**香港粗菜馆在线订座系统开发**

摘要

均质充量压缩着火（HCCI）燃烧，作为一种能有效实现高效低污染的燃烧方式，能够使发动机同时保持较高的燃油经济性和动力性能，而且能和碳烟排

**1 绪论**

* 1. 研究背景及意义

当前,我国的经济保持了持续稳定发展,居民的消费水平不断提高,消费观念也逐渐有了新的变化,传统的餐饮服务已经难以满足群众日益增长的高端化、特色化需求。在市场的调解作用下,我国的餐饮服务市场正在经历着巨大而复杂的深刻变革市场空间得到进一步拓展,市场机会逐渐增多,服务模式日趋多样化、智能化。范文旭. 基于网络订餐平台的智能餐饮发展研究[J]. 无线互联科技, 2018 (12): 48. 为了节省就餐所需要的排队时间，用户开始选择提前预约的方式。传统的电话预定方式因为复杂繁琐，管理成本较大已经逐渐不能满足用户和商家的需求。在线订座系统可以使用户随时随地了解餐厅的人流状况，合理的选择自己的用餐时间，挑选自己喜欢的座位，并能实时查看自己的排队进度，减少实地排队浪费的时间提高用户消费体验。商家可以根据客流状况自主选择接单或者拒单，并根据实际效果调整用餐时长规定，以达到利润最大化。为香港粗菜馆开发一套线上订座系统可以给顾客提供更好地用餐体验，有效实现用餐高峰分流。同时对餐厅来说也是一种有效的线上宣传方式。

伴随着互联网的发展和普及，越来越多的人开始选择网上订餐，再加上用户订餐的时间比较集中，使得系统需要处理的用户数据发生了爆炸性的增长。传统的单体服务已经不能满足高并发情况下的性能需求。选用集群的方式可以有效的进行负载均衡处理。集群方式下各个节点功能复杂且相同，而实际情况下用户对各功能模块的访问频率却有较大差异，这就造成了资源的浪费。此外各节点都包含众多复杂的功能也使系统的耦合性增高，不利于系统的维护和扩展。选用微服务的架构将各模块拆分成一个独立的服务单元，它们之间通过RPC方式通信。在满足了性能要求的同时，有效的降低了系统的耦合度，可以有效的拓展、复用某些服务。

本系统因此定位为一个微服务架构的在线订座系统。用户可以提前在线查看香港粗菜馆的用餐环境，订购特色菜品，同时选择自己的就餐时间、人数、位置偏好，订餐之后还可以查看自己的排队进度。同时商家可以通过后台管理界面直观简便的查看顾客的消费走势与人员分布特征。可以有效节省店家和顾客双方的时间。提高用户就餐体验，节省商家运营管理成本，可以给食堂管理员提供配餐数量信息，精准备餐，避免浪费。扩大餐馆知名度，实现利润最大化。

* 1. 国内外研究现状

由于智能手机的普及，已经有很多比如大众点评这样的app提供了线上订餐、订座等服务。用户预订座位后可以随时查看自己的排队进度。但这些应用大多功能不够完善，比如没有充分考虑到用户的用餐人数，预计用餐时间，就餐位置偏好，以及超时过号未到的处理方式，还有用户即将到号时提醒用户，商家无法确认用户是否可以到达。有些商家选择直接线上预订排号的方式，但没有考虑用户取号后不去就餐的情况，导致过多人拍号，真正需要就餐的人又因为看到等待时间过长放弃就餐。一些商家为避免上述情况只提供实地排号服务，但这又同时给用户带来了额外的不便。

本系统在提供基本的订座软件功能的同时。着眼于解决用户长时间排队体验较差和商家可能损失潜在用户的问题。用户可以在线填写用餐人数、预计用餐时间以及座位偏好等信息，同时预定后可实时查看排队进展。同时用户还可以提前浏览特色菜品与订餐。

微服务架构是近年来流行的一个新词汇，但其背后的想法并不新鲜。事实上，它与几年前非常流行的SOA模式类似。微服务和SOA都是将应用程序分解为更小的服务，以便更高效地扩展和管理应用程序生命周期。虽然SOA是一个非常普遍的概念，可能意味着很多事情，但微服务体系结构描述了使用非常小的服务构建应用程序的特定方式，每种方法都专注于做好一件事情。由Netflix领导的大型互联网公司在实施微服务体系结构方面的成功，对于考虑切换的人来说起到了很大的激励作用与动力。Spring Cloud是一系列框架的有序集合。它利用Spring Boot的开发便利性巧妙地简化了分布式系统基础设施的开发，如服务发现注册、配置中心、消息总线、负载均衡、断路器、数据监控等，都可以用Spring Boot的开发风格做到一键启动和部署。Spring Cloud 也被开发者认为是最好的开发框架。因此本系统选用Spring Cloud为开发框架。

* 1. 研究内容

香港粗菜馆在线订座系统作为一个在线订座平台，旨在为广大用户提供一个美观、便利、可靠的在线预订平台。用户可以根据自己的不同情况通过在线平台提前预定满足自己要求的就餐时间、餐位、就餐人数等个性化餐位。同时用户可以通过在线平台了解餐厅环境、历史文化、附近交通等信息。特色菜品列表可供用户提前订餐。以便商家可以根据用户需求进行提前准备，提高效率，节约成本。用户预订座位之后可以实时查看自己的预订就餐时间，避免现场排队所需要的巨大时间开销。商家可以根据店内就餐人数状况选择接受或拒绝用户的订座请求，可以从后台添加、修改、删除菜品信息，对用户信息进行分析管理，以及通过后台数据对客流状况，菜品销售状况等进行统计分析。以便合理调整经营策略，提高餐厅收入。

* 1. 本章小结

本章从研究背景与意义入手，分析介绍了目前线上订座系统的发展状况与必要性，然后介绍了线上订座及微服务架构的研究现状，最后介绍了香港粗菜馆在线订座系统的主要功能。本章是对整个论文的一个概述。

1. **相关技术**
   1. MySQL数据库

本系统选用MySQL作为存储数据库。

MySQL是⼀个关系型数据库管理系统，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。。MySQL使⽤C和C++编写，并使⽤了多种编译器进⾏测试，保证了源代码的可移植性。MySQL支持多种操作系统并为多种编程语言提供了丰富的API。MySQL是一个真正的多用户、多线程SQL数据库服务器。它是以客户机/服务器结构实现的，由一个服务器守护程序以及很多不同的客户程序和库组成。它能够快捷、有效和安全地处理大量的数据。相对于Oracle等数据库来说，MySQL的使用非常简单。MySQL主要目标是快速、便捷和易用。MySQL 软件采⽤了双授权政策，它分为社区版和商业版，由于其体积⼩、速度快、总体拥有成本低，尤其是开放源码这⼀特点，⼀般中⼩型⽹站的开发都选择MySQL作为⽹站数据库。

·2.2 Redis数据库

本系统选为Redis实现分布式情况下的Session会话共享，保存用户的登录状态信息。

redis是一个key-value存储系统。和Memcached类似，它支持存储的value类型相对更多，包括string(字符串)、list(链表)、set(集合)、zset(sorted set --有序集合)和hash（哈希类型）。这些数据类型都支持push/pop、add/remove及取交集并集和差集及更丰富的操作，而且这些操作都是原子性的。在此基础上，redis支持各种不同方式的排序。与memcached一样，为了保证效率，数据都是缓存在内存中。区别的是redis会周期性的把更新的数据写入磁盘或者把修改操作写入追加的记录文件，并且在此基础上实现了master-slave(主从)同步。

Redis 是一个高性能的key-value数据库。 redis的出现，很大程度补偿了memcached这类key/value存储的不足，在部 分场合可以对关系数据库起到很好的补充作用。它提供了Java，C/C++，C#，PHP，JavaScript，Perl，Object-C，Python，Ruby，Erlang等客户端，使用很方便。

Redis支持主从同步。数据可以从主服务器向任意数量的从服务器上同步，从服务器可以是关联其他从服务器的主服务器。这使得Redis可执行单层树复制。存盘可以有意无意的对数据进行写操作。由于完全实现了发布/订阅机制，使得从数据库在任何地方同步树时，可订阅一个频道并接收主服务器完整的消息发布记录。同步对读取操作的可扩展性和数据冗余很有帮助。

* 1. SpringBoot

 Spring Boot是由Pivotal团队提供的全新框架，其设计目的是用来简化新Spring应用的初始搭建以及开发过程。该框架使用了特定的方式来进行配置，从而使开发人员不再需要定义样板化的配置。spring boot其实不是一个新的框架，它默认配置了很多框架的使用方式，就像maven整合了所有的jar包，spring boot整合了所有的框架。

  Spring Boot 简化了基于 Spring 的应用开发，通过少量的代码就能创建一个独立的、产品级别的 Spring 应用。Spring Boot为 Spring 平台及第三方库提供开箱即用的设置，这样你就可以有条不紊地开始。Spring Boot的核心思想就是约定大于配置，多数 Spring Boot 应用只需要很少的 Spring 配置。采用 Spring Boot 可以大大的简化你的开发过程，所有的常用框架，它都有对应的组件支持。

SpringBoot可以快速创建独立运行的Spring项目以及与主流框架集成。使用嵌入式的Servlet容器，应用无需打成WAR包。有效解决了J2EE笨重的开发、繁多的配置、低下的开发效率、 复杂的部署流程、第三方技术集成难度大等问题。

SpringBoot使编码、配置、部署、监控都变得非常简单。Spring Boot 的这些特性使其能够非常方便、快速构建独立的微服务。所以使用 Spring Boot 开发项目，会给我们的开发带来非常大的便利，有效提高开发效率与质量。

* 1. SpringCloud

Spring Cloud 是一系列框架的有序集合，它利用 Spring Boot 的开发便利性简化了分布式系统的开发，比如服务发现、服务网关、服务路由、链路追踪等。Spring Cloud 并不重复造轮子，而是将市面上开发得比较好的模块集成进去，进行封装，从而减少了各模块的开发成本。换句话说：Spring Cloud 提供了构建分布式系统所需的“全家桶”。

相比于Dubbo 只是实现了服务治理而言， Spring Cloud 实现了微服务架构的方方面面。香港粗菜馆在线订座系统中主要运用到了Spring Cloud的如下组件：

1. Eureka

Eureka 是 Netflix 开发的，一个基于 REST 服务的，服务注册与发现的组件。

它主要包括两个组件：Eureka Server 和 Eureka Client。

Eureka Client：一个Java客户端，用于简化与 Eureka Server 的交互（通常就是微服务中的客户端和服务端）。

Eureka Server：提供服务注册和发现的能力（通常就是微服务中的注册中心）。

各个微服务启动时，会通过 Eureka Client 向 Eureka Server 注册自己，Eureka Server 会存储该服务的信息。也就是说，每个微服务的客户端和服务端，都会注册到 Eureka Server。

Eureka的工作原理如图2-1。

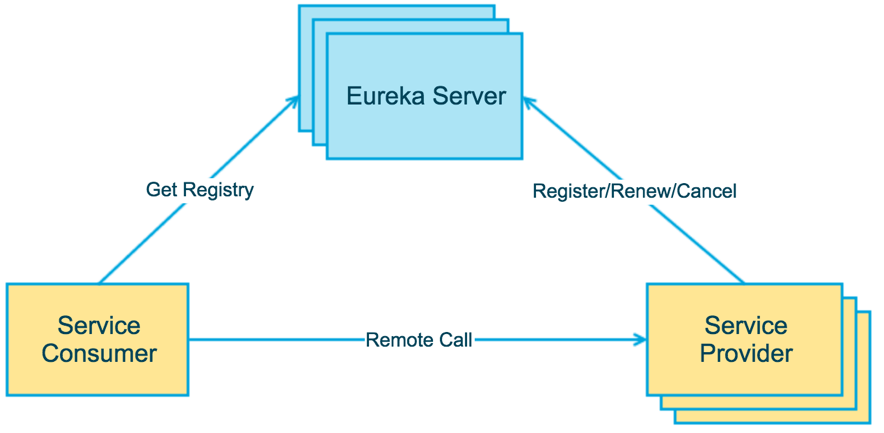


图2-1 Eureka工作原理架构图

1. SpringCloudConfig

SpringCloudConfig是SpringCloud创建的用来为分布式系统中的基础设施和微服务应用提供集中化的外部配置支持，它分为客户端和服务端两部分。服务端也称为分布式配置中心，是一个独立的微服务应用，用来连接配置仓库并为客户端提供获取配置信息，加密/解密信息等访问接口。而客户端则是微服务架构中各微服务应用或基础设施，通过指定的配置中心来管理应用资源与业务相关的配置内容，并在启动的时候从配置中心获取和加载配置信息。 引入配置中心后可以对各个微服务的配置进行集中统一的管理。并且可以实现在不重启微服务的前提条件下对微服务的配置进行更改。避免了传统情况下修改配置需要重启系统的巨大开销。SpringCloudConfig可以将配置文件存储在本地，也可以将配置文件存储在远程Git仓库，可以创建ConfigServer，通过它集中统一管理所有的配置文件。SpringCloudConfig架构如图2-2。

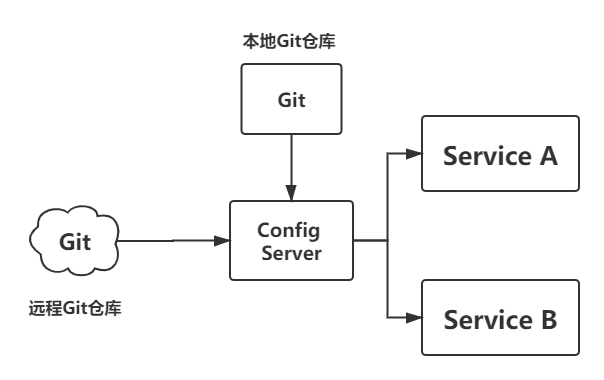


图2-2 SpringCloudConfig架构图

1. Feign

Feign默认集成了Ribbon，并和Eureka结合，默认实现了客户端负载均衡的效果。

在开发 Spring Cloud 微服务的时候，服务之间都是以 HTTP 接口的形式对外提供服务的，因此消费者在进行调用的时候，底层就是通过 HTTP Client 的这种方式进行访问。可以使用JDK原生的 URLConnection、Apache 的 HTTP Client、Netty 异步 Http Client，Spring 的 RestTemplate 去实现服务间的调用。但是最方便、最优雅的方式是通过 Spring Cloud Open Feign 进行服务间的调用 Spring Cloud 对 Feign 进行了增强，使 Feign 支持 Spring Mvc 的注解，并整合了 Ribbon 等，从而让 Feign 的使用更加方便。

Feign 是一个声明式的 Web Service 客户端。它的出现使开发 Web Service 客户端变得很简单。使用 Feign 只需要创建一个接口加上对应的注解。Feign 是一种声明式、模板化的 HTTP 客户端。在 Spring Cloud 中使用 Feign，可以做到使用 HTTP 请求访问远程服务，就像调用本地方法一样的，开发者完全感知不到这是在调用远程方法，更感知不到在访问 HTTP 请求。它整合了 Ribbon 和 Hystrix，从而不需要开发者针对 Feign 对其进行整合。Feign 还提供了 HTTP 请求的模板，通过编写简单的接口和注解，就可以定义好 HTTP 请求的参数、格式、地址等信息。Feign 会完全代理 HTTP 的请求，在使用过程中我们只需要依赖注入 Bean，然后调用对应的方法传递参数即可。

* 1. Bootstrap

Bootstrap是基于HTML、CSS、JavaScript开发的简洁、直观、强悍的前端开发框架，使得Web开发更加快捷。Bootstrap提供了实用性很强的组件，包括：导航，标签，工具条，按钮等供开发者使用。内置jQuery插件，支持html5 css3，支持less动态样式，跨设备，跨浏览器，响应布局。灵活简洁，代码优雅，美观大方，是当下最流行的前端框架。

* 1. Layui

本系统选用layui进行管理员后台管理页面的开发。

layui 是一款采用自身模块规范编写的前端 UI 框架,遵循原生 HTML/CSS/JS 的书写与组织形式,门槛极低,拿来即用。其外在极简,却又不失饱满的内在,体积轻盈,组件丰盈，从核心代码到 API 的每一处细节都经过精心雕琢，非常适合界面的快速开发。layui 首个版本发布于2016年秋，它区别于那些基于 MVVM 底层的 UI 框架，却并非逆道而行，而是信奉返璞归真之道。准确地说，它更多是为服务端程序员量身定做，开发者无需涉足各种前端工具的复杂配置，只需面对浏览器本身，使得开发十分便捷。

layui在组织形式上毅然采用了几年前的以浏览器为宿主的类 AMD 模块管理方式，更多的是面向于后端开发者，它拥有自己的模式，更加轻量和简单。layui 定义为“经典模块化”，并非是刻意强调“模块”理念本身，而是有意避开当下 JS 社区的主流方案，试图以尽可能简单的方式去诠释高效！它的所谓经典，是在于对返璞归真的执念，它以当前浏览器普通认可的方式去组织模块！ layui 认为这种轻量的组织方式，仍然可以填补 WebPack 以外的许多场景。所以它坚持采用经典模块化，也正是能让人避开工具的复杂配置，重新回归到原生态的 HTML/CSS/JavaScript本身。

layui 提供了丰富的内置模块，他们皆可通过模块化的方式按需加载。它兼容除了IE6/7外正在使用的全部浏览器，可作为 PC 端后台系统与前台界面的速成开发方案。

2.7 本章小结

本章主要对香港粗菜馆在线订座系统所用到的一些主要关键技术进行了简要介绍。分别介绍了数据库、分布式系统后台以及前端所用的相关技术。

**3 系统分析**

3.1 可行性分析