# 摘 要

此项目主要研究与实现命名为基于微服务的影院信息化系统。此系统主要服务于各大电影院，实现在线展示影片信息、影片的评论与评分、智能的影片搜索、在线购票与选座等一系列功能。此系统与传统的影院系统所不同的是，它是基于微服务，面向服务开发，实现了真正的高并发、高性能。

影院信息化系统根据业务进行拆分，形成单独的服务。基于springboot框架，快速形成一个web系统，而且能与其他的中间键快速整合，比如缓存中间键Redis，消息中间键RabbitMQ。相对于采用spring来构建项目，消除了很多繁琐且臃肿的配置。各个服务之间采用dubbo调用，而dubbo又是基于高性能通信框架Netty，实现了高并发，多人同时访问成为可能。而Netty是基于NIO，非阻塞I/O实现了高性能，快速响应用户的请求。

系统拆分为七个服务：用户服务、影片服务、影院服务、订单服务、支付服务、系统监控日志服务、后台管理服务。各个服务发布到docker容器中，互不干扰，可动态的实现服务的上下线与发布。利用高性能的web服务器nginx实现负载均衡。

**关键字**：微服务，springboot，高并发，高性能，dubbo

# Ab s t r a c t

dsadasdasdsdsad

# 目 录

# 1绪论

## 1.1开发背景

在传统的IT行业，一个软件大多是由很多系统相互堆砌而成，其导致的结果必然是可扩展性差，可靠性不高，维护成本高。到后面引入了SOA服务化，但是由于在早期SOA使用的是总线模式，这种总线模式与我们使用的某种技术栈强绑定，比如J2EE。这导致了企业遗留的系统很难对接，系统迁移时间长，成本高，新系统稳定性的收敛也需要一些时间。最终 SOA 看起来很美，但却成为了[企业级](https://www.baidu.com/s?wd=%E4%BC%81%E4%B8%9A%E7%BA%A7&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd" \t "https://blog.csdn.net/chinafire525/article/details/_blank)奢侈品，中小公司都[望而生畏](https://www.baidu.com/s?wd=%E6%9C%9B%E8%80%8C%E7%94%9F%E7%95%8F&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd" \t "https://blog.csdn.net/chinafire525/article/details/_blank)。

而微服务讲究的是微，狭义来讲就是体积小， 服务一个或者一组相对较小且独立的功能单元，是用户可以感知最小功能集。服务基于业务能力构建，并能够通过自动化部署机制来独立部署，这些服务使用不同的编程语言实现，以及不同数据存储技术，并保持最低限度的集中式管理。

一个微服务就是一个独立的实体。尽量避免把多个服务部署到同一台机器上服务之间均通过网络调用进行通信，从而加强了服务之间的隔离性，避免紧耦合。这些服务应该可以彼此间独立进行修改，并且某一个服务的部署不应该引起该服务消费方的变动。对于服务，需要考虑的是什么应该暴露，什么应该隐藏。如果暴露得过多，那么服务消费方会与该服务的内部实现产生耦合。这会使得服务和消费方之间产生额外的协调工作，从而降低服务的自治性。服务会暴露出API（Application Programming Interface，应用编程接口），然后服务之间通过这些API进行通信。API的实现技术应该避免与消费方耦合，这就意味着应该选择与具体技术不相关的API实现方式，以保证技术的选择不被限制。

## 1.2国内影院系统现状

影院票务管理系统每天需要实时向电影票务数据平台上报实时售票记录、定时上报票务统计记录。因此，专资办的数据是目前国内唯一精准的电影票房数据来源。影院隶属于院线，因此院线也会搭建本院线总部的数据平台，收集影院统计票房。

电影售票系统是经过广电总局认可，必须24小时联网的售票软件。牌照不是轻易可以拿到的，所以国内目前做影院票务系统的公司屈指可数。

## 1.3微服务的发展前景

从IT技术发展趋势看，无论硬件、[软件](http://www.siilu.com/biaoqian/284.html" \o "软件)、还是基础架构都在朝着轻量化的方向发展。微服务通过化整为零的概念，将复杂的IT部署分解成更小、更独立的微服务。相对传统的建设方法，传统企业更看重微服务如下四方面的优势：技术选型灵活，更轻松采用新架构和语言（28%）降低系统内部服务冗余，提升开发效率（27%）独立部署（22%）更好的容错机制（20%），在涉及复杂项目时，和单体架构的对比中，微服务从多个角度显示出了压倒性的优势。因此微服务的发展前景还是非常可观的。

# 2系统分析

## 2.1系统可行性分析

系统可行性分析是软件开发的生命周期一个重要阶段，它从工程管理、技术实现、研发经费等方面进行调查研究和分析比较，并对项目建成以后可能取得的财务、经济效益及社会环境影响进行预测，从而提出该项目是否值得投资和如何进行建设的咨询意见，从而为系统决策提供了良好的依据的一种综合性的系统分析的方法。在进行系统设计前，需要先进行系统分析，因为整个系统作为研究性课题来实现。下面通过系统技术可行性与运行可行性来分析系统可行性。

### 2.2.1系统技术可行性分析

技术可行性是指决策的技术和决策方案的技术不能突破组织所拥有的或有关人员所掌握的技术资源条件的边界。系统是基于Java语言开发。Java语言简单易学，去掉了头文件，指针运算，结构，联合，操作符重载，虚基类等，它是面向对象的，将重点放在了对象与对象的接口上，当然Java最大的优势就是跨平台了，这是JAVA的一个重要的优势。JAVA代码或者说字节码、二进制码可以跨平台的移植，而不用管具体的操作系统和硬件环境。JAVA本身就是面向网络的，只有在网络环境中才能显示出他的优势，比如：现在我有一个网络环境，要让我的笔记本和手机交互，笔记本环境是windows发出一个让手机定时录音的JAVA代码，手机只有简单Linux内核操作系统，照样可以完美的执行这个代码。

“一次编写，随处运行”：“write once, run anywhere”。

整个系统是在SpringBoot框架的基础上搭建的，SpringBoot相对于Spring就有很大的优势了，遵循"习惯优于配置"原则,使用Spirng Boot只需很少的配置,大部分时候可以使用默认配置;项目快速搭建,另外还可以无配置整合第三方框架;可完全不使用xml配置,只使用自动配置和Java Config;内嵌入Servlet如Tomcat容器,应用可用jar包运行，但是基本的思想依然是IOC与AOP，对象解耦与面向切面编程。

数据的持久层是Mybatis与Redis。前者是一种轻量级的ORM框架，实现了sql与业务代码的解耦，采用配置文件的方式统一管理sql，方便管理与维护。后者是高性能的缓存中间键，它是单线程，分布式，高可用的，提供了5中基本的数据存储类型，支持事务处理。

### 2.3.2系统运行可行性分析

系统的运行基于Tomcat，

## 2.2系统需求分析

# 3系统研究基础

## 3.1开发环境简介

## 3.2所用技术简介

### 3.2.1 SpringBoot

### 3.2.2 Mybatis

### 3.2.3 Gateway

### 3.2.4 Dubbo

### 3.2.5 Redis

### 3.2.6 Docker

## 3.3 本章小结

# 4 影院系统总体设计

# 5 影院系统详细设计

# 6 影院系统编码实现

# 7 影院系统测试与运行

# 8 影院系统总结评估