大学生论文检测系统

文本复制检测报告单(全文标明引文)

№:ADBD2022R_20220610232355466974318791　　检测时间:2022-06-10 23:23:55

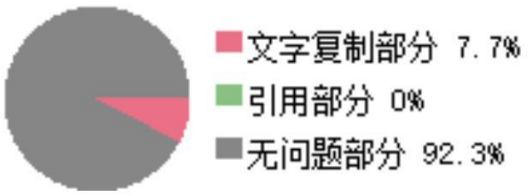
篇名: 基于Hadoop的电商用户行为数据分析系统的设计与实现
作者:
指导教师:
检测机构: 浙江工业大学
提交论文IP: 36.***.***.***
文件名: XXX-2毕业论文.docx
检测系统: 大学生论文检测系统
检测类型: 大学生论文
检测范围: 中国学术期刊网络出版总库
中国博士学位论文全文数据库/中国优秀硕士学位论文全文数据库
中国重要会议论文全文数据库
中国重要报纸全文数据库
中国专利全文数据库
图书资源
优先出版文献库
大学生论文联合比对库
互联网资源(包含贴吧等论坛资源)
英文数据库(涵盖期刊、博硕、会议的英文数据以及德国Springer、英国Taylor&Francis 期刊数据库等)
港澳台学术文献库
互联网文档资源
源代码库
CNKI大成编客-原创作品库
机构自建比对库
时间范围: 1900-01-01至2022-06-10

可能已提前检测，检测时间：2022/6/2 20:56:00，检测结果：8%

检测结果

去除本人文献复制比： 7.7%　　跨语言检测结果：0%
去除引用文献复制比：7.7%　　总文字复制比：7.7%
单篇最大文字复制比：3.7%　（张锐_201950915330_基于Hadoop的电商用户行为分析系统的设计与实现_计算机科学与技术_张玉中）

重复字数：	[2216]	总段落数：	[8]
总字数：	[28722]	疑似段落数：	[6]
单篇最大重复字数：	[1066]	前部重合字数：	[126]
疑似段落最大重合字数：	[1612]	后部重合字数：	[2090]
疑似段落最小重合字数：	[48]		



指标： 疑似剽窃观点　疑似剽窃文字表述　疑似整体剽窃　过度引用

相似表格： 0　　相似公式： 没有公式　　疑似文字的图片： 0

0%(0)　　0%(0)　　中英文摘要等（总2567字）

2.7%(55)	2.7%(55)	第一章绪论（总2059字）
2.3%(71)	2.3%(71)	第二章核心技术介绍（总3030字）
2.8%(48)	2.8%(48)	第三章系统需求分析（总1733字）
6.7%(373)	6.7%(373)	第四章系统设计（总5609字）
16.1%(1612)	16.1%(1612)	第五章系统实现_第1部分（总10012字）
0%(0)	0%(0)	第五章系统实现_第2部分（总2731字）
5.8%(57)	5.8%(57)	第六章总结与展望（总981字）

（注释：无问题部分　　文字复制部分　　引用部分）

指导教师审查结果

指导教师:

审阅结果:

审阅意见:　指导老师未填写审阅意见

1. 中英文摘要等　　总字数：2567

相似文献列表

去除本人文献复制比：0%(0)　　文字复制比：0%(0)　　疑似剽窃观点：(0)

原文内容

本科毕业设计说明书（论文）
（2022届）
论文题目基于Hadoop的电商用户行为
数据分析系统的设计与实现
作者姓名
指导教师
学科(专业) 软件工程（大数据方向2）
所在学院计算机科学与技术学院、软件学院
提交日期 2022年6月

摘要

由于互联网带来的便利，日益增长的用户体验产生了持续增长的海量数据，只有取其精华、去其糟粕，才能提炼出对企业有用的信息，这让大数据处理技术显得尤为重要。数据分析技术的价值在于如何对数据进行准确的分析，使数据分析的结果显示出一定的联系，从而获得最具商业价值的数据信息。在很多电商网站的运营过程中，正确的数据分析结果对平台的发展起到了至关重要的作用。传统的大数据分析只能处理关系型数据库中的结构化数据，而不能处理大量的半结构化或非结构化数据。

为了解决传统数据分析方法在处理结构复杂的海量数据方面的局限性，本文基于Hadoop架构设计并实现了的适用于电商网站的数据分析平台，主要功能包括数据采集、数据计算和数据可视化三大模块。其中，数据采集模块选用Flume、Kafka来收集电子商务系统前端埋点的用户行为数据；数据计算模块运用Hive对数据进行分层处理，并进行保存、清洗、合并、拆分、统计等工作，以达到数据解析的目的；数据可视化模块使用Sqoop将结果数据导入MySQL，并采用Superset对最终需求的实现进行Web页面展示。

使用本平台，企业可以通过对用户行为的探究，反思其经营发展状况，并对业务需求进行改善和优化，给用户提供更好的体验，从而创造进一步的商业价值。

关键词：用户行为，数据分析，数据仓库，Hadoop，Flum，Hive

Abstract

Due to the convenience brought by the Internet, the increasing user experience has produced a continuous growth of massive data. Only by taking the essence and discarding the dregs can useful information for enterprises be extracted, which makes the big data processing technology seem particularly important. The value of data analysis technology lies in how to accurately analyze data and make the results of data analysis show a certain relationship symbol, so as to obtain the data information that shows the most commercial value. In the operation process of many e-commerce websites, correct data analysis results play a key role in the development of the platform. The traditional big data analysis can only deal with the structured data in the relational database, but cannot deal with the massive semi-structured or unstructured data.

In order to solve the limitations of traditional data analysis methods in dealing with massive data with

complex structure, this paper designs and implements a data analysis platform suitable for e-commerce websites based on Hadoop architecture, with three main functions including data collection, data calculation and data visualization. Among them, the data acquisition module uses Flume and Kafka to collect the user behavior data of the embedded point in the front-end of e-commerce system. Data calculation module uses Hive to stratified data processing, and save, clean, merge, split, statistics and other work, in order to achieve the purpose of data analysis; The data visualization module uses Sqoop to import the result data into MySQL, and uses Superset to display the realization of the final requirements on Web pages.

Using this platform, enterprises can reflect on their business development by exploring user behaviors, improve and optimize business needs, provide users with better experience, and thus create further business value.

Keywords: User behavior, Data analysis, Data warehouse, Hadoop, Flum, Hive

目录

图目录

表目录

2. 第一章绪论 总字数：2059

相似文献列表

去除本人文献复制比：2.7%(55) 文字复制比：2.7%(55) 疑似剽窃观点：(0)

1	基于SOA架构的食品电子商务平台的构建 赵瑾(导师：苏生;张爱英) - 《电子科技大学硕士论文》- 2010-09-01	2.7%（55） 是否引证：否

原文内容

第一章绪论

1.1 研究背景

现如今，由于人们对网络技术的研究和互联网的蓬勃发展，使网络技术在近些年获得了很大的发展。特别是随着像淘宝、京东这样的电子商务网站的出现，网络购物以其便捷和快速而获得了许多民众的喜爱，而这种“线下线上相结合”的商业模式也使网络渐渐地变成了我们生活中不可或缺的部分。随着我国网民规模的持续扩大，企业的竞争压力也在增加，许多公司正面临着衰落或破产的困境[1]。国内外的学者都对这一现象进行了深入的研究，但大多数都是基于传统的研究方法，如问卷调查、焦点访谈等。它们总是基于消费者的消费因素或满意度，无法获得消费者内心最深处的想法或更深层次的价值驱动。只有在企业了解了消费者之后，根据他们内心深处的感受可以为消费者提供更有针对性、更满意的产品，以达到促进销售的目的。因此，在商业竞争中，利用数据分析来支持运营已成为一种有效的手段。

传统的分析方法耗费大量昂贵的计算资源，效能和安全性都不高，而有效的时间分割和对计划任务的合理分配则可以借助更复杂的程序来完成。Hadoop分布式技术的发展将有助于缓解这些问题，它是一种大规模的数据分析平台，目前广泛应用于大型应用，拥有大量廉价的硬件设备，也是一个高度可靠、可扩展、并行的分布式系统[2]。企业可充分利用Hadoop集群的计算能力与存储容量，实现企业对海量数据信息资源的管理。

在此背景下，本文构建了一个基于Hadoop的数据仓库平台，并利用Hadoop的并行计算能力对电子商务数据进行分析和处理，从而帮助企业更清楚地了解客户的兴趣。然后为消费者提供更具体的产品或服务，并满足消费者的实际需求。这将有助于推广产品，提高企业在新形势下的竞争力。

1.2 研究现状

数据仓库，不同于数据库，它是利用海量历史数据面向实际需求进行指标分析，通过分析结果为整个公司各个部门的运作提供决策支持，从而提高企业产品竞争力的工具[3]。数据仓库，尽管名称听上去很像，但它并不是存储数据的地方，而是为合理地存储数据做好准备。这些准备包括对数据的：备份、清理、聚合、统计等等。数据仓库的输入系统，通常是指前台埋点的使用者操作数据信息、后台生成的服务数据信息，以及一些爬虫数据信息等；输出系统通常为报表管理系统、使用者画像系统、推荐管理系统、机器学习系统(多用于语音识别、图像识别)、风险监控系统(多用于金融行业)等。

随着网络数据的爆炸式增长和高速接入，网络数据分析面临着可扩展性问题，数据仓库在实际的生产运用中发挥了重大作用。虽然国外电子商务企业在一开始并不看好数据仓库的应用价值，但随着企业之间竞争压力的加大，数据仓库在电子商务应用领域的逐步实施，越来越多的企业利用数据仓库对网站上的海量数据进行数据分析，以辅助企业运营，提升自身产品的竞争力。各大公司推出的数仓框架也逐渐变多，如Apache的开源Hive框架、Cloudera公司推出的企业级大数据管理平台CDH等[4]。

数据仓库的概念在国内引入较晚，但由于市场竞争非常激烈，数据仓库发展迅速。例如，阿里巴巴推出了基于阿里云的数据仓库系统。早在2008年，阿里巴巴就开始通过Hive架构研发使用Hadoop技术的大数据平台“云梯”[4]。在国家的大力支持下，国内越来越多的IT企业自主研发了众多的数据分析平台与大数据平台，如帆软、神策数据、Gowing IO等。

1.3 论文主要内容

为了解决传统数据分析在处理大量数据时的局限性问题，本文在深入分析主流大数据平台相关技术的基础上，设计并实现了基于Hadoop的电子商务数据分析系统。本数据仓库系统基于海量数据生成环境构建，其功能涵盖数据采集、数据计算、数据可视化展示等大数据处理的整个过程，在数仓搭建的过程中，也对Hadoop、Flume、Hive等相关技术工具进行了大量调研，对电商领域的重要需求指标进行了实现。系统通过Flume-Kafka-Flume来实现海量电商用户行为数据的ETL，并落盘到HDFS中，然后使用Hive搭建分层的数据仓库，对不同维度的数据进行统计分析，并将底层默认的MapReduce计算引擎换成Spark，提高执行效率。采用Azkaban对数仓业务实现全流程自动化调度，提高开发效率。最后实现电商业务需求的可视化。

1.4 章节安排

第一章调研了论文主题的研究背景和研究现状，并对论文进行了主要工作概述和章节安排。

第二章介绍了系统实现涉及的相关核心技术，分析调研了Hadoop下的HDFS、MapReduce两大关键部分和Flume、Hive的架构和原理。

第三章从功能性需求和非功能性需求两个方面对系统实现展开了需求分析。

第四章对系统主要功能模块进行了详细设计，包括采集模块业务设计、数据仓库业务设计和数据可视化业务设计。

第五章根据系统设计对系统进行了总体实现，描述了环境搭建步骤和各模块的详细实现，重点展现了数仓搭建和需求指标分析的实现过程。

第六章进行了总结，并归纳了系统的待改进之处。

第二章核心技术介绍

数据分析平台最重要的技术核心是数据的存储与分析，而在基于Hadoop的框架下进行分析和处理的时间明显优于传统的方法，也更适用于电商用户行为分析系统。本数据分析系统平台研发拟采用的组件有Hadoop、Hive、Flume、Zookeeper、Kafka、Azkaban、Sqoop、Superset等，以下对系统研发过程中要用到的核心组件和对分析系统搭建过程中需要了解到的数仓理论进行了简单介绍。

2.1 Hadoop架构

Hadoop是一个大众流行的大数据处理平台，在广义的概念上指的是Hadoop生态圈[5]。Hadoop具有优秀的跨平台能力，可以部署到成本较低的计算机集群中。目前Hadoop主要由HDFS、MapReduce、Yarn和Common四部分组成。Hadoop有1.0、2.0和3.0版本，在Hadoop1.0中，MapReduce具有业务逻辑运算和资源调度两方面的功能，而Hadoop2.0和Hadoop3.0中将两种功能分开，MapReduce只负责计算，Yarn负责资源调度。下面介绍Hadoop的关键组成部分，高可用性分布式存储系统HDFS和并行计算平台MapReduce。

2.1.1 HDFS架构原理

数据在Hadoop集群上处理之前，必须先存储在Hadoop集群中。数据保存功能主要是通过将控制系统使用的大量电子商务数据存储在HDFS中来实现。大多数电子商务数据是非结构化的。为了提高系统效率、可扩展性和数据安全性，可以使用HDFS来存储数据。

HDFS是一种分布式文件系统，用于以流数据访问方式存储大文件。它可以通过主/从结构来管理资源文件系统[5]。如图2-1所示，HDFS集群由一个主节点NameNode（名称节点）和多个从节点Datanode（数据节点）组成[6]。NameNode管理文件系统的命名空间结构，并保护整个文件系统树或任何文档系统或文档系统树中的目录的元数据信息。它还记录每个文档中每个块的数据节点信息。客户机想要读数据块时，它首先会联系指定的NameNode并通知客户端将数据块存储在什么地方，然后在新发现的数据节点上读取新的数据，或在名称节点的统一调度下重新创建、复制和删除数据块[6]。

图 2-1 HDFS整体结构[6]

HDFS的特点是：数据写入一次，读取多次。HDFS的优点有：高容错性(多副本机制)，适合处理大数据(数据规模可达GB、TB级别)、存储大文件(文件规模可达百万以上)，可构建于廉价机器上。

2.1.2 MapReduce计算框架

MapReduce是一种分布式计算方式。MapReduce从不同的角度分析数据，如产品类型、期限、价格等，得到整齐有序的数据。MapReduce分布式计算任务基本过程可以概括为：读取文件数据，先进行Map（映射）处理，再进行Reduce（归约）处理，最

后将处理结果写入文件[6]。详细过程如图2-2所示。

图 2-2 MapReduce计算模型[6]

Map阶段获取切割后的数据并将这些键值对输入值<k1，v1>映射成中间输出值<k2，v2>。在此过程中，Map各函数独立工作，互不影响。之后，MapReduce通过Combine函数合并处理中间结果，再用Shuffle函数排序数据，可以加速数据传输，减小Reduce函数的工作压力。Reduce阶段对键值对输入值<k2，list<v2>>中同一个key的value累加，得到合并后的新键值对<k3，v3>作为输出。

MapReduce框架具有以下优点：

(1) 易于编程：用户只需要关心业务逻辑，简单地实现框架接口。

(2) 良好的可扩展性：当资源得不到满足时，只需扩充服务器就能够实现动态地扩充计算能力。

(3) 高容错性：当任何一台机器出现故障时，任务可以转移到另一个节点，这个过程由Hadoop内部完成，无需人工干预。

(4) 适合海量数据计算(TB、PB级别)：允许数千台服务器集群并发工作、共同计算。

同时具有以下缺点：

(1) 不擅长实时计算：MapReduce无法达到类似于MySQL等关系型数据库的毫秒级查询速度。

(2) 不擅长流型计算：Sparkstreaming、Flink等流算法可以动态输入大量数据，而MapReduce则是静态数据源。

(3) 不擅长DAG有向无环图计算：Spark在这方面做得更好。

2.2 Flume原理

Flume是一种分布式日志收集系统，它可以有效地收集、聚集和移动大量日志数据（包括日志文档和事件），并将应用程序产生的数据存储到任何集中存储设备（如HDFS、HBase）中[7]。它是一个高安全性和可分配的工具，一般用来把流数据信息(日志数据信息)从多个web服务器上复制到HDFS。

Flume的event被定义为具有字节负载和一组可选字符串属性的数据流单元，Agent是托管这些event从外部数据源到下一个目的地的进程。如图2-3所示，Agent由三个组件构成，其用途是从服务端接受数据并转送到存储器中。

图 2-3 Flume核心组件[7]

组件详细介绍如下：

(1) Source从数据生成器接收数据，并以Flume的event格式将其传输到一个或多个Channel[7]。Flume支持多种类型的Source，每个Source从指定的数据生成器接收event。

(2) Channel是一个临时存储，它从Source接收event，并充当缓冲区，直到event被sinks消耗，它充当Source和Sink之间的桥梁。Channel是一个完整的事务，它保证了发送和接收数据的一致性，可以连接任意数量的Source和Sink[7]。

(3) Sink将数据送往集中存储设备，如HDFS、HBase等。它使用来自Channel的数据（events），并将其传递到目的地。目的地可能是另一个Sink、HDFS、HBase等[7]。

2.3 Hive架构

Hive构建在Hadoop上，是一种数据信息仓储工具。它提供一种类SQL语句，称为HQL(Hibernate Query Language)[8]，可以对海量日志数据进行查询分析，完成数据仓库任务。Hive 的核心架构主要包含四个部分：用户接口、元数据、Hadoop和驱动器，如图2-4所示。

图 2-4 Hive架构[8]

(1) 用户端口(Client)：主要有三个，CLI、thriftClient和WUI。当中最常见的是CLI(command line interface)，在企业内可以使用手堡垒机连接ssh hdp_lbg_ectech@10.126.101.7，可以直接进入Hive，也可以连结到HiveServer。

(2) 元数据(Metastore)：用来存放和管理与Hive有关的元数据，默认上存放到自带的数据库系统中，也支持转换成MySQL等关系式数据库。这些元数据通常包含表名、表的属性、分区和目录等。

(3) Hadoop部分：HDFS存储Hive生成的数据，MapReduce对数据进行处理和计算。

(4) 驱动器(Driver)：由编译器、优化器、执行器组成，通过分析HQL语言和优化编译器形成一个自动执行计划，然后使用MapReduce计算架构[8]。

第三章系统需求分析

子商务网站产生的海量用户行为数据，如果被加以利用，是可以提取出宝贵信息是。为了挖掘数据的潜在价值，有必要建立数据仓库系统。数据仓库系统从数据采集过程开始，将数据统一收集到数据仓库中进行合理的分析、分类、存储和计算。

在开始本项目之前，规划一个全面的需求分析是很重要的，这样开发人员可以了解产品的需求，然后检查系统功能和业务流程是否完成，以及需要改进的地方。如果没有详细的需求说明，在项目的开发工作完成之后要进行的改动将会对项目进度产生不利影响。本章主要针对电子商务数据分析项目的需求，分别对功能需求和非功能需求进行描述，从而确定系统的开发目标。

3.1 功能需求分析

本数据仓库系统主要实现数据采集平台、数据计算平台和数据可视化三大功能模块，如图3-1所示。

图 3-1 系统功能模块

1. 用户行为数据采集功能

用户在启动和使用应用时中都会产生大量的日志数据，如启动应用会产生加载时间、广告播放时间、广告页面ID等信息，用户浏览页面会有日志记录，跳转到新页面会有跳入时间的信息，用户操作过程中也可能出现错误等。这些数据通过埋点的方式获取并以JSON格式存储在日志服务器中[9]。大数据分析系统的海量数据都是采集自日志服务器中的埋点用户行为日志，这些日志数据主要包括页面数据、事件数据、曝光数据、启动数据和错误数据五类数据[10]。采集模块的主要任务就是将这些用户行为日志数据收集存储到Hadoop上，采集过程中要保证数据传输效率并具备缓存功能。

2. 数据仓库分析功能

使用Hive对采集到的日志数据进行离线分析，并将分析结果分级存储在数据仓库中。这提高了分析结果的可重用性，简化了分析结果的管理。本系统主要对设备相关指标进行分析，最终要求实现如图3-2所示需求。

图 3-2 用户行为指标分析

在移动统计中，用户是通过设备来判断的，每个独立设备代表一个独立用户。

(1) 活跃设备数指标分析用以统计当日、本周和本月的活跃设备数。启动过应用即为活跃。周和月指的是某个自然周和某个自然月。对于在某周(月)内启动了应用的设备，在该周(月)内多次启动，只计算一个活动设备。

(2) 新增设备数指标分析用以统计该应用被首次使用的设备数量。首次安装并打开使用某个APP的用户被当做新增用户；若卸载重装再次使用该APP，则不能被看做新增用户。这里统计的是每日新增用户。

(3) 沉默设备数指标分析用以统计从首次安装使用过一次后，超过一周未再次使用应用的设备数。

(4) 流失设备数指标分析用以统计一周内未活跃的设备数。

(5) 本周回流设备数指标分析用以统计上周未使用而这周使用了应用的设备数，也就是这周活跃的沉默设备数。

(6) 设备留存率指标分析用以统计某段时间内的新增设备，经过一段时间后继续使用应用的留存设备，占当时新增设备的比例。

(7) 持续活跃设备数指标分析用以统计指定时间段内连续活跃的设备数量。例如近3周活跃设备数、近7天内持续3天活跃设备数等。

3. 可视化功能

需求指标分析完后，为直观地展示数据分析结果，需要将结果数据导入MySQL，供用户查询或进行可视化展示。

3.2 非功能需求分析

非功能需求是指软件产品为满足用户业务需求而必须具有且除功能需求以外的特性[11]。做好非功能需求的实现可以提高产品质量，使产品的适用性更加普遍，增强用户体验。本系统需要满足以下几个方面的非功能需求：

(1) 稳定性

本数仓系统的硬件环境和软件环境须满足长时间高效的稳定运行，避免出现影响系统稳定性的因素。

(2) 可靠性

在系统发生程序故障、断电、节点宕机等突发情况时，保证数据完整不丢失，系统可以快速重启。面对数据采集过程中可能出现的异常，系统能够自己处理某个环境出现的错误，及时抛出异常信息，保证系统能够回归正常运行状态。

(3) 可扩展性

随着平台的搭建和完善，会有新服务的增加或新框架的替换，这需要系统有良好的的可扩展性。

(4) 易用性

要注重代码编写规范，添加必要的注释，统一接口和SQL语句，方便相关人员理解沟通；数据可视化界面应简单直观，使得用户能够快速上手使用。

一个Flume-Kafka-Flume的结构，其中两个Flume的作用是不同的，前者是采集层Flume，后者是消费层Flume。

首先，在服务器将模拟生成的数据写入磁盘日志文件后，采用Flume实时收集落盘到文件的数据，通过配置Source和Sink可以控制接收和发送的数据。Channel可以保存从Source接收的数据并直接传递给下一个组件，避免数据丢失。

为防止采集层Flume数据量过大而导致消费层Flume无法及时处理数据造成数据堵塞，增加Kafka用作日志处理的临时存储，从而降低文件上传到HDFS的速率，保证采集通道的稳定运行。

最后通过消费层Flume来读取Kafka缓存中的数据，传输到HDFS上，为数据仓库提供原始数据。Hadoop中的HDFS起到海量数据的存储作用。

该模块采用的数据传输方案如图4-2所示。

图 4-2 数据传输方案

4.3 数据仓库业务设计

数据仓库是电商系统中的核心部分。数仓搭建的设计需要确定要分析的主题领域[12]。本系统针对用户行为数据进行分析，因此以电商用户行为主题作为研究领域，对采集通道收集的日志文件进行分析，找出用户行为数据的内部规律，为企业决策提供参考。使用Hive作为分析工具，底层配置Spark引擎，提高执行速度。数据保存、清理、合并、拆分和收集，这个过程主要是通过对数据仓库的合理划分来完成的，数据仓库分为ODS、DW、DWS、DWT、ADS五层，在本系统中每层的作用如表4-1所示。

表 4-1 数据仓库分层设计

数据仓库层次名称	作用
ODS (Operation Data Store)原始数据层	存放采集至HDFS的原始数据，直接加载原始数据，无需复杂的逻辑处理
DWD (Data Warehouse Detail)明细数据层	粒度和结构与原始表一样，对ODS层的非结构化用户行为数据进行轻度清洗
DWS (Data Warehouse Service)服务数据层	每天对DWD层数据进行汇总，统计各个主题对象的当天行为，构建多个主题宽表
DWT (Data Warehouse Topic)数据主题层	进一步聚合DWS层数据，统计各个主题对象的累积行为，构建多个主题的全量宽表
ADS (Application Data Store)数据应用层	分析电子商务系统的主题指标，存储数据分析结果，为各种统计报表提供数据

数据仓库层次名称作用

ODS (Operation Data Store)原始数据层存放采集至HDFS的原始数据，直接加载原始数据，无需复杂的逻辑处理

DWD (Data Warehouse Detail)明细数据层粒度和结构与原始表一样，对ODS层的非结构化用户行为数据进行轻度清洗

DWS (Data Warehouse Service)服务数据层每天对DWD层数据进行汇总，统计各个主题对象的当天行为，构建多个主题宽表

DWT (Data Warehouse Topic)数据主题层进一步聚合DWS层数据，统计各个主题对象的累积行为，构建多个主题的全量宽表

ADS (Application Data Store)数据应用层分析电子商务系统的主题指标，存储数据分析结果，为各种统计报表提供数据

对数据仓库分层有以下几方面的好处：

(1) 将复杂的问题简化[13]。使数据仓库结构更加清晰，复杂的大问题被分解到每层变成简单的小任务，每一层都有其定位和作用。在使用表时便于快速定位和分析问题，识别影响范围，若某些数据出现错误，只用修改这个范围内的数据，方便维护数据的准确性。

(2) 减少重复开发。通过规范数据仓库的分层处理，可以提高一些常用中间层的分析结果的重用率，减少重复计算，节省计算资源[14]。

(3) 隔离原始数据。将原始的真实数据与用于分析的统计数据解耦，以避免原始数据变化或异常带来的影响，出现问题可以在中间层兼容处理。

设计数据仓库分层之前，还须对命名规范进行约束，这样可以更加方便地管理数据仓库。规范各层各表命名，可以通过表名直观地区分该表属于哪一层哪个主题以及这张表大概是用来干嘛的。通常每层表都以该层名称开头，如ODS层以“ods_表名”形式命名、临时表一般包含tmp、用户行为表以log为后缀。脚本以“数据源_to_目标.sh”形式命名。

4.3.1 数仓设计-ODS层

ODS层存放整个数仓最原始的数据，直接加载从HDFS收集的原始日志，不做任何处理，保持数据的完整性。ODS层持久化最细粒度的数据，而其它层次则是经过计算和汇总的粗粒度数据，只有ODS层能满足详细数据的查询需求[15]。该层只用创建一张日志表，由于日志数据都是JSON格式，表中只用一个字段存储加载的所有数据信息就行。由于日志数据量很大，表存储时需要进行压缩。因为该层只存放原始数据，所以不涉及逻辑处理，不提供复杂的查询功能。ODS层的表信息如表4-2所示，日志表的设计如表4-3所示。

表 4-2 数据仓库ODS层设计

序号	表名称	表含义
1	ods_log	日志表

序号表名称表含义

1 ods_log 日志表

表 4-3 日志表字段信息

字段名称	数据类型	字段描述
line	string	日志数据

字段名称数据类型字段描述

line string 日志数据

4.3.2 数仓设计-DWD层

DWD层的任务主要是基于ODS层对日志数据解析和对核心数据进行判空过滤。从这一层开始，设计数据仓库的逻辑处理，并将数据作为数据仓库计算和分析的基础。在ODS层的设计中，日志数据被存储为只有一个字段的分区表。日志表中的日志结构分为普通页面埋点日志和启动日志两种，需要分别从中解析出启动日志、页面日志、动作日志、曝光日志和错误日志到5张表中，具体如表4-4所示。

表 4-4 数据仓库DWD层设计

序号	表名称	表含义
1	dwd_start_log	启动日志表
2	dwd_page_log	页面日志表
3	dwd_action_log	动作日志表
4	dwd_display_log	曝光日志表
5	dwd_error_log	错误日志表

序号表名称表含义
1 dwd_start_log 启动日志表
2 dwd_page_log 页面日志表
3 dwd_action_log 动作日志表
4 dwd_display_log 曝光日志表
5 dwd_error_log 错误日志表

以启动日志表（dwd_start_log）为例，其字段信息如表4-5所示。

表 4-5 启动日志表字段信息

字段名称	数据类型	字段描述
area_code	string	地区编码
brand	string	手机品牌
channel	string	渠道
model	string	手机型号
mid_id	string	设备id
os	string	操作系统
user_id	string	会员id
version_code	string	app版本号
entry	string	icon 手机图标 notice 通知 install 安装后启动
loading_time	bigint	启动加载时间
open_ad_id	string	广告页 ID
open_ad_ms	bigint	广告总共播放时间
open_ad_skip_ms	bigint	用户跳过广告时间
ts	bigint	启动时间

字段名称数据类型字段描述
area_code string 地区编码
brand string 手机品牌
channel string 渠道
model string 手机型号
mid_id string 设备id
os string 操作系统
user_id string 会员id
version_code string app版本号
entry string icon 手机图标 notice 通知 install 安装后启动
loading_time bigint 启动加载时间
open_ad_id string 广告页 ID
open_ad_ms bigint 广告总共播放时间
open_ad_skip_ms bigint 用户跳过广告时间
ts bigint 启动时间

4.3.3 数仓设计-DWS层

DWS层以ODS层作为数据源，关联ODS层的多张表，对用户行为数据将DWD层的表引入时间维度进行汇总，通常时间维度有日、周、月等。该层只创建一张每日设备行为表，用于统计设备主题对象的当天行为，每行对应一个设备主题对象一天的数据。对该宽表设置设备id为主键，以天作为数据维度进行轻度汇总，导入当天活跃的所有设备信息，这样就可以为后续的数据分析做准备，利用这些数据统计每日的活跃设备数，累积后统计每周、每月的活跃设备数，以及沉默设备数、流式设备数等等，最后将一系列的分析结果支持精细化报表，协助企业运营决策。该层表信息如表4-6所示，表字段信息如表4-7所示。

表 4-6 数据仓库DWS层设计

序号	表名称	表含义
1	dws_uv_detail_daycount	每日设备行为表

序号表名称表含义
1 dws_uv_detail_daycount 每日设备行为表

表 4-7 每日设备行为表字段信息

字段名称	数据类型	字段描述
mid_id	string	设备id
brand	string	手机品牌
model	string	手机型号
login_count	bigint	活跃次数
page_stats	array<struct<page_id:string,page_count:bigint>>	页面访问统计

字段名称数据类型字段描述
mid_id string 设备id
brand string 手机品牌
model string 手机型号
login_count bigint 活跃次数
page_stats array<struct<page_id:string,page_count:bigint>> 页面访问统计

4.3.4 数仓设计-DWT层

DWS层对设备主题以天为单位进行聚合，得到每天设备主题的相关度量数据，DWT层将在此基础上进一步聚合，得到设备主题的全量宽表，重点关注累计至今和累计某段时间的度量值、首次和末次时间。该层表信息如表4-8所示，表字段信息如表4-9所示。

表 4-8 数据仓库DWT层设计

序号	表名称	表含义
1	dwt_uv_topic	设备主题宽表

序号表名称表含义
1 dwt_uv_topic 设备主题宽表

表 4-9 设备主题宽表字段信息

字段名称	数据类型	字段描述
mid_id	string	设备id
brand	string	手机品牌
model	string	手机型号
login_date_first	string	首次活跃时间
login_date_last	string	末次活跃时间
login_day_count	bigint	当日活跃次数
login_count	bigint	累计活跃天数

字段名称数据类型字段描述
mid_id string 设备id
brand string 手机品牌
model string 手机型号
login_date_first string 首次活跃时间
login_date_last string 末次活跃时间
login_day_count bigint 当日活跃次数
login_count bigint 累计活跃天数

4.3.5 数仓设计-ADS层

ADS层作文整个数据仓库最高的一层，存储了粒度最高的数据，直接对接最终的统计报表，为企业运营提供决策支撑[16]。该层的主要任务为，基于DWD、DWS、DWT层的数据，对设备主题面向实际需求进行指标分析，形成各种统计报表，最终导入到关系型数据库（RDBMS）如MySQL中以供数据应用系统查询使用。该层表的数量较多，对应要分析的指标创建了8张表，如表4-10所示。

表 4-10 数据仓库ADS层设计

序号	表名称	表含义
1	ads_uv_count	活跃设备数表
2	ads_new_mid_count	每日新增设备表
3	ads_silent_count	沉默用户数表
4	ads_wastage_count	流失用户数表
5	ads_back_count	本周回流用户数表
6	ads_user_retention_day_rate	留存率表
7	ads_continuity_wk_count	最近连续3周活跃用户数表
8	ads_continuity_uv_count	最近7天内连续3天活跃用户数表

序号表名称表含义
1 ads_uv_count 活跃设备数表
2 ads_new_mid_count 每日新增设备表
3 ads_silent_count 沉默用户数表
4 ads_wastage_count 流失用户数表
5 ads_back_count 本周回流用户数表
6 ads_user_retention_day_rate 留存率表
7 ads_continuity_wk_count 最近连续3周活跃用户数表
8 ads_continuity_uv_count 最近7天内连续3天活跃用户数表

以活跃设备数表（ads_uv_count）为例，其字段信息如表4-11所示。

表 4-11 活跃设备数表字段信息

字段名称	数据类型	字段描述
dt	string	统计日期
day_count	bigint	当日用户数量
wk_count	bigint	当周用户数量

mn_count	bigint	当月用户数量
is_weekend	string	Y/N 是否是周末用于得到本周最终结果
is_monthend	string	Y/N 是否是月末用于得到本月最终结果

字段名称数据类型字段描述
dt string 统计日期
day_count bigint 当日用户数量
wk_count bigint 当周用户数量
mn_count bigint 当月用户数量
is_weekend string Y/N 是否是周末用于得到本周最终结果
is_monthend string Y/N 是否是月末用于得到本月最终结果

4.4 数据可视化业务设计

仅将分析结果以数据表格的形式展现难以发现其中的规律，在与用户交互过程中不友好，所以需要用图形化界面来直观地展示结果数据信息，便于用户理解[17]。数据展示层主要工作为根据ADS层的结果在MySQL中创建相应的表，使用Sqoop将最终需求结果数据定期导入MySQL，并使用Superset将结果数据直观地显示在Web页面上。

4.5 运行环境

1. 系统开发环境和工具

(1) 系统平台：Microsoft Windows 10，Linux CentOS 7

(2) 开发工具：IDEA、Maven

电商用户行为数据分析系统的软件全部运行在Linux系统上。为了简化部署，系统使用VMware虚拟机构建Hadoop集群。一台主机作为NameNode主节点，配置16G内存和50G硬盘；另外两台主机作为DataNode从节点，配置4G内存和50G硬盘。安装并配置Hadoop、Zookeeper、MySQL、Hive、Flume、Kafka等软件。

在Windows环境下主要是用来编写Flume拦截器等代码，最终也会部署到大数据集群。

采用IDEA工具编写代码，最终用Maven进行项目的构建打包。

2. 服务器集群规划

(1) 消耗资源的软件要尽量分开安装在不同服务器

(2) 传输数据比较紧密的Zookeeper、Kafka 、Flume安装在同一台服务器

(3) 为了方便外部访问，将客户端放置在同一台服务器上

服务器集群规划具体见表4-12。

表 4-12 服务器集群规划

服务名称	子服务	hadoop102	hadoop102	hadoop102
HDFS	NameNode	√		
	DataNode	√	√	√
	SecondaryNameNode			√
Yarn	NodeManager	√	√	√
	ResourceManager		√	
Zookeeper	Zookeeper Server	√	√	√
Flume	Flume	√	√	
Kafka	Kafka	√	√	√
Flume	Flume			√
Hive	Hive	√		
MySQL	MySQL	√	√	√
Sqoop	Sqoop	√		
Superset	Superset	√		
Azkaban	AzkabanWebServer	√		
	AzkabanExecutorServer	√		
服务数总计		12	7	7

服务名称子服务 hadoop102 hadoop102 hadoop102
HDFS NameNode √
DataNode √ √ √
SecondaryNameNode √
Yarn NodeManager √ √ √
ResourceManager √
Zookeeper Zookeeper Server √ √ √
Flume Flume √ √
Kafka Kafka √ √ √
Flume Flume √
Hive Hive √
MySQL MySQL √ √ √
Sqoop Sqoop √
Superset Superset √
Azkaban AzkabanWebServer √
AzkabanExecutorServer √

指　　标

疑似剽窃文字表述

1. loading_time bigint 启动加载时间
   open_ad_id string 广告页 ID
   open_ad_ms bigint 广告总共播放时间
   open_ad_skip

2. ring 手机品牌
   model string 手机型号
   login_count bigint 活跃次数
   page_stats array<struct<page_id:string,page_count:bigint>>

6. 第五章系统实现_第1部分　　总字数：10012

相似文献列表

去除本人文献复制比：16.1%(1612)　　文字复制比：16.1%(1612)　　疑似剽窃观点：(0)

1	张锐_201950915330_基于Hadoop的电商用户行为分析系统的设计与实现_计算机科学与技术_张玉中 张锐 - 《大学生论文联合比对库》- 2021-05-20	6.9%（693） 是否引证：否
2	张锐_201950915330_基于Hadoop的电商用户行为分析系统_计算机科学与技术_张玉中 张锐 - 《大学生论文联合比对库》- 2021-05-24	5.9%（590） 是否引证：否
3	张锐_201950915330_基于Hadoop的电商用户行为分析系统的设计与实现_计算机科学与技术_张玉中 张锐 - 《大学生论文联合比对库》- 2021-06-04	5.9%（590） 是否引证：否
4	侯思露-杰普大数据-5720171305-基于hadoop的电商用户行为分析 杰普大数据 - 《大学生论文联合比对库》- 2021-05-31	5.5%（549） 是否引证：否
5	秦军浩-基于Hadoop的电商用户行为分析系统设计与实现 秦军浩 - 《大学生论文联合比对库》- 2021-05-19	5.2%（521） 是否引证：否
6	秦军浩-基于Hadoop的电商用户行为分析系统设计与实现 秦军浩 - 《大学生论文联合比对库》- 2021-05-24	5.0%（496） 是否引证：否
7	1706142224_李越_电商用户行为数据分析系统的设计与实现 李越 - 《大学生论文联合比对库》- 2021-06-02	3.0%（301） 是否引证：否
8	1706142224_李越_电商用户行为数据分析系统的设计与实现 李越 - 《大学生论文联合比对库》- 2021-05-19	2.9%（295） 是否引证：否
9	1706142224_李越_电商用户行为数据分析系统的设计与实现 李越 - 《大学生论文联合比对库》- 2021-05-21	2.9%（295） 是否引证：否
10	1706142224_李越_电商用户行为数据分析系统的设计与实现 李越 - 《大学生论文联合比对库》- 2021-05-17	2.8%（278） 是否引证：否
11	201502731961_马越_基于Hadoop的用户行为分析系统设计与实现 马越 - 《大学生论文联合比对库》- 2019-06-03	2.2%（223） 是否引证：否
12	大数据数据采集与数仓系统开发 张忠伟 - 《大学生论文联合比对库》- 2020-06-05	1.9%（189） 是否引证：否
13	大数据数据采集与数仓系统 张忠伟 - 《大学生论文联合比对库》- 2020-06-07	1.7%（167） 是否引证：否
14	大数据数据采集与数仓系统开发 张忠伟 - 《大学生论文联合比对库》- 2020-06-16	1.7%（167） 是否引证：否
15	毕业论文 - 《大学生论文联合比对库》- 2020-11-17	1.7%（167） 是否引证：否
16	1612190309_张忠伟_大数据数据采集与数仓系统开发(蒲飞)_毕业论文	1.7%（167）

	张忠伟 - 《大学生论文联合比对库》- 2020-11-17	是否引证：否
17	基于媒体融合的广播互动平台探讨	0.9%（92）
	张娇; - 《电声技术》- 2018-05-05	是否引证：否
18	基于大数据的电子书城数仓搭建及报表分析	0.9%（89）
	李开放 - 《大学生论文联合比对库》- 2020-05-20	是否引证：否
19	牛客网在线学习数据分析系统的设计与实现	0.5%（53）
	罗小兵 - 《大学生论文联合比对库》- 2020-05-03	是否引证：否
20	2016330109_栗世帅_牛客网在线学习数据分析设计与实现_赵怡	0.5%（53）
	栗世帅 - 《大学生论文联合比对库》- 2020-05-06	是否引证：否

原文内容

第五章系统实现

5.1 实验环境搭建

5.1.1 Hadoop环境搭建

本数据分析系统基于Hadoop框架构建，首要任务就是搭建Hadoop实验环境。Hadoop有3种运行模式：本地运行模式、伪分布式运行模式、完全分布式运行模式[18]。由于系统数据存储在HDFS上，即多台服务器同时工作，因此选择完全分布式运行模式。搭建Hadoop集群时，每台服务器的配置是类似的，所以只需在hadoop102节点上配置好再分发到其它服务器。

(1) 准备3台虚拟机，配置好IP、主机名、主机名称映射和改windows的主机映射文件（hosts文件）等信息。

(2) 安装JDK。配置环境变量并使其生效。

(3) 安装Hadoop。下载Hadoop安装包，上传并解压到/opt/module/目录下。配置环境变量并使其生效。

(4) 修改配置文件（在Hadoop的etc/hadoop目录下）。

配置core-site.xml，如图5-1所示。

图 5-1 core-site.xml配置

配置hdfs-site.xml，如图5-2所示。

图 5-2 hdfs-site.xml配置

配置yarn-site.xml，如图5-3所示。

图 5-3 yarn-site.xml配置

配置mapred-site.xml，如图5-4所示。

图 5-4 mapred-site.xml配置

配置workers，如图5-5所示。

图 5-5 workers配置

(5) 为方便分发文件到其它节点，需要编写集群分发脚本，并将配置好的文件分发到其它节点上。

(6) 启动Hadoop集群。第一次启动时要格式化NameNode（在hadoop102上）[18]。在配置了 NameNode的hadoop102上启动HDFS，在配置了ResourceManager的hadoop103上启动YARN，用jps命令可查看进程状态，在web端可以查看HDFS 上存储的数据信息和YARN上运行的Job信息。

在浏览器输入http://hadoop102:9870，出现如图5-6所示界面。

图 5-6 HDFS的NameNode界面

在浏览器输入http://hadoop103:8088，出现如图5-7所示界面。

图 5-7 YARN的ResourceManager界面

(7) 由于Hadoop集群的启动停止命令太多，每次启动都要敲一遍太麻烦，所以需要将hdfs、yarn、historyserver的启动停止命令集中编写成脚本方便操作。

(8) 配置Hadoop支持LZO压缩。在Flume传输数据到HDFS的过程中，需要将数据保存为压缩格式，这对海量数据的存储尤为重要。由于采集到HDFS的用户行为日志通常数据量非常庞大，对一些大文件的处理可能还会涉及到切片操作，所以选用压缩率较高且支持切片的LZO压缩格式。将编译好后的 hadoop-lzo-0.4.20.jar上传到Hadoop的share/hadoop/common目录下并同步到其它节点。在core-site.xml中添加如图5-8所示配置。

图 5-8 core-site.xml配置

5.1.2 Zookeeper环境搭建

本系统使用Kafka作为数据缓冲区。在分布式环境下，要求Kafka集群中所有节点的配置信息是一致的[19]。因此需要安装Zookeeper对其进行统一配置管理，保证Kafka的正常运行。

(1) 下载Zookeeper安装包，上传并解压到/opt/module/目录下。

(2) 配置zoo.cfg文件。重命名/opt/module/zookeeper-3.4.10/conf下的 zoo_sample.cfg为zoo.cfg，并修改数据存储路径为后续创建的文件夹zkData，添加server id和节点IP路径的对应关系配置。

(3) 配置服务器编号。在安装目录下创建zkData文件夹用于存储zookeeper相关数据。在zkData目录下创建并编辑myid文件，按照zoo.cfg文件的配置添加与server对应的编号。

(4) 将配置好的文件同步到其它机器，注意分别在其它机器上修改myid文件的内容，分别在3台服务器上启动Zookeeper。

(5) Zookeeper不提供脚本用来同时在多个服务器上启动。因此，为了方便操作，要将Zookeeper的启动和停止命令封装到自定义脚本中。

5.1.3 MySQL环境搭建

(1) 用rpm -qa|grep mariadb命令检查，若安装过MySQL，需要卸载重装。
(2) 下载MySQL安装包，上传并解压到/opt/software/目录下。
(3) 执行rpm安装。
(4) 使用root权限查看存储在/root/.mysql_secret中的临时密码，登录MySQL后要及时修改。
(5) 初始化MySQL数据库。
(6) 启动MySQL服务。
(7) 登录MySQL数据库。及时修改密码并记住。

5.1.4 Hive环境搭建

(1) 下载Hive安装包并解压到/opt/module/目录下，重命名解压后的文件夹为hive。配置环境变量并使其生效。
(2) Hive连接第三方数据库MySQL需要将MySQL的JDBC驱动拷贝到Hive的lib目录下。
(3) 配置Metastore到MySQL配置Hive的conf目录下的hive-site.xml文件。
(4) 登录MySQL，新建Hive元数据库并初始化。
(5) 检查Hive是否能启动。
(6) 配置Spark引擎，为数仓搭建做准备。

5.1.5 Sqoop环境搭建

(1) 下载Sqoop安装包并解压到/opt/module/目录下,重命名解压后的文件夹为sqoop。配置环境变量并使其生效。
(2) 在Sqoop的conf目录下，重命名配置文件sqoop-env-template为sqoop-env.sh，并修改其中的配置，添加相关软件路径。
(3) 将JDBC驱动拷贝到Sqoop的lib目录下。
(4) 验证Sqoop是否安装成功。
(5) 测试Sqoop是否能够成功连接数据库。

5.1.6 Superset环境搭建

(1) 安装Python环境。下载Miniconda安装包，上传并解压到/opt/module/目录下，配置环境变量并使其生效。创建Python3.6环境，激活Superset环境。
(2) 部署Superset。安装Superset依赖，安装（更新）setuptools和pip，安装Superset。初始化Superset数据库。创建管理员用户用于后续登录Superset。初始化Superset。
(3) 安装gunicorn，启动Superset。
(4) 将Superset启停命令封装成脚本方便操作。
(5) 登录Superset。

5.2 数据采集模块实现

5.2.1 采集日志到Kafka层配置

采集日志Flume层的主要任务是将日志从落盘文件中采集出来发往Kafka的topic_log主题。针对本系统的需求，在编写Flume Agent配置文件之前，需要进行组件选型。

(1) Source选型

应该选用TailDir Source从动态写文件夹读取数据。Exec Source在Flume停止后容易造成数据丢失；Spooling Directory Source不能实时采集数据；而TailDir Source可以记录数据读取位置，实现断点续传而不丢失数据，还可以监控多个目录，达到实时采集数据的目的[20]。

(2) Channel选型

因为选择Kafka作为缓存，所以选用Kafka Channel，也就不需要sink了。

(3) Flume拦截器

采集日志到kafka的过程中需要进行一个初步的简单的数据清洗，过滤掉格式不正确的不合法的脏数据，这可以通过部署Flume ETL拦截器来实现。ETL 代表Extract（抽取）-Transform（转换）-Load（加载）[21]。

(4) 详细配置

采集层Flume的具体配置思路如图5-9所示。Flume从/opt/module/applog/log目录下直接读取模拟生成的文件名为日期格式的log日志数据。为了对日志数据进行初步过滤，需要部署一个拦截器。然后根据业务需求编写采集方案配置文件，结合Flume官网配置，具体描述Flume三大组件的配置信息，用于将日志发往Kafka缓存的topic_log主题。最后，为方便执行调用，将采集层Flume程序的启动停止命令封装成脚本，放到家目录的bin下。

图 5-9 采集层Flume实现

在Flume的conf目录下配置 file-flume-kafka.conf 文件，如图5-10所示。

图 5-10 采集层Flume配置

5.2.2 消费Kafka日志的Flume配置

将日志缓存到Kafka后，要最终落盘到HDFS上，该部分工作也交由Flume完成。

(1) Source选型

因为数据源保存在Kafka缓存中，消费层Flume主要从Kafka读取消息，所以选用Kafka Source。

(2) Channel选型

要保证数据不丢失应该选用FileChannel。尽管MemoryChannel传输效率更高，但容易丢失数据，无法保障数据质量[21]。

(3) Sink选型

选用HDFS Sink将日志落盘到HDFS上。

(4) 详细配置

消费层Flume的具体配置思路如图5-11所示。根据组件选型，结合官网编写消费方案配置文件，使Flume读取Kafka中topic_log主题的数据，存储到HDFS上的/origin_data/gmall/log/topic_log目录下。最后将消费层Flume程序的启动停止命令

封装到脚本中。日志收集全部完成后，需要将数据采集模块内的所有命令封装成采集通道的启停脚本，以便后续工作。

图 5-11 消费层Flume实现

在Flume的conf目录下配置 kafka-flume-hdfs.conf ，如图5-13所示。

图 5-13 消费层Flume配置

5.3 数仓模块实现（ODS-DWT）

数据计算模块通过使用Hive调用HDFS上的日志数据搭建数据仓库并进行离线分析来实现。本系统总共搭建了ODS、DWD、DWS、DWT、ADS五层数仓，实现了从ODS层最细粒度的JSON数据转化为ADS层经过汇总统计后的最粗粒度的数据分析结果，最终精细化数据报表。再导入采集到的日志数据之前，需要在数据仓库创建相应的数据库和表，首先建立gmall数据库，用于存放本项目的所有数仓表。HDFS上的Hive数据仓库存储如图5-14所示。

图 5-14 HDFS上的Hive数据仓库目录

5.3.1 数仓搭建-ODS层

ODS层直接把HDFS中的日志数据拉取到大数据集群，保持数据原貌不做更改，不涉及任何逻辑处理，既作为数据备份也作为后续大数据计算的元数据。针对用户行为日志数据创建一张日志表。

建表思路：

(1) 若要创建的表已存在，则要先删除该表重新创建。

(2) 创建外部表，这样删除表数据时只会删除元数据，而保留表的原始数据，更加安全。

(3) 表每行的字段就是一个string类型的JSON，JSON数据中存储的就是一个个键值对，之后DWD层将从每行的JSON字符串中提取出相应的值。

(4) 创建分区表，按照日期创建分区，以供后续按照日期进行查询。

(5) 指定存储方式为LZO压缩格式处理，减少磁盘空间占用。

(6) 指定日志表在HDFS上的存储路径在/warehouse/gmall/ods/下。

由于HDFS不支持对LZO压缩格式文件进行分片，所以还需要为LZO压缩文件创建索引。最后将导入数据和创建索引的语句封装成加载数据脚本hdfs_to_ods.sh方便执行。

ODS层具体实现如图5-15所示。

图 5-15 ODS层具体实现

5.3.2 数仓搭建-DWD层

DWD层对ODS层的日志表进行数据解析，将每行的JSON字符串转化为一个个单一的字段加载到启动日志表、页面日志表、动作日志表、曝光日志表和错误日志表5张表中。创建表时，同样创建外部表和分区表（按照日期分区），采用parquet列式存储，保存为LZO压缩格式以减少存储空间占用，并指定5张表存储在HDFS上的/warehouse/gmall/dwd/路径下。

(1) 启动日志解析：启动日志表每行对应设备的一次启动记录，需要包含公共信息、启动信息和启动时间。过滤出包含“start”字段的日志，然后用get_json_object函数解析每个字段。

(2) 页面日志解析：页面日志表每行对应一个页面浏览记录，需要包含公共信息、页面信息和跳入当前页时间。过滤出包含“page”字段的日志，解析每个字段。

(3) 动作日志解析：动作日志表中的每一行记录页面上的一个动作，需要包含公共信息、页面信息和动作信息。过滤出包含“action”字段的日志，解析每个字段。还需要用自定义UDTF函数来“炸开”动作数组，将一条包含多个动作记录的日志解析成多条只包含一个动作记录的日志。

(4) 曝光日志解析：曝光日志表每行对应用户在某个页面的一个曝光记录，需要包含公共信息、页面信息、曝光信息和跳入当前页时间。过滤出包含“display”字段的日志，解析每个字段。和动作日志解析一样，需要用自定义UDTF函数来“炸开”曝光数组。

(5) 错误日志解析：错误日志表每一行对应一条应用使用过程中的错误记录，为了便于定位错误，需要保留与错误相关的所有因素。因此，一条错误记录要包含与其对应的公共信息、页面信息、启动信息、动作信息、曝光信息、错误信息和时间。过滤出包含“error”字段的日志，解析每个字段。若要分析错误与单个动作和曝光的关系，则可以先用自定义UDTF函数来“炸开”动作数组和曝光数组，再解析字段。

通过自定义UDTF函数解析日志数组，可以达到“一进多出”的效果，将一条包含多个记录的日志解析成多条只包含一个记录的日志。实现函数需要编写Java程序，打包上传到HDFS 的/user/hive/jars路径下，并创建永久函数关联开发好的Java Class。

自定义UDTF函数的工作流程如图5-16所示。

图 5-16 自定义UDTF函数流程图

根据上述日志解析思路，创建5张表，并将所有表的导入数据语句封装成脚本ods_to_dwd.sh方便执行。

DWD层具体实现如图5-17所示。

图 5-17 DWD层具体实现

5.3.3 数仓搭建-DWS层

DWS层以设备为主题构建一张每日设备行为宽表，统计设备主题对象的当天行为，主要按照设备id进行分组统计，每行对应一个设备一天的累计行为。

解析思路：

(1) 创建外部表和分区表（按照日期分区），使用parquet格式存储，保存为LZO压缩格式，并指定存储在HDFS上的/warehouse/gmall/dws/路径下。

(2) 表字段包含3个设备信息和2个与流量相关的汇总值，在这里把设备信息字段看成一个整体，选择和分组时都一起写，方便查询分析。

(3) 对DWD层启动日志表按照设备id聚合，得到一个设备一天的活跃次数。

(4) 对DWD层页面日志表按照设备id和页面id聚合，得到一个设备一天的页面访问统计。由于一个设备一天可能浏览多个页

面，所以需要用数组来保存多个页面的访问统计。先按照页面id聚合，得到一个设备在一个页面的访问次数。再将页面id和页面放访问次数封装到结构体中，使用collect_set函数实现聚合，存放到数组。

(5) 对聚合后的临时表进行全外联，使用nvl函数过滤出非空数据。

将该层导入数据语句封装成脚本dwd_to_dws.sh方便执行。

DWS层具体实现如图5-18所示。

图 5-18 DWS层具体实现

5.3.4 数仓搭建-DWT层

DWT层构建设备主题宽表，统计设备主题对象的累积行为，以设备id作为唯一标识，每行对应一个设备截止到当天的全量设备行为，每天将新增的设备信息添加到表中，更新首末次活跃时间和当日及累计活跃次数，为后续分析设备相关指标做准备。

解析思路：

(1) 创建外部表，无需分区，使用parquet格式存储，保存为LZO压缩格式，并指定存储在HDFS上的/warehouse/gmall/dwt/路径下。

(2) 表字段包含3个设备信息和4个累计值。

(3) 将DWT层设备主题表作为旧表、DWS层每日设备行为表作为新表，全外联得到非空数据。

将该层导入数据语句封装成脚本dws_to_dwt.sh方便执行。

DWT层具体实现如图5-19所示。

图 5-19 DWT层具体实现

5.4 数仓搭建-ADS层

ADS层对实际的具体需求进行实现，对电商系统的设备主题指标分别进行分析，每个分析结果都有对应的日期字段，数据从DWS和DWT层导入，如果要算历史上某几天的数据，就从DWS层导入，如果算的是截止至今的累计数据就从DWT层导入。在该层创建外部表，存储在HDFS上的/warehouse/gmall/ads/路径下。表不用分区，因为一天就插入一条数据；不用进行列式存储，因为ADS层存放的已经是结果数据了，之后使用Sqoop全表导出MySQL，不用选特定的哪几列导出；也不用压缩，因为一天就只有一条数据，数据量很少。需要注意导入数据时，使用insert into追加会有任务重跑数据重复问题和HDFS小文件问题，应该使用insert overwrite+union取出原文件数据和新增数据合并去重后再覆盖插入。最后将所有表的导入数据语句封装成脚本dwt_to_ads.sh方便执行。

ADS层具体实现如图5-20所示。

图 5-20 ADS层具体实现

5.4.1 指标分析-活跃设备数

需求定义：

(1) 日活：当日活跃的设备数。

(2) 周活：当周活跃的设备数。

(3) 月活：当月活跃的设备数。

思路分析：指标以统计日期作为标识，对DWT层设备主题宽表按照日期聚合，得到日活、周活、月活。3个活跃设备数指标都统一计算频率，一天算一次，非周末月末的日期就计算截止到当天之前一天的结果即可。还要增加判断当天是否是周末和月末的字段，以得到周活、月活的最终结果。

活跃设备数表（ads_uv_count）字段为日期、日活、周活、月活、是否是周末、是否是周末，统计结果如图5-21所示。

图 5-21 活跃设备数指标分析结果

5.4.2 指标分析-每日新增设备

需求定义：当日首次活跃的设备数。

思路分析：指标以统计日期作为标识，对DWT层设备主题宽表按照首次活跃时间字段聚合，得到首次活跃时间为当天的某天的新增设备数。

每日新增设备数表（ads_new_mid_count）字段为统计日期、新增设备数，统计结果如图5-22所示。

图 5-22 每日新增设备数指标分析结果

5.4.3 指标分析-沉默设备数

需求定义：首次活跃时间在7天前，之后未再活跃的设备数。

思路分析：指标以统计日期作为标识，对DWT层设备主题宽表按照首次活跃时间字段聚合即可。只在首次使用应用时活跃过，即首次末次活跃时间都在启动应用当天，累计活跃天数也只有那一天；在7天前活跃，即首次末次活跃时间小于等于7天前。

沉默设备数表（ads_silent_count）字段为统计日期、沉默设备数，统计结果如图5-23所示。

图 5-23 沉默设备数指标分析结果

5.4.4 指标分析-流失设备数

需求定义：最近7天未活跃的设备数。

思路分析：指标以统计日期作为标识，对DWT层设备主题宽表按照末次活跃时间字段聚合即可。最近7天未活跃，即末次活跃时间在7天前。

流失设备数表（ads_wastage_count）字段为统计日期、流失设备数，统计结果如图5-24所示。

图 5-24 流失设备数指标分析结果

5.4.5 指标分析-本周回流设备数

需求定义：上周未活跃，本周活跃，且不是本周新增的设备数。

思路分析：

(1) 指标以统计日期作为标识，一天计算一次本周一截止到当天的回流数，直到周末得到最终结果。统计日期所在周字段以“周一日期_周日日期”形式作为周id，用以区分是哪周的回流设备。

(2) 先查询出本周活跃设备，然后去掉其中的本周新增和上周活跃设备，即可得到本周回流设备。

(3) 本周活跃设备可以从DWT层查询，末次活跃时间在本周范围内，非本周新增设备也可以通过首次活跃时间在本周一之前过滤出来；上周活跃设备要从DWS层查询，活跃时间在上周范围内，按照设备id聚合；最后用left join上周活跃设备，选出右表为空的，即可从本周活跃设备中去掉上周活跃设备，需要注意由于HQL不能在where中放子查询，所以不能用not in来过滤掉上周活跃设备，否则会报错。

本周回流设备数表（ads_back_count）字段为统计日期、统计日期所在周标识、回流设备数，统计结果如图5-25所示。

图 5-25 本周回流设备数指标分析结果

5.4.6 指标分析-留存率

在互联网行业中，开始使用一个应用程序一段时间，并在一段时间后继续使用该应用程序的用户被认为是留存用户[21]。这些用户占当时新增用户的比例就是留存率，每1个单位时间(日、周、月)计算一次。顾名思义，留存率指的就是“有多少用户留下来了”。留存用户和留存率反映了应用程序的质量及其留住用户的能力[21]。某天的几日留存率是指几日后的留存用户占当天新增用户的比率。

需求定义：计算每天的前7日留存率。

思路分析：

(1) 要统计某天如2022-04-04的前几日的留存率，应该按照该公式计算：2022-04-04的前3日即2022-04-01的3日留存率=2022-04-01新增且2022-04-04活跃的设备数/2022-04-01新增设备数。

(2) 计算留存率指标应该从DWT层查询。留存率以设备新增日期和留存天数作为标识，按设备新增日期分组聚合，计算每天的前7日留存率将会每天插入7条数据，每行代表一条过去的设备新增日期在该统计日期的留存率信息。

(3) 先过滤出统计日期的前7日新增设备，新增日期为设备首次活跃时间，然后按新增日期分组，每组统计出一条留存率数据，其中留存天数为统计日期与新增日期的差值，留存数为在统计日期活跃的总和，新增数就是分组聚合的总数，留存率为留存数与新增数的比例。

留存率表（ads_user_retention_day_rate）字段为统计日期、设备新增日期、截止当前日期留存天数、留存数、设备新增数、留存率，统计结果如图5-26所示。

图 5-26 留存率指标分析结果

5.4.7 指标分析-最近连续3周活跃设备数

需求定义：统计最近连续3周活跃设备数，即连续3周，每周至少启动过1次应用的设备数。

思路分析：指标以统计日期作为标识，从DWS层查询出本周、上周、上上周的活跃设备，对3周内的所有活跃设备按设备id进行分组，分组后设备个数为3的设备即为最近连续3周活跃设备。

最近连续3周活跃设备数表（ads_continuity_wk_count）字段为统计日期、持续时间标识、活跃设备数，统计结果如图5-27所示。

图 5-27 最近连续3周活跃设备数指标分析结果

5.4.8 指标分析-最近7天内连续3天活跃设备数

需求定义：统计最近7天内持续3天活跃设备数，即7天的其中3天连续活跃的设备数，但7天中有可能前3天和后3天都有活跃记录的情况，应该去重只记一个活跃设备。

思路分析：指标以统计日期作为标识，最近7天日期字段形式为“7天前日期_今天日期”，方便查看统计的时间段。要查询历史上某一时间段的数据，应该从DWS层导入。先对近7天的活跃设备按照设备id分组，按照活跃日期排序；取日期值和排序值的差值作为连续标志，值相同代表连续；按照设备id和连续标志再分组聚合，不小于3个即为近7天连续3天活跃的设备。

指　　标
疑似剽窃文字表述

1. 配置core-site.xml，如图5-1所示。
   图 5-1 core-site.xml配置
   配置hdfs-site.xml，如图5-2所示。
   图 5-2 hdfs-site.xml配置
   配置yarn-site.xml，如图5-3所示。
   图 5-3 yarn-site.xml配置
   配置mapred-site.xml，如图5-4所示。
   图
2. core-site.xml中添加如图5-8所示配置。
   图 5-8 core-site.xml配置
   5.
3. 配置zoo.cfg文件。重命名/opt/module/zookeeper-3.4.10/conf下的 zoo_sample.cfg为zoo.cfg，
4. MySQL的JDBC驱动拷贝到Hive的lib目录下。
   (3) 配置Metastore到MySQL配置Hive的conf目录
5. 目录下，重命名配置文件sqoop-env-template为sqoop-env.sh，并修改其中的配置，
6. 验证Sqoop是否安装成功。

(5) 测试Sqoop是否能够成功连接数据库。

7. 采集日志到Kafka层配置
采集日志Flume层的主要任务是将日志从落盘文件中采集出来

8. 针对本系统的需求，在编写Flume Agent配置文件之前，需要进行组件选型。
(1) Source选

9. 如图5-10所示。
图 5-10 采集层Flume配置
5.2.2 消费Kafka日志的Flume配置
将日志

10. 消费层Flume主要从Kafka读取消息，所以选用Kafka Source。
(2) Chan

11. (3) Sink选型
选用HDFS Sink将日志落盘到HDFS上。
(4) 详细配置
消费层Fl

12. 由于HDFS不支持对LZO压缩格式文件进行分片，所以还需要为LZO压缩文件创建索引。

13. 启动记录，需要包含公共信息、启动信息和启动时间。过滤出包含“start”字段的日志，然后用get_json_object函数解析每个字段。
(2) 页面日志解析：页面日志

14. 动作记录的日志。
(4) 曝光日志解析：曝光日志表每行对应用户在某个页面的一个曝光记录，需要包含公共信息、页面信息、曝光信息

15. 错误记录要包含与其对应的公共信息、页面信息、启动信息、动作信息、曝光信息、错误信息和时间。过滤出包含

16. ADS层具体实现
5.4.1 指标分析-活跃设备数
需求定义：
(1) 日活：当日活跃的设备数。
(2) 周活：当周活跃的设备数。
(3) 月活：当月活跃的设备数。
思路分析：指标以统计日期

17. 图 5-22 每日新增设备数指标分析结果
5.4.3 指标分析-沉默设备数
需求定义：

18. 设备数指标分析结果
5.4.4 指标分析-流失设备数
需求定义：最近7天未活跃的设备数。
思路分析：指标

19. 分析结果
5.4.5 指标分析-本周回流设备数
需求定义：上周未活跃，本周活跃，且不是本周新增的设备数。
思路分析：
(1) 指标

20. 图 5-27 最近连续3周活跃设备数指标分析结果
5.4.8 指标分析-最近7天内连续3天活跃

最近7天内连续3天活跃设备数表（ads_continuity_uv_count）字段为统计日期、最近7天日期标识、连续3天活跃设备数，统计结果如图5-28所示。

图 5-28 最近7天内连续3天活跃设备数指标分析结果

5.5 数据可视化模块实现

5.5.1 集群启动

在VMWare中启动虚拟机，如图5-29所示。

图 5-29 启动虚拟机

使用XShell连接虚拟机，如图5-30所示。

图 5-30 连接虚拟机

脚本介绍。

1. cluser.sh 整个集群启停脚本（start/stop）
2. jpsall 查看所有节点java进程
3. xcall 群执行脚本
4. xsync 同步分发脚本
5. lg.sh 模拟日志数据生产脚本
6. azkaban.sh azkaban启停脚本（start/stop）
7. zk.sh zookeeper框架启停脚本（start/status/stop）
8. hdp.sh hadoop集群hdfs/yarn启停脚本（start/stop）
9. kf.sh kafka启停脚本（start/stop）
10. f1.sh 第一层flume(采集)（start/stop）
11. f2.sh 第二层flume(消费)（start/stop）
12. superset.sh superset启停脚本（start/stop/status/restart）

一个脚本群启集群，如图5-31所示。

图 5-31 群启集群

用jpsall命令查看集群进程启动情况，如图5-32所示。

图 5-32 查看集群进程

其中NameNode、DataNode、SecondaryNameNode是HDFS服务进程，NodeManager和ResourceManager是Yarn服务进程，JobHistoryServer是历史服务器进程，QuorumPeerMain是Zookeeper服务进程，Application是Flume服务进程，AzkabanWebServer和AzkabanExecutorServer是Azkaban服务进程，RunJar是Hive服务进程。若有集群启动失败，要单独执行该集群的启停脚本，停止该集群服务后再次启动即可。

5.5.2 Azkaban全流程调度

为了有效组织系统工作流程，减少人力消耗、节省时间，可以将本数仓复杂的执行计划利用Azkaban工作流调度器来调度执行[26]。Azkaban通过将任务调度和任务间的依赖关系封装在后缀为.job的脚本中来调度任务，只要将.job文件压缩成.zip文件上传到Azkaban的Web客户端，然后就可以手动执行或者定时调度了。

要注意的是，执行过的任务不能再执行，若执行失败，要把执行成功的任务disable掉，然后从上次执行失败的任务开始再次执行。

在浏览器输入http://hadoop102:8081/，登录Azkaban，用户名和密码是在Azkaban的conf目录下的azkaban-users.xml文件中添加的用户配置，如图5-33所示。

图 5-33 Azkaban登录界面

打开本数仓流程项目，如图5-34所示。

图 5-34 Azkaban任务流项目界面

查看用户行为主题任务流调度，如图5-35所示。

图 5-35 Azkaban任务流调度界面

查看任务流详细信息，如图5-36所示。

图 5-36 Azkaban任务流预览界面

添加执行任务调度的时间参数，点击Execute执行，如图5-37所示。

图 5-37 Azkaban任务流参数界面

点击Continue查看执行结果，如图5-38所示。

图 5-38 Azkaban任务调度提交执行界面

执行成功，如图5-39所示。

图 5-39 Azkaban任务调度执行成功界面

点击菜单栏的Job List可以看到任务都是根据依赖关系，按时间先后顺序执行的，如图5-40所示。

图 5-40 Azkaban任务调度详细信息

5.5.3 SuperSet报表可视化

在浏览器输入http://hadoop102:8787/，登录SuperSet，如图5-41所示，用户名和密码为部署Superset时创建的管理员用户的用户名和密码。

图 5-41 Superset登录界面

实现用户行为数据各项指标分析结果的可视化展示，图标类型选择数字和趋势图，横着以天为单位，纵轴统计的是每天或一段时间内累计到当天的数据，做好相关设置后效果如图5-42和图5-43所示。

图 5-42 数据可视化1
图 5-43 数据可视化2
各指标说明如下：
(1) 日活跃设备趋势图展示的是每天的活跃设备数，可见本月的活跃设备增长较平缓，用户使用频率较稳定。当天有269个活跃设备。
(2) 周活跃设备趋势图展示本周累计到当天的活跃设备数，显然，每个峰顶的横坐标就是该周周末，纵坐标就是每周的最终结果。可见，第二周的活跃设备数量较上周有所上升。产品的用户体验良好。本周截止今天共有681个活跃设备。
(3) 月活跃设备趋势图展示的是每个月累计到当天的活跃设备数，可见本月截止今天累计活跃1.37k设备数。
(4) 日新增设备趋势图可以看出，应用一开始有很多用户注册，之后新增用户越来越少，今天的新增用户只有2个。
(5) 日留存率趋势图统计的是每天的前1日、前2日和前3日留存率。数据从月初开始统计，所以月初的前几日是没有数据的。可见每天的用户留存率波动较大。今天的前1日、前2日和前3日留存率分别为33.33%、0、20%。
(6) 日沉默用户趋势图展示距离当天7天前启动的设备数变化趋势，可见沉默用户越来越少，今天的沉默用户有48个。
(7) 本周回流用户趋势图展示本周累计到当天的回流用户，每个峰顶的横坐标是该周周末，纵坐标是每周的最终结果。可见，第二周的回流用户明显减少。本周截止今天共有142个回流用户。
(8) 日流失用户趋势图展示每天的流失用户变化趋势，可见本月的流失用户逐渐减少，今天的流失用户有236个。
(9) 最近连续三周活跃用户趋势图展示累计三周截止到当天的活跃用户数。由于前三周统计日期没有完整的连续三周的数据，所以直到三周后才有记录。累计三周截止到今天有239个活跃用户。
(10) 最近七天内连续三天活跃用户趋势图展示截止到当天的七天中，连续三天活跃的设备数。图中可以看出本月呈上升趋势，而今天的最近七天内连续三天活跃用户有63个。

8. 第六章总结与展望　　总字数：981

相似文献列表

去除本人文献复制比：5.8%(57)　　文字复制比：5.8%(57)　　疑似剽窃观点：(0)

1	基于稀疏表示的快速人脸识别算法研究与实现 王文涛 - 《大学生论文联合比对库》- 2016-06-05	5.8%（57） 是否引证：否

原文内容

第六章总结与展望
6.1 完成的工作
(1) 调研了课题相关的背景和国内外研究现状，形成了系统的分析报告，并对项目开发中需要用到的相关技术做了介绍。
(2) 对系统要实现的功能需求和非功能需求进行了详尽分析，提出了数据仓库系统总体架构，并对电子商务系统前端埋点用户行为数据的采集存储、计算和可视化进行了详细设计。
(3) 根据总体架构方案搭建起了整套集群环境，重点实现数据仓库对采集日志的数据分析，挖掘有效信息，为电商网站提供决策支撑。
(4) 对搭建好的数仓基于Azkaban实现任务自动调度，对系统实现结果进行展示，验证项目的可行性。
(5) 系统的实现将很有效克服传统数据分析的局限，能够保存海量的数据，增强搜索性能；也可以管理大量的数据，有效地挖掘出有价值的信息，提供数据支持帮助企业决策人员更好地了解电商网站的运营现状和用户行为轨迹，并及时制定相应的商务决策与营销策略。
6.2 待改进的地方
(1) 由于时间和难度原因，只选择了用户行为数据进行分析，还可以增加业务数据的分析。
(2) 本项目只实现了数据的离线分析，若是能实现实时分析。可以提高查询效率和报表精确度。
参考文献
[1] 第47次《中国互联网络发展状况统计报告》. http://cnnic.cn/hlwfzyj/hlwxzbg/hlwtjbg/202102/t20210203_71361.htm, 2021年02月03日.
[2] 许长福. 日志数据分析系统的设计与实现[D]. 北京:北京交通大学, 2017.
[3] 王彦明. 近年来Hadoop国内研究进展[J]. 现代情报, 2014, 34(8): 14-19.
[4] Michael Frampton, Big Data Made Easy, A Working Guide to the Complete Hadoop Toolset[M], Apress , December 23, 2014, 165.
[5] Rajiv Tiwari, Hadoop for Finance Essentials[M], Packt Publishing, April 30, 2015, 76.
[6] Apache Hadoop. [EB/OL]. http://hadoop.apache.org/.
[7] Apache Flume. [EB/OL]. http://flume.apache.org/.
[8] Apache Hive. [EB/OL]. http://hive.apache.org/.
[9] 张波, 移动互联网时代的商业革命[M], 机械工业出版社, 2013-2, 34.
[10] Dong Gaifang, Fu Xueliang, Li Honghui, Xie Pengfei. Cooperative ant colony-genetic algorithm based on spark[J]. Computers and Electrical Engineering, 2016.
[11] 李爽. 基于Spark的数据处理分析系统的设计与实现[D]. 北京: 北京交通大学, 2015.
[12] 刘鹏. 基于Spark的数据管理平台的设计与实现[D]. 杭州: 浙江大学, 2016.

[13] Foster Provost, Tom Fawcett, Data Science for Business, What you need to know about data mining and data-analytic thinking[M], O'Reilly Media,August 19, 2013, 310.
[14] Aravinth S S, Begam A H, Shanmugapriyaa S, et al. An efficient HADOOP frameworks SQOOP and ambari for big data processing[J]. International Journal for Innovative Research in Science and Technology, 2015, 1(10): 252-255.
[15] Levin M, Krause GM. Incident detection: A Bayesian approach[J]. 1978.
[16] 孙卫琴, JAVA面向对象编程[M], 电子工业出版社, 2006, 64.
[17] 王孝成. 数据仓库的特征、应用类型和实施方法[J]. 计算机与现代化, 2002(11): 11-13.
[18] 金雯婷, 张松. 互联网大数据采集与处理的关键技术研究[J]. 中国金融电脑, 2014(11): 70-73.
[19] Apache Superset. [EB/OL]. https://superset.apache.org/.
[20] 怀特(Tom White), 周傲英, Hadoop权威指南(第2版)(修订•升级版)[M], 清华大学出版社, 2011-7, 66.
[21] 孙明铎. 基于Hadoop的电商数据分析系统的设计与实现[D]. 桂林: 广西师范大学, 2017.
[22] 林昆山. 基于Hadoop的电商离线数据挖掘系统的设计与实现[D]. 长沙: 湖南大学, 2014.
[23] 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.
[24] 王建辉, 李涛. 基于Hive的支付SDK日志分析系统的设计研究[J]. 计算机应用与软件, 2017, 34(7): 51-54.
[25] 朱淑献. 分布式电商主题搜索引擎研究[D]. 广州: 华南理工大学, 2016.
[26] 杨苏雁. 基于矩阵分解及深度神经网络的推荐排序学习方法研究[D]. 北京: 北京交通大学, 2018.
[27] 彭稣宇. 基于大数据分析平台的网络数据处理研究[D]. 南京: 东南大学, 2017.
[28] 李彬, 刘莉莉. 基于MapReduce的Web日志挖掘[J]. 计算机工程与应用, 2012, 48(22): 95-98.
[29] 任其亮, 谢小淞. 基于遗传动态模糊聚类的道路交通状态判定方法[J]. 交通运输工程与信息学报, 2007, 5(3): 13-15, 25.
[30] 戢晓峰, 刘澜. 基于交通信息提取的区域交通状态判别方法[J]. 三峡大学学报(自然科学版), 2009, 31(1): 94-97.

致谢

在毕业设计阶段，我的导师孔祥杰老师以及王凌云学长的指导和建议，对我的项目进展起到了有效的引导作用，使我加深了对项目领域专业知识的理解，对自己的人生规划有了更明确的目标，在此谨表示衷心的感谢。

跳丸日月，毕业将近，落笔至此已是槐序。生活在这所美丽宽阔的大学里，不仅仅是每天在课堂上听老师们传道、受业、解惑，更多的是学习如何从懵懂的青少年成长为一名独立自强的大人。四年期间，我于此结识了许多如磁石引针、琥珀拾芥的朋友，他们在我的学业和人格建设方面提供了很多帮助，特表谢意。除此之外，我最应向我的家人表达谢忱，正是他们无条件的支持，使我拥有了直面失败和从头开始的勇气。

今后我在工作学习的道路上必定还会再遇到其他困难，而我在完成毕业设计的这段时间领悟到的科研精神，将会在那个时刻托举我、激励我继续前进。最后，冀望自己脚踏实地，靠一步一步的努力实现抱负，青衿之志，履践致远；凡事以渺小启程，靠一点一滴的积累成就自我，山不让尘，川不辞盈。

附录

附件1 毕业设计文献综述

附件2 毕业设计开题报告

附件3 毕业设计外文翻译（中文译文与外文原文）