

分类号: TP37  
UDC: 004.6

学校代码: 83801  
密级: 公开

中国铁道科学研究院  
专业学位硕士研究生  
学位论文

12306 候补购票服务系统的研究与实现  
Research and Implementation of  
12306 Line Up Ticketing System

作者姓名: 李雯

工程领域: 计算机技术

论文类型: 应用研发

研究方向: 铁路客票系统维护与管理技术

导师姓名: 朱建生

学位授予单位: 中国铁道科学研究院

论文提交日期: 2019 年 6 月

分类号: TP37  
UDC: 004.6

学校代码: 83801  
密级: 公开

中国铁道科学研究院  
专业学位硕士研究生  
学位论文

12306 候补购票服务系统的研究与实现  
Research and Implementation of  
12306 Line Up Ticketing System

作者姓名 李雯

导师姓名 朱建生

申请学位 硕士

培养单位 电子所

工程领域 计算机技术

研究方向 系统维护与管理

答辩委员会主席 王富章

评阅人 王富章

马敏书

王明培

论文提交日期: 2019 年 6 月 (四号宋体、居中)



## 原创性声明

本人郑重声明： 所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。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。本声明的法律结果由本人承担。

论文作者签名：李雯

日期：2019年6月12日

## 关于学位论文使用授权说明

作者了解中国铁道科学研究院研究生部有关保留、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中国铁道科学研究院。中国铁道科学研究院有权保留并向国家有关部门或机构送交论文的纸质文档和电子文档，允许学位论文被查阅和借阅，可以公布学位论文的全部或部分内容，可以采用复印、缩印或其他手段保留、汇编学位论文。保密的学位论文在解密后遵守此规定。

☒公开      ☐保密（保密期：\_\_\_\_年\_\_\_\_月至\_\_\_\_年\_\_\_\_月）

学位论文作者签名：李雯      导师签名：朱世

签字日期：2019.6.12      签字日期：2019.6.13

## 致 谢

值此论文完成之际，衷心感谢我的导师朱建生研究员，导师知识渊博、学术态度严谨、学术造诣深厚、在整个研究生的学习生涯中对我的悉心指导使我受益匪浅，在论文完成之际，要向朱老师致以最真挚的谢意。

感谢我的领导单杏花主任、杨立鹏主任在我攻读硕士学位期间给予我的帮助、工作中的信任，在此表示衷心的感谢！

电子所 ASP 技术部是一个非常优秀的团队，很荣幸能够在这里工作，为客票事业的发展添砖加瓦，感谢梅巧玲、张龙、王拓等同事对我日常工作的支持和帮助。

感谢我的同学们在研究生学习生活里对我生活、工作、学习上的关心，感谢我的同门王拓硕士在论文完成过程中对我的帮助，感谢研究生院的所有老师。

感谢我的家人，你们在背后的默默支持和鼓励是我前进的动力！

感谢在百忙之中评阅我硕士论文和参加答辩的各位专家和学者，不足之处恳请各位专家学者批评指正，谢谢你们提出的宝贵意见以指导论文的完善！

最后，感谢所有曾经支持、帮助、关心、批评和鼓励过我的人们。

## 摘 要

随着 12306 客票系统的推广与应用，互联网购票已经成为旅客购票的最主要方式之一。根据对用户体验和客票销售情况的调查，发现售票规则存在需要完善的空间，例如：（1）在旅客想要购买的车次无票之后，旅客需要占用较长时间，时刻关注余票变化情况，在车票存量动态变化频繁的情况下，将会导致客流流失；（2）另外，其他旅客退票改签的原票中，有一定比例没有被再次销售。以上两点对旅客和铁路部门都会造成损失，同时也给第三方刷票程序以可乘之机。为了提升旅客从 12306 购票的便利性，从根源处减少刷票软件的高频刷票行为，打击第三方刷票软件市场，缓解 12306 客票系统服务器压力，充分利用票额，本论文提出一种候补购票服务系统，以候补队列的方式重新定义无票购票。

本论文按照工程应用的思路，主要介绍的是候补购票服务系统的设计，首先进行需求分析，由此入手确定技术选型，主要包括开源分布式内存数据库、消息队列、微服务等技术，在候补购票服务系统的应用，并从整体系统的需求、总体架构以及功能实现等方面详解候补购票服务系统的方案。通过生产环境的试点，验证业务功能和系统指标是否满足需求。

本论文主要研究内容如下：

一、结合铁路客票系统的发展综述，确定了基于 12306 客票系统的候补购票服务系统方案的必要性，确定以开源分布式内存数据库技术、排队算法等互联网技术为基础的 12306 候补购票服务系统方案作为论文的主要研究对象。

二、针对基于 12306 客票系统的候补购票服务系统方案作需求分析，主要从业务需求、系统需求、安全需求以及网络需求方面进行分析。

三、研究分析目前主流的开源分布式内存数据库技术及排队算法，明确主流的开源分布式内存数据库各自的技术特性及其在候补购票服务系统中的适用场景。

四、在明确 12306 候补购票服务系统方案需求的基础上，进行总体系统设计，形成 12306 候补购票服务系统方案的技术框架、建立相应的应急处理机制、确定

适应长期发展的扩展机制以及保证系统整体安全的安全方案。

五、将建立的候补购票服务系统进行应用，验证 12306 候补购票服务系统方案的实用性和可行性。

**关键词：**铁路客票系统；候补购票；内存数据库；Redis；兑现成功率

## ABSTRACT

With the development and application of the 12306 ticketing system, Internet Ticketing has become one of the most critical ways for passengers to buy tickets. During the long holidays, the difficulty of buying a ticket becomes the main contradiction between the 12306 ticketing system and the majority of passengers. As a result, the ticket-snatching behavior, triggered by the usage of third-party ticket-snatching software, brings huge operation and maintenance pressures to the web server of the 12306 ticketing system. The paper proposes a method called ‘No Ticket Line Up Purchasing’ and develops the ‘Line Up Ticketing System(LUTS)’ to reduce the number of ticket-snatching software from the source, shrink the market of the ticket-snatching software and release the unnecessary pressure of the server, which redefines the way of ticket purchase when out of tickets.

The paper elaborates the technology stack of LUTS by introducing the application of open source distributed memory database technology primarily and explains the system scheme design in detail by analyzing the overall system requirements, architectures, and functions. The application of LUTS significantly reduces the ticket-snatching phenomenon, totally eliminates the third party ticket-snatching actions, reduces the pressure of 12306 ticketing system, as well as improves the user experience, elevates the railway management efficiency and service quality.

The paper mainly studies the following aspects:

1. Determine the necessity of LUTS on the basis of the railway ticketing system. Confirm LUTS with technologies like the open source distributed memory database technology and the line-up algorithm as the main research target of the paper.
2. Research and analyze the mainstream open source distributed memory database technology and line up algorithms. Specify the application situation of these technologies.

3. Make requirements analysis based on LUTS system scheme, including system requirements analysis, security analysis, and network analysis, etc.

4. Design the system based on the requirements of LUTS. Form the emergency handling technical framework. Develop the long-term scaling mechanism and security solutions.

5. Applying the Line-Up Service and verifying the practicability and feasibility of LUTS.

**KEYWORDS:** ticket system; line up buying tickets; memory database; Redis; cashing success rate.



## 序 言

铁路客票系统发展至今已有二十多年。目前客票系统建成了线上、线下两大类销售渠道。线上以 12306 网站和 12306 手机 APP 为主；线下则包括车站窗口、自动售票机以及站外人工售票点和站外银铁通自动售票机。

在 2011 年之前，先后实现了计算机售票作业、地区中心联网售票、全路联网售票、收入清算、客票营销。在这一阶段中，国家铁路相关单位在宏观层面对中国铁路客票系统的发展动态进行了前瞻性预测，中国铁路客票系统的优化升级成为必然要求。中国铁道科学研究院电子计算技术研究所对中国铁路客票系统的技术方案进行了详细论述与实践。

在 2011 年 6 月，铁路 12306 互联网售票系统投产应用。近年来，围绕 12306 网站和手机 App 先后实现了包括排队系统、分布式内存计算、同城双活双中心、混合云、电子客票等诸多技术和功能上的优化革新。在这一阶段中，互联网售票量逐年大幅度提升，铁路客运服务主要精力逐渐转移至客票系统的优化、升级与维护当中。

尽管我国铁路客运运力逐渐提升、客票销售策略逐渐合理完善，但乘客在出行高峰期间仍然一票难求，需要时刻关注余票变化，造成较差的购票体验。由此滋生的第三方抢票软件给铁路客票系统造成了巨大的系统压力及不必要的资源浪费。同时，对于铁路部门，由于车票返回销售的时间较短，存在部分票额利用不充分的问题。

为此本论文在铁路客票系统领域中提出了一种候补购票服务系统，该系统旨在抑制当前海量的第三方软件刷票行为，从而为旅客的出行提供优质保障，进一步优化客票销售策略。系统在开发过程中依赖于中国铁道科学研究院电子计算技术研究所提供的技术支撑，该系统从属于铁路客票系统，是铁路客票系统优化的一种探索与实现，是为旅客提供更优质的出行购票体验的技术追求。

# 目 录

摘 要 .....	I
ABSTRACT.....	III
序 言 .....	V
1 引言 .....	1
1.1 研究背景.....	1
1.2 研究目的及意义.....	2
1.3 国内外研究现状.....	3
1.4 本文主要研究内容.....	4
2 候补购票服务系统需求研究 .....	7
2.1 业务需求.....	7
2.1.1 业务规则 .....	7
2.1.2 候补购票服务系统功能 .....	8
2.1.3 车票发售功能 .....	10
2.1.4 通知 .....	10
2.1.5 客服服务 .....	10
2.1.6 统计分析功能 .....	10
2.2 系统需求.....	11
2.2.1 并发处理能力 .....	11
2.2.2 数据存储能力 .....	11
2.3 网络需求.....	12
2.4 安全需求.....	12
2.4.1 系统安全性 .....	12
2.4.2 数据安全性 .....	12
2.4.3 数据保密性 .....	12
2.5 本章小结.....	13
3 候补购票服务系统相关理论及技术研究 .....	15
3.1 排队算法.....	15
3.2 拟合预测算法.....	16

3.3	内存数据库技术 .....	19
3.3.1	内存数据库 .....	20
3.3.2	Redis 内存数据库技术 .....	22
3.3.3	OpenResty web 平台 .....	23
3.3.4	消息队列技术 .....	24
3.4	技术选型 .....	26
3.4.1	数据库选型 .....	27
3.4.2	消息队列选型 .....	28
3.5	本章小结 .....	29
4	候补购票服务系统设计与实现 .....	31
4.1	总体架构设计 .....	31
4.1.1	设计原则 .....	31
4.1.2	架构设计 .....	32
4.2	候补购票服务系统功能实现 .....	34
4.2.1	系统入口及展示设计 .....	34
4.2.2	候补订单处理逻辑设计 .....	35
4.2.3	候补订金及结账对账 .....	39
4.2.4	候补需求整合处理 .....	40
4.2.5	候补订单 .....	40
4.2.6	候补余票计算 .....	44
4.2.7	候补订单兑现规则设计 .....	44
4.2.8	各渠道余票查询及售票控制 .....	45
4.2.9	其他功能 .....	46
4.3	应急处理机制 .....	46
4.4	扩展机制 .....	48
4.5	安全方案 .....	48
4.6	本章小结 .....	49
5	候补购票服务系统应用 .....	51
5.1	候补购票服务系统模拟测试 .....	51
5.2	候补购票服务系统应用 .....	53
5.3	本章小结 .....	54

6 总结与展望 .....	57
6.1 本文研究总结.....	57
6.2 展望.....	57
参考文献 .....	59
作者简历及攻读学位期间取得的科研成果 .....	63
学位论文数据集 .....	65

# 1 引言

## 1.1 研究背景

当今中国铁路已经进入高速时代，为了向高速铁路提供更优质、稳定、安全的服务，铁路软件技术不断发展。铁路客票系统发展到目前为止主要分为两个阶段，分水节点为铁路客票互联网系统的投产与运行。

在第一阶段中，国家铁路相关单位在宏观层面对中国铁路客票系统的发展动态进行了前瞻性预测，中国铁路客票系统的优化升级成为必然要求。相关单位据此加大了对中国铁路客票系统的发展目标研究投入，并对中国铁路客票系统的技术方案进行了详细论述与实践<sup>[1]</sup>。在实践过程中，相关单位应用时兴的机器学习算法实现了多个铁路客票系统研究任务方案，特别是攻克了中国铁路客票系统的中转路径最佳优选与优化这一难题。中国铁路客票的发售系统以及预订系统的优化与升级极大地提升了售票的服务质量以及旅客的购票体验，避免了个人选择客票的盲目性<sup>[2]</sup>。与此同时，传统 BP 神经网络模型的应用与改进，双层次正交神经网络模型以及数据挖掘技术在铁路客运量预测优化的应用，较好地解决了铁路客票数据及行车安全数据的偏类数据集问题<sup>[3,4]</sup>。

在第二阶段中，由于铁路客票互联网系统的投产与应用，以及互联网售票的逐年大幅度提升，国家铁路相关单位的大部分精力逐渐转移至客票系统的优化、升级与维护当中。2012 年基于互联网的新一代铁路客票系统总体技术方案被提出，方案对于中国铁路客票系统在现代化潮流中应该形成的铁路售票方式进行了确定性描述，定义了未来客运服务的转型方向，即把中国铁路客票系统一贯的以内部服务型的主要运营方式转向为以旅客为中心的外部服务型，并对新型服务运维方式的工作特点以及新的总体需求做出总结<sup>[5,6]</sup>。同年，铁科院电子所也进行了多方面相关任务研究主要涉及客票预售期和席位动态调整<sup>[7,8]</sup>。同时，为了实现互联网售票系统的高工作性能、实名制系统运行以及手机购票方式的扩展，2013 年对消息队列的分布式数据库技术以及铁路客票系统的缓存机制进行了极大优化与改善

[9,10]，并且于 2013 年底确定了手机售票系统结构与功能<sup>[11]</sup>，选用面向分布式计算数据库提高实名制系统读写效率和横向扩展能力<sup>[12]</sup>。至此，12306 互联网售票系统的第一轮架构优化基本完成，由于同阶段云计算技术的爆发式发展以及客运服务的比重增加，2014 年提出了基于云计算技术的第二轮架构优化方案<sup>[13]</sup>，主要通过大数据平台架构及技术优化信息管理<sup>[14]</sup>、优化票额预分<sup>[15,16]</sup>以及完善售票高峰期性能测试<sup>[17]</sup>，通过机器学习方法实现客流精准预测<sup>[18,19]</sup>以及掌握基本铁路运行的数据资源，并对与铁路管理相关的参数如收入率等进行准确分析<sup>[20]</sup>。

中国铁路客票系统在近 20 年发生的近乎跨越式的变革极大的方便了人民的日常生活。但是在春运、暑运、节假日等售票高峰期间，旅客及代购网站在无票后持续刷票的行为对客票系统服务造成了巨大的访问压力，消耗了大量的后台资源。第三方“抢票软件”主要通过计算机编写机器软件利用收集好的旅客身份信息以及购票关键信息，以固定的频率，在避开中国铁路客票风控系统的限制基础上对客票系统进行一次车票查询的访问请求。“抢票插件”不仅收费高昂，而且严重影响 12306 APP 及网站客户体验，损害了人民的利益。因此，遏制第三方“抢票软件”刷票行为，改善旅客购票体验是本论文主要解决的难题。

## 1.2 国内外研究现状

### （1）国外研究现状

目前国际上，当列车车票售罄的情况下，售票模式较为单一。印度铁路当前实行了无票情况下的 waiting list 购票服务，即当列车所有车票售罄后，乘客继续购票则会发放 waiting list，在该列车开通前 2 小时会通知乘客该 waiting list 是否兑现成功从而使得乘客具有上车资格。欧美发达国家由于国情的不同，铁路客运需求并不旺盛，在无票情况下，主要还是通过不断查询余票的方式来购票，并没有滋生出对社会而言为负面影响的第三方抢票软件机构。针对无票售票模式，可以借鉴在出行需求方面较为旺盛的航空售票模式，美联航空在 20 世纪九十年代在无票情况下提出了与印度铁路相似的无票售票模式，旅客可在已售空机票的相应登机口查看 waiting list 是否被兑现成功。



## （2）国内研究现状

国内当下常用的遏制第三方软件刷票行为的手段主要有提高验证码识别的难度，风控拦截以及身份证实名认证。提高验证码的难度主要通过将验证码加入中文逻辑，这对验证码破解方法有极强的遏制效果<sup>[21-25]</sup>。风控拦截主要根据用户行为分析增加行为逻辑控制，通过风控系统防范短时间提交大量订票请求、防范购票操作过快以及防范查询余票过于频繁，由此对非法用户进行刷票限制<sup>[26]</sup>。身份证实名认证是旅客必须使用自己的身份证号来购票，保证一人一票，限制第三方软件刷票占票。除了技术层面的防范手段以外，在售票模式上也可以借鉴其他行业的方法来优化旅客购票体验。国航在 2013 年提出了“登机口候补”的售票模式，当乘客在意愿购买的航班无票可售的情况下，乘客可以在机场的服务窗口购买相应席位座位的候补票，航班起飞前 30 分钟确定该航班具有空余座位后，可按照乘客购买候补票顺序引导乘客登机。以上方式虽然在遏制刷票行为方面有着一定的借鉴作用，但是并不完全适合我国铁路行情。

为了引导旅客有序购票，减少旅客盲目刷票，彻底遏制第三方网站的刷票代购行为，降低客票系统运行压力，提高铁路客运营管理效率和服务质量，本文以开源软件为依托，以分布式系统为基础，研究出最优的技术选型方案，设计出最优的技术架构，建立以 12306 客票系统候补购票服务系统方案。

## 1.3 研究目的及意义

无票候补购票服务是指在车票售罄后，旅客可以在 12306 官方网站及手机 APP 中提交“候补购票”订单。如果客票系统生成新的余票，系统将会以所有旅客提交的无票候补购票订单的前后顺序进行排队兑现。为了进一步提升铁路旅客出行体验，提高服务水平，仅研究在 12306 网站及手机 APP 上提供预售期内车次、席位无票的情况下的实现无票候补功能，车站窗口、代售点、自动售票机及电话订票不提供无票候补服务。以开源分布式技术为基础的 12306 客票系统候补购票服务系统方案实现后，具有如下积极意义：

### （1）提升用户体验

不仅极大遏制第三方软件刷票行为，还可以更加稳定可靠的大幅改善 12306 服务质量，使用户的查询速度更快，购票体验更好。

### （2）降低资源消耗

无票候补购票方案的广泛而深入应用，可进一步提升数据存储能力和并发访问能力，降低系统负载，进而降低系统建设投入成本，可以间接提高企业经营效益的需要。

### （3）推动技术革新

近年来，开源分布式数据库技术在国内外各种行业的信息系统中广泛应用。从开源分布式数据库技术角度出发，高并发应用场景的稳定安全应对是中国铁路客票系统最亟待解决的问题。这些技术提供了解决之道，并带动了技术堆栈的革新。

### （4）提供营销参考

可以为铁路加开列车、增加编组等业务提供营销参考，可以有效降低社会成本支出，为推动铁路供给侧改革发挥重要作用。

## 1.4 主要研究内容

针对第三方刷票软件的刷票行为，为了降低客票系统运行压力，提升用户购票体验，改善铁路售票秩序，提高铁路客运营管理效率和服务质量，本论文提出了一种基于 12306 客票系统的候补购票服务系统方案，该方案首次从根源层面对第三方刷票软件进行遏制，论文的研究主要内容如下：

一、结合铁路客票系统的发展综述，确定了基于 12306 候补购票服务系统方案的必要性，确定以开源分布式内存数据库技术为基础的 12306 候补购票服务系统方案作为论文的主要研究对象。

二、针对基于 12306 候补购票服务系统方案作需求分析，主要从业务需求、系统需求、安全需求以及网络需求方面进行分析。

三、研究分析目前主流的开源分布式技术，明确主流的开源分布式内存数据库、web 平台以及消息队列各自的技术特性及其在候补购票服务系统中的适用场

景。

四、在明确 12306 候补购票服务系统方案需求、主流技术的基础上，进行总体系统设计，形成 12306 候补购票服务系统方案的技术框架、建立相应的应急处理机制、确定适应长期发展的扩展机制以及保证系统整体安全的安全方案。

五、将建立的候补购票服务系统进行应用，验证 12306 候补购票服务系统方案的实用性和可行性。



## 2 候补购票服务系统需求研究

针对基于 12306 客票系统候补购票服务系统方案作需求分析，主要从业务需求、系统需求、安全需求以及网络需求方面进行分析，从而明确候补购票服务系统方案的功能和技术指标。

### 2.1 业务需求

#### 2.1.1 业务规则

候补购票服务系统在 12306 网站、手机 APP 渠道开展候补服务，主要服务对象为已激活的铁路畅行会员以及通过人脸核验的 12306 注册用户。对于人脸比对不通过及不具备在线核验条件的用户，手机端提供人脸图像采集功能，旅客可以上传证件照片和近照，后台通过人工校对（周期为 3 个工作日）旅客上传证件照的真实性，如人工初审通过，再进行线上人脸与上传照片的比对认证。

候补购票服务系统支持预售期范围内的车次提交候补请求。30 天预售期内没有余票时，均可提交候补需求。在提交候补需求至购物车时，如果候补队列人数过多、兑现成功率较低，则提示给用户。候补订单内每个候补需求的估算兑现成功率（如：较高、一般、较低），系统后台结合历史售票情况、当前用户候补排队位置综合估算兑现成功率。结合大部分旅客的出行需要，每个用户最多提交 4 个（天）待兑现实态的候补订单，每个候补订单中最多可添加 2 个乘车日期及发到站相同（发到站可同城）的不同“车次+席别”组合。每个用户在 1 个自然日内最多取消 5 次待支付候补订单（与既有互联网订单取消次数无关）。可视候补运行情况灵活调整候补订单数量和“车次+席别”组合数量上限。

提交的候补订单包含乘车日期、车次、发站、到站、席别、票种和乘车人身份信息。为提高候补单的兑现成功率，候补订单提交时不支持选座。候补购票服务系统支持用户自定义截止兑现时间（用户选择日期和小时），需不晚于系统默认截止兑现时间。允许用户在提交候补订单之前选择自定义截止兑现时间，提交候补订单之后，不允许修改。所有车次默认限定最大累计候补请求数量均为 500（参照整车定员 60%左右），可以根据运行情况进行优化调整。

为保证候补需求的真实性及避免用户二次支付，用户需要预付候补需求订单

中最高席别的票款（卧铺按下铺票价进行计算），支付时间为 30 分钟，同一时间仅允许存在 1 个待支付候补订单。生成候补订单时，不检查候补需求与已购车票是否存在违反实名制购票规则和行程冲突的情况。兑现候补订单时，检查实名制和行程冲突。若有，则兑现失败，提示给用户。待兑现的候补订单之间不判断实名制购票规则和行程冲突。当旅客先候补、后买票时，提示旅客。先买票，后候补时，候补兑现成功时提示旅客。用户提交候补订单时，页面提示用户兑现的车票可能不连续，既有其他相关提示也一并梳理。

候补订单提交后不可进行任何修改。用户可主动取消待兑现候补订单；到期无法兑现时，系统自动取消待兑现候补订单，上述两种取消均全额退还候补订金。到期兑现成功时，系统自动生成已支付订单，对于预付款大于实际票款情况，系统自动退差。对于有候补的车次、区间和席别，当出现余票时，优先兑现给候补需求，窗口、自动售和电话订票等渠道显示为“无票”；当候补队列中的需求处理完毕后，其他各渠道才显示余票信息，然后比照正常流程购票。

兑现车票时，同一候补需求不支持部分兑现。候补购票服务系统支持共用和通售用途的兑现，其他用途车票的发售与候补无关。后台系统进行兑现时，技术上支持集中时间段和分散时间段兑现，根据实际运行情况灵活调整。将轮询频率设置为 1-2 小时，并且通过运行情况进行灵活调整。候补兑现成功或失败时，向用户推送通知消息。

## 2.1.2 候补购票服务系统功能

### （1） 候补购票服务系统入口功能

候补购票服务系统入口为 12306 网站及手机 APP 的余票查询页面，在预售期内车次、席别无票的情况下，提供候补购票功能。车站窗口、代售点、自动售票机及电话订票不支持候补服务。

### （2） 候补购票服务系统服务对象及办理范围

候补购票的服务对象是已激活的铁路畅行会员以及通过人脸核验的 12306 注册用户。现有互联网可售的所有车站、车次、席别，当没有余票时，均可提交候补需求。其中学生票支持 30 天内预售期的车次候补（同成人票一致）。

无座席别不参与候补，“其他”包含的席别同样不参与候补。在候补购票上线



时，向符合条件的用户推送通知，介绍候补购票服务，引导用户使用候补购票。

### （3） 候补购票服务系统订单的提交和变更

#### 1) 提交候补订单

结合大部分旅客的出行需要，每个用户最多同时存在 4 个待兑现状态的候补订单，1 个候补订单中最多可添加 2 个“同一乘车日期+站车次+席别”组合（发到站一致或同城）。每个用户在 1 个自然日内最多连续取消 5 次候补订单（支付前的取消），与既有互联网订单取消次数无关。

提交的候补订单包含乘车日期、始发日期、站车次、始发车次、上车站、下车站、席别、票种和乘车人身份信息等内容。提交候补订单时不显示既有候补人数，在候补订单支付页面向用户提示候补订单内每个候补需求的估算兑现成功率。

生成候补订单时，不判断候补需求与已购车票是否存在违反实名制购票规则和行程冲突的情况。在兑现候补订单时，检查实名制和行程冲突，若有，则提示给用户。候补订单之间不判断实名制购票规则和行程冲突。用户需要支付当前候补订单中最高票款（卧铺按下铺票价进行计算）作为“候补订金”。所有车次默认限定最大累计候补请求数量均为 500，可根据需要进行调整。

#### 2) 查询候补订单

候补订单提交后，用户查询候补订单时，包含：待支付、待兑现、已处理订单。针对兑现失败的特殊情况显示兑现失败原因。

#### 3) 候补订单变更

用户可主动取消候补订单；到期无法兑现时，系统自动取消候补订单，以上两种取消候补订单的情况均不收取费用。

### （4） 候补购票服务系统订单兑现

在兑现过程中，遵循以下原则：

- 1) 按候补订单支付的先后顺序进行处理。
- 2) 为避免频繁退票、业务逻辑与页面交互复杂以及旅客纠纷等情况，同一候补需求不支持部分兑现。
- 3) 候补购票后台系统支持灵活的兑现时间，尽可能分散兑现。
- 4) 兑现成功后，如候补订金金额大于实际票款，系统将自动退差；如到截止

时间或因行程冲突等原因无法继续兑现，则全额返还候补订金。如没票或数据库整理导致失败等情况则继续兑现，不提示给用户。

5) 兑现过程中进行实名制卡控及行程冲突控制。

6) 候补订单截止兑现时间为乘车日期前一天系统停售前 1 个小时，即 22:00。截止提交时间早于截止兑现时间，以分钟为单位，进行配置。

### (5) 候补订金、退款及对账结账功能

旅客提交候补订单后，需支付最高票款（卧铺按下铺票价进行计算），待车票候补成功后，如有差价，系统自动退差。候补候补订金的收款和退款明细，标识后随票款交易明细，发送支付平台，支持对账处理。

## 2.1.3 车票发售功能

所有线上（网站、手机 APP、微信）和线下（车站窗口、代售点、电订、TVM、自动售、车站大屏）单独的余票查询服务和售票服务，如果对应的“始发日期+上车站+下车站+席别”有候补，返回无票；如果没有，则显示实际余票数量并可继续发售。

## 2.1.4 通知

候补订单的状态变化需要及时通知到旅客，通知方式包括短信、支付宝、微信等，用户可自主选择以何种方式接收通知。通知内容包括候补订单主要信息以及兑现成功率（非常高、较高、一般、较低、非常低）。应避免夜间（23:00-06:00）期间发送大量短信，给旅客带来不便。

## 2.1.5 客服服务

为客服提供候补需求订单的查询服务及对应票款的支付、退款状态查询服务。

## 2.1.6 统计分析功能

以准实时的方式将用户提交的候补订单传输到营销数据仓库，用于数据挖掘

分析。分析结果可为候补购票服务调整策略提供依据，可为列车增开、调整提供依据。

## 2.2 系统需求

按预售期 30 天，全路车次总数 6000 个，主要席别 4 个进行估算，共有 72 万个“乘车日期+车次+席别”组合。在春运售票高峰期间，假设每个组合最多接收 500 个候补需求，这种情况下，最多有 3.6 亿个候补需求。以此为依据估算并发处理能力和数据存储能力。

### 2.2.1 并发处理能力

因提交候补购票的场景发生在查询无票之后，因此候补购票服务的并发处理能力应强于售票能力：2018 年春运售票高峰期间售票量约 625 张/秒，预计 2020 年春运售票高峰期间售票量约 833 张/秒。

1 个候补余票查询集群的处理能力应在 5000 次/秒以上，此时可在 144 秒内完成 72 万个“乘车日期+车次+席别”组合的余票数据轮询任务。候补轮询和兑现服务的处理能力取决于候补余票结果、路局中心处理扣票请求的能力、候补订单数据库的处理能力。系统通过流控机制动态调整轮询频率、兑现频率。

### 2.2.2 数据存储能力

数据存储需满足海量数据的集中处理、存储需求，能够存放 30 天预售期内的候补购票订单数据，包括：待兑现候补订单、已取消候补订单、到期无法兑现候补订单。不存储早于 30 天的历史候补订单。为保证数据安全，数据存储在客服内网。

候补订单拆分数据库中存储候补订单的拆分归类信息，每行数据约 150 字节，预售期 30 天，全路车次总数 6000 个，主要席别 4 个进行估算，共有 72 万个“乘车日期+车次+席别”组合，假设每个组合最多接收 500 个候补需求，这种情况下，该表最多有 3.6 亿行数据。候补订单拆分数据库具备 50G 以上的存储能力（不含主备、索引、日志等）。

候补订单详情数据库中存储候补订单主要信息，每行数据约 800 字节，由于系统最多接收 3.6 亿候补需求，最多对应 3.6 亿候补订单，这种情况下，该表最多有 3.6 亿行数据。候补订单详情数据库具备 268G 以上的存储能力（不含主备、索引、日志等）。

## 2.3 网络需求

候补购票服务涉及环节较多，对客服外网、客服内网、客票网均有网络访问。候补购票服务系统的对外部互联网供给服务的服务器设定在外部服务系统中，候补购票服务系统的客票交易服务器设定在内部服务系统中，他们之间通过安全平台进行联通。

## 2.4 安全需求

### 2.4.1 系统安全性

系统的安全主要体现在系统的三个方面：物理安全、逻辑安全以及管理安全。物理安全主要体现在系统的硬件设备不应受到物理伤害；逻辑安全主要体现在系统中所有数据资源的安全性，必须具备保密、完整和可用的特点；管理安全主要体现在系统的管理机制是否保密完整。

### 2.4.2 数据安全性

要求隐私数据的存储需要放置在内部服务网，通过内外网安全平台进行通信，保障数据的安全性。

### 2.4.3 数据保密性

候补购票服务系统涉及旅客的个人隐私和订单信息，需严格限制访问人员及访问权限，不越界传输、不越权使用、不遗留、不提前释密。建立管理用户访问日志，记录管理用户对系统的访问操作，具备行为审计能力。

## 2.5 本章小结

本章主要针对基于 12306 客票系统候补购票服务系统方案作需求分析，主要从业务需求、系统需求、安全需求以及网络需求方面进行分析，从而明确候补购票服务系统的设计思路。





### 3 候补购票服务系统相关理论及技术研究

根据以上需求分析，候补购票服务系统的技术堆栈丰富，运用大量先进的成熟的具备高性能的互联网以及计算机技术，本章节主要介绍与之相关的关键技术开源分布式内存数据库技术和排队算法。

#### 3.1 排队算法

无票候补候补购票系统对提交候补订单的所有用户进行排队模式的请求兑现，目前主流的排队算法主要有如下几种：

##### （1）FIFO 排队算法

FIFO（First In First Out）排队算法是指在排序队列中使用先进先出的逻辑方式，优先提交候补订单的用户在很大程度上可以被优先兑现购票请求。

当前排队算法主要应用于电商行业，例如当前大型购物活动双十一以及双十二期间，许多淘宝以及天猫店铺选择抢售方式，虽然采取了多线程的方式保证销售速率，但是当产生冲突情况下依旧安装先进先出的原则给予用户购买权。由于在电商行业中，每一个用户购买需求所需处理的时间都几乎相同，不存在一个过长处理时间的任务会造成后续排队成员有较长等待时间，因此 FIFO 排队算法适用于每一个处理需求所需时间相近的情况。

##### （2）PQ 排队算法

PQ（Priority Queueing）排队算法主要指对不同排序对象使用优先级的权重标记，对处于高优先级的排队对象进行优先处理，低优先级的排队对象进行靠后处理，在每一级的范围内依旧采取 FIFO 排队算法。

权重排队算法当前适合处理同一队列中不同任务需要不同响应时间的应用场景，在餐饮行业中该排队算法应用较多。针对不同人数的排队用户，人数较少的优先解决，一般在餐厅内小桌的供应量更大，因此当前自动叫号机会按照输入的人数进行自动排位，当内部空间出现多余的位置时会按照人数设定的权重进行自

动服务。

### （3）WFQ 排队算法

WFQ (Weighted Fair Queueing) 排队算法是以流为对象，对不同流所指派的队列进行先后处理，首先对不同的流来源进行权重设置，其次在不同流来源内部的排队对象进行 FIFO 处理。

### （4）LLQ 排队算法

LLQ (Low Latency Queueing) 排队算法是优先级排队算法、加权公平排队算法以及自定义排队算法的综合使用，它可以很好的满足对延迟要求低的系统需求，主要优先处理具有绝对优先级的排列对象，其次处理剩余的排列对象。

以上算法中，FIFO 先进先出的排队算法虽然简单，但是却十分符合用户候补排队购票的请求模式，该算法在业务逻辑上可以保证旅客候补购票的公平性，其简单的实现方法也降低了对系统硬件的要求，十分适用于候补购票服务系统。

## 3.2 拟合预测算法

候补购票服务系统需要满足用户预判一张候补订单是否能够被兑现的需求，因此在用户提交候补订单的同时应当在展示页面的某个模块向用户显示出当前提交候补订单可能被成功兑现的概率。

影响一趟车次相应席位的兑现成功率因素众多，从社会生态学出发，乘客乘坐列车出行属于自然迁徙，因此在某种程度上该迁徙具备一定的规律性。比如工作日的周一至周四出门的人相对较少，而由于周末出去游玩或者固定返乡的原因，周五以及周日会有较多的乘客选择火车出行。而当某个地点举办大型活动或者明星演唱会时，通往该地点的列车的乘客会骤然增多。一趟列车能够提供的座位是一定的，如果提交的候补订单数越大，那么当该数趋近于无穷时，候补订单的兑现成功率越低。因此在众多复杂因素影响兑现成功率的计算结果状况下，候补兑现成功率的计算算法显得尤为重要，以下是当前常用的一些数据拟合与预测算法：

### （1）多项式线性拟合算法

多项式线性拟合是当前对于离散点进行规律性拟合最常用的拟合算法，使用

该算法对兑现成功率进行拟合需要利用兑现成功率历史数据。

假设当前需要拟合列车的兑现成功率为  $P_{N+1}$ ，那么该区间兑现成功率非线性多项式拟合模型如下所示：

$$P_n = C_0 + C_1 \cdot n + C_2 \cdot n^2 + \cdots + C_k \cdot n^k$$

公式中  $n$  的取值范围为  $[1, N+1]$ ， $C_0, C_1, \cdots, C_k$  为多项式拟合系数， $k$  的取值不同将决定该式的拟合效果， $k \geq 2$ 。此时将上述前  $N$  个不同时期兑现成功率以及相应  $n$  值代入，并自取不同的固定  $k$  值便可计算求得  $C$  系列系数，最后将  $N+1$  带入便可求得  $P_c$ 。并非通过该拟合算法对历史数据的拟合效果越贴合便代表对未来兑现成功率的预估越准确，因为该公式的系数十分灵敏，因此在选区  $k$  值时应该适当的牺牲一定精准度从而保证预测结果不会出现极大误差。

该算法的优点是在预测兑现成功率时，计算简单，并且当下各个应用组件附带的数学函数都有成熟的解决方案，不需要再研发。缺点是当面对突发情况，例如某地大型活动的临时举办等现象造成的兑现成功率的突降，没有很好的平滑计算方法，可能会给乘客提供较大误差的兑现成功率。

## （2）机器学习预测算法

当下时兴的基于机器学习的基础算法延伸出多种数据挖掘与预测函数，例如支持向量机、随机森林以及概率图模型等等。这些机器学习算法主要的计算逻辑都需要将兑现成功率的历史数据分为训练集以及测试集，主要步骤为下：

1) 设计好相应机器学习算法模型，主要关键点是设计好合适的评估函数，以决定向何种梯度方向逼近更接近真实的兑现成功率。

2) 对兑现成功率历史数据的训练集进行训练，得到最优符合训练集的相关经验参数。在进行训练的过程中，容易出现评估函数陷入局部最小，因此在设计模型时应当加大函数多样性以及训练集的样本多样性，多样性主要体现在数量的多以及数据类型的丰富。除此以外还可以从多梯度方向对评估函数进行计算，从而增大跳出局部极小的概率。

3) 用测试集对训练好的机器学习模型进行验证，并在该过程中反馈优化机器学习模型从而达到该机器学习模型可以更好地预测多种复杂情况下的兑现成功

率。

机器学习算法相对于多项式拟合算法的优点是它可以给出更趋近历史数据样本模型的预估值，缺点是机器学习算法过于依赖训练集样本，训练集越全面，预测结果才越准确。

### (3) 神经网络算法

从以上分析得知兑现成功率的预测是复杂度较高的一种物理模型，神经网络算法是处理复杂度高并且需要多次迭代的物理模型的较优选择。一般简单的神经网络算法分为三层，分别为输入层神经网络、隐含层神经网络以及输出层神经网络。复杂的神经网络算法都是在该简单神经网络算法的基础上进行延伸，比如在隐含层神经网络进行复杂化，采用多隐含层神经网络，加大层与层之间神经元的黑匣联系度；也可以复杂神经网络模型层与层之间的联系，比如在输入层加入向隐含层神经网络的负反馈，并层层逆向传递形成负反馈神经网络。

神经网络算法的计算逻辑大体相似，具体步骤为下：

1) 设计好神经网络模型架构，依据兑现成功率具体计算业务进行神经网络层数、函数模型以及神经元个数设计。

2) 使用训练集样本对设计好的神经网络模型进行训练，从而得到每个神经元的相应权数及参数。神经元的权数以及参数在初始化过程中可以使用随机方法，因为神经网络模型可以无限次迭代，随机的方法可以使得模型更容易跳出预测结果的局部极小。

3) 使用测试集样本对神经网络模型进行验证与调整，从而使得模型更加优化。

神经网络算法的优点是它可以更准确的在历史数据样本的基础上对兑现成功率进行预测，缺点是该算法对计算机系统的硬件要求较高，会消耗较多资源并且计算速度与系统硬件水平直接挂钩，不适合实时系统。

经过分析以上主流数据拟合预测算法，采用兑现成功率的简单业务逻辑算法，主要通过提交候补订单数与可提交候补订单数上限的比值定性告知乘客当前排队人数较多、中等或较少，从而辅助旅客做决策（具体见 4.2.5）。机器学习与神经网络算法的预测精度虽然较高，但是对系统硬件的过高要求导致模型应用性价比较低。

### 3.3 内存数据库技术

随着 IT 技术的快速发展,分布式技术和内存数据库技术都有了革命性的突破。分布式技术能够更加充分地将分布在各处的存储资源、运算资源高效整合为统一的整体<sup>[27-32]</sup>。内存数据库技术则可以更有效地使用内存和 CPU 时钟周期。自 2012 年以来,铁路 12306 互联网售票系统开始使用商业化的分布式内存数据库产品,负责具有高并发访问需求的业务,主要应用业务场景有:余票查询、票价查询、订单查询、实名制信息查询、用户查询、常用联系人查询等。针对铁路客票系统业务,数据量已从最初的 1 亿增长至 19 亿,高峰期访问 TPS 已从 1000 增长至 3000 以上。

候补购票服务系统基于以上技术实现的数据存储关系如图 3-1 所示,高频使用数据存储在第三模块数据库数据集群中,该集群中的存储信息基本能够满足请求,保证了系统整体数据存储的可靠稳定性<sup>[33-38]</sup>。

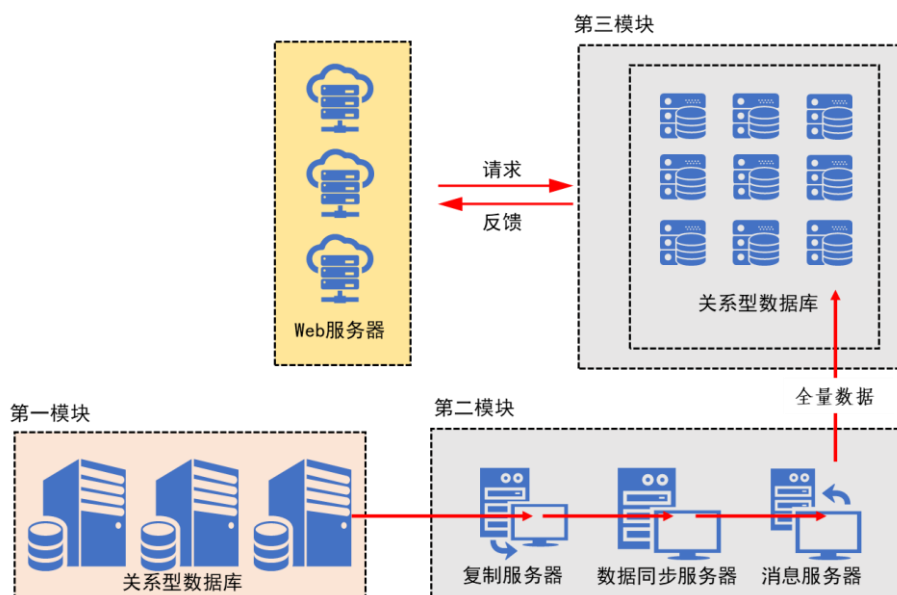


图 3-1 候补购票服务系统的数据存储关系

Figure.3-1 Data Storage Relation of No Ticket Line Up Buying System

在第一模块中,无票候补订单数据的所有根数据在关系型数据库中存储,该数据库主要通过无票候补业务的某个关键字段来划分存储,关系型数据库位于最深层次安全网络结构中,被直接操作与访问的次数极低,是保证无票候补业务数据有据可审的基础。

第二个模块是无票候补业务数据同步模块,它是实现降低关系型数据库被直

接操作与访问次数的必不可少环节。复制服务器的功能是以固定频率监听关系型数据库，当关系型数据库存储的无票候补业务数据发生改变时，复制服务器便捕获该数据变化并将该数据变化传递给数据同步服务器。数据同步服务器将会以消息的形式把数据变化发送到消息服务器，消息服务器属于透明的消息转发中间件模块，它可以对接受的消息形式的 SQL 语句进行解析，从而将数据写入第三模块的数据库中。

第三个模块是高效存储运算模块，该模块数据库对高频无票候补业务数据字段信息进行存储与运算，极大程度提高了请求处理效率。该模块对接无票候补业务功能接口，对负载均衡优化算法分配的请求进行处理与反馈。

正是由于这样的数据存储规则，当前项目面临的问题比较多，主要有如下几点：

（1）稳定性不强：庞大的数据量给老旧集群带来沉重负担，Java GC 更加频繁，对应用正常运行产生影响。

（2）扩展性很差：业务数据和逻辑都运行在同一容器中，受制于数据量和应用架构，无法从容地增加表、增加逻辑。

（3）容灾机制面临瓶颈：因数据量过于庞大，当发生宕机、掉点等异常情况时，业务数据 rebalance 压力很大；同时，集群已经不具备快速初始化、创建索引的条件，一旦集群持久化数据损坏，将无法恢复。

（4）高可用机制不完善：没有流量调度、集群切换机制，一旦某个集群故障，只能重启相关应用服务，切换集群。

（5）对外耦合度高：外层应用接入时，只能通过产品专用的客户端连接池接入服务。

针对面临的突出问题，研究开源分布式内存数据库技术在候补购票服务系统方案系统的应用。

### 3.3.1 内存数据库

#### （1）基本概念



内存数据库是一种将其库中的所有数据经常驻防在内存中的数据库类型。当计算机的 CPU 运行后，只用通过计算机内存的输入输出流操作就可以对数据库的所有数据进行存放与读取操作，运行效率极高，从而内存数据库体系结构设计的初衷就是以提高内存以及 CPU 使用效率为根本目的。

内存数据库的概念在 20 世纪 80 年代被提出，以下两点为内存数据库爆发式发展的原因：

1) 高精尖领域如军事防卫系统的雷达、金融领域的所有交易操作以及基于大数据的专家分析系统、自然语言处理和复杂机器人系统等都对传统数据库的功能提出了更高的要求，主要需求数据库要有极短的响应时间，这种不满促进了内存数据库的研发。

2) 内存在硬件技术方面有了极大的提升，在容量、稳定性、价格、主频率方面都有了极大的优化，性价比极大提升。

## (2) 与基于硬盘的数据库的对比：

下表列出了内存数据库与磁盘数据库的主要不同点：

表 3-1 内存数据库与磁盘数据库不同点  
Table.3-1 Differences between memory database and disk database

比较项	内存数据库	磁盘数据库
数据存储	采用列级和分区式存储模型，可以在不同的物理机上，不要连续存放。	采取连续存放，数据存储效率与物理条件相关
索引结构	使用 T 树或 Hash 索引结构，适合内存存储。	一般使用 B 树索引，适合磁盘存储。
查询优化	查询速度主要取决于处理器性能和内存性能。	查询速度主要取决于磁盘、内存、处理器之间的 IO 性能。
灾难恢复	采用预提交、批量提交等方式来存储日志记录。	采用备份、日志、检查点、保存点等技术。

数据库数据的长期存储方式是主要区分磁盘数据库和内存数据库的方法，内存数据库将数据备份存储在稳定且可靠的磁盘上以备用于数据恢复。当前成熟的内存数据库产品有多种类型，本文主要介绍 Redis 内存数据技术。

### 3.3.2 Redis 内存数据库技术

#### (1) 基本概念

Redis 是内存数据库技术中的代表技术,它具有典型的内存存储数据结构系统,Redis 既可以当作数据库来使用,也可以用于缓存,作为消息中间件,整个技术的开发获得 BSD 许可认证。Redis 可以用于多种类型的数据存储,常见的字符串(strings),散列(hashes),列表(lists),集合(sets),有序集合(sorted sets)都是 Redis 常用范围,Redis 也可以用于范围查询以及地理空间(geospatial)索引半径查询。

Redis 内部设置多项应用,其中主要包括复制,LUA 脚本,LRU 驱动事件,事务以及不同级别的磁盘持久化。Redis 为了提高其应用系统的本身高可用性,它使用了哨兵和自动分区。Redis 对于使用其本身的用户提供一些类型的原子操作,这些类型包括字符串中的 append 命令;散列函数中的 hincrby;列表函数中的 lpush 命令;集合函数中用于计算交集的 sinter 命令等等。

Redis 通过使用把所有数据放在内存中的运行方式来保障整个系统的优质性能。在不同应用场景中,可以使用 Redis 以固定频率把其数据库中的相应数据集拷贝至磁盘,也可以使用 Redis 关闭其拥有的持久化功能,变相的把 Redis 应用为具备缓存数据功能的数据库。

#### (2) 与基于硬盘的数据库的对比:

Redis 分布式内存数据库可以带来以下优势<sup>[39-43]</sup>:

##### 1) 增强节点扩展性

Redis 集群可以动态地向集群中增加新的存储、运算节点,扩展性极强。

##### 2) 提高并行处理能力

当客户使用 Redis 组装成集群,用户可以根据自身设计的存储数据规则,使用多个不相同的物理节点把不同范围不同类型的数据记录存储在集群当中,在相应的物理节点上也会分布着与之对应的并行查询和并行处理,综上,使用控制节点进行汇总,因此,集群性能将得到提升。

### 3) 加强数据可靠性

Redis 数据库数据的备份版本存储在若干个不同的节点中, 并且可以实现数据持久化, 每一个数据库节点也具有恢复备份机制。

### 4) 良好的负载均衡

负载均衡主要体现在两个方面:

首先, 数据存储的负载均衡, Redis 技术集群主要分为两级架构, 两级架构之间是松耦合状态, 处于下级的分布式模块主要实现了数据节点负载均衡功能, 优化了传统分布式系统数据节点容易倾斜的问题。

其次, 并发访问的负载均衡。有些分布式内存数据库产品一般会有控制节点, 该控制节点的作用是管理服务节点与客户端的连接、路由客户端访问请求到负载较低的服务节点上。

## 3.3.3 OpenResty web 平台

### (1) OpenResty 概述

通过对 OpenResty 综述学习, 本论文认为 OpenResty 技术的定位为: OpenResty 是一个具备优良性能的 Web 平台, 它的技术基础是在 Nginx 以及 Lua 上建立完成的, 因此, OpenResty 对 Lua 库以及大多数的依赖项有着优良集成, 它的最大优点是可以搭建 Web 应用、服务以及动态网关, 所有这些应用都可以很好的处理高并发并且具备极高的扩展性。

在 OpenResty 构建 Web 应用平台的应用场景中, OpenResty 可以使用多种优良的 Nginx 模块, 通过这种汇聚优势, 可以使得 Nginx 构建为一个通用的应用平台。在这种通用的 Web 应用平台基础上, 开发人员可以使用相应的脚本语言以及模块搭建 1000K 以上优良并发连接性能的 Web 应用系统。OpenResty 最主要的作用可以让使用者的 Web 服务运行在 Nginx 模块服务内, 这样可以充足的利用 Nginx 的非阻塞 I/O 模型。

### (2) 融合 Redis

OpenResty 融合 Redis, 也即是通过 OpenResty 的 Nginx 模块对 Redis 进行访

问，可以通过多种模块技术完成，例如 `HttpRedis` 模块、`HttpRedis2Module` 模块以及 `HttpLuaModule` 模块的 `lua-resty-redis` 库等。不同的模块库技术具备不同的优缺点，`HttpRedis` 模块应用较局限，主要是由于其本身可提供的指令类型单一，但是具备一定的扩展性。相较于 `HttpRedis` 模块，`HttpRedis2Module` 模块可以使得用户根据自己的需求设计逻辑处理，通过提供可以被使用的 `Redis` 接口让模块操作灵活。`HttpLuaModule` 模块的 `lua-resty-redis` 库使用户可以操作 `with-luajit`。

### 3.3.4 消息队列技术

#### (1) 消息队列的概念

消息队列（`Message Queue` 简称 `MQ`）是常用的一种通信方法，主要目的是使得应用程序与应用程序之间建立消息传递关系。在不同架构的系统中，消息队列是关键应用技术，它可以在不同架构的系统之间建立消息包，不同架构系统的应用程序由此在消息包中给彼此传递数据，这保证两个异构的应用程序不会直接传递通信。综上，消息队列可以被用作中间件技术，在使用消息队列技术的过程中，可以忽略异构应用程序同时执行的要求，这便可以间接实现异构系统应用程序的异步通信，在理论上，性能强大的消息队列服务器可以实现“准同步通信”<sup>[44-48]</sup>。

#### (2) RabbitMQ 消息队列技术

`RabbitMQ` 是消息队列的代表技术之一，它的基础编写语言为 `Erlang`，由于其消息队列系统的协议为 `AMQP`，并且开源，所以 `RabbitMQ` 技术得到广泛应用。从框架角度来说，`RabbitMQ` 分为客户端和服务端两部分：客户端可以是消息生产者(`Producer`)和消息消费者(`Consumer`)；服务端就是 `RabbitMQ` 消息队列服务器本身。其组成框架如下图所示：

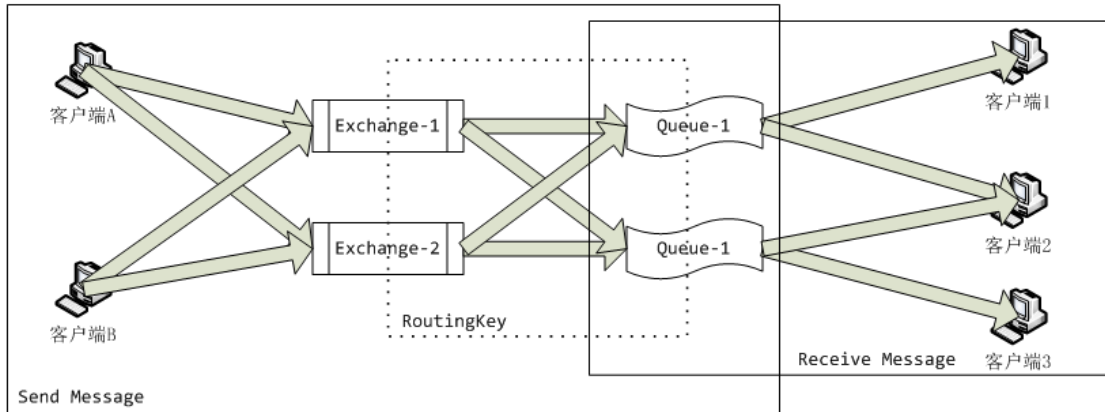


图 3-2 候补购票服务系统的数据存储关系

Figure.3-2 Data Storage Relation of No Ticket Line Up Buying System

组成框架的各个组件（服务）的功能如下：

**Broker：**代表了消息队列服务器实体本质为服务进程。

**Binding：**绑定协议，该协议的主要功能是提供了路由的规则，使得 Exchange 和 Queue 有了确切关联，主要内容包含了 Exchange 和 RoutingKey。

**Exchange：**消息交换机，主要包括完全匹配交换机(amq.direct)、模式匹配交换机(amq.topic)等，该消息交换机主要通过系统的 RoutingKey 属性对消息进行投递。

**RoutingKey：**路由关键字。

**Queue：**消息队列实体。

**VirtualHost：**虚拟主机。

**Producer：**产生消息的系统应用程序。

**Consumer：**接收消息的系统应用程序。

**Channel：**消息通道，一个 channel 代表一个会话任务。在客户端的每个连接里，可建立多个消息通道。

RabbitMQ 消息队列技术的消息处理流程主要有以下几步：

- 1) 使得消息队列的服务器与客户端相互连接，通过使用一个 channel 使得连接建立，并且对 Exchange 中的相应属性进行修改。
- 2) 客户端建立 Queue 并设置相关属性。
- 3) 在客户端处使用确定的路由关键字 RoutingKey 在消息交换机以及信息队列之间建立绑定关系，从而使得客户端能够在消息交换机中发送消息。

- 4) 消息交换机接收到消息后, 将消息发送到一个或多个队列里。
- 5) 接受消息的系统应用程序可以随时启动, 接受指定 Queue 中的消息, 进行后续处理工作。

### (3) RocketMQ 消息队列技术

如图 3-3 所示为 RocketMQ 技术架构, 由 NameServer、Broker、生产集群以及消费集群构成。NameServer 的节点与节点之间没有消息同步, Broker 的构成相对复杂, 它通常分为两类, 一类为 Master 另一类为 Slave。一个 Master 可以有多个 Slave 从属, 但是 Slave 只能与唯一的 Master 对应。

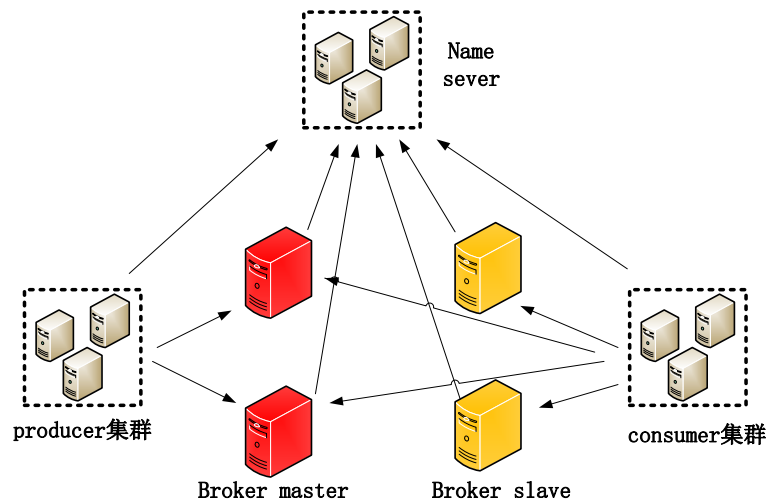


图 3-3 RocketMQ 技术架构  
Figure.3-3 Technical framework of RocketMQ

RocketMQ 技术在储存消息对消息进行操作的过程中使用了零拷贝原理, 该原理一般通过两种方式实现, 第一种是 mmap 和写方式结合的技巧, 另一种是使用发送文件的方式。RocketMQ 消息队列技术具有消息队列轻量, 每一个队列所具有全部数据量少的特点, 并且对系统磁盘进行了串行化访问, 从而减少了磁盘竞争。

## 3.4 技术选型

在关键技术研究, 对数据库、Web 平台、消息队列等技术进行了研究, 以此为基础, 在系统设计阶段, 结合产品特性和业务产品需求。

### 3.4.1 数据库选型

在进行数据库选型时，从技术、业务这 2 个角度进行分析筛选。

#### (1) 技术角度

在数据库选型阶段，从技术角度对比基于内存的 KV 数据库和基于硬盘的关系型数据库，对比分析结果如下：

表 3-2 KV 数据库和关系型数据库对比

Table.3-2 Comparison of KV databases and relational databases

名称	基于内存的 KV 数据库	基于硬盘的关系数据库
存储介质	内存	硬盘
数据结构	Key-Value	关系型 row-oriented
复杂查询	弱	强
事务支持	一般	强
性能	高	低
存储能力	依赖内存容量	依赖硬盘容量
高可用性	依赖产品	依赖产品
稳定性	依赖产品	依赖产品
扩展性	普遍较强	一般
安全性	依赖产品	依赖产品
技术应用程度	较高	一般

如上表，基于内存的 KV 数据库的并发处理能力、扩展性更强，基于硬盘的关系数据库的复杂查询能力、事务支持能力更强。

对于服务来说：

- (1) 并不需要非常强的并发处理能力；
- (2) 系统建设时预留 3 年-5 年的数据冗余量，短期内没有扩展集群规模的必要。
- (3) 在数据写入环节，需要强一致性的保障。同时需要尽可能避免脏读。

从数据库产品特性角度来看，两者基本一致，基于内存的 KV 数据库略有优势。

#### (2) 业务角度

从业务角度来看，查询业务逻辑、数据结构并不复杂：业务逻辑上，需要支持按照用户名查询信息，并且需要经过合法性校验等逻辑。数据结构上，包含多种索引，包含多个表，单表字段 16 个，不含表关联。转换为 KV 内存数据库时，需要建立映射关系。

这种业务逻辑和数据结构的特性，使得在使用 KV 数据库实现时，较为繁琐，需要人为建立并维护索引数据，也使得后续业务扩展时，受到比较大的掣肘。而使用基于硬盘的关系库时，以用户名为依据，对数据进行分库存储，此时也需要建立映射关系。

从业务角度来看，两者基本一致。

### (3) 结论

选用技术应用程度更高的 Redis 内存数据库技术更符合候补购票服务系统的需求。

## 3.4.2 消息队列选型

可选消息队列主要包括两种：RabbitMQ、RocketMQ。

### (1) 技术角度分析

RabbitMQ 它在可靠性、可用性、扩展性、功能丰富等方面较为突出。RocketMQ 是阿里巴巴开发的开源消息队列技术，它在吞吐量、高可用性等方面具备优良特点<sup>[49,50]</sup>。

候补购票服务对消息队列的使用场景是业务数据同步，目的在于解耦异构数据库。对于消息队列中的一些高级特性并没有使用。但是对高可用、持久化、消息顺序性、消息幂等性有着严格要求。在这方面来看，以上两种 MQ 均可满足需求。

### (2) 实际应用角度分析



RabbitMQ 广泛应用于 GemFire、Codis 的业务数据同步，对上游 ctmsx 和下游数据库都有较好的支持，且运行稳定。

RocketMQ 广泛应用于通知系统、异步余票、监控数据等服务中，是后续重点研究使用的产品。

### (3) 结论

两者都可。为降低开发工作量、保障业务稳定，决定采用 RabbitMQ。

## 3.5 本章小结

本章主要介绍了候补系统应用的排队理论及候补系统所应用的存储技术。这里主要对开源分布式内存数据库技术的关键技术进行介绍，并横向对比商业版软件和开源版软件，针对自身应用，对关键技术进行技术选型。明确在内存数据库技术方面选择了以 Redis 为主的数据库，在消息队列技术方面选择了 RabbitMQ，此后候补购票服务系统可直接应用该技术架构，优化系统。



## 4 候补购票服务系统设计与实现

### 4.1 总体架构设计

#### 4.1.1 设计原则

##### (1) 成熟性与先进性的原则

系统架构设计时，考虑充分利用既有成熟的、先进的技术和软硬件平台，提高候补购票需求的处理能力和存储能力，同时也充分预想，考虑系统未来发展需求，引入先进的架构、技术及理念，在未来一段时间内保持先进性。

##### (2) 可靠性与可用性的原则

重构用户登录及查询服务之后，不应降低高可靠性和高可用性，用户登录及查询服务自身也需加强对资源的有效保护以及建立应急处理机制，通过良好的系统设计，使系统的可靠性及可用性达到最优设计。

##### (3) 经济型的原则

系统设计建设考虑充分利用既有软硬件资源，提高资源利用效率，实现资源的有机整合，避免重复建设，提升整体经济性。

##### (4) 可扩展性的原则

系统设计考虑良好的可扩展性，满足现有需求的同时，充分考虑远期互联网售票在访问量、交易量和用户数增长以及应用系统的增加带来的用户数量快速增长情况，架构中各个环节需具备良好的可扩展性，满足业务发展的需要。

##### (5) 安全性的原则

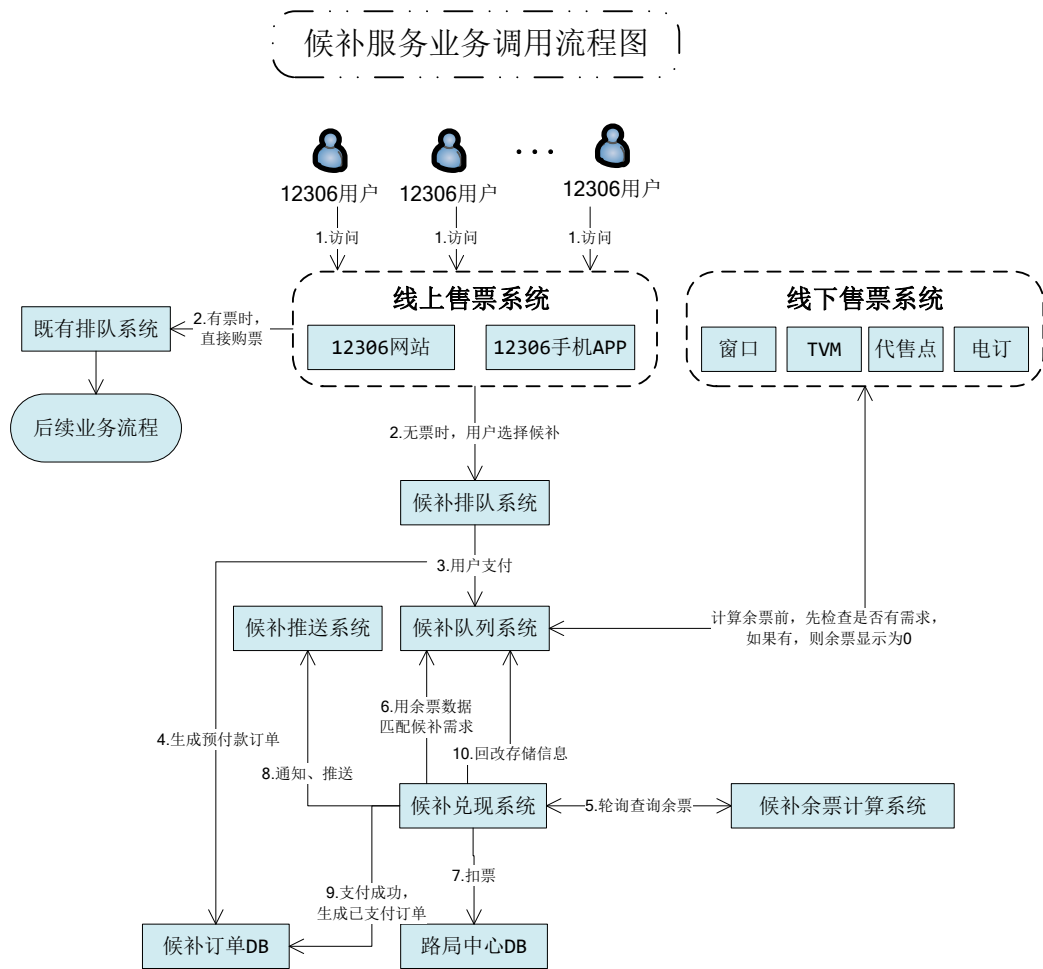
12306 候补购票服务系统应具有良好的安全边界防护，确保客票核心系统安全、确保互联网售票交易安全、确保用户隐私数据安全。

根据前述分析，设计技术架构如下：

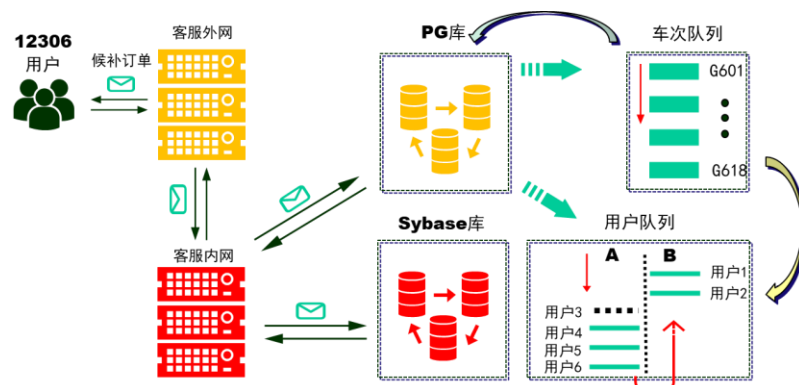
图 4-1 候补购票服务系统各层级功能架构

客服外网部署：候补入口服务、候补排队服务。

客票网部署：候补订单数据库（电子票库）。



(a) 候补购票服务系统业务调用流程



(b) 订单生成兑现

图 4-2 业务调用与订单生成兑现过程

Figure.4-2 Order Generation and Cash-in Process

当生成许多候补订单时，客服内网会通过功能接口调用库形成候补订单的两个队列，一个是车次队列另一个是所有候补订单用户队列。候补兑现的过程是首先对车次队列进行轮询，并查看库中是否有相应车次余票。其次，如果有余票那

么将对 A 用户队列进行轮询，依次查找可以满足该余票需求的候补订单用户并将该用户从队列删除（如用户 3），之前未满足条件的用户将按照顺序移至队列 B（如用户 1、2）以此在下一轮轮询时能够保证先提交候补订单的用户依然可以排在顺序前列。

如果在库中查询相应车次没有余票，则会一直对车次队列轮询直到候补购票订单所规定的有效截止时间。当一个候补订单无论兑现成功与失败，候补订单数据库与查询数据库中存储的候补订单状态信息最终都会被修改为相应状态，以便后续对候补购票订单数据进行统计。

4.2 候补购票服务系统功能实现

4.2.1 系统入口及展示设计

无票候补购票服务的入口是 12306 网站及 APP 余票查询功能，用户可根据自己的实际需求，选择具体的车次和席别。为提升用户体验，采用购物车的形式存放用户当前的候补订单，如图 4-3 所示：



图 4-3 购物车展示形式

Figure.4-3 Shopping cart display form

购物车内的所有候补需求构成 1 个候补订单。系统针对购物车里的用户候补需求进行排重处理，校验购物车内候补需求上限。1 个候补订单内，要求发到站同方向，即发站同城、到站同城，要求同一乘车日期，允许多个不同席别。用户向

购物车内添加候补需求时，最多添加 4 个，即 1 个候补订单中，可最多选择 2 个“同一乘车日期+站车次+席别”的组合，可以选择 2 个相邻的日期。当用户购物车内有候补需求时，不允许修改乘车日期或发到站区间。当用户清空购物车之后，可以更改乘车日期和发到站区间。购物车数据不存入数据库，用户退出登录或者关闭浏览器，则购物车信息清空。用户确定自己的候补需求后，点击“添加乘客”按钮，如图 4-5 所示：

请支付预付款，支付剩余时间：28分53秒

候补列车信息

车次	日期	始发站	终到站	时间	席别	金额
G143	2019年6月6日	始 北京南	终 上海虹桥	07:50 - 13:12	一等座	¥933.0
G111	2019年6月6日	始 北京南	终 上海虹桥	08:35 - 14:22	商务座	¥1748.0
G5	2019年6月7日	始 北京南	终 上海	07:00 - 11:40	二等座	¥558.0
G1	2019年6月7日	始 北京南	终 上海虹桥	09:00 - 13:28	二等座	¥553.0

截止兑现时间：2019-06-06 19:00

序号	姓名	证件类型	证件号码	票种
1				

应付总金额 ¥1748.0

- 候补购票订单生效后，相关候补需求不可修改，如需变更，可终止订单后重新操作。
- 完成候补预付款交付后，候补购票订单立即生效，兑现时，按照候补订单生效的时间顺序，优先兑现符合条件的候补需求。
- 预付款按本单各候补需求中票款最高的额度计算（卧铺按下铺票价计算）；候补购票成功后，预付款大于实际票款的，系统将自动退回差额票款。用户主动终止或系统自动终止候补的，系统自动原额退还预付款。

取消订单 立即支付

图 4-4 添加乘客展示形式

Figure.4-4 Add passengers display form

用户需要确认候补订单信息、乘客信息，确认无误后，点击“提交订单”按钮。

## 4.2.2 候补订单处理逻辑设计

这里主要应用了排队理论，对用户提交的候补订单进行处理。用户提交候补订单时，需要首先经过候补排队系统。相比于既有排队系统，候补排队系统主要是为了缓冲巨大的瞬时并发访问压力，降低对候补订单数据库的冲击。候补排队系统不访问路局中心票库，不是扣票操作的必经之路。

为提高候补排队系统的高可用性和并发处理能力，采用 Master-Slave 主从模式增强候补排队系统的高可用性。采用多消费端（Processor）的运行方式提高并发处理能力。主用存储为磁盘，当发生服务器重启、磁盘 ReadOnly 等异常情况时，原始数据不丢失。

用户提交候补订单后，经过候补排队系统，就会生成待支付状态的候补订单，此时系统后台向电子票数据库写入“待支付”状态的候补订单信息，页面跳转至支付倒计时页面。

为实现候补队列的相关需求，需要计算影响客票发售的相关业务数据，并满足定义数据变化时及时触发相关数据自动更新。结合客票发售业务所需要的参数类定义，和候补队列需求，实现候补队列的主要功能点如下：

(1) 建立候补需求表

该表来源于候补队列的全部需求集合，将需求按照具体的始发日期、全车次、席别进行合并，统一计算具体席别的车票需求数量，生成候补需求表。

该表用于归纳有候补需求的车次，能够有效完成需求收敛，避免全路所有车次、日期、发到站组合造成的计算量过大、范围盲目等现象。

该表主要字段如下，此表作为实现余票计算过程的入参：

表 4-1 需求字段  
Table.4-1 Demand field

名称	字段名	类型	值域
始发日期	train_date	char(8)	Not Null
始发车次	train_code	char(12)	Not Null
上车站	from_station	char(3)	Not Null
下车站	to_station	char(3)	Not Null
席别	seat_type_code	char(1)	Not Null
有座无座标记	w_flag	char(1)	Not Null
总张数	total_num	int	Not Null
上铺数	up_num	int	Null
中铺数	mid_num	int	Null
下铺数	down_num	int	Null
生成时间	operate_time	datetime	Not Null

(2) 建立共用关系缓存表

客票发售时，后方站何时可共用前方站余票、可共用的具体用途，是由共用定义、共用策略、共用例外定义共同作用实现的。一般来说，该定义变化周期较



长。针对上述三类定义，生成共用关系缓存表，能有效缩短共用定义的计算时长。该表主要字段如下：

表 4-2 关系缓存字段

Table.4-2 Relational cache Fields

名称	字段名	类型	值域
起日期	start_date	char(8)	Not Null
止日期	stop_date	char(8)	Not Null
始发车次	train_code	char(8)	Not Null
始发站名	start_station_name	char(10)	Not Null
终到站名	end_station_name	char(10)	Not Null
始发站电报码	start_station_telecode	char(3)	Not Null
终到站电报码	end_station_telecode	char(3)	Not Null
车厢	coach_no	char(2)	Not Null
席别	seat_type_code	char(1)	Not Null
无座席标记	wseat_flag	char(1)	Not Null
用途	purpose_code	char(2)	Not Null
被共用站	old_assign_station	char(3)	Not Null
共用站	new_share_station	char(3)	Not Null
提前共用小时数	pre_saletime	int	Not Null
乘车日期	train_date	char(8)	Not Null
共用站开车时间	start_time	char(4)	Not Null
共用日期	share_date	char(8)	Not Null
共用时间	shart_time	char(5)	Not Null
缓存更新时间	operate_time	datetime	Not Null
更新标记	operate_flag	char(1)	Not Null

### (3) 建立取消限售缓存表

席位带有以远站属性，客票发售时，何时可以突破以远站限制，发售短途车票是由取消限售定义决定的。一般来说，该定义针对不同用途，时间设置有很大差别，且允许批量定义。针对此类定义，生成取消限售缓存表，能有效缩短取消限售定义的计算时长。该表主要字段如下：

表 4-3 取消限售缓存字段

Table.4-3 Cancel restricted cache fields

名称	字段名	类型	值域
起日期	start_date	char(8)	Not Null

止日期	stop_date	char(8)	Not Null
始发车次	train_code	char(8)	Not Null
始发站名	start_station_name	char(10)	Not Null
终到站名	end_station_name	char(10)	Not Null
始发站电报码	start_station_telecode	char(3)	Not Null
终到站电报码	end_station_telecode	char(3)	Not Null
车站	station_telecode	char(3)	Not Null
取消限售时间	ahead_time	int	Not Null
席别	seat_type_code	char(1)	Not Null
用途	purpose_code	char(2)	Not Null
列车类型	train_type	char(1)	Not Null
售票方式	sale_mode	char(1)	Not Null
缓存更新时间	operate_time	datetime	Not Null
更新标记	operate_flag	char(1)	Not Null

#### (4) 建立候补队列兑现关系表

按照候补需求表中的车次及发站，生成后续每一站的共用特征及取消限售特征，计算出满足此特征的余票数量、车厢范围。应用服务根据候补需求，对照候补队列兑现关系表逐条兑现。该表主要字段如下：

表 4-4 队列兑现字段

Table.4-4 Queue cash fields

名称	字段名	类型	值域
始发日期	start_date	char(8)	Not Null
乘车日期	train_date	char(8)	Not Null
全车次	train_no	char(12)	Not Null
用途	purpose_code	char(2)	Not Null
上车站	board_station	varchar(256)	Not Null
以远站	far_from_station_no	char(2)	Not Null
限售站	limit_station	char(2)	Not Null
席别	seat_type_code	char(1)	Not Null
有座无座标记	w_flag	char(1)	Not Null
总张数	total_num	int	Not Null
上铺数	up_num	int	Null
中铺数	mid_num	int	Null
下铺数	down_num	int	Null
生成时间	operate_time	datetime	Not Null

完成兑现后，候补队列兑现关系表中的总张数逐次减少，同时候补需求表中已兑现的需求标记为已兑现。实时调用存储过程后，生成上表，是结合了调令、共用定义、取消限售等定义的余票记录表，按照以远站合并后的结果。结合旅客的候补需求，读取此表决定是否能够取票。取票时需要遵循以下规则记录兑现张数。如存在以下余票记录：

表 4-5 余票记录字段  
Table.4-5 Left tickets record fields

上车站	以远站	限售站	张数
02 03 04 05	05	10	20
03 04 05	05	10	30
05	05	10	30

#### 4.2.3 候补订金及结账对账

用户提交候补订单后，进入到支付页面，如下图所示：



图 4-5 支付页面

Figure.4-5 Payment page

支付渠道与既有互联网售票保持一致。用户需要支付当前候补订单中最高车次席别票价。例如，用户选择 12 月 6 日北京南-上海虹桥 D103 次车的高级软卧、

动卧、二等座，那么用户需要支付高级软卧席别的乘车人数量对应的票款。

用户支付候补订金后，如果主动取消候补订单或候补订单到期无法兑现，则系统自动退还全部候补订金。如果兑现成功，且兑现席别票价低于预付票款，则系统自动退还票款差价。对于用户主动取消订单的情况，可以直接在候补服务中调用支付平台完成退款。对于到期无法兑现的情况，系统通过定时任务的方式在数据库中检索符合条件的候补订单，并发起退还候补订金、检查车次加锁等操作。

结账统计：针对候补订金的收款和退款，在铁路局互联网售票结帐统计中，增加统计报表。

对账：候补订金的收款和退款明细，标识后随票款交易明细，T+1 发送支付平台，支持对账处理。

#### 4.2.4 候补需求整合处理

用户提交候补订单并完成支付以后，系统按照预先设定的规则对订单进行整合处理，根据支付回调信息分别更改电子票数据库中的候补订单状态等信息，并将该“用户名+候补订单号+银行卡号”信息推送至消息队列中，由候补兑现服务消费后，再次查询出支付后候补订单信息，并写入数据库。这种通过反查电子票库候补订单信息更新数据库候补订单信息的方式，可以保障中已支付订单数据的准确性。

#### 4.2.5 候补订单

##### (1) 候补订单的构成

候补订单保存有用户提交的候补订单详细信息，其构成包括列车相关信息、用户相关信息、订单相关信息等。

##### (2) 候补订单的查看与取消

用户的候补订单兑现之后，用户可在“候补订单”页面中查看未支付、待兑现、兑现后的候补订单，如下图所示：



图 4-6 订单查看与取消

Figure.4-6 Order viewing and cancellation

未支付订单：用户需要在限定时间内（如 30 分钟）完成支付操作，否则系统将自动取消该候补订单。

待兑现订单：用户可以随时取消。如果用户取消了待兑现候补订单，系统将会在指定时间内发起退还候补订金。

兑现后订单：用户可以查看到兑现成功或兑现失败的候补订单。

### (3) 候补订单数据的物理存储

候补订单数据存储在电子票库和查询库两个位置。候补订单详细信息收集写入电子票数据库，作为可信数据源；同时写入查询数据库，提高后续查询效率，降低电子票数据库访问压力。

在候补订单电子票数据库中，存储候补订单全部字段信息，承担订单（存根）信息管理、候补订金信息维护、加锁信息管理、故障单信息管理、日志记录等功能。在既有电子票库中新建分库，采用 Hash 存储方式，可以更加均衡存储数据、避免热点。数据存储在 A 层，进行双中心复制，向 AT 层复制，通过 ctmsx+MQ 向 GemFire 订单查询集群复制。

在候补订单查询数据库中，包含详情表和拆分表，存储候补订单主要字段信息，承担订单主要信息查询和兑现功能、计数器功能。兑现时，根据候补余票数

量，以候补余票查询入参（始发日期+始发车次+席别）为条件从拆分表中分页取出一批候补订单，逐个轮询匹配，寻找符合条件的候补订单号和用户名，返回给应用，再到候补订单详情表中查询乘车人信息，进行扣票，按照支付时间进行升序排序。

#### (4) 兑现成功率

在车票售罄之后，可能产生新票的方式包括：

- 1) 退票、改签操作产生的返票
- 2) 取消限售、转用途产生的车票
- 3) 共用、复用产生的车票
- 4) 临时加挂、新增临客产生的车票

以上 4 种方式产生的新票日志、订单等数据在客票中心、营销数据库中均没有完全记录的数据。其中，返票（退票、改签、废票、取消票）转为可售状态没有日志记录；取消限售数据在路局中心有日志，数据在 AD 数据库中；共用数据存储在 AD 数据库；临时加挂、新增临客数据在 AD 数据库中有日志表；营销数据库中只有售票、退票、改签、废票存根等统计数据。

系统后台估算候补需求的兑现成功率，计算的依据是历史售票情况和当前候补排队情况。对兑现成功率进行模糊化处理，以“较高、较低”的字样提示给用户，可以在用户支付前页面进行预估成功率的提示。

根据候补购票兑现成功率的定义，在考虑当前所掌握的历史数据及未来可采集数据的情况下，目前主要提出以下三种实现方案。

##### 1) 方案一：

根据目前营销数据库中掌握的历史交易数据，可通过聚类、分类等方法统计分析旺季时期、热门线路列车 OD 售票时间分布情况。若旅客出行旺季时期或热门线路，票额售出时间段呈现明显的差异化分布，售出时间距离开车时间越近，为转用途的新票的概率越大。

##### 2) 方案二：

在车票售罄后，对新产生新票的方式记录日志，记录新票在转用途前的状态、数量等数据，并每日收集汇总统计至大数据集群。根据这类数据，可准确计算转

用途后生成的新票在总售出票数当中的比例。

### 3) 方案三:

记录列车售票的起售时间（目前客票所有数据当中，均没有对列车售票起售时间的定义），统计分析列车售出时间和起售时间间隔  $N$  的分布情况，估算一个  $N$  值，定义时间段  $N$  之后售出的车票为车票售罄后的新票。

上述三种方案，都是针对历史数据进行统计分析得到新票的概率。计算候补队列中旅客成功买到新票的概率，还需要看候补总体需求情况。旅客候补需求张数不超过新票张数时，才有较大概率买到新票，旅客需求排队越靠后，候补成功率越低。

根据目前掌握的数据情况，可优先采用第一种方案初步分析列车票额售出时间的分布情况。候补兑现成功率的表现形式如图 4-7 所示。



图 4-7 兑现成功率表现形式

Figure.4-7 Cash success rate display form

该方案的实施过程如下：首先通过计数器统计库中存储的同一车次相同席别的候补订单数，记该数为  $N_{\text{提交}}$ ，其次将目前限定的一个车次的最大允许候补人数记为  $N_{\text{上限}}$ ，此时当前排队人数百分比  $\eta$  的计算方式为：

$$\eta = \frac{N_{\text{提交}}}{N_{\text{上限}}} \quad (1)$$

假设一个车次的二等座起售时间为 2018 年 1 月 1 日，起发时间为 2018 年 2 月 1 日，那么每当有 12306 用户在该时间段提交候补订单， $N_{\text{提交}}$  则加 1，所以排队人数百分比  $\eta$  是动态变化，将  $\eta$  值分为三个不同区间，根据  $\eta$  大小来决定 12306 界面对用户显示的候补人数为较少、中等还是较多。由此，优先提交候补订单的

用户将会看到 12306 界面候补人数较少的提示，提交顺序越靠后的乘客将会看到 12306 界面候补人数较多的提示。

#### 4.2.6 候补余票计算

候补余票计算的作用在于得知当前候补需求是否存在余票。为满足此功能，需要新增候补余票计算功能。候补轮询服务根据用户需求作为候补余票查询条件，经过计算之后。输出当前“始发日期+始发车次+席别”条件下，各个有余票数量的停靠站组合。

例如：如 G1 北京-上海，有旅客候补北京-济南车票，已放开限售及允许共用其他车站车票，此时有北京到上海的退票，则旅客可在界面上查询到除北京到济南外的其他区间有票，也可购票。但 G1 北京到济南需全部候补需求兑现完成后才正常显示余票。

候补余票计算服务处于候补购票服务的关键位置，为候补需求的兑现提供票额。为提高并发处理能力，降低系统改造难度，候补余票计算服务仍然使用分布式内存数据库集群。

#### 4.2.7 候补订单兑现规则设计

候补订单轮询及兑现服务是整个候补购票服务中承上启下的环节。

候补订单轮询服务消费候补需求队列中的“始发日期+始发车次+席别”组合，以此作为入参调用候补余票计算服务，在得到输出结果以后，到候补订单数据库拆分表中，按照支付时间先后顺序分页获取一批用户提交的候补需求简要信息（包含当前候补需求中，用户需要的票数），之后匹配余票数量和候补需求票数，寻找符合兑现条件的元素。在兑现过程中，遵循以下原则：

- （1）先到先得；
- （2）优先兑现符合余票数量的候补需求；
- （3）不支持部分兑现。

在完成兑现以后，需要进行一系列后续操作，包括：生成已兑现订单、通知



旅客完成兑现、调用支付平台完成差价退款、回改候补缓存中的信息等。

调图情况不对之前需求进行兑现，旅客自行再次提交候补需求。

目前的兑现方式对候补需求中车票少的用户有利，而且给候补产生的票额也较小，因此可以考虑积攒一批车票再放出来，尽可能保证公平。

#### 4.2.8 各渠道余票查询及售票控制

根据候补购票服务的定位，票额优先分配给候补购票服务中用户提交的需求。所有线上（网站、APP）和线下（窗口、电订、TVM、自动售、代售点等）售票渠道在查询余票之前，需要首先判断是否存在相应的候补需求。

在 AD 数据库中建立余票加锁表。对于普通车次：以乘车日期、始发车次、始发/终到站、乘车区间、席别、用途（仅互联网使用的用途）为单位，线上和线下余票查询及扣票过程中，需要判断是否存在加锁情况，如果有，返回无票。

加锁表数据通过电子票数据库复制机制向各个路局中心实时传输数据；通过查询数据库复制+ctmsx+MQ 机制向 GF 集群实时传输数据。

为降低对 AD basic 库中余票加锁表的访问次数，在数据库中同样设置余票加锁表。

##### （1）加锁数据的生成

用户提交候补订单并支付成功后，候补系统首先到数据库中查询是否存在余票加锁数据，若存在，则不做操作；若不存在，则向 AD 数据库中添加余票加锁数据。

##### （2）加锁数据的删除

情况 1：每个候补需求兑现完成后，都要到数据库中检查一下“始发日期+始发车次+席别+上车站电报码+下车站电报码”这个条件，判断是否还有候补拆分表数据，如果有，则不做操作；如果没有，则以此条件删除加锁表数据。

情况 2：没有候补需求，但开始执行兑现。这时拆分表查不到数据，认为当前“始发日期+始发车次+席别”没有要兑现的候补需求，需要删除加锁表数据，以此为条件，删除数据。

### 4.2.9 其他功能

#### (1) 通知信息发送

在用户候补订单兑现成功、到期无法兑现、存在业务异常等情况下，系统将会通过微信、短信、支付宝等方式通知用户（以用户选择的通知方式为准）。候补购票服务调用通知系统发送接口，并发送相应信息。

#### (2) 客服相关

为配合客服处理旅客咨询投诉，向客服提供根据用户名、证件号等信息查询候补订单信息的接口。客服桌面系统通过 http 方式调用综合查询平台接口服务，相应内容中如有敏感数据根据业务要求进行加密。

#### (3) 监控报警

系统硬件 CPU、内存等运行状态通过既有一二中心监控系统监控，针对业务运行状态，通过松耦合的方式集成，动态对数据进行采样，支持数值、柱状图、饼状图、环状图、雷达图、折线图、事件序列图等多种直观的展现方式。业务监控指标包含汇总及分渠道的发送速率、发送成功率、响应时间数据等，针对各项系统及业务监控指标，设置合理的报警阈值。

## 4.3 应急处理机制

系统在架构设计和技术选型时，充分考虑高可用性，建立较为完善的应急处理机制，各个环节采用集群或主备设计节点，提高系统高可用性，避免单点风险。系统具备自动化的故障发现、级别判定、应急处理能力。

#### (1) 候补排队故障应急

候补排队服务采用主备运行的方式，当某个在线运行候补排队服务故障时，迅速切换至备用系统运行。对于已经在排队过程中的请求，系统将会返回提示信息，引导旅客进行重试。为避免对电子票库造成较大冲击，故障时一般不会切换至同步模式。

## （2） 候补队列故障应急

候补队列服务采用集群部署（数据分布式存储、运算分布式执行）、主备运行的方式。当某个在线运行候补队列服务整体出现故障时，迅速切换至备用系统运行。当某个集群节点故障时，会影响存储在当前节点上的数据存储和运算操作，此时可以立刻启用故障节点的从节点，数据不丢失，运算会有闪断，但是用户无感知。

## （3） 候补轮询及兑现故障应急

候补轮询服务采用主备运行的方式，当某个在线运行候补排队系统故障时，迅速切换至备用系统运行，候补兑现服务采用集群部署（节点对等），多节点并行运行的方式，当某个服务节点故障时，不影响其他正常运行节点，以上两种情况皆可满足用户无感知。

## （4） 候补余票计算故障应急

候补余票计算服务采用集群部署（数据分布式存储、运算分布式执行）、多集群并行运行的方式。当某个在线运行候补余票计算服务整体出现故障时，迅速切换至其他集群运行。当某个集群节点故障时，会影响存储在当前节点上的数据存储和运算操作，数据不丢失，运算会有闪断。

## （5） 候补订单故障应急

候补订单故障时，如果某一分库或节点发生故障，则进行“用户迁移”操作，将当前分库或节点的用户的新增候补订单迁移至其他已有分库或节点，为避免异常范围扩大，需停止候补轮询和兑现服务。

## （6） 通知信息发送故障应急

通知系统将会具备双中心主备运行模式，具备主备切换能力。对于候补购票服务来说，由于用户已经预付票款，兑现成功或到期无法兑现时，如果通知消息

无法及时送达用户，那么用户可使用候补订单查询功能查看候补订单状态。

## 4.4 扩展机制

系统设计考虑良好的可扩展性，满足现有需求的同时，充分考虑远期互联网售票在访问量、交易量和用户数增长以及应用系统的增加带来的候补需求数量快速增长情况，相关日志分析能力、日志持久化存储能力需具备良好的可扩展性，满足业务发展的需要。

## 4.5 安全方案

### (1) 网络安全

涉及用户信息、乘车人信息、候补订单信息的应用部署于 12306 客服内网，通过安全平台确保网络安全。

### (2) 主机安全

相关主机设备安全配置均需要经过安全测评，并且在主机进行任何操作都应保留记录，确保故障可追踪。

### (3) 应用安全

#### 1) 应用权限管理

后台应用涉密数据和配置部署在客服内网，需要通过安全平台进行数据跨网络访问。通过严格的权限管理来进行访问控制，包括角色管理、权限分层授权和授权记录审计。针对不同的业务信息采取隔离方式，避免互相影响。

#### 2) 容灾和高可用

关键系统模块建立冗余系统，缓存、消息队列、数据库采用主从方式运行，应用服务采用多活方式运行。建立定期备份和应急切换机制，提高应用的容灾能力和故障恢复能力，提升应用系统的高可用性。

## 4.6 本章小结

本章主要进行了候补购票服务系统的搭建与实现，在确定设计原则的基础上，提出整体架构设计，分业务分模块对候补购票服务系统的各个功能进行相应实现。除此以外，还对系统所必须具备的应急处理机制、扩展机制以及安全方案进行设定，至此候补购票服务系统完全实现。



## 5 候补购票服务系统应用

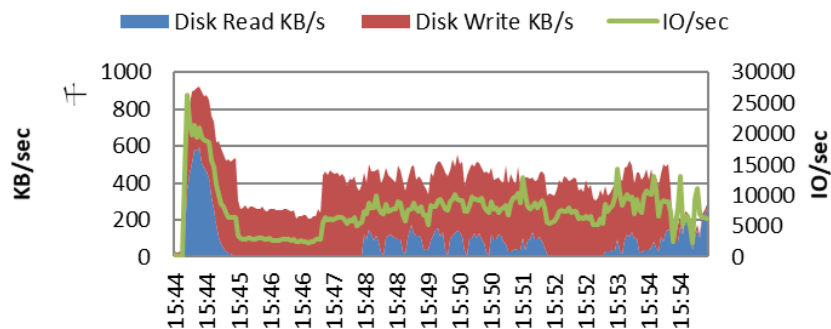
### 5.1 候补购票服务系统模拟测试

对候补购票服务系统进行了严谨的分业务模式的模拟测试，对于单个业务场景模拟测试有着 14 种不同场景，主要为插入候补订单、支付后更新候补订单状态、更新候补订单的截止兑现时间、成功率估算、短信提醒当前排队订单数等等。

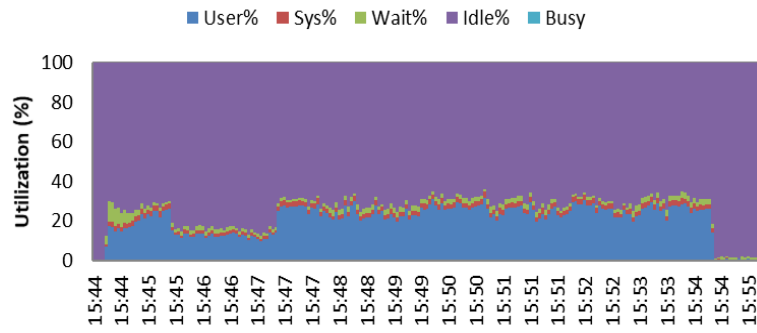
图 5-1 为插入候补订单模拟测试的系统 TPS、IO 以及 CPU 占用率变化情况，测试时间为 10 分钟，模拟了 40 个客户端以及 40 个多线程。插入候补订单业务的平均延迟为 4.027ms，延迟的标准差为 21.118ms，TPS 达到 9932.45 的水平。在系统 IO 方面以及 CPU 利用率方面也基本满足上线要求。其余单个业务场景模拟测试结果也都达到上线水平，不再赘述。

```
scaling factor: 1
query mode: prepared
number of clients: 40
number of threads: 40
duration: 600 s
number of transactions actually processed: 5959484
latency average = 4.027 ms
latency stddev = 21.118 ms
tps = 9932.450740 (including connections establishing)
tps = 9932.660603 (excluding connections establishing)
script statistics:
- statement latencies in milliseconds:
  3.997 insert into hb10_reserve_list_details
```

(a) TPS 及相关参数测试结果



(b) 系统 IO 测试结果

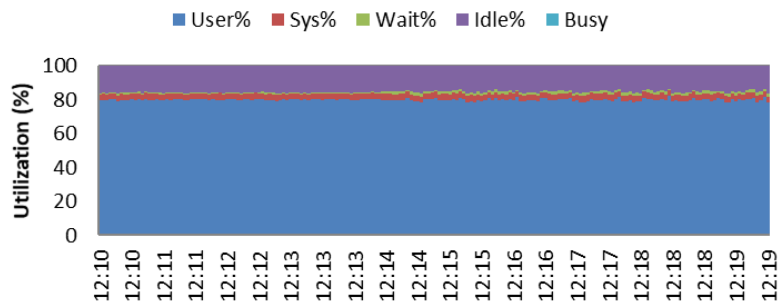


(c) CPU 利用率测试结果

图 5-1 插入订单模拟测试结果

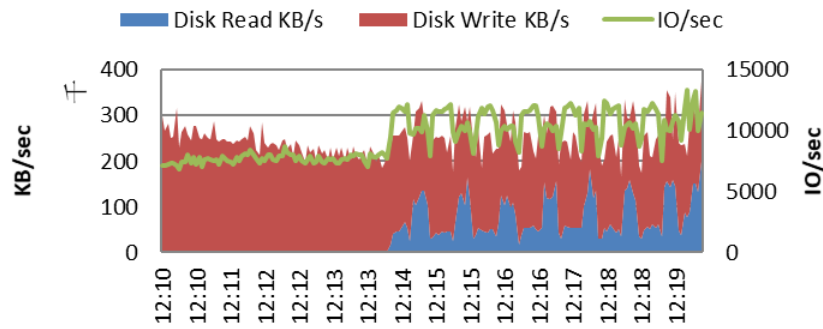
Figure.5-1 Pressure test results of orders

对于业务场景混合的模拟测试有着 8 种不同场景，主要为候补订单详情表混合均衡测试、读操作占大比例测试、写操作占大比例测试以及候补订单拆分表的混合读操作占大比例测试、写操作占大比例测试和更新操作占大比例测试。图 5-2 为候补订单详情表混合业务均衡测试，四个混合业务分别为根据用户名查询候补订单、插入候补订单、更新候补订单状态以及定时删除候补订单，四个业务量的占比分别为 40%、30%、20% 以及 10%，根据用户名查询候补订单、插入候补订单、更新候补订单状态测试的 TPS 分别为 82925、4492、3838，基本达到上线水平，并且 IO 变化以及 CPU 利用率也符合标准。其余混合业务测试结果也都符合标准，并且十分稳定。



(a) CPU 利用率测试结果





(b) 系统 I/O 测试结果

图 5-2 详情表混合业务均衡测试结果

Figure.5-2 Details table mixed business balance test results

## 5.2 候补购票服务系统应用

在完成调研、需求分析、总体方案设计之后，技术团队完成了研发工作。经过严谨充分地测试工作之后，候补购票服务系统于 2019 年春运期间在北京、上海、广州、成都铁路局进行试点运行，按照节前节后分方向试点。2018 年 12 月 27 日，12306 候补购票项目试点运营，截止 2019 年 2 月 21 日 20 点，累计已支付候补订单 478530 笔，终止兑现 268447 笔(其中主动退单 165949 笔，自动退单 102498 笔)，兑现失败 44228 笔，成功兑现 207667 笔，去除主动退单与自动退单的净候补订单成功兑现率为 66%，总候补订单成功兑现率为 43%。

在春运试运营期间候补购票项目优势已经开始彰显，如图 5-3 以及图 5-4 所示为自 2018 年 12 月 29 日起春运期间部分热门正常车次与部分热门候补车次提交申请订单平均值分布状况。图 5-3 中候补车次申请订单数目比较平稳，正常车次申请订单数目波动较大。图 5-4 中候补车次的提交申请的失败率更低。两图的对比结果说明第三方平台对候补车次带来的刷票压力极大减小，基本达成无票候补购票项目的最主要目的。以下两图中的车次都具备普化效应绝非特例。

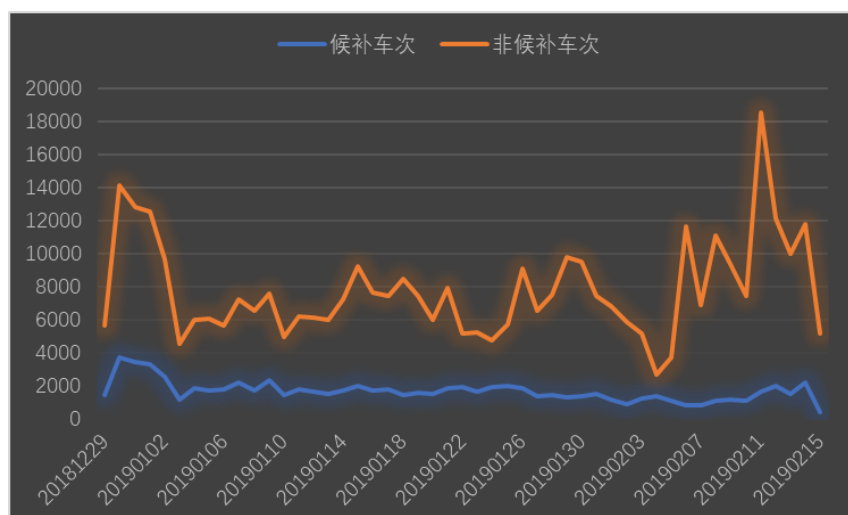


图 5-3 提交订单对比

Figure.5-3 Contrast of submission orders

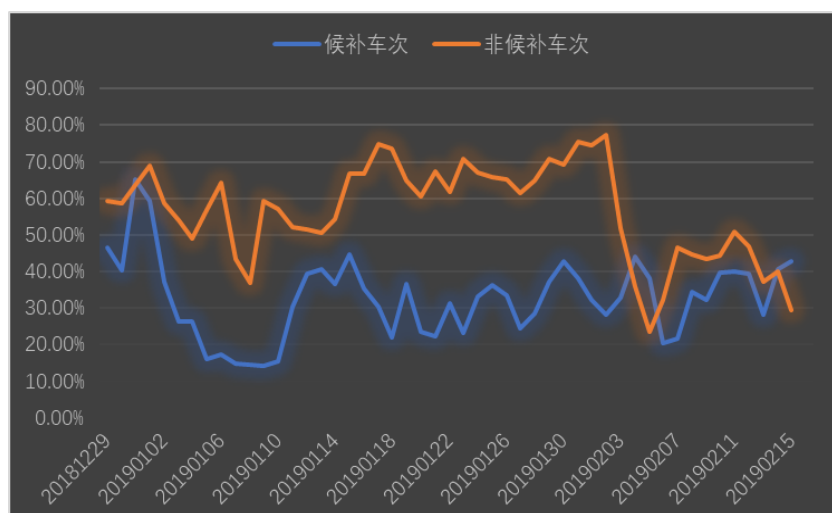


图 5-4 兑现失败对比

Figure.5-4 Contrast of cash failure

在 12306 用户使用候补购票体验反响方面，候补系统极受欢迎。通过在微博、微信上的调查，用户对候补购票系统普遍接受并赞扬，且有强烈的普及愿望，这极大的打击了第三方刷票软件，方便旅客出行。

### 5.3 本章小结

本章主要进行了候补购票服务系统的模拟测试以及实际应用，分别对候补购票服务系统的单个业务场景以及混合业务场景进行测试，测试结果显示的系统 TPS、IO 以及 CPU 利用率都符合上线标准。2019 年春运期间对候补购票服务系统

的试点应用有着平均 66% 的高成功兑现率，在一定程度上打击了第三方刷票软件，证明该系统具有重要意义。



## 6 总结与展望

### 6.1 本文研究总结

本文首先介绍了我国高速铁路和 12306 售票系统的发展历程，因运力紧张导致的票额不足和 12306 互联网售票系统功能不够完善，导致一些铁路旅客转向使用第三方代购网站从而给 12306 互联网售票系统带来较大的压力。为了解决以上问题，本论文在以下方面进行研究：

（1）结合铁路客票系统的发展综述，确定了基于 12306 客票系统候补购票服务系统方案的必要性，确定以开源分布式内存数据库技术为基础的 12306 客票系统候补购票服务系统方案作为论文的主要研究对象。

（2）研究分析目前主流的开源分布式内存数据库技术，明确主流的开源分布式内存数据库各自的技术特性及其在候补购票服务系统中的适用场景。

（3）针对基于 12306 客票系统候补购票服务系统方案作需求分析，主要从业务需求、系统需求、安全需求以及网络需求方面进行分析，从而明确候补购票服务系统方案的功能。

（4）在明确 12306 客票系统候补购票服务系统方案需求的基础上，进行总体系统设计，形成 12306 客票系统候补购票服务系统方案的技术框架、建立相应的应急处理机制、确定适应长期发展的扩展机制以及保证系统整体安全的安全方案。

（5）将建立的候补购票服务系统进行应用，验证 12306 客票系统候补购票服务系统方案的实用性和可行性。

### 6.2 展望

候补购票服务系统目前取得了一定的应用效果，在新技术和新需求的推动下，应当逐步优化系统，使其能够在便民利民、节约开支等方面发挥更大作用。本文仍然存在一些不足，后续工作可以从下面展开：

（1）在候补购票服务系统中，兑现成功率的计算方法有待提升，可以使用更加精准的算法为旅客提供量化兑现成功率计算。

（2）目前候补购票服务系统仅在部分区间试点运行，可以向全国推广。

（3）由于候补购票服务系统仍处于试运行阶段，尚未明确上限，需要根据运行情况不断总结分析，以便对系统构成、业务指标进行灵活调整。

## 参考文献

- [1] 朱建生, 马钧培, 史天运等. 铁路客票发售和预订系统升级的可行性研究[C] 国际节能与新能源汽车创新发展论坛. 2009.
- [2] 于澎, 朱建生. 基于人工神经网络的铁路客票系统中转路径选取研究[J]. 铁道运输与经济, 2010, 32(3):42-46.
- [3] 汪健雄, 刘春煌, 单杏花等. 基于双层次正交神经网络模型的铁路客运量预测[J]. 中国铁道科学, 2010, 31(3):126-132.
- [4] 吕晓艳, 刘春煌, 朱建生. 基于关键度度量的决策树算法改进及其在铁路运输中的应用[J]. 铁道学报, 2011, 33(9):62-67.
- [5] 朱建生. 新一代客票系统总体技术方案的研究[J]. 铁路计算机应用, 2012, 21(6):1-6.
- [6] 王洪业, 吕晓艳, 朱建生. 铁路客票系统售票组织管理自动化研究[J]. 铁路计算机应用, 2012, 21(4):30-32.
- [7] 王红爱, 吕晓艳, 朱建生等. 基于粒子群算法的铁路客票预售期研究[J]. 铁道运输与经济, 2012, 34(3):76-79.
- [8] 王拓, 朱建生, 王明哲. 分布式内存数据库在铁路互联网售票余票查询中的应用研究[C] 中国智能交通年会. 2013.
- [9] 王红爱, 朱建生, 刘文韬等. 铁路客票系统中缓存机制的应用模型研究[J]. 铁路计算机应用, 2013, 22(2):30-33.
- [10] 朱建生, 汪健雄, 张军锋. 基于 NoSQL 数据库的大数据查询技术的研究与应用[J]. 中国铁道科学, 2014, 35(1):135-141.
- [11] 邢国军, 王明哲, 朱建生等. 铁路 12306 手机售票系统的研究与实现[C] 2014 中国智能交通年会大会. 2014.
- [12] 沈海燕, 贺晓玲, 朱建生. 铁路信息系统运维管理信息化需求分析[J]. 中国铁路, 2014(10):25-28.
- [13] 朱建生, 王明哲, 杨立鹏等. 12306 互联网售票系统的架构优化及演进[J]. 铁路计算机应用, 2015(11):1-4.
- [14] 单杏花, 王富章, 朱建生等. 铁路客运大数据平台架构及技术应用研究[J]. 铁路计算机应用, 2016, 25(9):14-16.
- [15] 张圣, 朱建生, 王洪业. 铁路大站票额预分预测模型研究[J]. 铁道运输与经济, 2017, 39(6):27-31.
- [16] 张圣, 周亮瑾, 吕晓艳等. 铁路售票组织策略探讨[C] 中国智能交通年会. 2016.
- [17] 于澎, 刘文韬, 朱建生等. goTicket 性能测试平台的开发与应用[J]. 铁路计算机应用, 2015(6):1-4.
- [18] 李丽辉, 朱建生, 强丽霞等. 基于随机森林回归算法的高速铁路短期客流预测研究[J]. 铁道运输与经济, 2017, 39(9):12-16.

- [19]李丽辉, 朱建生, 徐彦等. 京沪高速铁路客流空间分布预测方法研究[J]. 铁道运输与经济, 2017, 39(6):32-36.
- [20]张永, 朱建生, 吕晓艳等. 高铁列车停靠站客票收入率分类及收入预测研究[J]. 铁道学报, 2018, 40(3):24-28.
- [21]郭慧峰. 基于卷积神经网络的 12306 验证码识别[D]. 华南理工大学, 2017.
- [22]田朝辉, 金鑫, 赵耿等. 基于深度学习的物体识别验证码破解方法[J]. 计算机仿真, 2018.
- [23]陈思茹. 基于电子商务视角下的交通运输订票网站管理研究—以 12306 网站验证码为例[J]. 信息化建设, 2016(4).
- [24]王雪峰, 王明哲, 朱建军等. 如何防范针对 12306 网站的恶意抢票插件[J]. 铁路计算机应用, 2015(3).
- [25]张砚枫, 寇玉龙, 金玉. 基于自然图像混合的验证码[J]. 辽宁科技学院学报, 2017(1).
- [26]吴广. 网易火车票订票系统的设计与实现[D]. 北京交通大学, 2014.
- [27]秦秀磊, 张文博, 魏峻等. 云计算环境下分布式缓存技术的现状与挑战[J]. 软件学报, 2013.
- [28]吕明育, 李小勇. NoSQL 数据库与关系数据库的比较分析[J]. 微型电脑应用, 2011, 27(10).
- [29]王德文, 杨力平. 智能电网大数据流式处理方法与状态监测异常检测[J]. 电力系统自动化, 2016, 40(14):122-128.
- [30]陆鑫, 王艳蓉, 孙超等. 一种高性能多模式的内存数据库系统[J]. 计算机应用与软件, 2019(1).
- [31]杨艳, 李炜, 王纯. 内存数据库在高速缓存方面的应用[J]. 现代电信科技, 2011, 41(12):59-64.
- [32]林金芳, 欧锋. 分布式混合数据库系统[J]. 计算机系统应用, 2015, 24(10):32-37.
- [33]杨立鹏, 梅巧玲, 陈爱华等. 铁路互联网售票系统的研究与实现[J]. 铁路技术创新, 2012(4):32-34.
- [34]陈吉荣, 乐嘉锦. 基于 Hadoop 生态系统的大数据解决方案综述[J]. 计算机工程与科学, 2013, 35(10):25-35.
- [35]田文洪, 李国忠, 陈瑜等. 一种兼顾负载均衡的 Hadoop 集群动态节能方法[J]. 清华大学学报(自然科学版), 2016(11):1226-1231.
- [36]孟永伟, 黄建强, 曹腾飞等. Hadoop 集群部署实验的设计与实现[J]. 实验技术与管理, 2015, 32(1).
- [37]夏靖波, 韦泽鲲, 付凯等. 云计算中 Hadoop 技术研究与应用综述[J]. 计算机科学, 2015, 43(11):6-11.
- [38]杨来, 史忠植, 梁帆等. 基于 Hadoop 云平台的并行数据挖掘方法[J]. 系统仿真学报, 2013, 25(5):86-94.
- [39]李翀, 刘利娜, 刘学敏等. 一种高效的 Redis Cluster 的分布式缓存系统[J]. 计算机系统应用, 2018, 27(10):95-102.
- [40]曾超宇, 李金香. Redis 在高速缓存系统中的应用[J]. 微型机与应用, 2013, 32(12):11-13.



- [41]葛宝龙, 邓欣, 秦嘉罗. SpringBoot 集成 Redis 缓存在电信运营系统中的应用研究[J]. 中国新通信, 2017(23):106-108.
- [42]马豫星. Redis 数据库特性分析[J]. 物联网技术, 2015(3).
- [43]张敬伟, 丁志均, 杨青等. 异构 Redis 集群大规模评论数据存储负载均衡设计[J]. 华东师范大学学报(自然科学版), 2017(05):29-38.
- [44]韩忠华, 刘春光, 戴敬等. 系统服务和消息队列技术在生产数据采集中的应用[J]. 沈阳建筑大学学报(自然科学版), 2010, 26(4):808-812.
- [45]梅雪峰, 赵文静. 基于消息队列和 Web 服务的分布式系统异步交互方式体系架构[J]. 西北大学学报(自然科学版), 2004, 34(6):655-658.
- [46]陆庆, 周世杰, 秦志光等. 消息队列中间件系统中消息队列与消息分发技术研究[J]. 计算机应用研究, 2003(8):51-53.
- [47]杨超, 徐如志, 杨峰. 基于消息队列的多进程数据处理系统[J]. 计算机工程与设计, 2010, 31(13).
- [48]王小霞, 陈亮. 一种消息队列中间件的设计与实现[J]. 计算机工程, 2005, 31(21).
- [49]马跃, 颜睿阳, 孙建伟. 基于 RocketMQ 的 MQTT 消息推送服务器分布式部署方案[J]. 计算机系统应用, 2018, v.27(06):85-88.
- [50]欧志芳. 基于 RocketMQ 实现异构数据库同步[J]. 网络安全技术与应用, 2016(12).



## 作者简历及攻读学位期间取得的科研成果

### 一、作者简历

[内容采用五号宋体] 包括教育经历、工作经历、攻读学位期间发表的论文和完成的工作等。

### 二、攻读学位期间科研成果

一、参加科研项目					
序号	项目名称	项目来源	项目下达编号	本人承担的任务	
1	12306 综合信息服务系统的研究与实现	院基金课题	2017YJ104	执行负责人	
二、发表学术论文					
序号	发表论文名称	发表时间	发表刊物、会议名称	ISSN 刊物号或 SCI、EI、ISTP 检索号	作者排名
1	基于 Zabbix 的海量服务器监控系统的设计与实现	2017-11-22~26	第十二届中国智能交通年会	否	1
2	基于指数权重算法的铁路互联网售票异常用户智能识别的研究与实现	2018年10期	铁路计算机应用	否	1
三、获得各类科技奖项					
序号	获奖项目	获奖等级	获奖时间	证书号/专利号	颁奖部门
1	基于“双中心双活”的铁路12306互联网售票系统研究与应用	三等	2018		北京市

(说明: 此表如一页未填完可续页)



## 学位论文数据集

作者姓名	李雯	学号	TK1715	
关键词（3个）	密级（公开、内部、秘密、机密、绝密）	中图分类号	UDC	论文选题来源（国家级、省部级、自选）
铁路客票系统；候补购票；内存数据库	公开	TP37	004.6	省部级
学位授予单位	学位授予单位代码	学位类别	学位级别	
中国铁道科学研究院	83801	专业学位	硕士专业学位	
前置学位授予单位	前置学位授予时间	前置学位证书编号	前置学位授予专业	
湖南文理学院	2010年7月	1054942010000827	网络工程	
论文中文题目		论文英文题名		论文语种
12306 候补购票服务系统的研究与实现		Research and Implementation of 12306 Line Up Ticketing System		中文
培养单位名称	中国铁道科学研究院	培养单位地址	北京市海淀区大柳树路2号	
培养单位代码	83801	邮编	100081	
导师姓名	朱建生	导师职称	研究员	
答辩委员会主席	王富章	论文评阅人		
答辩委员会成员	王明哲 朱建军 马敏书 王炜炜	王明哲 王富章 马敏书		
电子版论文提交格式	MSword			
论文总页数	66			