

文章编号: 1000-5641(2018)02-0089-12

基于 WebSocket 的 PK 答题的设计与实现

祝 翔, 董启文, 郁可人

(华东师范大学 数据科学与工程学院, 上海 200062)

摘要: 互联网的发展给人们的生活和工作都带来了极大的方便, 相应人们对互联网应用也产生了更多的需求. 如何帮助学生通过互联网进行在线答题也越来越成为广大中小学以及各在线教育平台的现实需求. 从这一现实需求出发, 提出了一款基于线上的PK (Player Killing) 答题功能. 通过在线PK答题, 可以提升学生学习的趣味性, 提高学生对知识的掌握能力.

关键词: 在线答题; B/S 模式; 题库; PK 对战

中图分类号: TP391 **文献标志码:** A **DOI:** 10.3969/j.issn.1000-5641.2018.02.009

Design and implementation of PK answer based on WebSocket

ZHU Xiang, DONG Qi-wen, YU Ke-ren

(School of Data Science and Engineering, East China Normal University, Shanghai 200062, China)

Abstract: The development of the Internet has brought great convenience to people's life and work, at the same time, people have more demand for Internet applications. How to help students answer questions online is increasingly becoming a reality demand for primary and secondary schools as well as online education platforms. From this realistic demand, an online PK (Player Killing) answer function is proposed. Through the online PK answer, students' interest in learning and their ability to grasp knowledge can be enhanced.

Key words: online answers; B/S mode; question bank; PK battle

0 引 言

近几年, 在线教育逐步走向大众的视野. 在线教育相比于传统线下教育所存在的, 如资源分配不均、投入产出失衡等问题有着明显的优势, 它可以突破时空的限制, 结合移动终端的发展让学生实现碎片化学习, 让教育的方式变得更加多样化^[1-2].

从2010年开始, 各种在线教育平台如雨后春笋般地涌现, 除了像新东方、学大教育等一系列的老牌教育公司, 还出现了如网易公开课、网易云课堂以及百度教育平台腾讯课堂等新的势力. 互联网巨头们纷纷涉足互联网教育领域, 可见当下在线教育已经成为一个极具发展潜力的领域^[3].

收稿日期: 2017-06-19

基金项目: 华东师范大学信息化软科学研究课题 (41600-10201-562940/008)

第一作者: 祝 翔, 男, 硕士研究生, 研究方向为数据科学技术应用. E-mail: 965730955@qq.com.

本论文的主要内容是, 研究在线答题系统中 PK 答题模块的设计与实现, 提供解决在线答题系统中 PK 答题模块的技术实现方案.

全文的结构安排如下: 第 1 节为系统简介, 分析在线答题系统的主要功能; 第 2 节为技术介绍, 主要是 SpringBoot 的框架及其特点介绍、Bootstrap 框架的介绍、Websocket 的基本原理、安全框架 Shiro 的技术架构简介, 以及 Redis 非关系型数据库的相关介绍; 第 3 节为需求分析, 分析在线答题系统 PK 模块的具体的需求; 第 4 节为在线答题系统中 PK 答题模块的设计与流程, 简要讨论 PK 答题的设计思路与基本流程; 第 5 节为在线答题系统中 PK 答题模块的实现, 详细阐述如何使用 WebSocket 进行 PK 用户匹配, 以及如何控制多线程并发等细节; 第 6 节为 PK 答题模块的测试部分, 主要包括客户端功能测试以及服务器端的测试; 第 7 节为总结与展望, 对全文进行总结, 并对下一步工作进行展望.

1 在线答题系统简介

在线答题系统中 PK 答题模块为该系统中的核心模块之一, 此外还有练习模块、试题管理模块.

试题管理模块主要用于实现对试题的导入功能, 系统中的试题可以实现单题导入, 也可以通过 Excel、CSV 等文档实现试题的批量导入. 试题按照年级、科目、章节进行分类.

答题模块主要用于实现试题解答的功能. 在用户登录后, 后端可以根据用户的 ID 号寻找到该用户之前的答题记录, 并根据上一次的答题历史记录继续答题. 当用户点击交卷后, 系统会自动评阅客观题的答案, 并提供试题的标准答案与解析.

PK 对战模块是极具特色和新意的模块, 用户在进入 PK 答题模块后能够自动寻找对手, 进行 PK 答题. 在 PK 答题模块中, 匹配成功的用户将共同的去完成同一套试题, 并得出胜负. 该模块很大程度的增强了用户在答题过程中的趣味性, 对用户具有较大的吸引力.

该答题系统采用了目前的主流技术和框架 (Spring Boot、Bootstrap、WebSocket、MySQL、Shiro、Redis) 来进行具体的开发和实现. 系统整体是采用的 B/S (Browser/Server, 浏览器/服务器) 结构来搭建的, B/S 结构是一种基于 Web 的网络结构模式, Web 浏览器为客户端最主要的应用软件. 这种模式将页面展示的功能交由客户端来处理, 将系统的核心部分集中到服务器上, 简化了系统的开发、维护和使用. 用户利用客户机上安装的浏览器通过应用服务器同数据库进行数据交互. B/S 结构实际上是由表示层、业务逻辑层和数据存储层构成的 3 层结构, 如图 1 所示.

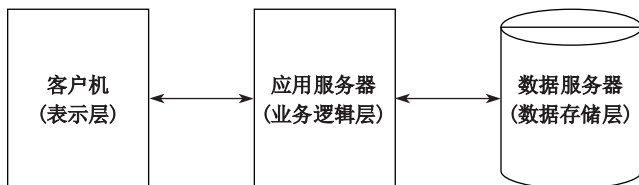


图 1 B/S 架构

Fig. 1 B/S architecture

2 在线答题系统相关技术介绍

2.1 Spring Boot

Spring Boot 为 Spring 的衍生产品. Spring 是一款轻量级的 Java 开源框架, 最早是从

2004年由Rod Johnson开发,到如今经过了数年的发展和完善,已经成为了当下最为流行一款的Java开源框架^[4].

Spring框架凭借其控制反转、面向切面、MVC等技术特点,很好地避免了由于硬编码所造成的过度程序耦合问题,用户可以更加专注于编写上层的应用.因而在常见的JavaEE项目开发设计中, Spring框架主要应用于系统的业务逻辑层.

为了简化利用Spring框架进行的项目开发、配置、调试和部署工作,2013年Pivotal团队提出了Spring Boot项目,该项目集成了大量简单实用的基础框架.例如一些内嵌容器(Tomcat、Jetty、Undertow)、日志框架、JMS框架、持久化框架、流行的NOSQL数据库(Cassandra、MongoDB)、缓存框架等^[5].

应用Spring Boot框架可以大幅度地提高开发效率,其具有以下特性^[6-7].

- (1) 遵循软件开发简约性原则,尽可能根据项目依赖特性自动生成Spring框架.
- (2) 可以生成独立运行的Spring应用.
- (3) 没有冗余代码生成,也没有XML配置文件.
- (4) 提供支持强大的Restful风格的编码,非常简洁.
- (5) 提供更多的应用开发模块及功能.
- (6) 将Tomcat或Jetty Web应用服务器内嵌其中,无需提供War包.

2.2 Bootstrap

在早期Web开发的过程中,Web前端开发者每天都在重复地使用HTML、CSS、Javascript编写模版、样式、和动态交互效果^[8].这种重复工作不仅单调乏味而且跟不上网络发展的需求.此时作为Web前端开发者就需要寻找新的途径提高工作效率.Bootstrap的出现就在很大程度上帮助Web前端开发者摆脱了这种现状^[8-9].

Bootstrap是由Twitter著名前端开发工程师Mark Otto和Jacob Thornton合作开发的一款目前很流行的前端开发框架.Bootstrap基于HTML、CSS、JAVASCRIPT,是一个桌面应用程序,它简洁灵活,并且包含了几十个组件,支持响应式和非响应式Web设计,有较强的兼容性,可在多种设备上良好地运行^[10],能大幅提高Web开发者的开发效率,降低开发成本.

2.3 WebSocket

WebSocket是Web浏览器和Web服务器之间的全双工通信标准,其中WebSocket协议是由IETF定为标准,WebSocket API是由W3C定为标准^[11].

当Web服务器与客户端建立起WebSocket协议的通信连接.之后所有的通信都依靠这个专用协议进行.通信的过程可以互相发送JSON、XML、HTML或者图片等任何格式的数据^[12].

由于是建立在HTTP基础上的协议,因此连接的发起方仍是客户端,而一旦确立是WebSocket通信连接,无论是服务器还是客户端,任意一方都可以直接向对方发送报文^[13].

这种全双工的通信方式使得WebSocket具有以下的一些特点.WebSocket支持由服务器向客户端推送数据的推送功能,这样,服务器可以直接发送数据,无需等待客户端的请求.由于在建立了WebSocket连接之后,就可以一直保持着连接状态.和传统的HTTP协议相比,不仅每次的HTTP连接时的总开销减少,而且由于WebSocket首部的信息量很小,通信量也相应地减少了^[13-14].

2.4 Shiro

Apache Shiro 是一个强大而灵活的开源安全框架, 它主要用于处理身份认证、授权、企业会话管理和加密等问题.

Shiro 的技术架构主要由 Subject、SecurityManager 和 Realm 等 3 个核心组件构成^[15], 如图 2 所示.

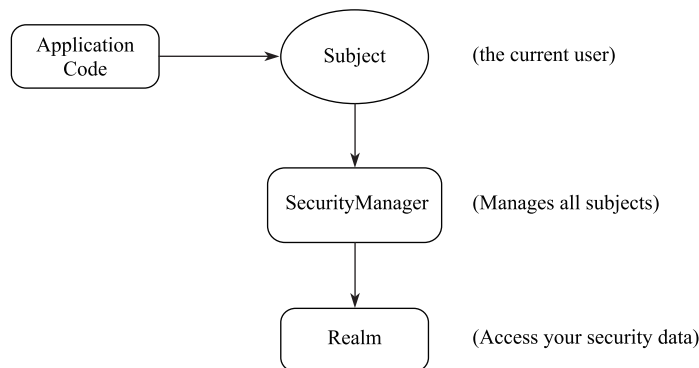


图 2 Shiro 核心组件

Fig.2 Shiro core components

Subject: 表示当前登录对象, 泛指的是与当前应用进行信息交换的对象.

SecurityManager: 是 Apache Shiro 技术框架的核心, 通过它来提供安全管理及权限控制.

Realm: 是 Apache Shiro 与应用系统数据库之间的纽带, 它封装了应用系统数据库与 Apache Shiro 之间的连接细节, 并在需要时将相关权限数据提供给 Shiro. 在配置 Apache Shiro 进行权限控制时, 必须至少指定一个数据库 DAO(Realm), 用于登录或授权认证. Apache Shiro 框架内部已经实现了包括如 LDAP、关系数据库 (JDBC)、类似 INI 的文本配置资源以及属性文件等大量数据源实现. 根据应用系统需求选择合适的数据源实现, 如果上述实现不能满足系统要求, 还可以自定义数据源, 只需要实现 Realm 接口即可^[16-17].

2.5 Redis

Redis 是一款开源的支持网络、基于内存亦可持久化的日志型、Key-value 数据库. 它拥有着许多区别于同类产品的特点. 例如, Redis 作为一个键值存储系统, 相比于 Memcached 拥有更多的数据类型. Redis 利用内存来实现数据的存储, 并且可以通过搭建 Redis 集群, 将主服务器上的数据和任意数量的数据从服务器进行共享^[18].

由于 Redis 是利用内存作为介质来存储数据的, 所以 Redis 拥有着极高的读取速度, 它每秒可以执行大概 110 000 设置操作和 81 000 个读取操作, 并且这些操作都是原子性的, 很大程度上保证了多个用户的同时访问 Redis 服务器时可以获得最新的数据^[19].

3 在线答题系统 PK 模块的需求分析

在具备了上述的基础知识和基本原理的准备后, 本节主要分析 PK 答题模块的需求与用例.

3.1 需求分析

在线 PK 答题模块需要完成用户身份的获取、PK 用户的匹配、匹配成功后随机生成 PK 题目、将 PK 题目并行发送给成功匹配的用户、同步评阅等几个基本功能.

PK模块中需要获取每一个用户的身份信息,并将该用户之前的做题记录以及PK的胜负情况显示在用户自己的PK界面中。当PK用户匹配成功之后,还需要将用户自己的基本信息发送给对手。

PK用户的匹配需要能够满足高并发的要求,当多个用户同时请求PK对战时,需要能够正确地实现匹配,避免产生匹配混乱的情况。在用户发送PK请求失败后,要能够向前端用户及时反馈,并给予提示信息,让用户重新请求PK服务。

在PK用户匹配成功后,为了保证PK答题时的公平性,需要保证向匹配成功的这一对用户发送同一套练习题,并且该练习题需要通过随机算法,从后台中获取。

当匹配成功的一对用户中,倘若其中一个PK用户完成了自己的PK题目,另一方还未完成,这时就不能得出并显示PK结果。只有当两个PK用户都完成题目之后才能显示这一对PK用户的最终结果。在一次PK的过程中需要有时间的设置,当有用户未能在指定时间内完成所有习题的作答,系统会自动提交并且视未作答的题目为错误。

在线PK模块是系统中最为新颖且具有特点模块,必须要做到响应迅速、支持大规模的并发操作。

3.2 用例分析

在线答题PK模块的用户主要为学生、管理员。学生在PK模块的主要用例有答题、PK做题、查看成绩、查看题目分析。

管理员在PK模块中的主要用例有题目的录入、随机生成PK试题。

PK模块的系统用例如图3所示。

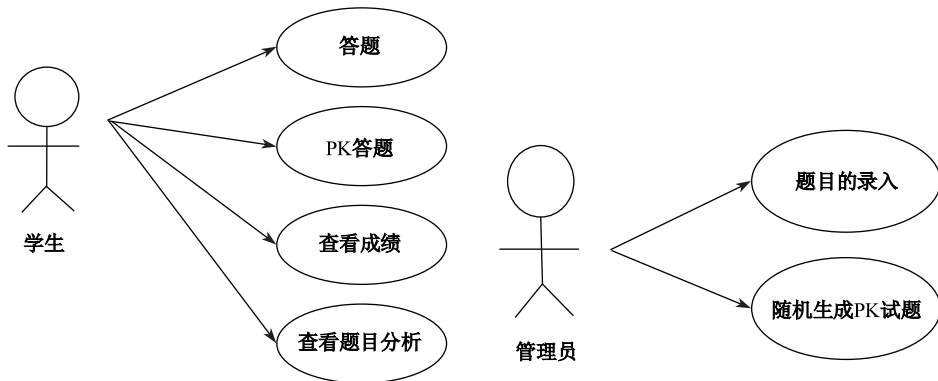


图3 PK模块的系统用例图

Fig. 3 The system use case diagram for the PK module

4 PK答题模块的设计与实现

通过上述的需求分析和模块用例的介绍,本节将详细阐述PK答题模块的设计与实现。

4.1 PK答题模块设计的目标

模块设计的目标是为了满足模块的需求。在线答题系统中PK模块的设计目标主要是能实现用户在线PK答题的功能,要能够支持一定规模的用户同时在线PK答题,以达到可实际应用的目的。

在线答题系统中PK模块的设计和实现需要达到以下目标。

(1) 能够完成基本的在线PK用户匹配。

- (2) 能够确保每一对 PK 用户有各自的一套 PK 习题.
- (3) 能够确保只有当一对 PK 用户双方都答题结束后再同时返回评阅结果.
- (4) 能够达到单台服务器支持千人以上在线 PK 答题的服务请求.

4.2 PK 模块流程

在线答题系统中 PK 模块的基本流程以下.

(1) 在线用户的匹配. 在线用户的匹配是当用户发送了 PK 请求时, 此时就需要判断是否已经有其他用户发送了同样的 PK 请求, 如果存在其他用户也发送了 PK 请求, 就需要将两人进行匹配, 并将匹配成功的响应返回给用户. 如果此时未能成功匹配, 则提示用户要求重新发送 PK 请求.

(2) PK 套题的随机生成. 当一对 PK 用户匹配成功之后, 需要生成一套测试习题, 可以通过随机算法, 从题库中任选一定数量的题目.

(3) 并行发送 PK 试题. 在提示用户匹配成功并随机生成测试题之后, 需要将题目同时发送给匹配成功的一对用户.

(4) 同步答题. 当匹配成功的用户接收到生成的一套随机测试题后, 需要两个用户同时答题. 倘若其中一个用户没有答题完成, 另一个用户则需要等待, 直到两个用户都答题完成后才能发送最终答题结果.

(5) PK 结果评阅. 当匹配成功的一对用户都提交答题结果后, 客观题将被自动评分, 并将评阅结果同时发送给这一对进行 PK 的用户.

在线答题系统之 PK 模块的流程如图 4 所示.

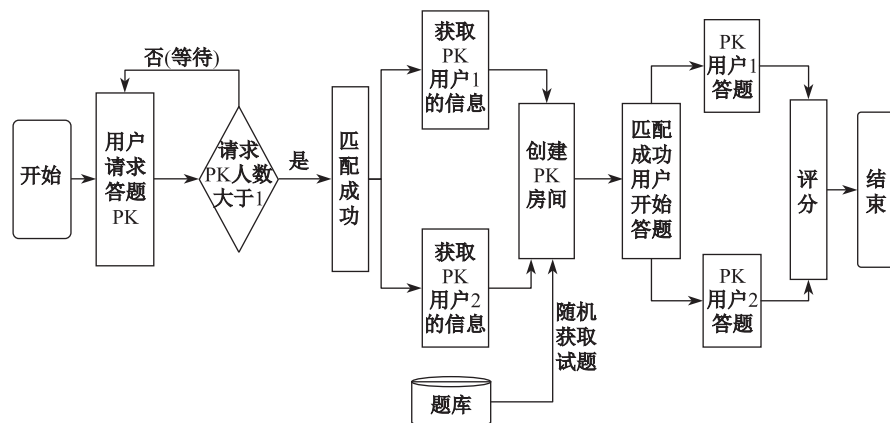


图 4 PK 模块流程图

Fig. 4 PK module flow chart

5 在线答题系统 PK 模块的实现

前面几节介绍了在线答题系统中 PK 模块的需求和流程与设计. 本节将具体阐述在线答题系统中 PK 模块的实现.

5.1 系统开发环境

系统使用 Linux 作为服务器操作系统, 通过 Node.js 作为前端的 Web 服务器, 使用 Java 作为开发语言, 使用 MySQL 作为数据库管理系统.

Node.js 作为 Web 服务器, 对服务请求提供响应, 当外部请求到达时, Web 服务器负责前端页面的跳转, 并通过 Ajax 异步请求后端的数据; 后端的服务器接受到异步请求后交给由 Spring Boot 框架搭建的微服务进行处理, 如果需要从数据库获取数据, 则通过 SpringDataJPA 的相关接口访问数据库并获取数据。

5.2 在线答题系统PK功能的实现

PK模块的主要功能包括在线用户的匹配、随机出题并发送试题、在线答题、返回PK结果等。

5.2.1 在线用户匹配

在线PK用户的匹配主要利用了 WebSocket 协议, 通过将传统的 HTTP 请求升级为 WebSocket 请求, 建立 WebSocket 长连接, 这样可以实现服务器端与客户端实时发送消息的功能。前端 WebSocketAPI 提供了客户端和服务端之间交互的常用接口, 当 WebSocket 连接建立成功时, 就会创建一个 WebSocket 实例对象, 通过该实例对象可以执行一些在长连接过程中前后端交互的方法。这些方法可以在 WebSocket 建立连接时、消息传递时、发生错误时, 以及关闭连接时执行一系列自定义指令。

当有一个用户发起PK请求时, 前端就会发送 WebSocket 的连接请求。此时, 后端接收到请求也会相应地为该用户创建 WebSocket 实例对象, 并打开属于该用户的 WebSocket 连接通道, 为该 WebSocket 对象维护一个专属的 WebSocketSession。

为了记录每一个发出PK请求的用户信息, 在后端还需要一个存放 WebSocketSession 的并发队列。若在某一时刻存放 WebSocketSession 的并发队列为空, 则将该 WebSocketSession 对象添加到并发队列中, 发送响应让前端处于等待状态, 即 PK 用户处于等待匹配状态。若此时并发队列中已经存在 WebSocketSession 对象, 则将并发队列的队首元素出队, 返回用户成功匹配的信息。

5.2.2 用户匹配的并发控制

当只有少量用户发出PK答题的请求时, 后端服务器完全可以为这些请求创建新的线程来执行操作。但当有数以万计的用户同时发出PK答题请求时, 就需要考虑到单台服务器对于用户请求的承载量^[20]。

为了解决多线程并发可能出现的线程安全问题, 可以利用 JDK 提供的一套线程框架 Executors。Executors 充当线程工厂的角色, 可以通过 Executors 创建特定功能的线程池。

线程池可以预先创建一定数量的工作线程并限制其线程数量, 来控制线程对象的内存消耗, 从而降低系统开销和资源消耗。通过对多个请求重用线程, 线程创建、销毁的开销被分摊到了多个请求上^[21]。另外通过限制线程数量, 也可以降低虚拟机在垃圾回收方面的开销。

线程事先已被创建, 请求到达时就可直接进行处理, 消除了因线程创建所带来的时间延迟。另外多个线程可以并发处理, 提高系统响应速度^[22]。

当通过 Executors 创建了一个新的线程池时, 在该线程池内部会维护一个任务队列用于存放待执行的任务请求, 并且还提供了核心线程数 corePoolSize、最大线程数 maximumPoolSize 等参数设置。

当有新的任务需要执行时, 如果线程池中的实际线程数小于 corePoolSize, 则会优先创建线程; 若线程池中的实际线程总数大于 corePoolSize, 则将任务加入任务队列; 若任务队列中的任务已满, 则在总线程不大于 maximumPoolSize 的情况下创建新的线程; 若线程数大

于 maximumPoolSize, 则执行拒绝策略, 停止新线程的创建.

通过以上机制, 可以很好地控制服务器端的并发量, 提高服务器的可用性.

5.2.3 随机出题并利用 Redis 缓存发送试题

当有一对 PK 用户匹配成功之后, 则创建 PK 房间即 PKRoom 实例对象, 并从后端 Session 中获取用户信息, 将用户信息保存到新创建的 PKRoom 实例对象中.

然后通过随机算法获得一定数量的题目编号, 判断这些题目编号是否存在于 redis 缓存中. 倘若该题目在 redis 缓存中并不存在, 则将题目存储在 redis 缓存数据库中, 以便于在前端发出请求时能够及时地响应. 当接受到请求, 即可通过 redis 缓存获取到具体的题目信息, 并将题目信息同时发送给匹配成功的 PK 用户.

5.2.4 在线答题并返回 PK 结果

当匹配成功的 PK 用户接收到发来的题目信息后就可以开始答题. 此时两个 PK 用户的答题过程可以不需要实时同步. 但是 PK 的评阅结果需要 PK 双方都答题结束后才可返回, 即当其中一个 PK 用户交卷之后, 必须要等待另一个用户也交卷才能将两人的 PK 结果返回.

当两个 PK 用户都申请交卷之后, 后台会将两个用户的 PK 结果进行比较, 并返回胜负情况.

6 PK 答题模块功能测试

根据上文 PK 模块的设计目标, 该模块的测试范围主要围绕功能实现、界面操作、吞吐量等几个方面.

6.1 客户端功能测试

该 PK 答题模块是基于 B/S 架构的. B/S 架构的软件系统具有的特殊性, 决定了对其进行测试的技术也有所不同. B/S 结构的软件系统有以下特征: 客户端通常使用浏览器访问后台服务; 以网页表单的形式展示界面; 采用 B/S 架构的软件系统, 客户端一般只能完成浏览、查询、数据输入等简单的功能, 绝大部分工作由服务器承担^[23].

客户端功能测试涉及 3 个部分: 用户请求 PK 答题、PK 用户匹配及用户获取题目等页面功能的测试.

用户请求 PK 答题的主要功能是向服务器端发送 WebSocket 连接请求. 对于发送 WebSocket 请求的测试方法是查看浏览器, 审查元素中是否提示有 WebSocket 连接成功的消息. 如果 WebSocket 请求连接成功则显示 101 状态码; 如果 WebSocket 请求没有连接成功, 则会出现两种连接失败的状态提示, 其中 403 表示该用户 Session 已失效, 需要用户重新登录, 500 表示服务器端发生错误, 连接失败.

表 1 是用户请求 PK 答题页面的测试情况, 经过反复测试, 用户请求 PK 答题部分功能正常.

PK 用户匹配主要是用来判断该用户是否找到了 PK 的对手. 要保证该功能可以正常工作, 首先要确保 WebSocket 连接请求成功. 在 WebSocket 连接成功的情况下, 如果服务器端同时也有人在等待, 则会返回匹配成功的提示信息, 否则返回正在等待匹配的提示信息.

表 1 请求 PK 答题页面的测试结果

Tab. 1 Request the test results of the PK answer page

测试内容	测试数据	期望结果	实际结果
用户已登录, 发送 PK 请求	点击 PK 请求按钮	返回 101 状态码, 连接成功	正常
用户未登录, 发送 PK 请求	点击 PK 请求按钮	返回 403 状态码, 拒绝访问	正常
服务器关闭, 发送 PK 请求	点击 PK 请求按钮	返回 500 状态码, 服务器异常	正常

图 5 是匹配页面测试数据及结果. 经多次测试, 结果表明匹配页面功能无异常.



图 5 PK 模块匹配页面

Fig. 5 PK module match pages

用户获取题目的主要作用是在用户匹配成功后, 获得后台随机发送的题目. 当用户匹配成功后就能够获取 PK 习题. 每次发送五题, PK 双方接收到的习题应该相同, 并且不同 PK 用户组之间的题目需要不同.

图 6 是用户获取题目页面测试数据及结果. 经多次测试, 结果表明用户获取题目功能无异常.



图 6 PK 答题页面

Fig. 6 PK page

6.2 服务器端测试

PK 答题模块服务器端的功能测试包括 WebSocket 连接请求、PK 用户匹配、题目发送等功能.

WebSocket 连接请求主要用于处理用户的 PK 连接请求, 构造请求回复报文来触发浏览器客户端中 WebSocket 对象的 onopen 事件以建立双向连接. 当用户与 WebSocket 服务器端断开连接时, WebSocket 服务器端也能够检测到. 测试方法是在 PK 模块服务器端启动时, 不停地让用户来请求连接, 并在浏览器客户端检测是否能成功建立连接, 在保持连接一段时间后, 由客户端主动断开连接. 为了检测服务器端连接处理模块工作的稳定性, 可以控制多个用户同时申请连接.

服务器端在使用 JMeter 压力测试工具下的测试数据: 图 7 为不同用户数量下服务器的平均响应时间; 图 8 为不同用户数量下服务器吞吐量.

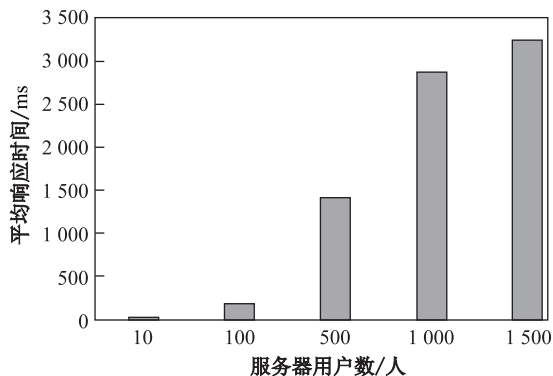


图 7 不同用户数量下的平均响应时间

Fig. 7 Average response time for different users

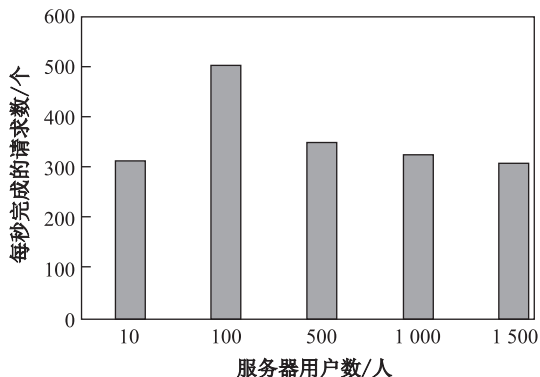


图 8 不同用户数量下吞吐量

Fig. 8 Throughput of different users

从图 7、图 8 可以看出, 当用户数量增多时, 服务器对用户的平均响应时间和服务器每秒能够完成的请求数会受到影响. 在服务器端用户数量增多时, 服务器对用户的平均响应时间会逐渐增加. 这是因为在网络条件相同的情况下, 服务器端用户数量越多, 在服务器中处理所需要的时间越长.

从图 8 可以看出, 当服务器端用户数量约为 100 人时, 服务器端每秒完成的请求数即吞吐量能达到一个较高水平; 当服务器端用户数量不断增加时, 服务器端的吞吐量达到一个较为稳定的

数值,每秒约为320次请求,单机情况下基本能达到千人同时访问的并发要求。

PK用户匹配的功能测试比较简单。当同时有多个浏览器客户端发送WebSocket连接请求,倘若服务器端能够接收到请求,则会向浏览器客户端返回匹配成功的响应。

题目发送的测试是基于PK用户匹配成功来进行的。在用户匹配成功之后,服务器端会随机生成一套题目,并将题目发送给匹配成功的两个用户。如果匹配成功的这一对用户能够收到服务器端发来的题目,则说明服务器端发送题目的功能正常。

7 总结与展望

7.1 总结

本文给出了一个在线答题系统中PK模块的设计与实现过程,该模块主要利用WebSocket来实现PK答题的功能,并利用MySQL作为目标数据库。

首先介绍了在线教育在互联网时代的重大意义,然后对Spring Boot、Bootstrap、WebSocket、Shiro、Redis等相关技术进行了详细的介绍。

之后通过对在线PK模块的需求分析,了解了用户在PK答题过程中的主要难点。通过用例分析,介绍了PK答题的基本流程,阐述了在用户进行PK答题时的基本环节和流程,建立了在线PK答题模块的基本框架。

在线PK答题模块的实现中详细介绍了PK模块中包括在线用户匹配、随机出题、发送试题、在线答题,以及返回PK结果等流程的实现细节。

7.2 下一步工作的展望

由于研究时间和相关条件的限制,对在线答题系统PK答题模块的研究还有很多后继工作需要探讨。

在线PK答题模块中会产生各种的数据,比如对用户的PK结果还需要进行分析和总结,让用户可以按照PK等级来选择PK对手,使PK系统更加完善。

在线答题系统应该是一个完整的系统,而不是一个独立的系统,应该可以与其他系统如学生管理系统、教务管理系统等紧密结合在一起,使功能更加完善。

最后,在线答题系统还应该紧跟电子商务的步伐,如提供在线支付的功能,为系统盈利提供技术支持。

[参 考 文 献]

- [1] 陈池,王宇鹏,李超,等. 面向在线教育领域的大数据研究及应用[J]. 计算机研究与发展, 2014, S1: 67-74.
- [2] MORENO-GER P, BURGOS D, MARTÍNEZ-ORTIZ I, et al. Educational game design for online education [J]. Computers in Human Behavior, 2008, 24(6): 2530-2540.
- [3] 管佳,李奇涛. 中国在线教育发展现状、趋势及经验借鉴[J]. 中国电化教育, 2014, 8: 62-66.
- [4] 王永和,张劲松,邓安明,等. Spring Boot 研究和应用[J]. 信息通信, 2016, 10: 91-94.
- [5] 汪云飞. JavaEE 开发的颠覆者: Spring Boot 实战[M]. 北京: 电子工业出版社, 2016: 122-124.
- [6] 张峰. 应用 Spring Boot 改变 Web 应用开发模式[J]. 科技创新与应用, 2017, 23: 193-194.
- [7] 杨家伟. 基于 Spring Boot 的 Web 设计与实现[J]. 轻工科技, 2016, 7: 86-89.
- [8] 贾英霞. 浅谈 Bootstrap 制作响应式网站布局[J]. 福建电脑, 2015, 8: 122-123.
- [9] 丁莲,张玲,杜巍. 基于 Bootstrap 的 Web 前端开发应用研究[J]. 电子制作, 2016, 20: 43.
- [10] 高榕岭. Bootstrap 在前端开发中的优势[J]. 计算机光盘软件与应用, 2015(1): 74+76.
- [11] PIMENTEL V, NICKERSON B G. Communicating and displaying real-time data with WebSocket [J]. IEEE Internet Computing, 2012, 16(4): 45-53.
- [12] CHEN B J, XU Z Q. A framework for browser-based multiplayer online games using WebGL and WebSocket [C]//International Conference on Multimedia Technology. IEEE, 2011: 471-474.
- [13] JIANG F Y, DUAN H C. Application research of WebSocket technology on Web tree component [C]//International Symposium on Information Technology in Medicine and Education. IEEE, 2012: 889-892.

- [14] 赵晗. 基于 WebSocket 协议的 Web 应用性能测试 [D]. 北京: 中国科学院大学, 2014.
- [15] 翁云翔. Java 安全框架 Shiro 在 Web 中的研究与应用 [D]. 武汉: 武汉邮电科学研究院, 2016.
- [16] 徐孝成. 基于 Shiro 的 Web 应用安全框架的设计与实现 [J]. 电脑知识与技术, 2015, 16: 93-95.
- [17] 荣艳冬, 冯建平. 深入探索 Shiro 权限框架的体系结构和实际应用 [J]. 电脑知识与技术, 2015, 22: 67-68.
- [18] 柳皓亮, 王丽, 周阳辰. Redis 集群性能测试分析 [J]. 微型机与应用, 2016, 35(10): 70-71.
- [19] 曾超宇, 李金香. Redis 在高速缓存系统中的应用 [J]. 微型机与应用, 2013, 32(12): 11-13.
- [20] 杨开杰, 刘秋菊, 徐汀荣. 线程池的多线程并发控制技术研究 [J]. 计算机应用与软件, 2010, 27(1): 168-170.
- [21] 封玮, 周世平. Java 中的线程池及实现 [J]. 计算机系统应用, 2004, 13(8): 16-18.
- [22] 李昊, 刘志镜. 线程池技术的研究 [J]. 现代电子技术, 2004, 27(3): 77-80.
- [23] 宋俊雅, 王鹏彪, 黄俊爽, 等. B/S 结构软件的系统测试技术 [J]. 科技信息, 2010, 10: 247-248.

(责任编辑: 李 艺)

(上接第 76 页)

- [8] CHEN Z B, SHEN H T, ZHOU X F. Discovering popular routes from trajectories[C]// 2011 IEEE 27th International Conference on Data Engineering (ICDE). 2011: 900-911.
- [9] DENG C Y, XUE L F, LI W Q, et al. The real-time monitoring system for inspecting car based on RFID, GPS and GIS[C]// 2010 International Conference on Environmental Science and Information Application Technology. IEEE, 2010: 772-775.
- [10] IQBAL U, ZAMIR S W, SHAHID M H, et al. Image based vehicle type identification[C]// 2010 International Conference on Information and Emerging Technologies. IEEE, 2010, Page(s): 1-5. DOI: 10.1109/ICIET.2010.5625675.
- [11] 卢晓春, 周欣, 蒋欣荣, 等. 基于网格化监控的套牌车检测系统[J]. 计算机应用, 2009, 29(10): 2847-2848.
- [12] 毛嘉莉, 金澈清, 章志刚, 等. 轨迹大数据异常检测: 研究进展及系统框架[J]. 软件学报, 2017, 28(1): 17-34.
- [13] LEI P R. A framework for anomaly detection in maritime trajectory behavior[J]. Knowledge and Information Systems, 2016, 47(1): 189-214.
- [14] ZHU J, JIANG W, LIU A, et al. Time-dependent popular routes based trajectory outlier detection[C]// International Conference on Web Information Systems Engineering-WISE 2015. Berlin: Springer International Publishing, 2015: 16-30.
- [15] PAN B, ZHENG Y U, WILKIE D, et al. Crowd sensing of traffic anomalies based on human mobility and social media[C]// Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. 2013: 344-353.

(责任编辑: 李 艺)