

毕业设计（论文）题目： 信息安全技术竞赛平台研制

学 院： 信息与电子学院

专 业： 信息对抗技术

班 级： 05611102

姓 名： 朱帅

指导教师： 罗森林

摘 要

随着信息安全技术竞赛在全球信息安全领域内的兴起，各种信息安全技术竞赛数量及参赛者数量也在逐年逐步稳定增长。每次竞赛的赛前准备、系统部署都会耗费大量的时间及人力物力，所以一套高可部署性、高可控性、高可扩展性的信息安全与对抗技术竞赛平台具有重要的实际应用价值。

本文针对目前竞赛系统人工部署耗时长、人力成本高、数据分析实时性差、可视效果不强等问题，提出了一种信息安全技术竞赛平台自动部署方法。该方法基于 B/S 架构，采用 MVC 框架和动态 Web 应用技术，支持弹性云计算技术。设计实现了一个信息安全与对抗技术竞赛快速部署系统。竞赛题目子系统和管理子系统相隔离支持系统的安全性和稳定性，利用软件工程思想保证系统的可扩展性，基于选择题库为选手随机生成试题、选手积分自动积分和排序、日志数据自动统计分析等方案实现平台的自动化管理。该系统具有用户注册与登录认证、用户管理、题库管理、在线答题及积分管理等功能。基于软件工程思想对平台进行了测试，结果表明，能够在 2 小时内完成竞赛平台的部署，再经过第 12 届信息安全与对抗技术竞赛的试用，平台已不间断稳定运行时间超过 30 天，支持超过 500 个用户并发操作，兼容 IE、FireFox、Google Chrome、Sogou、Safari 等 8 类浏览器。系统的应用有利于降低竞赛运行和维护的人力、物力、财力成本。

关键词：信息安全技术竞赛；信息安全；MVC 框架；动态 Web

Abstract

With the information security technology competition becoming popular in the global information security field, the number of entrants and the number of various information security technology competition are gradually steady growth year after year. Before each contest, system deployment will spend a lot of time and manpower, so a high deployability, high controllability, high scalability of information technology security and confront competition platform has important practical applications .

The current competition system deployed manually is time-consuming, high labor costs, poor real-time data analysis, visual effect is not strong and other issues. In this paper, a new information security technology platform competition automated deployment method. The method is based on B / S architecture, using MVC framework and dynamic Web application technology, support elastic cloud computing. Design and implement an information security technology competition and confrontation rapid deployment system. Contest subject subsystem and management subsystem to isolate the security and stability of the support system, the use of software engineering to ensure system scalability, based on the selection exam questions randomly generated for the players, the players integral automatic integration and sort log data automatically statistics analysis and other programs to automate the management platform. The system has a user registration and login authentication, user management, bank management, online questionnaires and integral management. Thought based software engineering platform has been tested, the results show that can be completed within two hours of deployment contest platform, and then after the 12th information security and technology race against trial, the platform has been uninterrupted uptime more than 30 days, support over 500 user concurrent operation, compatible with IE, FireFox, Google Chrome, Sogou, Safari browser and other eight categories. The application system will help reduce operation and maintenance of race human, material and financial costs.

**Keywords: Information Security and Countermeasures Technological Race;
Information Security;MVC Framework ;Dynamic Web**

目 录

第 1 章 绪论	1
1.1 研究背景和意义.....	1
1.2 研究历史和现状.....	1
1.2.1 发展历史.....	1
1.2.2 研究现状.....	4
1.2.3 问题总结与分析	5
1.3 研究内容和结构安排	5
1.3.1 研究内容.....	5
1.3.2 结构安排.....	5
第 2 章 涉及的理论与技术基础	7
2.1 引言	7
2.2 WEB 动态应用设计	7
2.2.1 静态页面设计方法.....	7
2.2.2 动态页面设计方法.....	8
2.3 模型-视图-控制器（MVC）框架	9
2.3.1 MVC 框架介绍.....	9
2.3.2 MVC 框架应用	11
2.4 数据存储方法.....	13
2.5 服务器防护方法.....	15
2.5.1 信息包过滤	15
2.5.2 透明代理.....	18
2.6 小结	20
第 3 章 系统设计与实现	21
3.1 引言	21
3.2 系统总体设计.....	21
3.2.1 技术路线和设计原则	21
3.2.2 目标和功能需求	21

3.2.3	系统总体结构	22
3.2.4	数据库设计	24
3.2.5	用户界面设计	28
3.3	主要功能需求说明	30
3.3.1	用户管理	30
3.3.2	题库管理	31
3.3.3	积分管理	31
3.3.4	页面管理	33
3.4	关键功能模块实现	34
3.4.1	用户管理	34
3.4.2	题库管理	34
3.4.3	积分管理	36
3.4.4	页面管理	37
3.5	系统测试分析	38
3.5.1	测试环境和条件	38
3.5.2	评价方法	39
3.5.3	测试过程	39
3.5.4	结果及分析	40
3.6	小结	43
第 4 章	结束语	44
4.1	全文总结	44
4.2	工作展望	45
致 谢	46
参考文献	47

图表索引

图 2-1 MVC 部件关系图	10
图 2-2 iptables 原理图	16
图 2-3 netfilter 原理图	17
图 2-4 透明代理示意图	19
图 3-1 系统架构图	22
图 3-2 系统功能结构图	23
图 3-3 普通用户界面	29
图 3-4 管理员界面	30
图 3-5 关卡答题流程图	32
图 3-6 选择题答题流程图	33
图 3-7 系统网络拓扑	39
表 1-1 历届 ISCC 数据统计	2
表 1-2 国内外部分著名信息安全技术竞赛	3
表 2-1 MySQL 操作命令	14
表 2-2 iptables 规则说明	18
表 3-1 数据库 config 表设计	24
表 3-2 数据库 challenges 表设计	24
表 3-3 数据库 choice_questions 表设计	25
表 3-4 数据库 papers 表设计	25
表 3-5 数据库 files 表设计	26
表 3-6 数据库 teams 表设计	26
表 3-7 数据库 pages 表设计	27
表 3-8 数据库 solves 表设计	27
表 3-9 数据库 wrong_keys 表设计	27
表 3-10 数据库 tracking 表设计	28
表 3-11 实验环境	38
表 3-12 实验软件	38
表 3-13 系统部署时间记录	40

表 3-14 并发性测试结果	41
表 3-15 兼容性测试结果	41
表 3-16 系统功能测试结果.....	42

第 1 章 绪论

1.1 研究背景和意义

随着信息技术的不断发展，计算机的使用越来越普遍，随之而来的信息安全问题也日渐突出。在当今这个信息高度发达的社会，信息安全已涉及到国家政治、经济、文化、社会和生态文明建设，随着信息系统向更高级的阶段发展，人们对信息的依赖性越来越强，在某种意义上信息系统更加容易被攻击，被攻击后带来的危害越来越严重。信息是一种重要战略资源，在社会发展的过程中起着不可替代的作用，世界各国关于信息的获取、使用 and 控制的斗争越来越激烈，信息安全在维护国家安全和稳定的过程中扮演着重要的角色，各国都在信息安全领域投入的大量的人力、物力、财力，在政策方面鼓励优先发展信息安全产业^[1]。

信息安全技术的发展离不开信息安全技术人员的努力与支持。在信息安全领域中，信息安全技术竞赛是信息安全技术人员之间相互切磋提高能力的一种方式。CTF 即夺旗赛作为信息安全技术竞赛的一种比赛形式，发展至今已经成为全球范围网络安全领域流行的竞赛形式。

随着信息安全技术竞赛在全球网络安全圈内越来越流行，竞赛数量及参赛者数量也在逐步稳定增长。每次竞赛的赛前准备、系统部署都会耗费大量的时间及人力物力，所以一套高可部署性、高可控性、高可扩展性的竞赛答题系统具有重要的实际应用价值。

1.2 研究历史和现状

1.2.1 发展历史

信息安全技术竞赛起源于 1996 年 DEFCON 国际黑客大会，作为一种新的技术竞赛方式来代替之前黑客们通过互相发起真实攻击进行技术比拼。

1996 年在第 4 届 DEFCON 全球黑客大会上举办第 1 届 CTF 夺旗赛，当时的比赛形式是参赛者之间互相攻击对方的服务，由竞赛组委会根据参赛者的攻防结果来进行评分。

1997 年第 2 届和 1998 年第 3 届 DEFCON CTF 竞赛中，参赛者可以通过两

种方式得分：第一种方式是提供攻击目标；第二种方式是攻击其他参赛者提供的目标。由于当时的竞赛规则还不太成熟，这种得分方式的灵活性让竞赛现场非常混乱。

1999 年第 4 届至 2001 年第 6 届的 DEFCON CTF 赛事中，参赛者不仅仅专注于比赛内容，他们还投入了大量的精力来攻破竞赛系统的服务器。

从 2002 年第 7 届 DEFCON CTF 竞赛开始，CTF 竞赛的形式、内容开始逐步成熟，以攻击其他参赛者的服务，加固防护自己的服务作为竞赛的主要内容。每一届的竞赛组委会都会在这个模型的基础之上打造竞赛平台，目的是为了维护一个公平的竞赛环境。最近几届的竞赛组织方在竞赛开赛之前都会对竞赛布局保密，参赛者不知道竞赛系统的网络结构、系统平台及评分方法。竞赛考察的技术内容范围包括加密算法、SQL 注入、逆向、缓冲区溢出等等。

2004 年北京理工大学信息系统及安全对抗实验中心罗森林教授在国内率先提出信息安全与对抗技术竞赛（ISCC）的概念，并于同年 5 月成功举办第一届 ISCC。ISCC 包括线上个人挑战赛和线下分组对抗赛两个环节。线上个人挑战赛环节以计算机信息网络为竞赛内容和考察重点，分为 BASIC、WEB、REVERSE、PWN、MISC 和 REALITY 六个关卡。关卡考察内容涉及 WEB 知识、ASP/PHP 脚本、缓冲区溢出、软件脱壳破解、系统漏洞利用、社会工程学等信息安全知识。ISCC 组委会会邀请在线上个人挑战赛中表现优秀的选手到北京理工大学参加线下分组对抗赛。线下比赛分为多个小组进行对抗，在封闭的真实对抗环境（包括 DMZ 区、数据区、开发测试区、内网服务区、终端区）中展开攻防角逐，充分展示了各位选手的个人水平和小组的协同合作能力。主要采取阵地夺旗、占领高地等模式进行攻防比拼。比赛时，各队伍通过对指定的服务器进行入侵攻击达到获取旗子的目标而得分，并在攻上高地后防御其他队伍的攻击。ISCC 每年举办一届，至今已经成功举办了 12 届，参赛人数也在逐年增长，历届 ISCC 数据统计如表 1-1 所示。

表 1-1 历届 ISCC 数据统计

	关卡数量	首页点击量	参赛人数
ISCC 2004	6	6124	131
ISCC 2005	14	11347	282
ISCC 2006	19	12794	230

ISCC 2007	32	20360	393
ISCC 2008	50	37524	300
ISCC 2009	40	48532	429
ISCC 2010	41	65214	446
ISCC 2011	31	115736	1802
ISCC 2012	31	187496	1953
ISCC 2013	31	204891	2146
ISCC 2014	46	218765	2258
ISCC 2015	26	235484	2342

2007 年 8 月，ISCC 进一步得到了教育部高教司、工业和信息化部人事司的肯定。经批准，在全国大学生电子设计竞赛中增设一项信息安全技术专题邀请赛，即增设《全国大学生电子设计竞赛信息安全技术专题邀请赛》，且于 2008 年起每两年举办一次，为全国大学生提供了更多的机会，对向全国范围普及和推动信息安全技术具有十分重要的作用。

随着信息安全技术竞赛的兴起，近些年来各种各样的信息安全技术竞赛逐渐发展起来，每年都会有各种各样的赛事举办，据统计 2013 年全球举办的信息安全技术竞赛超过 50 场。随着信息安全技术竞赛的成熟，每年参赛队伍都会定期参与信息安全技术竞赛，参赛队伍首先要参加线上资格赛，资格赛成绩排名靠前的队伍将会被竞赛组织方邀请参加现场决赛。在线上资格赛时，参赛选手通过答题来获取积分，线上赛主要考察的内容包括安全编程、数据隐写、加密解密、逆向、漏洞挖掘与利用、web 渗透等。现场决赛的模式是传统的攻防模式，在比赛开始时，竞赛组织方会给所有参赛队伍分配相同的一些服务，参赛选手的任务是对这些服务进行维护，对服务进行漏洞修补及加固来抵挡对手的攻击，同时可以攻击对手的服务来获取积分。

发展至今，各种信息安全技术竞赛的绝对数量一直在稳步增加。如表 1-2 所示，目前国内外有很多知名的信息安全技术竞赛：国际知名的 CTF 赛事有 DEFCON CTF、UCSB iCTF、XXC3 CTF 等；国内知名的 CTF 赛事有北理工的 ISCC、阿里巴巴公司的 AliCTF、西电的 XDCTF 等。

表 1-2 国内外部分著名信息安全技术竞赛

序号	竞赛名	举办单位
1	DEFCON CTF	DEFCON 黑客大会

2	UCSB iCTF	UCSB
3	XXC3 CTF	CCC 黑客大会
4	PHD CTF	俄罗斯 Positive Hacking Day 会议
5	RwthCTF	德国 OldEurOpe
6	Hack.lu CTF	卢森堡黑客会议
7	ISCC	北京理工大学
8	AliCTF	阿里巴巴集团
9	XDCTF	西安电子科技大学
10	0CTF	上海交通大学
11	360 信息安全技术大赛	360 公司
12	XCTF	清华大学
13	HCTF	杭州电子科技大学

1.2.2 研究现状

现在国际上主流的 CTF 比赛有 DEFCON CTF、UCSB iCTF，国内的 CTF 比赛有 ISCC、AliCTF、XDCTF 等。无论是国内还是国外的 CTF 比赛，它们的比赛模式基本上是一致的。在竞赛举办前，竞赛组织方都会发公告通知竞赛开赛时间，选手注册账号后先参加线上资格赛，线上资格赛结束后，竞赛组织方会根据参赛者在资格赛中的表现，邀请资格赛积分榜上排名考前的参赛者到现场参加决赛。

在线上资格赛阶段，竞赛组织方会提供一个网址，参赛选手通过这个网址进入竞赛平台，这个平台是一个动态 web 应用，具有题库管理、积分管理、用户管理、用户管理及在线答题等功能。线上资格赛考察的内容很广，包括安全编程、数据隐写、加密解密、逆向、漏洞挖掘与利用、web 渗透等。

现场决赛的模式是攻防赛，在竞赛开始时竞赛组织方会给每个队伍分配一台 VM，每台 VM 上运行着很多网络服务，这些服务大都存在着安全缺陷。每个服务中包含一个 flag，每个参赛队伍的 flag 不同，参赛队伍的任务就是获取其他参赛队伍服务中的 flag，或者把自己的 flag 写 入其他参赛队伍的服务中去。同时需要发现己方服务中的漏洞，对其修补加固来防止其他参赛队伍获取己方的 flag 或者将 flag 写入己方的服务中。在攻防赛中可以通过三种方式得分：第一种方式是 Breakthrough，第一个获取某一项服务的控制权的参赛队伍可以获得该项服务对应的分值；第二种方式是 Steal，参赛队伍获取某一服务中的 flag，并将 flag 提交到积分服务器上获取积分；第三种方式是 Overwrite，

参赛队伍在获取某一项服务的控制权限后，用自己的队伍的 flag 去覆盖其他队伍的 flag 来获取积分。

1.2.3 问题总结与分析

目前的信息安全技术竞赛使用的竞赛平台存在以下问题：赛前竞赛平台的部署采用人工部署的方式，完成系统部署耗时长、人力成本高；在并发数较大的情况下响应过慢甚至服务器崩溃，平台稳定性较差；安全防护措施少，容易给不怀好意的人破坏平台的机会，平台安全性较差；功能单一，可扩展性差；数据分析实时性差，可视效果不强，系统管理自动化程度较低。

1.3 研究内容和结构安排

1.3.1 研究内容

本文介绍了采用 B/S 架构，基于 MVC 框架实现一套高可部署性、高可控性、高可扩展性的竞赛答题系统。论文的主要研究内容包括以下几个方面：

1. 介绍 MVC 框架的原理，总结 MVC 框架的优点，研究使用 Flask 框架来提高系统的稳定性的方法；
2. 介绍透明代理的原理，研究使用 Squid 代理服务器及 iptables 提高系统安全性的方法；
3. 研究使用动态 web 技术和 python 脚本实现题库自动管理、选手积分自动积分和排序、日志数据自动统计分析的方法；
4. 基于以上方法实现信息安全技术竞赛平台，平台具备良好的安全性、稳定性和高度自动化；
5. 研究一套快速部署信息安全技术竞赛平台的方法，包括平台部署和题库部署及平台配置。

1.3.2 结构安排

论文共分四章：第 1 章是绪论部分，主要内容是对研究背景、目的和意义进行论述，同时对国内外研究历史和现状做简要分析；第 2 章论述理论基础和相关技术，主要是对 WEB 设计、MVC 框架、数据存储及服务器防护相关技术进行介绍；第 3 章提出一种基于 MVC 框架的信息安全技术竞赛平台的研究方

法，实现一个信息安全技术竞赛平台，并对平台的各项功能进行测试，分析测试结果；第 4 章结束语，对全文进行总结和工作的展望。

第 2 章 涉及的理论与技术基础

2.1 引言

本系统采用 B/S 架构，所涉及的知识包括 Web 设计、系统框架、数据存储及服务器防护等方面。本章对系统涉及到的几个技术进行介绍，主要内容包括以下几个方面：Web 动态应用基础知识及设计方法；MVC 框架介绍及使用方法；数据存储方法；服务器防护方法。

2.2 web 动态应用设计

2.2.1 静态页面设计方法

在 WEB 设计中，静态页面是一种最简单的页面，静态页面不需要从其他地方获取数据，它的所有内容都记录在 HTML 中。静态页面不是指其页面上的内容看起来是静态的，在静态页面上可以出现各种像 Flash 动画、GIF 动画之类的视觉动态效果。通常情况下静态页面的文件扩展名是.html、.htm、.shtml 等。

静态页面有如下特点：

1. 每个静态页面都有一个固定的 URL，其 URL 的后缀一般为.html、.htm、.shtml 等，作为与动态页面的区别，静态页面的 URL 中没有“？”；
2. 静态页面的内容相对比较稳定，因为静态页面中没有特殊代码，所以静态页面更容易被搜索引擎检索；
3. 静态页面的每一个页面都是存储在服务器上的一个独立的文件，无论用户是否访问，它们都存在；
4. 由于静态页面没有数据库的支持，所以静态页面网站的制作和维护工作量很大；
5. 静态页面的访问速度很快，因为静态页面不需要数据库工作；
6. 由于没有数据库的支持，静态页面的交互性比较差，对页面功能有很大的限制。

静态页面设计、制作的核心技术包括 HTML、CSS、和 JavaScript。

HTML 是用来描述网页的一种标记语言。HTML 使用标记标签来描述网页，

HTML 标记标签由尖括号和其包围的关键词组成。网页浏览器按顺序读取网页文件，同时根据 HTML 标记标签来解释和显示其标记的内容。

CSS 即层叠样式表，也叫级联样式表，是一种计算机语言，主要用来表现 HTML 或 XML 等文件样式。为了解决 Netscape 和 IE 这两种主要的浏览器不断将新的 HTML 标签和属性添加到 HTML 规范中，创建一个文档内容清晰地独立于文档表现层的站点越来越困难的问题。万维网联盟创造了样式表。样式表定义了 HTML 元素在浏览器上显示的样式，样式表通常保存在一个后缀名为.css 文件中，这个文件与 HTML 文件是分离开的。使用外部样式表的站点可以通过编辑 CSS 文件的内容，同时改变站点中所有页面的 HTML 元素的外观和布局。在 WEB 设计时可以给每一个 HTML 元素定义一个样式，然后将定义的样式应用于所有的页面中，如果要对全局样式进行升级时，只需要在外部样式表中编辑新的样式，站点中所有页面的元素都会自动的更新。

JavaScript 是目前最流行的脚本语言之一，被广泛应用于 WEB 应用客户端开发。网页可以通过 JavaScript 添加各种各样的动态效果，从而使页面更加美观。JavaScript 脚本通常是内嵌在 HTML 文件中。JavaScript 嵌入 HTML 页面后由浏览器来执行。

2.2.2 动态页面设计方法

动态页面是与静态页面相对应的一种网页设计方法。动态页面的 URL 不固定，可以通过后台程序与用户进行交互来完成请求、数据提交等动作^[2]。常用的语言有 Python、PHP、ASP、JSP 等。

动态页面有如下特点：

1. 动态页面的 URL 不固定，其后缀名一般为 .asp、.jsp、.php 等，与静态页面相比，动态页面的 URL 一个很大的特点就是它的 URL 中有个标志性的符号“?”^[3];
2. 动态页面以数据库技术为基础，能够很大程度的减少网站维护的工作量;
3. 动态页面具有很好的交互性，当用户在页面上操作后，页面会根据用户的操作类型实时改变和响应。由于交互性很好，和静态页面相比动态页面能实现更丰富的功能，例如账号注册、账号信息管理、在线问卷调查等等;
4. 动态页面不是独立存在于服务器上的网页文件，只有当用户请求服务时，

经过后台程序处理服务器才会返回一个完整的页面；

5. 由于动态页面的 URL 通常含有“？”，一般情况下搜索引擎不能访问一个网站数据库的所有页面，搜索引擎搜索时不会检索 URL 中“？”后面的内容，所以动态页面不容易被搜索引擎检索。

系统采用 B/S 架构，本质上是一个 Web 动态应用。系统采用 Python 作为后台程序编程语言，使用 HTML、CSS、JavaScript、JSON 等进行前端界面开发。

JSON(JavaScript Object Notation) 是一种轻量级的数据交换格式。它是基于 JavaScript 的一个子集。虽然 JSON 采用一种完全独立于语言的文本格式，但是也使用了一些类似于 C 语言的习惯。正是因为这些特性使得 JSON 成为理想的数据交换语言。

2.3 模型-视图-控制器（MVC）框架

2.3.1 MVC 框架介绍

MVC(Model View Control)是一种框架模式，它将应用程序的输入、处理和输出分开^[4]，使用 MVC 框架的应用程序被分为模型、视图和控制器三个核心部件。如图 2-1 所示，M 是指业务模型，表示应用程序的核心，在应用程序中用于处理应用程序数据逻辑；V 是指用户界面，在应用程序中处理数据显示；C 是控制器，处理数据输入，在应用程序中控制器主要用于处理用户交互。为了使同一个程序可以使用不同的显示形式，使用 MVC 将应用程序的模型和视图两个核心部件的实现代码分离。控制器作为三大核心部件之一是为了保证模型和视图的同步，视图随着模型的改变同步更新^[5]。

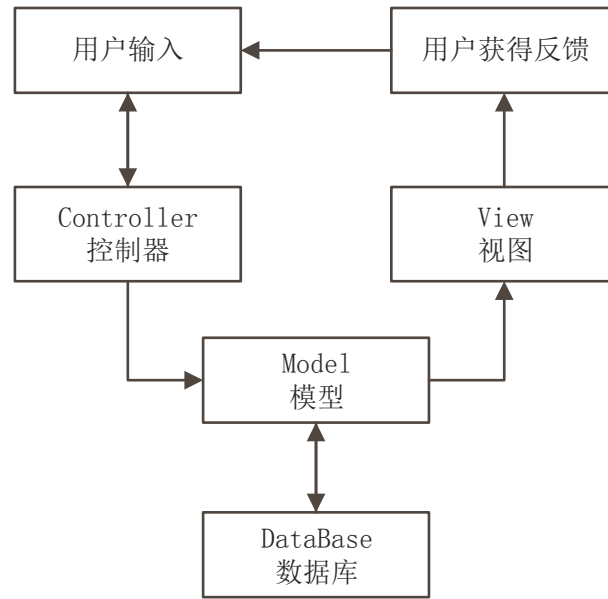


图 2-1 MVC 部件关系图

在 MVC 框架中，模型（Model）主要负责应用程序业务状态/流程的处理，并制定业务规则。业务处理的过程对于其他模块来说是透明的，模型接收视图发送的请求数据。如果模型发生变化，视图会得到通知，并且视图可以通过模型查询模型相关状态^[6]。同时，控制器可以通过模型访问封装在模型内部的应用程序功能。

视图（View）用来组织模型的内容，它负责提供用户交互界面。视图从模型那里获取数据并将这些数据按照制定的格式显示出来。当模型发生变化的时候，视图负责维持数据表现的一致性，并且将用户要求告知控制器。

控制器（Control）定义了应用程序的行为，它的主要作用是接收用户的请求。控制器对来自视图的用户请求进行解释，并把这些请求映射成相对应的行为，然后通过模型来实现这些行为。在一个 Web 应用程序中，用户请求的有很多种，比如来自网页的 GET、POST 或 HTTP 请求等^[7]。模型可以实现的行为分为两种，一种是修改模型的状态，另外一种是为处理业务，控制器根据用户请求和模型行为的结果选择一个视图作为对用户请求的应答。通常情况下一个控制器负责一组相关的功能集。

MVC 框架的处理用户请求包含三个过程^[8]：1）控制器接收用户请求，并根据用户请求的类型决定调用哪个模型来处理用户的请求；2）模型根据用户

请求进行相应的业务逻辑处理，并返回数据；3）控制器调用相应的视图格式化模型返回的数据，并通过视图呈现给用户。

MVC 框架具有如下几个优点：1）在一个应用程序中同一个模型可以被不同的视图重用，这样大大提高了代码的重复利用率；2）MVC 的三个模块相互独立，其中某一个模块的改变不会影响其他两个模块，根据这种设计思想能构造良好的松耦合的构件；3）控制器大大提高了应用程序的灵活性和可配置性，控制器可以用来连接不同的模型和视图去完成用户的请求，控制器为构造应用程序提供了强有力的手段。

2.3.2 MVC 框架应用

常见的 MVC 框架有 Struts、Spring MVC、Zend Framework、.NET 等，Flask 也是 MVC 框架的一种。Flask 是一个微框架（microframework），主要面向简单需求的小型应用。Flask 是一个 Web 应用框架，采用 Python 编写。Flask 的 WSGI 工具箱采用 Werkzeug，模板引擎使用 Jinja2^[9]。系统采用 Flask 框架，下面将根据 Flask 的使用方法分 9 个方面详细介绍 Flask 框架的内容。

1. 调试模式

在使用 Flask 框架开发 web 应用程序时，调试模式可以大大提高开发的效率，在调试模式下服务器会在修改应用之后自动重启，并且当应用出错时 Flask 还会提供一个有用的调试器。打开 Flask 调试模式的方法有两种：1）在应用对象上设置，代码为 `app.debug = True ;app.run()`；2）传递参数给 `run`，代码为 `app.run(debug=True)`。

2. 路由

现在的 Web 应用都使用规范便于人们记忆的 URL，Flask 使用 `route()` 装饰器将一个函数绑定到一个 URL，用户可以通过点击页面上的图标直接跳转到需要的 URL。

通过变量规则可以动态的改变 URL 的某一部分，把 URL 的一部分标记为 `<variable_name>` 就能在 URL 中添加变量。规则可以用 `<converter:variable_name>` 指定一个可选的转换器。

Flask 可以通过 `url_for()` 来给指定的函数构造 URL。`url_for()` 接受函数名作为第一个参数，也接受对应 URL 规则的变量部分的命名参数。未知变量部

分会添加到 URL 末尾，用来作为查询参数。

HTTP 协议里有不同的方法用于访问 URL，在默认情况下，一个 URL 只回应 GET 请求，使用其他方法需要通过 `methods` 参数设定。常用的 HTTP 方法有 GET、POST、HEAD、PUT 等。

3. 静态文件

动态 web 应用也使用静态文件，通常是 CSS 和 JavaScript 文件。在 Flask 应用开发过程中，只需要在包或者模块旁边创建一个名为 `static` 的文件夹，然后将静态文件放入其中就行，使用方法是 `url_for('static', filename='style.css')`。

4. 渲染模板

Flask 使用 Jinja2 模板引擎，`render_template()` 方法用于模板渲染，只需要提供模板名称并将需要的参数传递给模板变量即可。需要在包或者模块旁边创建一个名为 `templates` 的文件夹，然后将模板文件放入其中，Flask 会自动在 `templates` 文件夹中寻找模板。

5. 操作请求数据

Flask 中的请求信息由全局对象 `request` 提供。使用时首先需要从 `flask` 模块导入请求对象：`from flask import request`。使用 `methods` 属性可以设定不同的请求方法，然后通过 `form` 属性来处理表单数据。使用 Flask 上传文件的时候，需要在 HTML 表单中设置 `enctype="multipart/form-data"` 属性，否则浏览器根本不会传送文件。上传完成的文件存储在内存中或是文件系统中一个临时的位置。可以通过请求对象的 `files` 属性访问它们。Flask 访问 cookies 使用 `cookies` 属性，使用请求对象的 `set_cookies` 方法设置 cookies。

6. 重定向和错误

在 Flask 中使用 `redirect()` 函数可以实现重定向功能，使用 `abort()` 函数可以提前退出请求并返回错误代码。

7. 响应

Flask 的视图函数的返回值会被自动转换为一个响应对象。如果返回值是一个字符串，它被转换为该字符串为主体的、状态码 200 OK 的、MIME 类型是 `text/html` 的响应对象。

8. 会话

Flask 中除了请求对象之外，还有一 `session` 对象。`session` 对象允许在不同

请求间存储特定用户的信息。`session` 对象是在 `Cookies` 的基础上实现的，并且对 `Cookies` 进行密钥签名。用户可以查看 `Cookie` 的内容，但却不能修改它，除非用户知道签名的密钥。

9. 日志

在系统运行时可能会遇到数据出错需要纠正的情况，这时候就需要查看日志来分析错误原因，日志记录了系统运行时的每一个操作。`Flask` 配置好了日志工具 `logger()`，`logger()` 是一个标准的 `Python logger()` 类。

2.4 数据存储方法

无论什么平台、开发环境，数据都是核心。一个开发平台如果对数据的存储有良好的支持，这对应用程序的开发将会有很大的促进作用。总体来说，数据存储方式有三种：文件、数据库和网络^[10]。其中文件和数据库用的比较多，文件使用比较方便，可以通过程序自定义格式；数据库使用稍微复杂一些，但数据库具有很多优点，比如在存储海量数据时性能优越、具有查询功能、可以加密、可以跨平台等；网络存储一般用于比较重要的数据，比如科研、勘探、航空等实时采集的数据需要立即通过网络传输到数据处理中心进行存储并进行处理。

数据库是依照某种数据结构模型组织起来并存放在二级存储器中的数据集合。这种数据集合具有如下特点：数据尽可能不重复，以最优方式为某个特定组织的多种应用服务；数据结构独立于使用它的应用程序；对数据的增加、删除、修改、查询由统一的软件进行管理和控制。

数据库可以分为关系型数据库和非关系型数据库^[11]。关系型数据库即建立在关系模型上的数据库，关系型数据库借助集合代数等数学概念和方法来处理数据库中的数据。关系型数据库把所有的数据都通过行和列的二元表形式表现出来。典型的关系型数据库有 `Oracle`、`DB2`、`MySQL`、`Microsoft SQL Server`、`Microsoft Access` 等。

关系型数据库具有如下优势：事务处理时保持数据的一致性；由于以标准化为前提，数据更新的开销很小，相同的字段基本上都只有一处；可以进行 `JOIN` 等复杂的查询；其中能够保持数据的一致性是关系型数据库最大的优势。

关系型数据库的不足之处如下：不擅长大量数据的写入处理；不擅长为有

数据更新的表做索引或表结构变更；不适合字段不固定时应用；不擅长对简单查询需要快速返回结果的处理。

关系型数据库的应用十分广泛，能进行事务处理和表连接等复杂查询。而 NoSQL 数据库仅应用在一些特定的领域，基本上不用进行复杂的处理，但是 NoSQL 恰好弥补了关系型数据库的不足之处。

NoSQL 全称 Not Only SQL，泛指非关系型的数据库。NoSQL 数据库是为了解决大规模数据集合多重数据种类带来的挑战而产生的。典型的 NoSQL 数据库有 Oracle BDB、HBase、MongoDb、Neo4J 等。NoSQL 数据库不支持 JOIN 处理，各个数据都是独立设计的，很容易把数据分散在多个服务器上，因此减少了每个服务器上的数据量，在处理大量数据写入时变得更加容易，同样数据的读取操作也更加容易。

NoSQL 数据库具有如下特征：

1. 不需要预定义模式：不需要预先定义数据类型和表的结构；
2. 无共享架构：NoSQL 将数据划分后存储在各个本地服务器上；
3. 弹性可扩展：可以在系统运行时，动态增加或删除节点，数据自动迁移；
4. 分区：NoSQL 将数据进行分区，将记录分散在多个节点上；
5. 异步复制：NoSQL 中的数据复制往往是基于日志的异步复制；

NoSQL 适用于以下几种情况：数据模型比较简单；需要灵活性更强的信息系统；对数据库性能要求较高；不需要高度的数据一致性；对于给定的 Key 比较容易映射复杂值的环境。

本系统采用 MySQL 作为数据库管理系统。MySQL 数据库是一个关系型数据库管理系统，是目前最流行的关系型数据库管理系统。下面将介绍 MySQL 数据库的操作方法^[12]，如表 2-1 所示。

表 2-1 MySQL 操作命令

操作类型	操作命令
创建数据库	create database 数据库名;
使用数据库	use 数据库名;
创建表	create table 表名;
向表中插入数据	insert [into] 表名 [(列名 1, 列名 2, 列名 3, ...)] values (值 1, 值 2, 值 3, ...);
查询表中数据	select 列名称 from 表名称 [查询条件];

更新表中的数据	update 表名称 set 列名称=新值 where 更新条件;
删除表中的数据	delete from 表名称 where 删除条件;
在表中添加列	alter table 表名 add 列名 列数据类型 [after 插入位置];
在表中修改列	alter table 表名 change 列名称 列新名称 新数据类型;
在表中删除列	alter table 表名 drop 列名称;
表重命名	alter table 表名 rename 新表名;
删除表	drop table 表名;
删除数据库	drop database 数据库名;

insert 语句可以用来将一行或多行数据插到数据库表中，其中[]内的内容是可选的，按回车键确认后若提示 Query Ok, 1row affected (* sec) 表示数据插入成功。若插入失败请检查是否已选择需要操作的数据库。

selec 语句常用来根据一定的查询规则到数据库中获取数据，也可以使用通配符"*"查询表中所有的内容。select 语句可以按特定条件查询:where 关键词用于指定查询条件，用法形式为: select 列名称 from 表名称 where 条件。

where 子句不仅仅支持 "where 列名 = 值" 这种名等于值的查询形式，对一般的比较运算的运算符都是支持的，例如=、>、<、>=、<=、!= 以及一些扩展运算符 is [not] null、in、like 等等。还可以对查询条件使用 or 和 and 进行组合查询，后这里不再多做介绍。

2.5 服务器防护方法

2.5.1 信息包过滤

在 Linux 下防火墙用于访问控制，防火墙分为硬件防火墙和软件防火墙两种。无论在哪个网络中，防火墙都是工作在网络的边缘^[13]。可以定义防火墙策略来对出入网络的 IP、数据进行检测。

iptables 由 ipfirewall 演变而来，工作在用户空间中，是一款对数据包进行检测的访问控制工具。iptables 可以将多条规则组成一个列表，让这些规则共同发挥作用来实现绝对详细的访问控制功能。如图 2-2 所示，iptables 定义的规则可以让内核空间中的 netfilter 来读取^[14]，并且实现让防火墙工作。规则放入内核的地方必须是 TCP/IP 协议栈经过的地方，TCP/IP 协议栈必须经过的地方中可以实现读取规则的位置叫做 netfilter。

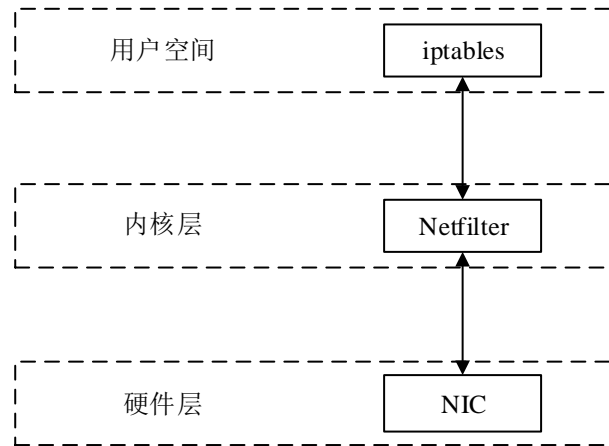


图 2-2 iptables 原理图

netfilter 的原理图如图 2-3 所示，在内核空间选取 5 个位置来作为控制的地方，这 5 个位置被称为五个钩子函数，也叫五个规则链^[15]：

1. PREROUTING，数据包第一次进入网络层时通过的位置，在这个位置进行网络地址转换；
2. INPUT，数据包入口，经过路由查找后，发送到本地的数据包通过的位置；
3. FORWARD，数据包转发，需要进行转发的数据包通过的位置；
4. OUTPUT，数据包出口，本地发送出去的数据包通过的位置；
5. POSTROUTING，数据包离开网络设备通过的位置，在这个位置进行源地址转换。

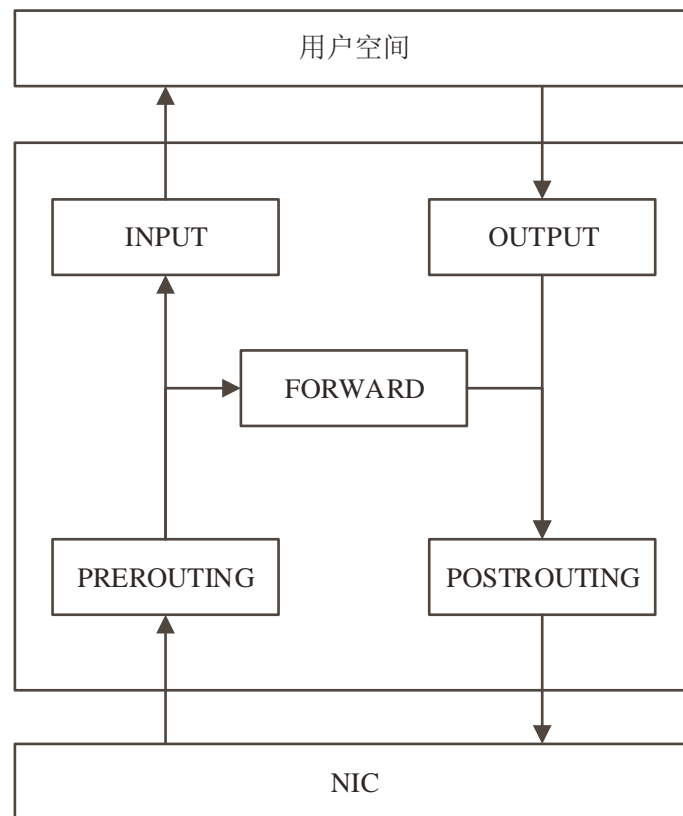


图 2-3 netfilter 原理图

防火墙策略一般分为两种：一种是通策略，默认端口是关着的，必须定义哪些数据包能进；一种是堵策略，默认端口是打开的，数据包必须要有身份认证，否则就不能进^[16]。所以需要定义一些策略，让需要进来的数据包进来，需要出去的数据包出去。当定义策略的时候，要分别定义多种功能，为了让这些功能交替工作，使用表来定义、区分各种不同的工作功能和处理方式。iptables 可以定义不同的表，每个表都包含几个内部的链，也能包含用户定义的链。每一个链都是一个规则列表，与对应得包进行匹配，每一条规则指定应当如何处理与之相匹配的包。现在使用比较多的功能有 3 个：1) Filter，缺省情况下使用的表，主要负责防火墙的功能，用于过滤流入流出本地的数据包，Filter 一般工作在 INPUT、FORWARD、OUTPUT 这三个规则链上；2) Nat，主要负责网络地址的转换，用于网络共享服务或特殊端口转换服务，Nat 一般工作在 PREROUTING、OUTPUT、POSTROUTING 这三个规则链上；2) Mangle，主要负责将报文拆开来修改报文的标志位，然后再重新将报文封装起来，Mangle 一般工作在 PREROUTING、INPUT、FORWARD、OUTPUT、POSTROUTING

这五个规则链上。

iptables 定义规则的格式为：

iptables [-t table] COMMAND chain CRETIRIA -j ACTION

表 2-2 对 iptables 规则中的每一个元素进行了说明。iptables 工作在用户空间，它可以让规则生效。iptables 本身不是一种服务，但现在 iptables 被做成了一个服务，可以控制规则的启用和停止。当 iptables 启动后，规则立即生效；当 iptables 停止后，规则失效。

表 2-2 iptables 规则说明

规则元素	作用
-t table	表名，filter、nat、mangle 等
COMMAND	定义如何对规则进行管理
chain	指定接下来的规则在哪一个链上操作，当定义策略的时候，可以省略
CRETIRIA	指定匹配标准
-j ACTION	指定如何处理

2.5.2 透明代理

透明代理即客户端根本不需要知道代理服务器的存在，代理服务器改变数据包并传送真实 IP，多用于路由器的 NAT 转发中^[17]。代理服务器可以进行内网地址和外网地址的转换，将内网细节屏蔽以防止非法人员探知内网结构。为了禁止用户使用容易造成攻击的不安全的命令，代理服务器提供了特殊的筛选命令，从根本上来抵御攻击^[18]。

透明代理的原理图如图 2-4。用户 A 和用户 B 并不知道代理服务器的存在，也就是说代理服务器对用户 A 和用户 B 是透明的，当用户 A 或用户 B 向服务器 A 或服务器 B 提交请求的时候，代理服务器根据设置的防火墙规则拦截并修改用户的报文，并作为实际的请求方向服务器 A 或 B 发送请求，当接收到服务器返回的数据时，代理服务器再根据自身的设置把允许的报文发回至用户 A 或 B，这些连接过程都是自动进行的，不需要客户端手工配置代理服务器，用户根本不会发现代理服务器的存在，所以对用户来说代理服务器是透明的。如果代理服务器设置禁止访问服务器 B，那么用户 A 或者用户 B 就无法

访问服务器 B。

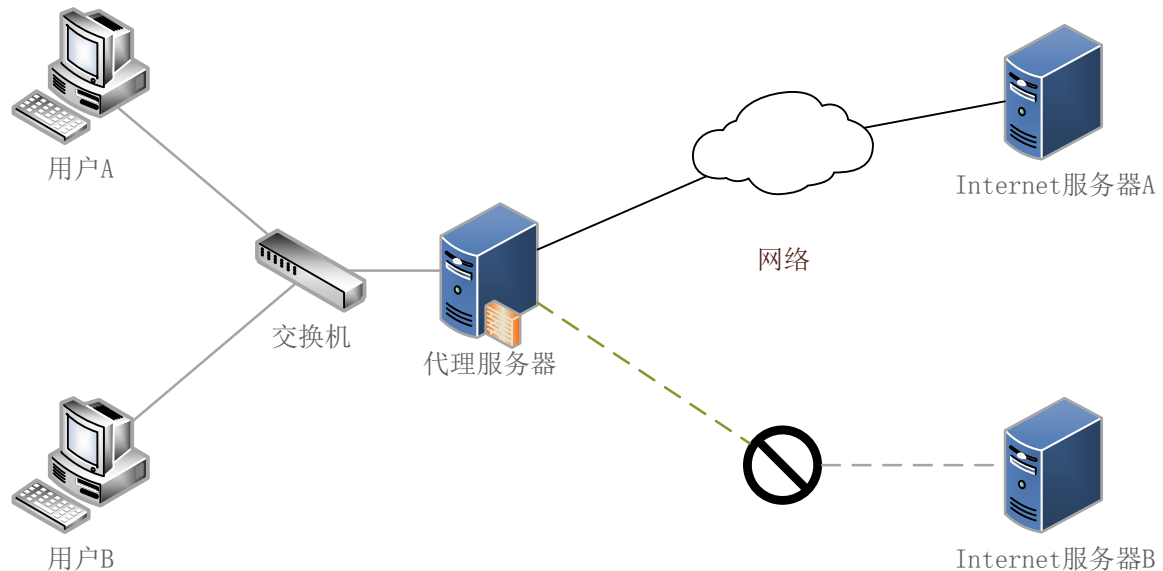


图 2-4 透明代理示意图

系统使用 Squid 透明代理服务器来保证系统安全，下面详细介绍 Squid 透明代理的步骤。首先编辑 squid 配置文件，配置过程如下：

```
vim /etc/squid/squid.conf
httpd_accel_host virtual    //指定虚拟主机模式
httpd_accel_port 80         //被加速主机端口
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
```

由于要访问的外网主机有许多台 virtual，即为虚拟的主机，virtual 指定了 Squid 使用虚拟主机模式，采用虚拟模式时，Squid 的缓存及 ICP 功能就会被禁用。httpd_accel_with_proxy 选项定义为 on 后，Squid 可以作为缓存代理服务器，同时可以为 Web 请求加速^[19]。httpd_accel_uses_host_header 选项定义为 on 后，Squid 针对要访问的主机使用主机头来识别不同的主机，这时透明代理服务器配置时必要的配置项。

Squid 配置完成后，在代理服务器上利用 iptables 配置端口重定向功能，配置过程如下：

```
#echo "1">/proc/sys/net/ipv4/ip_forward
//让流量在不同网卡之间流动，也可理解为开放路由功能

#iptables -t nat -F

#iptables -t nat -O POSTROUTING -o ppp0 -j MASQUERADE
//使 iptables 成为 NAT 服务器

#iptables -t nat -A PREROUTING -i eth1 -p tcp --dport80 -j REDIRECT --
to 3128
```

iptables 配置完成后，最后进行客户机网关配置^[20]，客户机网关必须配置成 squid 服务器的内网 IP 地址，另外可以通过 iptables 将 DNS 设置成智能 DNS 来减少客户机的工作量。

2.6 小结

本章从静态页面设计、动态页面设计设计两个方面介绍了 web 页面设计的基础知识及技术方法；对 MVC 框架的基本概念进行了介绍，详细介绍了 Flask 框架的使用方法；分关系型数据库和非关系型数据库两方面介绍了数据存储方法并详细介绍了 MySQL 数据库管理系统的使用方法；介绍了通过 iptables 和透明代理来防护服务器的理论基础并详细介绍了 Squid 透明代理服务器的使用方法。

第 3 章 系统设计与实现

3.1 引言

本章将介绍系统的功能要求、系统总体结构、系统数据库设计及用户交互界面设计。从系统设计、系统主要功能、关键功能模块实现和系统测试四个方面说明实现目标系统的过程及实现效果。

3.2 系统总体设计

3.2.1 技术路线和设计原则

系统采用 MVC 框架，对系统的各个功能进行模块化设计，在保证系统各个功能均能实现的基础上，减少不必要的功能、资源的使用，以减少 CPU 占用率、系统内存及其它相关硬件的使用。

3.2.2 目标和功能需求

目标系统要求能够简易完成部署、管理。系统功能需求如下：

1. 系统采用 B/S 架构，能够快速完成系统部署、题库管理，题库管理具备题库浏览、新建题目、修改题目、删除题目、生成试题功能；
2. 支持管理员用户和普通用户两类用户使用，其中管理员用户具有用户管理、题库查询、新建题目、修改题目、删除题目、设置题目可见性、新建页面、编辑页面、页面权限设置功能，普通用户具备用用户注册、用户信息修改、在线答题、页面浏览功能；
3. 具备积分管理功能，可以实现自动计分、积分自动排序、自动生成积分曲线的功能；
4. 具备全局设置功能，包括竞赛起止时间设置、选择题分值设置、选择题题量设置、用户注册方式设置、注册保护设置。

系统的性能要求指标如下：

1. 能够支持超过 3000 个用户同时在线；
2. 能够支持超过 500 个用户并行操作，在强并发情况下，系统能够快速响应用户请求；

3. 兼容 3 类以上浏览器。

3.2.3 系统总体结构

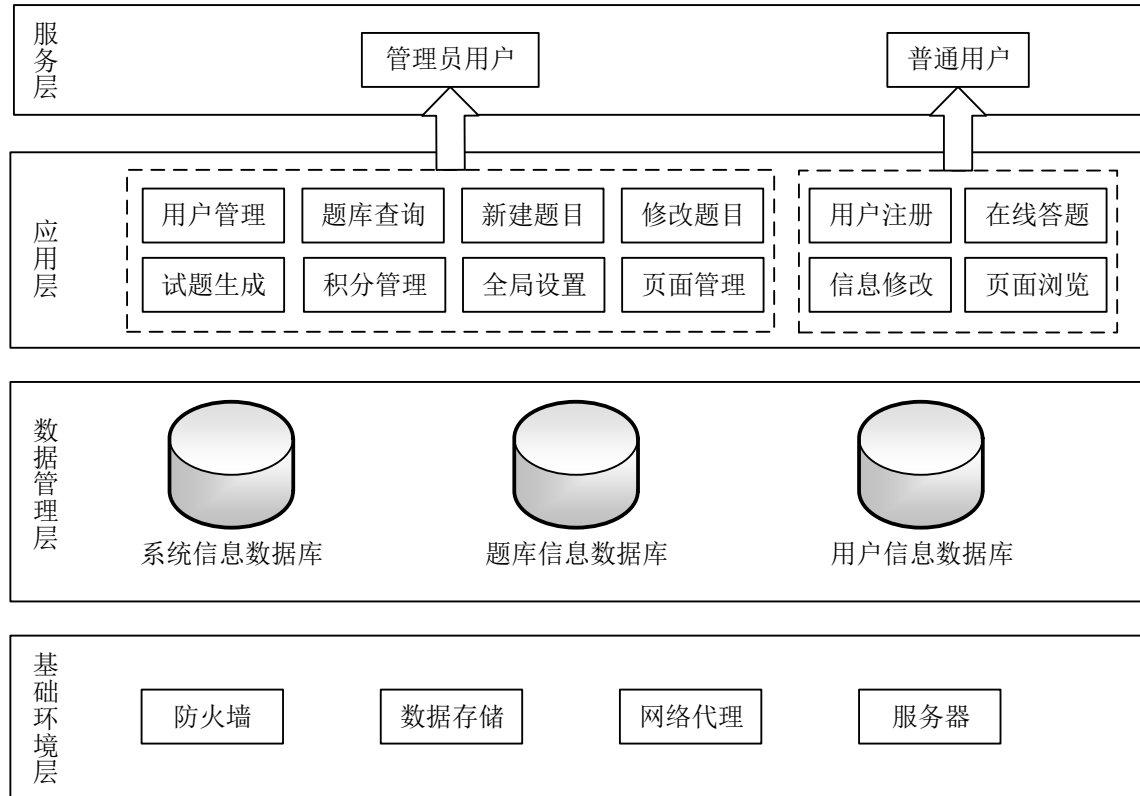


图 3-1 系统架构图

系统的结构自底向上分为基础环境层、数据管理层、应用层、服务层四层，系统架构图如图 3-1 所示。基础环境层分为四个部分：服务器操作系统采用 Ubuntu Server 系统；采用本地存储的方式存储数据；使用防火墙来对流入/流出系统服务器的数据包进行过滤；使用网络代理服务器保证系统内部网络的安全。数据管理层分为系统信息数据库、题库信息数据库、用户信息数据库三部分，使用 MySQL 数据库管理系统对它们进行统一管理。应用层是系统最核心的部分，应用层包含系统的所有功能。应用层的功能分为两类分别对应管理员用户功能和普通用户功能。管理员用户具有以下功能：用户管理，可以删除用户；积分管理，设置用户在积分板上是否可见；题库查询，浏览所有的关卡题和选择题题库；新建题目，包括新建选择题和新建关卡题；修改题目，可以修改选择题的题目、选项及正确答案，可以修改关卡题的题目描述、flag、附件

及关卡可见性；试题生成，在每位用户第一次进入选择题答题环节时，从题库随机抽题给用户生成一套试题；页面管理，包括新建页面、页面内容编辑、页面访问权限设置功能；全局设置，包括竞赛起止时间设置、选择题分值设置、选择题题量设置、注册方式设置和注册保护设置功能。普通用户具有以下功能：用户注册，普通用户可以根据系统的设置完成用户账号注册；信息修改，用户注册后可以修改账号注册邮箱、用户联系方式，并且可以通过注册邮箱修改密码；在线答题，包括关卡答题和选择题答题，在关卡答题环节用户可以下载题目附件、提交 flag；页面浏览，用户可以访问系统的竞赛规则、学习资料、系统公告、致谢页面，从页面上获取竞赛相关的信息。服务层包含两类用户，管理员用户和普通用户，他们对应应用层的两组功能。

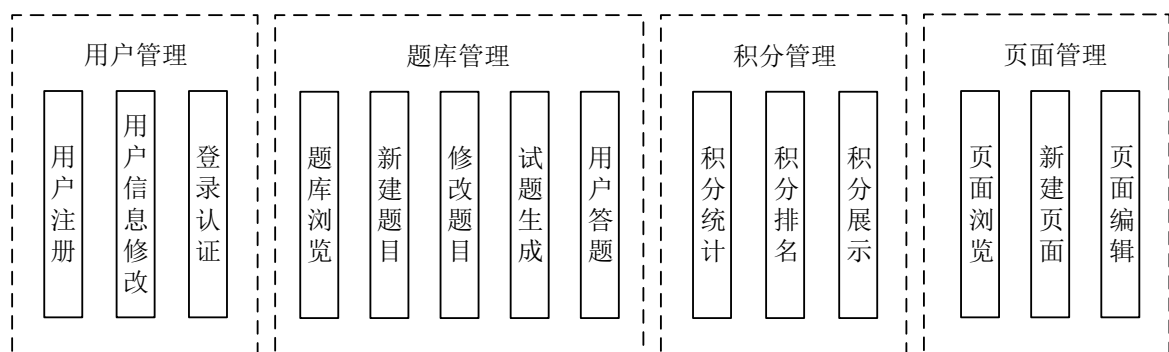


图 3-2 系统功能结构图

系统主要包括用户管理、题库管理、积分管理、页面管理四部分。系统功能结构图如图 3-2 所示。其中用户管理负责用户注册、用户登录认证以及用户信息修改的功能，用户注册有快速注册和邮箱注册两种方式；题库管理负责提供题库浏览、新建题目、修改题目、试题生成以及用户答题的功能，其中修改题目包括修改选择题的题目、选项及正确答案，修改关卡题的题目描述、flag、附件及关卡可见性；积分管理负责用户得分统计、用户积分排名以及用户积分展示；页面管理负责页面浏览功能、新建页面以及页面内容的编辑。

系统采用 MVC 框架来设计，控制器部分使用 Python 语言开发，对用户的输入请求进行处理，从数据库中调取相应的数据后，反馈到用户界面上，用户根据反馈信息进行下一步的操作。本系统使用 MySQL 数据库管理系统对系统数据进行管理，根据控制器的请求将数据以正确的格式返回。交互界面使用

html 语言、CSS 层叠样式表及 JavaScript 脚本语言进行开发。

3.2.4 数据库设计

本系统采用 MySQL 数据库管理系统，根据系统需求共设计了 10 个表：config、challenges、choice_questions、papers、files、teams、pages、solves、wrong_keys、tracking。下面将详细介绍这 10 个表：

config 记录全局设置相关信息，系统设计了 8 个全局变量，分别是：1) start，用于设置竞赛开始时间；2) end，用于设置竞赛结束时间；3) view_challenges_unregistered，用于非注册用户访问时，控制其浏览关卡的权限；4) prevent_registration，用于注册保护；5) set_password_by_email，用于设置用户注册方式；6) set，用于记录系统是否已初始化；7) score，用于设置选择题每道题目的分值；8) number，用于设置给每位用户分配的选择题数量。config 的详细设计如表 3-1 所示。

表 3-1 数据库 config 表设计

字段名称	数据类型	主键	描述
id	int(11)	是	
key	text		全局设置变量
value	text		全局设置变量值

challenges 记录关卡信息，challenges 中的每一条记录对应一个关卡，里面记录了关卡的详细信息，包括关卡名称、关卡描述、关卡分值等，challenges 的详细设计如表 3-2 所示。

表 3-2 数据库 challenges 表设计

字段名称	数据类型	主键	描述
id	int(11)	是	
name	varchar(80)		关卡名称
description	text		关卡描述
value	int(11)		关卡分值
flag	text		关卡通过 flag
visible	tinyint(1)		关卡对用户是否可见
visit	int(11)		关卡被浏览次数
submit	int(11)		关卡提交 flag 次数

passwd	int(11)		关卡通过次数
category	varchar(80)		关卡类型

choice_questions 记录选择题信息，choice_questions 中的每一条记录对应一道选择题，里面记录了选择题题目详细信息和正确答案，choice_questions 的详细设计如表 3-3 所示。

表 3-3 数据库 choice_questions 表设计

字段名称	数据类型	主键	描述
id	int(11)	是	
subject	varchar(255)		选择题题目
optionA	varchar(128)		选择题 A 选项
optionB	varchar(128)		选择题 B 选项
optionC	varchar(128)		选择题 C 选项
optionD	varchar(128)		选择题 D 选项
answer	varchar(10)		选择题正确答案

papers 记录为每一位用户生成的选择题试题信息，每一条记录对应一套试题，里面记录所有题目的题号、用户提交的选项、用户得分，还记录了用户当前答题题号，让用户在答题中途可以退出，下次登录时能继续答题，papers 的详细设计如表 3-4 所示。

表 3-4 数据库 papers 表设计

字段名称	数据类型	主键	描述
id	int(11)	是	
teamid	int(11)		试题所属用户 id 号
teamname	varchar(128)		试题所属用户用户名
questionlist	varchar(500)		试题题目列表
answerlist	varchar(255)		用户答案列表
num	int(11)		用户当前答题题号
value	int(11)		试题最终得分

files 记录了关卡的附件信息，每一条记录对应一个附件，里面记录了附件所属的关卡和附件的存储路径，files 的详细设计如表 3-5 所示。

表 3-5 数据库 files 表设计

字段名称	数据类型	主键	描述
id	int(11)	是	
chal	int(11)		文件所属关卡 id 号
location	text		文件存储路径

teams 记录了所有用户的相关信息，每一条记录对应一个用户，里面记录了用户的登录信息、用户的联系方式、用户的答题情况及得分情况等，teams 的详细设计如表 3-6 所示。

表 3-6 数据库 teams 表设计

字段名称	数据类型	主键	描述
id	int(11)	是	
name	varchar(128)		用户名
email	varchar(128)		用户注册邮箱
password	varchar(128)		账号密码
grade	varchar(128)		用户职业
phnumber	varchar(128)		用户手机号码
place	varchar(128)		用户工作单位
score	int(11)		用户总得分
last_score_time	datetime		用户最后得分时间
chall_score	int(11)		用户关卡答题得分
last_chall_score_time	datetime		用户关卡最后得分时间
choic_score	int(11)		用户选择题答题得分
last_choic_score_time	datetime		用户选择题最后得分时间
banned	tinyint(1)		用户是否在积分榜上显示
admin	tinyint(1)		用户是否是管理员

pages 记录了每一个页面的信息，每一条记录对应一个页面。系统设计了 6 个初始页面：1) index，竞赛平台首页；2) rule，竞赛规则页面，用于显示竞赛详细规则；3) choice，显示选择题答题规则并提供选择题答题环节入口；4) notice，竞赛公告页面，用于实时发布竞赛相关信息；5) study，学习资料页面，为用户提供竞赛相关学习资料的书籍目录及网页链接；6) thanks，致谢页面，用于显示竞赛举办方的相关信息。pages 的详细设计如表 3-7 所示。

表 3-7 数据库 pages 表设计

字段名称	数据类型	主键	描述
id	int(11)	是	
title	varchar(128)		页面名称
route	varchar(80)		页面路由
comment	varchar(128)		页面简介
html	text		页面内容
needauth	tinyint(1)		查看页面是否需要登录

solves 记录了用户每一次通关的相关信息，包括通关的关卡 id、通过该关卡用户的 id、通关时用户的 ip 地址及通关时间，solves 的详细设计如表 3-8 所示。

表 3-8 数据库 solves 表设计

字段名称	数据类型	主键	描述
id	int(11)	是	
chalid	int(11)		被解答的关卡 id 号
teamid	int(11)		解决该关卡用户的 id 号
ip	int(11)		用户的 ip 地址
date	datetime		用户解答关卡的时间

wrong_keys 记录了用户每一次提交错误 flag 的相关信息，包括用户 id、关卡 id、用户提交的错误 flag 以及提交时间，wrong_keys 的详细设计如表 3-9 所示。

表 3-9 数据库 wrong_keys 表设计

字段名称	数据类型	主键	描述
id	int(11)	是	
chal	int(11)		被提交错误 flag 的关卡 id 号
team	int(11)		提交错误 flag 用户的 id 号
date	datetime		用户提交错误 flag 的时间
flag	text		用户提交的错误 flag

tracking 记录了用户登录的浏览器信息，包括用户登录 ip 地址、用户使用的浏览器类型以及登录时间，tracking 的详细设计如表 3-10 所示。

表 3-10 数据库 tracking 表设计

字段名称	数据类型	主键	描述
id	int(11)	是	
ip	bigint(20)		用户的 ip 地址
useragent	text		用户使用的浏览器类型
team	int(11)		用户的 id 号
date	datetime		用户使用该浏览器登录的时间

3.2.5 用户界面设计

系统采用动态 Web 页面技术设计用户界面，采用 CSS 层叠样式表来进行页面布局和数据信息显示，保证页面的一致性。用户界面分为普通用户界面和管理员界面两部分。

普通用户界面如图 3-3 所示，界面菜单栏包括 8 个选项：ISCC，进入竞赛网站首页；竞赛规则，查看详细竞赛规则及竞赛主办方联系方式；选择题，用户可以浏览选择题答题规则，进入选择题答题环节；关卡列表，用户可以查看关卡列表，进入关卡答题环节；积分板，查看当前已注册用户中前 50 名用户的得分排名及前 10 名用户的得分曲线；公告栏，查看管理员在竞赛期间发布的通知公告；学习资料，查看竞赛主办方推荐的相关技术资料；致谢，查看竞赛主办单位及协办单位。



图 3-3 普通用户界面

管理员界面如图 3-4 所示, 管理员登录后首先点击界面右上角下拉菜单中的“管理”进入管理模式, 管理模式菜单栏包括 9 个选项: ISCC, 进入竞赛网站首页, 同时进入答题模式; 分析图表, 查看关卡答题统计分析图; 选择题管理, 查看选择题题库中的所有题目, 进行选择题修改、新建、删除操作; 关卡管理, 查看所有关卡, 进行关卡修改、新建、删除操作; 用户管理, 查看所有注册用户, 删除用户; 页面管理, 查看所有页面, 初始页面包括首页、竞赛规则、选择题、公告栏、学习资料和致谢, 进行新建页面、页面内容编辑、页面权限设置操作; 积分板, 查看所有注册用户的积分排名; 信息统计, 查看竞赛相关统计数据, 包括总关卡数、用户注册数、关卡 Flag 总错误/正确提交数、通过人数最多的关卡及通过人数、通过人数最少的关卡及通过人数; 全局设置, 查看当前的全局设置情况, 设置竞赛时间、选择题分值、选择题题量、用户注册方式等。

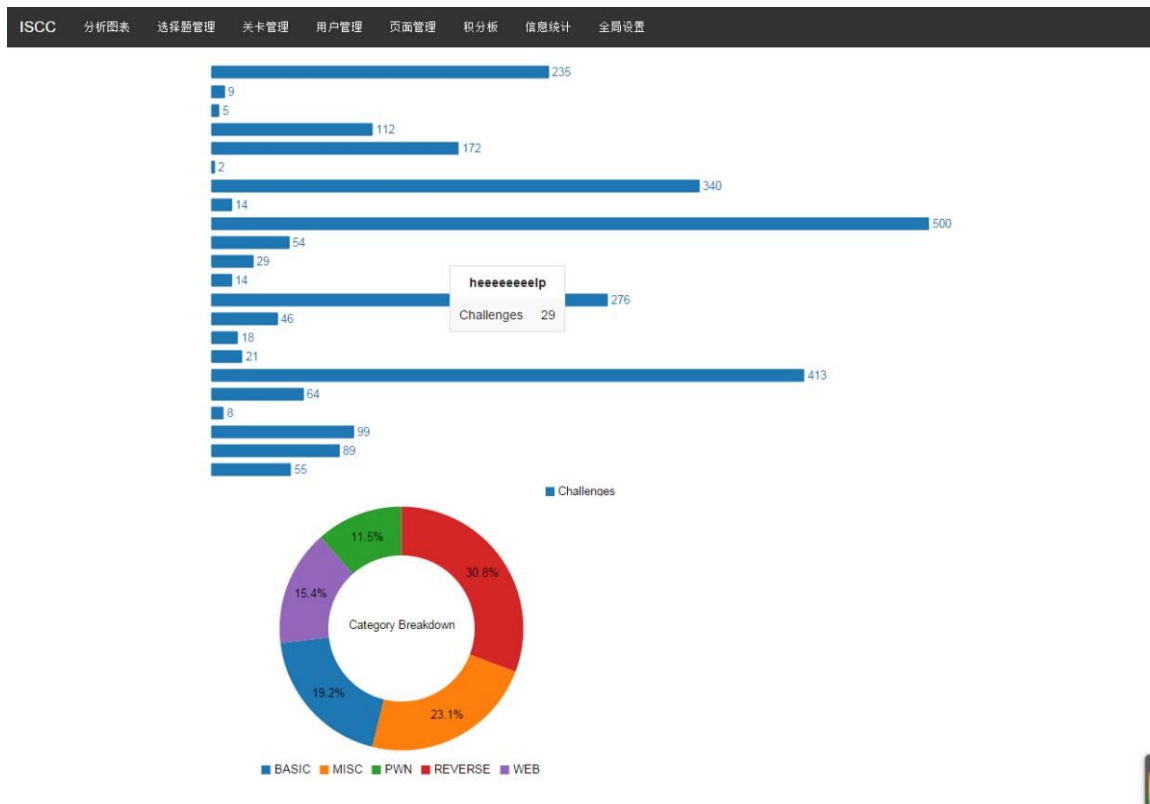


图 3-4 管理员界面

3.3 主要功能需求说明

3.3.1 用户管理

用户管理模块由用户注册、登录认证和信息管理三部分组成。

用户注册支持两种注册方式：普通注册和邮箱注册。普通注册也叫快速注册，这种注册方式适合只想先期体验竞赛平台的用户，用户只用输入一个独一无二的用户名，设定好密码后就可以完成注册。由于普通注册的身份认证缺失等可能导致安全性的问题存在，所以提供了邮箱注册方式，用户在注册账号时需要填写自己的邮箱，随后系统会将一个注册验证链接发送到用户邮箱，用户打开验证链接完成注册^[21]。邮箱注册有几点好处：1、可以避免很多恶意注册；2、可以及时公布信息，方便与用户之间的联系；3、当用户忘记密码时可以通过邮箱来找回密码。

在用户登录认证时，采用 MD5 加密技术对用户密码进行加密。当用户登录的时候，系统对用户输入的密码进行 MD5 Hash 运算，然后将运算的结果和

保存在文件系统中的 MD5 值进行匹配，如果匹配成功，就能通过登录认证。通过采用 MD5 加密技术，系统不需要存储用户密码的明码，只需要存储用户密码的 MD5 值就可以确定用户登录系统的合法性。通过这种方式避免用户的密码被管理员知道，杜绝了密码泄露的隐患。用户信息管理提供用户信息修改、查询等功能。

3.3.2 题库管理

题目管理模块由竞赛题库、分值设计和赛题录入三部分组成^[22]。竞赛题库包括历年 ISCC 竞赛的题目及每次竞赛新出的题目。根据题目难度的大小给题库中的每个题目设定不同的分值，分值大小与题目难度系数成正比。在竞赛时根据需要从竞赛题库中选题进行赛题录入。赛题分为关卡题和选择题，关卡题管理模块具有新建题目、修改题目、删除题目、附件上传以及题目浏览控制的功能。选择题管理模块具有题库浏览、新建题目、修改题目、删除题目及试题生成的功能，在用户第一次进入选择题答题环节时，系统根据管理员设置，从题库中随机挑选一定数量的题目生成一套试题供用户作答。

3.3.3 积分管理

积分管理模块由答题积分管理、奖励积分管理和积分排名展示三部分组成。

答题积分管理分为两部分：关卡答题积分管理和选择题答题积分管理。关卡答题积分管理中，当用户提交正确的 flag 后，系统会将对应关卡的分值加入到用户的总得分中，同时记录答题的时间点。如图 3-6 所示，在选择题答题积分管理中，在用户答题过程中，用户的得分系统会实时记录，但不会加入用户的总得分中，只有当用户答完试题中的所有题目后，得分一次性加入到用户总得分中。

图 3-5 为答题积分管理流程图。用户登录成功后选择题目解答，答题完成后向系统提交结果，然后系统对用户提交的结果进行判定。如果结果正确，用户获得积分，然后选择是否继续答题；如果结果错误，系统会提示用户答题错误并询问用户是否继续答题，如果用户继续答题，可以选择继续解答原题或者重新选题。

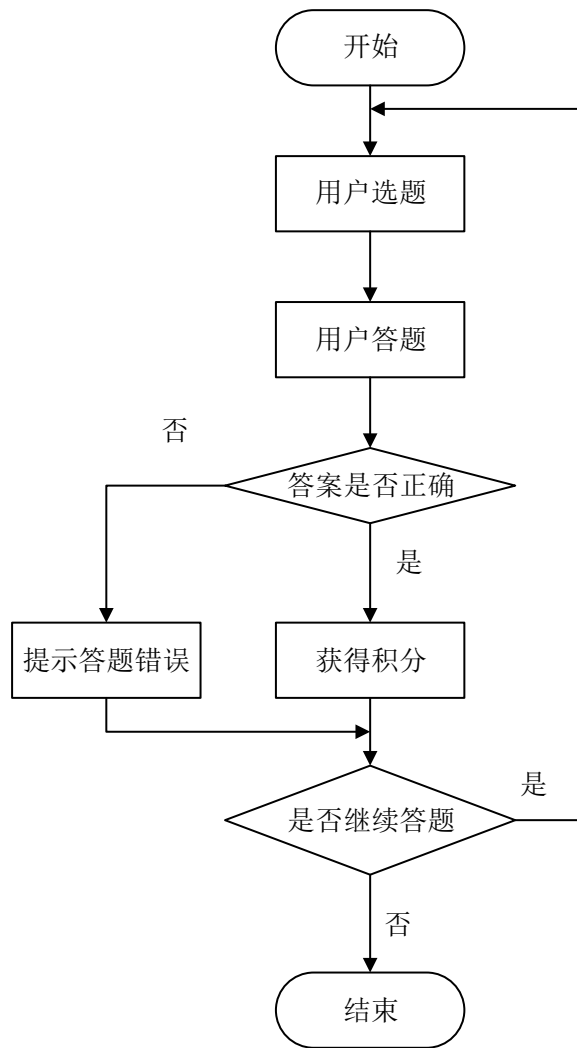


图 3-5 关卡答题流程图

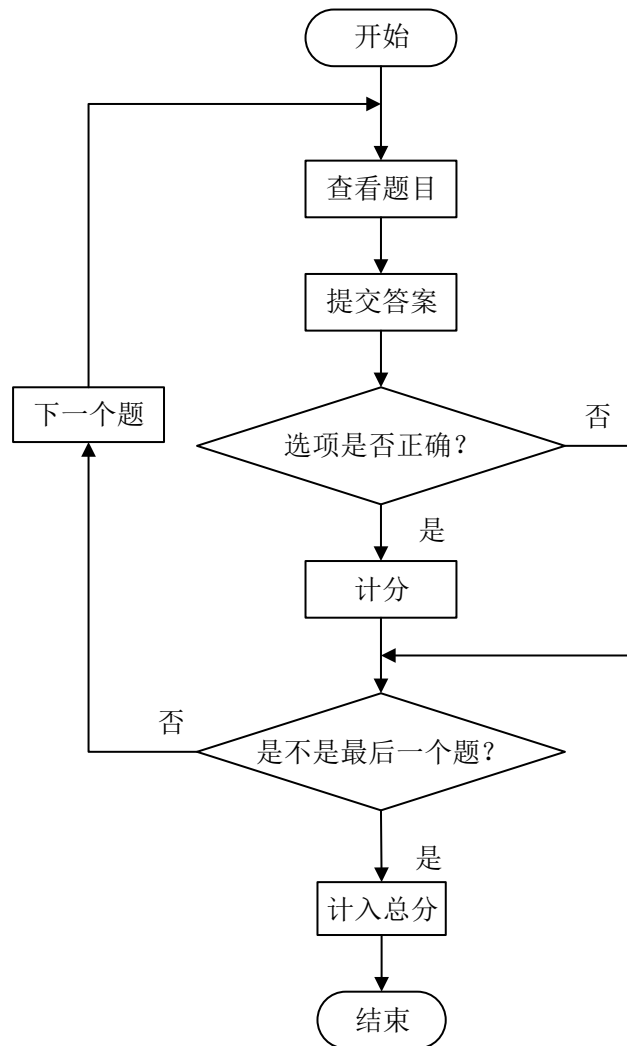


图 3-6 选择题答题流程图

奖励积分是当用户发现竞赛答题系统的漏洞或者给系统提出改进建议并被采纳后，给予用户一定的奖励积分，并计入总积分中。排名展示用于实时展示用户的积分排名，便于用户了解自己的排名状况。积分多的用户排名靠前，相同积分的用户根据其最新得分时间排名，最新得分时间较早的用户排名靠前。

3.3.4 页面管理

在系统中，页面管理模块管理的对象是用于信息展示的相关页面^[23]。系统初始化时，页面管理模块管理的页面包括：1）竞赛平台首页；2）竞赛规则，用于显示竞赛详细规则；3）选择题答题环节首页，显示选择题答题规则并提供选择题答题环节入口；4）竞赛公告，用于实时发布竞赛相关信息；5）学习

资料页面，为用户提供竞赛相关学习资料的书籍目录及网页链接；6）致谢页面，用于显示竞赛举办方的相关信息。

页面管理模块包括以下功能：新建页面、页面内容编辑、删除页面及页面浏览权限设置。管理员可以根据竞赛需要新建页面，设置页面浏览是否需要登录。

3.4 关键功能模块实现

3.4.1 用户管理

用户管理模块负责用户注册、用户信息修改、用户登录认证以及用户密码找回功能。用户如果需要使用本系统，需要先注册一个账号。当用户忘记密码时可以通过注册邮箱重置密码。用户管理相关函数集中在 `auth.py` 文件中，下面对用户管理模块的核心代码进行介绍：

`register()`：用户注册函数，往数据库写入用户注册信息，并向用户发送邮件，让用户通过邮件中的链接设置密码，核心代码如下：

```
db.session.add(team)          //添加用户信息到数据库
msg=Message("Message from {0}".format(app.config['CTF_NAME']),
sender = app.config['ADMINS'][0], recipients = [addr])
mail.send(msg)                //发送密码设置邮件
bcrypt_sha256.encrypt(request.form['password'].strip()) //用户密码加密
```

`login()`：用户登录认证函数，核心代码如下：

```
team=Teams.query.filter_by(name=request.form['name']).first()
//获取用户信息
bcrypt_sha256.verify(request.form['password'], team.password)
//登录密码验证
```

3.4.2 题库管理

题库管理模块分为关卡管理模块和选择题管理模块，关卡管理模块的功能

包括创建关卡、修改关卡等，选择题管理模块的功能包括题库浏览、创建题目、修改题目、删除题目和试题生成。关卡管理模块的核心代码在 `admin.py` 和 `challenges.py` 文件中，选择题管理模块的核心代码在 `admin.py` 和 `auth.py` 文件中。下面对题库管理模块的核心代码进行介绍：

`chals()`：获取所有关卡信息，核心代码如下：

```
Challenges.query.add_columns('id','name','value','description','category').o  
rder_by(Challenges.value).all()    //获取所有关卡信息
```

`admin_create_chal()`：新建关卡，核心代码如下：

```
chal=Challenges(request.form['name'],request.form['desc'],request.form['v  
alue'], request.form['category'])  
db.session.add(chal)    //新建关卡，将关卡信息写入数据库
```

`admin_update_chal()`：关卡修改，核心代码如下：

```
challenge=Challenges.query.filter_by(id=request.form['id']).first()  
//获取指定关卡信息
```

`chal()`：对用户提交的 flag 进行判断，核心代码如下：

```
keys = Keys.query.filter_by(chal=chalid).all() //获取关卡的所有 flag  
key = request.form['key'].strip().lower()    //获取用户提交的 flag
```

`admin_files()`：处理管理员上传的附件

```
db_f = Files(chalid, os.path.join(app.config['UPLOAD_FOLDER'],  
md5hash, filename))  
db.session.add(db_f) //将附件添加到数据库中
```

`admin_choice()`、`admin_newchoice()`、`admin_update_choice()`、`admin_delete`

_choice()分别对应选择题管理部分的题库浏览、新建题目、修改题目以及删除题目，核心代码如下：

```
questions = ChoiceQuestions.query.all()    //获取题库所有题目
question=ChoiceQuestions(subject,optionA,optionB,optionC,optionD,answer)
db.session.add(question)    //将新建的题目写入数据库
question = ChoiceQuestions.query.filter_by(id=id).first() //获取指定题目
db.session.delete(question)    //删除题目
```

paper(): 给用户生成选择题试题并对用户提交的答案进行判断，核心代码如下：

```
num = str(number.value)
papers = db.engine.execute("select * from choice_questions order by rand()
limit" + " "+num+"")    //随机生成试题
paper = Papers(team.id,team.name,questionlist,answerlist,0,0)
db.session.add(paper)    //将试题信息写入数据库
paper = Papers.query.filter_by(teamname = team.name).first()
//获取指定试题
```

上述代码中的 number.value 是生成的选择题题量，这个数值可以在全局设置中修改，questionlist 和 answerlist 是两个按照特定规则生成的特殊的字符串，里面分别记录了试题中题目的题号 and 用户提交的选项。

3.4.3 积分管理

积分管理模块主要用于用户的积分统计以及用户积分排名展示，其中积分展示部分包括积分板和积分曲线展示。积分管理模块的核心代码在 admin.py 和 scoreboard.py 两个文件中。下面对积分管理模块的核心代码进行介绍：

scores(): 对所有用户进行积分排序核心代码如下:

```
user = Teams.query.filter_by(id=teamid).first()
user.banned = 1    //禁止用户得分在积分版上展示
user = Teams.query.filter_by(id=teamid).first()
user.banned = None //允许用户得分在积分板上展示
```

topteam(): 获取的分排名前十的用户, 绘制他们的得分曲线, 核心代码如下:

```
score = db.func.sum(Challenges.value).label('score')
teams=db.session.query(Solves.teamid,Teams.name,score,db.func.max(Solves.date).label('quickest')).join(Teams).join(Challenges).filter(Teams.banned==None).group_by(Solves.teamid).order_by(score.desc(),Solves.date).limit(count)    //获取的分排名前十的用户信息
```

ban()和 unban(): 用于设置用户得分是否在积分板上展示, 核心代码如下:

```
user = Teams.query.filter_by(id=teamid).first()
user.banned = 1    //禁止用户得分在积分版上展示
user = Teams.query.filter_by(id=teamid).first()
user.banned = None //允许用户得分在积分板上展示
```

3.4.4 页面管理

页面管理模块用于管理用于信息展示的相关页面, 页面管理模块的功能包括新建页面和页面内容编辑。页面管理模块的核心代码在 admin.py 文件中。下面对页面管理模块的核心代码进行介绍:

admin_pages(): 用于新建页面和修改页面内容, 核心代码如下:

```
page = Pages.query.filter_by(route=route).first()    //获取指定页面信息
page = Pages(title, route, comment, html)
db.session.add(page)    //新建页面
```

3.5 系统测试分析

测试系统部署所有时间，测试系统的并发操作性能，测试系统可靠性，测试系统应用浏览器的兼容性等

3.5.1 测试环境和条件

实验环境如表 3-11 所示，实验软件如表 3-12 所示。

表 3-11 实验环境

操作系统	Linux Ubuntu Server(X64)
处理器	2.4GHz×8
内存	8GHz

表 3-12 实验软件

软件名	版本号
APScheduler	v3.0.1
Python	v2.7
Flask	v0.10.1
Flask-Mail	v0.9.1
Flask-SQLAlchemy	v2.0
Flask-Session	v0.1.1
SQLAlchemy	v0.9.8
passlib	v1.6.2
py-bcrypt	v0.4
six	v1.8.0

测试环境网络拓扑图如图 3-7 所示，竞赛平台服务器和脚本关卡服务器处于内网之中，用户通过代理服务器来访问竞赛平台。代理服务器使用 Squid 作为 web 服务代理，同时在 Squid 中设置防火墙规则 iptables 来控制访问以保证竞赛平台的安全^[24]。

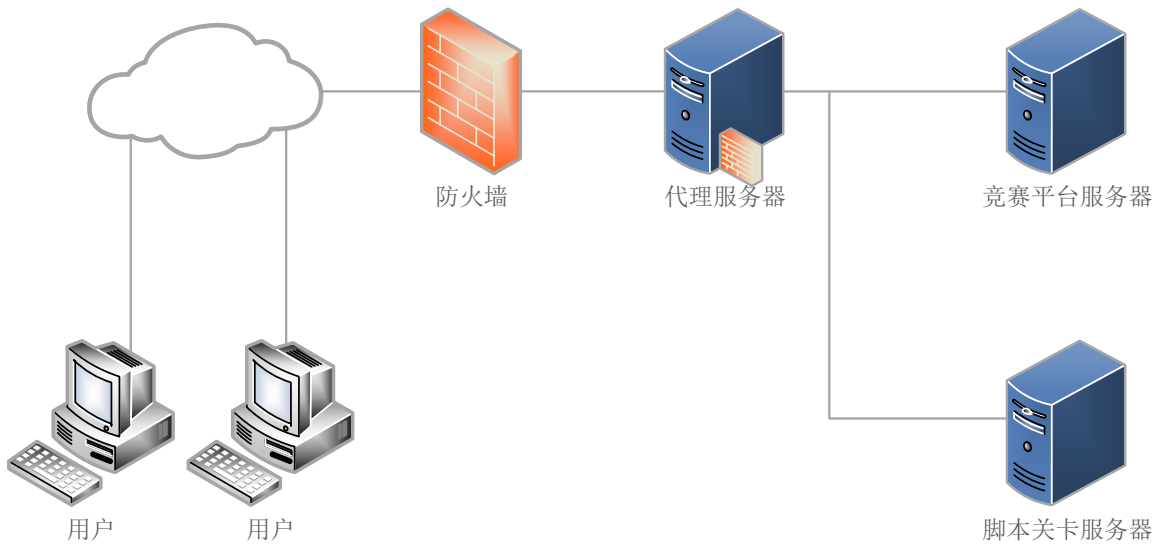


图 3-7 系统网络拓扑

3.5.2 评价方法

系统共进行五项测试，分别是可部署性测试、并发性测试、稳定性测试、兼容性测试、功能完整性测试。这五项测试分别对应五个评价指标：系统部署耗时(t)、平均响应时间(\bar{t})、持续稳定运行时间、支持浏览器种类数、系统功能完成度。

系统部署耗时 t = 系统部署结束时间 t_2 - 系统部署开始时间 t_1

$$\text{平均响应时间 } \bar{t} = \frac{\sum \text{每个请求响应时间 } t_i}{\text{并发用户总数 } n}$$

$$\text{系统功能完成度 } \eta = \frac{\text{系统实现功能数 } n_2}{\text{系统设计功能数 } n_1} \times 100\%$$

3.5.3 测试过程

1. 系统可部署性测试

在系统功能测试和性能测试之前首先进行系统部署，在系统部署时记录开始部署系统的时间和完成系统部署的时间，从而得到系统部署耗时。

2. 系统并发性测试

使用 Microsoft 的 Web Application Stress Tool (WAS) 这个工具软件进行并发性测试，WAS 能有效测试一个网站的负载性能，这个软件可以通过脚本

模拟数百个强并发用户的访问，并模拟实际用户的一些点击操作。实验使用不同的并发用户数进行了 8 次测试，每次测试设置的并发用户数分别为 100、200、300、400、500、600、650、700，测试中记录每一次的响应时间并计算平均响应时间，同时记录 CPU 和内存的占用率。

3. 系统可靠性测试

实验期间连续运行系统，记录系统稳定运行的最长时间。

4. 系统兼容性测试

在电脑上安装几类主流的浏览器，分别通过这些浏览器访问系统 Web 服务器，并在系统界面上进行操作来测试系统的兼容性。测试的浏览器包括 IE、FireFox、Google Chrome、Sogou Explore、Safari、360chrome、360SE、Baidu Explore。

5. 系统功能完整性测试

访问系统主页，对系统设计的功能逐项进行测试，记录每项功能是否达到系统设计要求。

3.5.4 结果及分析

1. 系统可部署性测试

整个系统部署的过程耗时 1 小时 54 分，结果表明系统可以在 2 小时内完成部署，具备高可部署性，测试结果详情如表 3-13 所示。

表 3-13 系统部署时间记录

序号	事件	时间点
1	开始部署系统	2015 年 4 月 30 日 18:36
2	完成系统必备软件安装	2015 年 4 月 30 日 19:09
3	完成系统安装	2015 年 4 月 30 日 19:30
4	完成题库部署	2015 年 4 月 30 日 20:03
5	完成代理服务器设置	2015 年 4 月 30 日 20:26
6	完成系统初始化	2015 年 4 月 30 日 20:30
7	完成系统部署	2015 年 4 月 30 日 20:30

2. 系统并发性测试

测试结果表明，随着并发用户数的增加，平均响应时间、CPU 占用率、内

存占用率都会上升。当并发数在 500 以内时，平均响应时间在 1s 内，CPU 占用率和内存占用率均保持在 50% 以内。并发数超过 500 后，平均响应时间、CPU 占用率和内存占用率迅速上升。并发数达到 700 时，平均响应时间超过 3s，CPU 占用率超过 75%，内存占用率超过 70%。当响应时间在 1s 以内时，用户体验最好。实验结果表明系统满足支持超过 500 个用户并行操作的性能要求，详细测试结果如表 3-14 所示。

表 3-14 并发性测试结果

序号	并发用户数	平均响应时间	CPU 占用率	内存占用率
1	100	0.26s	12%	17%
2	200	0.31s	19%	23%
3	300	0.45s	28%	29%
4	400	0.68s	39%	38%
5	500	0.96s	50%	47%
6	600	1.51s	61%	56%
7	650	2.43s	73%	63%
8	700	3.78s	82%	72%

3. 系统可靠性测试

系统持续稳定运行时间超过 30 天。

4. 系统兼容性测试

测试结果表明，系统兼容测试的 8 类浏览器，满足兼容 3 类以上浏览器的性能要求，详细测试结果如表 3-15 所示。

表 3-15 兼容性测试结果

序号	浏览器名称	浏览器内核	是否兼容
1	IE	IE	兼容
2	FireFox	Gecko	兼容
3	Google Chrome	Webkit	兼容
4	Sogou Explorer	IE	兼容
5	Safari	AppleWebKit	兼容
6	360chrome	IE	兼容
7	360SE	IE	兼容
8	Baidu Explore	IE	兼容

5. 系统功能完整性测试

实验结果表明，系统设计的功能均已实现，系统功能完整性为 100%，测试详情如表 3-16 所示。

表 3-16 系统功能测试结果

序号	功能项	测试结果	是否实现
1	系统初始化	成功设置系统管理员，完成系统初始化	是
2	用户快速注册	系统设置为快速注册方式时，不需要邮箱验证就能完成用户注册	是
3	用户邮箱注册	系统设置为邮箱注册方式时，注册时填写注册邮箱，邮箱收到一个设置账号密码的链接，点击链接完成密码设置，完成用户注册	是
4	关卡答题	进入关卡答题页面，下载附件后，提交 flag，flag 错误时，页面会有提示，flag 正确时，页面会有提示，同时对应关卡分值计入总分	是
5	选择题答题	进入选择题答题页面，系统随机生成一套试题，答题结束后页面提示选择题环节得分，并将得分计入总分	是
6	页面展示	点击各个页面，均能浏览相应的信息	是
7	竞赛时间设置	在系统全局设置中设置竞赛起止时间，在竞赛时间段意外无法使用竞赛系统相关功能	是
8	选择题分值设置	在系统全局设置中设置选择题分值后，选择题计分会发生相应变化	是
9	选择题题量设置	在系统全局设置中设置选择题题量后，选择题试题生成会发生相应变化	是
10	注册保护设置	在系统全局设置中勾选注册保护后，不能注册新用户	是
11	关卡题库浏览	在关卡管理中可以浏览所有关卡题	是
12	新建关卡题	在关卡管理中可以新建关卡题	是
13	修改关卡题	在关卡管理中可以修改关卡题	是
14	删除关卡题	在关卡管理中可以删除关卡题	是
15	设置关卡题可见性	在关卡管理中可以设置关卡题的可见性，设置关卡可见后，用户能够浏览该关卡并解答，设置关卡不可见时，用户无法浏览该关卡	是
16	选择题题库浏览	在选择题管理中可以浏览所有的选择题	是

17	新建选择题	在选择题管理中可以新建选择题	是
18	修改选择题	在选择题管理中可以修改选择题	是
19	删除选择题	在选择题管理中可以删除选择题	是
20	用户管理	管理员可以删除用户	是
21	积分管理	系统可以对所有用户积分排序,并按降序排列展示,将用户设置成积分不可见后,用户不会出现在积分榜	是
22	新建页面	在页面管理中可以新建页面	是
23	编辑页面	在页面管理中可以编辑页面内容	是
24	设置页面访问权限	在页面管理中可以设置页面访问权限	是

3.6 小结

本章详细介绍了系统的功能要求、系统的整体结构和系统数据库的设计。简单介绍了用户界面的设计。详细说明了系统的几个关键模块的设计过程。对系统进行了可部署性、并发性、稳定性、兼容性、功能完整性的实验,通过实验结果分析可以知道,系统的功能均已实现,达到预期效果。

第 4 章 结束语

4.1 全文总结

随着信息系统的高速发展与普及，信息安全已涉及到国家政治、经济、文化、社会和生态文明建设，信息安全也成为了世界各国关注的一个焦点，我国在 2014 年 2 月 27 日成立了网络安全和信息化小组，由国家主席习近平担任组长。信息安全技术的发展离不开信息安全技术人员的努力与支持，研究一套能够简易快速部署的信息安全技术竞赛平台对于信息安全技术人才的培养是一件非常有意义的事。本文对信息安全技术竞赛的概念、发展历史以及研究现状进行了总结，对动态 Web 技术、Flask 框架、数据库及透明代理的方法进行了研究，实现了一个能够快速部署的信息安全技术竞赛平台。

本文的主要工作包括：

1. 对信息安全技术竞赛的概念、发展历史及研究现状进行了总结。

介绍了信息安全技术竞赛的基本概念，信息安全技术竞赛的起源、发展壮大过程，每一个发展阶段的信息安全技术的竞赛形式及竞赛使用的竞赛平台、竞赛考察内容。对国内外现在主要的信息安全技术竞赛进行了介绍，包括竞赛的模式、使用的竞赛平台及竞赛考察的内容。

2. 提出了一种信息安全技术竞赛平台自动部署方法。

针对目前竞赛系统人工部署耗时长、人力成本高、数据分析实时性差、可视效果不强等问题，提出了一种信息安全技术竞赛平台自动部署方法。该方法采用 B/S 架构和动态 WEB 技术便于用户应用，使用 MVC 框架提升系统的稳定性和安全性，通过透明代理实现用户的安全接入，竞赛题目子系统和管子系统相隔离支持系统的安全性和稳定性，利用软件工程思想保证系统的可扩展性，基于选择题库为选手随机生成试题、选手积分自动积分和排序、日志数据自动统计分析等方案实现平台的自动化管理。

3. 设计实现了一个信息安全与对抗技术竞赛平台。

平台基于 linux 操作系统，采用 C++ 和 Python 语言开发，数据库采用 MySQL。系统 web 服务使用 HTML 语言、CSS 层叠样式表及 JavaScript 脚本语言实现。

平台主要功能包括：用户注册与登录认证、用户管理、题库管理、在线答题及积分管理等。基于软件工程思想对平台进行了测试，结果表明，能够在 2 小时内完成竞赛平台的部署，再经过第 12 届信息安全与对抗技术竞赛的试用，平台已不间断稳定运行时间超过 30 天，支持超过 500 个用户并发操作，兼容 IE、FireFox、Google Chrome、Sogou、Safari 等 8 类浏览器。

4.2 工作展望

本文设计并实现的信息安全技术竞赛平台已经能够完成必要的基本功能，但仍然有一些不足的地方需要改进。在本文的基础之上，在后续工作中还将对以下方面进一步研究与改进。

1. 公告信息实时提醒。

目前系统的公告页面不具备实时提醒功能，当新公告发布时用户需要手动去查询，所以需要采用其他技术来实现公告信息实时提醒功能，当新公告发布时，系统会提醒用户查看公告。

2. 增加竞赛文化展示页面。

信息安全技术竞赛平台不仅仅是信息安全技术人员的竞技场，同时也是信息安全竞赛文化传播的平台，在下一阶段的研究中将增加一个竞赛文化展示界面，用以展示竞赛文化、宗旨以及历年竞赛海报。

3. 页面展示统计信息。

在竞赛过程中，竞赛的一些统计数据对管理员及参赛者有很大的参考价值，在下一阶段的研究中在竞赛系统页面上展示注册人数及当前在线用户数、各个关卡的通过率、各种类型关卡的通过比例等供管理员及参赛者参考。

致 谢

时光荏苒、岁月如梭，转眼间本科毕业设计即将结束，四年的大学生活也即将画上句号。在毕业设计的这几个月的时间里，有汗水、有苦恼、有伤感，也有进步、有发现、有自豪。由于缺乏系统性的思维，再加上没有项目经验，在毕业设计过程中难免会有一些考虑不完善的地方，如果没有导师的督促与指导以及实验室师兄师姐的帮助，凭我一个人的能力想要完成毕业设计是难以想象的。

在这里首先要感谢我的毕业设计导师罗森林教授，罗老师教会了我要用系统的角度观察审视问题，从顶层开始思考分析问题，做研究要讲求精益求精，这对我在做每一项学术研究时都产生了潜移默化的影响。

在毕业设计期间贾丛飞师兄给予了我极大的指导和帮助。在此向贾丛飞师兄致以最诚挚的感谢。在毕业设计期间，贾丛飞师兄给我提供了最大的技术支持，在毕业设计中遇到技术难题无法解决时，经过贾丛飞师兄的指导后问题都迎刃而解。

除此之外，实验室其他的师兄师姐对我的帮助也很大，每周的组会使我在进行毕业设计的过程中还汲取了其他方面的知识，扩展了思路，为以后的学习、研究工作开了一个好头。

参考文献

- [1] 潘丽敏,罗森林,张笈等. 信息安全与对抗竞赛中个人挑战赛的设计方法[J]. 实验技术与管理,2012,29(12):203-206.
- [2] 柳杨,范冰冰. 一种通用的动态 Web 应用加速方案[J]. 计算机工程,2012,38(11):247-250.
- [3] 田红玉. 静态网页制作技术与动态网页制作技术的比较[J]. 黑龙江科学,2013,4(9):116.
- [4] 刘祺彬. 基于开源技术的 MVC 框架的构建[J]. 信息技术,2013,(8):125-129.
- [5] 张宇,王映辉,张翔南等. 基于 Spring 的 MVC 框架设计与实现[J]. 计算机工程,2010,36(4):59-62.
- [6] 赵玉伟,肖敏,熊前兴等. 一种简单 MVC 框架的实现方案[J]. 交通与计算机,2006,24(4):136-138.
- [7] 刘亚鹏. 基于 MVC 架构的电子商务平台的设计与实现[D]. 华中科技大学,2011.
- [8] 于海雯,娄芳. 用 MVC 处理 Web 系统的逻辑编程和页面设计的分离问题[J]. 电脑知识与技术（认证考试）,2004,(8):69-71.
- [9] 李绘英. 基于 WSGI 规范的 Web 数据库应用系统研发[D]. 武汉理工大学,2012.
- [10] 王贵生. MySQL 数据库索引的研究[J]. 电脑知识与技术,2014,(34):8079-8080.
- [11] 宋珍玉,王华军. Sql Server、Mysql 数据库统一管理解决方案[J]. 电脑知识与技术,2013,(23):5203-5205,5210.
- [12] 吴垠,余瑾,郭磊等. MySQL 数据库在嵌入式 Linux 平台上的移植和应用[J]. 电子技术,2011,38(7):12-13,3.
- [13] 胡安磊,周大水,李大兴等. Linux 中 Netfilter/IPtables 的应用研究[J]. 计算机应用与软件,2004,21(10):56-57,66.
- [14] 李玉峰,杨婷,卜永波等. Linux 下基于 Netfilter/Iptables 防火墙的研究与应用[J]. 内蒙古农业大学学报（自然科学版）,2012,33(1):198-200.

- [15] 季刚,姚艳,唐怀瓿等. Linux 下基于 Netfilter/iptables 防火墙的构建[J]. 电脑知识与技术,2011,07(19):4550-4552.
- [16] 赵亚楠,马兆丰. Linux 中 Netfilter / iptables 的研究与应用[J]. 中国科技论文,2014,(10):1174-1177,1187.
- [17] 陈国伟,伍小明. 透明网关与透明代理结合的防火墙的设计与实现[J]. 计算机应用研究,2003,20(5):138-140.
- [18] 朱萍. 基于透明代理的 Linux 防火墙的设计与实现[J]. 合肥工业大学学报(自然科学版),2007,30(5):576-578,586.
- [19] 陶丽娟. 基于 Linux 网络系统的 Squid 代理服务器的构建[J]. 消费电子,2014,(4):163-163.
- [20] 姜明铭. squid 透明代理服务器的设计[J]. 中国科技纵横,2012,(19):56.
- [21] 朱媛媛. 基于 Web 技术的在线考试系统的设计与实现[D]. 天津师范大学,2014.
- [22] 汪永明. 基于 B/S 模式的在线考试系统[J]. 微机发展,2004,14(1):59-60,85.
- [23] 郭文荣,高宗余,于丽杰等. 大学网上考试系统的设计与实现[J]. 计算机与网络,2014,(18):59-61.
- [24] 周奇 李震阳. LiNuX 系统网络服务器组建、配置和管理实训教程.[M].