

密级： 保密期限：

北京邮电大学

硕士学位论文



题目： 基于参与式感知的激励机制的
研究与实现

学 号： 2012111445

姓 名： 赵露名

专 业： 计算机科学与技术

导 师： 阙喜戎

学 院： 网络技术研究院

2015 年 2 月 4 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）

非保密论文注释：本学位论文不属于保密范围，适用本授权书。

本人签名：_____ 日期：_____

导师签名：_____ 日期：_____

基于参与式感知的激励机制的研究与实现

摘 要

近年来，随着硬件技术的不断发展以及移动设备内置传感器的不断丰富，为移动协作感知的发展提供了良好的硬件基础。参与式感知要求用户参与到数据收集的过程中来，这无疑会消耗用户的时间、精力和金钱，无论用户是否因其上传的数据获得益处，用户的参与行为本身都需要合理的反馈和鼓励，否则用户很有可能丧失动力。以往的事实证明，合理的激励机制能够有效地提高用户的能动性，增加主观因素的影响提高数据的质量，这对参与式感知的发展有重要意义。

本论文的研究依托于移动协作感知这一国家自然科学基金课题。在总结现有参与式感知技术以及激励机制的基础上，设计了一个高可扩展性的参与式感知平台。它能够实时的采集用户数据，并在服务器端对用户数据进行存储、分析和可视化。本论文改进和实现了一种综合考虑数据数量、数据分布和预算约束的反向竞价激励机制，该算法增加了收集数据的数量、减少平台资金的投入、并使参与者获得更高的收益。本论文通过实现参与者之间的移动协作，减少参与者移动设备的资源消耗和上传数据的冗余，实现参与式感知平台和参与者的共赢。

首先，本论文介绍了参与式感知以及激励机制的研究背景和发展现状。其次，对参与式感知平台进行了需求分析和详细设计。然后，对激励模块的内部结构进行了详细的设计，并详细阐述了激励机制算法的原理与实现。之后，对系统的功能展示场景进行设计，并完成对激励模块的单元测试，从而表明系统达到预期效果。最后，对全文进行了总结，对未来的工作进行了展望。

关键词 参与式感知 激励机制 反向竞价 移动协作

RESEARCH AND IMPLEMENTATION OF INCENTIVE STRATEGY BASED ON PARTICIPATORY SENSING

ABSTRACT

In the last decades, hardware technology in the world has advanced with each passing day, resulting in a richer set of embedded sensors and larger capacity battery in smart phones. These changes have enabled new applications across a wide variety of domains and given rise to a new area of research called mobile phone sensing. Participatory sensing requires the user to participate in the process of data collection. During this process, participants will undoubtedly contribute their time, effort and money to fulfill the tasks. If participants cannot receive justifiable incentive, they will lose patience for our task. According to our experience, Reasonable incentive mechanisms can effectively improve the user's initiative to increase the influence of subjective factors and to improve the quality of the data.

This thesis' research based on a national natural fund project named mobile collaboration awareness. According to the basis of summarizing the existing participatory sensing technology and incentive mechanism, the author designs a participatory sensing platform, which can collect user data in real time, and finish the data storage, analysis, and data visualization on server. This paper improves and implements the incentive strategy algorithm, which take data quantity, data distribution and budget constraints into consideration. By using the new incentive strategy, our platform increase the number of data collection, reduce capital investment, and make our participants get higher profit. We realize the win-win of the participatory sensing platform and participants by implement the cooperation between participants.

First, this thesis introduces the related theories and technical background of participatory sensing and incentive strategy. Second, requirement analysis and system design of the participatory sensing platform. Third, the detailed design and implementation of each subsystem are highlighted. Then unit and

integration testing are adopted, which indicates that the entire system achieves the prospective design goals. In the end, the thesis summarizes the whole work of the system, looks forward to future work.

KEY WORDS : participatory sensing incentive strategy reverse auction
mobile cooperation

目 录

| | |
|------------------------|----|
| 第一章 绪论 | 1 |
| 1.1 研究背景 | 1 |
| 1.2 课题主要研究内容 | 2 |
| 1.3 主要工作 | 3 |
| 1.4 论文结构 | 3 |
| 第二章 相关技术和概念 | 5 |
| 2.1 参与式感知 | 5 |
| 2.1.1 参与式感知应用场景 | 5 |
| 2.1.2 参与式感知应用技术 | 6 |
| 2.1.3 参与式感知存在问题 | 7 |
| 2.2 激励机制 | 7 |
| 2.3 系统开发技术 | 9 |
| 2.3.1 系统中的通信技术 | 9 |
| 2.3.2 数据存储技术 | 14 |
| 2.3.3 系统开发运行环境 | 14 |
| 2.4 本章小结 | 15 |
| 第三章 系统的需求分析及总体设计 | 16 |
| 3.1 系统的需求分析 | 16 |
| 3.1.1 系统拓扑结构 | 16 |
| 3.1.2 功能性需求 | 17 |
| 3.1.3 非功能性需求 | 21 |
| 3.2 系统框架设计 | 22 |
| 3.2.1 系统物理框架 | 22 |
| 3.2.2 系统功能框架 | 23 |
| 3.2.3 系统分层设计 | 23 |
| 3.2.4 系统模块划分 | 24 |
| 3.2.5 模块逻辑关系 | 26 |
| 3.3 系统交互设计 | 27 |
| 3.3.1 客户端交互设计 | 28 |
| 3.3.2 服务器交互设计 | 29 |
| 3.4 系统原始数据生成方案设计 | 34 |

| | |
|--------------------------|----|
| 3.5 本章小结 | 36 |
| 第四章 激励模块的研究与实现 | 37 |
| 4.1 激励模块流程设计 | 37 |
| 4.1.1 服务器端流程 | 37 |
| 4.1.2 客户端模块调度流程 | 38 |
| 4.2 激励模块详细设计 | 40 |
| 4.2.1 激励模块子模块划分 | 40 |
| 4.2.2 激励模块类设计 | 42 |
| 4.2.3 激励模块数据库设计 | 42 |
| 4.3 激励机制算法研究与实现 | 44 |
| 4.3.1 激励算法研究 | 44 |
| 4.3.2 算法实现 | 48 |
| 4.4 激励模块功能模块实现 | 49 |
| 4.4.1 进程间通信模块实现 | 49 |
| 4.4.2 推送模块实现 | 51 |
| 4.5 本章小结 | 53 |
| 第五章 平台系统和激励机制的测试 | 55 |
| 5.1 测试环境 | 55 |
| 5.2 测试方法 | 56 |
| 5.3 系统功能测试 | 57 |
| 5.3.1 客户端基本功能 | 57 |
| 5.3.2 PM2.5 分析模型更新 | 58 |
| 5.3.3 客户端-Web 数据通信 | 59 |
| 5.3.4 Web 功能验证 | 61 |
| 5.4 激励模块测试 | 62 |
| 5.5 本章小结 | 66 |
| 第六章 结束语 | 67 |
| 6.1 全文总结 | 67 |
| 6.2 可继续开展的工作 | 67 |
| 6.3 研究生期间主要工作 | 68 |
| 参考文献 | 69 |
| 致 谢 | 70 |
| 作者攻读学位期间发表的学术论文 | 71 |

第一章 绪论

1.1 研究背景

近些年来,随着硬件技术的不断发展,移动设备的续航能力不断提升,内置传感器设备也不断丰富,使得移动协作感知^[1]这一课题成为可能。随着移动设备的普及,每个人对周围的环境都有丰富的感知功能,每一个内置了传感器(GPS、加速度、时间、声音、图像等)的设备都是一个综合的传感器平台。如果能够将城市中每一个人感知到的数据收集起来,可以形成强大的感知能力,这就是参与式感知。

如今已经有很多成功的参与式感知平台,例如亚马逊的 MTurk^[2] (Amazon Mechanical Turk)。一般来说参与式感知任务由三部分协作完成,任务发布者、系统平台、任务接受者。任务发布者可以在平台上发布参与式感知任务并且给出相应的激励,平台将任务广播给所有的参与者,接收任务的参与者在完成该任务后能够从平台上获得相应的回报。参与式感知技术与固定部署的传感技术相比具有诸多优点:1) 服务开发部署成本极低,贴近生活,便于大规模部署;2) 移动终端用户的流动性决定了参与式感知能够覆盖更大的时空范围;3) 与用户利益息息相关,用户不仅是参与式感知系统的使用者,同时也是内容和服务的提供者,更是最终的受益者。但是,参与式感知在现阶段的相关系统的实际部署和运行过程中存在着诸多不足,面临着巨大的挑战。由于移动节点的不确定性,在数据收集过程中会受到参与者的影响,参与者的文化和教育背景等差异,会对感知数据收集的精度、可信度、时延等造成影响。因此,应该在基础设施的层面上建立起一种数据认证、校验的机制来尽量规避这些影响,保证采集数据的可靠性。如何保护用户的隐私,如何在多维数据空间中利用稀疏的采样数据对缺乏统一模型的现象建模,如何通激励机制提高用户积极性、增加数据数量、提高数据质量以及评定用户的信誉都是这一课题迫切需要解决的问题。

参与式感知要求用户参与到数据收集的过程中来,这无疑会消耗用户的时间、精力和金钱,无论用户是否因其上传的数据获得益处,用户的参与行为本身都需要合理的反馈和鼓励,否则用户很有可能丧失动力。以往的事实证明,合理的激励机制^[3],能够有效地提高用户的能动性,增加主观因素的影响提高数据的质量。这对参与式感知的发展有重要意义。

最初的激励机制主要是服务器定价的激励机制。服务器提供固定的回报来收取用户的数据。该激励机制又包含按照提供数据时长报价和按照数据条数报价两种。该激励机

制的缺点是对于用户的吸引力较小,不能够充分发挥用户的主观能动性。如今热门的反向竞价的激励机制是服务器首先下发任务,用户上报其能接受任务的价格,服务器根据用户的反馈按照一定的算法选择能够参与的用户,待到用户完成任务,则发放事先约定好的激励。这种激励机制的优点是以用户满意的回报来调动用户的主观能动性,从而增强数据的可靠性。

如今在参与式感知中的激励机制大多是通过反向竞价的方式完成,需要一定的现金来支持,鉴于用户行为在参与式感知中所起到的重要作用,如何降低激励成本、保障数据质量、调整数据的时空分布已经成为了亟待解决的问题。

本论文设计了一个完整的参与式感知平台,该平台采用的激励机制将会对预算约束以及数据分布均衡条件进行综合考虑,以便达到降低成本、保证数据数量和分布的效果。

1.2 课题主要研究内容

本论文需要在实现参与式感知平台的基础上,对激励机制进行研究和实现的工作。因此本论文的主要研究内容包括:参与式感知平台的方案设计,激励算法的研究与实现,以及推送功能的研究与实现。

● 参与式感知平台的方案设计

本论文要设计并搭建一个基于 C/S (Client/Server, 客户端/服务器) 结构参与式感知平台,其主要有客户端和服务端组成,提供了数据的采集、存储、分析、展示四大功能。客户端提供收集光照、噪声、经度、纬度、螺旋加速器、电池电量等传感器数据和图片信息的功能,并提供相应的传感器读取接口;采用相关的采集策略来减少资源的消耗;客户端提供本地数据存储的功能,避免数据的丢失和浪费;同时,客户端还会提供数据的展示功能,来使用户得到更加及时的反馈。服务端提供了数据的接收、存储和分析的功能;为不同数据分析算法预留统一的数据读取和存储接口,使得整个平台具备更好的可拓展性;与此同时,服务端提供了将数据根据其经纬度信息整合并展示的功能。

● 激励算法的研究

良好的激励机制不仅仅对于参与者有良好的鼓励作用,还可以通过充分发挥参与者的主观能动性提高所收集数据的完整性和数量,扩大数据的时空覆盖范围。然而现有的激励机制在综合考虑数据分布均衡和预算约束方面存在不足,能够提供均衡分布的激励机制往往需要任务发布者提供高额的资金投入。本论文研究并采用了一种预算约束下的激励机制,该机制结合信息熵和期望的概念综合考虑数据分布情况和预算约束条件来发放激励任务激励,从而达到节约预算,调整数据分布的效果

● 推送功能的研究与实现

激励机制试验的实现需要消息推送技术作为支撑。本论文调研了在目前实现的针对 Android 客户端消息推送的四种方案: C2DM (Cloud to Device Messagin, 云推送) 云推

送方案, MQTT (Message Queuing Telemetry Transport, 消息队列遥测传输) 协议的封装, 基于 XMPP (Extensible Messaging and Presence Protocol, 可扩展消息处理现场协议) 协议封装, 以及第三方推送 API (Application Programming Interface, 应用程序编程接口) 服务。其中 C2DM 是谷歌提供的官方解决方案, 由于国内网络环境的限制服务极其不稳定, 不适用于在项目中采用。对 MQTT 和 XMPP 协议的封装工作量大, 不适宜在需短期上线的系统中应用。因此本论文采用国内第三方推送 API 服务。激励机制需要实现动态的用户分组推送, 第三方推送服务不能满足本论文的需求, 需要进行进一步的研究。

1.3 主要工作

本论文研究中, 论文作者主要负责的工作内容如下:

1. 技术调研和可行性分析: 充分了解参与式感知的研究背景、参与式感知平台的设计思路、激励机制的研究现状。同时, 对典型的激励机制算法进行学习。
2. 参与式感知系统的需求分析: 根据平台预期提供的功能, 对应用场景进行具体化, 完成对参与式感知系统的用例描述, 充分了解其应用场景。
3. 参与式感知系统的详细设计, 以及原始数据生成方案设计: 根据参与式感知系统的需求分析, 对系统进行整体设计; 完成前后端功能模块的划分, 细化各模块的职责以及调用关系; 设计与激励模块相关的调用接口。
4. 激励机制的研究与实现: 对激励机制在系统中的实现做出详细设计与功能实现, 研究现有的激励算法分析其不足并进行相应的改进。
5. 设计系统演示场景, 以及激励模块的功能测试: 为验证系统的完整性, 同时在尽可能少的场景下, 尽可能多的展示系统的功能, 对系统演示场景进行设计和实现, 通过演示素材的整理、编辑, 形成最终的演示方案; 完成激励模块的功能测试。

1.4 论文结构

本文的主要内容是参与式感知系统的设计以及激励机制的研究与实现。全文围绕着该系统的设计与实现展开, 首先进行了需求分析, 接着主要介绍各子模块的概要设计和详细设计, 最后给出了功能实现的测试工作。论文的组织结构如下所述:

第一章, 绪论。介绍了参与是感知以及激励机制的研究背景, 总体描述了本论文的研究内容, 并介绍了论文作者的主要工作。

第二章, 相关技术和概念。介绍参与式感知以及激励机制的技术背景和发展现状, 对于系统的实现所使用的技术进行相关介绍。

第三章, 系统的需求分析及总体设计。首先通过用例图说明的方式对系统的需求进行分析, 然后介绍系统总的模块划分、逻辑结构、交互设计等内容。

第四章, 激励模块的研究与实现。详细介绍了系统激励模块的调度流程, 内部结构, 数据库设计, 以及激励模块内部功能和算法的设计与实现的方案, 并对激励模块进行算法分析和验证。

第五章, 平台系统和激励机制的测试与验证。介绍整个系统的测试及展示方案设计, 并对激励模块的功能进行测试。

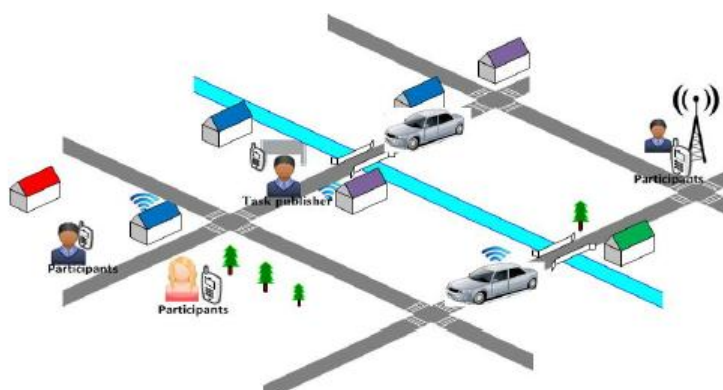
第六章, 结束语。对已完成工作进行总结, 并指出需要完善的地方和相应建议。

第二章 相关技术和概念

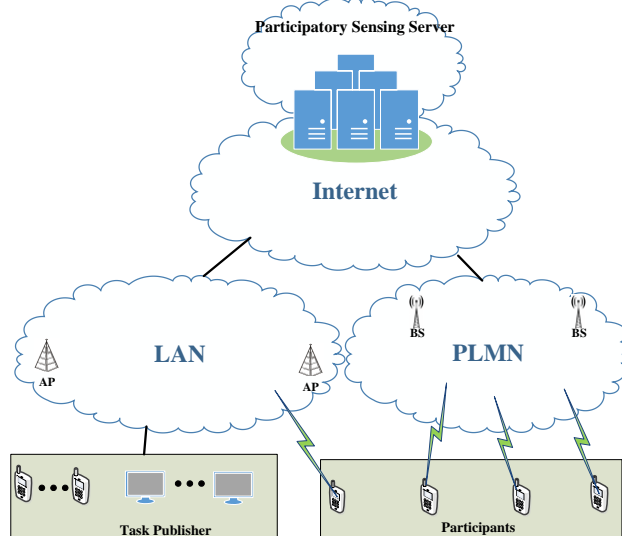
2.1 参与式感知

2.1.1 参与式感知应用场景

硬件的高速发展使得智能手机内置了丰富的传感器，给参与式感知提供了良好的硬件基础。智能手机近些年得到飞速的普及，仅国内智能手机用户数量都将突破 5 亿，给参与式感知提供了良好的用户基础。设计合适的参与式感知方案，智能手机就可以作为传感器节点和移动感知数据收集工具。参与式感知系统的应用场景如图 2-1（a）所描述。凡是用于内置传感器并且能够连接网络的设备，都能够成为参与式感知的参与者。



(a) 参与式感知的应用场景



(b) 参与式感知平台

图 2-1

参与式感知平台的组成如图 2-1 (b) 所示。它由三部分构成：任务发布者，参与者和服务器平台。任务发布者通过移动客户端和 Web 端发布任务信息（所需数据类型、数据的位置等信息）以及相应的激励信息。服务器在接收该任务以后，通过多播或者广播的方式，将任务信息发布给参与者，并且从所有表达参与意愿的用户中选取有限个参与者完成本次任务。参与者在接收任务后，到指定位置采集指定类型的传感数据，通过网络将数据上传到服务器。服务器通过对数据进行分析，给出激励。

2.1.2 参与式感知应用技术

参与式感知平台发布任务的目的是要获得可用、可靠的数据。数据的应用程度依赖于数据处理方式的科学性。而参与式感知平台数据的来源是参与者，参与者的广泛性和不确定性导致了数据完整性和可靠性的不确定。本论文设计的参与式感知平台着重选择数据融合、轨迹分析和激励机制三种技术对平台的数据质量和数据应用进行提升。

● 数据融合

参与式感知网络中参与者覆盖范围广泛，但是由于移动节点的移动性、随机性、分布不均匀和异构性等特点，将会导致数据的不准确、不均匀和不可靠，必须进行相应的冗余数据处理，修补缺失数据，最终得到真实可靠的数据，以保证后续工作的持续进展。

数据融合技术能够利用计算技术和通信技术中数据的时序特征对数据中的有用信息进行提取、分析、整合和过滤等操作，最终得到满足决策需求和评估任务的结果。根据不同层次对数据进行处理将数据融合技术分为三个层次，从低到高依次是：数据层融合，在拥有丰富的原始数据和高精度的详细信息时，可以采用。该层次数据融合原始数据容易受到噪声污染，耗时长实时性差，需要融合系统拥有很好的容错能力；特征层融合，实现了对原始数据的压缩处理，减少大量干扰数据。但是需要在处理前对数据进行预处理，将数据按照特征向量分组；决策层融合，所需要的信息量少，方法的容错性高，但数据的处理精度较低，同时数据处理的代价较大。综合考虑数据处理的代价和实效性，本系统中采用的数据融合技术属于特征层融合。

● 轨迹分析

参与式感知平台收集到的数据，结合位置信息能够进行深入的挖掘。从而能够提高系统激励的效率。轨迹是有一系列根据时间顺序连接起来的位置点序列，对轨迹数据的有效挖掘和分析，能够为未来的决策和数据应用提供有效的支撑。例如通过历史轨迹能够分析和掌握个体的生活规律，从而能够预测个体未来的轨迹分布情况。该预测结果能够帮助系统判断未来用户的分布状况，便于系统能够有针对性的安排参与式感知任务。

2.1.3 参与式感知存在问题

参与式感知具有数据时空分布广、数据丰富、不需要额外硬件投资等优点，是一个十分有潜力的研究领域。但它还有很多问题亟待解决，这是一个机遇与挑战并存的领域，吸引了大量的优秀人才投身其中，该领域的研究存在的亟待解决的问题主要集中于以下几个方面^[3,4,5,6]：

- 隐私保护

为了保证数据的有效性、可信性，参与式感知系统需要对用户的个人信息进行验证和记录，虽然可以通过模糊处理隐藏真实的个人信息，但如何保证这些个人信息的使用不会造成用户的困扰和不安，是所有参与式感知项目都必须面对和解决的问题。

- 用户资源优化问题

用户再上传数据过程中，需要耗费客户端资源，服务器对数据进行存储和分析也需要耗费服务器资源，若大量重复冗余数据上传，造成了双方的资源浪费且无益于数据的收集。随着众包思想的盛行，在参与式感知中应对用户协作进行考虑。

- 反馈与激励的问题

用户参与到活动中需要参与式感知任务中需要耗费经历，用户的参与行为需要合理的反馈和鼓励，否则用户很有可能丧失动力。不合理的激励方式将会造成数据质量并未提高并且出资方的经济损失严重的情况。要保持用户的积极性和能动性需建立合理的激励机制。

- 数据的有效性与可信性。

参与式感知比较传统感知方式的一大弊端就是许多时候信息受到用户主观因素的影响，未必可靠，节点的不确定性和网络情况的复杂性也增加了数据的不可靠性，这就要求后台处理程序有能力鉴别、剔除不可靠的信息。

2.2 激励机制

根据 2.1.3 节提出的参与式感知研究中所存在的问题，本论文投身于激励机制的研究，致力于解决激励与反馈这一重要问题。

所谓激励机制^[7]，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。激励机制应用到参与式感知项目中，参与式感知平台为激励主体，参与者为激励课题。它能够激励更多的用户参与到实验中来，并且对于参与式感知所收集的数据的可靠性也有提高的功能。目前对激励机制的研究已经到达很深入的阶段，有许多复杂的算法对其进行优化，并且通过综合考虑用户信誉、地理位置等信息对激励机制进行优化。鉴于用户行为在参与式感知中所起到的重要作用，降低激励成本、保障数据质量已经成为了亟待解决的问题。

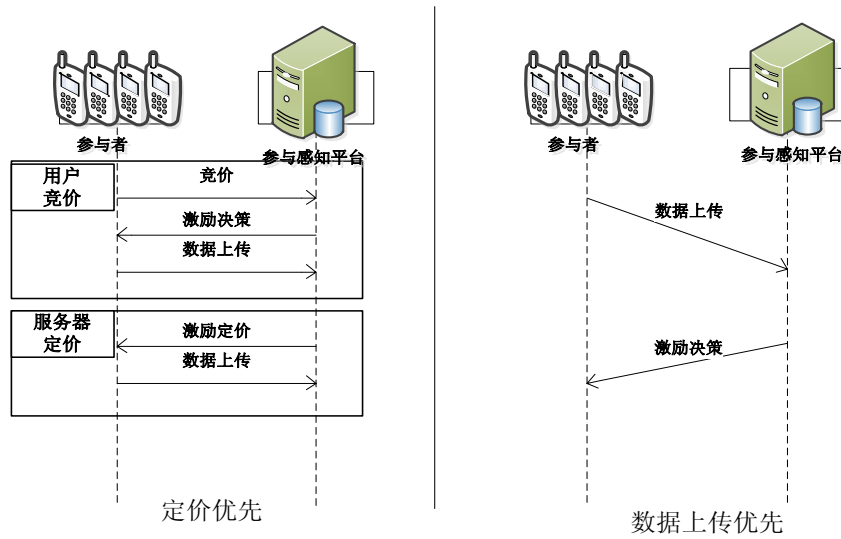


图 2-2 激励机制流程

现有的激励机制主要分为预先定价（Price-Decision-First）^[8,9]和数据上传优先（Data-Upload-First）^[10,11]两种。主要流程如图 2-2 所示。预先定价是指，在数据上传之前参与者已从服务器获得本次上传数据的价格，该价格可以通过竞价或者是服务器定价的方式获得。在参与者上传数据以后即可获得约定到的激励；数据上传优先是指参与者在数据上传之前并不知晓本次上传数据的价格，在数据上传以后服务器根据数据的质量以及数据稀缺程度等因素决定该数据的激励值，并发放给参与者。

表 2-1 激励机制分类

| 现金激励 | | | | 非现金激励 | |
|--------|-------|------|------|-------|------|
| 数据上传优先 | | 定价优先 | | 信誉奖励 | 排名奖励 |
| 竞价 | 服务器定价 | 动态定价 | 固定定价 | | |

影响参与者行为的因素有很多，根据激励机制的决策方式不同，其大致划分如表 2-1 所示。现金激励对参与者来说收益较为直接，根据定价时间的差异可以分为数据上传优先和定价优先两种。在数据上传优先的激励机制中，又可以分为用户竞价和服务器定价两种方式。在定价优先的方式中，分为动态定价和固定定价方式。非现金激励中有针对人的好胜心，对于赢的欲望进行排名激励，以及通过对用户对平台贡献进行评定的信誉激励方式。

研究表明，定价在数据上传之前能够是参与者提前评估参与任务所消耗的资源与回报是否合适，促使参与者合理的决定是否参与，从而提高任务完成率；反向竞价与服务器定价相比，更能够使得激励满足参与者的心理预期，从而提高参与者的参与程度；动态定价通过考虑数据质量、数据稀缺程度等因素对数据进行定价，能够促使参与者上传回报较高的数据，从而使得平台能够获得稀缺和完整的数据；非现金激励的方式，能够激发参与者参与热情，可以做为辅助手段，与现金激励方式相结合使用，达到意想不到

的效果。

2.3 系统开发技术

2.3.1 系统中的通信技术

2.3.1.1 HTTP 请求应答模型

HTTP^[12]-Hypertext transfer protocol (超文本传送协议) 是互联网上应用最为广泛的协议。它定义了浏览器向网络上的服务器请求资源以及服务器返回给浏览器资源的方式。通过 HTTP 协议请求的资源由统一资源标示符来标示, HTTP 是应用层的协议, 它是万维网进行可靠文件交换的重要基础。

HTTP 协议的主要有以下特点: 支持 B/S (Browser/Server, 浏览器/服务器) 模式, 应用广泛; 向服务器发起请求时只需传递请求方法和路径。通常采用 POST 和 GET 两种方式, 两种方式规定的客户端与服务器交互方式不同; HTTP 的服务器程序规模很小, 通信速度快; 传输类型可以通过 Content-Type 标记, 传输对象灵活可变; 无连接协议, 每次链接只能处理一个请求, 服务器处理完客户端请求并接收客户端应答以后即断开连接, 从而节省传输时间; 对事物缺乏记忆功能, 无状态协议, 因此信息若需要后续处理则不需进行重传。

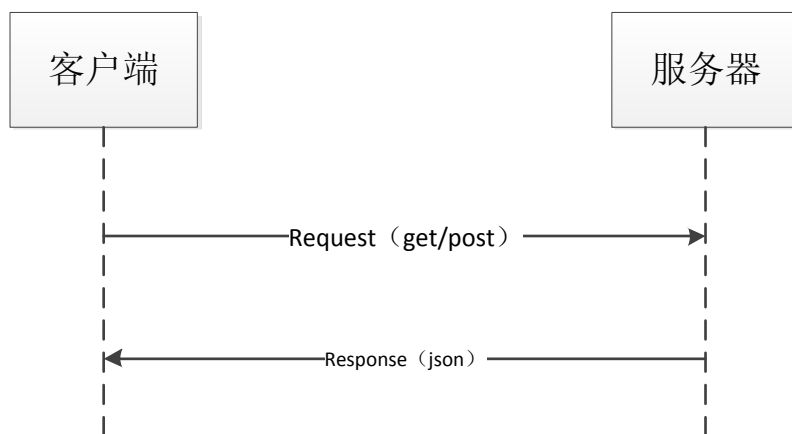


图 2-3 HTTP 请求应答模型

由于它的简单性、快速性和灵活性, 在系统设计时服务器与客户端间的通信采用了该协议。它的工作过程如图 2-3 所示, 客户端需要向服务器请求资源或上传数据, 则对服务器发起请求 (get/post), 服务器则向客户端返回对该请求的应答, 返回的资源可以有很多表达方式 (图片、文字、富媒体等)。本论文在数据上传和客户端验证的过程中都采用了 HTTP 协议进行通信。

2.3.1.2 消息推送

消息推送^[13]是实现客户端不定时被动获取服务器更新信息的一种通信方法。要获取服务器上不定时更新的信息一般来说有两种方法，第一种是客户端使用长轮询的方式，每隔一段时间就向服务器发起请求获取信息，查看服务器信息更新状况。该方法需要实现与服务器之间的通信，如果频率过低可能导致某些消息的延迟，如果频率过高，则会大量消耗网络带宽和电池电量。第二种就是服务器使用 Push（推送）的方式，当服务器端有新信息了，则把最新的信息 Push 到客户端上。短信消息推送在 Android 平台上，可以通过拦截短信消息并且解析消息内容来了解服务器的信息内容，但实际操作过程中成本过高，不适宜实验课题应用。持久连接的推送方案，是实现移动端平台消息推送的最佳选择。

2.3.1.2.1 C2DM 技术

C2DM^[13]是一个用来帮助开发者从服务器向 Android 应用程序发送数据的服务，其运行过程如图 2-4 所示。该服务首先通过 Android 的客户端向 C2DM 服务器注册标示客户端的 ID，然后将客户端用户名与注册的 ID 在应用服务器进行绑定。当应用服务器从网上获得推送给某个用户信息或用户信息发生更新时，应用服务器向 C2DM 服务器发送信息，C2DM 服务器将消息推送给客户端。经过以上过程完成了服务器主动下发信息的操作。

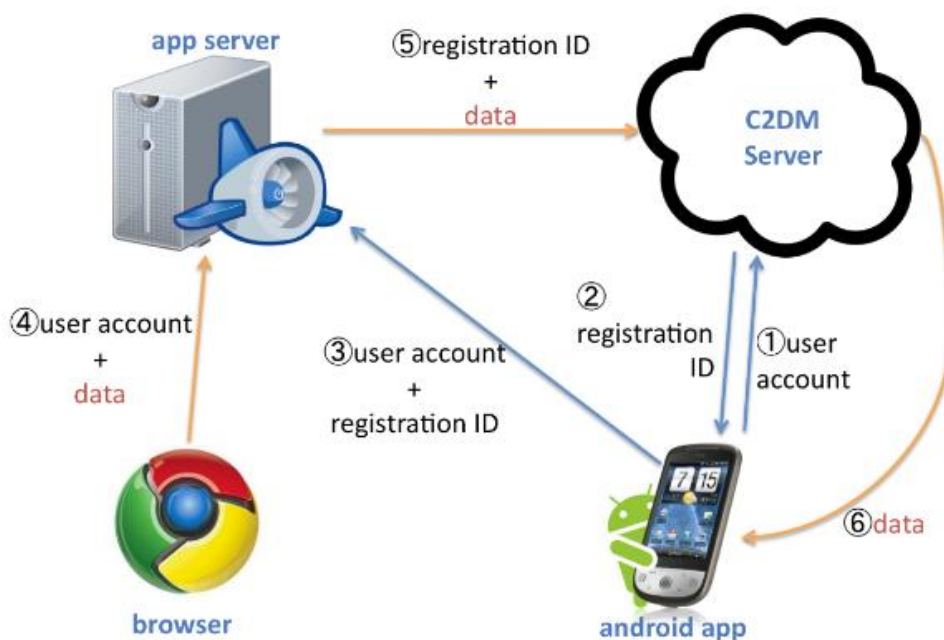


图 2-4 C2DM 操作过程^[13]

C2DM 是谷歌官方提供的消息推送机制，其资源消耗小，稳定性强，但其依赖于谷歌官方提供的服务器，由于国内网络环境的限制导致其经常不能访问，因此本论文无法选择该方案应用到工程实践中。

2.3.1.2.2 XMPP 协议

XMPP^[14]（可扩展消息处理现场协议）是基于可扩展标记语言^[15]（XML）的协议，它主要用于实现即时通信，通常所见的即时通信软件底层均采用该协议。这个协议允许互联网用户向互联网上的其他任何人发送即时消息，并且消息的传递不依赖于客户端的类型。**XMPP** 协议中定义了客户端，服务器，网关三个角色，通信能够在这三者的任意两个之间双向发生。服务器同时承担了客户端信息记录，连接管理和信息的路由功能。网关承担着与异构即时通信系统的互联互通。

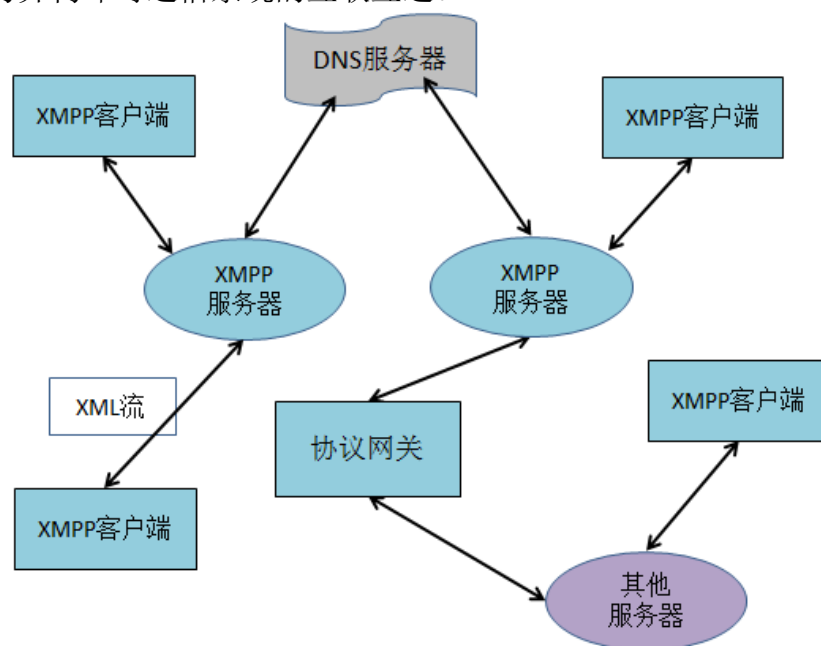


图 2-5 XMPP 工作原理

XMPP 工作原理如图 2-5 所示。一个 XMPP 客户端到另一个 XMPP 客户端的消息和数据都要通过 XMPP 服务器。客户端连接服务器，服务器根据本地目录系统的证书对客户端进行认证，客户端制定目标地址通过服务器获得目标状态。服务器进行 DNS 查找，并进行连接和认证。在连接建立后，客户端可以进行交互。

目前国内第三方提供的云推送服务，多是基于对 XMPP 协议的封装实现。考虑到 XMPP 应用的广泛性，综合调研评估各第三方推送服务的稳定性和时长反馈，本论文采用百度提供的云推送 API，并对其消息处理流程进行了改进，以满足本论文的功能性需求。

2.3.1.3 进程间通信

多进程和多线程在编程实现和性能上很大的区别。多进程间共享数据复杂，需要远程调用来实现，但数据分开存储导致数据同步比较简单，多线程由于能够共享进程的数据，故数据共享简单，但对数据同步造成了极大地困难；多进程需要为每个进程开辟内

存，内存消耗大，CPU 利用率低，多线程占用内存少，线程间切换简单因此 CPU 利用率高；进程的创建与销毁较为复杂，速度较慢，而进程创建销毁比较简单；在编程调试方面，进程具有显著优势，其编程调试难度远小于多线程编程；在可靠性方面，进程之间不会相互影响，而多线程会出现一个线程崩溃导致整个进程崩溃，系统无法运行的现象；多进程能够适用于多核多机的分布式系统，增强系统的可扩展性，但是多线程则无法支持多机的扩展。

表 2-2 多进程编程与多线程编程比较

| | 多进程 | 多线程 |
|-------|----------------------------|-----------------------|
| 数据共享 | 需考虑进程同步，较为复杂 | 可以共享进程数据，实现简单 |
| 内存消耗 | 为每个进程单独开辟内存，占用内存多，CPU 利用率低 | 占用内存少，上下文切换简单 |
| 创建与销毁 | 操作复杂，速度慢 | 操作简单，速度快 |
| 编程调试 | 进程独立性强，调试简单 | 线程相互依赖，调试复杂 |
| 可靠性 | 互不影响，可靠性强 | 单个线程崩溃可能导致整个进程崩溃，可靠性低 |
| 可扩展性 | 适用于分布式系统 | 不支持多级扩展 |

综合各方面考虑，为维护系统的稳定性和可扩展性，本系统采用多进程的方式。进程之间需要实现数据的共享、计算结果的传递等功能，因此需要实现进程间通信。本论文选择了远程调用的方式实现进程间的通信。

远程调用是指通过客户端进程调用远程服务端服务，等待应答。服务端的服务保持休眠状态直到调用信息到达为止，服务端在接收到调用信息之后，对服务进行执行，完成服务的计算后返回计算结果。

表 2-3 RMI 与 Http invoker 对比

| | RMI | Spring Http invoker |
|------|---------|---------------------|
| 语言 | Java | Java |
| 依赖框架 | 不依赖框架 | 依赖于 Spring 框架 |
| 应用难度 | 简单 | 相对复杂 |
| 访问限制 | 受到防火墙限制 | 不受限制，能够穿透防火墙 |
| 实现难度 | 步骤繁琐 | 能够使用代理工厂 Bean 简化配置 |

在服务器设计初期，经各模块需求分析，服务器选择 java 作为开发语言。因此在选择远程调用技术时，在 Java 本身提供的远程通讯协议 RMI 和 Spring 框架提供的 HTTP

invoker^[16]之间进行了比较。RMI 具有较高的稳定性，应用十分简单，不依赖于框架，只依赖 Java 语言本身，但其不能穿透防火墙，在实际应用中造成了很大困扰。而 HTTP invoker 其效率与 RMI 协议相差无几，但其通过 http 协议访问 Web 的端口的方式实现远程调用，能够良好的解决无法通过防火墙的问题。

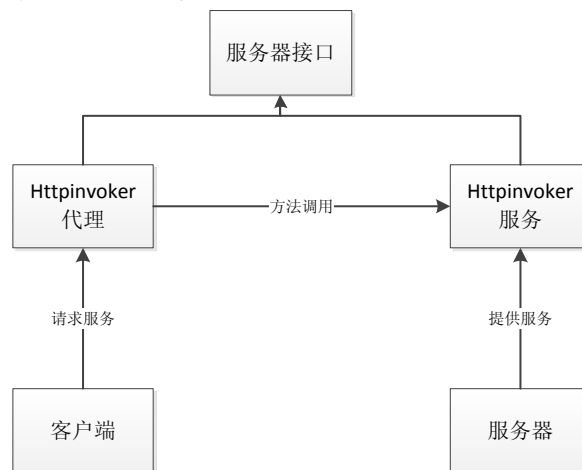


图 2-6 Httpinvoker 结构

HTTP invoker 是 spring 框架中的一个远程调用模型，执行基于 HTTP 的远程调用，并使用 java 的序列化机制在网络间传递对象。客户端可以很轻松的像调用本地对象一样调用远程服务器上的对象。其工作方式如图 2-6 所示。Spring 框架提供了一个代理工厂 Bean，使得我们能够将远程调用服务向本地 JavaBean 服务一样配置，极大简化了远程调用服务模型。

2.3.1.4 通信数据格式

JSON^[17] (JavaScript Object Notation) 是一种轻量级的数据交换格式。由于 JSON 是基于 JavaScript 的一个子集，这意味着在 JavaScript 中处理 JSON 数据不需要加载任何特殊的工具包（除了像 IE 6/7 这样的旧版本浏览器不支持 JSON 的解析和序列化，需要加载 json2.js 库）。相比 XML，JSON 容易阅读、解析速度更快、占用空间更少，C、C++、Java、Python、Ruby 等语言都有相应的 JSON 库支持。JSON 的这些特性使得它成为了前后端之间理想的通信格式，也是未来的流行趋势。

系统在设计中使用的通信方式，制定使用 JSON 格式传递数据的 API，不仅减少了单次数据传输字节流的大小，提升了用户体验，同时也便于扩展，让用户能够感受到更多由 Web 2.0 所带来的新特性和功能。

2.3.2 数据存储技术

2.3.2.1 客户端数据库

SQLite^[18], 是遵守 ACID 的轻量级关系型数据库管理系统, 它是针对资源紧张的嵌入式系统设计的。考虑到嵌入式系统的资源约束性, SQLite 数据库的系统资源占用率很低, 只需要几百 K 的内存既能满足其运行需求。它能够支持多种主流的操作系统, 并且支持多种语言的开发例如 C#, Java, Objective-c 等, SQLite 设计成一个进程内的数据库, 不需要单独的服务器, 避免复杂的配置, 使得它在移动端开发中脱颖而出。

由于他的灵活、资源消耗低、支持跨平台操作等特点, 在客户端我们选取它作为数据存储的数据库, 实现数据在客户端的持久化。

2.3.2.2 服务器数据库

HBase^[19] – Hadoop Database, 是如今最流行的分布式数据库管理系统, 具有良好的可伸缩和可扩展性。它提供非关系型的数据存储结构, 不必为维护数据关系牺牲效率, 仅通过主键(row key)和主键的 range 来检索数据, 主要用来存储非结构化和半结构化的松散数据。HBase 的计算和存储能力能够通过增加设备来提高, 这就避免系统的存储计算能力受到限制。

为将来大量数据的存储与读写做准备, 本论文设计系统服务器端选取了 HBase 作为数据库。其非关系型数据库能够支持高并发、高可用性的应用, 为系统的拓展提供了良好的基础。

2.3.3 系统开发运行环境

系统的开发环境如下:

- 后台编程语言: Java (JDK 1.7)
- 集成开发环境: Eclipse Java EE
- 服务器端框架: Spring MVC (HTTP invoker)
- 数据库: HBase、SQLite
- Web 前端编程语言: HTML、CSS、JavaScript
- Web 前端调试工具: Chrome 开发工具
- 版本控制: SVN
- 客户端编程语言: Java、android SDK

参与式感知系统和 Web 数据可视化服务器运行在 Linux 操作系统上, 保证了系统的安全性, 同时也提升系统的性能、增强可扩展性。Web 前端部署在了和服务器端同样的

环境中。

2.4 本章小结

本章着重介绍了参与式感知的概念以及应用场景，强调了参与式感知挑战与机遇并存的研究现状。对于本论文研究的重点激励机制的主要流程进行了详细的介绍，并且对现有激励机制根据不同原则进行划分，对现有的激励机制方案进行了简单的介绍。此外还介绍了对于项目实现方案 and 技术的调研、技术的选取，以及数据通信、数据存储技术的基本知识。

第三章 系统的需求分析及总体设计

首先要明确对参与式感知系统提出的功能性需求，根据系统应用场景以及所提供的功能对系统进行设计，然后需要明确对系统提出的非功能型需求，在系统的设计实现当中要对非功能性需求进行综合考虑。

3.1 系统的需求分析

参与式感知系统需要手机客户端和服务器共同构成。用户通过手机上传数据，服务器完成对用户周围空气质量、噪声状况、光照情况的统计分析，深入挖掘用户行为习惯，形成有用的数据统计结果集。为保证数据的可靠性，需要对多重数据循环校验，可能需要大量数据对结论进行佐证。在需要未获得数据时，可以通过激励机制鼓励用户参与，从而得到急需的数据。

3.1.1 系统拓扑结构

本论文所设计的参与式感知系统是移动设备采集传感器数据，对数据进行上传，在服务器上对数据进行存储和分析，用户通过上传数据获得奖励，并且能够对用户所拍摄的图片进行分析，得出该场景所对应的 $PM_{2.5}$ 值的系统。根据上述需求分析中所提到的功能性需求和非功能性需求，整个系统的网络拓扑设计如图 3-1 所示。本系统客户端和 Web 在访问服务器和数据库中的数据时，采用了开放平台的设计思想，通过 URL 来获取。

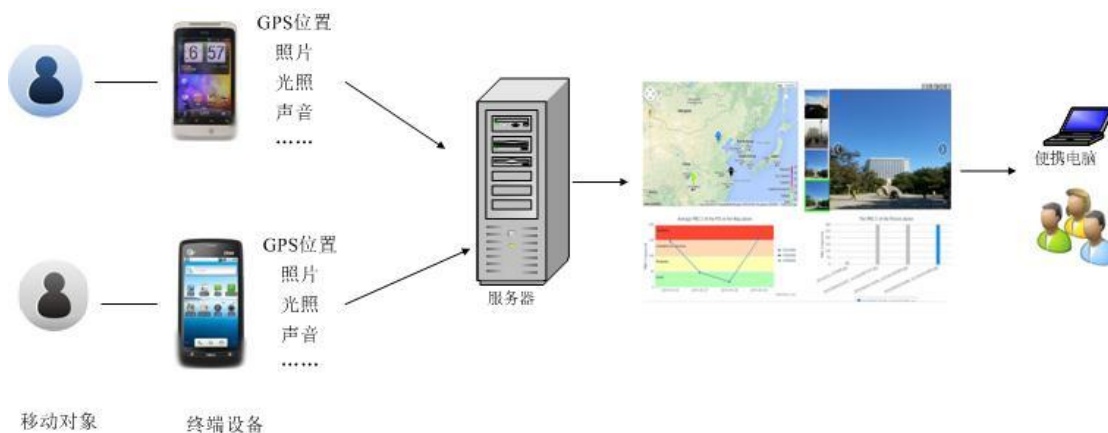


图 3-1 系统拓扑结构

该系统服务器主要包括参与式感知服务器和第三方 PM2.5 检测站数值收集服务器。参与式感知客户端既可以作为参与式感知的任务发布者，也可以作为参与者。任务发布者可以通过平板电脑、手机和电脑进行任务发布，任务信息可以通过因特网或者 GSM（Global System for Mobile Communication，全球移动电话网）网络传递到服务器上。服务器则通过网络将任务下发给所有的参与者。当移动端需要通过拍摄照片计算 PM2.5 数据时，移动端在建模过程中，需要使用第三方数据进行验证。为保证建模的顺利进行，参与式感知服务器会定时抓取第三方服务器数据，并进行存储，以确保在客户端需要数据时能够及时提供。同时参与式感知服务器，还需要对参与式感知任务的激励分配负责。他将根据现有数据的分布情况，以及参与者位置的分布情况，来决定参与任务的参与者以及所给出的激励。

3.1.2 功能性需求

参与式感知系统需要提供传感器数据采集、数据的存储、数据的应用与共享、以及数据的展示四部分功能。参与式感知需要客户端和服务端共同完成，分别负责不同的功能。本小结将对系统进行详细的需求分析，根据用例图，对系统功能进行说明。

3.1.2.1 客户端需求分析

根据对参与式感知系统的功能性需求进行分析，可知在参与式感知系统中需要客户端和服务端两部分，本论文设计的系统重点在于实现移动协作感知，因此客户端是实现数据收集的首要环节。为实现移动端的功能，保证用户参与的方便与快捷，降低用户的学习成本，我们对客户端的功能有如下需求：

首先客户端必须提供采集传感器数据的功能，这是参与式感知数据收集的首要条件，也是客户端的核心功能之一。其次，数据需要能够在客户端做持久化，避免由于客户端资源紧张、网络状态差等情况造成的数据丢失。此外，还要提供良好的通信功能，保证采集到的数据以及服务器的任务能够顺利的交互。

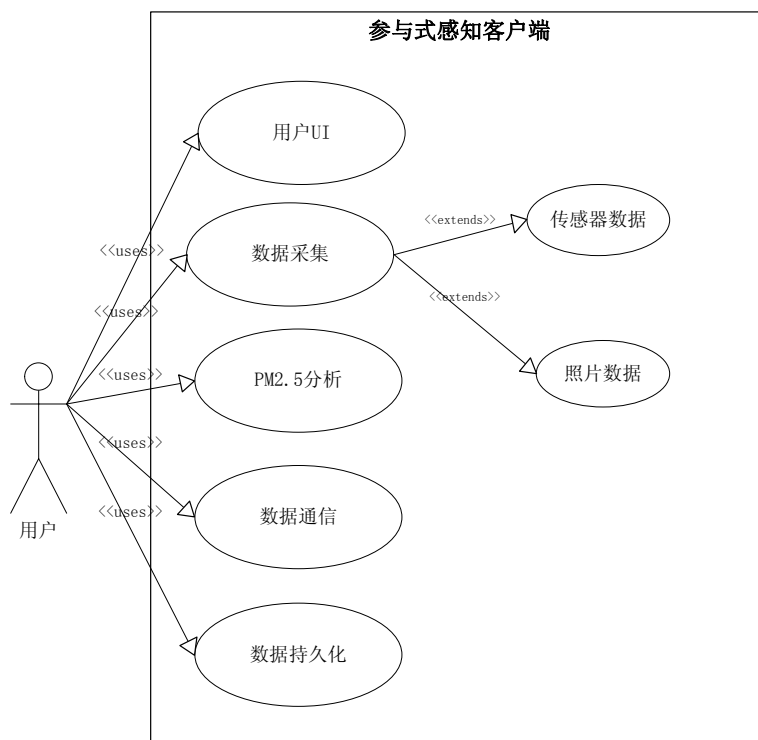


图 3-2 客户端功能用例

● 用户 UI

移动端应用，首先要提供良好的用户交互功能。简单易用的用户 UI，能够使用户投入最小的学习成本，快速上手，避免用户因操作过于复杂过多歧义造成弃用的情况。用户 UI 同时也是客户端其他功能的载体。用户 UI 控制了页面的逻辑，承载了其他的功能。

● 数据采集

对于参与式感知来说数据采集功能是必不可少的，根据移动设备现有传感器的状况我们所采集的数据有：GPS、光照、声音、电池电量、充电状态、螺旋仪值、网络状态、图片信息。根据采集的数据进行分析，能够充分感知用户所处的环境，用户的生活习惯等信息。数据的完全利用，对于轨迹分析等方向的研究，具有巨大的现实意义。

● PM2.5 分析

为了良好的利用所采集的图片数据，我们在客户端对图片进行了建模分析。根据对同一场景的建模能够计算出图片所对应的场景的 PM2.5 值。并且在采集照片的时候，会将其他的传感器信息一同写入图片的 EXIF 文件当中。

● 数据通信

数据在本地实现采集后，要上传到服务器进行存储和分析。因此需要提供数据通信的功能。数据通信功能又包括客户端主动上传和服务器主动下发消息两种。在客户端上传传感器数据时，采用客户端主动上传的通信方式，服务器会应答是否上传成功。在服务器发布任务时，采用的是服务器主动下发消息的模式。客户端通过对接收消息进行解析，从而展示。

● 数据持久化

数据的采集在客户端是定时或不定时进行的，在客户端采集的过程中用户所处的环境是不可预测的，客户端很可能不能及时的将数据上传至服务器进行存储。实时的将数据存储在本机，能够避免由于节点的不确定性造成的数据损失。为此我们需要在客户端做好数据持久化的工作。数据的持久化有很多种方式，可以将数据存储在内存中，或用键值对的方式存在硬盘上。由于数据量的不确定性和多样性，我们选用了数据库存储的方式。

3.1.2.2 服务器需求分析

服务器的主要职责是对数据进行接收、存储和分析，提供 Web 访问的接口，同时服务器还提供接口进行算法验证。

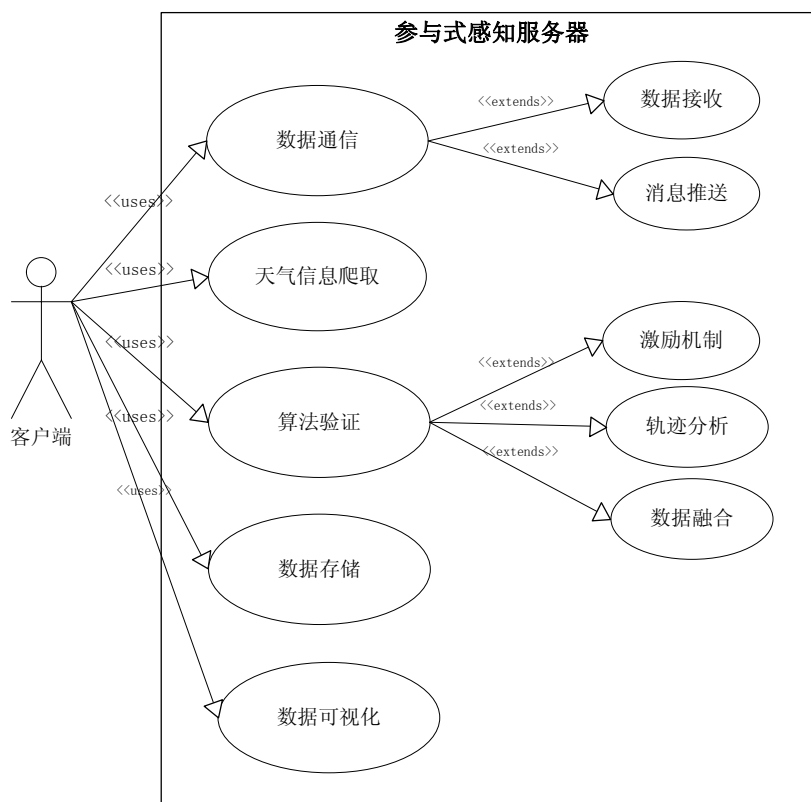


图 3-3 服务器功能用例

● 数据通信

服务器的数据通信功能，主要是接收客户端上传的传感器信息，在任务发布时推送任务信息给客户端。数据通信功能的稳定性确保了数据的完整性，是数据存储和应用的前提条件。由于客户端的数量是动态发生变化的，可能存在大量客户端同时向服务器发出请求的情况，这就对数据通信功能的性能提出了要求。它需要支持高并发的访问，并且完成对请求的迅速响应和数据的妥善存储。

- 天气信息爬取

天气信息爬取模块主要是为了配合客户端的 PM2.5 分析模块的需求。该模块主要完成对指定区域内 PM2.5 监测站点的监测数据的爬取。该数据将作为客户端 PM2.5 分析模块的对比信息，通过误差对比，对客户端的模型进行训练和调整。

- 算法验证

对数据进行采集，并不仅仅是为了将其存储在服务器上。为了达到物尽其用的目的，我们需要对数据进行分析 and 处理。在本系统设计初期提供了三个方向的算法验证和实现的接口。轨迹分析，是通过用户对用户的轨迹规律进行建模，从而能够得出在试验区域内用的分布情况；数据融合，能够通过数据库中已存在的数据，对数据稀疏的区域或时间进行数据补齐，从而达到数据在时空分布的均匀性；激励机制则能够通过一定的激励策略，鼓励用户参与，并对数据的质量提供一定的保障。这些算法的实验与验证，也是系统的重要组成部分，更是系统设计的重要目的之一。在多个算法实现与验证的过程中，也验证了服务器的可扩展性和接口的灵活性。

- 数据存储

数据的妥善存储，是数据应用的前提，因此对于数据存储的稳定性有较高的要求。本系统需要支持大量数据的存储与读写，对此需要设计合理的存储格式和读取策略。

- 数据可视化

本系统提供了数据可视化的功能，可以通过浏览器访问本服务器上的数据，并对图片、GPS 和 PM2.5 等信息进行展示。服务器数据通过经纬度信息相结合，在地图上同一位置对应的图片、光照和时间数据会通过网站加以展示。

3.1.2.3 激励模块需求分析

激励机制是本论文研究和实现的主要内容，也是参与式感知系统的重要功能之一，对于调动用户积极性，激励用户上传可靠的数据和稀疏的数据具有重要作用。合理的激励机制能够为任务发布者节约任务预算，并且为其获得时空分布更为均匀，数据可靠性更高的优质数据。本论文研究的激励机制包含了从任务发布到激励下发的整个过程，通过一定的奖励手段，来鼓励用户持续参与，而不是退出项目。在预算约束的情况下，获得最多的有效数据。激励模块的主要功能有任务发布、数据分析和激励计算。

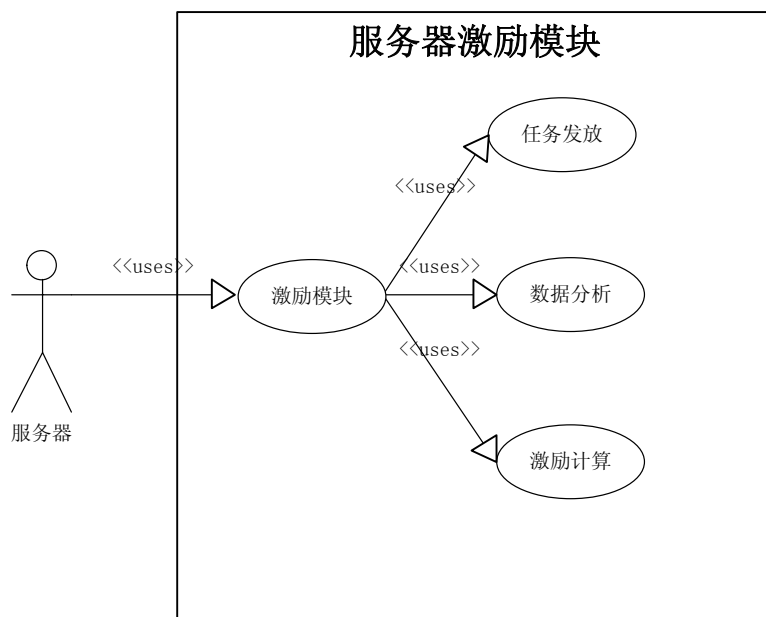


图 3-4 激励模块用例图

激励机制是服务器中提供的算法分析中的一种。要对用户进行有效的激励，首先要发放任务，通过消息推送的方式，将任务发送给一定数量/全部的用户。根据用户的反馈信息，分析用户的期望回报以及在约束的预算下用户可能参与的概率，并且根据用户的位置信息判断其数据的重要性（数据越稀疏的地方，该用户的数据越重要），这些信息都将在激励计算的过程中作为一定的比例。激励的计算还存在协商的过程，用户需与服务器进行协商。激励计算需迭代进行，计算所有的用户或实验分区发放激励以后，对数据数量和分布均衡性的影响，以期得到最优的结果。按照计算结果对激励进行发放。并且通过排名等方式，刺激用户争得前向排名的欲望，促使其共享更多的数据。

3.1.3 非功能性需求

对于整个系统来说，仅仅完成功能上的需求是远远不能满足系统正常运转的需求的。对系统的性能要提出更高的要求。

● 性能

随着用户数量的增多，突发大量数据同时上传的情况是存在可能的，系统需要支持高并发数据的上传和存储工作，就要求数据库能够快速的进行查询和读写，避免操作时间过长造成请求超时。同时 Web 端需要读取大量的图片信息，需要根据显示的方式，合理组织数据的存储，从而使得展示保持流畅。

● 稳定性

本系统存在大量数据通信，数据存储读取的操作，因此整个系统必须要稳定，最大限度的减少数据上传、下载失败的情况，减少系统崩溃的次数，从而能够保证数据的完整性和系统的正常运转。应提供容错处理、系统日志等功能，以便在系统发生崩溃时进

行问题追踪，避免在同样的错误上反复消耗时间。

- 可扩展性

本系统在客户端封装了读取传感信息的 API，在服务器端设计了多项算法验证的接口。一个开放性平台，为完成对数据的共享，必须要做好可扩展性的要求。在系统中添加新的功能或改进已有功能时，要保证整个系统能够平稳迭代，在保证新功能的正常运行的同时要保证原有功能不受影响。在系统设计时，要充分考虑各模块间的耦合程度、以及数据库对各部分的兼容性。

- 易用性

本系统的客户端，是提供给广大用户来为系统收集数据的。由于文化、教育背景的差异，个人对于客户端功能的理解存在重大差异。良好的用户交互设计，合理的页面布局，友好的交互设计能够降低用户的学习成本，提高用户体验，使其很快能够融入到整个系统当中。复杂的设计，将会使用户对该应用弃之不理，致使课题丧失大量的宝贵数据来源。

- 高效性

本系统存在大规模数据、图像上传的操作。因此系统必须支持高并发的接收和存储，避免由于效率问题造成的数据损失。

3.2 系统框架设计

3.2.1 系统物理框架

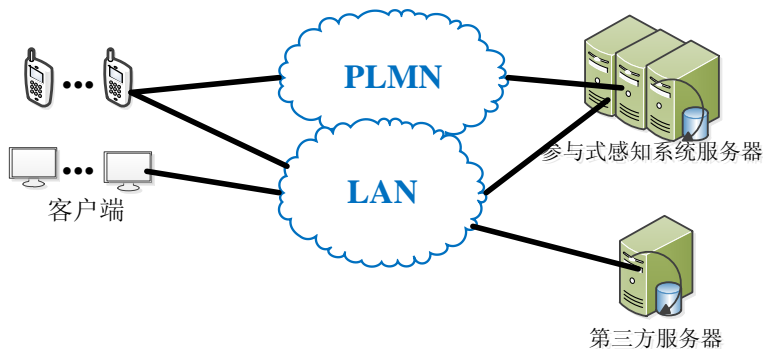


图 3-5 系统物理框架

参与式感知系统的物理结构如图 3-5 所示。系统的客户端有移动客户端和电脑两个部分。参与者可以用手机客户端可以通过城域网或移动互联网将收集到的数据上传给服务器并接收服务器下发的数据和指令。通过浏览器能够查询参与式感知系统服务器上的数据。参与式感知系统，能够通过互联网与第三方服务器进行通信，从而得到所需的数据。

3.2.2 系统功能框架

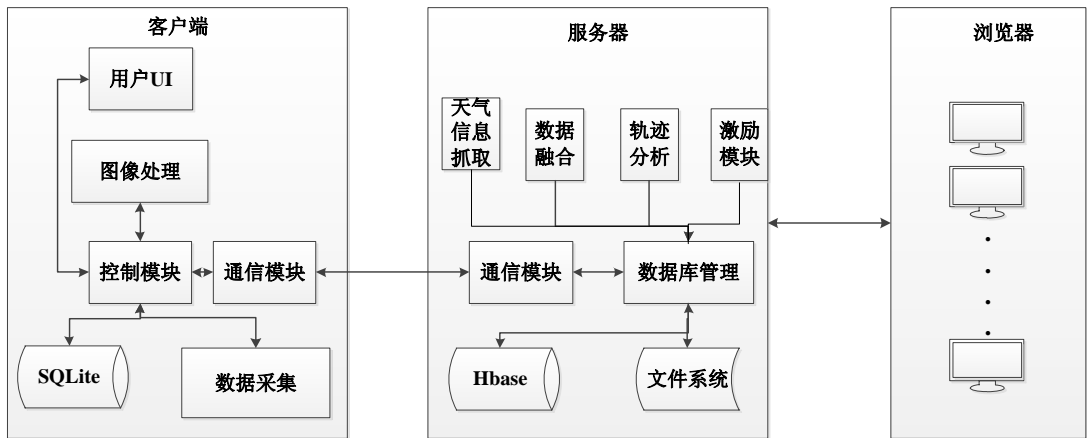


图 3-6 系统功能框架图

根据系统的功能性需求，将系统分为客户端、浏览器和服务端三个部分。图 3-6 展示了各部分的功能模块。

客户端需要完成数据的采集、数据存储、数据通信、逻辑控制、图像处理和用户 UI 这几大功能。在客户端各模块间的交互，通过控制模块调度完成。

服务器端需要完成数据通信、数据存储、图片存储、数据应用的功能。数据的增删改查等操作通过数据库管理模块统一管理。数据融合、轨迹分析、激励模块的算法验证通过数据库管理模块统一提供接口，采用插件式管理，便于服务器端的扩展。

浏览器，主要完成数据展示功能。采用 HTTP 协议的 GET 和 POST 方法，获得服务器的数据，并且通过浏览器加以展示。

3.2.3 系统分层设计

根据系统的功能框架，将该系统的模块分为：表示层、应用层、控制层、存储层和物理层。如图 3-7 所示，图左侧模块为客户端模块，右侧则是服务器模块。

表示层包括了客户端和服务器的展示模块，客户端通过用户 UI 进行展示，而服务器的数据则通过 Web 展示进行展示；应用层包含了客户端的 PM2.5 分析和服务器的各个算法模块，应用层的模块会对系统中的数据信息进行应用、分析和处理；控制层负责整个系统的逻辑控制和数据读写控制。在客户端逻辑控制与数据控制相分离，而服务器的控制模块主要是进行数据控制，各个模块间的逻辑关系通过远程调用来实现；存储层负责系统数据的存储。客户端应用轻量级数据库 SQLite 进行数据存储，服务器端除了数据存储外还需要存储相应的照片信息，采用了分布式数据库 HBase 和文件系统对数据进行存储；物理层是最接近硬件的部分，负责通过读取硬件获得数据采集部分，通过数据通信模块将服务器与客户端相连接。

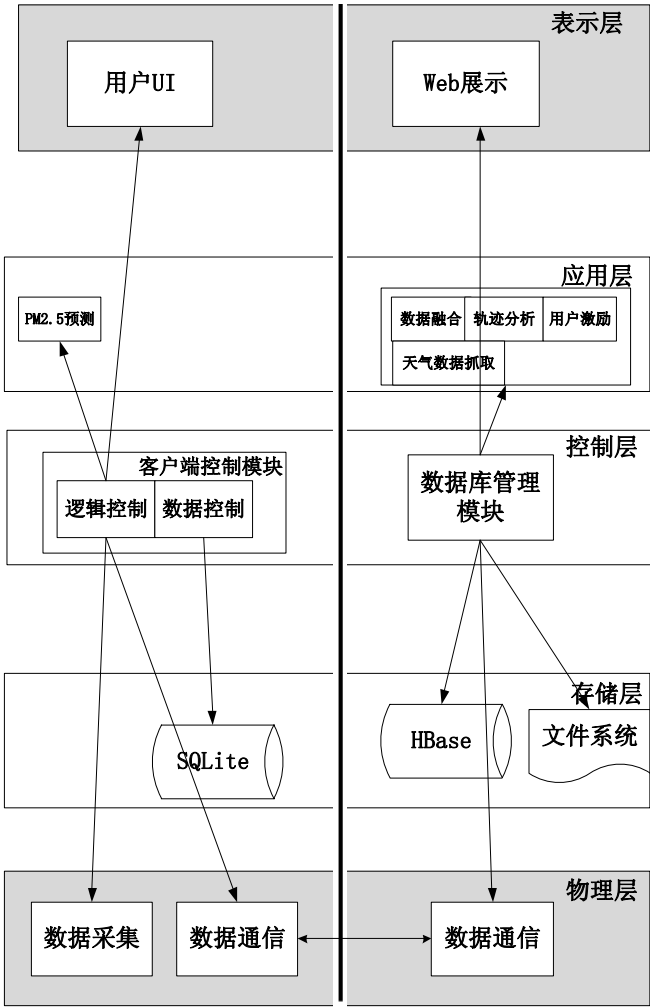


图 3-7 系统框架图

3.2.4 系统模块划分

根据系统的功能性需求，将系统主要划分为客户端和服务端两大模块。客户端主要完成数据采集、上传、存储的功能；服务器主要完成数据接收、存储、分析和展示的功能。该系统的模块划分如图 3-8 所示。

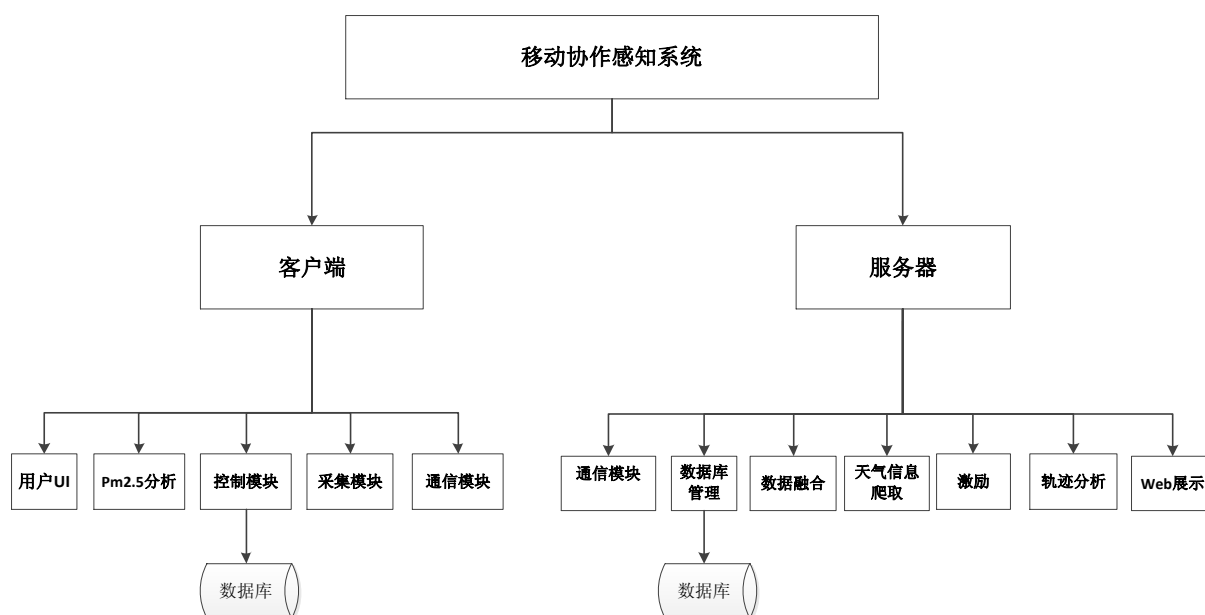


图 3-8 系统模块划分

- 客户端模块功能描述：

用户 UI：提供良好的用户交互，为用户提供简单易用的用户 UI，从而保证各项功能的良好运行，避免由于交互歧义对用户造成的困扰。对用户个人数据提供统计结果的展示，并根据服务器下发信息，展示其他用户的数据。

PM2.5 分析：根据拍摄图片，对场景进行建模，根据模型计算出该场景对应的 PM2.5 值。

控制模块：提供用户 UI 的逻辑控制，封装所有对数据库的读写操作，从而保证数据的透明性，避免由于多模块同步操作造成的数据不一致问题。

采集模块：封装客户端系统提供的传感器读取 API。根据用户意愿，读取移动客户端传感器数据。

通信模块：负责客户端的与服务器之间的通信功能，确保数据上传、下载的顺利进行。

- 服务器模块功能描述：

通信模块：负责服务器与客户端之间的所有通信功能，确保数据能够正确的接收到服务器端进行存储。支持并发访问。

数据库管理模块：封装所有的数据库读写查询操作，避免数据对外完全暴露。对数据存储负责。

数据融合：对客户端收集的数据进行区域性的补齐，使得数据在时空分布中尽量达到均匀。

天气信息爬取：爬取指定区域 PM2.5 检测站的数据，对于客户端的模型的计算结果进行佐证，同时为进行模型的训练提供数据支持。

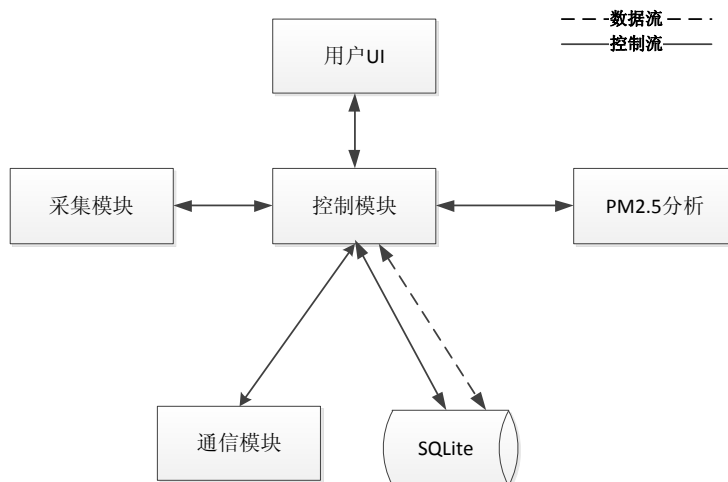
激励模块：对于缺失的数据类型和缺失数据的区域，通过给定报酬的方式激励用户长传数据，为系统增加用户以及数据均匀分布负责。

轨迹分析：对用户提供的地理位置信息进行分析，提供用户的运动规则，为后期提供基于位置的服务奠定良好基础。

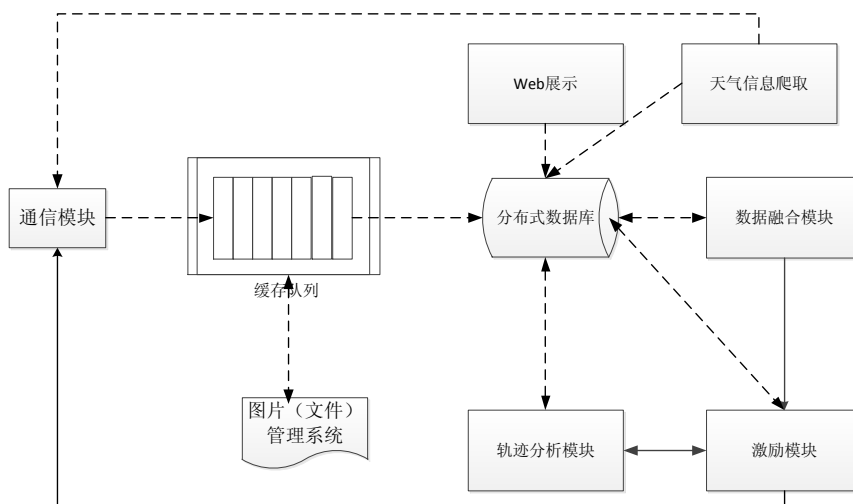
Web 展示：对采集的数据进行聚类、关联、展示。将数据通过 GPS 信息相关联，在地图上标注 POI 信息，通过该信息将图片和数据进行关联展示。为用户提供直观的展示功能。

3.2.5 模块逻辑关系

由于整个系统的复杂性，各功能模块各司其职，相互有着密切的联系。系统的逻辑结构如图 3-6 所示。客户端各模块通过控制模块相互联系，服务器端各模块通过分布式数据库管理模块相互联系。其中激励模块与轨迹分析和数据融合模块联系紧密，Web 展示模块和 PM2.5 信息抓取模块在服务器端独立运行。



(a) 客户端模块逻辑关系



(b) 服务器模块逻辑关系

图 3-9 逻辑结构

为了保证数据库操作的透明性，避免各模块操作数据为数据库带来的负担，我们在客户端和服务端都设计了数据库的管理模块，封装所有的数据库操作，为其他模块提供接口，因此系统数据流与这两个模块紧密相关。客户端的数据库管理功能包含在控制模块之中，控制模块除了完成对数据库操作的封装，还需协调用户交互的逻辑，以及各模块对通信模块的调用逻辑，从而完成客户端逻辑的控制功能。

客户端和服务端都有相对应的通信模块，通信模块主要负责客户端与服务端之间数据的交互。客户端的通信模块负责上传数据、图片信息和接收服务端返回的提示信息。而与之相对应的服务端的通信模块，则负责应答客户端的请求，并对客户端上传的数据信息进行存储。与此同时，为了完成数据推送的功能，服务端的通信模块还需负责对信息进行主动推送。

缓存队列是为了解决大量数据上传与分布式数据库写入耗时较长之间的矛盾而设计的。服务器首先将接收到数据存入缓存队列，然后按照数据的类型分布存入数据库和文件管理系统，并记录图片文件的存储路径。

数据融合模块、激励模块和轨迹分析模块之间联系紧密，三个模块之间存在调用关系。数据融合模块能够完成对数据分布的统计和缺失数据的补齐工作，轨迹分析模块能够给出相似轨迹的记录以及用户的转移矩阵（指用户在某时刻从 A 地到 B 地的概率）。根据这两个模块的计算结果，能够得出在数据分布稀疏甚至缺失的区域可能到达的用户有哪些。激励模块在制定激励策略之时，能够参考两个模块计算结果进行激励，从而达到更好的激励效果。

3.3 系统交互设计

系统功能模块复杂，各模块之间有紧密的联系，各个模块间存在复杂的消息交互流

程。本节对客户端和服务端各部分的消息交互进行详细设计。

3.3.1 客户端交互设计

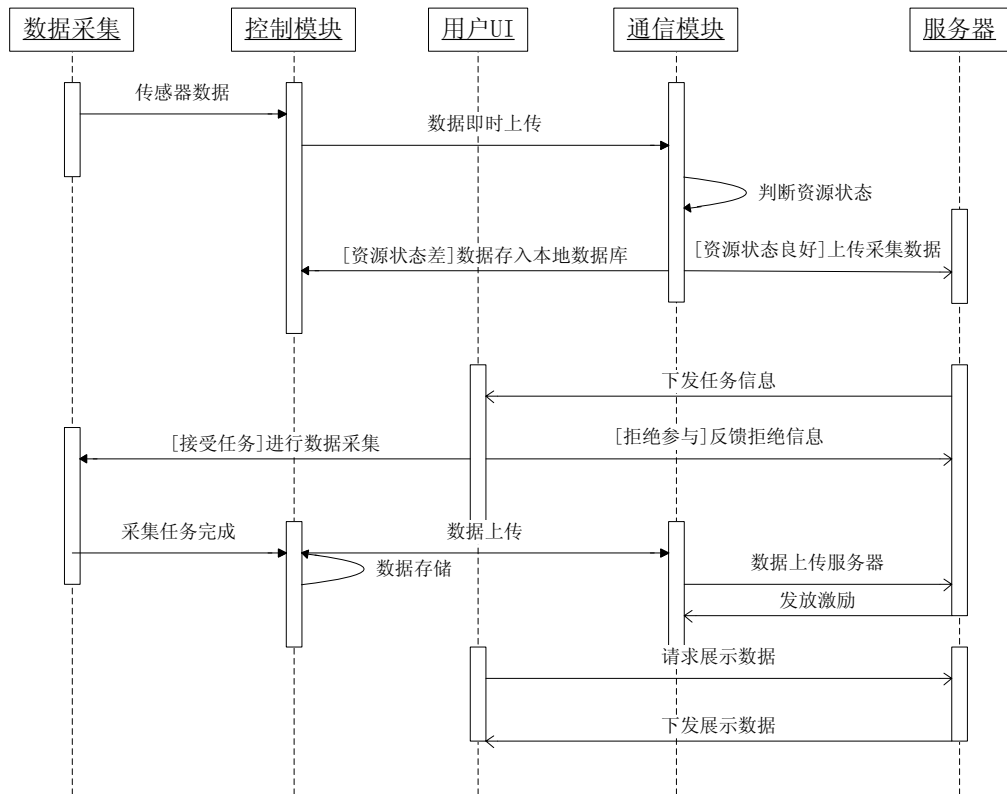


图 3-10 客户端消息交互

客户端对于传感器数据的收集方式分为了主动采集和被动采集两个方式。主动采集则在采集数据以后立即调用通信模块进行网络检测，在网络状态允许的情况下，对数据进行上传，如果网络状态不允许，则需要将采集的数据通过控制模块存储到本地数据库。被动收集，则是直接将数据存入本地数据库，等待数据库定时上传开启对数据进行上传。主动采集一般是由于服务器端下发的任务信息驱动其完成的。在客户端接收到服务器的任务信息时，首先需要通过用户界面对任务进行显示，然后用户可以根据自身情况选择是否接受本次任务。如果用户接受任务则变为主动采集的过程，如果用户拒绝本次任务，则直接反馈给服务器拒绝信息本次任务过程结束。

用户 UI 的部分会根据不同界面的功能设计不同的展示。其对本地信息的展示则需要通过控制模块读取数据库中存储的数据信息；若要展示服务器端的数据信息，则需要向服务器发起请求，等待服务器响应的信息并进行展示。

3.3.2 服务器交互设计

3.3.2.1 服务器响应客户端查询交互设计

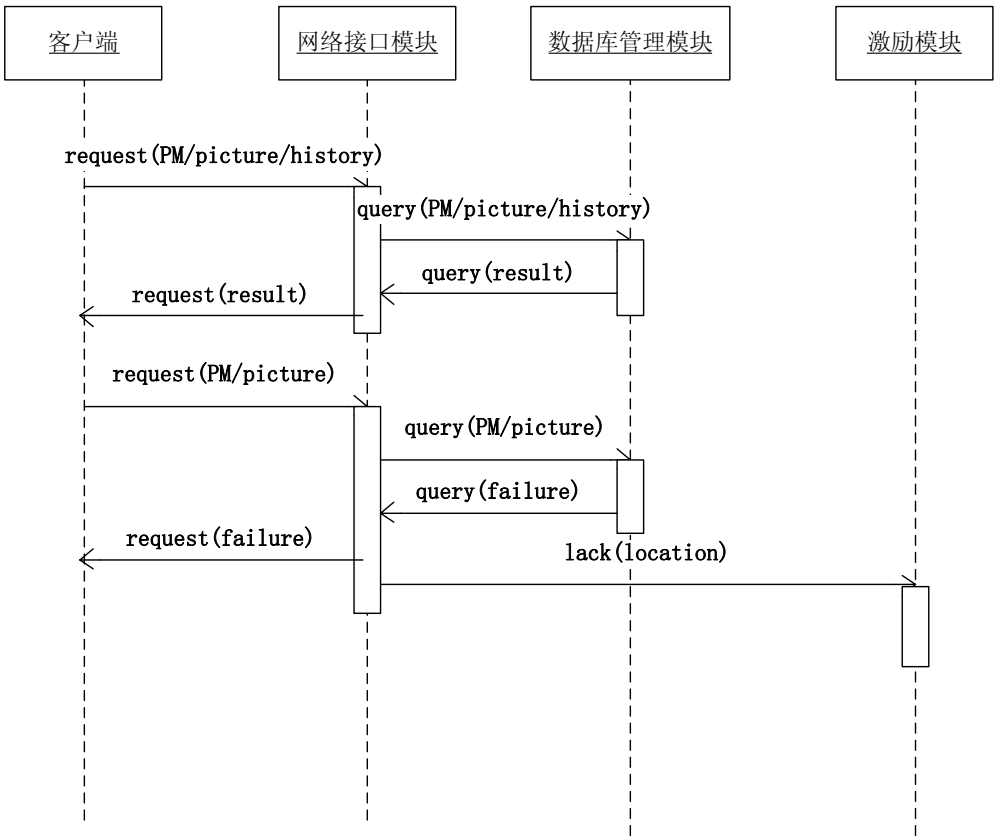


图 3-11 查询消息交互图

客户端和 Web 页面为显示服务器上的图片数据、PM2.5 数据和经纬度等信息，需要向服务器发起请求。服务器的通信模块在接收到该请求后，会根据请求约定的名称调用不同的方法，通过数据库管理模块向分布式数据库发起查询操作。在得到查询结果之后会将结果返回给客户端或 Web 端。前端能够根据返回数据进行显示。若客户端或 Web 页面请求的数据在查询时发生失败，通信模块能够返回响应的错误信息，以方便前端的后续处理的进行。

该过程在服务器端主要是由通信模块和数据管理模块共同完成。在查询过程中发现数据缺失会驱动激励模块的任务发布操作。

3.3.2.2 用户身份验证交互设计

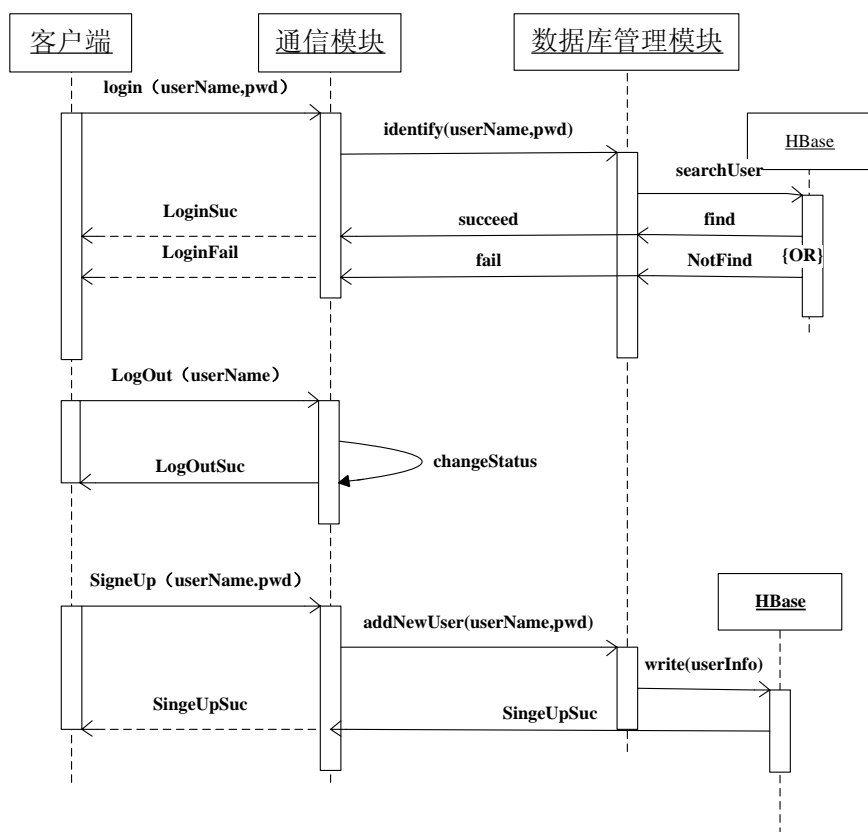


图 3-12 用户身份验证消息交互图

参与式感知系统要对用户提供的数据加以分析，因此需要对用户进行区分。用户的登录、注册和身份验证就成为了系统中必不可少的环节。用户身份验证的消息交互过程如图 3-12 所示。

客户端发起登录请求，携带参数用户名和密码到服务器端验证，通信模块根据方法名判断需要调用数据库管理模块进行查询。数据库管理模块通过在 HBase 中查询该用户的信息，返回登录成功/失败的信息给客户端。

客户端发起注销请求，只需要到服务器端更改用户的登录状态即可，不需要与数据库发生交互操作。

新用户注册，需要用户设置用户名密码。服务器需要检测用户名是否可用，若不可用则需要用户重新选定用户名，若用户名合法则在数据库中增加新的用户信息，并返回用户注册成功的消息给客户端。

3.3.2.3 上传数据交互设计

本系统中，客户端上传的数据分为 JSON 格式的字符串数据和 PNG 图片数据。对于两种数据格式，会产生不同的交互流程。下面分别对 JSON 数据和图片数据的交互流

程进行设计。

● JSON 数据上传交互

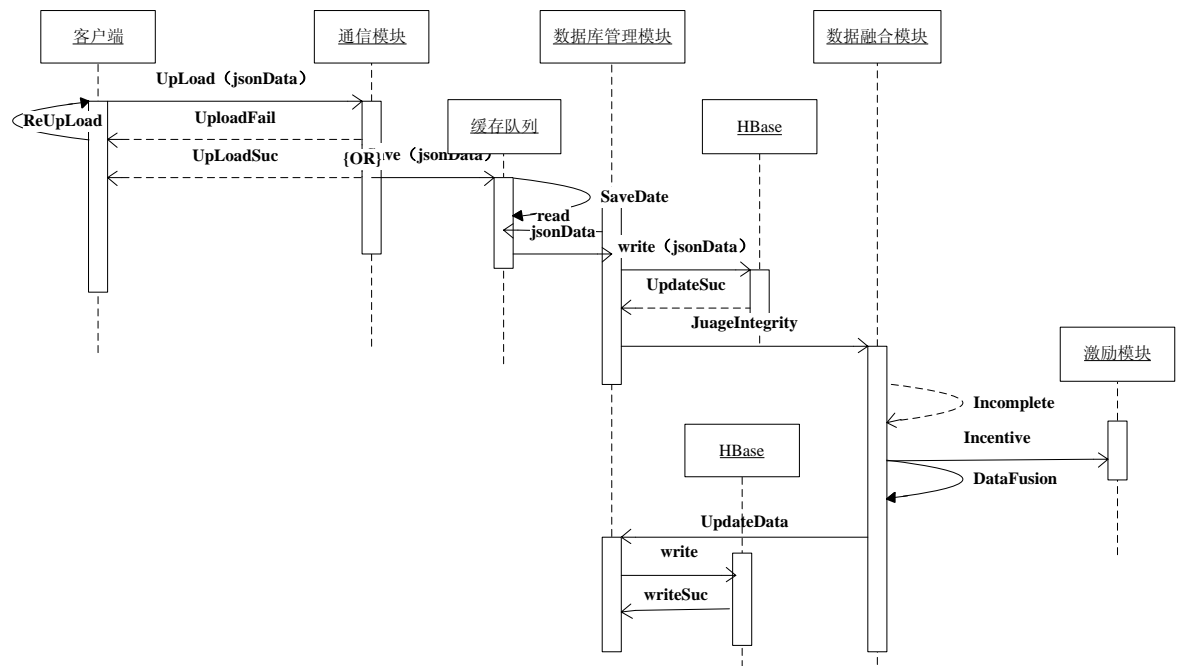


图 3-13 数据上传交互流程

客户端上传 JSON 数据，需要得到服务器返回的上传成功/失败信息并在客户端进行展示，告知用户本次操作结果。若上传失败，需要请求客户端对数据进行重传；若上传成功则可以进行后续流程。

由于分布式数据库写入速度与大规模客户端上传数据速度的差异性，服务器在成功接收数据后，先将数据写入缓存队列。数据库管理模块会从缓存队列读取数据，并在数据中进行存储。在数据存储完成后，需要通过数据融合模块进行数据完整性的检测。数据的完整性有单条数据完整性和数据区域完整性两个方面。在遇到不完整数据时，数据融合模块将会调用数据融合算法进行数据补齐和数据插值，同时会调用激励模块来获得更丰富的数据。

● 照片数据上传交互

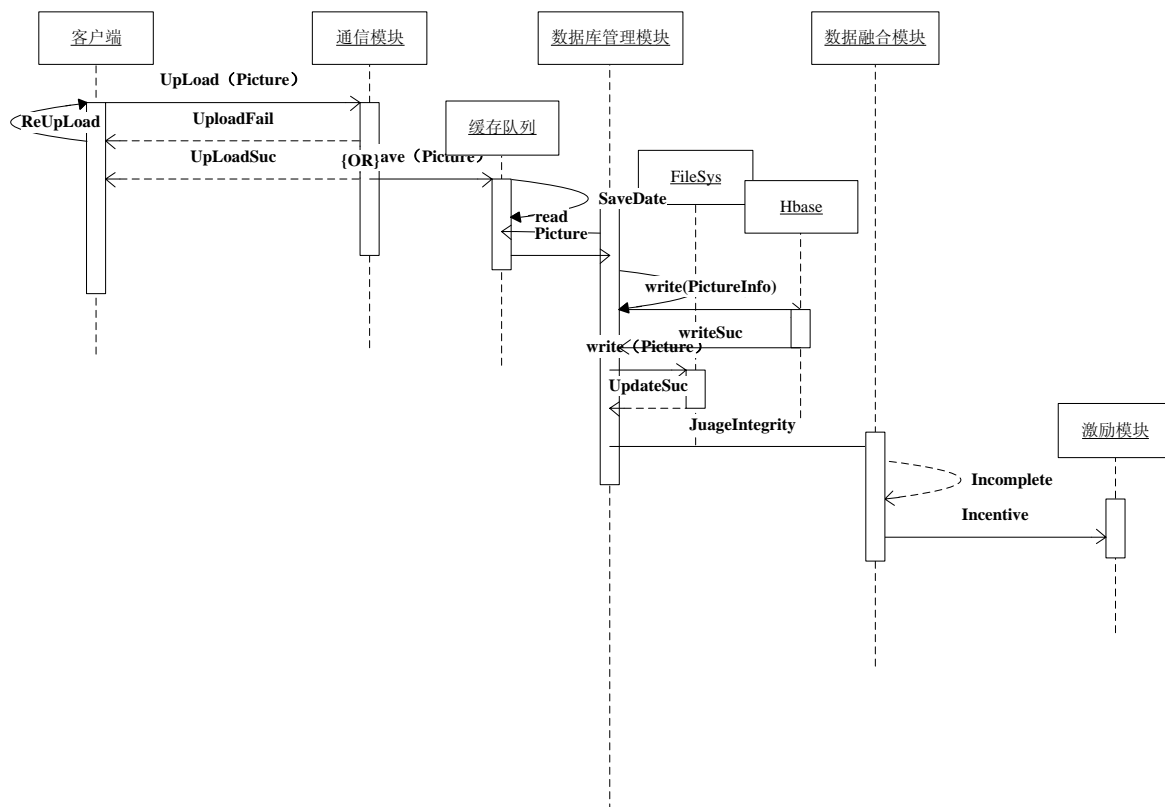


图 3-14 照片上传交互流程

照片数据上传的处理过程相较 JSON 数据上传更为复杂，交互流程如图 3-14 所示。照片的上传同样需要在上传失败时提供重传操作。

在客户端进行拍照时，将该照片拍摄时周围环境的 PM2.5、光线、噪声、时间和经纬度信息按照规定的 JSON 格式写入了照片的 EXIF 信息中。因此在照片上传成功后，数据库管理模块首先需要对照片信息进行解析，然后再将照片文件及其压缩文件存储数据库管理系统，将照片存储的路径、名称以及照片中所对应的 JSON 数据存入分布式数据库中。最后需要计算照片文件对应的 POI 信息，为 Web 展示做出相应的准备。

与数据上传相似，在上述计算和存储操作完成后，需要通过数据融合模块来判断数据的完整性。由于照片数据的不可融合性，若出现照片缺失的情况则只能够通过激励任务的方式获得更多的照片。

3.3.2.4 激励交互设计

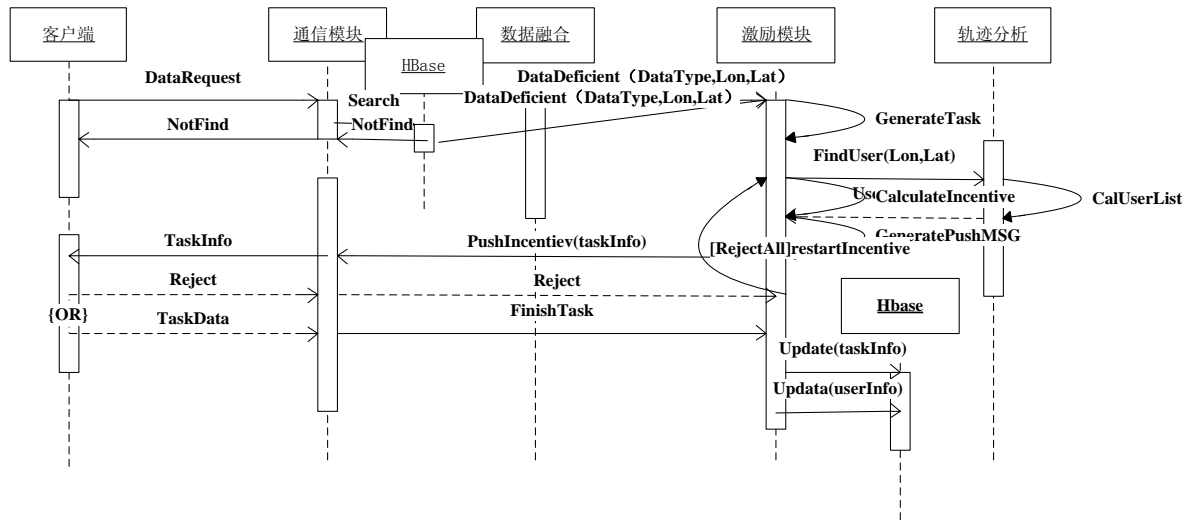


图 3-15 激励任务交互流程

激励任务的下发是参与式感知系统的重要工作之一。该任务的生成与下发在服务器端需要激励模块、轨迹分析、数据融合、数据库管理和通信模块的相互协作。任务的生成和反馈都需要复杂的计算过程。

激励任务的发起有两种情况驱动：数据融合模块在检查数据完整性时发现不可补齐的缺失（区域性缺失或数据类型缺失）；用户在请求数据时，发现数据缺失。根据缺失数据类型、数据位置请求激励模块生成激励任务。

激励模块将缺失数据的位置发送给轨迹分析模块，以便获得该区域可能出现的用户列表从而增加任务完成的可能性，与此同时，需要根据激励算法来计算本次任务的激励强度。得到轨迹分析模块返回的用户列表后，激励模块需要在数据库中存储本次激励任务的相关信息，并通过通信模块对任务进行下发。

客户端在收到本次任务后，可以选择参与或者拒绝。若用户拒绝本次任务，则直接将信息反馈给服务器，本次任务中客户端交互完成。激励模块需判断是否全部用户都拒绝参与本次任务，如果是则需重新计算激励强度并且增加任务下发范围，如果不是则可以继续等待其他用户的反馈。用户若选择参与本次任务，待到客户端上传所收集的信息后，激励模块根据约定计算本次任务的激励强度，并且在数据库中更新相关信息。

3.3.2.5 Web 展示交互设计

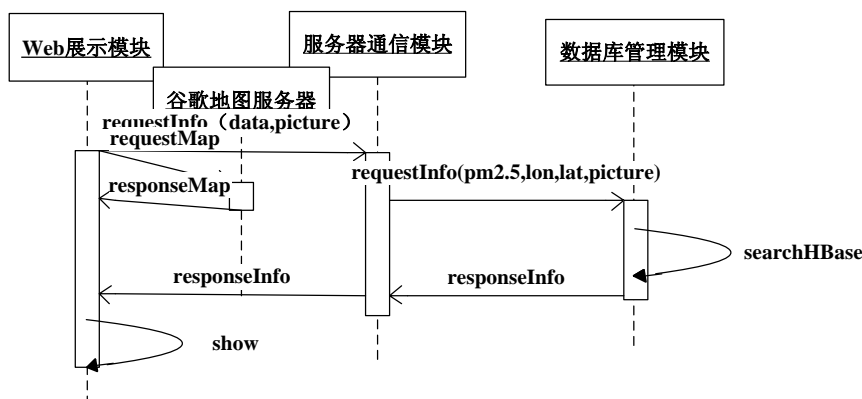


图 3-16 Web 展示交互设计

Web 展示部分，需要展示地图和地图展示范围内的 POI 所对应的 PM2.5 数据近期走势、照片及其所对应的 PM2.5 信息。在发起 Web 展示请求后，页面通过 JS 向谷歌地图服务器请求地图数据，同时向参与式感知服务器请求地理位置信息和相对应的图片、数据信息。服务器通信模块在接收到该请求后，通过数据库管理模块向数据库发起查询请求，然后返回给 Web 端查询得到的信息，Web 端将信息跟地图上的信息相结合对数据进行展示。

3.4 系统原始数据生成方案设计

● 数据库数据生成

为使得系统能够正常运转。需要为数据融合模块和轨迹分析模块以及 Web 展示模块生成原始数据。在系统搭建之初，未进行真实数据收集之时，我们将 GeoLife[21]中试验区域的轨迹数据、无锡物联网科研项目提供的传感器数据以及搜集到的图片数据相结合，形成了初始的数据集，供系统各模块使用。

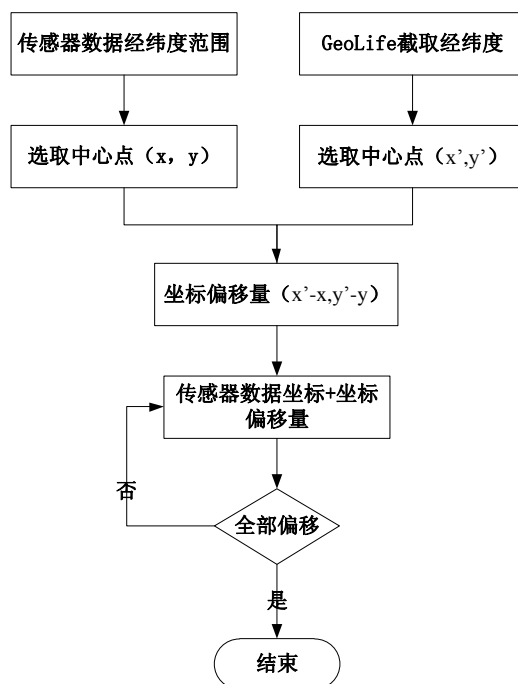


图 3-17 数据偏移流程

GeoLife 数据为本论文提供了位置信息，其存储了用户名、轨迹编号、经度、纬度、时间。传感器数据包括湿度、温度、时间、节点经度、纬度信息。为使得经纬度信息相对应，选取的位置信息区域大小与传感器数据的区域大小相同，选取区域中心点经纬度 (x, y) ，使得传感器信息中每个点与该中心点经纬度相减得到位置偏移。选取位置信息的中心点，用传感器信息中的经纬度偏移与位置信息中心点相加即得到了迁移以后的经纬度。从而将传感器数据，迁移到了试验区域。

● 照片数据

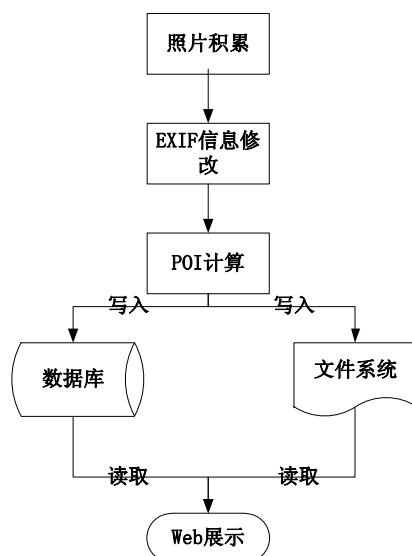


图 3-18 照片数据生成流程

照片数据的 EXIF 信息进行填充，增加相应的传感器信息。在数据库和文件系统中写入修改后的信息。计算照片所对应的 POI，为 Web 展示做好准备。

照片数据需通过地理位置信息与其他信息相结合进行展示，在系统搭建初期，将图片数据从属的 POI 信息固定写入数据库中，其从属于固定的经纬度，从而使得其能够与原始数据相结合展示。

3.5 本章小结

本章首先对整个系统的总体功能性需求进行了说明，并对系统提出的非功能性需求做出了初步阐述。通过对主要工作场景进行了分析和用例说明，总结出系统应具备的基本功能。然后对系统的框架进行了设计，并对该系统进行了模块划分，将各模块的基本功能、逻辑关系以及各模块间的消息流程做了简单的介绍，为接下来激励模块的详细设计的介绍做出了准备。

第四章 激励模块的研究与实现

激励模块是系统中重要的子模块之一。为实现激励模块的功能，并增强模块的可扩展性，将该模块分为消息推送、激励算法、进程间通信三个子模块。下面将详细介绍激励模块的设计、研究与实现。

4.1 激励模块流程设计

激励模块的作用是为参与式感知任务获得更多的用户或者是让已经参与到项目中的参与者更积极地为系统提供所需的数据。其应用场景有两种：1) 数据融合失效，需要用户提供某个地域的真实数据；2) 用户/系统请求数据缺失，需要通过激励其他用户上传获得。

激励模块的运转，需要服务器端和客户端的配合才能完成，模块流程也分为服务器流程和客户端流程两个部分。

4.1.1 服务器端流程

在整个参与式感知系统的服务器端，与激励模块存在调度关系的模块主要有：轨迹分析模块、数据融合模块、网络通信模块。

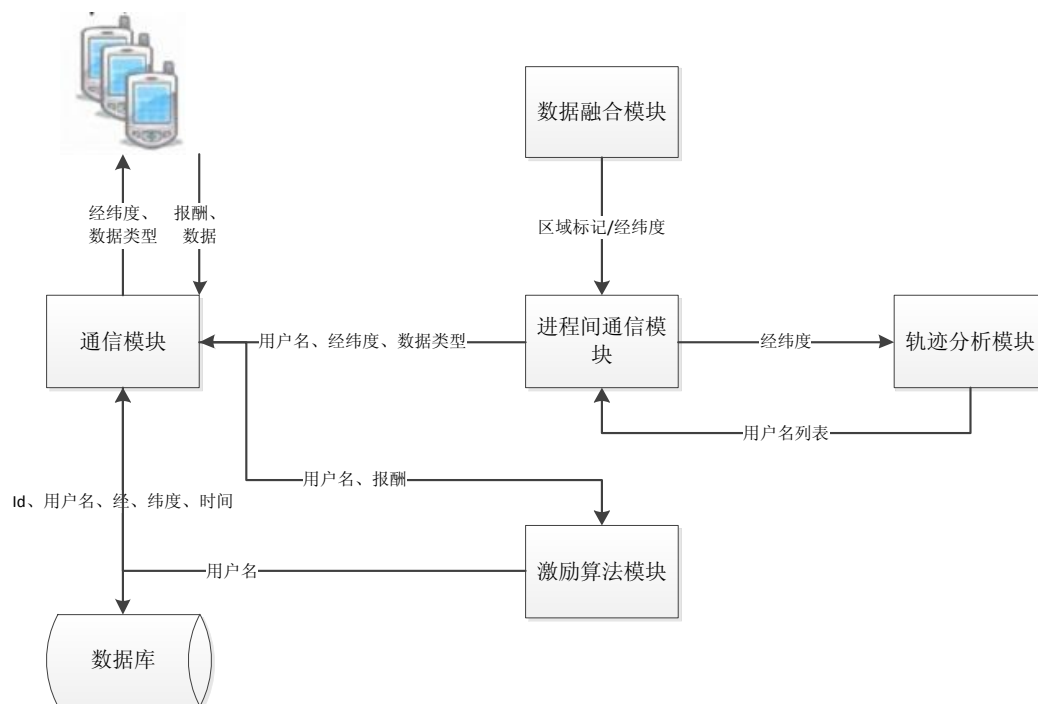


图 4-1 与其他模块调用关系

每一次参与式感知任务的发起是由数据融合模块驱动或者是激励模块定时驱动的。在数据融合模块发现某些区域有数据缺失后，会驱动参与式感知任务的启动，从而获得缺失的数据。通过远程调用的方式进行进程间通信，从而通知激励模块，传递经纬度参数和数据类型参数。激励模块会根据该消息，调用轨迹分析模块的轨迹预测功能，轨迹分析模块将计算出将会到达该位置的用户名列表。激励模块在获得该用户名列表信息后，将根据激励算法确定采用何种激励策略对用户进行激励，并通过网络通信模块与客户端进行交互，与此同时，激励模块将在数据库中记录下发激励任务的基本信息。其流程图如图 4-1 所示。

由于激励任务的推送需要进行大量计算，数据库中会存储每次任务的信息，激励模块则可以通过定时查询数据库，知晓是否还有未推送的激励任务，从而驱动上述过程在服务器端的进行。

4.1.2 客户端模块调度流程

参与式感知任务在下发到客户端后，需要用户的操作以及通信、存储等功能，它的流程是贯穿在用户 UI、数据采集、控制模块、通信模块之中的。

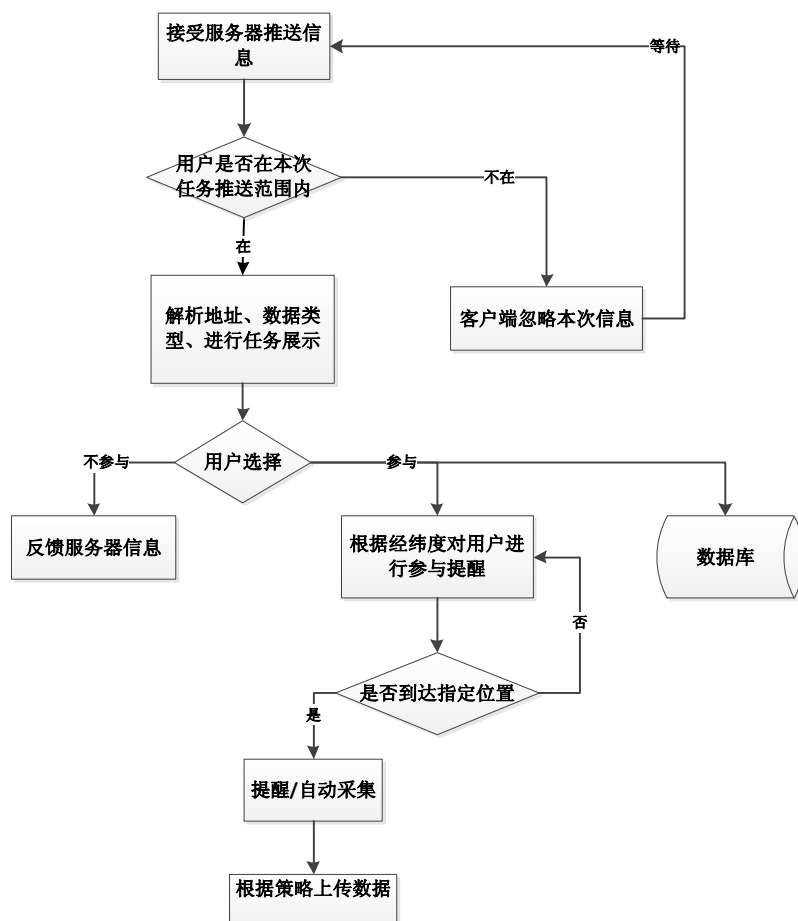


图 4-2 客户端流程

客户端的工作流程，主要是根据用户是否接受参与到本次任务当中来区别流程。参与式感知任务通过服务器端的网络通信模块下发到客户端后，客户端会向服务器查询该用户是否是本次激励任务下发用户，若不是则忽略本次激励任务，若是则会解析本次任务信息，并通过用户 UI 对任务给以展示。参与者可以选择是否接受任务，该信息将会直接返回给服务器。若用户不参与本次任务，则本次任务结束。若用户选择参与本次任务，则客户端将记录本次任务信息，包括任务时间、任务地点、所需数据类型等信息。用户采集模块将自动对比参与者位置信息与任务位置信息，在参与这到达指定位置时，采集模块将会自动采集所需要的传感器信息。如果网络状态允许则进行上传操作，若网络状态不允许，则存储数据库，等待用户主动上传。

由于参与式感知任务分为两种：服务器定价任务和用户报价任务。他们在客户端将存在不同的流程。

服务器定价：展示根据下发任务解析出来的信息，给用户是否参与的选择。如果用户选择忽略本次任务，则向服务器发送信息，记录用户拒绝参与的信息。若用户接受，服务器根据该信息更新本次激励的响应相关信息，等待用户上传数据。

用户报价：用户拒绝参与则不再显示本次任务。用户若向服务器报价，若跟服务器协商成功，则服务器等待用户上传数据，若是协商失败这在本本地删除该任务。

4.2 激励模块详细设计

4.2.1 激励模块子模块划分

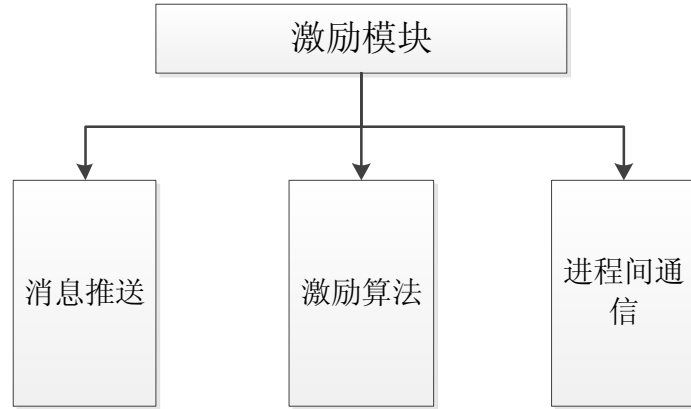


图 4-3 激励模块划分

如图 4-3 所示，激励模块根据功能性需求划分为了消息推送、激励算法和进程间通信模块三个模块。三模块之间相互调用，完成整个激励模块的功能。其中消息推送模块和进程间通信模块作为本模块的功能性模块实现。

● 消息推送

消息推送模块负责实现服务器对参与式感知任务的主动下发功能。参与式感知任务准确的推送到客户端，是激励任务顺利进行的首要条件。该模块的实现需要客户端和服务器的共同配合，考虑到客户端网络资源和电池资源的限制，该模块在实现的过程中需要保证消息推送的稳定性、较少的延迟和资源消耗。

本论文中采用长连接的方式进行消息推送，节约资源。消息推送模块需要根据在用户注册的过程中，通过随机函数生成标记位，对用户的类型进行划分。根据生成的任务类型以及数据库中记录的用户类型，决定下发任务是服务器定价还是用户报价的模式。服务器下发的数据是 JSON 格式的，下发数据的内容如表 4-1 所示。根据推送任务模式不同，Payment 字段可为空。

表 4-1 服务器下发的数据类型

| 数据名称 | 说明 |
|-------------|---------------------|
| IncentiveID | 表明激励任务的编号 |
| PSType | 0 表示服务器定价，1 表示用户报价 |
| PublishTime | 任务发布的时间 |
| Longitude | 经度 |
| Latitude | 纬度 |
| DataType | 所需的数据类型 |
| Payment | 在服务器定价中是固定值，用户报价则为空 |

由于本系统中跟随环境变化而发生变化的数据有光照、声音和图片三种信息。本论文中采用位运算的方式，对数据类型进行定义。其中光照为 1，声音为 2，图片为 4。例如采集光照和声音两种信息，则 $\text{DataType}=1+2=3$ 。具体对应关系如表 4-2 所示。

表 4-2 DataType 与数据类型对应表

| DataType 值 | 数据类型 |
|------------|----------|
| 1 | 光照 |
| 2 | 声音 |
| 3 | 光照、声音 |
| 4 | 照片 |
| 5 | 光照、照片 |
| 6 | 声音、照片 |
| 7 | 光照、声音、照片 |

● 进程间通信

在本系统进行设计之初，比较了多进程编程和多线程编程的差别。为保证系统的稳定性、数据同步的简便性和调试的简洁性。本系统采用了多进程编程的方式。由于各个进程并不是完全独立的,需要进行数据交互，因此进程间通信模块是必须的。

进程间通信模块主要负责完成进程之间的数据共享和相互调用工作。激励模块的顺利进行需要与数据融合模块、轨迹分析模块进行良好的进程间通信。激励模块需要定时从数据融合模块获取位置信息，并调用轨迹分析模块，获得用户名列表。这两种信息的获取均通过进程间通信完成。由于该操作均需要大量计算，为避免过长时间的等待，该模块需采用异步调用的方式进行实现。

● 激励算法

本模块的主要功能就是实现激励算法。激励算法的研究也是本论文的核心内容之一。因此为实现多种激励算法，在进行系统设计时要保持本模块相对独立，在算法进行更改时，对系统产生最小的影响。

激励算法主要有两方面的功能做：首先是决定参与式感知任务的参与者有哪些，其次是决定激励的强度。本论文中研究的激励算法，是希望通过激励的接入，达到参与式感知数据时空分布和数量的最大平衡。因此要选择最可能参与的用户以及对时空分布和数据量平衡影响最有的用户进行激励。

4.2.2 激励模块类设计

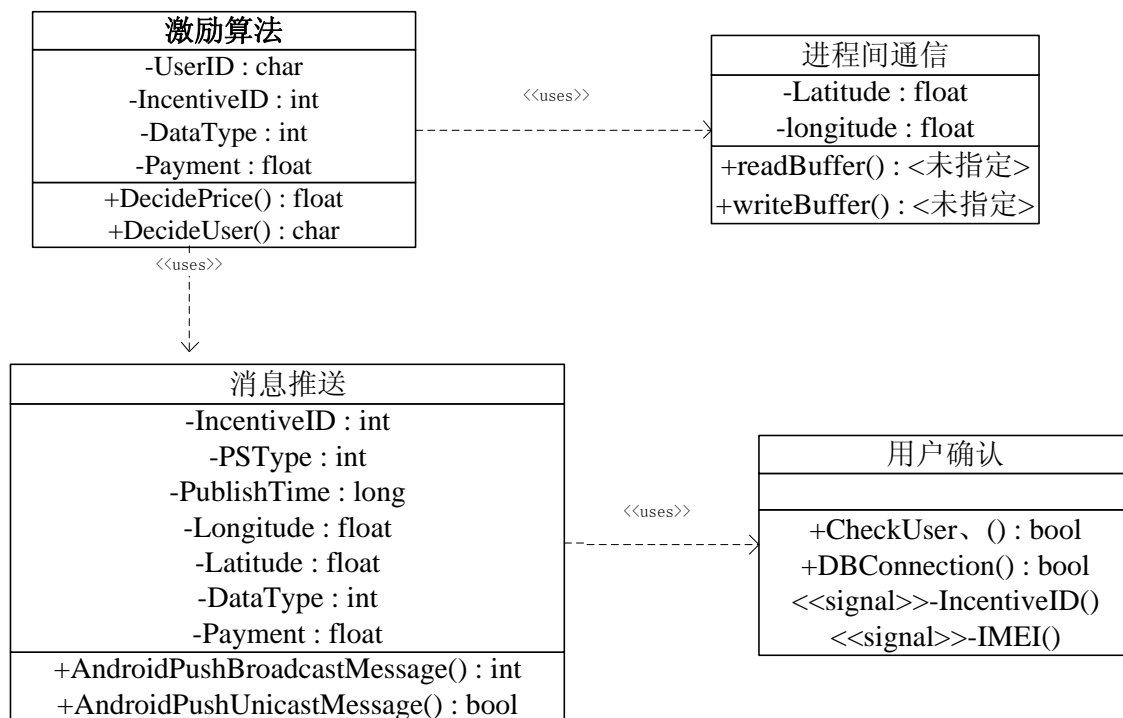


图 4-4 激励模块类图

激励模块根据功能划分实现了以上四个类。首先激励算法模块会调用消息推送模块和进程间通信模块。激励算法主要实现了确定参与任务的用户列表和激励任务的报酬两个功能；消息推送模块实现了广播和单播两个功能，在组播的实现中调用用户确认类，来查询用户是否参与到本次激励任务中；进程间通信模块，主要实现了写入激励结果和调用轨迹分析等模块获得结果的功能。

4.2.3 激励模块数据库设计

激励模块在进行用户激励的过程中，需要对用户和任务的信息进行记录和更新。本论文的设计中其主要涉及到三张表：用户信息表、激励任务记录表、用户激励信息表。

用户信息表 **UserInfo** 如表 4-3 所示，主要用于存储用户的个人信息以及相关的激励信息。本模块会通过更新 **incentive** 和 **participate** 信息来确定用户对激励任务的参与情况。

表 4-3 UserInfo

| 属性 | 类型 | 描述 | 主键 | 非空 |
|----------|--------|-------------|----|----|
| userId | string | IMEI 标示用户身份 | √ | √ |
| userName | string | 用户名 | | √ |
| passWord | string | 登录密码 | | √ |
| PSType | int | 表明用户激励类型 | | √ |

(续上表)

| | | | | |
|-------------|--------|-------|--|--|
| incentive | double | 激励值 | | |
| participate | Int | 参与活跃度 | | |

原始任务信息表 **Originalincentive** 如表 4-4 所示。它主要存储了本次激励任务所需的数据类型, 该数据所在的位置, 以及激励和本次任务的推送情况。为服务器记录激励的推送情况, 通过查询该表, 获得未进行推送任务列表, 对该任务进行计算和推送。

表 4-4 Originalincentive

| 属性 | 类型 | 描述 | 主键 | 非空 |
|-------------|--------|--------------|----|----|
| incentiveId | int | 自增 | √ | √ |
| dataType | int | 规定激励所需数据类型 | | √ |
| lon | double | 纬度 | | √ |
| lat | double | 经度 | | √ |
| time | Date | 发出本次任务的时间 | | √ |
| payment | double | 对于本次任务的激励 | | |
| flag | int | 判断本次任务是否已经推送 | | √ |

用户激励信息表 **Userincentive** 如表 4-5 所示, 主要记录对于发送激励用户对于本次激励的响应情况。通过该表可以查询用户响应激励的历史信息, 以及对每次任务的接受情况, 根据该结果能够优化已有的激励策略, 从而提高任务接受和响应的概率。

表 4-5 Userincentive

| 属性 | 类型 | 描述 | 主键 | 非空 |
|-------------|--------|--------|----|----|
| id | int | 自增 | √ | √ |
| userName | string | 用户名 | | √ |
| incentiveID | int | 激励 ID | | √ |
| time | Date | 响应激励时间 | | √ |
| pay | double | 约定激励 | | √ |
| willing | Int | 是否愿意参与 | | |
| flag | int | 是否已经响应 | | √ |

激励模块主要对激励信息相关表进行修改和更新, 从而记录用户响应激励的情况以及激励任务分下发情况。激励信息所需的地理位置信息, 将同轨迹分析和数据融合模块共享。

用户信息表中记录的信息主要用来用户查询自身的信息, 每次任务结束都会更新响应用户在用户信息表的值。其中用户权限, 是在用户注册时通过随机函数分配的权限(管理员权限除外)。任务信息表记录了每次任务的生成时间, 任务的数据类型以及任务的推送情况, 方便检测未进行推送的任务, 推动该模块的运行。

4.3 激励机制算法研究与实现

推送模块和进程间通信模块的实现对于任何激励算法的实现都是适用的，在本论文的 2.2 节中详细描述的数据上传优先和定价优先的激励机制均能良好适配。预先定价是指，在数据上传之前参与者已从服务器获得本次上传数据的价格，该价格可以通过竞价或者是服务器定价的方式获得。在参与者上传数据以后即可获得约定的激励；数据上传优先是指参与者在数据上传之前并不知晓本次上传数据的价格，在数据上传以后服务器根据数据的质量以及稀缺程度等因素决定该数据的激励值，并发放给参与者。本章节将详细介绍本论文采用的一种预算约束下的激励机制的原理和代码实现方法，在本算法中实现参与者和参与式感知平台的共赢。

4.3.1 激励算法研究

4.3.1.1 激励算法描述

本论文采用了一种轻量级的动态定价机制^[20]，并根据本论文作者的研究对[20]算法中计算数据数量和分布的方法进行了改进。该机制使得平台能够根据数据实时的可用性和时空分布进行动态定价，并且使用迭代的算法来计算最优结果。本算法中所用的所有变量解释如下表 4-6 所示。

表 4-6 算法变量表

| 变量名 | 说明 |
|---------------|---|
| S | 实验区域分配集合 $\{s=1,2,\dots,S\}$ |
| T | 时间槽划分集合 |
| \mathcal{R} | 区域划分集合 |
| O | 激励的预算报价， o_r 表示对 r 区域的预算是 o |
| \mathcal{A} | 区域划分，每个子区域大于 \mathcal{R} 的子区域，为方便计算可采取令该划分中每个区域是 \mathcal{R} 中子区域的整数倍方式 |
| O | 激励的预算报价， o_r 表示对 r 区域的预算是 o |
| \mathcal{B} | 预算约束 |
| \mathcal{M} | 参与者的集合 |
| n_r | 在 r 区域可能存在的参与者数量 |
| $d(o)$ | 参与者在激励值为 o 的情况下参与任务的概率。 |
| $p_r(o_r)$ | 区域 r 内，在激励值为 o 的情况下能够收集到数据的概率 |
| $\alpha(o)$ | 激励值为 o 的情况下，获得的样本数量 |
| $\rho_a(o)$ | 表示在 o 预算的情况下，数据落在 a 区域的概率 |

(续上表)

| | |
|------------|--------------------------|
| $\beta(o)$ | 激励值为 o 的情况下，获得的样本的分布情况 |
| θ_x | 对区域 x 进行激励以后得到的效果 |

首先，对实验区域进行划分，划分后的结果集为 $S = \{s = 1, 2, \dots, S\}$ 。然后，将实验的持续时间划分为连续的时间槽，方便将轨迹信息与时间信息相结合 $T = \{t = 1, 2, \dots, T\}$ 。考虑到收集数据分布的时空特性，还需要对区域进行空间和时间上的划分 $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_T\}$ 。对区域进行时空分布，下角标 i 表示的是区域，上角标 t 表示的是时间槽的值。

算法的计算关系如下图 4-5 所示。根据 $d(o)$ 计算出在区域 r 内收集到数据的概率值 $p_r(o_r)$ 。其中样本数量和样本分布的计算有依赖于 $p_r(o_r)$ 的计算结果，而最终效果则是由样本份数量和样本分布的结果共同决定的。

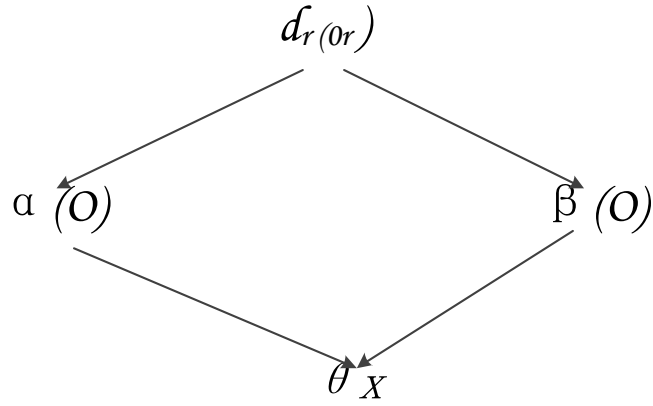


图 4-5 计算关系

根据问卷调查得到的用户在激励为 o 的情况下参与本次任务的概率如公式 4-1 所示。其中 $k = 0.042, \tau = 0.172$ 。

$$d(o) = ko^\tau \quad (4-1)$$

若为每个区域分配的激励用 o_r 表示，则该区域每个参与者获得激励可以表示为 $\frac{o_r}{n_r}$ ，那么每个人参与的概率可以表示为公式 4-2。

$$d(o_i) = ko_i^\tau, \forall i \in n_r \quad (4-2)$$

对于任意区域而言，获得数据的概率越大，所有区域获得数据的概率和越大，分布人数越多，则最终获得的数据的数量越多。 $\alpha(o)$ 的计算方法如公式 4-3 所示。对区域人员分布的期望进行计算来代表数据总量的分布情况。

$$\alpha(o) = \sum_{\forall r \in R} d_r(o_i^r) n_r \quad (4-3)$$

为根据不同区域的数据获得概率，计算数据分布的情况，本算法对区域进行了划

分。 $A = \{a = 1, 2, \dots, A\}$ ，并且为确保试验区域是同一个区域所以需要在 $\bigcup_{\forall r \in A} r = R$ 这一约束下进行。 a 所代表的区域大于 r 所代表的区域。计算 $\rho_a(o)$ 代表样本落在 a 区域的概率。

$$\rho_a(o) = \frac{\sum_{\forall r \in a} d_r(o_r^r) n_r}{\sum_{\forall r \in R} d_r(o_r^r) n_r} \quad (4-4)$$

众所周知信息熵是信息论中用于度量信息量的一个概念。消息出现的概率大，则它的信息量就小；它出现的概率小，则它的信息量就大。而信息熵表示的是信源的平均信息量。应用到本论文的算法当中，可以用信息熵公式表达数据分布的均匀性。

利用 w_a 表示 a 区域的历史数据分布情况，通过查询历史情况，得到该区域所存在的数据量，在数据越系数的区域增加数据，则对数据均匀性越有积极的影响。 w_a 就代表了区域 a 的权值。结合 $\rho_a(o)$ 和 w_a 可以计算出激励分配后，样本的分布变化情况 $\beta(o)$ 。计算公式如 4-5 所示。

$$\begin{aligned} \beta(o) &= -\sum_{a \in A} w_a \rho_a(o) \log \rho_a(o) \\ &= -\sum_{a \in A} w_a \frac{\sum_{\forall r \in a} p_r(o_r)}{\sum_{\forall r \in R} p_r(o_r)} \log \frac{\sum_{\forall r \in a} p_r(o_r)}{\sum_{\forall r \in R} p_r(o_r)} \end{aligned} \quad (4-5)$$

在本算法中用 $\alpha(o)$ 、 $\beta(o)$ 分别表示数据样本的数量和空间分布情况。而本论文中提出的算法是使得两者达到最优解。 O^* 是我们所求的最优解。

$$O^* = \{o_r \mid \forall r \in R\} = \underset{o_r}{\operatorname{argmax}} \alpha(o) + (-\lambda) \beta(o) \quad (4-6)$$

o_r 的和在总体约束下 $\sum_{\forall r \in R} o_r \leq B$

● 算法计算流程

1. 该算法是一个迭代的过程。每次激励之初，给每个区域分配的预算 o_r 均为 0。
2. 本算法的思想是将剩余的预算迭代的分配给子区域，来获得公式 4-6 中的最优解。 θ_x 表示的是在对 x 区域多分配一单位的预算后，数据分布和数量的变化情况。计算方法如公式 4-7 所示。 $\theta_{x \max} = \max\{\theta_x \mid \forall x \in R, \forall t \geq t_0\}$ ， $\theta_{x \max}$ 表示本轮计算的最大值。在本轮迭代进行之前对 x 区域已经分配出去的预算值为 $o_x \times p_x(o_x)$ ，而在本轮迭代进行以后分配的预算为 $(o_x+1) \times p_x(o_x+1)$ 。因此剩余的预算值为 $B' = B - (o_x+1) \times p_x(o_x+1) + o_x \times p_x(o_x)$ 。

$$\theta_x = (\lambda\alpha(O'_x) + (1 - \lambda)\beta(O'_x)) - (\lambda\alpha(O) + (1 - \lambda)\beta(O)) \quad (4-7)$$

3. 迭代进行步骤 2 的计算，得到 $B' \geq 0$ 情况得到的预算分配情况。使得 θ_x 得到最大值的即为所求的最优解。

4.3.1.2 激励算法分析

本论文采用的算法主要参考了[20]中描述的算法。[20]中描述的算法思想是结合每次预算分配后可能获得数据数量和数据分布的情况，来决定每一次激励的发放区域。其采用仿真的方式证明该算法主要有以下三点优势：

1. 在同等强度激励下，本论文采用的算法能够获得更多的参与者参与到感知任务中，这就为数据数量的增加提供了保障。
2. 在相同激励强度下，该算法能够获得平均误差更低的数据集。这是由于在本算法发放激励的过程中考虑数据分布均匀性对数据质量带来的提升。
3. 通过对比参与者平均收益，发现[20]下的参与者能够在较低报价的激励任务中获得较高的收益，因此[20]中的参与者更倾向于以较低的报价完成较多的任务，从而达到为任务发布者节约预算并提供更多数据且参与者获得更高收益的效果。

但[20]中描述的算法在计算 $\alpha(O)$ 和 $\rho_a(O)$ 时，使用的是参与者参与任务的概率来进行计算，本论文作者认为概率不能表明每个区域参与人数的多少，例如当区域 A 只有一个参与概率为 1 的参与者，而另一区域 B 拥有两个参与概率为 0.5 的参与者时，两个区域收集到数据的个数期望是相同的，在[20]中，则是区域 A 收集到的数据要多于区域 B。因此本论文在对 $\alpha(O)$ 和 $\rho_a(O)$ 的计算方法进行了改进。

通过对可能收集到数据数量和数据分布在某区域的概率计算方式进行改进，能够提高所收集的数据的数量，同时降低数据的平均误差率。

根据仿真实验得到的图 4-6 可知，向相同数量的参与者发布任务的情况下，改进后的算法能够获得更多的数据。因此算法的改进对数据收集的数量得到了有效的提升，表明使用数学期望的概念表述数据数量的方法能够更准确的描述区域的真实状态。

本论文实现的激励算法，由于目前系统功能的限制，主要通过实验仿真和问卷调查的方式进行了算法效果的验证，并没有在参与式感知系统平台中通过大量用户参与进行算法效果的验证。可以在后续工作中，通过对系统的不断完善，提供对大量用户参与的验证。

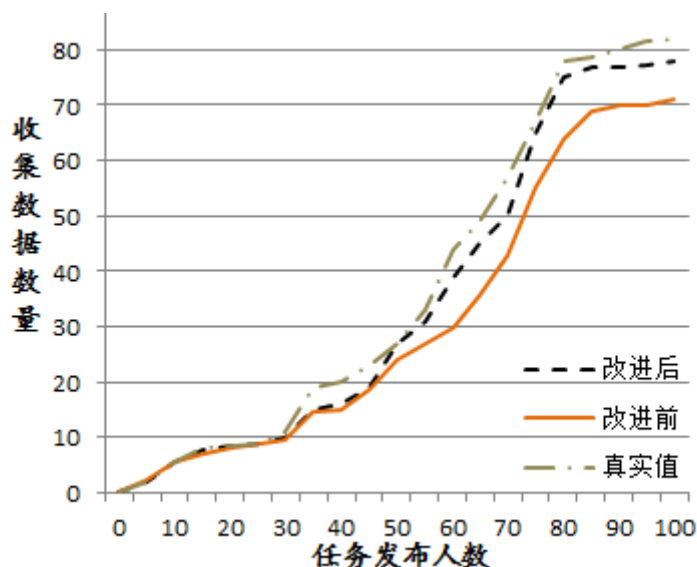


图 4-6 收集数据数量对比

4.3.2 算法实现

算法中要求有 \mathcal{R} 和 \mathcal{A} 两种区域划分，并且 \mathcal{A} 的每个区域要小于 \mathcal{R} 中的区域。在程序是现实通过 xml 配置将两个 \mathcal{R} 中的区域设置成一个 \mathcal{A} 中的区域。

实现上述算法需实现每个公式的计算，并通过迭代函数完成整个算法。算法实现的函数接口如图 4-7 所示。

```
public interface Algorithm{
    // 计算  $d(o)$ 
    public static double d ( double o );
    // 计算数据数量
    public static double  $\alpha$  ( List<Region> regionsList );
    // 计算数据落入 area 区域的概率
    public static double  $\rho$  ( Area area );
    //计算数据分布状况
    public static double  $\beta$  ( List<Area> areaList );
    //计算  $\theta$  的值并进行记录
    public static double  $\theta$ ;
}
```

图 4-7 算法实现接口

其中用到了 Region 和 Area 两个类。分别表示 \mathcal{R} 和 \mathcal{A} 两种区域划分，分别含有不同的方法和属性。Region 有区域编号、参与任务的人数、区域预算，Area 有区域编号和区域权重两个属性，其中区域权重表示的是两个 Region 区域的历史数据分布情况，通过查询数据库获得。通过读取 xml 配置文件确定区域划分的关系，方便计算。

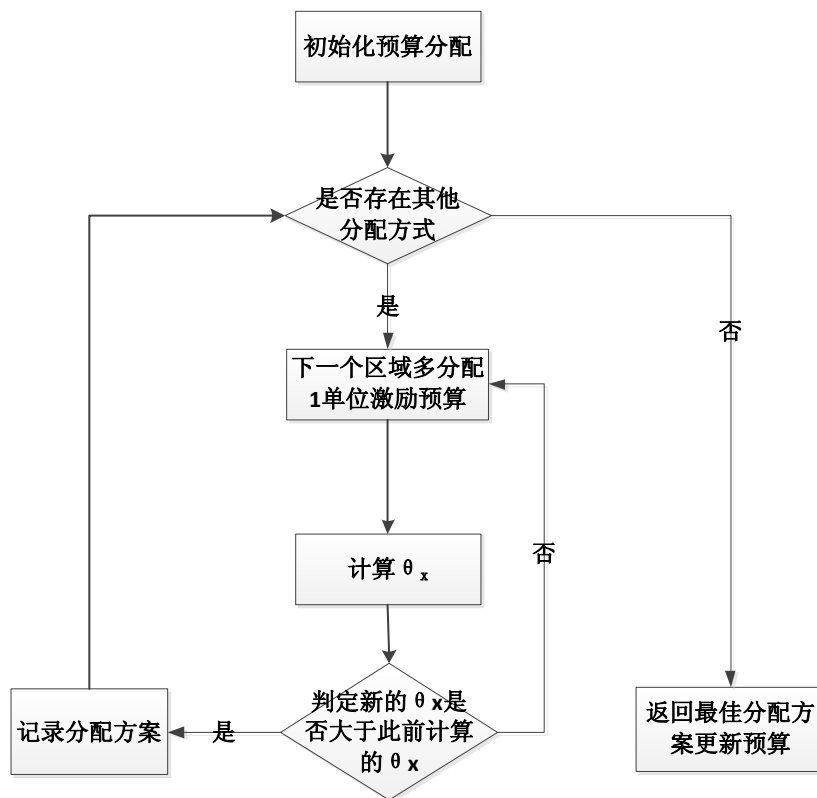


图 4-8 算法计算流程

算法的实现流程如图 4-8 所示，首先对预算和各个区域的权值进行初始化，然后通过区域进行划分，对该区域增加 1 单位预算来计算此时的 θ 值，若该值大于此前的最大值，则记录本次预算分配方案，并查看区域遍历是否完成，若未完成，则继续遍历，若已完成则返回最优的分配方案，并根据该方案发布激励任务。

对算法按照功能定义接口函数的方式，方便了算法的更换和移植。当系统中实现其他算法时，可以直接跟换算法接口函数即可完成。该算法中的原始数据和统计数据，可以通过数据库查询获得，也可以通过配置文件的读写获得。便于调试和修改的进行。

4.4 激励模块功能模块实现

4.4.1 进程间通信模块实现

本模块为完成进程间通信功能，采用了 Spring 框架下的 HTTP invoker 的方式通过远程调用实现进程间通信。

远程调用在激励模块的主要作用有：通过进程间通信模块调用轨迹分析模块的方法；激励模块可以通过进程间通信被数据融合模块调用；激励模块可以通过进程间通信调用推送模块。

远程调用实现分为服务端和客户端两部分，服务端提供服务供客户端调用。采用 HTTP invoker 的方式，服务端将服务部署在网络上，客户端可以通过 http 协议以及 Http

Invoker 代理对网络上的服务进行调用。远程调用的模型如图 4-10 所示。对于数据融合模块而言，激励模块扮演服务端角色，而对于轨迹分析模块而言，激励模块则扮演了客户端的角色。

- 服务提供方配置：

激励模块作为激励服务的提供方，需要将其服务部署到 Web 环境中，供数据融合模块进行调用和访问。

服务的配置需要导出 HTTP invoker 服务，为导出该服务，需要使用 `HttpInvokerServiceExporter` 类，因此需要先在 Spring 中配置一个 `HttpInvokerServiceExporter` Bean。通过配置 `service-ref` 的属性值置入了实现这个服务的 Bean 的引用，可以指明所实现的服务的位置。`serviceInterface` 属性用于标示这个服务实现的接口，在该接口中标识的服务即为服务器提供的支持远程调用的服务，即需要在接口中写明提供的服务的服务名、参数列表、返回值。

`HttpInvokerServiceExporter` 的本质是一个 Spring 的 MVC 控制器，它通过 `DispatcherServlet` 接收来自客户端的请求，并将这些请求转换成对实现服务的 POJO（Plain Ordinary Java Object，简单 Java 对象）的方法的调用，流程如下图 4-9 所示。

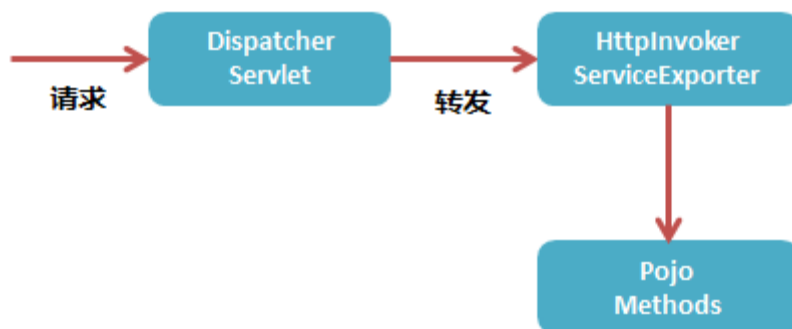


图 4-9 请求接收流程

因为 `HttpInvokerServiceExporter` 是一个 Spring 的 MVC 控制器，我们首先需要在部署描述符文件（`Web.xml`）中声明 `DispatcherServlet`，它是一个前段控制器，其任务是将请求的 URL 通过处理器映射来决策其下一步的工作内容，之后配置 `<servlet-mapping>`，表明所提供的服务对应的访问路径。最后，我们在 Spring 的上下文文件中建立一个 URL 的处理器映射，映射 HTTP URL 到对应的服务上面。通过映射配置定义了服务访问的路径。

- 客户端配置：

激励模块在服务器中，有需要调用推送模块和轨迹分析模块得到相应结果。在对其他模块发起调用请求时，在 HTTP invoker 的处理过程中则处于客户端的角色。

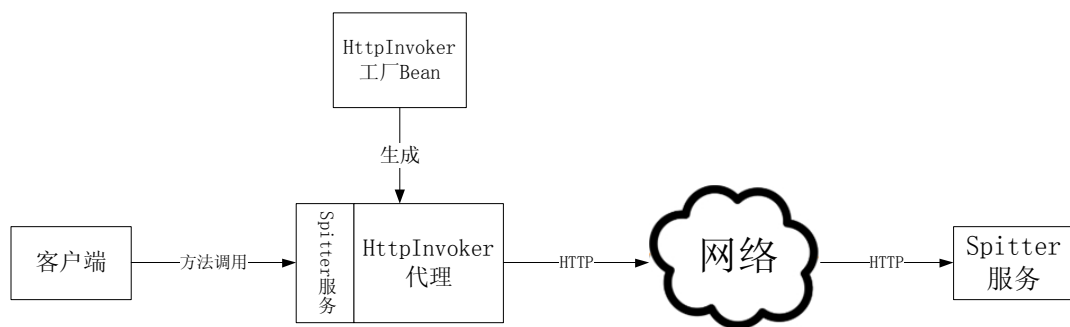


图 4-10 客户端远程调用流程

为了访问基于 HTTP Invoker 的远程服务，客户端必须在 Spring 的应用上下文中配置一个 `HttpInvokerProxyFactoryBean` 的 Bean 来代理它。`HttpInvokerProxyFactoryBean` 是一个代理工厂 Bean，用于生成一个代理，该代理使用 Spring 特有的基于 HTTP 协议进行远程通信。

其中，`serviceInterface` 属性用于标示服务所实现的接口，该接口标示了所能提供的服务。`serviceUrl` 属性用于标示远程服务的位置。完成上述两个属性的配置后，就能够像调用编写在本地函数一样访问远程服务了。

激励模块需要调用轨迹分析模块获得轨迹数据，需要调用推送模块进行消息推送。因此我们需要定义两个以上 bean，来标示两个不同的服务提供方，并且在 `serviceInterface` 所指向的接口函数中，写明分别提供的服务，即完成了本模块的功能。

4.4.2 推送模块实现

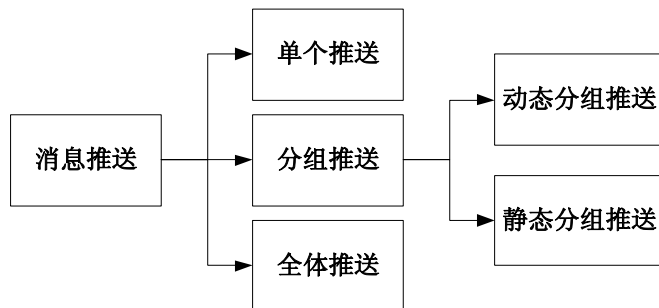


图 4-11 消息推送类型

本论文所设计的推送模块，需要能够实现单个推送、分组推送和全体推送的功能。单个推送是指给指定的用户，单独下发信息；全体推送，是给所有的用户下发信息；分组推送，是指给指定分组用户下发信息。其中分组推送又分为静态分组推送和动态分组推送。静态分组是指分组不会发生变化的情况，动态分组则是用户分组会根据系统功能要求发生动态变化。推送模块的实现难点是对不同模式推送的统一实现，尤其是对动态分组推送的实现方式。

本论文使用云推送 API^[22]来实现信息的推送。其提供两种消息的推送机制：通知和透传消息。通知消息在下发到客户端后会立即在通知栏显示通知，而透传消息可以根据

客户端的解析按照程序所需要的格式显示。由于本论文中下发到客户端的信息需要经过客户端的解析来确定消息的是否展示以及展示方式，因此我们选择透传消息作为下发消息类型。

在如今开放的 API 中，实现了单个推送和全体推送的功能，云推送平台开发的接口中实现的组播，是需要对客户端打标签的方式实现。但在本论文研究中，需要对用户动态指定分组，进行信息下发和展示，开放的 API 远远不能满足本论文研究的需求，因此本论文作者对实现方案进行了重新的设计。

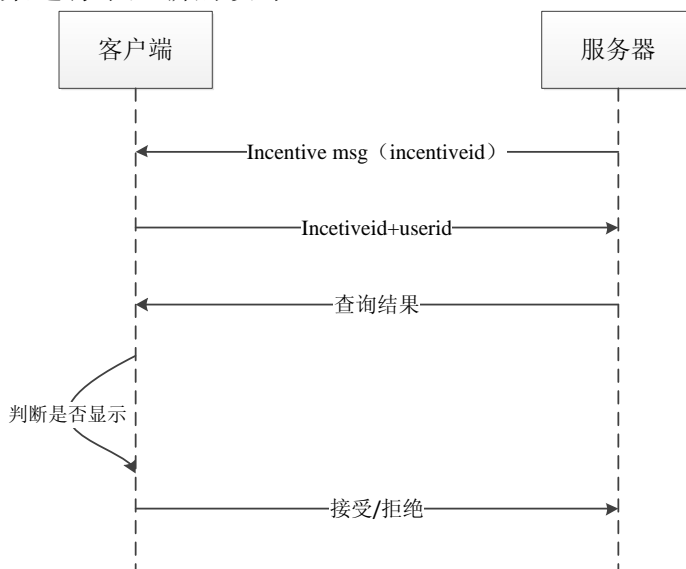


图 4-12 分组推送流程

分组推送流程如图 4-12 所示。首先服务器利用全体推送方法为所有用户下发广播，在 4.2.2 节中介绍的激励信息数据库表设计中，记录了每次激励下发的用户名。当激励任务信息下发到客户端后，客户端会携带激励任务编号和本客户端对应的用户名到服务器端查询，得到本次激励是否指定该用户参加，如果未查询到自己的用户名，则会忽略本次任务；若查询得到该用户为指定用户，则会对收到的透传信息进行解析，并且调用系统通知进行展示。从而使得激励任务进入响应阶段。

```

//建立广播类型
PushBroadcastMessageRequest request = new PushBroadcastMessageRequest();
request.setDeviceType(3); // device_type => 1: Web 2: pc 3:android4:ios 5:wp
String messageFormat = "{\"title\":\"%s\",\"description\":\"%s\"}"; //定制内容格式
String[] titleContent = new String[2];
titleContent[0] = "我的标题";
titleContent[1] = "通知内容";
request.setMessage(String.format(messageFormat, titleContent));
//调用推送接口
PushBroadcastMessageResponse response = channelClient.pushBroadcastMessage(request);
  
```

图 4-13 广播推送实现

广播推送的实现方式主要代码如图 4-13 所示，其建立广播的类型，并根据定义的格式对信息进行推送即可。在推送完成后，调用 `response.getSuccessAmount()` 回调函数，来检验推送是否成功。在实现组播时，下发推送只是跟广播推送是相同的方法，其区别在于客户端需要验证自身是否需要展示。

单播推送对应的类为 `PushUnicastMessageRequest`，与广播推送不同的是，在设置单播推送方法时需要绑定用户的 `userid`，从而确认是对哪一个用户进行推送。单播也可以采用组播的方式实现，但是会浪费其他用户的资源，因此采用云推送提供的 API 进行实现。

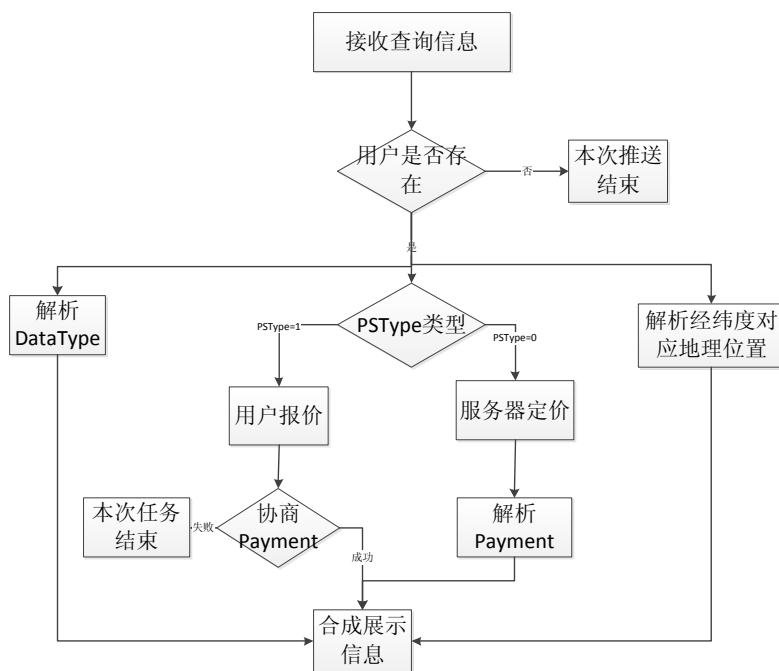


图 4-14 推送信息解析过程

推送信息在成功推送到客户端后，客户端通过查询服务器得到其是否在本次参与式感知任务中的反馈。若该客户端在本次任务指定范围内，需要对推送信息进行解析和展示。通过使用百度地图提供的 API，能够通过经纬度反向解析出该地理位置的名称。根据 4.2.1 中介绍的 `DataType` 的解析规则，解析需要采集的数据类型。如果用户属于服务器定价的激励方式，则可直接提取 `Payment` 值合成展示信息，若属于用户报价的激励方式，则需要跟服务器协商得到 `Payment`，而后合成展示信息。展示信息形成后，客户端则可进行展示。

4.5 本章小结

本章首先介绍了激励模块的内部结构和模块的划分、介绍了其在客户端和服务器的 workflows；然后，对激励模块的数据库设计进行了详细介绍；接着，着重介绍了本论文中采用的激励机制算法，以及算法的实现方式，并给出激励算法改进后对参与人数表述更精确地效果验证；在本章最后，描述了消息推送和进程间通信的实现方案。

第五章 平台系统和激励机制的测试

5.1 测试环境

参与式感知平台的测试环境拓扑搭建如图 5-1 所示。

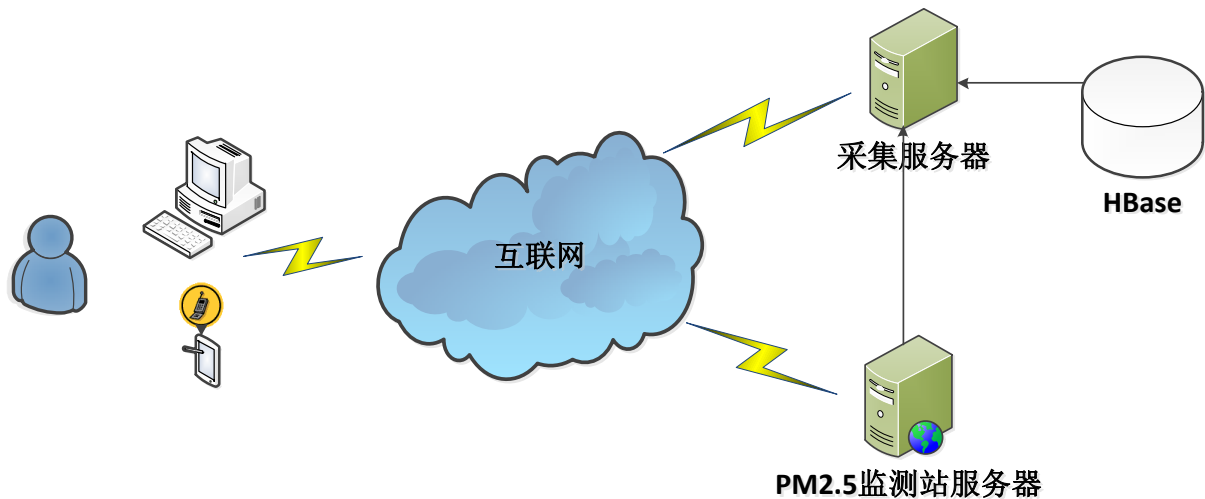


图 5-1 测试环境拓扑

由于参与式感知平台系统需要协调客户端、服务器和网站的功能。因此我们在测试时，需要通过客户端的上传、网站的展示来验证数据的存储、上传和计算的稳定运行。因此在测试环境的拓扑结构中存在移动客户端、电脑、以及通过网络连接的服务器。

其中，客户端功能测试机为三星 Galaxy 系列的产品，应用程序适配 Android4.2 以上系统版本。服务器操作系统采用的是 Ubuntu 12.04，数据库采用的是非关系型海量存储 HBase 数据库。详细的功能测试硬件环境如表 5-1 所示。

表 5-1 功能测试硬件环境

| 手机客户端 | | |
|--------------------|------|--------------------------------------|
| 测试机型号 | | Galaxy S3/S4 |
| 操作系统 | | Android 4.2 及以上 |
| 服务器 | | |
| Web 服 务 器 | 操作系统 | Ubuntu 12.04 |
| | CPU | Intel(R) Xeon(R) E5-2620 0 @ 2.00GHz |
| | 内存 | 16GB |
| | 硬盘 | 500GB |

(续上表)

| | | |
|----------------------|------|--------------------------------------|
| HBase 服 务 器 | 操作系统 | Ubuntu 12.04 |
| | CPU | Intel(R) Xeon(R) E5-2620 0 @ 2.00GHz |
| | 内存 | 16GB |
| | 硬盘 | 500GB |

5.2 测试方法

对于系统的测试分为黑盒测试和白盒测试两种，本小节对两种常见的测试方式进行介绍。

● 黑盒测试

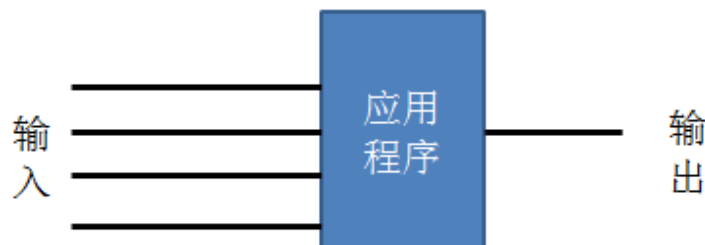


图 5-2 黑盒测试

黑盒测试又称为功能性测试。黑盒测试是由功能驱动或数据驱动的测试。在已知系统拥有功能的基础上，测试各功能是否能够正常使用。黑盒测试主要基于系统的需求分析和设计，不基于内部结构和代码逻辑。

黑盒测试主要用于发现以下几类错误：功能不正确或遗漏；界面展示错误；输入输出错误；数据库访问错误；性能错误；初始化和终止错误。黑盒测试只能找到错误的表象，并不能找到错误发生的原因，如果没有准确的需求规格说明书，测试用例将难以设计。相交白盒测试而言，黑盒测试很容易发生遗漏和冗余。

● 白盒测试

白盒测试是结构测试，基于程序代码。需要对代码的每个分支进行覆盖，按照程序的内部结构进行测试，检验是否每个分支都按照预定的要求正确工作。

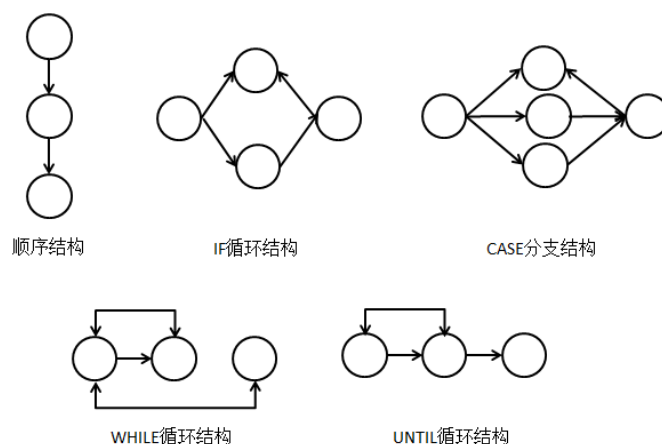


图 5-3 白盒测试逻辑覆盖

白盒测试的测试方法有代码检查法、静态结构分析法、静态质量度量法、逻辑覆盖法、基本路径测试法、域测试、符号测试、路径覆盖和程序变异。白盒测试法的覆盖标准有逻辑覆盖、循环覆盖和基本路径测试。其中逻辑覆盖包括语句覆盖、判定覆盖、条件覆盖、判定/条件覆盖、条件组合覆盖和路径覆盖。

对本系统的功能性验证和激励模块的功能性测试都属于黑盒测试的范围。

5.3 系统功能测试

为确保系统功能的完整实现和展示，需是的设计展的示场景尽可能全面的展示系统功能，由此来验证系统的良好运行。本论文作者由于负责系统的整体设计，对系统整体把握最为全面，因此负责设计整个系统的展示场景。并对收集客户端操作和 Web 端操作进行视频录制和剪辑，从而形成视频来展示系统的功能。

为尽可能多的展示系统功能，并且避免冗余场景，本阶段针对已实现系统的功能设计了四个场景。分别介绍了数据采集、存储、本地展示、图片拍摄、对已建立模块的图片分析和新建模型场景的分析、数据上传等功能。

5.3.1 客户端基本功能

客户端提供的基本功能有数据采集、数据的即时展示和统计展示、照片 PM2.5 预测模型建立和预测结果展示。

首先通过采集页面的数据表格展现验证本地数据读取和存储的正确性；然后通过采集页面右上角按钮点击操作，观察页面采集数据数值的变化验证了数据采集和即时展示的功能；最后在 PM Collection 页面点击照相机按钮，新建立模型并增加照片，验证 PM2.5 分析模型建立过程。

表 5-2 基本功能验证场景

| 步骤编号 | 操作描述 | 操作结果 |
|-------|---|----------------------------------|
| 1-1-1 | 点击采集页面右上角刷新按钮 | 观察到光照/声音标题下方数字发生变化, 表明数据实时采集的成功。 |
| 1-1-2 | 观察采集页面折线图 | 显示近一周内每天所采集数据平均值, 验证本地数据存储的正常运行。 |
| 1-1-3 | 在 PM Collection 页面, 点击相机按钮, 新建模型, 进行拍照, 点击模型。 | 提示正在建模, 模型未建立完成提示分析失败 |

根据图 5-4 所示, 在点击刷新按钮以后, 光照标题下的数据发生了变化验证了数据(光照)采集模块功能, 而图中的折线图则表明了对本地数据库数据的读取和本地数据库对采集数据存储的功能。在点击添加新的模型后, 出现场景命名对话框并在拍摄图片完成后, 出现了新添加的场景, 显示了图片数量不够导致模型建立失败的提示。表明新模型添加和建模的工作过程。系统客户端的基本功能在本场景得到验证。



图 5-4 基本功能验证场景

5.3.2 PM_{2.5} 分析模型更新

客户端在已建立成功的模型中, 增加图片会出发 PM_{2.5} 分析操作的进行。对已建立模型分析完成后, 会更新在 PM Collection 页面的背景图片和文字。长按该模型的按钮, 能够显示该模型中的图片信息。

表 5-3 已建立模型场景

| | 操作描述 | 操作结果 |
|-------|---|---|
| 1-2-1 | 在 PM Collection 页面，点击相机按钮，对已有模型拍照 | 相机模块的正常调用，并且成功拍摄照片，进入预测阶段 |
| 1-2-2 | 长按该模型，显示模型的详细信息 | 展示存储在本模型中已有的图片，验证数据库存取的正确性 |
| 1-2-3 | 在系统提示分析完成(一般在图片保存后 1-2 分钟)后，返回 PM Collection 页面 | 观察到背景图片更换为新拍摄图片，并且 PM2.5 的值更新为图片所对应场景的值，验证了 PM2.5 分析正确运行。 |

该场景验证了相机模块的良好运行，以及 PM2.5 分析模块与控制模块之间工作的协调运行。本地图片的 PM2.5 分析模型的建立和运行有赖于服务器端天气信息抓取模块抓取数据存储情况。在客户端能够得到验证结果，证明了服务器端天气信息抓取模块数据抓取和存储成功。验证结果如图 5-5 所示。

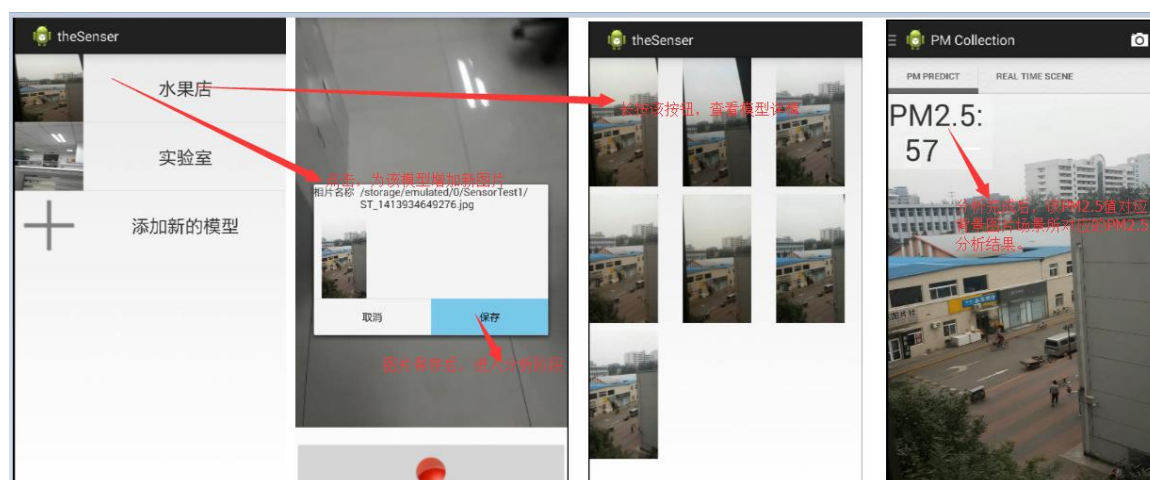


图 5-5 PM2.5 模型更新场景

5.3.3 客户端-Web 数据通信

客户端上传的数据，经过服务器的存储，也能够供 Web 展示应用。因此，最直观的验证客户端与服务器数据通信、服务器数据存取的良好运转的场景，就是通过客户端上传数据后能够在 Web 端给予展示的情况。

表 5-4 数据上传场景

| 步骤编号 | 操作描述 | 操作结果 |
|-------|--|--|
| 1-3-1 | 确保网络状况良好，在 PM Collection 页面，点击拍照按钮 | 调用相机正常 |
| 1-3-2 | 拍摄照片后点击保存，并切换到实景界面 | 在实景界面显示刚刚拍摄的照片，并显示其他用户上传的照片，验证数据下载的成功。 |
| 1-3-3 | 访问 Web 网站，并且找到刚刚拍摄图片所对应的地理位置，点击该 POI | 显示刚刚拍摄的照片，表明数据上传成功并存取成功，并在能够完成 POI 归属计算，和 Web 展示功能 |

如图 5-6 所示，图中蓝色边框中是同一张照片。在客户端拍摄完成并且上传照片以后，能够通过客户端的照片墙上看到新上传的照片，表明数据上传和下载功能良好完成。在照片上传以后，需要对照片中的信息进行计算，在 1-2 分钟后能够在 **Web** 端选择地图上刚刚上传照片对应的 **POI**，则能够在照片轮询展示时看到该照片。表明 **Web** 端对数据库信息的读取功能完成。



图 5-6 数据通信场景

5.3.4 Web 功能验证

对于 Web 页面主要功能的综合展示。观察 Web 页面各模块间数据的关系，展示了数据按照不同方式组织存储的情况。

表 5-5 Web 页面功能展示

| 步骤编号 | 操作描述 | 操作结果 |
|-------|---------------|---|
| 1-4-1 | 访问 Web 页面 | 展示由地图、图片、折线图、柱状图组成的 Web 界面 |
| 1-4-2 | 对网页上的地图进行缩放操作 | 展示 POI 的分离与聚合，折线图会根据地图上所显示的 POI 个数和标记颜色发生变化 |
| 1-4-3 | 点击 POI | 图片将展示跟所选中 POI 同样颜色标记的图片，即为在该点所采集的图片，柱状图随着图片的轮询展示而发生变化 |

Web 展示功能如图 5-7 展示。本系统设计的 Web 页面主要展示地图、各个 POI PM2.5 近期走势、照片和照片对应的 PM2.5 信息四个部分。数据按照地理位置信息进行组织，通过在地图上不同颜色的标记表明了其他几项数据所在的地理位置。验证了数据通过地理位置信息关联成功。



图 5-7 Web 功能展示

5.4 激励模块测试

为方便激励模块的单元测试，我们启动程序远程调用模块，调用模拟轨迹分析模块的接口函数。通过线程休眠模拟其他进程的负责运算，观察是否能够主动得到返回值。

为了实现推送模块的测试，进行激励模块的定时启动，采取广播和单播的模式分别模拟全体推送和单独推送的实现，验证推送模块的正常进行。

● 远程调用

远程调用模块需要验证其作为客户端调用其他模块提供的服务成功。激励模块作为客户端调用了轨迹分析模块和推送模块，来完成激励任务的下发。

我们需要验证，激励模块能够通过远程调用成功获得用户名列表，以及激励模块能够成功调用推送模块进行推送，在该测试中使用全体推送任务即可。

表 5-6 调用服务测试

| 用例编号 | 测试用例 | 预测结果 | 测试结果 |
|-------|----------------------------|--------------|--------|
| 2-1-1 | 调用模拟的轨迹分析模块，并获得返回的用户 id 列表 | 获得用户 id 列表 | 如图 5-8 |
| 2-1-2 | 调用推送模块，对已获得 id 列表进行推送 | 在客户端收到所推送的消息 | 如图 5-9 |

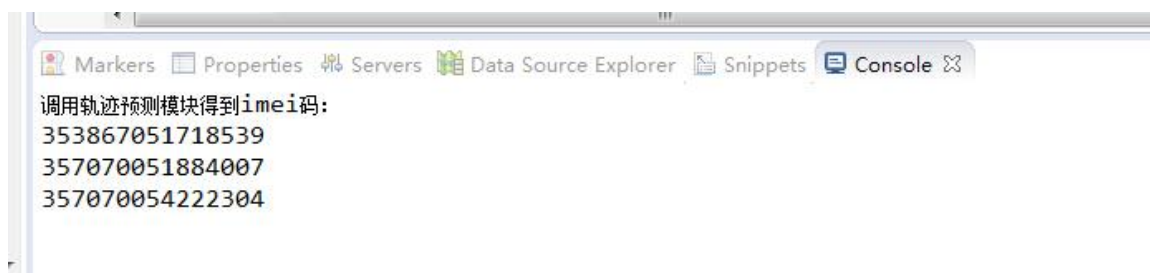


图 5-8 远程调用获取用户 id 列表

如图 5-8 所示，服务器端激励模块成功调用轨迹分析模块，获得相应的 IMEI 码，表明远程调用模块的功能能够正常进行。远程调用功能的实现，保证了激励机制服务能够顺利的被其他模块调用，同时也保证了激励模块能够成功的调用其他模块所提供的功能，完成了服务器各模块间的进程间通信。



图 5-9 推送成功

图 5-9 展示了调用推送模块向客户端推送这一功能的成功。客户端能够显示提醒，并且在通知栏显示激励任务消息。用户能够及时得到任务提醒，从而能够及时的响应参与式感知任务。

● 消息推送

消息推送功能，分为单个推送、分组推送和全体推送三种方式。通过不同的推送方式能够指定不同数量的参与者接收到参与式感知任务。消息推送功能的正常运行，是参与式感知任务能够顺利下发的首要前提，在进行功能测试时需要对其进行验证。

在给指定单一用户发送推送任务时采用单个推送的方式；在任务涉及的范围比较广发时，可以采用全体推送的方式；在任务指定了所需要的位置时，可以对参与者进行动态分组，并且采用分组推送的方式进行。

表 5-7 推送服务测试

| 用例编号 | 测试用例 | 预测结果 | 测试结果 |
|-------|-------------------|-----------------|---------|
| 2-2-1 | 对指定单个用户进行消息推送 | 单个用户获取到推送信息 | 如图 5-10 |
| 2-2-2 | 调用广播推送方法对全体用户进行推送 | 在所有客户端收到所推送的消息 | 如图 5-11 |
| 2-2-3 | 发起一次激励任务,选取一个用户 | 只有一个用户显示了激励任务信息 | 如图 5-11 |

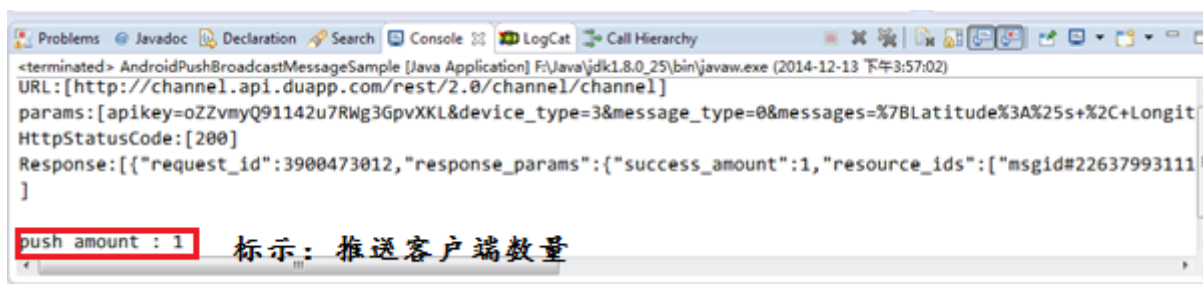


图 5-10 单播日志

如图 5-10 所示,在调用推送模块时,服务器端会在控制台打印本次推送任务推送的客户端信息以及该客户端响应本次推送的信息,以及完成推送的客户端数量。



图 5-11 推送日志

在选择单播和广播模式时,服务器会打印推送成功的客户端个数,从而能够判断单播还是广播。根据客户端的推送日志,可以看到客户端成功接收了服务器推送的激励人无消息,表明消息推送功能的完成情况。

● 客户端解析

在 4.2 节中介绍了数据推送的内容,在客户端收到数据推送内容后,需要根据规则对内容进行解析和展示。首先在收到消息后,会根据信息的内容向服务发送请求,得到

本次激励任务是否有该用户参与的确认信息。由图 5-11 可知,本次任务是服务器定价机制的任务;任务规定的采集数据类型即 $\text{DataType}=4$,则表示要采集图片任务;任务的报酬为 0.15 元。因此在客户端会根据以上信息显示任务的位置、任务内容、报酬,并询问用户是否接受。显示方式如图 5-12 所示。

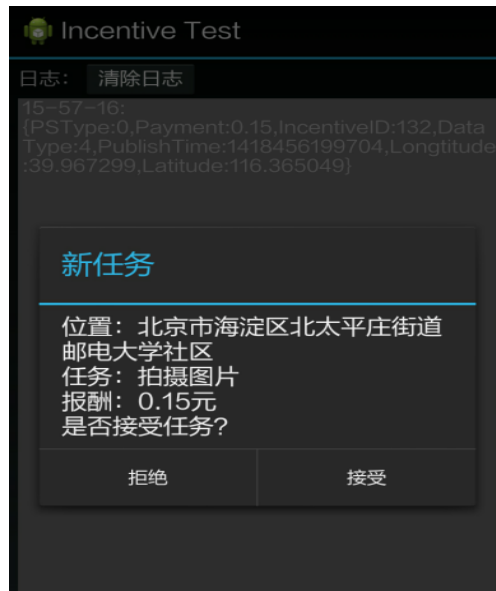


图 5-12 激励任务展示

● 激励机制测试

本论文中采用的激励策略,目的是在降低服务器投入的情况下获得尽可能多且分布均匀的数据。为了能够衡量该目标,采用该算法和普通激励的算法[24]进行对比,采用了两个方面的数据对其进行说明。

首先对比用户的平均收益,如图 5-13 所示本论文算法用户能够在激励值较低的情况下获得较高的平均收益,也就说明服务器能够通过较少的投入获得较多的数据。

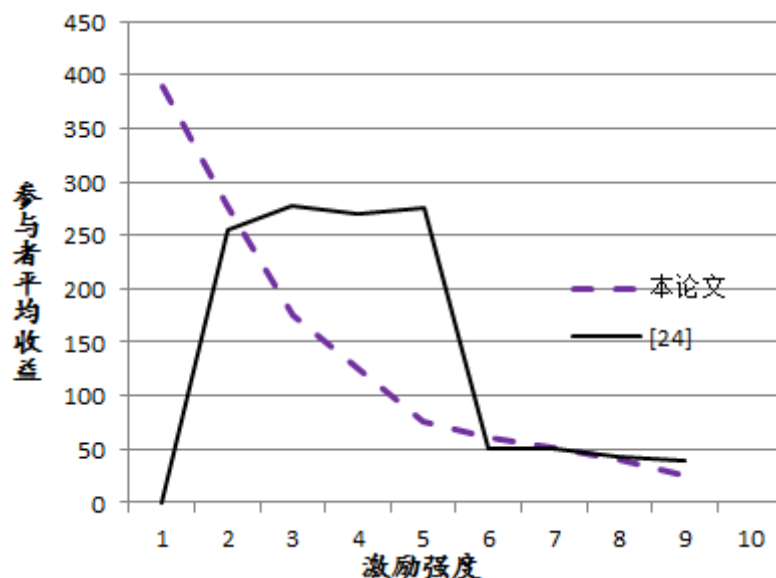


图 5-13 不同强度对应的参与者平均收益

其次，对比不同激励预算下得到的数据的平均误差。当预算相同时，数据平均误差越低，就说明数据的分布越均匀，得到的质量越高。对比结果如图 5-14 所示。

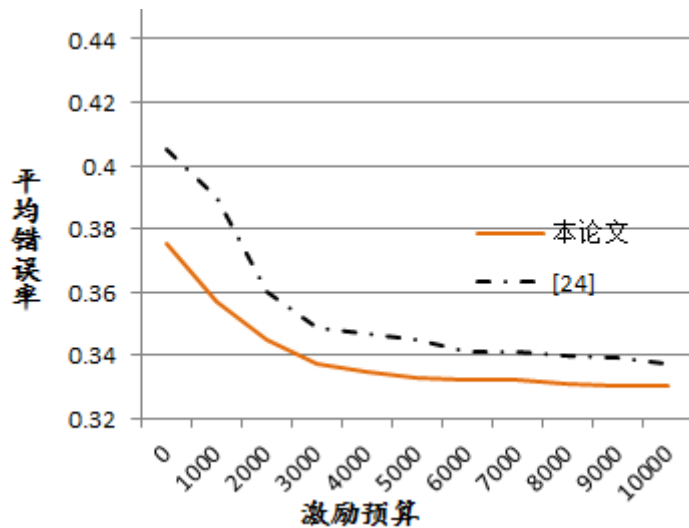


图 5-14 不同激励预算下的平均错误率

5.5 本章小结

本章详细描述参与式感知系统的测试工作，首先介绍系统的测试环境，然后针整个系统的功能性测试以及展示方案进行了设计和介绍，接着对激励模块进行了单元测试，验证论文工程工作的完成效果。

第六章 结束语

6.1 全文总结

本论文主要讨论了参与式感知系统平台的设计方案，并设计实现了平台中激励模块的算法和功能。参与式感知平台能够通过参与者收集到的传感器数据，对数据进行存储、分析、可视化展示和共享。通过合理的激励机制，吸引更多用户参与任务，获得数据量增长和分布均匀的效果。

本论文参与式感知平台的设计主要有三方面的特点，首先，在设计上对可扩展功能（数据采集、算法验证）采用插件式管理，便于后期对系统扩展工作的进行。其次，本平台采用分布式数据存储技术，为大量数据的存储提供了先决条件。最后，为实现数据能够直观展示，本平台设计了数据可视化的功能，对数据根据地理位置信息进行组织和展示。

本论文中研究和实现的激励机制，为区别于普通的反向竞价的激励机制，引入了数学期望和信息论中信息熵的概念来表述数据的分布情况。结合用户激励期望调研生成的模型，得知用户对参与式感知任务的参与概率，从而估算预算分配对各区域数据数量的影响情况。根据对数据分布均匀情况和数据数量影响情况的分析，得出最终的激励方案。

本论文首先对参与式感知平台和激励机制的相关背景和技术进行介绍；之后详细阐述了系统的需求分析，并对系统进行模块划分，讲述各模块之间的逻辑关系和消息交互的流程；接着详细阐述了激励机制的内部结构、算法和功能模块的实现方式；最后通过对整个系统功能演示的场景设计，对系统功能进行了检验，并对激励模块进行了功能测试。

6.2 可继续开展的工作

随着移动互联网技术的普及和发展，使得参与式感知拥有了更广阔的实践平台，如何更为有效的激励参与者参与并保证参与者自身权益，是激励机制研究的主要目标。本论文工作重点在于对系统功能的实现和对数据分布描述准确性算法的改进两个方面，由于论文作者自身精力和时间的限制，没有对激励算法研究中的所有约束条件都加以考虑，后续可以进一步对激励机制进行以下三方面的研究和探索：

- 在激励机制研究过程中，运用了用户的轨迹信息，并将轨迹与具体用户相对应，但并未充分考虑用户隐私保护的问题。在后续研究中，需对用户隐

私保护机制进行研究。

- 参与式感知任务的响应，虽然考虑了资源状态较差情况下数据的保存从而避免数据丢失的情况，但该方式并未彻底解决资源浪费问题。后续研究中可通过参与者间的协同合作、资源调配等方式进行优化。
- 在研究过程中，假设参与者的可信度较高，这一前提本就是不切合实际的。在后续研究中，可以通过引入用户的信誉机制，评估数据和参与者的可靠性，对激励的策略进行调整。

6.3 研究生期间主要工作

在研究生期间，论文作者参与的工作主要有：

1. 企业合作项目 公交助手 时间：2013.03-2013.06
 - 负责设置页面的布局、功能的编写
 - 负责应用中广告功能的添加、维护
 - 负责数据库的生成、更新和维护
 - 负责用户数据的采集、存储和展示
2. 青年基金项目 智能图书馆 时间：2013.09-2013.11
 - 负责应用中广告功能的添加、维护
 - 负责用户数据的采集、存储和展示
3. 国家自然科学基金项目 移动协助感知 时间：2013.12-2014.12
 - 负责 12 人项目团队的协调工作
 - 参与前期的技术调研和可行性分析
 - 负责系统的需求分析
 - 负责系统总体架构的设计和详细设计
 - 负责系统激励模块的设计与实现
 - 负责系统演示方式的设计与激励模块的测试

参考文献

- [1] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell. A survey of mobile phone sensing[J]. Communications Magazine, IEEE. 2010. 48(9):140–150
- [2] Amazon MTurk. <https://www.mturk.com/mturk/welcome>.
- [3] Shilton K, Ramanathan N, Reddy S, et al. Participatory design of sensing networks: strengths and challenges[A]//Proceedings of the Tenth Anniversary Conference on Participatory Design 2008[C]. Indiana University, 2008: 282-285.
- [4] Burke J A, Estrin D, Hansen M, et al. Participatory sensing[J]. Center for Embedded Network Sensing, 2006:1–5.
- [5] Reddy S, Estrin D, Srivastava M. Recruitment framework for participatory sensing data collections[A]//Pervasive Computing[C]. Springer Berlin Heidelberg, 2010: 138-155.
- [6] 参与式感知的挑战 <http://blog.csdn.net/moonshooter/article/details/8556539>
- [7] 王慧贤. 社交网络媒体平台用户参与激励机制研究[D]. 北京市:北京邮电大学, 2013.
- [8] Chon Y, Lane N D, Kim Y, et al. A Large-scale Study of Mobile Crowdsourcing with Smartphones for Urban Sensing Applications[J]. 2013.
- [9] 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.
- [10] Luo T, Tham C K. Fairness and social welfare in incentivizing participatory sensing[C]//Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2012 9th Annual IEEE Communications Society Conference on. IEEE, 2012: 425-433.
- [11] Faltings B, Li J, Jurca R. Incentive mechanisms for community sensing[J]. 2013.
- [12] HTTP 介绍 <http://www.w3.org/Protocols/>
- [13] android 推送介绍 <http://www.cnblogs.com/hanyonglu/archive/2012/03/04/2378971.html>
- [14] XMPP 介绍 <http://xmpp.org/>
- [15] XML 介绍 <http://www.w3.org/XML/>
- [16] Walls C, Breidenbach R. Spring In Action, Updated For Spring 2.0[M]. Dreamtech Press, 2007.
- [17] JSON 介绍 <http://www.json.org/json-zh.html>
- [18] SQLite 介绍 <http://www.sqlite.org/>
- [19] HBase 介绍 <http://HBase.apache.org/>
- [20] Song Z, Ngai E, Ma J, et al. A novel incentive negotiation mechanism for participatory sensing under budget constraints[C]. Quality of Service (IWQoS), 2014 IEEE 22nd International Symposium of. IEEE, 2014:326 - 331.1
- [21] GeoLife <http://research.microsoft.com/en-us/projects/geolife/>
- [22] 推送工具 <http://developer.baidu.com/cloud/push>
- [23] Y. Liu, Y. He, M. Li, J. Wang, K. Liu, and X. Li, “Does wireless sensor network scale? a measurement study on greenorbs,” Parallel and Distributed Systems, IEEE Transactions . 2013. 24(10): 1983–1993
- [24] J.-S. Lee and B. Hoh, “Sell your experiences: a market mechanism based incentive for participatory sensing,” in IEEE Percom’10. IEEE, 2010: 60–68.

致 谢

转眼间两年半的研究生时光即将结束。在此，我要感谢所有在研究生求学期给予过我指导和帮助的人。

我首先要感谢我学位论文的指导老师王文东教授。在硕士研究生生涯期间，王老师虽然工作繁忙，但仍坚持每周为我们项目把关，并对项目指明发展方向。在学术交流中，王老师对项目的实现方式提出独到的见解，认真的工作风格给我留下了深刻的印象，在此谨向王老师致以最真挚的感谢。

同时，我还要感谢龚向阳教授和我的导师阙喜戎副教授，他们在科研项目上给了我很多的支持和帮助，对我的项目工作和论文写作提出了宝贵的建议和意见，在这里我要向龚老师和阙老师表示由衷的感谢。

感谢宋峥师兄、张波师兄和高慧师兄，他们在学术研究方面对我的帮助和启发很大，为我提供了不同的研究思路，为学位论文的顺利进行提供了很多帮助。

感谢所有参与式感知项目组的同学。我们在项目开发和学术研究中团结一致，互相切磋互相进步，确保了项目和研究的顺利进行。

此外，我还要感谢我的家人，特别是我的父母，他们的关心和鼓励是我一生中最宝贵的财富，他们的支持与鼓励为我提供了坚强的后盾，解除一切后顾之忧。

最后，非常感谢诸位专家教授们在百忙之中评审我的学位论文，并提出宝贵意见。

作者攻读学位期间发表的学术论文

- [1] 赵露名. 基于位置的视频监控系统前台展示交互的设计与实现[EB/OL].北京: 中国科技论文在线 [2014-11-13]