电子科技大学

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

专业学位硕士学位论文

MASTER THESIS FOR PROFESSIONAL DEGREE

论文题目基于银行客户数据分析的用户画像系统设计与实现

专业学位类别	工 程 硕 士
学　　号	201651220120
作 者 姓 名	马　晓　冬
指 导 教 师	陈波　副教授

分类号 _____ 密级 _____

UDC [注1] _____

学 位 论 文

基于银行客户数据分析的用户画像系统设计与实现

（题名和副题名）

马晓冬

（作者姓名）

指导教师 **陈 波** **副教授**

电子科技大学 **成 都**

陆 太 **高 工**

中国电信四川广元分公司 **广 元**

（姓名、职称、单位名称）

申请学位级别 **硕士** 专业学位类别 **工 程 硕 士**

工程领域名称 **软件工程**

提交论文日期 2020.3.31 论文答辩日期 2020.5.9

学位授予单位和日期 **电子科技大学** 2020 年 6 月

答辩委员会主席 _____

评阅人 _____

注 1：注明《国际十进分类法 UDC》的类号。

Design and Implementation of User Portrait System Based on bank Client Data Analysis

A Master Thesis Submitted to

University of Electronic Science and Technology of China

Discipline: **Master of Engineering**

Author: **Xiaodong Ma**

Supervisor: **Bo Chen**

School: **School of Information and Software Engineering**

独创性声明

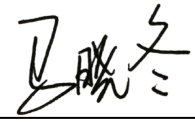
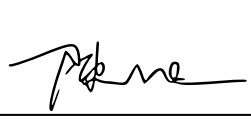
本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

作者签名：  　　日期：2020年 5月 9日

论文使用授权

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，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，允许论文被查阅和借阅。本人授权电子科技大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后应遵守此规定）

作者签名：  　　导师签名： 

日期：2020年 5月 9日

摘　要

近年来，我国信息化快速发展，大数据应用越来越广泛和深入，截止 2019 年我国网民规模已达到 8.5 亿。互联网用户的规模效应推动了大数据各类应用产生。本课题将围绕大数据相关技术，对四川省内某家地方性中小银行为目标，分析该银行数据并挖掘业务需求，开展基于客户数据的用户画像系统设计与实现的研究工作。该银行地处四川偏远落后地区，如何面对电子银行、互联网银行的新浪潮，适应大环境及各种新兴技术的快速发展和广泛应用，将银行科技发展战略与大数据等新技术相结合。如何将以自身业务为中心逐渐转向以客户为中心的发展模式、主动了解客户、提高客户忠诚度、开发适合市场不同群体的各类产品，为银行提供二次产能。本课题基于以上问题，展开基于该银行数据的用户画像系统设计与实现的研究工作。

本课题主要借鉴现有的用户画像理论研究成果，结合大数据相关技术，主要研究内容及目标为以下几方面：

一是通过对该行综合情况分析，分析该行现有的各类业务系统，开展对已有数据的分析、梳理，并制定整合及进一步挖掘的明细需求，为建立一套清晰分析逻辑的大数据系统提供需求依据。

二是根据业务需求，开展用户画像系统的设计研究。通过对大数据技术、数据分析技术、用户标签体系及展示系统技术的相关理论和方法介绍，结合业务需求，自底向上完成大数据及用户画像系统的架构设计工作。

三是通过搭建、开发和实现用户画像系统，研究具体大数据、数据计算、用户标签化及系统展示等问题。本论文重点为大数据架构搭建、数据的处理与分析、标签体系建设、应用及归类，通过将该行核心系统、信贷系统等主要业务系统的数据采集、分析、处理后，通过前端数据可视化系统，最终实现一套完整的用户画像系统的目标。

该目标实现的意义在于，使该行的系统用户能够在使用系统时，可根据不同类别标签、进一步人工分析。最终通过产品与服务，在竞争激烈的当下，实现留住老客户、争夺新客户，争夺更多市场资源，提高核心竞争力，从而提高盈利水平。

关键词：用户画像，用户标签，Hadoop 大数据，银行数据价值，数据分析

ABSTRACT

In recent years , with the rapid development of information technology in China , the application of big data is further extensive and in-depth . By 2019 , the scale of Internet users in China has reached 850 million , and the scale effect of Internet users is conducive to the production of various applications of big data . The topic will focus on the related technology of big data , and take a local small and medium-sized bank in Sichuan Province as the research object to analyze the bank data and explore the business needs , so as to study the design and implementation of user portrait system Based on customer data . The bank is located in the remote and backward areas of Sichuan Province , facing the following problems : e-banking and Internet banking are in a new wave , how to adapt to the environment and the rapid development and wide application of various emerging technologies , and combine the bank's technology development strategy with new technologies such as big data . How to make use of the development mode of gradually changing from business-centered to customer-centered , actively understand customers , improve customer loyalty , and develop various products suitable for different groups of the market to provide secondary capacity for banks . Based on the above problems , this paper studies the design and implementation of user portrait system based on the bank data .

The topic mainly draws lessons from the existing research results of user portrait theory , and combines with big-data-related technologies . The main research contents and objectives are as follows :

First , after the study of the bank's comprehensive situation , analyze its existing business systems , analyze and sort out the existing data , and formulate the detailed requirements for integration and further mining , so as to provide the basis for the establishment of a set of big data system with clear analysis logic .

Second , according to the business requirements , design and research user portrait system . By introducing related theories and methods , such as big data technology , data analysis technology , user label system and display system technology , the architecture design of big data and user portrait system is completed from bottom to top according to business requirements .

Third , build , develop and implement user portrait system , study specific big data ,

data calculation , user tagging and system display . This paper focuses on the construction of big data architecture , data processing and analysis , the construction , application and classification of the label system . After the data collection , analysis and processing of the bank's core system , credit system and other major business systems , a complete set of user portrait system is finally achieved through the front-end data visualization system .

The significance of this goal is to enable the system users of this bank to further analyze manually according to different categories of labels when using the system . Finally , in the fierce competition , products and services are used to retain old customers , compete for new customers and more market resources , and improve the core competitiveness , so as to improve the profitability .

Keywords: user profile, user label, Hadoop big data, bank data value, data analysis

目 录

第一章 绪 论

1.1 研究背景和意义

近年来，我国信息化建设速度已位居国际前列，随着信息化水平与普及率日益提高，根据 CNNIC 统计数据[1]，我国网民人数规模已经突破 8 亿大关，截止 2019 年 06 月，我国网民规模已达到 8.54 亿，其中网络支付用户规模已近 6 亿。信息化快速发展 “互联网+”应用越来越广泛和深入，众多互联网龙头企业开始涉足金融领域，倒逼金融业加快改革自己的步伐。银行业金融机构间竞争激烈，加之互联网企业进入金融领域，银行业务由被动服务、以自我业务为中心逐渐转变为以市场为导向、以客户为中心的服务模式。为了适应国家大数据战略，紧跟我国金融领域快速发展更迭的脚步，随着金融行业不断开放及与各类行业不断融合，日新月异的金融产品也越来越多。金融行业各类机构尤其是银行业机构需要快速实现产品创新，满足客户需求，提高自身核心竞争力，就必须使用好现有数据资产，使数据产生价值，为银行提供二次产能[2]。

银行业金融机构要想“快、准、狠”，首先需要更加有效的客户资源竞争能力，就需要产品创新、正确决策提供准确数据，利用数据计算，挖掘潜在低价值客户，即忠诚度不高、贡献度不高的客户[3]，找到此类客户兴奋点，制定有针对性的营销手段。其次是需要创新更快见效的产品，实现快速回收资金及收益，从而提高市场占有率，这对银行在市场深入分析及了解提出了新的标准，而用户画像能有效提供客户特征，为产品创新提供支撑条件。市场的根本是客户，是与客户匹配的数据，是银行业金融机构必须去挖掘的“金山银山”。只有了解市场、了解客户、了解竞争对手，才能制定出物美价廉的产品，获得等多的市场青睐[4]。最后是需要突破传统并建立新规则，传统银行业金融机构想了解用户，更多的是选择依靠客户经理，依靠人与人的沟通，本文研究的用户画像系统，是打破客户经理与客户资源绑定规律的基础要素之一，任何人都可以通过用户画像系统，快速了解客户，而最终目标是通过各类系统及产品，实现客户与银行绑定，实现真正意义的高用户忠诚度，此用户忠诚于系统，忠诚于银行的产品和服务，而不受限于人。

1.2 国内外历史与现状

互联网行业率先提出并应用了用户画像技术，该概念最早源自交互设计之父 Alan Cooper[5]提出的 Persona 概念：“Personas are a concrete representation of target

users.” Alan Cooper 解释 Persona 是真实用户对应的虚拟用户，是建立在一系列真实数据（Marketing data,Usability data）之上的目标用户模型。

1.2.1 基础概念

用户画像系统最初由 California 大学的 Syskill&Webert 通过显示的收集用户对当前页面的满意程度逐步学习，建立用户的兴趣模型。后来，由 CUM 大学开发的 Web Watcher，面向的不是个体用户而是群体用户，它使用数据采集器，从而记录用户在互联网上的各种行为模式与各种浏览行为及用户的喜好，并且利用这些统计信息来统计得出用户群体的浏览行为，并对用户的兴趣模型进行构建。WebWatcher 更进一步的贴近我们现在针对用户画像系统的定义[6]。

1.2.2 发展现状

在国外，以 Facebook、Twitter 等为代表的主流社交网站，及 Ebuy、Amazon 等为代表的商城类网站，都已拥有了成熟可靠的用户画像应用。在我国，用户画像在电商行业、电力行业及三大运营商都有较为成熟的应用。在电商行业中，京东、阿里这些以 TB 计的高质量、多维度的数据记录着用户大量的网络行为，用户画像就是对这些数据的分析而得到的用户基本属性、购买能力、行为特征、社交网络、心理特征和兴趣爱好等方面的标签模型，从而指导并驱动业务场景和运营，发现和把握在海量用户中的巨大商机。

美国的亚马逊、中国的天猫商城、京东商城、苏宁易购等大型电子商务网站都在对用户进行建模，为用户推荐其可能的喜欢的商品。新浪微博、腾讯微博等社交平台中，也提供按照用户兴趣显示内容链接的服务。

用户画像首先在互联网公司发展起来，是基于互联网数据的开放性、竞争性及互联网公司开发灵活性。随着社交平台针对用户画像后，实现了对用户准确推送其兴趣内容，到后来电商行业借助社交平台的画像能力展开营销推广，再转变为电商行业自行根据用户消费习惯、消费能力进行主动画像与营销。用户画像基于大数据分析，驱动行业主动了解用户，使用户得到了良好体验，同时推荐消费、关联消费、兴趣消费等新型消费因素带来的经济价值，又促进了行业公司不断优化数据分析与用户画像模型，形成了良好的循环机制。

银行业机构没有互联网公司的灵活优势，但有可靠的数据来源。可靠的数据来源能够更加真实的反应客户特性，结合数据挖掘与分析技术，推断出其数据背后的其他特性，是目前国内很多银行已经或正在做的事情。用户画像系统已在银行业中兴起，基于银行可靠的数据，利用业务系统逻辑，结合大数据相关技术手

段，即可实现对银行客户进行用户画像。有了用户画像，使银行主动了解客户，从而提高银行的产品开发准确度、销售水平，促进银行业务发展。

以上案例都基于数据挖掘、分析建立了用于用户模型构建或者用户画像，越来越多的公司意识到以客户为中心的经营方式的重要性。目前，随着行业竞争的不断加剧，产品服务的同质化，客户忠诚度也在逐步降低，导致用户流失率也越来越严重。因此如何对用户进行建模，如何了解用户并留住用户变的越来越重要。综上，从国内外学者的研究成果和应用情况可以总结出，用户画像其实就是通过对客户多维度数据的挖掘、信息的分析[7]，将各类信息结合在一起，形成一定类型上的独特特征与气质，通过标签定义，最终以表格或格式化数据的形式呈现出来。需要指出的是，用户画像不是一成不变的，画像应根据时间、空间的推移，而发生变化，与真实的人相对应。

1.3 用户画像系统解决的问题及实现目标

基于以上的研究，本课题将围绕用户画像系统相关技术，对四川省内某家地方性中小银行为目标开展相关技术研究。该银行地处四川偏远落后地区，如何面对电子银行、互联网银行的新浪潮，适应大环境及各种新兴技术的快速发展和广泛应用，将银行科技发展战略与大数据、云计算等新技术相结合。如何将以自身业务为中心逐渐转向以客户为中心的发展模式、主动了解客户、提高客户忠诚度、开发适合市场不同群体的各类产品[8]等。以上该行考虑的问题，更是全国各类中小银行同样面临的问题。本课题基于以上问题，展开基于该银行数据的用户画像系统设计与实现的研究工作。

本课题主要借鉴现有的用户画像理论研究成果，结合大数据相关技术，主要研究内容及目标为以下几方面：一是通过对该行综合情况分析，分析该行现有的各类系统情况，开展对已有数据的分析、梳理，并制定整合及进一步挖掘的明细需求，为建立一套清晰分析逻辑的大数据系统提供需求依据。二是根据业务需求，开展用户画像系统的设计研究。通过对大数据技术、数据分析技术、用户标签体系及展示系统技术的相关理论和方法介绍，结合业务需求，自底向上完成用户画像系统的架构设计工作。三是通过搭建、开发和实现用户画像系统，研究具体大数据、数据计算、用户标签化及系统展示等问题。本论文重点为大数据架构搭建、数据的处理与分析、标签体系建设、应用及归类，通过将该行核心系统、信贷系统等主要业务系统的数据采集、分析、处理后，再借助前端数据可视化系统，最终实现一套完整的用户画像系统的目标。

该目标实现的意义在于，使该行的系统用户能够在使用系统时，可根据不同

类别标签、进一步人工分析，为业务创新及金融产品开发、甚至对优化客户营销[9]、提升客户关怀、强化风险管理等方面工作提供支持，提高整体工作绩效。最终通过产品与服务，在竞争激烈的当下[10]，实现留住老客户、争夺新客户，争夺更多市场资源，提高核心竞争力，从而提高盈利水平。

1.4 论文结构安排

本文的章节结构安排如下：

第一章为本论文的开篇绪论，引入了用户画像的研究背景和意义，简单介绍了国内外历史及基础概念与发展现状，最后按照需求、设计、实现三个大阶段表明了本论文计划解决的问题及实现目标。

第二章为本课题的用户画像概念和大数据技术基础概念，简要阐述了用户画像概念在国内外学者中的研究情况，同时介绍了大数据技术及 Hadoop 相关组件的基础理论知识，最后分析了金融行业的用户画像应用情况。

第三章为用户画像系统的需求分析，带领该行项目组成员开展了需求调研与分析工作。该章节明确了用户画像要做什么，用户想要什么。将主要从系统建设目标、需求概述、需求建模及非功能性需求展开描述。

第四章是用户画像系统的具体设计章节，基于需求分析，与项目组各成员确定了用户画像系统要做成什么样子，做到什么程度。将主要从用户画像系统设计目标、设计原则、系统整体结构设计展开介绍。

第五章是用户画像系统的实现部分介绍，通过搭建大数据平台，根据需求、设计开展了数据采集与分析工作。该章节重点对软硬件平台搭建、系统数据采集层、业务处理层开展介绍，对数据可视化层及测试与效果进行展示。

第六章是用户画像系统的总结与展望，该章节总结了本论文成果，指出论文成果到现阶段的不足与欠缺之处，并计划下一步深入研究及持续优化进行展望。

第二章 用户画像和大数据技术理论概述

第一章介绍了本课题的研究背景和意义，简要说明了国际国内用户画像系统的发展水平，同时分析了研究方向及希望解决的问题，最后阐述了论文的章节编排结构。本章主要介绍用户画像和大数据技术理论概述，将从用户画像概念入手，引入大数据技术概念介绍，最后分析了该系统在金融行业的应用情况。

2.1 用户画像概念

从国际角度来看，互联网行业率先提出并应用了用户画像技术，该概念最早源自交互设计之父 Alan Cooper《About Face：交互设计精髓》于 1983 年提出的 Persona 概念。G. Amato[11],R.M. Quintanan[12]等将用户画像意思理解为“通过大量数据分析获取的、由用户信息基本组成的形象集合”，通过这个集合，可以进一步分析用户的需求、用户个性化兴趣等等。信息提供者的最终目标是满足用户的信息需求。也就是说，通过正确的方式、在正确的时间、为用户提供正确的信息，而发展个性化服务的先决条件是依赖于通过用户画像文件分析，得出的代表用户需求的信息。S. Gauch,M. Speretta[13]等也将用户画像理解为一种集合，但他们理解的集合与上面两位学者的理解有所不同，其主要区别在于组成集合的成分认知不同，S. Gauch,M. Speretta 认为应该包含“keyword profile,semantic net profile,concept profile”即由关键字属性、语义属性、概念属性组成用户画像。D. Travis[14]提出了用户画像的 7 个组成要素，Primary（基本性）、Empathy（同理性）、Realistic（真实性）、Singular（独特性）、Objectives（目标性）、Number（数量性）、Applicable（应用性）[15-16]，并使用这 7 个要素的首字母排列组合成 Persona 这个英文单词，翻译过来就是现在我们研究的用户画像一词。

从国内角度看，根据王宪朋[17]、余孟杰[18]等学者的分析，可将用户画像的定义划分为三层，第一层也是最底层，应为数据采集层，该层是构建用户画像的基础和必要条件；第二层为中间层，该层可以是抽象层也可以是真实层，与数据底层和上层展示层做交互，实现用户画像的具体业务逻辑，负责展示业务特色；第三层为最上层，通过数据可视化工具或系统，对中间层已实现的数据结果、模型实现可视化展现。最终这个数据可视化的结果就是用户画像。

2.2 技术概述

本小节将介绍利用大数据技术[19]，结合业务需求，搭建大数据环境，实现对

结构化、非结构化数据的采集、归集、存储和后期可视化处理。

2.2.1 Hadoop 及 Hadoop 架构分析

本论文大数据方面主要使用的技术应用为 Hadoop 及其相关组件。本小结将先对 Hadoop 概念、功能及其架构进行分析介绍。

Hadoop 是 Apache 软件基金会旗下的一个开源分布式计算平台[20]，其源头项目是 Apache Nutch 搜索引擎项目，该项目于 2002 年开始，是 Apache Lucene 的子项目之一。2004 年 Doug Cutting 等人基于 Google 的 Operating System Design and Implementation 会议时发表的 MapReduce：Simplified Data Processing on Large Clusters 论文，开始展开对 MapReduce 的研究，并将其与 NDFS 相结合。2006 年 2 月，NDFS 与 MapReduce 在 Nutch 引擎中已经运行良好，成为了一套完整而独立的软件，起名为 Hadoop。Hadoop 其实为分布式文件系统，最初核心包含 HDFS（Hadoop Distributed File System）和 MR（MapReduce），现在 Hadoop 已经形成了自己的生态圈[21]，拥有众多丰富功能的组件，让用户可以根据自身需求自由组装，实现不同的功能。

Hadoop 使用 HDFS 分布式存储方式，在提高了读写速度的同时，扩大了存储容量，结合 MapReduce 计算核心，实现高效的分布式数据的并行计算和处理。所以，Hadoop 的优点主要为：高可靠性、高扩展性、高效性、高容错性[22]。

Hadoop 的 HDFS 分布式文件系统框架，如下图 2-1：

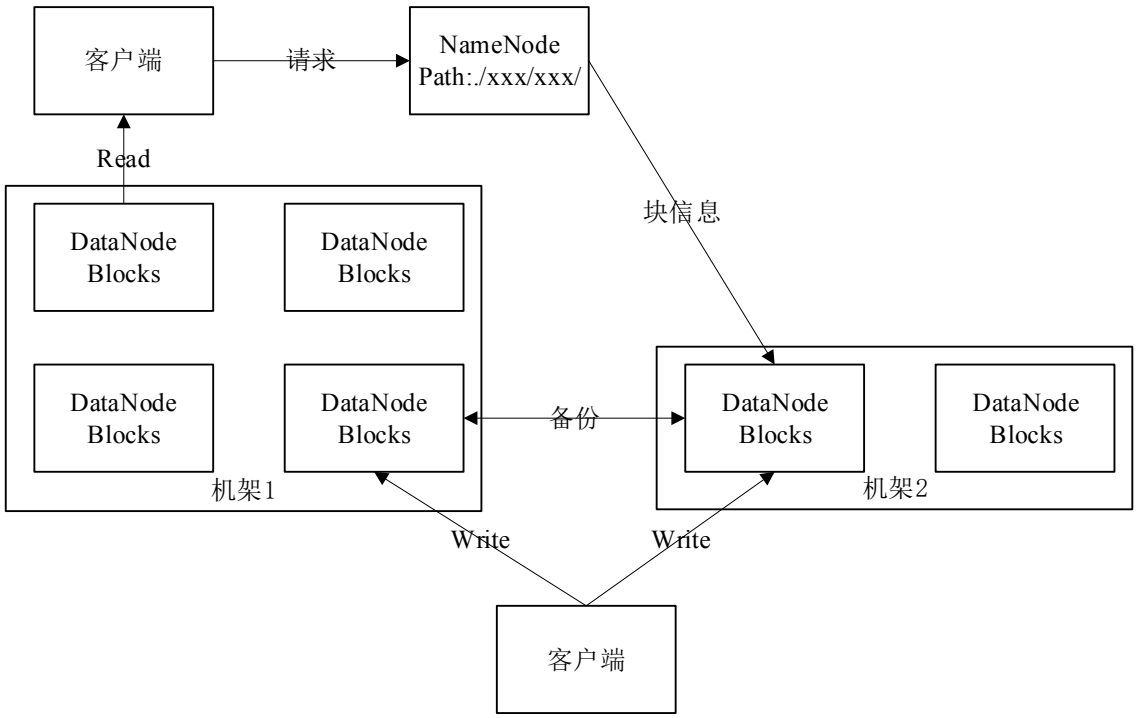


图 2-1 Hadoop 文件系统架构

HDFS 分布式采用 Master-Slave 模式结构，从图 2-1 可以看出，一个 HDFS 集群是由一个主节点即名称节点和 N 个从节点即数据节点组成的，图中 NameNode 扮演了 Master 角色，DataNode 扮演了 Slave 角色。

NameNode 作为主服务器，仅存储目录结构及元数据，允许客户端用户以文件的方式开展增删改查的读或写操作，各类操作均被记录到日志中[23]。该文件从 HDFS 内部来看，会被分割为若干个数据块 Block，存储到 DataNode 中，最新 V2 版本的 Hadoop 文件分割默认大小为 128M。DataNode 中存储 Data Block，在 V2 版本的集群环境中，默认的建议副本数值为 3，所以当 Data Block 被增删改查时，不会做日志记录，仅作副本备份及同步操作。

NameNode 主要操作两个数据表的映射关系，一是 Namespace 与 Block 之间的映射关系，该映射关系被持久化在硬盘中，另一个是 Block 与 DataNode 的映射关系，此映射关系取决于每次启动的 DataNode 的就绪先后，仅存储与内存中。

DataNode 的副本策略决定了 Hadoop 的高容错特性，副本策略可以由应用程序指定、创建时参照配置指定还可通过后期修改。NameNode 周期性的接收来自 DataNode 的状态心跳报告，从而掌握各 DataNode 的状态，以便于新数据写入时，决策具体的 DataNode 的分配。DataNode 的分布参考均衡策略，避免造成某节点的压力过大或数据过重，以保障吞吐性能，其样例见下图 2-2。

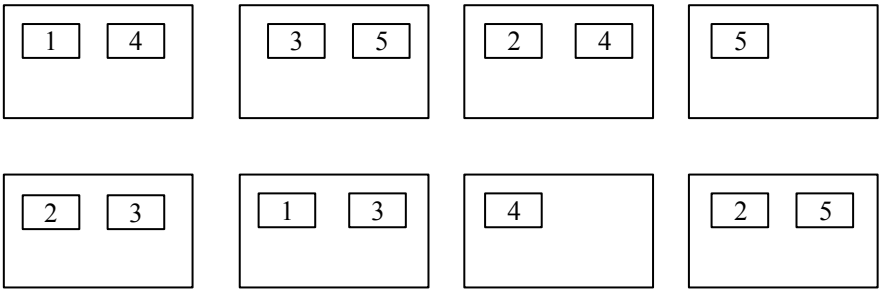


图 2-2 HDFS 副本节点策略图

Hadoop 从 2006 年正式面世，发展至今，已包含了多个子项目的集合，各集合与 Hadoop 相互依存、相互补充，共同构成了 Hadoop 生态圈，见图 2-3，本小节将针对用户画像系统使用到的相关技术组件展开介绍。

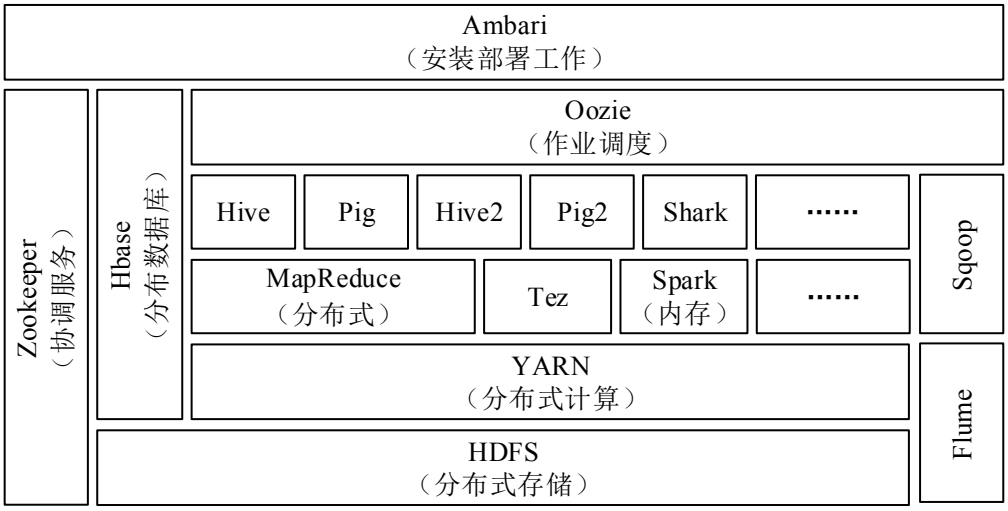


图 2-3 Hadoop 生态圈

2.2.2 Zookeeper 介绍

Zookeeper 是 Hadoop 生态圈中的组件之一，主要为了解决分布式服务冲突问题，为用户提供同步、配置管理、分组和命名等服务。分布式环境下的程序和活动为了达到协调一致的目的，Zookeeper 可以将分布式配置文件集中存储，架构如下图 2-4，当程序需要调整配置文件或目录时，不用再依次变更，Zookeeper 会将变化通知到每个节点。

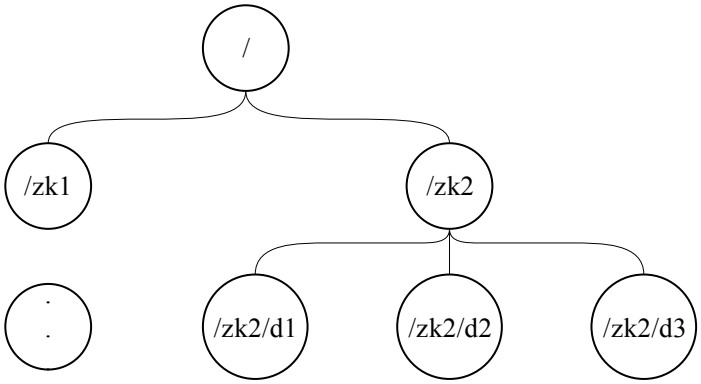


图 2-4 Zookeeper 目录管理架构

这些节点类似树状结构，每一个 Node 节点都进行数据维护、访问控制、时间戳及包含交换版本号数据结构的操作，执行协调及更新操作。

2.2.3 HBase 介绍

HBase 是 Hadoop 生态圈中重要角色之一，可以理解成 Hadoop 生态圈中承担数据库角色的组件，特点实时读、写、访问非常快。

HBase 一改传统数据库的模型，使用了面向列的方式。可以与 Hadoop 的 HDFS

文件存储架构[24]。HBase 存储是松散型的结构，其数据关系介于映射(Key-Value)与关系型数据之间，从逻辑上看就像一张巨大的表，可以根据需求动态增加。本论文选择 HBase 组件，主要为解决一些实际业务需求中的应用场景：一是数据量较大，并且访问需要满足随机、快速响应的需要，二是需要满足动态扩展的规划，三是不需要满足关系型数据库中的特性，如事务、连接、交叉表等，四是写数据时，需要拥有高吞吐的能力。

2.2.4 Hive 介绍

Hive 是 Hadoop 生态圈中重要的计算组件之一，利用 MapReduce 编程技术，实现了部分 SQL 语句，提供类 SQL 的编程接口[25]。Hive 基于 Hadoop 的 HDFS 可以提供数据仓库文件类型的架构，提供数据 ETL 过程（ETL 是数据抽取 Extract 、转换 Transform 、加载 Load 的简写）、存储管理和大型数据集的查询与分析能力。

Hive 与 HBase 相比，Hive 更突出计算能力，而不是实时服务的能力，在本论文中，Hive 将主要使用于逻辑中间层。

2.2.5 Flume 介绍

在本课题中，实际数据源并非全部是结构化数据，也并非全部都存储在数据库中。要想把这些非结构化的、非固定格式的文件存储到 Hadoop 的分布式文件系统上，就不得不用到 Flume 这个日志采集服务组件了[26]。

Flume 与其他大多组件相同，拥有分布式、高可用、高可靠的特性，它实现了不同的海量数据收集、传输、存储到一个分布式文件系统中。Flume 框架轻量、配置简单，适用于收集各种日志，并且支持故障漂移和负载均衡。

Flume 逻辑上可以理解为三层架构，分别是代理 Agent、链接 Collector、存储 Storage 层。Agent 层用于具体与各类数据接触，将产生的数据流传输至 Collector 层，Collector 汇总各个 Agent 的数据流，加载到 Storage 中，多个 Collector 之间遵循负载均衡原则。Storage 更靠近文件系统，接收 Collector 上传的数据流后，以文本、文件甚至 HBase 的方式存储。

Flume 的核心架构见图 2-5，由 Source、Channel、Sink 三个组件组成，Source 负责收集数据，传输给 Channel，Channel 中转临时数据并保存，Sink 从 Channel 中读取数据，最终发给存储介质，如 HBase、HDFS 等。

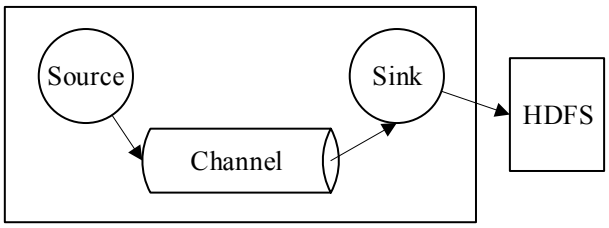


图 2-5 Flume 组件架构示意图

2.2.6 Sqoop 介绍

Sqoop 面向关系型数据库，解决了关系型数据库与 Hadoop 之间的数据传递问题[27]。Sqoop 使用 MapReduce 计算框架来完成数据的导入和导出，继承了 MapReduce 的并行能力与容错性，数据传输过程如图 2-6 所示。

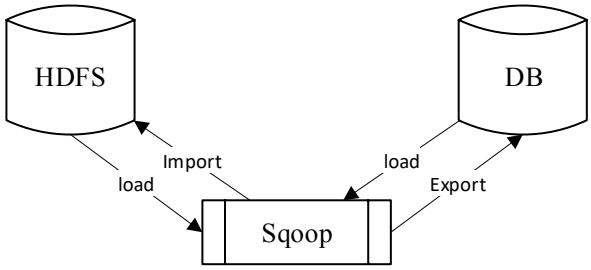


图 2-6 Sqoop 功能示意图

通过 Sqoop 可以读取 Load 关系型数据库 DB 中的数据，然后将读取的数据导入 Import 到分布式文件系统中 HDFS 中；也可也使用 Sqoop 读取 Load 分布式文件系统中的数据，导出 Export 至关系型数据库中。

2.3 标签算法介绍

本课题研究的重点在于如何将大数据抽象、分类后实现对用户的画像并通过系统实现画像的可视化。通过对大数据的采集与整合，结合抽取及清洗工作后，首先应使用权重算法对目标数据进行权重计算，获取权重高的目标标签，再通过分类算法根据权重标签，对数据进行分类计算，获取到同类标签相关数据。下面将简要介绍本课题使用的两种算法：TF-IDF 权重计算与 K-means 聚类算法。

2.3.1 TF-IDF 权重计算

数据采集及清洗工作完成以后，就需要对数据进行词频统计计算。本课题选用 TF-IDF 权重计算方法，实现对标签的权重计算。

TF-IDF 权重计算核心思想是：如果某个词语在一篇文章中出现的频率越高，同时在其他文章中很少出现，则认为该词语具有较高的权重值，具有代表性。TF

为 Term Frequency 词频，含义为在文章中出现的次数。IDF 为 Inverse Document Frequency 逆向文件频率，含义为总文章数除以出现过该词语的文章数，取其结果对数所得。

TF-IDF 的主要应用为，搜索引擎、关键词提取、文本相似性及文章摘要。

2.3.2 K-means 聚类算法

分类是根据数据的某些字段或属性，将样本类型归为同一类别中，但需要提前知道各种类别信息。当对大数据进行分类时，为了提高效率，本课题选用 K-means 算法。

K-means 算法核心思想是：根据给定的样本数据，事先确定聚类簇数 K，使得簇内样本数据尽可能的紧密靠近，而簇与簇之间的距离尽可能的远离。

K-means 算法步骤为，输入样本数据集 A，簇的个数为 K，最大迭代数设为 N：第一步，为 K 个簇选择一个初始聚类中心；第二步，将样本集数 A 据按照最小距离原则，分配到最邻近的聚类；第三步，使用每个聚类样本均值更新聚类中心；第四步，重复二三步骤，直到聚类中心值不再发生变化。通过以上几步，最终输出聚类中心与 K 个簇划分结果。

2.4 用户画像金融行业应用

用户画像的概念源于互联网，源于互联网公司对客户精准营销的需求，实现用户画像后，可以准确推送营销信息，提高订单转化率、挖掘新需求及获得新客户。

支付宝、微信都开展了较为成熟的用户画像应用，如 2017 年底支付宝推出的年度账单，为用户统计了 2017 年的全年支出以及网购、出行、手机充值、生活缴费、转账等不同分类的消费情况，进而“预测”出用户在 2018 年的消费关键词及个人标签。微信则推出《微信数据报告》，以不同年龄段为主要维度，对用户年龄占比、睡眠时间、出行方式、兴趣爱好及饮食偏好做了画像分析，提供了有趣且生活的标签体系[28]。

传统的金融行业包含银行、保险、证券三种机构，随着支付宝、微信等第三方机构引领起新的支付方式浪潮，银行业金融机构，必须快速适应互联网发展及支付方式变革，了解客户需求以保住市场份额。

各大型商业银行都先后根据自身的数据及业务需求，建立了使用于自身的用户画像系统。2018 年中国银行大数据应用平台发布，该平台建设了一套客户 360°画像的标签体系，涵盖个人客户、企业客户及外部数据等多达 1000 个左右的客户

标签，覆盖客户基础信息、兴趣爱好、社会属性、金融特征、客户价值和互联网特征等多种丰富的客户属性[29]。同年，华夏银行发布了其用户画像系统，根据其不同的支付结算渠道，对用户进行了人群分布、用户概况、金融产品偏好、渠道访问特征等维度分析及标签建模。

保险业金融机构则更加关注其产品销售的市场及风险程度，通过同时切入不同的消费场景，向用户提供保险服务来积累用户在不同消费场景下的信用标签，由此构建出全面而立体的用户画像。这些多场景、细颗粒度、实时动态的保单数据是帮助用户触达资金成本较低的、传统金融机构的重要信用数据。如车险客户可参考外部养车 APP 活跃信息；意外险和保障险可导入移动设备信息，找到户外运动人群和商旅人群；寿险、养老险、教育险等可依靠用户所处的生活环境及就医信息等。

证券业金融机构的服务用户与保险、银行的服务对象有求知欲强的明显区别[30]，一是对浏览器和综合资讯有很高的需求，二是因用户经常在移动端进行资金交易，用户安全意识高，十分注重移动端安全管理，三是证券投资用户年龄主要分布在 24-35 岁，此年龄段用户处于事业成长期，对投资理财需求较大，风险承受能力较强。所以证券业金融机构在制定用户画像策略时，主要考虑用户在理财方面有多方面需求，包含记账、购买理财产品、购买基金、P2P 投资、信用卡管理等等。

银行、保险、证券三种金融行业在构建用户画像时重点不同[31]，但仅以整个银行业金融机构来说，其内容及关注点是相似的。因为其上下游系统、客户信息、基础业务数据都大同小异，只是各家银行在建立用户画像系统初期时，因业务侧重、营销策略而选择不同，导致标签体系、管理及展示的暂时性区别[32-33]。随着行业发展、时间推移，银行业金融机构的用户画像系统应趋于标准化，仅存在各银行的主营业务特色、地域文化等很小的差异[34-35]。

2.5 本章小结

本章介绍了用户画像的基本理论，根据用户画像理论知识，选择以 Hadoop 为主要技术架构，并对 Hadoop、HDFS、Zookeeper、HBase、HIVE、Flume、Sqoop 逐一进行了基本介绍和解读，在使用 Hadoop 框架前，对 Hadoop 生态圈及组件有了初步的认识和了解。有了以上的基础理论知识和技术基础解读，下一章将基于本章的理论知识，开展针对广元市贵商村镇银行用户画像系统的需求分析。

第三章 用户画像系统需求分析

第二章介绍了本课题的用户画像概念和大数据技术基础概念，简要阐述了用户画像概念在国内外学者中的研究情况，同时介绍了大数据技术及 Hadoop 相关组件的基础理论知识，最后分析了金融行业的用户画像应用情况。本章节是本课题重点章节之一，也是重点工作之一，本小节进入用户画像需求分析阶段，将引导项目组成员开展需求调研、需求分析，最终完成需求编写与建模。

3.1 系统建设目标

广元市贵商村镇银行在国内银行中属于小微银行序列，地处四川省广元市，截至 2019 年底全市存款总量逾 1100 亿，该行存款规模过 100 亿，约仅占全市总量的 9%；广元市有常驻人口 266 万，其中是该行有效客户的近 18 万，约占全市人口总量的 6%。该行 100 亿存款中，个人存款 29 亿元、对公存款 71 亿元，个人存款占比仅为 29%，对公存款占比过高。随着国家逐渐收紧财政资金策略，整体经济形势较为低迷，对公存款组织困难，且维护成本高。广元市贵商村镇银行要想立足广元，在广元市甚至四川省银行中站稳脚跟，就必须要调整存款结构，扩大个人存款组织和营销。

调整储蓄结构是短期需求，但该行其根本需求是了解客户，掌握客户的基本信息、渠道使用偏好、消费能力及贷款需求等，以便各业务部门根据分类标签，挖掘空白市场、挖掘客户需求，从而研发出可以有效打开市场的新产品。

基于以上的短期与根本需求，结合大数据相关技术，对广元市贵商村镇银行的相关业务数据进行采集、归集、存储和清洗，利用结构化与非结构化数据整合技术，搭建用户画像系统，实现客户信息查询、分类、标签化，让系统使用者可以快速了解客户及客户群。

3.2 用户需求概述

本文用户需求主要来源于广元市贵商村镇银行的各主要业务部门和抽样支行。需求采集工作由科技部牵头成立的需求讨论小组负责，小组成员包含该行科技部、互联网金融部、公司业务部、授信评审部、三农业务部、嘉陵支行、利州支行及科技支行的成员代表，负责参与需求讨论、需求提出及需求整理。采集方式主要采用走访、调查问卷和会议方式。

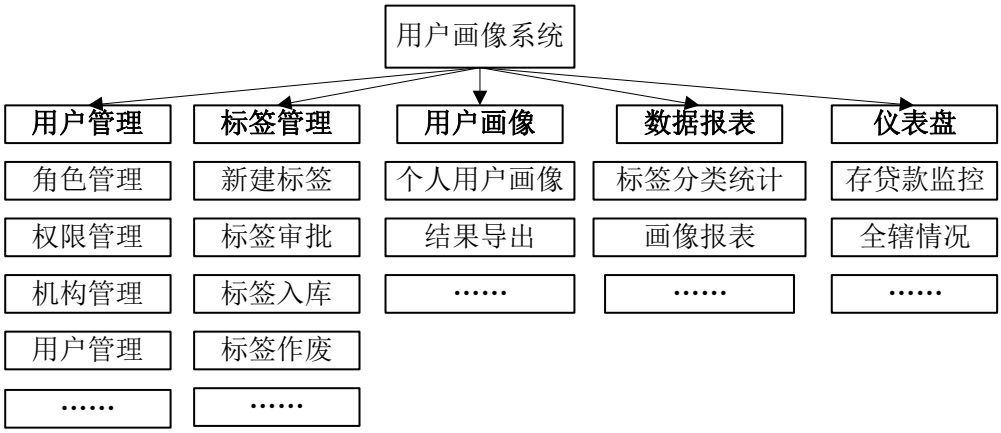


图 3-1 用户画像系统功能示意图

通过原型引导、市场现有系统分析学习，初步确定该行的用户画像主要功能结构图，见图 3-1，包含用户管理、标签管理、客户画像、数据报表、仪表盘等。

用户管理为公共功能，通过用户角色对应菜单权限，实现用户的基本权限划分与功能使用，后续小节将对其他模块需求展开详细介绍。

3.2.1 标签管理

标签是抽象的复合性词组，是从大数据中脱离出来的行为。银行业金融机构内部的信息分布在各个系统中，例如关于客户属性的信息存在存款考核或核心系统，贷款及信用类信息主要存在信贷系统、征信系统中，而消费类特征存储在支付渠道类与各产品系统中。

用户标签的形成，由原始数据清洗、结构化后，通过统计抽取技术，形成属性分类，再通过属性分类、词频统计等手段，形成基础标签、习惯标签、特征标签与其他标签。见图 3-2。

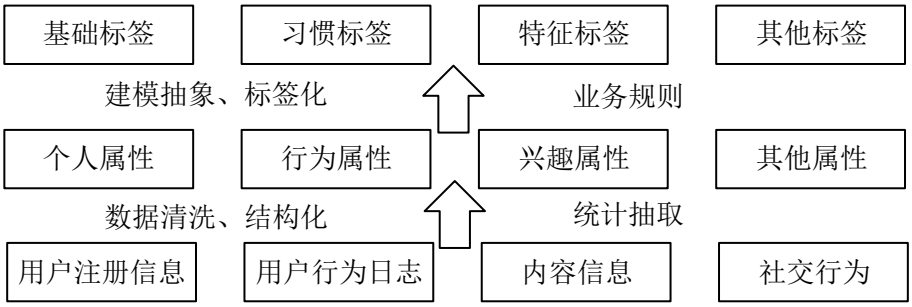


图 3-2 标签分类流程

该银行标签体系管理需求，主要包含新建标签、标签审批、标签入库及作废。标签可由具有该菜单操作权限用户进行新建，经标签审批员审批后，根据申请人部门级别入库生效。标签审批流程设计如图 3-3。

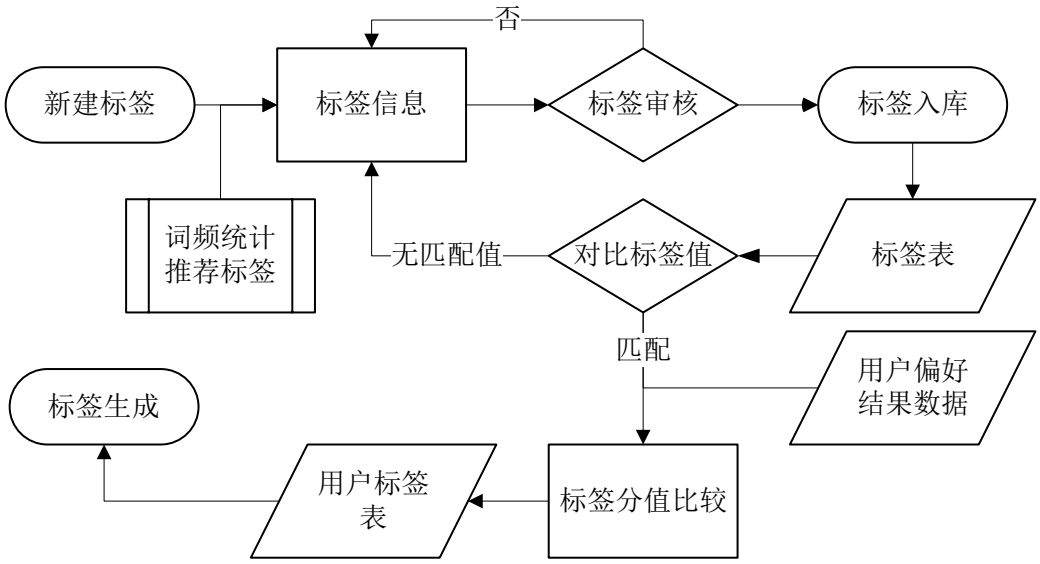


图 3-3 标签审批流程

标签可以由人工录入或者系统跑批推荐入库，系统跑批入库针对关键系统的重要信息表进行词频统计，得出词频排名靠前的结果，并将该结果插入标签审批表等待审批。人工录入的标签应由系统管理员或部门总角色审批，将数据结果存入标签结果表，该结果表与用户偏好数据结果数据进行匹配计算后，根据权重、阈值等计算参数，形成用户画像，用户画像标签将存储至标签结果表以供查询、调用。

3.2.2 用户画像

前小节重点介绍了标签如何产生与管理，本小节将重点介绍用户画像的标签如何设计与分类。用户画像实际是将标签向用户靠拢的过程描述，是站在用户维度的行为，标签是客户群体的高度浓缩性词汇，具有独特性同时兼顾代表性。本课题设计的用户画像功能为：输入用户姓名、身份证号、电话号码等唯一标识，通过用户画像系统查询到该用户的标签结果，通过图表的形式展现出该用户描述信息。

该银行需求参照人生理财的几大阶段，将客户按照年龄区间分为形成期 25 至 35 岁、成长期 30 至 55 岁、成熟期 50 至 60 岁、衰老期 60 岁以上等阶段。基于各人生阶段的金融需求的不同，便可以为寻找目标客户时提供目标定位。再结合客户的消费、收支及学历、资产负债等信息情况，可将客户进一步分为低、中、高几级，提供其需要的不同金融服务。还可参考消费记录和资产信息，以及交易产品，购买的产品，将客户消费特征进行定性描述，例如户外爱好者，奢侈品爱好者，科技产品发烧友，摄影爱好者，高端汽车需求者等信息。

用户画像需求的根本是帮助银行将复杂数据抽象化，通过数据处理与分析，将交易数据定性后分类，结合分析要求，对数据进行价值附加，最终目的是使系统使用者通过系统快速了解客户，如图 3-4 所示。

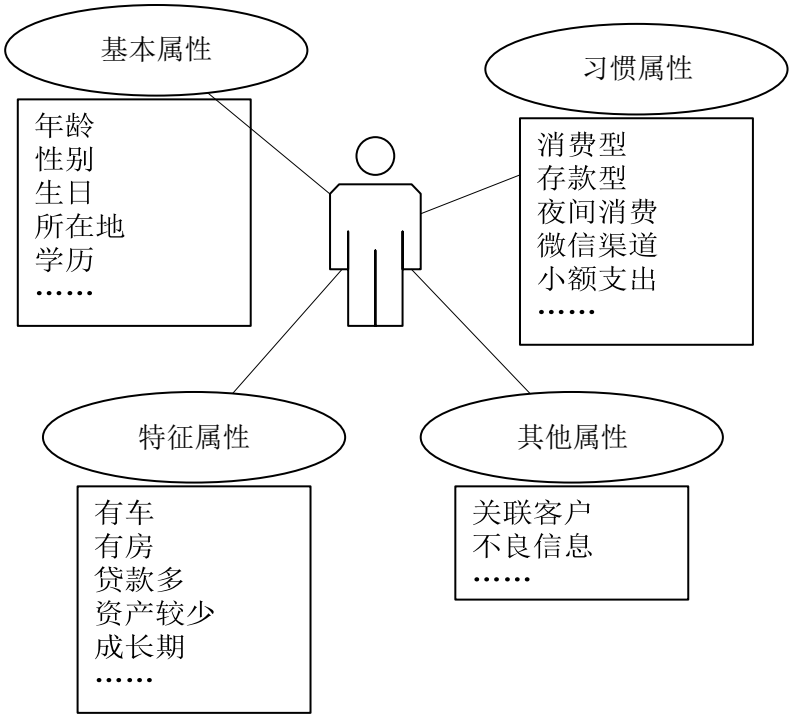


图 3-4 用户画像属性示例图

用户画像需求还包括通过特征属性、标签，使客户经理快速找到包含某特征的目标客户群，这个过程是将客户向标签靠拢的描述。通过标签查询客户，可以进一步精准营销，提高转化率。例如利用发卡数据、物业费代扣交易数据（别墅／高档小区）、稳定的交易额度流水、车贷信息数据或加油数据，可挖掘出在我行资产较少、消费额大于资产的用户，可判定为优质客户，为其提供高端资产管理服务。

为了实现用户画像，对基础定性信息进行分类是一个重要的工作环节，有效的信息分类使用户画像实现真正的需求转化。根据银行需求，至少包含用户的收支记录、TopN 的标签图，以及客户的基本信息，包含姓名、性别、电话、地址、消费偏好、人生阶段分析、产品推荐等。

用户画像查询完毕后，可将结果导出，形成 pdf 或者格式化电子表格，供客户经理进一步分析使用。需要注意的是，用户画像的结果应随着时间的推移、数据的堆积及其他因素的变化而变化，用户画像由大数据及标签库提供基础画像，经该行各条线业务专家不断对标签进行调整，不断优化数据计算规则，用户画像趋于准确平均值。

3.2.3 数据报表及仪表盘

数据报表为该行的自身企业画像，也是将客户向标签靠拢的过程，是站在标签角度的行为。

数据报表统计该行下辖的各分支行的存贷款情况、不良贷款占比、支行排名等信息。该行数据报表主要是为了满足管理与监控需求，目的是让总行领导、部门领导对各支行的经营情况能够实时查看、掌握第一时间数据，解决现在数据统计归口过多、口径不一甚至数据错误、造假的问题。

仪表盘的需求是以客户为中心，根据支行、渠道维度，统计客户的交易情况，按照各标签单独或组合统计的不同维度报表展示，以数字化、柱状、饼状等分析类图标为主。全辖标签情况主要展示各标签的使用情况、频率、词频数等相关数据，监控仪表主要展示广元市贵商村镇银行全辖营业情况，根据基础数据对自身画像，展示存款数、贷款数、营业柜台数、各渠道交易笔数等，效果图如 3-5 所示。

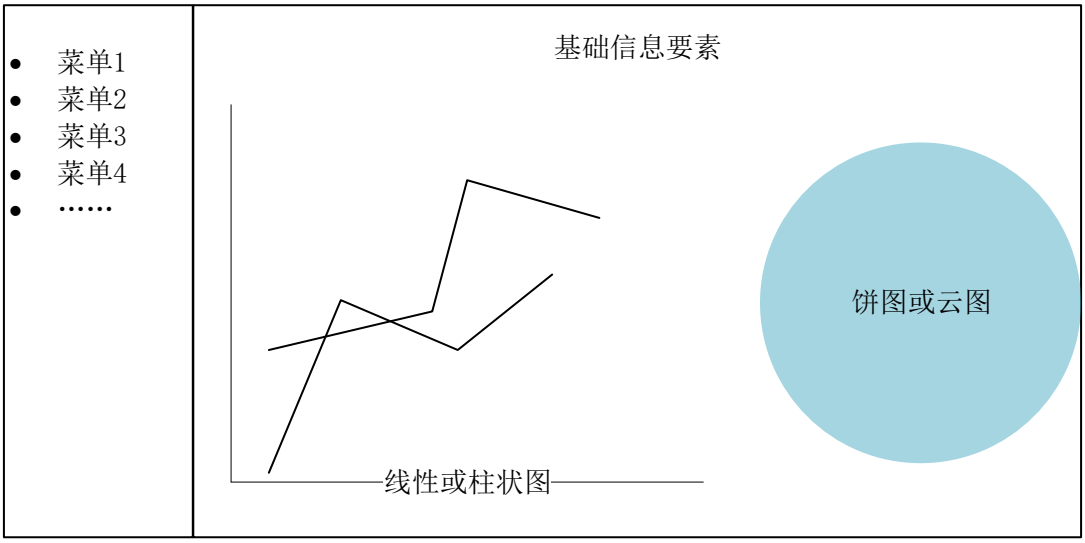


图 3-5 仪表盘示意图

3.3 系统需求建模

3.3.1 标签数据采集建模

用户画像需要建立标签体系，标签体系需要建立大数据平台数据仓库支撑，大数据平台将在具体设计第四章节介绍，本小节将对标签管理建模进行介绍。

标签管理需要数据支撑，用户偏好结果表可以设计为次数表或权重表，根据贷款系统的贷款信息相关表、核心系统交易流水相关表、网银系统日志表、短信系统日志表等，统计出用户对应的偏好值。标签系列的数据通过系统分析计算后，经审批入标签表，最终根据标签结果表与用户偏好值表计算得出该用户的匹配标

签。

本课题对该行全部系统进行了数据筛选，最终确定采集范围包含：核心系统、银行卡系统、网银系统、信贷系统、银联渠道、微信、支付宝、大小额渠道、短信系统等系统的数据库表及日志数据，如表 3-1 所示。数据采集规则根据每个系统数据格式、内容的不同，采取不同的采集规则，除初始化需导入全量数据外，其余日期规则主要分为 T+1 日采集及 T+7 日采集两种模式，按数据结构类型主要分为结构导入、非结构导入方式。

表 3-1 系统及数据分类表

系统类型	系统名称	数据库	数据类型
交易类	核心系统	Informix	库或文本
	银行卡系统	Informix	库或文本
	网银系统	Oracle	库或文本
渠道类	银联	Log	报文或日志
	大小额	Log	报文或日志
	财付通、支付宝	Log	报文或日志
信贷类	信贷系统	Informix	报文或日志
	闪贷	MySql	库或文本
其他类	短信系统	Oracle	库或文本

数据类型以系统重要程度区分，分为核心类业务系统、渠道类系统、外围类系统。该行的主要核心类业务系统包含：核心系统、银行卡系统、网银系统及信贷系统，以上各系统数据采用 T+1 日采集模式，数据类型为结构化数据。其中核心系统拥有柜面交易、账务信息、核算数据明细等，银行卡系统拥有卡类信息、绑定及部分签约关系等，网银系统拥有网上客户信息、网上交易信息等，信贷系统拥有贷款客户详细信息、客户资产信息、贷款签约信息、贷后信息等。

该行的主要渠道类系统包含：银联渠道、微信渠道、支付宝渠道、大小额渠道及四川省同城支付系统，为该行的转账、汇兑及结算渠道，采用 T+7 日采集模式，数据类型以结构化为主，有部分非结构化日志数据参与。此类渠道包含了渠道交易信息、商户信息、对手信息等。该行的主要外围业务系统包含：个人征信系统、企业征信系统、短信系统、存款考核系统及历史流水查询系统，采用 T+1 日采集模式，数据类型为结构化数据，有部分非结构化日志数据参与。此类系统主要功能以报送、查询及补充核心类系统功能为主，包含了征信类信息、贷款信用情况、短信内容、存款日均、历史交易流水等，数据采集图流程见图 3-7。

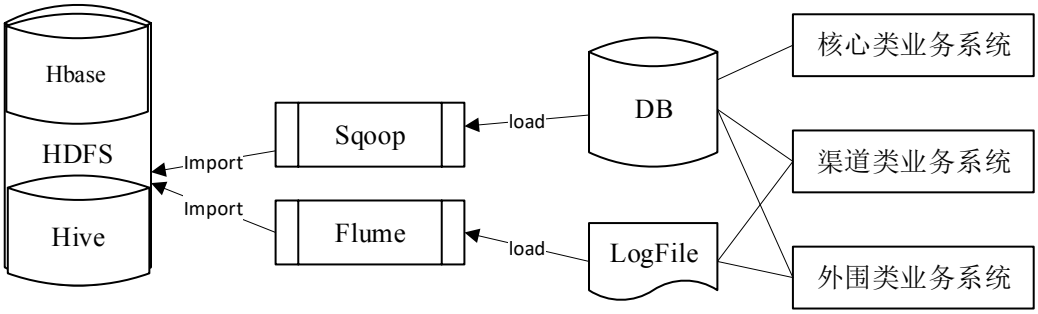


图 3-7 Sqoop 与 Flume 的具体使用

数据导入规则主要按照结构化、非结构化进入大数据平台，借助 Sqoop 导入结构化数据、Flume 导入非结构化数据，数据从源头采集后，通过清洗转换等操作，进入 HDFS、HBase 及 Hive[36-38]。数据处理及转换主要为了过滤不合格数据，包含脏数据、逻辑错误数据、不完整数据等。处理及清洗规则主要由业务部门先按逐项业务制定，再根据此种业务相关联的实体表数据的实际情况进行筛查，最终优化完善清洗规则后确定。良好的清洗规则是提高转换效率的有效手段之一，转换的目标主要目的是统一各业务系统、渠道、平台的数据标识，并实现统一编码，便于后期系统查询、数据运算及进入业务处理层的分析计算、挖掘等工作流程。

3.3.2 画像标签计算建模

画像标签计算建模依托于数据层数据采集，属于业务处理层，该层不负责实现对数据的采集动作，主要实现对已收集数据的处理、分析与结果输出。标签的分析、计算与生成，数据采集处理完成后，根据人工初步定义的特征标签词典，开始进入数据分类、关联、聚类分析。首先需要对数据进行切词，可以利用词频计算等方式，初步得到分词词典。其次根据词典，判断各个特征词语在某一客户数据中的权重，因为各词语在每个客户身上的权重是不一样的，此处使用 TF-IDF 计算权重，生成标签结果。当有一定对象或元组数据后，采用 K-means 算法对生成的用户画像结果进一步进行聚类分析。

数据分析主要依赖 Hadoop 生态圈的相关计算组件，如 Hive、Spark 等，通过搭建此类组件，不用关心底层 MapReduce 等运作机制，只需要编写核心计算代码即可，能够将更多的精力投入到挖掘与分析工作中。

业务处理层主要规划实现三大功能，一是针对结构化表数据、非结构化日志数据的分析与计算，将用户属性与偏好进行打分。二是计算标签，将库表数据中标签来源字段进行分词、词频统计，形成结果后，将此结果反馈保存至标签结果库。通过编写函数对逐个用户的数据进行计算，形成用户标签。三是用户偏好与标签匹配计算功能，通过标签与用户偏好的阈值计算匹配，得到用户画像结果。

最后是数据统计与仪表盘功能，是针对广元市贵商村镇银行自身数据的分析与统计，将定期需要查询的报表或实时数据，通过数据可视化系统展现出来，功能整体流程如下图 3-8 所示。

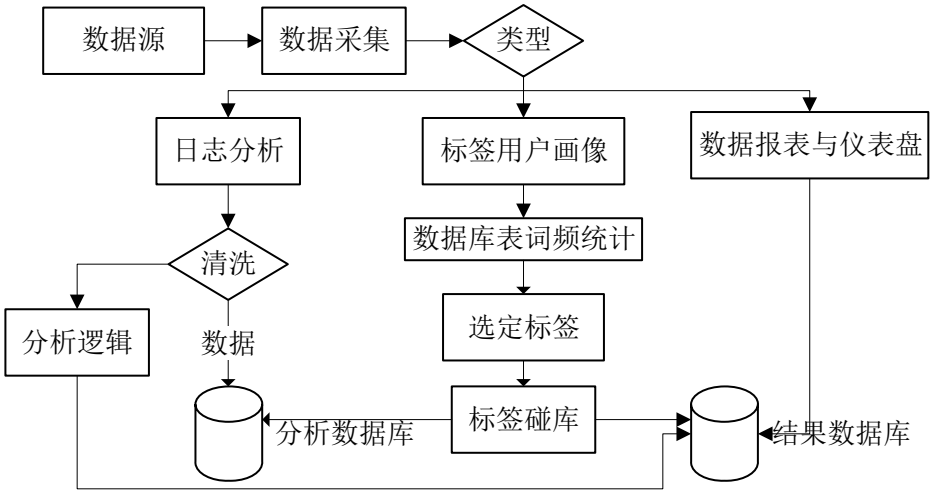


图 3-8 数据采集到标签形成流程图

为计算出用户偏好标签表，需要在用户行为标签的基础上计算用户行为标签对应的权重值，然后将同类标签权重归类汇总，算出用户偏好的标签。

偏好标签算法使用：用户标签权重=行为类型权重×时间衰减×用户行为次数×TF-IDF 计算标签权重。行为类型权重参数含义为用户浏览、搜索等不同行为对用户而言有着不同的重要性，一般而言操作复杂度越高的行为权重越大。该权重值业务人员或数据分析人员主观给出；时间衰减参数含义为用户的行为会随着时间的过去，历史行为和当前的相关性不断减弱；行为次数含义为记录用户在同一天中，同一行为的次数。

偏好标签算法解决了用户标签问题，而聚类算法可对不同特征属性的用户进行聚类，可以更好的了解和对客户群分类。K-means 算法是一种划分式聚类算法，在大数据运算时有高效性，其计算流程如下图 3-9 所示：

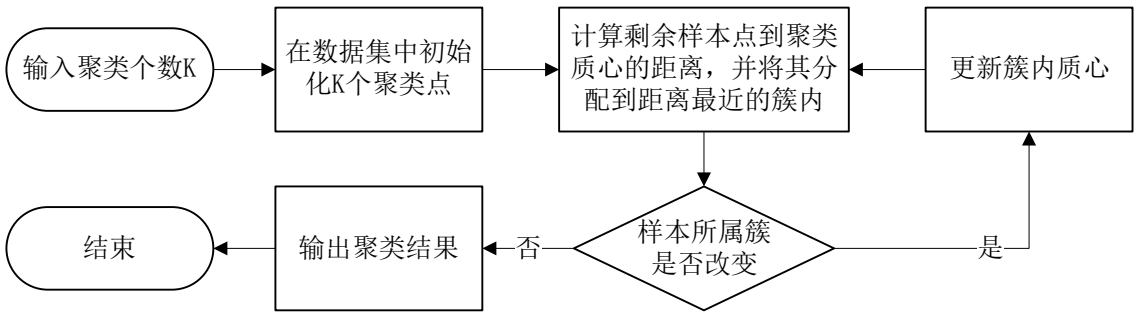


图 3-9 K-means 算法流程图

3.3.3 展示层建模

展示层是面向用户的接口，主要面向广元市贵商村镇银行总行各营销及管理职能部门。超级管理员具有公共功能包含用户、权限、角色管理功能及所有模块的权限。其次用户用例中，客户经理可以使用个人用户标签、企业用户标签、标签营销数据功能，部门总为总行各业务部门负责人，可以使用客户经理已有权限功能及标签管理功能模块。支行行长与总行行领导功能模块权限一致，只是数据展示上有所区别，总行行领导查看的为全行数据，支行行长查看的是其辖内支行的汇总数据，结果报表及仪表盘的有权人员，UML 角色用例图如 3-8 图：

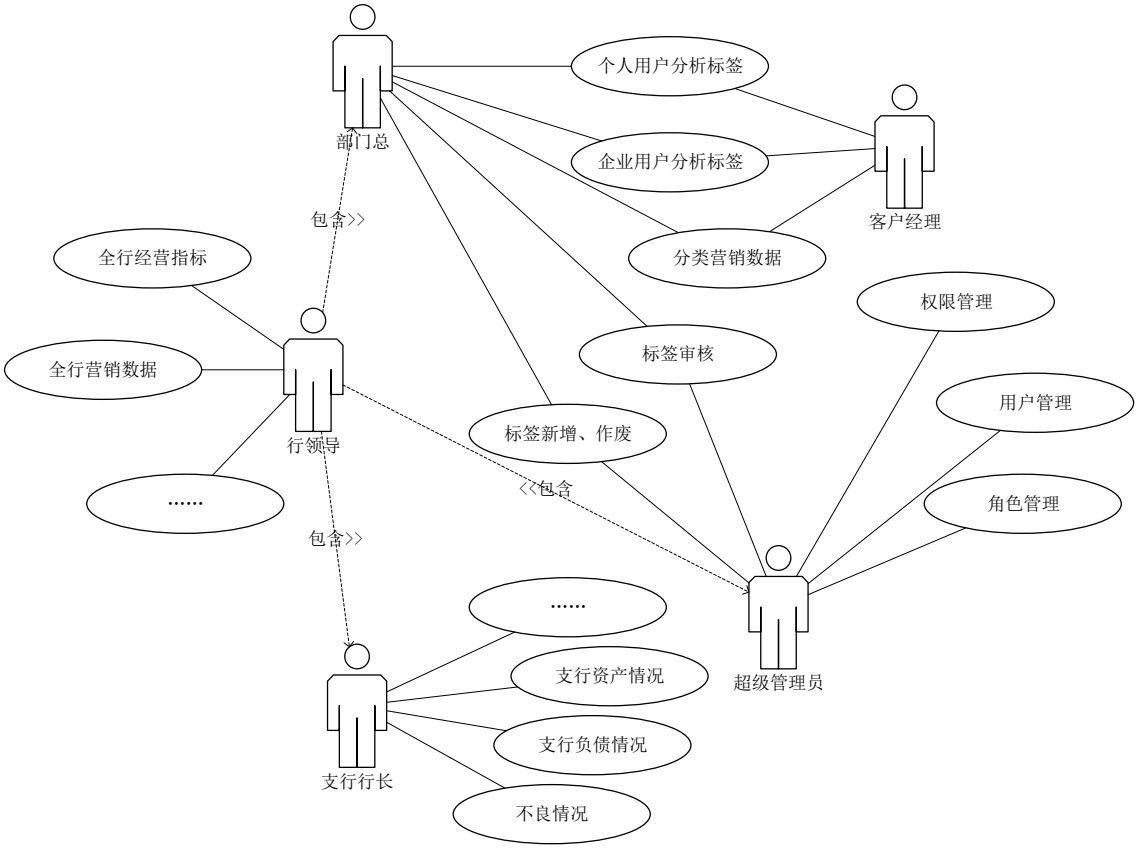


图 3-10 UML 角色用例图

图 3-10 中，总行领导应拥有所有查看权限，主要查看仪表盘内的全行经营指标、全行营销数据、不良总资产等，但同时在需要时可以查看总行职能部门总经理的权限表，包含其部门提交的需求分析结果、标签、营销数据等。同时总行行领导连挂管理若干个一级支行，应具有查看其所连挂管理支行数据的查看权限。

3.4 系统非功能性需求

3.4.1 软硬件需求

硬件是系统的运行基础设施，梳理出准确的硬件需求，是节约成本、保障系统稳定运行、性能达到预期的前提条件。本硬件需求将基于较准确的硬件需求，进行性能、数量的适当放大，以考虑满足广元市贵商村镇银行用户画像系统 3 年内的扩容需求。硬件选用 X86 架构平台，品牌及型号选择 Lenovo 联想品牌的 X3650-M，具体配置见表 3-2。

表 3-2 服务器选型详细配置表

指标	指标项	指标要求
CPU	CPU 类型	英特尔®至强 E5-2630V4
	CPU 单核主频	≥2.2GHz
	服务器总 CPU 主频数	≥2.2GHz*10（22GHz）
内存	内存类型及容量	DDR4=64GB
硬盘	硬盘数量及容量	4*600GB 2.5" SAS 磁盘 15000rpm
	硬盘槽位数量	8（RAS 支持热插拔）
	硬盘背板	支持 raid10、raid5 功能
I/0	网卡配置	集成 4 口千兆网卡
	HBA 卡配置	2 块 8Gbps 双端口 HBA 卡
电源	双电源	550W 白金双电源

为将硬件性能及资源利用到极致，同时进一步节约和有效利用硬件资源，计划将此批次服务器使用虚拟化技术，规划数量及预计分配节点，如图 3-11 所示。

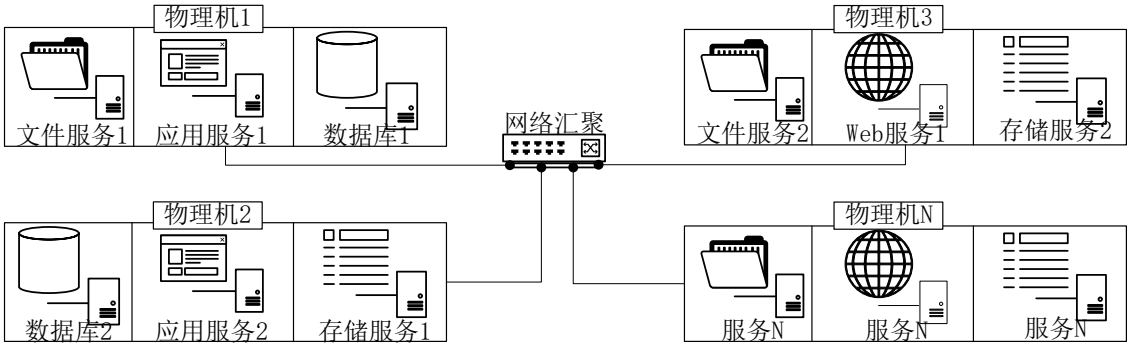


图 3-11 虚拟化架构示意图

软件是系统运行的环境保障，在虚拟化后，硬件资源成倍扩大，将冗余资源充分利用起来。用户画像系统需要使用 Hadoop 大数据框架相关技术，基于虚拟化搭建平台，最终实现相关系统展示，需要的相关软件及版本如下表 3-3 所示。

表 3-3 软件及版本信息表

名称	版本
Vmware ESXi	5.5
Linux	Ubuntu 18.10 bit64
Hadoop	2.9.2
Zookeeper	3.4.14
Sqoop	1.4.7
Flume	1.9.0
HAproxy	1.8.10
Hive	3.1.1
Hbase	2.1.5
SSH	OpenSSH 7.7
JDK	11.0.1

3.4.2 开发及运算需求

Hadoop 默认自带 MapReduce 运算框架，MapReduce 使用了分布式计算框架，将计算任务分布在集群的各个节点上运行，主要分为两个阶段，一是 Map，二是 Reduce。Hadoop2.x 之后，配套了新的运算框架 Yarn，Yarn 在设计上大大减小了 JobTracker 的资源消耗，并且让监测每一个 Job 子任务 Task 状态的程序分布式化了，更安全、更优美。将原有结构重新切分成了三个职能结构：一个全局的资源管理器 ResourceManager 一个负责每个节点代理 NodeManager，一个表示每个应用的 ApplicationMaster。三个职能服务分布在架构的不同节点上，如果 ResourceManager 出现问题，可以在其他节点上重启服务或另起服务，避免了单点故障引起的问题，而 NodeManager 替代了原来的 JobTracker，平均分布了各个节点上，由各节点的 NodeManager 自己监控自己的资源与进程，如此节点出现问题，ResourceManager 将会调用其他节点资源启动计算，大大提高了内存资源利用率。

3.4.3 数据安全及保密需求

广元市贵商村镇银行是银行业机构，本次计划纳入用户画像系统的上游系统数据，均有数据安全要求。一是需要保障数据能够安全、完整、准确的通过内网环境，通过各类 Hadoop 组件进入 Hadoop 集群存储框架中持久下来。二是需要保障各类数据不能丢失，要梳理好数据源头到数据尽头的完成流程图，把控好数据流中的各个节点访问及读取权限，实现“最小化”权限管理。三是定期检查数据

情况，同时做好数据、程序的备份工作，借助 Hadoop 框架本身的高可用性，进一步提高数据安全性。四是对部分敏感数据进行脱敏华处理，根据相关银行业监管要求，对相关数据进行清洗和脱敏，满足数据的保密性需求。

3.5 本章小结

本章明确了用户画像的系统建设目标，本着用户需求第一的角度出发，对用户管理、数据 ETL 规则、标签体系管理、用户画像管理、数据报表及仪表盘等需求做了详细介绍和建模描述，为下一步的设计工作打下了良好基础。下一章将结合本章需求及初步的建模，开展用户画像系统设计。

第四章 用户画像系统的设计

第三章介绍了用户画像系统的总体建设目标，对用户管理、数据 ETL 规则、标签体系管理、用户画像管理、数据报表及仪表盘等需求做了介绍和建模描述。本章同样为本课题的重点工作之一，将结合上一章节需求及建模，开展用户画像系统架构设计工作。

4.1 系统设计目标

基于项目组成员采集的需求结果，与该行科技部讨论最终确定技术框架，以便开展对用户画像系统的架构设计工作。

系统设计应参考当下广元市贵商村镇银行的发展经营情况，结合现有软硬件资源条件，同时针对未来发展冗余预留一定的资源。设计应考虑从整体到细节、自底层至顶层依次有序的描述，设计工作完成应召开阶段评审会，确定后便于指导后续系统进一步的开发及实施工作。

系统设计原则应遵循广元市贵商村镇银行已发布的相关软件系统开发类制度中的原则，同时应做到：“统一标准，灵活适用，指导开发”。

4.2 系统体系结构设计

为了方便业务人员沟通与理解，将架构设计以功能分类分为三大部分：第一部分为最底层，用户画像需要大数据支撑，该层设计的目标是基于 Hadoop 的 HDFS 存储文件、数据；第二部分为中间层，该层设计的目标是为了实现收集、导入、归纳各类文件、数据，并实现数据脱敏、清理及其他加工工作，同时完成一些针对数据的基础计算工作使用 Hive、Hbase 等实现词频统计、权重计算；第三部分为最顶层，该层设计的目标是为了实现基于 JavaWeb 或其他数据可视化类 UI 系统[39-43]，是业务部门使用人员直接使用的系统，界面应在满足需求的前提下尽可能的友好。整体系统体系架构如图 4-1 所示。

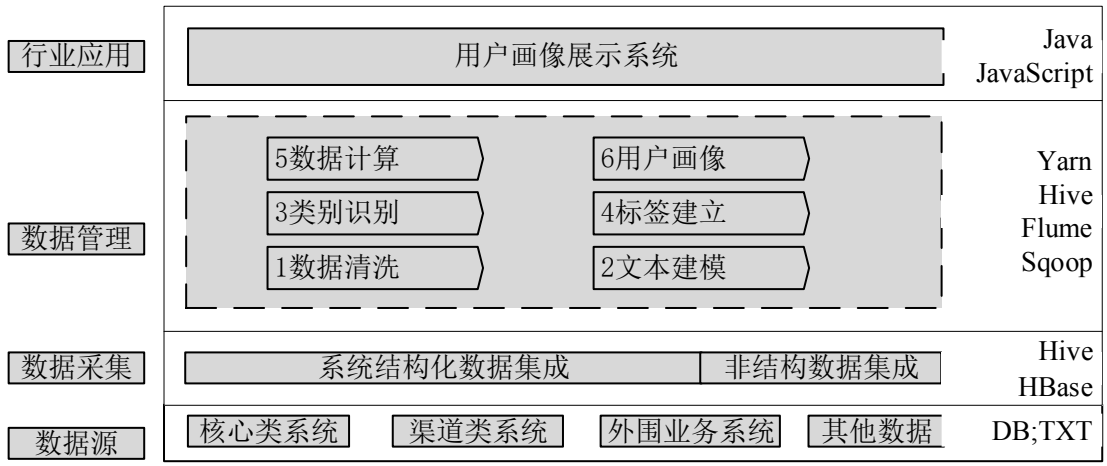


图 4-1 系统架构及内部组件

针对广元市贵商村镇银行的用户画像系统，图 4-1 中的体系架构满足了高可用需求。其需求主要体现在科技部方面，根据银行业相关监管规定，该系统实现后为非实时非交易系统，但需要实现高可用方案，以免在关键时段产生系统宕机或其他未知原因引起的系统服务停止。

本论文设计的用户画像系统，采用多种方式以保障高可用的实现。一是从系统层面，使用了 Vmware 自带的快照功能，定期由科技值班人员拍摄快照，并冗余硬件资源，随时可以恢复故障节点。二是从存储层面，先从 Hadoop 本身层面来说，Hadoop 本身架构设计就实现了高可用的方式，本论文在搭建环境中也参照其高可用配置的主从方案：名称节点搭建了双名称节点，一个主 NameNode，一个 SecondaryNameNode。三是从应用层面，本架构选用了 Zookeeper、Haproxy 两款负责协调、同步、高可用的 Hadoop 组件，由这两款软件负责 HBase、Hive 的高可用实现。

数据层是用户画像系统的最底层，该层的主要目的是为了实现数据的 ETL（Extract & Transform & Load）过程并实现 Hadoop 存储，不管数据是文件、数据库、图片或其他形式的数据，都应持久化在本层，并通过 Hadoop 的高可用高可靠，保障数据使用的效率与存储的安全性。

数据层应至少由 5 个不同物理机的节点共同构成，设计上由浅入深由粗到细大概分为三步：第一步，设计明确应用操作系统，经过科技部人员的推荐、结合实际应用的经验，加之网络文章的参数比对，最终设计明确了操作系统使用 Linux 的 Ubuntu 18.10 版本。第二步，设计明确存储架构。操作系统选定之后，Hadoop 及 Hadoop 组件选择了 2.9.2 版本及对应匹配版本的组件。设计中共五个节点，其中两个节点选择成为 NameNode 即名称节点，这两个节点一主一备又互为备份，另外三个节点设计成为 DataNode 即数据节点。第三步，安装其他组件服务至压力

较小节点，是性能平均分配。至此，集群式存储架构与数据、文件、日志等具体的应用组件架构设计完成，如表 4-1 所示。

表 4-1 物理机集群分配示意图

IP 地址	CPU&内存&硬盘	节点名称	HDFS 用途	组件规划
192.168.71.131	4C,16G,80G	nna	NameNode	sqoop1,HAProxy
192.168.71.132	4C,16G,80G	nns	Secondary NameNode	Flume,HAProxy
192.168.71.133	4C,16G,100G	dn1	DataNode1	zk1,hbase1,hive1
192.168.71.134	4C,16G,100G	dn2	DataNode2	zk2,hbase2,hive2
192.168.71.135	4C,16G,100G	dn3	DataNode3	zk3,hbase3,hive3

4.3 模块设计

本节将对数据层、业务层、展示层分别进行设计介绍。

数据层主要负责数据清洗、采集与导入，结构化数据从数据库或数据库文件中直接获取，需要对某些空字段、乱码、多余字符等进行处理；非结构化数据需要对文本内容进行格式化预定义处理，比如定长截取、字符分割等等，最终把两种数据转化为结构化可用数据。

业务层主要负责标签的分析、计算与生成，数据采集处理完成后，根据人工初步定义的特征标签词典，开始进入数据分类、关联、聚类分析。首先需要对数据进行切词，可以利用词频计算等方式，初步得到分词词典。其次根据词典，判断各个特征词语在某一客户数据中的权重，因为各词语在每个客户身上的权重是不一样的，此处使用 TF-IDF 计算权重，生成标签结果。当有一定对象或元组数据后，采用 K-means 算法对生成的用户画像结果进一步进行聚类分析。

展示层主要负责将业务处理层得到的数据结果进行可视化展示[44-46]，通过对不同系统角色权限人的判断，提供不同的菜单功能。展示层主要使用 Javascript 图表技术表示数据结果，通过 Json 串由 Ajax 变量获得数据传入 Javascript 对应的 data 域中，实现可视化。

三层各层之间的逻辑处理流程图，如图 4-2 所示。

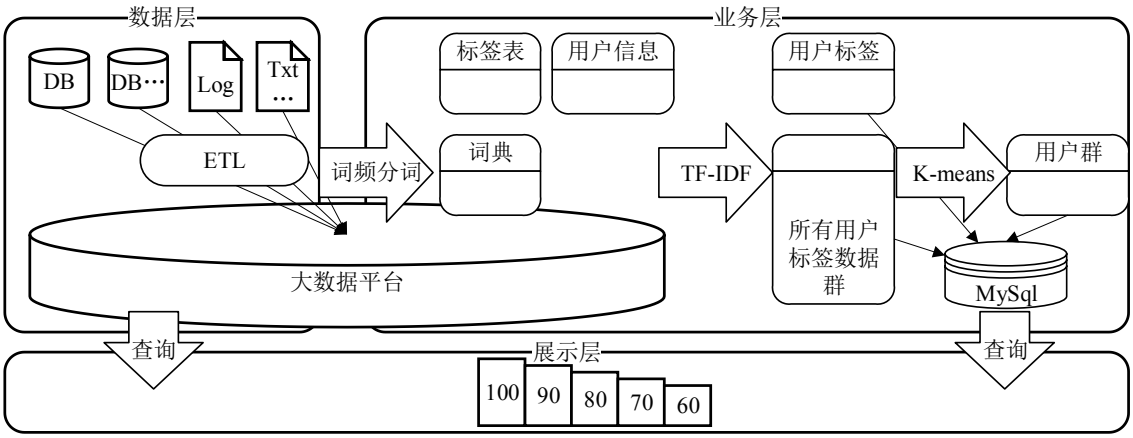


图 4-2 各层逻辑处理流程

4.3.1 系统数据层设计

根据广元市贵商村镇银行已有的数据类型进行初步分析发现，其数据主要为数据库文件、日志文件两大类，各系统数据采集导入流程如图 4-3 所示。

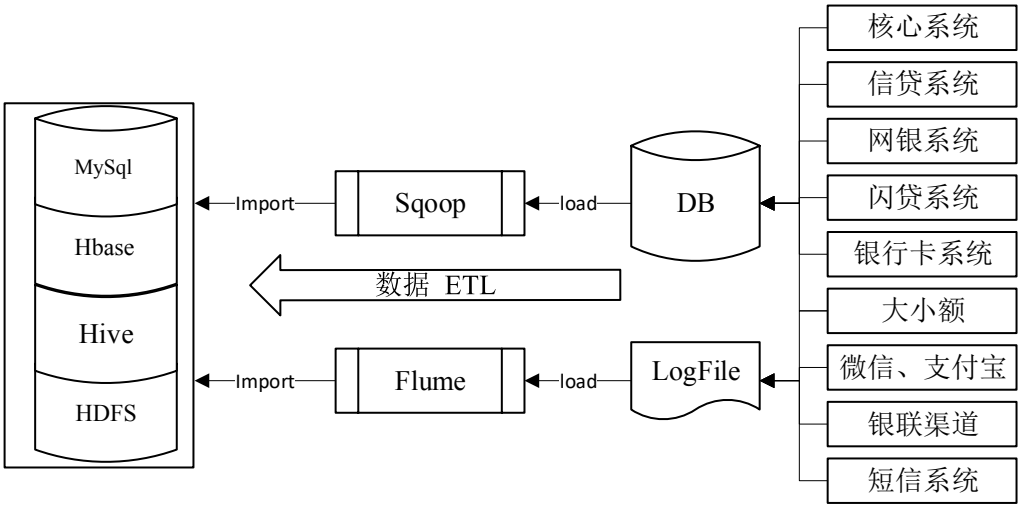


图 4-3 各系统数据采集流程

针对第一类数据库文件，设计使用 Sqoop、Hive 或 HBase 组件来实现数据库文件及数据的存储。Sqoop 可以单独部署在架构五个节点中的任意一个[47]，由 Sqoop 实现与外围业务系统数据库的对接与数据导入[48]，导入后 Sqoop 将数据库表以文件方式存放于 Hadoop 的 HDFS 系统中。HBase 可以看做是 Hadoop 的数据库，通过 Key-Value 方式存储数据。通过 Sqoop 组件工具实现数据的导入或通过命令行、脚本的方式实现数据的存储。各类系统对应的采集工具及对应入库设计如下表 4-2。

表 4-2 系统数据入仓关系表

系统名称	数据类型	入仓工具	存储系统
核心系统	Informix	Sqoop	HDFS、Hive
银行卡系统	Informix	Sqoop	Hive、HDFS
网银系统	Oracle	Sqoop	Hive、HDFS
银联	报文或日志	Flume	HDFS
大小额	报文或日志	Flume	HDFS
财付通	报文或日志	Flume	HDFS
支付宝	报文或日志	Flume	HDFS
信贷系统	Informix	Sqoop	Hive
闪贷	MySql	Sqoop	HBase
短信系统	Oracle	Sqoop	Hive

经过与各业部门的多次讨论，系统建设初期先将核心数据库数据全部导入至大数据平台中，以便后期使用。除核心系统以外，其余系统必须导入的表如下表 4-3 所示：

表 4-3 各系统及必入库特色数据表

系统名称	表名/文件名
核心系统	ADSHQ,ASDFH,AGHFH,AGDFH,ACPLS,ADKFH,AKMZZ,ANBFH,AQXFH,AZHJX,ABWMX,ADKMX,AGDMX,AGHMX,AJXMX,ANBMX,ASDMX,ASHMX,BAJDK,BDSKH,BDGKH,BGRXD,BGDDJ,BDJDJ,BDWXX,BTZDJ,BTZMX,PJGCS,PGYCS,PYWDH,PKMCS,PJYKM,PJYKZ,PJZLV,PZLCS 等
银行卡系统	IC_BUS_SYSPARA, IC_P_O_PARA, IC_S_CLASS, IC_P_MK_DET,VYKTD 等
银联、微信、支付宝	CUPS.log.yyyyMMdd、Alipay.gygscb.yyyyMMdd、Cft.log.yyyyMMdd 等
大小额渠道	BDEDJ,BXEDJ 等
信贷系统	CUSTOMER_INFO,ENT_INFO, BUSINESS_APPLY,BUSINESS_APPROVE,BUSINESS_CONTRACT,BUSINESS_PUTOUT,BUSINESS_DUEBILL,BUSINESS_HISTORY,BUSINESS_EXTENSION,BUSINESS_WASTEBOOK,BUSINESS_TYPE,BUSINESS_APPLICANT,CONTRACT_INFO 等

闪贷	FLASHCREDIT_CONTRACT,FLASHCREDIT_INFO 等
短信系统	MSG_FLOW_IPP,MSG_REBACK,MSG_SEND_CUSTOMER,MONI_Worker,MSG_FLOW_IPP_HIS,MSG_REBACK_HIS,MSG_SEND_CUSTOMER_HIS,MONI_Worker_HIS 等

以上数据为结构化业务数据，各业务系统数据通过客户号或客户身份证件号统一关联，通过导入至目标存储系统时进行数据的 ETL 操作。

针对第二类日志文件，设计选择 Flume 组件工具，配置 Flume 与 Zookeeper 集群的架构，可以避免 Flume 的单节点故障[49-51]，保障对日志文件目录的稳定采集，配置日志采集目录时，可以通过广元市贵商村镇银行已有的日志服务器进行采集，也可在初期单独采集各个业务系统的日志，下表为某业务系统日志 json 字段含义，如下表 4-4 所示：

表 4-4 系统日志 Json 字段含义

字段名	字段含义	长度
ADDRESS	地址	20
COUNTRY	国家	20
PROVINCE	省	20
CITY	市	20
DEVICE_ID	设备 id	60
USER_ID	用户 id	40
ACTIVE_NAME	活动页面	80
IP	ip 地址	16
SESSION_ID	sessionid	60
ACTION_PATH	路径	120
TIME_TAG	时间戳	20
REQ_URL	请求链接	120

Json 字符串值："address":{"country":"中国","province":"四川","city":"广元"},"device_id":"1f9367ed5ea2528d9126739bc48f4225","user_id":"28110000115881","active_name":"pageview","ip":"171.216.106.41","session_id":"0000f7714f3c48f4838513a65ad7383b","action_path":["https://ebank.gygscb.com/category"],"time_tag":1527604188966,"req_url":"https://ebank.gygscb.com/category "将以上所有表按照其表结构导入到对应的 HDFS、Hive 或 HBase 中后，才能开始下一步的业务层逻辑开发工作。

4.3.2 系统业务层设计

系统业务层是指衔接底层数据存储层与顶层展示层之间，主要实现数据预处理、业务处理及数据运算、数据统计等功能。

业务处理层设计实现三大功能，一是针对日志的分析，应通过对日志文本信息的分析与计算，实现日志数据的格式化落地，以便通过 Hive 等组件对其展开后期分析。二是计算标签，将库表数据中标签来源字段进行分词、词频统计，形成结果后，将此结果反馈保存至标签结果库。通过编写函数，与标签库碰库的方式，对逐个用户的数据进行计算，形成用户标签，将用户打好的标签存到用户标签库表中，形成用户画像表。三是数据统计与仪表盘功能，是针对广元市贵商村镇银行自身数据的分析与统计，将定期需要查询的报表或实时数据，通过数据可视化系统展现[52-55]。

在需求建模阶段，已将标签类别主要分成了基础类、习惯类、特征类三种类型，首先需要将各表中数据与以上三种类型进行初步分析归类，再计算分析将对应标签权重计入客户偏好表中，最后与标签权重阈值匹配，得出用户画像标签值。本课题设计的用户标签及规则如表 4-5 所示：

表 4-5 标签类别及规则定义

类别	标签内容	规则定义
基础	性别年龄	男、女；按照形成 25-35、成长 30-55、成熟 50-60、衰老 60 上分段
	所在地	取身份证前 6 位判断地区，对照公安发布的地区码表取值
	资产比例	资产负债比=年度资产/年度负债*100%
	是否有车有房	是否有房贷、车贷、大修基金、契税、月度加油消费记录
习惯	登录频次	月登录次数和
	日平均在线	月在线时长(m)/30
	在线时段	客户办理业务时段
特征	交易时段	客户交易时段：上、下、晚、凌晨
	交易类型	客户月交易类型统计：消费、转账、提现、收入
	交易额度	客户交易日均额度=月总额/30
	交易频率	客户交易频率=月总次/30
	标签推测	根据其年龄段+资产能力，推荐标签，形成 30，成长 40，成熟 50，衰老 80 比例

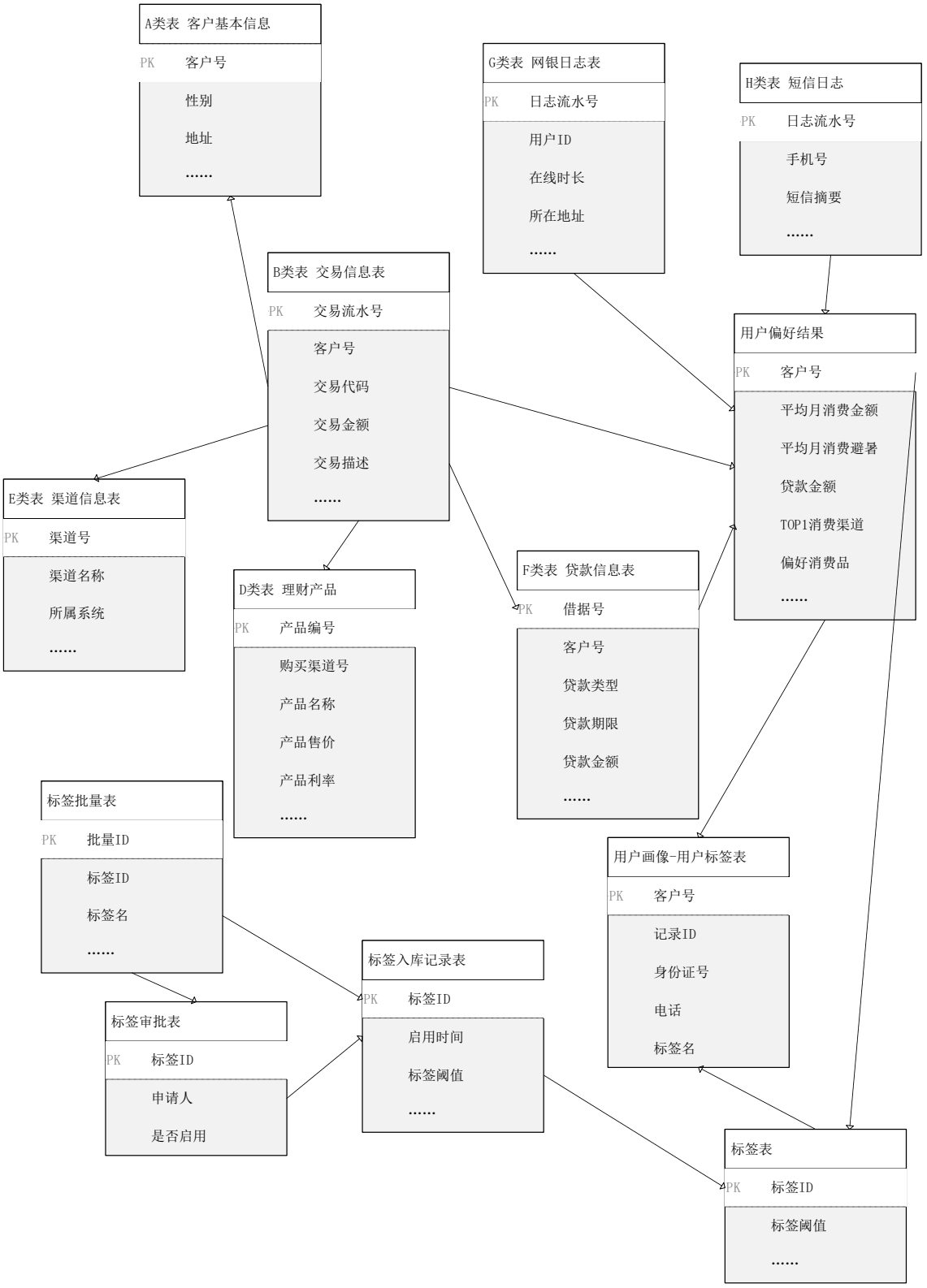


图 4-4 标签对象图

数据采集工作主要为了标签管理功能的实现，该功能实现需要数据支撑，如图 4-4 标签对象图所示，用户偏好结果表可以设计为次数表或权重表，根据贷款系

统的贷款信息相关表、核心系统交易流水相关表、网银系统日志表、短信系统日志表等，统计出用户对应的偏好值。标签系列的数据通过系统分析计算后，经审批入标签表，最终根据标签结果表与用户偏好值表计算得出该用户的匹配标签。

业务层标签的计算逻辑与设计实现，该层的标签计算过程是基于用户已发生的事实数据分析产生的，根据规则定义表中的具体规则，按照用户维度，遍历所有用户数据，当满足规则定义条件时，该标签匹配给用户。标签计算流程示意图如下图 4-5 所示：

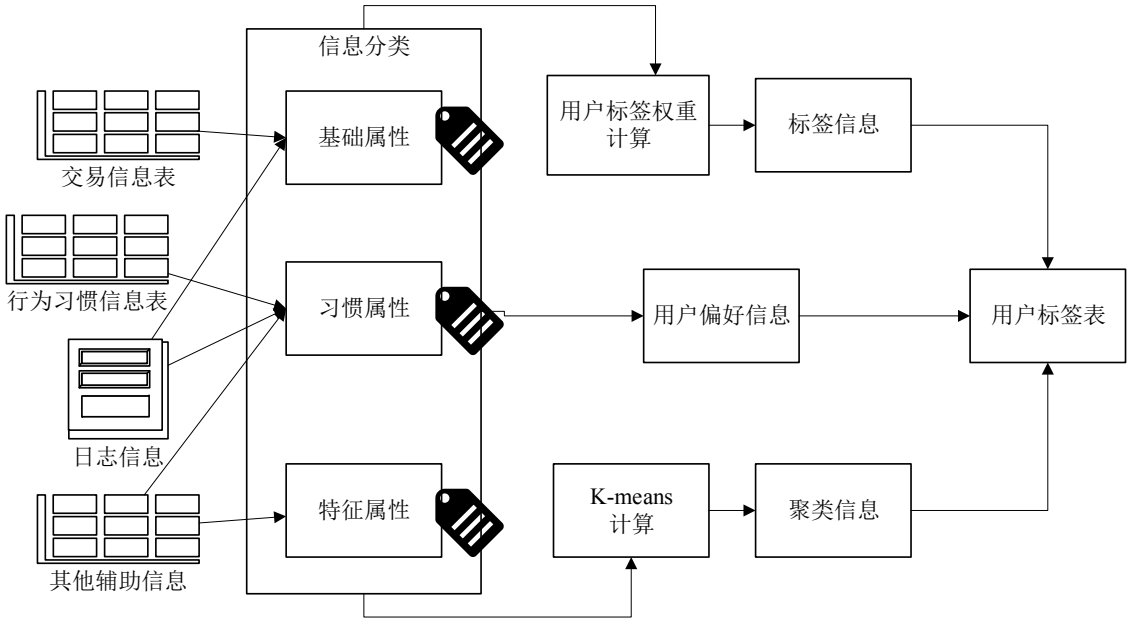


图 4-5 标签计算流程示意图

基础属性类标签，包含核心系统客户基础信息表、相关注册业务系统信息表、渠道开通登记表等，基础属性类标签属于半结构化标签，基础属性信息中多为事实信息，计算过程图下图 4-6 所示。

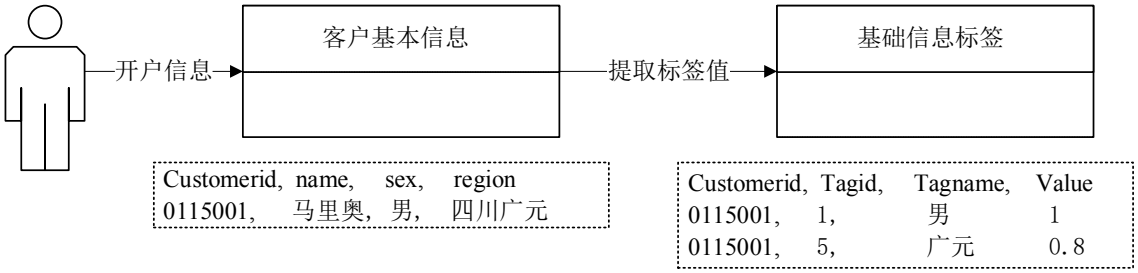


图 4-6 基础标签计算流程示意图

习惯属性类标签，其至少涉及的行为习惯信息有：核心系统交易流水表、网银系统交易信息表、日志信息、渠道的微信交易信息表、支付宝交易信息表、银

联交易信息表、大小额交易信息表、外围系统的短信日志信息表等至少 8 张信息表，首先建立习惯属性主表，建立临时表 1 获取交易类型、交易金额等表内信息，建立临时表 2 获取日志类信息中的交易类型、交易金额等信息；第二步需将交易时产生的标签值信息插入到习惯属性主表中，需要明确交易类型，如小额支出、日常支出、生活开支、房贷支出、车贷支出、大宗消费等。最后形成用户偏好信息，与计算权重结果进行匹配后，得出该用户习惯属性类用户标签，如图 4-7 所示。

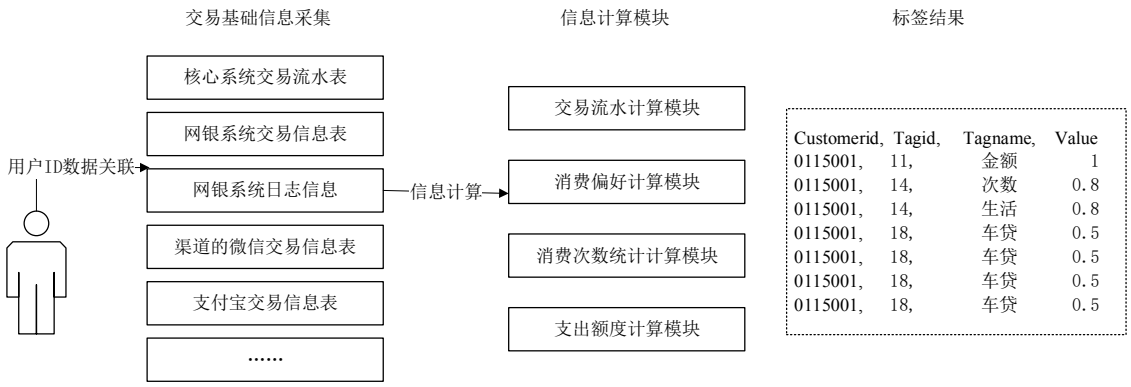


图 4-7 习惯标签计算流程示意图

特征属性类标签需结合现有标签结果，进行进一步分析与计算，得出该用户的特征属性类标签，如图 4-8 所示。

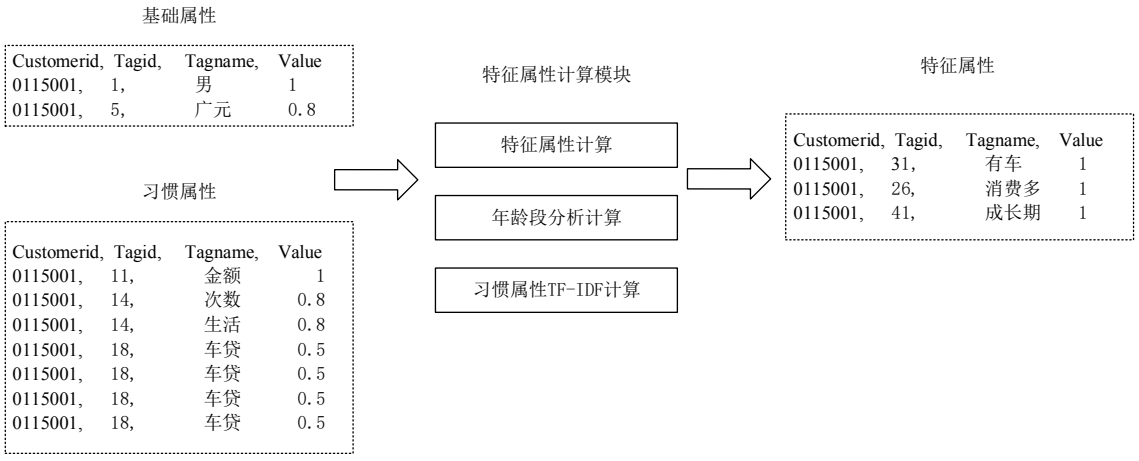


图 4-8 特征标签计算流程示意图

4.3.3 系统展示层设计

系统展示层满足数据可视化实现需求，设计有用户管理、标签管理、客户画像、数据报表、仪表盘等模块功能，下面就系统展示层设计开展介绍。

用户管理模块为公共功能模块，所有注册用户均可使用该模块的用户管理功能，用户可修改基本信息、重置密码等操作。超级管理员则根据角色配置不同权

限，使不同用户拥有不同权限类别，使用不同的系统菜单功能，数据库实体关系图如 4-9 所示。

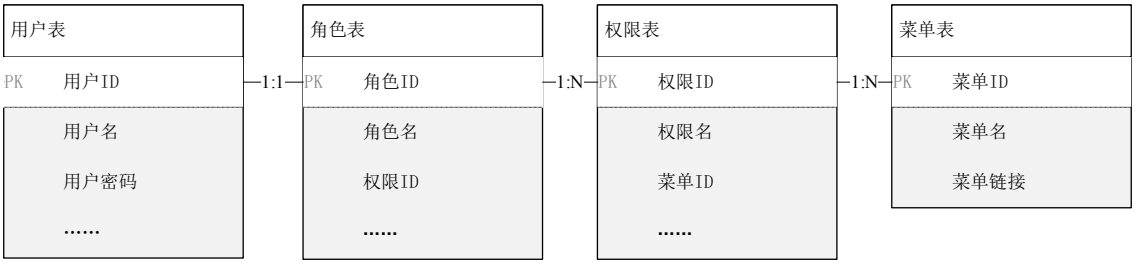


图 4-9 用户管理实体关系图

标签管理模块实现了标签的新建、审批、作废与启用管理，标签表与用户偏好表需合并计算，最终产生用户画像表，该部分如图 4-10 所示。

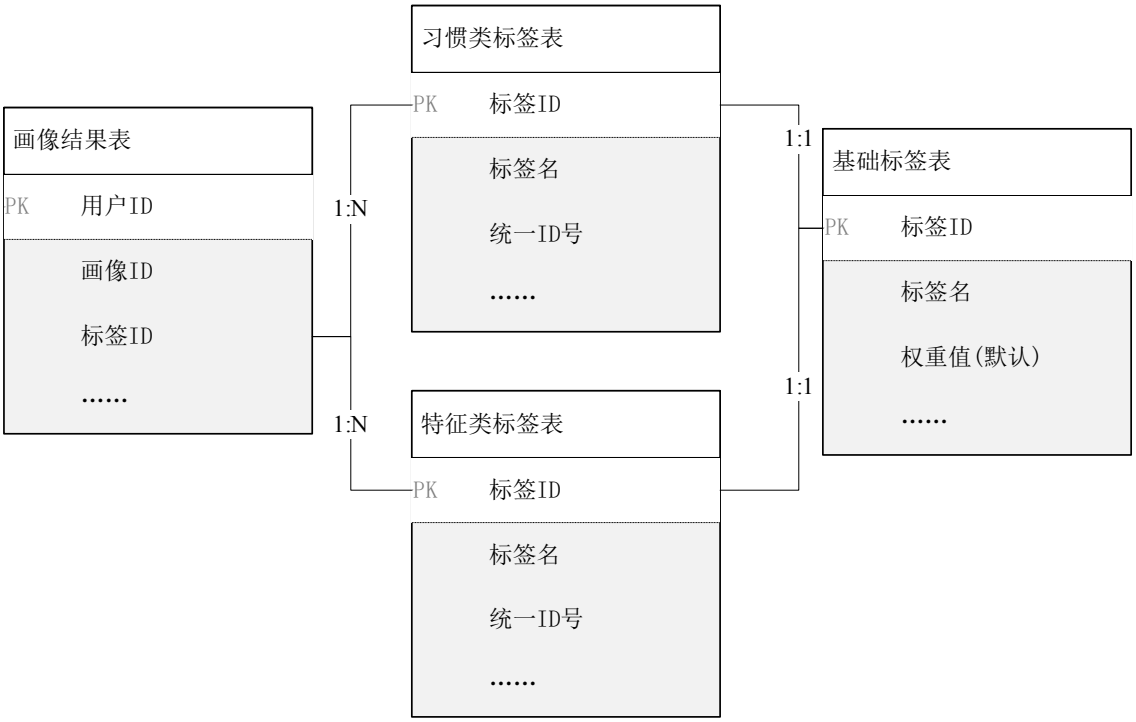


图 4-10 标签与画像实体关系图

4.4 本章小结

本章节针对广元市贵商村镇银行用户画像系统的设计进行了阐述，主要包含系统设计目标、系统设计原则等方向方针性质的内容及模块设计内容。下一章将在本章设计的基础上，进行用户画像系统的具体实现工作，从软硬件环境准备到各层功能的具体实施等重点部分分别介绍。

第五章 用户画像系统的实现

第四章针对广元市贵商村镇银行用户画像系统的设计进行了阐述，主要包含系统设计目标、系统设计原则等方向方针性质的内容及模块设计内容，其中重点对数据层、功能层展开了介绍。本章节将介绍用户画像系统的具体实现工作，介绍重点研究的工作，从软硬件环境准备、数据底层搭建与采集、业务逻辑层实现等重点介绍。

5.1 银行用户画像系统依赖的系统及软件平台

本课题研究的用户画像系统，依托于 X86 架构的 3650 服务器，为了能够实现现阶段的资源最大化利用，同步采取了虚拟化技术。下面将就软硬件平台搭建的实现工作开展详细介绍。

5.1.1 系统实现的系统平台

为了节约硬件资源，将更多的资源投入到软件平台中，虚拟化软件选择了 Vmware Esxi5.5。开发环境中，使用了 Vmware Workstation 15。首先安装虚拟机 Ubuntu Linux 18.10，Ubuntu 本身安装较为简单。先下载 ubuntu 镜像，可以在 old-releases.ubuntu.com/releases/处下载，该地址是 ubuntu 官方地址，在本地中为已发布的版本。选择好版本后，点击下载，将 ISO 文件下载到本地后，打开 Vmware WorkStation 新建虚拟机时选择对应 ISO 镜像。

选择 ISO 完成硬盘、CPU、内存等配置后，点击打开电源，即可开启安装系统。安装完成后，需要进行几处基本的初始化配置，配置完成后即可开始拍摄快照或者克隆本快照，实现快速系统复制的目的，虚拟化分配如下图 5-1 所示。

其余配置如修改软件源、安装 ssh、jdk 等操作，可以根据使用需要安装，如使用不受影响，则可以进入下一小节，下一小节将介绍 Hadoop 框架的安装与搭建实现，包含 ssh、jdk 等配置安装内容。

5.1.2 系统实现的软件平台

Hadoop 及相关组件的下载可以在 http://mirror.bit.edu.cn/apache/等国内源进行下载，通过虚拟机共享目录上传至节点后安装，软件平台各组件详细版本如下表 5-1 所示。

表 5-1 软件平台详细配置表

名称	版本	主要用途
Vmware ESXi	5.5	虚拟化
Linux	Ubuntu 18.10 bit64	各服务操作系统
Hadoop	2.9.2	提供 HDFS 分布式存储
Zookeeper	3.4.14	提供协调服务
Sqoop	1.4.7	数据库导入导出服务
Flume	1.9.0	提供日志导入服务
HAproxy	1.8.10	提供高可用负载均衡服务
Hive	3.1.1	非结构化数据存储
Hbase	2.1.5	结构化数据存储

Hadoop 安装要注意 HDFS 的配置，需要按照虚拟化规划表进行配置以下配置文件：core-site.xml,hdfs-site.xml,map-site.xml,yarn-site.xml,fair-scheduler.xml，core-site.xml 为 Hadoop 核心配置文件，指明了服务的地址、端口、集群目录等信息。hdfs-site.xml 是 HDFS 文件系统配置文件，指明了 HDFS 节点别名、地址、端口等信息。map-site.xml、yarn-site.xml 是 MapReduce 计算资源配置、Yarn 资源管理器配置，yarn-site 与 fair-shceduler 协同配置，可实现 HDFS 的高可用[56-57]。

安装完成后，可执行 Hadoop 的样例程序，已验证当前环境是否已安装成功，样例程序执行结果如图 5-1 所示。

```
02:03:14 INFO mapred.LocalJobRunner: Finishing task: attempt_local432030412_0001_r_000000_0
02:03:14 INFO mapred.LocalJobRunner: reduce task executor complete.
02:03:15 INFO mapreduce.Job:  map 100% reduce 100%
02:03:15 INFO mapreduce.Job: Job job_local432030412_0001 completed successfully
02:03:15 INFO mapreduce.Job: Counters: 35
File System Counters
        FILE: Number of bytes read=607044
        FILE: Number of bytes written=1569296
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
        HDFS: Number of bytes read=28
        HDFS: Number of bytes written=18
        HDFS: Number of read operations=13
        HDFS: Number of large read operations=0
        HDFS: Number of write operations=4
Map-Reduce Framework
        Map input records=1
        Map output records=2
        Map output bytes=22
        Map output materialized bytes=32
        Input split bytes=99
        Combine input records=2
        Combine output records=2
        Reduce input groups=2
        Reduce shuffle bytes=32
        Reduce input records=2
        Reduce output records=2
        Spilled Records=4
        Shuffled Maps =1
        Failed Shuffles=0
        Merged Map outputs=1
        GC time elapsed (ms)=0
```

图 5-1 WordCount 样例程序执行截图

其余组件 Zookeeper、Sqoop、Flume、HBase、Hive 的安装过程较为简单，下载对应版本的安装包加压后，配置环境变量 profile、再修改软件目录下的对应配置文件即可。

5.2 系统数据层实现

上一节主要阐述和介绍了用户画像系统依赖的系统及软件平台搭建工作，基本完成了 Hadoop 集群环境的搭建，并安装了相关上层应用组件。接下来将对系统数据层展开进行介绍，数据层实现了数据归集及存储，为上层功能层、展示层提供可靠数据的数据源输出，下面将展示数据层数据归集的核心功能部分。

本系统数据归集功能存储在 Hadoop 的 HDFS 架构系统之上，因此不用担心文件存储及安全性问题，可以将更多的精力集中在数据库数据、非结构化数据的处理与存储工作上。

5.2.1 HDFS 文件系统数据采集

简要介绍 Hadoop 的 HDFS 系统操作，HDFS 实现了普通文件的分布式存储。HDFS 本身的操作与普通文件系统类似，只是每次使用前需要加上 hdfs 的命令行生命，再结合 linux 习惯的相关命令即可，如命令：“max@nna:~$ hdfs dfs -ls /”意思为使用 ls 列示查看 dfs 系统的根目录下的所有文件。普通文件可通过架构中的五个节点，通过-put 命令直接向 HDFS 系统传送，如：“max@nna:~$ hdfs dfs -put bankNumber.txt /tmp/bankNumber.txt”， -put 后的第一个 bankNumber.txt 为本地文件，发送到 HDFS 文件系统的/tmp/目录下并使用名称 bankNumber.txt，再次使用-ls 即可查看该文件是否上传成功。

5.2.2 Sqoop 数据导入采集

基于 HDFS 文件系统，使用 Sqoop 组件，可以实现 Sqoop 与各类数据的数据交互及导入操作。广元市贵商村镇银行主要系统使用的数据库为 Informix、Oracle、MySql 三种，根据查询时效，分别将库表导入到 HDFS、Hive 或 HBase，下面将列示主要的核心代码，使用时根据需要组合即可。Sqoop 与 Flume 的数据导入流程如图 5-2 所示。

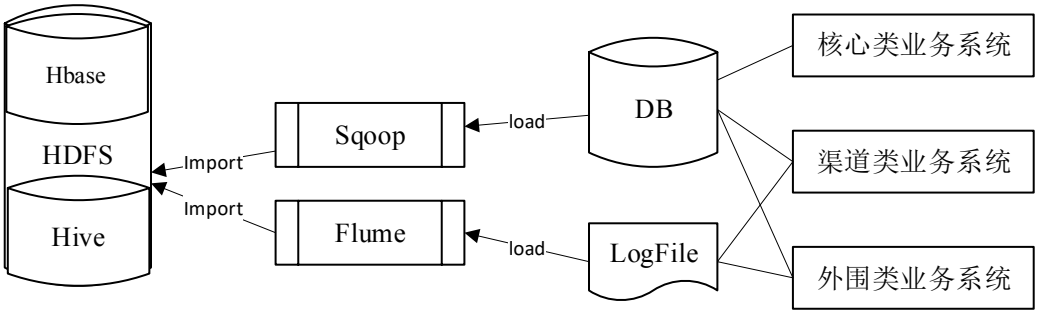


图 5-2 Sqoop 与 Flume 的具体使用

Sqoop 连接 Informix，并将 cbs 整个数据库导入到 HDFS 中，核心代码如下：

```
sqoop import-all-tables \
```

```
--connect
"jdbc:informix-sqli://99.6.12.189:10001/cbs:INFORMIXSERVER=online_sop;NEWC
ODESET=GBK,8859-1,819,Big5" \
--username informix \
--password informix \
--driver com.informix.jdbc.IfxDriver
--warehouse-dir "SqoopImport/sopDataAll" \
--autoreset-to-one-mapper \
-m 2        >>结束。
```

import-all-tables 参数为导入整个数据库所有表的关键字，--为 sqoop 的关键字执行符号，“\”为 sqoop 命令的换行符，为了操作整洁及查阅方便，每个关键字参数进行了换行。此处将用户名、密码放入了连接串中，所以没有单独列示 sqoop 的用户名、密码参数。“--autoreset-to-one-mapper”参数设置为解决 cbs 核心库表部分表没有主键问题，这样对于没有主键的表将自动生成一个 map 去处理导入工作。“-m 2”表示指定 sqoop 生成几个 map 任务处理导入工作，sqoop 默认采用 4 个 map 任务，m 指定为 n，则在 HDFS 中导入结束后执行结果中就将生成几个 part-m 文件。

Sqoop 连接 Oracle，并将指定表导入到 Hive 中：

```
sqoop import \
--connect jdbc:oracle:thin:@99.16.12.11:1521:msg \
--username msgserver \
--password msg123456 \
--table TABLENAME    \
--hive-import \
--hive-database hivecbs \
-m 1        >>结束。
```

与导入至 HDFS 相比，执行语句多了 Hive 的相关参数：“--hive-import”参数指定了导入方式为将数据表导入到 hive 数据库中，此参数必须指定，否则无法导入至 Hive 中。“--hive-database”参数指定了导入的目标 Hive 库名为 hive_cbs，若为指定此参数，数据将默认导入至 Hive 默认库 default 中。以上语句导入执行结果为将 Oracle 的 msg 库中 msg_customer_number 表导入到了 Hive 库 hive_cbs 的 msg_customer_number 表中，Hive 自动创建了一张与 Oracle 导入表一样的表。也可以根据需求指定导入 Hive 后的表名，指定新表名参数为：--hive-table

newtbName。

Sqoop 连接 MySql，并将指定表的指定数据导入到 HBase 中：

```
#!/bin/bash
#声明指定需要连接的数据库字符串
dbmysql_connstr="jdbc:mysql://99.16.12.12:3306/flashcredit"
dbmysql_driver="com.mysql.jdbc.Driver"
dbmysql_username="flashcredit"
dbmysql_passwd="123456"
#指定需要导入的表名及列
dbmysql_table="customer_apply"
dbmysql_columns="SERIALNO,RELATIVESERIALNO,OCCURDATE,CUSTOMERID,CUSTOMERNAME,BUSINESSTYPE,BUSINESSSUBTYPE,OCCURTYPE,FUNDSOURCE,OPERATETYPE,CURRENYLIST,CURRENCYMODE,BUSINESSTYPELIST,CALCULATEMODE,USEORGLIST,CYCLEFLAG,FLOWREDUCEFLAG,CONTRACTFLAG,SUBCONTRACTFLAG,SELFUSEFLAG,CREDITAGGREEMENT,RELATIVEAGREEMENT,LOANFLAG,TOTALSUM,OURROLE,REVERSIBILITY"
#指定 HBase 表名及列簇名
hbase_table="customer_apply"
hbase_rowkey="credit"
datenow=$(date +%Y-%m-%d)
#Sqoop 导入 MySql 数据表
$SQOOP_HOME/bin/sqoop import
--connect ${dbmysql_connstr}
--driver ${dbmysql_driver}
--username ${dbmysql_username}
--password ${dbmysql_passwd}
--table ${dbmysql_table}
--columns "${dbmysql_columns}"
--hbase-table ${hbase_table}
--hbase-row-key ${hbase_rowkey}
--check-column ACTION_TIME
--incremental lastmodified
```

--last-value ${datenow}

-m 5 >>结束。

与前两种方式相比，此处核心代码区别更加明显，一是使用了 shell 脚本结合 sqoop 命令实现导入功能，二是脚本核心功能为每日日终 21:00 点，自动导入当天"customer_apply"表的新增数据，通过“--check-column”的 ACTION_TIME 字段去判断增量，“--incremental”指定导入方式为增量导入，“--last-value”传入当天日期参数，三个参数结合实现对新增数据的判断。

5.2.3 Flume 非格式化数据采集

针对系统提示信息、打印输出及日之类等非格式化文件，Flume 组件提供了可靠的采集方案。Flume 有三个重要的配置参数，在/usr/soft/flume-1.9.0/conf 目录下的 flume-conf-local.properties 文件中，分别是：source、channel、sink，其中 source 指定要从哪里采集数据，channel 指定了使用哪个通道传输数据，sink 指定数据发送的目的地，这三个参数的 type 属性是必须配置的。

启动时指定使用 flume-conf-tohive.properties 配置文件，具体启动命令为：flume-ng agent -n agent1 -c conf -f $FLUME_HOME/conf/flume-conf-local.properties -Dflume.root.logger=DEBUG,CONSOLE &，其余配置不变，至此 Flume 导入日志文件到 Hive 的基本配置参数就设置完成了，进入 hive 命令行，建立以 flumelogs1 名称的数据库及 flume_user 表，当目录中有日志文件上传或产生时，即可采集至 Hive 中。

广元市贵商村镇银行用户画像系统设计时规划的日志采集为非实时采集，即采集的日志为上日甚至上周的历史日志，分析工作针对历史日志开展。但为了使系统功能尽量优化，考虑到该行发展及实时分析的可能性，针对各线上生产系统实时通过日志工具类似 log4j 等产生的日志，输出到本地服务器磁盘上，为了满足今后实时针对此类线上生产系统开展分析，行为日志分析实现方式为：对实时产生的日志分析处理，需要在各线上生产系统服务器上安装部署 Flume 软件，将其作为 Agent 节点，通过配置 sources 的 type 为 exec 类型，可以将线上生产系统实时输出打印的日志采集到 FlumeCollector 服务节点中。

自定义 Flume 的 intercept 拦截器，将收集的 Event 数据进行格式化，Flume 日志收集架构设计成扇入流的方式，将各个线上系统服务器上的 Flume Agent 收集的 event 数据流扇入到另外一台/多台（ha）Flume 服务器上，汇总的 Flume 将 Event 数据库保存到 hdfs 上；（/t 分隔的文本格式）hive 建立外表，读取 flume 收集到 hdfs 上的数据。自定义的 Flume Intercept 过滤拦截器伪代码如下：

```
public class MyFlumeInterceptor implements Interceptor {
    public Event intercept(Event event) {
        Map<String, String> headers = event.getHeaders();
        String body = new String(event.getBody(), Charsets.UTF_8);
        以 personalRecommend():分割，\\转义()
        String[] split = body.split("personalRecommend\\(\\):");
        if (split == null || split.length <2) {
            为空时返回;
        } else {
            取出分割的后半段内容
            String logStr = split[1];
            转 Map 后再赋值给 Log 实体
            Map<String, String> fieldMap = getLongStr4Map(logStr);
            LogEntity logEntity = getLogEntityFromMap(fieldMap);
            返回 event 对象;
          }
    }
```

将传入的字符串转为 Map，代码片段如下：

```
    根据分号；分割字符串
    public Map<String,String> getLongStr4Map(String str) {
        Map<String,String> map = new HashMap<>();
        String[] split = str.split(";");
        根据分隔符数组进行 for 循环{
        if (是否为结束符字符串) {
            否则继续；
        }
        根据等号进行过滤，并取出 key 与 value 值，存入 map 中
        String[] split2 = fileds.split("=");
        if (split2 == null || split2.length<2) {
        }
        String key = split2[0];
        String value = split2[1];
        map.put(key.trim(), value.trim());
```

}

建立 maven 项目，编写自定义 intercept 并编译打包，挑选已经部署 Flume 的服务器节点，配置 Flume 作为 Agent 角色，将生成的 jar 放到 flume_home 的 lib 中。配置文件中一共需配置四个 interceptors 过滤拦截器：fillt1 正则表达式过滤器，过滤线上系统的日志数据；filt2 是 host 拦截器拦截 flume 的 event 数据；filt3 是 timestamp 拦截器拦截 header 中放置时间信息；filt4 是自定义拦截器，用来格式化日志数据。

启动 Flume Agent、Flume Collector，此时就能将线上生产系统产生的实时日志，通过扇入流的 Flume 保存到 hdfs 了。

创建 Hive 表：

```
CREATE EXTERNAL TABLE 'ttengine_api_data' (
   'uid' string, 'ppuid' string, 'ch_id' string, 'f_num' int, 'cost' int, 'usg' int,……
   'p_15' string)
PARTITIONED BY (
   'dt' string,
   'hour' string)
ROW FORMAT DELIMITED
   FIELDS TERMINATED BY '\t'
STORED AS INPUTFORMAT
   'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
   'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
   'hdfs://hadoop-jy-namenode/data/qytt/flume/ttengine_api'
```

这里需要注意，外表的路径要和 Flume 收集到 hdfs 上的主目录保持一致（不包含分区目录，分区目录 hive 会自动建立）；表的列要和 Flume intercept 解析数据的格式相互对应；该表有两个分区，PARTITIONED BY dt 和 hour。

5.3 业务处理层实现

业务处理层的实现依托于数据层的数据采集功能，业务处理层不负责实现对数据的采集动作，主要实现对已收集数据的处理、分析与结果输出至 Hbase、MySql 关系数据库表或格式化文件中。数据采集及清洗完成后，将对日志数据、数据库表数据等数据展开统计与计算，具体流程如下图 5-3 所示。

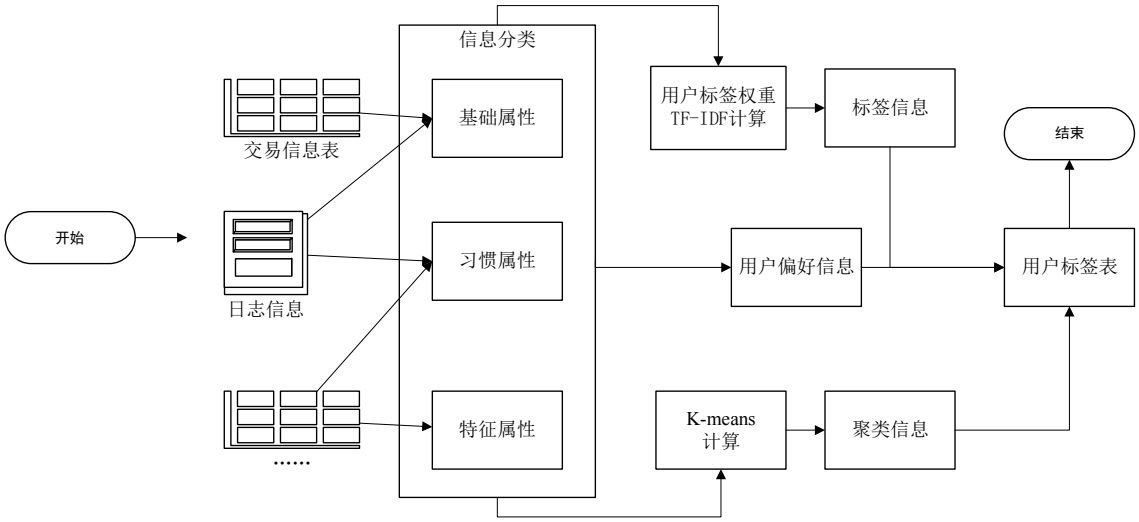


图 5-3 标签生成流程

5.3.1 行为日志分析实现

行为日志的分析是对客户情况的进一步了解，便于开展客户关怀、营销工作。广元市贵商村镇银行的用户行为日志主要存放于核心系统日志、网银系统日志、短信平台日志、手机银行日志等。下面将针对网银系统日志展开分析实现：

网银日志样式如下：

{"address":{"country":" 中 国 ","province":" 四 川 ","city":" 广 元 "},"device_id":"1f9367ed5ea2528d9126739bc48f4225","user_id":"28110000115881","active_name":"pageview","ip":"171.216.106.41","session_id":"0000f7714f3c48f4838513a65ad7383b","action_path":["https://ebank.gygscb.com/category"],"time_tag":1527604188966,"req_url":"https://ebank.gygscb.com/category "}

创建 bigdata 库，create database if not exists bigdata，然后创建 eweblog 表，即用户行为表 create external table if not exists 'bigdata.eweblog' 并为 eweblog 添加分区，随后创建用户表：create external table if not exists 'bigdata.member' 与订单表：create EXTERNAL table `bigdata.orders` 及其他辅助信息表。

统计男性、女性花钱总数谁更多，平均每个男性和每个女性谁花钱更多？尝试将 member 表和 gender 表进行关联，然后根据 gender 分为 2 个组，进行 join 聚合操作即可。用户画像标签库属性：性别、年龄、设备类型、注册时间、首次下单时间、最近一次下单时间、下单次数、下单金额、最近一次下单地理位置。最简单的方法可以通过多表关联，但是这样不好维护，如果新增新的标签，需要原标签库和新标签再次进行关联。这里使用一种按照标签语义分区，灵活运用行转

列、列转行的方法。第一步建立中间表和大宽表。中间表 user_tag_value 只有 3 个字段：user_id， tag 和 value，1 个 user 的所有标签和值对应了多行。大宽表 user_profile 只有 2 个字段：user_id， profile，profile 是一个 json 串，存放了用户的所有可能标签。第二步将不同标签写到相应分区。性别、年龄、设备类型、注册日期写入 basic_info 分区。

查询男女性花钱总数与平均值比较的实现语句：

```
select gender, count(1) num_gender, sum(t2.pay_amount) sum_amount,
  avg(t2.pay_amount) vg_amount
from
  (select user_id, gender from member) t1
join
  (select user_id, pay_amount from orders) t2
  on t1.user_id=t2.user_id
group by gender;
```

5.3.2 用户标签计算实现

标签是用户画像的重点功能，在没有明确标签选择规则前，用 Hadoop 的 MapReduce 或 Yarn 提供的计算功能，统计词频，经过人工筛选后，保留有价值的标签进入标签库。使用命令：hadoop jar $HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.9.2.jar wordcount /tmp/SopKeyWordCount /tmp/SopTags20191020.txt，将该结果存入标签库表。

接下来，通过创建标签计算节点，实现对用户数据的逐个标签计算，初始化标签节点数组，数组大小(size)最好与标签库的数量相同，因为数组是顺序存储的，通过下标查找，将提高运行速度。根据 StringPointer 类中 matchTags 函数的方法，得到用户数据的标签匹配计算，主要过程为先对数据进行文本预处理，可通过 replaceAll()函数进行替换，再对预处理后的数据进行匹配计算。计算标签头 2 个字符的 Hash 值(hash)，计算标签应该存到数组的位置(hash & (size - 1))。如果数组该位置为空，为此标签生成节点，添加此节点到该位置。如果数组该位置不为空，判断标签和此位置的节点的 headTwoCharMix 是否相等，若相等，则将标签添加到 TreeSet 中，若不相等，则生成新的节点。计算完成后，将结果插入标签库表中。用户标签计算结果表如下表 5-1 所示：

表 5-1 标签计算结果表

客户 ID	标签 ID	标签值
10000011	201	贷款
10000011	202	车贷
10000011	205	消费贷款
10000011	206	当期已结清
10000011	305	支付宝
10000023	101	活期存款
10000023	102	定期存款
10000023	201	贷款
10000037	101	活期存款
10000037	301	银联

5.3.3 TF-IDF 计算实现

词频算法 TF-IDF 是业务逻辑层计算模块的主要计算功能[58-60]，主要负责根据词组计算权重。词频统计 TF（Term Frequency，缩写为 TF）的思路就是找出库表中或某字段中，出现次数最多的词，如果某个词很重要，它应该在该表或字段数据中出现次数远超过其他词语出现的次数，这是进行词频统计的主要思路。TF 正向词频统计，在银行流水表中的表现为，“账户”、“余额”、“交易”、“金额”等词语出现的词频最高，但此类词语是需要过滤掉的，所以需要一个调整系数，来判断该词语是否为常见词语。而标签需要的词语是比较少见，但在库表或字段中多次出现，该词能够反映特性才是所需要的关键词。为了将此类关键词挑选出来，就需要使用逆文档频率 IDF（Inverse Document Frequency，缩写为 IDF）。TF-IDF= 词频统计 TF * 逆文档频率 IDF。实现的 Java 关键代码片段如下：

```
// 计算其他领域包含候选词文档数
for (Map.Entry<String, Integer> entry : containsKeyset) {
    if (!entry.getKey().equals(f.getName())) {
        otherContainsKeyDoc += entry.getValue();
    }
}
// 计算其他领域文档总数
for (Map.Entry<String, Integer> entry : totalDocset) {
    if (!entry.getKey().equals(f.getName())) {
        otherTotalDoc += entry.getValue();
```

```
        }
    }
    // 计算 idf
    idf = log((float) otherTotalDoc / (otherContainsKeyDoc + 1), 2);
    // 计算 TF-IDF = TF * IDF 并输出
    for (Map.Entry<String, Double> entry : tfSet) {
        if (entry.getKey().equals(f.getName())) {
            tfidf = (double) entry.getValue() * idf;
            System.out.println("tfidf:" + tfidf);
        }
    }
```

5.4 展示层实现

展示层主要功能是为了实现数据可视化，将业务处理层加工、计算后的结果表，通过可视化图表技术，结合用户角色、权限功能，最终实现该层功能。

5.4.1 展示层框架

展示层框架又分为后端与前端两大部分：一是后端框架，主要使用了 Java 的 SSM 框架即 SpringMVC+Spring+MyBatis，并没有采用更为成熟的 SSH 框架即 Struts2+Spring+Hibernate。这是因为在实际开发中，SpringMVC 可以完全替代 Struts，配合注解的方式，编程非常快捷可以有效缩短开发时间。另外，MyBatis 也可完全替代 Hibernate，因为 MyBatis 的半自动特点，我们可以更灵活的使用 SQL，这样考虑是为了后期需要，可以灵活、高效的使用 SQL 语句，实现更丰富的查询组合与结果集。二是前端框架，主要为了具体实现页面内容展示，主要选用了 Bootstrap 封装的 AdminLTE、Jquery、Echars 等 JS 包，为了实现更丰富的图表功能。

展示层框架搭建 IDE 使用 Eclipse，创建动态 Web 项目后，建立好相应的目录与包。这里重点介绍几个包的作用：一是 DAO 包，也就是常说的数据访问层包，这个包中的实现类负责与数据打交道，可以是数据库操作、访问数据层处理后的结果表或数据采集后的源数据，结合 Mybatis 可以直接在配置文件中实现接口的每个方法。二是 WEB 包，也就是控制器包。SpringMVC 就是在这层通过 Controller 控制器，代替了 Struts 中的 Action。其他的 Entity 实体类、DTO 数据传输层、Service 业务逻辑接口、ServiceImpl 业务逻辑实现包等不做详细介绍。

后台框架通过 Java 开发，结合数据库设计的用户、角色、权限、标签信息、标签结果等数据表结构，前端使用 AdminLTE 模板实现了用户管理模块、标签管理等功能。

5.4.1 数据可视化实现

数据可视化通过图表技术实现，本课题中主要组合使用了 AdminLTE、Jquery、Echars 等 JS 包，通过 SSM 获取后台数据并经业务代码加工处理后，传送至不同的图表进行展示，此处难点是如何在单一页面中同时展示多个不同类型的图表，解决思路为模块化地对需要展示的数据表进行操作，不同的图表调用不同的函数。实现关键代码如下：

```
Var balanceOp = {
    xAxis: [指定 x 轴信息],yAxis: [指定 y 轴信息],
    grid:[指定坐标轴位置，大小],
    series: [{type: 'line',//折线图 data: lineData[0]//折线图数据赋值}]
};
//饼图 1
var pieChars1={data :pieData[0]//数据赋值};
//饼图 2
var pieChars2={roseType : 'area',//玫瑰饼图 data :pieData[0]//数据赋值};
//将饼图添加到主 series 中,完成折线图和饼图的合并
balanceOp.series.push(pieChars1);
balanceOp.series.push(pieChars2);
```

5.5 本章小结

本章主要介绍了用户画像系统的实现，从系统依赖的软硬件平台搭建，系统数据采集层、业务处理层、展示层分别展开了阐述。数据层作为 Hadoop 大数据的核心功能部分，而业务处理层重点放在数据分析、标签形成、结果输出等功能上。下一章将测试工作及展示层进行效果展示与阐述。

第六章 用户画像系统测试与效果展示

6.1 系统测试

系统测试主要为了验证系统的稳定性、可靠性与健壮性，系统测试应尽早开始，根据项目管理 V 模型，测试工作与需求工作应是同时开始，并且与系统交付同时结束，期间与各个项目阶段相互对应测试，从而提高测试与项目工作质量。由于广元市贵商村镇银行无专业测试人员，测试工作较为薄弱，本论文中，该行主要通过展示层为入口，使用黑盒测试方法，根据所有需求人员、业务部门提交的需求，对需求开发的结果进行测试与校验。

6.1.1 系统测试目标与方法

本论文的系统测试目标为验证系统功能与业务需求是否一致。测试方法主要为黑盒测试，主要步骤为：系统讲解与培训、拟定测试计划、根据系统功能拟定测试案例、测试分工、测试记录统计、BUG 修改与回归测试，最终达到系统功能基本满足业务需求，达到上线运行标准。功能测试分配表 6-1 反映了系统主要功能模块与功能、简要描述了基本案例及对口测试业务部门。

表 6-1 功能测试分配表

模块名称	功能名称	测试案例	测试部门
用户管理	角色管理	新增、删除、停用用户及角色	个金部
	权限管理	调整对应角色的菜单权限	
	机构管理	增加、删除、修改机构信息	
	用户管理	新增、删除、修改用户信息	
标签管理	新建标签	新建标签	互金部
	标签审批	新建标签的审批、复核	
	标签入库	查看标签计算结果、启用标签	
	标签作废	删除无用标签	
用户画像	用户画像	输入姓名、电话、客户号进行画像	运营部
	结果导出	将画像导出成 pdf 或打印	
仪表盘	标签统计	查看结果各部门口径类报表	各部门
	存贷款监控	分支行存贷款数据	行领导
	全辖情况	全辖总体经营情况	行领导

6.1.2 系统重点功能测试

根据系统功能分配表，对各功能对应的测试扩口部门人员进行测试指导，通过系统基本功能培训后，开展系统重点功能的测试工作。主要包含基本功能测试、用户管理模块测试、标签管理模块测试、用户画像模块测试、仪表盘模块测试。

基本功能元素测试，验证主界面中所有链接、菜单、页面、按钮及输入等功能是否显示正常、是否跳转正常、是否提交正常、是否返回正常、是否输入校验正常。

用户管理模块测试，主要针对机构管理、用户管理、系统维护进行测试，主要测试功能点有：

新建功能测试：输入新机构、新用户，并对用户进行相应授权;；使用新建用户登录系统，查看该用户是否属于该新机构、是否拥有已授权菜单及功能。

修改功能测试：修改机构要素、用户要素、对用户授权进行增减；使用修改后用户登录系统，查看该用户相关信息是否已变更、权限是否已生效。

标签管理测试，对标签入库、标签审核、标签删除模块进行测试，主要测试功能点为：

标签入库测试：输入新标签 id、标签值等必输要素后，选择提交，查看是否提交成功，未审核的数据，本人可以进行修改与删除。

标签审核测试：审核人员或超级管理员权限查看，点击后查看当前提交的待审核标签，对待审核标签进行审核或驳回操作，查看是否生效。

标签删除测试：标签权重较低或词频较低的、或审核有误的标签，审核人员或超级管理员可以选择删除，选择对应的删除标签后，系统将不会对其计算。

用户画像测试，主要含个人客户画像查看、结果导出测试，主要测试功能点为：

个人客户画像测试：输入客户号、客户姓名或电话号码，点击提交，系统将展示该客户已匹配的标签结果。该页面主要包含四部分，一是展示客户的基本信息与权重最大的前三位标签，二是展示最近 1 年的月度收支流水对比图，三是展示其支出渠道偏好雷达图，四是展示该客户的标签云图。可以根据输入的测试客户数据、流水数据、标签云图进行后台数据比对，完成该界面的测试。

结果导出测试：可以再个人客户画像界面导出该客户的明细数据，也可通过本页面输入客户号、客户姓名或电话号码后，选择需要导出的标签信息、流水信息等。

仪表盘主要包含标签统计、存贷款监控、全辖情况，是对广元市贵商村镇银行自身的画像实现，主要测试功能点有：

存贷款监控测试：该模块可作为分支行经营画像使用，点击该菜单后，应显示全辖存款总额、贷款总额，并显示近 1 年存贷款比较柱状图与支行贡献度云图。存款贡献度系数为 0.4，贷款贡献度系数为 0.6。

全辖情况测试：该模块可作为仪表盘使用，点击该菜单后，应显示全行客户群年龄分部、存款结构对公对私比例、助农扶贫贷款比例与各支行几大支付渠道交易笔数柱状图。

测试结果如下表 6-2 所示。

表 6-2 测试结果表

模块名称	功能名称	测试结果	问题描述	是否影响上线
用户管理	角色管理	通过	无	否
	权限管理	通过	冲突菜单未限制	否
	机构管理	通过	无法增加网点，后期优化	否
	用户管理	通过	无	否
标签管理	新建标签	通过	增加字数限制长度	否
	标签审批	通过	无	否
	标签入库	通过	无	否
	标签作废	通过	无	否
用户画像	用户画像	通过	无	否
	结果导出	通过	无	否
仪表盘	标签统计	通过	无	否
	存贷款监控	通过	无	否
	全辖情况	通过	无	否

6.2 系统重点功能展示

展示层主界面用户管理模块，如下图 6-1 所示。系统该模块主要实现了用户权限配置与角色控制。通过新建权限，将权限 ID 与菜单 ID 匹配后，使权限生效。对用户分配权限时，系统根据不同权限 ID，初始化不同的功能菜单，实现系统角色区分与控制。

图 6-1 用户权限配置截图

标签管理界面。需要标签管理部门将需要申请的字段勾选后，选择对应日期，后台将根据提交的记录记性该字段的试跑批，计算该字段对应内容下的词频与标签结果。如 5-3 图所示。

图 6-2 标签试算字段勾选截图

用户画像界面，效果如下图 6-3 所示。个人客户用户画像功能根据输入的用户名、客户号或电话号码确定唯一客户，提交系统后，由展示层系统查询标签结果库，将该客户相关标签展示出来，使用云图等方式展现。该页面主要包含四部分，一是展示客户的基本信息与权重最大的前三位标签，二是展示最近 1 年的月度收支流水对比图，三是展示其支出渠道偏好雷达图，四是展示该客户的标签云图。

图 6-3 用户画像测试数据截图

数据报表与仪表盘界面，效果如下图 6-4 所示。数据报表与仪表盘中，存贷款监控展现了该行各条线与全行的数据情况，选择最顶级机构，展示全辖机构下所有分支行的资产、负债、支行经营指标排名等情况。

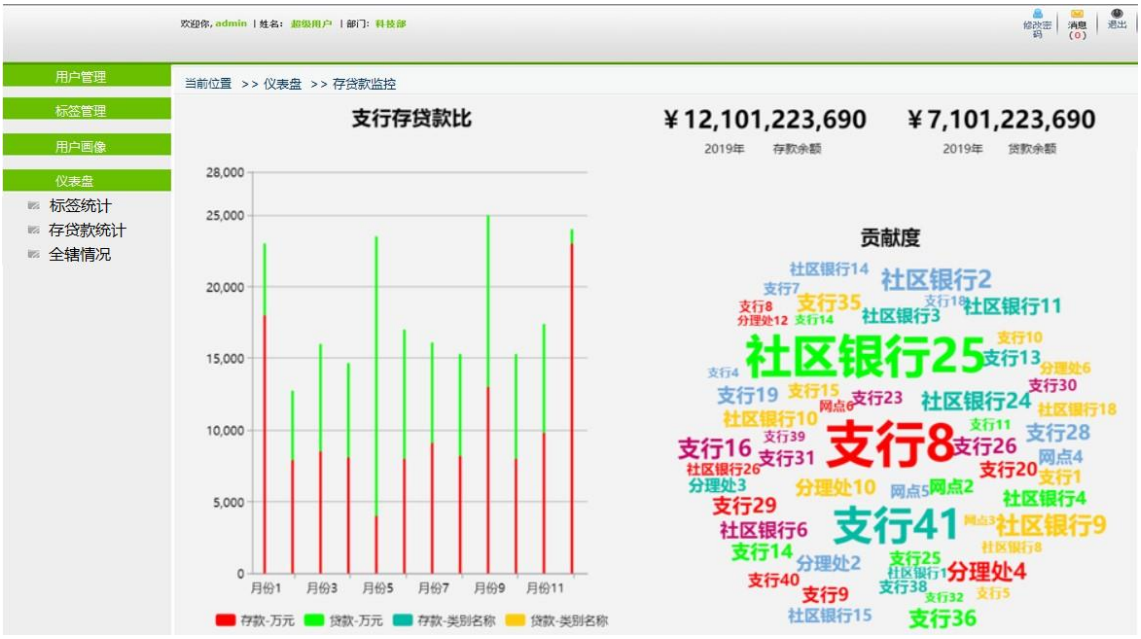


图 6-4 全辖仪表盘截图

全辖情况一览，如 6-5 所示，该仪表盘实现了对客户群年龄分布、个人存款对公存款占比、助农扶贫贷款占比及支行各渠道交易笔数的数据可视化展示，该功

能主要使用部门为运营部及其分管行长，为了美观，该功能中年龄分布使用了南丁格尔图、存款结构占比使用饼状图、助农扶贫贷款使用了环形图、支行渠道避暑使用了柱状图。

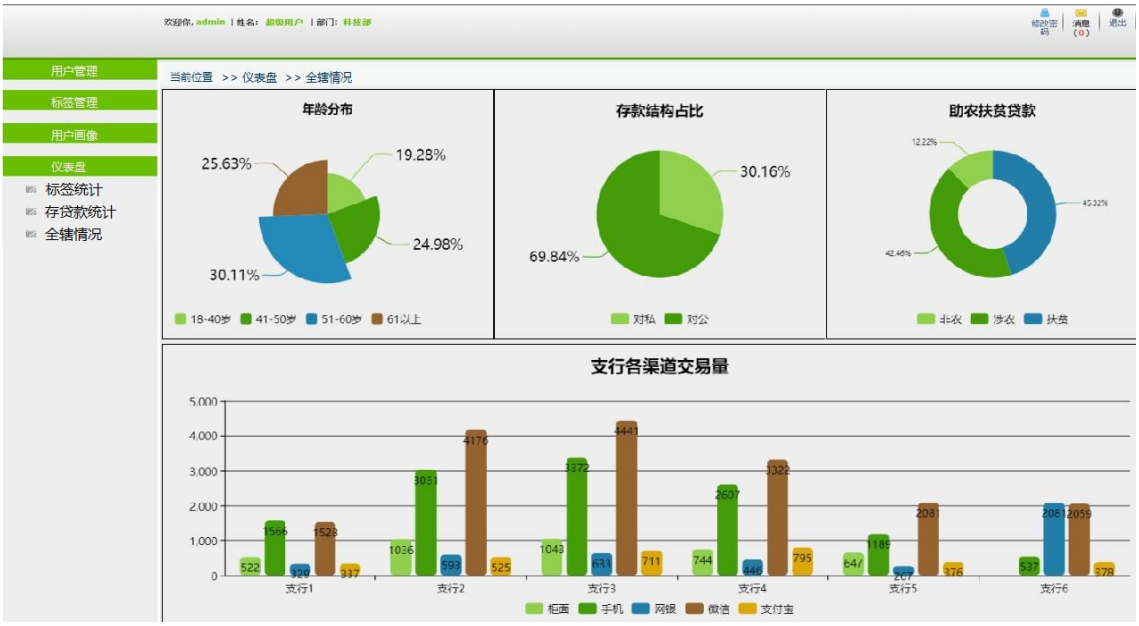


图 6-5 全辖仪表盘截图

标签统计情况如下图 6-6 所示，主要实现了我行贵宾客户占比，即存款日均数达到 100 万元及以上的个人客户，方便各支行对贵宾客户进行明细导出并开展后续工作。该功能还对营销活动、签约客户分布、交易客户分布进行了柱状展示。

图 6-6 渠道客户情况截图

6.3 本章小结

本章节对用户画像系统的测试工作进行了简要描述，同时对用户画像最终数据可视化的重点功能进行了展示与功能阐述。下一章将对前面几章的内容进行总结，并对今后工作提出展望。

第七章 总结与展望

7.1 全文总结

前章主要介绍了用户画像系统的实现与效果展示，重点对系统软硬件平台搭建、数据采集层、业务处理层展开了阐述，并对展示层及测试结果进行了效果截图展示。本章将对本课题内容进行全文总结，并对今后工作提出展望。

用户画像系统的意义对于广元市贵商村镇银行来说，更多的体现在了解客户、了解经营、指导经营上。本论文课题的重点为基于银行客户数据分析，但如何分析数据、分析什么样的数据才是有效的，本论文在编写过程中曾与广元市贵商村镇银行各部门骨干业务人员进行了多次沟通，并根据沟通结果确定了需求。

为了更好满足当下及以后不断产生的新需求，同时有效利用 Hadoop 大数据框架，灵活运用生态圈内各组件，并将其功能与广元市贵商村镇银行的实际需求结合起来，本论文在编写过程中也做了很多次尝试与调整。如安装没有选择使用 Ambari，而是先搭建 Hadoop 的 HDFS 底层，再逐个组件进行下载、安装与配置，通过此过程对 Hadoop 架构有了进一步的深入了解，在后期扩展时能够灵活的进行扩容增配。

有了可靠、可用、安全的大数据平台后，本论文根据各业务部门提交的需求对各业务系统的数据开展逐个采集工作，并通过 Sqoop、Flume 等工具对数据进行了初步的 ETL 工作，便于在 Hive 中对数据展开计算、分析使用。通过编写 HQL 实现业务需求，并将结果存储在 Hbase、MySql 中，最终由前端 UI 系统调用并显示。

本论文实现的用户画像系统包含了客户画像、自身画像两部分，将重点放在了银行数据分析上，业务逻辑、系统展现存在一些明显不足，分析有以下几个原因：

（1） 项目在工作中受重视程度不够，配合部门的力度不足，在前期调研、需求采集、系统设计等环节都较为被动，对该系统的期望不高。

（2） 业务人员能力不足，村镇银行的业务人员将更多的精力放在了营销与创收上，对自身的数据分析不到位，不知道自己手中数据背后的价值。

（3） 系统需求与设计环节重点不均，将更多的精力放在了大数据平台的需求与设计上，虽是为了规避需求不够明确带来的后期实现风险，但造成了系统目前功能不够完善，数据分析不够透彻，挖掘工作有待进一步深入，展示层系统需进一步细化提高使用友好度。

虽然有诸多不足，目前的用户画像系统功能不够完善，但也有其创新的意义

与价值，总结如下：

（1） 搭建了可靠的大数据平台，并能够根据后期需求灵活扩容增配，实现中长期需求的满足。

（2） 数据采集、数据逻辑分析工作已有一定成效，对今后的业务人员如何提出需求，提出什么样的需求有指导和借鉴意义。

（3） 搭建了能够灵活更换、调整系统架构，为后期进一步完善系统功能打下良好基础。

7.2 后续工作展望

经过不懈的努力与付出，本论文工作得到了一些成果，搭建的基于银行客户数据分析的用户画像系统也初步完成，但在很多方面都还不够深入、不够细化，需求也较为简单。希望在今后的工作中，能够多向行业中应用的画像类系统学习，指导本单位及自身将我行的数据更加有效的利用起来，分析出成果、挖掘出价值，为单位的管理政策制定、营销方案设计、支行客户了解等工作提供精准的数据支撑。需求明确是前提，所以在今后要更深入的了解我行各类业务及各系统，了解了业务数据，才能帮助、引导业务人员提出更合理、明确的需求，从而促使下一步分析工作、分析结果更高效、准确。总之该系统还有很多需要完善、优化的地方，数据采集与分析也应加入更多复杂的数据，通过需求与开发的不断迭代，将广元市贵商村镇银行用户画像系统建设的更加完善、稳定。

致　谢

在攻读硕士学位期间我完成了这篇硕士论文。首先衷心感谢我的导师陈波教授，感谢导师在我写作论文期间，不断的提醒与悉心的指导，帮助我在工作中不断提升自己的思维方式，给与了我很多中肯的意见，对指导我的论文修改、完善更是细致入微。陈波教授其渊博的专业知识，严谨的治学态度，精益求精的工作作风，让学生深受感动、获益匪浅。在这里，我向我的导师陈波教授表示最诚挚的谢意！

我还要感谢我的单位，感谢单位领导们对我工作的支持与帮助，让我能够全称参与、负责整个项目的开发与管理，更深入的了解了银行业务，提高了自己的各方面能力。

最后，我要感谢我项目组的同事还有我的家人，感谢他们对我的理解，大家能够达成共识，朝着共同的目标前进，并鼓励我、敦促我、帮助我坚持完成学业！

参考文献

[1] 中国网信网.中国互联网络发展状况统计报告[EB/OL]. http://www.cnnic.net.cn/hlwfzyj/hlwxzbg/hlwtjbg/201908/t20190830_70800.htm,2019年08月30日

[2] 刘伟,郑小六.中国商业银行产能利用率及影响因素[J].金融经济学研究(03):87-96

[3] 胡明国.大数据时代下客户立体画像在银行业应用研究[J].中国城市金融, 2016(1):40-42

[4] 丁咏.数据挖掘在商业银行客户关系管理中的应用[D].重庆:重庆大学计算机系,2006,35-50

[5] Alan Cooper. The origin of personas[EB/OL]. https://www.cooper.com/journal/2008/05/the_origin_of_personas. 2005

[6] M. Pazzani, J. Muramatsu, D. Billsus. Identifying interesting web sites[C].1996

[7] 高玉龙.基于文本挖掘的用户画像研究[D].汕头:汕头大学,2014,28-39

[8] 张万军.基于大数据的个人信用风险评估模型研究[D].北京:对外经济贸易大学, 2016,41-55

[9] 孙浩楠.左边还是右边?电商网站个性化推荐位置研究[D].大连:大连理工大学, 2015,13-19

[10] 朱静.互联网金融背景下商业银行发展策略研究[J].中国市场(35):58-59

[11] G. Amato, U. Straccia. User profile modeling and applications to digital libraries[C]. The 3rd European Conference on Research and Advanced Technology for Digital Libraries, 1999, 184-197

[12] R. M. Quintana, S. R. Haley, A. Levick, et al. The persona party: using personas to design for learning at scale[J]. CHI Conference Extended, 2017:933-941

[13] S. Gauch, M. Speretta, A. Chandramouli, et al. User profiles for personalized information access[C].The Adaptive Web,Methods and Strategies of Web Personalization DBLP,2007,54-89

[14] D. Travis. E-Commerce Usability[M]. 2002

[15] J. A. Iglesias, P. Angelov, A. Ledezma, et al. Creating evolving user behavior profiles automatically[J]. IEEE Transactions on Knowledge and Data Engineering,2012,24(5):854-867

[16] S. Ramkumar, G. Emayavaramban, A. Elakkiya. A web usage mining framework for mining evolving user profiles in dynamic web sites[J]. International Journal of Advanced Research in Computer Science and Software Engineering,2017:889-894

[17] 王宪朋.基于视频大数据的用户画像构建[J].电视技术,2017,41(6):20-23

[18] 余孟杰.产品研发中用户画像的数据建模[J].设计艺术研究,2014,4(6):62-64

[19] 百度百科.大数据[EB/OL]. https://baike.baidu.com/item/%E5%A4%A7%E6%95%B0%E6%8D%AE/1356941?fr=aladdin

#ref_[1]_13647476. 2008 年 8 月

[20] V. Gottfried. Big data as the new enabler in business and other intelligence[J]. Vietnam Journal of Computer Science, 1(1):3-14

[21] X. Qin, B. Qin, C. Li, et al. Landscape of Unified Big Data Platforms[J]. 2014

[22] 邓杰. Hadoop 大数据挖掘从入门到进阶实战[M]. 机械工业出版社, 2018, 24-25

[23] 田承平. 基于 B/S 架构的统一配置管理系统的设计与实现[D]. 南京:东南大学, 2014,15-25

[24] 张智, 龚宇. 分布式存储系统 HBase 关键技术研究[J]. 现代计算机(专业版)(32):35-39

[25] 张野. Hadoop Hive 实现日志数据统计[J]. 电脑编程技巧与维护(4):115-117

[26] S. Hoffman. Apache Flume: Distributed Log Collection for Hadoop[J]. 2015

[27] 陈吉荣, 乐嘉锦. 基于 Hadoop 生态系统的大数据解决方案综述[J]. 计算机工程与科学, 2013, 35(10):25-35

[28] 万向怡. 一种基于 Hadoop 的电商数据分析系统的设计与实现[D].杭州:浙江大学, 2016,11-18

[29] 董慧嵘. 基于学习行为分析的学习能力评估与学习意图预测方法研究[D].哈尔滨:哈尔滨工业大学,2019,9-11

[30] 中国互联网证券用户画像[J].现代商业银行,2018(17):19-23

[31] 陆岷峰, 虞鹏飞. 互联网金融背景下商业银行“大数据”战略研究——基于互联网金融在商业银行转型升级中的运用[J]. 经济与管理(3):33-40

[32] 卢小宾,王涛.Google 三大云计算技术对海量数据分析流程的技术改进优化研究[J].图书情报工作,2015(3):6-11

[33] 裴国才.基于用户画像的电信精准营销模型研究[J].信息通信, 2017, 000(012):240-241,243

[34] 贾怡敏.银行要跟上客户需求的变化价值.金融博览[M].北京:中国金融出版社,2008

[35] 刘寒, 孙晶. 金融业大数据应用研究[J]. 电信网技术, 000(2):9-13

[36] W. J. Xu, N. R. Juri, A. Gupta, et al. Supporting large scale connected vehicle data analysis using HIVE[C]. 2016 IEEE International Conference on Big Data (Big Data). IEEE, 2016

[37] 冯朝阁. 基于 YARN 的工业大数据处理平台研究与实现[D]. 西安:西安电子科技大学, 2015,25-37

[38] 赵颖. Hadoop 环境下的动态资源管理研究与实现[D]. 上海:上海交通大学,2015,28-35

[39] N. Maheswari, M. Sivagami. Large-Scale Data Analytics Tools: Apache Hive, Pig, and HBase[M]. Data Science and Big Data Computing,2016

[40] 张悦, 杨学全. 基于 Hadoop+HBase+Hive 分布式技术的云计算平台的设计[J]. 现代装饰(理论), 2014(9):256-256

[41] J. Rod. Expert One-on-One J2EE Design and Development[M]. Wrox Press Ltd.2002

[42] W. U. Renbiao, L. Chao, Q. U. Jingyi. Storage method for flight delay platform based on HBase and Hive[J]. Journal of Computer Applications, 2018

[43] 宋美娜,崔丹阳,鄂海红,等.一种通用的数据可视化模型设计与实现[J]. 计算机应用与软件(9):45-49+103

[44] 吴义.基于 Hadoop 和 Django 的大数据可视化分析 Web 系统[D].上海:东华大学,2016,6-20

[45] B. Z. Wu, S. Z. Liang, D. X. Niu. Building Web Application System Based on Architecture of Struts 2 & Spring & Hibernate[J]. Computer & Modernization, 2009

[46] J. Beres. Build a Better UI[J]. Visual Studio Magazine, 2004, 14(10):p.36-37,39-40

[47] K. Ting, J. J. Cecho. Apache Sqoop Cookbook[M]. 2013

[48] Pippal, Sanjeev, Singh, et al. Data Trasfer From MySQL To Hadoop: Implementers' Perspective[J]. 2014

[49] S. Hoffman. Apache Flume: Distributed Log Collection for Hadoop[J]. 2015

[50] L. Bhushan. Implementing SQOOP and Flume-based Data Transfers[M]. Apress, 2016

[51] S. Wadkar, M. Siddalingaiah. Log Analysis Using Hadoop[M]. Pro Apache Hadoop. 2014

[52] K. Pushpendra, S. T. Ramjeevan. A Framework for Weblog Data Analysis Using HIVE in Hadoop Framework[M]. Proceedings of International Conference on Recent Advancement on Computer and Communication. 2018

[53] N. Maheswari, M. Sivagami. Large-Scale Data Analytics Tools: Apache Hive, Pig, and HBase[M] . Data Science and Big Data Computing. 2016

[54] Z. Zhao, Y. Jiang. An efficient Join-Engine to the SQL query based on Hive with Hbase[C]. 2015

[55] E. Capriolo, D. Wampler, J. Rutherglen. Programming Hive[M]. 2013

[56] H. Lu. HBase Write Performance Optimizations[J]. 2014, 9:67-72

[57] 马延辉,孟鑫,李立松. HBase 企业应用开发实战[M]. 2014

[58] 石凤贵.基于 TF-IDF 中文文本分类实现[J].现代计算机,2020(06):51-54+75

[59] 赵银春,付关友,朱征宇.基于 Web 浏览内容和行为相结合的用户兴趣挖掘.计算机工程[J].2005(12):93-94

[60] 刘越,李锦涛,虎嵩林.基于代价估计的 Hive 多维索引分割策略选择算法[J]. 计算机研究与发展, 2016, 53(4):798-810