

分类号	
UDC	
密 级	
学 号	2151220042

西安理工大学

硕士学位论文

移动边缘计算环境下的任务迁移策略研究

孔月

学 科 门 类： 工学

学 科 名 称： 软件工程

指 导 教 师： 张毅坤 教授

申 请 日 期： 2018 年 6 月

独创性声明

本人所呈交的学位论文是在导师指导下进行的研究工作及取得的成果。尽我所知，除特别加以标注的地方外，论文中不包含其他人的研究成果。与我一同工作的同志对本文的研究工作和成果的任何贡献均已在论文中作了明确的说明并已致谢。

本论文及其相关资料若有不实之处，由本人承担一切相关责任。

论文作者签名：孔月 2018年6月19日

学位论文使用授权

本人作为学位论文作者了解并愿意遵守学校有关保留、使用学位论文的规定，即：在导师的指导下创作完成的学位论文的知识产权归西安理工大学所有，本人今后在使用或发表该论文涉及的研究内容时，会注明西安理工大学。西安理工大学拥有学位论文的如下使用权，包括：学校可以保存学位论文；可以采用影印、缩印或其他复制手段保存论文；可以查阅或借阅。本人授权西安理工大学对学位论文全部内容编入公开的数据库进行检索。本学位论文全部或部分内容的公布（包括刊登）授权西安理工大学研究生院办理。

经过学校保密办公室确定密级的涉密学位论文，按照相关保密规定执行；需要进行技术保密的学位论文，按照《西安理工大学学位论文技术保密申请表》内容进行保密（附《西安理工大学学位论文技术保密申请表》）。

保密的学位论文在解密后，适用本授权。

论文作者签名：孔月 导师签名：张强 2018年6月19日

论文题目：移动边缘计算环境下的任务迁移策略研究

学科名称：软件工程

研究生：孔月

签名：孔月

指导教师：张毅坤 教授

签名：张毅坤

摘 要

随着移动互联网和物联网技术的快速发展与普及，人类已经进入万物互联的时代。网络边缘设备数量迅速增加，物联网感知层产生海量数据，导致云计算网络传输负载量急剧增加，造成较长的网络延迟；移动边缘计算作为未来 5G 网络提高用户体验度的关键技术，具有靠近数据源的特性，使其能够有效降低网络传输延迟。边缘计算将业务下沉到网络边缘，提供计算服务和存储服务，并为用户提供任务迁移平台。

任务迁移技术可以将移动设备上复杂的任务通过无线网络迁移到远端边缘服务器执行，依靠远端服务器丰富的计算资源，完成计算任务，并将结果回传给移动设备，从而解决移动终端计算能力不足、电池容量有限等问题。

目前现有任务迁移策略均是在迁移服务节点已经确立的前提下进行迁移决策的，并未考虑到多服务节点可选时的情景，而且绝大多数迁移策略中未考虑移动终端应用内部具有多依赖关系的复杂情况，这些策略的适用场景较为单一，不能充分发挥移动边缘计算的特性和优势。因此本文在对云计算和移动边缘计算的学习基础上，对任务迁移的两个方向：任务迁移目的地的选取、如何根据能耗优化目标实现最优化迁移进行研究。

首先，从降低时延，提高资源利用率均衡度的角度出发，解决多服务节点可选场景下“迁移到何处”的问题。提出基于供需相似度与动态价格模型的任务迁移节点选取策略，在移动边缘计算环境中选取最佳迁移地点，将任务迁移至最优的边缘微型数据中心执行，提高资源利用率和任务吞吐量，在 CloudAnalyst 改造仿真平台下的实验结果表明，该选取算法能够有效降低网络延时和响应时间，提高资源利用率和任务吞吐量。

其次，由于目前移动设备的电池发展速度远远跟不上其处理器和存储器的发展速度，成为制约移动终端设备发展的重要因素，无法满足新兴物联网应用的能耗需求，因此，从降低能耗的角度出发，解决“如何实现复杂依赖关系型应用低能耗迁移”的问题，在任务迁移目的地已经确定的基础上，对于复杂多依赖关系的移动终端应用场景，针对移动边缘计算特性，建立细粒度有向无环图状任务划分模型，对划分后的每一个子任务及其之间的关系进行分析，构建执行时间限制下的最小化能耗问题，并利用遗传算法求最优解，得出针对每一个子任务的迁移决策结果，即整个移动终端应用的最优化节能迁移方案，实验结果显示，本文所提出的细粒度任务迁移策略充分发挥了移动边缘计算的优势，在满足任务

执行时延的前提下，能够有效减少移动终端设备的能耗。

论文得到国家自然科学基金（No.61602376）、陕西省自然科学基金（No.2017JQ6020）和陕西省教育厅科技专项（No.16JK1573）的资助。

关键词：物联网；5G；移动边缘计算；任务迁移；节约能耗

**Title: THE RESEARCH ON TASK MIGRATION STRATEGY IN MOBILE
EDGE COMPUTING ENVIRONMENT**

Major: Software Engineering

Name: Kong Yue

Signature: Kong Yue

Supervisor: Prof. Yikun Zhang

Signature: Yikun Zhang

Abstract

With the rapid development and popularization of mobile Internet and Internet of Things, People have entered the era of Internet of Everything. With the rapid increase in the number of network edge devices, mass data is generated by the perception layer of the Internet of Things, which leads to a sharp increase in the load of the cloud computing network, resulting in a long network delay; Mobile edge computing is the key technology to improving the user experience of 5G network in the future, which is close to the data source, so that it can effectively reduce the network transmission delay. Edge computing will sink the business to the edge of the network, provide computing services and storage services, and provide task migration platform for users.

Task migration technology can transfer complex tasks on mobile devices to remote edge server through wireless networks, rely on the rich computing resources of remote servers, complete computing tasks, and return the results to mobile devices, so as to solve the problems of inadequate computing power and limited battery capacity of mobile terminals.

At present, the existing task migration strategy is to make the migration decision under the premise that the migration service node has been established. It does not take into account the scenarios when the multi service nodes are available, and most of the migration strategies do not take into account the complex situation of the multi dependency relationship within the mobile terminal application. The application scenarios of these strategies are relatively simple and cannot give full play to the characteristics and advantages of mobile edge computing. On the basis of the learning of cloud computing and mobile edge computing, this paper studies the two directions of task migration: the selection of task migration destination and how to optimize the migration according to the target of energy consumption optimization.

First of all, from the perspective of reducing the delay and improving the resource utilization equilibrium, the issue of "where to migrate" is solved under the scenario of multi service nodes. A task migration destination selection strategy based on similarity between

supply and demand and dynamic price model is proposed. The optimal migration location is selected in the mobile edge computing environment. The task is migrated to the optimal edge micro data center to improve the resource utilization and task throughput. The experimental results under the CloudAnalyst retrofit simulation platform show that this selection algorithm can effectively reduce the network delay and response time, and improve resource utilization and task throughput.

Secondly, due to the current battery development speed of mobile devices is far behind the development speed of its processor and memory, it has become an important factor constraining the development of mobile terminal equipment, unable to meet the energy consumption requirement of emerging Internet of Things applications. Therefore, to solve the problem of "how to realize the low energy migration of complex dependency application", from the perspective of reducing energy consumption, based on the determination of task migration destination, for mobile terminal scenarios with complex multi-dependencies, a fine-grained directed acyclic graph task partition model is established based on the characteristics of mobile edge computing, and the relationships among the divided subtasks are analyzed to construct the minimization energy consumption problem under the execution time limit, then uses the genetic algorithm to find the optimal solution, and obtains the result of the migration decision for each sub task, that is, the optimal energy-saving migration plan for the entire mobile terminal application. The experimental results show that the fine-grained task migration strategy proposed in this paper makes full use of the advantages of mobile edge computing, and can effectively reduce the energy consumption of mobile terminal devices on the premise of meeting the task execution delay.

The paper was supported by National Natural Science Funds of China (No.61602376), National Natural Science Funds of Shaanxi (No.2017JQ6020), and Science Technology Project of Shaanxi Education Department (No.16JK1573).

Key words: Internet of Things; 5G; Mobile Edge Computing; Task migration; Energy saving

目 录

1 绪论.....	1
1.1 课题研究背景及意义.....	1
1.1.1 研究背景.....	1
1.1.2 本研究的目的及意义.....	3
1.2 国内外研究现状.....	3
1.2.1 移动边缘计算研究现状.....	4
1.2.2 任务迁移研究现状.....	5
1.3 主要研究内容.....	6
1.4 论文组织结构.....	7
1.5 本章小结.....	8
2 任务迁移研究思路与框架.....	9
2.1 移动边缘计算基本架构分析.....	9
2.1.1 移动边缘计算的基本概念及架构.....	9
2.1.2 移动边缘计算的特征及优势.....	10
2.2 任务迁移系统软件架构.....	12
2.2.1 任务迁移系统软件架构.....	12
2.2.2 任务迁移决策过程.....	13
2.3 本文研究的整体思路.....	16
2.4 本章小结.....	19
3 基于供需相似度与动态价格模型的迁移地点选取策略.....	21
3.1 供需相似度与动态价格模型分析.....	21
3.1.1 任务迁移场景.....	21
3.1.2 供需相似度思想.....	22
3.1.3 动态价格模型.....	23
3.2 任务迁移地点选取算法.....	23
3.2.1 任务迁移系统框架.....	23
3.2.2 任务迁移地点选取算法思想.....	25
3.3 迁移地点选取算法实现.....	26
3.3.1 供需相似度的计算.....	26
3.3.2 动态价格的计算.....	27
3.3.3 迁移权重的计算.....	28
3.4 实验结果及分析.....	29
3.4.1 仿真场景与参数设置.....	30

3.4.2 仿真结果分析.....	31
3.5 本章小结.....	33
4 基于遗传算法的任务迁移节能策略.....	35
4.1 任务划分模型.....	35
4.1.1 现有的任务划分模型.....	35
4.1.2 细粒度有向无环图状任务划分模型.....	37
4.2 能耗和时间模型.....	38
4.2.1 任务执行时间和能耗模型.....	39
4.2.2 数据传输时间和能耗模型.....	40
4.3 最小化能耗问题建模.....	41
4.4 基于遗传算法的细粒度任务迁移节能策略.....	43
4.4.1 初始种群的选取和编码.....	44
4.4.2 适应度函数.....	45
4.4.3 交叉方式的设计.....	45
4.4.4 变异方式的设计.....	46
4.4.5 遗传终止条件.....	47
4.5 实验结果及分析.....	47
4.5.1 仿真场景与参数设置.....	47
4.5.2 仿真结果分析.....	48
4.6 本章小结.....	50
5 总结与展望.....	51
5.1 研究总结.....	51
5.2 研究展望.....	52
致谢.....	53
参考文献.....	55
在校参与项目成果及获奖.....	59

1 绪论

1.1 课题研究背景及意义

1.1.1 研究背景

物联网，又被称为传感网，是信息产业在新时代下的最新发展，仅需在物体中嵌入一个小小的芯片，它就能够将不同的事物连接起来^[1,2]。物联网技术被广泛应用于安防、农业、医疗和交通等领域，为人们生活提供了便捷的服务。

根据思科互联网业务解决方案集团预测^[3]，到 2020 年，连接到网络的无线设备数量将达到 500 亿，在万物互联的时代背景下，网络边缘设备产生的数据已经达到海量级别，而且新增的业务场景，如无人驾驶汽车、AR、智能电网等，也会涵盖农业、交通、医疗、零售和公共事业等多个方面，这样就对时延、功耗等提出了更高的要求，日后基于物联网平台的应用服务通常需要更短的响应时间^[4]。

简而言之，未来物联网的发展，需要海量的连接和毫秒级的时延，这都是当前 4G 网络无法实现的，为了应对移动互联网和物联网的超高速发展，5G 概念应运而生，5G 网络，又被称为“第五代移动网络”，它是在 4G 网络的基础上发展而来的，传输速度是 4G 网络的百倍之上，5G 移动网络具有低延时、低能耗、广覆盖和超密集组网的特点，它的应用能够为物联网灵活处理多样化的数据提供支持，符合未来物联网发展的需求，使物联网功能更为强大^[5,6]。

图 1-1 是 5G 网络的总体愿景图，从图 1-1 可以看出，未来 5G 将逐渐渗透到社会的各个领域，以用户为中心，构建全方位的信息生态系统，便捷地实现人与万物的智能互联和极佳的交互体验^[7,8]。

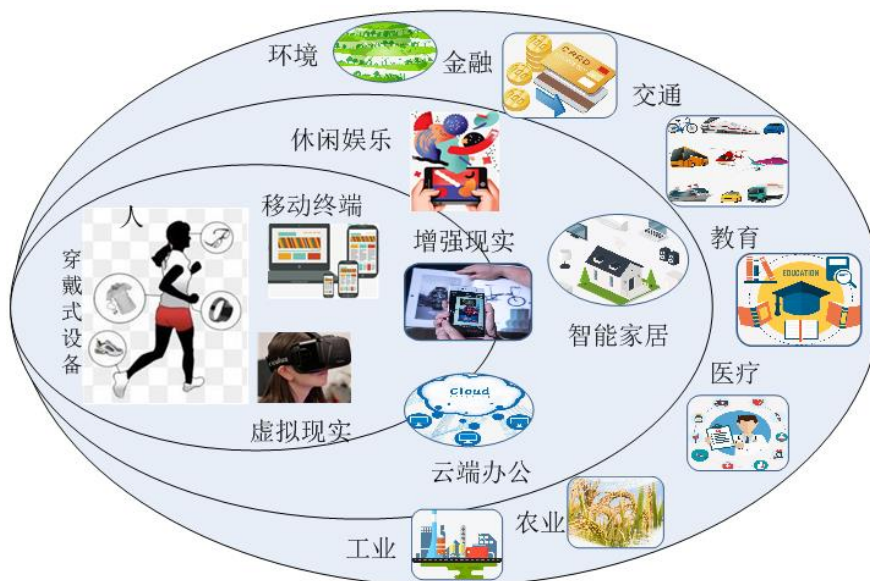


图 1-1 5G 总体愿景图

Figure 1-1 5G Overall vision

5G 技术以“大容量、大带宽、低时延、低功耗”为诉求^[9]，根据联合国国际电信联盟（ITU）对 5G 的标准要求，5G 标准包括增强型移动带宽（eMBB）、低延时高可靠（URLLC）以及海量机器类通信（eMTC）三大应用场景，并定义了以下关键指标：峰值吞吐率 10Gbps、时延 1ms、连接数 100 万/km²、高速移动性 500km/h。4G、5G 时代下，各类移动应用对网络的要求更高、更复杂，但是在目前的网络架构中，由于核心网的高位置部署传输延时比较大，不能满足超低时延的业务需求；此外，业务完全在云端终结并非完全有效，尤其一些区域性业务不在本地终结，既浪费带宽，也增加时延。

因此，时延指标和连接数指标决定了 5G 业务的终结点不可能全部都在核心网后端的云平台。对带宽、时延要求苛刻的业务数据的转发路径止于无线侧的理念越来越成为业界共识，而伴随该理念产生的移动边缘计算（Mobile Edge Computing, MEC）则成为了未来 5G 网络提高用户体验的关键技术^[10]。

移动边缘计算系统允许移动终端设备将计算任务迁移到网络边缘节点处理，如无线接入点、基站等，移动边缘计算系统基本框架图如图 1-2 所示，移动边缘计算系统的设计理念，既满足了移动终端设备扩展计算能力的需求，同时也弥补了云计算平台传输时延较长的缺点。

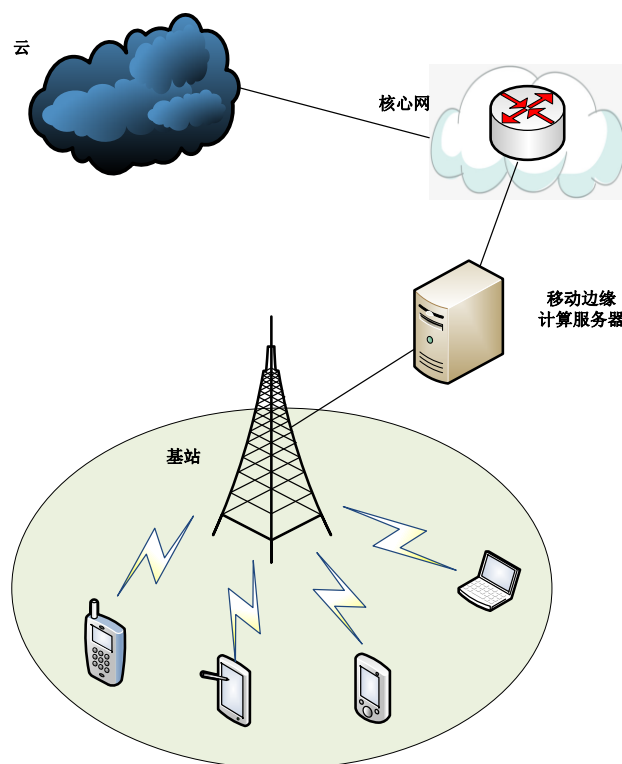


图 1-2 移动边缘计算系统基本框架图

Figure 1-2 Basic frame diagram of mobile edge computing system

从图 1-2 可以看出，在 MEC 平台中，边缘计算服务器部署在无线接入网侧，大大缩短了其与用户设备的距离。传统的云计算中，数据需要通过无线接入网和核心网才能到达远端的云服务器，云服务器处理完数据后，再经过漫长的回传链路将结果返回给用户，

可见利用云平台进行任务迁移的方式是十分耗时的。由于传输距离的缩短，MEC 的任务迁移不再需要经过漫长的回传链路和核心网，减少了时延开销。另一方面，由于边缘服务器的计算能力远大于移动设备，从而大幅度缩短了任务处理时间^[11]。

1.1.2 本研究的目的及意义

在万物互联的时代背景下，物联网和移动互联网蓬勃发展，爆炸式的数据流量增长和海量的设备连接为 5G 带来挑战^[11]，新型物联网应用不断涌现，理想时延要求达到毫秒级别，但是终端设备的计算资源往往有限，而且随着各式各样的传感器和越来越丰富的应用程序的出现，移动设备的电池和节能技术的发展速度也远远跟不上人们急速增长的能耗需求^[12]，需要依赖其他资源丰富的平台来弥补自身资源的不足，然而云计算平台较长的通信链路导致高延时，已经不再满足该发展场景下的业务需求^[13]，随着移动边缘计算的兴起，可以将智能移动终端的任务迁移至移动边缘计算平台上，解决移动终端资源不足的问题，并且有效降低时延，降低能耗。

以往的迁移策略都是针对云计算平台提出的，一般都是将整个任务作为一个迁移对象进行迁移，迁移粒度较大^[14]。有的改进策略中，将任务划分成细粒度的具有链式关系的多个子任务^[15]，并且构建最小化任务完成时间的问题模型，进行求解，它虽然细化了任务的划分，但是也仅仅适用于子任务间是链式关系的情况，不适应于子任务间依赖关系复杂的应用。而且以上策略均是在服务节点已经确立的前提下进行迁移决策的，并未考虑到多服务节点可选时的情景。因此，研究适用于移动边缘计算环境的迁移策略，发挥移动边缘计算的优势，有效降低时延和能耗，是目前亟需解决的问题之一。

本文分别从移动边缘计算环境下的任务迁移节点的选取、实现复杂依赖关系型应用低能耗迁移展开研究。针对现有迁移策略均是在服务节点已经确立的前提下进行迁移决策，未考虑移动边缘计算平台中存在多服务节点场景的问题，提出多服务节点场景下的任务迁移目的地选取策略，将任务迁移至与其最匹配、合适的服务节点中执行，达到低时延、低开销的要求；针对现有迁移策略未考虑子任务间存在多依赖关系，以及移动终端电池发展速度满足不了移动终端应用日益增长的能耗需求的问题，提出适用于复杂依赖关系型移动终端应用的任务迁移节能策略，以保障在时延约束下有效降低能耗。本文的任务迁移策略普适性更强，适用范围更广。

1.2 国内外研究现状

为了充分发挥移动边缘计算的优势，研究适用于移动边缘计算环境的迁移策略，需要从移动边缘计算和任务迁移两个方面进行深入学习，本小节对移动边缘计算和任务迁移的研究现状进行较为详细地分析，阐述现阶段的对于任务迁移策略的研究方法及存在问题。

1.2.1 移动边缘计算研究现状

边缘计算是指将云计算的能力“下沉”到网络边缘,它是在网络边缘设备上执行任务的一种新型计算模型,边缘计算的“边缘”指从网络边缘终端设备所产生的数据源到云计算中心数据路径之间的任意资源^[16],其基本理念就是将计算任务在接近数据源的计算资源上运行。边缘计算与云计算不是非此即彼而是相辅相成的关系,它们各自弥补了对方的不足,共同促进物联网发展^[17]。

移动边缘计算是边缘计算的一种形式,它位于无线接入点与有线网络之间,使传统的无线接入网具备了业务本地化和近距离部署的条件,进一步为终端用户提供更高带宽、更低时延的数据服务,并大幅度减少核心网的网络负荷,同时降低数据业务对网络回传的带宽要求^[18]。

移动边缘计算技术(MEC)的概念最早提出于2009年卡内基梅隆大学所研发的Cloudlet平台^[19],2011年,思科全球研发中心的总裁DR. Flavio Bonomi首次提出雾计算(Fog computing)的概念,用其形象地形容了介于云计算和终端计算的中间态^[20, 21]。欧洲电信标准协会(ETSI)于2014年正式定义了MEC的基本概念,成立相关工作组(ETSI Mobile Edge Computing Industry Specification Group),同时发布移动边缘计算概况技术白皮书(Mobile-edge computing introductory technical white paper),对MEC的产业目标、服务场景、网络架构和技术挑战等方面进行详细阐述。2016年,华为联合英特尔、ARM和软通动力等公司在国内倡议发起的“边缘计算产业联盟”在北京正式成立^[22],该联盟旨在搭建边缘计算产业合作平台,推动运营技术(Operation Technology)和信息与通信技术(Information and Communication Technology)开放协作。

在学术界中,移动边缘计算也成了近几年的研究热点,Sabella D等人介绍了移动边缘计算在物联网中的应用价值,提出MEC在物联网应用中的部署架构^[23, 24];Huertacanepa G等人提出了将移动终端和其周围的终端共同组成一个虚拟的云的方案,协同解决计算任务,用于解决网络条件不好情况下,移动终端无法接入云或者接入云需要产生高昂带宽费用的问题^[25];Shanhe研究团队实现一种面向面部识别应用的概念验证平台,将人脸识别应用的计算任务从云计算中心迁移到边缘,从云到边缘的响应时间从900ms减少到169ms^[26];李洪星研究移动边缘计算组网模型,实现接入网络与云计算的融合,搭建移动边缘计算平台^[27];Ha K提出了一种实现边缘计算的体系结构,在该结构中,由移动设备、Cloudlet以及云三层构成,Cloudlet位于云和终端设备之间^[28];祁兵等人提出基于边缘计算的家庭能源系统框架,设计了基于异构平台间数据融合和操作协同的四层架构,为边缘计算技术在智慧用能领域的应用提供新思路^[29]。

以上研究都是着重于提出移动边缘计算框架,实现移动边缘计算平台,并验证其可行性,任务迁移技术作为提升移动终端运行性能的主要技术,在新兴的移动边缘计算环境下的相关研究却是相对较少的,目前现有的计算迁移策略大都是针对于云计算环境的,

因此，本文将对此领域进行深入研究，根据移动边缘计算的特性，研究适用于移动边缘计算环境的任务迁移策略。

1.2.2 任务迁移研究现状

任务迁移是解决移动终端资源受限问题的一个重要途径，它通过将移动终端的计算密集型任务从本地发送到远端设备中执行，利用远端资源来扩展本地资源，最终再将结果返回给移动终端。

整个任务迁移过程如图 1-3 所示，首先，智能移动终端向移动边缘计算平台发送任务请求，然后，移动边缘计算平台根据任务请求执行相应的任务，最后，移动边缘计算平台将任务结果返回给用户。目前，移动边缘计算所支持的移动终端应用有：导航地图、图像处理、视频处理、游戏应用、多媒体服务、语音识别、医疗服务、汽车自动驾驶和智能交通等。

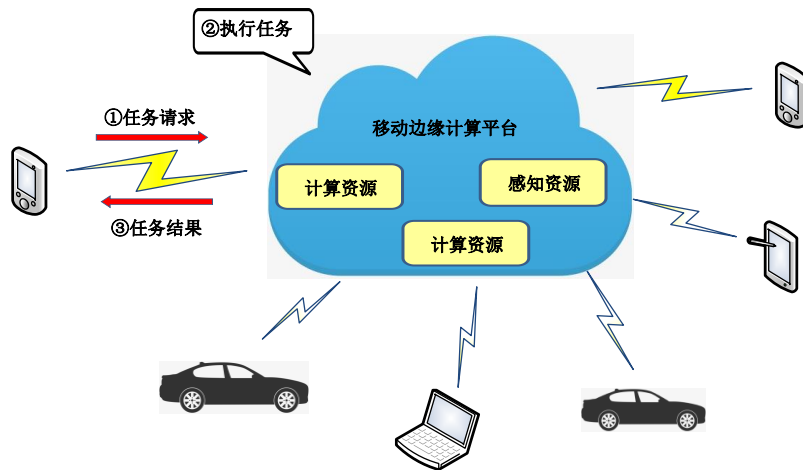


图 1-3 任务迁移过程图

Figure 1-3 Task migration process diagram

Cuervo E 和 Cho D K 等人提出 MAUI 任务迁移平台，通过对应用程序代码进行标识以支持任务迁移^[30]；Chae D 和 Kim J 等人提出着重成本开销的任务迁移平台 CMcloud，将尽可能多的移动应用程序迁移到单个服务器上，最小化服务器成本^[31]。

以上都是针对迁移平台实现的研究，由于不同的任务具有不同的计算量和数据传输量，为了确保智能移动终端能够通过任务迁移提高性能，需要使用任务迁移策略来进行迁移指标衡量，判断任务是在本地移动终端执行，还是迁移到远端服务器上执行，目前，国内外针对于云计算环境的任务迁移策略研究相对较多。

Lagerspetz E 提出了最基本的迁移评判标准：当进行任务迁移所需的能耗小于本地执行的能耗时，进行迁移计算，否则，在终端进行计算，并且得出一个基本规则：当数据量越小，计算量越大，带宽越高的时候，使用任务迁移的节能效果越好；当数据量越大，

计算量越小,带宽越低的时候,使用本地执行的效果会更好一些^[32]。但是这种策略只单纯考虑了能耗一个因素,并未顾及到时间要求。Zhang H 考虑了能耗和时间两个因素,认为如果任务在终端的执行时间超过用户的延迟容忍阈值,则将任务迁移至远端服务器,反之,继续判断任务在终端的执行能耗是否大于在远端执行的能耗,如果大的话,将任务迁移至远端服务器,否则就在本地执行^[33]。

但是以上研究的卸载粒度过大,仅能描述整个移动终端应用在本地移动终端执行或者被整体迁移至移动边缘计算平台执行,随着代码分解技术的发展,MEC 中的任务划分模型得到了广泛关注,也就是将一整个复杂的移动终端应用划分为多个功能模块,将各功能模块作为任务卸载决策的最小单位。

Jia M 将任务分解成多个子任务,并且这些任务的关系是链式的,一个任务的输出结果是下一个任务的输入源,每一个任务可能在终端执行,也可能在远端执行,它们之间的距离,就代表了数据传输能耗,因此将最小化能耗问题构建成最短路径问题,并且利用基于花销的拉格朗日松弛算法 LARAC (Lagrangian Relaxation Based Aggregated Cost) 来获取近似最优解^[34]; Huang D 和 Wang P 等人根据李雅普诺夫优化理论 (Lyapunov optimization method),提出根据无线网络环境动态改变任务迁移策略,最小化移动应用的执行能耗^[35]; Li T 和 Wu M 等人综合能量消耗、时间延迟和服务器执行成本三个因素,提出了优化方案,用于移动边缘计算环境下的任务迁移^[36],但是这种多目标优化的迁移算法往往复杂度较高,为了能够有效降低复杂度,Wang J 等人提出了一种基于李雅普诺夫优化理论的低复杂度任务迁移算法,同时降低智能移动终端的能耗和任务执行时间^[37]。

从以上的任务迁移相关研究现状可以看出,现有的任务迁移策略中,有的是面向能耗的单目标决策方法,有的是面向响应时间和能耗的多目标决策方法,有的是以整个移动终端应用作为迁移对象,有的是将整个移动终端应用划分为多个具有链式依赖关系的子任务作为最小迁移单位。但是这些迁移策略都是在计算服务地点已经确定,即迁移目的地已经确定的前提下进行迁移决策,但是实际应用场景中,可能存在多个可供选择的远程计算实体;而且以上迁移策略中,未考虑到移动终端应用内部关系较为复杂的情况,即子任务间具有多依赖关系,一个子任务的输入源来自于多个前置子任务的输出结果的情况。

因此,本文的研究重点是针对移动边缘计算环境下多服务节点可选的场景,研究任务迁移节点选取策略,并且针对具有多依赖关系的移动终端应用进行最优化节能迁移决策,以达到任务限制下最小化移动终端能耗的目标。

1.3 主要研究内容

本文首先总结了物联网的发展现状,分析现有移动通信网络状况,引出 5G 网络的兴起,说明移动边缘计算的必要性;接着,指出新兴物联网应用对低时延和低能耗的要求,以及移动终端资源不足的现状,引出移动边缘计算环境下任务迁移策略的重要性,

对任务迁移的现状进行详细分析与对比,针对其未考虑移动边缘计算平台中具有多服务节点和移动终端应用内部具有多依赖关系的场景的问题,提出任务迁移目的地选取策略和最小化能耗迁移策略实现最优化迁移。具体研究内容如下:

a. 针对现有迁移策略均是在服务节点已经确立的前提下进行迁移决策,未考虑 MEC 平台中存在多服务节点场景的问题,提出多服务节点场景下的任务迁移目的地选取策略,解决“任务迁移到何处”的问题。从降低时延和花销,均衡资源利用率的角度出发,提出基于供需相似度和动态价格模型的迁移地点选取策略,使用资源利用率均衡度变化率来代表任务需求与服务节点可用资源之间的相似程度,使用动态价格模型为服务资源定价,结合两者,选取供需相似度高,定价较低的服务节点作为迁移目的地,有效降低时延,降低花销,并且提高资源利用率和任务吞吐量。

b. 针对现有迁移策略未考虑子任务间存在多依赖关系的情况,以及移动终端电池发展速度满足不了移动终端应用日益增长的能耗需求的问题,提出适用于复杂依赖关系型移动终端应用的任务迁移节能策略,从降低能耗的角度出发,解决“如何实现复杂依赖关系型应用低能耗迁移”的问题。在任务迁移目的地已经确定的基础上,建立细粒度有向无环图状任务划分模型,对划分后的每一个子任务及其之间的关系进行分析,构建执行时间限制下的最小化能耗问题,利用遗传算法求最优解,得出针对每一个子任务的迁移决策结果,即最优化迁移方案,在满足任务执行时延的前提下,有效减少移动终端设备的能耗。

1.4 论文组织结构

第一章 绪论。本章介绍了论文的研究背景,引出研究目的和意义,描述移动边缘计算和任务迁移技术在国内外的研究现状,最后归纳和总结本文的主要研究内容。

第二章 任务迁移研究思路与框架。本章首先移动边缘计算基本架构进行分析,描述其框架和特征,再对任务迁移系统组成进行详细分析,对迁移决策这一模块进行具体介绍,最后,针对现有迁移策略存在的问题提出本文整体研究流程与路线。

第三章 基于供需相似度和动态价格模型的迁移地点选取策略。本章针对移动边缘计算环境下具有多服务节点的场景,利用资源利用率均衡度的变化率来代表任务资源需求和服务节点能力的相似度,结合基于服务节点资源剩余量的动态价格模型,提出任务迁移目的地选取策略,具体介绍了迁移地点选取策略的设计思路及算法流程和实验效果。

第四章 基于遗传算法的任务迁移节能策略。本章具体介绍了在确定任务迁移地点后,针对具有多依赖关系的复杂移动终端应用场景,利用细粒度有向无环图状任务划分模型,构建时间限制下的最小化能耗模型,利用遗传算法求解,得出最优迁移方案,实现任务迁移优化节能的策略。

第五章 总结与展望。对本文研究的基于移动边缘计算环境的任务迁移策略进行总结,并对后续的研究工作和研究方向进行了展望。

1.5 本章小结

本章主要介绍了基于移动边缘计算环境的任务迁移策略的研究背景，引出其研究目的及意义，总结了国内外对移动边缘计算、以及任务迁移策略的研究现状，并且详细叙述了本文主要研究内容，最后，简要介绍了本文的章节安排。

2 任务迁移研究思路与框架

本文是针对移动边缘计算平台的任务迁移策略进行研究，因此，首先需对移动边缘计算的架构及特性进行详细了解，再对任务迁移过程和系统组成进行具体分析，并归纳总结现有移动边缘计算环境下任务迁移策略流程，对其存在的问题进行详细分析，最终提出本文研究的整体流程和思路。

2.1 移动边缘计算基本架构分析

2.1.1 移动边缘计算的基本概念及架构

边缘计算是指在靠近物或数据源头的一侧，采用网络、计算、存储、应用核心能力为一体的开放平台，就近提供服务，其应用程序在边缘侧发起，产生更快的网络服务响应，满足行业在实时处理、智能应用、安全和隐私保护等方面的基本需求。边缘计算处于物理实体和工业连接之间，或处于物理实体的顶端，而云端计算，仍然可以访问边缘计算的历史数据。

如果将云计算比作整个计算机智能系统的大脑，那么边缘计算就是这个系统的眼睛耳朵和手脚。核心服务器让智能系统具有很强的人工智能，但是如果这个人工智能是聋子瞎子，它也发挥不了太大的作用。大数据应用中常常面对的一个难题就是没有采集到合适的数据，边缘计算可以为核心服务器的大数据算法提供最准确、最及时的数据来源。

边缘计算和云计算的结合让整个智能系统不但头脑清楚，而且耳聪目明，手脚灵便。完全依赖云计算的计算机系统就好比每一件事都要请示司令部的军队，在需要大量和外界互动的时候会显得僵化，反应迟缓，而且一旦出现网络问题，系统将彻底瘫痪。加上边缘计算之后就好比让中低层军官也开始发挥主观能动性，能一定程度上自主做出智能判断和行动决策，同时也只需要把一部分经过筛选的信息上传到司令部，大大缓解了网络通讯的压力，即使在和总司令部暂时失去联系的情况下，也能自主做出部分决策。

MEC 的架构和应用示意图如图 2-1 所示：

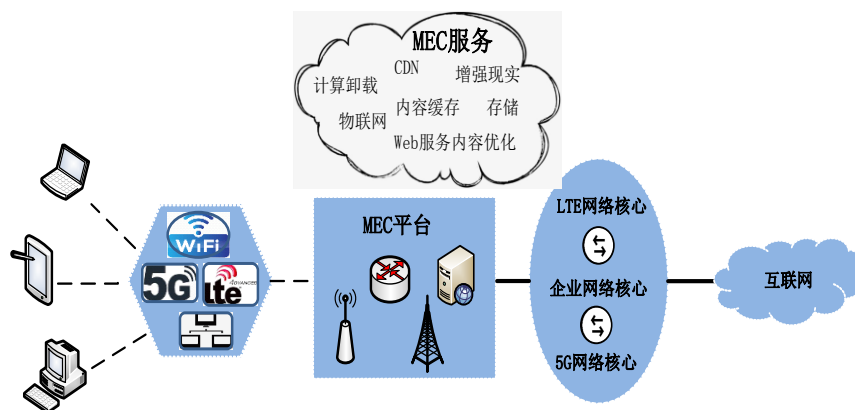


图 2-1 MEC 架构和应用示意图

Figure 2-1 MEC architecture and application schematic diagram

在 5G 时代即将来临的背景下，MEC 被视为向 5G 过渡的关键技术和架构性概念。MEC 正在推动传统集中式数据中心的云计算平台与移动网络的融合，将原本位于云数据中心的的服务和功能“下沉”到移动网络的边缘，在移动网络边缘提供计算、存储、网络 and 通信资源。MEC 强调靠近用户，从而减少网络操作和服务交付的时延，提升用户体验。

欧洲电信标准化协会（ETSI）对 MEC 的定义是：在移动网络边缘提供 IT 服务环境和云计算能力。需要说明的是，在初期，MEC 中的“M”代表的是“Mobile”，特指移动网络环境（移动蜂窝场景）。2016 年 9 月，ETSI 把 MEC 的概念扩展成多接入边缘计算（Multi Access Edge computing），意为将边缘计算能力从电信蜂窝网络进一步延伸至其他无线接入网络（如 WIFI）以及有线网络，因此在保持英文 MEC 缩写的基础上，ETSI 将其的概念改为多通路边缘计算^[38]。但是目前业界以及 ETSI 等标准制定组织研究的重点依旧是移动场景下的边缘计算，因此现在业界仍多以“移动边缘计算”称之。

2.1.2 移动边缘计算的特征及优势

移动边缘计算是延伸到网络边缘的计算，它和云计算类似，都为终端用户提供了存储、计算和网络服务。但是，移动边缘计算作为一个新的架构，意味着它具有自己的特性，使其成为现有研究热点。它与云计算的区别主要在于它的部署位置、资源性能和服务质量等。

移动边缘计算的主要特征如下：

a. 邻近的物理位置

如图 2-2 MEC 与云计算相对位置图所示，相对于云计算，移动边缘计算在物理位置上更接近“地面”，而云计算则是浮动在遥远的“天空”中。

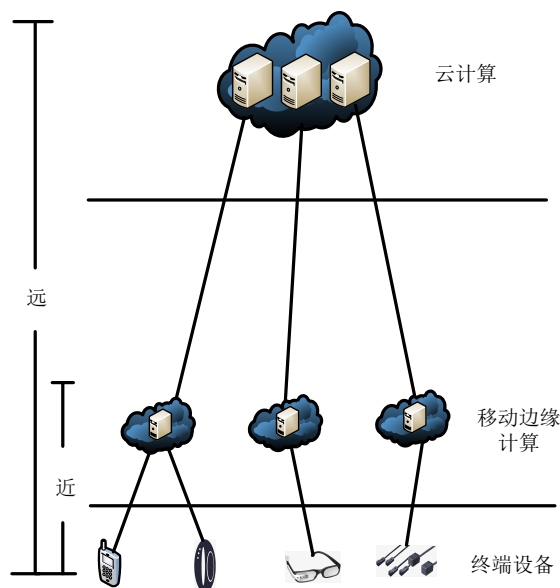


图 2-2 MEC 与云计算相对位置图

Figure 2-2 Relative position diagram of MEC and cloud computing

云计算和移动终端设备在物理距离上相距很远，用户设备需要通过核心网，经过众多的路由器，才能连接到云数据中心的服务器。移动边缘计算被部署在网络边缘，用户设备和 MEC 服务器位于同一个局域网（Local Area Network, LAN）中，局域网的数据传输速率显然比广域网（Wide Area Network, WAN）或者互联网的数据传输速率快。在物联网环境中，移动终端设备会实时采集不同类型的数据，当 MEC 服务器接收到数据的时候，会根据数据的紧急程度和类型进行分类处理和反馈。

b. 服务节点性能低

传统的云计算都是依靠超高性能的基础设施驱动的，但移动边缘计算仅需对小范围的数据进行预处理，提供实时响应功能即可，不提供复杂数据运算，如数据挖掘等功能，因此，MEC 平台的基础设施性能相较于云计算平台都不太高，计算和存储能力较弱，各项资源性能较低。

c. 支持多种通信协议

移动边缘计算不仅要支持各类异构设备不同的网络接入协议，实现物联网中短距离的传输，比如：ZigBee、2G、3G、4G 和 Wi-Fi 等；另外还要支持连接云计算数据中心的 TCP/IP 协议，实现互联网远距离的网络传输。

d. 提供分布式计算

与云计算集中式的方式相比，移动边缘计算呈现分布式的架构。云平台将计算任务划分为多个子任务，分配给多个 MEC 节点处理，云计算平台再将 MEC 节点的处理结果聚合起来，进行更加深入的分析。随着传感网络海量数据的产生，频繁的数据传输造成巨大的网络带宽压力，移动边缘计算分布式处理的架构，极大减少物联网和云计算数据中心的频繁交互，有效缓解了网络带宽 congestion 的问题。

基于以上阐述的移动边缘计算的特点，总结得出移动边缘计算的优势有：

a. 低时延

MEC 将计算能力和存储能力都“下沉”到网络边缘，由于距离用户更近，用户的请求不需要经过漫长的传输网络到达遥远的核心网被处理，而是由部署在本地的 MEC 服务器将一部分流量进行卸载，直接处理并且响应用户，因此，通信时延将会大大降低。MEC 的低时延特性，在视频传输以及 VR 等时延敏感型的相关应用中表现的尤其明显。

以视频传输为例，在不使用 MEC 的传统方式下，每个用户在发起视频内容调用请求时，首先需要经过基站接入，然后通过核心网，连接目标内容，再逐层回传。最终完成终端和该目标内容间的交互。可见这样的连接和逐层获取的方式是十分耗时的。因此，引入 MEC 解决方案，在靠近 UE（User Equipment）的基站部署 MEC 服务器，利用 MEC 提供的存储资源将内容缓存在 MEC 服务器上，用户可以直接从 MEC 服务器获取内容，不再需要通过漫长的回传链路，从相对遥远的核心网获取内容数据。这样大大缩短了从用户发出请求和被响应之间的等待时间，从而提升 QoS（Quality of Service）。

b. 改善链路容量

部署在移动网络边缘的 MEC 服务器能对流量数据进行本地卸载,从而极大地降低对传输网和核心网的带宽的要求。许多演唱会、发布会及球赛,经常是以直播这种高并发的方式发布,用一时间内,就会有大量的用户接入,请求同一资源,因此,对带宽和链路状态的要求极高。通过在网络边缘部署 MEC 服务器,可以将直播视频实时缓存在距离用户更近的地方,在本地进行用户请求的处理,从而减少了对回传链路的带宽压力,也同时可以降低发生链路拥塞和故障的可能性,从而改善链路容量。

c. 提高能效

在移动网络环境下,网络的能量消耗主要包括人物计算能耗和数据传输能耗两个部分,MEC 的引入能够极大的降低网络的能量消耗,由于 MEC 自身具有计算和存储资源,能够在本地进行部分计算的卸载,对于需要进行大量计算能力的再考虑上交给距离更远、处理能力更强的云数据中心进行处理,这样可以尽量避免不必要的传输能耗以及核心网的计算能耗。另一方面,随着缓存技术的发展,存储资源相对于带宽资源来说成本逐渐降低,MEC 的部署也是一种以存储换带宽的方式,内容的本地存储可以极大地减少远程传输的必要性,从而降低传输能耗。

2.2 任务迁移系统软件架构

随着移动互联网和物联网的发展,移动终端设备更加普及,用户对于移动终端的要求也越来越高,但是这些设备都存在了一些问题,比如:CPU 运算能力,内存大小以及能源等方面,这些问题都制约着用户在使用移动设备时的良好体验。移动边缘计算的出现对传统数据的计算和存储模式进行重大改进成为可能,移动边缘计算通过将大量计算和存储任务转移到靠近用户的网络边缘端,进而大幅降低了终端的能耗,解决了计算能力受限的问题。

任务迁移是指将终端设备应用程序的全部或者部分迁移到资源丰富的云端或者边缘端,借助它们强大的处理能力,加快计算速度,缩短应用程序的执行完成时间,为终端设备节约电池能量和花销^[39]。迁移的位置可以是云计算平台、MEC 平台或者其他终端设备等。这种任务的迁移方式可以缩短任务的执行时间、均衡系统能耗、获取更好的用户体验。不过在任务迁移的过程中也需要额外消耗一些资源,如传输功耗、流量等。当发生任务迁移的时候,该任务迁移时间的最小化也是实现满意的服务质量的关键因素,因此如何权衡负载任务在边缘节点和云端是否应该迁移,以及应该如何迁移都是这种边缘计算系统架构中需要解决的问题。

2.2.1 任务迁移系统软件架构

在进行任务迁移时,需要遵从一个统一的软件架构,任务迁移系统架构图如图 2-3 所示:

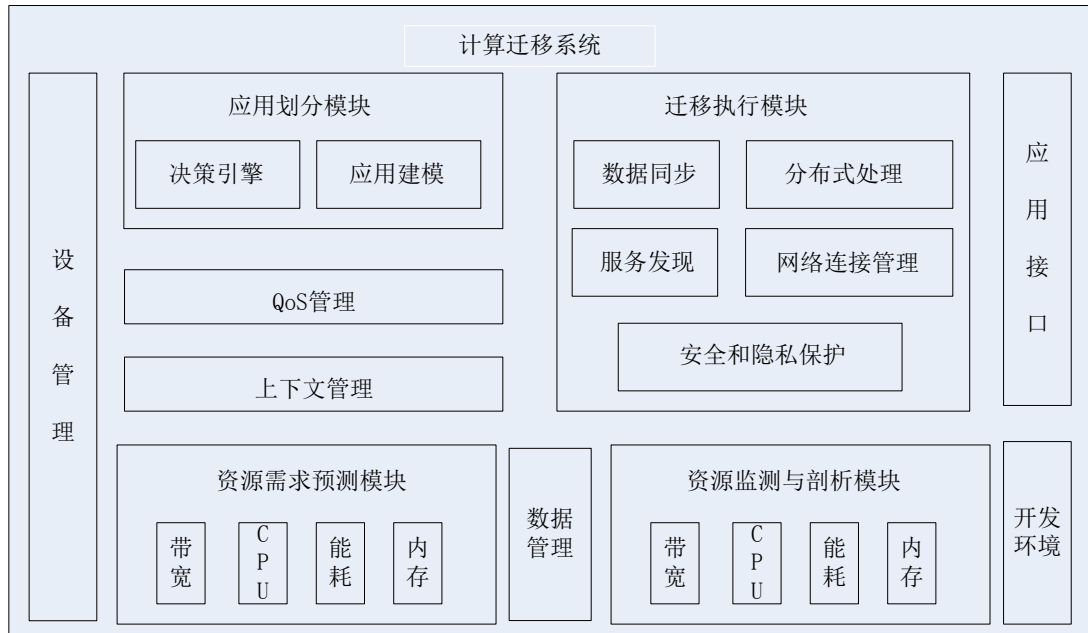


图 2-3 任务迁移系统软件架构图

Figure 2-3 Task migration system software architecture

由图 2-3 可以看出，一般将迁移系统架构大致分为以下 4 个关键模块：

a. 应用划分模块：任务迁移系统的核心模块，通过某种切分算法将一个整体的移动应用，划分为多个子任务，这些子任务有的是必须在本地执行，比如用户交互任务、设备 I/O 任务和外围设备接口任务等，还有一些是可在本地执行，也可进行迁移的任务，它们一般都是不需要与本地设备交互的任务，通常都是数据处理型任务，计算量较大。划分后的子任务彼此之间有数据交互，但是又能够独立执行，这是进行迁移决策的前提条件。

b. 资源需求预测模块：预测计算任务的资源需求，计算本地执行成本和迁移执行成本，对比成本预测信息，进行迁移决策。

c. 资源监测与剖析模块：主要对迁移平台可用资源量进行记录，比如服务器的状态和信息、虚拟机的数量和性能以及无线网络的信道条件等，当执行迁移后，可用资源量发生动态变化，需要及时更新记录。

d. 迁移执行模块：当移动客户端向远端服务器提交任务后，远端服务器会为该任务分配一个虚拟机，作为一个独立的应用来执行计算任务的执行，并将最终结果再返回给移动客户端。

2.2.2 任务迁移决策过程

移动边缘计算的任务迁移主要包括环境感知、任务划分、迁移决策、任务上传、边缘服务器执行、结果返回等六个步骤，其中任务划分、迁移决策是比较核心的两个步骤。任务迁移具体的流程图如图 2-4 所示：

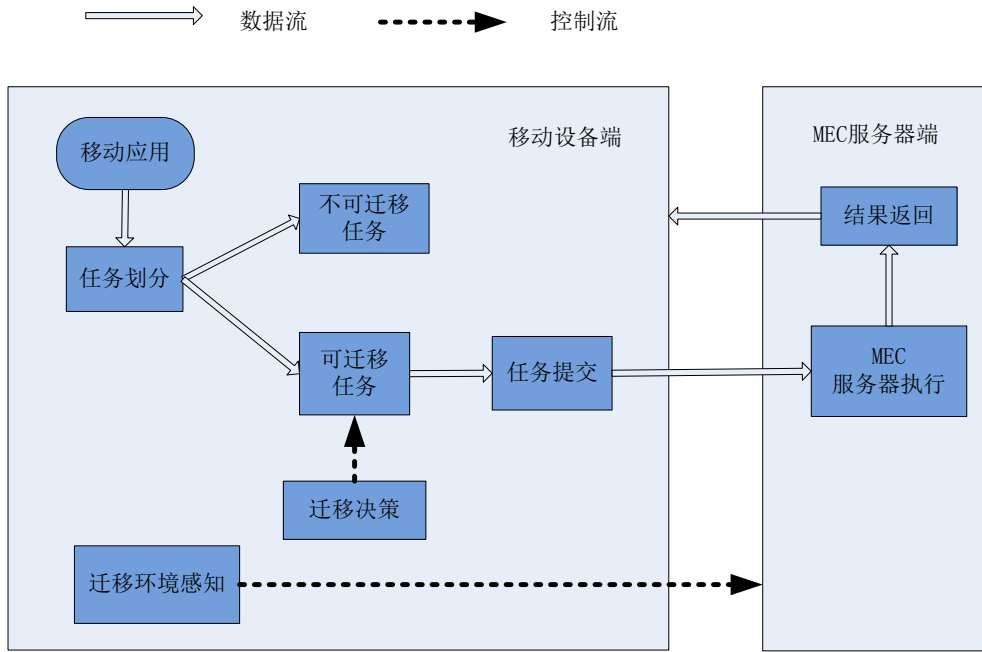


图 2-4 任务迁移流程图

Figure 2-4 Task migration flow chart

迁移决策是任务迁移过程中的重要步骤，它是指以某种指标为优化目标，通过相关的科学方法，衡量迁移效用，做出将哪些子任务迁移至远端服务器的过程。任务迁移通常分为基于方法层面迁移，基于任务层面的迁移，基于应用程序层面的迁移^{【40】}。通常研究的都是基于任务层面的迁移。

基于任务层面的迁移是指以任务为粒度，将应用程序代码划分为多个部分，并进行部分修改，以保证每个任务能够独立运行，移动设备以某种指标为优化目标，将监测到的带宽、移动设备以及远端服务器的cpu、内存等参数反馈至迁移决策模块，通过计算和衡量，做出是否需要卸载、将哪些子任务卸载的过程。一般的任务迁移具体步骤如下：

- 移动用户允许迁移，则迁移过程开始；反之，将应用程序在移动设备执行；
- 应用程序判断远端资源的可用性，检测与远端资源的有效连接性，若有可用资源且连接有效，则进行下一步；反之，将应用程序在移动设备执行；
- 分析移动应用程序内部依赖关系，进行任务划分；
- 执行迁移决策算法，做出将哪些任务进行迁移，哪些任务在本地执行的决策；
- 根据决策结果执行任务，并将最后结果返回给移动用户。

任务迁移的决策是多方面因素共同作用的结果，其中包括用户的期望目标、移动设备的性能、远端服务器的性能、应用程序的结构以及网络状况等^{【39, 40】}。影响任务迁移决策的主要因素如图 2-5 所示：



图 2-5 影响迁移决策过程主要因素

Figure 2-5 Main factors affecting the decision process of migration

从图 2-5 可以看出，影响迁移决策过程的主要因素有以下 5 点：

a. 用户因素

用户的意愿是关系到迁移决策的因素之一，受到应用程序执行时间，获取网络资源和远端资源的金钱花销等因素的影响。其次，还与用户所期望的执行完成时间、节约电池能耗等因素有关。

b. 网络连接因素

不同种类的通信技术具有各自的特性，比如，基于 Wi-Fi 的网络连接具有高带宽和低延迟的特点。相比于 Wi-Fi，3G 带宽较低，网络延迟较大。当两者均可用的时候，用户更加倾向于使用 Wi-Fi，但在特殊情况下，例如终端设备处于连续移动的情景时，Wi-Fi 的连接并不可靠，只能选择 3G/4G。因此，在网络方面，影响迁移决策的主要因素是：网络延时、带宽和花销等。

c. 移动设备因素

移动设备的处理器、内存和存储等方面近年来都得到了显著的提升。因此，对于持有高性能移动终端设备的用户而言，自身终端设备的性能较高，任务迁移对其作用细微，必要性不是很大，但是对于所持设备性能较差的用户而言，任务迁移可以提高执行速率，减轻设备负载。

d. 服务器因素

任务迁移的前提就是远端服务器上具有对所上传的应用程序运行的环境，而且远端服务器资源必须足够充足以满足移动终端任务的需求，如果移动终端设备与远端服务器中部署的虚拟机在内存和 CPU 处理速度等方面性能相差无几，即使任务迁移可以节省电池能耗，但是由于网络延时、数据传输时延等因素都可能导致最终做出不迁移的决策。

e. 应用程序因素

应用程序的自身特点会影响迁移策略，一般认为，应用程序的计算复杂度越小，数据传输量越大，就更偏向于在本地执行；应用程序的计算复杂度越大，数据传输量越小，就更偏向于在远端执行。

2.3 本文研究的整体思路

在上述的理论基础上，归纳总结出现有迁移策略的流程并分析出其所存在的问题。原始的任务迁移策略是以整个移动终端应用为单位进行迁移决策的，这种粗粒度的迁移流程图如图 2-6 所示：

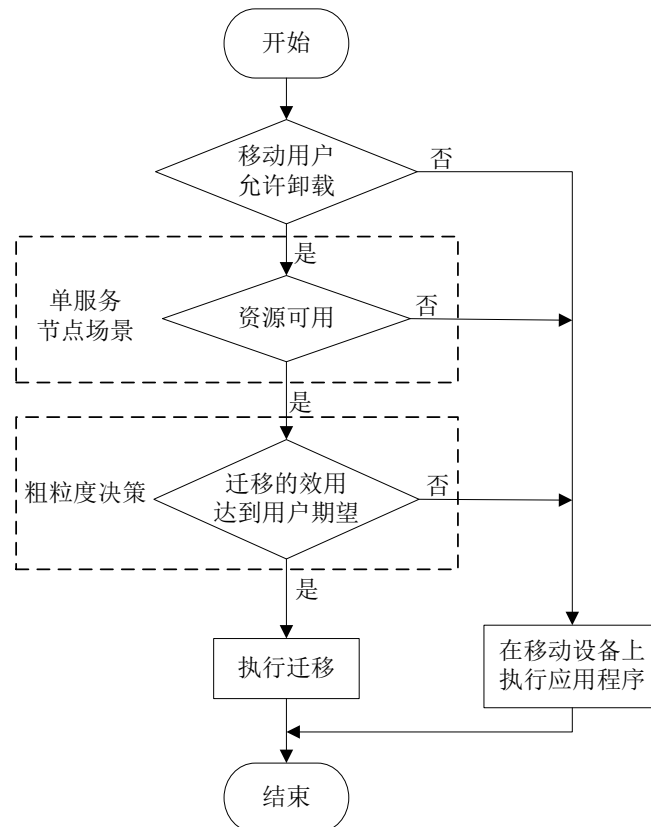


图 2-6 粗粒度迁移流程图

Figure 2-6 Coarse grained migration flow chart

在移动边缘计算环境下，用户与 MEC 平台的距离比较近，数据传输距离相较于云平台较短，传输速率提高，传输延时大大降低，因此，任务迁移在通信上的开销会变得很小，移动终端设备倾向于数据传输更为频繁的细粒度迁移方式来提高性能，降低开销。现有改进后的任务迁移策略对移动终端应用进行细粒度链式划分，并进行迁移决策，以

子任务为单位进行迁移，这种细粒度链式任务划分迁移流程图如图 2-7 所示：

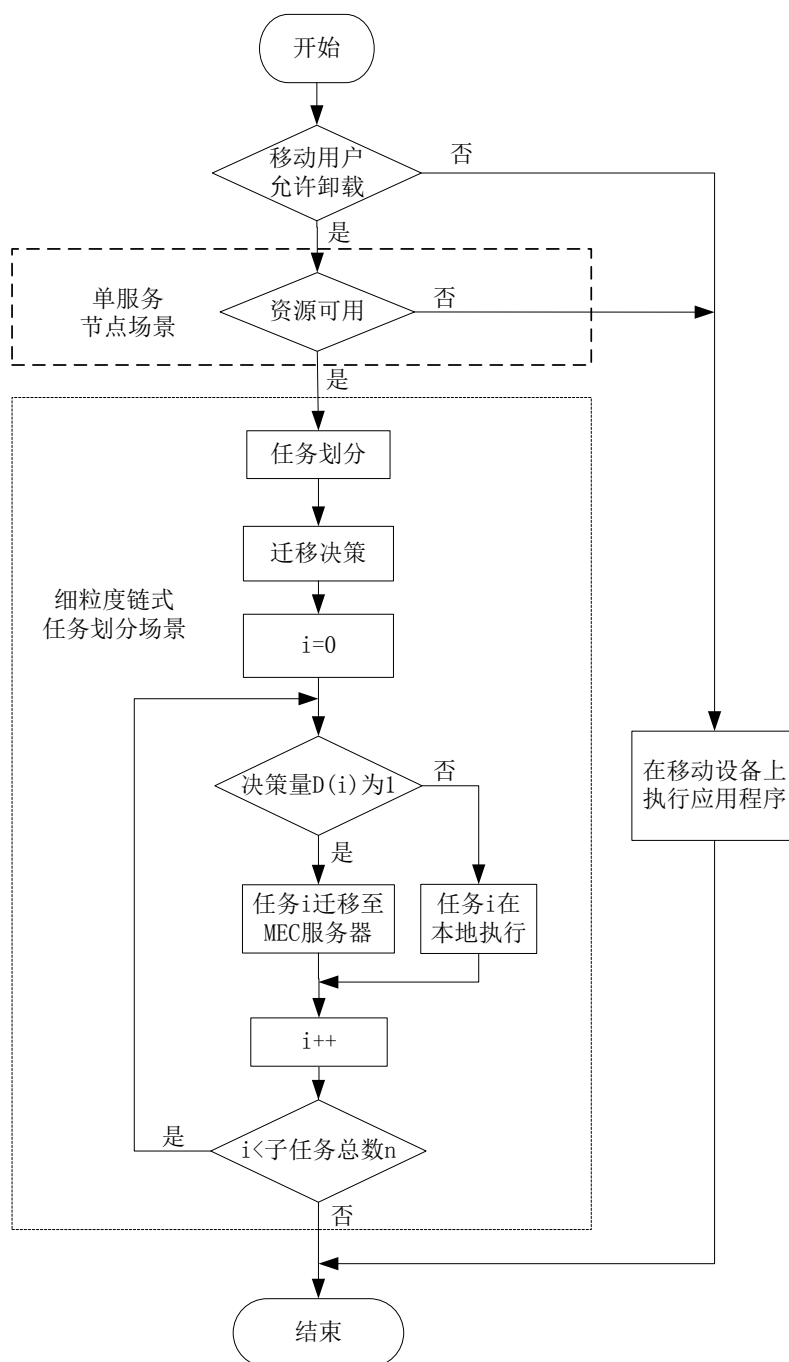


图 2-7 细粒度链式划分迁移流程图

Figure 2-7 Fine-grained chained mode migration flow chart

但是上述迁移策略均是在服务节点已经确定的前提下进行迁移决策，未考虑 MEC 平台中存在多服务节点可选的场景，而且细粒度链式任务划分模型未考虑移动终端应用内部存在复杂的多依赖关系，即一个子任务的输入数据来源于多个前置子任务的情况。

因此，本文旨在针对多服务节点场景提出任务迁移目的地迁移策略，以选取与移动终端应用最匹配的服务节点，在此基础上，针对细粒度有向无环图状任务划分场景，提出最优化节能迁移策略。本文的主要研究流程如图 2-8 多服务节点场景有向无环图状划

分任务迁移流程图所示：

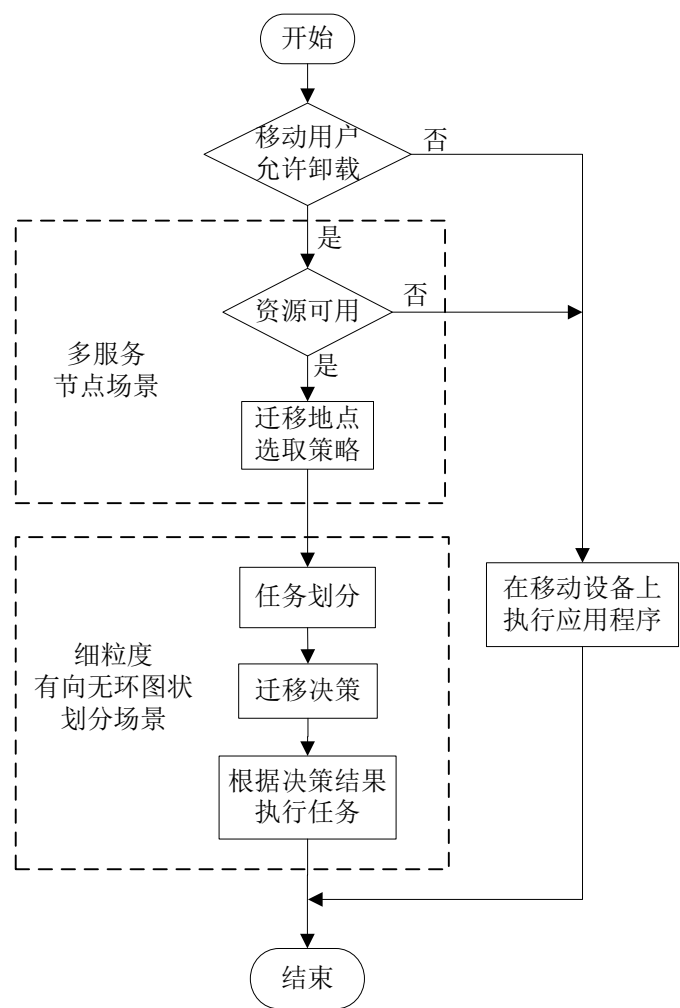


图 2-8 多服务节点场景有向无环图状划分任务迁移流程图

Figure 2-8 Fine-grained directed acyclic graph mode migration in multi-service nodes scene flow chart

课题整体研究路线图如图 2-9 所示：

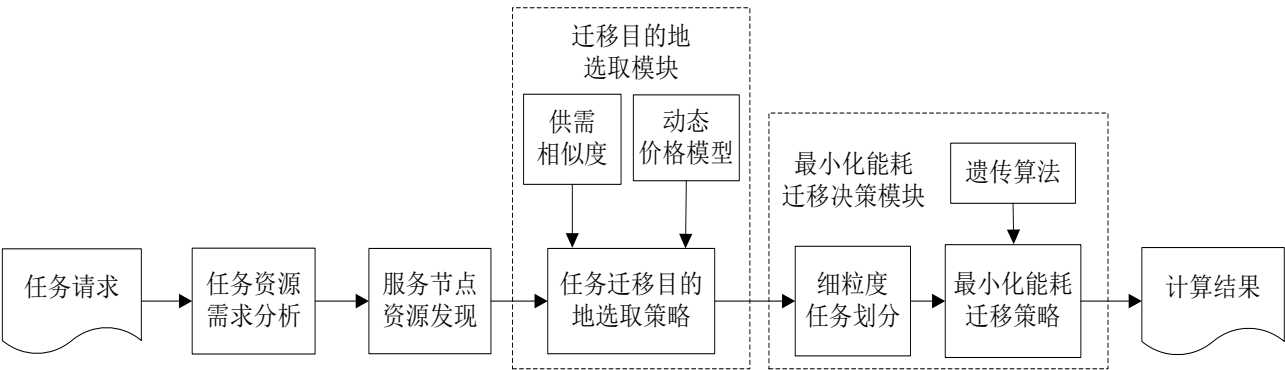


图 2-9 研究路线图

Figure 2-9 Research roadmap

由图 2-9 可以看出，本文主要研究在于任务迁移目的地的选取和最小化能耗迁移决策两大模块，为了解决在实际应用场景中存在多个服务节点时的迁移目的地选取问题，

本文提出基于供需相似度和动态价格模型的迁移目的地选取策略，以保障移动边缘计算平台负载均衡，提高资源利用率，有效降低时延和用户花销；并且在此基础上，针对移动终端电量有限的问题，且应用程序内部具有多依赖关系的情况，提出细粒度有向无环图状任务划分模型，构建任务完成时间限制下的最小化能耗问题模型，利用遗传算法求解，得出针对每一个子任务的迁移决策结果，获取最优化迁移方案，达到节约能耗的目的。

2.4 本章小结

本章首先对移动边缘计算基本架构进行分析，详细介绍它的由来和发展状况，描述其框架和应用前景，罗列其特征和优势；接着，对任务迁移系统组成进行详细分析，列出任务迁移一般步骤，并对迁移决策这一重要模块进行详细介绍，罗列出对迁移决策的重要影响因素；最后，归纳总结现有迁移策略的流程并分析出其所存在的问题，并针对问题，提出本文整体研究流程与思路。

3 基于供需相似度与动态价格模型的迁移地点选取策略

由于用户所持的移动终端各项性能指标较差，无法满足某些资源富集性的移动应用的资源要求，必须将应用任务迁移到资源丰富的边缘计算平台中执行，并将结果返回给移动终端，以满足用户对移动终端应用的需求。在实际应用场景中，可能存在多个可供选择的远程计算实体，在进行任务迁移时，选取合适的迁移地点起着至关重要的作用，但是传统的任务迁移策略都是建立在迁移服务节点已经确定的基础上，并未考虑到移动边缘计算平台中存在多个服务节点的情景，因此，如何综合用户所在意的多种因素，选取最优迁移地点，尽可能降低时延，降低成本，提高资源利用率，并且达到负载均衡的要求，即第二章研究总体思路中的迁移目的地选取模块，是本章主要研究内容。

3.1 供需相似度与动态价格模型分析

3.1.1 任务迁移场景

实际任务迁移场景图如图 3-1 所示：

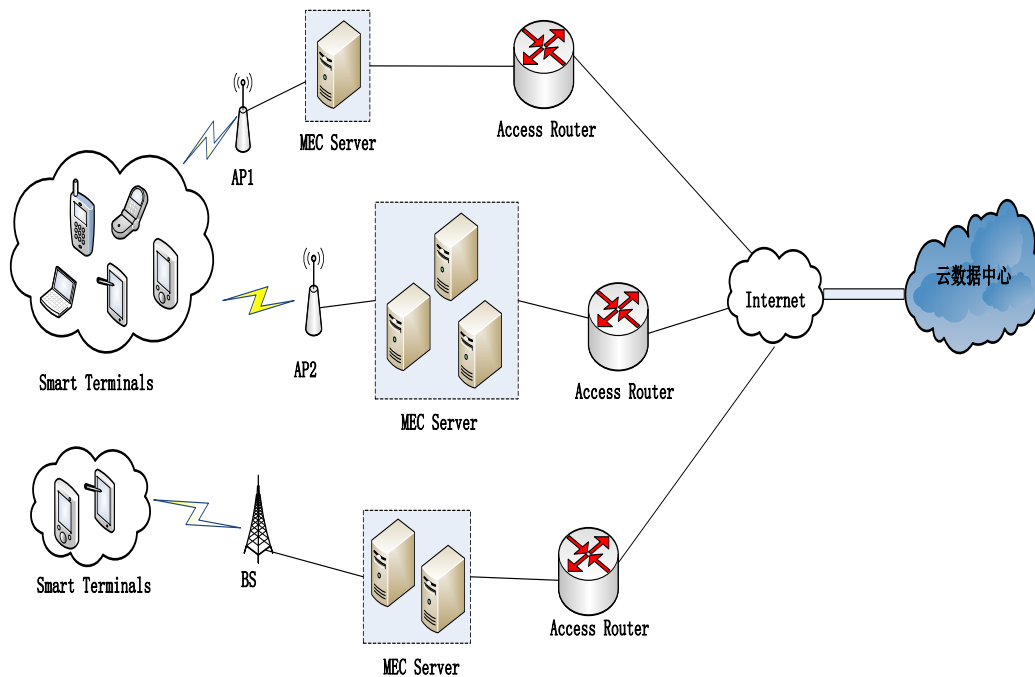


图 3-1 任务迁移场景图

Figure 3-1 Task migration scene graph

为了便于描述，先给出 2 个定义：

定义 1 迁移目的地：给定一些 MEC 服务器，通过对一些参数的评估，综合之后，为这些服务器赋予迁移权重，最终选取权重最大的 MEC 服务器作为任务执行地点。

定义 2 迁移权重：从资源利用率和成本花销等方面进行评估，综合这些参数后，为每个可用的 MEC 服务器赋予权值，认为权重最高的 MEC 服务器是迁移目的地。

在图 3-1 任务迁移场景图中，左上角的移动终端在两个无线接入点 AP1、AP2 的信

号覆盖范围内，左下角的移动终端在基站 BS 的信号覆盖范围内，它们之间的网速不尽相同；AP1 侧部署的 MEC 服务器只有 1 个，AP2 侧部署的 MEC 服务器有 3 个，BS 侧部署的 MEC 服务器有 2 个；若左上角的移动终端有任务迁移需求，应该选择 AP1 还是 AP2 接入网络，若选择具有 3 个 MEC 服务器的 AP2 接入网络，又将面临一个问题：3 个 MEC 服务器，哪一个作为任务处理载体？简言之，当确定要进行任务迁移时，任务迁移目的地的选取，是本章需要进行研究的内容。

在实际的资源查找即迁移环境感知过程中，一般都会有不止一处 MEC 服务器满足用户的迁移要求，能够为用户提供足够的资源来完成任务，因此，为了能够在较短的时间内完成任务，并且尽可能的减少资金花费，均衡资源利用率，需要从这些可选的服务节点中选择一个最合适的作为任务迁移目的地。

3.1.2 供需相似度思想

目前，移动边缘计算平台中的基础设施都已经被完全虚拟化，所有的任务都分配给虚拟机，并且将虚拟机部署到物理机上^[41]。可以把虚拟机看作是一个向量，它的每一维都代表一种类型的服务器资源，如内存、CPU 等，服务器可以看作是一个箱子，部署虚拟机可以看作是一个装箱问题。在传统的虚拟机调度策略中，总是将任务分配给性能最好的服务器中，并在此节点中创建虚拟机执行任务，但是这样会造成整个计算中心负载不均衡，并且容易导致基础设施的资源利用率不均衡，只要某一项资源不达标，都无法进一步部署虚拟机，造成极大的资源浪费。

先仅从内存和 CPU 两个维度简述资源利用率均衡问题，图 3-2 表示 1 台服务器上部署了 3 台虚拟机，每次部署虚拟机时，仅考虑任务的内存和 CPU 两种资源需求，从图 3-2 观察到当 3 台虚拟机均部署完成后，服务器中剩余了较多的内存资源，但是 CPU 资源却所剩无几，由于两种资源的利用率失衡，导致无法在服务器中进一步部署新的虚拟机，以至于内存资源被浪费。因此，在进行任务迁移选择迁移目的地时，应该尽可能的均衡资源利用率，避免造成资源浪费。

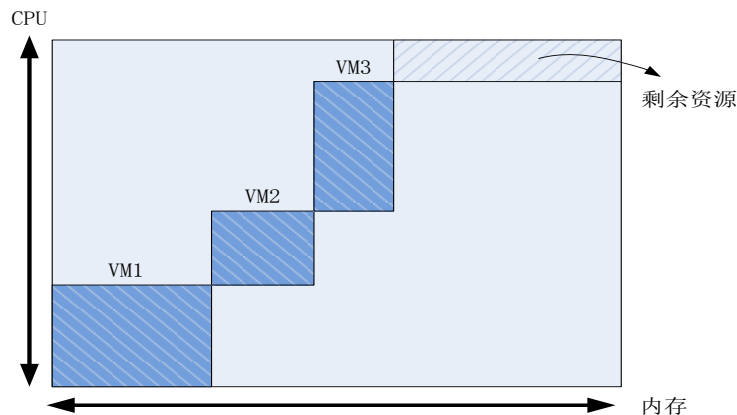


图 3-2 虚拟机部署图

Figure 3-2 Virtual machine deployment diagram

综上所述,在进行任务迁移目的地选取的过程中,需要充分考虑如何均衡服务器资源利用率,不能单纯只考虑基础设施的性能,每次都选择性能最好的边缘计算节点作为迁移目的地,而是应选择与任务最匹配的边缘计算节点作为迁移目的地。针对此问题,本文提出供需相似度的思想,根据任务资源需求量和 MEC 服务器的资源使用情况,评估若将任务部署到 MEC 服务器后的资源利用率均衡度的变化率,用其来代表任务与 MEC 服务器之间的相似程度,任务若部署到某 MEC 服务器后,资源利用率均衡度将大幅增加,则代表它们之间的供需相似程度越大,更适合作为任务迁移目的地;反之,代表它们之间的供需相似程度较小,不适合作为任务迁移目的地。

利用供需相似度的思想能够均衡各项资源利用率,增强利用率均衡度,能够极大的增加每台 MEC 服务器上的各类资源的利用率,减少资源浪费。

3.1.3 动态价格模型

仅仅根据供需相似度来选择任务迁移目的地,是不完善的,它未考虑到整个计算平台的多个 MEC 服务器的负载均衡需求,也未考虑到用户的金钱花销。移动边缘计算平台的资源价格与移动边缘计算服务提供商的定价模型有关,为了解决上述问题,采用基于资源剩余量的动态价格模型,动态价格模型是指根据某种指标,在不同的时间下,对价格进行动态调整,使得有限的供给能够匹配变化的需求,从而实现利益的最大化。在相同的场景下,在对于同样的移动终端任务,在不同的时间,迁移到不同的 MEC 服务器上,其成本指数均不相同。

在动态价格模型下,计算出各 MEC 服务器的成本指数,综合供需相似程度,最终选取成本指数较低且供需相似程度较高的 MEC 服务器作为最终的任务迁移目的地,完成任务并且返回结果给移动终端。在完成迁移任务的基本前提下,使边缘计算平台中的基础设施负载均衡,且提高了资源利用率,降低了用户成本,从而提高用户服务体验。

3.2 任务迁移地点选取算法

3.2.1 任务迁移系统框架

为了对算法进行更好的阐述,本文首先给出几个重要定义,重要定义如下:

定义 1 资源调度器:运行在移动边缘计算平台中的某个 MEC 服务器上,它可以根据移动终端用户的任务要求,以及移动边缘计算平台各服务器的运行状态,为移动终端用户任务分配相应 MEC 服务器并且创建虚拟机。

定义 2 状态监控进程:运行在每一个 MEC 服务器上,它能够实时监控移动边缘计算平台的基础设施状态信息,具体包括各 MEC 服务器的实时资源剩余量和实时资源价格等,将汇总后的状态信息报告给资源调度器,为迁移目的地的选取策略提供了基础。

定义 3 状态信息表:资源调度器中的一个存储模块,专门用于存放各状态监控进程收集到的各个 MEC 服务器的状态信息,主要包括资源价格,剩余资源量和时延等,其

中剩余资源量主要是指当前时刻的 CPU 资源，内存储器 and 带宽等。

MEC 平台的状态信息由运行在各 MEC 服务器中的状态监控进程实时监控，并且汇报给运行在某个 MEC 服务器中的资源调度器，资源调度器汇总各个状态监控进程所报告的信息，更新移动边缘计算平台状态表。三个定义之间的关系图如图 3-3 所示：

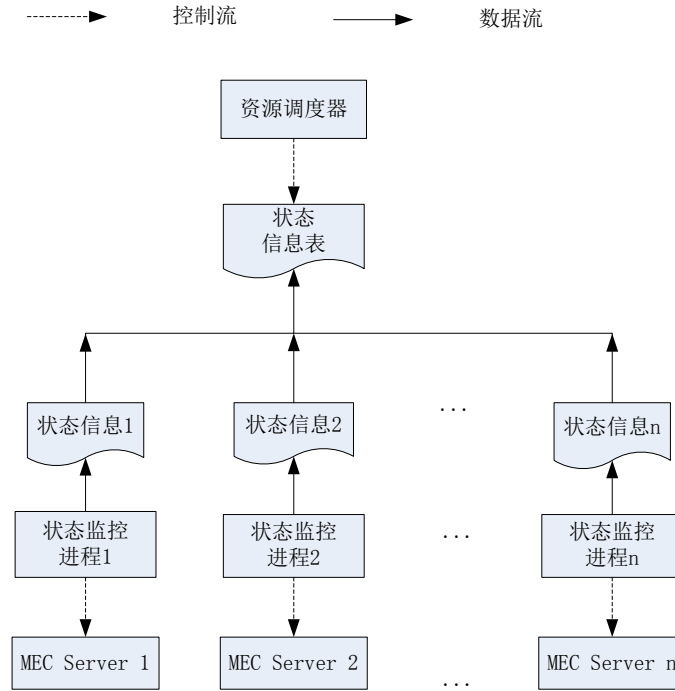


图 3-3 定义关系图

Figure 3-3 Definition diagram

具体的任务迁移系统架构图如图 3-4 所示，图 3-4 中标出任务迁移系统重要的部件和参数。迁移系统架构图 3-4 所示：

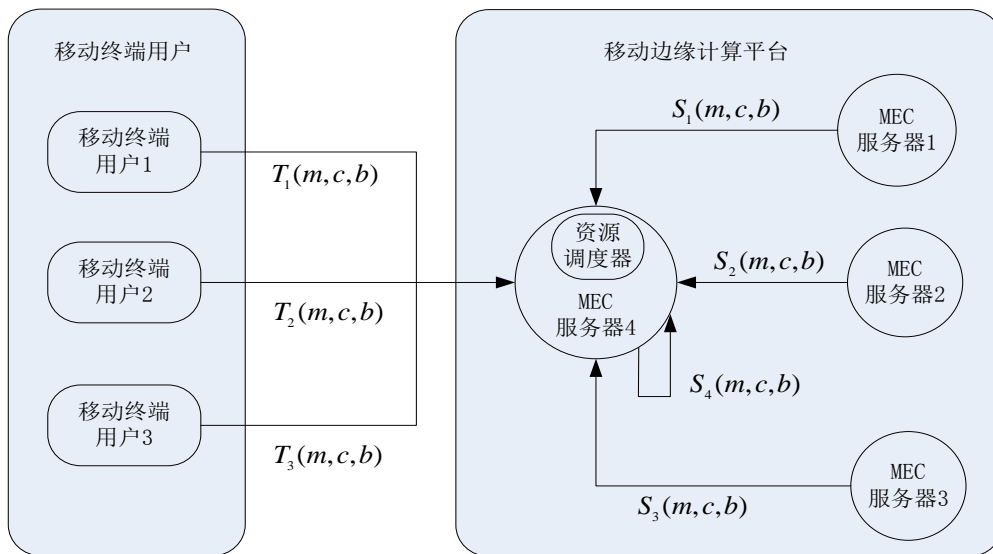


图 3-4 迁移系统架构图

Figure 3-4 Migration system architecture diagram

在图 3-4 中, $T_i(m,c,b)$ 表示移动终端用户向资源调度器提交的任务资源请求, 它分别包含了内存、CPU 和带宽要求, 其中, $T_i(m)$ 表示用户 i 的内存要求, $T_i(c)$ 表示用户 i 对 CPU 的要求, $T_i(b)$ 表示用户 i 对带宽的要求。 $S_i(m,c,b)$ 表示第 i 台移动边缘服务器的剩余资源量, $S_i(m)$ 表示其内存剩余量, $S_i(c)$ 表示其 CPU 剩余量, $S_i(b)$ 表示带宽剩余量。

3.2.2 任务迁移地点选取算法思想

基于供需相似度与动态价格模型的迁移地点选取算法思想如下:

当移动终端有任务迁移请求时, 首先, 选取出网络延时最小的接入点, 其次, 判断该接入点侧的 MEC 服务器数量是否为 1, 若为 1, 则选择该 MEC 服务器; 否则, 根据供需相似度与动态价格模型的思想, 评估若进行任务迁移前后 MEC 服务器的资源利用率均衡度的变化率, 用其代表任务所需资源和 MEC 服务器的可用资源的相似程度, 再为 MEC 服务器的资源赋予和剩余量相关的价格, 根据供需相似度和动态价格, 为各个 MEC 服务器赋予权重并排序, 最终选择权重最高的 MEC 服务器作为迁移目的地, 被选取的 MEC 服务器将用于执行该任务, 并将结果返回给移动终端。

基于供需相似度与动态价格模型的迁移地点选取算法流程图如图 3-5 所示:

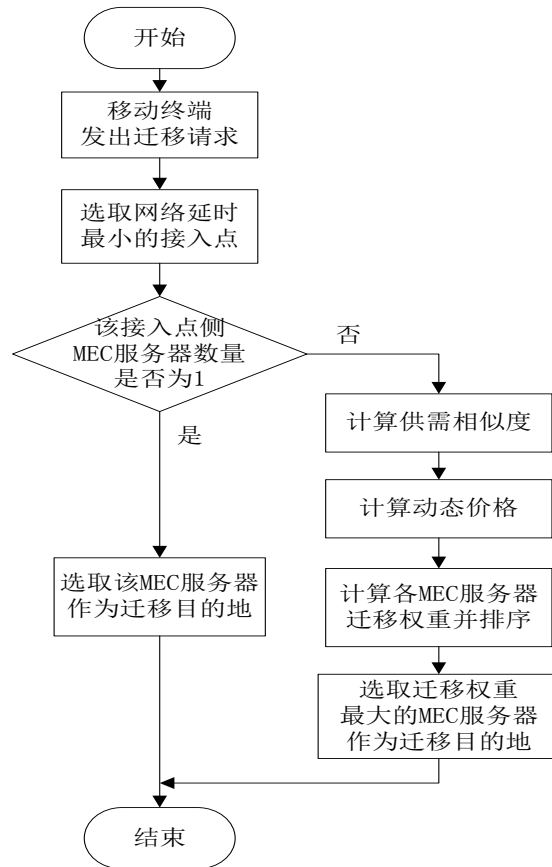


图 3-5 迁移目的地选取算法流程图

Figure 3-5 Migration destination selection algorithm flow chart

3.3 迁移地点选取算法实现

3.3.1 供需相似度的计算

利用资源利用率均衡度变化率来代表任务需求与服务节点能力的相似度，为了便于后文描述，首先给出资源利用率标准差 $RUSD$ （resource utilization standard deviation）、资源利用率均衡度 $RUBD$ （resource utilization balance degree）和资源利用率均衡度变化率 $CRRUBD$ （change rate of resource utilization balance degree）的定义。

定义 1 资源利用率标准差 $RUSD$ ：表示移动边缘计算服务器上的多维资源利用率的离散程度， $RUSD$ 值越小，代表各项资源利用率越均衡。第 i 台 MEC 服务器的资源利用率标准差计算公式如（3.1）所示：

$$RUSD_i = \sqrt{\frac{1}{K} \sum_{k=1}^K (u_i^k - u_i^{ave})^2} \quad (3.1)$$

其中， u_i^k 代表 MEC 服务器 i 上的第 k 维资源的利用率， $u_i^{ave} = \frac{1}{K} \sum_{k=1}^K u_i^k$ 表示 MEC 服务器 i 的资源利用率平均值。当 MEC 服务器上的某一种资源的利用率较高，而其他资源的利用率却十分低而且很均衡时，该 MEC 服务器的 $RUSD$ 值也是较低的，但是实际上却会由于该类资源剩余量较少，导致无法进一步部署虚拟机。因此，需要采用资源均衡率 RB （resource balance）来限制某一维资源利用率过高的情况，第 i 台 MEC 服务器的资源均衡度计算公式如（3.2）所示：

$$RB_i = \frac{u_i^{ave}}{\max_{k \in \{1, \dots, K\}} u_i^k} \quad (3.2)$$

在式（3.2）中， RB_i 的值是小于等于 1 的， u_i^{ave} 代表 MEC 服务器的多维资源利用率平均值， $\max_{k \in \{1, \dots, K\}} u_i^k$ 表示多维资源利用率中的最大值，当 RB_i 接近 1 时，代表限制某一种资源利用率过高的效果比较好。

定义 2 资源利用率均衡度 $RUBD$ ：综合考虑资源利用率标准差 $RUSD$ 和资源均衡率 RB ，第 i 台 MEC 服务器的资源利用率均衡度计算公式如（3.3）所示：

$$RUBD_i = \frac{RB_i}{RUSD_i} \quad (3.3)$$

定义 3 资源利用率均衡度变化率 $CRRUBD$ ：根据任务资源需求，评估若将任务分配到 MEC 服务器后的资源利用率均衡度 $RUBD^{after}$ ，第 i 台 MEC 服务器的资源利用率均衡度变化率计算公式如（3.4）所示：

$$CRRUBD_i = \frac{RUBD_i^{after} - RUBD_i}{RUBD_i} \quad (3.4)$$

当 $CRRUBD_i$ 为正数时,代表如果将任务分配到第 i 台 MEC 服务器上,会提高该 MEC 服务器的资源利用率均衡度,反之,则会降低该 MEC 服务器的资源利用率均衡度。更倾向于选择 $CRRUBD$ 值更大的 MEC 服务器作为任务迁移目的地。

计算供需相似度算法伪代码如表 3-1 所示:

表 3-1 供需相似度伪代码

Table 3-1 The calculation of similarity between MEC Server and task
Algorithm 1 calculation of similarity
Input : <i>Task configuration, MEC ServerList</i>
Output : <i>SimilarityList</i>
Steps:
$MECServerSize \leftarrow MECServerList.size()$
$i \leftarrow 0$
while $i < MECServerSize$ do
$RUSD.get(i) \leftarrow \sqrt{((u_i^m - u_i^{ave})^2 + (u_i^c - u_i^{ave})^2 + (u_i^b - u_i^{ave})^2) / 3}$
$RB.get(i) \leftarrow u_i^{ave} / \max_{k \in \{1, \dots, K\}} u_i^k$
$RUBD.get(i) \leftarrow RB.get(i) / RUSD.get(i)$
$RUBDAfter.get(i) \leftarrow \text{fun}(\text{Task configuration}, MEC \text{ ServerList})$
$CRRUBD.get(i) \leftarrow (RUBDAfter.get(i) - RUBD.get(i)) / RUBD.get(i)$
$Similarity.get(i) \leftarrow \text{transfer}(CRRUBD.get(i))$
$i++$
end while
return <i>SimilarityList</i>

3.3.2 动态价格的计算

在计算供需相似度的步骤中,尚未把用户花销考虑在内,因此,在此基础上,使用基于资源剩余量的动态价格模型,资源剩余量越少,资源价格越高,用户较倾向于选取单价较低的服务节点作为迁移目的地,从而保证移动边缘计算平台负载均衡,在降低用户花销的同时提高资源利用率。

公式 (3.5) 是基于资源剩余量的动态价格模型:

$$p_{n,t} = \lambda_1 p_{n,0} + \lambda_2 f(L_{n,t}) \quad (3.5)$$

在公式 (3.5) 中, n 代表移动边缘计算服务器的 ID, t 代表当前时间, λ_i 是权重因子, $\sum \lambda_i = 1$, $p_{n,t} = p(c_{n,t}, m_{n,t}, b_{n,t})$, $c_{n,t}$, $m_{n,t}$, $b_{n,t}$ 分别是边缘计算服务器在 t 时刻的处

理器资源，内存资源和带宽资源， $p_{n,0}$ 是基准资源价格，它是一个固定值， $f(L_{n,t})$ 是与资源剩余量有关的函数。

计算动态价格模型算法伪代码如表 3-2 所示：

表 3-2 动态价格模型伪代码

Table 3-2 The calculation of dynamic pricing model

Algorithm 2 dynamic pricing model

Input : *MEC ServerList*, *time*

Output : *PriceList*

Steps:

```

MECServerSize ← MECServerList.size()
i ← 0
while i < MECServerSize do
    MECServer ← MECServerList.get(i)
    PriceList.get(i) ←  $\lambda_1 * MECServer.getPrice()$ 
                     +  $\lambda_2 * f(MECServer, time)$ 
    i++
end while
return PriceList

```

3.3.3 迁移权重的计算

获取到任务和移动边缘服务器之间的供需相似度，再求出当前时刻下，各个移动边缘服务器的各项资源价格，使用公式（3.6）为网络延时最小的接入侧的各个移动边缘计算服务器赋予迁移权重。

$$W_{n,t} = \alpha * Similarity(n) + \frac{\beta}{P_{n,t}} \quad (3.6)$$

在公式（3.6）中， $W_{n,t}$ 代表第 n 个移动边缘计算服务器在 t 时刻的迁移权重， $Similarity(n)$ 代表该任务与第 n 个移动边缘服务器之间的供需相似度， $P_{n,t}$ 是代表了第 n 个移动边缘计算服务器在 t 时刻的价格。

当获取到迁移权重列表后，对其按照由大到小的规则进行排序，最终选取迁移权重最大的移动边缘计算服务器作为任务迁移目的地，对任务进行迁移处理，并将最终结果返回给移动终端。计算迁移权重的算法伪代码如表 3-3 所示：

表 3-3 计算迁移权重算法伪代码

Table 3-3 The calculation of migrate weight

Algorithm 3 calculation of weight
Input : <i>SimilarityList</i> , <i>PriceList</i>
Output : <i>WeightList</i>
Steps:
$MECServerSize \leftarrow SimilarityList.size()$
$i \leftarrow 0$
while $i < MECServerSize$ do
$WeightList.get(i) \leftarrow \alpha * Similarity.get(i) + \frac{\beta}{PriceList.get(i)}$
$i++$
end while
return <i>WeightList</i>

3.4 实验结果及分析

CloudAnalyst 是一款基于 CloudSim 的可视化模拟器,由 Buyya 云计算研究小组开发的云计算仿真软件,能够实现用户群和数据中心之间的最优调度^[41-45]。CloudAnalyst 具有简单易用的用户接口,也可图形化输出实验结果,便于进行实验和分析。

移动边缘计算就是将云计算的能力下沉到网络边缘,使“计算”更加接近用户,它与云计算内部运行原理较为相似,因此,可以通过对 CloudAnalyst 仿真平台代码进行修改,对其主要组件进行改进,能够将其改造成移动边缘计算仿真软件,在改进后的模拟器上进行仿真实验。

改进后的仿真平台主要组件如下:

- Region:** 该组件将该物理范围内划分为 6 个区域。User 和 AP 等组件都位于区域 Region 之中。
- Internet:** 网络属性模拟了真实世界的网络属性,其中主要包括网络数据传输延时和带宽等重要属性。
- User:** 该组件主要用于提交任务。
- AP:** 网络接入点,可以部署一个或多个 MEC 服务器以提供计算、存储能力。
- MEC Server:** 该组件用于处理 User 的任务请求,并且返回结果给 User。
- Application Service Broker:** 该组件主要实现把用户的任务迁移请求分配到某个具体的 MEC 服务器中,该组件是本章迁移目的地选取算法的承载体。
- GUI:** 该组件是实现模拟器的图形界面。

3.4.1 仿真场景与参数设置

本章的仿真场景如下：

在一定地理范围内，不同的网络接入点，信号覆盖范围不同，多个移动终端用户也是随机散布在各网络接入点的信号覆盖范围内。仿真场景中，有 4 个移动终端用户 User1、User2、User3 和 User4，它们分别在 2 号、1 号、3 号、0 号区域，有 4 个网络接入点 AP1、AP2、AP3、AP4，它们分别在 1 号、2 号、0 号和 3 号区域。共有 9 台 MEC 服务器，其中 MECS7、MECS8 和 MECS9 在接入点 AP3 侧，位于 0 号区域，MECS4 在接入点 AP1 侧，位于 1 号区域，MECS1、MECS2 和 MECS3 在接入点 AP2 侧，位于 2 号区域，MECS5 和 MECS6 在接入点 AP4 侧，位于 3 号区域。

网络接入点配置如表 3-4 所示：

表 3-4 网络接入点配置

Table 3-5 Access Point configuration

Name	AP1	AP2	AP3	AP4
Region	1	2	0	3

用户配置如表 3-5 所示：

表 3-5 用户配置

Table 3-4 User configuration

Name	User1	User2	User3	User4
Region	2	1	3	0
Request Per Hr	60	60	60	60
Data Size per Request (bytes)	100	100	100	100

时延矩阵配置如表 3-6 所示：

表 3-6 时延矩阵

Table 3-6 Delay Matrix(Units In Millisecond)

Region	0	1	2	3	4	5
0	25	100	150	250	250	100
1	100	25	250	500	350	200
2	150	250	25	150	150	200
3	250	500	150	25	500	500
4	250	350	150	500	25	500
5	200	200	200	500	500	25

移动边缘计算服务器配置如表 3-7 所示：

表 3-7 移动边缘计算服务器配置

Table 3-7 Mobile Edge Computing server configuration

Name	MECS1	MECS2	MECS3	MECS4	MECS5	MECS6	MECS7	MECS8	MECS9
AP	AP2	AP2	AP2	AP1	AP4	AP4	AP3	AP3	AP3
Region	2	2	2	1	3	3	0	0	0
Arch	X86	X86	X86	X86	X86	X86	X86	X86	X86
OS	Linux	Linux	Linux	Linux	Linux	Linux	Linux	Linux	Linux
VMM	Xen	Xen	Xen	Xen	Xen	Xen	Xen	Xen	Xen
Cost per VM \$/Hr	0.11	0.09	0.12	0.10	0.13	0.12	0.08	0.13	0.09
Memory Cost \$/s	0.04	0.05	0.06	0.05	0.06	0.04	0.05	0.05	0.07
Band Width Cost \$/s	0.10	0.09	0.11	0.08	0.09	0.10	0.12	0.11	0.10

带宽矩阵如表 3-8 所示：

表 3-8 带宽矩阵

Table 3-8 Bandwidth Matrix(Units In Mbps)

Region	0	1	2	3	4	5
0	20	10	10	10	10	10
1	10	8	10	10	10	10
2	10	10	25	10	10	10
3	10	10	10	15	10	10
4	10	10	10	10	5	10
5	10	10	10	10	10	20

3.4.2 仿真结果分析

为了分析基于供需相似度与动态价格模型的迁移地点选取策略（Selection Strategy of Migration Sites Based on Similarity and Dynamic Price Model，简称 MBOSADP）的性能，本章将另外两个迁移地点选取策略作为对比策略：一个是最快响应时间策略（Best Response，简称 BR），即选取上一轮任务处理完成时间最短的节点作为迁移目的地，另一个是随机选取策略（Random），即在 MEC 平台中随机选取任意一个服务节点作为迁移目的地。三种策略的任务完成时间对比图如图 3-6 所示，图 3-6 显示了在 MEC 环境中，三种策略在不同数量的应用任务参与迁移的情况下，各自的平均任务完成时间对比情况。任务完成时间是指任务从被提交开始，到该任务完成并且返回结果给用户的时间间隔。

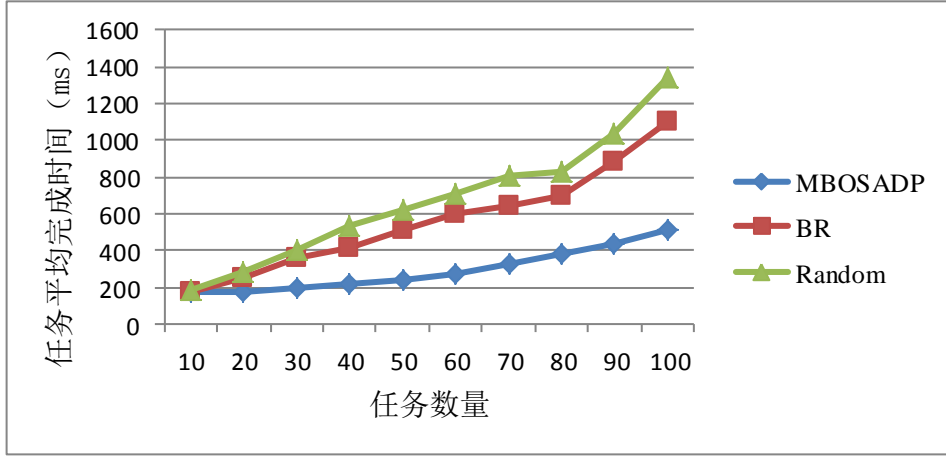


图 3-6 三种策略的任务完成时间对比图

Figure 3-6 Comparison of task completion time under three strategies

从图 3-6 可知，基于供需相似度和动态价格模型的迁移策略（MBOSADP）总是优于将任务随机迁移到任意一个计算资源上的策略（Random）和选取响应时间最快的计算资源的策略（Best Response），它的任务完成平均时间总是最小的。在迁移任务数量较少时，MBOSADP 策略在任务平均完成时间上的优势并不明显，这是因为一开始的可用资源较为充足，资源利用率相对均衡，可部署相应的虚拟机。随着参与迁移的任务数量的增加，MBOSADP 策略的平均任务完成时间远远低于其他两种策略，这是因为移动边缘计算的资源总量是固定的，MBOSADP 策略在进行迁移决策时，充分考虑了移动终端应用的资源需求与 MEC 服务器的资源剩余量，总是尽可能的将任务迁移到最合适的计算资源上执行，从而均衡 MEC 服务器的资源利用率，可以部署更多的虚拟机去执行任务。

图 3-7 是三种策略下的任务开销对比图，从图 3-7 中显示了三种算法各自的平均任务开销成本的对比情况。任务开销成本是指在移动边缘计算环境中，用户为完成一个任务的迁移所付出的实际调度开销费用。

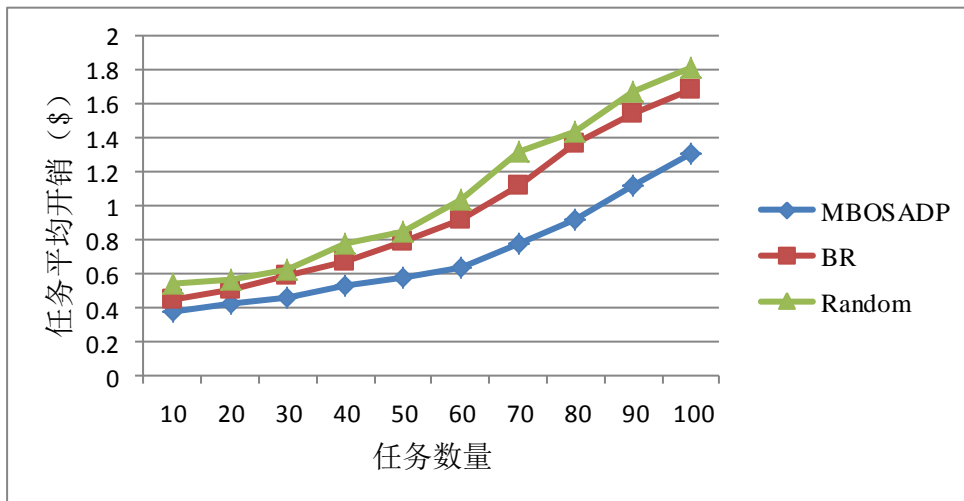


图 3-7 三种策略的任务开销对比图

Figure 3-7 Comparison of task costs under three strategies

从图 3-7 可知，随着任务数量的增加，任务之间的竞争激烈，任务等待时间较长，因此任务平均完成时间随之增加，而且资源单价也是动态变化增加的，任务平均开销也会随之增长。基于供需相似度和动态价格模型的迁移策略（MBOSADP）的平均任务迁移开销明显优于随机迁移策略（Random）和最快响应时间策略（Best Response），这是由于 MBOSADP 策略在进行迁移决策时兼顾了计算资源的利用率和价格，总是更倾向于将任务迁移到与其资源需求较匹配，价格相对较低的计算节点，这在一定程度上降低了应用任务的迁移开销成本。

3.5 本章小结

任务迁移目的地的选取是移动边缘计算任务迁移研究的一个重要方向，由于不同区域的不同任务的资源需求不同，如何降低任务时延和迁移成本，是一个研究热点。本文基于供需相似度和动态价格模型的任务迁移策略，充分考虑了任务的资源需求与移动边缘计算环境的计算资源的相似程度，提出供需相似度模型，保证 MEC 服务器的各项资源利用率的均衡度，同时，为了使整个移动边缘计算平台负载均衡，又提出基于资源剩余量的动态价格模型，最终将两个模型结合，为各 MEC 服务器赋予迁移权重，将任务迁移到迁移权重最大的 MEC 服务器中执行，并返回最终结果给终端。在改进的 CloudAnalyst 模拟器中的仿真结果表明，在相同的移动边缘计算环境中，在任务平均完成时间和平均花销等方面，基于供需相似度和动态价格模型的任务迁移目的地选取策略明显优于随机选取策略和最快响应时间策略。

4 基于遗传算法的任务迁移节能策略

传统的任务迁移策略中，一般是将整个移动终端应用作为迁移对象，或者将任务划分为链式线性关系的多个子任务后再进行迁移决策，均未考虑到移动终端应用内部具有多依赖关系的复杂情况；而且现有智能移动终端电池和节能技术的发展速度已经远远跟不上迅速增长的能耗需求。因此，依据第二章研究的总体思路框架下，在第三章已经选出最佳迁移目的地的基础上，本章将针对复杂拓扑关系的移动终端应用进行迁移决策，得出针对每一个子任务的迁移决策结果，即最优化迁移方案，在保证满足任务执行时间限制的前提下，有效减少移动终端设备的能耗。

本章首先分析现有的任务划分模型，并且根据移动边缘计算平台的近距离、低时延、能力相对有限的特性，建立细粒度有向无环图状任务划分模型，对划分后的每一个子任务及其之间的关系进行分析，提出相应的能耗和时间模型，构建执行时间限制下的最小化能耗问题，并利用遗传算法求最优解，得出针对每一个子任务的迁移决策结果，即该移动终端应用的最优化迁移方案。

4.1 任务划分模型

4.1.1 现有的任务划分模型

a. 粗粒度任务划分模型

传统的任务迁移策略中，往往将一整个移动终端应用作为一个迁移对象，并未再将其划分为多个子任务，这种粗粒度的迁移策略中，未充分利用移动边缘计算平台的低时延优势，也未考虑到移动边缘计算平台的各项能力均低于云计算中心的特点。

在上述粗粒度的任务迁移场景中，一般将整个任务迁移到移动边缘端的任务执行时间加上任务数据来回传输时间的总和记为 T_e ，移动终端的传输能耗和休眠能耗总和记为 E_e ，任务在移动终端的执行时间记做 T_l ，执行能耗记为 E_l ，将两者进行比较，若 $T_e < T_l$ 且 $E_e < E_l$ ，则将任务迁移到移动边缘端执行，否则，就将其放在移动终端侧执行。

b. 细粒度线性链式任务划分模型

有些改进的迁移策略中，假设一个移动应用是由一组子任务呈线性拓扑顺序组成，每一个子任务按顺序依次执行，每个子任务的输入数据都是由上一个子任务提供，它的输入数据是上一个子任务的输出数据，整个移动应用都有一个时间限制为 T 。图 4-1 是一个简单的细粒度线性任务拓扑图，认为一个移动应用由 n 个子任务线性顺序组成，第 k 个子任务的计算负载为 w_k ， $k=1,2,...,n$ ，因此，整个移动应用的计算负载为 $\sum_{k=1}^n w_k$ 。第 k 个子任务的输入和输出数据量分别是 α_k 和 β_k 。

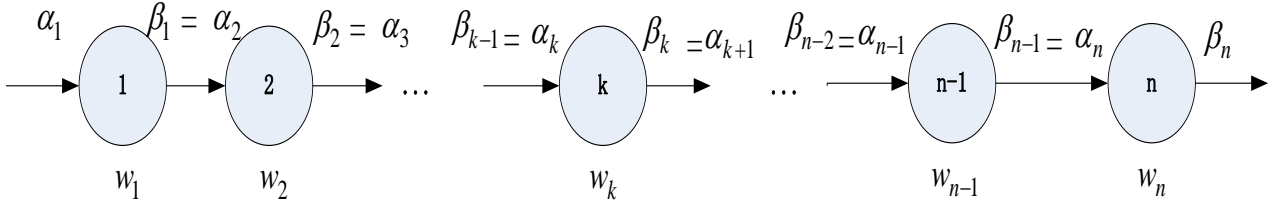


图 4-1 细粒度线性任务拓扑图

Figure 4-1 Fine-grained linear task topology

细粒度线性任务迁移场景的执行流程图如图 4-2 所示，它认为每一个子任务要么在移动终端侧执行，要么在移动边缘侧执行，将在移动终端侧执行任务时产生的计算能耗记为 $C(ME)$ ，将在移动边缘侧执行任务时产生的计算能耗记为 $C(EE)$ ；而且，当前一个子任务的执行地点和后一个的子任务的执行地点不同时，会产生传输能耗。当前者在移动终端执行，后者在移动边缘侧执行时，会产生发送时延和发送能耗，将发送能耗记为 $C(SID)$ ；当前者在移动边缘侧执行，后者在移动终端侧执行时，会产生接收时延和接收能耗，将接收能耗记为 $C(ROD)$ ；需要注意的是，最开始的任務数据存放在移动终端侧，并且当所有子任务完成后，需要将最终处理结果返回给移动终端。

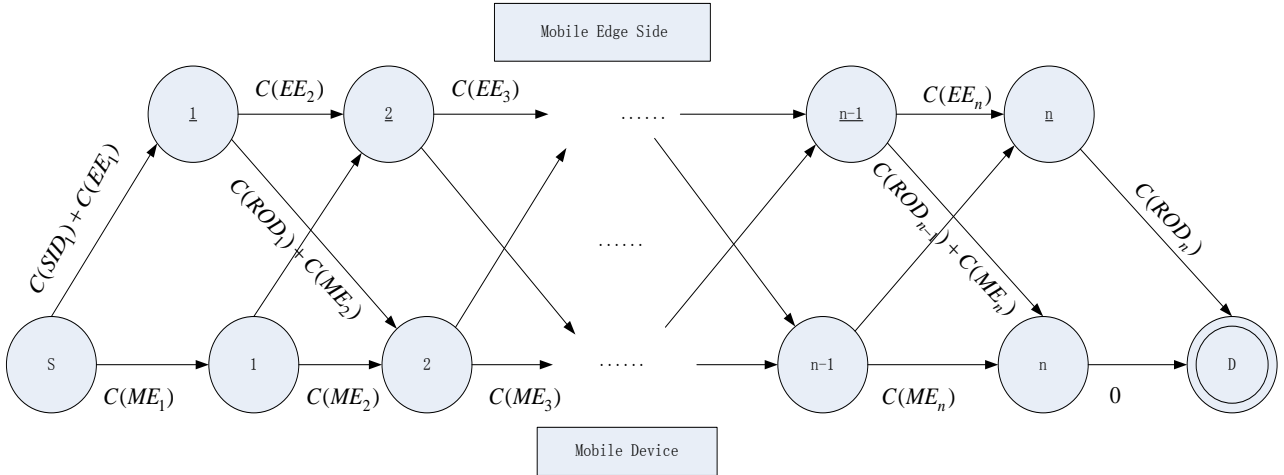


图 4-2 细粒度线状任务执行流程图

Figure 4-2 Fine-grained linear task execution flow

根据图 4-2 细粒度线性任务执行流程图，可以构建出在任务完成时间限制下的最小化能耗问题，在这种线性任务拓扑结构的情况下，可以将问题抽象成最短路径问题，使用迪杰斯特拉算法，求解最短路径，所得最短路径即最小能耗，若这条路径不能满足任务完成时间限制，则不断调整参数进行优化，最终得出的一组二进制位的解，代表各个子任务的执行地点。

但是以上 2 种任务划分模型都未考虑某个子任务的输入数据可能来自于多个子任务的输出，它必须在这些子任务都完成之后，才可以开始执行。用有向无环图来表示一个并行任务工作流，即将一个移动应用，划分为多个子任务，而且这些子任务之间互相有依赖关系，不是简单的链式单依赖关系，而是复杂的有向无环图式多依赖关系。

4.1.2 细粒度有向无环图状任务划分模型

如图 4-3 细粒度有向无环图状任务拓扑示例图所示，5 号子任务的输入数据源来自于 3 号子任务和 4 号子任务，只有当 3 号子任务和 4 号子任务均完成，5 号子任务收到来自双方的数据后，才可以开始执行。

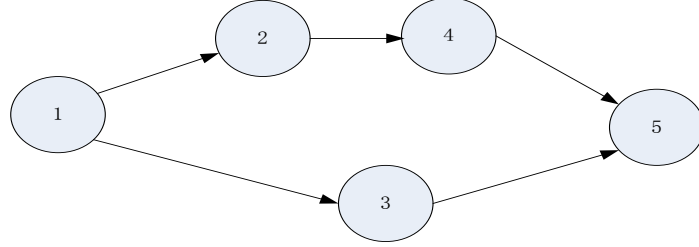


图 4-3 细粒度有向无环图式任务拓扑示例图

Figure 4-3 The sample of Fine-grained directed acyclic graph task topology

由图 4-3 示例图可知，移动终端应用的子任务间存在并行处理的情况，任务之间具有多依赖关系，传统的链式线性划分模型已经不适用于这种情况。

图 4-4 是细粒度有向无环图状任务拓扑模型图，用一个有向无环图（Directed acyclic graph, DAG）来表示子任务之间复杂的依赖关系，对于图 G ， $G = (V, E)$ ，其中节点集合 V 代表待处理的任务集合， $v \in V$ 代表划分出的子任务， E 代表有向边集合，表示任务之间的依赖关系， $e_{uv} \in E$ 表示任务 u 与任务 v 之间的转移数据量，它代表当任务 u 执行完毕后，会传输 e_{uv} 大小的数据给任务 v ，任务 v 只有在任务 u 执行完毕，且接收到任务 u 传输的数据后，才可以开始执行。

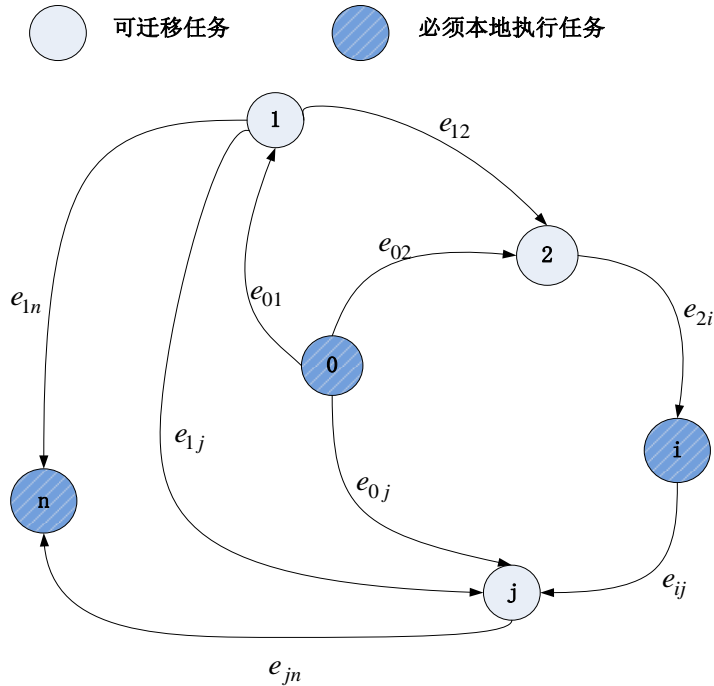


图 4-4 细粒度有向无环图状任务拓扑模型图

Figure 4-4 Fine-grained directed acyclic graph task topology model

由图 4-4 可知，移动终端应用进行划分后得到的子任务可以分为 2 类，一类为必须在本地执行的任务，比如任务 0，任务 i ，任务 n ，也就是图中的蓝色斜纹型任务，它们均属于不可迁移的任务集合 V_{local} ；另一类为可以迁移到移动边缘侧执行的任务，比如任务 1，任务 2，任务 j ，也就是图中的白色任务，它们均属于可迁移的任务集合 $V_{offload}$ ；但是它们不一定全部都必须被迁移到移动边缘侧执行，是否进行迁移是需要根据具体网络 and 任务数据量等因素而决定。

关于各个子任务之间的依赖关系，使用一个二进制变量 R (Rely) 来表示， $R \in \{0,1\}$ ，它的表达式如式 (4.1) 所示：

$$R_{uv} = \begin{cases} 1, & \text{任务 } u \text{ 是任务 } v \text{ 的前置任务} \\ 0, & \text{其他} \end{cases} \quad (4.1)$$

表达式 (4.1) 的含义是：若任务 u 是任务 v 的前置任务，则 $R_{uv} = 1$ ，代表两个任务之间有依赖关系，只有当任务 u 完成后，任务 v 才有可能开始执行，例如图 4-4 中， $R_{01} = 1$ ， $R_{02} = 1$ ， $R_{12} = 1$ ，而 $R_{in} = 0$ ， $R_{0i} = 0$ ；当某个任务有多个前置任务时，必须得等每一个前置任务都完成后，才能开始执行。例如图 4-4 中，任务 j 有 3 个前置任务，分别是任务 0，任务 1，任务 i ，唯有这 3 个任务均执行完毕，任务 j 接收到它们的执行结果之后，任务 j 才可以开始执行。

4.2 能耗和时间模型

任务迁移节能算法是为了在保证满足任务完成时间限制的前提下，尽可能的降低计算迁移能耗，主要是通过比较移动终端侧和 MEC 侧的时间，能耗，做出任务迁移的决策，因此，需要建立相应的任务能耗模型和时间模型。

为了方便后续描述，首先列出能耗和时间模型参数对应表，列出了本小节所使用到的符号名称及其含义，如表 4-1 所示：

表 4-1 能耗和时间模型参数对应表

Table 4-1 Parameter table for energy consumption and time model

名称	含义	单位
w_v	任务 v 的计算量	CPU 指令
f_l	移动终端的计算速率	MIPS
f_e	移动边缘计算服务器的计算速率	MIPS
p_l	移动终端执行任务时的功率	W
p_i	任务在 MEC 服务器执行，移动终端休眠时的功率	W
p_s	移动终端发送数据时的功率	W
p_r	移动终端接收数据时的功率	W
e_{uv}	任务 u 和任务 v 之间的数据传输量	bits
B_s	从移动终端往 MEC 服务器发送数据的速率	bits/s
B_r	移动终端从 MEC 服务器侧接收数据的速率	bits/s
T_v^l	任务 v 在移动终端执行所耗费的时间	s
T_v^e	任务 v 在 MEC 服务器执行所耗费的时间	s
T_{uv}	将任务 u 的处理结果发送给任务 v 所耗费的时间	s
E_v^l	任务 v 在移动终端执行所耗费的能量	J
E_v^e	任务 v 在 MEC 服务器执行所耗费的能量	J
E_{uv}	将任务 u 的处理结果发送给任务 v 所耗费的能量	J

用一个迁移决策量来表示各个子任务的执行地点，即使用二进制变量 D (Decision) 来表示， $D \in \{0,1\}$ ，它的表达式如式 (4.2) 所示：

$$D_v = \begin{cases} 1, & \text{任务 } v \text{ 迁移到 MEC 服务器上执行} \\ 0, & \text{任务 } v \text{ 在移动终端上执行} \end{cases} \quad (4.2)$$

表达式 (4.2) 的含义是：若任务 v 迁移到 MEC 服务器上执行，则 $D_v = 1$ ；若任务 v 在移动终端上执行，则 $D_v = 0$ 。注意：对于不可迁移的任务集合 V_{local} 里的所有任务都是 $D_v = 0$ ，即： $\forall v \in V_{local}, D_v = 0$ 。

4.2.1 任务执行时间和能耗模型

a. 移动终端执行时间

若任务 v 就在移动终端执行，即 $D_v = 0$ ，则任务的执行时间 T_v^l 与任务量大小 w_v 和移动终端的 CPU 计算速率 f_l 有关，任务的执行时间 T_v^l 的计算公式如式 (4.3) 所示：

$$T_v^l = \frac{w_v}{f_l} \quad (4.3)$$

b. MEC 服务器执行时间

若任务 v 迁移到 MEC 服务器执行, 即 $D_v = 1$, 则任务的执行时间 T_v^e 与任务量大小 w_v 和移动边缘计算平台侧的 CPU 计算速率 f_e 有关, 任务的执行时间 T_v^e 的计算公式如式 (4.4) 所示:

$$T_v^e = \frac{w_v}{f_e} \quad (4.4)$$

c. 移动终端执行能耗

若任务 v 就在移动终端执行, 即 $D_v = 0$, 此时移动终端处于工作状态, 则任务的本地执行能耗 E_v^l 与它的执行时间 T_v^l 和移动终端执行任务时的功率 p_l 有关, 任务的执行能耗 E_v^l 的计算公式如式 (4.5) 所示:

$$E_v^l = T_v^l p_l \quad (4.5)$$

d. 移动终端休眠能耗

若任务 v 就在移动边缘计算平台侧执行, 即 $D_v = 1$, 此时移动终端处于休眠状态, 移动终端休眠状态时的功率是 p_i , 则任务的执行能耗 E_v^e 计算公式如式 (4.6) 所示:

$$E_v^e = T_v^e p_i \quad (4.6)$$

4.2.2 数据传输时间和能耗模型

在任务 u 是任务 v 的前置任务的前提下, 即 $R_{uv} = 1$ 的前提下:

a. 若任务 u 和任务 v 的执行位置相同, 即 $R_{uv} = 1$ 且 $D_u = D_v$ 。两个任务之间不会产生数据传输, 因此 $T_{uv} = 0, E_{uv} = 0$ 。

b. 任务 u 和任务 v 的执行位置不同, $R_{uv} = 1$ 且 $D_u = 0, D_v = 1$ 。

即当任务 u 在移动终端执行, 任务 v 在 MEC 服务器上执行时, 会产生数据发送时延, 它与数据传输量 e_{uv} 和发送数据速率 B_s 相关, 任务之间的数据发送时延 T_{uv} 计算公式如式 (4.7) 所示:

$$T_{uv} = \frac{e_{uv}}{B_s} \quad (4.7)$$

发送数据能耗 E_{uv} 计算公式如式 (4.8) 所示:

$$E_{uv} = T_{uv} p_s \quad (4.8)$$

c. 若任务 u 和任务 v 的执行位置不同, $R_{uv} = 1$ 且 $D_u = 1, D_v = 0$ 。

即当任务 u 在 MEC 服务器上执行, 任务 v 在移动终端执行时, 会产生数据接收时延, 它与数据传输量 e_{uv} 和接收数据速率 B_r 相关, 任务之间的数据接收时延 T_{uv} 计算公式如式

(4.9) 所示:

$$T_{uv} = \frac{e_{uv}}{B_r} \quad (4.9)$$

接收数据能耗 E_{uv} 计算公式如式 (4.10) 所示:

$$E_{uv} = T_{uv} p_r \quad (4.10)$$

4.3 最小化能耗问题建模

任务迁移节能策略的目标是在迁移目的地确定的情况下, 将整个应用划分细化成更小粒度的子任务, 优化迁移方案, 尽可能的降低终端能耗, 达到为智能终端节能的目的, 但是与此同时, 任务完成时间还需要满足用户时延期望约束, 不能超出任务完成时间阈值。因此, 在构建最小化能耗问题模型时, 需要有时间约束条件, 确保在最小化能耗的同时, 能够保证任务完成时间不超时, 保证用户体验质量。

在本小节, 建立在任务完成时间限制下的最小化问题能耗模型。为了方便后续描述, 首先建立参数对应表, 列出了本小节所使用到的符号名称及其含义, 如表 4-2 所示:

表 4-2 最小化能耗问题模型参数对应表

Table 4-2 Parameter table for minimization of energy consumption model

名称	含义	单位
E_{total}	移动终端总能耗	W
T_{total}	任务完成总耗时	s
T_{max}	任务完成时间阈值	s
T_v^b	任务 v 的开始时刻	s
T_v^{ex}	任务 v 的执行时间	s
T_v^f	任务 v 的完成时刻	s
E_v^{ex}	任务 v 的执行能耗	W
R_{uv}	任务 u 和任务 v 的依赖关系	无
D_v	任务 v 的执行地点	无

任务 v 的执行时间 T_v^{ex} 与其执行地点 D_v 有关, 它的计算公式如 (4.11) 所示:

$$T_v^{ex} = (1 - D_v) T_v^l + D_v T_v^e \quad (4.11)$$

在式 (4.11) 中, 当 $D_v=0$ 时, 代表任务在移动终端上执行, 则前半部分的终端任务执行时间非 0, 后半部分 MEC 服务器任务执行时间为 0; 当 $D_v=1$ 时, 代表任务在移动边缘计算平台上执行, 则前半部分的终端任务执行时间为 0, 后半部分 MEC 服务器任务执行时间非 0;

任务 v 的开始执行时刻 T_v^b 取决于它的前置任务完成时刻和它们之间的数据传输时延

之和，若任务 v 有多个前置任务，则需要等到最后一个前置任务完成且所有数据传输完，才可以开始执行，任务 v 的开始执行时刻 T_v^b 的计算公式如（4.12）所示：

$$T_v^b = \max_{u \in V} R_{uv} (T_u^f + |D_u - D_v| T_{uv}) \quad (4.12)$$

在式（4.12）中， R_{uv} 代表只有任务 v 的前置任务才可以参与到它的开始执行时刻的计算； T_u^f 代表前置任务的完成时间；若任务 u 和任务 v 的执行地点不同，则会产生数据传输时延，则 $|D_u - D_v| T_{uv}$ 非0，否则，不会产生数据传输时延， $|D_u - D_v| T_{uv}$ 为0。从上式可以看出，任务 v 开始执行时间的计算是一个迭代过程，只有从初始任务开始计算才能得出结果。

求出任务 v 的开始执行时刻 T_v^b 和执行时间 T_v^{ex} 后，可以根据式（4.13）求出任务 v 的完成时刻：

$$T_v^f = T_v^b + T_v^{ex} \quad (4.13)$$

注意：任务完成时间阈值 T_{max} 是指所有子任务都在移动终端完成所需的时间，即 $D_v = 1, \forall v \in V$ 时的任务完成时间。

以上是对任务执行完成时间的分析，接下来分析任务完成能耗。

使用 $D(n)=[D_1, D_2, \dots, D_n]$ 来表示决策结果，即由 n 个子任务的执行地点决策量构成最小化能耗问题的解。

移动终端的能耗主要由任务执行能耗和数据传输能耗两部分构成。

任务 v 的执行能耗 E_v^{ex} 与其执行地点 D_v 有关，它的计算公式如（4.14）所示：

$$E_v^{ex} = (1 - D_v) E_v^l + D_v E_v^e \quad (4.14)$$

在式（4.14）中，当 $D_v=0$ 时，代表任务在移动终端上执行，则前半部分的终端任务执行能耗非0，后半部分MEC服务器执行任务，移动终端休眠能耗为0；当 $D_v=1$ 时，代表任务在移动边缘计算平台上执行，则前半部分的终端任务执行能耗为0，后半部分MEC服务器执行任务，移动终端休眠能耗非0。

因此，整个移动应用的能耗 E_{total} 的计算公式如（4.15）所示：

$$\begin{aligned} E(total) = E_{D(n)} &= \sum_{v \in V} [(1 - D_v) E_v^l + D_v E_v^e] + \sum_{u, v \in V} [|D_u - D_v| E_{uv}] \\ \text{subject to:} \\ D(n) &= [D_1, D_2, \dots, D_n] \\ D_v &\in \{0, 1\}, \forall v \in V_{offload} \\ D_v &= 0, \forall v \in V_{local} \end{aligned} \quad (4.15)$$

式（4.15）的后半部分，代表只有当 D_u 和 D_v 不同，即两个任务的执行地点不同且具有任务依赖关系时，才会产生传输能耗。

基于以上分析，本文提出任务完成时间限制下的最小化移动终端能耗问题模型，如式（4.16）所示：

$$\begin{aligned}
 \min E(total) &= \min_{D(n)} E \\
 \text{subject to:} \\
 T_{total} &\leq T_{max} \\
 D(n) &= [D_1, D_2, \dots, D_n] \\
 D_v &\in \{0, 1\}, \forall v \in V_{offload} \\
 D_v &= 0, \forall v \in V_{local}
 \end{aligned} \tag{4.16}$$

$D(n)=[D_1, D_2, \dots, D_n]$ 是该问题模型的解，它包含 n 个二进制变量，由于每一个可迁移任务的执行位置都有 2 种选择，若 $V_{offload}$ 包含 M 个元素，则迁移决策就有 2^M 组可能的解，且每一组解都必须通过迭代递推的方式，才能求出移动终端应用的完成时间 T_{total} 。若使用枚举法，求出最优解，计算复杂度过高，而且呈指数级增长，这并不是一个可行的方法。通过观察，发现迁移决策 $D(n)$ 是一组二进制变量构成的解，可以选择遗传算法，进行近似求解。这样做的好处是：即避免了枚举求解的高复杂度，又能保证解的准确性，

4.4 基于遗传算法的细粒度任务迁移节能策略

最小化能耗问题是一个非线性限制的 0-1 规划问题，对于该类问题，无法通过相应公式求解，但若使用枚举法求解，计算复杂度又过高，因此，需要使用启发式算法来求最优解，观察到迁移决策 $D(n)$ 是一组二进制变量构成的解，本文选择二进制位编码的遗传算法解决最小化能耗求解问题。

遗传算法是一种高效的全局搜索算法，在解决非线性、多峰、大空间、全局化等比较复杂的问题时具有一定的优势，它因模拟自然界中生物进化过程的“优胜劣汰”规律而出名，它主要思想是将目标优化问题的解编码成染色体，再通过模拟染色体的遗传进化过程来逐渐逼近获取最优解。

表 4-3 列出一些遗传算术语语：

表 4-3 算术语表

Table 4-3 Algorithm Glossary

生物遗传概念	遗传算法中的作用
个体	解
染色体	解的编码
基因	解中每一分量的特征（如分量的值）
适应性	适应函数值
种群	根据适应函数值选取的一组解
交叉	根据交叉原则产生一组新解的过程
变异	编码的某一个分量发生变化的过程

基于遗传算法的任务迁移策略流程图如图 4-5 所示：

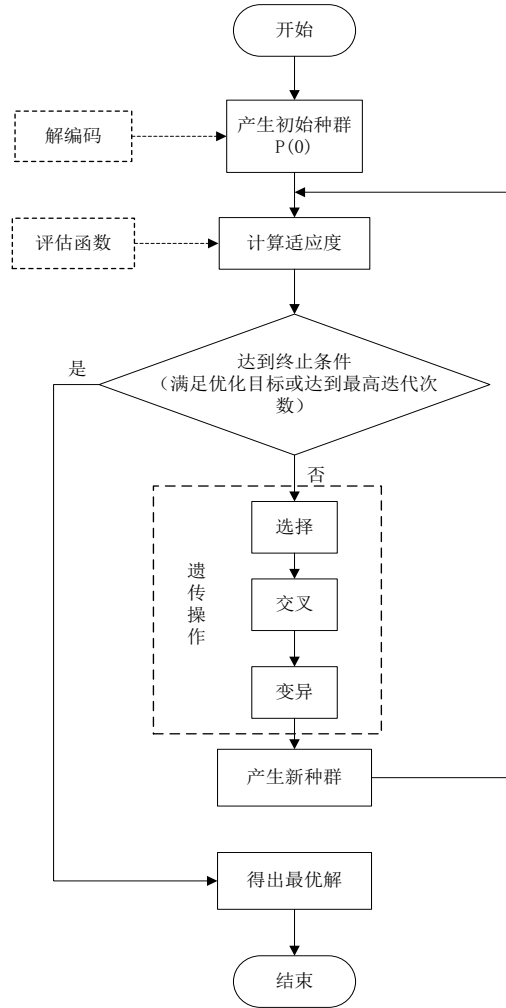


图 4-5 遗传算法流程图

Figure 4-5 Genetic algorithm flow chart

4.4.1 初始种群的选取和编码

利用遗传算法对任务的迁移策略进行全局求解时，需要对每一种迁移方案编码成一个相应的染色体，方便后期的交叉与编译等遗传操作。编码其实就是将实际问题的可行解从其解空间转换到遗传算法的搜索空间的过程，选择合适的编码是解决遗传算法问题的基础工作^{【46】}。

编码方式丰富多样，常见的有：浮点数编码、二进制编码、格雷码编码和排列编码等^{【47】}。由于本章的任务迁移最小化能耗问题模型中，迁移决策变量 $D(n)=[D_1, D_2, \dots, D_n]$ 是一个二进制的向量，它的每一个子元素都是一个二进制变量，因此，在最小化能耗迁移问题中，使用二进制编码，操作简单易行，也符合最小符号集的编码原则。指定种群规模 K ，将随机生成的相应数量可行任务迁移决策基于二进制染色体编码方法进行编码，即可得到初始的染色体种群 $P(0)$ 。

由于一个迁移决策变量 $D(n)$ 包含 n 个决策量，但是已知那些只能在本地产执行的子任务，无需进行决策，它们的决策量始终为 0，即 $D_v = 0, \forall v \in V_{local}$ 。假设在一个移动终端应用包含 n 个子任务，其中有 N 个子任务必须在本地产执行，即 $|V_{offload}| = N$ ，有 M 个子

任务可以进行迁移，即 $|V_{local}|=M$ ，且 $M+N=n$ 。因此，仅需对这 M 个子任务进行迁移决策即可，即在遗传算法中使用 M 个二进制位（基因）构造一条染色体，在得出最优解后，将其与 N 个值为0的二进制位组合起来，构成完成的迁移决策 $D(n)$ 。

图4-6组合编码求解示例图是一个划分为10个子任务的移动终端应用迁移卸载求解示例图，它有4个本地执行任务，6个可迁移任务，仅需对6个可迁移子任务的决策变量进行编码求解，得出最优解后，再与其他4个本地子任务的决策变量进行组合，得出最终的迁移决策。

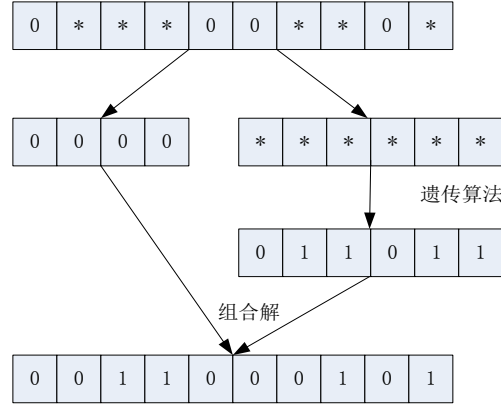


图 4-6 组合编码图

Figure 4-6 Combination of coding

4.4.2 适应度函数

适应度值是借鉴自然界种群个体对环境的适应程度，适应度函数是评价个体性能高低的标准，依靠函数值的大小区分每个个体的优劣，函数值越大，相应的染色体个体在遗传算法的进化迭代操作中，越有利于被保留下来，对应的迁移决策也就越好。在本章提出的任务迁移最小化能耗问题模型中，以移动终端应用总能耗的倒数作为评价相应染色体优劣的适应度函数，表示如式（4.17）所示：

$$\begin{aligned}
 fitness &= \frac{1}{E(total)} \\
 &= \frac{1}{\sum_{v \in V} [(1 - D_v) E_v^l + D_v E_v^e] + \sum_{u, v \in V} [|D_u - D_v| E_{uv}]}
 \end{aligned} \tag{4.17}$$

基于上述的适应度函数对染色体的个体进行优劣性评估，一条染色体个体的适应度函数值越大，则其所对应的任务迁移决策的能耗越小，这正是基于遗传算法的细粒度任务迁移策略的求解目标。依据个体的适应度值，计算每个个体在子代中出现的概率，并且依照此概率使用轮盘赌选择法，从父代中选取一部分个体，构成子代种群。

4.4.3 交叉方式的设计

交叉操作是模仿自然界中的繁殖和交配过程，通过将父代染色体两两优化组合，交

换部分基因，产生新的优良个体，使其优良品质保留至新的种群中。

常见的交叉方式有以下几种：

单点交叉：在父代染色体中随机选择一个交叉点，交换交叉点之前或之后的全部基因，以产生新的子代。单点交叉示意图如图 4-7 所示：

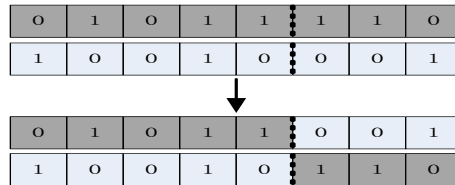


图 4-7 单点交叉

Figure 4-7 Single point crossing

双点交叉：在父代染色体中随机选择两个交叉点，交换两点之间的基因，以产生新的子代^[48]。双点交叉示意图如图 4-8 所示：

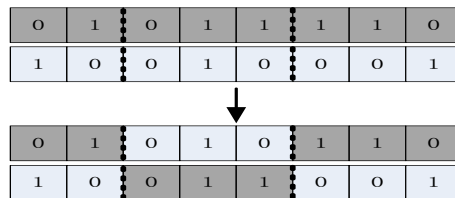


图 4-8 双点交叉

Figure 4-8 Two points crossing

均匀交叉：将父代染色体中的每一位基因都以相同的概率进行交换，以产生子代^[48]。均匀交叉示意图如图 4-9 所示：

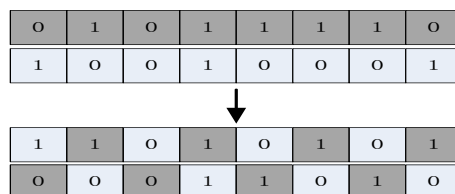


图 4-9 均匀交叉

Figure 4-9 Evenly crossing

一般的遗传算法中都是选取一种交叉方式来构造新的染色体，但是单一的交叉方式会造成个体结构单一，使种群陷入“早熟”，目前，很多研究者提出了自适应的遗传算法，主要是通过改变交叉的概率来实现。在本章中，为了增加种群多样性，且遗传个体的优良特性，采用的交叉方式是利用随机算法，等概率的选择其中任意一种交叉方式进行交叉，避免因为单一的交叉方式，造成种群的单一性。

4.4.4 变异方式的设计

所谓变异运算，是根据变异概率将染色体编码串里面的某些基因值用其他基因值来替换，从而形成一个新的个体，它是产生新个体的辅助方法，增强了遗传算法的局部搜

索能力，同时保持种群多样性，常用的几种变异方式有：基本位变异、均匀变异、边界变异和高斯近似变异等，在此使用适合于二进制编码的变异方式：基本位变异。

基本位变异示意图如图 4-10 所示：

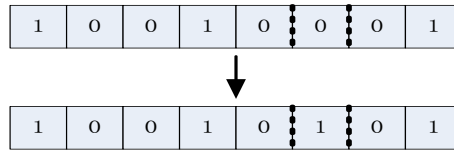


图 4-10 基本位交叉

Figure 4-10 Basic bit variation

4.4.5 遗传终止条件

经过上述遗传迭代进化的操作，可以得到新一代的染色体种群，根据提出的任务完成时间约束下的最小化能耗策略的求解目标，即公式 (4.16)，对得到的新一代染色体种群进行评估。如果遗传算法的迭代次数已经达到了预先设定的最大迭代次数，或者在连续的若干代染色体种群中，适应性最好的染色体个体的适应度函数值没有改进，则目标优化求解过程的终止条件得到满足，算法终止；否则，令迭代次数加 1，并继续重复上述遗传迭代进化操作过程，直至算法的终止条件得到满足。

4.5 实验结果及分析

4.5.1 仿真场景与参数设置

本章的仿真场景如下：

有 1 个移动终端设备 (User Equipment)，1 个基站，基站侧部署了 1 台 MEC 服务器，MEC 为移动终端用户提供任务迁移服务。仿真实验参数表如下表 4-4 所示：

表 4-4 仿真实验参数表

Table 4-4 Simulation parameters

参数	含义	参数值
f_l	UE 计算速度	300MIPS
f_e	MEC 服务器计算速度	5000MIPS
B_s	UE 发送速率	2Mbps
B_r	UE 接收速率	2Mbps
p_l	UE 运算功率	0.50w
p_i	UE 休眠功率	0.04w
p_s	UE 发送功率	0.03w
p_r	UE 接收功率	0.01w

关于移动终端应用的构建，每次采取随机方式为各个移动终端应用创建 20 个有关联

关系的子任务，并设定其中五分之一的任务仅限于本地执行，每个子任务的计算量和数据传输量都均匀分布。

设定初始染色体种群规模为 50，交叉概率为 0.6，变异概率为 0.15，最大种群迭代次数为 200，连续 5 代种群最小进化速率为 0.2%，即种群迭代进化最多次数为 200 次，或者连续 5 代种群中适应性最好的染色体个体的适应度函数值进化速度低于 0.2% 时，算法终止。

4.5.2 仿真结果分析

为了分析本章节的基于遗传算法的细粒度任务迁移算法性能（Fine-Grained Migration Based on GA，简称 FGMBGA），首先将其与粗粒度迁移策略(Coarse)、本地执行策略(Local)、全部迁移至 MEC 服务器执行策略(All Migration，简称 AllMig)进行能耗量对比分析，再将其与穷举法的结果进行对比分析。

以随机构造的方式，构造 15 个移动终端应用，各个移动终端应用的子任务不尽相同，以便更好的体现出迁移策略在不同场景下的适应性和优越性，计算在不同策略下的能耗值，画出四种策略下的移动终端能耗折线图如图 4-11 所示：

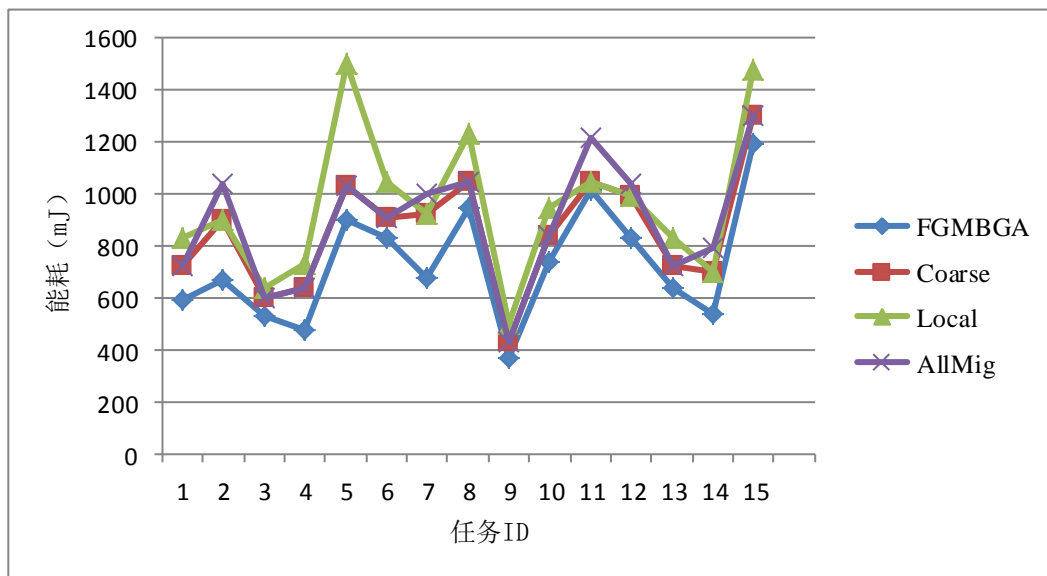


图 4-11 四种迁移策略能耗对比图

Figure 4-11 Comparison of task energy consumption under four strategies

图 4-11 展示了不同迁移策略下的移动终端能耗，图中的 FGMBGA、Coarse、Local 和 AllMig 分别代表了基于遗传算法的细粒度任务迁移策略、粗粒度迁移策略、本地执行策略和全部卸载至 MEC 服务器四种策略。从图 4-11 中可以观察到，粗粒度的任务迁移策略结果总是和本地执行策略或者全部卸载至 MEC 服务器策略中的结果一致，因为它就是以整个移动终端应用为对象，进行指标衡量分析后得出结果，在本地或者 MEC 服务器选一个较优的为任务执行地点。同时，还可以观察到，对于 15 个随机生成的移动终端应用，基于遗传算法的细粒度任务迁移策略总是比其他三种策略更加节能，尤其是对

于任务 5，基于遗传算法的细粒度任务迁移策略相对于本地执行策略可以节约近 40% 的能量。

图 4-12 迁移策略节约能耗对比图展示了基于遗传算法的细粒度任务迁移策略与其他三种策略相比的节约能耗百分比，图 4-12 显示，基于遗传算法的细粒度任务迁移策略与粗粒度的任务迁移策略相比，它最多能节约 26.093% 的能量，15 个应用平均节能 14.784%；与本地执行策略相比，它最多节约 40% 的能量，15 个应用平均节能 23.111%；与 AllMig 策略相比，它最多节约 35.387% 的能量，15 个应用平均节能 17.625%。

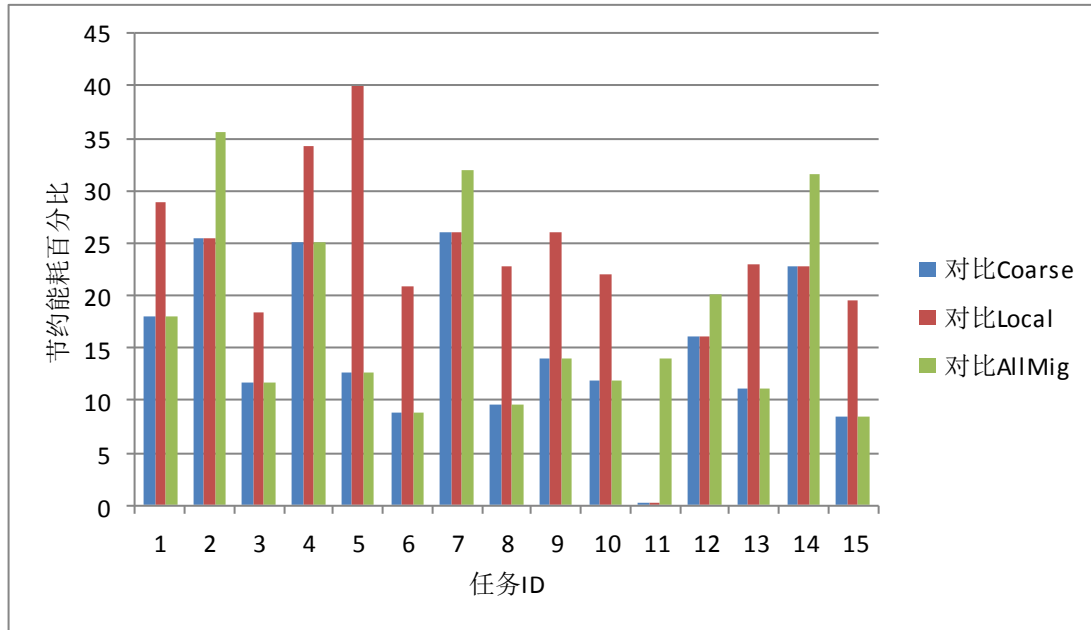


图 4-12 FGMBGA 与其他三种迁移策略节能对比图

Figure 4-12 The comparison of energy conservation between FGMBGA and the other three strategies

图 4-13 是对于基于遗传算法的细粒度任务迁移策略（FGMBGA）和穷举法(Enum)在能耗方面的对比图，可以观察到，Enum 和 FGMBGA 算法的折线图是基本重叠的，说明了 FGMBGA 策略虽然没有遍历整个解空间，但是最后得出的最优解与穷举法的效果相差无几，证明了本算法在降低时间复杂度的同时，也保障了准确性。

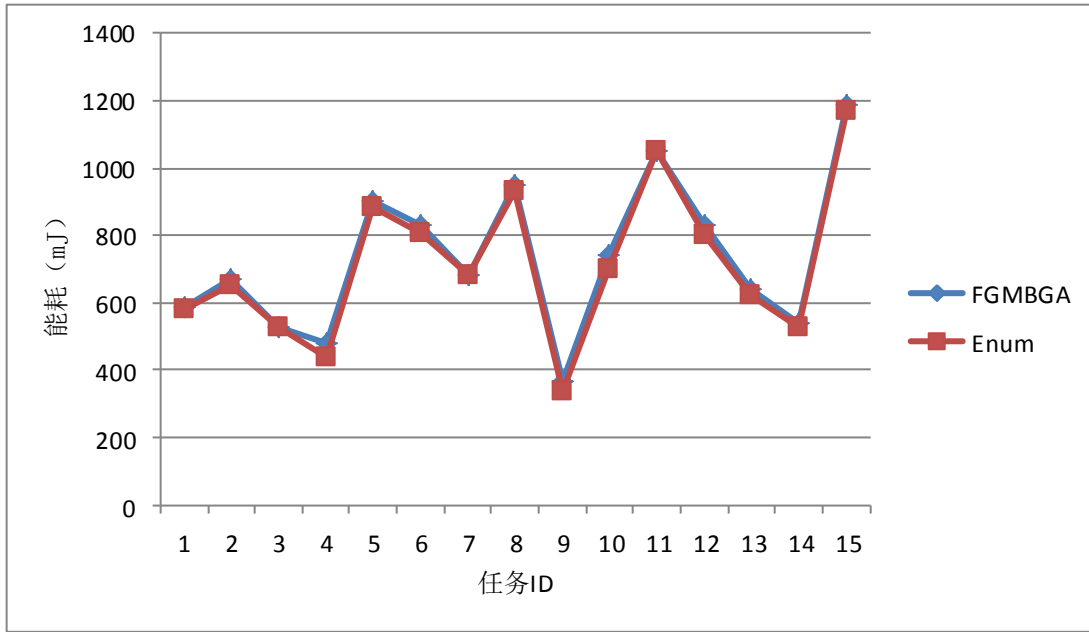


图 4-13 FGMBGA 与 Enum 能耗对比图

Figure 4-13 Comparison of task energy conservation between FGMBGA and Enum

4.6 本章小结

本章提出了基于遗传算法的细粒度任务迁移节能策略（FGMBGA），在任务迁移目的地已经确定的基础上，提出了在完成时间限制的最小化能耗模型并求解。本章首先分析了现有的任务划分模型，即粗粒度和细粒度线性划分模型，分析了它们各自的缺点，针对未考虑子任务之间的多依赖关系，而且不能充分的利用移动边缘计算低时延的特性问题，提出了更加符合实际情况的有向无环图状的细粒度任务划分模型，体现出子任务之间的依赖关系，即一个子任务的输入数据来自于多个子任务的输出结果；建立时间模型和能耗模型，提出时间限制下的最小化能耗问题模型，使用改进交叉方式的遗传算法，求解上述问题，得出针对每个子任务的迁移决策结果，获取最优化迁移方案，最终达到节约能耗的目的。仿真结果充分证明了 FGMBGA 策略的节能性，相较于其他三种策略，均不同程度的节约了能量，另外，还对比了 FGMBGA 策略与穷举法的迁移决策的能耗，仿真实验结果表明，两种策略所得最优解的耗能量误差在 4%之内，证明了所提 FGMBGA 算法的准确性。

5 总结与展望

5.1 研究总结

随着移动互联网和物联网的快速发展,人类即将迎来 5G 时代,5G 技术以“大容量、大带宽、低时延、低功耗”为诉求,而移动边缘计算正是 5G 网络提高用户体验的关键技术,它部署在网络边缘位置,靠近数据源,反馈迅速,能够将计算内容和能力下沉,使业务本地化。

移动边缘计算架构的提出,为移动终端用户提供了存储、计算和网络服务,为任务迁移提供了一个良好的平台。任务迁移就是将移动设备上计算密集型的应用程序的全部或者部分迁移到资源丰富的平台上,解决其计算能力不足、电量有限等问题。利用移动边缘计算近距离部署的特性,它可以为用户提供强大的计算能力和随时随地都能接入的超高速网络,使任务迁移的传输路径更短、能耗更小。

传统的任务迁移策略未考虑对多服务节点场景,也未考虑移动终端应用内部具有多依赖关系的场景,不能够充分发挥移动边缘的新特性的优势,本文针对移动边缘计算,从任务迁移策略的两个研究角度出发,即提出相应的任务迁移目的地选取策略和任务迁移优化节能策略,主要研究内容如下:

1) 任务迁移目的地选取策略,即在移动终端资源不足,确定必须将任务迁移至移动边缘计算平台中,才能够完成任务的情景下,解决如何在多个可选服务节点中,选取最优迁移地点的问题,以达到降低任务完成时间,降低任务迁移开销的目的。

对此,本文提出基于供需相似度与动态价格模型的迁移地点选取策略,摒弃传统的随机选取和最快响应时间思想,提出供需相似度的概念,考虑任务的资源需求量和移动边缘计算平台的计算节点资源量的相似程度,提高资源利用率均衡度,以保障更多的虚拟机能够被部署以执行任务;并且构建基于资源剩余量的动态价格模型,均衡资源利用率;最终选取供需相似度较高,价格较低的服务节点作为迁移目的地,有效的降低了任务时延和迁移成本。在改造的 CloudAnalyst 模拟平台下进行仿真实验,结果表明,在任务平均完成时间和平均花销方面,基于供需相似度和动态价格模型的迁移策略优于随机选取策略和最快响应时间策略,尤其在任务数量较多的情况下,它的优势更加明显。

2) 任务迁移优化节能策略,即在选取了合适的迁移目的地之后,从能耗、时间等多个指标进行权衡,做出针对于每一个子任务的迁移决策,得出最节能的迁移方案。

由于传统的任务迁移策略中都是将整个移动终端应用作为一个整体进行迁移决策,即迁移最小单位是一整个应用,而移动边缘计算的传输速率更高,适合于频繁的数据传递,可以将一个应用划分为多个子任务,不单单是普通的链式依赖关系,而且有向无环图式的依赖关系,即一个任务的输入数据来自于多个任务的输出结果,根据此思想,构建细粒度有向无环图状任务划分模型,并且构建其对应的任务能耗和时间模型,提出时间限制下的最小化能耗问题模型,使用改进了交叉方式的遗传算法进行求解。得出针对

每个子任务的迁移决策结果，获取最优化迁移方案，最终达到节约能耗的目的。

5.2 研究展望

本文所提出的任务迁移相关策略都是只考虑了当前环境状态，认为移动终端始终在信号覆盖范围内，且网络状况稳定，未考虑移动终端的移动性，忽略了移动边缘计算资源可用时长，增大了任务执行失败的可能性。

下一步研究工作中，可以考虑移动终端的移动性，对其的移动方向和轨迹进行预测，计算出移动终端与计算资源的连接时长，判断是否可以在其连接时间内完成任务的迁移后，再进行进一步的迁移决策。

本文在考虑细粒度任务迁移节能策略时，仅从任务特性出发，未考虑用户的偏好、信道的变化、电池剩余量等因素，未来研究工作中，应该考虑多方面因素，对迁移策略进行动态调整，以适应更多复杂的情况。

致谢

时光荏苒，岁月如梭，转眼间，在理工大的求学生涯即将结束，站在毕业的门槛上，回首往昔，百感交集，酸甜苦辣集结于心头，但此刻心中充盈更多的是感激。

感谢母校—西安理工大学，让我拥有了知识的海洋，见识世界的广阔。

感谢我的导师张毅坤教授，自进入研究室学习以来，您给予我耐心的指导、极大的关怀和热情的帮助，您严谨的治学态度、一丝不苟的学术精神和勤勉的工作作风时刻激励着我，是我学习的好榜样；您高尚的为师品德、博大包容的胸襟情怀以及对我的中肯教诲，使我受益终身。

其次，感谢张毅坤教授和王一川老师，谢谢你们在我的课题研究方向上给予指导和帮助，为我提供外出学术交流与学习的机会，参加学术论坛，开阔视野，了解课题前沿研究方向，谢谢你们们的敦促，使我积极进取，坚持不懈的做最好的自己。

感谢一路同甘共苦的同窗好友。刘航舟、任瑞国、刘航舟，与你们一起奋斗成长的日子，弥足珍贵。刘航舟积极乐观，吃苦耐劳，有踏实认真的学习态度；任瑞国沉着冷静，兴趣广泛，待人温和体贴；董俊华聪明帅气，思维敏捷，工作效率高。你们身上都有值得我学习的优点，祝你们前程似锦，生活美满，愿我们的友谊长存。

感谢教研室的师兄师姐们，李昊、王标帅师兄在我遇到编程难题时，总能帮我整理思路，找出解决方案，陈浩师兄在我科研遇到瓶颈时，悉心指导，使我醍醐灌顶，卢晓静、高科云、温晶师姐在我迷茫无助时，总能给我帮助和关怀，让我继续奋进；感谢朱赫、张丽师妹和王昌舒师弟，从你们身上我不仅学到了专业技术方面的知识，你们乐观向上的生活态度更是深深打动了我。

感谢我最亲爱的家人，你们是我漫长求学生涯中最强大的后援团，在我挫败、气馁的时候给予支持和鼓励，在我浮躁、焦虑的时候给予安慰和忠告，是你们培养了我坚强勇敢的性格，塑造了我正直善良的品性，祝愿你们身体健康，快乐永远。日后工作中，我也会继续严格要求自己，用自己的实际行动和努力报答你们。

最后，感谢百忙之中评阅论文和参加答辩的各位专家、教授。

参考文献

- [1] 钱志鸿, 王义君. 物联网技术与应用研究[J]. 电子学报, 2012, 40(5):1023-1029.
- [2] 刘强, 崔莉, 陈海明. 物联网关键技术与应用[J]. 计算机科学, 2010, 37(6):1-4.
- [3] Evans D. The internet of things how the next evolution of the internet is changing everything. CISCO White papers, 2011.
- [4] Shi W, Dustdar S. The Promise of Edge Computing[J]. Computer, 2016, 49(5):78-81.
- [5] 朱浩, 项菲. 5G 网络架构设计与标准化进展[J]. 电信科学, 2016, 32(4):126-132.
- [6] 王胡成, 徐晖, 程志密,等. 5G 网络技术研究现状和发展趋势[J]. 电信科学, 2015, 31(9):149-155.
- [7] 张元鑫. 5G 移动通信技术应用及其发展前景探索[J]. 通讯世界, 2017(22):7-8.
- [8] 窦笠, 孙震强, 李艳芬. 5G 愿景和需求[J]. 电信技术, 2013, 1(12):8-11.
- [9] 魏运锋, 刘庆东, 杨锐. 5G 标准及关键技术[J]. 电信工程技术与标准化, 2016, 29(12):55-60.
- [10] 戴晶, 陈丹, 范斌. 移动边缘计算促进 5G 发展的分析[J]. 邮电设计技术, 2016(7):4-8.
- [11] 田辉, 范绍帅, 吕昕晨,等. 面向 5G 需求的移动边缘计算[J]. 北京邮电大学学报, 2017, 40(2):1-10.
- [12] 付钰. 基于智能手机的能源消耗分析及对策[J]. 电脑知识与技术, 2016, 12(8):193-194.
- [13] Ahmed A, Ahmed E. A survey on mobile edge computing[C]// International Conference on Intelligent Systems and Control. IEEE, 2016.
- [14] 李华辉. 基于计算迁移移动设备的节能算法[J]. 电脑编程技巧与维护, 2015(10):18-18.
- [15] 李子姝, 谢人超, 孙礼,等. 移动边缘计算综述[J]. 电信科学, 2018(1):87-101.
- [16] Satyanarayanan M. The Emergence of Edge Computing[J]. Computer, 2017, 50(1):30-39.
- [17] Mao Y, You C, Zhang J, et al. A Survey on Mobile Edge Computing: The Communication Perspective[J]. IEEE Communications Surveys & Tutorials, 2017, PP(99):1-1.
- [18] 俞一帆, 任春明, 阮磊峰,等. 移动边缘计算技术发展浅析[J]. 电信网技术, 2016(11):59-62.
- [19] Jararweh Y, Tawalbeh L', Ababneh F, et al. Scalable Cloudlet-based Mobile Computing Model[C]// The, International Conference on Mobile Systems and Pervasive Computing. 2014:434-441.
- [20] 吴磊. 思科:“雾计算”的标语与干货[J]. IT 经理世界, 2014(11):68-69.
- [21] Bonomi F, Milito R, Zhu J, et al. Fog computing and its role in the internet of things[C]// Edition of the Mcc Workshop on Mobile Cloud Computing. ACM, 2012:13-16.
- [22] 黄海峰. 边缘计算产业联盟成立影响几何?[J]. 通信世界, 2016(33):52-53.
- [23] 李佐昭, 刘金旭. 移动边缘计算在车联网中的应用[J]. 现代电信科技, 2017(3):37-41.
- [24] Sabella D, Vaillant A, Kuure P, et al. Mobile-Edge Computing Architecture: The role of MEC in the Internet of Things[J]. IEEE Consumer Electronics Magazine, 2016, 5(4):84-91.
- [25] Huertacanepa G, Lee D. An Adaptable Application Offloading Scheme Based on Application Behavior[C]// International Conference on Advanced Information NETWORKING and Applications - Workshops. IEEE, 2008:387-392.

- [26] Yi S, Hao Z, Qin Z, et al. Fog Computing: Platform and Applications[C]// Third IEEE Workshop on Hot Topics in Web Systems and Technologies. IEEE Computer Society, 2015:73-78.
- [27] 李洪星. 移动边缘计算组网与应用研究[D]. 北京邮电大学, 2017.
- [28] Ha K, Chen Z, Hu W, et al. Towards wearable cognitive assistance[C]// International Conference on Mobile Systems. ACM, 2014:68-81.
- [29] 祁兵, 夏琰, 李彬,等. 基于边缘计算的家庭能源管理系统:架构、关键技术及实现方式[J]. 电力建设, 2018(3).
- [30] Cuervo E, Balasubramanian A, Cho D K, et al. MAUI:making smartphones last longer with code offload[C]// International Conference on Mobile Systems, Applications, and Services. DBLP, 2010:49-62.
- [31] Chae D, Kim J, Kim J, et al. CMcloud: Cloud Platform for Cost-Effective Offloading of Mobile Applications[C]// Ieee/acm International Symposium on Cluster, Cloud and Grid Computing. IEEE, 2014:434-444.
- [32] Lagerspetz E, Tarkoma S. Mobile search and the cloud: The benefits of offloading[C]// IEEE International Conference on Pervasive Computing and Communications Workshops. IEEE, 2011:117-122.
- [33] Zhang H, Zhang Q, Du X. Toward Vehicle-Assisted Cloud Computing for Smartphones[J]. IEEE Transactions on Vehicular Technology, 2015, 64(12):5610-5618.
- [34] Jia M, Cao J, Yang L. Heuristic offloading of concurrent tasks for computation-intensive applications in mobile cloud computing[C]// Computer Communications Workshops. IEEE, 2014:352-357.
- [35] Huang D, Wang P, Niyato D. A Dynamic Offloading Algorithm for Mobile Computing[J]. IEEE Transactions on Wireless Communications, 2012, 11(6):1991-1995.
- [36] Li T, Wu M, Zhao M. Consumption considered optimal scheme for task offloading in mobile edge computing[C]// International Conference on Telecommunications. IEEE, 2016:1-6.
- [37] Wang J, Peng J, Wei Y, et al. Adaptive application offloading decision and transmission scheduling for mobile cloud computing[J]. 中国通信(英文版), 2017, 14(3):169-181.
- [38] 张建敏, 谢伟良, 杨峰义,等. 移动边缘计算技术及其本地分流方案[J]. 电信科学, 2016, 32(7):132-139.
- [39] 汤海燕. 移动云计算中任务卸载及虚拟机调度策略的研究[D]. 吉林大学, 2017.
- [40] 郎为民, 安海燕, 姚晋芳. 移动云计算卸载问题研究[J]. 电信快报:网络与通信, 2016(12):3-7.
- [41] 朱亚会, 陈丹, 庄毅. 云数据中心资源利用率均衡的虚拟机调度算法[J]. 小型微型计算机系统, 2017, 38(2):232-237.
- [42] Wickremasinghe B, Calheiros R N, Buyya R. CloudAnalyst: A CloudSim-Based Visual Modeller for Analysing Cloud Computing Environments and Applications[C]// IEEE International Conference on Advanced Information NETWORKING and Applications. IEEE, 2010:446-452.

- [43] Calheiros R N, Ranjan R, Beloglazov A, et al. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms[J]. Software Practice & Experience, 2011, 41(1):23-50.
- [44] Patel S, Patel R, Patel H, et al. CloudAnalyst : A Survey of Load Balancing Policies[J]. International Journal of Computer Applications, 2015, 117(21):975-8887.
- [45] Sharma V, Rath R, Bola S K. Round-Robin Data Center Selection in Single Region for Service Proximity Service Broker in CloudAnalyst[J]. International Journal of Computers & Technology, 2013, 4(2).
- [46] 王小平, 曹立明. 遗传算法:理论、应用与软件实现[M]. 西安交通大学出版社, 2002.
- [47] 赵诗奎. 基于遗传算法的柔性资源调度优化方法研究[D]. 浙江大学, 2013.
- [48] 杨大地, 张春涛. 均匀两点交叉遗传算法[J]. 重庆师范大学学报(自然科学版), 2004, 21(1):26-29.
- [49] 牛海春. 基于移动 Agent 的移动云计算任务迁移机制研究[D]. 河南科技大学, 2015.
- [50] 齐彦丽, 周一青, 刘玲,等. 融合移动边缘计算的未来 5G 移动通信网络[J]. 计算机研究与发展, 2018, 55(3):478-486.
- [51] Deng M, Tian H, Fan B. Fine-granularity based application offloading policy in cloud-enhanced small cell networks[C]// IEEE International Conference on Communications Workshops. IEEE, 2016.
- [52] 邓见光. 云计算任务调度策略研究[D]. 华南理工大学, 2014.
- [53] 施莹娟. 雾计算平台的软件构架分析与应用研究[D]. 浙江师范大学, 2016.
- [54] 王赞, 樊向宇, 邹雨果,等. 一种基于遗传算法的多缺陷定位方法[J]. 软件学报, 2016, 27(4):879-900.
- [55] 施巍松, 孙辉, 曹杰,等. 边缘计算:万物互联时代新型计算模型[J]. 计算机研究与发展, 2017, 54(5):907-924.
- [56] 周悦, 芝张迪. 近端云计算:后云计算时代的机遇与挑战[J]. 计算机学报, 2018.
- [57] 宋晓诗, 闫岩, 王梦源. 面向 5G 的 MEC 系统关键技术[J]. 中兴通讯技术, 2018(1):21-25.
- [58] 郎为民, 安海燕, 姚晋芳. 移动设备自适应计算卸载研究[J]. 电信快报, 2017(1):2-5.
- [59] Shi W, Cao J, Zhang Q, et al. Edge Computing: Vision and Challenges[J]. IEEE Internet of Things Journal, 2016, 3(5):637-646.

在校参与项目成果及获奖

- 【1】 Kong Y, Zhang Y, Wang Y, et al. Weight-polling based task classification towards flexible computing[C]// IEEE International Symposium on Microwave, Antenna, Propagation, and Emc Technologies. IEEE, 2017:489-494. (EI 检索)
- 【2】 张毅坤, 孔月, 高校离退休工作处信息管理系统软件 V1.2, 软件著作权, 2017SR503362.
- 【3】 荣获“优秀学生干部”称号;
- 【4】 连续三年荣获研究生学业奖学金。