]云提供商建立一个计算平台，通常包括操作系统，各种程序语言执行环境以及网络服务器。用户无需了解平台背后的设备部署情况就可以用平台开发并运行他们的软件，并免去了购买和管理底层硬件和软件的成本和麻烦。[5]平台即服务可以通过公共云服务或者将软件安装在私人数据中心上来提供存储、计算和其他处理服务。

1.1.3 基础设施即服务

基础设施即服务是云提供商将资源虚拟池中的存储计算等资源，通过提供电脑实体机或者虚拟机的形式，按照用户的不同需求分配给用户。[5]其优点主要是云操作系统中的资源池能够支持大量的虚拟机，从而可以利用汇聚增益（Pooling Gain）降低用户和云提供商在硬件上的开销。

1.2　移动计算（Mobile Computing）

移动计算的含义是用户使用有计算能力的移动设备和移动通信技术，从而能够在世界上任何时间任何地点获取网络上的信息或能够获取相应所需计算环境下的资源。[9]移动计算是一种人机互动的过程，在正常使用时，设备负责传输数据、声音以及影像。移动计算主要包括移动通信、移动硬件和移动软件。[10]

1.3　移动云计算（Mobile Cloud Computing, MCC）

1.3.1　概念

移动云计算是能够充分利用网络中的共享资源的、动态连接移动终端的协作式任务处理方式。[11]移动云计算整合了云计算、移动计算和无线通信技术，利用无线网络为移动终端接入云端丰富的存储、计算资源，极大提高移动用户的体验质量（Quality of Experience，QoE）。[12]

1.3.2　优点

移动云计算的优势主要有通过将能耗大的任务转移到云端处理来节省终端能耗从而延长电池寿命，[13]允许移动终端运行复杂的应用并且提供更高的数据存储能力，通过集中处理提高系统灵活性。[14]

1.3.3　挑战

同时移动云计算也面临着延时和终端能耗受限以及保证无缝切换的挑战，其中用户到达的随机性和用户的移动性极大影响着移动云计算的性能。[15]

1.4　移动云计算架构平台

关于移动云计算的研究近年比较丰富，产生了多种不同架构，这里考虑将无线资源和计算资源拉近用户端从而保证用户低延时需求的架构。如图 1.1 是一个分布式的云场景[16]，小基站通过微云互相连接起来。微云通过连接的基站管理虚拟机给用户的分配。如果用户的请求可以在本地微云完成，任务就在本地计算，否则基站会向远端云服务器发出请求。这样，无线资源和计算资源都离用户更近，这和文献[14]Concert 中云的概念很类似，把无线和计算资源放到离用户近的位置，也就是有本地云，微云，或者本地服务器的概念，这样可以让不同类型规模的计算任务分配到不同地方计算。这样的架构可以在获得集中汇聚增益的同时更加灵活地满足不同类型的业务需求，同时由于一个本地服务云可以覆盖多个基站，也可以有效解决用户的移动切换问题。

在本地云的基础上，异构云由于给用户服务有更多选择所以也有其独特优势。我们在[17]中通过综合利用本地云和因特网云的各自特点，设计了基于阈值的任务调度策略来提高移动云计算中的服务质量。本地云距离基站以及用户近，所以能够保证用户比较小的延时约束，但是计算资源有限；因特网云距离基站远一些，可能会导致任务处理延时较大，但是计算存储资源丰富。所以当本地云已经负载过重时，利用远端云的帮助可以极大提高用户的服务质量。如果根据用户的延时界松紧不同给用户排优先级，从而让延时界更紧的用户享有更高的优先权（即让延时更敏感的用户更靠前服务），可以满足更高的服务质量要求。我们进一步在[18]中利用云端资源的价格以及本地云和因特网云的竞争关系，研究了边缘云应该部

署的计算资源量。

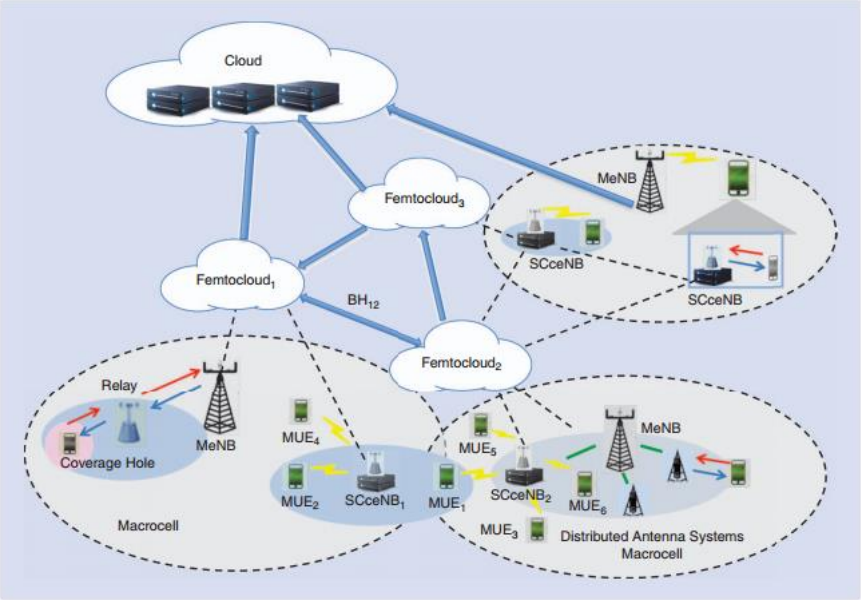

图 1. 1　分布式云架构[16]

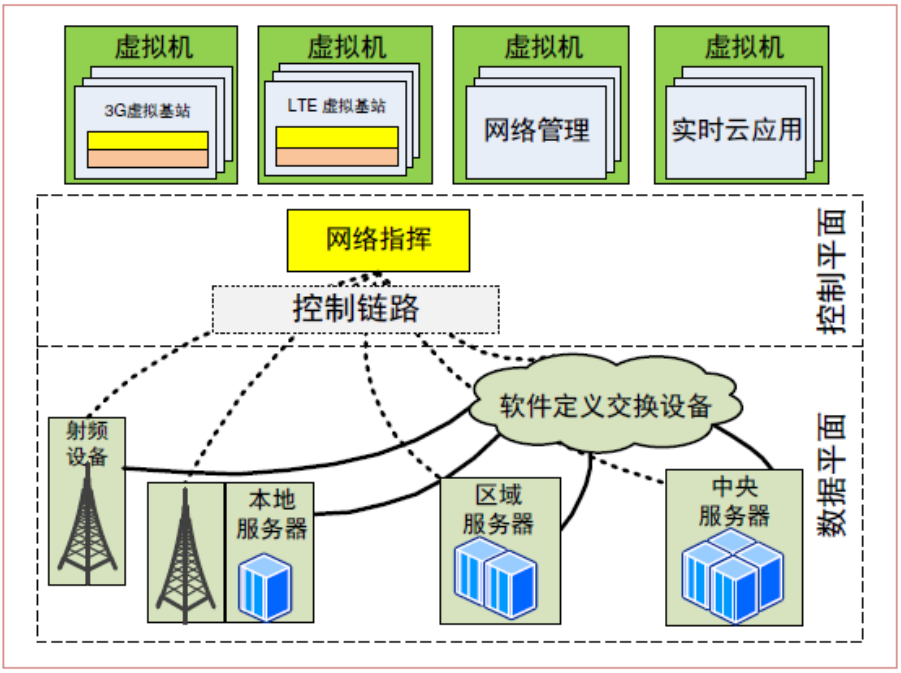

图 1. 2 CONCERT 架构[14]

同时，中国移动研究院提出的云化无线接入网（Cloud Radio Access Network, C-RAN）[19]也利用云计算技术构建新型无线接入网架构，如图 1.3 所示。C-RAN 系统将不同的基带处理模块集中在一起，形成一个虚拟基站资源池，这样资源就可以被集中管理并且根据需求动态分配。C-RAN 架构主要包括基带单元（Baseband Unit, BBU）池，远程无线射频单元

（Remote Radio Head, RRH）以及传输网络。[20]C-RAN 的主要特点与优势有基带单元集成，资源虚拟化以及更加灵活的资源调度。

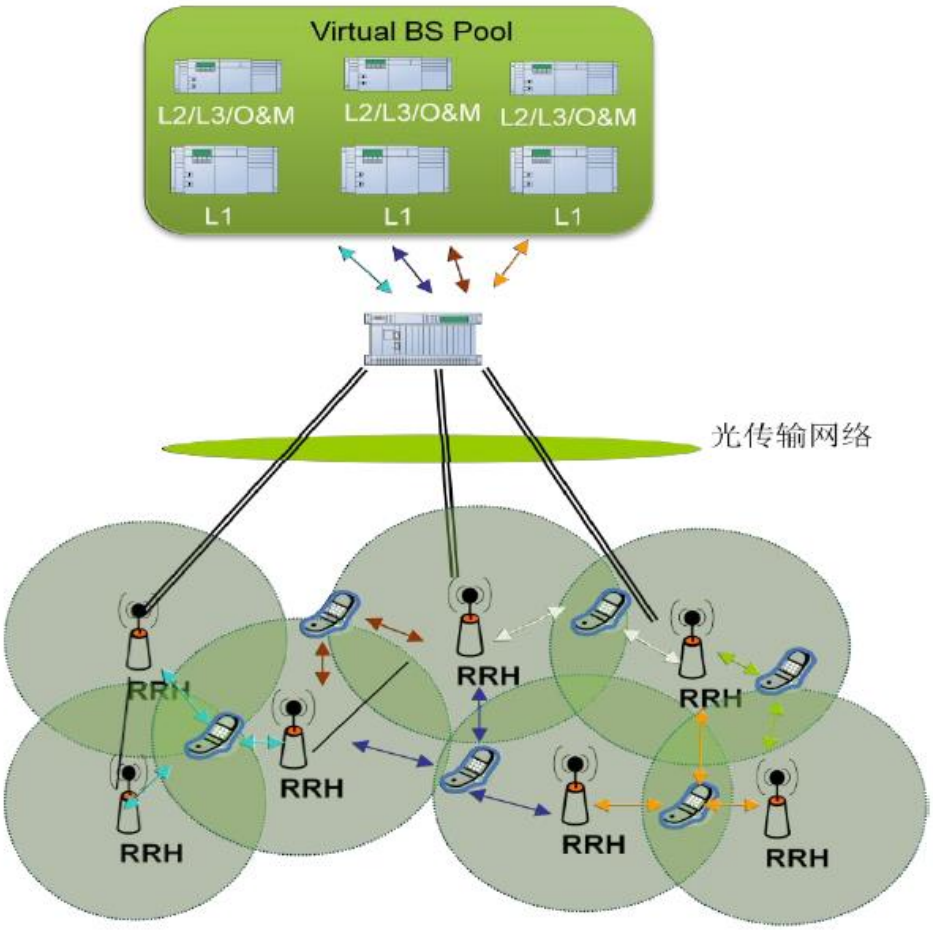

图 1. 3 C-RAN 架构[19]

1.5　各种应用

移动云计算可以帮助用户处理多种应用，如终端多媒体应用，图像处理和图像识别应用需要以图片或视频的形式传输大量的数据，如果这样的数据已经在服务器端，则负载转移的效果非常明显；[21]也可以应用到自动驾驶中，自动驾驶机器人需要实时地在车辆没有发生碰撞之前识别到周围的障碍物；也包括例如象棋等一些在线游戏；还有针对一些面向未来的新型应用，如虚拟现实，增强现实以及针对老年人的特殊医疗检测仪器等。

1.6　终端能耗危机

近年来，随着移动通信网的普及，移动终端设备的功能日趋丰富，使

用量也逐年高速攀升。以智能手机和平板电脑为主要代表的移动终端设备吸引着越来越多的用户。多媒体文件、视频会议、手机网游等终端应用能流畅地运行在各个终端上，极大丰富移动用户体验的同时也带来了移动数据量的剧增。从思科公司发布的图 1.4 智能移动终端全球增长趋势[22]可以看出，智能手机在移动终端中所占的比例逐年攀升。在 2015 年，智能手机所占比例接近 4 成，到了 2016 年这个比例已经超过了 4 成。预计到 2017 年，智能手机在移动终端中的比例会超过半数。根据数据预测，到 2020 年，智能手机将占移动终端的 67%。从图 1.5 2015 年 11 月移动用户每月使用流量分布[22]来看，几乎所有的用户当月使用的流量都超过 20MB，94%的用户当月的流量超过 200MB。更可观的是，当月流量超过 2GB 的用户数已经超过一半。2 成用户当月使用超过 5GB 的流量。还有接近 1 成用户当月流量超过 10GB。

智能移动终端以及移动数据流量的激增会给终端电池的能耗带来巨大的压力。如果终端能耗问题不能得到良好解决，会严重影响用户体验，阻碍移动通信的进一步发展。所以，如何减少终端能耗，近年来也成为研究的热门话题之一。

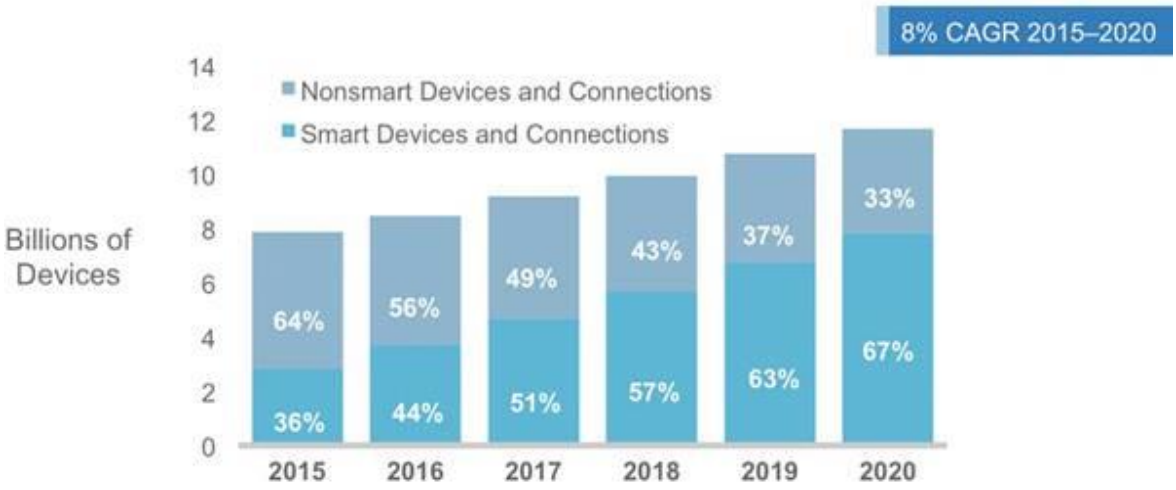

图 1.4 智能移动终端全球增长趋势[22]

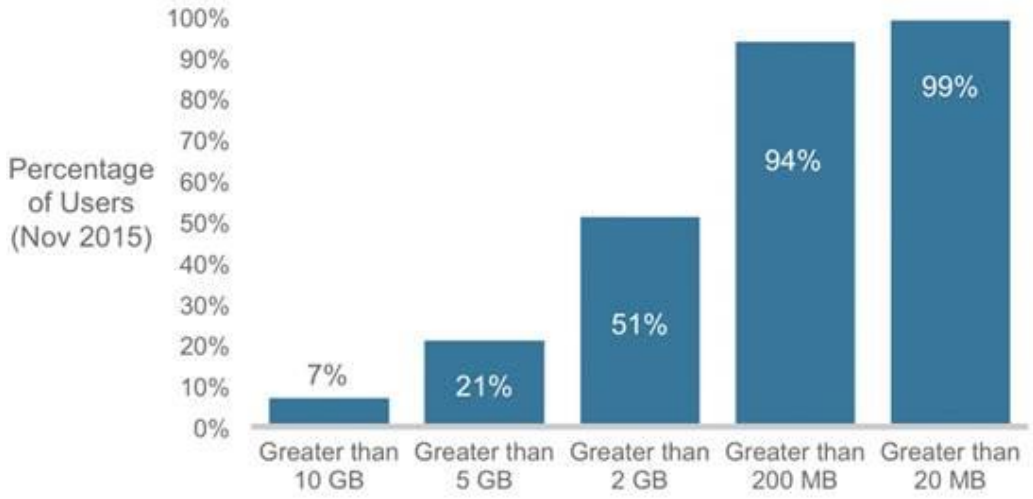

图 1.5 移动用户每月使用流量分布[22]

1.7　资源分配问题

得益于云端的额外储存能力和计算资源，移动云计算使得资源欠缺的移动终端能够处理资源消耗量大的应用。但是，如果没有无线资源和计算资源的协调，移动云计算很难充分发挥其优势。无线资源和计算资源的耦合问题在多用户的场景下非常具有难度的。本文移动云计算中的资源管理研究借鉴传统无线资源分配和蜂窝网中的移动性管理理论和技术。

传统无线资源分配负载均衡问题大体可以分为两类：第一类是具体的小区设计、用户关联算法优化，建模成一个比较复杂的非凸的优化问题，然后通过近似算法之类的数值方法求解；参考文献[23]通过最大化全网络的效用函数优化了用户关联机制；参考文献[24]通过调整小区大小和形状提高系统容量的同时保持最小传输功率。第二类是系统参数优化，一般是在求闭式解，如参考文献[25]基于泊松点过程（Poisson Point Process, PPP)模型分析了最优的宏基站和微基站密度，文献[26]在六边形异构网中优化了信道分配机制。

关于移动云计算资源分配的最新研究有文献[27]中分析了分布式移动云计算中静态的无线和计算资源联合分配问题。场景是 Nb 个基站，Nc 个云，K 个用户，在满足最大延时的条件下，通过分配无线资源和计算资源最小化手机发射能耗 E，把问题建模成一个 NP 难的组合规划问题。但该建模假设每个用户的任务量要么全部在移动终端计算，要么全部上传到云端去处理，而没有涉及任务分割的概念，这样无法最大化利用云端资源。

移动云计算区别于传统云计算的主要不同点在于无线网络的随机性以及用户的移动性。无线网络的随机性会对任务计算的最终延时造成影响，而移动云计算中的移动性管理问题同样和传统的移动性管理有着本质区别：一个用户在移动中逐渐远离基站 1，而靠近基站 2。不同于传统切换会选择接到提供更好 SINR 的基站，这里还需要考虑所谓的云切换，就是云端计算带来的影响，即需要综合考虑无线资源和云端资源的分配，或者考虑云切换，尤其当用户在移动过程中从基站 1 切换到基站 2 但是在基站 1 对应的服务器端有该用户的计算任务正在进行，这里会涉及到虚拟机迁移以及基站和云之间后传网的传输开销。

整体而言，移动云尽管解决了终端存储和计算资源不足的问题，但同时也面临着延时受限、终端能耗受限以及无缝切换这几个瓶颈。因此，研究如何更好地配置无线资源和计算资源具有重要意义。而且从现有研究来看，学者们主要考虑移动云场景中单用户的任务调度策略，缺少对多用户场景同时支持任务分割的系统考虑，也缺乏同时考虑无线和计算资源的切换研究。因此，对移动云计算中针对多用户的资源分配以及用户移动性的分析是很有必要的。

1.8　论文主要内容和结构安排

上文分析了移动云计算中的相关概念、需要解决的终端能耗问题以及传统的无线资源分配管理方法。移动云尽管目标是解决终端存储和计算资源不足的问题，但同时面临着延时受限、终端能耗受限以及如何无缝切换这几个瓶颈。因此，移动云计算中的高能效资源分配成为一个很有意义的重要课题。此外，现有研究主要考虑移动云场景中单用户的任务调度策略，或者云端资源不受限的情况从而用户之间没有互相的耦合制约关系,所以本文为了满足用户的延时需求，主要考虑离用户距离近但资源相对有限的本地云（也称作近端云，边缘云或者微云），讨论多用户移动云计算场景中对无线资源和计算资源的分配以及移动性管理。

第二章旨在能够减少终端的能耗，为多用户在使用移动云计算时提供了无线资源和计算资源的集中式联合分配机制，并且，基于延时约束和不同移动终端的应用类型大小，为资源分配提供一种启发的策略。数值结果表明，与没有负载转移移动计算的制度相比，我们设计的负载转移策略能够在满足用户延时约束的同时在系统中有 3 个移动终端的情况下有效减少

40%的移动终端能耗。此外，算法的性能在复杂度低的情况下接近于搜索算法。这部分工作已经整理为 2015 年 IEEE ICCC 的文章。

考虑到集中式联合资源分配需要用户和中心基站之间大量的信令开销，第三章针对耦合的多用户联合优化问题，提出了将多用户问题解耦分解的分布式资源分配策略，利用对偶函数将原多用户耦合的优化问题分解为每个用户以及云端的分别优化。分布式的资源分配策略主要有传输信令开销小和复杂度低的优点。

由于云提供商的目标终究是最大化自己的收益，所以在第四章我们用经济学的方法研究云提供商通过调整云资源的价格来分配云端有限资源，同时尽可能满足各个用户的延时需求并降低终端能耗。通过求解云提供商和用户之间博弈的纳什均衡点得到系统平衡时的资源分配情况。

不同于传统蜂窝网的移动性管理，移动云计算中的除了要解决无线端的基站切换还要考虑云端计算任务的切换，即云切换。在第五章，针对移动云计算中特殊的移动性管理问题，设计了相应的切换机制流程，综合考虑无线端和云端的延时来降低用户切换中的延时及开销。

最后一章对研究内容进行总结，并展望未来可以展开的研究方向。

第 2 章　多用户资源受限场景中的集中式高能效资源分配

2.1　本章引论

移动云计算（MCC）的出现是为了处理资源消耗量大的应用与资源缺乏的移动终端之间紧张的不匹配关系。这种灵活的技术使得用户可以通过无线连接从云端获取到相对终端更多的计算资源和存储资源。尽管近些年移动云计算发展迅猛，但同样也面临一些技术瓶颈，其中就包括移动终端的电池续航能力以及不同应用的延时需求，而这都会极大限制系统的性能。[12]近些年，移动终端上出现了各种各样媒体影音的应用，吸引了越来越多的用户的同时也促使新型移动应用的不断更新，还带来了指数式的数据量增长。然而因为有限的物理尺寸大小，移动终端则要受限于电池的续航能力。人们的智能手机几乎每天都需要充电。[27]另一方面，下一代的移动通信系统（5G）旨在达到毫秒级的延时和相对 4G 一百倍数据率的性能指标，所以研究人员集中了更多精力在移动用户的延时需求上。[1]因此，在解决移动云计算中的终端能耗和延时需求问题中，合理地分配有限资源是非常有必要的。

为了在资源池中实现灵活的资源分配，以及协助终端降低能耗[29]，近些年研究者们提出了不同移动云计算架构，例如 Cloudlet[30]，Clone Cloud[31]，和 CONCERT[14]，这其中以分布式移动云架构[16]最为突出。具体来讲，小蜂窝基站连接到有限计算存储资源的微云。如果用户的需求可以通过本地微云得到满足，那么所有任务都可以在本地微云完成；否则，基站则会向远端具有更大计算存储能力的互联网云提供商寻求帮助。通过这样的方式，无线资源和计算资源都离用户终端更近从而更容易满足用户的延时需求，这与 CONCERT 框架中的原理非常相似。

需要强调的是，负载转移(offloading)和任务分割（partitioning）的具体流程方法会随用户的不同应用而有所不同。本章只考虑应用程序开发的时候就支持任务分割的情形，分割的开销主要有延时和终端能耗两个方面。为了着重体现出在移动云计算中负载转移的本质折中关系，假定用户的任务可以在移动终端和微云端并行处理，[32]而没有具体研究相关分割算法的详细细节。

移动云计算中的资源分配一直是一个重要的研究问题。在文献[33]，数

据中心的任务分配问题被建模为一个马尔可夫决策过程（Markov Decision Process, MDP）并提出了一个指数策略进行任务调度。但是，这篇文章只包括云端数据中心的任务分配。文献[32]分析了在单一用户场景下计算和无线资源的优化。文献[27]、[34]、[35]分析了多用户场景，并且假定该系统中所有的移动用户都已经决定将其所有的任务都负载转移至移动云端去处理。不同于之前的参考文献，我们考虑多用户的场景，并且假定每一个用户的任务可以部分转移到云端处理。我们试图为系统设计最优的资源分配策略，从而在满足用户延时需求的同时尽量降低用户终端能量消耗。

本章的主要贡献包括：

1）建模了优化问题分析在资源受限多用户移动云计算场景下如何进行无线资源和计算资源的联合分配。

2）设计了一个基于用户计算任务和延时需求的，易于实施的资源分配策略。数值结果表明设计的机制性能接近搜索算法。

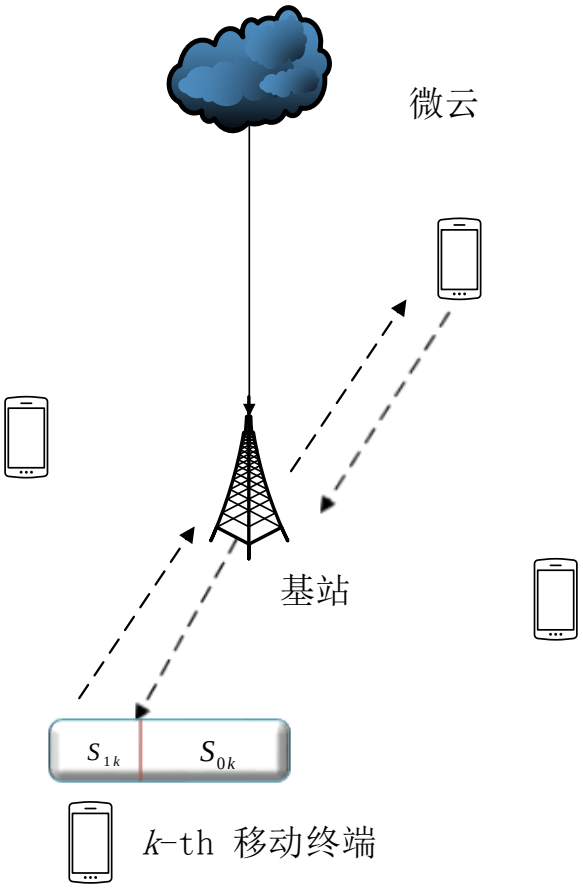

图 2.1 问题场景

2.2　系统模型

不失一般性，我们考虑一个基站连接到有一些存储资源和计算资源的微云，系统中有K个用户，如图 2.1 所示。每一个用户都有一个应用需要在延时界L_k内完成。用户可以选择是否去向微云请求计算服务。在需要部分负载转移的情况下，应用支持任务分割，这意味着用户的任务都可以被分割为任意大小的两部分。一部分将在终端完成，另一部分转移到微云中去完成。这里假设任务分割不带来任何开销是为了简化问题，这样的分割模型对最终系统性能（延时和终端能耗）在一定程度上是实际情况的下界或者最好情况。因此分割模型可以描述如下[32]:

$$S_{0k}+S_{1k}=S_k. \tag{2-1}$$

S_k是第 k 个用户端应用总的大小，其中S_{0k}部分在用户端执行，S_{1k}部分通过无线网负载转移至微云端完成。

在第 k 个移动终端完成任务所需的时间与在终端执行的任务量成比例:

$$\tau_{\mathrm{k}}S_{0k}.$$

假设第 k 个用户完成一个 1 *bit* 任务量所需的时间是τ_{k}，τ_{k}的具体值取决于终端 CPU 的运行速度和应用的复杂程度。

类似地，用于处理非负载转移部分任务（在移动终端处理的任务）的能量被设为：

$$\xi_{\mathrm{k}}S_{0k}.$$

其中ξ_{k}代表着第 k 个移动终端完成 1 *bit* 计算任务产生的终端能量消耗。这里ξ_{k}的大小也取决于终端 CPU 的运算速度和用户处理的应用种类。

当需要有负载转移到云端计算时，第 k 个用户上传的任务部分执行需要的总延时Δ_k包括如下四个部分：

$$\Delta_k=\Delta_k^t+\Delta_k^e+\Delta_k^r+T_B. \tag{2-2}$$

这里Δ_k^t是上传到云端去执行任务所需的状态和输入所需的上行传输时间。我们假设上行传输的总比特数为$\mu_{\mathrm{uk}}S_{1k}$，其中系数$\mu_{\mathrm{uk}}>0$是表示用户k

在上行传输时对计算任务量的压缩或者增加的开销比例。利用香农公式（Shannon Formula）计算得到上行传输速率，然后上行传输时间 Δ_k^t 可以表示为：

$$\Delta_k^t = \frac{\mu_{\mathrm{uk}} S_{1k}}{B_k \log_2(1+\frac{P_k g_k}{N_0 B_k})}, \tag{2-3}$$

其中 B_k 是分给第 k 个移动终端的无线频谱带宽，P_k 代表上行传输的终端发射功率。g_k 代表第 k 个移动终端和基站之间的无线信道增益。N_0 是热噪声的功率谱密度。

Δ_k^r 是将计算得到的结果返回到第 k 个移动终端所需要的时间。通常来说由于基站的发射功率远大于用户终端的发射功率，所以下行传输时间相对上行传输时间要小得多，在总延时计算中可以忽略。

Δ_k^e 代表在云端处理第 k 个移动终端的 S_{1k} 任务量所需的时间。假定第 k 个移动终端获取的计算资源（CPU 频率）是 f_k，那么

$$\Delta_k^e = S_{1k} / f_k. \tag{2-4}$$

后传网（backhaul）的传输时间 T_B 这里建模为一个常量，取决于基站和微云之间的距离和连接方式（例如光纤）。

2.3　问题建模

我们的最终目标是要为 K 个移动用户提供高能效的资源分配策略，从而使得在满足每个用户的延时约束的条件下最小化终端能耗。这里的终端能量消耗包括无线传输能耗和在终端处理任务的能耗。可以调整的有无线资源、计算资源以及每个用户上传到云端的任务量大小。具体说来，我们在问题建模中使用频分复用（FDM），并将全部带宽切分成若干子频带（sub-bands）供各个用户使用，从而避免用户之间的无线信号干扰。微云端的服务器支持多任务同时运行，这样 K 个移动用户可以共享云端的计算能力。用户终端可以选择是否要分割应用以及有多少任务需要负载转移。我们考虑的是在用户终端和微云的计算过程可以并行进行，为了简化分析，任务分割和处理后的整合不考虑引入任何开销（或者可以认为引入固定的能耗和延时开销）。

我们的目标是最小化每一个移动终端的能量消耗加权和：

$$E=\sum_{k=1}^{K}\beta_k E_k. \tag{2-5}$$

其中系数 $\beta_k \geq 0$ 是加权因子，它会为满足不同移动终端的电池能力或对终端能耗的敏感度而调整。

$$E_k = P_k \Delta_k^t + \xi_{0k} S_{0k}. \tag{2-6}$$

是第k个移动终端的能量消耗，它包含无线传输和本地终端计算的能量消耗。P_k 是第k个移动终端的传输功率。由于基站的发射功率远大于用户移动终端的发射功率，下行传输中的延时和能量消耗相对上行要小很多，所以可以忽略不计。

当用户 k 在移动终端处无法在延时界 L_k 内完成自己的计算任务量时，则用户不得不借助云端计算资源去计算，此时基站和云端必须给用户分配一定的无线资源和计算资源；当用户 k 的延时界相对较松时，用户可以选择在终端还是云端完成计算任务，这时上传到云端处理计算任务更多是为了节省终端能耗。

基于以上分析，我们将多用户资源分配问题建模为如下优化问题：

$$\begin{aligned}
&\min_{S_{0k},S_{1k},P_k,B_k,f_k} E=\sum_{k=1}^{K}\beta_k E_k \\
&s.t.\ i) S_{0k}+S_{1k}=S_k,\ \forall k \\
&\quad ii) \max\{\tau_{0k}S_{0k},\Delta_{knm}\}\leq L_k,\ \forall k \\
&\quad iii) P_k \leq P_{kt},\ \forall k \\
&\quad iv) \sum_{k=1}^{K} f_k \leq F, \\
&\quad v) \sum_{k=1}^{K} B_k \leq B
\end{aligned}$$

所列 5 个约束条件的含义分别如下：

i）第k个移动终端的应用可以以任意比例被分割成两个部分：一部分在本地终端运行，另一部分上传到云端去完成；

ii）在各个移动终端以及在微云的任务运行过程可以同时进行。在两边的执行时间都必须在延时界 L_k 以内；

iii）第k个移动终端的发射功率不得超过最大发射功率 P_{kt}；

iv）分配到各个应用的计算资源 f_k 的总和不得超过云端可提供的最大计算能力 F；

v）分配给每一个移动终端的子带宽 B_k 的总和不得超过全部带宽 B；

2.4　问题简化

然而很不巧，上面建模的最优化问题是非线性规划，它的目标函数和约束函数都是非凸的。不过，问题存在一些好的性质，利用这些性质我们可以简化原始的资源分配问题：

根据约束 *i*)，把 S_{0k} 用 S_{1k} 表示成：

$$S_{0k} = S_k - S_{1k} \tag{2-7}$$

从而可以在优化问题中删除优化变量 S_{0k}。

虽然优化问题针对所有的优化变量是非凸的，但针对每个用户的任务上传量 S_{1k} 其实是线性规划问题，S_{1k} 仅在目标函数和前两个约束中出现。通过解该线性规划问题，S_{1k} 可以被 P_k，f_k ，B_k 用分段函数表示：

$$S_{1k} = \begin{cases} \max\{0, S_k - \dfrac{L_k}{\tau_k}\} & if \ \dfrac{P_k \mu_{uk}}{B_k log_2(1+\dfrac{P_k g_k}{N_0 B_k})} > \xi_k \\ \dfrac{L_k}{\dfrac{\mu_{uk}}{B_k \log_2(1+\dfrac{P_k g_k}{N_0 B_k})} + \dfrac{1}{f_k}} & \text{else} \end{cases} \tag{2-8}$$

2.5　基于任务和延时的资源分配策略

尽管我们利用问题针对单一变量 S_{1k} 的线性特性去简化问题，但因为延时表达中的分数形式，要获得一个一般的封闭式解并非易事。我们可以使用穷举搜索去获得一个目标解决方案。但是，因为穷搜算法的计算复杂度太大，特别是当用户数量太多时，我们转而去寻找一个启发式的算法从而达到一个次优的解决方案。事实上，正如上文问题简化中所提到的，S_{1k} 和 S_{0k} 都可以通过分配给第 k 个移动终端的无线资源 B_k 和计算资源 f_k 得以表达。所以，这里的关键是要找到一个资源分配机制去解决问题。要注意的是在本部分我们不考虑功率控制，只是将分配给各用户的带宽作为无线资源，从而便于研究移动云计算中无线资源和计算资源的对应关系。对能耗更敏感（有着更大 β_k）以及有适当延时界的移动终端应当有机会获取更多资源，基于这样的考虑，我们假定资源应当按照每个用户的

$\beta_k(\tau_k S_k - |\tau_k S_k - L_k|)$成比例分配，全部任务在终端计算时间$(\tau_k S_k)$接近延时界$(L_k)$的用户获得更多的频谱资源和计算资源。

根据以上资源分配方式和之前结论即S_{1k}和S_{0k}可以用f_k和B_k表达，在算法 2.1 中提出一种启发式的资源分配机制从而尽可能降低移动终端的能耗。

Algorithm 1 Task & Delay based Resource Allocation Scheme

for $k = 1$ to K **do**

$$B_k = \frac{\beta_k(\tau_k S_k - |\tau_k S_k - L_k|)}{\sum_{i=1}^{K} \beta_i(\tau_i S_i - |\tau_i S_i - L_i|)} B$$

$$f_k = \frac{\beta_k(\tau_k S_k - |\tau_k S_k - L_k|)}{\sum_{i=1}^{K} \beta_i(\tau_i S_i - |\tau_i S_i - L_i|)} F$$

end for

if $\frac{P_k \mu_{uk}}{B_k \log_2(1+\frac{P_k g_k}{N_0 B_k})} > \xi_k$ **then**

$$S_{1k} = \max\{0, S_k - \frac{L_k}{\tau_k}\}$$

else

$$S_{1k} = \frac{L_k}{\frac{\mu_{uk}}{B_k \log_2(1+\frac{P_k g_k}{N_0 B_k})} + \frac{1}{f_k}}$$

end if

return $E = \sum_{k=1}^{K} \beta_k E_k$

算法 2.1 基于任务和延时的资源分配策略

2.6 数值结果

在这一部分，我们采用上面设计的资源分配策略并讨论数值结果。表 1 列出了仿真所采用的参数。其中移动终端的能耗和延时参数来源于文献[36]。我们设定所有的移动终端都有同样的信道条件以及相同的任务处理能力（各个终端处理 1 比特任务的时间和能耗都相同）。利用参考文献[36]中的 N810 设备的参数，我们给出这里参数设置中每个移动终端的ξ_k和τ_k。需要注意的是，我们假定在微云的 CPU 计算速度 F 比在用户终端的计算速

度快 4 倍（这可以通过同时装配 5 个相同处理能力的服务器来实现）。不失一般性，我们固定第一个移动终端的参数，并通过改变其他移动终端的应用大小和延时需求来研究系统内全部终端能量消耗的变化。参考文献[32]中的设置，设定第一个移动终端的计算任务大小为$S_1 = 5Mb$，而其他$S_k, k \neq 1$则是S_1的整数倍。

表 2.1 系统参数设置

参数	取值
终端发射功率$(P_k, \forall k)$	$0.2W$
总频谱带宽(B)	$20MHz$
云端总计算资源(F)	$5 \times 10^7 cycles/second$
$\xi_k, \forall k$	$8.6 \times 10^{-8} J/bit$
$\tau_k, \forall k$	$10^{-7} s/bit$
$\mu_{uk}, \forall k$	1
$\beta_k, \forall k$	1
$\frac{g_k}{N_0}, \forall k$	$8Hz/W$

图 2.2 和图 2.3 分别展示了在系统中有两个移动终端时分配给移动终端 2 的带宽资源和计算资源情况。我们采用前面设计的基于任务和延时的资源分配策略以及穷搜算法去分析延时约束和资源分配的关系，变化第二个移动终端处的计算任务量从而得出两组曲线。这里设定第一个移动终端的延时界$L_1 = 0.4s$。比较图 2.2 中用户 2 分配到的无线资源和图 2.3 中用户 2 分配到的计算资源可以发现，分配给用户 2 的计算资源会与分配的带宽资源波动趋势相似。这个现象比较容易理解，因为移动终端通过无线连接负载转移越多的任务，就会要求在云端分配到越多的计算资源去处理计算任务。具体说来，当用户 2 的延时约束L_2小于在移动终端完成全部任务所需的时间$\tau_2 S_2$时，分配到的无线资源和计算资源会随延时约束的放松而增加。在这样的情况下，用户 2 不得不向云端寻求帮助以满足延时要求。延时约束越宽松可以允许移动终端负载转移越多的任务到云端以节约终端能耗，这样的情况下更多负载转移的任务量通常需要更多的无线资源和计算资源供传输和计算。然而，当L_2大于$\tau_2 S_2$时，继续增大L_2会导致用户 2 对无线和

计算资源需求的降低。这是因为此时用户 2 有相对宽松的延时约束，用户 2 无需太多的资源就可以满足延时要求，因此可以留出更多的资源给系统中的其他延时界更紧的用户，从而降低系统中总的终端能耗。而且，我们所提出的基于任务量和延时的任务分配机制的波动趋势和穷搜算法的结果在图 2.2 和图 2.3 中非常吻合，这也证实了我们之前的推理。

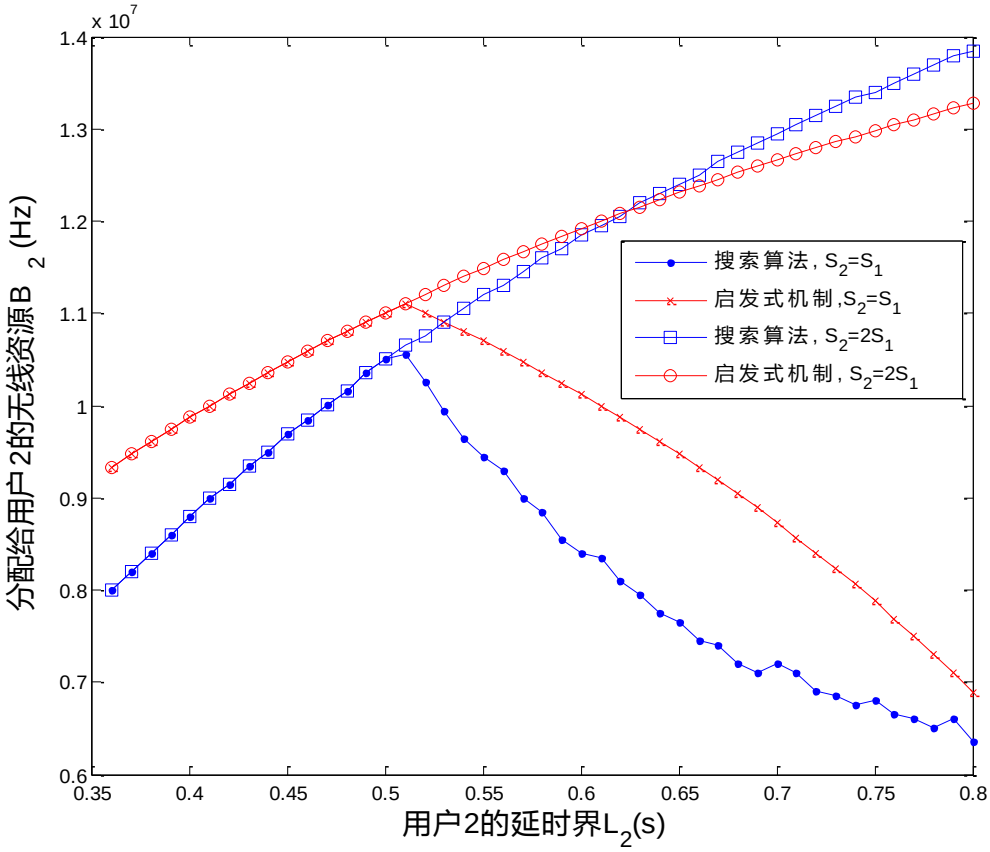

图 2. 2　系统中有两个用户时给用户 2 分配的无线带宽资源

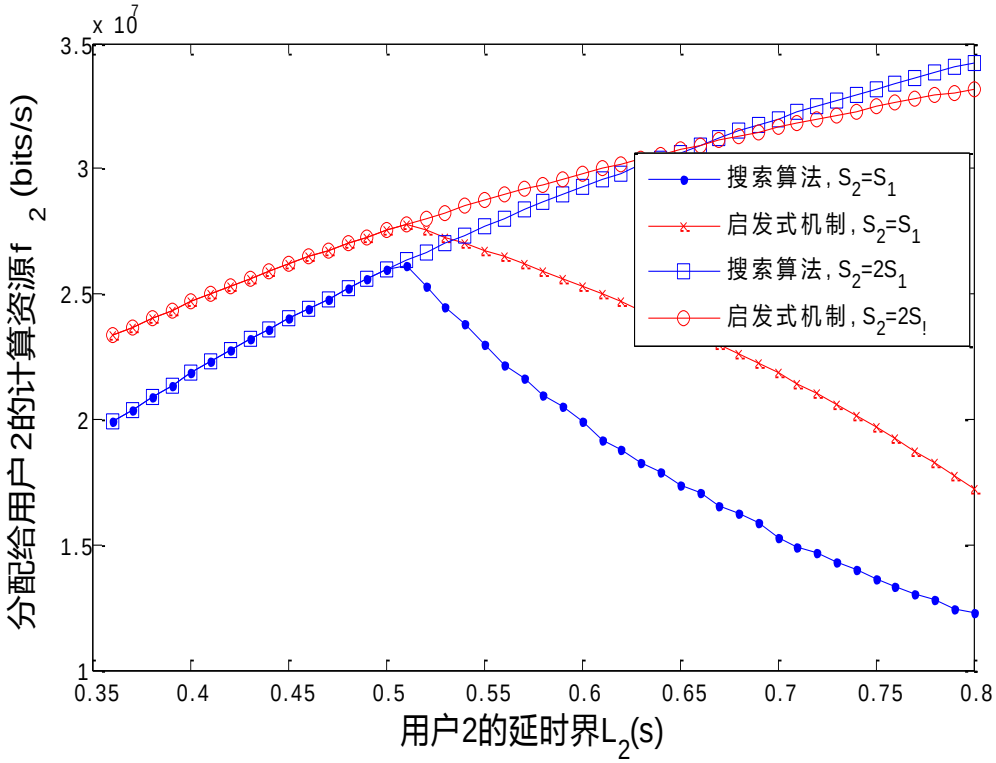

图 2. 3　系统中有两个用户时给用户 2 分配的计算资源

图 2.4 展示了用户的延时要求对于终端能量消耗的影响。随着用户 2 延时约束L_2的增加，终端总能耗会显著降低。这表明移动用户在牺牲处理延时的情况下，可以通过移动云计算 帮助节省相当可观的终端能耗。在移动终端处理 2 个大小为$S_1 = 5Mb$的应用需要消耗 0.86W 的终端能耗，然而利用移动云计算负载转移之后，需要的终端能耗小于 0.2W，用户实则在处理同样规模的任务时借助移动云计算技术降低了超过 75%的终端能耗。同样，从图 2.4 中可以看出，我们设计的启发式资源分配策略性能相当好，特别是在延时约束相较于在移动终端处理所有任务所需的时间相对严格的情况下性能尤其接近搜索算法。

表 2.2 对比了在系统中有 3 个用户时，提出的启发式资源分配策略和搜索算法的性能表现。设定 3 个移动终端的总计算任务量均为$S_k = 5Mb, \forall k$。关于表中的二维数组，每一组中的第一个数是通过穷搜得出的总终端能耗结果，第二个数则是采用我们设计的启发式资源分配策略得出的终端能耗结果。具体而言，相较于没有负载转移的情况，我们提出的资源分配平均减少了 40%的终端总能耗，同时我们的资源分配策略在 3 个用户的情况下比搜索算法的最终终端总能耗数值结果差 10%。不过，我们提出的资源分配策略只有 $O(K)$的（K 是系统中总的用户数）计算复杂度，所以

具有实际可行性，然而原始的优化问题是非凸的分式优化问题，搜索算法的计算则太过于复杂。

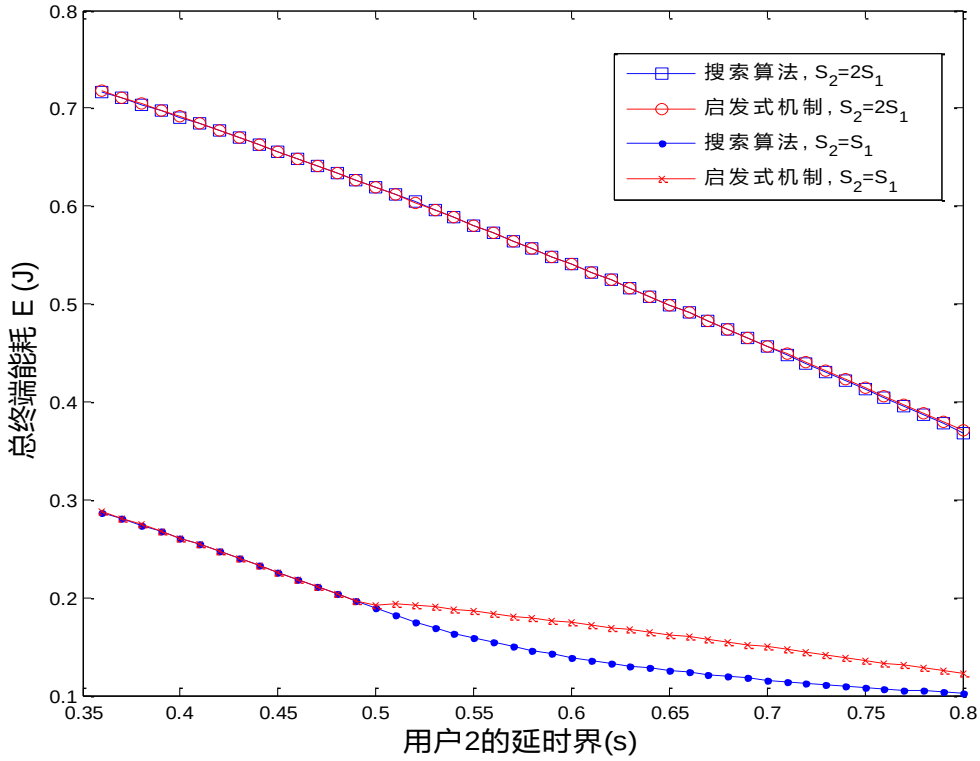

图 2.4 系统中有两个用户时总的终端能耗

表 2.2 三个用户时终端总能耗

E/J L_2/s / L_1/s	0.55	0.65	0.75
0.55	(0.5947,0.5973)	(0.5572,0.5581)	(0.5272,0.5460)
0.65	(0.5275,0.5509)	(0.5060,0.5134)	(0.4731,0.5021)
0.75	(0.4731,0.4967)	(0.4464,0.4558)	(0.4125,0.4405)

2.7　本章小结

在本章中，我们解决了移动云计算中在满足用户延时需求的同时最小化用户终端能耗时分配有限无线资源和计算资源的问题。我们建模了一个带有非线性延时约束的优化问题，利用问题针对部分变量的线性特征简化问题，根据用户的延时约束和每个用户对能耗的敏感程度，我们提出了基于

任务和延时的资源分配策略。即用户延时界(L_k)接近全部任务在终端计算时间($\tau_k S_k$)的用户获得更多的频谱资源和计算资源。我们提出的资源分配策略非常容易实施，只有 $O(K)$的复杂度，数值结果表明其具有良好的性能。未来的工作中，我们可以进一步考虑在多基站和多微云场景下的资源分配问题。本章节主要内容已经整理成会议论文，发表在 2015 年的 IEEE ICCC(International Conference on Communications in China）会议上。

第 3 章　基于优化问题分解的分布式高能效资源分配

3.1　本章引论

在多用户移动云场景中，为了在满足用户延时需求的同时最小化终端能耗，在第 2 章将系统中的有限资源分配问题建模为带有延时约束的中心优化问题，然而在实际系统中，中心的优化控制需要在云端搜集用户的总任务量大小，延时需求，对能耗的敏感程度以及信道状况等详细终端信息，这会带来较大的信令开销，所以本章考虑将集中式资源分配问题分解为 K（系统总用户数）个子问题进行优化，进而设计分布式资源分配算法。

本章的主要贡献如下：

优化问题分解的核心观点是将原本的中心控制问题利用拉格朗日对偶函数分解成为分布式的针对每一个用户的子问题，然后再通过少量信令传输由一个云端控制的主问题（the master dual problem）来得到全局最优的资源分配策略。[37]

3.2　系统模型

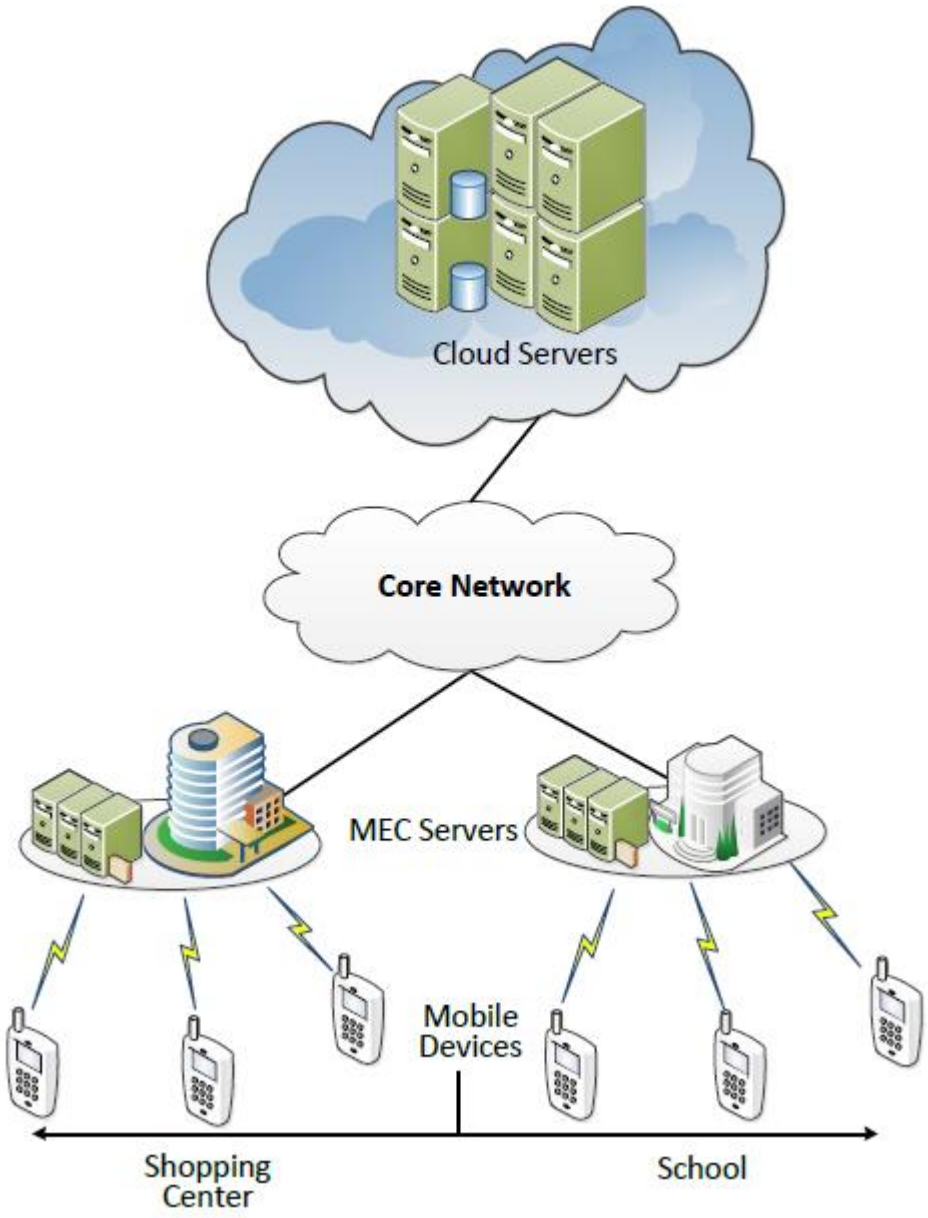

图 3.1 移动边缘云计算架构[38]

我们依然考虑移动边缘云计算架构（Mobile Edge Computing）[38]，如图 3.1 所示。不失一般性，我们考虑一个基站储备有若干存储资源和计算资源，系统中共有K个用户，其中每一个用户都有一个应用需要在规定时间内完成。是否去向边缘云请求计算服务由用户自行选择，在需要部分负载转移的情况下，应用支持任务分割(task partitioning)，即用户的任务都可以

被分割为任意大小的两部分，分别在终端完成，和转移到云中完成。为了简化问题，我们假设任务分割不带来任何开销。在一定程度上，这样的分割模型对最终系统性能（延时和终端能耗）是实际情况的下界或者最好情况。因此分割模型可以描述如下[32]:

$$S_{0k}+S_{1k}=S_k. \tag{3-1}$$

S_{0k} 部分是用户在终端执行任务的大下，S_{1k} 部分通过无线网负载转移至微云端完成，S_k 是第 k 个用户端应用总的大小。

在第 k 个移动终端完成任务所需的时间与在终端执行的任务量成比例:

$$\tau_k S_{0k}.$$

假设第 k 个用户去完成 1 bit 任务量所需的时间是 τ_{k}，τ_k 的具体值由终端 CPU 的运行速度和应用的复杂程度来决定。

与之相类似，用于去处理非负载转移部分任务（在移动终端处理的任务）的能量被设为：

$$\xi_{\mathrm{k}} S_{0k}.$$

其中 ξ_{k} 代表第 k 个移动终端完成 1 bit 计算任务产生的终端能量消耗。这里 ξ_{k} 的大小也取决于终端 CPU 的运算速度和用户处理的应用种类。

当需要有负载转移到云端计算时，Δ_k 代表第 k 个用户上传的任务部分执行需要的总延时，其由四个部分构成：

$$\Delta_k=\Delta_k^t+\Delta_k^e+\Delta_k^r+T_B. \tag{3-2}$$

其中，第一项 Δ_k^t 是上传到云端去执行的任务部分 S_{1k} 所需的状态和输入的上行传输时间。假设上行传输的总比特数为 S_{uk}，表示用户在上行传输时必须上传的程序包大小。这里假设给第 k 个用户分配固定的上传容量 C_k，则上行传输时间 Δ_k^t 可以表示为：

$$\Delta_k^t=\frac{S_{uk}}{C_k}. \tag{3-3}$$

第二项 Δ_k^r 是将计算得到的结果返回到第 k 个移动终端所需要的时间。通常情况，基站的发射功率远大于用户终端的发射功率，所以，相对

上行传输时间，下行传输时间要小得多，小到在总延时计算中可以忽略不计。

第三项 Δ_k^e 代表在云端处理第 k 个移动终端的 S_{1k} 任务量所需的时间。假定第 k 个移动终端获取的计算资源（CPU 频率）是 f_k，那么

$$\Delta_k^e = S_{1k} / f_k. \tag{3-4}$$

第四项 T_B 应当是一个常量，代表后传网（backhaul）的传输时间，取决于基站和微云之间的距离和连接方式（例如光纤）。

3.3　问题建模

集中式资源分配问题是在满足用户延时需求的同时最大化利用云端的有限资源从而降低终端总能耗：

$$\begin{aligned} &\min_{f_k, S_{1k}} \sum_{k=1}^{K} E_k \\ &\text{s.t.} \sum_{k=1}^{K} f_k \le F \\ &\tau_k (S_k - S_{1k}) \le L_k \\ &\frac{S_{1k}}{f_k} + \frac{\mu_{uk} S_{1k}}{S_k C_k} \le L_k \end{aligned} \tag{3-5}$$

优化目标是最小化 K 个用户的终端总能耗，代入各变量的表达式并且整理可得：

$$\begin{aligned} &\min_{f_k, S_{1k}} \sum_{k=1}^{K} \left(\frac{P_k \mu_{uk}}{S_k C_k} S_{1k} + \xi_k (S_k - S_{1k}) \right) \\ &\text{s.t.} \sum_{k=1}^{K} f_k \le F \\ &\frac{\tau_k S_k - L_k}{\tau_k} \le S_{1k} \le \frac{L_k}{\frac{\mu_{uk}}{S_k C_k} + \frac{1}{f_k}} \end{aligned} \tag{3-6}$$

(3-6)中第二个关于用户 k 上传任务量 S_{1k} 的约束，左边代表为了满足用户的延时需求，至少应该上传的任务量大小，而右边意思是当用户 k 被分配到 f_k 的计算资源时最多能上传的任务量多少。显然，如果(3-6)中没有

$\sum_{k=1}^{K} f_k \le F$ 的约束，优化问题即可分解为针对 K 个用户的分别优化。因此，先考虑其对偶问题：

$$\begin{aligned}&\min_{f_k,\mathrm{S}_{1k}} \sum_{k=1}^{K}\left(\frac{P_k\mu_{uk}}{S_kC_k}S_{1k}+\xi_k(\mathrm{S}_k-\mathrm{S}_{1k})\right)+\lambda\sum_{k=1}^{K} f_k - F\\ &\mathrm{s.t.}\ \frac{\tau_k S_k - L_k}{\tau_k} \le S_{1k} \le \frac{L_k}{\frac{\mu_{uk}}{S_kC_k}+\frac{1}{f_k}}\end{aligned} \tag{3-7}$$

这样就可以将优化问题(3-5)解耦分解为两层优化，用户终端的优化以及云端的优化。

在每个用户端：

$$\begin{aligned}&\min_{f_k,\mathrm{S}_{1k}} \frac{P_k\mu_{uk}}{S_kC_k}S_{1k}+\xi_k(\mathrm{S}_k-\mathrm{S}_{1k})+\lambda f_k\\ &\mathrm{s.t.}\ \frac{\tau_k S_k - L_k}{\tau_k} \le S_{1k} \le \max\left\{\frac{L_k}{\frac{\mu_{uk}}{S_kC_k}+\frac{1}{f_k}}, S_k\right\}\end{aligned} \tag{3-8}$$

当 $\xi_k - \frac{P_k\mu_{uk}}{S_kC_k} \ge 0$ 时，意味着对于第 k 个用户，将任务上传到云端可以节省终端能耗，从而可以求出第 k 个用户的任务上传量：

$$S_{1k} = \frac{L_k}{\frac{\mu_{uk}}{S_kC_k}+\frac{1}{f_k}} \tag{3-9}$$

将(3-9)代入(3-8)优化问题的约束条件可得：

$$\min\left\{0, \frac{1}{\frac{L_k\tau_k}{\tau_kS_k-L_k}-\frac{\mu_{uk}}{S_kC_k}}\right\} \le \mathrm{f}_k \le \frac{S_k}{L_k-\frac{\mu_{uk}}{C_k}} \tag{3-10}$$

上式(3-10)左边代表为了让用户 k 在延时界 L_k 内完成 S_k 的计算任务量，必须至少需要请求的计算资源量；而右边则代表将全部任务量 S_k 都上传到云端所需要的计算资源量大小。

将(3-9)式代入(3-8)优化问题的目标函数，并令 $\theta_k = \frac{P_k\mu_{uk}}{S_kC_k}$ 得：

$$\begin{aligned}&\min_{f_k} \mathrm{h}(f_k)=(\theta_k-\xi_k)\frac{L_k}{\frac{\mu_{uk}}{S_kC_k}+\frac{1}{f_k}}+\lambda f_k\\&\mathrm{s.t.}\min\{0,\frac{1}{\frac{L_k\tau_k}{\tau_kS_k-L_k}-\frac{\mu_{uk}}{S_kC_k}}\}\le \mathrm{f}_k\le\frac{S_k}{L_k-\frac{\mu_{uk}}{C_k}}\end{aligned}\tag{3-11}$$

通过对$\mathrm{h}(f_k)$求导可以求得：

$$h'(\mathrm{f}_k)=\lambda-(\xi_k-\theta_k)\frac{L_k(\mathrm{S}_k\mathrm{C}_k)^2}{(\mathrm{f}_k\,\mu_{uk}+\mathrm{S}_k\mathrm{C}_k)^2}=0\tag{3-12}$$

$$f_{\mathrm{k}}=\frac{S_kC_k(\sqrt{\frac{L_k(\xi_k-\theta_k)}{\lambda}}-1)}{\mu_{uk}}\tag{3-13}$$

上式说明λ越大，用户请求的计算资源量则越小，λ类似于惩罚因子或者是价格因子，总计算任务量S_k越大以及延时界越松的用户会请求更多的计算资源。

中心控制端的主对偶问题（master dual problem）通过解下面的对偶问题负责更新对偶变量λ：

$$\begin{aligned}&\max_{\lambda} j(\lambda)=\sum_{k=1}^{K}(\frac{P_k\mu_{uk}}{S_kC_k}S_{1k}+\xi_k(\mathrm{S}_k-\mathrm{S}_{1k})+\lambda f_k)\text{-}F\\&s.t.\lambda\ge 0\end{aligned}\tag{3-14}$$

云端计算资源总量F越大，为使问题(3-14)中的$j(\lambda)$最大，λ会越小，根据(3-13)各用户相应请求更多的计算资源。

由于(3-11)中优化问题关于f_k是凸的，并且满足 Slater 条件，故强对偶性成立，对偶问题的解就是原问题的解。

终端能耗则相对不借助移动云计算节省了$\frac{L_k}{\frac{\mu_{uk}}{S_kC_k}+\frac{1}{f_k}}\times(\xi_k-\frac{P_k\mu_{uk}}{S_kC_k})$。

3.4　分布式资源分配策略

本小节，我们设计一种分布式的资源分配机制，从而减少用户和基站之间的信令开销，云端只需要知道用户的计算资源请求量，而用户只需要

知道基站公布的 λ 的值。

根据 3.3 小节，我们可以得到 f_k 的取值，利用梯度下降法可以求得(3-14)任意精度（通过调整步长值）的解：

表 3.1 分布式资源分配策略

分布式资源分配策略
一、各用户求得 f_k：
if $\min\{0,\frac{1}{\frac{L_k\tau_k}{\tau_k S_k - L_k}-\frac{\mu_{uk}}{S_k C_k}}\} \le \frac{S_k C_k(\sqrt{\frac{L_k(\xi_k-\theta_k)}{\lambda}}-1)}{\mu_{uk}} \le \frac{S_k}{L_k-\frac{\mu_{uk}}{C_k}}$，
f_k 为 $\frac{S_k C_k(\sqrt{\frac{L_k(\xi_k-\theta_k)}{\lambda}}-1)}{\mu_{uk}}$、$\frac{S_k}{L_k-\frac{\mu_{uk}}{C_k}}$、$\min\{0,\frac{1}{\frac{L_k\tau_k}{\tau_k S_k - L_k}-\frac{\mu_{uk}}{S_k C_k}}\}$三个值中使
得 $h(f_k)=(\theta_k-\xi_k)\frac{L_k}{\frac{\mu_{uk}}{S_k C_k}+\frac{1}{f_k}}+\lambda f_k$ 最小的值。
else f_k 为、$\frac{S_k}{L_k-\frac{\mu_{uk}}{C_k}}$、$\min\{0,\frac{1}{\frac{L_k\tau_k}{\tau_k S_k - L_k}-\frac{\mu_{uk}}{S_k C_k}}\}$两个值中使得
$h(f_k)=(\theta_k-\xi_k)\frac{L_k}{\frac{\mu_{uk}}{S_k C_k}+\frac{1}{f_k}}+\lambda f_k$ 最小的值。
二、云端利用梯度下降法求 λ：
1、求 $j(\lambda)$ 的梯度值 $\nabla j(\lambda)$。
2、沿着梯度的方向移动 λ：$\lambda \leftarrow \lambda+\gamma\nabla j(\lambda)$，$\gamma$ 为步长。
3、循环迭代步骤 2，直到 λ 值的变化使得 $j(\lambda)$ 在两次迭代之间的差值小于设定的精度值，说明此时 $j(\lambda)$ 已经达到了该精度下的最大值。
4、输出 λ，这个 λ 就是使得 $j(\lambda)$ 最大时 λ 的取值 。

3.5　数值结果

该小节展示我们分布式资源分配策略的数值结果。表 3.2 是采用的系统参数取值。图 3.2 展示了利用梯度下降法求解 λ 的过程。图 3.3 和图 3.4 分别是当系统中有两个用户时（固定用户 2 的延时界 L_2 和计算任务量 S_2），对偶变量的取值以及用户 1 分配到的计算资源量 f_1 和用户 1 的延时界 L_1 的关系。从图 3.4 可以看出，用户 1 分配配到的计算资源 f_1 随着用户 1 延时界 L_1 的增大而增大，即在用户 2 延时界 L_2 和计算任务量 S_2 不变的时候，用户 1 的延时界越松，越有机会上传更多的任务量到云端处理从而节省终端能耗，从而会被分配到更多的计算资源。图 3.5 和图 3.6 分别是当系统中有两个用户时（固定用户 2 的延时界 L_2 和计算任务量 S_2），对偶变量的取值以及用户 1 分配到的计算资源量 f_1 和用户 1 的任务量 S_1 的关系。根据图 3.6，用户 1 分配到的计算资源量随着用户 1 任务量的增加逐渐增加，即终端任务量越多越应该向云端请求更多的计算资源，从而在延时约束内完成任务。

表 3.2 系统参数设置

参数	取值
终端发射功率$(P_k, \forall k)$	0.2W
上行传输速率(C_k)	100kb/s
云端总计算资源(F)	$1 \times 10^8 cycles/second$
$\xi_k, \forall k$	$8.6 \times 10^{-8} J/bit$
$\tau_k, \forall k$	$10^{-7} s/bit$
$\mu_{uk}, \forall k$	1

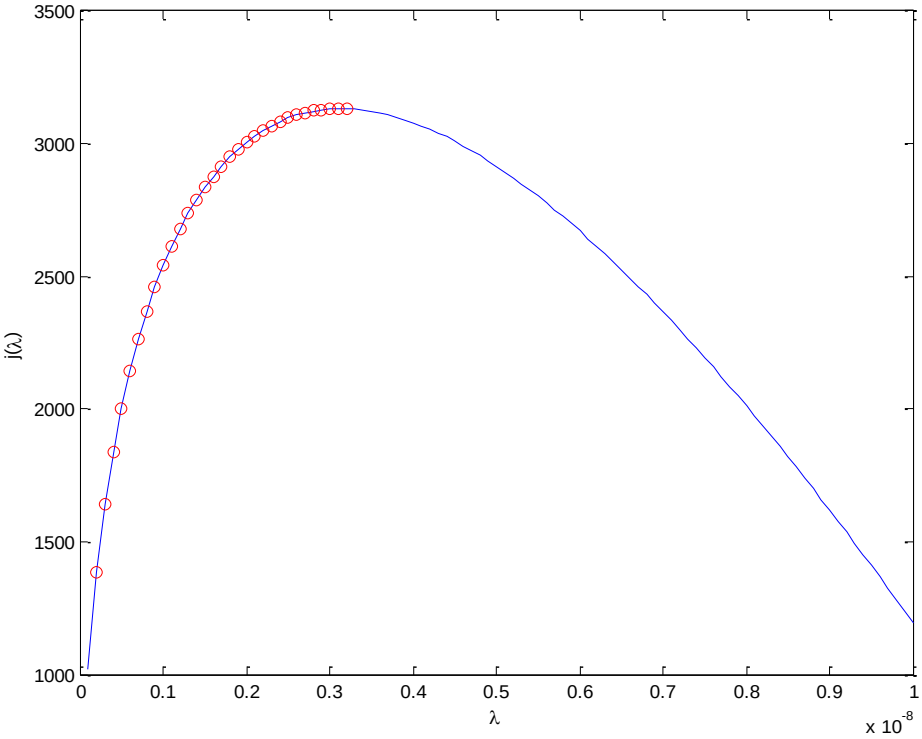

图 3. 2　梯度下降法寻找使得 $j(\lambda)$ 最大的 λ

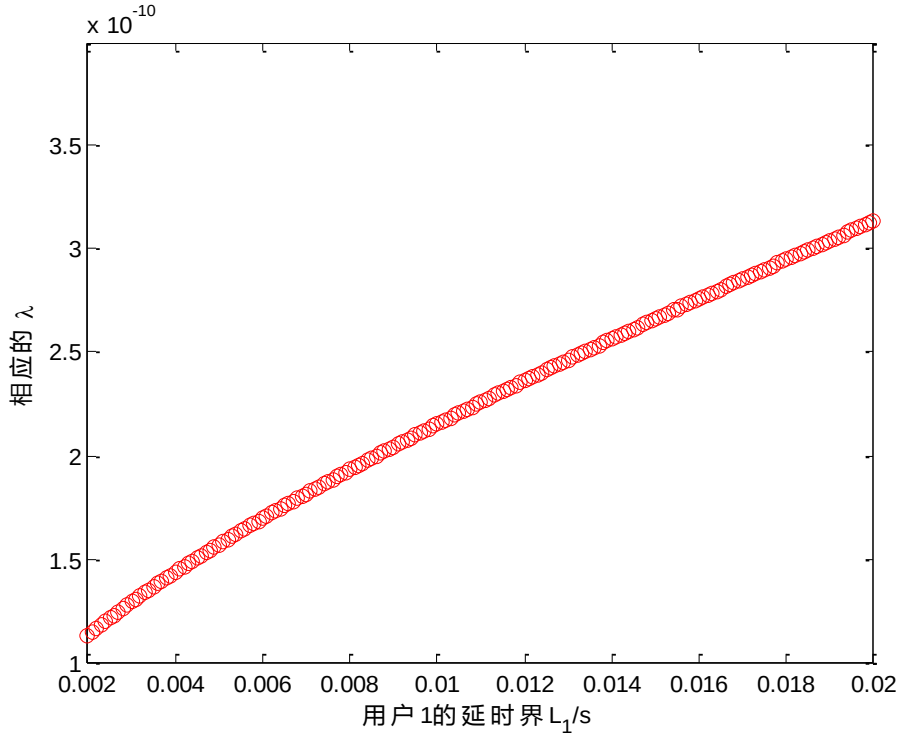

图 3. 3　最终 λ 和用户 1 延时界 L_1 的关系

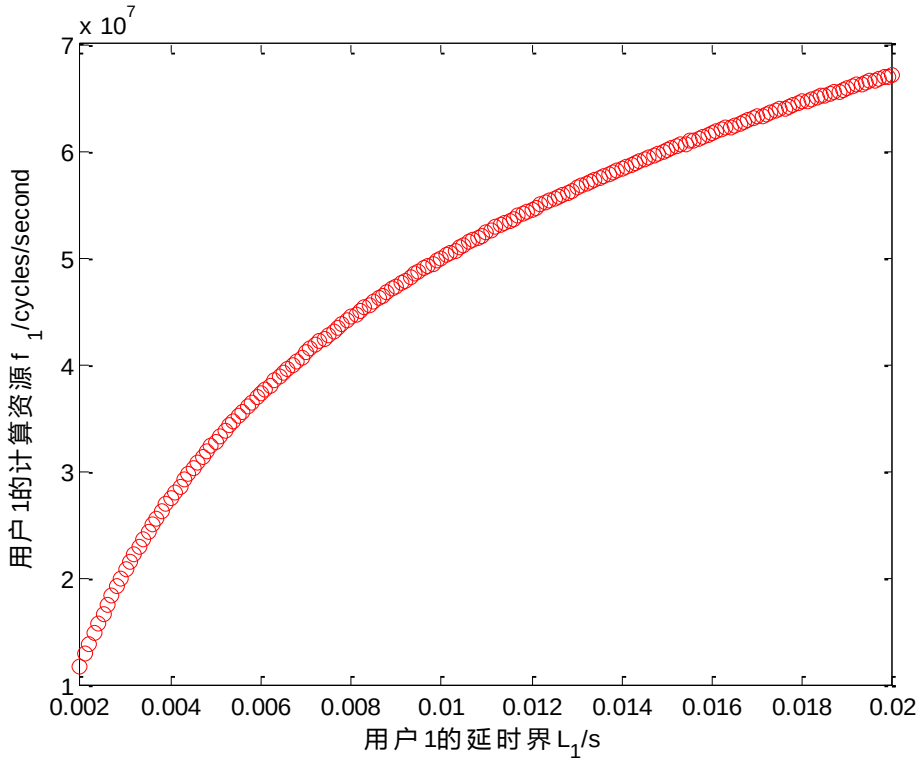

图 3. 4　用户 1 分配到的计算资源 f_1 和用户 1 延时界 L_1 的关系

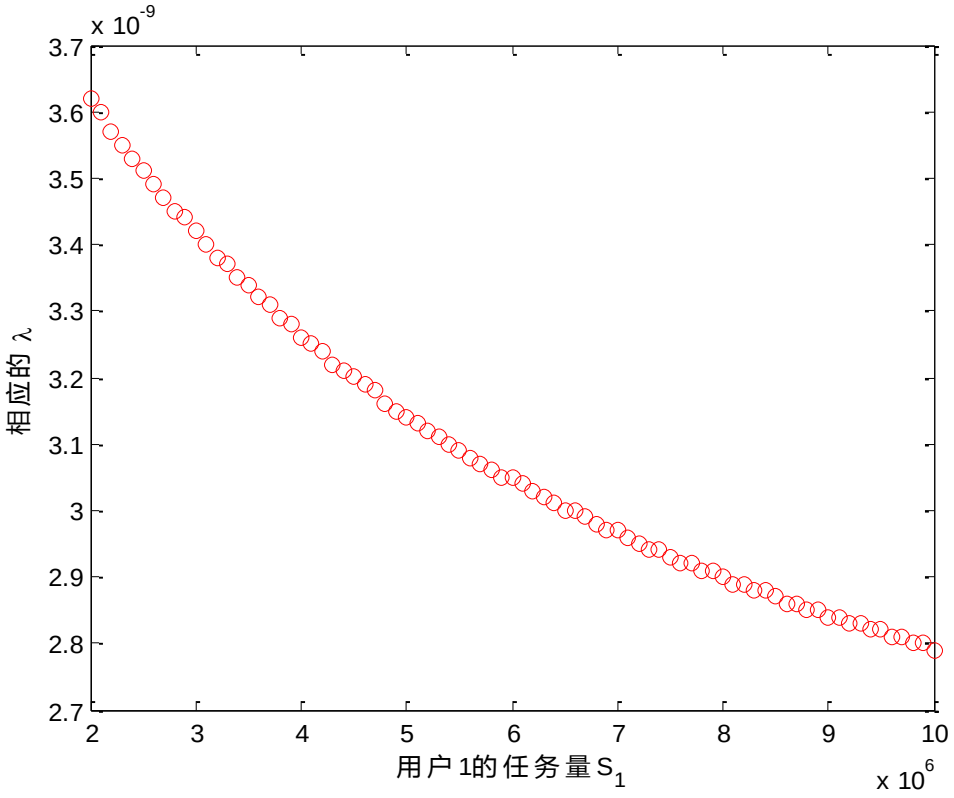

图 3. 5 最终 λ 和用户 1 任务量 S_1 的关系

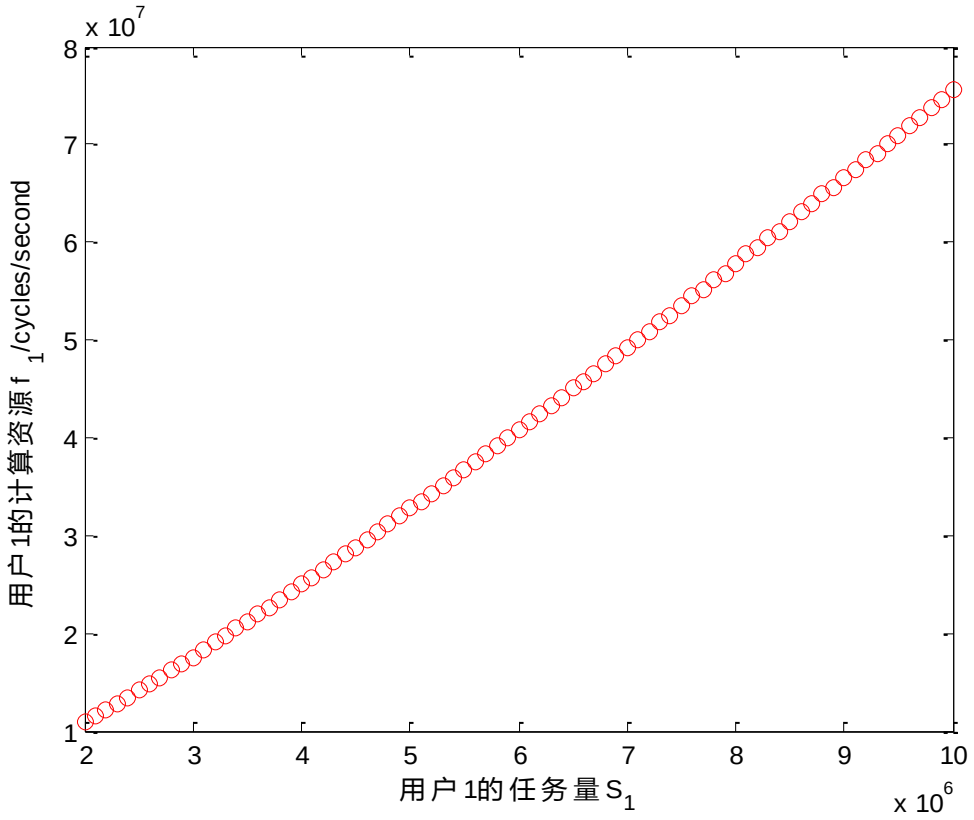

图 3. 6 用户 1 分配到的计算资源和用户 1 任务量 S_1 的关系

3. 6　本章小结

本章中继续研究移动云计算多用户场景，在资源受限的约束里，K 个用户通过将集中式的资源分配优化问题解耦分解为 K 个分布式优化子问题及一个主对偶问题（master dual problem）负责更新对偶变量 λ 。我们通过分解问题设计了分布式的资源分配策略，用户根据对偶变量决定自己需要上传的计算资源量和所需的计算资源，而云端则负责更新对偶变量，这样用户和云端之间只要传输对偶变量及用户所需的计算资源量即可，故该策略具有计算复杂度低以及传输开销小的优势。

第 4 章　用经济学的方法分布式分配有限资源

4.1　本章引论

在实际的场景中，云计算的有限资源需要用户付费使用，本章在移动云计算场景中引入计算资源的价格，利用博弈论为多用户系统提供分布式资源分配策略。博弈论在研究多参与者决策尤其是用户竞争行为中得到广泛应用。[39]博弈论解决各参与者行为会直接相互影响的问题，并假设各个参与者都是自私的、理性的，即任何时候都会做最利于自己的决策，这很符合现实移动云计算场景中用户对资源的竞争。

移动云计算场景中的资源分配问题可以采用分布式的方法求解，分布式资源分配基于用户自身对上传任务量和请求计算资源多少的控制，相对集中式控制减少网络对中心分配设备的依赖，减轻基站和用户间的信令开销。用户基于云端的价格决定任务上传量和请求的计算资源量，而云服务提供商则根据用户的这些信息决定最有利于云端的计算资源价格策略。

本章的主要贡献如下：

综合考虑用户满足延时以及节省终端能耗的需求以及对云端有限计算资源的购买，将多用户移动云场景中的经济模型问题建模成为一个完全信息动态博弈问题，利用逆向递归法求解该动态博弈问题的纳什均衡解，研究如何通过调整云端计算资源的价格来调整给各用户的资源分配量。

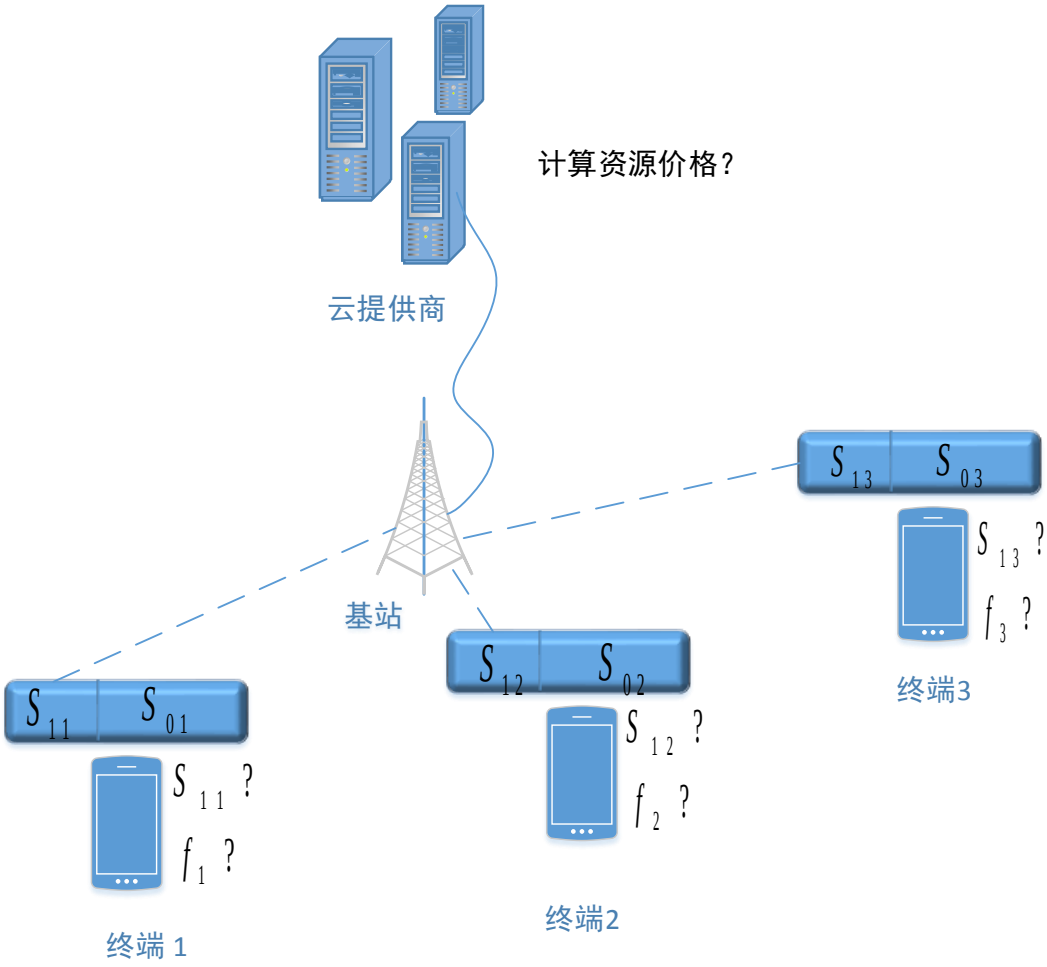

图 4. 1　分布式移动云计算场景（云提供商定计算资源的价格，用户自己决定上传任务量和计算资源购买量）

4. 2　系统模型

如图 4.1 所示，我们研究的是一个基站连接有一定计算能力的云提供商的场景，其中有 K 个用户，每个用户 $k(\forall k \in \{1,2,...,K\})$ 都有一个大小为 S_k 的计算任务需要在规定延时界 L_k 内完成。由于是分布式机制，用户可以自行决定上传到云端的任务量大小和需要购买的计算资源量。云提供商则通过调整计算资源的价格最大化有限计算能力的云端收益。

不失一般性，我们考虑一个基站连接到有一定计算资源本地云的分布式云场景，系统中有K 个用户，每一个用户都有一个应用需要在规定的延时要求内完成。用户可以选择是否将计算任务转移到云端计算。在需要负载部分转移时，应用支持任务分割，且假设用户的任务都可以被分割为任意大小的两部分。一部分任务在终端处理，另一部分转移到本地微云中去

完成。这里，为了问题的简化没有考虑任务分割带来的延时及能耗等开销。从一定程度上讲，这样的分割模型对最终系统性能（延时和终端能耗）是实际情况的下界或者最好情况。根据上述分析任务分割模型可以描述如下[32]:

$$S_{0k}+S_{1k}=S_k. \tag{4-1}$$

S_{0k} 表示在用户端完成的部分，剩余的 S_{1k} 表示通过无线网负载转移至微云端去处理的部分，S_k 是第 k 个用户端应用的总任务量大小。

在第 k 个用户端完成任务 S_{0k} 所需的时间与在终端执行的任务量 S_{0k} 成正比:$\tau_k S_{0k}$。这里假设第 k 个用户去完成一个 1 *bit* 任务量所需的时间是 τ_k，τ_k 的值取决于终端 CPU 的运行速度以及应用的种类。

假定用于去处理非负载转移部分任务量所需的终端能量为 $\xi_k S_{0k}$，其中 ξ_k 代表着本地移动终端处理 1 *bit* 任务的能量消耗。终端 CPU 的运算速度的不同和用户处理应用的复杂程度的不同，这里 ξ_k 的大小也相应有所变化。

当需要有计算任务转移到云端计算时，第 k 个用户上传的任务部分 S_{1k} 执行需要的总延时 Δ_k 包括如下 4 个部分：

$$\Delta_k=\Delta_k^t+\Delta_k^e+\Delta_k^r+T_B. \tag{4-2}$$

其一，Δ_k^t 是上传到云端去执行的任务部分 S_{1k} 所需的状态和输入的上行传输时间。假设上行传输的总比特数为 S_{uk}，表示用户在上行传输时必须上传的程序包大小。假设给每个用户 k 分配固定的上传容量 C_k，则上行传输时间 Δ_k^t 可以表示为：

$$\Delta_k^t=\frac{S_{uk}}{C_k}. \tag{4-3}$$

其二，Δ_k^r 是将在云端计算得到的结果通过无线网络下行传输到第 k 个移动终端所需要的时间。通常来说由于基站的发射功率远大于用户终端的发射功率，所以下行传输时间相对上行传输时间要小得多，在(4-2)总延时计算中可以忽略。

其三，Δ_k^e 代表在云端处理第 k 个移动终端的 S_{1k} 任务量所需的时间。假定第 k 个移动终端获取的计算资源（CPU 频率）是 f_k，那么

$$\Delta_k^e=S_{1k}/f_k. \tag{4-4}$$

其四，后传网（backhaul）的传输时间 T_B 应当是一个常量，随基站和

微云之间的距离和连接方式（例如光纤）不同而变化。

用户 k 上传 S_{1k} 任务量并购买 f_k 的计算资源需要付的费用（也就是云端为用户 k 用 f_k 的计算资源处理 S_{1k} 任务量的收益）是 $M_k=cS_{1k}f_k^{\alpha}$,这里 $c>0$ 是常数，$\alpha \geq 1$ 体现出计算资源的价格特点，云端对用户的计算资源收费呈指数增长[40]，云提供商可以调整 α，将有限的计算资源分配给 K 个移动终端，从而最大化自己的收益。

4.3　问题建模

用户和云服务提供商的相互关系为:云提供商对计算资源价格说一不二，但用户可以自主决定任务上传量以及购买的计算资源量。这样形成一个完全信息动态博弈问题，可以用逆向归纳法去解该问题的纳什均衡点。当达到纳什均衡点时，云提供商和用户都不会单方面改变自己的策略来提高自己的效用函数（Utility Function）。[41]即

$$U_{cloud}(\alpha^*, f_k^*) \geq U_{cloud}(\alpha, f_k^*), \tag{4-5}$$

$$U_{MTk}(\alpha^*, f_k^*) \geq U_{MTk}(\alpha^*, f_k), \ \forall k=\{1,2,\ldots,K\}, \tag{4-6}$$

其中 $U_{cloud}(\alpha, f_k)$ 和 $U_{MTk}(\alpha, f_k)$ 分别代表云提供商和第 k 个用户的效用函数，α^* 是云端针对均衡时 K 个用户的计算资源请求量得到的最优 α，即是如下问题的解：

$$\max_{\alpha} U_{cloud}(\alpha, f_k^*) \tag{4-7}$$

f_k^* 是用户 k 针对均衡时云端出示的资源价格决定的最优计算资源请求量 f_k，即是如下问题的解：

$$\max_{f_k} U_{cloud}(\alpha^*, f_k) \tag{4-8}$$

在用户端，每个用户通过决定最优的上传量和需要的计算资源量在满足延时需求的同时最小化自己的费用函数 $\gamma_k E_k + \beta_k M_k$，其中 $E_k = P_k \Delta_k^t + \xi_{0k} S_{0k}$ 是用户 k 完成任务所产生的终端能耗，P_k 是用户 k 的上行发射功率。γ_k 和 β_k 是权重因子，分别代表用户 k 对于终端能耗和费用的敏感程度。γ_k 越大，代表用户 k 对终端能耗越关心；β_k 越大，代表用户 k 对费用金额越在乎。则用户 k 一侧的优化问题是在满足延时需求的同时，最小化终端能耗和所需的资源购买费用：

$$\begin{aligned} \min_{S_{1k}, f_k} \quad & \gamma_k E_k + \beta_k M_k \\ s.t. \quad & S_{ok} + S_{1k} = S_k, \forall k \\ & max\{\tau_k S_{0k}, \Delta_k\} \le L_k, \forall k \end{aligned} \tag{4-9}$$

其中延时约束通过化简实则可以表示为对于任务上传量S_{1k}的限制：

$$max\{0, \frac{\tau_k S_k - L_k}{\tau_k}\} \le S_{1k} \le \min\{f_k(L_k - \Delta_k^t), S_k\}. \tag{4-10}$$

在实际情况中，移动云计算服务的方式是云端收取适当费用的情况下为移动终端节省能耗，用户愿意将更多的任务量转移到云端计算，即满足：

$$\gamma_k \xi_k > c\beta_k f_k^{\alpha}, \tag{4-11}$$

那么用户 k 最优的任务上传量应该是:

$$S_{1k} = min\{f_k(L_k - \Delta_k^t), S_k\}. \tag{4-12}$$

将这个计算任务量(4-12)代入上面用户的优化问题(4-9)的优化目标中，优化问题可以写为：

$$\min_{f_k} \quad g(f_k) = \beta_k c f_k^{\alpha+1} - \gamma_k \xi_k f_k. \tag{4-13}$$

利用求导的方法，我们令

$$g'(f_k) = \beta_k c(\alpha+1) f_k^{\alpha} - \gamma_k \xi_k = 0, \tag{4-14}$$

通过求解上述方程，我们得到最优的用户计算资源请求量

$$f_k = (\frac{\gamma_k \xi_k}{(\alpha+1)c\beta_k})^{\frac{1}{\alpha}}. \tag{4-15}$$

在云端，云提供商通过调整计算资源的价格最大化自己的收益 M，同时要分配给各用户相应的资源量 f_k 从而满足用户的服务需求：

$$\begin{aligned} \max_{\alpha} \quad & M = \sum_{k=1}^{K} M_k \\ s.t. \quad & \sum_{k=1}^{K} f_k \le F, \\ & S_{1k} \le S_k, \forall k \end{aligned} \tag{4-16}$$

其中，$M = \sum_{k=1}^{K} M_k$ 是云端从 K 个用户得到的总收益，F 是云端总的计算

资源量。

利用每个用户上报的计算任务量S_{1k}和请求的计算资源大小f_k，云端的优化问题变为：

$$\begin{aligned} \max_{\alpha} \quad & M=\sum_{k=1}^{K}(\frac{\lambda_k}{\alpha+1})^{\frac{\alpha+1}{\alpha}} \\ s.t. \quad & \sum_{k=1}^{K}(\frac{\lambda_k}{\alpha+1})^{\frac{1}{\alpha}} \le F, \\ & (\frac{\lambda_k}{\alpha+1})^{\frac{1}{\alpha}}(L_k-\Delta_k^t) \le S_k, \forall k, \end{aligned} \tag{4-17}$$

这里，$\lambda_k=\frac{\gamma_k \xi_k}{c\beta_k}$表示第 k 个用户对于终端能耗和花费的敏感程度，λ_k越大，代表该终端越愿意为了节省终端能耗而付费。

当系统中只有一个用户且云端资源不受限时，(4-17)云端优化问题的目标函数简化为：

$$\max_{\alpha}(\frac{\lambda_k}{\alpha+1})^{\frac{\alpha+1}{\alpha}}=\max_{\alpha} g(\alpha) \tag{4-18}$$

利用求导的方法得出目标函数$g(\alpha)$关于价格因子α的单调性：当$\lambda_k \ge 1$时，云端收益函数随着价格因子α的增加而减小。

单用户云端资源受限时，可以采用惩罚函数法求得带约束优化问题(4-17)的近似解，当$\lambda_k \ge 1$时，云端收益函数随着价格因子α的增加而减小，所以只需要找到满足延时约束和最大传输任务量约束的最小的α即可。

多用户且云端资源受限时，当$\lambda_k \ge 1, \forall k$时，云端收益函数随着价格因子$\alpha$的增加而减小，故只需要找到满足延时约束和最大传输任务量约束的最小的α即可。

4.4　数值结果

该小节是一些仿真数值结果。图 4.2、图 4.3、图 4.4 和图 4.5 分别显示了系统只有一个用户或者 K 个用户参数（λ_k、S_k、L_k）都一致的情况下，云端的均衡价格参量α、均衡的计算资源请求量f_k、均衡的计算任务上传量S_{1k}以及均衡的云端收益随着λ_k的变化情况。图中当λ_k刚开始增大时，随着λ_k的增加，用户愿意购买越多云端的计算资源，进而上传越多的

计算任务到云端处理；而出现拐点是由于云端资源受限（云端总计算资源 $F = 5 \times 10^7 cycles/second$），此时云端只能通过提升计算资源价格（即增大价格参量 α）来限制用户对于计算资源的请求购买量。

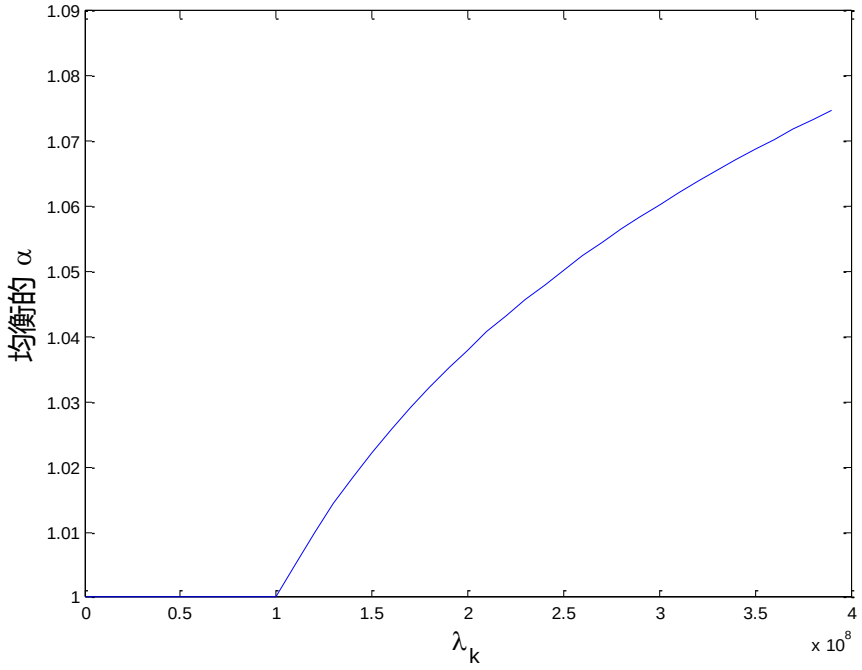

图 4. 2　均衡的 α

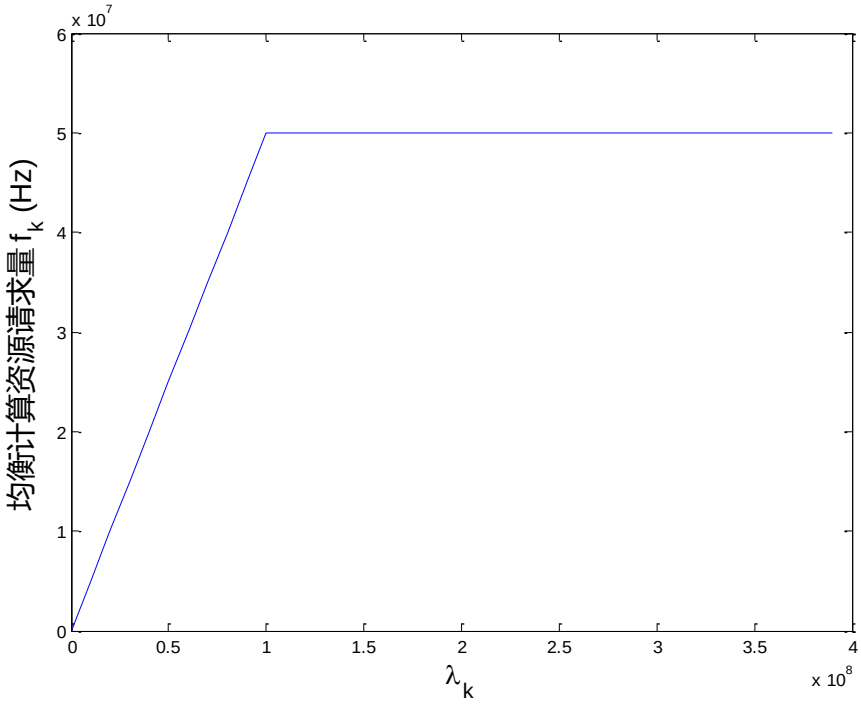

图 4. 3　均衡计算资源请求量 f_k

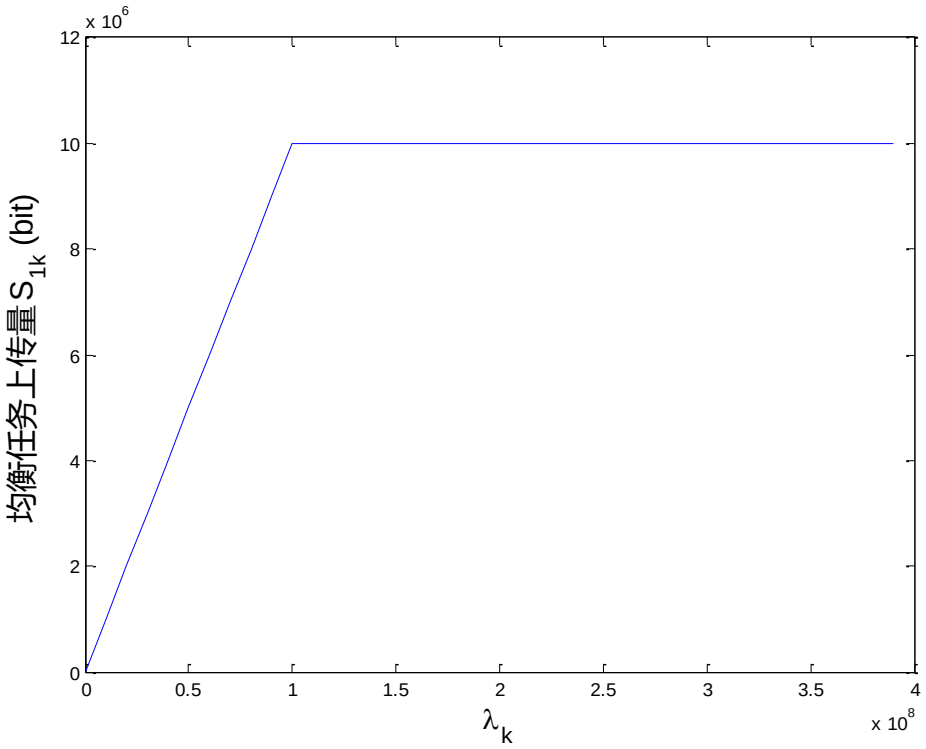

图 4.4 均衡任务上传量 S_{1k}

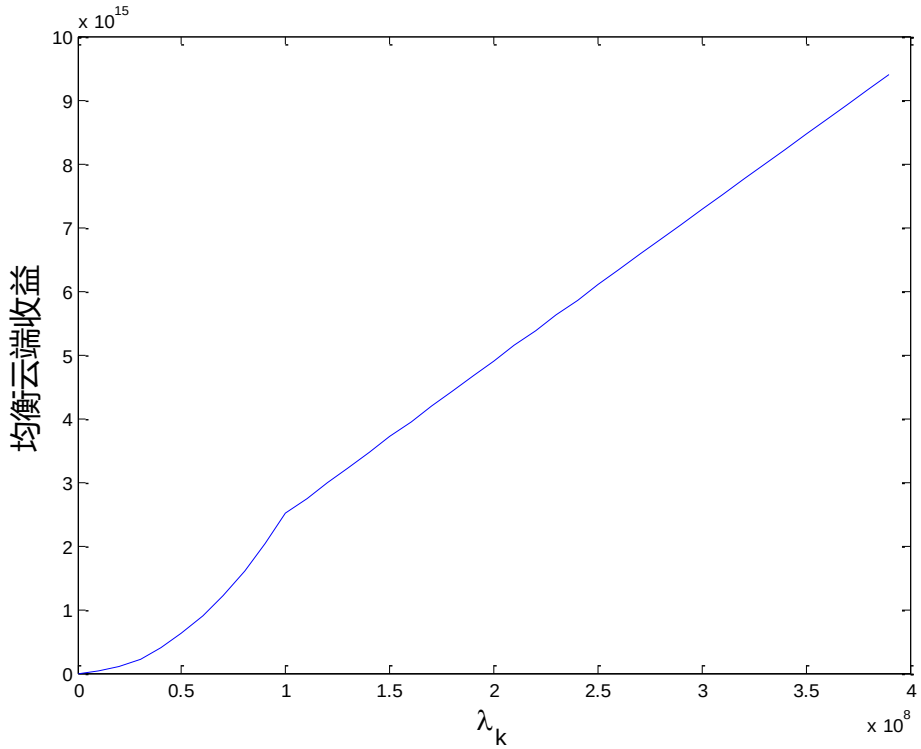

图 4.5 均衡云端收益

4.5　本章小结

本章建模了多用户和资源受限的边缘云提供商的博弈问题，云端通过调整资源价格来为用户分配有限的计算资源。通过逆向归纳法求得该博弈问题的纳什均衡点，即在该点用户和云都不可能通过单独改变自己策略来增大自己的收益。数值结果表明如果用户更愿意为了节省终端能耗付费时，云提供商可以在均衡时获益更高。

由于单纯靠博弈参与者的理性竞争得到的纳什均衡很多时候不会是全局最优，所以通过设计一些机制，使得用户和云端都不完全是理性，利用合作博弈达到更好的资源利用可以作为未来工作。

第 5 章 移动云计算中的移动性管理

5.1　本章引论

移动性管理一直是移动通信网络中非常关键的问题，对于移动性管理的研究现状，同构网中的用户移动性管理研究比较成熟，如有文献[42]利用排队论分析预留信道对切换成功概率和系统容量的影响；而对于异构网的切换策略，根据切换标准或者优化目标不同，可以分为基于接收信号强度、接收干扰功率、用户速度、业务类型、能效以及综合考虑其中若干因素的代价函数的切换策略。[43]

而对于移动云计算中的切换，一个用户在运动中逐渐远离基站 1，而靠近基站 2. 异于传统切换会选择接到提供更好 SINR（signal-to-interference-plus-noise ratio）的基站，这里还需要考虑所谓的云切换就是云端计算带来的影响，即需要综合考虑无线资源和云资源的分配，或者考虑云切换，尤其当用户在移动过程中从基站 1 切换到基站 2 但是在基站 1 对应的服务器端有该用户的计算任务正在进行，这里会涉及到虚拟机迁移以及后传网的传输开销。

本章考虑用户移动性对移动云计算系统的影响，图 5.1 是最基本的移动场景,云架构是每个基站都对应一个本地服务器，一个用户在运动中逐渐远离基站 1，而靠近基站 2。传统切换会选择接到提供更好 SINR 的基站接入，但这里还需要考虑所谓的云切换就是云端计算带来的影响。

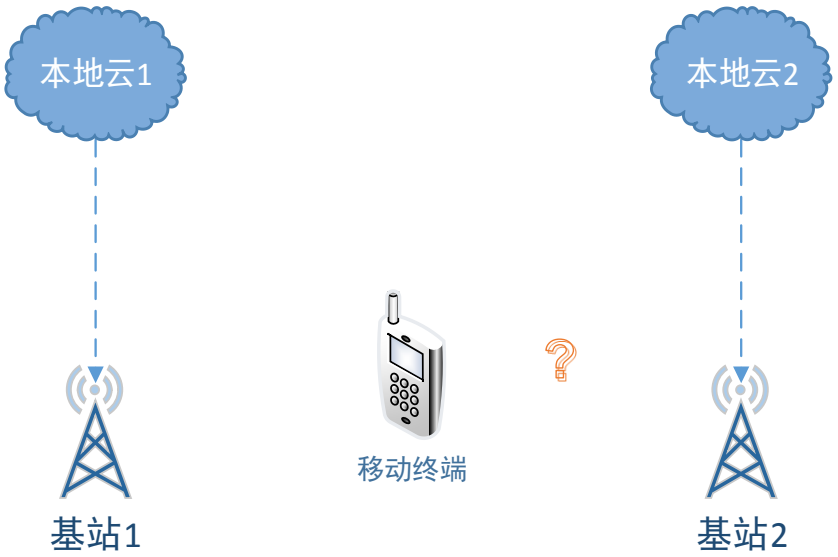

图 5.1 移动云计算中用户移动场景

下一代（5G）通信系统目标延时是 1ms，我们希望最小化在移动云计算中用户端的延时，同时由于每次切换都会引入延时以及其他信令开销，所以应该尽可能减少不必要的切换。

5.2　基于延时的移动云场景切换策略

在图 5.2 移动云计算中用户延时最小切换策略所示的流程图设计了一个以延时为标准的移动云计算场景切换策略。只有当用户接到基站 2 的延时小于原来连接到基站 1 的延时减去切换迟滞量（HHM）时，用户才切换到基站 2，HHM 的引入是为了消除无线信道的变化并且减缓乒乓效应。

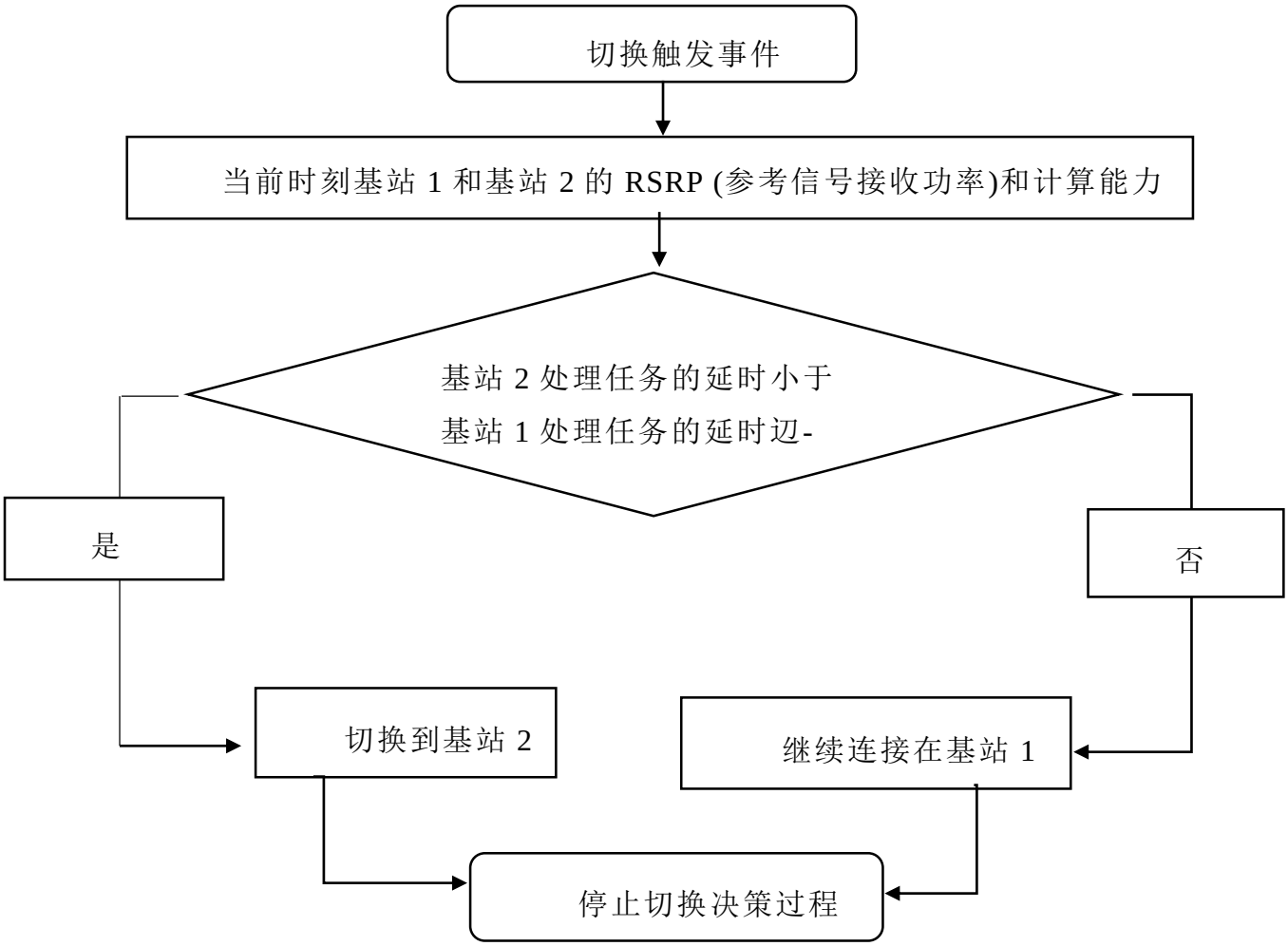

图 5. 2 移动云计算中用户延时最小切换策略

移动终端处理计算任务经历的总延时 T 由如下四部分构成：

$$T = T_{tx} + \frac{\omega}{f} + T_{rx} + T_{bh}, \tag{5-1}$$

其中，T_{tx} 是将用户把计算任务所需的信息传到基站的上行传输延时，利用香农公式求得上传速率可得：

$$T_{tx} = \frac{N}{B log_2(1+SINR)}. \tag{5-2}$$

这里N是需要上传的信息量，$\frac{\omega}{f}$是在 CPU 频率为 f 的服务器完成任务量大小为ω计算任务所需要的计算时间。T_{rx} 是基站把计算任务所得结果返

回给用户的下行传输延时。T_{fh} 是基站把任务通过后传网传到云端的耗时。

下面这个式子体现了刚才流程图中的切换决策的思想，只有当用户接到基站 2 的延时小于原来基站的延时减去切换迟滞量时，用户才切换到基站 2：

$$\arg min_{c\to u} T := \{c \mid T_{c\to u} < T_{s\to u} - HHM_{dB}\} \tag{5-3}$$

其中u 代表目标用户，s是当前正在服务用户的基站，c是目标可能切换到的服务基站。

5.3　数值结果

这里是一些仿真结果，关于系统的参数设置，我们借鉴了文献[44]：

基站分布假设服从泊松点过程（PPP）,基站的发射功率为 43dBm。

用户移动性模型：用户在 t 时刻的速度大小服从均值为$\bar{v}$，标准偏差为 s_u^2 的正态分布：

$$v_t = N\left(\bar{v}, s_u^2\right). \tag{5-4}$$

其中，$\bar{v}$是用户速度的均值，典型值有 $\bar{v} = 3m/s$ 和 $\bar{v} = 40m/s$，分别表示低速移动和高速移动的用户，$s_u = 1km/h$ 是速度标准偏差。

用户在 t 时刻的速度方向角度依赖于上一时刻用户的运动方向：

$$\varphi_t = N(\varphi_{t-1}, 2\pi - \varphi_{t-1}\ tan\left(\sqrt{\frac{v_t}{2}}\right)\Delta t). \tag{5-5}$$

仿真中单位时间间隔为 $\Delta t = 1s.$

路径损耗模型：

$$PL(dB) = 15.3 + 37.6log_{10}d. \tag{5-6}$$

总仿真时间为 $1000s$。

平均切换数定义为在 $1000s$ 的仿真时间中的切换数。如图 5.3 所示，平均切换数随着 HHM 的增加基本减小（中间不太规律是因为 PPP 撒点的时候随机性以及仿真时长较短造成的），而且用户的移动速度越快，平均切换数目会越大。因此 HHM 可以通过避免切换中的乒乓效应减少一些不必要

的切换。平均延时是在 1000s 的仿真时间中用来传输和处理任务的平均延时，如图 5.4 所示，大的 HHM 意味着为了减少不必要的切换，能够容忍当前基站的更大延时，因此会导致更大的平均延时。图 5.5 和图 5.6 分别对比了我们提出的基于延时的移动云场景切换策略和传统的将用户接入信号最强基站的策略的平均延时和平均切换数。观察两图可以发现，采用基于用户延时的切换策略之后，平均延时比最强信号基站接入策略高一倍左右，而切换次数并没有太多增加。

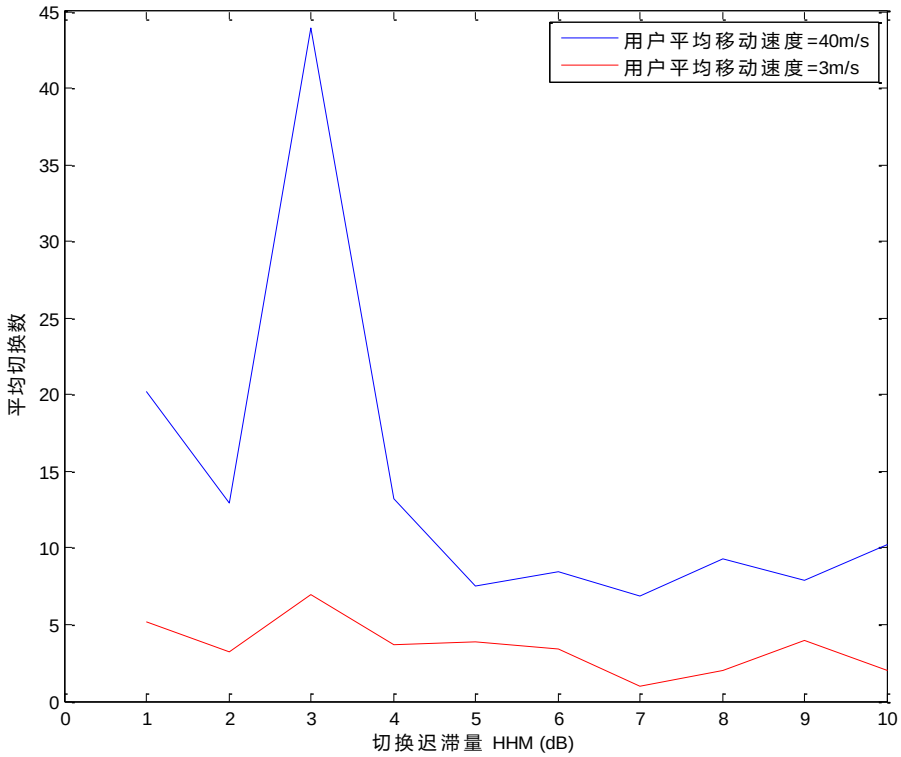

图 5.3 平均切换数

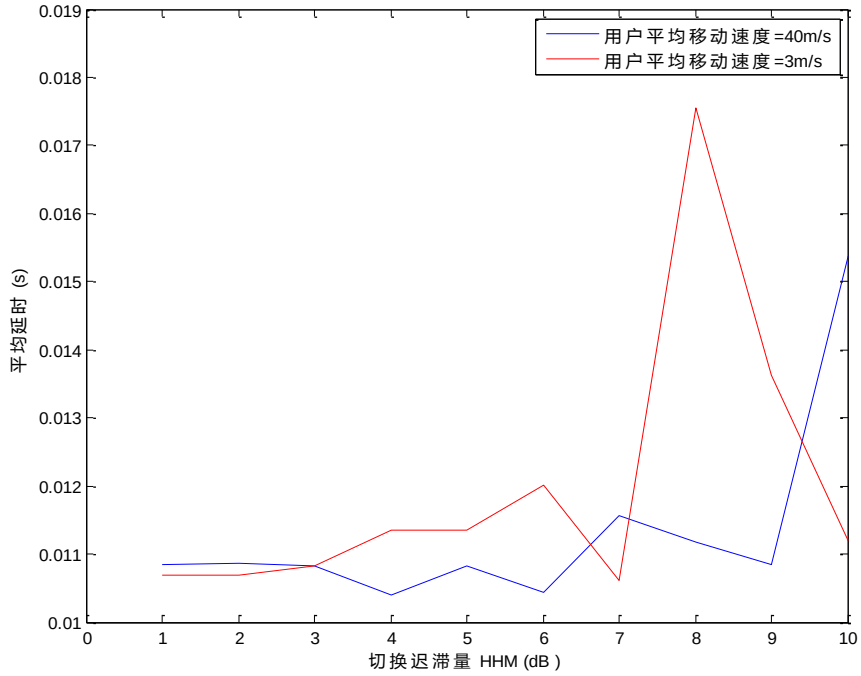

图 5. 4　平均延时

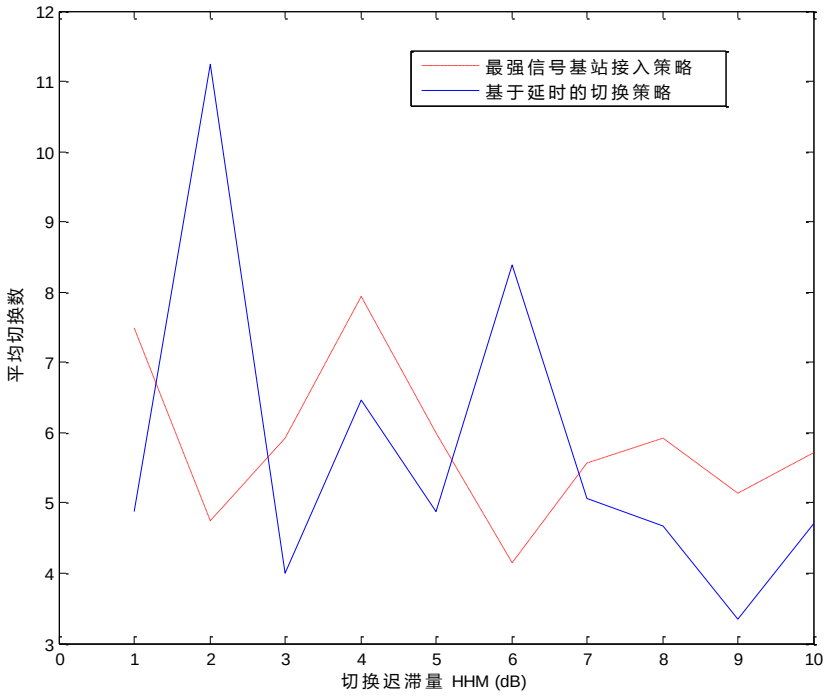

图 5. 5　两种策略的平均切换数

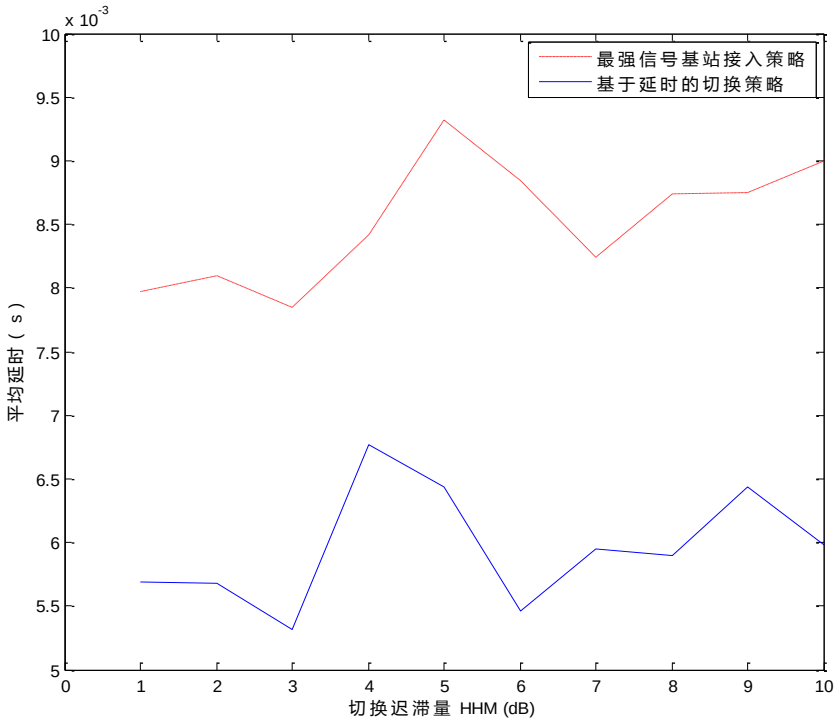

图 5. 6　两种策略的平均延时

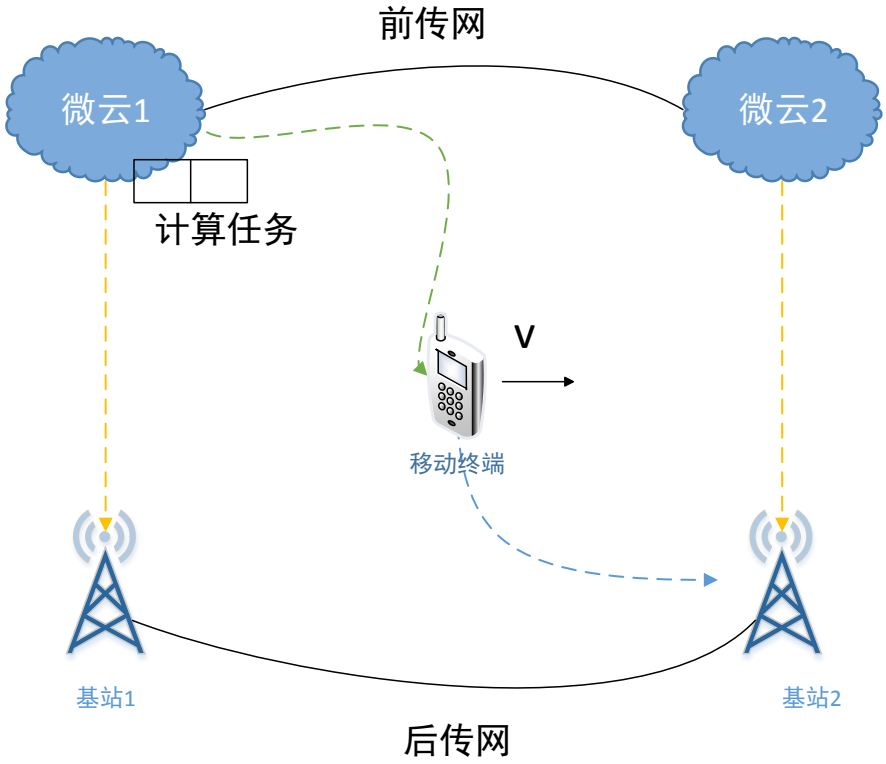

图 5. 7　移动云计算中云切换场景

5.4　移动云切换的排队模型

移动云切换中一个特殊场景是有任务在云端计算时而无线端由于移动用户移出原来基站的覆盖范围而必须切换，如图 5.7 移动云计算中云切换场景所示。综合考虑无线资源和云资源的分配，需要考虑云切换。如果用户在移动过程中从基站 1 切换到基站 2 但是在基站 1 对应的服务器端有计算任务正在进行，这时候有两个选择：一、算完再传，就要考虑占用微云 1 的计算资源，这样会导致其他用户无法接入基站 1，增加切换失败概率，还有要考虑把在微云 1 处算好的结果传到基站 2 带来的延时；二、虚拟机迁移，会带来很大的延时和资源开销。[45]

对于同时有无线切换和云端切换的场景可以建模为 Jackson 网络,如图图 5.8 移动云计算切换排队网络模型所示，但由于 Jackson 单独分析各节点的延时分布比较困难[46]，所以简化为下面图 5.9 移动云计算中的级联多跳排队系统，第一个队列是有两类业务的 M/G/1 队列，其中λl代表本地业务到达率，λh是切换业务到达率，λch表示云切换任务到达率。利用排队论进一步分析业务排队延时以及不同任务在系统中的阻塞率。

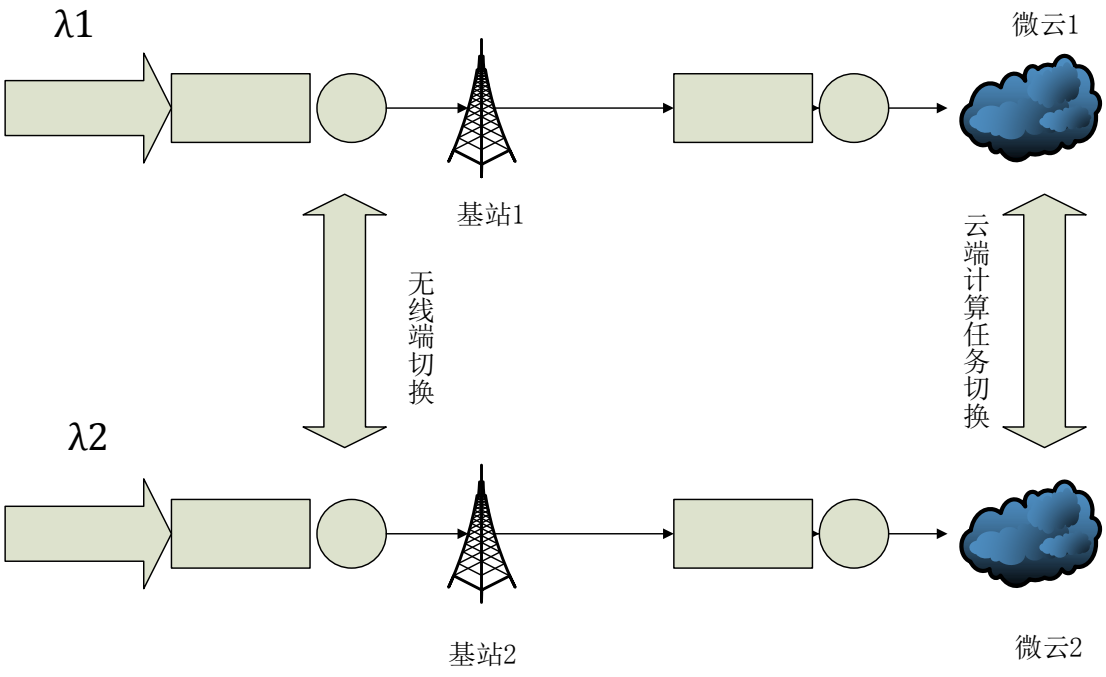

图 5.8 移动云计算切换排队网络模型

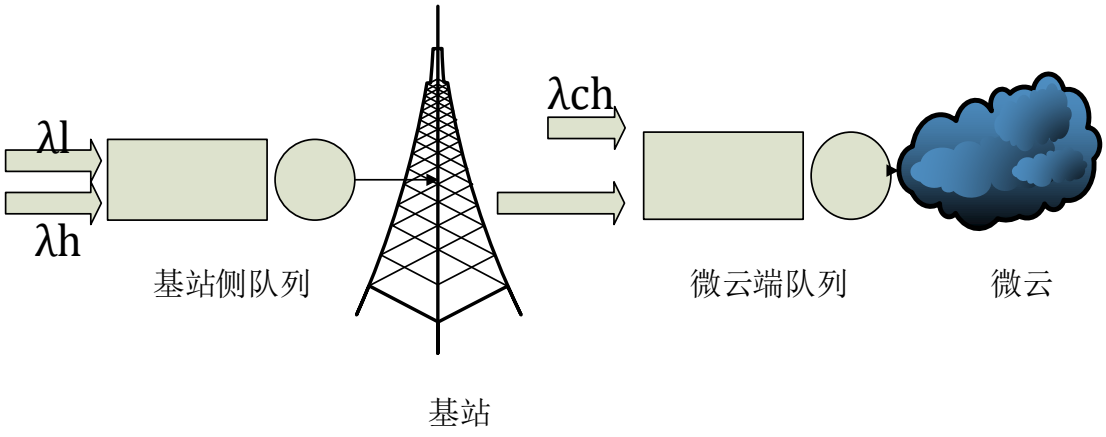

图 5.9 移动云计算级联多跳模型

5.5 本章小结

在这一章中，我们分析了移动云计算中的移动性管理和传统蜂窝网中的移动性管理问题的本质不同，即存在云切换的概念。综合考虑无线端和云端的延时，提出了基于延时的移动云计算切换策略，通过引入切换迟滞量 HHM 减少移动用户的切换开销（如平均切换数目），并保证用户端的低延时。最后分析了移动云计算切换场景中的排队模型建模。

第 6 章　总结与展望

6.1　研究总结

移动云计算随着近年移动数据业务量的剧增以及终端能耗问题的凸显越来越受到人们的关注，然而依然面临着延时、终端能耗以及移动性管理的核心挑战。本论文的关注点是研究如何在多用户移动云计算场景中满足各用户延时需求的同时，高能效（特指终端能耗）地分配系统中有限的无线资源和计算资源。同时针对移动云切换的概念讨论了移动云计算中的低开销切换策略。此外，移动云计算的实现应用方法可以拓展到物联网，车联网以及赛博物理系统等有多传感器的场景中。

本文主要结论总结如下:

1、在多用户集中式联合分配有限无线资源和计算资源中，为延时界 L_k 更接近全部任务在终端计算时间$(\tau_k S_k)$的用户分配更多的无线资源和计算资源。当延时界较紧时，由于将任务上传到云端会带来更多延时，所以应该传少量的任务到云端去计算从而节省能量，此时也即需要较少的无线资源和计算资源；当延时界较松时，用户只需要相对较少的资源就可以在自己的延时要求内完成任务计算，此时可以将更多资源留给系统中的其他用户。

2、在多用户移动云计算分布式资源分配中，基站通过调整拉格朗日乘子来调整用户的计算资源请求量。通过对偶的方法将原本 K 个用户互相耦合的优化问题解耦，从而由用户自行决定上传任务量及请求的计算资源量，云端则负责控制对偶变量来调整用户对计算资源的请求量。

3、在经济学模型中，云端可以调整云端资源的价格从而使用户购买合适的计算资源来达到用户和云端之间的纳什均衡，在该纳什均衡点，云提供商和用户都不能通过单独改变自己的策略来增加自己的收益。

4、移动云计算中的用户切换问题异于传统蜂窝网中切换问题的最大不同在于除了无线端基站的切换，背后还有计算任务在云端的切换，综合考虑无线端和计算端的延时，我们提出了移动云计算中基于延时的用户切换机制，通过切换迟滞量 HHM 在不明显引入额外附加延时的同时减少平均切换次数，降低切换开销。

6.2　工作展望

由于移动云计算是一种面向未来复杂应用的新型技术概念，未来工作

可以搭建实际的移动云平台，选取一种特定的移动终端应用，通过实际的能耗和延时测量，对文中分割、终端能耗、延时模型进行调整。

在应用博弈论解决云端为多用户分配有限计算资源时，可以在假设参与者都自私且理性的时候求解到纳什均衡的基础上，设计一些合作博弈的机制，从而让用户和云端能够达到比纳什均衡更优的帕累托最优解。

关于移动云计算中的移动性管理问题，可以借助排队论建立多跳排队系统分析用户的延时情况及切换成功概率。

伴随着近年大数据方法的普及以及各种新型移动应用的诞生，可以预见移动云计算作为额外计算能力提供技术有着广阔的应用前景，而要想充分利用云端丰富的计算、存储资源，无线端的随机性和移动性对整体系统性能（如终端能耗，处理延时，切换开销等）的影响将是移动云计算中的关键问题。

参考文献

[1] IMT-2020(5G)推进. 5G 愿景与需求白皮书, 2014.

[2] M. H. Kang, J. N. Froscher and B. J. Eppinger. Towards an infrastructure for MLS distributed computing. Computer Security Applications Conference, 1998. Proceedings. 14th Annual, Phoenix, AZ, 1998, pp. 91-100.

[3] Xu M, "Effective Internet grid computing for industrial users," Cluster Computing and the Grid, 2001. Proceedings. First IEEE/ACM International Symposium on, Brisbane, Qld., 2001, pp. 34-.

[4] 林闯, 苏文博, 孟坤,等. 云计算安全:架构、机制与模型评价[J]. 计算机学报, 2013, 36(9):1765-1784.

[5] 维基百科云计算. https://en.wikipedia.org/wiki/Cloud_computing

[6] NICE Cloud Computing Report. http://www.nice-software.com/solutions/cloud-computing

[7] 韩燕波等，“云计算导论——从应用视角开启云计算之门”，电子工业出版社 2015 年版，第 27 页.

[8] 维基百科 PaaS. https://en.wikipedia.org/wiki/Platform_as_a_service

[9] 张德干，“移动计算”，科学出版社 2009 年版，第 3 页.

[10] 维基百科移动计算. https://en.wikipedia.org/wiki/Mobile_computing

[11] [丹麦] Frank H.P. Fitzek,[芬兰] Marcos D. Katz 编著《移动云计算：无线、移动及社交网络中分布式资源的开发利用》，郎为民等译，机械工业出版社 2014 年版，第 13 页.

[12] 维基百科移动云计算. https://en.wikipedia.org/wiki/Mobile_cloud_computing

[13] Cuervo E, Balasubramanian A, Cho D, et al. MAUI: making smartphones last longer with code offload. Proceedings of the 8th international conference on Mobile systems, applications, and services. ACM, 2010: 49-62.

[14] Liu J, Zhao T, Zhou S, Cheng Y, and Niu Z. CONCERT: a cloudbased architecture for next-generation cellular systems. IEEE Wireless Commun. Mag., vol. 21, no. 6, pp. 14–22, Dec 2014.

[15] Sanaei Z, Abolfazli S, Gani A, et al. Heterogeneity in mobile cloud computing: taxonomy and open challenges. Communications Surveys & Tutorials, IEEE, 2014, 16(1): 369-392.

[16] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo. Communicating while computing: Distributed mobile cloud computing over 5G heterogeneous networks. IEEE Signal Process. Mag., vol. 31, no. 6, pp. 45–55, Nov 2014.

[17] Zhao T, Zhou S, Guo X, Zhao Y, Niu Z. A Cooperative Scheduling Scheme of Local Cloud and Internet Cloud for Delay-Aware Mobile Cloud Computing. in 2015 IEEE GLOBECOM Workshop, Dec 2015.

[18] Zhao T, Zhou S, Guo X, Zhao Y, Niu Z. Pricing Policy and Computational Resource Provisioning for Delay-aware Mobile Edge Computing. accepted by IEEE ICCC 2016.

[19] 中国移动通信研究院.C-RAN:无线接入网绿色演进，2013.

[20] Chih-Lin I, Huang J, Duan R, et al. Recent Progress on C-RAN Centralization and Cloudification. Access IEEE, 2014, 2:1030-1039.

[21] Kumar K, Liu J, Lu Y H, et al. A Survey of Computation Offloading for Mobile Systems. Mobile Networks & Applications, 2013, 18(1):129-140.

[22] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015–2020 White Paper. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html

[23] Lin Du; Biaham, J. Cuthbert, L. A bubble oscillation algorithm for distributed geographic load balancing in mobile networks. INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies

[24] Peng Jiang; Bigham, J. Anas Khan, M. Distributed Algorithm for Real Time Cooperative Synthesis of Wireless Cell Coverage Patterns. Communications Letters, IEEE , vol.12, no.9, pp.702,704, September 2008

[25] Cao D, Zhou S, Niu Z. Optimal base station density for energy-efficient heterogeneous cellular networks. Communications (ICC), 2012 IEEE International Conference on. IEEE, 2012: 4379-4383.

[26] Zhang S, Zhou S, Niu Z, et al. A practical channel allocation scheme based on the weighted conflict graph in heterogeneous networks. Communications in China (ICCC), 2014 IEEE/CIC International Conference on. IEEE, 2014: 615-620.

[27] S. Sardellitti, S. Barbarossa, and G. Scutari. Distributed mobile cloud computing: Joint optimization of radio and computational resources. 2014 Globecom Workshops (GC Wkshps), Dec 2014, pp. 1505–1510.

[28] S. Abolfazli, Z. Sanaei, E. Ahmed, A. Gani, and R. Buyya. Cloudbased augmentation for mobile devices: Motivation, taxonomies, and open challenges. IEEE Commun. Surveys Tuts., vol. 16, no. 1, pp. 337–368, Jan 2014.

[29] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. Wu. Energy-optimal mobile cloud computing under stochastic wireless channel. IEEE Trans. Wireless Commun., vol. 12, no. 9, pp. 4569–4581, Sep 2013.

[30] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. The case for vm-based cloudlets in mobile computing. IEEE Pervasive Comput.,vol. 8, no. 4, pp. 14–23, Oct 2009.

[31] B. Chun and P. Maniatis. Augmented smartphone applications through clone cloud execution. in Conference on Hot Topics in Operating Systems, 2009.

[32] O. Munoz-Medina, A. Pascual-Iserte, and J. Vidal. Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading. IEEE Trans. Veh. Technol., vol. PP, no. 99, pp. 1–1, 2014.

[33] Xueying Guo, Rahul Singh, Tianchu Zhao, Zhisheng Niu. An Index Based Task Assignment Policy for Achieving Optimal Power-Delay Tradeoff in Edge Cloud Systems. in Proc. IEEE ICC 2016.

[34] S. Sardellitti, G. Scutari, and S. Barbarossa. Distributed joint optimization of radio and computational resources for mobile cloud computing. in 2014 IEEE 3rd International Conference on Cloud Networking (CloudNet), Oct 2014, pp. 211–216.

[35] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo. Joint allocation of computation and communication resources in multiuser mobile cloud computing. in 2013 IEEE 14th Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Jun 2013, pp. 26–30.

[36] A. P. Miettinen and J. K. Nurminen. Energy efficiency of mobile clients in cloud computing. in Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing, ser. HotCloud'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 4–4. [Online]. Available: http://dl.acm.org/citation.cfm?id=1863103.1863107

[37] D. P. Palomar and Mung Chiang. A tutorial on decomposition methods for network utility maximization. in IEEE Journal on Selected Areas in Communications, vol. 24, no. 8, pp. 1439-1451, Aug. 2006.

[38] Arif Ahmed and Ejaz Ahmed. A Survey on Mobile Edge Computing. accepted by 10th IEEE International Conference on Intelligent Systems and Control (ISCO 2016).

[39] 段翔.无线网络中基于效用的无线资源管理与服务质量控制[博士学位论文]. 北京：清华大学电子工程系.2004.

[40] https://help.aliyun.com/document_detail/slb/buy-guide/slb-price.html?spm=5176.775974053.0.0.1Pl6mz

[41] [美] Robert Gibbons 编著《博弈论基础》，高峰译，中国社会科学出版社 1999 年版，第 6 页.

[42] Velez F J, Correia L M. Traffic from mobility in mobile broadband systems[J]. TELEKTRONIKK, 1998, 94: 95-101.

[43] Xenakis D, Passas N, Merakos L, et al. Mobility management for femtocells in LTE-advanced: key aspects and survey of handover decision algorithms. Communications Surveys & Tutorials, IEEE, 2014, 16(1): 64-91.

[44] Xenakis D, Passas N, Verikoukis C. An energy-centric handover decision algorithm for the integrated LTE macrocell–femtocell network[J]. Computer Communications, 2012, 35(14):1684-1694.

[45] Sharma S, Chawla M. A technical review for efficient virtual machine migration//Cloud & Ubiquitous Computing & Emerging Technologies (CUBE), 2013 International Conference on. IEEE, 2013: 20-25.

[46] Giovanni Giambene, Queuing Theory and Telecommunications: Networks and Applications (2005 Springer Science + Business Media). p.562.

[47] Zhao Y, Zhou S, Zhao T, and Niu Z. Energy-Efficient Task Offloading for Multiuser Mobile Cloud Computing. in IEEE ICCC 2015 , Shenzhen, China, Nov. 2015.

致　谢

短短三年的学习时光如白驹过隙般匆匆而过。纵然短暂，但亦让我收获良多，不仅开拓了视野，找准了人生方向，更平添了些豁达与成熟。这期间的精彩的生活自不必多说，更重要的是，老师们给予我的谆谆教诲，和同学们建立的深厚友情，都将是我今后人生路上的宝贵财富。

我的导师牛志升教授之于我，既是传道授业的师长，亦是和蔼可亲的家长。和牛教授相处的点点滴滴都让我真切地感受到老师对学生最朴实真挚的关怀。论文选题时牛教授着重对大方向的把控，尊重学生兴趣，鼓励我们自由探索的科研精神。在论文写作时，牛教授又对细节严格要求，一丝不苟，敦促我养成扎实严谨的学术作风。在我决定留学深造时，无论是学校申请上还是拿到录取后学校的选择上，牛教授都给予了我支持、鼓励与帮助，在我需要做出关键选择的时候给我醍醐灌顶的人生指导。

同时，我也要感谢周盛老师一直以来对我的指点和启发。周老师如兄长一般，不厌其烦地帮我解答学业上的问题，这份耐心背后藏着的是对学生的关爱。

不仅老师们平易近人，师兄师姐们也都热情友善。无论是学习上的问题，还是生活中的困扰，他们都耐心倾听并真诚地给予建议。

本论文课题承蒙 Intel 项目资助，特此感谢。

此外，还要感谢实验室 Niulab 的小伙伴们，我们从陌生到熟悉好像并没有花费太多的时间，友情就这样自然而然地生根发芽，枝繁叶茂。从学习讨论，到集体聚会，和各种学生活动，一路走来我们有太多灿烂的回忆。祝福大家都能有似锦的前程，如诗的人生！

最后，一定要感谢我的父母，他们对我无私的爱与支持始终是我最坚实的后盾，让我敢于去尝试人生更多的可能。

声　明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签　名：_____日　期：_____

个人简历、在学期间发表的学术论文与研究成果

个人简历

1992 年 12 月 07 日出生于内蒙古乌兰察布市。

2008 年 9 月考入浙江大学通信与通信工程专业，2012 年 6 月本科毕业并获得工学学士学位。

2013 年 9 月进入清华大学电子系攻读工学硕士至今。

学术论文

[1] Yun Zhao, Sheng Zhou, Tianchu Zhao, and Zhisheng Niu. Energy-Efficient Task Offloading for Multiuser Mobile Cloud Computing. in IEEE ICCC 2015 , Shenzhen, China, Nov. 2015. (EI)

[2] Tianchu Zhao, Sheng Zhou, Xueying Guo, Yun Zhao, and Zhisheng Niu. A Cooperative Scheduling Scheme of Local Cloud and Internet Cloud for Delay-Aware Mobile Cloud Computing. in IEEE Globecom Workshop 2015 , San Diego, CA, USA, Dec. 2015. (EI)

[3] Yangtian Yan, Bangcheng Sun, Yun Zhao, Zhenhui Huang, Hui Yang, and Jian Song. A Bi-directional Visible Light Communication System Based on DTMB-A. in IEEE VTC 2016 ,Nanjing, China, May, 2016. (EI)

[4] Tianchu Zhao, Sheng Zhou, Xueying Guo, Yun Zhao, and Zhisheng Niu, “A Cooperative Scheduling Pricing Policy and Computational Resource Provisioning for Delay-aware Mobile Edge Computing,” accepted by IEEE ICCC 2016 , Xi’an , China, Jul. 2016. (EI)