**密级：商业秘密 保密期限：三年**



**硕士学位论文**



**题目： 参与式感知平台的数据处理与可视化设计与实现**

**学 号： 2013111244**

**姓 名： 杨婷婷**

**专 业： 计算机科学与技术**

**导 师： 龚向阳**

**学 院： 网络技术研究院**

**2015年 1月 2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：

# 参与式感知平台的数据处理与可视化设计与实现

## 摘 要

智能手机的普及正在改变着人们的生活。手机往往配备强大的中央处理器和多种类型的传感器，可以收集和分析大规模的感知数据。 “参与式感知”利用智能手机，使我们第一次有机会以较低的成本更广泛、更及时、更全面的采集人们普遍关心的数据，吸引更多的人群有效参与环境、社会等问题的监测和处理参与式感知平台将参与式感知融合了移动互联网、无线传感器网的技术，对人们关注的公共问题如大气质量（PM2.5、噪音等）的监测具有深远意义。平台中采集的海量数据的处理和展示是该平台的关键，数据类型繁多，数据量随着平台的应用于也越来越大，通过平台中的数据可视化技术，可以借助于图形化手段，清晰有效传达与沟通信息，对加工、传播、分享海量数据方面具有很大优势[4]。

本文在参与式感知背景和可视化相关技术的分析的基础上，分析并设计出一个完整的参与式感知平台，其中包含客户端、服务器端和web展示端，平台主要是以手机移动客户端采集的数据为基础，服务器端进行数据处理，web端把整理后的数据进行可视化展示，主要有基于百度地图的POI（POI，Point of Interest）的地理位置展示、POI点的聚合、图片的展示以及相关的图表展示。论文首先对参与式感知平台的概念和研究现状以及web端的可视化相关的技术进行简单介绍；之后对参与式感知平台的具体功能进行分析设计，在此基础上对平台、数据管理系统以及web可视化系统进行需求分析和总体设计，接着对数据管理系统以及web可视化系统的各个子模块进行详细设计和实现，之后对web可视化系统进行了功能性测试，验证可视化系统以及数据处理系统的实现符合预期目标；最后，论文对全文进行了总结，对本文的不足和下一阶段的共组进行了展望，并总结了作者在研究生期间的所有工作和成果。

关键词： 参与式感知平台 数据可视化 数据处理 兴趣点

# DATA PROCESSING PLATFORM AND VISUAL

# DESIGN AND IMPLEMENT OF PARTICIPATORY

# SENSING PLATFORM

## ABSTRACT

The popularity of intelligent mobile phones is changing people's lives. Mobile phones which equip with a powerful central processor and a variety of types of sensors can collect and analyze the scale of sensed data. Participating sense that use intelligent mobile phone can collect the data which are concerned by people more timely more comprehensive and more extensive with the minimum price, can attract more people to monitor and handle the environmental social problem effectively.  That participating sensing platform use the technology which fuse mobile Internet and wireless sensor network has far-reaching significance for people to pay more attention to public problems such as atmospheric quality (PM2.5, noise, etc.). The handling and displaying the mass data are the key points for this platform. With the application of the platform, the types and quality of data are more and more. Through the data visualization technology that uses Graphical method in the platform, this platform has a great advantage in respect of collecting processing and commuting information.

In this thesis, based on the background of Participatory sensing platform and related technologies of data visualization, a web-side data visualization system based on Participatory sensing platform is designed and implemented. Which contains the client, server and web display side, the platform is mainly based on the data collected by the mobile phone client, data processing dose the data processing. First, this thesis introduced the Participatory sensing platform background and related web-side data visualization techniques briefly; After that the architecture of mobile collaborative sensing platform is designed, on the basis of that the requirement analysis and system design of data visualization system is made; Next, detailed design and implementation of each sub-module are depicted in detail, including web front-end data visualization module and server-side data processing module; Then the whole system functional tests show visualization system to achieve the desired design goals; and finally, the summary the whole work, the inadequate and the future work of this thesis are briefly described, and the summary all of the work and achievements of author during the graduate students are given.

KEY WORDS: Participatory sensing platform; data visualization; data processing;point of Interest,

目录

[第一章 绪论 1](#_Toc439807497)

[1.1课题背景 1](#_Toc439807498)

[1.2拟解决问题 1](#_Toc439807499)

[1.3 课题主要工作 2](#_Toc439807500)

[1.4 论文结构 3](#_Toc439807501)

[第二章 相关技术与理论 4](#_Toc439807502)

[2.1 参与式感知 4](#_Toc439807503)

[2.2 数据可视化 4](#_Toc439807504)

[2.3 Apache Hbase及其相关技术 4](#_Toc439807505)

[2.4 前端浏览器与服务器之间的通信技术 4](#_Toc439807506)

[2.5 聚合算法 4](#_Toc439807507)

[2.6 百度地图API简介以及坐标体系的概述 4](#_Toc439807508)

[第三章 系统需求分析和概要设计 4](#_Toc439807509)

[3.1 移动协作感知平台的需求分析和总体设计 4](#_Toc439807510)

[3.2 数据库系统的需求分析与总体设计 4](#_Toc439807511)

[3.3 Web可视化的需求分析与总体设计 4](#_Toc439807512)

[第四章 数据处理与可视化系统的详细设计与实现 4](#_Toc439807513)

[4.1 数据库管理系统的详细设计与实现 4](#_Toc439807514)

[4.2 Web可视化系统的详细设计与实现 4](#_Toc439807515)

[4.2.2 地图展示区域 4](#_Toc439807516)

[4.2.3 图片展示 4](#_Toc439807517)

[4.2.4 柱状图展示 4](#_Toc439807518)

[4.3 数据库开发环境 4](#_Toc439807519)

[4.4 Web可视化开发环境 4](#_Toc439807520)

[4.5 本章小结 4](#_Toc439807521)

[第五章 系统测试 4](#_Toc439807522)

[5.4 测试总结 4](#_Toc439807523)

[第六章 结束语 4](#_Toc439807524)

[6.1 全文总结 4](#_Toc439807525)

[6.2 不足以及下一步工作 4](#_Toc439807526)

[6.3 研究生期间的工作 4](#_Toc439807527)

[致谢 4](#_Toc439807528)

[参考文献 5](#_Toc439807529)

[致谢 5](#_Toc439807530)

[攻读学位期间发表的学术论文和科研情况 6](#_Toc439807531)

## 绪论

### 1.1 课题背景

目前，智能手机的普及正在改变着人们的生活，国际电信联盟（ITU）发布的全球互联网情况报告显示，到2015年底，全球网民数量将达到32亿。国际电联预计到2015年年底将有70亿活跃的蜂窝连接，这个数字在2000年仅为7.38亿。而早在2013年10月18日，ITU发布的年报称全球移动互联网设备数将达到68亿台，几乎等于地球上的人口总和。总体来说，全球使用固线上网或移动上网的网民总数已经到了27亿人。ITU称，有些网民不止一部手机，总的移动渗透率大概为每百家住户/家庭有96.2名用户[1]。Facebook、谷歌等公司在全球尤其是移动平台上拓展它们的业务，ITU提供的一些数据表明每百家家庭约有41.3名用户正在使用某种形式的上网服务，其中，固线宽带用户约为9.8名，而移动宽带用户约为29.5名。换句话说，如果你想将你的服务提供给最新的网民，你最好在移动平台上推广它。

每个内置了传感器（GPS、加速度、时间、声音、图像等）的手机也都是一个总和的传感器平台。如果城市中的每一个人将自己或自己的手机所感知的数据汇集起来，可以形成强大的感知能力，这就是参与式感知。

传统的传感网络的特点是信息由传感器产生、搜集，然后自动的（可以是主动也可以使被动）上传至某个中心节点或者区域节点[12]。而“参与式感知”更加注重“人”的参与，数据主要是由用户创建、筛选和控制，然后上传。所以“参与式感知”是以人为主，用户出于个人或者经济兴趣，有意识的响应感知需求，用户既是数据的消费者又是数据的提供者[3]。而随着智能手机的普及，当前情况下用户用于共享数据的设备大部分都是智能手机，而共享的途径普遍就是移动互联网。

数据可视化主要旨在借助于图形化手段，清晰有效传达与沟通信息。海量数据带来的大量信息，对加工、传播、分享海量数据带来挑战。数据可视化已经遍布我们的生活的的每个细节，例如现在的新闻播报以及天气预报也越来越多的用到数据图表，动态演示且立体化的呈现报道内容。人们期待更加直观、高效的信息呈现方式，而数据可视化正好弥补这项需求。在智能手机、平板电脑等平台日渐普及的当下，新的交互手段将成为数据可视化的趋势[10]。

### 1.2 拟解决问题

本课题的研究目标是参考现有的参与式感知技术和理论，运用移动互联网已有的感知系统，参与者通过智能移动终端的传感器收集并上传所感知的数据到指定的服务器，在服务器端进行存储、分析和处理。而数据可视化即可对这些数据进行地理位置的分析，以图像、图表等形式直观形象地把这些感知数据展示出来，让我可以更容易了解收集数据的分布情况以及变化规律。

基于上述研究目标的分析，本课题主要解决以下几个问题：

1. 移动协作感知平台的设计

参与式移动协作感知的主要是利用人们手持的带有传感器的移动智能设备来感知环境，收集实时数据，然后通过各种网络技术传送感知数据，服务器再将这些数据存储、分析、处理、展示[11]。到目前为止，研究人员对移动协作感知的概念提出多种形式化框架，归结起来可以用四层结构描述：感知层利用传感设备实习对数据采集；网络传输层为数据提供可靠的传输服务，由感知主题将移动设备采集到的传感信息介入现有的通信设施，安全可靠地上传到服务器；数据处理层即服务器对数据进行分析、整理、计算、编程等加工过程；应用层主要是在数据信息展示的基础上，人们可以据此来了解数据信息，以进行智能的决策和调节控制。

由上述的移动协作感知平台的框架的介绍和分析，我们需要针对每一层功能进行具体的设计，以使该平台符合我们进行感知数据收集、传输、存储和管理的需求。

1. 数据可视化系统的研究与设计

数据可视化是关于数据的视觉表现形式的研究，数据可视化作为参与式感知平台的应用层的应用，需要对采集的感知数据进行web端的展示，包括对于智能移动终端上传的感知数据在百度地图地理位置的展示，对上传的图片、光照、噪声以及图片的PM2.5等数据的变化的展示。

### 1.3 课题主要工作

本课题的研究任务主要是，基于移动协作感知平台，对客户端收集的各种传感数据进行过滤、分析、处理等操作后，在web前段进行各种形式的可视化展示。具体的工作内容包括：

1. 研究移动协作感知技术，对移动协作感知平台进行总体设计

通过查看与移动协作感知技术的相关文献资料，了解移动协作感知的研究现状。参与式感知（Participatory Sensing）是目前移动物联网领域的研究热点之一。，早在2006年，Burke就提出移动协作感知的概念【7】，并描绘一个可以提高数据的质量、安全性、可信度的共享性的体系架构。

基于对移动协作感知平台的系统架构的分层，对平台的每一层进行对应的设计。首先，感知平台的客户端负责完成感知层的功能，主要是通过手机终端的传感器进行感知数据信息的收集并且将所收集的数据上传至服务器；对于感知平台中的网络传输层功能，我们主要是利用现有的移动数据网络或者WIFI网络等进行数据的传送；感知平台的服务器会负责数据处理层面的功能，主要包括对感知数据的接受、分析处理和存储；在应用层，我们会通过web界面对经过过滤、分析、整理的感知数据信息进行各种形式的可视化展示。

1. 数据可视化系统的设计与实现

数据可视化系统的目标，是把用户在参与式感知平台上利用智能终端设备收集、上传的大量的感知数据以图形图像图表的形式展示出来，用户可以通过简单地交互，从不同的维度对数据进行观察分析，了解数据间的相互关系，以及变化趋势和规律等。综上，web端可视化的具体工作内容有：

1. 地图展示

首先，web端可视化是基于地图的展示，因为参与式感知平台的感知层由智能终端设备的来完成，感知的数据都是反应所在地理位置的信息，手机收集的GPS信息、图片、噪声、光照等信息，离开 了所在的地理位置就不再有意义。地图的展示功能主要包括上传图片等感知数据信息所在的地理位置（PM站点）的展示，以及这些位置点随地图缩放的聚合展示。

1. 图片展示与相应的图形图表

图片展示是对用户收集上传的海量信息进行整理，从 不同维度如不同时间段、不同月份、不同地理位置等，展示相应的图片。

用户上传的感知数据主要包括基于地理位置的信息如GPS、光照、噪声以及图片的相关的PM值等，可以利用折线图和柱状图等形式来展示。如折线图主要用来展示地图上某一个PM站点的某一个时间段的PM值趋势以及不同PM站点在同一个时间段的PM值的趋势，并且，折线图会依据地图的缩放和移动实现相应的变化；柱状图主要是展示图片相关的PM值以及位置信息等，并且有一定的交互功能。

### 论文结构

论文的结构安排如下：

1. 绪论。本章阐述了论文的课题背景，拟解决问题，课题主要工作和论文结构。
2. 相关技术与理论。讲述参与式感知的概念和架构，可视化研究，Apache Hbase及其相关技术，前端浏览器与服务器之间的通信技术，聚合算法以及百度地图API简介及其坐标体系。
3. 系统需求分析和概要设计，主要设计平台、数据库系统以及web可视化系统的需求分析和总体设计。
4. 系统的详细设计与实现，主要是对数据库管理系统和web可视化系统的详细设计介绍以及对开发环境的介绍。
5. 系统的测试与验证，包括测试环境、目的以及方案过程的介绍，最后对各个模块的测试总了总结概括。
6. 结束语，对全文进行总结，并之处目前工作中的不足之处和下一阶段的工作目标和对未来工作的展望。

## 第二章 相关技术与理论

本章将在论文研究背景和研究内容介绍的基础上，对论文涉及的参与式感知以及可视化等概念进行介绍。而优于我们的可视化系统的展示是基于web页面的，所以，我们将对web端的数据可视化技术、浏览器与服务器之间的通信技术进行阐述分析。

### 2.1 参与式感知

#### 2.1.1 概念

参与式感知的概念是在2006年首次提出的，美国加州大学洛杉矶分校的Estrin教授对参与式感知给出的定义是，个体或群体使用功能越来越强大的移动手机和云端服务器来收集和分析语义的感知过程，参与感知强调人的作用，通过将任务部署给移动设备来组建一个可互动的传感网络以方便用户收集、分享和共享本地知识【2】

#### 2.1.2 架构

参与式感知网络按层次划分可分为三层，分别为感知层、传输层和处理层。

感知层，包括参与者以及其所用的参与式设备，这层主要是负责任务的的执行，及采集相关数据；传输层，主要负责任务所需数据的传输以及数据加密；处理层，主要负责对所传输数据的分析和处理。

服务器首先发布一定的感知任务，参与者按照相应要求执行该感知任务，并将采集的结果上传到服务器，处理层对上传的数据进行筛选、分析、处理，推断出感知任务所需要的有效信息。

传统的传感网络的特点是信息由传感器产生、搜集，然后自动的（可以是主动也可以使被动）上传至某个中心节点或者区域节点[12]。相对于传统的传感器网络，参与式感知的特点是加入了人的因素，在这个特殊的传感器网络中，每个参与者中的人的因素起到了至关重要的作用，每个人都可以看作一个传感节点，这样的节点可以携带智能移动终端，可以自由移动，并在这个移动的过程中完成感知任务，并上传数据，然后继续向其原始目的地移动，如图二所示。

#### 2.1.3 参与式感知的优点和应用

“参与式感知”更加注重“人”的参与，数据主要是由用户创建、筛选和控制，然后上传，所以“参与式感知”是以人为主，用户出于个人或者经济兴趣，有意识的响应感知需求，用户既是数据的消费者又是数据的提供者。而随着智能手机的普及，当前情况下用户用于共享数据的设备大部分都是智能手机，而共享的途径普遍就是移动互联网[3]。

参与式感知中人的分析判断能力大大减轻了系统的负担，并且，优于人的移动性更具自由自主性，极大地扩大了系统的地域覆盖范围、参与者范围以及专业知识领域，因此，参与式系统相对于传统传感器网络，可以获得实时的，覆盖范围更广的，种类更丰富的数据[13]。

每个人对周围的环境都有丰富的感知功能，每个内置了传感器（GPS、加速度、时间、声音、图像等）的手机也都是一个总和的传感器平台。如果城市中的每一个人将自己或自己的手机所感知的数据汇集起来，可以形成强大的感知能力。目前，参与式感知主要应用在三个方面：

* 环境感知的相关应用，比如空气质量监测、噪声监测等
* 公共设施服务感知的相关应用，比如停车空位引导和交通路况等
* 社会化感知（以参与者为中心的感知）的相关应用，如监测并共享参与者的生活饮食习惯、社交媒体增强等

### 2.2 数据可视化

数据可视化技术起源于20世纪中期，即计算机图形学时代和数据可视化发展的萌芽时期，人们就开始研究如何在计算机中表示图形；而到了20世纪80年代后期，科学计算的可视化才被提出并逐渐发展起来。科学计算可视化（Visualization in Scientific Computing）的诞生为数据可视化奠定了重要的基础[18]。

#### 2.2.1 可视化技术概况

数据可视化（Data Visualization）是信息图形、信息可视化、科学可视化和统计图形的结合体，实现了科学可视化与信息可视化领域的统一。数据可视化的研究内容是数据的视觉表现形式，运用计算机图形学和图像处理等技术，通过对数据的可视化设计，将数据转换为图形或图像展现出来并进行交互处理【18】，以清晰有效地传达与沟通信息的技术。当前，在研究、教学和开发领域，数据可视化是一个极为活跃而又关键的方面。

随着计算机技术的飞速发展，计算机在图像处理方面越来越强。如今，人们在表现数据方面便不再局限于通过观察关系型数据来分析数据信息。计算机强大的处理能力、可视化算法和计算机图形学算法使数据可视化技术越来越丰富和强大，利用可视化技术可将大量的复杂的多维的数据信息转换为静态或动态图像并具有一定的人机交互功能，以更好地分析、挖掘数据，寻找数据间的关系或者数据规律趋势。

可视化是数据的图形化表示形式，数据可视化技术可使大量复杂的数据信息通过图表的方式解释，以使数据包含的信息一目了然，更有效地突出核心数据，方便用户快速洞察问题所在，数据可视化技术主要有如下三个特点：

1. 与用户的交互性强。在管理和开发数据上用户可以很方便地以交互式的操作进行，用户的角色并不局限于信息传播中的接受者。
2. 数据显示的多维性。针对多维数据进行可视化分析，对每一维数据的值进行分类、排序、组合和显示，进而看到所表示对象或时间的数据的多个属性。
3. 最直观的可视性特点。数据可以用图像、二维图形、曲线、动画和三维体等来显示，并可对其模式和相互关系进行可视化分析。【3】

#### 2.2.2 web可视化技术

Web可视化研究已经开展多年，其包含功能可视化、结构可视化以及操作可视化等，而数据可视化是其中最重要的部分。基于web的可视化不同于传统可视化方案，必须充分考虑网络的传输速率、分布性、高并发性以及浏览器短的展示效率等因素。

本课题中用的可视化技术主要应用在地图的展示、图片的展示以及图像图表的展示。如今，Flash、Siliverlight等富客户端技术逐渐淡出，人们更多的使用Javascript、HTML5、CSS3等原生的Web端技术实现数据的绘制和展示。当前较为流行的和认可的Web端优秀的可视化工具的类型包含图表类、图谱类、三维图形、WebGIS类，下面依次简要的介绍这些可视化工具。

1. 图表类

大量而复杂的数据的展示形式有多种，其中人们使用最为广泛的就是图表类，图表也是在工作、学习和生活中最常见的数据表示形式。常见的图表类型有曲线图、柱形图、折线图、条形图、饼图、散点图和地图。

不同类型的图的应用情景不同。比如，曲线图主要可以用来反映随时间变化的趋势，例如1年终12个月的销售额变化趋势；柱形图主要用来反映不同事物之间的比较，当然也可以用来反映时间趋势，如同组之间的销售成绩；条形图，与柱状图的应用条件相似，主要看具体的数据展示需求；饼图最常用在部分占总体的比例，即事物的构成；散点图主要用来反映事物A和B之间的相关性，或整体的分布，例如展示员工的工资与满意度之间是否存在关系；而地图主要使用与地区之间的分类比较，例如各区域的分公司的销售额比较。

将复杂数据映射成相应的图表，需要使用一定的Web端可视化工具，这些工具中使用比较普遍的主要有eCharts、iCharts、Google Chart、HighChart、D3。

1. echarts

echars（Enterprise Chars 商业产品图标库）的出现，可以说重新定义了数据图表，是大数据时代的一个亮点。随着echarts技术的不断完善，其应用越来越广泛，现在已经是商业产品的常用图表。echarts底层是基于ZRender（一个全新的轻量级canvas类库）， 创建了坐标系、工具箱、图例、提示等基础组建，并在此基础上构建出折线图、散点图、柱状图、雷达图、饼图、地图、力导向布局图、仪表盘等，同时支持任意纬度的堆积和多图表混合展现。

1. Highcharts

Highcharts Highchats支持的图表类型有柱状图、曲线图、区域图、饼状图、散状点图和综合图表等。是一个用纯Javascript编写的一个图标库，不需要像Flash和Java那样需要插件才可以运行，它能很简单便捷的在web应用程序和web网站上添加有交互性的图表，并且免费提供给个人网站、个人学习和非商业用途使用。而且还有界面美观、运行速度快、主题多、操作简单和动态交互性好的优点，所以深受广大可视化研发人员的青睐，也是本课题所应用的技术之一。

1. D3

D3的全称是（Data-Driven Document），即被数据驱动的文档。D3是一个开源项目，作者是纽约时报的工程师，它将可视化的复杂步骤精简到了几个简单地函数，只要你输入几个简单地数据，就能将纷繁复杂的数据转换为各种绚丽的图形。

1. 图谱类

图谱类相对于其他应用较少，主要应用于生物、化学等领域，如醌类图谱分析、HPLC指纹图谱、人类基因图谱等，其主要的应用技术有Arbor.js、Sigrra.js、Processing.js等。

1. Arbor.js

Arbor.js是一个利用Web Works和jQuery创建的可视化图形库，它为图形组织和屏幕刷新处理提供了一个力导向的、高效的布局算法。

1. Sigma.js

Sigma是一个致力于在Web应用中进行图形绘制的Javascript库，它的突出特性在于定制渲染，即开发者可以使用Canvas或WebGL内建渲染器，或是什么自行编写，内建渲染器还提供了许多定制渲染过程的途径。而且sigma有面向交互，具有强大的图形模型、很好的扩展性和兼容性，所有的这些特点使Sigma曾被推荐为最流行的代码库。

1. 三维图形

3d是three-dimensional的缩写，即三维图形。而在计算机里显示3d图形就是在平面里显示三维图形。而在web可视化领域，常用的绘制三维图形的技术有PhiloGL、Three.js等。

1. PhiloGL

PhiloGL是由Sencha实验室开发的一个新的WebGL开源框架，开发者可以使用该技术轻松的开发WebGL并整合到Web应用中。

1. Three.js

Three.js是一个开源的Javascript 3D引擎，它以创建低复杂、轻量级的3D库为目标，是目前用最简单、直观的方式封装WebGL中的常用方法。

1. WebGIS

WebGIS（网络地理信息系统）指基于Internet平台，客户端应用软件采用网络协议，运用在Internet上的地理信息系统。WebGIS可采用多数据库、多主机进行分布式部署，通过Internet/Intranet实现互联，是一种浏览器/服务器（B/S）结构，服务器端向客户端提供所需要的空间信息和人文信息。

WebGIS数据可视化受制于客户端浏览器的能力，客户端和服务器的内存、处理器等不确定因素也对可视化产生了冲击。目前，比较被认可和常用的webGIS技术有ArcGISWeb、GoogleMaps、百度地图、高德地图、OpenPlayer等。

1. GoogleMaps

Google地图是由Google公司于2005年4月推出的，它是以矢量地图、卫星影像、混合三种服务模式面向全世界的用户，免费提供地图查看、地点搜索以及地图缩放的地图服务。借助Google地图的API，开发者可以在自己的web应用程序中嵌入Google地图，从而可以自主使用Google地图提供的服务。

Google地图有特点如下[36]：

1. 地图操作。Google Maps的地图操操作简单，主要有移动、自由缩放。
2. 地图预生成。滴入并不是根据用户的请求动态生成，而是预先处理成图片金字塔，切换后做四叉树编码，存放在服务器端。当地图窗口发生移动、缩放时，只需要下载新的图片来填充新的区域。另外，下载过的图片无需向服务器重新下载。
3. 分析功能。Google Maps可以实现距离测量、最近分析、路径分析功能。
4. 百度地图[46]

百度地图API为开发者免费提供一套基于百度地图服务的应用接口，包括Javas百度地图的API、Web服务API、AndroidADK、LBS云等多种开发工具服务，提供了地图标注、地图数据、地图显示、司徒查询展现、搜索、定位等网络地图服务功能。并且，适用于PC端、服务器、移动端等多种设备。它将平台和地理信息数据捆绑，把复杂的GIS底层逻辑进行封装和隐藏，以一种直观的方式提供服务，让使用者只需要按照百度的要求进行注册使用，在客户端不需要下载安装任何软件、控件或地图，所有运算均在百度服务器中运行，因此，开发者只需要具备一定的HTML和Javascript编程基础，通过应用百度地图API提供的方法、事件和封装的类，就可以将百度地图服务链接到自己的网站，并制定构建交互性强、功能丰富的应用程序。【5】

1. Openlayers

Openlayers是一个专为Web GIS客户端开发提供的Javascript类库包，用于实现标准格式发布的地图数据数据访问。OpenLayers可以在浏览器中帮助开发者实现地图浏览的基本效果如放大、缩小、平移等操作，还可以进行选取线、选取面、图层叠加、要素选择等，甚至可以对已有的OpenLayers操作和数据支持类型进行扩充以赋予更多的功能。

### 2.3 Apache Hbase及其相关技术

随着web2.0的快速发展，非关系型、分布式数据存储得到了快速发展，NoSQL的概念在2009年被提出来，在2010年风生水起，NoSQL被用的最多的是Key-value存储，还有其他的文档型的、xml数据库、列存储、图型数据库等方式存储数据模型。

NoSQL的整体架构如图所示，主要由四部分组成：接口层，主要是为上层应用提供数据读写的接口；数据逻辑模型层，主要是描述数据的逻辑表现形式；数据分布层，主要是定义数据如何让进行分布；数据持久层，定义数据存储形式。

NoSQL数据模型的这些特点使它具有易扩展、大数据量、高性能、灵活的数据模型等优势，随着Web2.0技术在网络中的广泛应用，市场上的NoSQL产品也逐渐增多，例如：Hbase，Redis，MonogoDB等，在此简单介绍Hbase的技术。

Hadoop是Apache软件基金会的一个开源分布式计算平台，其核心是并行计算模型MapReduce、分布式数据库HBase和分布式文件系统HDFS。其中HBase是Google云计算最核心技术MapReduce、Bigtable和GFS的开源实现。HBase是Apache Hadoop一个顶级项目，是一个支持非结构化和结构化数据存储的分布式数据库，能够对大型数据提供实时的读写访问。

#### 2.3.1 Hbase结构

Hbase是一个分布式的、面向列的开源数据库，主要技术来源于Fay Chang所撰写的Google论文“BIgtable：一个结构化的分布式存储系统，它与传统的关系数据库不同之处主要表现在两点，第一，它是一个适合非结构化数据存储的数据库；第二，Hbase是基于列而不是基于行的模式。

Hbase中有表（table）、行（row）、列（column）、单元格（cell）、列族（column family）等概念。一个表中有若干行，并且由唯一的行键来确定存储，一行由一列或者多列构成，若干个列又构成一个列族（column family），。Hbase中最基本的单位是列，每列有多个版本，在每一个单元格中存储了不同的值。每一列的值或者单元格的值都有时间戳，默认识由系统指定，也可以由用户显示设置。时间戳可以用来区分不同的版本的值。

Hbase是按照BigTable模型实现的，是一个分布式的、稀疏的、持久化的、多维的映射，主要是由行键、列键和时间戳进行索引。由上述的特点以及结构描述，可以得到Hbase的数据存储模式：

（Table， Rowkey， Family， Column， Timestamp）-> value

#### 2.3.2 Hbase存储API

Hbase的API提供了建表、删表、增加列族和删除列族操作，同时还提供了修改表和列族元数据的功能，如压缩和设置块的大小，而且，还提供了客户端对给定的行键值进行查找、删除、增加等操作功能。Hbase的API主要是Java代码，但是也可以用其他的编程语言来实现数据的存取。

除了上述的基本功能，Hbase还提供了客户端API的高级特性，包括过滤器、计数器和协处理器，其中，get和scan两个类都支持过滤器（filter），可以进行细粒度的筛选功能比如基于正则表达式对行键或者值进行筛选，可以大大提高处理表中数据的效率。

### 2.4 前端浏览器与服务器之间的通信技术

#### 2.4.1 HTTP请求应答模型

HTTP协议是一种无状态的协议，客户端和服务器的连接是基于某种请求应答模式。一个客户端（通常是一个web浏览器），为得到服务器上的一种资源，面向服务器发送一个请求，服务器则返回一个与所请求的资源相对应的应答，上述的资源可以是简单地HTMl文件也可以是一个产生动态应答的程序。

Http请求方式主要分为GET和Post请求

* + - * Get请求

get请求是最常见的请求方式，get方法主要是要求服务器将URL定位的资源放在资源相应报文部分，回送给客户端，客户端通过解析，得到所需要的资源。

* + - * Post请求[47]

Post请求可以允许客户端给服务器提供信息较多，Post方法将请求参数封装到Http请求数据中，以名称/值的形式出现，可以传输大量数据，这样POST方式对传送的数据大小没有限制，而且也不会显示在URL中。POST请求方式安全性比GET请求的安全性高。

#### 2.4.2 Ajax技术【48】

AJAX即“**A***synchronous***J***avascript****A****nd***X***ML*”（异步Javascript和XML），是指一种创建交互式网页应用的网页开发技术。通过后台与服务器进行少量数据交换，AJAX可以使网页实现异步更新，这样就可以在不重新加载整个网页的情况下，对网页的某部分进行更新。

AJAX是通过XmlHttpRequest对象来向服务器发异步请求，从服务器获得数据，然后用Javascript来操作DOM而更新页面。XMLHttpRequest是ajax的核心机制，是一种支持异步请求的技术，即Javascript可以及时向服务器提出请求和处理响应，但是不会阻塞用户操作，以此达到无刷新的效果。

2.4.3 servlet技术

Servlet是sun公司提供的一门用于开发动态web资源的技术，其在API中提供了一个servlet接口，主要定义两个默认实现类，分别为GenericServlet、HttpServlet，其中的HttpServlet在实现Servlet接口时，覆写了service方法，该方法可以自动判断用户请求方式。

### 2.5 聚合算法

数据挖掘即是从大量的、不完全的、有噪音的、模糊的、随机的实际应用数据中，发现并提取隐含在其中未知的、可信的、有用的模式的过程。目前，数据挖掘已广泛应用于大中型企业、商业、银行、保险等领域，成为未来3-5年内对工业有重大影响的关键技术之一[49]。

聚类是数据挖掘中的一类重要技术，是分析数据并从中发现有用信息的一种有效手段。由聚类所生成的簇是一组数据对象的集合，这些对象与同一簇中的对象彼此相似，与其他簇的对象相异。聚类算法有划分聚类算法、层次聚类算法、基于密度的聚类算法、基于网格的聚类算法、模糊聚类算法、子空间聚类算法。其中，K-means算法属于聚类方法中一种基本的划分方法，传统的K-means算法是输入聚类个数k，以及包含n个数据对象的数据库，输出满足方差最小标准的k个聚类【1】

#### 2.5.1 k-means算法的基本思想概述

K-means算法是硬聚类算法，是典型的基于原型的目标函数聚类方法的代表，它是数据点到原型的某种距离作为优化的目标函数，利用函数求极值得方法得到迭代运算的调整规则。K-means算法以欧式距离作为相似度测度，它是求对应某一初始聚类中心向量V最优分类，使得评价指标J最小。算法采用误差平方和准则函数作为聚类准则函数[50]。

#### 2.5.2 k-means算法工作原理与具体步骤

K-means算法以K为参数，把n个对象分为K个簇，以使簇内具有较高的相似度，而簇间的相似度较低。首先随机选择K个对象，每个对象初始地代表了一个簇的平均值或中心。对剩余的每个对象根据其与各个簇中心的距离，将它赋给最近的簇。然后重新计算每个簇的平均值。不断重复该过程，直到准则函数收敛[51]。

问题提出：给定一个元素集合D，其中每个元素具有n个可观察属性，将D划分为k个子集，要求每个子集内部的元素之间相异度尽可能低，而不同子集的元素相异度尽可能高。

算法步骤如下[52]：

1. 从D中随机取k个元素，作为k个簇的各自的中心
2. 分别计算剩下的元素到k个簇中心的相异度，将这些元素分别划归到相异度最低的簇
3. 依据聚类结果，重新计算k个簇各自的中心，计算方法是取簇中所有元素各自维度的算数平均值。
4. 将D中全部元素按照新的中心重新聚类。
5. 重复第4步，直到聚类结果不再变化。

将结果输出。

#### 2.5.3 k-means算法存在的问题

原始的K-means算法选取K个点作为初始聚类中心，然后进行迭代操作，其中存在以下的几个问题。

1. K值的确定。

K-means算法首先选择K个初始质心，其中K是用户指定的参数，即所期望的簇的个数，这么做的前提是已知数据集中包含簇的个数，但是在很多情况下，我们并不知道数据的分布情况，实际上聚类就是我们发现数据分布的一种手段，所以，K值的确定对K-means聚合结果至关重要。

1. 初始质心的选取

选择适当的初始质心是基本K-means算法的关键。常用的方法是随机选取初始质心，但是，这样簇的质量常常很差。

1. 距离的度量[53]

常用的距离度量方法包括：欧几里得距离和余弦相似度。两者都是评定个体间差异的大小的。欧几里得距离度量会受指标不同单位刻度的影响，所以一般需要先进性标准化，同时距离越大，个体间差异越大；空间向量余弦夹角的相似度度量不会受指标刻度的影响，余弦值落于[-1, 1]，值越大，差异越小。

### 2.6 百度地图API简介以及坐标体系的概述

百度地图API是一套有Javascript编写的将百度地图嵌入到网页应用程序接口，它能够帮助您在网站中构建功能丰富、交互性强的地图应用程序。百度地图API为开发者提供丰富的函数、空间、事件和封装的类，提供很多的专题服务，如本地搜索、路线规划、地址解析等接口工用户使用【54】。开发者只需要按照百度的要求进行注册使用，通过API，利用Javascript脚本语言就可以将百度地图服务连接到自己的网页中【6】。

在百度API中，有如下坐标系[55][56]：

1. 经纬度

通过经度（longitude）和纬度（latitude）描述地球上的某一个位置

1. 平面坐标

投影之后的坐标（用x和y描述），用于在平面上标识某个位置，百度地图API默认使用墨卡托投影（Mercator Projection），平面坐标是以最大级别18级为基准的。即在18级下，平面坐标的一个单位就代表了屏幕上的一个像素。平面坐标与地图所展示的级别没有关系，也就是说在1级和18级下，天安门位置的平面坐标都是一致的。某个位置的平面坐标可以通过BMap.MercatorProjection类来完成，该类提供经纬度与平面坐标互相转换的方法。例如天安门的经纬度大约是116.404, 39.915，经过转换即可得到平面坐标：

var projection = new BMap.MercatorProjection();

var point = projection.lngLatToPoint(new BMap.Point(116.404, 39.915));

得到结果就是12958175， 4825923.77就是平面坐标。可以理解为18级下，天安门距离坐标原点的位置差为12958175， 4825923.77， 单位为像素。

1. 像素坐标

描述不同级别下地图上某点的位置，在第18级别下，我们直接将平面坐标向下取整就得到了像素坐标，而在其他级别下可以通过如下公式进行换算：

像素坐标 = |平面坐标 × 2 zoom - 18| （2-1）

比如经过计算，在第4级天安门位置的像素坐标是：790， 294.

1. 图快坐标[56]：

地图图块编号，百度地图API在展示地图时是将征地地图切割成若干图块显示的，当地图初始化或是地图级别、中心点位置发生变化时，地图API会根据当前像素坐标计算出视野内需要的图块坐标（也叫图块编号），从而加载对应的图块用以显示地图。百度地图的图块坐标原点与平面坐标一致，从原点向右上方开始编号为0，0。

某个位置的图块坐标可以通过如下公式计算：

**图块坐标 = |像素坐标 ÷ 256| （2-2）**

256实际上是每个图块的宽度和高度，我们用像素坐标除以这个数就知道图块坐标了。还以天安门为例，在第4级下天安门所在的图块编号为：3, 1，而在第18级下，图块编号为：50617, 18851

1. 可视区域坐标

地图可视区域的坐标系（用x和y描述），地图都是显示在确定大小的矩形框中的，这个矩形框通常是开发者在初始化地图传入的某个容器元素。这个矩形框也有自己的坐标系，在百度地图API中称之为可视区域坐标系，它的原点位于矩形的左上角。通过Map类的pointToPixel和pixelToPoint方法可以相互转换经纬度坐标与可视区域坐标[57]。

1. 覆盖物坐标

覆盖物相对于容器的坐标（用x和y描述），覆盖物在实现上就是若干DOM元素，这些元素会被放在若干覆盖物容器内(具体请参考地图API开发指南)，那么覆盖物的坐标实际上就是相对于这些覆盖物容器的坐标。

## 第三章 系统需求分析和概要设计

本文主要是关于参与式感知平台的数据处理以及可视化系统的设计与实现，在之前的章节中，对参与式感知平台、数据处理技术以及可视化技术进行了详细介绍，余下的章节主要对参与式感知平台进行需求分析和总体设计，阐述数据处理以及web端可视化系统的开发过程，包括需求分析、概要设计、详细设计、系统实现以及系统测试等内容。

### 移动协作感知平台的需求分析和总体设计

为了更清晰明了地阐述数据处理以及web可视化系统，首先，我们有必要对其所基于的平台有简单地了解，下面是关于参与式感知平台的需求分析以及总体设计的介绍。

#### 参与式协作感知平台的系统拓扑结构

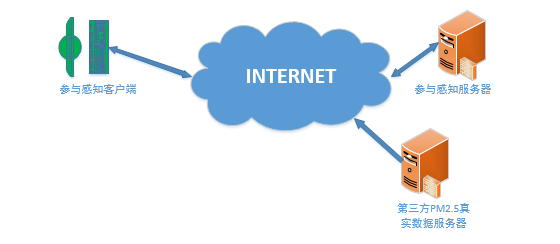
本论文所设计的参与式感知平台是移动客户端采集并上传传感数据，在服务器端对上传的数据进行整理、存储和分析，在web端进行感知数据的可视化展示，而且该平台可以预测用户轨迹，采用适当的激励机制激励用户参与进来，并且对参与的用户进行适当的奖励，以保证参与者数量以及上传的感知数据的质量和数量。根据伤处需求分析中所提到的功能性需求和非功能性需求，整个系统的网络拓扑扑设计如图3-1所示。

图3-1 系统拓扑图

参与式感知平台主要包括客户端、服务器端和web端三大部分，其中：

1. 客户端主要包括移动客户端。移动客户端，是参与式感知平台的主要数据来源，它主要是指内置各种传感器装置的智能手机客户端，人们通过这个手机客户端，可以拍照上传照片，与此同时，手机传感器感知的数据如温度、湿度、噪声等数据以及感知数据的地理位置信息，安装客户端的终端有关的状态信息如终端的网络状况、电量、充电状态等信息都可以收集并上传至服务器。主要是展示用户轨迹以及激励机制相关的内容。
2. 服务器端主要负责数据的接收、存取以及分析，并且提供web访问接口和控制台访问接口，还包含了对图片的PM2.5分析计算模块。具体的功能模块有数据通信、天气信息爬取、PM2.5分析计算模块、算法验证、数据存储等模块。
3. web端又包含了控制台和可视化展示部分。其中的控制台部分，主要负责展示参与者的轨迹，以及系统管理员发布任务、消息推送、激励奖励等功能的实现。Web可视化部分主要负责对经过服务器处理以及数据库中的存储的数据进行可视化展示，包括基于百度地图的地理位置展示、图片展示以及与图片相关的图表图像的展示等。这些可视化展示可以方便用户更直观、清晰地认识感知数据呈现的环境信息以及变化规律。

#### 平台需求分析

参与式感知平台主要由客户端、web端和服务器端三个部分组成，上面的章节简单介绍了平台的拓扑结构，下面简单分析平台三个部分的功能性需求。

1. 客户端

客户端主要是指移动智能终端包括参与者和带有各种传感器的智能手机终端。此处的参与者主要是指应用该平台的持有移动手机终端的用户，参与式感知平台，主要是以“人”为本，发挥人的最大的主观能动性，“参与”到平台中，客户端的职责在于充分感知周围有用信息，并把这些数据上传至服务器，以供后续使用。

1. 用户UI

客户端属于参与式感知层的最上层，也是与用户最直接接触的一层，所以要充分考虑用户使用体验，尽量让软件的操作变得舒适、简单、自由。用户界面的设计不仅要追求美观，还有使软件清晰、简介、熟悉便于操作、响应迅速、风格一致，这也是现代对用户UI的设计规范的统一。

1. 数据采集

数据采集是参与式感知平台中数据主要来源渠道，根据现有的移动设备所配备的传感器状况，我们能够采集的数据有：GPS、声音、光照、电池电量、充电状态、网络状态、图片信息、螺旋仪值。将这些采集来的数据在进行整理分析处理，就能够充分感知用户所处的环境，并且对于研究人们行为对环境影响以及环境变化等具有重大的现实意义。

1. 数据通信

客户端采集完数据需要上传至服务器进行存储和分析，并且服务器端的激励机制中，也需要与客户端进行通信来往，因此数据通信功能也是必不可少。这里的数据通信包括客户端主动上传和服务器端主动下发两种。 客户端主动上传照片等信息，服务器端会反馈给客户端是否上传成功。服务器端会依据轨迹分析，下发任务给相应的客户端，客户端通过对接收的消息进行解析，从而反馈响应给服务器并展示到客户端。

1. 服务器端

服务器端主要职责是对感知数据的接收、存储和分析和处理，对上传的图片分析计算其对应的PM2.5的值，提供Web访问的接口。

1. 数据通信

服务器的通信功能主要负责接收客户端上传的照片、任务反馈等信息，主动推送任务子信息、奖励信息等。数据通信功能是保证数据完整性的必要过程，也是数据存储与展示的中间环节。考虑到会有同时向服务器发送大量请求的情况，这就对数据通信功能提出了性能上的要求，需要通信功支持高并发访问的同时，能快速响应客户端并且对完成数据的妥善存储。

1. 数据存储

客户端上传的感知数据，数据结构多样且纷繁复杂，这就对数据存储带来极大的挑战。数据存储功能需要了解不同数据类型，保证数据库设计的合理性和数据访问的高效性，以提高数据共享的效率和质量。本平台涉及大量数据的存储与读写操作，因此需要对这些数据进行合理的设计存储格式和读取策略。

1. 天气信息抓取

天气信息爬取模块主要是服务于PM2.5计算模块，该模块主要负责对指定区域内的PM2.5监测站点的监测数据的爬取。该数据将作为服务器PM2.5分析模块中估计模型建立的基础，并且对后期PM2.5分析模块的对比实验以及误差对比分析提供数据基础。

1. PM2.5计算模块

PM2.5计算模块主要是对客户端上传的图片以及感知数据进行分析过滤处理，估计出图片对应的PM2.5的值，需要实现的主要功能是收集相关图像，提取雾霾特征，建立估计模型及估计图片PM2.5等。

1. 激励机制

激励机制是参与式感知平台中的一个重要功能之一，对充分调动用户积极性，激励用户参与任务、上传有效的感知数据具有重要作用。合理的激励机制能使用最少的任务预算，获取更多更高的优质数据。这就需要设计一个合理的奖励手段，来鼓励更多的用户参与进来，同时在预算约束的情况下，获取更多有效的优质数据，这个模块中的主要功能有任务发布、激励计算和数据分析。

1. 轨迹分析与预测

轨迹分析与预测模块主要是指在指定的某一个运动区域，在一段时间内通过对持有移动终端的参与者的轨迹等感知数据的收集，然后对这些感知数据进行分析挖掘，建立相应的预测模型，依据用户当前的运动情况，结合之前收集的历史轨迹信息，得出用户将来的运动趋势，并且预测参与者到达指定地点的概率，这也为上述的激励机制提供潜在的用户，使得奖励机制更加具有针对性。

1. 浏览器端

浏览器端主要分为两部分，一个是web可视化，一个是平台管理系统。用户通过手机客户端将感知数据上传至服务器之后，服务器对这些纷繁复杂的数据进行筛选、整理之后存储至数据库，web可视化就是在web浏览器端，对这些整理后的数据进行可视化再现，通过图形、图像、图表等形式，直观地把这些数据展示出来，并且以简单明了的方式展示数据之间潜在的规律。平台管理系统，主要有用户注册登录，用户轨迹展示，用户轨迹预测，任务发布等功能。

#### 平台的总体设计

上一节中简单介绍了平台的组成部分以及各个部分的功能需求，这一节中将介绍平台的总体设计，包括平台总体架构以及平台的模块设计。参与式感知平台主要由客户端、服务器端和浏览器端构成，其整体架构图如下所示。

图3-2 系统功能架构图 1

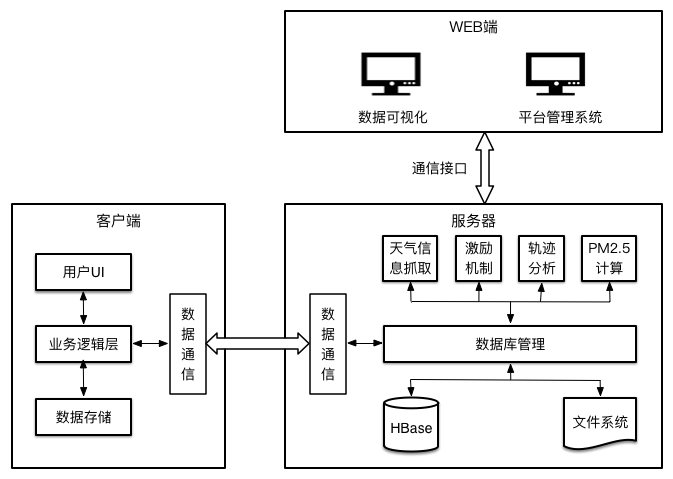


图3-2 功能框架图 1

客户端主要分为三层，用户UI、业务逻辑层和数据存储，主要实现数据的采集、数据通信、逻辑控制、数据存储和用户UI这几大功能，客户端的各个模块之间的交互通过中间业务逻辑层来完成，与服务器之间的通信通过通信模块完成。

服务器端按照功能划分，与上一节中的需求分析相对应，服务器端分为天气信息抓取模块、激励模块、轨迹分析模块、PM2.5计算模块、数据库管理模块以及数据通信模块。其中的数据通信模块与手机客户端完成通信功能，如对上传信息的响应、推送任务、推送奖励等。数据库管理模块主要负责数据库的增删查改等操作，并且提供其他服务的访问存储等服务支持。另外，服务器端还提供了web可视化接口，以提供web可视化所需要的数据信息。

浏览器端按照功能划分主要包括web可视化和平台管理系统。其中，web可视化也是本文中重点研究和设计功能，主要是把由服务器端处理过的大量纷繁复杂的数据以一种简单直观形象的方式展示到浏览器端，展示的方式主要有图形、图像、图表等可视化方法，以更好更清晰地展示数据之间的关系以及数据变化趋势和隐含的规律等。

#### 系统的分层设计

图3-3 整体框架图

依据平台的整体框架，按照网络层次的角度，可以将该平台的模块划分为：表示层、应用层、控制层、存储层和物理层。如上图所示，左端是客户端，主要是分为物理层、存储层、控制层、应用层和表示层，其中物理层是最接近硬件的部分，主要涉及到移动手机客户端的数据采集和与服务器端的数据通信；存储层主要是指客户端用嵌入式的和轻量级的sql数据库SQLite实现客户端数据的简单存储；控制层主要是对客户端的各个功能的逻辑控制和数据传输控制，是整个客户端框架的核心，应用层包含了客户端的Ar算法模块，主要是对系统中的数据信息进行应用、分析和处理；表示层主要指客户端的UI对客户端数据的展示。右端是服务器以及web展示模块，服务器端与客户端的层次大致相同，物理层主要是与客户端相连接的数据通信部分；存储层主要负责对复杂数据的存取，主要采用了分布式数据库Hbase和文件系统对数据进行存储管理；控制层负责对其他模块见的逻辑关系以及数据传输的控制，各个模块间的逻辑关系通过远程调用来实现；应用层主要是指轨迹分析预测、激励机制、天气预报抓取等模块对上一层提供的接口的应用；服务器端的表示层也就是web展示层和平台管理系统，是对服务器端经过分析处理整理后的数据的可视化展示。

### 数据库系统的需求分析与总体设计

数据库系统主要是负责实时存储客户端上传至服务器端的感知数据信息，然后依据上传数据的类型，对数据进行整理分析与处理，主要是合并冗余信息、筛选无效说、建立二级索引等，最后依据其他模块如用户轨迹分析预测、激励机制、web端展示、图像分析模块、PM2.5计算模块等的具体需求和应用场景来设计不同的存储表结构，为以上的应用模块提供高效可靠的数据读写接口。

#### 数据库系统的需求分析

数据库管理系统是对参与式感知平台的数据处理核心，是该系统运行的不可缺少的环节。参与式感知平台采用分布式存储系统进行数据的管理，其中的网络接口模块为其他应用模块如图像分析模块、激励模块、轨迹分析预测模块、web可视化等设计存储表结构，并提供可靠有效高速率的读取接口，主要负责接收客户端上传的数据，并将数据写入数据库中，负责接收和处理来自客户端或者web展示系统发送的HTTP请求，并且可以主动发布任务和激励给相应的参与者。不同的模块实现的功能不同，需求也不同，以下是该系统依据对各个模块的具体需求分析：

##### 功能性需求分析

1. 用户管理

参与式感知平台以人为主体，每个参与者过注册获取唯一的Id，可以上传数据，并且可以查看自己上传的数据信息。并且，用户信息也是轨迹分析模块不可缺少的，轨迹分析就是以用户为中心，根据用户的轨迹点序列进行用户轨迹的分析与预测。激励机制也需要针对用户来推送激励，进行奖励。用户管理的功能主要包括用户注册和用户登录，扩展功能主要有用户的权限分配以及受益等。

1. 实时上传数据

实时上传的数据主要是指客户端传感器感知的感应器数据如GPS信息、光照、湿度、温度、声音以及手机姿态等和多媒体数据如图像、录音以及视频等，所有的感应数据都有对应的收集时间和用户id。感应器数据通过JSON格式进行编码，多媒体数据通过EXIF技术进行添加属性。

1. 数据展示

参与式感知平台的目的就是把感知的纷繁复杂数据以易于理解的方式直观形象地展示给用户，使用户以此挖掘出数据的变化趋势和隐含的规律，更好的指导人们的行为，所以，数据展示是平台中直接面向用户的不可或缺的重要环节。收集数据展示主要分为web展示和客户端的展示。

1. 气象数据抓取

气象数据抓取模块主要负责全国主要城市天气数据以及PM2.5浓度数据的抓取和存储。这些数据是图像分析模块以及PM2.5计算模块的数据基础。抓取过程主要是利用第三方Webmagic【58】开源框架在特定气象数据与PM数据公开网站进行抓取。主要抓取每个城市的所有监测点的每个整点时刻观测到的PM2.5浓度、PM10（Inhalable Particles, 可吸入颗粒物）浓度以及AQI（Air Quality Index， 空气质量指数）以及城市在每个整点时刻的温度、湿度、风速、降雨量、压强等。该系统需要将这些数据快速存储到数据库，并可以进行快速检索访问。

1. 图像模型存储

PM2.5计算模块包括图像上传、归一化处理、分析、建模、预测几个阶段。图像分析模块分析整理数据并建模结束后，需要依据不同的手机型号存储并且会随着上传数据的增多而更新优化模型，至此分析阶段结束。预测阶段指当新的图像到来，会对图像进行简要分析，获取图像对应的拍照时间、手机姿态、GPS等信息，结合从数据库读出的新图像对应的手机型号在拍照时刻之前的最新模型，预测出该图像对应的PM2.5的值。在此要求系统存储模型的所有版本，并且提供对版本时间的快速检索能力。

1. 轨迹预测模型存储

轨迹分析与预测模块需要存储所有参与者的历史轨迹，通过分析参与者的历史运动轨迹，依据相应的算法建立预测模型，以此来预测参与者到达指定地点的概率，并且可以预测出某一个地点的可能到达者。轨迹分析与预测模块是激励算法的重要决策属性，激励机制的实现以及实现效果很大程度依赖于轨迹分析与预测模块，该模块预测的结果可以更精准的推送激励任务，可以丰富数据量和提高数据质量，也鼓励更多的参与者贡献数据和激励更多的用户参与进来。随着用户量的增加，用户每日的轨迹的增加，轨迹分析模块存储的数据也十分庞大，这些都对该数据库管理系的用户轨迹信息和预测结果的检索速度和信息完整性提出更严格的要求。

1. 发布、跟踪激励任务

激励任务是指让参与者到指定地理位置采集指定的数据，可以选择稍后上传或者立即上传数据。激励模块会发布合适的激励任务，同时调用轨迹分析预测模块获取指定时间内最有可能到达指定位置的参与者列表，以及相应的用户信息，通过激励算法得到合适的参与者列表，调用网络接口模块将激励任务推送给相应的参与者。激励模块还会跟踪激励任务的接收情况和完成情况，更新参与者的诚信度、收益等数据。该模块要求数据库系统提供对用户信息、激励任务信息的快速读写接口。

##### 3.2.1.2 非功能性需求分析

非功能性需求主要是包括对数据库可靠性、安全性、可维护性、可扩展性、可测试性等，下面就简单介绍该系统的非功能性需求。

1. 性能需求

参与式感知平台主要是采集和展示感知数据，数据类型多种多样，数据量也比较庞大，所以对数据库要求比较高。首先，数据库管理系统必须能承受住可能的大量并发的数据上传和存储，确保数据库的实时数据的高效写性能，防止因为网络等原因的数据丢失；其次为了更好地支持web可视化展示和交互功能以及客户端UI的数据展示，要求系统具备高效的读性能；此外，还有激励模块、轨迹分析模块、PM2.5计算模块等都要求可接受的数据库的数据读写性能。性能测试主要指标有响应时间、吞吐量、并发数等，在数据库存储表结构的设计过程中需要充分考虑不同的数据的类型和使用方式，以达到功能和性能的统一。

1. 可靠性与可用性

参与式感知平台中数据是核心，平台的存在的意义就是数据信息，这些长时间积淀下来的数据一旦丢失，不能再重来，所以一定要保证数据库系统的高可靠性。可靠性是指数据库在一个给定的时间间隔之内不产生错误的概率。它主要强调了数据库的正确进行，即符合某种格式化的要求。可用性是指当需要访问数据库时数据库能正常运行的概率。

可靠性和可用性主要涉及数据持久性、数据可访问性和数据一致性三方面的要求。其中，数据持久性是指在任何情况下都不会出现数据丢失，这就需要把数据写入可靠的持久性存储设备上，并且可以备份到不同的物理设备，以防止某个存储故障或者灾害发生时造成数据丢失；数据可访问性是指数据库在任何时间间隔都是可以访问的，计入因为一个存储设备的损坏，需要将数据访问切换到另一个数据存储设备上时，如果这个过程不能很快完成或者需要停止终端用户访问数据，这些都是不可访问的；数据一致性，数据为保证持久性需要有多个副本，而不可预知的网络、服务器或者软件故障，往往会导致部分副本写入失败，这就造成了副本之间的数据不一致的情况，造成数据内容冲突。

1. 伸缩性

伸缩性是指在不改变系统的软硬件设计的前提下，增加或者减少服务器的数量就可以扩大或者缩小服务器的服务处理能力，这样通过廉价的服务器集群实现分布式来线性提升性能的同时还降低了资源成本和系统维护成本。参与式感知平台随着时间的推移，更多的用户也会参与进来，数据也会越来越多，数据类型会越来越丰富，所以实现可伸缩性也是数据库管理的必然趋势。

#### 数据库管理系统的概要设计

##### 系统模块划分

##### 3.2.2.2 模块概要设计

图3-4 模块划分图

1. 针对轨迹分析预测模块的数据结构接口设计
   1. 数据结构设计

轨迹分析模块从感应数据存储表中读取用户的运动轨迹信息，然后将运动轨迹信息映射到网格中，得到用户的运动轨迹的网格序列，并将此数据存入到数据库中。然后以这些用户的历史轨迹数据为基础，依据轨迹预测算法，输出对用户的轨迹预测表。

格子预测表结构设计 如下所示：

|  |
| --- |
| private static final byte[] INFO\_CF = "info".getBytes();  private static final byte[] GRID\_COL = "gridid".getBytes();  private static final byte[] PREDICT\_COL = "predict".getBytes();  private static String[] family = {"info"};  private static final String tableName = "grid"; |

轨迹表结构如下：

|  |
| --- |
| //表名  public static final String tableName = "traj";  //列族info  public static final byte[] INFO\_CF = "info".getBytes();  public static final String[] family = {"info"};  //path列  public static final byte[] PATH\_COL = "path".getBytes(); |

* 1. 读写接口设计：

表3-1 轨迹接口表

|  |  |
| --- | --- |
| **public** **boolean** setPath(String userId, String trajId, **double** lat, **double** lon, String gridId,Date timestamp) | |
| 参数（按照上述顺序） | 用户Id，轨迹Id，经度，纬度，格子Id，时间戳 |
| 返回 | 0代表成功，否则失败 |
| 描述 | 向轨迹表中添加某用户的path轨迹信息 |
| **public** List<String> getTrajIds(String userId, Date timestamp) | |
| 参数（按照上述顺序） | 用户Id，时间戳 |
| 返回 | 0代表成功，否则失败 |
| 描述 | 向轨迹表中添加某用户的path轨迹信息 |
| **public** String getNeartraj(String userId) | |
| 参数（按照上述顺序） | 用户Id |
| 返回 | String 轨迹Id |
| 描述 | 返回用户最近的轨迹id |
| **public** List<String> getAllUserNames4GPS(**double** lon, **double** lat) | |
| 参数（按照上述顺序） | 经度，纬度 |
| 返回 | 用户列表 |
| 描述 | 根据经纬度，来获取预测的用户列表 |
| **public** List<String> getPath(String userId, String trajId, Date timestamp) | |
| 参数（按照上述顺序） | 用户Id，轨迹Id，时间戳 |
| 返回 | **return** List<String> Arraylist={grid1/timestamp1,......} |
| 描述 | 向轨迹表中添加某用户的path轨迹信息 |
| **public** List<String> getTrajIds(String userId, Date timestamp) | |
| 参数（按照上述顺序） | 用户Id，时间戳 |
| 返回 | **return** List<byte[]> 对应于用户ID的轨迹序列 {grid1，。。。。。} |
| 描述 | 获取用户的轨迹序列 |
| **public** String findGridId(**double** lat, **double** lon) | |
| 参数（按照上述顺序） | 经度，纬度 |
| 返回 | **String格子Id** |
| 描述 | 依据经纬度得到其对应的格子id |
| **public** **boolean** setPredictUsers(String gridId, Date timestamp,List<String> users) | |
| 参数（按照上述顺序） | 格子id，时间戳，用户列表 |
| 返回 | **return** {@link List}list<String> {usrID1,userID2.... |
| 描述 | 在某一个时刻，一个格子，可能要去那个格子的用户列表 |

1. 针对激励机制模块的数据结构接口设计
   1. 数据结构

|  |
| --- |
| public class UserTable extends AbstractTable{  // 表名  private static final String tableName = "user";  // 列族名  private final static byte[] COLFAM\_NAME = "info".getBytes();  // 密码  private final static byte[] PWD = "pwd".getBytes();  // 权限  private final static byte[] PRIO = "prio".getBytes();  // 总共获得的激励  private final static byte[] EARN = "earn".getBytes();  // 参与次数  private final static byte[] TIMES = "times".getBytes();  // 推送token  private final static byte[] TOKEN = "token".getBytes();  private final static byte[] TRANSACTION\_CF = "transaction".getBytes(); |

任务表结构设计如下：

|  |
| --- |
| private static final String tableName = "task";  private static final byte[] INFO\_CF = "info".getBytes();  private static final byte[] PROPERTIES\_COL = "proper".getBytes();    private static final byte[] STATE\_CF = "state".getBytes();  private static final byte[] TASKPHASE\_COL = "taskphase".getBytes();  private static final byte[] PARTICIPANTS\_COL = "participants".getBytes();    private static final byte[] BID\_CF = "bid".getBytes(); |

* 1. 接口设计：

表3-2 推送接口表

|  |  |
| --- | --- |
| public boolean createTask(Task task, Date timestamp) | |
| 参数（按照上述顺序） | 任务对象，时间戳 |
| 返回 | 0代表成功，否则失败 |
| 描述 | 创建一个任务信息 |
| public boolean modify(String taskId, int taskphase, List<String> participants,Date timestamp) | |
| 参数（按照上述顺序） | 任务Id，任务类型，参与者列表，时间戳 |
| 返回 | 0代表成功，否则失败 |
| 描述 | 修改任务类型或者参与者列表 |
| public List<byte[]> getTasksByTaskphase(int taskPhase) | |
| 参数（按照上述顺序） | 任务类型 |
| 返回 | 任务行键列表 |
| 描述 | 根据taskPhase来返回task行健列表 |
| public Task getTaskInfo(byte[] rowKey) | |
| 参数（按照上述顺序） | 任务行键 |
| 返回 | 任务对象 |
| 描述 | 返回具体的任务信息 |
| public boolean insertBid(String taskId, String userId, double bidPrice, Date timestamp) | |
| 参数（按照上述顺序） | 任务Id，用户Id，竞价，时间戳 |
| 返回 | **0代表成功，1代表失败** |
| 描述 | 输入一个用户竞价 |
| public List<JSONObject> getBids(String taskId, int n) | |
| 参数（按照上述顺序） | 任务Id，数字 |
| 返回 | n个投标者 |
| 描述 | 找到前n个最低投标者 |
| public List<JSONObject> getFaildBids(String taskId, int n) | |
| 参数（按照上述顺序） | 任务Id，数字n |
| 返回 | **用户报价** |
| 描述 | 未选择用户的所有报价 |
| public Boolean set(String username,String pwd,int priv,String token) | |
| 参数（按照上述顺序） | 用户名，密码，优先级，token |
| 返回 | **0代表成功，1代表失败** |
| 描述 | 存储用户信息 |
| public String getPwd(String username) | |
| 参数（按照上述顺序） | 用户名 |
| 返回 | **密码** |
| 描述 | 查找用户密码 |
| public boolean setTransation(String userId, String taskId, double earn, int phase, Date timestamp) | |
| 参数（按照上述顺序） | 用户Id，任务Id，收益，阶段标志，时间戳 |
| 返回 | **0代表成功，1代表失败** |
| 描述 | 输入用户的一条交易 |
| public List<Transaction> getAllTransactions(String userId) | |
| 参数（按照上述顺序） | 用户Id |
| 返回 | **交易记录** |
| 描述 | 返回用户的所有交易记录 |
| public List<Transaction> getAllTransationByPhase(String userId, int phase) | |
| 参数（按照上述顺序） | 用户Id，阶段标志 |
| 返回 | **交易记录** |
| 描述 | 查询某个用户的某一个阶段的所有交易记录 |
| public boolean modifyTransationPhase(String transation, int phase, Date timestamp) | |
| 参数（按照上述顺序） | 交易，阶段标志，时间戳 |
| 返回 | **0代表成功，1代表失败** |
| 描述 | 修改一条交易记录的阶段值 |
| public List<String> getAllUserNames() | |
| 参数（按照上述顺序） | Null |
| 返回 | **用户名列表** |
| 描述 | 获取所有用户名 |
|  |  |

1. 针对PM2.5计算模块数据结构接口设计
   1. 结构设计

|  |
| --- |
| public class WeatherDataUnit{  // 温度  private int temp;  // 湿度  private int humi;  // 风速  private int wspend;  // 降雨量  private int precip;  // 压强  private int pressure;  // 天气类型  private int weather;  // 表名  private static final String tableName = "weather";  // 列名  public final static byte[] TEMP = "temp".getBytes();  public final static byte[] HUMI = "humi".getBytes();  private final static byte[] WSPD = "wspeed".getBytes();  private final static byte[] PRECIP = "precip".getBytes();  private final static byte[] WEATH = "weather".getBytes();  private final static byte[] PRESSU = "pressure".getBytes();  //PM2.5浓度  private int fpm;  //PM10浓度  private int cpm;  //AQI指数  private int aqi; |

* 1. 接口设计：

表3-3 PM2.5计算模块接口表

|  |  |
| --- | --- |
| public Boolean set(String city,Date timestamp,int temp,int humi,int wspeed,int precip,int pressure,int weather) | |
| 参数（按照上述顺序） | 城市，时间戳，温度，湿度，风速，降雨量，压强，天气状况 |
| 返回 | 0代表成功，1代表失败 |
| 描述 | 存储天气信息 |
| public WeatherDataUnit getNearst(String city,Date timestamp) | |
| 参数（按照上述顺序） | 城市，时间戳 |
| 返回 | **天气信息对象** |
| 描述 | 获得与时间戳最接近的天气记录 |
| public Boolean delete(String city,Date timestamp) | |
| 参数（按照上述顺序） | 城市名，时间戳 |
| 返回 | 0代表成功，1代表失败 |
| 描述 | 删除一个城市的该时间戳的记录 |
| public Boolean set(String city, int siteId, Date timestamp, int fpm,int cpm, int aqi) | |
| 参数（按照上述顺序） | 城市名，观测点id，时间戳，Pm2.5浓度，PM10浓度，AQI指数 |
| 返回 | **Ture代表成功，否则失败** |
| 描述 | 存储一个监测点的信息 |
| public Boolean delete(String city, int siteId, Date timestamp) | |
| 参数（按照上述顺序） | 城市名，监测点Id，时间戳 |
| 返回 | **True代表成功，否则失败** |
| 描述 | 删除监测点信息 |
| public MonitorSiteDataUnit getNearst(String city, int siteId, byte[] COLName, Date timestamp) | |
| 参数（按照上述顺序） | 城市名，监测点Id，监测的信息，时间戳 |
| 返回 | **观测信息** |
| 描述 | 读取与时间戳之前或之后的最接近整点的观测信息 |

1. 针对web展示端的POI聚合的数据结构与接口设计
   1. Point是数据库存储的POI点的数据，其数据结构设计如下：

|  |
| --- |
| public class Point {    private double pixelX;  private double pixelY;  private double lon;  private double lat;  private String poiString;  private int cid; //所归属的k值  } |

* 1. 接口设计

表3-4 POI聚合接口表

|  |  |
| --- | --- |
| public boolean setPonit(int zoom, double titleX, double titleY, String pointxy,Date timestamp) | |
| 参数（按照上述顺序） | 缩放级别，图块坐标x，图块坐标y，point信息，时间戳 |
| 返回 | 0代表成功，1代表失败 |
| 描述 | 在kmeansTable中添加数据源 |
| public List<JSONObject> getCenters(int zoom, double minX, double minY, double maxX, double maxY) | |
| 参数（按照上述顺序） | 缩放级别，范围的像素坐标的最小xy以及最大xy |
| 返回 | list<JSONObject> 即是[centerx:###,centery:###],[][][][][]**]** |
| 描述 | 在某个区域之内的某个缩放级别下的center的值 |
| public List<JSONObject> getNears(int zoom) | |
| 参数（按照上述顺序） | 缩放级别zoom |
| 返回 | list<JSONObject> 返回zoom缩放比例下k个簇的k个质心最近的点 |
| 描述 | 根据zoom得到所有的near |
| public String getNearpoiInCenterZoom(int zoom, Point center) | |
| 参数（按照上述顺序） | 缩放级别zoom，质心center |
| 返回 | 返回最近的POIstring |
| 描述 | 根据zoom，center得到对应的nearest点 |
| public List<JSONObject> getPoisnfoForZoomCenter(int zoom, Point center) | |
| 参数（按照上述顺序） | 缩放级别zoom，质心 |
| 返回 | List<JSONObject>即POI信息 |
| 描述 | 得到zoom下的center下的所有的pois的所有的信息 |
| public ResultScanner getResource(int zoom) | |
| 参数（按照上述顺序） | 缩放级别zoom |
| 返回 | ResultScanner 扫描器 |
| 描述 | 得到某一个zoom下所有的点集合 |
| public boolean clusterAndImport(DataSource datasource, int zoom) | |
| 参数（按照上述顺序） | 所有的点信息，缩放级别zoom |
| 返回 | Ture代表成功，false失败 |
| 描述 | 将某个缩放级别下的点进行聚合并导入到数据库 |
| public boolean cluster(int zoom) | |
| 参数（按照上述顺序） | 缩放级别zoom |
| 返回 | Ture代表成功，false代表失败 |
| 描述 | 对某一个集合下的所有的点 取出、聚合、导入到数据库 |

### 3.3 Web可视化的需求分析与总体设计

数据可视化主要是指借助于图形化手段，清晰有效传达与沟通信息。海量的额数据对数据的加工、传播与分享带来巨大的挑战。

#### web可视化的需求分析

web可视化系统的主要功能就是对移动客户端收集上传的感知数据在浏览器端的可视化展示，可视化展示是一种简单明了直观的方式如图形图像图表的形式展现数据之间的关系与隐含的规律。我们可以得到数据可视化系统的拓扑结构如图。

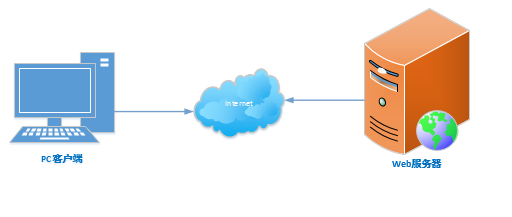


图3.5 可视化系统拓扑结构图

用户通过PC客户端的浏览器注册登录我们的站点，可以查看可视化数据的效果，浏览器向服务器请求所需要展示的数据，然后以图形图像图表等形式展示出来，用户可以通过简单地交互，从不同的维度对数据进行观察和分析。

web可视化系统的web界面主要完成用户注册登录、查看用户个人上传数据的统计展示功能，以及对平台收集到的所有感知数据的展示功能。其中，用户信息主要包括用户名、邮箱、轨迹、信誉值等基本信息以及用户上传的图片、GPS等数据信息。

Web可视化界面主要展示的功能有POI点的地理位置的展示、POI的PM2.5的变化趋势、基于POI的是图片的展示以及图片相关的PM2.5信息的展示。具体的需求有：

1. 用户注册与登录

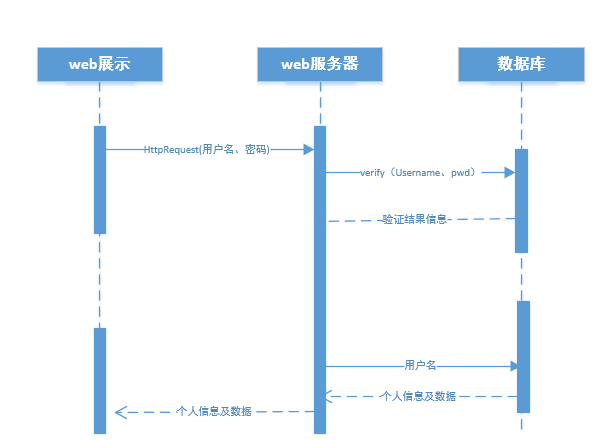
以下两张表是用户注册登录模块的交互过程，服务器端的

图3-6 用户注册过程交互

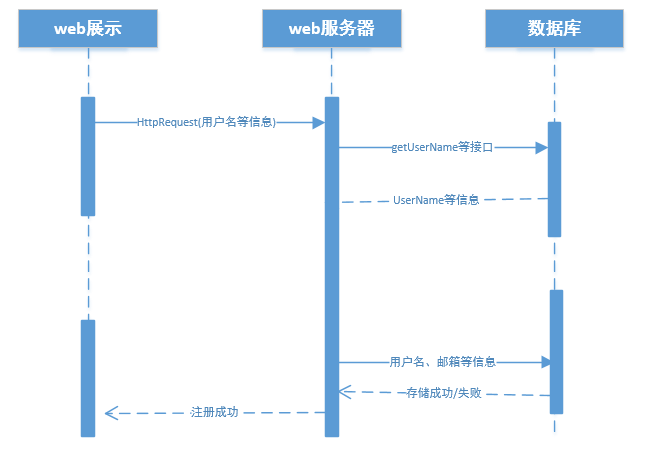


图3-7 用户登录过程交互

1. 百度地图展示

移动客户端收集上传的感知数据有GPS、光照、温度、噪声等以及多媒体数据如照片、视频和录音等。所有的这些信息都是基于某一个时间点和某一个地理位置的，这些数据信息离开了时间与地点也就没有了参考和使用价值。所以感知数据所在地理位置是展示一个重要因素，所有的收集到的感知数据与多媒体数据都是基于地图上的某一个地理位置的。地图平台有百度地图、google地图、高德地图等，但是考虑到百度地图简洁明了的界面符合大多数中国人的使用习惯，并且更新速度快等优势，本系统使用的是百度地图。

百度地图可以根据POI的经纬度信息准确定位该点，并用相应的图标展示。可是由于参与式感知平台的参与者收集的数据越来越多，在进行展示时就需要考虑到在有限的可视范围内POI的展示数量，将大量的POI直接展示在地图上，会导致图标的覆盖、重叠现象，这就会直接影响地图的可视化效果，还会影响到地图性能，从而影响到用户在使用web可视化界面时的体验，所以我们需要对展示的POI点根据地图的缩放级别进行聚合，以防止在放大或者缩小地图时，出现POI点的覆盖重叠现象，提高可视化界面的使用体验。

1. 展示的POI点会因为地图的缩放而进行聚合，从而减少地图可视化区域内POI的数量。由于POI对应的PM2.5浓度值、光照强度、噪声大小为具体的数值，并且与时间空间一一对应，它们会随着时间空间的变化而而得到不同的数值。对于同一个POI点，我们可以通过随时间变化的曲线图来对这些数据进行可视化展示。对于聚合的POI点的这些数据信息，我们通过一定的数据融合算法来实现数据的融合（数据信息的聚合）。
2. 默认情况下，我们会展示地图可视化区域中的所有POI对应的图片，当店家某一个POI图标的时候，图片展示区域只展示点击的POI所对应的图片。每一个POI点所包含的数据有基于该POI地理位置的所有时间点的所有图片，而对应不同时间点的图片的展示可以有多种展示方式，采用单图模式，以幻灯片切换的方式可以对图片采集地点的PM2.5值有一个直观的认识，采用多图模式，一个POI点的所有的图片都展示出来使得不同PM2.5值的图片对比鲜明，视觉冲击性更强、效果更佳。
3. 相应的图形图表展示

与图片展示相对应的图片PM2.5值的展示也分为单图和多图模式。PM2.5值的展示会随着图片的切换而相应的变化。由于图片的PM2.5值是一个具体的数值，可以考虑使用柱状图的高度来表示PM2.5浓度值的大小。

综上，整个数据可视化展示需求中，可以把地图作为整个展示效果的触发点。地图中展示POI点，对地图进行缩放、拖动等操作时，地图上展示的POI点都会发生变化。因此POI对应的PM2.5值、图片、光照强度、噪声等数据都会随着POI的变化而变化。

#### web可视化的概要设计

依据上述对web可视化系统的需求分析可以得出数据可视化系统的主要功能是对感知数据和多媒体数据在web端的可视化展示，以及服务器端对web浏览器端的请求的处理以及相关数据的计算。如下图为可视化系统的功能框架图：

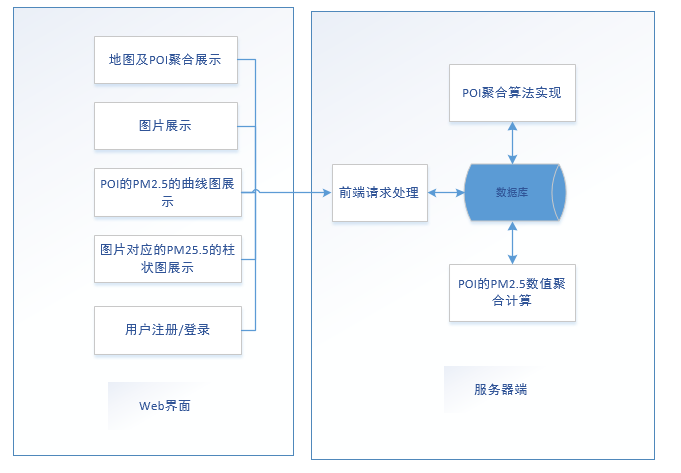


图3-8 功能架构图 1

Web可视化展示主要包括web界面和服务器端，其中web界面展示包含地图及POI聚合展示、图片展示、图片PM2.5柱状图展示、POI点的PM2.5折线图展示。服务器端主要功能涉及到对浏览器端的请求处理、POI聚合计算、POI对应的PM2.5值聚合计算等。

数据可视化系统的主体流程是用户注册登录web站点，浏览器向服务器端发送展示数据的请求，服务器端通过数据库接口返回web端所需要的数据，浏览器将接收到的数据进行可视化展示。由上述的主体流程，可以得到如下的数据可视化系统的总体架构图。

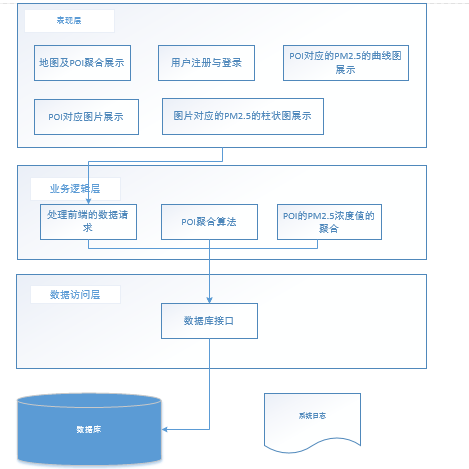


图3-9 可视化系统的总体架构图 1

由系统总体架构图，最底层是数据库和系统日志，数据库是采用分布式数据库Hbase，将移动客户端的感知数据与多媒体数据以不同的表结构存储于数据库，系统日志实时保存了所有对数据库的操作记录，这一层是架构的数据基础；数据访问层主要是指数据库接口，此处主要是包括为数据可为上层业务逻辑层提供的数据增删查改接口；业务逻辑层与数据访问层的交互主要是由数据库接口实现，主要是实现计算POI聚合、POI点PM2.5值聚合以及处理浏览器端交互时对服务器端的实时请求；表现层主要是web界面中地图及POI聚合的展示、POI点的PM2.5值的趋势图、图片的展示、图片的PM2.5柱状图。

#### POI聚合算法研究

参与式感知平台海量数据都是基于地图上地理位置来展现的，随着数据量的增多，地图上的POI点也随之增加，如果将POI点直接展示到地图上，就不可避免的出现POI图标覆盖、重复的现象，这样不仅会影响界面的美观，还会影响用户交互体验性。因此，需要在地图不同的额缩放比例下，对POI点进行简单地聚合。本课题中依据传统K-means聚合算法，并结合百度地图的丰富的坐标体系，实现了POI的聚合算法，即基于百度地图的基于网格的K-means算法。该算法针对传统K-menas算法存在的问题，提出新的网格确定、k值选择和k个初始质心选择的新的方法。

（1）基于网格

由第二章我们了解到百度地图的坐标体系以及传统的K-means算法，我们可以把百度地图看作是整个地图图片切割成若干个图块显示，每个图块可以看作一个网格。这样在百度地图每个缩放级别下，每个POI点都有对应的唯一的图块坐标。基于网格的思想就是指在不同的地图缩放级别下，每个POI都有对应的网格，而网格的数目就是K值，即该缩放级别下聚合的簇数，在此，每个POI图块坐标的计算过程如下：

1. 可以依据百度地图提供的API计算出其平面坐标：

var projection = new BMap.MercatorProjection();

var point = projection.lngLatToPoint(new BMap.Point(116.404, 39.915));

1. 依据平面坐标可以根据公式计算出其像素坐标：

**像素坐标 = |平面坐标 × 2 zoom - 18|**

1. **依据像素坐标，即可得到POI对应的图块坐标：**

**图块坐标 = |像素坐标 ÷ 256|**

（2）k个初始质心的选择

在百度地图某个缩放级别下，所有POI的个数是n，即原始数据集合（x1,x2,…,xn），此处的每个xi为二维向量，所有POI所在的图块为k个，即将原始数据分成k类S={S1,S2,…,Sk}。在此，每个初始质心指对应图块中所有的POI点的各维度的算数平均数，则初始质心坐标为：



其中m为每个质心的POI个数。

本课题的POI展示主要是基于百度地图，实现POI点的聚合充分利用百度地图丰富的坐标体系，巧妙的实现网格的划分以及K-means聚合算法中的K值选择和质心的额确定，这样，即使POI点的数据增加，也不会出现覆盖现象。

#### POI点的PM2.5值的聚合算法

POI点在百度地图上的地理位置可以依据地图的缩放而聚合，web可视化中POI点PM2.5值的曲线图是展示地图中可视化区域内的POI点的PM2.5值，因此，当地图发生缩放、拖动等操作时，POI需要聚合，POI点的PM2.5值的曲线图也需要相应的变化。此处，需要实现对POI点的PM2.5值的聚合，本课题中选择基于距离的加权平均算法：

在POI点进行聚合之后，得到k个质心，每个质心会包含若干不等的POI点，我们知道每个POI点的PM2.5值信息，依据该算法可以得到k个质心的PM2.5值。上述公式中，权值的分配原则是每个POI点到对应质心的距离倒数，每个质心对应的PM2.5值是这个质心所包含所有的POI点的PM2.5浓度值的加权平均值。

上述的距离的计算方法是，按照平面地图坐标中的像素坐标来计算的，假设质心坐标是（X，Y），该质心中某一个POI点的坐标是（x1，y1），那么它们之间的距离就是：

## 第四章 数据处理与可视化系统的详细设计与实现

结合上一章对平台的需求分析，以及对数据库管理系统和web可视化系统的需求分析和总体结构的设计，我们这一章将要介绍数据管理系统和web可视化系统的详细设计与实现。

### 4.1 数据库管理系统的详细设计与实现

数据库管理系统的详细设计主要是包括数据可对平台其他模块如轨迹分析预测模块、激励模块、PM2.5计算模块、天气信息爬取模块、web可视化模块等提供的存储表结构以及这些存储表的增删查改操作的接口。

#### 数据库管理系统的数据库设计

##### 轨迹预测模块

轨迹预测模块，主要是通过记录用户的历史轨迹信息，并且在服务器端进行存储，利用轨迹分析预测算法，计算出用户到达指定地理位置的概率。这些数据是为精确推送激励任务提供了重要支持。

轨迹分析预测模块，首先要把用户的轨迹路线的轨迹点按照一定的规则，一一映射到地图上的连续排列的正方形网格中，这样的网格是有一定规律的呈现到百度地图的平面坐标上，称为区域。

为实现轨迹预测模块的相关功能，数据库管理模块主要涉及三个存储表，分别为轨迹数据表、轨迹格子划分表和预测结果表。

原始轨迹数据表主要是对参与者轨迹数据的记录，表中的数据主要有用户Id，轨迹Id，经纬度以及当前位置所在的格子的编号以及时间戳，这是后来对用户轨迹分析算法的输入，该数据来源主要是微软的GeoLife数据集，主要是截取了北邮以及北邮周围的轨迹数据，其中轨迹条数为977条，用户数目为92个，位置点位66559个。

轨迹格子区域划分表，可以依据某点的经纬度坐标得到该点的格子编号。将所截取的区域为1.5km\*1.5km，将区域划分为15\*15的正方形格子区域，每个格子的大小为100m\*100m。每个正方形格子记录下格子的对角线的经纬度，以判断给定点的经纬度坐标属于哪个格子范围。

预测结果表，主要是将预测结果存储到数据库，将可能的用户存入到相应的格子区域中，可以根据用户Id和轨迹Id来读取轨迹点经过的格子序列和时间。

对应的用户轨迹表的表结构设计如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Table | RowKey | Family | Columns | Value | Timestamp |
| TrajTable（traj） | {userName， Traj\_id,  } | Info | Path1 | {lat, lon, grid\_id} | Gps\_time |
| Path2 | {lat, lon, grid\_id} |  |
| pathn | {lat, lon, grid\_id} |  |

格子区域表结构设计如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Table | RowKey | Family | Columns | Value | Timestamp |
| GridTable  （grid） | {  Upper\_left\_lat,  Upper\_left\_lon  Bottom\_right\_lat  Bottom\_right\_lon  } | info | grid | Grid\_id |  |
| predict |  |  |

##### PM2.5分析计算模块

PM2.5分析计算模块在数据库端主要涉及到气象数据抓取和图像模型存储。

1. 气象数据抓取

气象数据抓取主要是实时存储爬取的空气质量和天气数据，负责全国主要城市天气数据以及PM2.5浓度数据的抓取与存储。数据库中主要是Monitor表和Weather表，其中，Monitor表主要包含假侧站点id，PM10,PM2.5与AQI指数，记录时间。Weather表主要包含城市id，天气状态，湿度，温度，风速，气压，降雨量，记录时间等。

数据库中对观测表结构设计如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 行键 | | | INFO列族 | | |
| 城市 | 观测点ID | 观测时间 | Fpm | Cpm | Aqi |

数据库中对天气表结构设计如下：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 行键 | | INFO列族 | | | | | |
| 城市 | 观测时间 | 温度 | 湿度 | 风速 | 降雨量 | 压强 | 天气状况 |

1. 图像模型存储

PM2.5计算模块中的图像分析步骤结束之后，需要存储模型结果。图像分析模块认为，不同的图像以及拍照手机型号不同，模型就不同，所以数据库端需要为每种手机型号存储结果模型。但是，随着时间以及数据的不断更新，结果模型的版本也会不断的变化和优化。当需要利用结果模型对图像进行处理的时候，图像分析模块需要查询最近最新版本的结果模型，所以，数据库管理系统需要记录下模型的所有版本，并且可以针对时间戳来进行模型的查询检索。

数据库中对图像模型存储Rtf表结构设计如下：

|  |  |
| --- | --- |
| 行键 | INFO列族 |
| 手机型号 | T1 - CRF模型  T2-CRF模型 |

##### 4.1.1.3 激励模块

推送和激励模块主要涉及到三个表，任务表、用户表和报价表。其中任务表主负责任务的创建、查询、删除操作以及任务字段的插入与更改。用户表主要是负责用户信息的录入，用户交易信息的查询。报价表主要是对任务的报价统计。在数据库的表结构设计上将报价表和任务表合二为一。如下表所示：

用户表主要是：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Table | RowKey | Family | Columns | Value | Timestamp |
| userTable | userID | Info | Info：info |  |  |
| task | Task:taskID | Earn:double |  |

任务表如下表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Table | RowKey | Family | Columns | Value | Timestamp |
| skTable | Task\_ID | Info | Info: Proper | {  taskType:1/2/3/4,  IncentiveType:1/2,  Budget:9999.00,  dataNumber:787,  begintime: long,  biddingdeadline:long  endingtime: long  } |  |
| State | State: taskphase | 1/2/3/4 |  |
| State:participants | [“”,””,””] |  |
|
| Bid | Bid:userId | {userId:XXX,bidPrice:XXX} |  |

##### Web展示模块

Web展示模块主要负责POI聚合以及结果存储，客户端采集图像的存储和POI站点信息的存储。

1. POI聚合，POI聚合算法如第二章所述，在POI聚合表中的主要操作包括得到某一个zoom下的所有点的集合，得到某一个缩放级别zoom下某一簇的所有POI点，获取在地图某一个缩放级别zoom下某一个区域范围内的簇质心信息，获取某一个缩放级别zoom下距离k个簇质心最近的k个点。由此，该表结构设计如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Table | RowKey | Family | Columns | Value | Timestamp |
| kmeansTable | {  Zoom，  GridX，  GridY  } | Info | pois | [pixelX/pixelY/lat/lon/poiId] |  |
| Cluster | Center | pixelX/pixelY/0.0/0.0/null |  |
| include | [pixelX/pixelY/lat/lon/poiId], |  |
| Nearest | pixelX/pixelY/lat/lon/poiId |  |

1. 图片表，主要存储客户端采集上传的图片，由需求分析可知，手机客户端和web可视化系统会对图片表进行频繁访问和存取图片信息。

手机客户端的对图片表的操作主要是：

* 1. 图片的存储，客户端上传图片，将图片信息存储到图片表中，主要是存储该图片所在的经度、纬度、拍照时间、用户名、图片对应的PM2.5的值、图片的名称以及图片所在的POIId。
  2. 图片的存取，上传图片之后，在创建该图片对应的图像模型时，需要将对应的图片从数据库中读取出来。主要的读取方式有：依据经纬度的范围、时间范围来读取图片的信息。
  3. 图片的删除，当删除某一个POI时，相应的图片信息也需要删除，主要的删除方有：依据POIId删除，即删除某一个POI下的所有图片；依据图片的路径来删除某一张图片。
  4. 另外，数据库系统还提供了图片的导入接口。图片的存储途径有两个，一个是通过手机客户端上传，一个是通过本地图片手动导入。导入接口主要是将所导入图片信息放到一个文件内，然后以文件为参数，将图片信息导入到数据库。

综上，我们设计的表结构如下，行键由（时间戳，经度，纬度，用户名）组成，可以使图片表以经纬度或者时间作为条件查询数据库。列族有info和poi，其中列族info包含图片的名称（列名pic）和对应的PM2.5的值（列名fpm），列族poi包含列poi，即图片所在的POI的Id。即如下表所示：这样设计列族和列的目的在于可以依据图片名称和图片对应的POIId来删除某一张或者所有POI的图片。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Table | RowKey | Family | Columns | Value | Timestamp |
| Pic | {  Timestamp,  Lat,  Lon，  UserName  } | info | fpm | 该图片的PM值示例：  \x00\x00\x00\x1D |  |
| pic | 图片名称 |  |
| poi | poi | 所在的POIId |  |

1. POI表，关于POI表的操作主要是：
   1. 存储，存储POI点对应的图片信息（路径、PM2.5的值、拍照时间），存储所有的POI信息，存储POI点对应的图像模型。
   2. 读取，获取POI信息，包括获取某一个POI点的最新的n张图片、获取所有POI的最新的一张图片、获取所有POI的左右图片、获取某POI的所有图片、获取所有POI的Id，获取POI的图像模型、读取某POI的图片数目。
   3. 删除，删除POI信息主要是依据POIId来检索POI，并将该POI信息删除。

综上所述，POI表的表结构设计如下，行键有{POIid，时间戳}，{POIId， long.MIN\_VALUE}，列族有dayData和info，前者包含Poi——fpm、图片路径和模型信息的路径、图片的个数；后者包含poi\_info列，主要是poi的信息。主要表结构如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Table | RowKey | Family | Columns | Value | Timestamp |
| POITable | {Poi，  Timestamp  } | dayData | Poi\_fpm | {XXXX} |  |
| pic | {“pic”:XXXXX,  ”fpm”:XXXX} | Gps\_time |
| {Poi，-9223372036854775808  } | Model\_info | {XXXX}  (存的是model的路径) |  |
| count | {XXXX}  {long类型的值，表示照片的数量} |  |
| info | Poi\_info | value={  "poi":"XXX",  "lon":XXX,  "lat":XXX,  “builder”:name |  |

1. 格子区域表，格子是在地图上时一个正方形的区域，由左上和右下角的经纬度来确定。格子区域表主要是计算某一经纬度坐标属于哪个格子区域，查找距离某一范围的的POI点，存储格子信息，查找某一个格子是否存在，删除某一个POI所在的格子。综上，所有的查询等操作条件是某一个经纬度或者某一个POIId，而我们可以通过查询POIId来得到相应的经纬度，所以对各自区域表的操作都是基于某一个经纬度，而通过经纬度坐标可以计算出其对应的格子信息的左上角和右下角的经纬度，因此，这个格子区域表结构中的行键设计为左上角和右下角的经纬度坐标。列名为info：gridPoiId，value为{格子Id，PoiId，time，创建者}。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Table | RowKey | Family | Columns | Value | Timestamp |
| GridTable | {  Upper\_left\_lat,  Upper\_left\_lon  Bottom\_right\_lat  Bottom\_right\_lon  } | info | gridPoiId | {  "gridId":1,  "poiId":"904732353",  "time":1422405945000,  "builder":"lshuai“  } |  |

### Web可视化系统的详细设计与实现

结合上一章对数据可视化系统的需求分析，我们可以将数据可视化系统分为两个模块：web前端展示模块和服务器端的处理模块，本节主要阐述这两个模块的详细设计和具体实现。

#### 服务端POI聚合详细设计

上一节讲述了数据库对前端展示模块中用到的表的设计。现在对POI聚合的流程图如下：

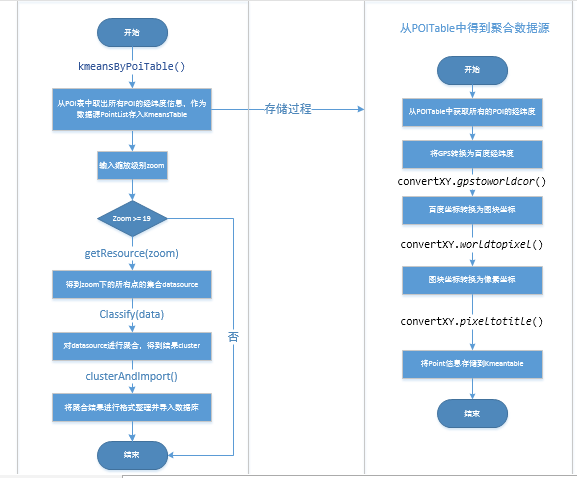
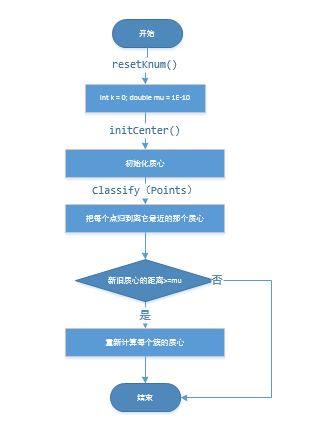


图4-1 POI聚合流程图



#### 4.2.2 Web前端展示详细设计

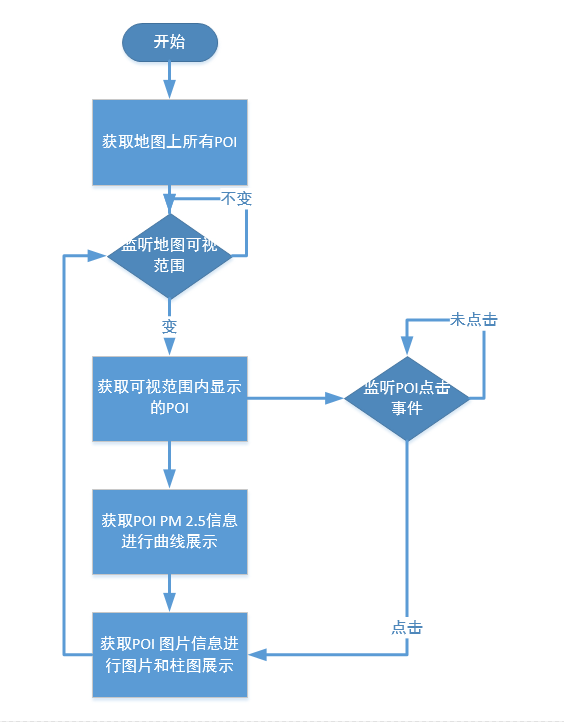
依据可视化系统的需求分析，可知web前端展示模块主要负责的工作包括百度地图的展示、地图中POI的聚合展示、地图可视范围中的POI一段时间的PM2.5值等数据的展示、POI对应的图片的展示以及每张图片对应的PM2.5浓度值的展示。由此我们可以按照展示内容区域划分为四个部分，分别为地图展示，POI对应的相关数据的折线图，图片展示和图片对应的相关数据的柱状图。

图4-2 可视化系统系统流程图

##### 4.2.2.1 用户注册登录模块

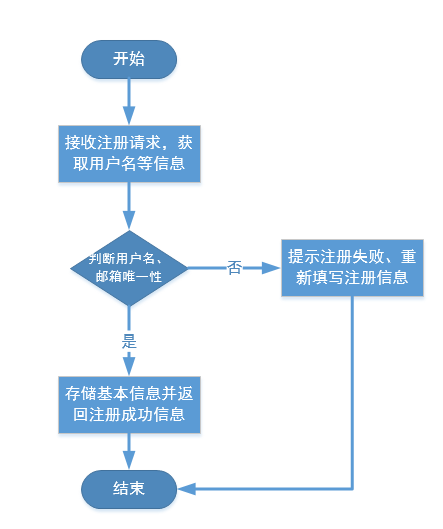
由需求分析可知，用户注册登录模块的流程图如下：

图4-3 用户注册流程图 1 1

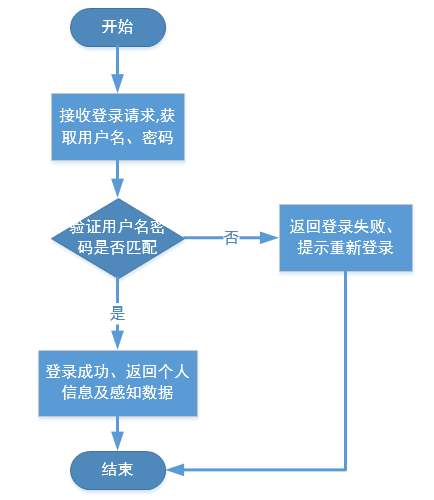


图4-4 用户登录模块流程图 1

##### 4.2.2.2 地图展示区域

图4-3 用户登录流程图 1

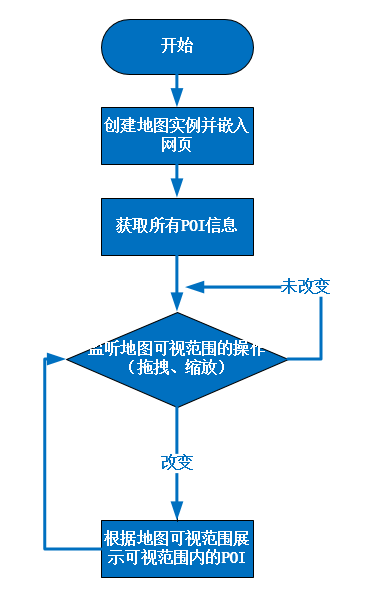
由需求分析可知，地图展示区域，主要是在百度地图的基础上，添加参与式感知平台自定义的一些PM2.5站点（POI）信息，主要是在地图上展示PM2.5站点的地理位置并且以特定的标志物来展示，随着百度地图的缩放以及拖动等操作，POI点会相应的发生聚合，那么地图可视化范围内的POI点会相应变化，对应的POI点的曲线图也会变化，当点击某一个POI点时，地图展示区域右侧的图片也会相应改变。该展示区域的主要流程如图所示：

图4-5 地图展示区域流程图 1

由上述流程图可知，我们首先应该利用百度地图API来生成一个地图实例，主要值利用百度地图Javascript API提供的应用程序接口，实现地图的基本功能展示（支持2D图、3D图、卫星图）、评议、缩放、拖拽等还有地图空间展示如比例尺、地图类型等。首先需要申请密钥（ak），ak主要通过如下方法将百度地图引用到网页中：

|  |
| --- |
| <script src="http://api.map.baidu.com/api?v=2.0&ak=您的密钥" type="text/javascript"></script> |

然后就可以利用百度地图提供的api生成地图实例，其创建以及配置代码如下在js文件中：

|  |
| --- |
| // 创建地图实例  var bmap = new BMap.Map(document.getElementById('map'));  bmap.addControl(new BMap.NavigationControl());  //添加默认缩放平移控件  bmap.addControl(new BMap.ScaleControl());  //添加默认比例尺控件  bmap.addControl(new BMap.OverviewMapControl());  //添加默认缩略地图控件  bmap.addControl(newBMap.MapTypeControl());  //2D图，卫星图  bmap.enableScrollWheelZoom();  //启用滚轮放大缩小，默认禁用  bmap.enableContinuousZoom();  var point = new BMap.Point(116.404,39.915);  //Point(lng:Number, lat:Number) 经度，纬度  bmap.centerAndZoom(point,12);  //将地图的缩放级别，默认为12 |

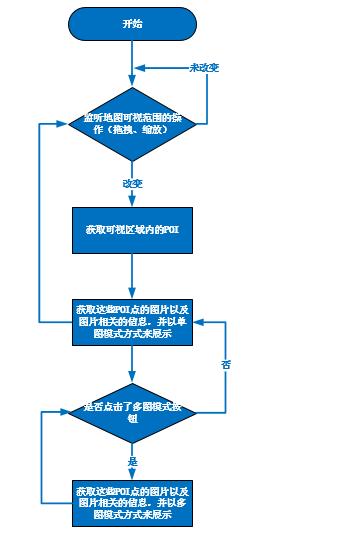
为有更好的用户体验性，将不同的POI用不同颜色来表示，同时在地图可视范围之内，为防止因为POI数目太多导致POI标志重叠拥挤现象，我们需要把这些POI随着地图的缩放而聚合，也是上文中提到的聚合算法。具体用到的百度地图的方法如下：

|  |  |  |
| --- | --- | --- |
| 方法 | 返回值 | 描述 |
| Map(container:String|HTMLElement[, opts:MapOptions]) | Map | 在指定的容器内创建地图实例，之后需要调用Map.centerAndZoom()方法对地图进行初始化。未进行初始化的地图将不能进行任何操作。 |
| getBounds() | Bounds | 返回地图可视区域，以地理坐标表示 |
| getZoom() | Number | 返回地图当前缩放级别 |
| getInfoWindow() | InfoWindow|Null | 返回地图上处于打开状态的信息窗的实例。当地图没有打开的信息窗口时，此方法返回null。 |
| pixelToPoint(pixel:Pixel) | Point | 像素坐标转换为经纬度坐标。 |
| pointToPixel(point:Point) | Pixel | 经纬度坐标转换为像素坐标。 |
| Click | {type, target, point, pixel, overlay} | 左键单击地图时触发此事件。  当双击时，产生的事件序列为： click click dblclick |

创建地图并且配置基本信息之后，利用百度地图API得到地图的可视区域的边界坐标和缩放级别，向服务器发送请求，返回可视区域和当前缩放级别下的POI点的坐标以及POI点的所有的信息，包括图片、Poi点的PM值等信息。

|  |
| --- |
| EventWrapper.addListener(bmap, 'tilesloaded', debounce(function(event) {  var zoomLevel = bmap.getZoom(); //得到地图的缩放级别  var myBounds = bmap.getBounds(); //得到地图可视区域的边界坐标  } |

##### 4.2.2.3 图片展示

当点击地图上某一个POI点时，会向服务器端放松请求，服务器端返回该POI点的所有的图片信息，包含图片路径、图片对应的PM2.5值以及时间戳。Web端的展示包含单图模式与多图模式。其流程图如图：

4-6 图片展示区域流程图 1

* 单图模式

单图模式，主要应用的技术是jquery的一个视频画廊插件blueimp.Gallery，在第二章中已经对该插件做了简单地介绍。

* + - * 多图模式

多图模式，主要是为了能更清晰地展示某POI点的所有图片，对于不同的PM2.5的的图片对比更加鲜明。

##### 4.2.2.4 柱状图展示

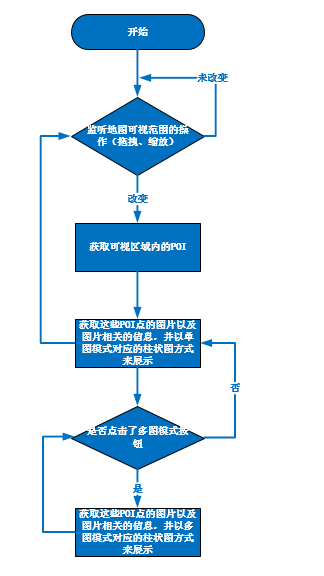
柱状图的展示主要对应于图片的PM2.5值的展示，柱子的颜色主要与Pm2.5颜色值相同，柱状图与图片展示对应，分为单图模式和多图模式。

图4-7 柱状图展示区域流程图

##### 4.2.2.5 折线图展示

主要是请求地图可视区域范围内所有POI点的某一段时间的PM2.5值的趋势，在时间维度是可选的，有三种选择模式，默认展示最近七天的PM2.5的值，还可以选择当天按24小时内的PM2.5浓度趋势，也可以通过选择某一个时间段来展示过去几天的PM2.5的曲线。

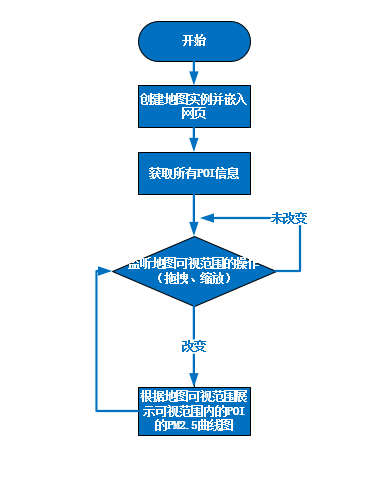


图4-8 曲线图展示区域流程图 1

### 开发环境介绍

#### 4.3.1 数据库开发环境介绍

系统的开发环境如下：

后台编程语言：java（JDK 1.7.0）

集成开发环境：Eclipse

打包与构建工具：Ant1.94

编辑器：sublime

版本控制：svn

系统的运行环境:

操作系统： Ubuntu 12.04

内核版本： Linx3.2.0-20-generic

非关系型数据库：Apache Hbase 0.94.27

底层文件系统： Hadoop Distributed File System 0.23.10

#### 4.3.2 Web可视化开发环境

系统的开发环境如下：

后台编程语言：java（JDK 1.7.0）

集成开发环境：Eclipse

编辑器：sublime

版本控制：svn

系统的运行环境:

操作系统： Ubuntu 12.04

内核版本： Linx3.2.0-20-generic

### 本章小结

本章主要介绍了数据库管理系统的详细设计包括对各个模块的表结构设计，以及对应的存储表的重要接口，然后介绍了web可视化的各个模块的详细表结构设计以及算法实现，最后介绍了数据库管理系统以及web可视化系统的开发和运行环境。

## 第五章 系统测试

前面对参与式感知平台的需求分析、总体设计和详细设计做了详细的阐述，并对其中的关键部分的具体实现做了简单介绍。本章详细描述参与式平台中的web可视化系统进行功能测试，主要是介绍系统的测试环境，测试方法，演示并检验运行效果，最后对测试结果进行分析。

### 5.1 测试环境

数据可视化系统主要是参与式感知平台的数据在PC客户端的web端页面的展示。可视化系统的测试可以直接通过在浏览器上展示的效果以及简单交互来完成，所以测试的工具主要是一台连接到服务器的安装有浏览器的PC客户端。

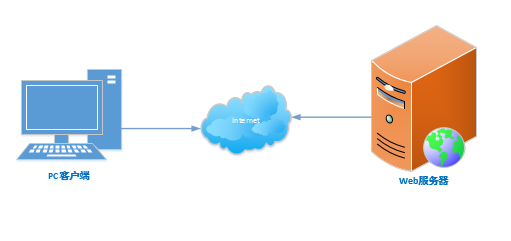


图5-1 测试环境拓扑图

在该测试环境中，PC端操作系统为Windows 7 旗舰版SP!，web展示采用Chrom 47.0.2526.106 m,服务器端，采用Tomcat7.0服务器软件进行服务请求处理，数据存储采用基于Hadoop的分布式数据库Hbase，版本号为0.94.27。

### 5.2 测试目的

对数据可视化系统的测试目的主要是检验该系统的实现是否符合需求分析中的所有的功能性和非功能性的要求，查看最后实现的功能是否运行正常，对功能的完善性以及交互界面的美观等进行测试、检验和评价。

界面视觉上，主要分为四个区域，一个是地图展示区域、图片展示区域、POI点的PM2.5值的曲线图区域和图片的PM2.5的柱状图区域。

系统的功能上，地图展示、POI位置点的展示、POI点的聚合、POI点的PM2.5时间趋势图、POI的图片展示、图片对应的PM2.5柱状图展示。首先有百度地图的展示，在地图上标有参与式感知平台的一些自定义的POI点，当对地图进行拖拽、缩放操作时，POI点会随之聚合，并且地图可视区域内的所有POI点的PM2.5浓度值的时间变化曲线也要随之变化，当点击地图上的某一个POI点时，图片展示部分会展示所点击的POI点的所有图片，图片展示会有单图模式和多图模式，图片的的单图模式和多图模式也有相对应的柱状图展示。

可视化界面的美观性上，界面中的四个区域，要相辅相成，和谐统一。由于地图上POI点标志与POI的PM2.5曲线有关联、图片与POI有关联、图片与图片的PM2.5值的柱状图也有关联，所以，为了更加清晰明了的展示相互间的关系，最简单直接的办法就是用颜色来区分不同的POI点。比如，地图上的POI点是红色，那么对应的曲线图也是红色的折线，POI对应的图片上方的标志也是红色。图片的PM2.5值的柱状图的每条柱子可以依据PM2.5的范围值来展示成相应的颜色，如下表所示：

表5-1 PM2.5浓度分级表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PM2.5 24小时平均（ug/m3） | 空气质量指数 | 空气质量指数级别 | 空气质量指数类别 | 空气质量表示颜色 |
| 0-50 | 0-50 | 一级 | 优 | 绿色 |
| 51-100 | 51-100 | 二级 | 良 | 黄色 |
| 101-150 | 101-150 | 三级 | 轻度污染 | 橙色 |
| 151-200 | 151-200 | 四级 | 中度污染 | 红色 |
| 201-300 | 201-300 | 五级 | 重度污染 | 紫色 |
| >300 | >300 | 六级 | 严重污染 | 褐红色 |

### 5.3 测试方案及过程

由于平台网络访问的限制，在本次数据可视化系统的测试中，参加测试的人员只是我们北京邮电大学的实验室的协作感知的学生，学生有统一的手机客户端来采集数据，为了增大数据量，提高实验质量，部分数据由网络上可用数据以及系统设定的默认填充或者手动填写以确保测试效果的完整性和真实性。下图是整个界面：

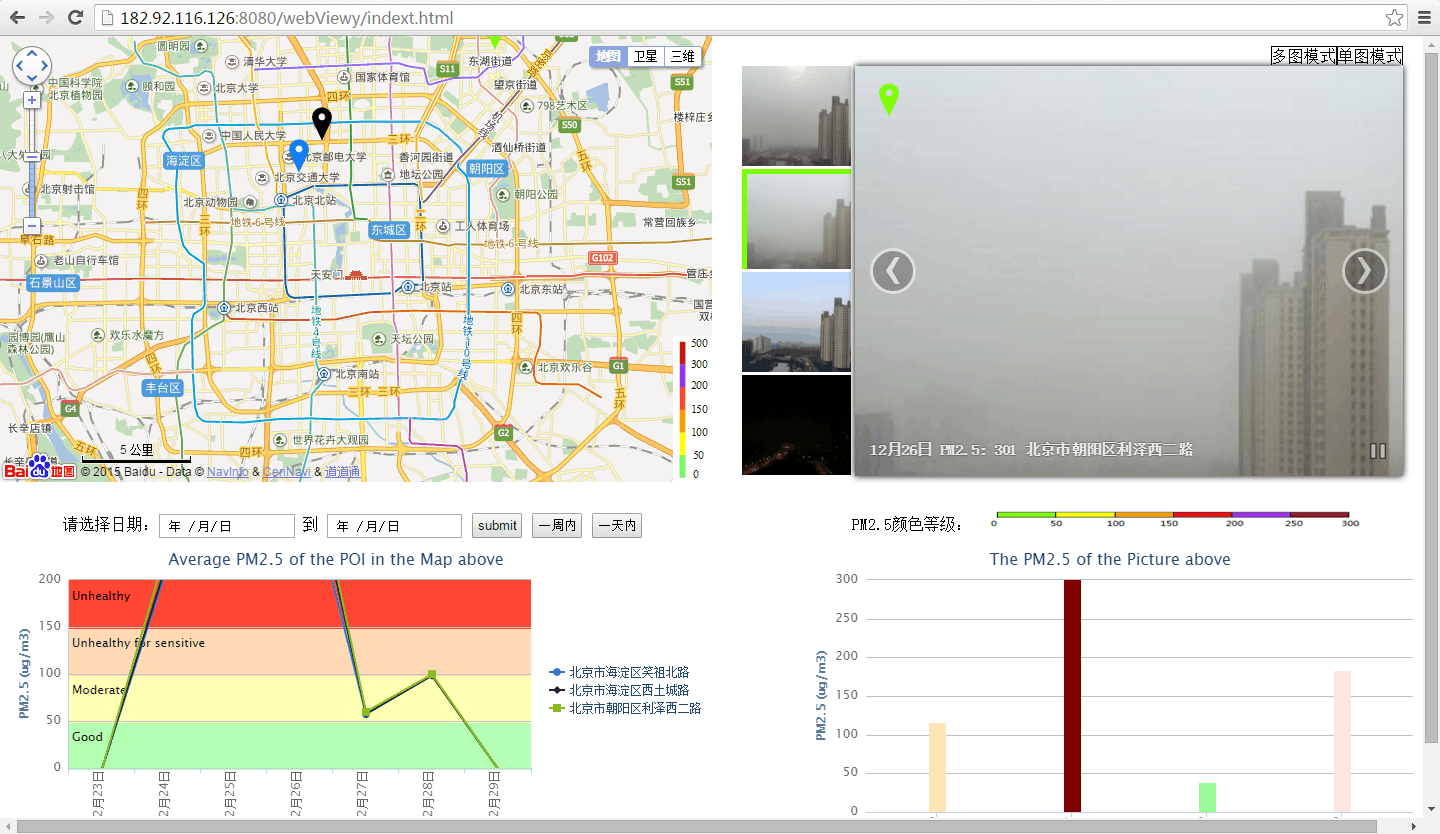


图5-2 web展示界面 1

#### 5.3.1 POI展示以及聚合测试方案

* 1. 测试内容：POI展示及聚合主要测试基于参与式感知平台所采集数据的百度地图的基本功能的展示、地图上POI点的展示、POI点随地图缩放的聚合与分散。
  2. 测试步骤：开打浏览器，在浏览器的地址栏输入web前端页面在服务器端的网址，查看地图以及地图上的POI标志；在地图展示区域点击鼠标左键，对地图进行拖拽操作，地图的可视范围进行相应的平移，对应的POI点的标志也随之改变；在地图展示区域滚动鼠标或者点击地图边栏的zoom标记（”+”,”-”），对地图进行缩放操作，可视区域会发生缩放和聚合，该区域内的 POI点也会进行相应的聚合。
  3. 预期结果：初始打开web展示页面，地图上显示一个POI点，代表北京，当对地图进行拖拽时，会看到上海天津的POI标志；当对地图进行放大时，会看到一个POI点会分裂为多个POI点，放大到一个程度，会看到北京邮电大学的几个POI点，包含北京邮电大学的几个建筑物所在的POI点，当地图缩小时，只展示一个中国，可视区域内将会把上海、天津以及北京的POI标志合成为一个。
  4. 测试结果：实际测试结果如下图所示，图。。是初始页面加载时的POI显示，当对地图进行放大操作时地图上的POI展示效果如图。。，当对地图进行缩小操作时，地图上的POI展示效果如图所示：

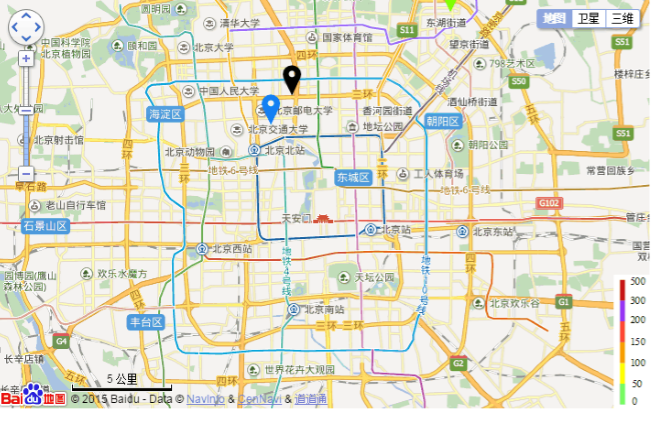


图5-3 地图展示区域测试图 1

* 1. 结果分析：由上图可知测试结果，首先地图在初始加载时能正常在地图上展示POI点标志，当对地图进行缩放操作时，地图上的POI点可以随之进行聚合和分散，经测试，系统实现符合我们对测试结果的预期效果。

#### 5.3.2 POI点的PM2.5变化曲线测试方案

* 1. 测试内容：POI点的PM2.5的变化曲线的测试主要是当我们对地图进行缩放操作时，地图上的POI点会相应的聚合，与此同时，POI点的PM2.5曲线图也会相应变化，曲线图默认显示可视区域内POI点的最近一周的PM2.5的曲线图，当点击“一天”按钮时，会展示这些POI点的当天的PM2.5的曲线图，当选择开始时间和结束时间时，会展示这段时间内的PM2.5的变化曲线图。
  2. 测试步骤：打开web前端展示页面，可以查看地图上的POI展示以及曲线图区域的PM2.5变化曲线；在地图展示区域点击鼠标左键，进行拖拽地图，查看曲线图区域的PM2.5变化曲线；在地图展示区域内滚动鼠标滑轮或者点击地图zoom标记（“+”，“-”）的操作可以改变地图的缩放级别，此时可以看到地图可视区域内的POI点的聚合效果，以及对应的曲线图的变化效果，并且可以观察曲线图中的各个POI的曲线颜色与地图展示区域中的POI点的标志颜色是否一致。
  3. 预期结果：初始地图上显示一个POI并且可以看到POI的PM2.5变化曲线只有一条曲线也即该POI点的最近一周内的PM2.5变化曲线图；当对地图进行放大时，地图展示区域内的一个POI点会分散为n个，相应的曲线图区域的曲线条数也会变为n条即这n个POI点的PM2.5变化曲线；当点击“一天”按钮时，曲线图会展示当天的PM2.5变化趋势；当选择完开始时间和结束时间后，曲线图就会展示这段时间的PM2.5的变化趋势。
  4. 测试结果：

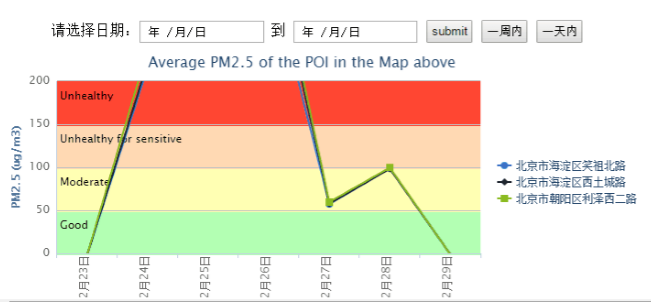


图5-4 POI的PM2.5曲线图测试 1

* 1. 结果分析：由上图可知，初始地图上只有一个POI点，对应的曲线图区域也只有一条该POI点的PM2.5曲线图，当对地图进行缩放时，可以看到一个POI点分散成三个，对应的曲线图区域也有三条曲线分别表示这三个POI点的PM2.5值变化趋势，当改变时间段时，曲线图也相应的展示该时间按段的PM2.5的曲线图，并且可以看到曲线图的每条折线的颜色与该折线对应的POI点的标志颜色是一致的。

#### 5.3.3 图片展示测试方案

1. 测试内容：图片展示测试主要是测试某个POI点的所有图片，可以有单图模式和多图模式，单图模式通过图片自动轮播的方式，将图片动态的展示，图片上可以展示图片相关的信息，如图片的GPS信息、拍照时间、PM2.5的值、对应的POI点的信息、对应的POI点的标志。多图模式，是将所选择的POI点的所有的图片全部展示到图片展示区域。
2. 测试步骤：首先打开web前端展示页面，查看地图上的POI点以及图片展示所显示的图片；在地图展示区域滚动鼠标或者点击鼠标右键平移拖拽地图时，地图上的POI相应改变，图片展示区域也相应变化；当点击地图展示区域中的某一个POI点时，图片展示区域会显示该POI点所包含的图片；当点击多图模式的时候，会展示相应POI点的多图模式。
3. 预期结果：初始web展示页面中，只有一个POI点，图片展示区域中会展示该POI点的所有图片的单图模式，每张轮播的图片上有该图片对应的GPS信息、PM2.5以及拍照时间还有对应的POI点的标志；当对地图进行拖拽、缩放德国操作时，图片展示区域也会相应的变化；当点击地图展示区域中的某一个POI点时，图片展示区域也会相应的展示该POI点的单图模式；当点击多图模式的按钮时，图片展示区域就会由该POI点的单图模式展示为多图模式；
4. 测试结果：



图5-5 图片展示测试结果 1

1. 结果分析：上述测试结果是单图模式下的图片展示，其中左侧边栏是依次要展示的单图，其中带绿色边框的是当前展示的大图，可以看到图片的左上方是一个对应的POI标志，是与地图上的POI标志颜色相一致；在图片的左下方是这个图片的拍照时间和图片的地址，可知，该测试结果符合预期需求。

#### 5.3.4 状图展示区域

1. 测试内容：柱状图展示区域的测试主要是测试图片切换时，柱状图也会随之动态切换；与图片展示区域的单图和多图模式相对应，柱状图展示区域也分为单图模式和多图模式。柱状图的每根柱子的颜色与PM2.5等级颜色相同。当鼠标移到某一根柱子时，会在柱子上方显示相应的图片的信息，如拍摄时间、对应的POI点位置和PM2.5的值。
2. 测试步骤：打开web初始化界面，可以看到图片展示区域是对应于百度地图的POI点的单图模式，柱状图展示区域是表示单图模式下的柱状图；当点击多图模式时，查看柱状图展示区域的变化。
3. 预期结果：web初始化界面上，地图展示区域有一个POI点，图片展示区域是展示该POI点的单图模式；当点击多图模式按钮时，对应的柱状图就展示为该POI点的所有图片的PM25柱状图；柱状图的每根柱子的颜色取决于该柱子对应的PM2.5值，依据PM2.5分级颜色来确定。
4. 测试结果：

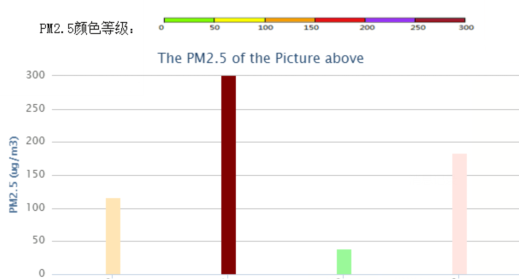


图5-6 图片的柱状图测试 1

1. 结果分析：有测试结果的截图是单图模式下的柱状图显示，图中每根柱子代表单图模式下的对应图片的PM2.5的值，其中柱子的颜色依据PM2.5值的范围不同而呈现出不同的颜色，可以看出符合预期需求。

### 测试总结

本章详细测试了参与式感知平台的web可视化系统，验证系统已经达到了需求分析章节中的功能。

## 第六章 结束语

前面对参与式感知平台的数据处理与可视化系统的背景介绍、相关技术分析、概要设计、详细设计与实现以及系统测试的进行了全面介绍，本章作为全文的结束，将对这个课题进行总结。同时，对本课题工作中的不足以及下一步的改进工作进行阐述。最后，对作者在研究生阶段的科研工作进行整理说明。

### 全文总结

本论文主要是基于参与式感知平台的数据处理和web可视化研究，主要包含数据处理和可视化两个方面。参与式感知平台是指大量参与者利用手持带有传感器的移动智能手机，将采集到的感知数据和多媒体数据（图片、视频、录音等）上传到服务器端，服务器端将这些数据进行分析、整理、存储，并应用相应的算法，实现参与者轨迹分析预测，并通过一定的激励机制来激励参与者上传高质量高数量的感知数据，也激励更多的用户参与进来，并且最终在PC的浏览器端进行数据的可视化展示。

其中数据处理是平台的核心，参与式感知平台通过移动客户端传感器采集周围的传感数据和多媒体数据，并且上传至服务器，数据处理工作就是将这些纷繁复杂的数据进行整理、过滤和分析，存储至服务器数据库中，并且提供上层应用模块如轨迹分析预测、激励、web可视化等所需的网络接口以及API。

可视化是参与式感知的最终表象，可视化界面处于表示层，之间面向用户，直观的表现了参与式感知平台的存在的意义，也是至关重要的一个环节。Web可视化的目标就是以图形图像等可视化技术来展现纷繁复杂的感知数据，简单明了直观地展现数据的变化规律以及数据之间的隐含的关系。该平台的可视化基于百度地图展示这些POI的地理位置，基于这些POI展示其中的感知信息如图片展示，POI点的PM2.5浓度值的变化趋势，图片对应的PM2.5值的变化趋势等。随着地图的缩放和拖动，POI点以及与之关联的数据都会发生便会，展示的过程中用到的可视化技术主要有highchar中的柱状图以及折线图等，图片的单图和多图模式，以及POI位置点的聚合和POI的PM2.5值的聚合。

本论文的创新点有两点：

首先，引入开源的分布式Key/value数据库HBase，Hbase面向列存储并且提供数据的版本控制功能，大大提高了写性能，为高并发上传数据提供技术基础。Hbase还有更好的扩展性和可靠性，为参与式感知平台发展成一个云平台成为可能。

其次，在web可视化方面，PM站点空间聚合算法。PM空间聚合是指基于百度地图的PM站点的聚合，在地图的不同缩放级别下，实现PM站点的聚合。笔者应用的算法是结合百度地图的改进的基于网格的Kmeans算法。该算法的创新之处在于基于“网格”，网格的选取是基于百度地图的不同坐标体系，百度地图缩放级别概念有关。kmeans算法中k值的选择以及k个初始质心的计算。PM站点的PM值的聚合算法。PM站点的PM值的计算，是对PM值的拟合，指根据周围的PM值来计算需要拟合的PM站点的PM值。笔者现在应用的是基于距离的加权平均算法。这个算法的创新之处在于结合了百度地图的特点，以及北京监测点的信息。

### 6.2 不足以及下一步工作

* + - * 柱状图和图片展示之间的交互。

问题：为有更好的可视以及可用性，图片的展示与柱状图有一定的关联。功能上包括：在多图模式下，点击某一张图片，会在柱状图突出这个图片对应的柱状图的信息；点击柱状图的某一柱子会突出显示对应的图片。

方法：突出主体的方法很多，可以借鉴摄影构图突出主体的方法，包括对比衬托法、利用线条指引突出主体、利用形象的完整与否突出主体、重要位置法等等。其中比较适合项目中的方法是对比衬托法，对比衬托法又包括虚实对比、大小对比、明暗对比、色彩冷暖对比、动静对比等等。考虑到图片的边框以及柱状图都有颜色，可以采用虚实对比以及大小对比的方法，来突出显示主体。

技术：highchar.js的相关方法。

具体的设计方案:多图模式下，点击某张图片时，柱状图中对应的柱子可以实，其他柱子可以虚化。当点击某一柱子时，该柱子对应的图片变大高亮显示，以突出主体。

2 折线图展示。

问题：折线图主要展示各个PM站点在某段时间之内的PM值趋势。可以比较直观的展示某个PM站点在一段时间内的PM的趋势，以及不同PM站点在同一时间段内PM值的比较。可是，由于各个站点距离较近，PM值的差距较小，折线图中各个PM站点的PM值折线区分不明显，甚至出现介乎重合的现象。

方案：现在PM站点的PM值的计算方案是利用上文提到的基于距离的加权平均算法。对应上述问题，有两种方案。第一：改进算法，之前的公式是

可以改变权值的计算方法，例如可以加一个系数，关于系数的确定，需要后期做实验来确定。第二：改进折线图的单位跨度，这种方案比较简单直接。

3 展示的性能方面。

问题：性能方面主要问题表现在打开网页较慢，以及折线图展示中，改变时间段时，折线图转换较慢，影响用户体验。

原因：对PM站点的PM缺省值的拟合算法，即PM值聚合算法，其中其中关于权值的计算部分需要实时计算，比较影响展示速度。

方案：针对原因，解决方案有两种，第一：定时（比如选择凌晨）计算每个PM站点的每个时间点（比如每个小时）拟合PM值，并且存储到数据库。此种方法的优点是提高网页性能，提高用户体验性，缺点是每次计算量有点大，数据库也会增加很多数据。第二：定时（比如选择凌晨）计算每个PM站点对应的权值，并且存储到数据库，当改变时间段时，可以实时计算PM站点的拟合PM值，可以节省权值的计算，相对于第一种方案性能稍逊，优点在于，实时计算，数据库相对存储的数据量较小。

### 研究生期间的工作

#### 参加的项目

#### 完成的文档

参与式感知平台的数据处理与可视化的需求分析文档；

参与式感知平台的数据处理与可视化的总体设计；

参与式感知平台的数据处理与可视化的详细设计；

参与式感知平台的数据处理与可视化的测试报告。

## 参考文献

[1] 业界要闻 -《移动通信》- 2013-10-15

[2] 用数字说话 -《IT时代周刊》- 2013-12-05

[3] “人+智能终端”，强大传感器的强大未来\_悟网不欢 - 《网络（http://blog.sina.com）》

[4] 大数据可视化之美 谢然; -《互联网周刊》- 2014-06-05

[5] 基于大数据时代的数据可视化应用分析 周嫣然; -《网络安全技术与应用》- 2014-11-15

[6] 大数据在智慧城市中的应用 李光亚;张敬谊;童庆; -《微型电脑应用》- 2014-12-20

[7] 大数据时代下网络舆论宣传的挑战与机遇 施畅; -《科教导刊(下旬)》- 2015-01-25

[8] 大数据可视化之美 谢然; -《互联网周刊》- 2014-06-05

[9] 基于大数据时代的数据可视化应用分析 周嫣然; -《网络安全技术与应用》- 2014-11-15

[10] 大数据时代背景下的数据可视化应用研究 涂聪; -《电子制作》- 2013-03-15

[11] 大数据可视化之美 谢然; -《互联网周刊》- 2014-06-05

[13] 参与感知在气象信息服务系统中的应用研究 孙晓燕;杜景林;周杰; -《计算机应用与软件》- 2014-05-15

[14] 数据可视化技术研究及其在铁路中的应用 李仕旺;潘跃;吕晓艳; -《铁路计算机应用》- 2015-10-25

[15] 嵌入式电极侧向场激励FBAR若干问题的研究 冯斌 -《浙江大学博士论文》- 2013-04-01

[18] 分子可视化建模及其软件实现 谢欣荣 -《华中科技大学硕士论文》- 2004-04-01

[19] 关联规则可视化技术的研究及实现 易先卉 -《湖南大学硕士论文》- 2008-05-19

[20] 数据可视化技术研究及其在铁路中的应用 李仕旺;潘跃;吕晓艳; -《铁路计算机应用》- 2015-10-25

[21] 分子可视化建模及其软件实现实现 谢欣荣 -《华中科技大学硕士论文》- 2004-04-01

[22] 多波层位标定可视化软件系统研究 杨昕梅 -《成都理工大学硕士论文》- 2004-05-01

[23] 拱坝可视化CAD的关键技术研究与系统开发 黄程 -《河海大学硕士论文》- 2005-03-05

[24] 气力输送管路流型虚拟可视化技术研究 张道建 -《山东大学硕士论文》- 2007-04-26

[25] 基于安全域的电压稳定在线监视系统 汪映辉 -《天津大学硕士论文》- 2007-06-01

[26] 大地电磁测深二维数值模拟可视化 张波 -《中国地质大学（北京）硕士论文》- 2008-05-01

[27] 关联规则可视化技术的研究及实现 易先卉 -《湖南大学硕士论文》- 2008-05-19

[28] 基于GIS的农业水资源信息可视化高效管理 张小光;胡杰; -《中国科技信息》- 2011-04-01

[29] 基于WebGS的大数据可视化研究与优化I 贺群;杨明川; -《电信技术》- 2015-06-25

[30] 消息分发系统中运营子系统的设计与实现 张烁 -《北京邮电大学硕士论文》- 2012-12-22

[31] 国家乒乓球队训练比赛知识交互平台的构建与应用 徐赟;张辉; -《中国体育科技》- 2014-07-15

[32] 优秀开源Web开发资源大集合\_头上的星空 - 《网络（http://blog.sina.com）》

[33] 给开发者提供的 35 款 JavaScript 图形图表库 -- 开源中国 --传送门 - 《网络（<http://chuansong.me/>）

[34] 数字城市管理系统的研究与设计 李炜 -《西安电子科技大学硕士论文》- 2012-06-06

[35] 基于WebGIS的企业大气污染物查询分析系统的设计与实现陆谊 -《苏州大学》- 2013-05-01

[36] 基于Google Maps API的移动基站信息管理系统研究\_代写论文 - 《网络（http://www.lw001.com）》

[37] 基于Google Maps API的Web地图服务系统研究及应用 耿庆斋;缪纶;段媛媛;李江华; -《中国水利水电科学研究院学报》- 2009-03-15

[38] Google Maps二次开发 何洪磊;王芳; -《电脑编程技巧与维护》- 2010-05-03

[39] 基于LBS的城市智能泊车系统研究与设计 高柏俊 -《西安电子科技大学硕士论文》- 2011-02-01

[40] 基于Google Map API的控制点发布系统设计与实现 远顺立;高彦涛; -《科学技术与工程》- 2011-10-18

[41] 大数据时代下网络舆论宣传的挑战与机 施畅; - 《科教导刊(下旬)》- 2015-01-25

[42] MongoDB在气象传感器数据处理中的应白长清;刘敏; - 《软件》- 2015-11-15

[43] 嵌入式电极侧向场激励FBAR若干问题的研究冯斌(导师：王德苗;金浩) - 《浙江大学博士论文》- 2013-04-01

[44] 消息分发系统中运营子系统的设计与实现张烁(导师：金跃辉) - 《北京邮电大学硕士论文》- 2012-12-22

[45] 基于安全域的电压稳定在线监视系 汪映辉(导师：贾宏杰) - 《天津大学硕士论文》- 2007-06-01

[46] 的农特产品地理信息系统设计与实现 张倩;靳云通;罗勇;张洪吉; -《安徽农业科学》- 2015-01-30

0

[47] Http协议通信相关资料（做笔记） - baitxttgchx@163.com -博客频道 - CSDN.NET - 《网络（http://blog.csdn.net）

[48] 基于开源的网络地图的实现技术——以MapServer和Ajax结合为例 于书媛;奚砚涛;王一冒;牛坤;于雪涛; -《测绘与空间地理信息》- 2010-12-25

[49] 一种改进的K-means算法 张玉芳,毛嘉莉,熊忠阳 -《计算机应用》- 2003-08-28

[50] 基于小脑模型的二级倒立摆控制仿真 冯岩 -《昆明理工大学硕士论文》- 2009-02-01

[51] 一种基于粗糙集的K-Means聚类算法 冯征; -《计算机工程与应用》- 2006-07-11

[52] 基于声纹识别的身份认证技术研究 彭诗雅 -《南京航空航天大学硕士论文》- 2010-12-01

[53] K-means算法实现 - 继续微笑lsj - 博客频道 - CSDN.NET -《网络（<http://blog.csdn.net>）

[54] 百度地图API在小型地理信息系统中的应用 杜传明; -《测绘与空间地理信息》- 2011-04-25

[55] 百度麻点图实现方式(转)\_pgis的涟漪 - 《网络（http://blog.sina.com）》- （是否引证：否）

[56] 基于WebGIS的车辆实时位置手机查询系统研究 陈峰 -《南京信息工程大学硕士论文》- 2013-05-01

[57] 基于GPS数据的道路交通状况的研究 王晨 -《浙江工业大学硕士论文》- 2013-11-05

[58] Webmagic http://webmagic.io/, 2014.

.

## 致谢

毕业论文的暂告收尾，也意味着我的研究生生活即将结束，近两年的学习和科研工作不仅使我的只是结构和科研能力以及动手编程能力都上了一个新台阶，各方面的素质也得到了提高。

最后的这段时间就是一直努力撰写毕业论文，论文的写作是枯燥和艰辛的，又是富有挑战性的科研工作。首先，感谢我的导师龚老师，非常感谢他在我的教育、学术以及学习精神上的指导，让我的研究生生活更充实、受益匪浅。还有我的指导老师王文东老师和阙喜戎老师，王老师在这个参与式感知平台上给我们太多的指导，在这个漫长艰辛又充实难忘的过程，因为有王老师平时的谆谆教诲和不断的鼓舞，才有今天得科研成果。阙老师也是耐心热情地和我探讨论文中的一些工作，给了我很多的建议和意见。

我还要感谢实验室的师兄师姐，袁龙运师兄在web可视化方面对我进行指导和帮助，给我提供一些参考性资料，是我日后进行可视化研究工作时的重要支持。感谢李莹师姐，在Hbase数据库的学习上的指导，以及表结构设计的提点，他们的孜孜不倦的精神以及严谨认真负责的学习态度给我做了很好的榜样。

感谢我的同学和朋友，感谢我们协作感知组的小伙伴，王东升、寇秦丽、冯云等人，我们在项目中各有分工，互相合作和帮助，一起画框架图，一起讨论设计和需求，当遇到问题时，互帮互助，互相鼓励，一起学习一起科研，使我们在科研的艰辛的道路上不孤单，并且乐此不疲。

还要感谢我的学弟学妹，李帅师弟在PM2.5计算工作上的杰出表现以及王璐师等的认真勤奋好学的精神，让我们看到参与式感知平台的未来，相信平台的后继工作会有更出色的表现。

此外，我还要感谢我的室友和我的家人，特别是我的父母，感谢他们对我的无条件的支持，以及给予的爱与关怀，让我拥有战胜一切的力量。

最后，再次感谢诸位专家教授们在百忙之中评审的学位论文，并给予指导

# 攻读学位期间发表的学术论文和科研情况