

密级：

保密期限：

北京邮电大学

硕士学位论文



题目： 参与式感知平台激励机制研究与实现

学 号： 2013111243

姓 名： 王东升

专 业： 通信与信息系统

导 师： 龚向阳

学 院： 网络技术研究院

2016 年 2 月 5 日

独创性（或创新性）声明

本人声明所呈交的论文是本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京邮电大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名：_____ 日期：_____

关于论文使用授权的说明

学位论文作者完全了解北京邮电大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属北京邮电大学。学校有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存、汇编学位论文。（保密的学位论文在解密后遵守此规定）

保密论文注释：本学位论文属于保密在三年解密后适用本授权书。

本人签名：_____ 日期：_____

导师签名：_____ 日期：_____

参与式感知平台激励机制研究与实现

摘 要

近年来,随着智能设备普及、可穿戴设备的兴起和网络带宽的提速、移动网络的广泛覆盖,具有联网能力、内置了丰富传感器的智能终端深刻的改变了人们的生活。在此背景下,参与式感知这种新型的获取感知数据的方式应运而生。参与式感知结合众包、移动互联网的优势,以配备了智能联网设备的人为中心,有能力获得多样的环境、个人健康等感知数据,具有超大规模的时空覆盖特性,体现了超越传统传感器网络性能的巨大潜力。

参与式感知真正发挥其潜力还面临诸多挑战,本文主要研究其中的一个重要方向-激励机制的设计和实现。参与式感知不需要投入固定传感器网路中的基础设施建设成本,但其高度依赖参与者的参与来贡献感知数据。参与者提供及时的、准确的、充分覆盖感知区域的数据观测值决定了参与式感知应用的成败。

为了深入研究参与式感知中的激励机制,并为本课题依托的实验平台设计和实现激励机制,本文首先介绍了参与式感知的概念、特点和典型应用,之后详细分析参与式感知中激励机制的研究现状,对其进行分类总结,并且设计一种动态分配预算的激励机制,通过仿真实验验证其较好的性能表现;然后为本文所依托的参与式感知实验平台设计激励机制的具体实现方案,从需求分析、概要设计到详细设计,使得激励模块集成到平台中;接下来介绍系统的部署情况,对激励系统的功能进行了测试;最后总结了全文工作,对未来工作进行展望,对研究生期间工作进行总结。

本文的主要创新点在于综合现有参与式感知激励机制中的固定价格激励和基于逆向竞拍的用户报价激励两种方式的优点,设计动态分配预算的激励机制,其适用于参与式感知中广泛的应用场景--绘制城市级的感知数据地图等场景,根据用户的参与情况分配预算以节省开销、满足不同用户对价格的需求,同时又避免竞价带来的博弈开销、不支持实时性的弊端,通过仿真实验对比其性能,提出进一步的改进策略;主要的工程实践在于在实验平台中设计并实现了激励模块,包括任务发布和激励分配、参与者选择等需求。

关键词 参与式感知 激励机制 动态分配预算

RESEARCH AND IMPLEMENTATION OF INCENTIVE MECHANISM FOR PARTICIPATORY SENSING PLATFORM

ABSTRACT

Nowadays, mobile Internet has been rising with the popularity of smart phones and wearable devices as well as increasing speed of network band and wider coverage of mobile cellular network. Mobile Internet based applications reshape our everyday life in all aspects. People are surrounded with the convenient provided by mobile Internet services. Mobile Internet may succeed not only in business field, but also presents huge potential in science, research, culture and public welfare. Participatory sensing is a typical mobile Internet paradigm which relies on ordinary citizens voluntarily making contribution to environmental sensing. Smart devices enabled with networking ability have powerful computing ability, various kinds of embed sensors and can be easily extended with increasingly popular wearable devices through Bluetooth WiFi protocol. Utilizing these devices participants can acquire a wide variety of sensing data about environment, health of themselves and share with others through network. This is participatory Sensing.

Hardware requirements for participatory sensing is ready, however, software requirements is still not satisfied. On one hand, software means participatory sensing applications, on the other hand, it means mechanisms or issues unresolved, including incentive mechanism, participant privacy protection, quality guarantee of sensed data and so on. This paper focused on incentive mechanism design and implementation in participatory sensing because recruit and maintain adequate participants is the key of success in participatory sensing.

This paper first introduced the notation, characters and typical applications of participatory sensing. Then, retrieved literature in the field of

incentive mechanism in participatory sensing carefully and classified the incentive mechanisms to summarize properties. Dynamic budget allocation incentive mechanism was proposed and simulation was done to compare the performance among three typical incentive mechanisms. Secondly, for the experimental platform, three incentive mechanisms were designed and implemented in its basic mode from requirement analysis to general design until detail design and implementation. Last, we deployed the incentive system and made the functional test, summarized the whole work in this paper and looked ahead the work in the future.

KEY WORDS participatory sensing, incentive mechanism, dynamic budget allocation

目 录

第一章 绪 论.....	1
1.1 研究背景与意义	1
1.2 研究内容与目标	1
1.3 论文结构	2
第二章 参与式感知及相关技术.....	4
2.1 参与式感知.....	4
2.2 本课题所依托的参与式感知数据收集平台	6
2.3 平台开发环境和技术路线.....	7
2.3.1 HTTP&HTTPs 协议	7
2.3.2 JSON 格式介绍.....	9
2.3.3 Spring 框架与 Spring MVC.....	10
2.3.4 Tomcat 服务器	12
2.3.5 REST	13
2.3.6 iOS 平台推送消息.....	13
2.4 本章小结	14
第三章 参与式感知平台激励机制的研究	15
3.1 引言.....	15
3.2 参与式感知平台激励机制相关工作	15
3.3 一种动态分配预算的参与式感知平台激励机制	16
3.3.1 设计原则	17
3.3.2 定价方式的确定	18
3.3.3 实时地选择参与者和上传数据	20
3.3.4 动态分配预算算法	21
3.4 参与式感知激励机制仿真实验	26
3.4.1 仿真场景建模	26
3.4.2 对比实验设置	27
3.4.3 性能比较	30
3.5 本章总结	34
第四章 参与式感知平台激励系统的需求分析和总体设计	36
4.1 功能性需求分析	36
4.1.1 发布任务	36

4.1.2	参与感知活动	37
4.1.3	获得激励报酬	37
4.2	非功能性需求分析	38
4.2.1	可用性	38
4.2.2	安全性	38
4.2.3	可扩展性	38
4.3	参与式感知平台激励系统的总体设计	38
4.3.1	参与式感知平台激励系统总体结构	38
4.3.2	参与式感知平台激励系统与其他系统的交互	39
4.3.3	参与式感知平台激励系统模块划分	40
4.4	本章小结	45
第五章	参与式感知平台激励机制的详细设计	46
5.1	任务管理模块	46
5.1.1	发布任务	46
5.1.2	任务列表管理	49
5.1.3	完成任务	50
5.2	参与者选择模块	51
5.2.1	竞拍管理	51
5.2.2	任务锁定	51
5.3	预算分配模块	52
5.3.1	每轮任务预算的确定	52
5.3.2	子任务预算分配	53
5.4	其他模块或功能实现	53
5.4.1	推送功能	53
5.4.2	用户系统	56
5.5	本章总结	57
第六章	系统部署与测试	58
6.1	系统部署	58
6.1.1	系统开发环境	58
6.1.2	系统运行环境	58
6.1.3	Tomcat 集群和 nginx 负载均衡的部署与配置	58
6.2	功能测试	60
6.2.1	任务管理模块	60
6.2.2	预算分配和用户选择模块	63

6.2.3	推送功能	63
6.2.4	用户系统	64
6.3	本章总结	65
第七章	总结与展望	66
7.1	工作总结	66
7.2	工作展望	66
7.3	研究生期间主要工作	67
参考文献	68
致谢	70
作者攻读学位期间发表的学术论文目录	71

第一章 绪论

1.1 研究背景与意义

参与式感知^[1]是近年来兴起的获取感知数据的方式,其使市民仅仅依靠手中的智能设备,利用内嵌其中的多种传感器,贡献城市级别的大范围感知数据。与传统的固定传感器网络相比,参与式感知方式具有部署成本低、无维护成本、覆盖范围广等优点,并且“传感器+人”的模式比固定的传感器节点具有更大的灵活性和更智能的感知能力。

然而,参与式感知模式中一项重要的挑战是设计有效的激励机制,动员参与者为参与式感知应用贡献数据。参与式感知应用虽然多是城市资源管理、环境监测等具有公益性的感知活动,主要靠市民的自愿参与,但合适的激励机制可以促使更多的参与者参与到感知活动中,相比于所付出的激励成本能获得成倍的回报,使活动发布者和参与的市民都收益。相关文献对参与式感知领域中的激励机制做了广泛而深入的研究:例如在大学校园里发起了一项拍照记录垃圾回收的感知实验活动,主要测试微支付方式的激励机制效果^[2];另一大类区别于简单的微支付方式的激励机制是将经济学中的逆向竞拍机制引入到参与式感知活动中^[3],通过用户上报所期望的激励报酬来改进平台定价所暴露的问题。之后大量文献基于逆向竞拍模型改进其算法,例如引入虚拟信誉值(VPC, Virtual Participation Credit)来促进公平性,避免感知开销爆炸现象^[4],或者从感知数据覆盖范围、感知数据质量保障、预算受限等方面提高基于逆向竞拍激励机制的性能^{[5][6][7]}。除了通过经济手段激励用户参与外,也可以从非经济激励方面设计激励机制,包括只有提供数据才能获得授权查看数据服务、用游戏、社交等元素设计感知任务的执行过程,这些非经济激励措施都丰富了设计激励机制的思路。

随着智能设备普及、传感器类型的日益丰富和网速的不断提高,参与式感知模式越来越受到人们的重视,基于参与式感知的应用也层出不穷。应用的使用场景不同,其最适合的激励机制方式也各不相同,总结现有激励机制的设计思路,结合参与式感知应用的具体使用场景,设计出适合的激励机制,是参与式感知应用成功的关键。

1.2 研究内容与目标

正如上节所提到的,参与式感知在展示巨大潜力、美好前景的同时,距离真正发挥作用还有很多基础工作要研究。本文主要关注如何设计激励机制吸引用户参与

到各种感知活动中来，并且利用这种机制促进用户上传优质数据，维持整个社区健康发展。

所谓激励机制，原本是经济学中常见的原理，是指在组织系统中，激励主体系统运用多种激励手段并使激励规范化和相对固定化，从而与激励客体相互作用、相互制约的结构、方式、关系及演变规律的总和^[8]。虽然参与式感知活动主要靠参与者或者称作志愿者的个体主动参与，但是感知过程中不可避免的会产生经济开销（手机的电量消耗、数据流量传输的费用等）和心理开销（定时采集数据需要人工干预、操作流程复杂枯燥等），如果没有适当的激励机制补偿参与者的经济及心理开销，参与者的参与热情必然会不断衰减，最终导致参与式感知应用的失败^[9]。

激励机制可以利用实际的经济报酬方式或者虚拟的积分方式。利用报酬方式时，付给参与者的报酬必须足够多以抵消参与者的感知开销，并且激励其持续参与到感知活动中，同时也要足够少使得感知活动发起者可以在预算范围内维持感知活动的运行。参与者的开销是用户的私人信息，参与同一次感知活动的不同参与者的开销不尽相同，而且参与者也有充分的动机虚假报告开销来获取更高的报酬，所以参与式感知应用的激励机制要设计得使感知活动发起者经济上可行，同时促进感知活动参与者真实报价并保证一定的服务质量来增加其收益形成良性的循环。

本文主要研究了参与式感知中通用的激励机制设计原则，分类比较已有的方案，并且根据论文依托的实验平台具体设计一种动态分配预算的激励机制，仿真对比激励机制效果。在本课题所依托的参与式感知实验平台中设计并实现了两种典型的激励方式，即固定价格激励机制和基于逆向竞拍的激励机制，包括需求分析、概要设计、详细设计和技术实现，还负责平台对感知数据的接收、平台开放的 REST 接口的设计与实现等开发工作。

1.3 论文结构

本文共分为七章：

第一章 绪论。介绍研究背景和意义、阐述研究内容和目标、论文结构。

第二章 参与式感知及相关技术。介绍了参与式感知的概念、本课题依托的参与式感知实验平台架构和平台中的关键技术。

第三章 参与式感知平台激励机制的研究。详细介绍激励机制研究现状，对已有激励机制研究进行分类分析，总结其优点设计一种动态分配预算的激励机制，通过仿真实验对比和验证其性能。

第四章 参与式感知平台的需求分析和总体设计。对实验平台中的激励系统进行充分的需求分析，通过总体设计将激励系统划分为任务管理模块、用户选择模块和预算分配模块三部分，还有一些用户管理、数据接收、推送通知等平台任务基础需

求，确定了对移动端的接口设计并提出对数据存取的需求。

第五章 参与式感知平台激励机制的详细设计。按模块划分进行详细设计，列举了重要的数据结构和关键算法、流程。

第六章 系统部署与测试。介绍了系统部署情况和系统功能测试，将展示激励系统和其他功能的实际实现效果。

第七章 总结与展望。对本文的工作进行了总结，展望以后的工作，列出研究生期间的工作和成果。

第二章 参与式感知及相关技术

2.1 参与式感知

近年来，乘着摩尔定律的浪潮，手机性能在飞速的提高，处理能力更强、嵌入的传感器更丰富、存储空间更大、网络传输速率更快。今天的手机已经从仅仅能打电话的功能机阶段进化到有丰富计算、感知和通信能力的智能设备时期。根据 ITU 的数据^[10]，截止到 2015 年末，全球移动蜂窝用户会达到 70 亿，其中有 20 亿智能手机用户，并且据调查，83% 的互联网用户更喜欢用他们的移动设备上网。随着手机技术和通信技术的进步，手机已进入到大众生活的各个角落，在此条件下，一种新型的完成大规模感知任务的方案——“参与式感知”应运而生。参与式感知的关键思路是使普通大众有能力对周边环境用手机进行感知，并且分享收集到的感知数据。

虽然手机不是特别制造来用作感知的，但事实上手机可以很好地作为复杂的传感器来工作。手机中的相机可以作为视频或图片传感器，麦克风是以高于二倍人声频率主要区间的采样频率采样的音频传感器，嵌入的 GPS 接收器可以提供位置信息，其他的嵌入传感器如陀螺仪、加速计和接近传感器可以综合利用来感知和估计手机携带者的环境信息，比如手机携带者是在走路还是骑行等等。另外，通过蓝牙或有线连接，外置传感器可以容易的和手机连接，例如空气质量检测仪等，手机作为传感器数据集中平台有灵活的可扩展性。

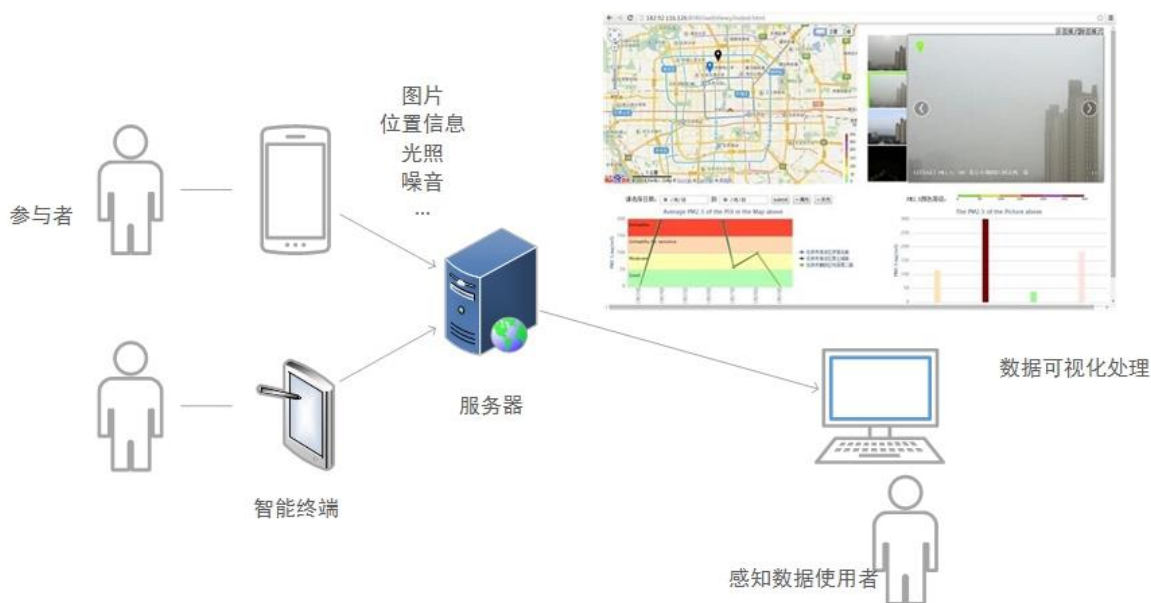


图 2-1 参与式感知应用场景

典型的参与式感知应用工作在一种中心型的方式，即参与者通过手机传感器采集的数据，通过无线通信链路将数据上传到中央服务器进行处理。图 2-1 展示了一个

典型的参与式感知用于收集环境数据的应用场景，手机端的感知任务可以手动触发、自动触发（更多的称作机会感知）或根据环境上下文触发；在服务器端，数据被分析、处理成可用的形式，通过数据可视化技术在个人的手机上或 Web 界面中展示出来。

参与式感知对比于传统的传感器网络有四点优势，后者通常需要部署大量固定的无线传感器设备，尤其是在都市范围的覆盖规模下。第一点优势，因为参与式感知利用现有的感知（手机中的传感器）和通信（蜂窝或 WiFi）基础设施，部署开销几乎为零；第二，手机用户固有的移动性使得参与式感知能提供空前的时空覆盖范围，而传统固定传感器网络所感知的范围是固定不变的；第三，手机应用的开发工具和发布平台都很成熟，使得参与式感知应用的开发和部署变得简单；最后，将普通大众吸引到参与式感知活动中，可以宣传环保等先进理念，感知的过程也是和市民互动交流的过程，这将巨大地改变人们的生活方式。

参与式感知概念最早提出时，构想将参与式感知应用到公众健康、城市规划、社会现象记录和自然资源管理等方面^[1]。文献^[1]中将参与式感知应用分为两类：个人中心型和环境中心型。个人中心型中，DietSense 用手机记录饮食信息，参与者可以将信息分享给医生或营养专家；PEIR (Personal Environmental Impact Report)使得用户通过手机了解到自己所处的环境的污染程度；BikeNet 提供了监测参与者骑行体验的系统；PetrolWatch 通过参与者对车外的加油站服务牌进行拍照，上传到中央处理器后处理分析得到汽油价格，人们可以检索感兴趣区域的最便宜的汽油价格。环境中心型中，Haze Watch 利用外接传感器测量 CO 、 O_3 、 SO_2 、 NO_2 的浓度，与固定的气象站对比，手机测量精确度可能要差，但是参与式感知的方式可以提供更大的覆盖范围，更细粒度的测量范围，参与者的移动性也有机会观测到突发的污染情况，这是固定观测站无法提供的服务，所以可以用参与式感知的方式，获取细粒度覆盖范围的观测样本，作为高保真固定观测站数据的补充；类似的，EarPhone 通过手机中的麦克风采集声音，绘制城市级别的噪音地图来研究噪音和相关社会行为的关系；Nericell 利用嵌入的加速计传感器、麦克风和定位系统（GPS、GSM 基站辅助定位）综合判断路况条件，比如坑洼、碰撞、刹车和鸣笛（可以进一步判断交通拥堵等信息），通过手机来采集并上传。概括起来，参与式感知应用的普遍目标是尽可能精确地观察、发现一些现象、过程或状态，然后分发给感兴趣的人们，取之于民、用之于民。

基于参与式感知概念的应用如雨后春笋般出现，展示出巨大的活力和潜力，但是成功部署参与式感知应用到人们的日常生活中还需要解决一系列问题。很多参与式感知应用的成功依赖于大量用户的参与，贡献足够数量、优质的数据，如何设计激励机制吸引用户参与是亟待解决的问题。由于参与式感知系统无法控制参与者的

行为,平台收到的数据在时间和空间上具有随机性,会造成数据集的不完整、不可靠。而很多应用需要推断用户所处环境上下文和活动状态信息,这需要综合各种传感器的数据结合机器学习技术来识别人类活动模式,应用采集了用户的各种周边环境数据,很多都会暴露用户隐私,例如位置信息、轨迹数据和拍摄的图片、视频等,如何保护用户隐私是消除用户疑虑、保证其积极参与的重要工作。同时,参与式感知应用还要有能力鉴别数据的真实性、准确性,否则良莠不齐甚至充斥伪数据的数据集将毫无价值。最后但也是很重要的一点,参与式感知应用要考虑减少手机的资源开销,参与者对于手机电量和流量的消耗比较敏感。

本文主要研究参与式感知课题中激励机制的问题,主要采用经济类的激励机制,促进用户参与到感知活动,合理分配预算,简化交互流程,促进数据的均匀分布,从而保证参与式感知活动的可靠运行。

2.2 本课题所依托的参与式感知数据收集平台

本课题依托的实验平台由客户端、服务器和浏览器三部分构成,整个平台的架构图如图 2-2 所示,客户端是在 iOS 和 Android 平台上的应用软件,可以通过 HTTP 请求与服务器通信,服务器也可以通过苹果公司的 APNs 服务向客户端主动推送消息。服务器端部署了 Tomcat 应用服务器,处理客户端和浏览器发起的 HTTP 请求,接收客户端上传的图片数据、JSON 格式的感知数据,也返回用户请求的附近图片、感知数据信息、感知任务信息等。服务器还部署了 HBase 分布式 NoSQL 数据库,实现海量数据的分布式存储和高性能访问。为了实现感知数据的可视化,实现了 Web 端浏览器中的感知数据展示页面,并且可以通过浏览器查看用户轨迹信息、轨迹预测对比效果图、发布感知任务等,实现了 Web 端的控制平台。

本课题所依托的参与式感知数据收集实验平台,现有功能包括图片、感知数据文本值的上传、接收和存储;客户端支持 Android 平台和 iOS 平台,可以采集感知数据,本地存储并上传到平台服务器。实验平台通过参与者拍摄的室外固定场景的照片,结合拍摄时间抓取的空气质量数据建立预测模型,之后通过同一场景的照片就可以估计拍摄照片当时当地的空气质量信息,此方案是结合图像处理、机器学习等热门技术解决空气质量监测问题的一种创新方案。传统的空气质量监测站点设备昂贵、维护成本高,采用上述方案结合参与式感知思想,号召大量市民利用手机拍摄周围室外环境照片,即可在城市范围内提供细粒度的空气质量监测数据,是固定部署的空气监测站点的补充和增强,实现“身边的空气质量监测站点”构想。

系统平台服务器现有天气抓取模块、PM2.5 计算模块,均已完成开发,轨迹分析模块和激励模块需要从无到有的开发,轨迹分析模块、客户端和 Web 端的开发工作由其他同学负责,我主要负责激励机制的设计和实现、通信接口 API 的设计和实

现。

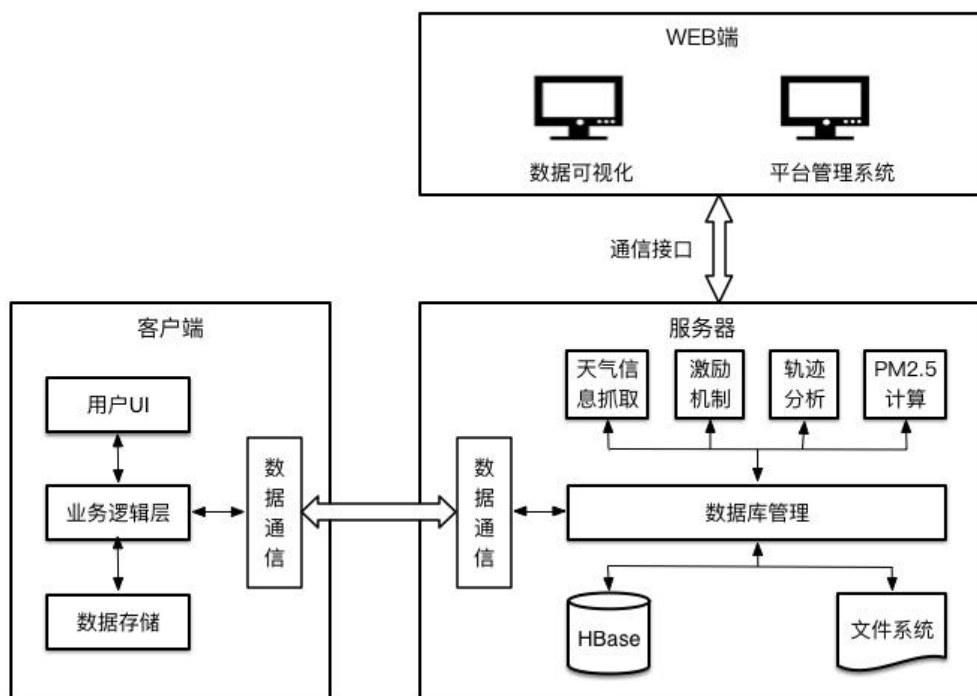


图 2-2 参与式感知数据收集实验平台

2.3 平台开发环境和技术路线

本课题所依托的实验平台要实现这样一种感知活动，即通过参与者拍摄、上传其所处环境的室外场景照片，通过和平台收集的空气质量信息、天气信息建立关联模型，之后仅依靠再次拍摄、上传照片，通过图像处理和机器学习相关技术，给出用户所处环境的空气质量估计数据。为实现这一参与式感知应用，本节介绍相关技术，在第四章的激励系统的设计和实现中会有应用。

2.3.1 HTTP&HTTPs 协议

由接下来的第四章中的激励系统需求分析将会看到，客户端需要上传文本文件、二进制文件到平台，当检索资源时也需要下载文本文件和图片等二进制文件到客户端，根据网络分层模型，可以利用可靠地面向连接的 TCP 协议实现客户端和平台的通信。但是通过 socket 链接，客户端和服务端都需要做很多连接、通信过程中的琐碎工作，而现有的应用层协议，比如 HTTP、FTP 都可以完成我们的需求，简化工作量，客户端开发中 iOS、Android 平台也提供了丰富的对于网络连接的 SDK 支持，所以我们不需要基于 TCP 定制应用层协议来开发。HTTP(HyperText Transfer Protocol)是无状态的应用层协议，应用于分布式的、协作的、超文本信息系统中^[12]。HTTP 建立在可靠的传输层或会话层连接之上，通过交换消息(message)来通信。HTTP

客户端与 HTTP 服务器建立连接，发送一个或更多的 HTTP 请求；HTTP 服务程序接受连接，通过发送 HTTP 响应来服务 HTTP 请求。

客户端通过请求消息给服务器发送 HTTP 请求，请求消息由请求行开始，包括请求方法、URI 和协议版本；接下来是头部，头部由一个空行来结束；最后是消息体，表 2-1 是请求 `www.example.com/hello.txt` 资源文件的请求消息示例。

表 2-1 HTTP 请求消息示例

HTTP 请求消息示例

```
GET /hello.txt HTTP/1.1
User-Agent: curl/7.16.3 libcurl/7.16.3 OpenSSL/0.9.7l zlib/1.2.3
Host: www.example.com
Accept-Language: en, mi
```

服务器通过一次或多次 HTTP 响应消息来回应客户端的请求。每个响应消息也包括三部分：开始行是状态行包括协议版本、成功或失败状态码，之后是用于说明状态码的文本；接着是响应头部（包括服务器信息、资源元数据和展示元数据），空行代表头部结束；最后是响应消息体。表 2-2 是得到了成功响应的消息体。

表 2-2 HTTP 响应消息示例

HTTP 响应消息示例

```
HTTP/1.1 200 OK
Date: Mon, 27 Jul 2009 12:28:53 GMT
Server: Apache
Last-Modified: Wed, 22 Jul 2009 19:15:56 GMT
ETag: "34aa387-d-1568eb00"
Accept-Ranges: bytes
Content-Length: 51
Vary: Accept-Encoding
Content-Type: text/plain

Hello World! My payload includes a trailing CRLF.
```

采集的相关传感器信息，可以构造成 JSON 格式的文本字符串发送给服务器，而二进制的文件（例如照片、音频等）可以通过二进制比特流放在消息体中来传输。

考虑到采集数据中有用户的位置等隐私信息，处于安全性考虑，如果采用 HTTP 协议，面临窃听、篡改和冒充风险。而 HTTPS 是 HTTP 的安全版本，基于 SSL/TLS

协议之上，其所有的信息都是加密传播的，第三方无法窃听，具有校验机制，一旦被篡改，通信双方会立刻发现，同时配备身份证书，以防止身份被冒充。SSL (Secure Sockets Layer)^[13]由 NetScape 于 1994 年发布，后互联网标准化组织 ISOC 接替 Netscape，发布了 SSL 的升级版 TLS，最新版本是 2011 年的修订版 TLS1.2。

SSL/TLS 协议采用公钥加密算法，客户端先向服务器索要公钥，然后用公钥加密信息，服务器对收到的密文用自己的私钥解密还原客户端的信息。为了保证开始获取的服务器公钥不被篡改，需要将公钥放在数字证书中，只要数字证书可以验证，公钥就是可信的。而公钥加密计算量很大，为了减少时间消耗，SSL/TLS 为每一次会话生成一个会话密钥，公钥只用来加密会话密钥，之后用会话密钥来对称加密本会话中的数据，这样就减少了加解密的计算开销。

iOS9 开始苹果也大力推进使用 HTTPs 进行数据通信，综上所述，我们采用 HTTPs 协议来进行感知数据的上传和下载。

2.3.2 JSON 格式介绍

表 2-3 JSON 字符串示例

JSON 字符串实例

```
{
  "students": [
    {
      "姓": "冯",
      "名": "云"
    },
    {
      "姓": "寇",
      "名": "秦荔"
    },
    {
      "姓": "杨",
      "名": "婷婷"
    }
  ]
}
```

JSON (JavaScript Object Notation) 是一种轻量级的数据交换格式^[14]。人们很容易的阅读和构建这种格式，而对机器来说，解析和构建 JSON 同样非常方便。JSON

的标准是 JavaScript 编程语言标准的一个子集，JSON 是编程语言独立的一种文本格式，是理想的数据交换语言。表 2-3 为展示了三个学生姓名的 JSON 字符串示例。

JSON 中由 {} 大括号包围一个对象，[] 中括号包围一组对象，对象中都是 key、value 形式的键值对，由冒号“:”分隔，键值对之间由“,” 逗号分隔。JSON 只有以上几种语法元素，非常简单，但由于各种元素可以嵌套，能提供强大的、灵活的数据表达能力。

2.3.3 Spring 框架与 Spring MVC

● Spring 框架介绍

Spring 框架是一个 Java 平台应用开发的优秀框架，提供了开发 Java 应用的基础设施支持，使得开发者可以专心于应用的开发^[15]。

使用 Spring 框架开发程序，好处除了可以利用优秀的第三方开源框架提供的工具，避免重复发明轮子，重要的是 Spring 框架依赖注入和控制反转的思想，使得程序具有高内聚、低耦合特性以便于扩展。图 2-3 是 Spring 框架运行时模块图，使用者根据自己的需要选择添加哪些模块。

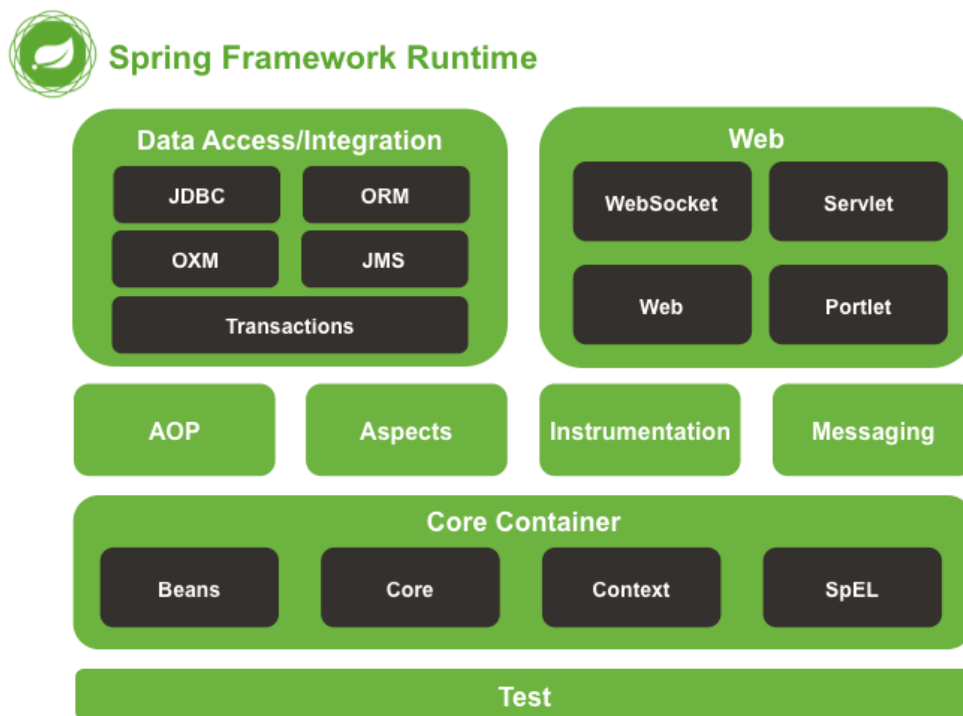


图 2-3 Spring 框架运行时模块图

Java 平台有多个版本，Java SE (Java Platform, Standard Edition) 包含 JVM 和 Java 最核心、最基本的类库 API；而 JavaEE (Java Platform, Enterprise Edition) 是在 Java SE 之上，定义的一套标准，用来支持企业级开发，比如用来接收 http 请求的 servlet

标准、和关系型数据库交互的 API (JPA, Java Persistence API) 等。Java EE 定义的每一条规范称为 JSR (Java Specification Request), 由 JCP (Java Community Process) 开发维护, 就像 RFC 之于 IETF。

基于 Java EE 标准, 业界公司实现了很多产品, 例如 IBM 的 WebSphere Application Server (WAS), Oracle 的 WebLogic, RedHat 的 JBoss/WildFly 等等, 这些实现都遵循 Java EE 的标准, 而且实现了 Servlet spec、JPA spec 等大部分重要的 specification, 称为应用服务器 (Application Server), Java EE 规范保证了我们编写的应用能够部署/运行在上面; 而在开源界更常用的 Tomcat/Jetty 则称为 Web 容器 (Web Container), 只实现了 Java EE 里的 Servlet spec、JSP spec 等和 Web 相关的 specification, 我们如果需要依赖注入, 就需要自己引入 Spring, 需要用 JPA 访问关系型数据库, 就需要自己引入 Hibernate。

Spring 中采用控制反转思想, 框架 (Framework) 和库 (Library) 的最大区别就是控制反转。调用库中编写好的函数, 是主动的调用; 而在框架下开发, 编写好的程序由框架来执行, 是被动的调用, 编程者负责向框架中注入依赖, 作为框架的回调、实现或被托管的对象, 这样做可以解耦, 解除作为主体的责任, 专注于业户逻辑的实现。Spring 中通过注解 (annotation) @Component 标示一个类需要框架来实例化对象, 通过注解 @Autowired 进行依赖注入, 合成一个对象。

● Spring MVC 模块

如图 2-3, Spring 框架所示, Spring 由许多模块组成, 最核心的有 Spring-core、Spring-beans, 和数据库交互的有 Spring-jdbc、Spring-orm 等, 我们用 Spring 来构建 RESTful 接口, 介绍一下 Spring Web MVC 模块。

Servlet 是 Java EE 里处理 HTTP 请求的技术/规范, Spring MVC 也是基于 servlet 来设计的, 基于一个核心的 DispatcherServlet, HTTP 请求进来之后由它分发给其他的 Controller 处理。其实现了 MVC 的设计模式, Model 负责封装应用数据层, View 负责渲染应用数据, 通常生成浏览器解释的 HTML 文件 (JSON 格式的字符串), Controller 负责处理用户请求, 绑定相应的模型并传给 view 层做渲染。

图 2-4 中的编号依次代表:

- 1、DispatcherServlet 接受 http 请求, 查询 Handler Mapping 找到对应的 controller;
- 2、DispatcherServlet 将请求传给对应的 controller, 根据请求方法、uri 调用特定的处理方法, 执行设定好的业务逻辑, 返回 view 名字给 DispatcherServlet;
- 3、DispatcherServlet 通过 View Resolver 得到对应的 View;
- 4、DispatcherServlet 得到了 view, 将其和 model 结合, 渲染后返回给浏览器 (或移动端应用)。

通过 MVC 设计模式, 使得应用程序的不同方面 (输入逻辑、业务逻辑和展示逻辑)

辑) 分离开, 在不同部分之间建立松散耦合的关系以利于扩展和复用。

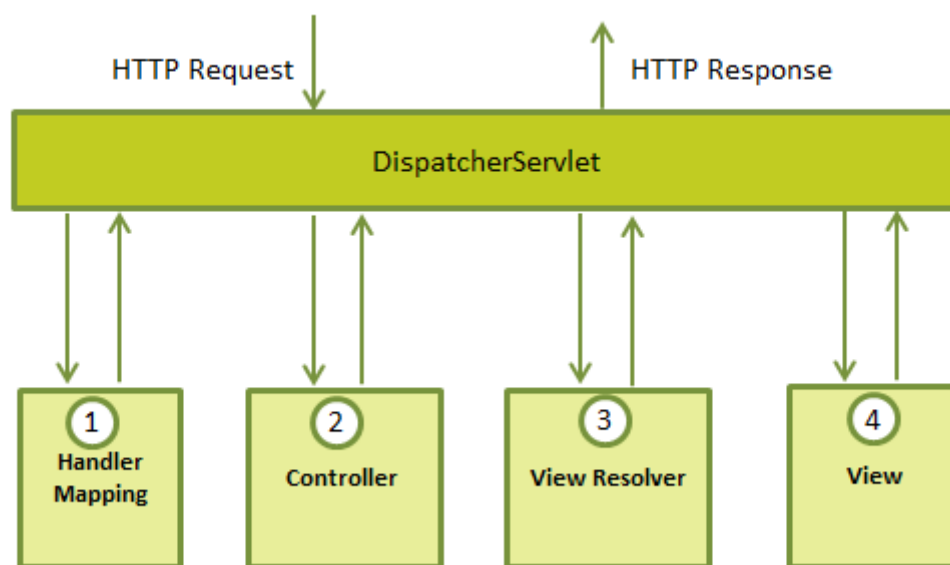


图 2-4 Spring MVC 处理流程

2.3.4 Tomcat 服务器

Tomcat^[16]是一个 Web 容器 (Web Container)，我们用 Spring 构建的 Web 应用是要借助一个容器来运行的，比如 Tomcat、Jetty，GlassFish。因为无论是用 Spring 还是其他 Java EE 框架编写出来的 Web 应用，都是一个个独立的 servlet 或者基于 servlet 的扩展，这些独立的 servlet 不能直接运行，也不能直接接收来自浏览器的 HTTP 请求。Tomcat/Jetty 这些容器一方面是一个 Web 服务器——即和浏览器交互，接收/响应 HTTP 请求；另一方面是 Servlet 容器——加载 servlet，负责管理他们的生命周期，根据 mapping 规则把分发给响应的 servlet^[11]，如图 2-5 所示。

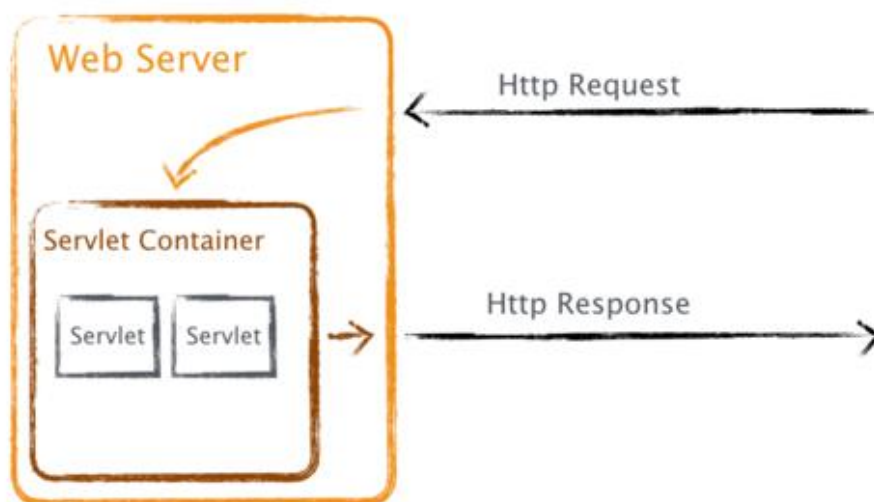


图 2-5 Servlet 原理图

这一整套都是 Java EE Servlet 规范，Spring 和容器都得遵守这些规范，用 Spring 写的 Web 应用才可以部署到 Web 容器里去。

2.3.5 REST

RESTful 网络服务是基于 REST 架构的网络服务，在 REST 架构中，一切都是资源^[17]。RESTful 网络服务具有轻量级、高可扩展性和高可维护性的特点，非常普遍的用于基于网络的应用 API 的实现。

REST (Representational State Transfer)，是一种基于 Web 的网络应用标准架构，采用 HTTP 进行数据通信。在 REST 架构中，REST 服务器提供对资源的访问，REST 客户端访问并展示资源，其中资源由 URI 表示，JSON 在 Web 服务中是最流行的资源交换格式。

HTTP 中熟知的方法普遍应用于 REST 架构中，分别有不同的语义，如表 2-4 所示。

表 2-4 HTTP 方法与 REST 语义对照

HTTP 方法	REST 架构中的语义
GET	对资源的只读访问
PUT	创建一个新资源
DELETE	移除一个已有资源
POST	更新一个已有资源或创建一个新资源
OPTIONS	得到一个资源所支持的操作类型

类似于单机中的进程间通信，计算机网络，比如互联网中的 Web 服务就是软件应用之间的通信，软件可以用不同的语言编写实现，可以部署在不同的平台，但是通过 Web 服务这一开放的协议和标准，软件应用可以方便的交换数据。基于 REST 架构的 Web 服务称为 RESTful 风格的 Web 服务，其通过 HTTP 协议通信，是一种 Web 服务的规范。对于本课题依托的参与式感知实验平台，客户端与平台服务器之间的数据交换正适合使用 REST 架构来实现。

2.3.6 iOS 平台推送消息

推送通知 (push notification) 是服务器端主动发消息给客户端的一种通信方式，本文主要关注 iOS 平台下的消息推送原理及实现方式。

推送通知是一种重要的与用户的交互方式，特别是 iOS 平台的应用，在退出前台之后无法主动发起对服务器的请求，推送是唯一和服务器交互的方式。推送也实现了一种观察者模式，用户同意接受服务器端推送消息之后，相当于注册成为了观察者，不必轮训查看所关注的事件，事件发生时会自动通知给用户。本文依托的实验平台需要服务器对客户端推送任务描述信息、用户参与感知活动进展等信息，所以简要介绍 iOS 平台的推送技术原理，在后续章节具体介绍推送的实现细节。

服务提供者将推送消息和用户信息（token）以安全的通信协议发送给 APNs（Apple Push Notification Service），苹果推送通知服务器通过鉴权认证后，确定用户信息，将消息发送给用户设备，用户设备维持着与 APNs 的长连接，收到推送通知后转交给相应的应用，无论应用是否是启动状态，前台运行还是后台运行，都可以对通知做出响应，完成与用户的交互。图 2-6 展示了多个服务者通过 APNs 推送消息通知给多个手机用户。

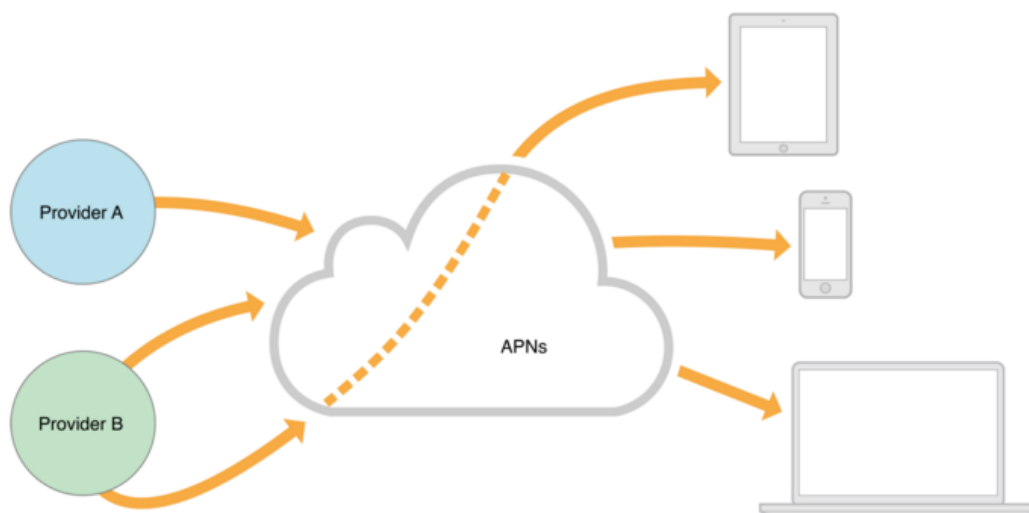


图 2-6 多服务提供者推送到多设备

具体实现环节，服务器端包括证书的准备、用户设备 token 的维护、推送内容的构建和与 APNs 的安全通信；客户端需要关注 token 上传、收到通知之后的显示与跳转。

2.4 本章小结

本章首先详细介绍了参与式感知的概念以及典型应用场景，之后介绍本课题所依托的实验平台平台的架构、现有功能，提出本文的工作中一部分就是扩充现有平台融入激励系统的功能，最后介绍平台开发中的技术路线和选择理由，为后文新系统和新功能的开发做准备。

第三章 参与式感知平台激励机制的研究

3.1 引言

参与式感知就是通过收集大量参与者上传的平台感兴趣的数据来观察和揭示一种现象的新型的感知方式。仅有一人上传数据，必然不能提供足够的、精准的感知数据来观测城市范围的观测现象，但是通过整合众多参与者的感知数据，就可以获得高品质、广覆盖的观测现象的测量值。因此，为了使参与式感知获得成功，必须调动大量的参与者投入到感知活动中，在参与式感知上下文中激励机制的研究与设计正是要解决这一问题。

参与式感知应用的成功需要大量用户的参与，除了应用本身有意义吸引用户之外，有效地宣传、推广和应用分发都是重要环节，用户第一次参与之后的用户存留和用户活跃度维持也需要重视。这些相关研究在互联网应用的领域得到了足够的重视，例如注册送红包等方式增加装机量，分享红包等方式通过社交网络促进装机量和成交量，通过设置升级制、荣誉勋章等方式提高用户粘度等等。这些措施多是结合用户心理、用户体验等因素制定推广、运营策略，而参与式感知中激励机制要讨论的问题与此侧重点不同，更多的是从传感器网络要达到的目标出发，即完成一定的感知任务、获得感知数据。本文假设用户有意愿参与到感知活动中（不讨论推广环节），并且是理性的、自私的，在此模型下研究如何维持用户参与热情，使用户提供更高质量的感知数据，谋求参与者和感知平台共赢的激励机制。

3.2 参与式感知平台激励机制相关工作

为了设计本课题所依托的参与式感知平台中的激励机制，广泛阅读了现有文献，研究其中的参与式感知背景中激励机制的设计，总结现有文献中参与式感知中激励机制的研究内容，为设计本平台中的激励机制提供理论基础。

文献^[1]首次提出参与式感知的概念，为日常生活中的手机等移动设备分配任务来形成交互式的参与式传感器网络，让公众和专业的参与者来收集、分析和共享本地信息。文献^[2]研究了微支付的激励模型，在校园中发起了参与式感知实验测试微支付的效果，证明了微支付（按数据份数支付）比宏支付（一次性付清所有报酬）效果更优，但是作者没有给出平台如何确定每份数据微支付报酬如何确定的方法。文献^[3]试图解决参与式感知中激励机制的经济模型建立问题，设计了基于逆向竞拍的动态价格激励机制，能最小化感知激励开销、稳定价格同时通过避免参与者退出感知活

动来维持足够的参与者数量，与随机选择固定价格的激励机制（类似于微支付方式）相比，逆向竞拍方法能降低激励开销 60%，同时具有激励分布的公平性和社会福利性，最重要的是解决了感知数据定价困难的问题。文献^[5]将逆向竞拍机制和预算受限的最大覆盖算法结合，解决了预算受限下感知数据覆盖范围的最大化的问题，其利用贪心算法，每次用户选择过程中，不是选择报价最低的参与者，而是根据其覆盖范围增量和开销的比值来选择参与者，得到预算受限下的最佳覆盖的感知数据集。文献^[18]提出多定语的竞拍方式，扩展了传统逆向竞拍中指考虑价格因素，参与者竞拍时提供一个竞拍向量，包含价格、数据质量、位置因素等属性，由平台根据向量计算综合的竞拍指数来选择用户，扩展了激励机制对数据采集的影响范围。文献^[19]考虑参与式感知激励机制的实时性，参与者到达感知区域后一段时间离开，平台无法获得所有参与者的信息就需要作出参与者选择决定，文献采用分阶段的采样、接受策略，解决基于逆向竞拍的参与式感知激励机制的实时性问题，并在公平性、预算效率方面做了优化。

3.3 一种动态分配预算的参与式感知平台激励机制

本文所依托的实验平台，主要功能是通过参与者持续拍摄室外照片，结合图像处理、机器学习等技术，获得当时空气质量数据的估计值，作为空气质量检测站点数据的补充参考，提供细粒度的空气质量监测值，绘制城市空气质量地图，实现“身边的空气质量监测站点”的构想，图 3-1 展示了平台通过收集参与者上传的空气质量相关的照片来绘制城市级别的空气重量感知数据地图的应用场景。



图 3-1 参与式感知用于绘制空气质量地图

这种绘制环境数据地图的参与式感知应用场景非常普遍，例如绘制城市的噪音地图、交通拥堵程度地图等，其共同特点是需要大量参与者持续一段时间周期性的参与到感知活动中。本课题中的感知数据需求是，城市范围内的室外环境照片，采集活动周期重复持续一段时间。根据本课题感知需求分析和以上对参与式感知激励

机制的研究，本文设计了一种感知预算动态分配的激励机制（Dynamic Budget Allocation）。

3.3.1 设计原则

激励机制的设计主要为了满足吸引参与者参与到感知活动中的要求，根据不同的参与式感知应用场景，还需满足不同的原则，有些原则之间甚至是矛盾的，只能做折中的优化设计，本小节将从经济可行性、数据质量、区域覆盖、公平性、参与度和自适应性六个原则具体阐述。

对于经济可行性，任何一个参与式感知项目，预算都是重要的要考虑的问题。参与式感知，需要维持大量的参与者来确保感知结果的可靠性和覆盖范围，但是有限的预算可能限制我们达到这个目标。如果采用非金钱的激励方式，可以尝试吸引具有特定兴趣和爱好的用户来参与。这类非金钱的方式包括增强应用的游戏属性、竞争性或者提供社交上的奖励（比如排名、勋章等）。这种想法试图减弱执行参与式感知任务时的心理负担，使其变得好玩。这类的激励机制可以维持大量兴趣驱动型的参与者，使得系统正常运转，但此类激励机制的设计需要时间和特定领域的知识，多是专家或有经验的设计者来完成。

另一方面，金钱类的激励机制提供了另一种解决问题的方案。通用的方法是估计出参与者愿意接受一个感知任务的最少期望报酬-保护价格工资，由系统提供这个保护价格工资来激励用户参与，从而使得参与式感知项目是经济可行的。文献^[20]指出用户保护价格工资会依赖于一些因素随着时间变化，例如手机剩余电量、通信套餐资费和用户当前对手机的使用情况等。将非金钱类的激励因素融入经济类的激励机制中，可以降低保护价格工资，发挥固有的激励因素降低开销。

对于数据质量，激励机制设计的是否合理，很大程度体现在是否促进高质量数据的上传。为了鼓励高质量数据的上传，可以应用信誉机制。每个具体的参与式感知应用对数据的要求各不相同，高质量的数据一般具有及时性、图片的清晰度高、位置分布均匀等属性，用户的信誉值可以通过该用户的历史参与表现还决定，也可以通过参与者互评，或者综合两种方式来得到。不同的信誉度的用户获得的激励是不同的，一方面可以整体促进高质量数据的上传，另一方面，在所需数据质量要求不高时，可以选择信用值适合的用户来执行任务，降低开销。

对于区域覆盖，参与式感知依靠随机分布的参与者替代固定部署的传感器来采集数据，优势是空前的时空覆盖范围，但同时也要解决感知数据分布具有随机性、不可控性的问题。现实应用中容易造成人员密集的学校、写字楼、居民区等区域数据丰富甚至冗余，而公园、郊区、道路等同样需要感知的区域在时间和空间上感知数据分布都很稀疏，无法提供可靠的、有代表性的感知数据。

感知数据分布问题主要有两方面：一是不同区域用户保护价格工资不同（感知相同的数据，有些区域价格便宜，有些区域价格过高），二是如何解决有些某些区域参与者缺失而另一区域参与者过多。如果没有设计好激励机制，做好用户选择，前者会造成只选择最低价格的用户，使得数据覆盖范围不好；后者更为严重，某些区域数据缺失。两种原因造成的结果类似，都是数据覆盖不好，但成因不同，前者是用户选择算法不合理，后者是没有充分利用激励机制和用户的移动性来促进数据分布^[21]。

对于公平性，公平性就是所有潜在参与者都有被选中的机会，是激励机制设计重点需要考虑的问题，对于维持用户参与程度、保证覆盖区域和节省预算都有重要影响。类似于现实社会中的福利制度，公平性在参与式感知中起到维持系统长时间有效运行的效果^[22]。如果仅考虑系统的开销最少，例如采用基于逆向竞拍的模型，选择报价最低的用户执行感知任务，单从一轮任务来看是最优的，但是随着高报价参与者的退出，剩余的参与者完全可以提高报价来获取更高收益，造成开销爆炸，反而增加了支出^[23]。当潜在参与者多于所需数据数量时，采用随机方式选择用户公平性最高，而设计激励机制时考虑的公平性就是公平性在支出、覆盖区域、数据质量等因素之间取得适合本应用的折中。

足够数量的参与者是参与式感知成功的关键，也是激励机制要解决的首要问题。如上所述，可以采取金钱类的和非金钱类的激励措施，维持用户持续参与。但过多的采集数据也是不必要的，造成预算的过多支出并增加了系统的传输和处理负担。所需维持的参与者数量，可以通过感知数据获取频率、目标区域的大小、观测现象的种类和变化速度、感知精度需求等因素综合来确定，之后根据激励机制维持不少于此数量的参与者持续贡献数据。

对变化的适应性，参与式感知平台可能新增加服务，感知区域可能变大，用户的开销也会随着时间增加或减少，也有其他同类的参与式感知平台作为竞争对手，所以激励机制要设计的能够适应这些变化，例如固定价格激励方式就需要引入价格确定算法，动态适应感知市场的价格变化，否则会因为激励过高造成预算浪费，或竞争力不足而达不到服务要求。

3.3.2 定价方式的确定

采用金钱类激励方式时，首先面临的问题是给用户多少奖励。用户的感知开销是可以计算的，比如用户手机电量的开销、数据传输流量费用、感知任务耗费的人工时间成本等，总计有一个感知开销。但是此开销是随时间变化的，且每个用户不尽相同，平台可以通过调查问卷的形式统计出用户的期望激励价格，定期更新，采用平台定价方式。也可以采用基于逆向竞拍的方式，由用户上报期望的保护工资价

格，完成任务后平台支付不低于此价格的激励。平台定价方式和基于逆向竞拍的方式各有优缺点，下面详细介绍。

● 平台定价方式

平台定价方式不仅要解决参与者感知开销的确定问题，还要设计激励的发放形式。文献^[2]中作者在 UCLA 大学校园发起了一项参与式感知的实验性运动，通过同学们上传对垃圾桶的拍照照片，来观察校园中垃圾分类、资源回收等环保问题。活动通过宣传单招募了 55 名志愿者参加为期 5 天的垃圾箱拍照活动，其中 25 名男性、30 名女性，年龄从 18 到 28，参与者在性别、年龄上大致均匀分布。将 55 名参与者随机的平均分成 5 份，测试 5 种不同的激励机制方式。详细实验设置如表 3-1 所示。

表 3-1 微支付中激励对照组设置

激励方式	分组说明
MACRO（一次付清）	宏支付，五天的报酬一次付清（50 美元）
MEDIUM μ	中等微支付，分次付款、中等激励（20 美分）
HIGH μ	高额微支付，分次付款、高额激励（50 美分）
LOW μ	低额微支付，分次付款、低额激励（5 美分）
COMPETE μ	竞争性微支付，分次付款、基于排名

不同组之间不了解彼此的激励程度，所以本实验可以很好地通过对比实验，测试激励程度对参与者贡献数据的影响。其中竞争组的规则是根据参与者的组内排名来决定参与者没上传一份数据获得的报酬。所有参与者最多获得 50 美元奖励。

除了照片数量作为考察标准，照片的清晰度和对照片添加的文字标注可以作为数据质量的考察标准，垃圾箱照片的位置作为数据分布的指标，综合数量、质量和分布加上感知活动结束后对参与者进行的回访，来分析各种激励方式的特点。

从数据数量上来看，最好的激励方式是竞争性的微支付方式，因为引入了竞争机制使得用户更具动力；最差的是宏支付方式，因为用户自己也不清楚 50 美元应该上传多少感知数据，会以为完成任务了而失去动力。所以按此付款的方式更适合，微支付优于宏支付方式。在微支付中，值得注意的是高等价格的分组，反而收获的数据量最少，因为受预算限制每人最多获得 50 美元，所以高等价格组用户获得了最大报酬后不再继续上传数据，总数据量反而最低。

至于数据质量和数据时空分布，竞争性的引入可以提高数据的时间空间覆盖，但是迫于竞争压力，用户上传的数据质量一般，表现为很少添加描述信息，因为描述信息是可选的，不影响用户的收益。

● 基于逆向竞拍的方式

在平台定价方式中，参与者获得的激励完全由平台来决定，每个用户的感知开销是不同的，这会增加平台的定价负担甚至因为缺少用户感知开销信息而无法给出

报价。Juong-Sik Lee 等^[3]首次将经济学中的逆向竞拍模型引入到参与式感知定价过程中，给出了不同于平台定价的另一种替代方式-基于逆向竞拍的定价方式。

在拍卖理论中，通常的竞拍是多个买家为了一个物品或者服务竞争，给出的报价不断提高，最后通常最高报价者获得物品或服务；而在逆向竞拍中，买家和卖家的位置互换，由多个卖家提供商品或服务，竞争获得买家的商业合作机会，给出的报价不断降低，最后通常最低报价者获得买家的商业合作机会。在参与式感知中，引入逆向竞拍模型后，平台发布一个感知活动描述，参与者上报完成每个任务的感知开销报价，由平台综合感知预算选择用户来完成任务，通过逆向竞拍，将激励定价的负担转移给参与者自己，并通过参与者之间的竞争、博弈，可以减少平台的感知开销。

表 3-2 不同定价方式的特点总结

	基于逆向竞拍的定价策略	非竞拍定价策略
优点	在参与者之间形成博弈；在一定程度上减少预算；能动态适应市场的变化	没有定价协商过程、简单；实时地选择用户
缺点	高的博弈开销会导致虚假竞标和开销爆炸；很难实时地选择用户；参与者报价也具有盲目性，只是把报价负担由平台转移给参与者	平台评估感知开销是一种负担；统一的固定奖励不能激励参与者上传比他人更好的感知数据

在本设计中综合两种定价方式的优点，任务发布时提供固定的激励报酬价格，没有用户报价过程简化了价格协商流程，同时根据任务的执行情况动态变化任务的激励价格，满足不同用户价格开销不同的实际情况，用户采用先到先得的选择机制同样引入了类似逆向竞拍的竞争关系，激励用户实时地迅速完成感知任务。

3.3.3 实时地选择参与者和上传数据

无论是平台定价还是用户报价，在参与者执行感知任务之前，参与者确定知晓完成感知任务将会获得的报酬的激励机制为定价优先的激励机制，而平台将任务描述信息广播给所有潜在参与者，参与者上传感知数据之后，平台根据感知数据对平台的贡献等因素确定参与者感知报酬的激励方式为数据上传优先的激励机制。数据上传优先的激励机制，因为没有价格协商的过程，其实现简单、参与者可以迅速执行感知活动，但相较于定价优先的激励机制的缺点是对上传的数据分布、参与者报价不可控，并非所有上传数据的参与者都会获得平台的激励报酬，获得报酬的参与者也可能没有达到其保护价格工资而退出感知活动，或者高于其保护价格工资很多

而造成平台预算的浪费。

参与式感知平台发布感知任务信息给所有潜在的参与者，潜在参与者根据自身情况反馈给平台感知信息，例如是否愿意参与此次任务、执行任务开销、用户所处位置等信息，平台根据潜在参与者的反馈信息进行参与者选择，被选中的用户执行感知任务，获得相应的感知激励报酬。在用户选择过程中，根据平台能否实时的选择参与者，参与者选择机制（激励激励）分为在线型和离线型两种^[20]。在线型的激励机制不用等待一个时间窗口的时间，直接根据潜在参与者的反馈信息做出是否选择该用户的决策，离线型的激励机制需要收集一定的潜在参与者的反馈信息，截止时间到达做出所有用户选择决策。传统的基于逆向竞拍的激励机制就是离线型的激励机制，需要一个报价时间窗口，报价结束揭晓竞标结果，数据上传优先的激励机制是在线型的激励机制，用户直接根据感知任务描述信息决定是否参与此次感知活动。

基于逆向竞拍的激励机制存在用户报价环节，如果支持实时性必须存在用户报价采样环节，根据掌握的部分信息来实时决定后续参与者的选择问题，如果参与者到达不是随机分布，会造成采样样本不合理而影响后续参与者的选择问题。本设计中省略报价环节，首先平台提供一个激励价格，在任务开放时间窗口根据任务执行情况动态调整价格，采用先到先得的用户选择策略，激励参与者及时贡献感知数据，引入竞争关系，参与者优先上传数据，之后获得与其贡献成正比的额外激励部分奖励。

3.3.4 动态分配预算算法

动态预算分配激励机制主要分为两部分，其一是总预算在各轮子任务中的分配，其二是每轮子预算的激励分配。

3.3.4.1 总预算的分配

首先，感知过程分为两种状态：定价状态和稳定价格转态。在定价转态下，平台大幅度的改变每轮之间的子预算分配来迅速的找到合适的预算价格；在稳定价格阶段，平台微调每轮之间的子预算来避免意外造成的市场波动，同时适应感知环境的价格变化。在图 3-2 中，红色展示定价状态、绿色展示稳定价格状态。

然后，确定第 i 轮的价格调整指数 D_i 。 D_i 由历史记录中任务执行情况计算得出。

- 情形 1. 如果感知过程处在定价状态（红色阶段），用 β_{dy} 代表， D_i 仅由最近一轮的任务完成情况决定。在这种方式下， D_i 对于供需关系非常敏感，相应的子预算变化差异较大。感知过程的初始状态适合设定为定价状态来迅速的找到合理定价。

- 情形 2. 如果感知过程处于稳定价格转态, 用 β_{sd} 表示, D_i 受最近的 M 轮任务完成信息影响, 此方法可以避免突发情况造成市场的剧烈波动, 因此适合于稳定价格阶段。



图 3-2 每轮任务预算在两个阶段的变化

默认的, 在初始化时, 感知数据收集过程处于定价阶段, $\beta = \beta_{dy}$, 第一轮的子预算设置为较低水平。

系统维持两个状态标记桶, 一个对应于动态定价状态, β_{dy} , 初始值为 1.0, 另一个对应于价格稳定状态, β_{sd} , 初始值为 0。

第 i 轮的任务执行情况, 标记为 α_i , 可以通过 t_{ai} 和 n_{ai} (分别为第 i 轮的任务完成时间和总共收集的数据份数) 来表征, 例如,

$$\alpha_i = \frac{t_{th}}{t_{ai}} \times \frac{n_{ai}}{n}, \quad (3-1)$$

t_{th} 为任务截止时间, n 为所需的数据份数。

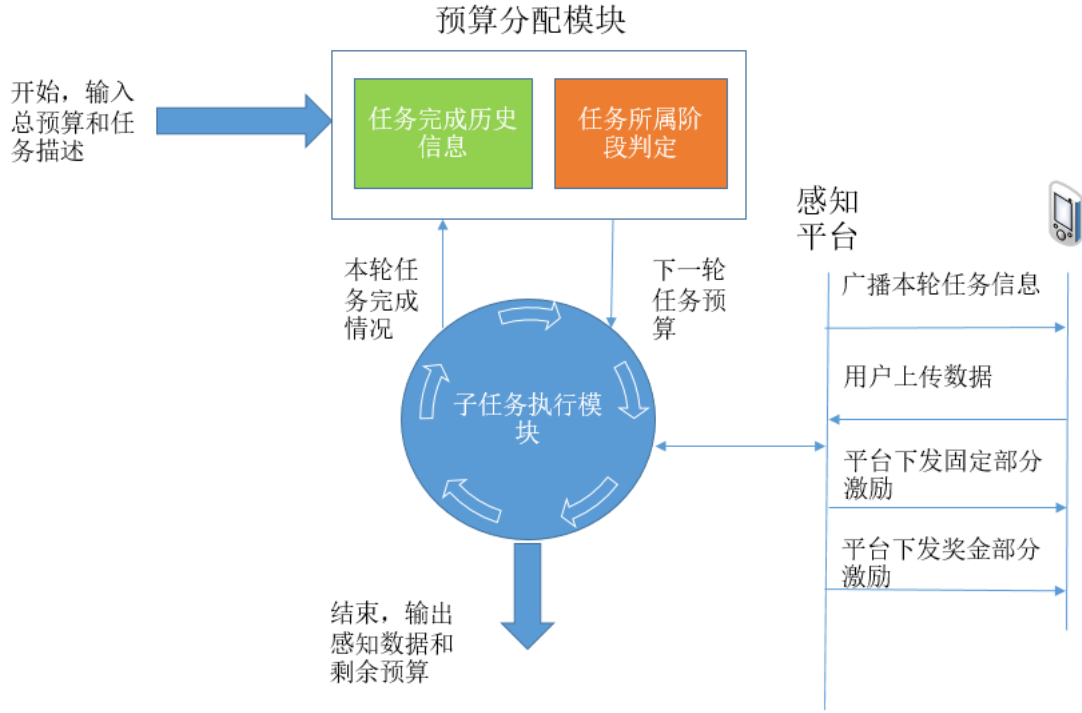


图 3-3 动态分配预算总体架构图

一旦我们得到了第 i 轮任务完成情况指标 α_i ，我们可以通过历史记录中的任务完成信息和当前的市场状态来得到价格调整参数 D_i ，如图 3-3 所示。

$$D_{i+1} = D(A, \beta) = \begin{cases} 1/\alpha_i & , \text{if } \beta = \beta_{dy} \\ \sum_{j=i-M+1}^i 1/\alpha_j & , \text{if } \beta = \beta_{sd} \end{cases}, \quad (3-2)$$

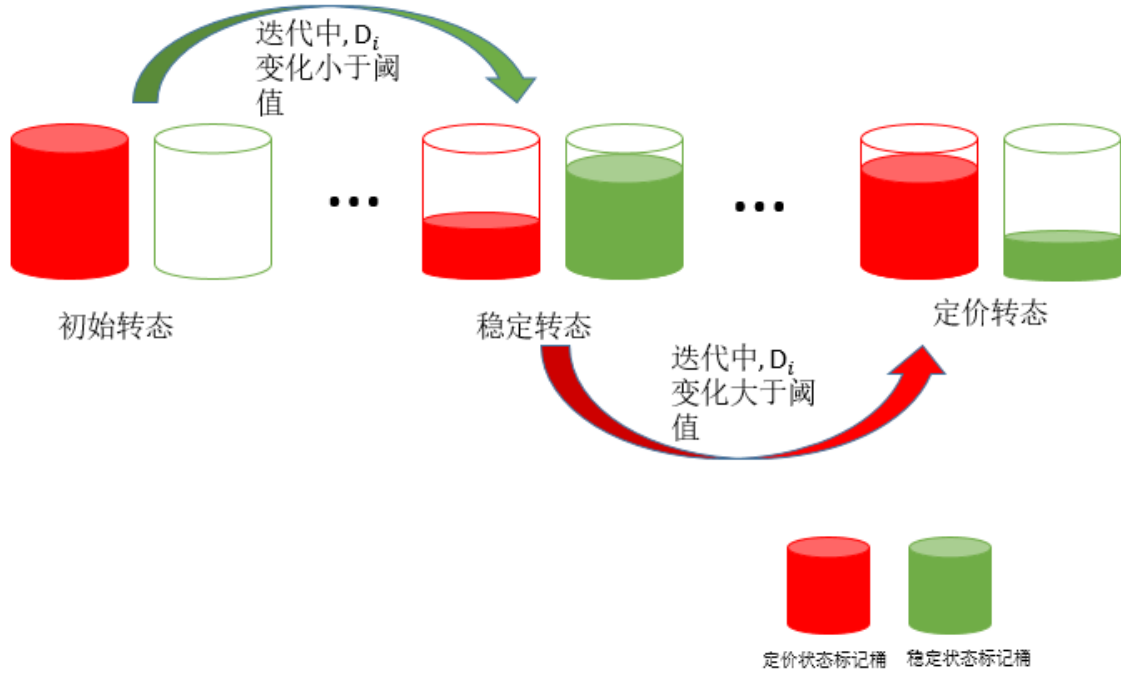
$A = \{\alpha_i, i \in [1, 2, \dots, R]\}$ 是所有历史任务执行情况集合， β 是当前市场状态。然后,根据 $|\delta(D_i)| = |D_i - D_{i-1}|$ 按照如下公式更新 β_{dy} , β_{sd} ,

$$\beta_{dy} = \begin{cases} \max\{\beta_{dy} + \varepsilon, 1\} & \text{if } |\delta(D_i)| > D_{th} \\ \min\{\beta_{dy} - \varepsilon, 0\} & \text{else} \end{cases}, \quad (3-3)$$

$$\beta_{sd} = \begin{cases} \min\{\beta_{sd} - \varepsilon, 0\} & \text{if } |\delta(D_i)| > D_{th} \\ \max\{\beta_{sd} + \varepsilon, 1\} & \text{else} \end{cases}, \quad (3-4)$$

当前市场状态 $\beta = \max\{\beta_{dy}, \beta_{sd}\}$.

图 3-4 展示了各轮之间转态桶的变化情况和转态转移示意。



根据 D_i ，我们将总预算分配到每轮任务的子预算中。假设总预算为 B ，总轮数为 R ，剩余可用预算为 B' ，剩余轮数为 R' 。根据 D_i 的情况， $D_i > 1$ 代表增加下一轮预算；相应的， $D_i < 1$ 代表减少下一轮预算；当 $D_i = 1$ 时下一轮预算较上一轮预算持平。为了避免赤字，我们将 $B_i \times D_{i+1}$ 和 $\frac{B'}{R'}$ 比较来确保下一轮预算不超过剩余可用平均预算。

$$B_{i+1} = \min \left\{ B_i \times D_{i+1}, \frac{B'}{R'} \right\}, \quad (3-5)$$

然后将 B_{i+1} 分为两部分，固定的基础部分 f_i 和变化的奖金部分 b_i ，根据一定的分配比例将每轮子任务的预算分配到参与者，下一节具体说明。

为了减少任务开始阶段收敛到合适预算的轮次，作为以上算法的改进，提出分时隙的预算递增算法来动态分配预算，初始时将总预算均分到每轮感知任务中，每轮感知任务的固定奖励部分预算按如图 3-5 所示分配，

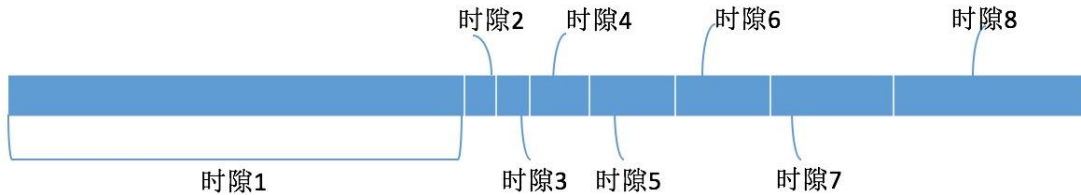


图 3-5 将任务开放时间分成时隙(上图中为 8 个时隙)，初始时分配一定的预算，随着时隙的增加如果任务没有完成，逐渐增加预算分配，到最后一个时隙时预算全

部投入。由于用户选择机制采用先到先得的方式，参与者之间存在竞争关系，在博弈条件下参与者不会出现集体等到最后一个时隙来完成任务的情况，通过此改进算法预算投入能更节省，每轮任务的预算投入还是根据上文的算法来动态确定，根据本轮任务执行情况和历史任务执行情况结合当前所处定价阶段来决定下一轮预算。

3.3.4.2 每轮任务执行时子预算的分配

当平台发起第 i 轮任务时，平台给所有潜在参与者广播任务描述信息。任务描述信息给出固定部分金额，同时告知参与者其可以获得与其贡献成比例的浮动部分奖金。

参与者根据任务描述信息选择参与此次活动或者忽略，平台检查上传的感知数据，直接发放固定金额的激励部分。参与者可以检索平台的感知数据收集情况来调整自己的参与策略来获取更大的收益。当平台获取到足够的感知数据或者截止时间到达，平台终止任务的执行流程并推送浮动的奖金部分给参与者。子预算的构成如图 3-6 所示，

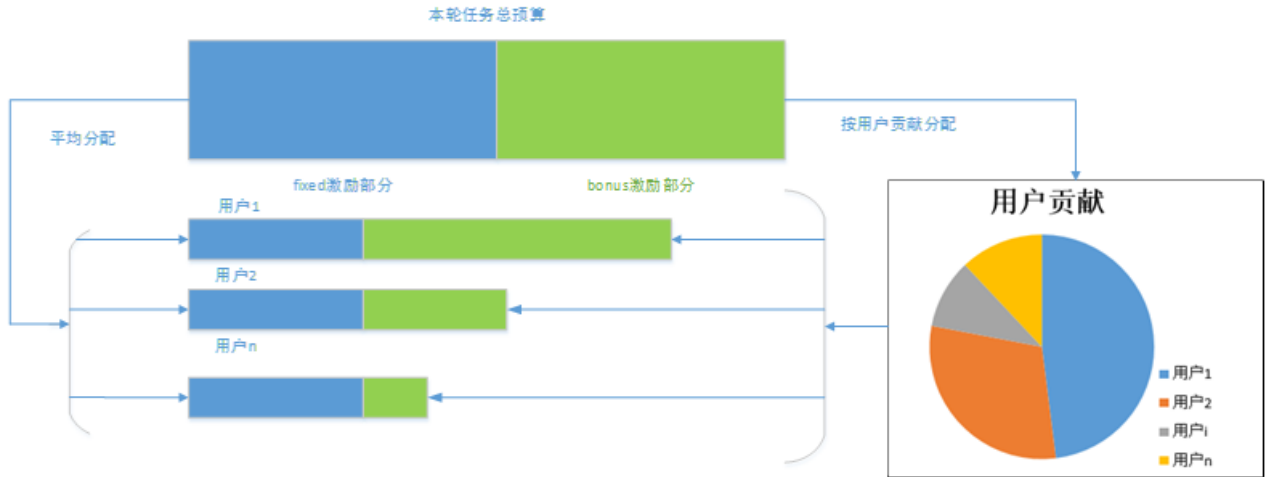


图 3-6 每轮任务激励分配方式

对于一个执行流程， B_i 是第 i 轮的子预算，固定价格部分预算 f_i ，浮动奖金部分 b_i 可以分别由如下公式得到，

$$f_i = a \times B_i, \quad (3-6)$$

$$b_i = (1 - a) \times B_i \times \frac{n_i}{n}, \quad (3-7)$$

a 是一个常数参数，用来调整固定部分和浮动部分的分配比例。

成功的参与者会立即获得固定金额部分的激励 $f_{is} = \frac{f_i}{n}$ ，当任务完成时获得浮动奖金部分，

$$\rho_{is} = \frac{U_0(d_{is})}{\sum_{s' \in S} U_0(d_{is'})}, \quad (3-8)$$

$$b_{is} = b_i \times \rho_i, \quad (3-9)$$

ρ_{is} 是某个参与者的贡献占有所有参与者的贡献比例，浮动奖金和其贡献比例成正比。

d_{is} 是用户 s 上传的感知数据， S 是所有参与者集合。系统效能函数 U_0 可以表示为及时性、空间性或完整性等参数的函数。

动态分配预算的激励机制继承了固定价格机制、数据上传优先激励机制的参与者和平台交互简单的优点，不需要用户报价和竞价以避免定价压力、竞价博弈开销，通过参与者的参与行为作为反馈渠道，自适应地收敛到合适的激励报酬价格，为了优化收敛过程，设置了定价状态、稳定价格状态和状态转移的算法，此方案适合于绘制城市级别的感知数据地图等参与式感知应用场景。

3.4 参与式感知激励机制仿真实验

3.4.1 仿真场景建模

为了评估激励机制的效果，在现有参与式感知实验平台开发还没有完成的情况下，采用仿真实验，建模来模拟用户行为，对比三种激励机制的效果，为在平台中设计和实现激励机制提供理论依据。

将北邮附近的一块区域作为感知数据绘图区域，将其分成 10×10 正方形的小块子区域，每块子区域每天需要拍摄三张照片来计算此区域的 $\text{pm}_{2.5}$ 值，本次感知活动持续两个月（10 天），总预算有 8333 元人民币。

通过传单等方式招募了 360（按照期望的参与人数 3×100 的 1.2 倍招募）名潜在参与者，用户开始时均匀的随机分布于 10×10 的子区域中，参与者随机分布于感知区域中，参与者对激励的反应分为两类，兴趣驱动型和利益驱动型。兴趣驱动型主要被感知活动的社会意义、自身的兴趣所吸引，感知开销的付出回报比值在没有达到一个较高的退出阈值之前不会离开感知活动；利益驱动型的参与者，感知开销付出回报比值退出阈值较低，如果累计收入低于预期值即开销回报比例高于退出的阈值，参与者将退出感知活动。同样，每轮任务结束后，平台会把当前数据采集情况广播给所有参与者，利益驱动型参与者会以较大概率转移到高价的感知区域，但是转移会增加其感知开销，兴趣驱动型参与者受激励价格影响较小，出于兴趣驱动或公益心理会以较小的概率转移到数据缺失区域，同样会增加其感知开销。每个区域采用先到先得的方式，模拟中采用随机参与者选择机制来模拟先到先得，未被选中的参与者不会产生感知开销，但是从原区域专门赶来的参与者会产生移动开销，累加到总计感知开销中。下面分三种激励形式详细阐述仿真设置。用户的感知开销服从正太分布，且附加一个很小的增长因子来模拟费用的不断上涨，兴趣性参与者和

利益驱动型参与者各占一半。仿真通用参数统计如表 3-3，

表 3-3 仿真参数表

总预算	8333 元	
持续时间	10 天	
空间范围分区数	10*10,100 个子区域	
子区域所需数据份数	3 张图片/天	
初始募集人数	360 人	
人群分布	随机分布，每天分 8 个时隙移动位置	
参与者参与反应模型	兴趣型	每次感知任务均要有利可图
	功利型	回报付出比低于 50% 不参与
用户选择方式	先到先得（随机选择）	
用户开销设定	用户拍照开销服从正太分布，均值 2.5 元/次，标准差 0.75	

3.4.2 对比实验设置

3.4.2.1 简单的平台定价激励方式设置

平台通过海报、传单、网络等方式招募所需的参与者，根据实际需要人数的 1.2 倍招募，例如有 100 个子区域需要执行感知任务，每个区域需要 3 人完成，则需要 $3 \times 100 \times 1.2$ 共招募 360 人参与。平台为每次照片上传给予固定的激励报酬，为了确定固定报酬的价格，假设通过问卷调查等统计方式获得了潜在参与者的保护价格工资（感知开销）的分布，如图 3-7 所示，根据累计概率密度达到 83.3% 的报酬价格（因为按 1.2 倍招募的用户），即可覆盖所需的目标人数，如图 3-8，固定价格为 3.22 元每份数据。此方案存在的问题是可能随着保护价格工资的上涨，平台没有及时提高激励价格，使得利益驱动型参与者退出感知活动。

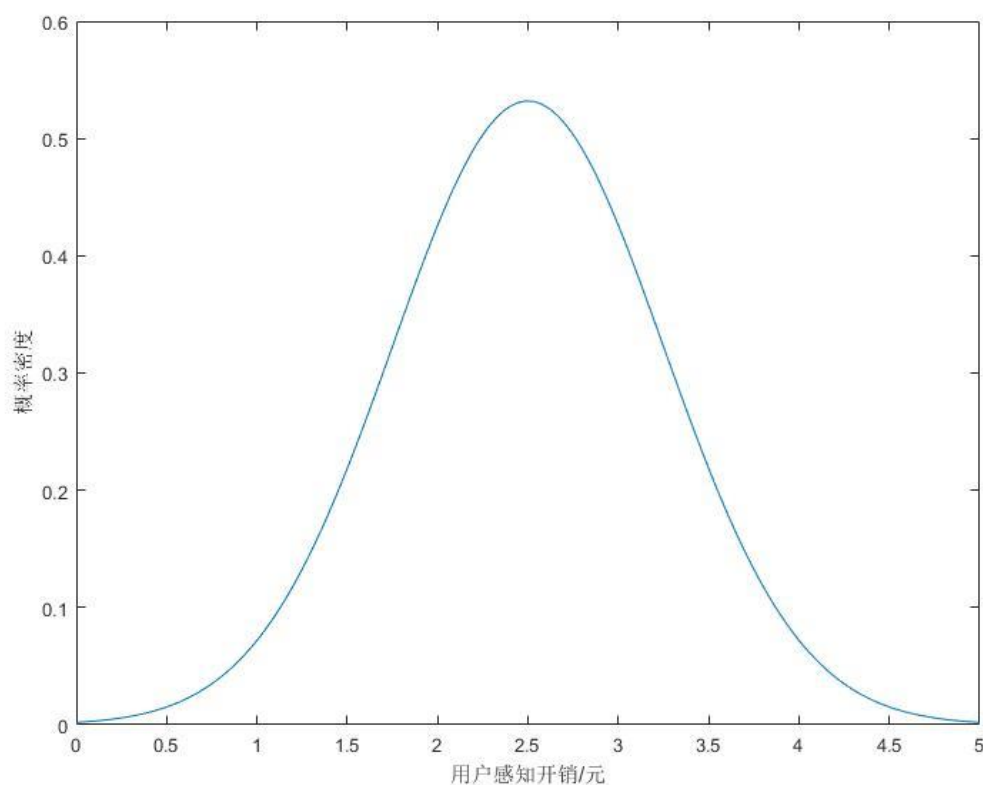


图 3-7 用户感知开销分布

确定了固定价格激励机制中的激励价格，仿真实验中共采集 10 轮照片数据，360 名参与者随机分布于 100 个格子中，每轮任务每个格子采集 3 张照片，每轮任务分为 8 个时隙，参与者在每个时隙开始时随机移动到相邻格子中（或者不移动）以模拟参与者的移动性特点。每个时隙中选择参与者时，在每个格子中选出潜在参与者，即参与者类型为兴趣型且付出回报率没有高于阈值，或者功利型的参与者但其感知开销低于报酬价格，在所有潜在参与者中随机选择剩余照片数量的参与者来执行任务模拟先到先得的用户选择策略。

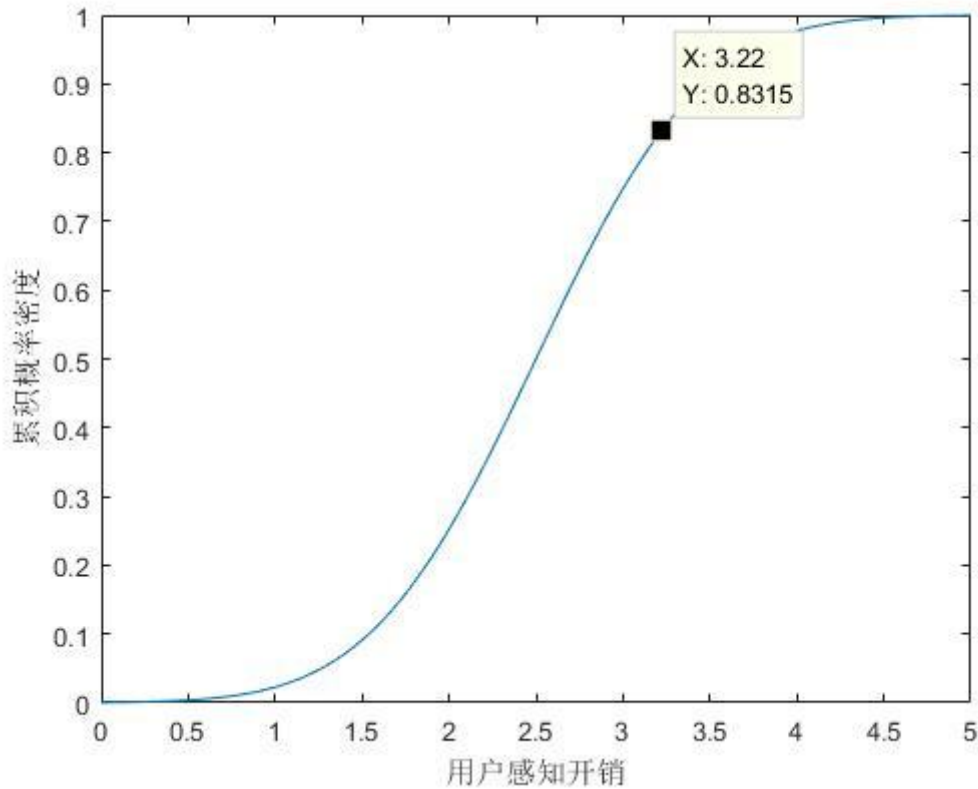


图 3-8 固定价格报酬的确定

3.4.2.2 基于逆向竞拍的激励机制设置

每一轮任务采集时，潜在参与者上报自己的感知开销（保护价格工资），平台选择保价最低的不超过三个用户采集本区域的数据，并支付相应的报酬。此方案存在的问题是保价较高的潜在参与者由于竞争失利，累计参与成本得不到激励回报，会退出感知过程。此激励机制方式的仿真过程和固定价格激励机制类似，只有在用户选择阶段，任务用户会真实报价，同一个格子中选择报价最低的所需照片数量的参与者来执行任务。

3.4.2.3 动态预算分配的激励机制设置

由上文介绍的，动态预算分配的激励机制方案中，初始化时将每轮任务的预算设置为总预算的平均分配水平，任务执行状态设置为定价阶段，之后根据每轮任务的完成情况更新任务执行阶段状态标记，根据标记计算出预算变化因子，得出下一轮任务的预算。整个任务执行过程会在定价阶段和稳定价格阶段相互转换。动态预算分配的激励机制可能遇到的问题是，在定价阶段由于价格不合理，需要一定轮次的任务执行过程才能收敛到合适的价格，这个过程可能导致感知数据达不到期望标准，一种解决的思路是适当增加感知任务执行轮次，类似于现实生活中餐馆的“试营

业”；本仿真实验中采用另一种改进方案，在第一个时隙中设置预算为本轮任务的可用预算的一半，如果任务没有完成进入下一个时隙，增加可用预算为剩余一半预算中的 $1/n$ ， n 为总的时隙数，使得进入最后一个时隙时本轮任务预算增加完毕全部投入。这种改进不需要适应市场环境的任务执行，能更好的保证收集到的数据数量，只需要根据市场情况动态调整每轮子任务的预算分配和各个时隙第一次预算投入占预算的比例即可，而这个动态调整的方式可以继续使用上文动态分配预算的流程。

3.4.3 性能比较

在仿真初始化过程中，按正太分布随机确定 360 名参与者的感知开销值，均值为 2.5 元，标准差为 0.75，然后将参与者随机分布到 100 个格子中，此条件存储下来为三种激励方案公用。如下对实验结果按照平台角度和参与者角度分别作出三种激励机制的效果对比。

3.4.3.1 平台角度

从平台角度主要考察总预算支出情况、总数据量和完成三份数据采集的格子分布三项指标，分别代表预算使用效率、数据数量和数据分布情况。

● 总预算支出情况

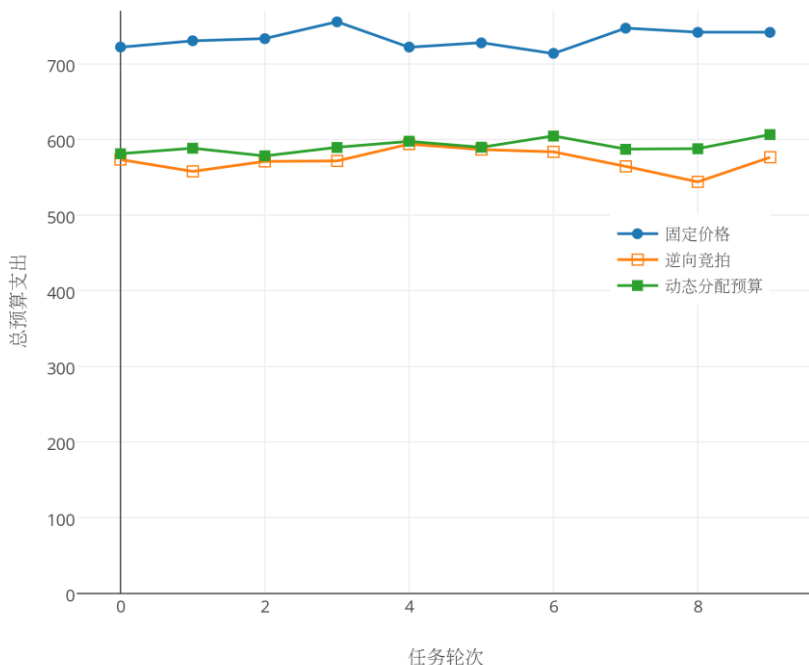


图 3-9 每轮任务的感知预算支出情况

三种激励方式完成同样的感知任务激励目标，10 轮任务分别支出预算情况如图 3-9，从图中可以看出，固定价格激励机制的预算开销最大，没有利用不同参与者感

知开销不同的特点来节省预算，每个参与者完成一次感知任务获得报酬一样多；在仿真中逆向竞拍的激励机制中，参与者都是真实报价，所以感知开销是理论上的最低值，而动态价格激励机制与其有类似的预算节省优势。分析动态价格激励机制之所以能节省预算，在于其预算是随着时隙的增加不断增加，而增加的预算可以激励用户转移到相邻格子中执行感知任务，使得用户移动性更有利于感知数据的收集。

● 总数据量

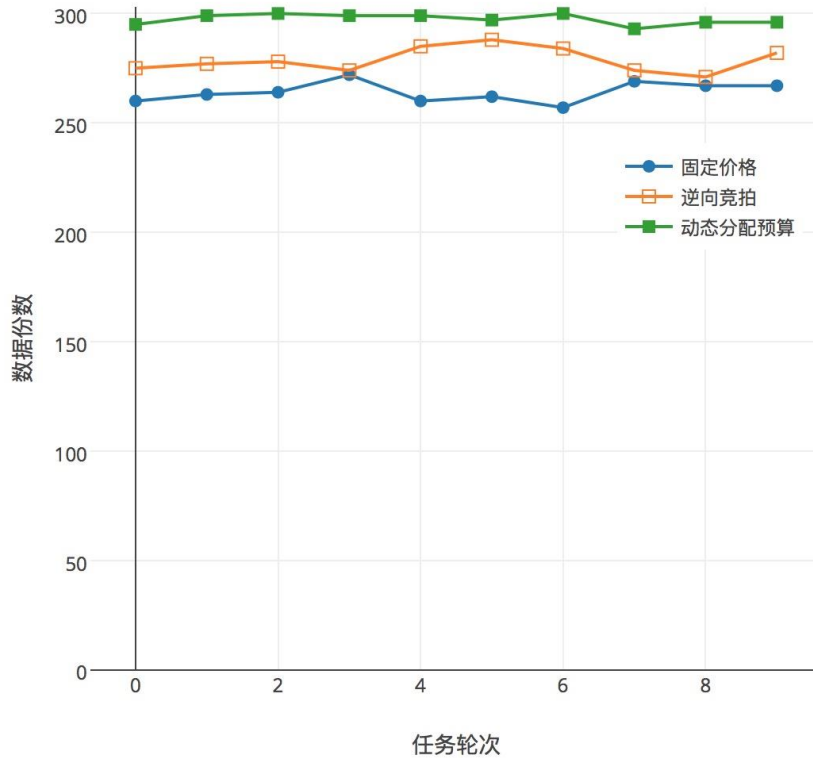


图 3-10 每轮任务采集的数据总量

由图 3-10 分析，动态价格激励机制收集了最多的感知数据，而固定价格激励机制性能最差。因为用户移动具有随机性，每个格子中的用户是否参与感知活动取决于其报酬价格能否抵消其开销，固定价格机制最不灵活，收集数据量最少；逆向竞拍机制能满足开销较高的参与者，但是其不能激励用户转移到数据缺失区域，因为参与者执行所有任务都是其感知开销相抵的报酬价格；动态预算分配激励机制中，随着时隙的流逝，未完成任务的报酬不断增加，有理由激励更多的用户转移到此区域中完成感知任务。

同时在仿真过程中注意到，如果不对激励用户移动做很好地处理，吸引了过多的用户到达一个区域，反而会造成数据采集效果变差，因为一个区域只需要三张照片，所以如何避免用户过度集中到一个数据缺失区域中，本文采用预约机制，区域已满参与者则选择周边其他激励区域或随机行走。

● 完成三份数据采集的格子数量

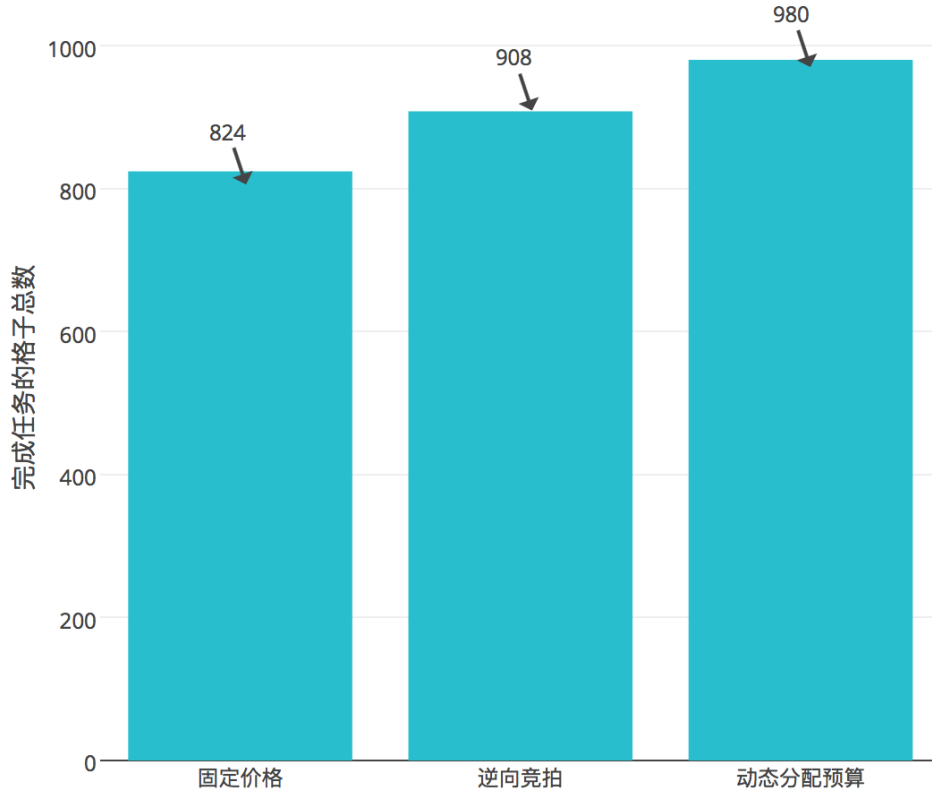


图 3-11 每轮任务采集的数据总量

对于感知数据质量的比较，实验中采用完成采集三份数据的任务数量来统计，每个区域完成了采集三份数据，证明其可以根据这三份数据综合出具有代表性的质量可靠的感知数据观测值，图 3-11 给出 10 轮任务总共完成 3 份数据采集的格子总数；另一方面，完成任务的区域数量越多，证明数据分布情况越好，高质量的数据的均匀分布，正是环境监测类参与式感知应用的目标，图 3-12、3-13 和 3-14 为三种激励方式在第五轮任务采集过程中完成 3 份数据采集的格子空间分布图，由上图可以看到，动态分配预算激励机制性能最优，其次是逆向竞拍和固定价格激励机制。

3.4.3.2 用户角度

从参与者角度来看参与结果，主要查看总参加人数、总收益情况两方面，代表激励机制的公平性和吸引程度。

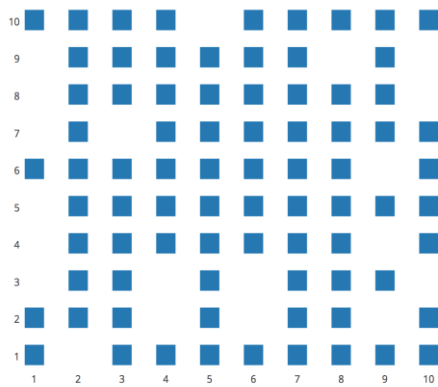


图 3-12 每轮任务采集的数据总量

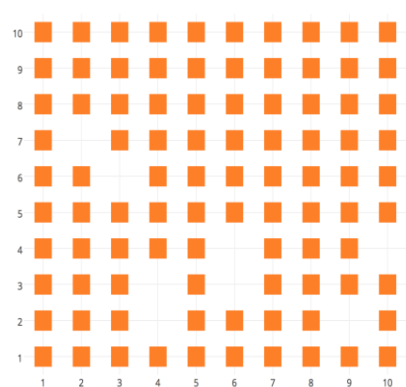


图 3-13 每轮任务采集的数据总量

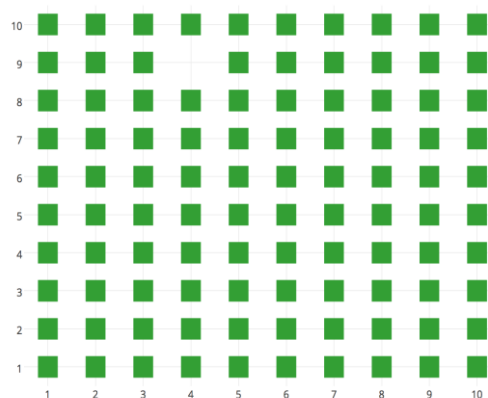


图 3-14 每轮任务采集的数据总量

● 总参加人数

考量参与到感知活动中的总人数，能体现激励机制设计的是否具有公平性，能吸引更多用户参与才能维持感知活动的长时间健康运行。理论上逆行竞拍的参与程度应该最低，因为总是选中报价最低的参与者，但是仿真中由图 3-15 看到，逆向竞拍的参与者反而最高，因为我们限制每个格子的用户在一起竞拍，所以竞争性没有那么明显，平均每个格子有 3.6 个参与者，而需要 3 份感知数据，所以逆向竞拍的公平性没有显示出劣势，这对以后的激励机制的设计给出启发，可以限制竞争程度来提高公平性，也能充分利用逆向竞拍机制的优点。动态预算分配每轮任务选择的用户数最少，平均每个被选择的用户都执行了两份以上任务，看似公平性不足，但 10 轮任务总的参与者数量是最高的 270 人，所以其用户参与效率最高，其中原因还需进一步分析。

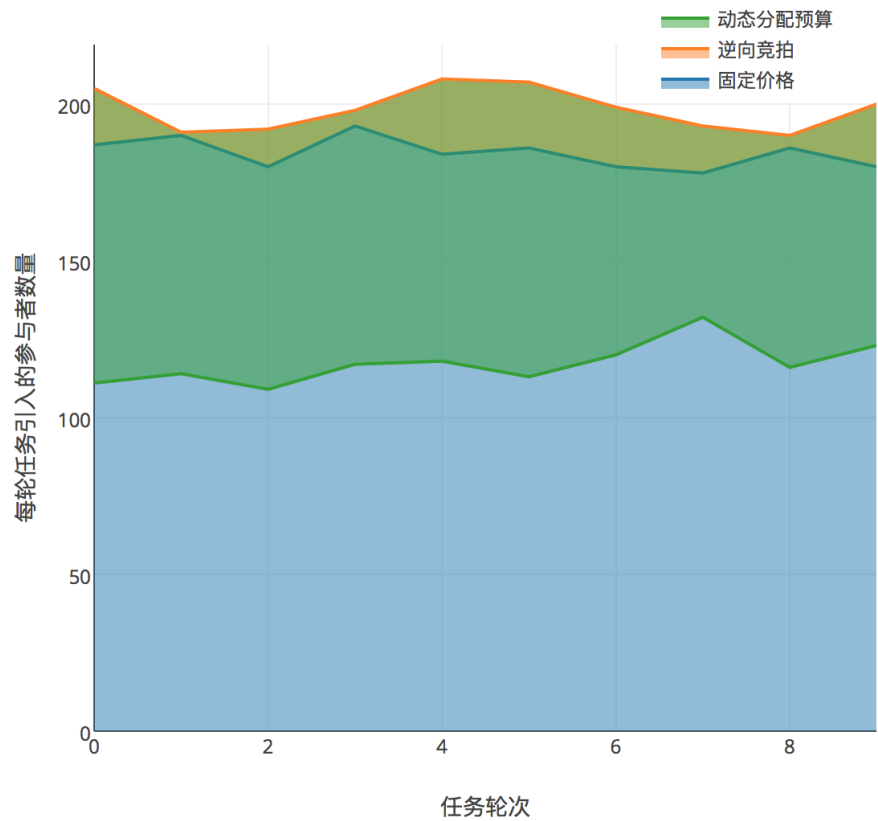


图 3-15 每轮任务选择的参与者数量

● 总收益情况

表 3-4 总收益与人均收益情况

激励类型	总参与人数	总激励发放金额	人均收益
固定价格	237	7336.10	30.95
逆向竞拍	258	5723.08	22.18
动态预算分配	270	5893.91	21.83

人均收益体现出固定价格激励机制最让利于参与者，但是大部分利润都由少数感知开销最低的用户分得，其公平性并不是最优。

3.5 本章总结

本章为参与式感知中激励机制的理论研究，首先在引言中阐述了参与式感知环境下激励机制的重要作用；之后简要总结现有文献对于参与式感知中激励机制的研究方向，其解决的问题和采用的方案，主要有固定价格和基于逆向竞拍的激励机制解决预算分配的问题、在预算受限条件下预算的最优化利用策略、激励机制促进数据的均匀分布和增强激励机制的时效性等主要研究方向；在第三节中，重点介绍本文创新提出的一种动态分配预算的激励机制，其设计原则、主要特性和核心的动态

分配预算算法；在最后，通过设置仿真实验，将动态分配预算激励机制与固定价格激励机制和基于逆向竞拍激励机制做对比，验证了其在节省预算、优化数据分布和简化交互流程等方面的性能优势，也通过仿真实验改进了激励方案，为后文参与式感知平台激励系统的设计和实现做好了理论准备。

第四章 参与式感知平台激励系统的需求分析和总体设计

4.1 功能性需求分析

本人主要负责参与式感知实验平台的激励机制的设计和实现，从最终的功能效果来看主要包括，任务发布者可以通过平台的激励系统发布感知任务，具体有指定感知任务的位置、指定激励机制类型（包括固定价格、基于逆向竞拍的激励机制和周期的动态预算分配的激励机制）；参与者可以通过激励系统参与到感知活动的数据采集过程、获得感知激励报酬。

图 4-1 为激励系统的用例图，下文对每个用例需求详细描述。

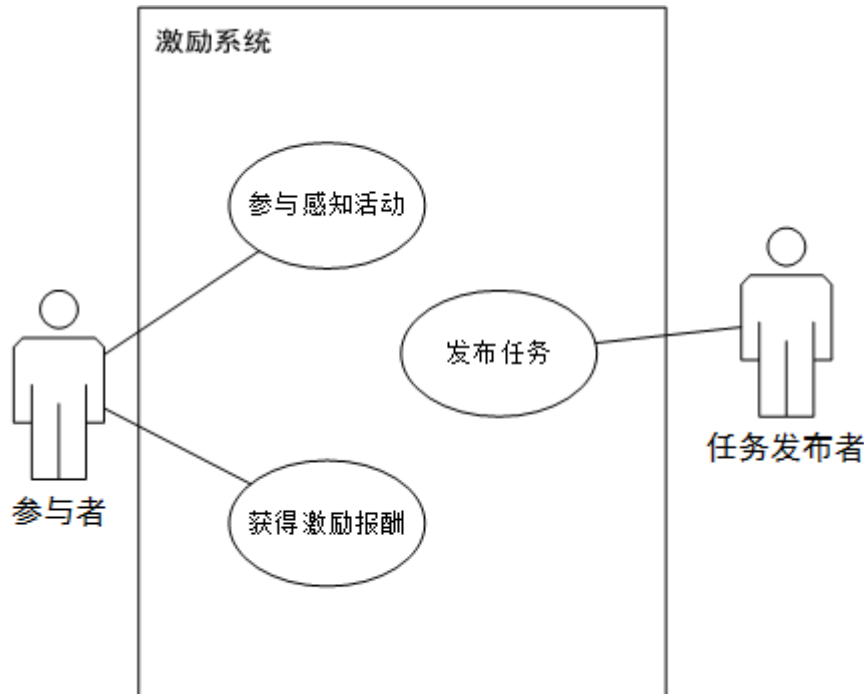


图 4-1 任务管理系统用例图

4.1.1 发布任务

任务发布者作为激励系统的一方交互者，有通过激励系统发布任务的需求。具体流程包括通过 Web 界面或移动客户端配置任务信息，包括感知任务所要采集的感知数据描述信息、所投入的用于激励的预算、激励机制类型选择等信息，之后发布一个感知活动。平台根据预算和任务发布者选择的激励机制类型，确定任务描述信息，根据任务的位置要求，向轨迹系统查询任务所在区域的潜在参与者信息，广播任务给相关潜在参与者。如果任务发布者没有成功提供预算，任务发起失败；如果没有轨迹信息和任务匹配的潜在参与者，广播任务描述信息给全体潜在参与者。

本课题依托的参与式感知数据收集实验平台主要采集照片、光照、噪音三类数据，每个感知活动持续一段时间，周期重复采集数据以绘制感知数据地图，任务发布者指定一块目标区域之后，用格子将区域分为子区域，每轮感知任务执行需要在每个格子收集指定份数的数据，周期重复此过程。

发布任务需求具体包括任务信息配置、根据位置信息获得潜在参与者列表、推送任务信息三个子需求。任务信息配置，主要配置任务的激励机制类型、任务重复执行周期、格子划分、感知数据类型和数量等参数；根据位置信息获得潜在参与者列表，需要和轨迹统计和预测系统交互，输入任务位置信息，返回潜在的参与者信息列表；推送任务信息是调用推送功能，将任务描述信息推送给适合的潜在参与者或者做广播推送给全体参与者。

4.1.2 参与到感知活动

参与者通过激励系统可以参与到感知活动中，具体流程包括收到任务推送信息或主动查看可以参与的感知活动任务列表，根据激励机制类型的不同通过不同方式获得参与感知活动的资格，执行感知任务采集感知数据，上传感知数据完成任务的执行过程。

本课题依托的实验平台中，主要根据用户位置来推送任务信息，用户查看可以参与的感知任务列表也是位置相关的；支持平台定价和基于逆向竞拍两种激励方式，平台定价方式采用先到先得的用户选择策略，用户锁定任务成功即可执行任务，基于逆向竞拍的激励机制需要一个竞价时间窗口，竞价结束时竞价成功的参与者获得执行感知任务的资格；上传感知数据时同时上传用户信息和任务信息以用于用户激励报酬的结算。

对于平台定价激励机制中的锁定任务，感知任务参与者处于任务详情界面，任务无需竞价环节，还需要召集参与者执行感知任务，参与者有意愿执行此感知任务，尝试锁定该任务，锁定成功，给用户提示信息，任务剩余参与者数量减一，如果并发锁定失败，提示失败信息。对于基于逆向竞拍的激励机制的任务，用户浏览任务详情信息，任务处于报价环节，还需要召集参与者执行感知任务，参与者上报竞拍价格，报价时间结束，系统通知用户竞拍结果。

4.1.3 获得激励报酬

参与者完成感知任务，成功上传感知数据后应获得相应的激励报酬，固定价格激励机制中激励报酬为固定值，基于逆向竞拍的激励机制为用户成功竞拍的报价值。参与者成功上传数据后，根据任务的激励机制类型，得到相应的激励报酬价格，更新用户账户金额，预算剩余金额减少相应值。

4.2 非功能性需求分析

4.2.1 可用性

可用性主要体现在当大量用户并发访问时，平台激励机制的运行情况，主要表现在锁定任务要处理好并发请求，逆向竞拍也要考虑有大量用户同时参与竞拍的情形。对于通常的并发请求设计，本平台也可以实现由 nginx 做反向代理实现负载均衡，增加冗余的应用服务器 Tomcat 来处理高并发请求。

4.2.2 安全性

安全性需要考虑用户隐私数据的安全，账号密码的安全，本系统采用 https 来实现安全通信。

4.2.3 可扩展性

系统采用 Spring 框架来增强可扩展性，按照 Spring 框架的控制反转和依赖注入思想，实现模块之间的松耦合，便于之后系统的扩展和重构。

4.3 参与式感知平台激励系统的总体设计

4.3.1 参与式感知平台激励系统总体结构

图 4-2 给出参与式感知数据收集实验平台系统功能层次划分图，最靠近用户的是 Jersey 实现的 Web 服务接口，在概要设计章节已经确定。JAX-RS, Java API for RESTful Services, 是 Java 语言为开发 RESTful 风格 Web 服务提供的编程接口，包括 javax.ws.rs、客户端、容器、核心、扩展五个包，为使用 Java 语言开发 RESTful Web 服务提供了便利条件。开源社区也开发了众多的符合 JAX-RS 标准的框架，其中 Jersey 就是一款优秀的产品级、开源的作为 JAX-RS 标准实现的框架。

业务逻辑层采用了 Spring 框架，可以使用 POJO (Plain Old Java Object) 来实现 model 类和业务逻辑；在持久层抽象出对数据库的访问接口，在概要设计中也作出约定，有持久层具体去实现对数据库的访问，无论数据库采用 MySQL 还是非关系型的 HBase，对上层的业务逻辑都不影响。

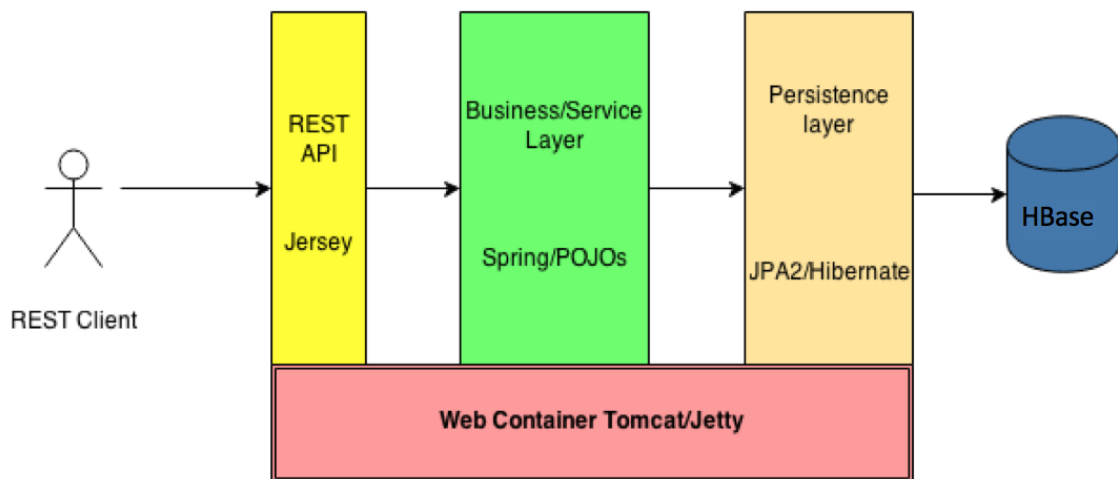


图 4-2 参与式感知平台系统功能分层

4.3.2 参与式感知平台激励系统与其他系统的交互

激励系统加入到参与式感知数据收集实验平台，主要起到促进感知数据收集的作用，同时也能测试不同激励机制对感知活动的促进效果。激励系统需要与其他系统配合，存在调用与被调用的关系。图 4-3 展示激励系统与其他系统的交互关系。

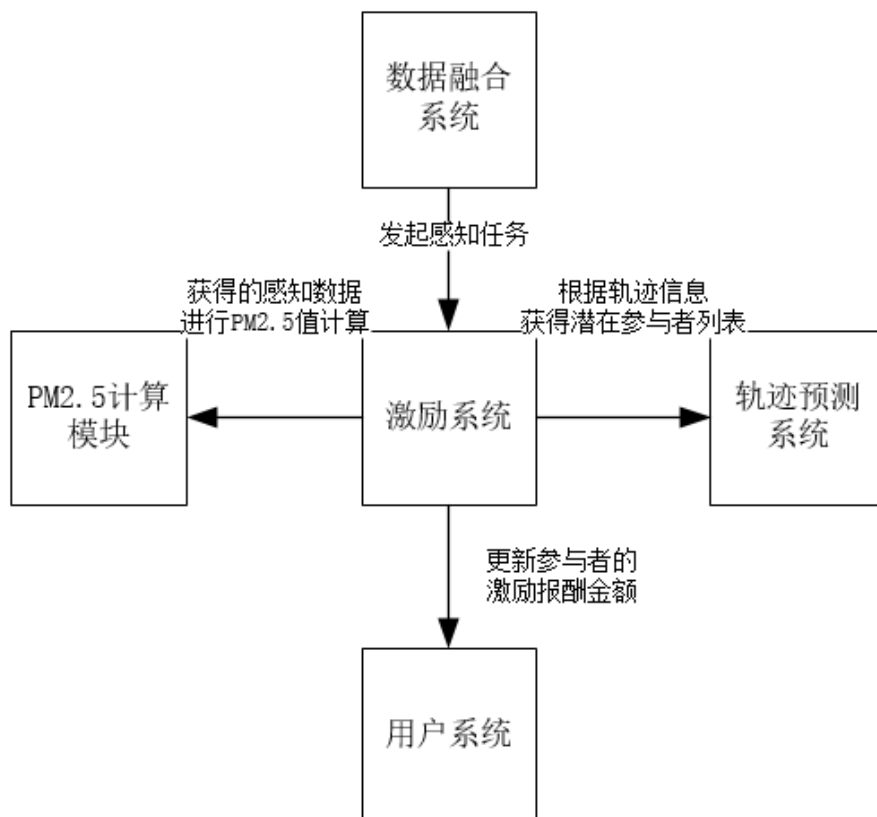


图 4-3 激励模块与其他模块交互图

激励系统除了本身受任务发布者调用发起感知活动外，也可能受到数据融合系

统、数据可视化系统等的调用，在其他系统发现某一区域感知数据缺失而无法提供可靠的数据融合或数据展示要求时，都可以调用激励系统发布感知任务。激励系统对外提供统一的接口来发布感知活动。

激励系统在发布感知活动时也要调用其他系统提供的功能，在确定潜在参与者列表时调用轨迹预测系统，传入位置信息返回参与者列表；在参与者完成感知任务时调用用户系统提供的功能修改用户账户金额；获取到感知数据后进行数据存储并交给 PM2.5 计算模块，计算空气质量估计值，在详细设计时确定各系统或模块间的接口设计。

4.3.3 参与式感知平台激励系统模块划分

根据以上的需求分析，将激励系统的功能划分为模块，模块及子模块的划分如图 4-4 所示。

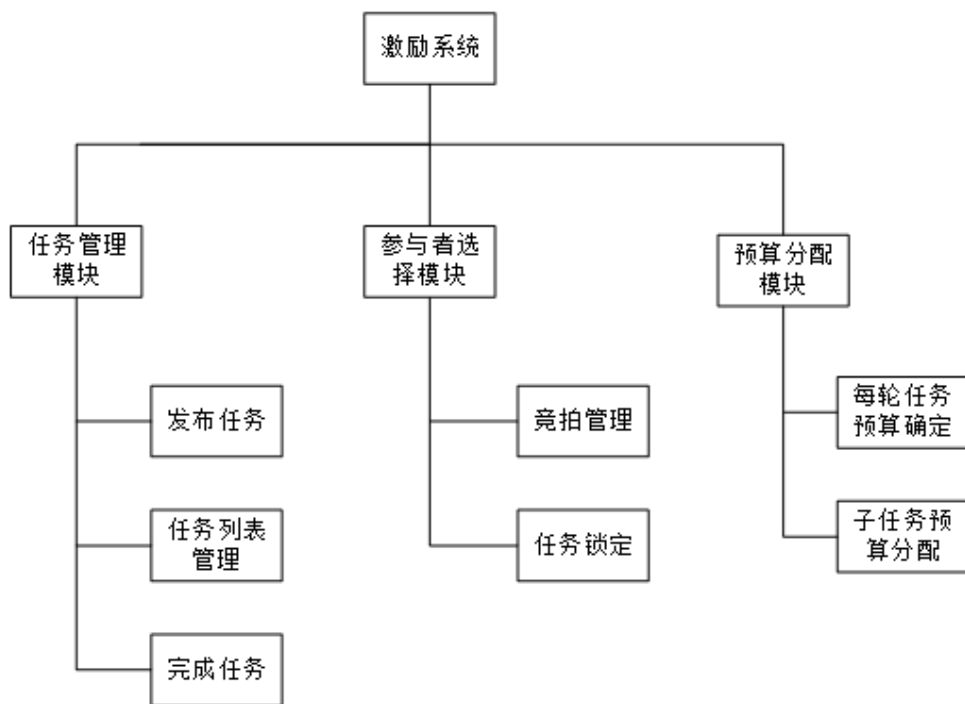


图 4-4 激励系统模块划分

激励系统分为三个模块，包括任务管理模块、用户选择模块和预算分配模块。任务发布者将参与式感知任务的任务描述信息和预算信息输入系统，系统通过任务管理模块和激励分配模块的功能，发布每一轮任务的详细信息，通过参与者选择模块召集参与者完成感知任务，最后输出参与者贡献的感知数据集和激励分配结果。具体每个模块的功能和接口设计分别阐述，我负责每项功能和移动客户端、Web 界面的接口实现，对数据存储提出需求，由其他同学负责 HBase 数据库的数据存储具体设计，所以接下来的小结会确定与客户端、Web 端的接口设计（只列出接口设计的核心部分，返回数据格式等信息可以查询接口文档），确定对数据存储的需求。

4.3.3.1 任务管理模块

任务管理模块包含发布任务、任务列表管理和任务完成逻辑的处理三个子模块。

发布任务为任务发布提供了统一的接口，通过接受感知活动的配置信息，生成任务描述信息，进而周期的生成每轮子任务，将每轮任务的具体信息推送给潜在参与者，完成任务发布过程。一个参与式感知活动需要配置预算信息、任务执行次数、数据份数、每次开始时间、时间间隔、任务的空间覆盖范围信息等，用户还要选择所采用的激励方式种类，可供选择的有平台定价激励机制、基于逆向竞拍的激励机制和动态分配预算的激励机制三种，如果选择了动态分配预算的激励机制，还有可选的配置选项提供给任务发布者进行更细粒度的任务发布管理。表 4-1 确定激励任务管理模块任务发布的接口格式。

表 4-1 任务发布接口参数表

URL	/task	HTTP 方法	post
参数	类型	说明	
budget	double	参与式感知活动的总预算	
roundNumber	int	参与式感知活动总轮数	
dataItmPerRndPerGrd	int	每轮任务每个格子收集数据份数	
gridNumber	int	格子数量	
deltaDegreeGrid	double	格子边的经纬度差值	
latitude	double	感知区域中心点纬度	
longitude	double	感知区域中心点经度	
incentiveType	int	枚举值，1 代表固定价格激励制度、2 代表基于逆向竞拍激励制度、3 代表动态分配预算激励制度	

URL 为 task 符合 RESTful 风格的以资源作为 URL 的设计思想，post 表示生成一个任务，即发布一条新的感知活动信息。对于数据存储的需求，只需写入数据库一条任务描述信息记录，

```

//*****发布任务数据存储需求*****
Boolean addTaskDescription(TaskDescription newTaskDescription);
//*****发布任务数据存储需求*****

```

TaskDescription 为包装了任务描述信息属性的类（POJO），返回值标志操作是否成功。

任务列表管理，客户端通过此接口请求参与式感知平台当前正在招募用户的感知任务列表，也可以查看用户个人的正在竞价的任务、已锁定任务和已完成任务的

任务列表。由于参与式感知需要大量不同用户执行感知任务，通过不同用户采集到的感知数据进行互相校验以确保感知数据的真实可靠，不希望同一个参与者执行一个感知任务多次，因为同一用户的操作习惯在多次感知操作中趋于一致，且手机的感知精度不变，较短的时间、空间间隔内贡献的感知数据差异非常小，所以本系统限制每个单独的参与者只可以执行一次感知任务，在查询可执行感知任务列表时必须是在已登录状态，已经参与的感知任务不会再呈现给参与者。参与者通过提供自己的位置信息，如果未打开手机的定位服务造成无法定位，则由平台根据用户轨迹系统推测用户所处位置信息，根据位置信息查询正在进行的感知任务列表，以 JSON 格式返回给用户。潜在参与者查看了正在执行的任务列表，可以对平台定价任务进行锁定，或者对于逆向竞拍任务参与报价竞拍，锁定成功或竞价成功的用户获得执行感知任务资格，客户端通过对此接口的请求，返回已获得感知任务执行资格的任务列表。

表 4-2 确定任务管理模块提供的获取任务列表接口格式。

表 4-2 获取任务列表接口参数表

URL	/task/list	HTTP 方法	get
参数	类型	说明	
latitude	double	感知区域中心点纬度	
longitude	double	感知区域中心点经度	
userID	string	用户 ID	
passToken	string	用户登录获得的 token	
beginItemIndex	int	开始的数据项索引，用于分页展示	
maxItemNum	int	最大返回数据条目	
tasklistType	int	枚举值，1 表示新任务、2 表示待完成任务、3 表示竞价中任务、4 为已完成任务	

HTTP 方法采用 get，为获取资源；对于数据存储的需求有

*****获取任务列表数据存储需求*****

//任务执行阶段枚举类型

```
public enum TaskState{
```

```
    TaskStateLocking, TaskStateBidding, TaskStateCompleting, TaskStateCompleted,
    TaskStateUncompleted, TaskStateAll
}
```

//按任务执行阶段和用户 ID 检索任务

```
Set<Task> queryTask(String userID, TaskState taskState);
```

//按地理位置检索正在执行的任务

```
Set<Task>queryExecutingTask(double lon, double lan, int maxItem);
```

```
//*****获取任务列表数据存储需求*****
```

对于查询待完成任务列表、竞价中任务列表、已完成任务列表只需调用一次 queryTask 函数传入相应的 TaskType 枚举值即可，对于新任务列表，需要调用 queryExecuting 函数得到本区域正在执行的任务集合，减去 queryTask(userID, TaskTypeAll)函数返回的以参与任务集合，即得到可以参与的新任务列表。

任务完成逻辑，是参与者上传感知数据、完成感知任务获取激励报酬的业务逻辑。获得了执行感知任务资格的参与者，需要在任务描述信息中任务完成时间段内执行感知任务，包括到达指定范围、拍摄照片、采集特定传感器数据，客户端应最大限度的方便用户使用，优化设计，减少参与者的人为介入工作，自动化采集过程。

完成任务包括两部分，参与者需要提供身份和任务信息，还要上传感知数据，为了简化交互流程，这两步可以在一个接口中实现，由表 4-3 确定。

表 4-3 完成感知任务接口参数表

URL	/task/complete	HTTP 方法	post
参数	类型	说明	
userID	string	用户 ID	
passToken	string	用户登录获得的 token	
taskID	string	感知任务 ID	
dataType	int	数据类型枚举值，1 表示二进制数据、2 表示文本数据	

感知数据放在 HTTP body 中，根据 dataType 的不同采用不同的解析方式处理；对于数据存取的需求有

```
//*****获取任务列表数据存储需求*****
```

```
//查询任务执行记录信息
```

```
TaskRecord queryTaskRecord(String taskID, String userID);
```

```
//更新任务执行记录信息
```

```
Boolean updateTaskRecord(TaskRecord taskRecord);
```

```
//*****获取任务列表数据存储需求*****
```

TaskRecord 为表示参与者执行任务的记录，包括任务执行阶段、所获得激励奖励等信息，任务完成时更新此记录为已完成阶段，设置相应的激励报酬值；同时在收到感知数据时调用数据存储功能，并执行 PM2.5 计算流程进行空气质量估计值计算。

4.3.3.2 参与者选择模块

根据任务选择的激励机制方式的不同，参与者选择包括竞价和锁定两种流程。

基于逆向竞拍的激励机制需要用户报价，执行竞拍流程来选择参与者；平台固定价格激励机制和动态分配预算激励机制不需竞价环节，采用先到先得的用户选择策略，需要用户来锁定任务以获得执行任务的资格。

对于竞价流程，在报价时间窗口内，潜在参与者都可以进行报价，上传自己可以接受的最低激励报酬（保护价格工资），报价结束之后，平台根据报价和其他决定因素通过用户选择算法确定被选中的参与者，通知其进行后续的感知任务执行流程，对于竞拍失败的用户通知其结果。

表 4-4 给出参与者选择模块竞拍接口格式。

表 4-4 竞拍接口参数表

URL	/task/bid	HTTP 方法	post
参数	类型	说明	
userID	string	用户 ID	
passToken	string	用户登录获得的 token	
taskID	string	感知任务 ID	
bid	double	参与者报价	

采用 HTTP 中的 post 方法上传用户报价值，对数据存储的需求为

```

//*****竞价流程数据存储需求*****
//增加一条报价信息，Bid 类描述了任务信息、用户信息、报价信息
Boolean addBidding(Bid bid);
//获取一个任务的竞价成功的报价列表
List<Bid> querySuccessfulBids(Task task);
//获取一个任务的竞价失败的报价列表
List<Bid> queryFailedBids(Task task);
//*****竞价流程数据存储需求*****

```

对于激励类型为固定价格激励方式和动态分配预算激励机制的感知任务，在招募参与者阶段，想要参与此次感知任务执行的潜在参与者都可以尝试锁定任务，锁定成功获得感知任务执行的机会，锁定任务参与者数量达到设定值之后关闭任务锁定窗口。本系统采用先到先得的方式选择用户，平台维护一个试图锁定任务的参与者队列，以先进先出的方式选择参与者，当剩余名额为零时关闭锁定任务入口，队列中如果有剩余参与者则出队并通知其锁定任务失败。

此处需要注意参与者并发的访问和修改执行任务剩余名额这个临界资源，需要做好互斥访问共享资源的操作，这在数据库操作中可以通过对表或列的加锁方式来实现，后面的详细设计中会再次介绍这部分的工作，表 4-5 给出具体的锁定任务接口

格式。

表 4-5 锁定任务接口参数表

URL	/task/lock	HTTP 方法	post
参数	类型	说明	
userID	string	用户 ID	
passToken	string	用户登录获得的 token	
taskID	string	感知任务 ID	

```
//*****锁定任务流程数据存储需求*****  
//更新一个任务的所处执行状态  
Boolean updateTaskState(TaskState taskState, Task task);  
//获取一个任务的剩余可以锁定的参与者数量  
int getRemainHeadCount(Task task);  
//设置一个任务的剩余可以锁定的参与者数量  
Boolean setRemainHeadCount(Task task, int headCount);  
//锁定任务成功，增加一条任务信息记录  
Boolean addTaskRecord(TaskRecord taskRecord);  
//*****锁定任务流程数据存储需求*****
```

4.4 本章小结

本章为参与式感知平台激励系统的需求分析和总体设计，通过第三章的理论研究，结合本课题现有参与式感知平台的软硬件资源和对图片数据的感知需求，进行了充分的需求分析；通过总结需求将系统进行模块划分，明确激励系统与其他系统和通用功能的调用交互关系，同时本人主要负责与客户端和服务端数据库访问的中间环节-通信接口和业务逻辑的实现部分，在本章中明确客户端的接口参数，对数据库存取提出需求，为第五章的详细设计和功能实现做好准备。

第五章 参与式感知平台激励机制的详细设计

5.1 任务管理模块

任务管理进程维持三个任务队列，分别为待发送任务队列、竞价中任务队列、待完成任务队列。任务发布时首先进入待发送队列，任务发布时间到达，由任务管理进程执行任务发布流程，根据任务所配置的激励机制类型，会进入到待完成任务队列（采用固定价格激励机制、动态分配预算激励机制）或竞价中任务队列（采用基于逆向竞拍的激励机制）；竞价任务竞价时间截止，也会进入待完成任务队列；任务收集完预定的数据或者过期时间到达，会被移出待完成队列，任务执行结束。图 5-1 的状态机展示了任务的生命周期。

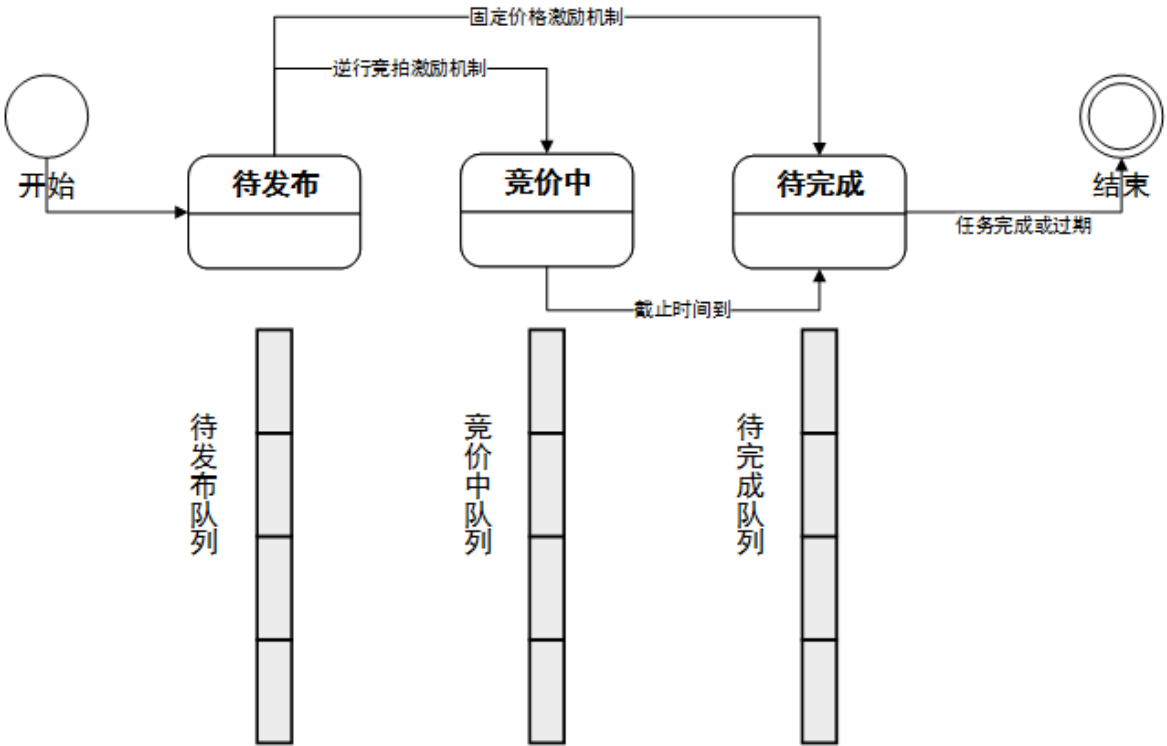


图 5-1 感知任务的状态转移

下面分别对发布任务子模块、任务列表管理子模块和完成任务子模块分别详细阐述。

5.1.1 发布任务

平台中有三个任务执行线程一直在后台运行，分别处理三个不同阶段的任务队列，队列为空时线程进入等待状态，队列虽然不为空但任务处理完毕，线程也进入休眠状态；当有任务需要发布时，发布任务线程将任务添加到相应的任务队列中，唤起任务执行线程，执行任务发布流程，发布任务的交互流程如图 5-2 所示。

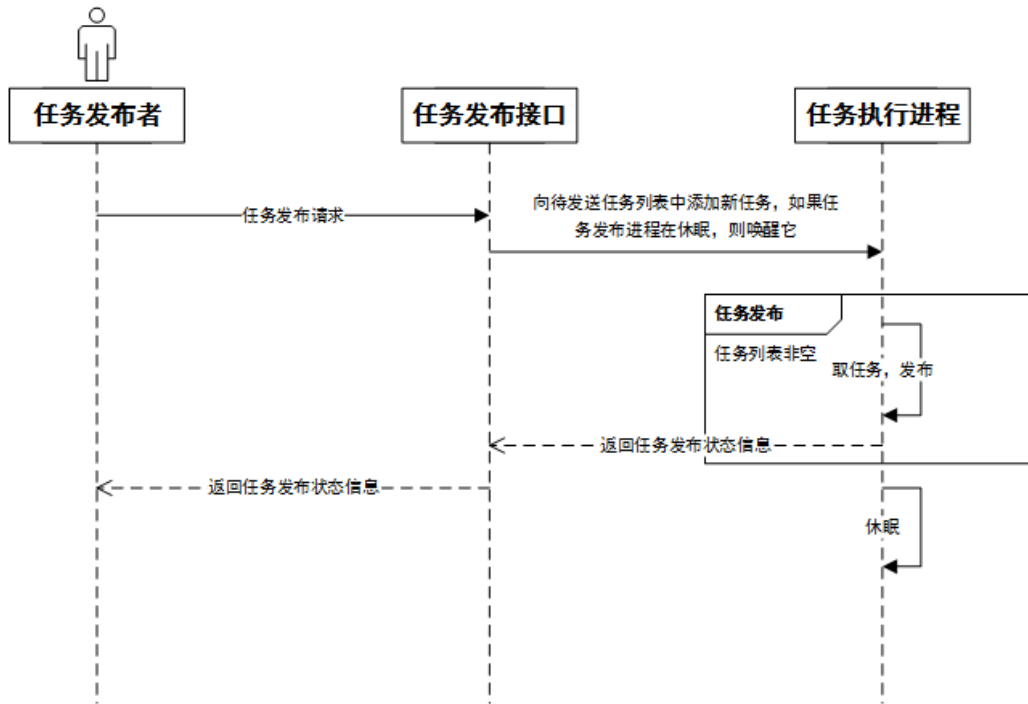


图 5-2 任务发布交互消息图

发布任务中会用到的关键数据结构有任务描述信息、任务两个类，任务描述信息代表了一个感知活动，通过任务描述信息可以生成每轮任务的具体信息，数据结构设计如类图 5-3、5-4 所示，

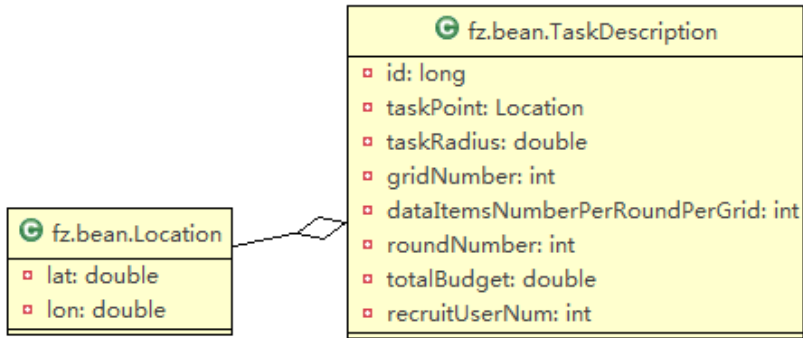


图 5-3 任务描述信息类图

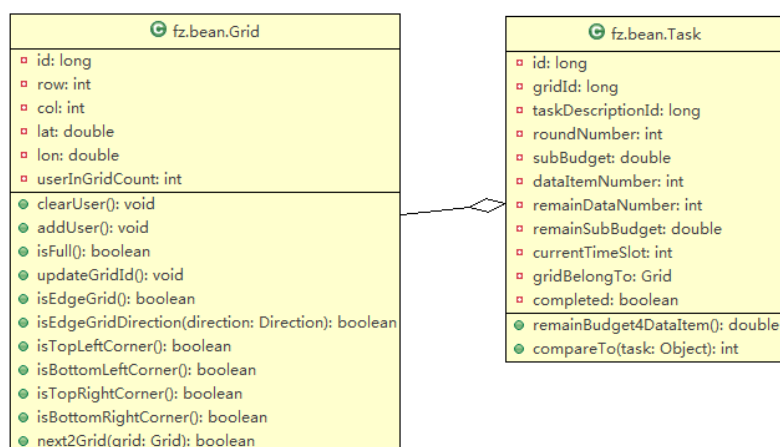


图 5-4 任务信息类图

Task 类通过 taskDescriptionId 维持对 TaskDescription 实例的引用，具有相同 taskDescriptionId 的任务都属于一个感知活动的不同轮次。

TaskManager 利用 Java 中的 RMI（远程方法调用）提供任务发布接口，其维持三个任务队列和三个任务执行线程，其属性和方法见类图 5-5，

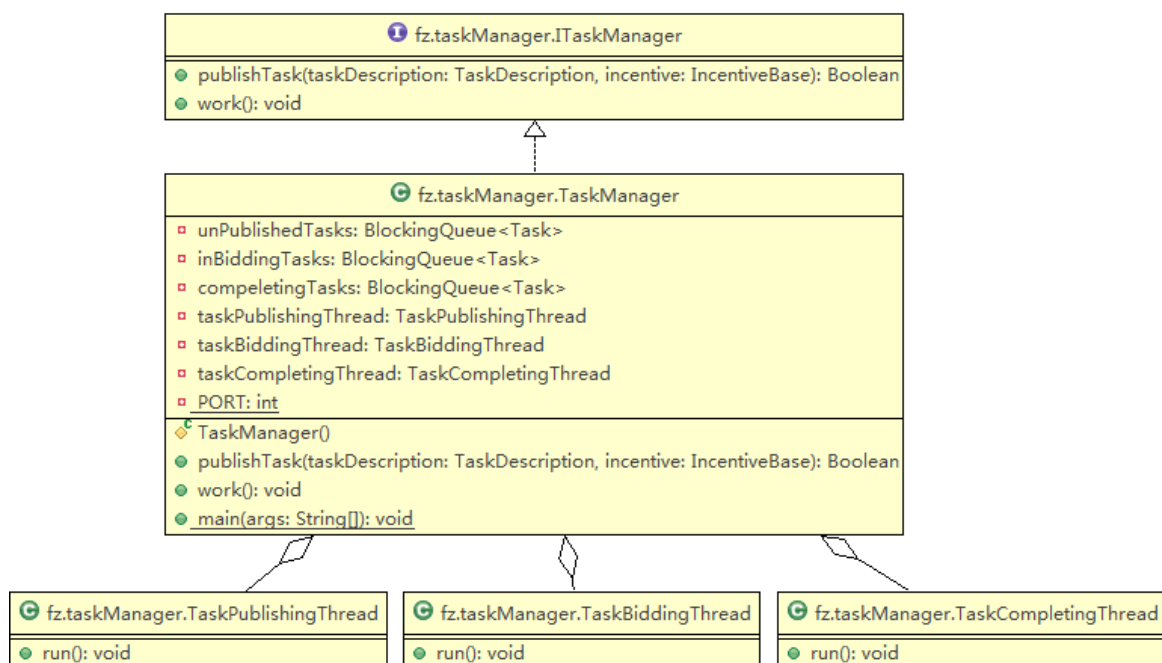


图 5-5 任务管理者类图

RMI 的注册代码如下

```

//*****锁定任务流程数据存储需求*****
// 启动 RMI 注册服务，指定端口为 1099 （1099 为默认端口）
    LocateRegistry.createRegistry(PORT);
    ITaskManager taskManager = new TaskManager();
// 把 taskManager 注册到 RMI 注册服务器上，命名为 TaskManager
    Naming.rebind("rmi://10.108.111.174:1099/TaskManager", taskManager);
// 启动三个工作线程，执行任务管理逻辑
    taskManager.work();
//*****锁定任务流程数据存储需求*****

```

由于 TaskManager 内部采用了线程安全的 blockingQueue，所以接口中的 publishTask 支持多线程发布任务。之后在 taskPublishingThread 中调用预算分配模块确定每轮任务的预算分配情况，生成任务写入数据库中，进行任务信息推送或提供给任务列表管理子模块生成任务列表，具体流程如图 5-6 所示，

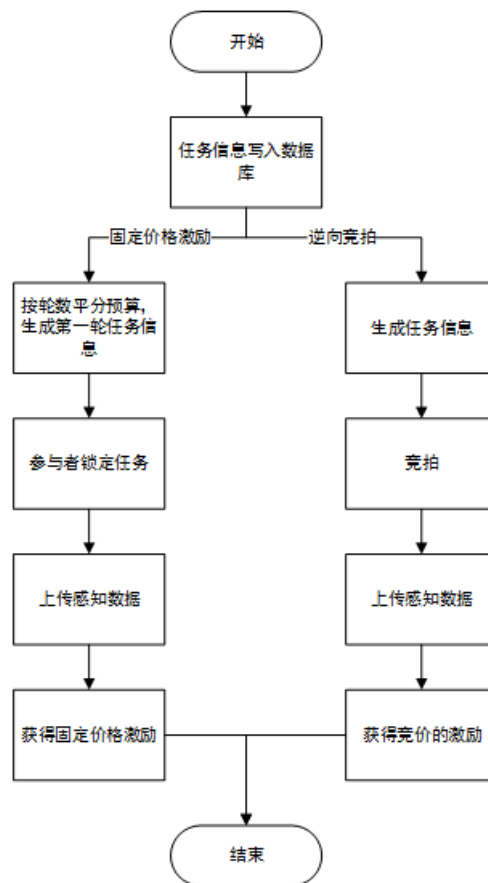


图 5-6 任务发布流程图

5.1.2 任务列表管理

用户参与的任务都会生成一个任务记录，通过持久化存储，可以按条件检索用户参与过的任务，包括锁定中任务、报价中任务、等待上传数据的任务、已完成任务等。用户获取自己的任务列表即向相应接口发送 HTTP 请求，相应的 Servlet 通过数据存储访问函数获得相应的任务列表，根据分页要求构造 JSON 数据返回给用户；用户请求可以参与的任务列表则需要单独处理，此时提供经纬度信息，数据存储访问函数通过地理位置信息检索正在开放招募的任务信息，返回的列表除去用户已参加的任务集合，即是用户可以参与的任务列表。

任务记录的数据结构如图 5-7，

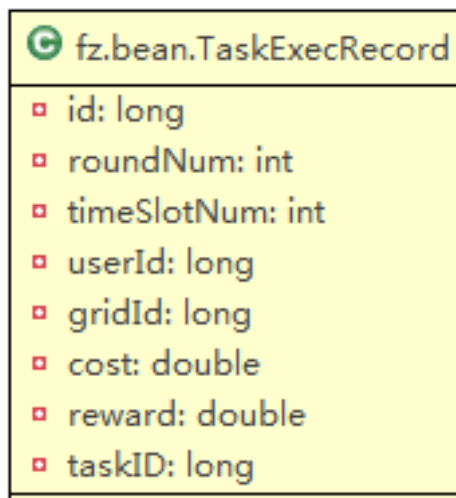


图 5-7 任务记录类图

通过 taskID 可以关联到 task 信息，根据 task 的不同阶段（taskStage 属性）来检索任务执行记录信息。

5.1.3 完成任务

完成任务时通过用户上传的 taskID 和 userID 信息，检索 TaskExecRecord 记录，根据记录中的激励报酬信息，调用用户系统提供的接口修改用户账户金额；同时将感知数据（目前只是照片）输入到 PM2.5 计算模块，图片文件的 EXIF 中存储了相关元信息，PM2.5 计算模块自己可以解析用于计算环节。

```

//*****锁定任务流程数据存储需求*****
//生成图片文件的路径
String sourcepath = path + uri;
//生成一个计算 PM2.5 值得实例
unitImgPrePm getFpm = new unitImgPrePm();
//调用方法计算 PM2.5 值
fpm = getFpm.returnFpm(sourcepath);
//*****锁定任务流程数据存储需求*****
  
```

5.2 参与者选择模块

5.2.1 竞拍管理

用户通过报价接口上报自己的竞拍价格，首先检查任务是否开放报价，开放则新增一条报价记录或更新已上报价格，报价记录中有 `userID`、`taskID`、`bid` 价格信息，当任务报价结束时检索 `taskID` 下的报价记录，根据设定的参与者数量选择报价最低的 `n` 个用户，对于报价成功的用户生成 `TaskExecRecord` 信息，记录其应得的感知报酬信息，且标记为待上传数据，数据上传后进行结算。

5.2.2 任务锁定

采用固定价格激励机制、动态分配预算的激励机制的感知任务，参与者在直接上传之前要先锁定任务以预约任务的执行资格，避免上传过多的数据造成预算浪费。锁定流程比较简单，注意多线程读写数据库时用同步的数据访问方法来互斥访问，同时采用数据库中的事物的原子操作特性来变更剩余任务数量。

```

//*****锁定任务流程数据存储需求*****
//用同步关键字修饰此方法
synchronized public static boolean lockTask(long taskID){
    Session session =
Application.sharedApplication().getSessionFactory().openSession();
    Transaction tx = null;
    try{
        tx = session.beginTransaction();
        Task task = queryTaskByID(taskID);
        if(task.getRemainDataNumber()>0){
            task.setRemainDataNumber(task.getRemainDataNumber()-1);
        }
        session.saveOrUpdate(task);
        tx.commit();

    }catch (HibernateException e) {
        if (tx!=null) tx.rollback();
        e.printStackTrace();
    }
}

```

```

    }finally {
        session.close();
    }
}

//*****锁定任务流程数据存储需求*****

```

如果返回锁定失败，则任务召集了足够数量的参与者，通知刷新任务列表，避免用户再次尝试锁定。

5.3 预算分配模块

5.3.1 每轮任务预算的确定

对于总预算在每轮任务中的分配，固定价格激励机制中的分配最简单，按照总预算和任务轮次平均分配即可；对于基于逆行竞拍的激励机制，目标收集 m 轮数据，每个格子 n 份数据，根据竞拍结果可能造成预算超支，本系统采用每轮任务的平均预算的 120% 为预算上限，根据竞拍结果选择参与者。最复杂的是动态分配预算的激励机制，在本小节中重点说明实现细节。

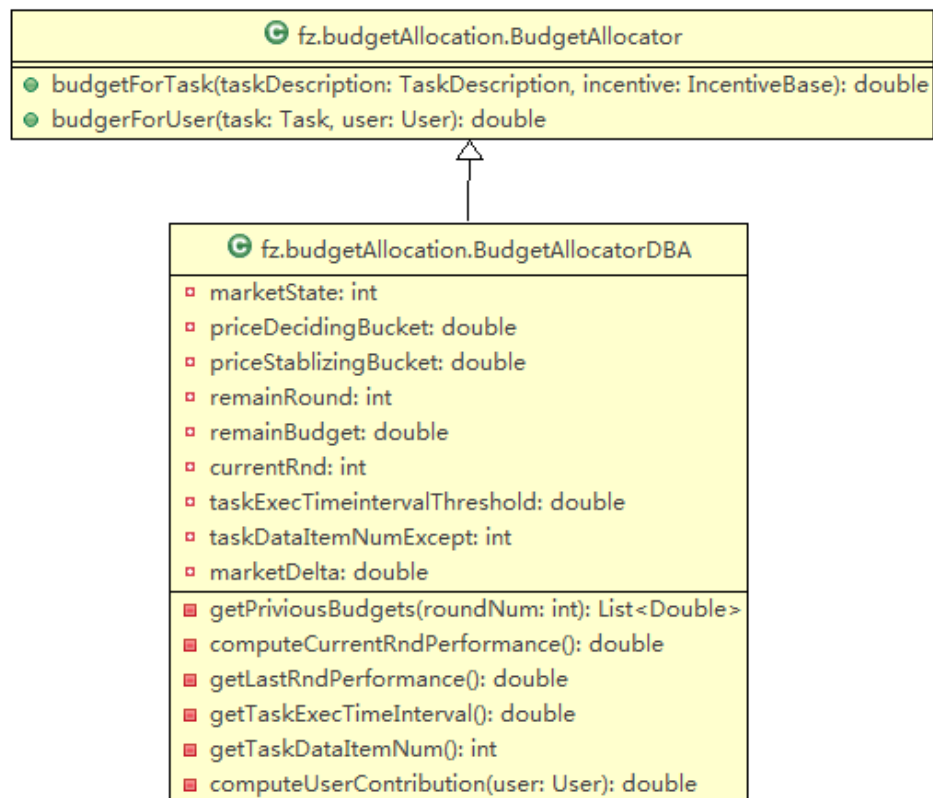


图 5-8 任务发布流程图

对于选择动态分配预算激励机制的感知活动，生成一个 `BudgetAllocatorDBA` 类的实例作为其预算分配工具，`BudgetAllocatorDBA` 类的属性和方法由类图 5-8 列出。

在每轮任务执行完成后，需要将任务完成情况信息持久化存储，用于之后的预算分配过程，任务执行情况信息的数据结构设计如下类图 5-9 所示，

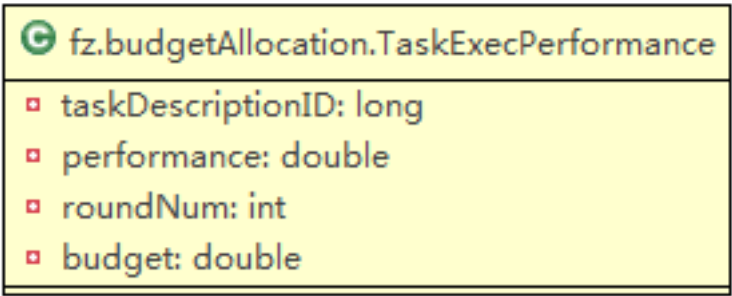


图 5-9 任务发布流程图

5.3.2 子任务预算分配

用户总的激励金额由两部分组成，固定激励部分和浮动奖励部分，固定激励部分很容易得出，通过本轮任务预算的固定支出部分和期望数据份数相除得到，浮动奖励部分要计算每个用户对平台贡献值的比重，根据其贡献比重分配剩余的浮动激励部分预算。通过 `BudgetAllocatorDBA` 的实例方法 `computeUserContributong` 可以计算本轮任务所有参与者的贡献值，贡献值由任务完成时间和数据份数共同决定。

5.4 其他模块或功能实现

5.4.1 推送功能

推送模块用例图如图 5-10，用户在打开手机是应用询问用户是否同意推送消息通知，用户也可以在个人设置里取消推送消息；平台角度需要广播消息和给特定用户推送定制消息两个需求。

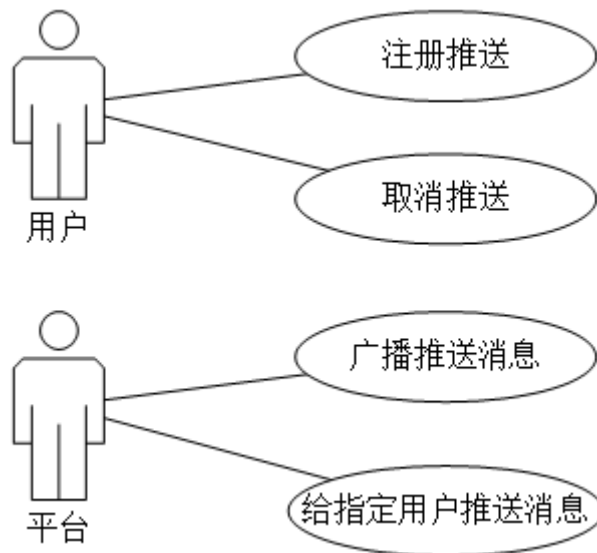


图 5-10 推送模块用例图

用户同意平台向其推送消息，则需要上传用户识别标志-推送 token 给服务器，如果用户已经注册，则连同用户 ID 信息一起上传，否则上传生成的 UUID 来唯一识别用户，

更新 pushtoken 的原理示意图如图 5-11 所示，用户设备向苹果 APNs 服务器发起 SSL 安全连接，APNs 返回设备唯一的 deviceToken 给手机，app 将此 token（32 字节长度的二进制序列）提交给推送服务提供者，也就是参与式感知收据收集平台，平台拥有此 token 才可以后续给参与者推送消息通知。

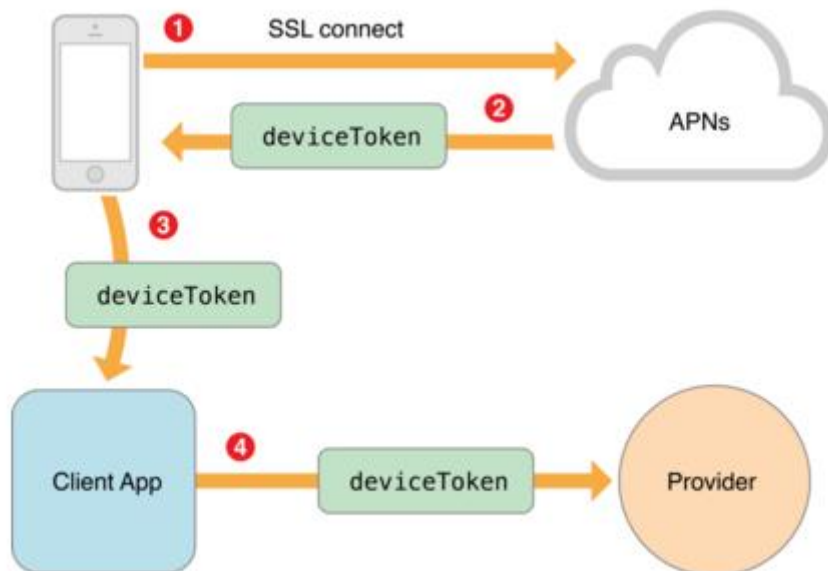


图 5-11 用户从 apns 获取推送 token 上传给平台

图 5-12 给出了参与者向感知平台上传或更新 token 流程，表 5-1 确定更新 token 的接口格式。

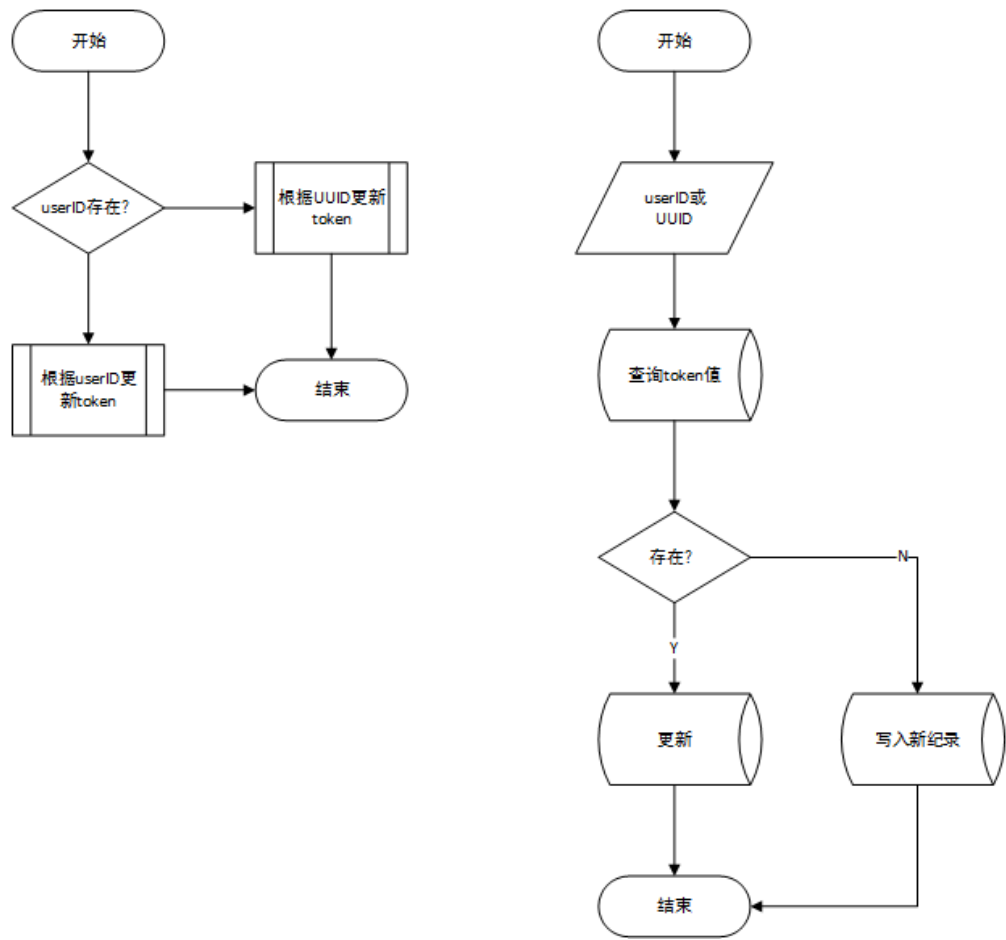


图 5-12 更新 token 流程图

表 5-1 更新 token 接口参数表

URL	/pushtoken	HTTP 方法	post
参数	类型	说明	
userID	string	用户 ID	
passtoken	string	用户登录 token	
pushtoken	string	用户设备推动 token	
UUID	string	用户设备唯一识别码，未登录时代替 userID 作为用户标示	

对数据存取提出的更新 token 需求如下函数，

Boolean updateToken(String userID, String UUID, String token);

优先使用 userID 管理 token，如流程图所示。

对于广播推送消息请求，目前采用的实现方式是遍历所有用户，使用对特定用户推送消息的方法实现广播消息。

对特定用户推送消息，流程图如下图 5-13，

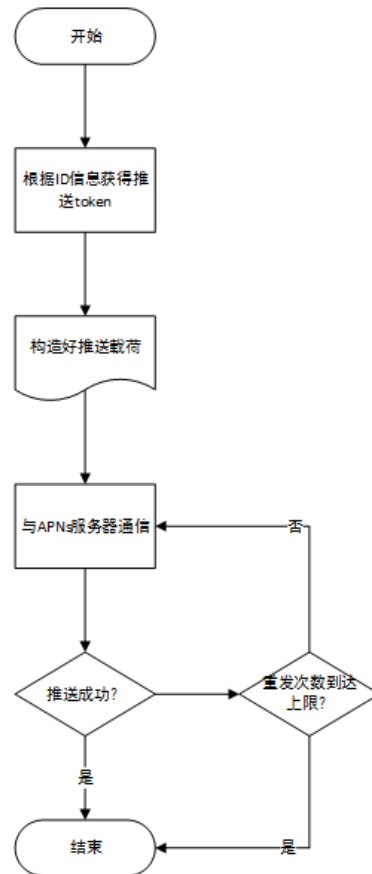


图 5-13 推送消息给特定用户

为其他模块提供的调用接口,为 `Boolean pushMessage(String message, String token)`。

5.4.2 用户系统

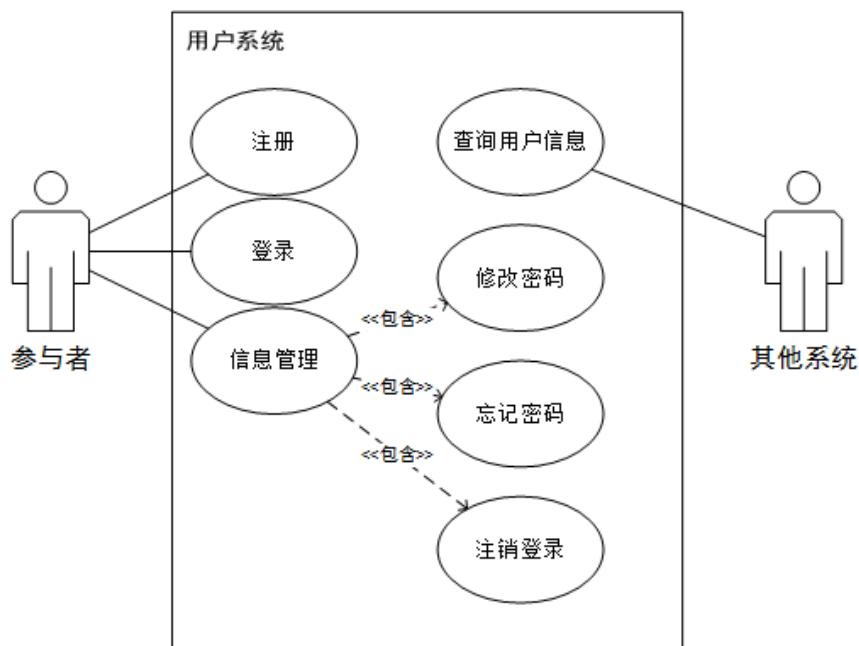


图 5-14 用户管理模块用例图

用户系统的用例图如图 5-14 所示，因为用户系统的开发比较成熟，本课题所依托的实验平台对于用户系统也没有特殊需求，对于用户系统的需求分析、设计不在赘述，在第六章的测试中给出各项功能的功能测试。

5.5 本章总结

本章为参与式感知平台激励系统的详细设计，同时设计了用户系统、推送功能等平台基础需求，主要分为数据结构设计、算法和流程设计，其中任务管理模块通过任务转移状态机来实现任务从发布、竞拍与锁定到完成的过程；参与者选择有两种方式，一是需要协商的基于逆向竞拍的参与者选择过程，二是先到先得的基于任务锁定的参与者选择过程；预算分配模块在固定价格激励机制、基于逆向竞拍的激励机制场景下比较简单，对于动态分配预算激励机制场景，设计数据结构和数据持久化存储方案实现了第三章中给出的动态分配预算算法。

第六章 系统部署与测试

6.1 系统部署

6.1.1 系统开发环境

编程语言：Java（JDK 1.7.0）

集成开发环境：Eclipse JavaEE 版

构建工具：Maven 3.3.9

编辑器：Vim

版本控制：SVN

脚本语言：shell

6.1.2 系统运行环境

操作系统：Ubuntu 12.04

内核版本：Linux 3.2.0-20-generic

Web 容器：Tomcat/Jetty

开发框架：Spring/Hibernate/Jersey

负载均衡：nginx

6.1.3 Tomcat 集群和 nginx 负载均衡的部署与配置

为了提高平台 Web 服务的可用性和可靠性，增强可扩展性，可以将不同的系统部署到单独的服务器中，对一个系统的改动不会影响到其他系统，在系统的最前端部署一台运行 nginx 的机器作为软件的负载均衡，每个系统可以部署到多台 Tomcat 服务器上提供冗余的处理能力，由 nginx 做反向代理的负载均衡，使得平台系统的可用性、可扩展性都大幅提高，图 6-1 展示 Tomcat 集群和负载均衡服务器示意图。

本文中在一台物理服务器中运行了多个 Tomcat 实例，分别加载不同的系统，这样在重新部署一个系统时不会中断其他系统的运行。具体设置方式如下：

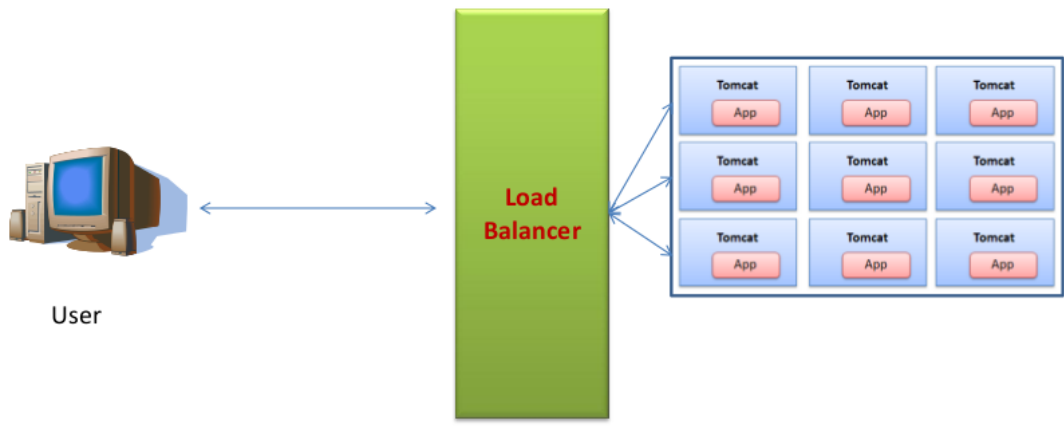


图 6-1 利用 Tomcat 集群和负载均衡提高服务可用性

将 conf, temp, Webapps and logs 文件夹复制一份到自己制定的位置，修改 start.sh 脚本中 CATALINA_BASE 环境变量为这个指定的位置，还需更改 Tomcat 运行所需要的三个端口号

shutdown port: 用于关闭 Tomcat 的端口号

connector port: 将应用暴露给外部客户端的实际端口号

ajp port: 这是 Web 服务器与 Tomcat 通信的端口号，当我们配置负载均衡服务器时要用到

这样我们就能运行多个 Tomcat 实例于一天物理机器上，分别部署不同的系统，增强了部署和更新系统的灵活性。

Nginx 做负载均衡的配置

Nginx 做负载均衡只需要两步

第一步，在 nginx 的配置文件 nginx.conf 中，设置好上游集群。一个上游集群提供相同的服务，由 nginx 做负载均衡，将请求发送给集群中不同的机器。

表 6-1Tomcat 集群配置文件

配置 Tomcat 集群文件示例

upstream psserver {
server 127.0.0.1:8181;
server 127.0.0.1:8282;
server 127.0.0.1:8383;
}

psserver 就是上游集群的名字，是部署了相同的参与式感知服务程序的 Tomcat Web 容器；

第二步，将 Web 请求的 URI 与相应的上游集群映射，

表 6-2Tomcat 上流映射键值对

配置 Tomcat 上流映射
<pre>location / { proxy_pass http://psserver; }</pre>

这条配置信心将所有请求导流到 psserver 上流集群，完成了 nginx 做负载均衡服务器的任务。

经过这两步，nginx 会将所有请求以轮训的方式分配到上流集群的机器中。

6.2 功能测试

功能性测试，也称为黑盒测试，指不了解程序内部运行逻辑的情况下，通过精心设计的测试用例，验证程序的每个功能在任何场景都能正确工作。对系统的功能测试我主要采用了黑盒测试，同时设计测试用例的时候考虑输入不合法、边界值等异常情况来验证系统的鲁棒性。

根据 4.1 节中的需求分析，现设计每个模块的测试用例，测试系统是否实现了各项需求。

6.2.1 任务管理模块

通过任务管理模块，系统目前支持通过 Web 端的控制台界面发布任务，通过配置任务信息、选择激励方式，成功发布一条任务的流程效果图如下图 6-2 所示。

固定价格任务

竞标任务

周期任务

任务描述:

开始时间:

--

时:

分

结束时间:

--

时:

分

总预算:

任务份数:

任务范围:

3

任务位置(请搜索):

发布任务

图 6-2 任务发布流程图

发布前查看数据库中的任务表，显示为空，如图 6-3，

```
hbase(main):014:0> scan 'task'
ROW                                COLUMN+CELL
0 row(s) in 0.0130 seconds
```

初始时任务表空

图 6-3 任务表为空

发布一条任务后，任务表记录增加一条，如图 6-4，

```
hbase(main):033:0> scan 'task'
ROW                                COLUMN+CELL
39.97/116.3524/100.01             column=info:proper, timestamp=1433406345242, value={"taskType":1,"incentiveType":1,"budget":99.9,"dataNumber":10,"beginTime":1433406540000,"biddingDeadline":0,"endTime":1433467680000,"taskDescription":"\xE6\x88\x91\xE4\xBB\xAC\xE9\x9C\x80\xE8\xA6\x81\xE4\xB8\x80\xE5\xBC\xA0\xE7\x85\xA7\xE7\x89\x87\xEF\xBC\x8C\xE5\x9C\xA8\xE5\x8C\x97\xE4\xBA\xAC\xE7\x94\xB5\xE5\xBD\xB1\xE5\xAD\xA6\xE9\x99\xA2\xE4\xB8\x9C\x432\xE7\xB1\xB3\xEF\xBC\x8C\xE5\xA5\x96\xE5\x8A\xB1\xEF\xBF\xA59.99\xEF\xBC\x8C\xE5\xB\xF\xAB\xE6\x9D\xA5\xE5\x8F\x82\xE4\xB8\x8E\xEF\xBC\x81","taskId":"39.97/116.3524/100.01"}
39.97/116.3524/100.01             column=state:participants, timestamp=1433406345242, value=[]
39.97/116.3524/100.01             column=state:taskphase, timestamp=1433406345242, value=\x00\x00\x00\x01
1 row(s) in 0.0250 seconds
```

成功发布一条任务

1代表待发布状态

图 6-4 任务表增加一条记录

任务处于竞价中状态，如图 6-5，

```
39.98/116.3524/100.01/1433689440433 column=info:proper, timestamp=1433689440608, value={"taskType":1,"incentiveType":2,"budget":99.9,"dataNumber":3,"beginTime":1433588400000,"biddingDeadline":1433689800000,"endTime":1433935680000,"taskDescription":"\xE6\x88\x91\xE4\xBB\xAC\xE9\x9C\x80\xE8\xA6\x81\xE4\xB8\x80\xE5\xBC\xA0\xE7\x85\xA7\xE7\x89\x87\xEF\xBC\x8C\xE5\x9C\xA8\xE5\xA1\x94\xE9\x99\xA2\xE9\x82\xAE\xE7\xA7\x91\xE7\xA4\xBE\xE5\x8C\xBA\xE5\x86\x850\xE7\xB1\xB3\xEF\xBC\x8C\xE4\xB8\x8A\xE6\x8A\xA5\xE4\xBB\xB7\xE6\xA0\xBC\xE5\x8F\x82\xE4\xB8\x8E\xE5\x90\xA7\xEF\xBC\x81","taskId":"39.98/116.3524/100.01/1433689440433"}
39.98/116.3524/100.01/1433689440433 column=state:participants, timestamp=1433689440608, value=[]
39.98/116.3524/100.01/1433689440433 column=state:taskphase, timestamp=1433689452246, value=\x00\x00\x00\x02
440433
```

无人参与竞价

2代表竞价状态

图 6-5 任务发布流程图

通过 Postman 工具测试报价接口，返回成功报价提示信息，如图 6-6，



图 6-6 Postman 工具测试报价接口

任务表中竞标数据成功增加一条 66 元的报价信息，如图 6-7 所示，

```

39.98/116.3524/100.01/1433689 column=bid:142, timestamp=1433689618186, value={"userId":"142","bidPrice":26}
440433
39.98/116.3524/100.01/1433689 column=bid:144, timestamp=1433689628229, value={"userId":"144","bidPrice":29}
440433
39.98/116.3524/100.01/1433689 column=bid:150, timestamp=1433689639656, value={"userId":"150","bidPrice":80}
440433
39.98/116.3524/100.01/1433689 column=bid:jack, timestamp=1433689519953, value={"userId":"jack","bidPrice":66}
440433
39.98/116.3524/100.01/1433689 column=info:proper, timestamp=1433689440608, value={"taskType":1,"incentiveType":2,"budget":99.9,"dataNumber":3,"beginTime":1433588400000,"biddingDeadline":1433689800000,"endTime":1433935680000,"taskDescription":"\xE6\x88\x91\xE4\xBB\xAC\xE9\x9C\x80\xE8\xA6\x81\xE4\xB8\x80\xE5\xBC\xA0\xE7\x85\xA7\xE7\x89\x87\xEF\xBC\x8C\xE5\x9C\xA8\xE5\xA1\x94\xE9\x99\xA2\xE9\x82\xAE\xE7\xA7\x91\xE7\xA4\xBE\xE5\x8C\xBA\xE5\x86\x850\xE7\xB1\xB3\xEF\xBC\x8C\xE4\xB8\x8A\xE6\x8A\xA5\xE4\xBB\xB7\xE6\xA0\xBC\xE5\x8F\x82\xE4\xB8\x8E\xE5\x90\xA7\xEF\xBC\x81","taskId":"39.98/116.3524/100.01/1433689440433"}
39.98/116.3524/100.01/1433689 column=state:participants, timestamp=1433689440608, value=[]
440433

```

jack 报价排第三位

报价截止时间未到达

图 6-7 成功增加报价信息记录

报价时间窗口截止，系统选出了报价最低的三个竞标者，如图 6-8 所示，

```

39.98/116.3524/100.01/1433689 column=info:proper, timestamp=1433689440608, value={"taskType":1,"incentiveType":2,"budget":99.9,"dataNumber":3,"beginTime":1433588400000,"biddingDeadline":1433689800000,"endTime":1433935680000,"taskDescription":"\xE6\x88\x91\xE4\xBB\xAC\xE9\x9C\x80\xE8\xA6\x81\xE4\xB8\x80\xE5\xBC\xA0\xE7\x85\xA7\xE7\x89\x87\xEF\xBC\x8C\xE5\x9C\xA8\xE5\xA1\x94\xE9\x99\xA2\xE9\x82\xAE\xE7\xA7\x91\xE7\xA4\xBE\xE5\x8C\xBA\xE5\x86\x850\xE7\xB1\xB3\xEF\xBC\x8C\xE4\xB8\x8A\xE6\x8A\xA5\xE4\xBB\xB7\xE6\xA0\xBC\xE5\x8F\x82\xE4\xB8\x8E\xE5\x90\xA7\xEF\xBC\x81","taskId":"39.98/116.3524/100.01/1433689440433"}
39.98/116.3524/100.01/1433689 column=state:participants, timestamp=1433689814205, value=["142","144","jack"]
39.98/116.3524/100.01/1433689 column=state:taskphase, timestamp=1433689814205, value=\x00\x00\x00\x03
440433

```

报价最低的三个用户被选中

图 6-8 逆向竞拍激励机制参与者选择

如上图的实际系统测试截图表示了发布任务、竞拍中的参与者选择都满足了需求分析中的需求，其他全面的关于激励系统的测试用例和测试结果在表 6-3 和 6-4 中给出。

表 6-3 任务管理模块测试用例

用例编号	测试用例	预测结果	测试结果
1-1	任务发布时任务信息写入到数据库	通过 Web 界面、手机端发布任务，数据库中任务表中增加一条记录	符合
1-2	任务定时时刻到达，任务执行进程执行任务	任务由待发布状态转移到相应转态	符合
1-3	任务列表查看	可以显示发布的测试任务列表，包括固定价格激励任务和逆向竞拍两类任务	符合
1-4	任务锁定	对于固定价格激励任务，成功锁定任务获得执行任务资格，在任务列表中归类到待完成任务中	符合

1-5	逆向竞拍报价	对于基于逆向竞拍激励的任务，用户报价，等待报价截止时间到来查询竞价结果	符合
1-6	完成任务，上传感知数据，包括照片和 JSON 数据	将照片，JSON 数据成功接收，计算 pm2.5 值，启动完成任务流程，分配激励	符合

6.2.2 预算分配和用户选择模块

表 6-4 激励分配模块测试用例

用例编号	测试用例	预测结果	测试结果
2-1	受任务管理模块调用，采用固定价格激励机制	固定价格激励任务，根据总预算平均分配给每轮任务，生成任务描述信息	符合
2-2	受任务管理模块调用，采用基于逆向竞拍的激励机制	生成竞拍任务的描述信息，预算不足时提前终止任务	符合
2-3	锁定任务，对于固定价格任务用户发起锁定请求	任务数据表中参与者列表中增加用户 ID	符合
2-4	逆向竞拍，对于报价用户，选出报价最低的 n 个用户	报价截止时根据报价对参与者排序，从小到大输出 n 个 ID	符合
2-5	激励结算	用户上传感知数据后，根据任务 ID 检索任务信息，确认用户有执行任务资格后付给激励报酬	符合

6.2.3 推送功能

推送功能在任务发布和用户选择中频繁被调用，图 6-9 为任务发布过程中将任务信息推送给经轨迹预测系统提供的潜在参与者实际截图，图 6-10 为在用户选择过程中将竞价成功信息推送给参与者，提醒其尽快完成感知数据采集。



图 6-9 任务发布流程图



图 6-10 任务发布流程图

推送功能的详细测试用例和结果在表 6-5 中列出。

表 6-5 推送模块测试用例

用例编号	测试用例	预测结果	测试结果
3-1	用户同意推送请求	通过用户 ID 可以检索出推送 token	符合
3-2	用户取消推送授权	通过用户 ID 检索推送 token 为空	符合
3-3	推送任务消息给用户	用户手机通知栏收到消息	符合
3-4	广播推送消息	所有同意推送请求的用户都收到推送消息	符合

6.2.4 用户系统

对于用户系统，同样采用 Postman 测试接口功能，如图 6-13 是新注册一个用户，返回用户注册成功信息，

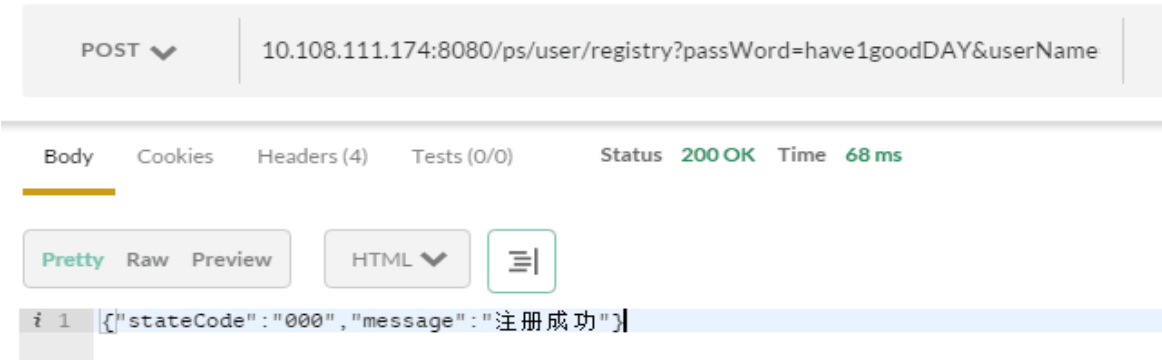


图 6-13 任务发布流程图

详细的测试用例和结果如表 6-6 中列出，

表 6-6 用户系统测试用例

用例编号	测试用例	预测结果	测试结果
4-1	注册	用户表多一条数据	符合
4-2	登录	返回 token 给用户	符合
4-3	上传（更新）推送 token	可以通过 userID 检索推送 token	符合
4-4	查看任务信息	分类显示参与者正在执行、已完成的各项任务	符合
4-5	查看激励金额	显示已获得的激励金额数字	符合

6.3 本章总结

本章首先介绍平台服务器中 Tomcat 的部署情况，通过单机部署多个 Tomcat 示例的方式结合 nginx 做反向代理实现了负载均衡的功能，在之后系统服务压力增大时可以通过增加物理服务器来透明的扩容服务能力；之后通过测试用例的设计，全面测试了激励系统、用户系统和推送服务的逻辑功能，通过线上系统的真实功能截图展示了各模块的功能。

第七章 总结与展望

7.1 工作总结

本论文对参与式感知中的激励机制进行了广泛和深入的研究，阅读大量现有激励机制文献，总结其优缺点，提出了一种参与式感知平台适用的动态分配预算的激励机制，在仿真实验中性能表现优于现有的常用方法。同时在本论文所依托的实验性参与式感知数据收集平台上，实现了最基本的任务发布和激励机制的加入，对论文中的激励机制提出设计方案。

论文首先对现有参与式感知环境下的激励机制进行广泛调研，用分类方法归纳已有研究，总结其解决问题所采用的方法，指出优缺点；之后综合基于逆向竞拍的激励机制和平台固定价格的激励机制，提出新颖的动态分配预算的激励机制，并将三种机制在仿真实验中进行性能对比；接下来对本论文所依托的实验平台任务发布和激励分配部分进行需求分析，设计并实现了激励系统，使得任务发布者（平台）可以提供一定的预算、选择合适的激励机制类型来发起参与式感知活动，参与者通过执行感知任务贡献宝贵的感知数据，同时也得到了激励补偿其感知开销；最后对系统进行了单元测试和系统联调，确保需求得到实现，系统可靠运行。

7.2 工作展望

参与式感知随着移动智能设备的普及和网络基础设施的完善而兴起，展示出了区别于传统传感器网络的巨大优势和潜力，在学术领域和工程实践中都获得了高度关注。本论文研究了参与式感知领域中的一个重要方向，激励机制。可以从激励机制这个纵向上继续深入研究，也可以关注参与式感知横向上其他问题的研究。参与式感知不同于任务分发平台（例如亚马逊 MTurk 平台^[28]），要考虑非经济类激励因素，本文的研究领域聚焦在经济类的激励机制上，未来可以考虑社交因素、公益性、游戏性等其他激励因素对参与积极性的影响。横向上的研究领域，感知数据质量控制是一个值得关注的方向，通过信用制度、感知数据的互相印证等方法，可以对感知数据的质量保证问题做进一步研究。

在平台的激励机制实现上，目前竞价时只有在截止时间到达才揭晓结果，可以基于即时通信的协议做成博弈性质的逆向竞拍，增强互动性；对于一些实时任务，目前采用轮训的方式发现任务，可以改成注册回调的方式减小开销。

7.3 研究生期间主要工作

研究生期间，主要参加的的工作有

一、青年基金项目“智能图书馆”、“公交助手”

- 智能图书馆应用在 APP Store 平台的上线发布
- 定期更新公交助手中的公交信息 SQLite 数据库，发布新版本

二、国家自然科学基金项目“移动协作感知”项目

- 负责参与式感知平台移动客户端设计的前期调研工作
- 负责参与式感知平台激励机制的研究
- 负责参与式感知平台激励系统的设计与实现
- 负责参与式感知平台对客户端的通信接口的设计和实现

三、申请诺基亚国际专利”AN OPTIMIZED BUDGET ALLOCATION METHOD IN PARTICIPATORY SENSING”

- 已通过专利申请，专利申请号为 PCT/CN2015/085406.
- 负责专利相关技术调研和专利文档撰写

参考文献

- [1] Burke J A, Estrin D, Hansen M, et al. Participatory sensing[J]. Center for Embedded Network Sensing, 2006.
- [2] Reddy S, Estrin D, Hansen M, et al. Examining micro-payments for participatory sensing data collections[C]//Proceedings of the 12th ACM international conference on Ubiquitous computing. ACM, 2010: 33-36.
- [3] Lee J S, Hoh B. Sell your experiences: a market mechanism based incentive for participatory sensing[C]//Pervasive Computing and Communications (PerCom), 2010 IEEE International Conference on. IEEE, 2010: 60-68.
- [4] Koutsopoulos I. Optimal incentive-driven design of participatory sensing systems[C]//Infocom, 2013 proceedings ieee. IEEE, 2013: 1402-1410.
- [5] Jaimes L G, Vergara-Laurens I, Labrador M A. A location-based incentive mechanism for participatory sensing systems with budget constraints[C]//Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on. IEEE, 2012: 103-108.
- [6] Tham C K, Luo T. Quality of contributed service and market equilibrium for participatory sensing[J]. Mobile Computing, IEEE Transactions on, 2015, 14(4): 829-842.
- [7] Xiong H, Zhang D, Chen G, et al. CrowdTasker: Maximizing coverage quality in Piggyback Crowdsensing under budget constraint[C]//Pervasive Computing and Communications (PerCom), 2015 IEEE International Conference on. IEEE, 2015: 55-62.
- [8] 李增泉. 激励机制与企业绩效: 一项基于上市公司的实证研究[J]. 会计研究, 2000 (1): 24-30.
- [9] Campbell A T, Eisenman S B, Lane N D, et al. People-centric urban sensing[C]//Proceedings of the 2nd annual international workshop on Wireless internet. ACM, 2006: 18.
- [10]<https://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2015.pdf>
- [11] Kanhere S S. Participatory sensing: Crowdsourcing data from mobile smartphones in urban spaces[C]//Mobile Data Management (MDM), 2011 12th IEEE International Conference on. IEEE, 2011, 2: 3-6.
- [12]<https://www.w3.org/Protocols/rfc2616/rfc2616.html>
- [13]https://en.wikipedia.org/wiki/Transport_Layer_Security

- [14]<http://www.json.org/json-zh.html>
- [15]<https://Spring.io/>
- [16]<http://Tomcat.apache.org/>
- [17]https://en.wikipedia.org/wiki/Representational_state_transfer
- [18]Krontiris, Ioannis, and Andreas Albers. "Monetary incentives in participatory sensing using multi-attributive auctions." *International Journal of Parallel, Emergent and Distributed Systems* 27.4 (2012): 317-336.
- [19]Zhao, Dong, Xiang-Yang Li, and Huadong Ma. "How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint." *INFOCOM, 2014 Proceedings IEEE. IEEE*, 2014.
- [20]Mukherjee T, Chander D, Mondal A, et al. CityZen: A cost-effective city management system with incentive-driven resident engagement[C]//*Mobile Data Management (MDM), 2014 IEEE 15th International Conference on. IEEE*, 2014, 1: 289-296.
- [21]Yang D, Xue G, Fang X, et al. Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing[C]//*Proceedings of the 18th annual international conference on Mobile computing and networking. ACM*, 2012: 173-184.
- [22]Lee J S, Hoh B. Dynamic pricing incentive for participatory sensing[J]. *Pervasive and Mobile Computing*, 2010, 6(6): 693-708.
- [23]Reddy S, Estrin D, Srivastava M. Recruitment framework for participatory sensing data collections[M]//*Pervasive Computing. Springer Berlin Heidelberg*, 2010: 138-155.
- [24]Ganti R K, Ye F, Lei H. Mobile crowdsensing: current state and future challenges[J]. *Communications Magazine, IEEE*, 2011, 49(11): 32-39.
- [25]Gaonkar S, Li J, Choudhury R R, et al. Micro-blog: sharing and querying content through mobile phones and social participation[C]//*Proceedings of the 6th international conference on Mobile systems, applications, and services. ACM*, 2008: 174-186.
- [26]Ra M R, Liu B, La Porta T F, et al. Medusa: A programming framework for crowd-sensing applications[C]//*Proceedings of the 10th international conference on Mobile systems, applications, and services. ACM*, 2012: 337-350.
- [27]Das T, Mohan P, Padmanabhan V N, et al. PRISM: platform for remote sensing using smartphones[C]//*Proceedings of the 8th international conference on Mobile systems, applications, and services. ACM*, 2010: 63-76
- [28]<https://www.mturk.com/mturk/welcome>

致谢

感谢王文东老师，作为我的项目指导老师，每一次例会都给出及时的指导，为项目的顺利推进把好关；在实习和找工作期间，充分理解同学们的想法，给予了理解和关心；最后的论文撰写过程中，王老师不顾自己的工作繁忙，抽出休息时间为我们指导论文结构和内容。您在我的学业和生活中都给予了指导和帮助，作为学生表达最真挚的感谢。

感谢龚向阳老师，您通过了我的研究生复试环节，使我有机会继续学习深造。您渊博的学识，严谨的治学态度，是我学习的榜样；在论文撰写阶段，耐心阅读我们的论文，给出修改意见，感谢您付出的时间和精力。

感谢阙喜戎老师，您是我们日常问题的解决者，我们有麻烦了都会找您，同时论文各个阶段的环节都是阙老师指导，感谢您的付出。

感谢我的家人，爸爸、妈妈、哥哥、姐姐，你们的支持让我心无旁骛的学习，今天毕业在即，感谢你们给的温暖。

感谢师兄师姐的指导，从你们手中接过参与式感知平台工程，经过耐心指导，从熟悉代码到可以开发新功能，薪火相传，感谢你们。

感谢参与式感知项目组三位同学，冯云、寇秦荔和杨婷婷（按例会发言顺序），有凌晨联调程序的苦，也有聚餐庆祝项目进展的开心，是同学更是好朋友，同甘共苦，巾帼不让须眉，感谢你们。

感谢师弟师妹交接任务的认真负责，虚心好学，感谢你们的传承，项目靠你们了。

最后感谢审阅论文的老师，感谢您的审阅和指导。

作者攻读学位期间发表的学术论文目录

- [1] 王东升. 无线双向中继系统用户调度算法研究[EB/OL].北京: 中国科技论文在线 [2015-12-03].<http://www.paper.edu.cn/releasepaper/content/201512-196>.