**硕士学位论文**

(工程硕士)

基于一致性哈希算法的广告投放系统的设计与实现

**DESIGN AND IMPLEMENTATION OF ADVERTISING SYSTEM BASED ON CONSISTENT HASHING ALGORITHM**

**武振朝**

**哈尔滨工业大学**

**2015 年 6 月**

国内图书分类号：TP311 学校代码：10213

国际图书分类号：621.3 密级：公开

**工程硕士学位论文**

基于一致性哈希算法的广告投放系统的设计与实现

|  |  |  |
| --- | --- | --- |
| 硕士研究 Th | ： | 武振朝 |
| 导 师 | ： | 涂志莹 讲师 |
| 副 导 师 | ： | 施炳文 高级工程师 |
| 申 请 学 位 | ： | 工程硕士 |
| 学 科 | ： | 软件工程 |
| 所 在 单 位 | ： | 软件学院 |
| 答 辩 日 期 | ： | 2015 年 6 月 |
| 授予学位单位 | ： | 哈尔滨工业大学 |

Classified Index：TP311 U.D.C.: 621.3

Dissertation for the Master‟s Degree in Engineering

**DESIGN AND IMPLEMENTATION OF ADVERTISING SYSTEM BASED ON CONSISTENT HASHING ALGORITHM**

|  |  |
| --- | --- |
| **Candidate：** | Wu Zhenchao |
| **Supervisor：** | Lecturer Tu Zhiying |
| **Associate Supervisor:** | Senior Engineer Shi Bingwen |
| **Academic Degree Applied for：** | Master of Engineering |
| **Speciality：** | Software Engineering |
| **Affiliation：** | School of Software |
| **Date of Defence：** | June, 2015 |
| **Degree-Conferring-Institution：** | Harbin Institute of Technology |

摘 要

网络广告的主要优势在于时间和空间上，既不受时间的约束又不受空间的限制，而且制作网络广告的费用相对于传统广告而言，成本较为低廉。本课题研究的广告投放系统的广告类型主要是以图片和动画为主。

在广告投放的过程中，来自广告投放网站的访问量是巨大的。直接请求关系型数据库会造成极高并发量，导致数据库的访问速度非常缓慢，延迟达到秒级。由于传统关系型数据库存储在磁盘上，在高并发量的情况下，响应较慢，故本课题在存储广告时使用处理速度快的非关系型数库redis。

在广告投放环节之后，外部系统访问之前，把不同的广告数据放在不同的缓存数据中，由于传统的哈希映射会造成非均匀映射，故本课题采用一致性哈希算法解决非均匀映射的问题，使缓存资源均匀地被利用进而使资源的利用率达到最大化。本课题使用Flask框架体系设计、实现和完成广告投放系统所涉及到的模块的

后台代码编写。主要采用bootstrape和jinja2编写前台页面。关系数据库采用数据库抽象层SQLAlchemy，非关系型数据库用redis进行数据交互。采用该系统可以满足企业或者其他中小型广告中介企业用最低的成本投放广告的需求。基于一致性哈希算法去操作数据库，能够使所使用的缓存中的数据得到均匀存放，最大优化资源的配置。在本论文的测试部分，采用Pytest进行单元测试；并分别对订单管理和广告投放进行功能测试；在性能测试部分主要测试在多数线程并发的情况下，订单管理部分的系统响应时间；在系统的非功能性测试部分，主要进行浏览器兼容性测试。

关键词：广告投放系统； Flask 框架；一致性哈希算法； redis；均衡映射

Abstract

In terms of time and space, online advertising has great advantages. It is not restricted by time nor space. And also, the cost of producing online advertising is relatively lower than that of traditional advertising. The advertisements in this research are mainly advertised in the form of picture and animation.

In the process of advertising, the access quantity from advertising website is huge. Direct request for relational database may lead to high concurrency, slowing down the speed of access to the database, delaying even to second level. Because the traditional relational database is stored on disk, the respond to request is very slow in the case of high concurrency. So in this paper, it uses redis, non-relational data, to store advertisements.

After advertising and before the external system visit, different advertisements are stored in different cache databases. The traditional hash map usually cause non-uniform map. This thesis uses consistent hashing algorithm to solve this problem, fully utilizing caching resources as far as possible.

This paper uses the Flask framework to design, implement and complete the background code of modules involved in the advertising system. And it adopts bootstrape and jinja2 to make the web page at the front desk. SQLAlchemy helpes the system with the building of the relational database. Redis is used for data exchange in non-relational database. The advertising system can meet the needs of the enterprises and other medium or small advertising agencies to advertise at the lowest cost. The database is created based on the consistent hashing algorithm, which can uniformly store data in the cache database, optimizing the configuration of the resource. In the test section, the Pytest carries on the unit testing; and order management and advertising placement test function; in performance testing is to test in most concurrent threads, the order management system response time; in the system of non functional testing part, mainly for browser compatibility testing. In the test section, it takes Pytest to do unit testing and also do function testing for order management and advertising placement; in performance testing, it mainly test the response time of order management in many concurrent threads; in the non-functional testing, it mainly test the browser compatibility.

**Keywords**: Advertising System; Flask Frame; Consistent Hashing Algorithm; Redis; Equilibrium Mapping

目 录

[摘 要](#_Toc686437999) 3

[Abstract](#_Toc686438000) 4

[第1章 绪论](#_Toc686438001) 5

[1.1 课题来源和项目名称](#_Toc686438002) 5

[1.2 广告投放系统的国内外研究状况](#_Toc686438003) 5

[1.2.1 国外研究现状](#_Toc686438004) 5

[1.2.2 国内研究现状](#_Toc686438005) 5

[1.2.3 与课题相关的问题](#_Toc686438006) 6

[1.3 网络广告类型和投放策略相关概念](#_Toc686438007) 6

[1.4 本论文章节安排](#_Toc686438008) 6

[第2章 广告投放系统需求分析](#_Toc686438009) 6

[2.1 系统业务流程描述](#_Toc686438010) 6

[2.1.1 订单管理业务流程](#_Toc686438011) 7

[2.1.2 广告投放系统业务流程](#_Toc686438012) 9

[2.2 系统基本功能需求分析](#_Toc686438013) 9

[2.3 系统用例需求分析](#_Toc686438014) 9

[2.3.1 订单管理业务用例分析](#_Toc686438015) 9

[2.3.2 广告投放业务用例分析](#_Toc686438016) 10

[2.4 实体对象分析](#_Toc686438017) 10

[2.5 一致性哈希算法需求分析](#_Toc686438018) 11

[2.6 非功能性需求和性能需求](#_Toc686438019) 11

[2.7 相关技术及算法简介](#_Toc686438020) 11

[2.7.1](#_Toc686438021) **[Flask](#_Toc686438021)**[框架体系](#_Toc686438021) 11

**[2.7.2](#_Toc686438022)****[Flask-SQLalchemy](#_Toc686438022)** 11

**[2.7.3](#_Toc686438023)****[Bootstrap](#_Toc686438023)** 12

[2.7.4 一致性哈希算法简介](#_Toc686438024) 12

[2.8 本章小结](#_Toc686438025) 13

[第3章 广告投放系统总体设计](#_Toc686438026) 13

[3.1 广告投放系统功能概要设计](#_Toc686438027) 13

[3.2 广告投放系统的总体结构](#_Toc686438028) 13

[3.3 系统核心数据流图](#_Toc686438029) 13

[3.4 数据库设计](#_Toc686438030) 14

[3.4.1 关系型数据库设计](#_Toc686438031) 14

[3.4.2 缓存数据库中键值对的设计](#_Toc686438032) 27

[3.5 系统各模块设计](#_Toc686438033) 28

[3.5.1 订单管理模块设计](#_Toc686438034) 28

[3.5.2 数据回写模块设计](#_Toc686438035) 29

[3.5.3 管理投放模块设计](#_Toc686438036) 30

[3.6 本章小结](#_Toc686438037) 31

[第4章 广告投放系统的实现](#_Toc686438038) 31

[4.1 订单管理模块实现](#_Toc686438039) 31

[4.2 数据回写模块实现](#_Toc686438040) 41

[4.3 广告管理投放模块实现](#_Toc686438041) 43

[4.4 一致性哈希算法的实现](#_Toc686438042) 45

[4.5 本章小结](#_Toc686438043) 48

[第5章 广告投放系统的测试](#_Toc686438044) 48

[5.1 测试环境](#_Toc686438045) 48

[5.2 单元测试](#_Toc686438046) 49

[5.3 功能测试](#_Toc686438047) 50

[5.3.1 订单管理功能测试](#_Toc686438048) 50

[5.3.2 广告投放功能测试](#_Toc686438049) 52

[5.4 性能测试](#_Toc686438050) 54

[5.5 非功能性测试](#_Toc686438051) 57

[5.6 本章小结](#_Toc686438052) 58

[结论](#_Toc686438053) 58

[参考文献](#_Toc686438054) 58

[哈尔滨工业大学学位论文原创性声明及使用授权说明](#_Toc686438055) 59

[学位论文原创性声明](#_Toc686438056) 59

[个人简历](#_Toc686438057) 60

# 第1章 绪论

## 1.1 课题来源和项目名称

网络技术的飞速发展，为互联网广告插上了腾飞的翅膀，使互联网广告带来的经济收益越来越显著，成为互联网企业重要的盈利方式之一[1]。与报刊、广播等传统媒体相比，网络广告不再采用文字、图片、声音等呆板的形式，而是采用文本、声音、图片、动画、视频相结合的形式，以更丰富的形式呈现出来。另外互联网广告具有受众清晰、信息量大的特点，而博得广告主的青睐[2]。互联网广告平台正在逐渐取代传统的媒体广告，成为广告类型的主体，也成为互联网企业的重要投资方向和发展方向。

传统的互联网广告通常采用“遍地撒网”的投放方式，这种方式导致页面上出现形形色色的广告，往往会引起受众的反感。展示在用户面前的广告数量很多，但是收效甚微，有效性得不到提高。铺天盖地的广告有时还干扰了网民的访问和浏览[3]。因此，掌握受众群体的特点，根据特点进行精准广告投放，才能使网络广告拜托以往的投放模式，才能提高网络广告的有效性，才使广告主的利益最大化。

本课题来源于所在实习公司，根据其实际需求，旨在开发出一套符合其业务规则和企业现状的广告投放系统。

本项目名称为基于一致性基于哈希算法广告的投放系统的设计与实现。本系统分为两个部分订单部分和投放部分。这两个部分中的大部分主要由本人来设计实现并完善，一些设计企业策略的部分由同小组其他人员开发，在订单部分我们需要实现用户权限管理、Web界面的设计与实现、数据操作，设计并实现数据库，为一些比较重要的操作提供脚本。目前此系统已在公司上线，订单部分服务内部300多的销售团队，广告投放部分也已投入使用。

## 1.2 广告投放系统的国内外研究状况

广告投放系统在国外的发展相对于国内的发展相对成熟，随着国内互联网的发展，目前国内的网络广告投放系统在技术上和理论上都取得了很大的进步，但是整个行业的技术和理论基础相比国外还有待进步。

### 1.2.1 国外研究现状

国外的大多数广告投放系统是依托于网络搜索引擎和社交平台的[4]，该类广告投放系统首先建立用户行为数据库，其后进行大数据分析。这类的广告投放技术主要是对用户的上网行为做个性化的精确分析，按照广告主的需求锁定广告受众，提供多种渠道的投放，并按照其投放效果进行付费[5]。这种技术采用的追踪系统发展已经很成熟了，可以确保数据的准确性[6]，保证数据真实地反映用户的行为和广告意向，可以深度防止作弊，还可以追踪在范围内的各种数据。

目前一些国外广告投放系统的创建开始采用全新的视角，并且加入更多的数学方法[7]，这样设计的主要目的在于及时跟踪用于的需求变化，当下流量轨迹对目标产品或品牌发生即时性的概率，进而来决策广告是否曝光。以逆概率来进行广告曝光决策[8]，能够最大的降低广告曝光的浪费，并提高曝光后发生实质效益的机率。这种类型的广告投放技术现已被广泛应用于欧美国家的国土安全方面，并且影响着社会政策、财经政策的决策，根据大数据运算出各种事件的发生机率，针对机率最高事件进行预判性决策[9]，这样借助大数据和数学方法使预测变得更加准确。

国外的网络广告投放系统已经开始融合统计学、计算机科学、信息科学以及微观经济学等多种学科，多种学科相互交融[10]旨在产生更加科学高效的广告投放系统。在广告投放系统的研究方面，能够实现语境、广告和受众三者的最佳匹配[11]，已经成为待实现的目标。研究的重点是解决用户与广告匹配的相关性，以及广告投放系统最佳的竞价模型。计算广告学涵盖了数据挖掘、销售、营销管理、语言处理等多领域的知识。

### 1.2.2 国内研究现状

我国网络广告产业的增长规模很大程度上取决于外生性增长，在网络广告理念、模式等方面的创新并不明显。

国内的大多数广告投放系统，很难进行流量管理[12]，所以用户的体验很差，不能给用户带来满意的投放效果。国内的百度联盟依托于百度搜索引擎，腾讯依托于巨大的用户数量，由于这些比较成熟的广告投放系统数量比较少，造成了广告投放系统的门槛相对较高，难以满足众多的广告投放者的需求[13]。

目前国内的广告系统主要基于两类技术：

第一类广告投放系统的技术主要进行受众人群分析[14]，依托于搜索引擎针对这些受众行为和用户在网络中的一些搜索习惯和浏览习惯进行行为分析。根

据百度自己的搜索排名技术选择适当的投放广告的网站[15]。然后广告进行正式的投放以及实时的检测优化。

第二类广告投放系统的技术依托于本身产品的庞大的用户库[16]，通过使用旗下的各种产品去判断用户的类型。并根据用户自己描述，个人的基本资料，用户设置的标签及其他用户为自己填写的资料[17]。用这些数据去做基础数据和上面的数据进行分析。获取日常用户的操作行为，判断出用户的倾向[18]。

从以上的内容可以看出国内外在广告投放系统的研究方面，大多是倾向于基于文本分析和基于用户行为分析的[19]。基于文本分析的广告系统主要是通过对获取的网页进行分析，并且结合文本的相似性分析，做出判断[20]。

国内的搜索引擎公司竞价排名的机制是通过对用户搜索记录的关键字和广告投放者要竞拍的关键字进行相似性计算[21]，实现在搜索结果的最前面插入和用户搜索相关的广告[22]。目前国外某些著名搜索引擎为了保证搜索结果的公平性，较早的放弃了竞价排名广告的策略[23]。

基于用户参数分析的广告投放系统可以为广告拥有者、广告投放公司和受众搭建一个平台[24]。该平台采用经济利益分成的方式让网民做为志愿者参与到广告创作和投放的环节中来。受众在浏览网页时，发现商品的信息，在商品信息所对应的位置上插入广告兴趣点，并附属链接。与之前的广告投放效果不同的是，其智慧在于用户对广告兴趣点有意向时，才会显示广告。

### 1.2.3 与课题相关的问题

与国内外广告投放系统相比，对于类似于百度大规模投放广告的系统，因为算法或策略的原因，会导致部分广告对投放效果不满意[25]。本课题所研究的系统能使广告投放的内容更加满足广告主的意愿，达到最好的广告投放效果。本系统能够直接选择广告主需要投放的网站或某网站的位置。

在本课题的广告投放系统中，在请求发送的过程里，广告存储在集群的缓存数据库中，对于高并发的外部请求，很容易使某台缓存宕机[26]，或是为了应对高并发的请求去增加一台机器。使用普通的hash方法的一般步骤是[27]：首先把数据映射到具体的节点上，采用key%N的方式实现映射[28]，key表示代码数据的广告代码的ID，N代表机器的节点数。如果出现从集群中增加或删除机器的情况，则所有的数据映射都无效。而且此哈希映射方法有一个严重缺陷[29]，比如某大型互联网企业的一款聊天交友软件，在存储有关数据时直接使用其ID序列，其ID序列为数字构成，在进行哈希取余时，某位特定数字结尾的数量特别多，造成数据库挂掉。其解决方案为再加一台数据，进行分库。这种做法并

非不可取，但并不是最优解决方案。

所以本课题引入一致性哈希算法，去解决广告信息在多个缓存数据库中存储不均匀以及增加或减少缓存节点时，能够保持原来数据库中广告代码的读取不受影响的问题。

本课题的主要研究内容是：

一方面是如何去解决面对高并发的外部广告请求，能够快捷的进行投放广告的响应，又能够使公司内部的缓存资源得到充分利用，在缓存出现问题之后，能够有比较好，不需要更多人力维护的容灾机制。

另一方面是如何做到规划提炼出广告中介公司的需求，开发出订单管理及权限审批流程，根据投放广告主的不同要求使用不同的投放策略，进行广告投放。

本系统的目标是开发出一套web广告投放系统（B/S结构），既要在系统内部控制缓存数据库的的选取和操作，还要构建开发订单管理和广告投放，主要包括选取投放策略，由投放模块投放广告，并在缓存数据库中实时计算广告数据。订单管理模块经由已有的合同管理系统提供的数据，制定详细的符合客户及系统要求的订单系统。广告投放模块要能够支持远程请求的广告请求，并返回广告，并在缓存中和数据库中及时计数。

## 1.3 网络广告类型和投放策略相关概念

网络广告的类型主要有文本广告和多媒体广告，本课题主要针对多媒体类型的广告，该多媒体广告主要采用图像和动画的形式。本课题实现的系统主要的受众媒体一般为访问量较大的门户网站或者某些知名网站的首页。在本系统中一般投放品牌广告，这些品牌广告的广告主也是一些知名品牌，广告主投放广告旨在宣传产品和提升公司的影响力。

CPC（Cost Per Click）用来表示广告单次点击的费用[30]，是一种广告服务商和开发者与广告主之间的一种计费方式，移动端软件使用者每点击一次广告主就要向广告平台提供者支付相应的费用。

CPM (Cost Per Thousand)与CPC相似，也是一种计费模式，CPM是广告显示1000次所需支付的费用[31]。

CTR (Click Through Rate)表示点击率，它的计算方法是：广告的目标人群点击广告的次数除以广告的展示次数。一般点击率越高说明受众的关注度越高，则说明广告的投放效果越好。因此CTR是衡量广告投放效果的一项很重要的指标。

本系统网络广告类型通常属于品牌广告，故计价模式大部分属于CPC 和

CPM。

配送：配送是根据购买量额外免费赠送的量，可包括CPC和CPM。

补余：补余是当投放量不符合CTR数据和预估量时所补充的量，可包括CPC

和CPM数据[32]。

## 1.4 本论文章节安排

在第1章中，首先论述选题背景及项目来源，其后通过分析国内外的研究状况来认识本课题存在的主要问题和难点，并介绍网络广告类型和常见投放策略，最后介绍本课题的主要研究内容。

在第2章中，将对本系统进行业务流程分析和基本需求分析及系统功能和非功能分析。本章内容中系统业务流程描述和系统基本功能需求分析两部分内容主要通过流程图的方式来呈现。通过系统用例需求分析介绍订单管理业务和广告投放业务。其后介绍实现广告投放系统用到的相关技术，并简要介绍一致性哈希算法。

在第3章中，介绍了广告投放系统的总体设计，进行了系统的概要设计和总体结构。并给出了系统核心数据流图，对系统中的数据库进行设计。对于一致性哈希算法具体的实现方法在本章做出介绍。最后介绍系统主要模块的设计，主要有订单管理模块、数据回写模块设计和管理投放模块设计。

在第4章中对系统进行了实现，并分模块进行了实现描述。主要包括订单管理模块实现、数据回写模块实现和广告管理投放模块的实现。最后在系统中实现一致性哈希算法。

在第5章中，对本系统进行了测试，测试效果以及数据的准确性，包括功能性测试和非功能性测试。

# 第2章 广告投放系统需求分析

## 2.1 系统业务流程描述

从所在实习公司的角度来看，从与广告投放者洽谈到真正的进行广告投放需要进行一系列活动。下面就是整个系统的业务流程的描述。

（1）广告中介者即本系统使用者首先与各大网络媒体，购买到广告位，得到广告位的基本属性[24]。比如：首页或者二级页面，侧栏或底部，广告位的尺寸。

（2）当销售与广告主或者代理商签订合同后，导入到订单系统中创建新的订单，并且设置广告的基本信息，例如广告的名称、广告的类别。并且需要选取广告的尺寸，设置广告的定向信息，一般为广告的目标网站。

（3）销售提出要基于订单申请，经过订单审批流程到达投放的阶段。

（4）媒体管理人员再次确认广告素材，并生成HTML代码提交给投放人员。

（5）投放人员投放此广告。

（6）当媒体的受众打开广告投放页面时，会通过加载JS代码向网站的服务器发出请求。

（7）当服务器的端口接收到页面发来的广告请求消息，经过广告投放管理部分的处理，将广告推送出去。

（8）广告中介提供数据给广告主验收，广告主会跟第三方平台做比对。

### 2.1.1 订单管理业务流程

将整个系统的业务流程进行模块化划分，可以抽离出订单管理模块，进而需要明确订单管理业务的流程，订单管理业务的流程如图2-1所示。

广告基础信息准备阶段，需要进行基础数据和信息的准备。大致包括合同号，投放数量，投放广告位置意向，投放期限，和投放素材。这个阶段属于创建订单的准备阶段，一般不在线上进行。

下订单准备阶段：在拿到广告基础信息准备阶段的线下数据之后，就可以输入基本信息，创建完成订单。在创建完成订单之后，就可以安排投放，需要完成审批流程，如通过审批，便要上传广告素材，检验广告素材的合法性，如广告素材通过合法性验证，便到达开启投放的阶段。订单准备阶段业务流程如图2-2所示。



开始

结束

广告基础信息准备阶段

订单准备阶段

批准投放阶段

订单维护阶段

图2-1 订单管理业务流程图

不通过



开始

通过

一级审核通过

二级审核

通过

投放检查

通过

结束

填写订单信息

安排投放

上传广告素材

不通过

不通过

图2-2 订单准备阶段业务流程图

批准投放阶段：在投放广告之后，会对广告进行期望检查，主要包括CPC，

CPM, CTR数据的检验是否达到广告主的要求，如果达不到要求，就会进行补余和配送，以便进行正常投放。详细流程如图2-3所示：



开始

不通过

期望检查

通过

结束

创建订单

开启投放

继续投放

修改订单量

图2-3 批准投放阶段业务流程图



开始

不通过

申请修改订单

（归档状态）检查订单状态

终 止 投放广告

通过不通过

（正常状态）

订单状态审核

（订单及状态未改变）

通过

继续投放广告

结束

图2-4 订单维护阶段业务流程图

在订单维护阶段，对于已存在的订单，会进行订单的状态维护，在某些极端情况下，会进行订单数据的修改，订单修改之后，会进行订单的状态检验，然后会再次进行审批，主要防止恶意修改订单，扰乱整个流程。详细流程如图2-4所示。

预下单审批未通过

下单审批



销售提交订单

预下单

预

预投放

审批通过

投放

预

预投放审批未通过

投放审批通过

图2-5 广告订单审批业务状态图

从以上内容可以看出订单管理的业务从广告基础信息准备一直到订单维护阶段，涵盖了整个广告投放的生命周期。在整个订单管理中，订单状态的变迁是非常重要的一部分，进入预下单状态的订单需要经过审批，才可以进行进入预投放，而经过预投放的订单，必须再次审批才可以进入投放状态。在预投放状态的订单必须经过审批才能达到投放状态的订单。用状态图来表示整个订单的变化，广告订单审批状态图如图2-5所示。

### 2.1.2 广告投放系统业务流程

在本系统中，订单管理部分是不可或缺的一部分，在订单业务梳理完成之后，便是本系统的重点部分，其业务部分相对简单，用软件系统实现有一定的

技术含量。广告投放业务的基本流程图如图2-6所示。

进行广告投放时，首先初始化订单，之后将广告上传，上传后的广告自动生成代码，代码生成后就开启投放，即广告的投放业务结束。



开始

结束

初始化订单

上传广告

生成代码

开启投放

图2-6 广告投放业务流程图

本系统最终投放的广告最终展示在页面上，当页面加载的时候，会解析内嵌的JS代码，根据数据会请求相应的在投放状态的广告。

在2.1节中首先描述了广告投放系统的业务流程，根据上述的描述抽离出订单管理业务流程和广告投放系统业务流程。通过介绍订单管理业务流程不仅明确了该部分要实现的主要功能也使订单管理业务部分的流程清晰化。在广告投放业务流程部分的介绍中，采用流程图的形式展现了广告投放的执行步骤。

## 2.2 系统基本功能需求分析

本广告投放系统就是方便广告中介者能够进行自动化广告业务管理的系统，最大程度提高工作人员的工作效率。由广告投放系统的系统业务流程可以得到整个系统的完整的功能需求。以下将完整功能需求分块来阐述。

（1）订单业务主要包括订单列表、订单项、创建订单、修改订单、删除订单、预下单情况列表等功能，其中订单信息数据的一部分信息由合同管理系统

（不在本系统中实现）提供，其它的在本系统中生成。在创建订单时，要将合

同的信息作为订单的基本信息导入到系统中。本部分主要处理来自于销售和媒体的业务，基本数据的录入和制定订单必备的数据。

（2）库存信息：该部分功能主要有查询库存和修改库存信息。查看库存功能可以查看所购买全部广告位置的利用率，提供视图形式的效果展示。修改库存用于特殊情况，紧急增加投放广告。本部分主要处理广告中介者手中的资源展示情况，以及特殊情况下，对库存的特殊修改处理。

（3）广告资源：主要实现媒体列表、广告单元列表、展示位置类表的展示等功能。广告位基础信息录入，新增媒体，新增媒体属性，新增广告单元，新建展示位置。本部分主要处理媒体与广告位的拥有者间的业务。

（4）客户代理：主要包括客户列表，新建客户，合同甲方，新建合同甲方，代理设置。本部分主要为合同系统的对接以及广告主（客户）间的业务。

（5）用户：主要包括团队和用户，新建团队，新建用户，团队控制。本部分为了跟风险控制系统和策划系统对接建立。

（6）投放控制：主要包括：开启投放，暂停投放，终止投放，投放内容审查。本部分主要对已下单状态的订单（准备投放的订单）的广告的投放有着直接的控制。

（7）日志分析：主要包括对每天的日志进行分析，并与缓存中数据做偏移计算，即提供精确访问CPC和CPM数据，又为实时计算做技术参考。本部分主要用于统计和计算前一天系统中CPC和CPM数据，并且将统计结果更新到数据库中，用于投放策略的参考和投放参数的设置。

（8）数据回写模块：在系统中实时地计算和操作CPC和CPM、CTR数据，并且确保这些数据的准确性，本部分是由于要实时控制订单是否已完成广告主要求而建立，结合到今天零点前的日志分析数据，做实时统计。

## 2.3 系统用例需求分析

### 2.3.1 订单管理业务用例分析

（1）广告资源：在进行广告资源管理时，媒体管理员首先登入广告投放系统，从展示位置、投放媒体、广告素材和广告尺寸等方面做出管理。前置条件：已经购买广告位，并取得一定广告位的详细信息。广告资源角色用例图如图2-7所示。

（2）订单业务：媒体管理员登陆广告投放系统后，可以进行订单业务管理，主要包括富媒体管理，客户管理和合同管理。从销售与客户签订的合同中可以

获取合同的状态。完成投放状态的订单需要广告资源的数据。订单业务的角色用例图如图2-8所示。



展示位置管理

投放媒体管理

媒体管理员

广告素材管理



广告尺寸管理

图2-7 广告资源用例图



富媒体管理

客户管理

合同管理

媒体管理员

图2-8 订单业务用例图



二级销售人员一级销售人员



库存查询

销售

库存修改



投放控制人员

图2-9 库存用例图

（3）库存：销售，一级销售主管，二级销售主管，和投放控制人员登陆系

统之后，可以查看当前（和之前）的库存。投放控制人员可以在紧急投放广告是，修改当前库存状态。库存的角色用例图如图2-9所示。

（4）客户，代理和用户，团队：系统管理员登陆系统时候，可以进行客户设置，代理管理（销售代理），团队控制管理（用户所属团队权限），用户管理

（包含用户权限控制）。客户，代理和客户，团队的角色用例图如图2-10所示。



客户设置

代理管理

系统管理员

团队控制管理

用户管理

图2-10 客户代理和用户团队角色用例图

### 2.3.2 广告投放业务用例分析

（1）投放控制：投放控制人员登陆系统之后可以修改库存（如图2-9所示），开启投放，暂停投放，终止投放。前置条件：待投放的订单已经准备好之后，才可以开启投放订单。当某个订单的CTR, CPC, CPM等数据不符合要求时，系统将终止投放订单。终止投放的情况为与广告主终止合同等[13]。投放控制角色用例图如图2-11所示。

（2）日志分析和数据回写：当投放控制人员登陆系统之后，可以进行分析日志管理，也可分析缓存数据。分析数据之后可以提供数据给风险控制团队，风险控制团队会根据提供的各种数据告知投放控制人员对于订单的投放控制动作。日志分析和数据回写角色用例图如图2-11所示。

在2.3节中，主要进行了订单管理业务用例分析和广告投放业务用例分析。在订单管理业务用例分析中，分别介绍了广告资源用例图、订单业务用例图、库存用例图、客户代理和用户团队用例，介绍了各个部分中媒体管理员和各级销售人员需要完成的职责和操作。在广告投放业务用例分析中，介绍了投放控制的依据，日志分析和数据回写的作用，也明确了投放控制人员的主要职责。



分析日志管理



开启投放广告管理



分析缓存数据

投放控制人员

终止投放管理



暂停投放管理



图2-11 投放控制用例图



## 2.4 实体对象分析



销售渠道

客户名称

合同号

执行

客户组

1

m

产生

策划

设计

状态

m

名字

描述

折扣

拥有

尺寸

预估

广告单元

1

投放控制

所属媒体

m

外边

投放量

拥有

状态

资源

1

广告素材

1

m

广告标志

拥有

单价

投放类别

投放策略

订单项

订单

客户

展示位置

图2-12 E-R实体图

在需求分析论述中，分析实体关系的结构对系统的数据模型有重要的意义，从使用系统的角度看待数据及处理要求和约束关系[14]，提出了一个用于反映系

统使用者的观点的概念模式。为了降低设计的复杂程度，可以将概念设计从系统总体的设计和实现中抽离出来，这样可以使各阶段待实现的任务相对单一化，也更方便进行数据库的设计。E-R图以关系图的形式描述了系统各个组成部分以及它们之间的数量关系与依赖关系，本论文中主要的E-R实体图如图2-12所示。

客户的管理方面主要用于管理客户及客户组的基本信息。根据客户可以一对多地创建订单，并对订单进行基本信息的设置，订单的基本信息主要包括：该订单正式的合同号、该订单的销售渠道的说明、该订单的折扣，以及该订单如何设计、如何执行和订单的策划方案。

订单项主要用于管理广告素材，设定投放量，并对投放进行控制的。

展示位置部分主要对广告的具体投放做出控制，包括广告的标志、单价及状态，还涉及到广告具体的投放类别和投放策略，直接关系着广告的投放效果。

广告单元主要控制着广告的具体信息，例如广告的尺寸大小、名字、描述等信息。

## 2.5 一致性哈希算法需求分析

在广告代码信息存储到集群的缓存数据库的过程中，本课题需要一种能够有强碰撞性的散列函数来保证所有散列到缓存数据库的广告代码ID的不同。

这样可以保证缓存数据库中的所有容量被均匀地使用，避免数据库的某些部分负载过高。

由于MD5自身具有很强的抗碰撞特性，所有想要找到具有相同的MD5值但是数据本身不相同的数据，是非常困难的，因此在此处采用MD5函数进行散列。

## 2.6 非功能性需求和性能需求

#### （1）非功能性需求分析

系统对CPC和CPM数据量及CTR数据进行的实时处理、计算直接决定着本人所实现系统和广告主的商业利益，故应该保证一定程度的精确性。本系统的数据直接影响着用户，也决定了广告使用者能够获取收益的多少，也决定了对三方能否对该系统给出较高额评价。所以对系统有较高的非功能需求。

可用性：开发的系统是否能够让客户通他们可用的渠道进行访问和使用，这是项目的关键。

准确性：能否准确地处理系统中的数据，主要确保包括CPC、CPM、CTR等数据的准确性[30]，只有保证这些数据的准确性才能让广告投放系统以正确的投递方式进行广告的投放，进而受到很好的效果。

鲁棒性：对第三方请求的数据有较强的鲁棒性，对其中的格式不正确的或者内容（如时间）不符合规则的进行自动处理，并且对经由第三方平台的请求有合法或非法的判断[31]。

可扩展性：本系统要符合软件工程设计相关概念、要实现“高内聚、低耦合”的结构。对项目后期的敏捷或迭代开发有着重要的作用。

易修改性：系统内的重要的部分均有相应的配置文件，如果需要修改某部分相关属性，比如数据库的端口号或者账号密码等系统属性[32]，直接修改配置文件即可轻松实现。

易维护性：本系统的重要步骤和敏感操作都有日志记录[33]，并釆用了邮件报警的方式，当遇到重要问题可及时发送邮件报警。

针对软件产品的技术适应性表现为软件产品的一种应变能力[34]，这种应变能力注重的是，当软件产品运行的条件，例如硬件平台、网络条件等，发生变化时，该产品依然能够准确无误地运行。

#### （2）性能需求分析

根据国际标准3-5-8原则，且本课题的订单系统只对内部员工进行开放。所以其性能指标的时间特性为：

订单访问登陆时间最长不超过3秒，页面之间跳转要在3s以内。其内部有一个300到400的销售团队，所以订单部分系统并发量要大于等于300以上。其广告投放部分每台服务器的并发量要在3000+，其计算原则为每个广告主

投放的广告一般会在4-6个网页上，依据于某较稳定的投放媒体年PV量在1.6

亿，通过计算得出每台服务器得并发量在3000。

对于广告的访问影响时间应在2s以内为最优，计算方法为3-5-8原则与滚落延迟的消耗。

## 2.7 相关技术及算法简介

### 2.7.1 **Flask**框架体系

Flask属于一个微型的Web框架[25]，这个框架是采用Python编写的。这里的“微型”并不是说Flask 框架在功能存在缺陷，这里的“微型”只是突出了

Flask框架的代码非常简洁。

Flask框架在代码方面的简洁并不代表该框架在功能方面存在不足，因为该框架是易于扩展的，并且该框架很容易做出修改，这样使用者就可以根据系统实际的需要，选择合适的数据库，甚至可以自主地转换模板引擎。一般情况下，数据库抽象层并不包含在Flask框架中，Flask通过扩展可以添加类似的功能。

在使用Flask框架的时候，伴随着Flask项目的搭建，将会有大量的可供使用的扩展在开发环境中集成出来。并且这些扩展在能够兼容高级版本，在未来的版本中仍然可以无障碍地使用。

代码库的不断增长能够帮助使用者在设计项目时做出正确恰当的决定。

Flask会根据Python的情况为应用开发提供一个简单易实现的粘合层，帮助使用者在多种数据库工具中实现高级模式。Flask框架中可以提供定制化的服务，使用者可以根据项目的实际情况继承Flask中的类。

Django, Flask, Pyramid, Tornado, Bottle, Diesel, Pecan, Falcon等等都是用

Python实现的Web框架。Pyramid和Django都是面向大型应用的框架，Flask则是备受小型应用青睐的框架。Flask 简洁与轻巧，自定义程度高。相比

django 更加轻量级[26]。

### **2.7.2** **Flask-SQLalchemy**

Flask框架虽然具有诸多优点，但是该框架仍然存在一个问题：Flask框架中没有集成原生的对象关系映射器（Object Relational Mappers, ORM），这就需要使用第三方的库来解决这个问题。在众多的第三方库中，SQLALchemy库可以轻松实现原生的对象映射器（ORM）。

Web框架在ORM方面一般会有两种处理：Django框架将自主开发的ORM

直接集成到框架中[27]，这种方法一般采用一种独立栈的方法，这种直接集成

ORM的方式主要的优点是能够提供非常连贯顺畅的体验。对于其他框架，例如

Pylons，这类框架采用第三方组件来添加ORM，这类框架一般通过增加基于组件架构可结合的第三方组件来实现，这种方式的优点是给设计增加了很大的灵活性。

SQLAlchemy是一个单独的项目，该项目是由Mike Bayer带领的一个核心团队潜心研究开发出来的。在Flask框架中使用单独的ORM的一个优势就是：开发人员可以充分地考虑自己的数据模型，并且可以通过命令行工具、GUI框架和Web框架等方式可视化数据，这种方式无疑为项目的开发提供了便利。

SQLAlchemy是一个正在发展中的项目，API在持续地更新中。SQLAlchemy的主要优势是：围绕SQLAlchemy的功能，开发团队为它建立了丰富的社区，

丰富和完善了SQLAlchemy 项目。SQLAlchemy 的扩展和插件主要包括：

Declarative、Migrate、Elixir、SQLSoup、django-sqlalchemy、DBSprockets 、

FormAlchemy 和z3c. sqlalchemy. SQLAlchemy 未来的发展方向是兼 容

SQLserver等众多数据库。

### **2.7.3** **Bootstrap**

Bootstrap是由twitter的设计师开发的Web前端的CSS框架。Bootstrap具有管理能力并且可以提高内部分析。Bootstrap是开源的，得到了广大工程师的关注，很多工程师无偿地为Bootstrap的发展贡献了智慧和汗水，所以Bootstrap的更新和发展是非常快的。

Bootstrap的核心是基于Less构建的CSS, Less是一个动态的CSS语言框架，Less用于扩展CSS的动态特性，Less为CSS提供了更加完善的功能和灵活性。通过使用Less，使用者可以轻松地在CSS中使用嵌入式申明、运算和颜色编辑功能函数。

Bootstrap的优势主要有：

（1）由于Bootstrap是利用LESS来编写，很多样式和设计的部分都已完成，所有里面有很多已经写好的代码片段。Web开发者可以减少大量的时间，更加专注于本身的工作上，只需把成型的代码粘贴到合适的位置就好。

（2）在Web开发者进行页面的布局时，可以使用Bootstrape平台的布局，这样能够使开发者减少其开发前端工作的难度[24]，能够让一个后端人员已最快的速度找到自己的功能。关于布局，其开发出了各种各样的样式，开发人员只要把自己编写的代码放到想要放的位置即可。

（3）Bootstrap开放源代码，允许开发人员根据工作的需要，进行修改源代码，封装成自己的特征的框架。它还提供了JavaScript库，使用这个库能够超越基本的样式和架构方案。通过它，可以简便地操作其组件[25]。它提供的组件也多样化包括按钮，日期插件，提示条等等。

（4）Bootstrap有很好的兼容性，是前端人员不用在费时费力的调整不同浏览器的显示效果。在safari，chrome，Ie中，有着统一的效果[28]。

（5）文档能为初学者提供很大的帮助，是一本不可或缺的参考书。Bootstrap团队提供了丰富的文档共广大学习爱好者来参考。

### 2.7.4 一致性哈希算法简介

麻省理工大学于1997年提出了一致性哈希算法，该算法旨在解决Internet中存在的热点(Hot spot)问题[29]，起初一致性哈希算法与CARP（Common Access Redundancy Protocol, 公用地址冗余协议）很相似。简单的哈希算法虽然有很多优点，但是也存在着很多缺点，一致性哈希算法弥补了简单哈希算法的缺点，使DHT（Distributed Hash Table，分布式哈希表）在P2P环境中的应用和发展。

一致性哈希算法的工作原理可以分为两步：

第一步：计算存储节点的哈希值，将完整的存储空间抽象成一个圆环，然后将存储节点分配到环上，圆环上的每个节点都对应一个值。一致性哈希算法示意图如图2-13所示。

节点G

计算数据键值的哈希值

计算节点的哈希值

节点A

节点F

节点E

节点B

节点C

节点D



图2-13 一致性哈希算法示意图

第二步：采用哈希算法运算数据，将数据映射到最近的节点上去，一般采用顺时针的方向映射。当出现某个节点出现故障需要离线的状况时，根据环上的数据分布可知，受影响的数据仅仅是该节点开始到下一个节点之间的数据，并不会造成太大的影响。当有节点出现删除和增加时，整个存储空间上的数据不需要全部重新映射。

## 2.8 本章小结

在本章中首先介绍了广告投放业务的流程，根据总体的流程抽离出订单管理业务流程和广告投放业务流程。订单管理业务流程主要包括订单准备阶段的流程、批准投放阶段流程、订单维护阶段的流程和广告订单审批业务的流程。各部分的叙述均采用流程图的形式直观地呈现出来。在广告投放业务流程中，首先初始化订单，其后上传广告、生成代码，最后开启投放。

然后进行了系统基本功能需求的分析，功能需求主要包括订单业务需求、库存信息呈现、广告资源管理、客户用户管理和关高投放控制等。根据系统人功能需求给出系统用例需求分析，形象地对每部分做出说明。

其后给出系统的E-R图，以关系图的形式描述了系统各个组成部分以及它们之间的数量和依赖关系。

其后对一致性哈希算法的需求做出了分析，在文章最后分析了本系统的非功能性需求，确保系统的可用性、可维护性等。

# 第3章 广告投放系统总体设计

## 3.1 广告投放系统功能概要设计

广告投放系统主要的功能模块主要有：订单基本管理、投放广告管理和数据回写管理。订单基本管理模块主要实现订单的审批、创建和管理；投放广告管理模块主要负责广告的验证、生成代码及推送广告；数据回写管理模块主要负责系统数据的实时计算和订正。广告投放系统具体的功能组成如图3-1所示。

在订单基本管理的模块下主要包括：预下订单详情，创建订单，订单项管理，订单审批管理。预下单详情，可以查看销售和广告主拟建立订单的情况跟踪。订单审批管理主要用于审批订单是否合法，只有合法的订单才可以进行下一步操作。



广告投放系统

订单基本管理

投放广告管理

数据回写管理

预下订单详情

创建订单

订单项管理

订单审批管理

验证广告

生成广告代码

推送合法订单

实时计算数据

数据订正

图3-1 具体系统各功能组成

销售可以通过创建订单，去和已经签订合同的客户去建立订单（此订单亦被称为客户订单），订单项用于查看或增加和删除客户下所对应的多项订单。订单详情可以查看某订单当前的状态。

投放广告管理模块主要负责广告的投放，是整个系统中的关键模块，主要由验证广告、生成广告代码和推送合法订单组成。验证广告主要用于检验广告是否合法，是否符合系统的要求。通过验证的广告要经过生广告代码功能，将广告素材生成HTML代码，之后将代码交给投放人员，进行广告的投放。

广告投放管理主要包括接口开放和调整[22]，一级定时的从缓存中拿到广告的相关信息。当页面请求广告时，处理此请求的逻辑就在此模块。包括投放策略中的内容处理，及返回广告的设定都在此处理[23]。

数据回写管理模块主要用于处理广告投放系统的数据，该模块要实时地计算数据，将计算结果反馈到系统中，如果出现错误，该模块还要及时订正数据，保证广告投放系统能够准确正常地运转。

## 3.2 广告投放系统的总体结构

本广告投放系统的总体结构图如图3-2所示，从分层的角度来划分主要可以分为：展现层、应用层、基础服务层、数据库。



web层

展现层

广告投放管理系统



应用层

订单基本管理

预下订单详情订单创建

订单项管理 订单审批管理

投放广告管理

验证广告

生成广告代码推送合法订单

数据回写管理

实时计算数据数据订正



基础

服务层

任务管理

日志管理

授权管理

广告代码管理

数据管理

数据库

缓存数据库

图3-2 广告投放系统总体结构图

应用层主要有订单基本管理、投放广告管理和数据回写管理三个模块。

订单基本管理模块：从销售签订合同到广告推送之前所有的步骤，均在此实现。投放广告管理：从验证广告、生成广告代码到推送广告合法订单。数据回写管理：主要实时计算数据和及时订正出错的数据。

基础服务层主要控制系统的任务管理、日志管理、授权管理、广告代码管理和数据管理。任务管理：主要负责管理一些系统的定时任务。日志分析：定时从缓存数据库中导出数据库的任务。

数据库主要有一般数据库和缓存数据库组成，主要负责系统的数据存储和实时的数据计算，以及及时订正错误的数据。缓存数据库：存放准备推送广告信息，和广告订单的实时投放量。

## 3.3 系统核心数据流图



4.查找数据

5.返回数据

2.请求

8.广告应答

9.更新数据

6.应答

10.更新数据

1.广告请求

7.记录日志

3.记录日志

11.更新数据

关系数据库

广告素材服务器

缓存数据库

广告投放服务器

负载均衡

Web服务器

日志服务器

在广告投放系统中，核心的数据流图可以分为订单部分的数据流图和广告投放部分的数据流图两部分。其中订单部分生成的广告投放HTML代码存放在缓存数据库中，其他的都存在数据库中。因此在这里只需要清晰地描述广告投放部分的数据流向，广告投放的数据流图如图3-3所示。



外部系统

图3-3 广告投放的核心数据流图

广告投放页面有广告请求时，会首先经过web服务器，然后再向广告投放服务器发出数据请求，此时日志服务器会作出日志记录。然后广告投放服务器从广告素材服务器中查找相关的数据，查找到的数据将被返回到广告投放服务器中。其后广告投放服务器向负载均衡的Web服务器发出应答，广告投放服务器会向日志服务器发送记录日志的数据，日志服务去再次作出记录，web 服务

器将此结果再返回给页面，让页面去加载。在广告投放服务器返回结果时，就会及时更新缓存数据库中的数据。订单管理模块会去拿缓存中的数据和关系数据库的数据做更新的实时计算，并更新数据库中的数据。日志服务器会在每天

24: 00之后，重新作出统计并更新关系数据库。

## 3.4 数据库设计

### 3.4.1 关系型数据库设计

以系统的需求和数据间的传递关系为基础，设计出系统的E-R实体图，如图3-4所示。实体主要包括订单项框架、订单项、订单项客户订单、订单项匹配、发送信息表、媒体信息、广告单元、广告位置、广告尺寸和素材信息。

|  |  |  |  |
| --- | --- | --- | --- |
| 框架 | | |  |
| PK | 框架ID | |
|  | 订单类型 | |
| 媒体合同 | |
| 媒体金额 | |
| 折扣 | |
|  |
| 投放数量 | |
| 投放策略 | |
| 媒体开始时间 | |
| 媒体结束时间 | |
| 创建者 | |
| 创建时间 | |
|  | |  | |

|  |  |
| --- | --- |
| 订单 | |
| PK | 订单ID |
|  | 客户ID 合同号  服务开始时间服务结束时间服务提醒时间素材类型  交易类型销售人员 |

|  |  |
| --- | --- |
| 媒体信息表 | |
| PK | 媒体ID |
|  | 备注缩写  广告位置数量 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 广告位置 | | | |  |
| PK | 广告位置ID | | |
|  | 广告位置名称 | |  |
| 广告位置描述 | |  |
| 广告尺寸ID | |  |
| 广告位置状态 | |  |
| 投放级别 | |  |
| 媒体id | |  |
| 广告类型 | |  |
| 投放策略 | |  |
|  |
| 价格 | |  |
| CPD数量 | |  |
| 订单最大数量 | |  |
|  | |  | | |

|  |  |
| --- | --- |
| 广告尺寸字典表 | |
| PK | 尺寸ID |
|  | 广告高度广告高度 |

|  |  |  |
| --- | --- | --- |
|  | 广告单元 | |
| PK | 广告单元ID |
|  | 广告单元描述广告尺寸ID 边距  广告单元状态广告媒体ID |
|  |

匹配信息表

PK

匹配ID

订单ID

投放目标ID 文件名称 匹配类型 匹配状态 创建者ID 创建时间

|  |  |
| --- | --- |
| 素材信息 | |
| PK | 素材ID |
|  | 素材类型描述  属性状态  素材创建者ID 素材创建这事件 |

|  |  |
| --- | --- |
| 发送信息表 | |
| PK | 发送信息 ID |
|  | 发送目标类型发送目标ID 发送时间  终止事件广告代码发送类型 |

图3-4 数据库设计图

本系统选择了Postgresql数据库，因为Postgresql开源免费、语法简洁、易于学习，同时性能也非常出色，当前Postgresql, Postgresql可以根据业务特点选择合适的使用，对千万级以上的读写的支持也很好[35]。

在广告订单中，我们需要广告位置表，广告单元表，广告尺寸表，订单表，订单项表，素材表，媒体表，用户团队表，客户代理表。

订单表记录订单的各种信息和合同的部分信息。订单项表用于记录某个客户在某个订单下投放广告的数量。

广告投放中，我们需要媒体页面表，记录未指定投放广告位的页面信息。下面就数据库主要的结构作出说明。

表3-1 广告位置信息表

| 编号 | 字段名称 | 数据类型 | 说明 |
| --- | --- | --- | --- |
| 1 | name | Character varying(100) | 名称 |
| 2 | description | Character varying(500) | 描述 |
| 3 | size\_id | integer | 表示大小的 id |
| 4 | standard | Character varying(100) | 投放标准 |
| 5 | status | integer | 广告位置状态 |
| 6 | level | integer | 投放级别 |
| 7 | medium\_id | integer | 媒体 id |
| 8 | ad\_type | integer | 广告类型 |
| 9 | cpd\_num | integer | Cpd 数量 |
| 10 | max\_order\_num | integer | 订单最大数量 |
| 11 | price | integer | 价格 |
| 12 | launch\_strategy | integer | 投放策略 |

广告位置表如表3-1所示，主要记录了广告位的位置，所在页面的级别，广告位所属媒体及其他信息。

广告单元表如表3-2所示，记录在订单系统中主要记录展示位置和广告尺寸综合的关系。主要表达广告主可能一次选取多个展示位置的信息。广告的信息主要包括广告的大小广告的边距以及广告的状态等。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | 表 3-2 广告单元信息表 |  |  |
|  | 编号 | 字段名称 | 数据类型 | 说明 |  |
|  | 1 | name | Character varying(100) | 名称 |  |
|  | 2 | description | Character varying(500) | 描述 |  |
|  | 3 | size\_id | integer | 广告大小 id |  |
|  | 4 | margin | Character varying(50) | 边距 |  |
|  | 5 | target | integer | 目标 |  |
|  | 6 | status | integer | 广告状态 |  |
|  | 7 | medium\_id | integer | 广告媒体 id |  |
|  | 8 | estimate\_num | integer | 创建数量 |  |

表3-3 广告尺寸信息表

| 编号 | 字段名称 | 数据类型 | 说明 |
| --- | --- | --- | --- |
| 1 | width | integer | 广告宽度 |
| 2 | height | integer | 广告高度 |

表3-4 订单项匹配信息表

| 编号 | 字段名称 | 数据类型 | 说明 |
| --- | --- | --- | --- |
| 1 | target\_type | Character varying(50) | 投放目标类型 |
| 2 | target\_id | integer | 投放目标 id |
| 3 | filename | Character varying(500) | 文件名称 |
| 4 | attachment\_type | integer | 匹配类型 |
| 5 | attachment\_status | integer | 匹配状态 |
| 6 | creator\_id | integer | 订单项创建者 id |
| 7 | create\_time | date | 创建时间 |

广告尺寸如表3-3所示，属于字典表，用于记录广告的宽度和高度。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | 表 | 3-5 订单项客户订单信息 | 表 |  |
|  | 编号 | 字段名称 | 数据类型 | 说明 |  |
|  | 1 | agent\_id | integer | 代理人 id |  |
|  | 2 | client\_id | integer | 客户 id |  |
|  | 3 | campaign | Character varying(100) | 业务名称 |  |
|  | 4 | contract | Character varying(100) | 合同 |  |
|  | 5 | money | integer | 钱款 |  |
|  | 6 | contract\_type | integer | 合同类型 |  |
|  | 7 | client\_start | date | 服务开始时间 |  |
|  | 8 | client\_end | date | 服务停止时间 |  |
|  | 9 | reminde\_date | date | 提醒时间 |  |
|  | 10 | resource\_type | integer | 素材类型 |  |
|  | 11 | sale\_type | integer | 交易类型 |  |
|  | 12 | contract\_status | integer | 合同状态 |  |
|  | 13 | creator\_id | integer | 订单项创建人 id |  |

表3-6 订单项框架信息表

| 编号 | 字段名称 | 数据类型 | 说明 |
| --- | --- | --- | --- |
| 1 | group\_id | integer | 团体 id |
| 2 | description | Character varying(500) | 描述 |
| 3 | contract | Character varying(100) | 合同 |
| 4 | money | integer | 金额 |
| 5 | contract\_type | integer | 合同类型 |
| 6 | client\_start | date | 开始时间 |
| 7 | client\_end | date | 结束时间 |
| 8 | reminde\_date | date | 提醒时间 |
| 9 | contract\_status | integer | 合同状态 |
| 10 | creator\_id | integer | 创建者 ID |
| 11 | create\_time | date | 创建时间 |

订单项匹配信息表如表3-4所示，主要记录订单的投放类型、匹配状态等信息，用于订单项的匹配。

用户团队表和客户代理表如表3-5所示，用于管理使用此系统的各个角色的权限和增加或删除角色以及相应的角色配置。

表3-7 订单项订单信息表

| 编号 | 字段名称 | 数据类型 | 说明 |
| --- | --- | --- | --- |
| 1 | campaign | Character varying(100) | 公司 |
| 2 | medium\_id | integer | 媒体 id |
| 3 | order\_type | integer | 订单类型 |
| 4 | medium\_contract | Character varying(100) | 媒体合同 |
| 5 | medium\_money | integer | 媒体金额 |
| 6 | medium\_money2 | integer | 折扣金额 |
| 7 | sale\_money | integer | 销售金额 |
| 8 | medium\_CPM | integer | CPM 数量 |
| 9 | sale\_CPM | integer | 销售 CPM |
| 10 | discount | integer | 折扣 |
| 11 | medium\_start | date | 媒体开始时间 |
| 12 | medium\_end | date | 媒体结束时间 |
| 13 | creator\_id | integer | 创建者 id |
| 14 | create\_time | date | 创建者时间 |

表3-8 订单项安排

| 编号 | 字段名称 | 数据类型 | 说明 |
| --- | --- | --- | --- |
| 1 | item\_id | integer | 订单 id |
| 2 | num | integer | 订单数量 |
| 3 | date | date | 创建时间 |
| 4 | start | date | 开始投放时间 |
| 5 | end | date | 结束投放时间 |

订单项框架信息表如表3-6所示，表中主要定义了订单的起止时间、合同类型等信息，用于存储该方面的数据。

订单项订单信息表如表3-7所示，主要记录了公司、订单类型、媒体的起止时间和一系列的金额。

订单项安排表如表3-8所示，主要记录了订单的数量、创建时间及投放的起止时间。

发送信息表如表3-9所示，主要记录了发送的广告的目标类型、发送类型、发送时间等。

素材表如表3-10所示，表示所要投放广告的信息，也在此生成广告所展示的HTML代码。

媒体表如表3-11所示，记录各媒体的信息（已签约或达成合作意向）。

表3-9 发送信息表

| 编号 | 字段名称 | 数据类型 | 说明 |
| --- | --- | --- | --- |
| 1 | target\_type | Character varying(50) | 发送目标类型 |
| 2 | target\_id | integer | 发送目标 id |
| 3 | date | date | 时间 |
| 4 | value | integer | 广告代码 |
| 5 | delivery\_type | integer | 发送类型 |

表3-10 素材信息表

| 编号 | 字段名称 | 数据类型 | 说明 |
| --- | --- | --- | --- |
| 1 | name | Character varying(100) | 名称 |
| 2 | type | integer | 类型 |
| 3 | item\_id | integer | 字段 id |
| 4 | code | text | 描述 |
| 5 | props | bytea | 属性 |
| 6 | status | integer | 状态 |
| 7 | creator\_id | integer | 素材创建者 id |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 表 3-11 媒体信息表 |  |
|  | 编号 | 字段名称 | 数据类型 | 说明 |
|  | 1 | name | Character varying(100) | 媒体 |
|  | 2 | abbreviation | Character varying(100) | 缩写 |
|  | 3 | owner\_id | integer | 所有者 id |

表3-12 out\_source\_target

| 编号 | 字段名称 | 数据类型 | 说明 |
| --- | --- | --- | --- |
| 1 | name | Character varying(100) | 媒体名称 |
| 2 | bank | Character varying(100) | 账户名称 |
| 3 | card | Character varying(100) | 卡号 |
| 4 | alipay | Character varying(100) | 支付宝账号 |
| 5 | contract | Character varying(100) | 合同 |
| 6 | remark | Character varying(100) | 标记 |
| 7 | type | integer | 类型 |

表3-13 team\_admin\_users

| 编号 | 字段名称 | 数据类型 | 说明 |
| --- | --- | --- | --- |
| 1 | user\_id | integer | 用户 id |
| 2 | team\_id | integer | 组 id |

### 3.4.2 缓存数据库中键值对的设计

选用缓存数据库最主要的原因是因为我们高频词的读取数据[36]，这也是

redis最大的优点之一，性能极高每秒能支持100k+的读写频率。

在缓存数据库中我们主要需要字符串格式的数据类型，我们存放要投放广告单元信息和广告的HTML代码[37]，其数据格式为：

Key为：“AdId: %sData: %s: ItemID: %s"其中%s表示对应的数据。

Value为一段包含广告的HTML代码。

html代码示例如下：

<html>

<head>

<base target="\_blank">

</head>

<body scroll="no">

<a href = "%%CLICK%%[http://www.baidu.com](http://www.baidu.com/)" target="\_blank">

<img src ="http://10.211.55.5:8000/files/test.jpg" border="0">

</a>

<img src="%%MONITOR%%" style="position:absolute:">

</body>

</html>

## 3.5 系统各模块设计

### 3.5.1 订单管理模块设计

订单管理模块是在客户与销售在线下签订合同之后，到广告投放之前贯穿这个阶段。严谨的订单流程能够是保证广告投放正常的先决条件。订单管理的订单创建，订单维护，广告资源的添加的流程将在下一章广告投放的实现中详细的描述。

订单管理的具体流程：

订单管理主要分为订单的增删改查和订单的状态维护。创建订单的基本流程为：录入基本信息。选择展示位置，展示位置的各种数据来自于广告单元。并和媒体单元相关联。订单创建的主要实体类图如图3-5所示。

在图订单管理的时序图中，AdItem类为创建广告订单的Controller类，它的creator\_order（）方法在得到具体的表单数据之后，会调用AdPositon 的

add\_postion（）方法，然后会调用DAO层的save（）方法进行保存。并返回信息，创建成功。通过直接调用DAO来进行数据查询，进而完成订单的查询。并返回订单相关信息。

在此部分可以完成的功能有，添加媒体、添加广告尺寸、增加广告单元、上传广告文件，生成广告代码预览，录入合同号，选择执行策略，审批流程，和开启订单投放状态即订单进入投放状态。

进入投放状态之前的订单，会有上传的动作，需检查广告资源的合法性，内容检查，文件名检查，图片本身校验。存储到数据库中的数据必须时经过校验的数据。进入投放状态的订单后，便会将本次广告代码与ID按照既定的缓存数据库中的数据结构存储到缓存数据库中去。如图3-6所示为创建订单的时序

图，如图3-7所示为广告上传的时序图。



**Ad\_item**

**Material**

**AdPosition**

+id

+client\_id

+campaign\_id

+medium\_id

+order\_type

+contract

+discount

+creator\_id

+id

+name

+type

+item\_id

+item

+code

+state

+id

+size\_id

+standard

+status

+level

+untis

+add\_position()

+estimate\_num()

+schedules\_by\_date()

+ordered\_num()

+retain\_num()

+can\_order\_num()

+storge\_percent()

+get\_orders\_by\_date()

+get\_width()

+get\_height()

+get\_path()

+preview\_path()

+is\_online\_by\_date()

+query\_order()

+check\_order()

+create\_item()

+create\_order()

+get\_order\_state()

+get\_order\_by\_user()

+delete\_order()

+items\_by\_state()

+can\_action()

+create\_ad\_Item()

+approval\_adItem()

+modify\_adItem()

+save\_Item()

**ImageMaterial**

+save()

+image\_file()

+image\_link()

+click\_link()

+monitor\_link()

+get\_html()

+upload\_ad()

+query\_ad()

**AdUnit**

**Medium**

+id

+size\_id

+magrgin

+target

+state

+medium\_id

+id

+name

+owner\_id

**Ad\_size**

**Agent**

+add\_size()

+check\_size()

+save\_size()

+id

+name

+position\_info\_by\_date()

+create\_mediun\_order()

+check\_medium\_name()

+add\_mediumItem()

+get\_name()

+schedule\_num()

+retain\_num()

+order\_items()

图3-5 订单创建的主要实体类图



用户:



1 : create\_order()

2 : add\_position()

3 : self.save()

4 : return

5 : 创建成功

6 : query\_order()

7 : self.query()

8 : return

9 : 获取订单信息成功

AdPosition

db.model

Ad\_item

图3-6 订单创建时序图





2 : validity\_ad()

3 : save()

8 : return

9 : 查询成功

ImageMaterial

MaterialCheck

db.model

1 : upload\_ad()

4 : return

7 : query()

5 : 上传广告成功

6 : query\_ad()

用户:

图3-7 广告上传时序图

### 3.5.2 数据回写模块设计

在数据回写模块主要保证数据的真实性，当外部系统请求广告之后，会解析他的请求，得到广告ID，然后会在缓存里面做计数。

广告CPM和CPC的计算规则为：

（1）将缓存中存储的CPM和CPC数据，作为当天展示的CPM和CPC等数据。当日展示的CPC和CPM的数量以缓存中存在为准。

（2）每日的精确数字，以分析访问日志为准。

（3）累计投放量为昨日的精确数字与今日缓存之和。

（4）偏差率的计算为：投放阶段的每日缓存数量之和/投放阶段中的日志分析出的数据之和。

（5）正确投放量的计算：如果计算所得的偏差率大于1，那么投放量将按照正常数据截止。如果得到的偏差率下于1，则按照正确数据与偏差率的倒数之积来计算正确的投放量。

（6）每天的到的CPC和CPM数据都要更新到数据库中。

对于订单管理的展示效果的数据是处于动态更新的阶段。数据回写模块的

主要类设计图如图3-8所示。

**Ad\_handle**

**Delivery**

+order\_id

+get\_order\_ids()

+get\_last\_name\_by\_id()

+modefiy\_cpc\_num()

+modfiy\_cpc\_num\_id()

+getItemById()

+compareWithLastTime()

+get\_cpm\_num()

+get\_cpc\_num()

+update\_cpc\_num()

+updateDeliveryById()

+update\_delivery()

**Redis\_Count**

**ad\_log**

+get\_cpc\_num\_by\_id()

+get\_cpm\_num\_by\_id()

+get\_end\_time()

+compare\_num\_with\_RCount()

+updateRedisById()

+get\_cpc\_num\_by\_id()

+get\_cpm\_num\_by\_id()

+getLogNum()

图3-8 数据回写模块主要类设计图

管理员 ：



Ad\_handle Delivery Redis\_count

ad\_log

1: get\_cpc\_num()

2: get\_cpc\_num\_by\_id()

3: get\_cpc\_num\_by\_id()

4: query()

5: return

6：查询成功

7: modefiy\_cpc\_num()

8: modifiy\_cpc\_num\_id()

9: get\_cpc\_num\_by\_id()

10: compare\_num\_with\_RCount()

13：数据回写成功

12: update\_cpc\_num

11: return

图3-9 数据回写时序图

Ad\_handle类是计算数据的辅助类，本身无属性，方法中定义了数据计算的规则。以计算到今日为止的CPC数据为例，说明取CPC和CPM的过程。CPC数据回写时序图和取得CPC数据的时序图是相同的。

通过调用Ad\_handle的get\_cpc\_num（）方法，开始计算CPC数据，首先数据库中广告投放的历史精确数据，即通过调用Delivery的get\_cpc\_num\_by\_id（）方法取得历史精确CPC数据，再通过redis\_count即缓存计数Controller的到今

日的CPC数据。

在进行修改的过程中，拿到前一天的数据通过缓存计数Controller 的

compar\_num\_wiht\_RCount（）方法做比对，最后更新数据到数据库中。数据回写功能的时序图如图3-9所示。

### 3.5.3 管理投放模块设计

在订单管理模块当中，当准备完毕的订单进入投放状态之后，系统的基础服务层会通过任务管理，来对符合要求的订单进行广告的投放。此时外部系统访问此广告便会得到响应。对于订单不同的状态响应，管理投放模块也会做出不同的动作，细节请看下章管理模块实现流程。结合订单管理模块及数据回写模块的设计，下面给出管理投放模块的主要类设计图如图3-10所示。

**ad\_monitor**

+check\_order\_state()

+make\_key\_value()

+save\_code\_to\_redis()

**ad\_delivery**

+start()

+stop()

+suspend()



|  |  |  |  |
| --- | --- | --- | --- |
|  | <<interface>>  **order\_unit\_delivery** | |  |
|  | |  | |
| **order\_unit\_delivery\_impl** | | | |
|  | | | |
| +is\_first\_delivery()  +query\_order\_by\_unit\_id()  +make\_unit\_code()  +save()  +delivery\_ad()  +get\_Item\_code() | | | |

3-10 管理投放模块主要类设计图





order\_unti\_delivery

用户 :

2 : query\_order\_by\_unit\_id()

3 : check\_order\_state

5 : start()

6 : return

7 : 投放成功

ad\_monitor

ad\_delivery

1 : delivery\_ad()

4 : make\_unit\_code

图3-11 广告投放模块时序图

系统用户进行广告投放时，会调用order\_unit\_delivery即订单投放controller的query\_order\_by\_id方法去查询订单的ID，根据这个ID去查询订单的基本信息，得到的这些信息中最主要的就是订单的状态。当订单进入投放状态时，便生成广告单元的代码，转到缓存数据中，开始真正的广告投放，此时合作媒体便会在页面中嵌入广告代码进行广告投放。广告投放模块时序图如图3-11所示。

## 3.6 本章小结

在本章中首先给出了广告投放系统的功能概要设计，主要的功能模块有：订单基本管理、投放广告管理和数据回写管理。其后从分层的角度给出广告投放系统的总体结构，主要分为：展现层、应用层、基础服务层、数据库和云服务器平台，使系统的设计更加直观。然后给出了系统的核心数据流的流图，更清晰地呈现核心数据是怎样交互和运行的。

数据库的设计也是在本章中做出阐述的，根据数据间的传递关系和系统的需求给出了数据库的总体设计图。主要包括广告位置表，广告单元表，广告尺寸表，订单表，订单项表，素材表，媒体表，用户团队表，客户代理表。

其后对于算法的设计做出了分析，最后对系统各模块进行了设计，主要包括：订单管理模块设计数据回写模块设计、管理投放模块设计，各个模块的设计均以时序图的形式呈现出来，更形象直观。

# 第4章 广告投放系统的实现

## 4.1 订单管理模块实现

在系统功能设计中，本系统给出了订单部分的总体功能。在这一部分将对功能做出详细的流程设计。本文用两张图来表示订单的订单的主要功能创建和管理，其中创建订单的主要流程如图4-1所示，订单管理部分流程如图4-2所示。



开始

输入订单基本

信息

结束

否

输出订单信息

否

表单信息是否

通过验证

是

是

是否通过

审批

否

是

调用approval\_adItem()

方法审批订单

是

调用create\_ad\_Item()方

法生成广告订单

表单信息是否

通过验证

否

是

输入媒体订单信息

库存容量是否大于0

调用check\_inventory（）

检查当前库存容量

否

媒体名字

是否存在

调用create\_Item()方法生成订单

调用check\_order()方法检查基础订单

调用create\_order() 方法生成客户订单

调用check\_order()方法检查客户订单信息

调用

create\_medium\_order()方法生成媒体订单

调用

check\_medium\_name()方法检查媒体名称

图4-1 创建订单实现流程图

创建订单实现流程图4-1中设计到很多方法，这些方法的具体说明如表4-1

到表4-8所示。

表4-1 生成订单create\_item()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | create\_item() |
| 传入参数 | Int contract\_id,Int cliect\_id, Data client\_start,Date client\_end |
| 返回值 | Int item\_id |
| 描述 | 生成订单 |

表4-2 检查订单check\_order()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | check\_order() |
| 传入参数 | Int item\_id |
| 返回值 | Boolean isControll |
| 描述 | 检查订单状态 |

表4-3 创建客户订单create\_order()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | create\_order() |
| 传入参数 | Int item\_id,Int cliect\_id,String campaign, Date reminde\_date |
| 返回值 | Int item\_id |
| 描述 | 创建客户订单 |

表4-4 创建媒体订单create\_medium\_order ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | Create\_medium\_order () |
| 传入参数 | Int medium\_id,Int item\_id,Int order\_id |
| 返回值 | Boolean isSuccess |
| 描述 | 创建媒体订单 |

表4-5 检查媒体名称check\_medium\_name ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | Check\_medium\_name () |
| 传入参数 | String medium\_name |
| 返回值 | Boolean isLegal |
| 描述 | 检查媒体名称 |

创建订单时，首先新增订单，之后录入合同，输入的合同要经过验证，若通过验证则创建客户订单，若合同未通过验证则重新录入合同。如果新创建的客户订单的信息是合法的，则创建媒体订单，若不合法则重新创建广告订单。媒体订单通过信息的合法化验证之后，则进行库存的验证，通过库存的验证之后，就可以生成订单、确认订单，通过订单确认之后，整个订单创建的流程结束。若媒体订单或库存信息没有通过验证，则创建订单的流程直接结束。

表4-6 检查库存容量check\_inventory()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | check\_inventory() |
| 传入参数 | Int medium\_id |
| 返回值 | Int inventoryNum |
| 描述 | 检查库存容量 |

表4-7 创建广告订单create\_ad\_Item ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | Create\_ad\_Item () |
| 传入参数 | Int Item\_id,Int client\_id,Int medium\_id |
| 返回值 | Int adItem\_id |
| 描述 | 创建广告订单 |

表4-8 审批订单approval\_adItem ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | Approval\_adItem () |
| 传入参数 | Int adItem\_id |
| 返回值 | Boolean isPass |
| 描述 | 审批订单 |

订单管理流程图4-2中涉及到的方法的说明如表4-9到4-18所示

表4-9 获取广告订单信息get\_adItems ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | Get\_adItems () |
| 传入参数 | Int adItem\_id |
| 返回值 | Aditem aditem |
| 描述 | 获取广告订单信息 |

表4-10 查看广告订单是否投放isnot\_delivery ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | Isnot\_delivery () |
| 传入参数 | Int adItem\_id |
| 返回值 | Boolean isEffective |
| 描述 | 查看广告订单是否投放 |

表4-11 检查订单状态check\_order\_state ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | Check\_order\_state () |
| 传入参数 | Int order\_id |
| 返回值 | String order\_state |
| 描述 | 检查订单状态 |

表4-12 修改订单状态modify\_adItem ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | Modify\_adItem () |
| 传入参数 | Int adItem\_id |
| 返回值 | String isSuccess |
| 描述 | 修改订单状态 |

订单管理实现的流程为：首先初始化订单信息，判断订单是否投放，若不投放则进入查看订单信息阶段。若投放则判断是否可控，对于不可控的订单，系统做出禁止修改的提示，随即结束订单管理的操作；如果订单可控，则进行订单信息的查看。

查看订单信息之后修改订单，并审批该订单的信息，若未通过审批则修改订单；若通过审批则修改库存。修改库存之后查看订单的投放量，若投放量为0，则终止该订单的投放，随即结束订单管理；若订单投放量不为0，则开启该订单的投放，之后结束订单管理操作。

表4-13 修改库存容量modify\_inventory ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | Modify\_inventory () |
| 传入参数 | Int Medium\_id |
| 返回值 | None |
| 描述 | 修改库存容量 |

表4-14 检查库存容量check\_inventory ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | Check\_inventory () |
| 传入参数 | Int Medium\_id |
| 返回值 | Boolean isLegal |
| 描述 | 检查库存容量 |

表4-15 保存订单save\_Item ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | Save\_Item () |
| 传入参数 | Item item |
| 返回值 | Boolean isSuccess |
| 描述 | 保存订单 |

表4-16 开启投放start\_delivery ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | Start\_delivery () |
| 传入参数 | Int adItem\_id |
| 返回值 | Boolean isSuccess |
| 描述 | 开启投放 |

表4-17 结束投放end\_delivery ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | End\_delivery () |
| 传入参数 | Int adItem\_id |
| 返回值 | Boolean isSuccess |
| 描述 | 结束投放 |

表4-18 发送信息send\_message ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | Send\_message () |
| 传入参数 | Int adItem\_id |
| 返回值 | Boolean isSuccess |
| 描述 | 发送信息 |

新建框架页面如图4-3所示，新建框架的内容主要用于订单信息的设置，包括代理集团的记录、合同金额的录入，订单执行的起止时间的设定，以及销售渠道、合同模板类型是设定。该页面为使用者提供可视化的界面，方便使用者准确简易地进行订单信息的设置。



开始

是否投放状态

是

订单状态是否可控

否

否

是

调用modify\_adItem()修改订单状态

调用approval\_adItem()方

法审批订单

否 审批通过

是

调用modify\_inventory

（）方法修改库存

调用check\_inventory

（）方法检查库存量

库存是否为0

是

否

调用save\_Item()保存订单

调用start\_delivery()开启投放

结束

调用get\_adItems()方法获取订单信息

调用isnot\_delivery()检查订单投放状态

调用check\_order\_state

（）方法检查订单状态

调用end\_delivery()方法结束投放

调用send\_message()方法发送提示修改信息

图4-2 订单管理实现流程图



图4-3 新建框架页面



图4-4 新建客户订单页面

新建客户页面如图4-4所示，新建客户主要用于该客户基本信息的记录，包括用户的名称、客户组的名称，也包括所签合同的金额，以及合同执行的起始时间等信息。



开始

录入尺寸信息

广告尺寸

是否存在

否

调用save\_size()方法

保存广告尺寸

调用add\_position（）

方法增加展示位置

调用approval\_adItem()

方法审批

展示位置

是否合理

是

是

否 调用add\_mediumItem()

方法添加媒体信息

调用modify\_inventory

方法修改库存

结束

调用add\_size()方法增加广告尺寸

调用check\_size（）检查广告尺寸

调用upload\_ad() 创建广告资源

图4-5 广告资源增加流程图

创建订单涉及到广告资源的管理，广告资源部分主要负责媒体资源的创建，主要包括广告单元、展示位置、媒体这些资源。创建广告资源由增加广告单元、

创建展示位置、附加到媒体组成，本系统的广告资源管理模块实现流程如图4-5

所示。

广告资源增加流程图4-5中涉及的方法的具体说明如表4-19至表4-24所示。

表4-19 增加广告尺寸add\_size ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | Add\_size () |
| 传入参数 | Int wight，Int hight |
| 返回值 | None |
| 描述 | 增加广告尺寸 |

表4-20 检查广告尺寸check\_size ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | Check\_size () |
| 传入参数 | Int wight，Int hight |
| 返回值 | Int size\_id |
| 描述 | 检查广告尺寸 |

表4-21 保存广告尺寸save\_size ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | Save\_size () |
| 传入参数 | Int wight，Int hight |
| 返回值 | Boolean isSuccess |
| 描述 | 保存广告尺寸 |

表4-22 增加展示位置add\_position()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | add\_position() |
| 传入参数 | Int size\_id |
| 返回值 | Boolean isSuccess |
| 描述 | 增加展示位置 |

表4-23 上传广告upload\_ad ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | Upload\_ad () |
| 传入参数 | Int adItem\_id |
| 返回值 | Boolean isSuccess |
| 描述 | 上传广告 |

表4-24 增加媒体信息add\_mediumItem ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | Add\_mediumItem () |
| 传入参数 | Int medium\_id |
| 返回值 | Boolean isSuccess |
| 描述 | 增加媒体信息 |

在广告资源创建时，首先新建广告尺寸，根据新建的广告尺寸判断该广告资源是否存在，若存在则直接为该资源新建展示位置，如果不存在需要将该资源添加到Ad\_size表中，再为该资源新建展示位置。新建展示位置之后要将该广告资源的广告位置信息添加到ad\_position表中，之后管理员会审批该广告资源的位置信息，若未通过审批则重新创建展示位置；若通过管理员的审批则创建新的广告资源。之后要判断刚创建的资源是否已经存在，若存在，则将该资源添加到库存中。对于不存在的广告资源，需要添加媒体，并将该资源的媒体存储到mediuns表中，再添加到库存中。添加到库存之后，store表也要随之改变，随即整个广告资源管理结束。



图4-6 新建展示位置

Banber页面点击之后跳转到新建广告位置页面，新建广告位置页面如图4-6

所示：如果点击非新建字段则跳转到展示页面，反之跳入新建页面。新建展示位置表单如图4-7所示：



图4-7 新建展示位置表单

在广告资源管理模块，创建一个新的展示位置需要以广告尺寸和广告单元为前置条件。

新建展示位置时，先新建广告尺寸，之后判断是否存在，如果已经存在则不需要新建广告位置了，直接退出该流程；若不存在则为其创建广告单元，然后再创建展示位置，创建完成后退出创建新展示位置阶段。

订单项页面如图4-8所示，在订单项列表中主要显示订单项的名称，所属订单，以及该订单项创建的时间。



图4-8 订单项页面

预下单审批页面如图4-9所示，在页面中显示订单信息和排期的状态，售卖类型和展示位置等信息。

预下单详情页面如图4-10所示，在页面中主要显示预下单的订单名称、投放媒体、开始时间、结束时间、创建者、创建时间以及投放状态等信息。





图4-9 预下单审批页面

图 4-10 预下订单详情页面

## 4.2 数据回写模块实现

数据回写模块控制着订单量是否完成，以及精确的目标。数据回写模块还关系着订单的数据更新，决定着订单的状态。并且与广告主所要求的计次规则相关。如图4-11所示为数据回写模块的实现流程。

表4-25 获取广告投放信息getItemById ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | GetItemById () |
| 传入参数 | Int itemId |
| 返回值 | Item item |
| 描述 | 获取广告投放信息 |

表4-26 与最后一个投放时间比较compareWithLastTime ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | CompareWithLastTime () |
| 传入参数 | Int itemId |
| 返回值 | String isExist |
| 描述 | 与最后一个投放时间比较 |



开始

时间差是否小于30s

否

调用updateRedisById()方

法更新缓存数据

调用updateDeliveryById()

方法更新缓存数据

调用hasYesterdayDelivery()

得到昨天投放信息

是

信息是否存在

是

调用getLogNum（）得到日志分析数据

否

调用update\_delivery()方法

订正数据

结束

调用getItemById()方法获取广告投放信息

调用compareWithLastTime（） 方法与上次访问时间做比较

图4-11 数据回写模块实现流程图

进行数据回写时，首先获取广告投放信息，并且记录时间，然后判断30s内是否访问，如果不访问，则直接结束数据回写；如果在30s内有访问量，则查看该订单前一天是否被投放。如果前一天广告投放系统没有投放该订单，就直接判断订单的状态。如果前一天投放了，那么获取日志分析的数据，并订正数据，然后计算订单状态。订单状态被计算出来之后，立即更新订单数据，此时数据回写的操作也全部完成。

表4-27 更新缓存中广告数updateRedisById ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | UpdateRedisById () |
| 传入参数 | Int itemId |
| 返回值 | Boolean isSuccess |
| 描述 | 根据广告 ID 更新缓存中广告数 |

表4-28 更新投放数据updateDeliveryById ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | UpdateDeliveryById () |
| 传入参数 | Int itemId |
| 返回值 | Boolean isSuccess |
| 描述 | 更新投放数据 |

表4-29 判断昨日是否投放hasYesterdayDelivery ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | HasYesterdayDelivery () |
| 传入参数 | Int itemId |
| 返回值 | String isExist |
| 描述 | 判断昨日是否投放 |

表4-30 得到日志准确数据getLogNum ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | GetLogNum () |
| 传入参数 | Int itemId |
| 返回值 | Int num |
| 描述 | 得到日志准确数据 |

数据回写模块实现流程图4-11 中涉及到的方法的具体说明如表4-25到表

4-31所示。

表4-31 更新投放数据update\_delivery ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | Update\_delivery () |
| 传入参数 | Int itemId，Int num |
| 返回值 | Boolean isSuccess |
| 描述 | 更新投放数据 |

## 4.3 广告管理投放模块实现



开始

是否存在第一

次投放时间

否

调用get\_Item\_code（）方法得到广告代码

调用make\_key\_value（）

方法生成键值对

是

调用save\_code\_to\_redis()

方法存储到缓存数据库中

调用start（）方法开启

广告投放

结束

调用get\_adItems() 方法获取广告信息

调用is\_first\_delivery()方法获取订单第一次时间

图4-12 投放广告实现流程图

投放广告实现流程图4-12中涉及方法的具体说明如表4-32至4-36所示。

广告的投放状态是由订单及库存状态决定的。控制投放广告需要实现三个功能：开始、暂停和终止。开始状态代表可以合法订单，生成广告代码。暂停状态涉及到订单投放广告的主动暂停（需要修改投放状态的订单须先暂停订单的投放状态[40]）。终止投放分为人为终止和自然终止，修改投放量为0时，人为终止订单投放。本广告投放系统的投放广告实现流程如图4-12所示。

表4-32 判断是否第一次投放is\_first\_delivery ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | Is\_first\_delivery () |
| 传入参数 | Int itemId |
| 返回值 | String isFirst |
| 描述 | 是否第一次投放 |

表4-33 获取广告代码get\_Item\_code ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | Get\_Item\_code () |
| 传入参数 | Int itemId |
| 返回值 | String adHtml |
| 描述 | 获取广告代码 |

表4-34 创建键值对对象make\_key\_value ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | Make\_key\_value () |
| 传入参数 | Int itemId |
| 返回值 | AD ad |
| 描述 | 广告代码的键值对对象 |

投放广告的流程如图4-12所示：首先初始化订单信息，再判断是否新建，若不新建，则判断是否终止投放广告，若确定终止投放则直接结束投放，对于不终止投放的广告则判断是否开启投放，若不开启则暂停任务，修改定时任务，若开启投放则进入开启投放阶段[41]。

表4-35 保存键值对对象到缓存数据库save\_code\_to\_redis ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | Save\_code\_to\_redis () |
| 传入参数 | AD ad |
| 返回值 | Boolean isSuccess |
| 描述 | 保存键值对对象到缓存数据库 |

表4-36 开启投放start ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | Start () |
| 传入参数 | Int itemId |
| 返回值 | Boolean isSuccess |
| 描述 | 开启投放 |

对于新建的广告，需要获取订单信息，之后生成广告代码，并组成键值对，经过合法性验证的广告直接存储到缓存数据库中，如果验证结果为不合法，那么需要重新生成广告代码。对于存储到缓存数据库中的广告，则直接开启投放，修改定时任务之后就开始投放广告，即实现了广告投放的功能[42]。

## 4.4 一致性哈希算法的实现

在进入到投放的环节前，已经生成好了存储广告信息的键值对，我们需要把此键值对存储到广告投放系统的缓存数据库中，这时候就需要把我们的广告代码的键值对通过一致性哈希算法来映射到缓存数据库中[43]，以保证缓存数据库存储数据量的存储均衡，并此算法应该包括增加或去除缓存数据库时，此键值对的有效性。

一致性哈希算法的实现流程图如图4-13所示。

表4-37 将节点保存为dict对象sort\_dict()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | sort\_dict() |
| 传入参数 | String nodes |
| 返回值 | Dict dict |
| 描述 | 将节点保存为 dict 对象 |

表4-38 将对象转换为ring对象make\_ring ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | Make\_ring () |
| 传入参数 | Dict nodes |
| 返回值 | Ring ring |
| 描述 | 将对象转换为 ring 对象 |



开始

输入广告代码key和

所有缓存节点地址

环是否

为空

否

是

keyNew

<=len(node)

是

否

结束

调用sort\_dict()方法将节点保存为字典

调用make\_ring()方法， 点生成环对象

调用make\_md5\_Obj() 方法生成md5对象

调用update\_md5\_Obj()方法用key更新上文md5对象

调用Obj\_digest()方法生成新的16字节的新key

执行is\_exsited\_ring()节点环获取

调用save\_item\_order() 方法存入广告信息到缓存数据库

图4-13 一致性哈希算法实现流程

在本课题中，我们根据一致性哈希算法的实现流程图进行方法层面的解耦。

首先输入广告代码key和所有缓存节点地址，其后调用sort\_dict（）方法将节点保存为字典，并调用make\_ring（）方法将点生成环对象，调用make\_md5\_Obj（）方法生成md5对象，并调用update\_md5\_Obj（）方法用key更新上文md5对象，调用

Obj\_digest（）方法生成新的16字节的新key，判断环是否为空，若不为空则判断

key的长度是否合法，若合法则将广告信息存入缓存数据库，即实现流程结束。在本课题中用HashRing实现Python环境下的一致性哈希算法时，可以先

实现一些方法，通过调用这些方法可以从整体上实现一致性哈希算法[45]。

一致性哈希算法的实现流程图4-13中涉及的方法的具体说明如表4-37至表

4-43所示。

表4-39 返回md5对象make\_md5\_Obj()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | make\_md5\_Obj() |
| 传入参数 | None |
| 返回值 | MD md5 |
| 描述 | 返回 md5 对象 |

表4-40 生成新的md5对象update\_md5\_Obj ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | Update\_md5\_Obj () |
| 传入参数 | String key |
| 返回值 | MD md5 |
| 描述 | 生成新的 md5 对象 |

表4-41 生成新的key 的Obj\_digest ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | Obj\_digest () |
| 传入参数 | MD md5 |
| 返回值 | String key |
| 描述 | 生成新的 key |

表4-42 节点环获取is\_exsited\_ring()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | is\_exsited\_ring() |
| 传入参数 | None |
| 返回值 | None |
| 描述 | 节点环获取 |

表4-43 把广告数据保存save\_item\_order ()方法的说明

| 类别 | 内容 |
| --- | --- |
| 函数名 | Save\_item\_order () |
| 传入参数 | Int Item\_id |
| 返回值 | Boolean isSuccess |
| 描述 | 把广告数据保存 |

## 4.5 本章小结

在本章中，首先阐述了订单管理模块的实现，通过创建订单实现流程图4-1和订单管理实现流程图来实现该模块，并且呈现出实现后相关的页面。其后实现和数据回写模块，数据回写模块关系着系统中数据的更新和订单状态。

然后阐述了广告投放管理模块的具体实现，通过实现广告投放流程来实现该模块。最后通过使用HashRing实现Python环境下的一致性哈希算法，并呈现了相关函数的输入输出参数，以及对方法的作用做出解释。

# 第5章 广告投放系统的测试

本章主要阐述测试的效果，依托于系统需求主要进行回归测试。通搭建本地环境，模拟真正的投放效果，从而达到本系统的准备性与正确性。

## 5.1 测试环境

在软件的开发实现过程中，所有功能的完成都在2-3人做Code Review之后，通过git版本控制软件来提交，每次提交都伴随着单元测试的提交。由本部分功能测试主要以测试CPC和CPM数据的用例来体现功能测试。由于本系统主要供内部使用，非功能性测试主要以测试浏览器兼容性为主。

通过CPU、内存、操作系统、开发语言和版本、数据库工具和测试工具等的配置，完成了广告投放系统的测试环境的搭建，测试计算机硬件环境配置如表5-1所示。

表5-1 测试计算机硬件环境配置

| 硬件名称 | 硬件条件 |
| --- | --- |
| CPU | 英特尔 Core i5 |
| 内存 | 4GB ( 金士顿 DDR3L 1600MHz ) |
| 硬盘 | 128G 闪存 |

用于广告投放系统的测试的计算机软件环境配置如表5-2所示。

表5-2 测试计算机软件环境配置

| 软件名称 | 所用软件 |
| --- | --- |
| 操作系统 | OS X 10.10 Yosemite |
| 数据库 | Postgresql redis |
| 开发工具 | Vi |

系统部署测试计算机要求包括如下：

服务器一台作为管理端，ubuntu，服务器最低配置为16G内存、1024G的磁盘大小。

## 5.2 单元测试

本系统进行单元测试主要采用Pytest，其特点主要是灵活和简便性。

Pytest不但具有参数和模块化的平台，还拥有参数化的测试函数，并且支持

Skip和xfail处理失败的用例。它的xdist插件分发测试给多个CPU，不断地重新运行失败的测试。单元测试用例如表5-3所示：

表5-3 单元测试用例

| 名称 | 说明 |
| --- | --- |
| 测试用例 | 增加用户 |
| 用例编号 | Pytest\_Validation\_addUser\_01 |
| 需求标识 | 增加用户是否成功 |
| 测试代码 | TestOrder |
| 前置条件 | 用户名称不存在 |
| 测试步骤 | 在命令行中 执行python test\_order.py |
| 预期结果 | 测试结果正常，能够增加用户 |
| 实际结果 | 符合预期结果 |
| 相关缺陷 | 无 |
| 测试结论 | 正常 |

在本系统中需要进行的单元测试比较多，下面举例来说明单元测试的过程，我们使用单元测试测试对增加用户进行验证，单元测试代码如下：

From models. order import Order

From helper import add\_user, add\_order def test\_order(session):

Order2 = order. get(order. id) assert order2 is not None

User2 = add\_user(„testuser2,‟pwd,‟2) order. designers=[user2]

Order3 = Order. get(order. id) assert user2 is not None

## 5.3 功能测试

### 5.3.1 订单管理功能测试

订单管理功能主要分为创建订单功能和新建广告单元两部分，下面将对这两部分分别功能测试。

#### （1）创建订单功能的测试

首先测试创建订单功能，测试中所用到的参数如表5-4所示。

表5-4 创建订单功能测试

| 测试项目名称 | 创建订单 |
| --- | --- |
| 测试内容 | 创建订单功能 |
| 执行者 | 销售 |
| 测试用例 |  |
| 输入： | 客户名称，活动名称，金额，折扣，合同号，代理公司，策划人员 |
| 输出： | 新增订单的提示信息，数据存储到数据库 |
| 错误输入： | 提示必须项或正确的数据格式 |
| 错误输出： | 新增订单信息错误，请重新输入 |
| 步骤及操作： | 1.销售正常登录系统  2.销售点击创建订单，弹出创建订单页面如图 5-1 所示  3.销售点击客户名称，活动名称，金额，折扣，合同号，代理公司，策划人员  4.销售点击提交，页面跳转  5.销售正确的信息，并会生成订单，系统会在消息栏提出测试。如图 5-2 所示  6.销售点击我的菜单能够看到新增订单。如图 5-3 所示 |

.



图5-1 创建订单的测试用例表



图5-2 新建订单成功提示消息



图5-3 订单显示

#### （2）新建广告单元的测试

对系统新建广告单元的功能进行测试，所用到的参数如表5-5所示。

表5-5 新建广告单元功能测试

| 测试项目名称 | 新建广告单元 |
| --- | --- |
| 测试内容 | 新建广告单元功能 |
| 执行者 | 媒体管理员 |
| 测试用例 |  |
| 输入： | 名称，描述，尺寸，广告标准，状态，媒体，资源类别 |
| 输出： | 新增订单的提示信息，数据存储到数据库。 |
| 错误输入： | 已经存在的名字 |
| 错误输出： | 新增广告单元名称重复 |
| 步骤及操作： | 1.媒体管理员正常登录系统  2.媒体管理员点击创建广告单元，弹出创建订单页面如图  5-4 所示  3.媒体管理员点击客户名称，活动名称，金额，折扣，合同号，代理公司，策划人员  4.媒体管理员点击提交，页面跳转  5.媒体管理员正确的信息，并会生成订单，系统会在消息栏提出测试。如图 5-2 所示  6.媒体管理员点击广告单元列表能够看到新增订单，如图  5-3 所示 |



图5-4 新建广告单元页面



图5-5 广告单元列表页面

### 5.3.2 广告投放功能测试

对于功能测试，这里列出的是一般情况，针对不同的应用系统，其测试内容的差异很大，但都可以归纳为为界面、数据、操作、逻辑、接口等几个方面。本系统主要对界面和数据进行功能测试。功能测试用例如表5-7所示。

表5-7 广告投放功能测试

| 测试项目名称 | 广告投放 |
| --- | --- |
| 测试内容 | 广告投放功能 |
| 执行者 | 投放控制人员 |
| 测试用例 |  |
| 前置条件 | 订单进入预投放状态 |
| 输入： | 名字 ，图片，跳转连接，展示检测，有效 |
| 输出： | 广告代码预览 |
| 错误输入： | 已经存在的名字和图片名字重复 |
| 错误输出： | 您提交的广告命名已存在，请重新输入。 |
| 步骤及操作： | 1.投放控制人员点击上传广告。弹出如图 6-6 页面  2.投放控制人员输入信息  3.投放控制人员点击确定  4.媒体管理员点击提交，页面跳转  5.广告代码预览页面展示如图 5-7 所示 |



图5-6 广告基本信息



图5-7 广告代码预览页面



图5-8 广告展示基本数据

下面对系统中的CPC和CPM数据进行测试，检验这两种数据是否正确，测试的用例如表5-8所示

表5-8 CPC和CPM数据测试

| 测试项目名称 | CPC 和 CPM 数据测试 |
| --- | --- |
| 测试内容 | 广告投放功能 |
| 执行者 | 访问广告页面人员 |
| 测试用例 |  |
| 前置条件 | 订单进入投放 |
| 输入： | 刷新或点击页面 |
| 输出： | 广告状态页面数据更新 |
| 错误输入： | 30s 内重复点击或刷新页面 |
| 错误输出： | 广告数据状态页面。 |
| 步骤及操作： | 1.用户登录广告页面  2.用户点击刷新页面  3.广告数据状态页面数据更新。广告数据状态页面如图 5-8 所  示 4。用户在 30s 内再次刷新或点击页面数据无法应答 |

## 5.4 性能测试

性能测试主要测试在并发的情况下，订单管理部分的系统响应时间。由于订单部分在内部使用，目前用户数量大约在300 左右。所以本课题在性能测试

的环节所设置的用户并发上限为300在性能测试的情况中，本课题对于查询的测试，针对订单部分最主要核心的几个功能点进行性能测试。

登录是指输入帐号密码到经过后台帐号密码及权限的验证最后到跳转到登录界面的过程。获取订单项列表是指用户登录系统以后，查询某个客户的订单项，获取其订单列表。上传广告是指客户完成订单的环节之后上传广告到图片服务器。生成广告代码是指在上传到图片服务器之后对信息生成相应的广告代码。

统计不同并发情况下的事务平均响应时间，不同并发情况下事物平均响应时间统计：单位（ms），如表5-9所示，表5-9对应的三维柱状图如图5-9所示。

表5-9 不同并发情况下的事务平均响应时间

| 并发线程数 | 1 | 100 | 200 | 300 |
| --- | --- | --- | --- | --- |
| 测试动作 |  |  |  |  |
| 登录 | 348 | 526 | 871 | 1034 |
| 获取订单项 | 597 | 1147 | 2017 | 2741 |
| 上传广告 | 492 | 1492 | 1928 | 2386 |
| 生成广告代码 | 356 | 2095 | 3045 | 3794 |



图5-9 不同并发情况下的事务平均响应时间三维柱状图

本课题中在读取广告代码信息到缓存数据库中，引入了一致性哈希算法本部分采取插入数据的方式来进行算法的测试。

#### （1）初始化4个缓存节点服务器，用A、B、C、D表示，并存入10000

条数据。相应的数据分布如表5-10所示。

表5-10 初始化4个缓存节点服务器的数据分布情况

| 服务器 | 数据分布 |
| --- | --- |
| A | 2395 个 |
| B | 2657 个 |
| C | 2450 个 |
| D | 2498 个 |

#### （2）去除服务器D，数据的分布情况如表5-11所示。

表5-11 去除缓存节点服务器D之后的数据分布情况

| 服务器 | 数据分布 |
| --- | --- |
| A | 3350 个 |
| B | 3177 个 |
| C | 3473 个 |

#### （3）增加服务器E数据的分布情况如表5-12所示。

表5-12 增加缓存节点服务器E之后的数据分布情况

| 服务器 | 数据分布 |
| --- | --- |
| A | 2479 个 |
| B | 2507 个 |
| C | 2613 个 |
| E | 2401 个 |

#### （4）增加服务器F，使存储的数据进行加倍，数据的分布情况如表5-13 所

示。

表5-13 增加缓存节点服务器F之后的数据分布情况

| 服务器 | 数据分布 |
| --- | --- |
| A | 4023 个 |
| B | 4093 个 |
| C | 3959 个 |
| E | 3978 个 |
| F | 3947 个 |

## 5.5 非功能性测试

表5-14 非功能性测试用例

| 名称 | 说明 |
| --- | --- |
| 测试用例 | 浏览器兼容性测试 |
| 用例编号 | Test\_browser |
| 需求标识 | 是否兼容 |
| 测试代码 | TestAdd |
| 前置条件 | 系统正常运行 |
| 测试步骤 | 在不同的浏览器中运行此系统 |
| 预期结果 | 每个页面均正常显示，使用无 bug |
| 实际结果 | 符合预期结果 |
| 相关缺陷 | 无 |
| 测试结论 | 正常 |

非功能性测试主要以浏览器兼容性测试为主，本系统需要在safari 和

Chrome浏览中正常运行。非功能性测试用例如表5-14所示。

## 5.6 本章小结

本章中主要进行广告投放系统的测试，首先进行测试环境的配置，随后进行单元测试。在功能测试中，主要对订单管理和广告投放功能进行功能测试，功能测试中给分别给出了测试用例。在非功能性测试中，主要进行浏览器兼容性的测试。

结论

论文对基于一致性哈希算法的广告投放系统的设计与实现进行了详细的描述。调查了国内外的研究现状，结合实际业务进行了系统需求分析和设计，并给出了实现过程。

主要的工作内容有：

（1）依据国内外对于当前广告投放系统的理论进行了深度分析，并与所在企业现状进行实际的结合，还有对其业务的特色进行了归纳，根据系统分析与设计的理论知识，对广告投放系统进行了细致的功能性需求和非功能需求分析。

（2）根据系统需求分析与当前软件行业的技术走向即敏捷开发的需求，采用精简的python语言和Flask框架体系进行开发，一系列丰富的类库，使开发人员的数量尽可能地减少，工作量也大大减少，能够更容易地实现易维护、开发敏捷的广告投放系统。在MVC的分层架构中，实现了高内聚和低耦合的设计，方便系统的迭代开发与后期维护。

（3）引入一致性哈希算法，并对广告投放系统的设计与实现的过程进行了详细的描述。引入算法的同时，考虑了其他应用于缓存案例的算法的不足之处，解决了在存放广告代码和增加减少缓存节点时，原数据受到影响的问题。

（4）提出了CPC和CPM的计算规则，当缓存库中广告数量的数据丢失时，有比较好的应对措施。主要根据缓存数据库中的数据与日志中的数据做比率计算来保证广告投放数据的准确性，进而满足广告主的要求。

本课题的发展趋势：

第一点：主要放在广告推荐的内容上，使整个广告的投放和推送达到最大效益化，建立起和企业合作的网站的资料库，记录投放网页的信息，进行内容匹配或者广告推荐。

第二点：现阶段计费还在线下计费，订单管理部分要开放出去，计费在线上计费，让想投放广告的客户进行自助下单，自助选取策略，自助投放，来响应这个行业的大趋势，逐步去完善本课题的研究进而优化本课题涉及的系统。

参考文献

[1] 刘晨. 移动广告平台投放子系统的设计与实现[D]. 北京: 北京邮电大学硕士学位论文, 2013: 4-6.

[2] 李志. 互联网广告精准投放平台的研究[D]. 武汉: 华中师范大学硕士学位论文, 2013: 15-23.

[3] Karypis G, Han Eui-Hong, Kumar V. Chameleon: Hierarchical Clustering Using Dynamic Modeling[J]. Computer, 2011' 32(8): 68-75.

[4] 张曦予. 基于微博的定向广告投放系统的设计与实现[D]. 长沙: 湖南大学硕士学位论文, 2014: 5-12.

[5] P. ChatterjeeDLa. HolWhlanandT. P. Novak. theeliek stream: ImPlieations for web-based advertising efforts[J]. Marketing Seience, 2003, 22(4): 520-541.

[6] 李平. 互联网广告精准投放系统平台的研究[D]. 武汉: 华中师范大学硕士学位论文, 2013: 2~3

[7] Feng, J., H. Bhargava, andD. Pennock. Implementing Sponsored Seareh in Web Seareh Engines: Computational Evaluation of Altemative Meehanisms[J]. Informs Journal on Computing, 2013, 19(l): 134-148

[8] Wang C, Zhang P, Choi R, et al. Understanding consumers attitude toward advertising[A]. Eighth Americas conference on information systems[C]. 2002: 1143-1148.

[9] P. O. Reily and A. Duane. Smart Mobile Media Services: Consumer Intention Model[A]. The 8th International Conference on Advances in Mobile Computing & Multimedia (MoMM)[C]: Paris, 2013: 188-194.

[10] Z. Yin, L. Cao, J. Han, C. Zhai and T. Huang. Geographical Topic Discovery and Comparison[A], 20th International World Wide Web Conference[C], India, 2011: 247-256.

[11] H. Yang, S. Chen, M. Lyu and I. King, Location-Based Topic Evolution[A].1 st International Workshop on Mobile Location–Based Service[C], Beijing, 2011: 89-98.

[12] C. Chow and M. Mokbel. Trajectory Privacy in Location-Based Services and Data Publication[A]. 17th International Conference on Knowledge Discovery and Data Discovery[C], San Diago, 2011: 19-29.

[13] C. Evans, Intelligent Retail Business: Location-Based Services for Mobile Devices[A]. 2nd International Conference on Pervasive Computing and Applications[C], UK, 2012: 1-5.

[14] P. Chatterjee, D. L. Hoffman, and T. P. Novak. Modeling the clickstream: Implications for web-based advertising efTorts[J]. Marketing Science, 2003: 520-541.

[15] Anisio Lacerda, Marco Cristo, Marcos Andre Goncalves, Weiguo Fan, Nivio Ziviani, and Berthier Ribeiro-Neto. Learning to advertise[A]. In SIGIR'06[C], 2006: 549-556.

[16] J. Hong and J. Landay, An architecture for privacy-sesitive ubiquitous computing[A]. MobiSys, June 6-9th, Boston, Massachusettes[C], USA, 2004: 335-377.

[11] Moore, P., Hu, B, & Wan, J. „Intelligent Context‟for Personalised Mobile Learning[A]. inArchitectures for Distributed and Complex M-Learning Systems: ApplyingIntelligentTechnologies[C], (Eds) S. Caballe; F. Xhafa; T. Daradoumis; A. A. Juan, Hershey, IGI Global, USA. 2010: 256-288.

[18] S. Dhar and U. Varshney (2011). Challenges and Business Models for Mobile Location-Based Services and Advertising[J], Communications of the ACM, Vol 54 (5): 121-129.

[19] BAGHER JEIRANA, PAREKHR. Combining behavioral and soeialnet workdatafo: online advertising[A]. 2008 IEEE International Confereneeon Data Mining Workshops[C]. 2008: 837-846.

[20] 高兰兰. 基于行为定向的精准广告投放系统的研究与实现[D]. 北京: 北京邮电大学硕士学位论文, 2012: 7-25.

[21] YLi, A. C. Surendran, and D. Shen. Datamining and audienee intelligence for advertising. SIGKDDExplor[J]. Newsl. 2007, 9(2): 96-99.

[22] 张淼. 人人网广告精准投放与反作弊系统设计与实现[D]. 长春: 吉林大学硕士学位论文, 2012; 13-39.

[23] Shvachko, K. The Hadoop Distributed File System[J]. Sunnyvale. May 2010.1-10: 46-50.

[24] 金欣. 移动广告精准分析平台的设计与实现[D]. 北京: 北京邮电大学硕士学位论文, 2012: 14-25.

[25] 段超. 键值存储的互联网广告投放系统需求方平台设计与实现[D]. 武汉: 华中师范大学硕士学位论文, 2014: 13-39.

[26] 李师童. 一种分布式广告投放引擎的设计与实现[D]. 北京: 北京邮电大学硕士学位论文, 2011: 9-23.

[27] Boldi P, Codenotti B, Santini M. UbiCrawler: A Scalable Fully Distributed Web Crawler [J]. Software: Practice & Experience. 2011, 34: 711-726.

[28] 张国燕. 基于标签的个性化广告精准营销系统设计与实现[D]. 武汉: 华中师范大学硕士学位论文, 2013: 19-30.

[29] 段徐蓉. 品友广告投放系统的设计与实现[D]. 北京: 北京交通大学硕士学位论文, 2013: 29-40.

[30] SongYao, CarlF. Mela. ADynamieModelofSponsoredSeareh Advertising[J]. Marketing Science Volume 30 Issue 3, May 2011: 447-468.

[31] Anisio Lacerda, Mareo Cristo, Mareos Andre Goncalves, weiguo Fan, Nivio Ziviani. Learning to advertise[J]. In SIGIR'06, 2010: 549 - 556.

[32] Zhang GuangWei, Li DeYi, Li Peng, Kang JianChu and Chen GuiSheng. A Collaborative Filter Recommendation Algorithm Based on Cloud Model [J]. Journal of Software, 2007, 18(10): 2403-2411

[33] Adrian Giurca, Emilian Pascalau. JSON Rules. In Proceedings of the Proceedings of 4th Knowledge Engineering and Software Engineering[J]. Informs Journal on Computing. 2008. Vol425: 7-18.

[34] 熊昱浩. 内文广告原型系统的设计及其关键技术的实现[D]. 北京: 北京邮电大学硕士学位论文, 2009: 23-45.

[35] 霍黎阳. 微博意见领袖影响力研究[D]. 北京: 北京邮电大学硕士学位论文, 2013: 14-21.

[36] 潘哲煦. MyAD广告投放系统的合同排期子系统的设计与实现[D]. 南京: 南京大学硕士学位论文, 2012: 15-23.

[37] A Likas, N Vlassis, J Verbeek. The global k-means algorithm [J]. Pattern Recognition. Vo. l 36, 2010, 451-461.

[38] Schwartz, Baron, PeterZaitsev, andVadimTkachenko. High Performance MySQL: Optimization, Backups, andReplication[J]. O'ReillyMedia, 2012.3, 1-34

[39] 何峥. 基于适宜度的网站广告自动投放系统的设计[D]. 成都: 电子科技大学硕士学位论文, 2010: 13-36.

[40] 钟云云. 基于地理信息的内容定向广告投放算法及性能研究[D]. 杭州: 浙江大学硕士学位论文, 2011: 30-50.

[41] 孙乃利. 基于微博意见领袖的个性化广告投放系统的设计与实现[D]. 北京: 北京邮电大学硕士学位论文, 2013: 33-40.

[42] 孙铭霞. 基于动态聚类分析的广告投放技术研究与实现[D]. 上海: 复旦大学硕士学位论文, 2013: 23-45.

[43] 刘佳璐. 网络广告研究[D]. 武汉: 华中师范大学硕士学位论文, 2005: 10-13.

[44] 兰小合. 基于SSH的数字电视广告投放系统的设计与实现[D]. 广州: 中ft大学硕士学位论文, 2013: 13-45.

[45] 胡建明. Green手机广告投放系统设计与实现[D]. 上海: 复旦大学硕士学位论文, 2009: 13-40.

# 哈尔滨工业大学学位论文原创性声明及使用授权说明

# 学位论文原创性声明

本人郑重声明：此处所提交的学位论文《基于一致性哈希算法的广告投放系统的设计与实现》，是本人在导师指导下，在哈尔滨工业大学攻读学位期间独立进行研究工作所取得的成果，且学位论文中除已标注引用文献的部分外不包含他人完成或已发表的研究成果。对本学位论文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。

作者签名：日期：2015年7月3 日



学位论文使用授权说明

学位论文是研究生在哈尔滨工业大学攻读学位期间完成的成果，知识产权归属哈尔滨工业大学。学位论文的使用权限如下：

（1）学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文，并向国家图书馆报送学位论文；（2）学校可以将学位论文部分或全部内容编入有关数据库进行检索和提供相应阅览服务；（3）研究生毕业后发表与此学位论文研究成果相关的学术论文和其他成果时，应征得导师同意，且第一署名单位为哈尔滨工业大学。

保密论文在保密期内遵守有关保密规定，解密后适用于此使用权限规定。本人知悉学位论文的使用权限，并将遵守有关规定。

作者签名：日期：2015年7月3 日



导师签名：日期：2015年7月3 日



致谢

短短的两年研究生生活马上就结束了。回忆起第一次刚到学院的第一天，仿佛就如昨日般记忆犹新。

本文是在校内导师涂志莹讲师的指导下完成，我在北京实习，由于我是

SAP方向的学生，本科又是网络工程的专业，是涂老师不厌其烦的在论文的理论基础和专业教育的基础上给予了最大的帮助。

涂志莹讲师给予我职业上有力的影响处是他在中期答辩时说过的一句话，如果不好好弄理论与基础又与码农何异。在此，致以衷心的感谢和崇高的敬意。

再一次感谢我的老师，导师，同事、同学。老师给我知识与关怀，导师给予我理论知识和职业影响，同学给予生活和学业上的帮助，同事教给我工作。再次感谢所有在我攻读硕士期间帮助过我的所有人，我将且记且行。

# 个人简历

武振朝，男，汉族，1985年7月10日生于河北大名。

2008年考入河北科技大学理工学院网络工程，2012年7月本科毕业并获得工学学士学位。

2013年9月--2015年7月在哈尔滨工业大学软件学院软件工程专业学习，攻读硕士学位。