# **论Web应用系统性能优化技术与应用**

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Web系统性能优化技术在项目中的具体应用，着重介绍了负载均衡技术、缓存技术、数据库主从部署三种技术。通过负载均衡技术结合服务器集群，提高网站的并发能力；缓存技术基于Redis内存数据库，降低系统数据库压力，提高页面加载速度；数据库主从部署实现读写分离，消除了数据库的负载瓶颈。最终系统顺利上线，并且稳定运行，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。

  系统需要支撑全校学生编程课程的学习考试，以及大量校外用户的使用，为保证系统的可靠性和用户体验，对系统进行性能优化显得尤为重要。传统的基于单块架构的编程考试系统早已无法支撑日益增长的用户需求，存在严重的性能问题，且很难进行扩展，因此OJ系统在架构设计中必须考虑到Web应用系统性能优化技术的应用。OJ平台以微服务架构作为基本架构风格，基于Spring Cloud框架，使用Java语言开发，分为前端Web服务、平台保障服务、业务服务三个部分。前端Web服务由Nginx负载均衡与服务器集群结合，解决前台界面的并发问题；平台保障服务以Eureka为中心，分为API网关、服务注册中心、监控平台，用以实现基础服务框架；业务服务分为多个微服务，实现具体业务功能，解决协同问题。

  在开发过程中，我们运用了多种性能优化技术。这里重点以负载均衡技术、缓存技术、数据库主从部署为例，介绍本系统Web性能优化的方案和达到的效果。

1. 负载均衡技术

  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. 缓存技术

  通过对系统业务的分析，我们发现后端业务服务在进行相关的业务逻辑操作时，会有一部分数据使用频率很高但不会经常变化，如系统基本设置和登录用户信息等。这一部分数据如果每次使用时都直接访问数据库，将会给数据库服务增加非常大的开销。为解决这一问题，我们引入了Redis内存数据库作为缓存。在系统启动时，将系统设置从数据库读出并缓存在Redis中，用户登录时，也会将用户信息缓存在Redis中，这样当服务再次使用这部分数据时，便可直接从效率更高的Redis缓存中读取。对于部分在一定时间内不会发生变化的数据库表内容，如试题信息、实验作业信息、课程信息等，我们也使用Redis进行了缓存。为保证Redis缓存与原数据库的数据同步，当读取数据时，会先读取Redis缓存中的数据，如果Redis缓存中不存在所要的数据，则从原数据库中读取，并同步写入至Redis缓存中，当写回数据时，写入到原数据库中，并同步淘汰Redis缓存中的数据。这样的架构极大地提升了系统的性能，并且缓解了页面加载缓慢等问题。

3. 数据库主从部署

  网站在使用缓存后，大部分数据读操作都可不通过数据库就能完成，但仍有一部分读操作（缓存不命中、过期）和全部写操作需要访问数据库，在网站用户达到一定规模后，数据库因负载压力过高而成为整个网站的瓶颈。对于OJ平台来说，后端数据库存储的性能是极其重要的，所有的程序提交记录、考试答题记录、成绩统计记录都将存储到数据库中。这里就要求数据库存储有非常强的性能，本系统主要采用主从式方法部署数据库结构，实现读写分离的架构。业务服务在写数据时，访问主数据库，主数据库通过主从复制机制将数据更新同步到从数据库，这样当业务服务读数据的时候，就可通过从数据库获得数据。业务服务通过使用专门的数据持久层访问模块来访问读写分离后的数据库，使数据库读写分离对应用透明。为了缓解由磁盘硬件带来的性能瓶颈，数据库所依赖的硬件存储采用了基于RAID6的固态硬盘阵列，能够在硬件方面提升数据库的随机存取性能。这样的设计提高了系统数据库的运行效率。

总结

  我们在这次系统性能优化设计中，还使用了很多其他的性能优化策略，这里不再一一赘述。系统在经过性能测试、负载测试、压力测试、稳定性测试后，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。日常使用过程中最高出现过600余用户同时在线进行实验作业提交、评测的情况，基本未出现页面加载缓慢、请求超时的问题，满足了高校编程课平台的基本性能需求。

  实践证明，OJ系统项目能够顺利上线，并且稳定运行，与系统采用了充分的性能优化设计密不可分。经过这次Web应用系统性能优化技术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，完善本系统的设计，使整个系统能够更加好用，更有效地服务于高校师生。