

论云原生架构及其应用

【摘要】

2019年3月，我单位联合某高校研发了《程序在线评测比赛考试系统》。系统以程序代码在线提交自动评测功能为核心，分为题库模块、评测机模块、实验作业模块、考试模块、比赛模块、抄袭判定模块、用户管理模块等，支持对接教务平台。在项目中我担任系统架构师，负责架构设计工作。

本文以该系统为例，主要论述了云原生架构在项目中的具体应用。系统以 Spring Cloud 微服务框架开发，分为前端 Web 服务、平台保障服务、业务服务三部分。前端 Web 服务由负载均衡与服务器集群结合，实现高并发的前台界面；平台保障服务以 Eureka 为中心，由 API 网关、服务注册中心、监控平台等构成，实现基础服务框架；业务服务划分为多个微服务，基于 Docker 容器，协同工作实现具体业务功能。最终系统顺利上线，获得用户一致好评。

【正文】

笔者在一个专为高校建设计算机专业智能教学一体化平台的单位任职，过往成果有《计算机组成原理仿真实验系统》等。2019年3月，我单位联合某大学研发了《程序在线评测比赛考试系统》项目（以下简称“OJ 系统”），以取代原有传统的编程上机考试平台。

系统以程序代码的在线提交自动评测功能为核心，主要分为题库模块、评测机模块、实验作业模块、考试模块、比赛模块、抄袭判定模块、用户管理模块等。题库模块主要负责试题和测试用例的管理，用户根据试题要求编写程序代码提交到系统，系统将测试用例与程序代码发送给评测机模块，由评测机自动编译、执行、判分，并将结果发送给其他相关模块进行统计；实验作业模块用于在线布置作业，从题库中选取试题，设置截止日期等要求；考试模块用于学生在线考试，按教师预先设置的参数自动从题库随机抽题生成试卷，以及向教务平台上传考试成绩；比赛模块主要用于 ACM 竞赛的培训；抄袭判定模块用于鉴定代码与他人代码雷同率；用户管理模块负责用户信息的管理。在这个项目中，我担任了系统架构师的职务，主要负责系统的架构设计相关工作。

云原生架构以微服务和容器技术为代表，有服务化、强韧性、可观测性和自动化四类设计原则。通过服务化的设计原则，应用被分解为多个服务，可分别选择不同的技术，单个服务模块很容易开发、理解和维护，无需协调其他服务对本服务的影响；通过强韧性的设计原则，微服务可以分布式云化部署，负载均衡管理请求的分发，避免单机失败对整体服务的影响，以及弹性调整资源容量；通过可观测性的设计原则，能够对系统进行健康检查、指标监控、日志管理和链路追踪，提高系统运维、管理和排错能力；通过自动化的设计原则，可实现系统的自动化部署、自动化扩展伸缩、自动化运维、持续交付和集成，有效减少人工操作的工作量。

OJ 系统基于 Spring Cloud 微服务框架开发，将平台服务划分为三类，分别为前端 Web 服务、平台保障服务、业务服务。下面针对这三类服务展开具体说明。

1. 前端 Web 服务

前端 Web 服务主要提供给用户使用的界面，分为前置 Nginx 负载均衡服务器、前端网站 Nginx 集群。当用户通过网络访问系统时，首先会访问到前置的 Nginx 负载均衡服务器，负载均衡服务器会将请求转发到前端网站的 Nginx 集群，前端网站通过发起 **http** 请求来和后端交互，具体是通过 **Ajax** 方式来调用后端 REST API 接口。用户访问网站通过前置的 Nginx 负载均衡服务器来转发到前端网站集群，以起到将用户请求进行分流的作用。当前端网站集群中的部分服务发生故障时，系统仍可正常地对外提供服务。前置 Nginx 负载均衡服务器使用软件反向代理的方式来实现负载均衡，部署为路由模式，系统内部网络与外部网络分属于不同的逻辑网络，以实现系统内部与外部网络的隔离。在负载均衡算法的选择上，使用最小连接法，每当用户的请求来临时，任务分发单元会将任务平滑分配给最小连接数的前端网站节点，这样的架构以廉价且透明的方式扩展了服务器和网络的带宽，可以大大提升系统的并发量，同时保证网站前端整体的稳定性和可靠性。

2. 平台保障服务

平台保障服务用以实现后端微服务的基础框架，包括 **API** 路由网关、服务注册中心、服务监控组件。**API** 网关收到前端的请求，不会直接调用后端的业务服务，而是首先会从服务注册中心根据当前请求来获取对应的服务配置，随后通过服务配置再调用已注册的服务。当后端微服务存在多个实例时，将采取负载均衡的方式调用。服务注册中心是整个云原生架构体系的核心部分，由 **Spring Cloud** 的 **Eureka** 组件来实现，专门提供微服务的服务注册和发现功能，涉及三种角色：服务提供者、服务消费者和服务注册中心。**API** 路由网关、所有业务服务，以及服务监控平台组件都注册到服务注册中心。通过服务注册中心两两互相注册、**API** 路由网关向服务注册中心注册多个实例等方式，来实现后端整体服务的高可靠性。服务监控平台通过注册到服务注册中心，获取所有注册到服务注册中心的后端业务服务，从而监控到所有后端业务服务的运行状态信息，最后收集并展示整个微服务系统的运行状态，更进一步保证整个后端的服务质量。

3. 业务服务

业务服务按功能模块，相应划分为题库服务、评测机服务、考试服务、比赛服务、抄袭判定服务等。各服务单独打包，基于 **Docker** 容器，连同运行环境一起封装，根据实际情况可在一台或多台物理机同时部署多个实例，服务启动后会将自身信息注册到服务注册中心。服务间协同工作，通过松耦合的服务发现机制，动态调用对方 **REST API** 接口。对于压力较大的服务，如评测机服务、抄袭判定服务等，将部署为多实例集群。以在线考试功能为例，用户进入考试时，考试服务核验考生信息通过，调用题库服务，题库服务返回试题，由考试服务组合成试卷，返回前端显示。用户交卷时，提交的程序代码到达考试服务，考试服务拆分后分发给题库服务，题库服务将程序代码和测试用例送入 **MQ** 队列排队。然后由负载均衡机制，依次将队列中待评测程序分发给评测机服务编译、执行、判分，完成评测后，题库服务统计试题通过率，考试服务统计成绩并向前端显示。在此期间服务请求者无需了解其他服务对数据如何具体处理和分析。

系统自 2019 年 10 月正式上线已运行一年有余，在学校的日常教学考试和竞赛培训中投入使用，至今已有 3000 名以上的学生用户，评测了 70000 份以上的程序代码，获得了单位同事领导和学校教师们的一致好评。在开发和试运行过程中，主要遇到了两个问题。一是跨域问题。**OJ** 系统前后端分离，前端通过 **Ajax** 访问后端服务。由于浏览器同源策略的限制，导致前端 UI 无法正常访问不同端

口和 IP 的后端服务。我们利用 Spring Boot 后端的 Cors 跨域机制解决了该问题。二是评测机宕机问题。评测机服务需要执行用户提交的代码，部分用户短时间内提交了大量不安全代码，导致评测机集群中所有实例全部宕机。我们引入心跳机制、快照回滚机制，以及基于机器学习技术的预判断机制，使评测机宕机时能够在 10 秒内自动重置恢复运行，最终解决了该问题。

实践证明，OJ 系统项目能够顺利上线，并且稳定运行，与系统采用了合适的架构设计密不可分。经过这次云原生架构的方法和实施的效果后，我也看到了自己身上的不足之处，在未来还会不断更新知识，完善本系统在各方面的设计，使整个 OJ 系统能够更加好用，更有效地服务于高校师生。

【原文链接：<https://zhuanlan.zhihu.com/p/337973885>】