

硕士学位论文

|  |
| --- |
| **基于Android的即时通讯和推送服务** |
| **研究与实现** |

|  |  |
| --- | --- |
| 作者姓名 | 贾建业 |
| 学科专业 | 软件工程 |
| 指导教师 | 陆璐 教授 |
| 所在学院 | 软件学院 |
| 论文提交日期 | 2017年04月 |

**Research and Implementation of Instant Messaging and Push Service Based on Android**

A Dissertation Submitted for the Degree of Master

**Candidate：Jia Jianye**

**Supervisor：Prof. Lu Lu**

South China University of Technology

Guangzhou，China

**分类号： TP302.7 学校代号：10561**

**学 号：201520134361**

华南理工大学硕士学位论文

**基于Android的即时通讯和推送服务**

**研究与实现**

作者姓名：贾建业 指导教师姓名、职称：陆璐　教授

申请学位级别：工学硕士 学科专业名称：软件工程

研究方向：计算机应用技术

论文提交日期： 论文答辩日期：

学位授予单位：华南理工大学 学位授予日期：

答辩委员会成员：

主席：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

委员：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**华南理工大学**

**学位论文原创性声明**

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写的成果作品。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律后果由本人承担。

作者签名： 日期： 年 月 日

**学位论文版权使用授权书**

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属华南理工大学。学校有权保存并向国家有关部门或机构送交论文的复印件和电子版，允许学位论文被查阅（除在保密期内的保密论文外）；学校可以公布学位论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存、汇编学位论文。本人电子文档的内容和纸质论文的内容相一致。

本学位论文属于：

□保密，在\_\_\_\_\_年解密后适用本授权书。

□不保密，同意在校园网上发布，供校内师生和与学校有共享协议的单位浏览；同意将本人学位论文提交中国学术期刊(光盘版)电子杂志社全文出版和编入CNKI《中国知识资源总库》，传播学位论文的全部或部分内容。

(请在以上相应方框内打“√”)

作者签名： 日期：

指导教师签名： 日期：

作者联系电话： 电子邮箱：

联系地址(含邮编)：

# 摘　要

随着移动互联网的迅猛发展，移动应用在人们的日常生活中起着越来越重要的作用。其中即时通讯和推送模块，几乎成了大部分移动应用不可或缺的部分。与此同时，移动应用的市场逐渐进入了相对成熟的阶段，应用开发商一方面越来越注重缩短应用的开发周期，使应用能够尽快进入并抢占市场，而另一方面，更注意收集用户在使用过程中所产生的数据，并对这些数据进行分析，从而可以根据这些数据为应用下一步发展提供合理的建议。

本文首先对国内外对即时通讯和推送技术的现状进行了研究，设计并实现了基于Android的即时通讯和推送服务，由以下三个部分构成：（1）即时通讯客户端SDK，其他移动应用可以使用该SDK快速集成即时通讯的功能，SDK提供了两种集成方式，一种是直接使用SDK的相关页面并进行定制化；另一种是使用自己的页面，只调用SDK的核心接口。开发者可以根据自己的需求选择集成方式。另外即时通讯SDK还会收集并上报用户的相关数据至数据分析中心。（2）推送客户端SDK，其他应用可以使用该SDK快速集成推送的功能，同时提供了一个推送端Web页面，推送者可以在推送端上设置推送信息的相关内容，将推送消息推送到移动设备上，同样，SDK提供了数据收集和上报的功能，包括用户设备数据，推送信息相关数据等。（3）数据分析中心，数据分析中心将即时通讯和推送SDK上传的数据进行接收，存储，计算，分析和图表的展现，开发者和运营商可以通过图表直观的了解应用目前的运行情况和用户的使用习惯等，为应用的下一步发展提供意见；同时数据分析中心还针对推送系统构建了对推送内容进行自动评分的模型，并使用基于线性拟合的协同过滤算法，构建了个性化的推送系统。最后，本文使用一个新闻类应用进行案例分析，验证本文的工作，该应用快速集成了即时通讯和推送的相关功能，同时数据分析中心给出了相关的图表，为应用的进一步优化提供了建议。

关键词：即时通讯；推送；移动应用；数据分析

# Abstract

With the rapid development of the mobile internet, mobile applications are playing more and more important role in the daily life. For the most applications, instant messaging and message push have become integral modules. In the meanwhile, the mobile application market has gradually entered a relatively mature stage. On the one hand, application developers pay more and more attention to shortening the developing time, so that apps can enter the market and seize the market share as soon as possible. On the other hand, they attach more importance to collecting and analyzing the data generated by users, through which they can make suggestions on the apps’ next step development.

This paper first makes a research on instant messaging and message push at home and abroad, moreover designs and implements an instant messaging service and a message push service based on Android. The platform consists of the following three parts: (1) Instant messaging SDK module, the other mobile apps can use it to quickly integrate related functions. The SDK provides two integration methods, one is to use relevant customized pages including activity and fragment in SDK directly; the other is to call the SDK’s core interface without using the pages in it. In addition, the SDK also collects and reports user’s relevant data to the data analysis center, by which developers and operators can have a deeper understanding of the app. (2) Message push SDK module, the other mobile apps can use it to quickly integrate relevant functions. It also provides a push-end web page. The pusher can set related properties on the page and push the message to the mobile devices.At the same time, the SDK provides collecting and reporting data functions which can collect user device data, push message data, and so on. (3) Data analysis center, it receives the data collected and uploaded by the tow SDKs. In the meantime it will store, compute, analysis and finally display the graphics.The developers and operators can have a deeper understanding of the apps’ running situation and the users’ using habits so that can make suggestions on the apps’ next step development. Simultaneously, aimed at the message push system, the Data analysis center also builds a model by which can score the push message automatically and uses the Collaborative Filtering algorithm based on linear regression to build a recommending system. Finally, the paper uses a news application as a case to verify the work the paper has done. The app integrates the corresponding functions quickly by using the SDKs. And the data analysis center provides related charts, making suggestions on further application optimization.

**Keywords:** Instant Messaging; Push; Service; Mobile Application; Data Analysis

# 目　录

[摘　要 I](#_Toc7665)

[Abstract II](#_Toc3823)

[目　录 III](#_Toc6889)

[第一章　绪论 1](#_Toc30859)

[1.1　课题研究背景 1](#_Toc16289)

[1.2　研究目标 2](#_Toc23644)

[1.2.1　即时通讯和推送服务概述 2](#_Toc14948)

[1.2.2　研究意义 3](#_Toc9487)

[1.3　国内外研究现状 3](#_Toc12917)

[1.3.1　国外研究现状 4](#_Toc20224)

[1.3.2　国内研究现状 4](#_Toc25691)

[1.4　本文主要工作 5](#_Toc30666)

[1.5　论文组织结构 6](#_Toc17833)

[1.6　本章小结 6](#_Toc31604)

[第二章　关键技术及整体架构设计 7](#_Toc5132)

[2.1　关键协议介绍 7](#_Toc5071)

[2.1.1 XMPP协议介绍 7](#_Toc4500)

[2.1.2 MQTT协议 7](#_Toc21509)

[2.2　Android客户端关键技术 8](#_Toc162)

[2.2.1　MVP工程架构 8](#_Toc26989)

[2.2.2　Android JNI开发 9](#_Toc17370)

[2.2.3　Android进程间通信技术 10](#_Toc5141)

[2.3　WebRTC视频通话技术 11](#_Toc27401)

[2.4　协同过滤推荐算法 11](#_Toc7345)

[2.5　整体架构设计 12](#_Toc31715)

[2.6　本章小结 13](#_Toc31423)

[第三章　即时通讯服务的设计和实现 14](#_Toc32257)

[3.1　即时通讯SDK实现 14](#_Toc13611)

[3.1.1　实现背景 14](#_Toc19340)

[3.1.2　即时通讯SDK整体架构设计 14](#_Toc6330)

[3.1.3　即时通讯SDK重要模块实现 16](#_Toc29809)

[3.1.3.1　用户管理模块 16](#_Toc31477)

[3.1.3.2　联系人管理模块 16](#_Toc12316)

[3.1.3.3　发送和接收消息模块 17](#_Toc13696)

[3.1.3.4　音视频通话模块 19](#_Toc31230)

[3.1.4　功能接口化实现 20](#_Toc9458)

[3.2　相关数据收集和上报 21](#_Toc6009)

[3.2.1　实现背景 21](#_Toc4015)

[3.2.2　用户统计和分析 21](#_Toc31197)

[3.2.3　消息统计和分析 23](#_Toc15444)

[3.3　本章小节 23](#_Toc17820)

[第四章　推送服务的设计和实现 24](#_Toc31919)

[4.1　推送SDK实现 24](#_Toc20453)

[4.1.1　实现背景 24](#_Toc22881)

[4.1.2　客户端推送SDK基本功能实现 24](#_Toc11842)

[4.1.2.1　推送消息的接收 24](#_Toc11157)

[4.1.2.2　推送消息的展示 26](#_Toc4073)

[4.1.2.3　离线推送消息的接收 28](#_Toc7542)

[4.1.2.4　推送消息的后续动作 28](#_Toc5860)

[4.2推送端的实现 29](#_Toc3911)

[4.2.1　实现背景 29](#_Toc22340)

[4.2.2　性能数据监测实现 29](#_Toc3660)

[4.2.3　后台推送功能实现 30](#_Toc5354)

[4.3　相关数据收集和上报 30](#_Toc13281)

[4.3.1　实现背景 30](#_Toc23733)

[4.3.2　推送消息数据采集 31](#_Toc13946)

[4.3.3　用户设备数据采集 32](#_Toc15388)

[4.4　本章小节 34](#_Toc856)

[第五章　数据分析中心的设计和实现 35](#_Toc12910)

[5.1 数据分析中心的整体结构 35](#_Toc12091)

[5.2 数据接收和对外提供模块 36](#_Toc4441)

[5.3 数据存储模块 36](#_Toc15710)

[5.4 数据分析模块 37](#_Toc28263)

[5.5　基于协同过滤的推荐系统模块 38](#_Toc30578)

[5.5.1　推送内容评分的构建 38](#_Toc5515)

[5.5.2　算法设计 39](#_Toc9861)

[5.6　结果展示模块 41](#_Toc11512)

[5.7　本章小结 41](#_Toc15339)

[第六章　案例分析 42](#_Toc31962)

[6.1　测试案例准备 42](#_Toc12966)

[6.1.1　应用介绍 42](#_Toc8615)

[6.1.2　服务端部署环境介绍 43](#_Toc29924)

[6.1.3　应用构建流程 43](#_Toc26144)

[6.2　即时通讯SDK的集成 44](#_Toc11908)

[6.3　推送SDK的集成 46](#_Toc10815)

[6.4　数据中心统计分析 48](#_Toc31948)

[6.4.1　即时通讯数据分析 48](#_Toc9885)

[6.4.2　推送数据分析 51](#_Toc11437)

[6.4.3　协同过滤推荐模型 56](#_Toc6178)

[总结与展望 59](#_Toc1098)

[总结 59](#_Toc32237)

[展望 59](#_Toc14753)

[参考文献 61](#_Toc9183)

[攻读硕士学位期间取得的研究成果 64](#_Toc1921)

[致　谢 65](#_Toc17974)

# 第一章　绪论

## 1.1　课题研究背景

近年来，国内的移动互联网始终保持着非常快速地发展。各类移动应用层出不穷，并且结合了物联网，共享经济，人工智能等技术，给我们的日常生活带来了极大地便利，起着越来越重要的作用。

互联网络信息中心发布了[最新的《中国互联网络发展状况统计报告》](http://n2.sinaimg.cn/tech/68acb9a7/20180131/CNNIC41.pdf" \t "http://tech.sina.com.cn/i/2018-01-31/_blank)[1]，报告对我国2017年度的移动互联网发展现状做了较为全面的调查和分析。报告中指出，我国网民中使用手机上网人群的占比由2016年的95.1%提升至97.5%，网民手机上网比例不断提高。而相反地，使用PC上网的比例分别为53.0%、35.8%，较2016年底均有所下降，其中使用台式电脑的比例变化尤为明显，下降7.1个百分点，可见移动互联网的发展依然如火如荼。

而在各种类型的移动应用中，即时通讯的用户量最大，达到了七亿两千万人，网民使用率高达百分之九十三。并且在历年的报告中，即时通讯的使用率几乎都是最高的。可以看出，即时通讯对于大部分移动网络用户来说都是最为重要的，使用频率最高的功能。而对于其他类型的移动应用，应用内部的即时通讯模块也越来越重要，如果应用内部没有即时通讯模块，或者即时通讯模块不够成熟，用户之间沟通则很可能会借助其他第三方即时通讯应用，不仅给用户带来很大的不便，还可能造成应用的使用率和黏着度的大幅度降低。

另一方面，随着今日头条等资讯类应用的崛起和火爆，推送技术也越来越成为一个非常受人瞩目的领域，尤其对于新闻、购物等类型的应用，推送系统甚至是整个移动应用非常重要的模块之一。另外，目前许多推送系统，只负责将内容推送给用户，但并没有对用户后续行为进行进一步的收集。收集用户收到推送后的行为，可以进一步分析用户的使用习惯和规律，改善整个推送系统，并且可以进一步结合推荐算法，构建一个个性化的推荐模型[2]。

随着过去几年移动互联网和移动应用市场的大规模扩张，目前移动互联网的发展已经进入了相对比较成熟的阶段，开发者和应用厂商一方面希望缩短应用的开发周期，降低开发成本，使应用可以尽快进入市场，提前抢夺用户；另一方面更加重视用户体验[3]，收集用户在使用过程中产生的各类数据，并对数据进行分析，对于针对性的提高用户体验是非常有帮助的。

最后，考虑到移动平台的多样性，而其中的Android操作系统具有开源，免费，可移植，市场占有率高[4]等特点，所以本文主要针对Android 平台研究即时通讯和推送服务以及相关用户行为的数据采集分析。

## 1.2　研究目标

### 1.2.1　即时通讯和推送服务概述

在诸如社交、购物、新闻、游戏等类型的应用中，应用内部的即时通讯模块在业务需求上可能不是最为核心的模块，但是在实际应用场景中却往往是非常关键、不可缺少的一个模块。而开发一个成熟，稳定的即时通讯系统却费时又费力，会极大程度的拉长项目进度，增加额外的开发成本，导致应用迟迟不能开发完成上线，甚至使整个项目胎死腹中。

另外，对于一个拥有大量用户的应用，推送功能同样十分重要，尤其是对购物，视频，新闻等类型的应用，推送功能甚至是最为重要的功能之一，是其用户的重要来源。而一个成熟推送系统，并不只是能够保证快速，高效地向客户端推送消息。推送者一方面想了解用户收到推送消息后的行为数据，根据用户的行为，对推送系统和推送内容进行针对性的改进。另一方希望能够对用户进行个性化的推送，对不同喜好的用户推送不同的内容，不仅可以使整个推送系统更加的准确和高效，还可以增加用户对于应用的好感度和黏着度。

无论对于即时通讯模块还是推送模块，仅仅关注于功能的实现是不够的，在同类软件竞争越来越激烈的今天，开发者和厂商开始更加注重用户的实际体验。所以针对用户的数据的采集是十分重要的，这些数据不仅为应用工程上的优化和版本迭代提供意见，更重要的是为应用的下一步的运营和发展方向提供数据上的支撑。

本文的即时通讯和推送服务具有以下特点：

1. 简单易用，本文设计的客户端SDK包括即时通讯SDK和推送SDK，因为其最终需要用户将本平台提供的独立的SDK集成到自己的工程中使用，所以必须要让开发者使用时感到非常得简单和方便，开发者只需要在应用启动时插入一行相应代码，即可完成基本功能的初始化，使用绝大部分的功能，真正做到即插即用。
2. 耦合度极低，本文设计的SDK与其宿主工程几乎没有耦合，SDK以aar包的形式提供，或者通过Gradle进行依赖，所以开发者无需担心SDK污染宿主工程。
3. 统计功能，只做到实现基本的功能是不够的，这也是目前其他类似的平台所缺少的。本文设计的SDK不仅有基本的即时通讯和推送功能，还具有较为丰富的统计功能，SDK会自动收集用户的各种数据进行上报，其中包括基础的用户数据，设备数据等，也包括针对推送和即时通讯功能而收集相关重要数据，并通过数据分析中心给开发者和运营者提供多种多样的数据分析报表。
4. 平台性，本文设计的即时通讯和推送服务旨在构建一个众多开发者可以一起使用的第三方工具。本文实现了客户端SDK的开发，服务端搭建，数据库搭建，数据分析中心的搭建，以及相应算法模型的构建等，提供了从前台到后台到算法等一整套完整的解决方案。如果平台的用户达到一定的数量时，甚至可以提供整个行业的数据进行统计分析，给行业的进一步发展提供大数据的指导意见。

### 1.2.2　研究意义

即时通讯和推送都可以归为广义的即时通讯，在应用内部，这些模块与应用其他模块往往是非常低耦合的。因此本文分别针对即时通讯和推送场景，设计并实现了即插即用的SDK，应用开发者只要进行简单地配置，即可在任何一款Android应用中集成一个成熟可靠的即时通讯系统或者推送系统，并且能够收集用户的相关行为数据，将数据上报至数据分析中心。数据分析中心一方面对收集的数据进行存储、分析、计算和展示，另一方面针对推送系统，根据收集的用户行为数据，提供基于机器学习方法的个性化推荐服务。从而使开发者无需花过多时间在这两个较为独立的领域，即可使自己的应用具有完整的即时通讯和推送功能，并进行相关数据的收集和统计，开发者可以将大部分精力集中于应用其他模块，这对于应用开发者来说无疑可以大大节省时间成本和经济成本，使应用能够尽快的投入并抢占市场，在移动应用市场竞争十分激烈的今天，是十分有意义的。

## 1.3　国内外研究现状

随着移动互联网的快速发展，国内外有许多对于即时通讯和推送的研究，但目前大部分都是根据某个具体的需求，开发具体的功能，最终形成一套独立的系统，不能针对不同的应用提供快速的集成，节约成本。另一方面，其中大部分的即时通讯和推送系统都没有进一步收集用户的相关数据进行统计分析，也没有使用算法针对推送系统做个性化的推送服务。

本文不仅开发了一套功能齐全的即时通讯和推送系统，并且将其SDK化，成为一个所有应用都可以即插即用的插件。同时，SDK中收集了用户的各种行为数据，通过数据分析中心统计、分析、计算，最终展现给开发者。

### 1.3.1　国外研究现状

[Yulianto, B](http://apps.webofknowledge.com/OneClickSearch.do?product=UA&search_mode=OneClickSearch&SID=7DAcUVgSuZcYOG1c5Ts&field=AU&value=Yulianto,%20B&ut=3811872&pos=1&excludeEventConfig=ExcludeIfFromFullRecPage" \o "查找此作者的更多记录) [5]等在其论文中实现了一个基于Android平台的针对教育机构的即时通讯系统，该系统使用了XMPP协议，实现了单聊，群聊，主要目的是促进教育机构中学生和教师之间的交流。但该系统并没有针对教育领域，收集用户的相关行为数据。

[Lübke, Robert](https://www.engineeringvillage.com/search/submit.url?CID=quickSearchCitationFormat&implicit=true&usageOrigin=recordpage&category=authorsearch&searchtype=Quick&searchWord1=%7bL%26%23252%3Bbke%2C+Robert%7d&section1=AU&database=1&yearselect=yearrange&sort=yr" \o "Search Author) 等[6]在其论文了实现了一个即时通讯的模块，它提供了即时通讯的常用的特性，可以很容易地集成到第三方的移动应用程序中，加快了第三方移动应用的开发速度。但是其并没有提供用户行为数据采集和分析的功能。

[Gudla, Suresh Kumar](https://www.engineeringvillage.com/search/submit.url?CID=quickSearchCitationFormat&implicit=true&usageOrigin=recordpage&category=authorsearch&searchtype=Quick&searchWord1=%7bGudla,+Suresh+Kumar%7d&section1=AU&database=1&yearselect=yearrange&sort=yr" \o "Search Author) 等[7]在其论文中实现了一个统一的推送服务系统，可以服务于多种移动平台，快速构建推送功能，并对推送架构的性能做出了优化。但是其没有针对推送服务收集用户的相关行为数据，并进行分析。

[Brüstel, Jonas](https://www.engineeringvillage.com/search/submit.url?CID=quickSearchCitationFormat&implicit=true&usageOrigin=recordpage&category=authorsearch&searchtype=Quick&searchWord1=%7bBr&%23252;stel,+Jonas%7d&section1=AU&database=1&yearselect=yearrange&sort=yr" \o "Search Author)[8]等在其论文中对推送技术进行了分析，并最终提出了一种统一的推送接口，可以适用于各种移动设备。但是同样也没有对用户数据进行采集和分析。

在国外也出现了一批即时通讯和推送相关的产品。

IMSDK是一个的即时通讯框架，支持了Andoird，iOS，.NET，JS等多种平台，为平台下的APP提供功能齐全的即时通讯功能，同时也集成了推送功能，但是平台缺乏对用户使用过程中产生的数据进行统计和分析，尤其是针对推送的统计信息非常少。

Urban Airship是一个已经被许多国外大型应用所认可和使用的推送框架，为这些应用提供了稳定的推送服务，并且收集了推送消息的相关数据，最终提供一个分析报告，但并没有实现个性化的推送系统。

Kumulos是一个功能丰富的推送框架，除了实现了基本的通知推送功能，还具有崩溃统计等功能，并且提供了较为丰富的针对推送消息的统计分析和图表展示，但是并没有提供个性化推送的功能。

### 1.3.2　国内研究现状

李鲲鹏[9]在其论文中实现了一个基于Android平台的即时通讯平台，使用XMPP协议和Openfire服务器进行消息通讯，并使用Tomcat服务器，实现了客户端之间的图片、文档等的分享。但本文并没有将其系统制作成一个插件式的SDK，同时也没有对用户的各种行为数据进行收集和分析。

徐鹏伟[10]在其论文中实现了一个基于XMPP协议的iOS即时通讯应用，提供了较为丰富的功能，并对应用进行了功能和性能方面的测试。但同样没有关注用户行为的数据采集和分析。

周艺[11]在其论文中实现了基于XMPP协议的Android终端消息推送平台，并使用动态连接算法，降低了移动终端的耗电量和流量，并对外提供消息推送功能。但其没有针对推送行为进一步收集分析用户的行为数据。

黄明恩[12]在其论文中实现了一个基于Android平台的云推送服务平台，使用了MQTT协议和ActiveMQ服务器提供推送功能，开发了一个客户端SDK，并进行了简单的统计，包括信息推送统计和终端信息统计。但其统计的数据较少，很不全面，也没有使用推荐算法，针对推送系统构建一个推荐模型。

郑峰[13]在其论文中实现了一个基于MQTT的云推送平台，较为深入的研究了MQTT协议，并且最终提供了推送客户端SDK和推送服务端SDK，但是其平台并没有对推送数据进行收集和上报和统计。

以上研究或者至关注于单个应用上，不能将即时通讯和推送的功能快速集成到第三方应用，或者没有对用户的行为数据进行采集和分析。而在国内的企业界，由于有较为广阔的市场，出现了一系列的产品[14]。

网易云信是网易开发的一套云服务产品，其中的即时通讯产品平台，为开发着提供了即插即用的即时通讯SDK，用户接入SDK后即使应用具有私聊，群聊等通讯能力，同时也提供了简单的用户统计功能。但是其并没有对用户行为数据进行收集、分析和统计。

友盟推送，极光推送等都为开发者提供了可以快速集成到开发者应用的推送SDK，并实现了较为丰富的数据统计分析功能。但是并有没有对针对推送实现一个基于机器学习的推荐系统。

另一方面，国内也涌现出了许多开源的即时通讯和推送的SDK。

MobileIMSDK是一个专为移动端开发的开源即时通讯框架，超轻量级、高度提炼，完全基于UDP协议，支持iOS、Android、Java平台，服务端基于Mina和Netty编写，但是其只实现了客户端的即时通讯功能，并没有收集相关的数据进行分析和展示。

MPush是一个开源的，支持多种客户端的实时消息推送框架，采用Java语言开发，具有协议简介，高度可配置，扩展性强等优点，但是MPush并没有提供相关推送数据的收集统计功能。

## 1.4　本文主要工作

根据上述分析，现有的即时通讯和推送服务还都不够完整和深入，它们或者没有提供一个可以供开发者应用快速集成的SDK，或者没有提供较为完整的用户行为数据的统计分析，和算法分析。而在开发者对快速开发和用户数据都越来越重视的今天，对该领域进行更加深入的研究是十分有意义的。

本文的主要工作是实现一个即时通讯和推送服务，首先提供了可以供第三方Android应用快速集成相关功能的SDK，该SDK一方面使应用具有基本的即时通讯或者推送功能，另一方面负责收集用户的各种行为数据，并发送至数据分析中心。数据分析中心对收集到的用户行为数据进行进一步的统计、计算、分析，将统计结果展示给用户，同时针对推送系统，使用了协同过滤算法构建了一个个性化推荐系统。通过这个平台，开发者可以迅速使自己的的应用具有即时通讯或者推送功能，大大降低了开发成本。同时可以根据用户的行为数据，更加清楚的了解用户的使用习惯，从而可以行针对性的对应用进行进一步的改进和完善。

在本文的余下章节中，会分别介绍关键技术，系统整体架构以及即时通讯服务模块，推送服务模块和后台数据分心中心模块的设计和实现，并最终给出一个案例，展示整个系统的使用流程。

## 1.5　论文组织结构

本文总共分为了七个章节：

1. ，绪论，本章阐述了研究背景和意义，对国内外即时通讯和推送服务的研究进展和现状进行了介绍，并阐述了本文的主要工作，最后一节对本文的组织结构进行了叙述。
2. ，关键技术和系统总体架构，本章分别介绍了XMPP和MQTT协议，Android客户端的相关技术，WebRTC视频通话协议和协同过滤算法，最终介绍了系统的总体架构。
3. ，即时通讯服务的设计和实现，本章介绍了客户端即时通讯SDK主要基本功能的实现方法和原理，以及SDK针对即时通讯收集上报的用户行为数据的数据类型和收集方法。
4. ，推送服务的设计和实现，本章首先介绍了客户端推送SDK主要基本功能的实现原理。然后介绍了SDK针对推送行为收集上报的用户行为数据的数据类型和收集方法。最终实现了Web前台推送页面。
5. ，数据分析中心的设计和实现，本章首先介绍了数据分心中心的整体架构，然后分别介绍了存储模块，分析模块，推荐算法模块和展示模块等的设计思路和实现方案。
6. ，案例分析，本章选取了一个有代表性的新闻类Android应用，快速集成了即时通讯和推送的SDK，并收集了相关的用户行为数据，并展示了最终的用户行为数据的统计和分析的结果。

总结与展望，本章总结了本文的整体工作，并针对本平台未来进一步的研究工作进行了展望。

## 1.6　本章小结

本章一开始论述了相关的研究背景和研究意义，说明了研究该课题的重要性。然后对目前国内外在该方面的技术研究的现状进行了讨论，并指出了其中的不足。最后介绍了本文的主要工作和本文的组织结构。

# 第二章　关键技术及整体架构设计

## 2.1　关键协议介绍

### 2.1.1 XMPP协议介绍

XMPP协议是一种基于XML的专门为即时通讯而设计的开源协议[15]。XMPP底层协议为TCP/IP协议，通过在服务器和客户端之间交换XML流[16]进行即时通讯。XMPP协议具有以下优点：

1. 开源，XMPP协议完全开源和开放，并且简单易懂，不仅如此，包括绝大多数的基于XMPP协议的客户端、客户端库，以及各类服务器也都是开源的，这也使得任何开发者都可以非常快速的入门并使用。XMPP的开源精神是它能够一直保持活力的重要原因之一。
2. 标准化，XMPP协议的标准由XMPP标准基金会制定并维护，所有其制定的标准都会公布在其官方网站上，这使的XMPP的核心协议具有高度标准化，主流的基于XMPP协议的客户端或者服务端都基于这个标准进行开发，从而能保证基于XMPP协议的不同客户端的主要功能都可以互联互通。
3. 可扩展[17]，尽管XMPP协议由一个专门的组织进行维护，但其设计又考虑到了不同应用之间的差异。开发者可以根据自己的需求，在XMPP协议的基础上轻松扩展自己的功能，快速完成软件的开发，当然这样做可能会导致这些功能无法和其他使用了XMPP协议对的软件进行互通。

XMPP协议从1999年发布至今，依然保持着其强大的活力，各种基于XMPP的开源、商业软件依然层出不穷。不仅如此，XMPP也被越来越多的应用到了物联网[18]和视频通话等其他领域。

Openfire服务器是一个使用Java作为开发语言的，开源的XMPP协议服务器，为即时通讯的提供了最为基本的通讯功能[19]。Openfire有丰富的插件系统，可以通过下载安装插件实现更为丰富的功能。同时Openfire支持分布式和负载均衡，可以应付较大用户量的场景。

### 2.1.2 MQTT协议

MQTT协议是IBM开发的一种基于发布/订阅模型的通讯协议[20]，移动端设备订阅相应话题，推送服务端只需向这些话题发送推送信息，移动端设备便可以接收到相应的推送消息。实现推送功能有很多不同的模型和方法[21]，而MQTT协议采用的发布/订阅模型不需要客户端轮询服务器端是否有推送消息[22]，从而大大减少客户端流量的消耗。MQTT协议适用于设备数量多，设备计算能力低，网络环境不稳定的环境，具有稳定，可靠，省点，省流量等优点，用于向移动设备推送消息的场景是十分理想的。

MQTT协议中有三个不同的身份，分别为消息发布者，消息代理者与消息订阅者，其中消息代理者为服务器端，消息推送者和消息订阅者都为客户端，并且同一个客户端可以同时即是发布者又是订阅者。发送消息的基本流程为：订阅者向代理者订阅某话题，订阅完毕后，发布者向代理者所持有的该话题发送消息，代理者则会向订阅了该话题的订阅者发送消息，从而完成一次消息的发布。

MQTT协议的报文[23]设计十分巧妙，既保证了报文足够承载协足够的信息，又使得报文十分简短和紧凑。MQTT协议的报文分为三部分，分别为固定头部部分，可变头部部分和负载部分。

固定头部部分是所有类型的MQTT报文必须具有的，其格式如图2-1所示。

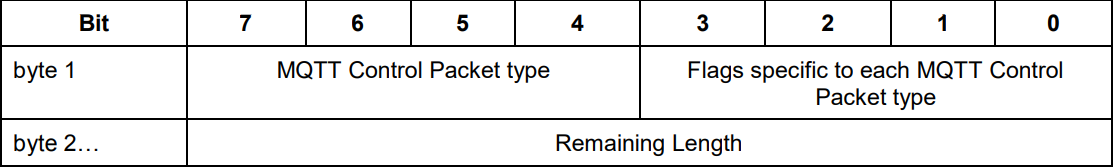


图2-1　MQTT协议报文固定头部头结构图

其中，第一个字节的7-4位表示MQTT控制报文的类型。第一个字节的3-0位作为标识位，只有特定类型的报文才有标识位，而在不使用标识位的报文中，这三位作为保留位。第二个字节用来保存变长头部和负载的总大小，而当总大小大于一个字节可以表示的长度时，则使前七位表示总大小，最后一位表示是否还有第二个字节，以此类推，最多支持四个字节，所以MQTT协议最大可以支持256M大小的消息。

可变头部位于固定头部和负载之间，不同类型报文的可变头部差异较大，以下以连接报文为例，其可变头部包括四部分，分别为协议名，协议级别，连接标识，以及响应时间。

Apache Apollo服务器是Apache旗下的开源服务器，其原型为大名鼎鼎的ActiveMQ，并在其基础上进行了多方面的优化，是新一代的的消息代理服务器，相比ActiveMQ，它更加可靠，速度更快，并同时支持STOMP、AMQP、MQTT等多种消息传输协议。

## 2.2　Android客户端关键技术

### 2.2.1　MVP工程架构

即时通讯SDK的工程代码量较大，模块较多，并且模块间的耦合较为严重，在这种情况下使用传统的工程架构会导致工程的开发和维护越来越困难，使用一个合适的能够解耦的工程架构对这类复杂工程来说十分重要。

MVP工程架构[24][25]，是一种能够对复杂工程有效解耦的架构模式，是对MVC模式的改进和发展，MVP三层分别表示Model层，View层和Presener层。其中，Model层包含业务逻辑和业务数据。View层负责显示界面，与用户交互，在Android系统中主要指Activity和Fragment。Presenter层是Model和View的中间层，一方面负责将Model层的数据和业务逻辑传递给View层，一方面响应用户在View层的动作，对Model层进行操作。

MVP架构与传统的MVC架构最大的区别和改进是，View和Model完全的解耦，View和Mode不允许直接交互，两者之间的交互必须通过Presenter进行中转和分发。MVP架构的结构图如图2-2所示。

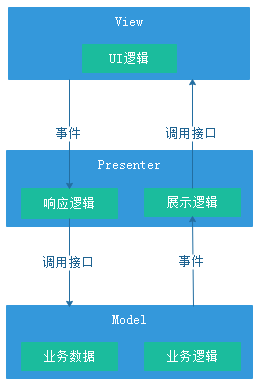


图2-2　MVP结构图

使用MVP工程架构模式具有以下优点：

（1）分离了业务逻辑和UI逻辑，极大程度上降低了工程模块间的耦合度。尤其是在UI逻辑中不需要掺杂任何业务逻辑代码，使Activity或者Fragment界面上的代码变得非常简洁，极大程度的提高了代码的维护成本和可阅读性。

（2）Presenter一般被抽象成接口，对于一些功能类似的界面可以复用Presenter接口，进一步简化了代码，增强了工程的逻辑性。

### 2.2.2　Android JNI开发

JNI技术允许 Java 和本地代码之间互相调用，在 Android 平台，此处的本地代码是指使用 C/C++ 或汇编语言编写的代码，编译后将以动态链接库so文件的形式供 Java 虚拟机加载。

Android工程一般使用Java开发，这对于大部分的场景都是足够的，但是Java层本身有一些固有的缺点，例如：性能低、容易被反编译、移植性差、难以和虚拟机交互等。所以对于一些特殊的场景，必须使用C/C++在Native层进行开发。Android中所谓的Native层是指Android架构体系中Library和Android Runtime层，主要包括Android系统中众多的底层库和虚拟机，Native层基本上都使用C/C++进行开发。

对于本设计中的即时通讯和推送的客户端SDK，都需要大量的网络操作，一方面这些网络操作的代码中包含了账号、密码、IP地址、端口号等敏感信息，另一方面频繁的网络通讯对性能的要求是比较高的，而网络性能对这类应用又十分重要。所以本设计的网络操作部分使用C/C++在Native层完成开发。既可以保证客户端不被轻易的反编译，从而暴露服务器的敏感信息，又能够提高网络通信的性能和效率，给用户带来更好的体验。

### 2.2.3　Android进程间通信技术

进程间通信是指两个独立的进程之间进行数据交换和通信的行为。Android系统中，每一个进程都被分配了一个独立虚拟机，每一个虚拟机在内存中分配了独立的地址空间，所以进程间通信往往不是一件比较困难的事情。Android系统基于Linux，Linux系统中进程间通讯的方法例如管道，Socket等理论上在Android系统上依然是可行的。但是Android系统为了效率和安全性，设计了Binder[26]通讯模型作为其进程间通讯的主要方式，Binder通讯模型的架构如图2-3所示。

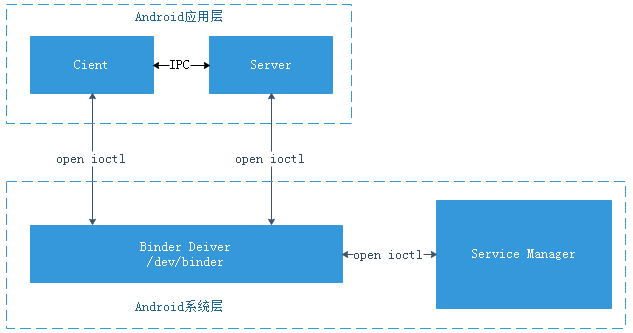


图2-3　Binder架构图

而由于Binder机制非常复杂，程序员实际上使用非常困难，Android系统又在Binder之上设计了AIDL[27]供开发者使用。AIDL是一门专门为Android应用进程间通讯而设计的语言，ADIL大大降低了开发人员进行跨进程通讯的开发难度。

## 2.3　WebRTC视频通话技术

WebRTC[28][29]是一种近年来十分流行的音视频通话技术，由Google开发和维护，并完全免费开源。WebRTC技术可以使两个客户端在通过NAT技术[30]建立点对点的连接，从而实现视频流任何其他数据的点对点传输。WebRTC 虽然一开始是Google针对Chrome浏览器所开发的技术，但是它一样可以使用到Android等移动端操作系统中。

WebRTC建立点对点连接的主要过程如下：首先客户端双发需要交换SDP即会话描述协议，协议内容包括媒体类型，传输协议，使用的编解码器，网络状况等等，用于客户端双发相互沟通。接下来需要使用ICE框架，通过STUN或者TURN服务器，进行NAT穿越，将两个在不同网段的设备拉入到同一网段中，如果NAT成功，最终两个客户端即可进行直接的点对点的数据通讯。

上述过程基本上就是进行实时通信需要经历的过程，整个过程看似很简单，但实际上端与端之间通常存在有很多层的防火墙和 NAT 设备阻隔着，需要经过一系列繁琐的信令交换，网络地址转换、打洞等过程。幸运的是，WebRTC提供了一些系列开源的解决方案，开发者无需过多考虑底层实现，从而可以将绝大多数的精力放在主要的业务逻辑上。

为了支持WebRTC我们需要架设三类服务器，一种是信令服务器，负责协调通讯的过程，主要包括会话控制信息、错误信息、元数据、网络数据等。 另外还需要打洞服务器，一旦会话建立，WebRTC会首先尝试使用点对点连接。而客户端往往都位于不同的网段内，中间还有防火墙等重重阻隔，所以需要专门的NAT服务器，利用STUN或者TURN服务器尝试将两个客户端连通。最后，还需要一个房间服务器，专门负责房间的建立和销毁等等逻辑。

## 2.4　协同过滤推荐算法

推荐算法[31]的目标是根据用户的历史行为，向用户推荐其最可能喜欢的物品。推荐算法领域有非常多种不同的算法思想，例如，基于关联规则的推荐算法，基于内容的推荐算法，以及协同过滤推荐算法等等。而协同过滤推荐算法[32] 是目前在学术界和工业界被讨论和使用最多的推荐算法之一。

协同过滤推荐算法是非常经典的一类推荐算法。协同过滤推荐算法可以分成两种类型，分别是基于近邻的[33]协同过滤算法和基于模型的协同过滤算法。

基于近邻的协同过滤算法还可以进一步分为基于用户的和基于物品的。基于用户的协同过滤算法，首先需要计算用户之间的相似度，从而发现相似的用户群，找出该用户群所最喜欢的物品，并且目标用户尚未知晓的物品，推荐给目标用户即可。类似地，基于物品的协同过滤算法[34]，首先需要算出物品之间的相似度，从而确定与目标用户所喜欢的物品的相似程度最高的物品，推荐给目标用户即可。

而基于模型的协同过滤算法，根据模型的不同，又有许多不同的具体实现，本文使用了基于线性拟合[35]的协同过滤推荐算法。由于该算法需要用户对物品评分，而对于本文的推送系统，让用户进行主动的评分显然是不现实的，所以本文构建了一个根据用户接收到推送消息后的一系列反应，自动对推送消息进行评分的模型。

## 2.5　整体架构设计

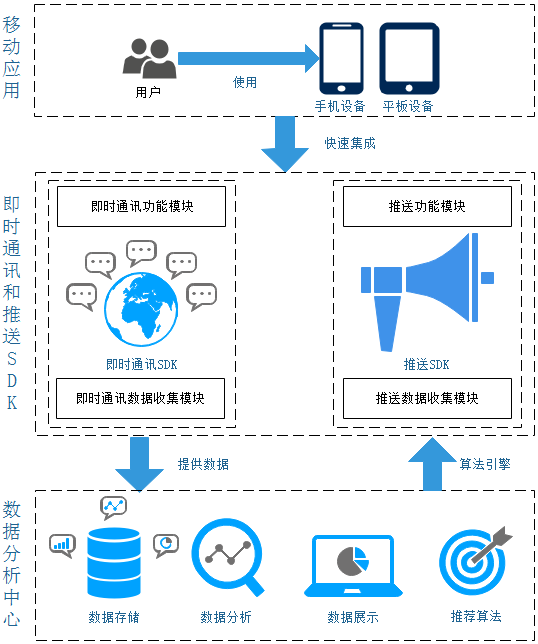


图2-4　整体架构设计图

本设计在以上技术的基础上，实现了即时通讯和推送服务。本设计在总体架构上分为三部分，（1）即时通讯客户端SDK，负责即时通讯基本功能的实现和相关用户数据的采集。（2）推送客户端SDK，负责推送功能的实现和相关用户数据的采集。（3）数据分析中心模块，负责分析来自客户端SDK的用户行为数据，一方面负责将收集到的各种数据做存储，分析并对开发者展示结果，另一方面针对推送服务使用协同过滤算法对用户进行推荐等。整体架构图如图2-4所示。

## 2.6　本章小结

本章首先对即时通讯的底层协议XMPP协议和推送的底层协议MQTT协议进行了介绍；由于本文的的客户端针对Android平台，所以对使用到的Android客户端技术进行了介绍，然后对协同过滤算法进行了介绍，最后对本系统的整体架构和运行流程进行了详细的描述。

# 第三章　即时通讯服务的设计和实现

## 3.1　即时通讯SDK实现

客户端即时通讯SDK为Android客户端提供即时通讯的功能包括登录。注册，单聊，群聊，发送图片，发送文件，发送语音，视频通话，新建群聊，添加联系人，个人设置等功能。SDK的使用分为两种方式，开发者可以选择接入页面，并可以对页面进行外观的定制；而另一种，可以选择只接入相关接口，页面由自己开发完成。开发者可以根据自己的需求选择适合自己的接入方式。

### 3.1.1　实现背景

即时通讯SDK具有比较多的功能，代码量较大，不同的模块之间耦合度较大，所以选择了MVP作为本工程的架构模式，大大降低了工程的耦合性，使模块之间的关系更加清楚，提高了工程的易读性和可维护性。本工程为多进程工程，使用一个专门的进程负责消息的发送和接受，进程间通过AIDL进行跨进程通讯，提高了SDK的稳定性。另外，SDK还负责收集用户的部分行为数据，并上传至数据统计服务器，供开发者参考。

### 3.1.2　即时通讯SDK整体架构设计

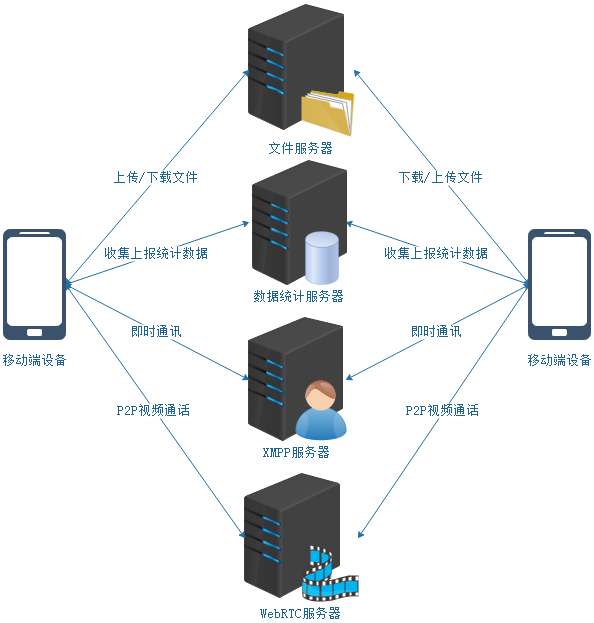


图3-1　即时通讯SDK服务器架构图

从服务器架构上，即时通讯平台需要搭建XMPP即时通讯服务器，文件服务器，数据统计服务器以及WebRTC服务器，服务器架构图如图3-1所示。

从功能模块上看，即时通讯SDK的功能可分为四个大模块，分别为：用户状态管理模块，联系人管理模块，收发消息模块，群聊模块。其中状态管理模块包括用户注册、用户登录、用户注销的功能；联系人管理模块具有获取联系人，添加联系人，删除联系人，查看联系人详情的功能；收发消息模块具有发送文本，发送语音，发送图片，发送与语音和发送文件的功能；群聊模块包括新建群聊，加入群聊，退出群聊和删除群聊的功能。整体的模块图如图3-2所示。

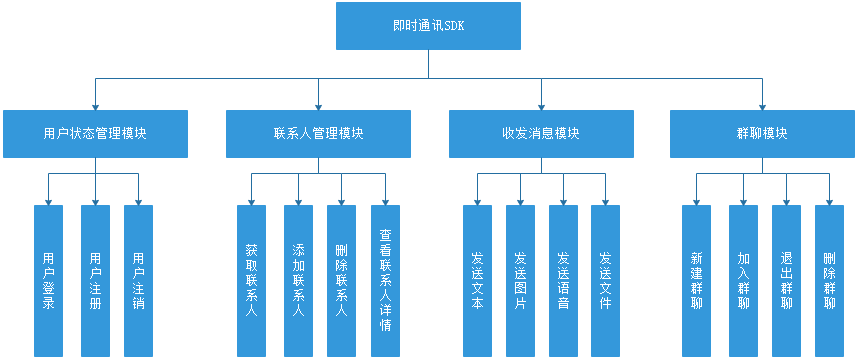


图3-2　即时通讯SDK模块图

另一方面，从提供的库结构上看，即时通讯SDK总体上分为两个库，分别是imui和imcore，其中imui依赖imcore。

imui中包含了所有界面，包括登录界面、单聊界面、群聊界面、会话展示界面、好友列表界面、个人设置界面等。imui还包括了所有界面的逻辑操作。MVP工程架构的View层和Presenter层都在这个Library中实现。imui对外提供各种界面，开发者可以十分方便的通过imui库直接获取到即时通讯的必须页面，并对外观进行各种定制，以最快满足自己的需求。

imcore包含了所有的基本数据和基本逻辑，例如登录，注册，获取好友，获取会话，发送消息，获取个人信息等，并对外提供这些功能的接口。imcore实现了MVP工程架构中的M层，是即时通讯SDK的基本核心库。Imcore使用了本地数据库对消息、会话等进行持久化，并对其提供三级缓存功能。

imui和imcore的依赖结构图如图3-3所示。

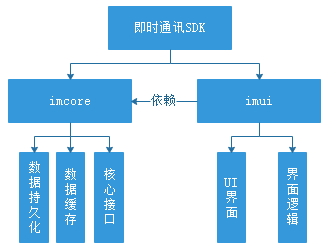


图3-3　即时通讯SDK依赖结构图

### 3.1.3　即时通讯SDK重要模块实现

#### 3.1.3.1　用户管理模块

登录过程包含两部分内容，连接远程服务器和本地初始化。

首先，登录接口被调用后，SDK尝试连接Openfire服务器，确保可以正常连接后登录服务器，并向服务器发送类型为available的Presence状态报文，通知服务器该用户已经上线，至此就完成了跟服务器的连接和账号认证的工作。

认证完成后，即开始对客户端做一系列初始化工作，首先需要开启负责消息的发送和接收的消息进程，并拉取尚未接收到的离线消息，同时从本地数据库缓存或者服务器拉取联系人列表，会话列表，个人信息等数据，接下来通过Android中的Preference将登录状态保存至本地，从而下次启动的时候不需要用户再次输入用户名和密码完成自动登录。

完成以上工作后即可进入主界面，将拉取到的所有信息展示到界面上，完成整个登录流程。

相反的，注销用户登录状态的流程，首先需要向Openfire服务器发送类型为unavailable的Presence状态报文，并断开和服务器的连接，在客户端方面，一方面需要将Preference中的登录状态删除，确保下次启动时无法自动登录；另一方面停止负责接收消息的Service。即可完成全部的注销流程。

#### 3.1.3.2　联系人管理模块

在XMPP协议中，所有的联系人信息都交由Roster来管理，客户端可以通过Roster获取所有联系人以及删除联系人。服务器又为每一个联系人维护了一个VCard信息，VCard用来记录联系人的详细信息，包括：昵称、性别，头像，生日等等。XMPP的设计考虑到了不同的应用对联系人具有的信息种类差别很大，所以除了这些基本的信息，VCard支持自定义任何的用户信息，具有很好的扩展性。通过Roaster得到联系人的id后，便可以通过VCardManager类的loadVCard方法获取任何一个联系人的详细信息。

相比获取联系人，添加联系人的流程较复杂，A用户添加B用户的流程如下：首先A用户需要向Openfire服务器发送类型为subscribe的Presence状态报文，告知服务器A用户想要添加B用户为联系人，Openfire收到订阅消息后，会向B用户发送订阅消息，B用户选择是否同意A用户的联系人申请，如果同意，B用户需要向服务器发送一个subscribe类型的Presence报文，告知服务器B用户接受了A用户的联系人申请，服务器收到订阅消息后，会向A用户发送用户B接受了联系人申请的消息。从而完成整个添加联系人的流程。

#### 3.1.3.3　发送和接收消息模块

系统中有很多种类型的消息，消息类型及其对应的含义如表3-1所示。

表3-1 消息类型表

|  |  |
| --- | --- |
| 消息类型 | 含义 |
| MYSELF\_TEXT | 自身所发的文本消息 |
| MYSELF\_IMAGE | 自身所发的图片消息 |
| MYSELF\_VOICE | 自身所发的语音消息 |
| MYSELF\_FILE | 自身所发的文件消息 |
| TARGET\_TEXT | 联系人所发的文本消息 |
| MYSELF\_IMAGE | 自身所发的图片消息 |
| MYSELF\_VOICE | 自身所发的语音消息 |
| MYSELF\_FILE | 自身所发的文件消息 |
| TARGET\_TEXT | 联系人所发的文本消息 |
| TARGET\_IMAGE | 联系人所发的图片消息 |
| TARGET\_VOICE | 联系人所发的语音消息 |
| TARGET\_FILE | 联系人所发文件消息 |
| VIDEO | 视频通话请求 |
| VIDEO\_CANCEL | 取消了自身发起的视频通话请求 |
| VICEO\_BACK\_ACCEPT | 接受了联系人所发起的视频通话请求 |
| VIDEO\_BACK\_REFUSE | 拒绝了联系人所发起的视频通话请求 |
| INVITE | 添加联系人请求 |
| INVITE\_BACK\_ACCEPT | 接受了联系人所发起的添加请求 |

发送文本消息较为简单，只需新建Chat类型的Message对象，将联系人用户名和内容通过Stanza报文发送至服务器，服务器即可发送至相应联系人，联系人收到后，展示到响应页面即可。

发送消息的关键代码如图3-4所示，传入的参数有三个，content为消息的内容，若是文本消息，content为文本内容，若为图片、文件、语音等消息，content则为其在文件服务器上的URL；type是消息的类型，isGroup为消息是否是群聊消息。方法首先实例化了一个Message对象，并将发送消息的对象的用户名和Message的类型chat传入其构造方法。然后将content传入message的setBody方法，将isGroup和type以JSON的格式传入message的setSubject方法。最终通过XMPPConnection的sendStanza方法将message发送给对方用户。

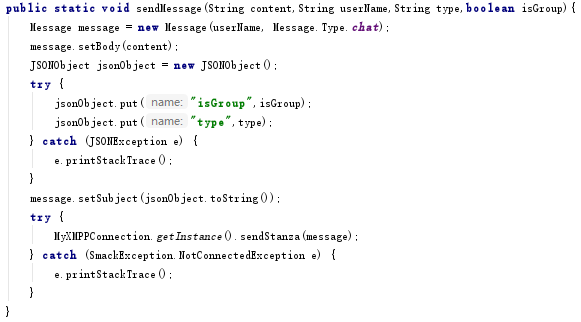


图3-4 发送消息关键代码

而对于图片，语音和文件的发送，则较为复杂。不管是图片，语音和一般文件，其本质都是文件，XMPP协议和Openfire服务器目前只支持文本的传输，而并没有现成的方案支持文件的传输。有一种方案可以是利用Base64编码[36]将文件转化为文本，这样处理从逻辑和工程上都非常简单，但是却有不少较为严重的问题，一方面使用Base64转化后的文本，会比原文件的数据量大三分之一左右，使用户白白损失流量；另一方面，使用Base64编码，无论是文本转文件，还是文件转文本，效率都是很低的，尤其是对于较大文件，编码时间会较长，增加了用户等待的时间。

考虑到Base64方案诸多严重的缺点，本设计使用了另一种方案，即搭建另外一个基于HTTP协议的文件服务器，该文件服务器对外提供文件的上传和下载服务。那么此时发送文件就非常简单了，A用户向B用户发送一个文件的流程如下：首先A通过HTTP协议将文件上传至文件管理服务器，上传完成后获取到图片的URL，再将URL信息发送至B用户，B用户收到消息后，通过URL从文件管理服务器上下载该文件，并展示到界面上，从而完成整个文件发送的流程。发送文件的完整流程如图3-5所示。

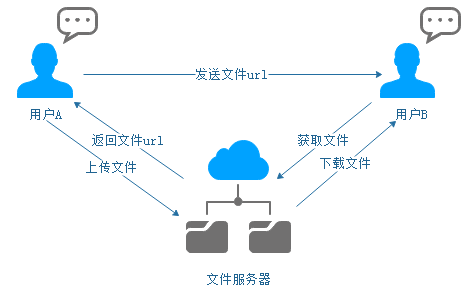


图3-5 发送文件架构图

#### 3.1.3.4　音视频通话模块

音视频通话是较为独立的一个模块，XMPP协议以及Openfire服务器显然没有提供这个功能。本设计的音视频通话通过WebRTC技术实现，WebRTC使用P2P的模型，一组用户之间通过NAT技术直接连接，并且具有免费开源，支持多终端等优点，非常适合我们的场景。

用户A跟用户B进行音视频通话的流程如下：首先用户A根据双方的ID和时间戳生成一个全局唯一的音视频通话房间ID，将音视频通话请求和房间ID一并发送给用户B，同时通过WebRTC技术，向音视频通话服务器发送请求，创建并进入音视频通话房间，然后等待用户B的响应。用户B收到邀请信息后，进入音视频通话接受界面，在该界面类似接听电话界面，用户B可以选择接受或是拒绝用户A的音视频通话请求，如果同意，则向用户A发送同意消息，并通过接收到的房间号进入改音视频通话房间；若拒绝，则向用户B发送拒绝消息，A用户收到拒绝请求后，关闭音视频通话房间。

### 3.1.4　功能接口化实现

由于本设计最终目的是提供一个即插即用的SDK给移动应用的开发者，所以代码设计上必须可以让开发者可以很方便的调用相关接口和页面。

对于imui库，需要对外暴露所有的页面，即所有的Activity和Fragment。同时考虑到开发者希望引入我们的界面风格，与本身应用的界面风格保持一致，所以对于每个界面，我们都开放了很多可定制化的选项，例如标题栏的颜色，聊天气泡的颜色，聊天文字的颜色，主界面图标的图片等等。开发者可以通过很简单的操作，对界面进行高度的定制化，满足自己的需求。

通过即时通讯SDK打开定制化的主页面的关键代码如图3-6表示，页面的定制化过程使用了建造者模式，开发者首先需要实例化一个主页面的Builder对象，通过该对象分别设置了标题栏颜色为蓝色，标题文字为“OuNews”，标题文字颜色为黑色，并通过ActivityRepository类中的静态类IMMainActivityTool中的静态方法startIMMainActiivty，将上下文和建造者Builder传入，即可打开一个定制的主页面。

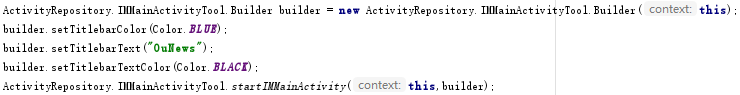


图3-6 打开定制主页面关键代码

对于imcore库，需要对外暴露所有的逻辑接口，对于一些耗时的操作，例如登录，发送消息，获取所有会话等，我们将这些方法规定为异步方法，通过接口回调的方式，获得结果；而对于一些瞬时操作，例如改变用户的登录状态等，我们直接规定为同步的方法，通过返回值获得结果。开发者可以根据自己的需求选择不同类型的方法。

开发者调用登录接口的关键代码如图3-7所示。首先通过IMCore的静态方法getInstance获取单例对象，通过调用该单例对象的login方法实现登录，传入的参数分别为用户名，密码和登录的回调接口。该方法是一个异步的登录方法，调用该方法后需要在OnLoginListener回调接口中实现onSuccess和onFailed，前者会在登录成功后回调，后者回在登录失败后回调，参数code为错误代码。

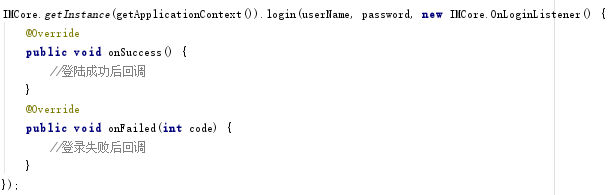


图3-7 登录接口关键代码

## 3.2　相关数据收集和上报

本设计的即时通讯SDK不仅提供了即时通讯的功能，更进一步的，SDK对用户在使用过程中产生的行为数据进行采集和上报。数据的收集，一方面用户设备的数据使用Android系统提供的类即可获取，另一方面，关于消息等用户具体的行为数据，需要在关键代码处埋点获取相应数据。数据的上报使用HTTP协议，将数据上报至数据分析中心。

### 3.2.1　实现背景

根据数据的产生方式可以将手机的数据分为两类，一类是用户基本数据，包括每日使用用户，新增用户，用户的地域分布等，这些数据时用户启动SDK后就可以马上获取的，开发者和运营商可以根据这些用户设备数据，了解该应用用户群的具体情况。而另一种数据是用户行为数据，包括每日产生的消息数，消息产生的时间分布等等。这些数据，是由用户进行了某个行为，使用了某个功能后才产生的，开者可以根据这些数据深入地了解某一个功能具体的用户使用场景是怎样的，以及用户的行为是怎样的，有助于进一步帮助开发者优化相关功能。

### 3.2.2　用户统计和分析

用户信息主要包括用户的日使用次数，以及日新增用户，运营人员可以根据这些数据判断应用的发展趋势，为运营决策提供可靠地依据。

用户日使用次数可以很直观的衡量一个应用当前的用户状况，统计方法较为接单。当用户每次打开应用时，即向数据分析中心发送一条用户使用的数据，数据格式如图3-8所示。

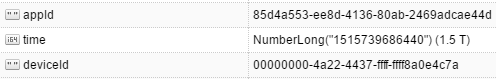


图3-8 日使用次数数据格式图

各字段的含义如表3-2所示。

表3-2 日使用次数数据字段含义表

|  |  |  |
| --- | --- | --- |
| key | 含义 | 数据类型 |
| appId | 应用唯一id | 字符串 |
| time | 服务器时间 | 长整型 |
| deviceId | 设备唯一id | 字符串 |

日新增用户是衡量一个应用发展趋势的重要标准，统计日新增用户可以根据用户是否是第一次打开应用来判断。当用户第一次打开应用时，即向数据分析中心发送一条新增用户的消息，并且包含了更为丰富的信息，包括用户使用的sim卡运营商，收集生产商，系统版本，所在地域等，可以使应用运营者更加了解用户的具体情况。消息的具体的消息格式如图3-9所示。



图3-9 日新增用户数据结构图

推送消息的各字段的含义如表3-3所示。

表3-3 日使用次数数据结构表

|  |  |  |
| --- | --- | --- |
| key | 含义 | 数据类型 |
| appVersion | app版本 | 字符串 |
| osVersion | 操作系统版本 | 字符串 |
| simProvider | sim卡供应商 | 字符串 |
| appId | 应用唯一id | 字符串 |
| time | 服务器时间 | 长整型 |
| deviceId | 设备唯一id | 字符串 |
| manufacture | 设备生产商 | 字符串 |

### 3.2.3　消息统计和分析

消息统计主要包括每天产生的消息个数，消息类型的个数占比，消息产生的时间占比。运营人员可以通过消息的相关统计，更加了解用户发送消息的习惯，和使用时间的习惯等。为客户端甚至服务端的改进提供参考依据。由于消息可能涉及到较多用户的隐私，是非常敏感的数据，所以我们并不收集消息的具体内容。

当用户发送消息时，向数据分析中心发送消息统计数据，数据格式如图3-10所示。

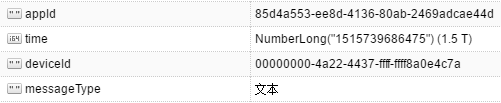


图3-10 消息数据结构图

消息统计数据的各字段的含义如表3-4所示。

表3-4 消息数据结构表

|  |  |  |
| --- | --- | --- |
| key | 含义 | 数据类型 |
| appId | 应用唯一id | 字符串 |
| time | 服务器时间 | 长整型 |
| deviceId | 设备唯一id | 字符串 |
| messageType | 消息类型 | 字符串 |

## 3.3　本章小节

本章首先讨论了客户端SDK的整体机构，详细介绍了用户管理，联系人管理，发送和接受消息以及群聊等几个重要模块的设计和实现，以及功能接口的实现方法，接着介绍了用户行为数据包括：用户、消息和群组的收集和采集的实现原理。

# 第四章　推送服务的设计和实现

## 4.1　推送SDK实现

客户端推送SDK为Android客户端提供最基础的推送功能，它一方面负责负责将从服务器收到的推送消息，以通知栏弹出的形式展示给用户；另一方面，推送SDK还负责收集用户收到推送消息后的各种行为数据。为了保证客户端的推送接收服务能够尽量保持活跃状态，推送接收模块使用Android的Service后台组件，并使用了很多方法对该Service进行保活。

### 4.1.1　实现背景

跟即时通讯SDK类似，SDK工程需要做到简单易用，尽量少的的与目标工程耦合。开发者只需依赖SDK，添加少量的配置和代码，即可推送完成SDK的接入。推送SDK的推送功能基于MQTT协议，服务端使用Apollo服务器，与服务器之间通讯使用JSON格式。

### 4.1.2　客户端推送SDK基本功能实现

#### 4.1.2.1　推送消息的接收

因为整个推送系统使用MQTT协议，所以客户端SDK启动后，需要先配置好连接服务器的相关参数，例如用户名，密码，超时时间等，配置好参数后，既连接并登陆MQTT服务器Apollo，关键代码如图4-1所示。



图4-1 推送SDK连接登录服务器关键代码

连接并登陆服务器后，需要订阅相关话题，本设计所有客户端SDK皆订阅了两个相关话题。一个话题是该应用的appId，即所有该应用的用户全部订阅了同一话题，从而推送端只需要向该话题发送推送消息，则该应用额所有用户便能够收到；另一个话题是设备唯一识别码，方便推送端可以单独向某个设备发送推送消息。

登录连接服务器，并订阅相应话题后，客户端便能够接受到来自服务器端推送来的消息。推送消息以JSON格式传输并存储，一条推送消息的格式如图4-2所示。

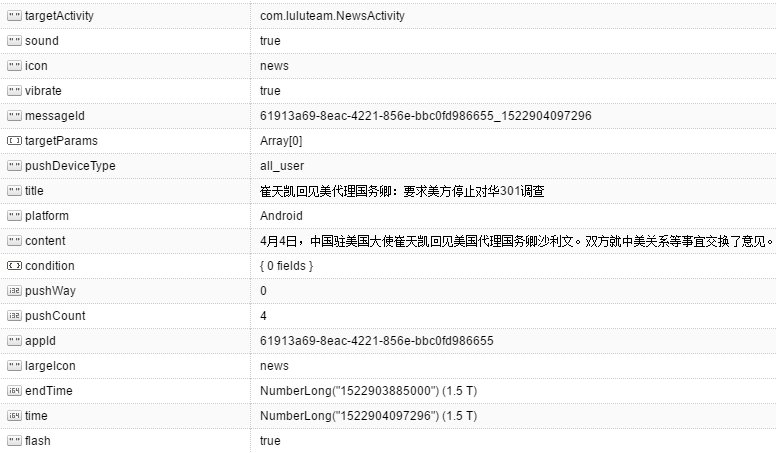


图4-2 推送消息数据结构图

推送消息的各字段的含义如表4-1所示。

表4-1 推送消息数据结构表

|  |  |  |
| --- | --- | --- |
| key | 含义 | 数据类型 |
| targetActivity | 点击后的目标页面的类名 | 字符串 |
| sound | 是否有铃声提示 | 布尔值 |
| icon | 推送消息图标的资源名称 | 字符串 |
| vibrate | 是否有震动提示 | 布尔值 |
| messageId | 推送消息唯一id | 长整型 |
| targetParmas | 跳转到目标页面后的参数 | 数组 |
| pushDeviceType | 目标设备类型 | 字符串 |
| title | 推送消息标题 | 字符串 |
| platform | 推送设备平台 | 字符串 |
| content | 推送内容 | 字符串 |
| condition | 目标设备条件 | 数组 |
| pushWay | 推送时间类型 | 整型 |
| pushCount | 该推送消息的目标用户数 | 整型 |
| appId | flash应用唯一id | 字符串 |
| largeIcon | 推送消息大图标的资源名称 | 字符串 |
| endTime | 推送消息截止时间 | 字符串 |
| time | 服务器时间 | 字符串 |
| flash | 是否有呼吸灯提示 | 字符串 |

本设计为了客户端能够持续不断的接收到推送消息，使用了Service进行推送消息的接收。顾名思义，Service是Android系统中的后台服务，它没有界面，往往有着更长的生命周期，跟Activity相同，Service同样必须在配置文件中声明。

Service的保活，即尽可能的保持Service一直活跃而不被杀死，这项技术在Android开发中常常会用到，而这对于我们的推送系统来说同样十分重要，本设计使用了以下措施对Service进行保活。

（1）开启专门的推送进程，Android中同一个应用支持多进程，专门开启一个推送进程，负责接受推送消息的Service在该推送进程中开启，可以有效避免主进程被杀死后，推送服务马上随之被杀死。

（2）提高Service所在进程的优先级，在Android系统中，Service的优先级被分为了五级。当Android设备出现内存不足等情况时，系统会优先选择低优先级的进程进行回收。提高Service优先级可以避免Service在出现内存不足等情况时被系统所杀死并回收。

（3）利用系统广播拉活，如果我们的推送Service已经被系统杀死，我们需要有另外的机制，在合适的时机，对推送Service进行重新拉活。利用系统广播拉活是十分简单而实用的方式之一。Android系统在发生特定事件时，例如开机、网络变化，锁屏解锁，会对整个系统发送特定的广播，我们只需在应用内定义一个静态广播接收者，每当接收到这些广播时，开启推送Service进程。

#### 4.1.2.2　推送消息的展示

客户端收到推送消息的JSON文本后，需解析文本，得到原始的推送消息，最终通过顶端通知栏弹出的形式展示给用户，在Android系统中，通知栏弹出的一条消息，被称为一个Notification[37]。

Notification在Android系统中是一种具有全局效果的通知，当 APP 向系统发出通知时，它即马上在设备顶端弹出一条消息显示给用户，而用户可以通过下拉顶部通知栏，查看改消息的详细信息，并进一步选择打开或者取消。Notification已经成为了Android系统中的展示推送消息的统一标准。

在Android 3.0之前，新建一个Notification只需使用new Notification()构造函数即可，而在Android 3.0之后的版本，Google加入了Notification.Builder类，使用建造者模式创建参数较为复杂的Notification对象。同时为了兼容低版本，support.v4包中加入了NotificationCombat.Builder类，其使用方法与Notification.builder类基本相同。创建一个简单的Notification对象的关键代码如图4-3所示。

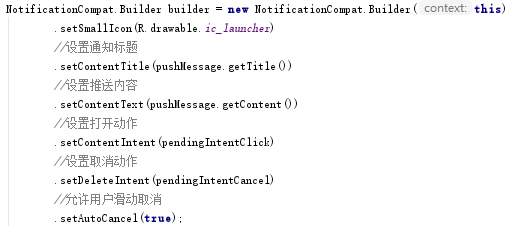


图4-3 创建Notification关键代码

而进行打开和取消推送消息的动作，则需要使用到Android系统中另一个十分重要的组件之一：BroadCastReceiver，即广播接收者，当用户打开或者取消推送消息时，这向对应的PendingIntent中设置好的广播接收者发送广播，对应的广播接收者接收到相应的广播后，进行页面打开和统计上报等操作。

广播接收者的注册，分为动态注册和静态注册，动态注册指在代码中进行注册，静态注册指在配置文件中进行注册，本设计采用了静态注册，配置文件中的关键代码如图4-4所示。

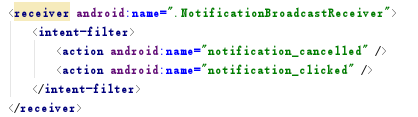


图4-4 注册广播关键代码

#### 4.1.2.3　离线推送消息的接收

上一小节使用了许多保活方案，保证Service尽可能的不被杀死。然而事实上，除了QQ，微信等大型应用，跟绝大多数手机厂商都有合作，从而保证进程不被杀死，对于一般的应用，我们很难保证某个Service一定不会被杀死。所以我们必须保证当推送Service被杀死，而没有接受到某些推送消息，当下次打开应用时会自动获取尚未过期的推送消息。

一方面当推送端向客户端推送消息时，会将推送消息保存到数据库中；另一方面，当客户端接收到推送消息后，会马上向数据分析中心发送一条统计数据，表示客户端顺利接收到了该条数据，数据分析中心接收到后会保存到数据中。而获取离线消息也正是使用到了这两种数据。当推送SDK启动时，会向服务器端获取离线消息。服务端只需查找数据库中，查找已经向该应用发送的并且尚未过期的推送消息中，该客户端上尚没有接受的推送消息，此即为离线消息，服务器获得离线消息后返回给该客户端，客户端使用Notification进行相应地展示。

#### 4.1.2.4　推送消息的后续动作

在绝大部分的场景中，当用户接收到推送消息并点击后，都会有后续动作，一般是跳转到某一个页面，本设计的SDK也包含了这个功能。Android系统中想要打开一个新的页面只需这个页面的类对象，所以用户只需在推送页面上，输入工程中相应页面的全类名。每条推送消的内容都是不同的，所以页面的跳转一般都会带有一个或者多个参数。当用户跳转到相应页面后，再通过参数从开发商的服务器上获取对应的内容。

在Android系统中，根据某个类的全类名获取到该类，需要使用反射技术。简单来说，Java反射技术是指在工程内虚拟机未加载某个类的情况下，通过类名动态获得某个类的信息，并进行相关的操作。

关键代码如图所示，首先通过Intent的getStringExtra方法的到后续目标页面的全类名，如果其不为空，则使用Class类的静态方法forName，通过反射获取后续目标页面的类，将其他的参数写入到新的Intent对象中，通过context的startActivity方法，打开相应页面，并传入相应参数，该页面接收到参数后地相应动作由开发者根据自己的逻辑进行开发，通过反射打开后续目标页面的关键代码如图4-5所示。

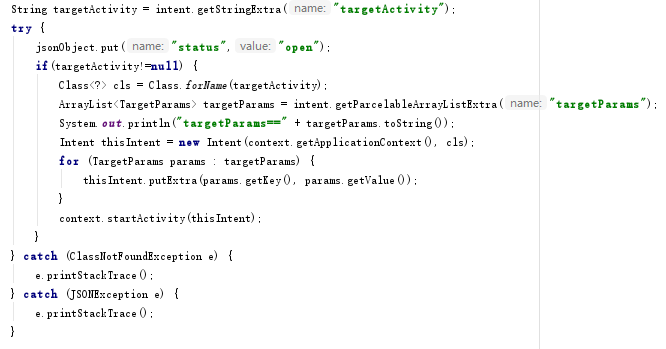


图4-5 打开后续目标页面关键代码

## 4.2推送端的实现

客户端集成推送SDK后，需要一个专门的推送端方便推送者进行消息推送，本设计使用Web页面作为推送消息的页面，推送者通过推送页面将推送消息发送给后台后，后台根据推送消息的参数，将推送消息通过MQTT协议最终推送给相应的客户端，完成整个推送的过程。其中，推送消息的参数主要包括：推送信息的标题，内容，是否开启呼吸灯、震动、铃声，后续目标页面的全类名，目标用户和推送时间。

### 4.2.1　实现背景

推送端对于一个推送系统来说也十分重要，推送端由两部分组成，一部分是推送端的前台页面，本设计使用Web页面作为前台页面，具有跨平台等优点，将界面设计的简洁清楚，可以使推送者使用起来更加方便。另一部分是推送后台，负责处理筛选推送用户，选择推送时间等逻辑，并将消息发送至MQTT服务器，后台设计要着重注意降低计算复杂度，从而提高整个推送过程的效率。

### 4.2.2　性能数据监测实现

推送端的前端Web页面，使用了BootStrap UI组件。BootStrap代码简洁，外观优美，可用于简单、快速的构建PC以及移动设备上的页面需求。本工程使用了BootStrap很多组件，以保证页面风格的统一，包括bootstrap-switch，bootstrap-datetimepicker，bootstrap-select等，三者分别用于构建页面中的开关、日历和下拉框组件。

推送者在推送前端填写各种推送消息的属性，包括：推送标题，推送内容，推送图标，是否开启声音、呼吸灯、震动，制定后续页面和参数，目标用户，和推送时间，推送者点击推送按钮后，把推送消息的所有属性组织成一个JSON格式的字符串，具体格式如图所示。并将该JSON对象通过AJAX的POST方法，发送至推送端后台。

### 4.2.3　后台推送功能实现

本设计的推送后台使用Java EE技术实现。一方面对推送者Web前端暴露接口，供推送端前端页面调用，一方面连接MQTT服务器，将推送消息发送至MQTT服务器。

推送后台启动时，首先连接并登录MQTT服务器，等待前端发送消息。当收到来自Web前端的推送消息后，取出JSON对象中的APPID，由于客户端的所有用户已经订阅了该APPID的话题，所有后台将该推送消息发送至该话题后，这个客户端的所有用户变能够接收到推送消息。

而在很多场景下，对于某些特定的推送消息，推送者并不想推送给全部用户。本系统支持根据设备的系统版本，APP的版本，和手机厂商进行部分用户的推送，使得推送更加有针对性。

另外，为了满足使用者一些比较特殊的场景，本系统还支持了定时推送和重复推送，定时推送是指推送者发出推送命令后，并不马上进行推送，而是根据推送者设定的时间，在某个时间点开始推送；重复推送是指，对于某一条推送信息，进行周期性的重复推送。

## 4.3　相关数据收集和上报

对推送数据的收集和上报模块，也属于推送客户端SDK中的重要组成部分，该模块主要的实现方法是在客户端相应的位置进行埋点，当用户有相应的操作后，使用HTTP协议的POST方法，将各类行为数据，以JSON格式上传至数据统计服务器，并保存至MongoDB服务器中，进而对这些数据进行分析并展示给推送者。

### 4.3.1　实现背景

推送系统可以准确、高效地向用户发送推送消息之后，推送者往往想进一步了解用户收到推送消息后的各种行为数据，本设计收集并上报的用户行为数据主要包括：用户是否打开了推送消息，用户打开推送消息的时间段，用户从接收达到打开推送消息之间的时间差，用户点击推送消息跳转到某个页面的停留时间等，客户端SDK将这些数据上传至服务器，为进一步优化推送系统，推送内容，以及相关推荐算法等提供可靠地依据。

### 4.3.2　推送消息数据采集

本文收集和上报的数据，主要包括：用户收到推送后打开还是取消了推送消息，用户从收到推送消息到打开推送消息所花的时间，用户打开或者取消推送消息的时间点，用户在后续目标页面停留的时间。

首先我们需要收集用户接受推送消息的数据。当客户端的推送Service接收到推送消息时，会向数据中心发送一条JSON格式的统计消息，表示客户端成功接收到了来自推送服务器的推送消息，消息的JSON格式如图4-6所示。

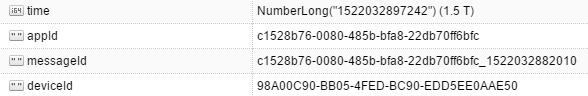


图4-6 接收到推送消息数据结构图

用户接收到推送消息数据的各字段含义如表4-2所示。

表4-2 接收到推送消息数据结构表

|  |  |  |
| --- | --- | --- |
| key | 含义 | 数据类型 |
| time | 服务器时间 | 长整型 |
| appId | 应用唯一id | 字符串 |
| messageId | 消息唯一id | 字符串 |
| deviceId | 设备唯一id | 字符串 |

用户对于推送消息的响应数据对于我们的分析来讲至关重要。当用户打开或者取消推送消息时，客户端会向数据中心发送一条JSON格式的统计消息，表示客户端对推送消息做出了回应，消息的JSON格式如图4-7所示。

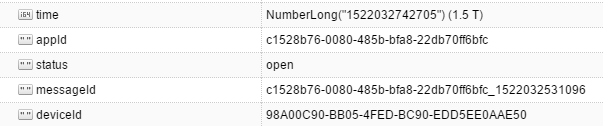


图4-7 回应推送消息数据结构图

用户回应消息数据的各字段含义如表4-2所示。

表4-2 回应推送消息数据结构图

|  |  |  |
| --- | --- | --- |
| key | 含义 | 数据类型 |
| time | 服务器时间 | 长整型 |
| appId | 应用唯一id | 字符串 |
| status | 用户打开或是取消了推送消息 | 字符串 |
| messageId | 消息唯一id | 字符串 |
| deviceId | 设备唯一id | 字符串 |

用户在后续目标页面的行为数据也十分重要，是衡量用户是否喜欢一个推送消息的重要参考标准。但因为目标页面存在于开发者的工程当中，所以需要开发者手动收集相关信息，并在在onPause方法中调用我们的接口，将用户在目标页面的停留行为数据上传至数据分析中心，目标页面停留数据的JSON格式如图4-8所示。

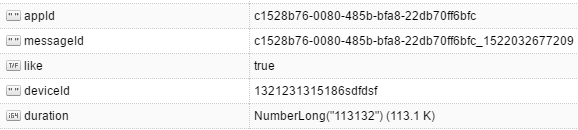


图4-8 目标页面停留数据结构图

目标页面停留数据的各字段含义如表4-3所示。

表4-3 目标页面停留数据各字段含义

|  |  |  |
| --- | --- | --- |
| key | 含义 | 数据类型 |
| appId | 应用唯一id | 字符串 |
| messageId | 消息唯一id | 字符串 |
| like | 用户是否有点赞或者收藏 | 布尔值 |
| deviceId | 设备唯一id | 字符串 |
| duration | 在页面停留时间 | 长整型 |

### 4.3.3　用户设备数据采集

用户设备的信息的收集，有两方面的意义，一方面有助于开发者更了解用户群的设备的具体情况，从而可以针对性的进行优化。另一方面也是实现针对用户设备进行部分推送的依据。用户设备数据主要包含：系统版本，APP版本，运营商，手机生产商等信息。这些信息通过Android SDK中的Build类可以方便的获得。

当用户第一次打开应用，对推送SDK进行初始化的时候，即收集相关信息向数据分析中心进行上报，数据同样适用JSON格式，具体格式如图4-9所示。



图4-9 用户设备数据结构图

用户设备数据的各字段含义如表4-4所示。

表4-4 用户设备数据各字段含义

|  |  |  |
| --- | --- | --- |
| key | 含义 | 数据类型 |
| appVersion | app版本 | 字符串 |
| osVersion | 操作系统版本 | 字符串 |
| simProvider | sim卡供应商 | 字符串 |
| appId | 应用唯一id | 字符串 |
| time | 服务器时间 | 长整型 |
| deviceId | 设备唯一id | 字符串 |
| manufacture | 设备生产商 | 字符串 |

SDK收集上报户设备信息的关键代码如图4-10所示，首先通过Preferences的getBoolean方法获取hasUploaded，hasUploaed表示是否已经上传过，如果其值为false，则表示没有上传过，即第一次打开，接下来通过SystemUtils工具类，获取设备id，app版本，设备生产商，操作系统版本，sim卡运营商等信息，并组织为JSON格式的字符串，最终通过HttpURLConnection将JSON字符串以POST方法发送至数据分析中心。确定上报成功即返回码为200后，将hasUploaed置为true，防止下次打开的时候重复上报。

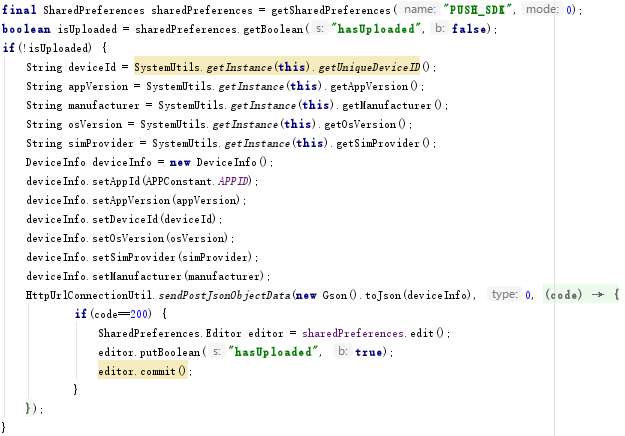


图4-10 上报用户设备数据关键代码

## 4.4　本章小节

本章介绍了推送服务的设计和实现。首先介绍了客户端SDK的设计和实现，以及关键问题的解决方法。接着介绍了推送端的设计和实现，包括推送前端和推送后台。最后介绍了客户端SDK中数据的采集和上报的实现方法。

# 第五章　数据分析中心的设计和实现

## 5.1 数据分析中心的整体结构

本设计的数据分析中心主要包含五个模块：数据接收和对外提供模块，数据存储模块，数据分析模块，基于协同过滤的推荐模块以及结果展示模块，数据分析中心的整体结构图如图5-1所示。

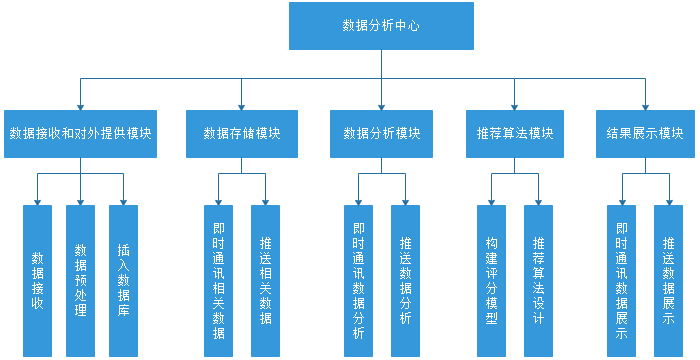


图5-1 数据分析中心整体结构图

数据接收和预处理模块负责接收来自客户端的各种JSON格式的数据，并且对数据进行合法性检验和预处理，预处理包括加入时间戳等，最终将数据插入MongoDB数据库。

存储模块负责将从客户端统计的各种JSON格式的数据，尤其是用户行为数据进行存储，并向其他模块提供数据来源。

数据分析模块负责将原本原始而杂乱的用户行为数据进行统计和分析，得出对开发者更直观，更有意义的数据。

基于协同过滤的推荐模块，主要针对推送系统，根据收集到的推送相关数据，建立评分模型，对推送消息进行自动评分，并使用基于线性回归的协同过滤算法进行推荐模型的构建。

结果展示模块负责，负责将数据分析模块得出的分析结果，在Web页面上使用各类图进行直观地展示。

在工程结构上，数据分析中心采用了J2EE分层架构[38]，使各个模块之间职责更加清楚，工程更加容易维护，从而大大提高了开发和维护的效率。

## 5.2 数据接收和对外提供模块

数据接收和预处理模块主要负责，一方面负责接收从客户端传过来的数据，并对数据进行合法性检验和预处理，并存储到数据库中。另一方面对前端Web页面提供数据，从而进行图表展示。

数据的接收主要使用Struts2进行HTTP接口的开发，接口主要分为两类，一类是数据的接收接口，接口URL以receive开头，另一类是对外提供相关数据，接口URL以get开头，主要的接口的路由如表5-1所示。

表5-1 接口路由表

|  |  |  |
| --- | --- | --- |
| URL路径 | 对应类名 | 功能 |
| /receivePushDeviceInfo | ReceivePushDeviceInfoAction | 收集用户设备数据 |
| /receivePushReceiveInfo | ReceivePushReceiveInfoAction | 收集用户接收推送消息 |
| /receivePushResponseInfo | ReceivePushResponseInfoAction | 收集用户推送消息相应 |
| /receivePushLoginInfo | ReceivePushLoginInfoAction | 收集推送用户使用次数 |
| /getWholePushInfo | GetWholePushInfo | 提供推送消息整体情况 |
| /getPushMessageDetail | GetPushMessageDetail | 提供推送消息整体情况 |
| /receiveIMRegisterInfo | ReceiveIMRegisterInfoAaction | 收集即时通讯日新增用户 |
| /receiveIMLoginInfo | ReceiveIMLoginInfoAction | 收集即时通讯使用次数 |
| /receiveIMMessageInfo | ReceiveIMMessageInfoAction | 收集即时通讯消息数据 |
| /getIMRegisterInfo | GetIMRegisterInfoAction | 提供即时通讯日新增用户 |

## 5.3 数据存储模块

数据存储模块是数据分析的核心模块之一，一方面它负责将即时通讯客户端和推送客户端的用户行为数据的持久化；另一方面，也为进一步的数据分析和推荐算法提供了数据源。本设计使用了MongoDB[39]作为数据库。MongoDB是一个由C++开发的，非常典型的NoSQL数据库，本设计使用MongoDB数据库，而非传统的关系型数据库如MySQL[40]等，主要考虑到以下几点：

本设计大部分的数据都使用JSON格式进行传输，而MongoDB本身就是使用的JSON格式进行存储，跟本设计的设计十分契合，JSON可以直接存储到MongoDB中，从而省去了大量序列化和反序列化JSON格式文本的时间。

本设计大部分数据表之间的关系较为松散，数据表之间的数据没有强关系性。同时，数据的格式并不固定。

本设计产生的数据，绝大部分是用户行为数据，这些数据会源源不断的产生，数据量较大，而MongoDB具有弱一致性，对海量数据进行存储和分析的效率更高。

本设计所用的数据表和相关含义如表5-2所示：

表5-2 数据表意义表

|  |  |
| --- | --- |
| 数据表名 | 数据表意义 |
| IMLogin | 即时通讯应用用户启动数据 |
| IMRegister | 即时通讯应用新增用户数据 |
| IMMessage | 即时通讯应用的消息数据 |
| PushLogin | 推送应用的用户启动数据 |
| PushRegister | 推送应用的新增用户数据 |
| PushMessage | 推送应用的推送消息数据 |
| PushReceive | 推送应用的消息接收数据 |
| PushResponse | 推送应用的用户响应数据 |
| PushPage | 推送应用的后续目标页面的相关数据 |

## 5.4 数据分析模块

数据分析系统最终的目的，是把用户的各种行为的数据经过各种分析处理，最终通过图表展示给开发者和运营者。而MongoDB数据库中存放的数据仅仅是原始的用户行为数据，想要获得更加直观有意义的数据还需要数据分析模块做进一步的分析和处理。

数据分析系统的分析模块，主要使用了MonngoDB的统计聚合功能，MongoDB主要提供了三种对数据进行聚合和统计的工具，分别为：count，distinct等基本的聚合命令；Aggregation框架；MapReduce分布式框架[41]。三种方式各有长处，适用于不同的场景，可以帮我们高效率的处理一些复杂的统计分析逻辑。

基本的聚合命令为聚合任务提供了几个较为简单的工具，例如count方法提供了计算数据数量的功能，distinct方法提供了寻找不同值的功能。

Aggregation框架提供了一系列的构建，包括排序、限制、投射、跳过、分组等等。

使用多个构件可以创建一个管道，通过管道对数据库中的文档进行复杂的处理。

MapReduce框架，MongoDB中的MapReduce聚合框架使用JavaScript作为其描述语言，相对于其他统计工具，功能最为强大。顾名思义，使用分布式计算的思想对文档进行处理。MapReduce框架适用于处理一些复杂的聚合逻辑，但是其执行速度相对较慢。

本文主要使用基本聚合命令和Aggregation对数据进行各种统计计算。

## 5.5　基于协同过滤的推荐系统模块

对于推送服务，除了一些重要的推送消息，需要推送给全部的用户，开发者和运营商往往希望能够根据用户的喜好进行个性化的推送，一方面对用户推送其喜欢的内容可以增加用户对应用的好感，相反减少了给用户推送其讨厌的内容，又减少了用户对于应用的厌恶感，所以构建一个高效准确的推荐系统对于一个应用来说是十分有意义的。

内容推荐是机器学习领域一个非常典型并且重要的应用领域，而协同过滤算法是推荐系统中非常经典的的一类算法。本文使用了基于线性回归的协同过滤算法。线性回归是机器学习中的一种模型。算法通过对线性回归的代价函数运行梯度下降算法，求出相应参数的最优解。最终通过参数得出用户最可能喜欢的推送内容。

### 5.5.1　推送内容评分的构建

本文使用的基于线性回归的协同过滤算法，必须使用到用户对于其收到的推送消息的具体评分，而我们很难让用户主动给推送内容进行评分，所以需要几个参数根据用户的行为数据，自动产生用户对推送内容的评分。这几个参数主要包括，用户是否点击了推送消息，用户从收到推送消息到打开或取消推送消息所用的时间，用户在目标页面所停留的时间，以及用户是否在目标页面进行收藏或点赞。这些参数都是判断用户是否喜欢一个推送消息的重要参数。

用户对于推送内容的评分可以分为两大类，一种是对于用户取消了推送消息的评分，一种是对于用户点击了推送消息的评分，我们规定用户对于推送消的评分的范围为0到10分，分数越高表示用户越喜欢该推送内容。则对于用户取消了的推送消息规定评分必须在0到5分之间，而对于用户点击了的推送消息的评分必须在5到10分之间。

对于用户的评分给出以下定义：

**定义5.1**：用户反应时间，表示为，含义是用户从接收到推送内容，到打开或者取消的时间间隔。

**定义5.2**：用户停留时间，表示为，含义是用户打开后续目标页面后，在该页面停留的时间。

对于用户取消了的推送消息的评分，并不能一概而论都设为0分。用户收到推送消息后，如果发现时自己十分讨厌的推送消息，往往会马上划掉。所以从收到推送的时间到用户取消的时间越短，用户的评分应该越低。

根据以上分析，对于用户取消了推送消息的情景，给出评分公式如（5-1）所示。

 (5-1)

其中表示用户对于推送内容的评分，其值域为，为调整系数，此处取值为15较为合适。

对于用户点击了的推送消息的评分，有更为多的纬度进行衡量。首先，用户收到推送消息后，如果发现时用户喜欢的消息，往往会快速的点击推送消息，所以从用户收到推送消息到用户点击推送消息的时间越短，用户的评分应该越高。另外，用户点击推送消息后，会跳转到后续目标页面，用户后续页面停留的时间代表着用户越喜欢这个推送消息，所以用户在后续页面停留的时间越长，评分应该越高。最后，用户在后续目标页面如果进行了收藏或者点赞，则可以最为准确的判断用户喜爱这个推送消息。

根据以上的分析，对于用户打开了推送消息的情景，给出评分公式如（5-2）所示。

 (5-2)

其中s表示用户对于推送内容的评分，其值域为，为调整系数，此处取值同样为15，表示用户是否有收藏或者点赞，有的话则等于1，否则等于0。

上文分别从用户打开了推送消息和用户取消了推送消息两种情况，分别构建了用户对于推送消息的评分公式。使用该评分公式，即不需要用户主动给推送内容评分，而是根据用户对推送消息的反应，自动进行较为客观的评分。

### 5.5.2　算法设计

为了能够更加清楚的对本算法进行介绍，本节首先给出算法中涉及的相关概念和符号表示。假设为系统中用户的个数，为系统中推送消息的个数，假设我们把推送消息分成了种消息类型。给出以下定义：

**定义5.1**：隐语义因子，表示为，含义是将推送消息分成的种不同的消息类型。

**定义5.2**：用户偏好矩阵，表示为，含义是用户对于种不同消息类型的偏爱程度，数值越大表示偏爱程度越高。

**定义5.3**：物品特征矩阵，表示为，含义是对于种不同消息类型，物品所符合的程度，数值越大表示符合程度越高。

**定义5.4**：分数矩阵，表示为，含义是通过评分模型得出的用户对于推送内容的具体评分，表示第个用户对第个推送内容的评分。

**定义5.5：**分数预测矩阵，表示为，含义算法预测的不同用户对于不同推送消息的预测分数，数值越大表示用户越喜欢该物品。

则用户偏好矩阵和物品特征矩阵进行点积即可获得分数预测矩阵，即有公式（5-3）。

 (5-3)

其中为的矩阵，是的矩阵，是的矩阵。

接下来我们定义代价函数，所谓代价函数是指我们的预测值跟实际值之间的差距，可以得出代价函数如公式（5-4）所示。

 (5-4)

其中表示代价函数，矩阵表示用户是否有对某个推送消息尽心评分，值为1表示用户已经对该推送消息评分，值为0表示用户尚未对该推送消息评分。只有用户已经评分的推送消息项才能参与代价函数的计算。公式最后两项是正则化项，用来防止特征数值过大或者过小。

算法的目的便是最小化代价函数，最小化一个函数一般有两种算法，梯度下降算法和最小二乘法，本文选择使用梯度下降算法最小化代价函数。假定我们有一个已知的用户偏好矩阵，则可以根据梯度下降算法，得出代价函数最小时的物品特征合矩阵。同样的，假定我们有一个已知的物品特征矩阵，则可以根据梯度下降算法，得出代价函数最小时的用户偏好矩阵。我们只需对用户偏好矩阵物品特征矩阵进行随机初始化，然后使用梯度下降算法最小化代价函数，并同时求出和即可。梯度下降算法的公式如公式（5-5）所示。

 (5-5)

使用梯度下降算法，求出用户偏好矩阵和物品推荐矩阵后，便可以使用公式计算出分数预测矩阵。将预测评分较高的推送消息推送给指定的用户。

而对于新加入的用户，由于系统中没有用户的任何评分数据，直接使用以上算法得出的结果没有任何的参考价值。这是典型的冷启动问题，是推荐算法经常会遇到的问题，本文使用均值归一化解决冷启动问题，

**定义5.5：**均值矩阵，表示为，含义是对于推送消息所有已评分用户的评分的平均值。

则在使用作为公式中的分数矩阵，最终使用梯度下降算法得出的的分数预测矩阵使后，使用作为最终的结果。这样做的实际结果是，对于完全没有任何数据的新用户，最终会将平均评分最高的消息推送给该用户。这是较为合理的一种结果。

## 5.6　结果展示模块

本系统使用前端Web页面，对数据分析中心最终产生的结果进行图表展示。结果展示模块的前端页面使用AJAX的GET方法获取数据分析模块提供的数据分析结果的接口，数据以JSON的格式传输，并最终使用Echarts[42]插件进行图表的渲染。

展示模块，分为两大部分，一部分是对即时通讯数据的展示，主要包括：用户的相关统计和消息的相关统计。其中用户的相关统计又包括每日新增用户折线图和用户日使用次数折线图；消息的相关统计包近消息的产生量折线图和消息产生时间点占比饼状图。

另一部分是对推送数据的展示，主要包括：用户设备的相关统计，和推送信息的相关统计。其中用户设备的相关统计有包括APP版本分布饼状图，操作系统版本分布饼状图和设备生产商分布饼状图；推送信息的相关统计又分为了推送系统整体的统计和具体推送消息的统计，主要有接收占比饼状图，打开/取消占比饼状图，打开推送消息时间点占比饼状图，取消推送消息时间点占比饼状图。

## 5.7　本章小结

本章首先介绍了数据分析中心的整体架构，将其分为了数据接收和对外提供模块，数据存储模块，数据分析模块，推荐算法模块和结果展示模块。在数据接收和对外展示模块介绍了接收和对外提供的接口路由；针对数据分析模块，详细介绍了了MongoDB数据库，以及使用它的原因；在数据分析模块，详细介绍了MongoDB的几个非常使用的聚合框架，并举例展示了聚合框架的使用方法；在推荐算法模块，实现了一个针对推送的个性化推荐系统，该系统使用了基于线性拟合的协同过滤算法。最后，在结果展示模块，则针对如何在前端web页面对数据进行简洁并美观的图表展示进行了描述。

# 第六章　案例分析

## 6.1　测试案例准备

本章选择了一个具有代表性的新闻类客户端进行案例分析，较为全面的介绍本平台的整个使用流程和最终效果。新闻类客户端越来越重视新闻发布者和读者之间的沟通，“今日头条”等新闻客户端都具有即时通讯的功能；另一方面，给用户推送新闻是新闻客户端的重要功能之一。本平台提供的即时通讯和推送SDK非常适合类似的新闻客户端快速集成即时通讯和推送功能，同时收集相关的用户数据，为应用的运营商和开发者提供运营和优化的依据。

### 6.1.1　应用介绍

在各式各样的移动应用中，新闻类APP是非常有代表性的一类应用，新闻的移动化阅读，在全世界范围内都成为主流，越来越多的用户选择在手机设备上阅读新闻。而对于一款新闻类APP，即时通讯和推送功能往往又是不可或缺的。本案例分析使用OuNews新闻客户端，它是一个完全开源的新闻类应用，具有浏览新闻，图片，视频，评论，点赞等功能。但是并不具有推送和即时通讯的功能，使用本文提供的SDK可以使其快速具有即时通讯和推送的功能，并且对相关数据进行统计、分析和展示。OuNews新闻客户端的主页面如图6-1所示。



图6-1 新闻客户端主页面图

### 6.1.2　服务端部署环境介绍

本平台的服务端部署在阿里云服务器上，服务器采用Linux系统，具体的环境配置如表6-1所示。

表6-1 服务器环境配置表

|  |  |
| --- | --- |
| 项目 | 配置 |
| 内存 | 4GB |
| CPU | Intel Xecon E3060 CPU 主频2.4GHz 两核 |
| 网络带宽 | 4Mbps |
| 操作系统 | Ubuntu 14.04 64位 |
| JDK | 1.7 |
| Web服务器 | Apache Tomcat 7 |
| 数据库 | MongoDB 3.2.3 |
| XMPP服务器 | Openfire 4.0.4 |
| MQTT服务器 | Apollo 1.7.1 |

### 6.1.3　应用构建流程

（1）注册账号并新建应用

用户首先需要在平台上注册自己的账号，并新建即时通讯和推送应用，新建应用成功和，平台会为该应用生成一个appid，每个应用拥有唯一的appid。后续开发者将SDK集成到应用中时，会使用到appid。

1. 下载并集成SDK

提供两种方式供开发者下载和集成SDK，一种是直接从网站上下载相应的aar库，开发者只需将相应aar导入并进行依赖即可。另一种是听过gradle进行依赖，本工程上传到了公共的maven仓库，开发者可以直接通过gradle进行更为方便和快速的集成。对于即时通讯的集成，提供了两个库分别是imcore，imui前者是即时通讯功能的核心库，包含了即时通讯的核心逻辑接口；后者包含了即时通讯的相关页面。开发者可以根据自己的需要选择集成。

1. 配置并启动SDK

在项目中集成相关SDK后，还需要进行简单的配置并添加启动SDK的代码。Android工程中的Application类中的onCreate方法是工程启动时必然会调用的生命周期方法，建议在这个方法中，进行SDK的配置和启动，对于即时通讯SDK，需要添加的关键代码如图6-2所示，将在平台上获取的相应appId作为参数传入IMCore类的init方法中即可完成即时通讯SDK的启动。



图6-2 即时通讯SDK启动关键代码

对于推送SDK需添加的关键代码如图6-3所示，通过PushClient的getInstance方法获取推送SDK的单例对象，再将从平台获取的的响应appId作为参数传入单例对象的start方法中即可完成推送SDK的启动。



图6-3 推送SDK启动关键代码

至此，应用便具有了即时通讯或者推送以及收集相关用户数据的功能。

## 6.2　即时通讯SDK的集成

集成即时通讯SDK后即可使用相关功能，可以选择使用SDK提供的界面，或者只使用相关接口。以下以使用界面为例，对几个较为重要的页面进行简单的介绍。



图6-4 会话页面图

即时通讯的会话页面如图6-4所示，该页面用来显示这个用户所有的历史会话信息，包括单聊和群聊，会话列表的每一项都回显示会话联系人的头像，昵称，最近的聊天记录和时间。用户点击会话即可进入具体的聊天页面。点击右上角的选项按钮，弹出菜单选项，可以进行添加好友和新建群聊。

即时通讯的单聊页面和群聊页面如图6-5所示，两个页面的功能相似，用户可以在这两个页面进行文本，图片，语音和文件的发送。点击图片消息会显示图片大图；点击文件消息会下载文件并使用相关应用打开；点击语音消息会播放语音。

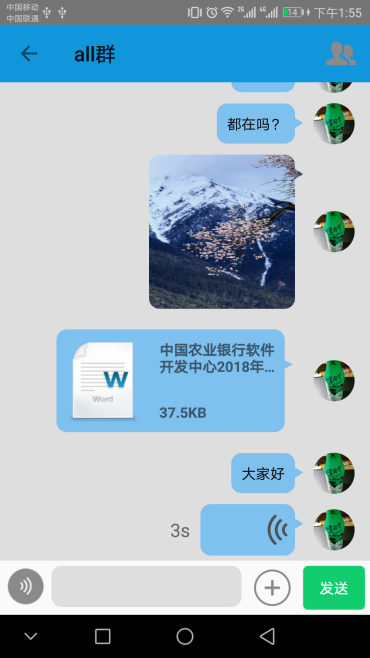


图6-5 聊天页面图

即时通讯的联系人管理页面如图6-6（左）所示，用户的所有联系人都显示在该页面中，点击某个具体的联系人，则显示联系人详情页面。联系人详情页面如图6-6（右），显示了该联系人的账号，昵称，性别，年龄和地区。另外，点击新的朋友会进入历史好友申请页面，该页面展示了所有的历史好友申请的记录。点击群聊则进入群聊列表页面，该页面展示了该用户所在的全部群聊。



图6-6 联系人管理和联系人详情页面图

即时通讯的“我”页面如图6-7（左）所示，用户可以在该页面注销登录状态，或者点击编辑名片进入个人信息的修改页面。个人信息的修改页面如图6-7（右）所示，用户可以在该页面进行个人信息的修改，包括修改头像，昵称，性别，年龄和地区。



图6-7 “我”和修改个人信息页面图

## 6.3　推送SDK的集成

移动端应用继承了推送SDK后，即可使用推送功能向移动端应用推送消息，推送端使用Web页面进行开发，以下以推送一条消息为例，对推送消息的流程做简单的介绍。

推送者首先需要登录平台，打开推送应用所对应的推送页面。推送者需要填写推送消息的基本内容，包括推送消息的标题，内容，小图标和大图标。图表需要填写工程内的资源名，相关设置界面如图6-8所示。

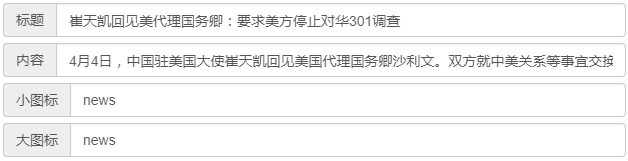


图6-8 推送页面基本内容设置图

如图所示，接着推送用户需要选择用户接收到推送消息时的提醒选项，包括是否开启铃声，是否开启呼吸灯和是否开启震动，相关设置界面如图6-9所示。



图6-9 推送页面提醒选项设置图

用户需要填写点击推送消息后的后续目标页面的信息，包括指定页面的全类名，以及相应参数的key值和value值。如果用户想要添加多个参数，需要点击添加参数，填写对应的key值和value值即可。相关设置界面如图6-10所示。



图6-10 推送页面后续动作设置图

用户需要选择推送的目标用户。若选择全部用户，则该应用的全部用户都会接收到推送消息；若选择部分用户，则需要推送者进一步填写推送条件筛选用户，提供了根据系统版本，APP版本和手机厂商对目标用户进行筛选。相关设置界面如图6-11所示。



图6-11 推送页面目标用户设置图

如图所示，用户需要选择推送时间，平台提供了三种模式，分别是立即推送，定时推送和重复推送。最后用户确认填写好全部信息后，点击提交按钮，即可完成推送端推送的全部流程。相关设置页面如图6-12所示。



图6-12 推送页面推送时间设置图

移动端设备即可以快速接收到推送端所发送的推送信息。点击通知栏后跳转到相应的新闻页面。客户端接收到推送消息的页面如图6-13所示。



图6-13 客户端接收到推送消息图

## 6.4　数据中心统计分析

数据分析中心向开发者和运营商展示了即时通讯和推送SDK所收集并通过数据分析中心分析后的用户相关数据，以下分别介绍即时通讯数据统计的结果，和推送数据统计的结果。

### 6.4.1　即时通讯数据分析

即时通讯应用最近七天内新增加的用户数量如图6-14所示，可以看出该应用在大趋势上的新增用户是在增加的，4月5日的新增用户最多，达到了30人。该图代表该应用的发展势头良好。

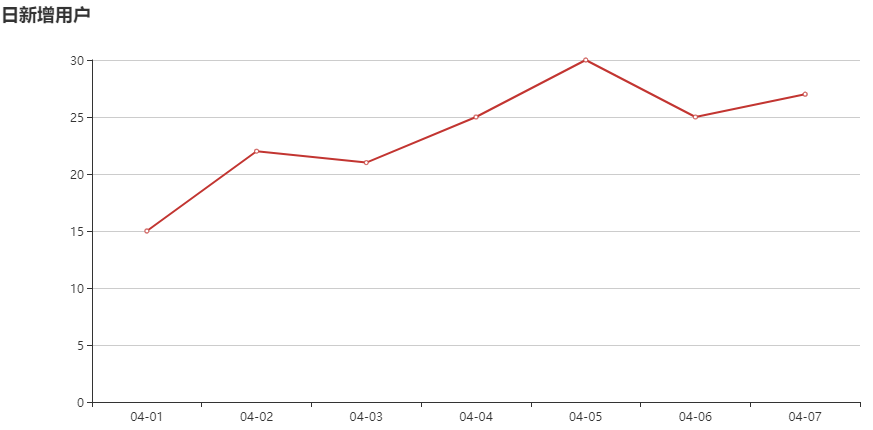


图6-14 即时通讯日新增用户折线图

应用最近七天内用户使用的次数如图6-15所示，即打开应用的次数，可以看出最近七天该应用的日使用次数有较为明显的增长，在最近一天到达了40次左右。代表该应用的用户量尤其是活跃用户量在持续增长。

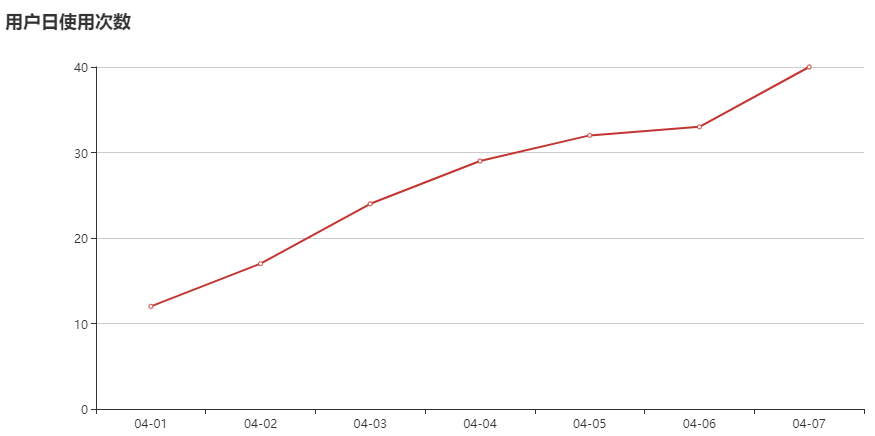


图6-15 即时通讯用户日使用次数折线图

应用最近七天内的日消息产生数量如图6-16所示，可以看出应用最近七天内产生的消息数量稳步升高，代表了应用内的即时通讯模块，尤其是发送消息模块被越来越多的用户所使用。

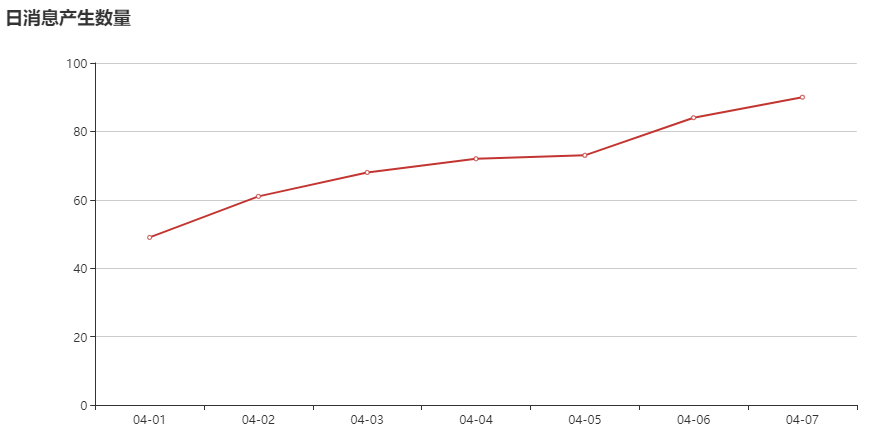


图6-16 即时通讯日消息产生量折线图

用户发送的消息的时间点的占比如图6-17所示，可以看出用户在19点，20点，21点的占比是比较高的，最高的20点有50条消息，占了总比中的12.26%。这代表了用户使用即时通讯模块最集中的时间段是19点到21点。

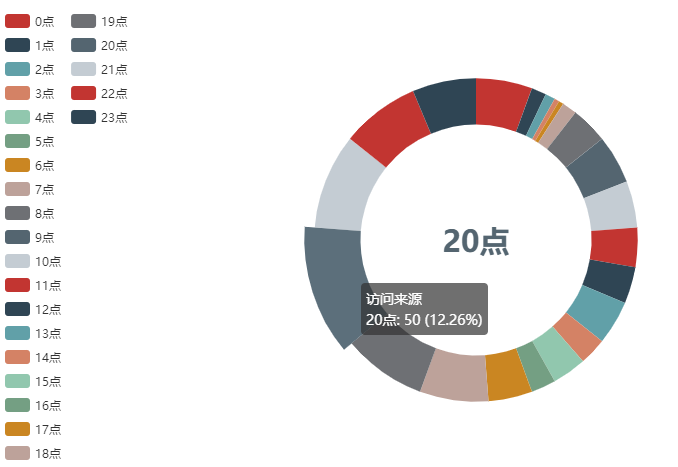


图6-17 即时通讯消息时间分布图

用户发送的消息的类型占比如图6-18所示，可以看出用户发送文本类型的消息总数最多，有121条，所占比重为56.54%，其次是语音类消息和图片消息，文件消息所占比重最小。代表用户最常使用的消息类型依然是文本消息。

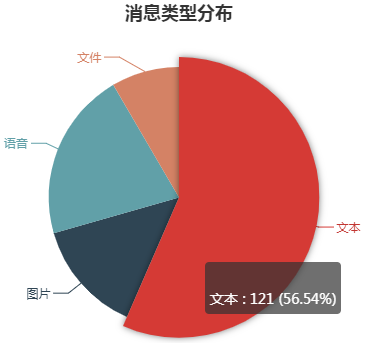


图6-18 即时通讯消息类型分布图

### 6.4.2　推送数据分析

推送数据的概览如图6-19所示，列出了用户总数，消息发送总数，消息总条数，消息接收总数，消息打开总数以及消息取消总数，供开发者和运营商对推送情况有一个整体的认识。



图6-19 推送数据概览图

推送SDK的推送消息列表如图6-20所示，即该应用推送过的所有历史推送消息，包括每条推送消息的发送人数，接收人数，打开人数和取消人数。发送人数是指该条应用应该发送给多少个用户；接收人数是指该条推送消息有多少人真正接收到了；打开人数是指接收到推送消息的用户有多少人打开了该推送消息；取消人数是指接收到推送消息的用户有多少人取消了该推送消息。



图6-20 推送数据列表图

集成了推送SDK的应用，各个APP版本 的占比如图6-21所示，可以看出，1.3版本的APP用户量最大，开发者和运营商可以主要针对1.3版本进行维护和优化，同时可以看出最新版本1.3.1的用户量较少，需要进一步推广应用的最新版本。

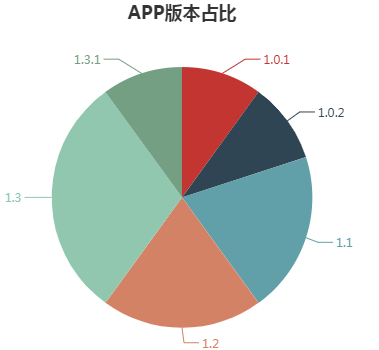


图6-21 推送客户端APP版本占比图

集成了推送SDK的应用，设备操作系统的占比如图6-22所示，可以看出Android 7.1的设备有4台，所占比重最大，开发者和运营商可以着重对应用在Android7.1上的表现和性能做深度的优化。

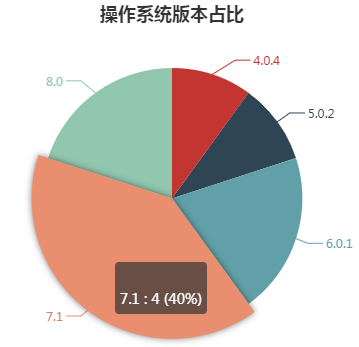


图6-22 推送客户端操作系统版本占比图

推送设备的生产商占比如图6-23所示，可以看出用户使用的所有设备中华为的设备最多有四台，占了总数的40%，运营商可以考虑针对华为设备和相应的系统进行对应用进行优化和升级。

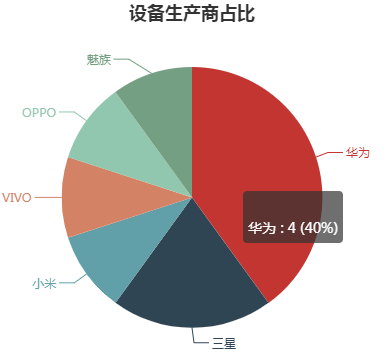


图6-23 推送客户端设备生产商占比图

推送系统发送的推送消息接收占比如图6-24所示，未接收到推送消息的原因是后台接收推送的Service被杀死，导致部分推送消息没有被设备接收到。可以看出推送SDK发送的消息的接收个数为81个，占比为81%。这代表推送服务的大部分推送消息是正常被设备接收到的。

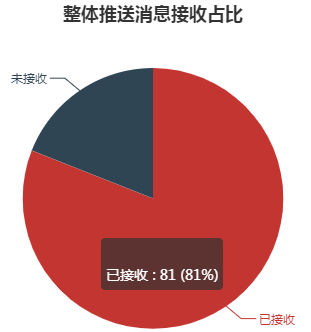


图6-24 整体推送消息接收占比图

用户接收到推送消息后的打开/取消占比如图6-25所示，可以看出用户总共打开消息有54条，占总比重的66.67%。代表半数以上的消息都被用户所打开，较大部分的推送的消息是受用户所欢迎的。

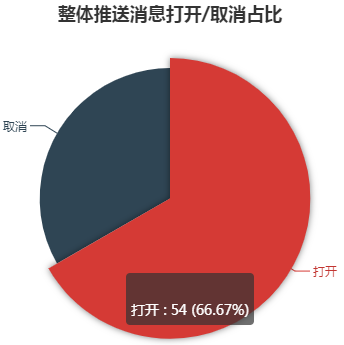


图6-25 整体推送消息打开/取消占比图

用户打开推送消息的时间段占比如图6-26所示，即所有的被打开的推送消息的时间段分布。可以看出，用户打开推送消息的时间段主要集中在20点到21点，占比为15.38%，说明这个时间段的用户打开推送消息的概率最高，推送者可以考虑将较为重要的推送消息在该时间段内进行推送，从而提高推送消息的打开占比。

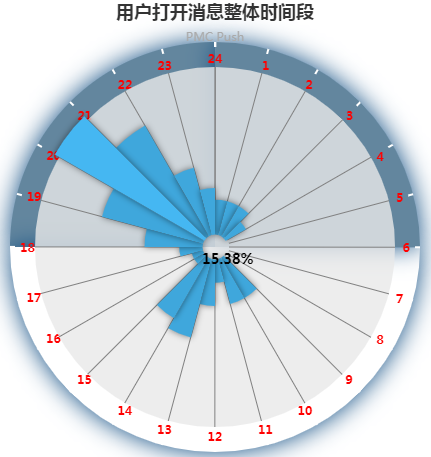


图6-26 用户打开消息整体时间段占比图

相对的，用户取消推送消息的时间段占比如图6-27所示，即所有的被取消的推送消息的时间段分布。可以看出，用户取消推送消息的时间段主要集中在9点到10点，占比为13.64%，说明这个时间段的用户取消消息的概率最高，推送者可以考虑减少在该时间段进行推送。

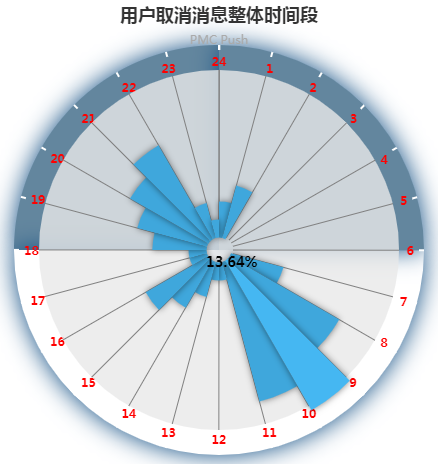


图6-27 用户取消消息整体时间段占比图

以上都是针对推送系统的整体情况得出的图表，可以从整体上了解推送系统的各个方面的运行情况。而另一方面针对某一条具体推送消息的统计也非常重要，可以非常直观的了解某条推送消息的具体情况。

某一条推送消息的打开取消占比如图6-28所示，可以看出，该条推送消息只有5人打开，所占比例为55.56%，并不是特别高，说明该条推送内容可能由于种种原因，并不受用户的喜欢，可以针对这条推送消息根据实际情况进一步分析具体的原因。

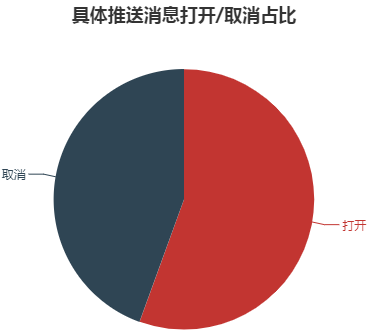


图6-28 具体推送消息打开/取消占比图

某一条具体的推送消息的用户打开时间段分布如图6-29所示，可以看出，该条推送消息绝大部分都是在19点到20点被打开的。可以进一步结合推送消息的推送时间，来判断该条消息是否得到了用户很快的回应。

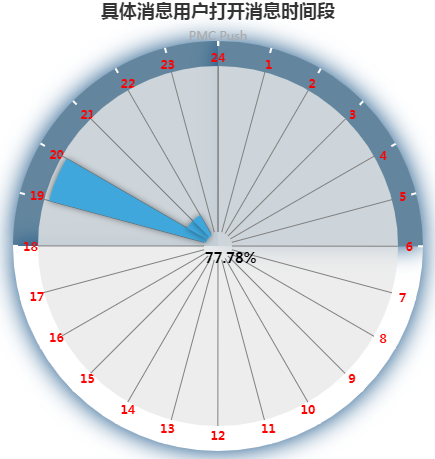


图6-29 具体消息用户打开消息时间段占比图

### 6.4.3　协同过滤推荐模型

本案例中有10个用户，向每个用户发送了10条消息，即总共100条消息。具体的消息标题和对应的消息类别如表6-2所示。

表6-2 推送消息内容表

|  |  |  |
| --- | --- | --- |
| id | 标题 | 类别 |
| 1 | 带伤上阵！曝小丁已报名12人名单 基本确定出战 | 体育 |
| 2 | 消除死亡之组！曝亚洲杯分档调整 国足避开乌兹卡塔尔 | 体育 |
| 3 | WTA蒙特雷赛穆古拉扎逆转4号种子 | 体育 |
| 4 | 小丁女友高调示爱秀才华 喊话辽宁为山东加油 | 体育/娱乐 |
| 5 | 财政部：研究制定政府非税收入管理条例 | 财经 |
| 6 | 阿里巴巴联合创始人蔡崇信收购篮网队49%股份 | 财经/体育 |
| 7 | 恒指涨1.22%报30207点 吉利汽车领涨蓝筹 | 财经 |
| 8 | 证监会处罚黄有龙、赵薇禁入证券市场5年 | 娱乐/财经 |
| 9 | 翁虹、杨恭如、郭可盈、欧阳震华重聚首，TVB回忆杀 | 娱乐 |
| 10 | 何洁回归家庭母爱满溢，素颜出镜接孩子放学 | 娱乐 |

使用5.5.1节中的评分模型，对100条消息进行评分，评分结果保留一位小数，则最终的评分结果如表6-3所示，行表示消息，列表示用户，则第行，第列表示第个用户对第条消息的评分，横线表示该用户由于网络等问题，没有成功对推送消息进行评分。

表6-3 评分结果表

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| id | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 1 | 9.2 | 2.5 | 7.6 | 1.0 | 8.5 | 4.4 | 0.6 | 2.8 | 3.9 | 8.1 | 8.3 |
| 2 | - | 3.9 | 9.1 | 3.6 | 6.7 | 3.6 | 1.4 | 1.5 | 7.5 | 7.7 | 7.9 |
| 3 | 9.2 | 1.3 | 10.0 | 2.4 | 7.5 | 0.9 | 4.4 | 4.1 | 6.5 | 9.6 | - |
| 4 | 4.4 | 4.9 | - | 4.2 | 3.1 | 2.5 | 3.5 | 3.4 | - | 10.0 | - |
| 5 | 3.9 | 9.1 | 10.0 | 5.1 | 1.8 | 7.4 | 8.6 | 5.5 | 8.5 | 4.7 | - |
| 6 | 6.1 | 8.8 | 7.9 | 2.4 | - | 6.1 | 9.8 | 4.5 | 6.4 | 5.3 | - |
| 7 | 5.4 | 7.6 | 6.2 | 3.8 | 2.4 | 9.1 | 7.5 | 7.6 | 7.4 | 3.3 | 4.9 |
| 8 | 2.8 | - | 1.8 | 7.6 | 3.4 | 8.9 | 0.7 | 1.5 | 9.6 | 1.1 | - |
| 9 | 0.4 | 2.8 | 4.4 | 8.9 | 1.4 | 6.5 | 1.7 | 4.7 | 7.2 | - | 1.5 |
| 10 | 4.9 | 1.8 | - | 9.9 | 3.1 | 7.2 | 1.8 | 3.4 | 8.6 | 4.5 | - |

此时第十一个用户进入系统，该新用户最新欢体育类新闻，其次是财经类新闻，最不喜欢娱乐类新闻，并对系统中的第一条体育类消息评分为8.3分，对第二条体育类消息评分为7.9分，对第七条财经类消息评分为4.9分，对第九条娱乐类消息评分为1.5分，我们的目的是找出该用户没有接收到的推送消息里，预测评分最高的。我们使用全部十一个用户产生的评分数据，对推荐模型进行训练，求出用户偏好矩阵和物品特征矩阵。则通过算法得出最后的该用户对十条新闻的评分如表6-4所示。可以看出，模型预测的评分最高的新闻类别为体育类，其次为财经类，最后为娱乐类，符合预期的结果。

表6-4 预测用户评分表

|  |  |  |  |
| --- | --- | --- | --- |
| id | 标题 | 类别 | 预测评分 |
| 3 | WTA蒙特雷赛穆古拉扎逆转4号种子 | 体育 | 8.7 |
| 6 | 阿里巴巴联合创始人蔡崇信收购篮网队49%股份 | 财经/体育 | 6.6 |
| 4 | 小丁女友高调示爱秀才华 喊话辽宁为山东加油 | 体育/娱乐 | 5.7 |
| 5 | 财政部：研究制定政府非税收入管理条例 | 财经 | 4.4 |
| 8 | 证监会处罚黄有龙、赵薇禁入证券市场5年 | 娱乐/财经 | 3.4 |
| 10 | 何洁回归家庭母爱满溢，素颜出镜接孩子放学 | 娱乐 | 1.7 |

则将最终将评分最高的id为3的体育类新闻推送给用户，结果如图6-30所示，即可完成整个个性化推荐的流程。



图6-30 客户端接收到推荐消息图

# 总结与展望

## 总结

通过对近年来即时通讯和推送技术的研究，本文设计并实现了基于Android客户端的即时通讯和推送服务，提供了客户端SDK，任何一个Android应用都可以快速集成SDK使应用具有即时通讯或者推送功能，同时该SDK收集了许多非常有意义的数据，包括用户数据，设备数据以及用户行为数据等。另外，该平台提供了数据分析中心，对从客户端收集到的数据进行存储、统计、计算和分析，给应用的开发者和运营人员，非常直观的图表展示，为应用进一步的发展提供数据支撑，同时也针对推送提供了个性化的推荐系统。具体来说，本文的工作如下：

1. 对即时通讯和推送服务需求的研究，设计了基于Android平台的即时通讯和推送服务从客户端到后台的整体架构和模块，并对每个模块的实现技术和原理进行了阐述。
2. 设计并实现了一个基于Android平台的即时通讯SDK，该SDK一方面实现了聊天，群聊，好友管理等即时通讯的基本功能，一方面收集并上报了用户和消息等数据。任何一个Android应用都可以集成该SDK，快速拥有上述功能。
3. 设计并实现了一个基于Android的推送SDK，该SDK一方面实现了推送的基本功能，一方面收集并上报了用户设备信息。任何一个应用都可以集成该SDK，快速拥有上述功能。
4. 设计并实现了数据分析中心，可以对即时通讯和推送SDK上报的数据进行接收，存储、分析、计算和展示。
5. 针对推送服务，利用推送SDK上报的数据，构建了自动对推送消息进行评分的模型，并使用基于线性拟合的协同过滤算法，实现了一个可以对用户进行个性化推荐的推荐模型和系统。

## 展望

本文实现了一个基于Android平台的即时通讯和推送服务，并对用户使用过程中产生的各种数据进行了收集、分析和展示。结合相关技术目前的发展和趋势，本平台还可以从以下方面进行改进：

1. 本平台的客户端SDK都基于Android系统，未来可以支持更多类型的移动端系统，例如iOS和Windows。
2. 协同过滤算法可以使用一些更加复杂的模型，提高推荐的准确率，例如基于深度学习的协同过滤算法等。

# 参考文献

1. 中国互联信息中心. 第41次中国互联网络发展状况统计报告[EB/OL]. http://www.cnnic.net.cn/hlwfzyj/hlwxzbg/hlwtjbg/201803/P020180305409870339136.pdf, 2018.
2. 刘凯, 王伟军, 黄英辉,等. 个性化推荐系统理论探索:从系统向用户为中心的演进[J]. 情报理论与实践, 2016, 39(3):52-56.
3. 邓胜利, 张敏. 用户体验——信息服务研究的新视角[J]. 图书与情报, 2008(4):18-23.
4. NETMARKETSHARE. Mobile/Tablet Operating System Market Share[EB/OL]. http://www.netmarketshare.com/operating-system-market-share.aspx, 2018.
5. Yulianto B, Heriyanni E, Dewi L C, et al. Architecture and Implementation of Instant Messaging in Educational Institution [J]. Procedia Computer Science, 2015,59:5-13.
6. Lübke R, Schuster D, Schill A. A Framework for the Development of Mobile Social Software on Android[C]. International Conference on Mobile Computing, Applications, and Services. Springer, Berlin, Heidelberg, 2011:207-225.
7. Gudla S K, Bose J, Sunkara S, et al. A unified Push Notifications service for mobile devices[C]. IEEE International Conference on Electronics, Computing and Communication Technologies. IEEE, 2016:1-6.
8. Brustel J, Preuss T. A Universal Push Service for Mobile Devices[C]. Sixth International Conference on Complex, Intelligent, and Software Intensive Systems. IEEE Computer Society, 2012:40-45.
9. 李鲲鹏. 基于Android的即时通讯平台研究与实现[D]. 华南理工大学,2013.
10. 徐鹏伟. 基于XMPP协议的iOS即时通讯应用研究与实现[D]. 华南理工大学,2015.
11. 周艺. 基于XMPP协议的Android终端消息推送平台的设计与实现[D]. 重庆理工大学,2017.
12. 黄明恩. 基于Android平台的云推送服务的设计与实现[D]. 北京交通大学,2015.
13. 郑峰. 基于MQTT的云推送平台的研究及应用[D]. 杭州电子科技大学,2016.
14. 唐苏. 基于通讯云技术的Android即时通讯系统的研究与实现[D].西南交通大学,2016.
15. 王忠民,曹洪江,范琳.基于XMPP通信协议的即时通讯研究[J].信息技术,2016(09):45-46.
16. Saintandre P. Streaming XML with Jabber/XMPP[J]. IEEE Internet Computing, 2005, 9(5):82-89.
17. 张哲. 基于扩展XMPP协议的即时通信系统的设计与实现[D]. 中国海洋大学, 2015.
18. Kirsche M, Klauck R. Unify to bridge gaps: Bringing XMPP into the Internet of Things[C]. IEEE International Conference on Pervasive Computing and Communications Workshops. IEEE, 2012:455-458.
19. 陈理兵. 基于Openfire的即时通信服务端系统设计与实现[D]. 华中科技大学, 2015.
20. Ren H, Yue M A, Yang H B, et al. Message Pushing Server Based on the MQTT Protocal[J]. Computer Systems & Applications, 2014.
21. 谢晓琦. 信息推送技术综述[J]. 科技创新与应用, 2016(23):45-45.
22. Hong-Jun N I. Research and Implementation of Message Push Based on Android[J]. Research & Exploration in Laboratory, 2014.
23. IBM. MQTT协议报文参考文档[EB/OL]. http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html
24. Ojeda-Guerra C N. A Simple Software Development Methodology Based on MVP for Android Applications in a Classroom Context[C]. IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing. IEEE, 2015:1429-1434.
25. 曾露. MVP模式在Android中的应用研究[J]. 软件, 2016, 37(6):75-78.
26. 李静. Android中Binder机制研究与应用[J]. 工业控制计算机, 2012, 25(4):66-67.
27. Hu A M. Research and Analysis Process of Android Communication Platform in AIDL Model[J]. Advanced Materials Research, 2014, 989-994:4893-4896.
28. Rhinow F, Veloso P P, Puyelo C, et al. P2P live video streaming in WebRTC[C]. Computer Applications and Information Systems. IEEE, 2014:1-6.
29. 林鸿, 王松, 杨鑫,等. 基于WebRTC技术的应用及平台技术开发与设计[J]. 电信科学, 2013, 29(9):20-25.
30. 朱雅琴.NAT技术在局域网中的应用[J].电子技术与软件工程,2017(09):9-10.
31. 孙光浩, 刘丹青, 李梦云. 个性化推荐算法综述[J]. 软件, 2017, 38(7):70-78.
32. 刘青文. 基于协同过滤的推荐算法研究[D].中国科学技术大学,2013.
33. Guo Y, Huang M, Lou T. A Collaborative Filtering Algorithm of Selecting Neighbors Based on User Profiles and Target Item[C]. Web Information System and Application Conference. IEEE, 2016:9-14.
34. Latha R, Nadarajan R. User Relevance for Item-Based Collaborative Filtering[J]. 2017, 8104(3):337-347.
35. Ge X, Liu J, Qi Q, et al. A new prediction approach based on linear regression for collaborative filtering[C]. Eighth International Conference on Fuzzy Systems and Knowledge Discovery. IEEE, 2011:2586-2590.
36. Tang W S, Tian L H, Cao W. A reaserch of Base64 coder realization and application[J]. Journal of Changchun University, 2006.
37. Google. Android开发者文档[EB/OL].https://developer.android.com/reference/android/app/Notification.html,2018.
38. 杨旭. J2EE企业级开发:Struts2+Spring+Hibernate整合技术[M]. 清华大学出版社, 2016.
39. Chodorow K, Dirolf M. MongoDB: The Definitive Guide[M]. O'Reilly Media, Inc. 2013.
40. Schram A, Anderson K M. MySQL to NoSQL:data modeling challenges in supporting scalability[C]. ACM Conference on Systems, Programming, Languages and Applications: Software for Humanity. ACM, 2012:191-202.
41. Patel C, Hasan M, Tanawala B. Benchmark of Map-Reduce Framework in MongoDB[C]. International Conference on Electrical, Electronics, Signals, Communication and Optimization. IEEE, 2015:1-4.
42. 百度echarts开发团队.echarts官网[EB/OL]. http://echarts.baidu.com/,2018.

# 攻读硕士学位期间取得的研究成果

一、已发表（包括已接受待发表）的论文，以及已投稿、或已成文打算投稿、或拟成文投稿的论文情况**（只填写与学位论文内容相关的部分）：**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **作者（全体作者，按顺序排列）** | **题 目** | **发表或投稿刊物名称、级别** | **发表的卷期、年月、页码** | **相当于学位论文的哪一部分（章、节）** | **被索引收录情况** |
| 1 | 陆璐  贾建业 | 一种在线播放HDFS中视频资源的新方法 | 计算机应用研究 | 2018年第Z1期 | 第一章 |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

注：在“发表的卷期、年月、页码”栏：

1如果论文已发表，请填写发表的卷期、年月、页码；

2如果论文已被接受，填写将要发表的卷期、年月；

3以上都不是，请据实填写“已投稿”，“拟投稿”。

不够请另加页。

1. 与学位内容相关的其它成果（包括专利、著作、获奖项目等）

发明专利：一种基于MQTT协议的移动终端试题分发与作答的方法（已受理，申请号：201710051907.4）

软件著作权：基于Android的粤教云课堂互动系统（已授权：登记号：2017SR616158）

# 致　谢

三年的研究生生涯马上就要结束，在这三年的宝贵时间里，我学习了很多计算机软件和数学方面的理论知识，同时也参与了许多实际的科研和实践项目，得到了技术上的锻炼，也开拓了自己的眼界。在此期间，我得到了导师，父母，同学的帮助和支持，在即将毕业之际，向他们表示诚挚的感谢。

首先感谢我的导师陆璐教授对我的栽培和指导，在研究生三年的时间里，陆老师在学习和生活方面都给了我莫大的帮助，在毕业论文的撰写方面也给了我许多宝贵的指导意见。陆老师在学术，科研，等方面都兢兢业业，一丝不苟，是我今后学习的榜样。

另外还要感谢我的父母，他们时常教育我要在学习上踏实认真，在为人处世上谦和善良，在这三年的时间里，他们在生活上给了我很多的关心和建议，给了我很大的帮助。

最后我要感谢我的师兄师姐们的耐心指点和帮助，感谢万腾师兄，陈勇师兄，何芦薇师姐，黎灿灿师姐，他们在科研和专业知识方面都对我有很大的帮助。感谢同届的涂逸欣同学，成斌，廖飞同学和邓德荣同学，三年的时间里和他们一起学习进步，受益匪浅。还要感谢关山旭师弟，黄小川师弟，贾志豪师弟，冯秋燕师妹和孙芳芳师妹，他们分担了我许多工作，使我能把更多的时间和精力投入到论文的撰写中。