# **论软件系统架构风格**

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，主要论述了软件系统架构风格在项目中的具体应用。系统采用了微服务架构风格，基于Spring Cloud框架。通过微服务架构，将系统分解为多个服务，前端Web服务解决前台并发问题，平台保障服务实现基础服务框架，业务服务实现具体业务功能，解决协同问题。各服务采用不同技术开发，独立部署，系统的性能、可靠性与扩展性得到了大的提升。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（以下简称为“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。

  常用的软件架构风格分为五类：数据流风格、调用/返回风格、独立构件风格、虚拟机风格、数据仓库风格。数据流风格包括批处理、管道-过滤器；调用/返回风格包括主程序/子程序、面向对象、层次结构；独立构件风格包括进程通信、隐式调用；虚拟机风格包括解释器、规则系统；数据仓库风格包括数据库系统、黑板系统、超文本系统。此外，架构风格还有MVC、SOA、微服务等。其中，层次结构风格将构件按层次组织，每层为上层提供服务，使用下层服务，只能接触邻层；进程通信风格将构件作为独立过程，通过消息传递调用；微服务架构将单一应用划分为多个松耦合的小服务，相互协调配合，采用轻量级通信机制，各服务可选择不同技术，支持独立部署。

  因OJ系统需要按功能模块进行拆分，且对性能、可靠性和扩展性要求较高，我们采用了微服务架构。系统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业务服务的基础框架，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后端服务架构体系的核心部分，由Spring Cloud的Eureka组件来实现，专门提供服务的注册和发现功能，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后端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

  业务服务按系统业务模块，相应划分为题库服务、评测机服务、实验作业服务、考试服务、比赛服务、抄袭判定服务、MQ队列服务等。服务间协同工作，通过松耦合的服务发现机制，来动态调用对方的REST API接口。其中，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后，根据预先设置的参数，调用题库服务，题库服务返回试题信息，由考试服务组合为试卷，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，依次将队列中排队的待评测程序分发给空闲的评测机服务编译、执行、判分，评测机服务完成评测后，将结果返回给题库服务和考试服务，题库服务进行试题通过率统计，考试服务进行成绩统计，以及向前端显示成绩。在这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。

总结

  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，截至目前已有3000以上的学生用户、评测了70000条以上的程序代码，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，但由于部分用户短时间内提交了大量不安全代码，导致所有评测机服务全部宕机。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，使评测服务宕机时能够在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风格密不可分。经过这次微服务架构应用的方法和实施的效果后，我也看到了自己身上的不足之处，在未来还会不断地更新知识，完善本系统的架构设计，使整个系统能够更加好用，更有效地服务于高校师生。