吴伟杰_201525050420_论文定稿

【原文对照报告-大学生版】

报告编号：57c6e7a4bc84df1d　　检测时间：2019-04-28 13:40:26　　检测字数：21,197字

作者名称：吴伟杰　　所属单位：华南农业大学

检测范围：

◎ 中文科技期刊论文全文数据库
◎ 中文主要报纸全文数据库
◎ 中国专利特色数据库
◎ 博士/硕士学位论文全文数据库
◎ 中国主要会议论文特色数据库
◎ 港澳台文献资源
◎ 外文特色文献数据全库
◎ 维普优先出版论文全文数据库
◎ 互联网数据资源/互联网文档资源
◎ 高校自建资源库
◎ 图书资源
◎ 古籍文献资源
◎ 个人自建资源库
◎ 年鉴资源
◎ IPUB原创作品

时间范围：1989-01-01至2019-04-28

检测结论：

全文总相似比	=	复写率	+	他引率	+	自引率	+	专业术语
9.35%		9.35%		0.00%		0.00%		0.00%

其他指标：

自写率：90.65%

专业用语：0.00%

高频词：系统,代码,用户,评判,资源

典型相似性：　无

指标说明：

复写率：相似或疑似重复内容占全文的比重

他引率：引用他人的部分占全文的比重，请正确标注引用

自引率：引用自己已发表部分占全文的比重，请正确标注引用

自写率：原创内容占全文的比重

专业用语：公式定理、法律条文、行业用语等占全文的比重

典型相似性：相似或疑似重复内容占互联网资源库的比重，超过60%可以访问

总相似片段：106

期刊：2 博硕：36 外文：1 综合：1 自建库：0 互联网：66

摘 要

本文主要介绍基于Scrapy框架的计算机专业学习资源网站的开发目的、设计思想和具体实现方法等内容。本系统是一个提供计算机专业理论知识学习与计算机语言编程实践的平台，目标用户是有学习计算机专业知识的需求的人群。面向用户的功能有：计算机专业知识学习资料的检索、编程题目在线评判。计算机相关学习资料来源主要有：网络爬虫索引各大博客、网站内用户整理并发布学习资料。编程题目由网站内具有题目管理权限的用户编写题目描述、测试用例，解题用户编写解题源码并上传到后端，代码评判器评判后将运行结果呈现给用户。本系统的学习资料与编程题目拥有分类标签，两者能够通过标签直接关联。与常见的通用搜索引擎、在线评判网站相比，本系统对计算机专业更具有针对性，理论与实践相结合，对用户学习效率的提高有所帮助。

本系统采用MVVM设计模式，前端是以Vue.js为主的一系列框架与插件开发的单页面应用程序。后端采用微服务架构，划分为4个模块：服务注册中心、配置服务器、资源服务器、代码评判器，支持分布式部署、集群。系统主要基于Spring体系的Spring Cloud、Spring Data等系列的框架开发，学习资料等资源存储、检索功能基于Elasticsearch，系统用户信息、编程题目相关数据、运行时产生的数据存储在关系型数据库MySQL中，并在应用层引入Redis实现缓存等功能，用户提供的图片、文档资料等文件采用了第三方服务阿里云对象存储OSS。代码评判器通过消息中间件RabbitMQ与资源服务器通讯。网络爬虫是本系统外部资源的重要来源，能够提供大量站外资源供本系统用户检索，基于Scrapy框架开发，引入Redis实现任务队列与URL去重。

关键词：网络爬虫 Scrapy Spring框架 在线评判

Computer Professional Learning Resource Website Based on Scrapy Framework

Wu Weijie

(College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China)

Abstract: This article mainly introduces the development purpose, design ideas and specific implementation methods of the "Scrapy framework-based computer professional learning resources website". This system is a platform to provide computer professional theoretical knowledge learning and computer language programming practice. The target users are people who have the need to learn computer professional knowledge. The user-oriented functions include: retrieval of computer professional knowledge learning materials, online evaluation of programming topics. The main sources of computer-related learning materials are: web crawler index major blogs, users within the website to organize and publish learning materials. The programming topic is written by the user with the topic management authority in the website, and the problem description source code is uploaded to the back end. After the code judger judges, the running result is presented to the user. The learning materials and programming topics of this system have classification labels, and the two can be directly related through labels. Compared with the common universal search engine and online evaluation website, the system is more targeted to the computer major, and the combination of theory and practice helps the user's learning efficiency.

The system adopts the MVVM design mode, and the front-end is a single-page application developed by a series of

frameworks and plug-ins based on Vue.js. The back-end uses a micro-service architecture and is divided into four modules: a service registry, a configuration server, a resource server, and a code evaluator to support distributed deployment and clustering. The system is mainly based on the Spring system, Spring Data and other series of framework development of the Spring system. The resource storage and retrieval functions such as learning materials are based on Elasticsearch, system user information, programming topic related data, and runtime generated data are stored in the relational database MySQL. And in the application layer to introduce Redis to achieve caching and other functions, the user-provided pictures, documents and other documents using a third-party service - Alibaba Cloud OSS. The code evaluator communicates with the resource server via the message middleware RabbitMQ. Web crawler is an important source of external resources of the system. It can provide a large amount of off-site resources for users of this system to search. Based on Scrapy framework development, Redis is introduced to implement task queue and URL deduplication.

Key words: Web Crawler Scrapy Spring Framework Online Judgement

目 录

1 前言

1.%2 研究背景

2017年，我国软件和信息技术服务业从业人员平均人数同比增速3.4%(中华人民共和国工业和信息化部，2018)；到了2018年，我国软件和信息技术服务业从业人员平均人数同比增速达到4.2%(中华人民共和国工业和信息化部，2019)。我国的软件业一直存在较大人才缺口，吸引了部分高校学生选择或转入软件、计算机相关专业，以及部分社会人士转行从事计算机软件相关行业。

计算机行业的专业性很强，尤其是开发类的工作，涉及大量专业知识，对从业者的基础如数据结构、常见算法、编程语言基础等有一定要求。基础知识的学习方式较多样化，校内有教师课堂教学，校外有培训机构开设课程，互联网上也有大量资料。由于课堂开班教学局限性较大，教师难以在有限时间内传授大量专业知识，通过互联网自学是计算机专业最重要的学习手段(黄洪波，2016)。

目前互联网上有较多计算机理论知识学习网站、数据结构与算法的可视化学习网站、计算机相关专业人员的博客，也有例如UVa等用于编程练习、比赛的网站，这些网站目前能够满足大部分用户的需求，但仍存在一些问题(胡二彪，2012)。

从使用者的角度，这些计算机专业学习网站的功能相对单一，理论知识学习网站只能基本满足理论学习的需求，实践还需要用户在自己的电脑或者其他在线评判系统进行，各个不同网站之间联系不大，导致所学的理论知识难以及时关联实践所需知识。

从系统架构的角度，目前大多数网站为单体架构，即所有应用模块集中在同一个项目中，开发流程较简单，但不易修改、扩展，系统复杂难以维护。以Java为例，在早期开发项目主要基于MVC分层模型开发Web应用，系统中不同模块都集中在同一项目的Model和Controller层，View层后端渲染Web页面模板文件。在扩展方面，单体系统只能水平扩展，无法只对系统中某一模块进行扩展，缺

乏灵活性。

随着互联网的发展以及用户的需求不断增加，各个不同的功能将会集成到一个大的平台中，例如数据结构、算法理论知识与编程实践题目对接，紧密关联，节省用户用于检索资料、寻找实践方法的时间。除了功能方面，由于互联网开放性，系统使用人数不确定，系统在可用性和并发量方面都有一定的要求。因此，模块高度耦合、难以扩展的传统MVC单体架构已经难以满足用户需求，越来越多的新项目基于微服务架构开发，也有少数较早的系统被重构为微服务架构(郑彬彬，2017)。微服务架构解决了复杂性问题，它将单体架构分解为一组服务，把系统各个不同的模块解耦。服务之间定义了一系列明确的接口，只要遵循接口等规则，每个服务的具体实现对于其他服务完全透明。对于压力较大的服务，可以有针对性地进行水平扩展，而无需像单体架构只能对整个系统水平扩展。微服务架构的优势非常明显，但也增加了开发过程的复杂度，开发基于微服务架构的系统需要解决服务发现、服务间调用、服务容错、服务部署等问题。

2.%2 研究意义

本系统面向的用户群体是需要学习计算机专业知识的人群，能够提供一个计算机专业理论学习与编程实践平台给计算机专业的学生用于课后学习，或需要自学计算机专业知识的用户用于自习。本系统结合理论与实践，能够提供一种更便利地学习计算机专业知识的方式，提高使用者的学习效率。

其次，互联网上现有的开源在线评判系统数量较少，国内常见的有基于Python的青岛大学qduoj、基于PHP的hustoj与uoj，基于Java以Spring体系框架开发的OJ数量更为稀少。开发基于Java的计算机专业学习资源网站，能够为目前为数不多的开源的在线评判系统做出一些贡献。

3.%2 系统设计思路

本系统是一个在互联网上开放的系统，面向的用户群体是计算机专业相关人员。与一些组织或个人内部使用的系统相比，开放网络的用户人数难以预计，因此系统的可用性与并发量是系统架构设计所需考虑的因素。

互联网上存在海量信息，可以作为本系统资料的主要来源，结合系统内用户人工整理、发布的资料，能够实现本站的资源检索功能。通用搜索引擎仅仅具备检索功能，且搜索结果不具有针对性。本系统开发的检索功能必须与通用搜索引擎相比更具有针对性，否则将失去使用的意义。

在编程实践方面，常见的在线评判系统的评判器在评判代码时能够精确统计时间、内存等信息，评判器也具有一定的健壮性，不会轻易被用户提交的恶意代码攻击系统。但这些评判器对操作系统的依赖性较强、评判代码时涉及进程创建、切换等操作，运行时开销较大。本系统评判器作为架构中的一个服务，如果能在保证健壮性与运行性能的前提下减少评判器对操作系统的依赖，在扩展、集群部署方面都会有一定的优势。

综上，本系统设计思路可以概括为：

（1） 采用微服务架构，支持分布式以保证可用性和并发性能；

（2） 学习资源来源以互联网各大计算机相关博客为主，站内用户整理发布为辅；

（3） 对文章资源等数据进行索引时，需要索引更多详细的信息；

（4） 理论学习资源与编程题目需要通过一定的关系进行关联；

（5） 系统架构的各个服务实现时需要降低对运行环境的依赖性，降低部署难度，保证系统运行与维护具有一定的灵活性。

4.%2 技术选型思路

爬虫模块：网络爬虫在爬取大型网站时需要部署较多实例，因此要求每个实例拥有较低的部署难度。基于Python语言的Scrapy是一款易用、成熟的爬虫框架，由于框架自身封装了大量与业务逻辑无关的代码，上手简单，开箱即用，只需编写少量逻辑即可启动。Scrapy的轻量级、开箱即用的特点恰好能够满足低部署难度等要求(刘思林，2018)。

系统前端：作为与用户直接交互的部分，前端的体验非常重要。近年来，前端出现了一种使用MVVM设计模式的应用类型SPA（Single Page Web Application，单页面应用程序）。与传统的多页应用程序相比，由于SPA首次启动便完成了应用所需的静态资源的加载，切换页面没有了请求应用自身静态资源的开销，具有非常快的响应速度，流畅度也高于传统的多页应用程序(李嘉,赵凯强,李长云

，2018)。本系统开发SPA应用使用JavaScript框架Vue.js及其生态圈下的Vue Router、Vuex插件，该框架上手容易，性能也相对良好。

系统后端：系统后端采用微服务架构，需要解决的问题包括服务发现、应用配置等。结合各方面考虑，系统后端开发选用Java作为主要编程语言。Java拥有庞大的生态圈，是很多企业级应用开发的合适选择。Spring Framework是以Java语言为主的一个庞大的框架体系，所提供的工具几乎涉及了应用开发的所有层次，包括了本系统所需的微服务架构、数据持久层、消息中间件、RESTful API开发，因此本系统开发技术首选以Spring Boot、Spring Cloud、Spring Data的一系列框架。

数据存储：本系统的资源服务器需要实现包括全文搜索、数据聚合等功能，且由于数据量会随爬虫的运行而增加，数据库性能、存储空间都具有一定的压力，数据库需要具有分布式部署能力。Elasticsearch是个性能强大、技术成熟的搜索引擎，天生支持分布式部署，使系统的灵活性和可用性都有一定的保证，故选用Elasticsearch存储、索引数据(王伟,魏乐,刘文清,舒红平，2018)。

1 关键技术介绍

5.%2 Scrapy

Scrapy是基于Python的开源爬虫框架，能够爬取网页并解析结构化的数据。Scrapy框架封装了较多繁琐的底层操作，使用简单，对于比较简单的网站，只需编写少量逻辑与修改一些配置项即可开始抓取数据。除了封装基本的流程，Scrapy框架可扩展性非常高，拥有大量强大的插件，根据各种应用场景使用对应的插件，能够极大提高使用者的开发效率(Scrapy，2018)。

6.%2 Redis

Redis是一个基于内存的NoSQL数据库，支持5种常用数据结构：String、List、Hash、Set、Sorted Set。由于数据存储在内存中，Redis拥有极高的吞吐量，应用场景非常广泛，例如缓存、计数器、分布式锁、队列、适用于大数据的位操作、排行榜等。为了应对内存的易失性，Redis提供了两种持久化的方式：一是保存当前内存快照，这种方式占用磁盘空间较小，但存在丢失少量数据的风险；二是通过追加命令的方式记录对Redis的所有操作，这种方式不易丢失数据，但日志文件会随着时间的推移一直扩大，对磁盘空间占用较大。Redis支持主从模式、集群，在可用性和并发性能方面有所保障(Redis，2018)。

7.%2 Vue.js

Vue.js是一款目前比较流行的JavaScript框架，结合ECMAScript6模块化后，能够以编写组件的方式前端页面。与直接使用HTML、CSS、JavaScript编写HTML等文件的传统方式相比，用组件构建前端页面的方式能够在一定程度减少代码冗余，有效提高代码复用率。Vue.js框架还有许多插件，扩展了框架能够实现的功能，能够满足绝大多数项目的需求。目前还有许多基于Vue.js框架开发的UI库，例如Element，提供了很多常用UI，能够快速搭建逻辑清晰、结构合理且高效易用的前端界面(Evan You，2018)。

2.%2.%3 Vue Router

Vue Router是Vue.js官方提供的路由管理器，该插件能够让构建SPA易如反掌。该插件包含的功能较多，包括常用的嵌套路由或视图、基于组件的模块化的路由配置、支持通配符的路由参数查询、视图过渡动画效果等。

3.%2.%3 Vuex

Vuex是专门为Vue.js应用开发的状态管理模式，可以简单地解为全局变量，但Vuex的功能远远强大于一个简单的全局变量。Vuex采用集中式存储管理应用的所有组件的状态，并以相应的规则保证状态以一种可预测的方式发生变化。可以用Vuex保存的信息有很多，例如用户登录后，服务端返回基于JWT的令牌可以存储在Vuex的数据部分，以便于后续操作调用服务端API时携带令牌。

8.%2 Spring Framework

Spring Framework是一个很庞大的体系，它实现了很多经典设计模式，封装大量繁琐与业务无关的底层代码，提供很多高级API，极大提高项目开发效率。Spring系列框架覆盖了绝大多数项目的各个层级，例如Web、数据库、中间件、批处理等，也提供了搭建分布式、微服务架构所需的工具。而且，Spring能够做到开箱即用，引入所需依赖，只需少量的配置，即可开始专注于实现业务逻辑代码。

4.%2.%3 Spring Boot

Spring Boot能够实现Spring Framework开箱即用。在Spring Boot出现之前，引入Spring框架的项目搭建与配置是一项非常繁琐的事

情。举个例子，搭建一个简单的基于Spring MVC的Web项目时，需要先手动或用Maven之类的项目依赖管理工具引入Spring框架的依赖，然后需要配置XML文件或者使用注解配置引入Spring MVC的DispacherServlet或其他相关Servlet与Filter，最后发布到Tomcat之类的Servlet容器才能使基本项目运行。如果要连接数据库，还需要配置数据源、事务管理器等，过程非常繁琐。但Spring Boot的出现使得这一切发生变化。

Spring Boot的理念是约定优于配置，底层隐藏了大量的自动配置逻辑，最后只需少量配置甚至实现零配置。除此，Spring可以内置Tomcat之类的Servlet容器，使得运行项目时无需再单独配置Servlet容器，实现开箱即用(Phillip Webb, et al，2018)。

5.%2.%3 Spring Cloud

分布式系统由于自身的特点，增加了开发过程的复杂性。Spring Cloud为开发人员提供了快速构建分布式系统中常见模式的实现，例如服务发现、配置管理、断路器、路由、分布式会话、集群监控等，是快速搭建微服务架构系统的利器。

6.%2.%3 Spring Data

Spring Data封装了操作各类数据源的底层代码，例如Spring Data JPA能够实现声明式编程，只需定义接口，无须实现代码甚至无须SQL语句即可实现数据库的增删查改。需要实现复杂查询的时候，也只需少量的代码，能够保持低耦合。

9.%2 RabbitMQ

RabbitMQ是基于Erlang语言编写的消息中间件。得益于Erlang语言本身并发性能较高的特性，RabbitMQ的性能较好；支持AMQP等多种协议，且大多数编程语言都提供了AMQP协议客户端，语言无关使其兼容性良好。RabbitMQ支持消息持久化以应对各种紧急情况，保证消息不会丢失，且提供了控制台，能够实时监控消息队列的各种状态或对队列进行管理配置。

10.%2 Elasticsearch

Elasticsearch是一个基于Lucene的分布式、RESTful风格的搜索和数据分析引擎。Elasticsearch具有很高的灵活性，提供了非常多的数据类型，能够对数字、文本、地理位置、结构化数据、非结构化数据进行存储、分析。(Elasticsearch B.V，2018)

在性能方面，Elasticsearch通过有限状态转换器实现了用于全文检索的倒排索引，实现了用于存储数值数据和地理位置数据的BKD树，以及用于分析的列存储。而且由于每个数据都被编入了索引，不会因为某些数据没有被索引而遗漏。

Elasticsearch的可扩展性也非常强，既能在性能不高的机器运行单个实例，也可以在数百个节点的集群运行，能够自动管理索引和查询在集群中的分布方式，使得在各种不同环境下能够以相同的操作方式实现查询与分析，实现开发环境和生产环境无缝切换。

11.%2 MySQL

MySQL是一个流行、开源的关系型数据库管理系统，具有性能优秀、部署难度低、使用成本低的特点。与常见的大型数据库Oracle、SQL Server相比，MySQL体积小，占用资源较低，具有灵活性。而且MySQL为多种语言提供了API，覆盖了常用的编程语言，在编程语言上不存在使用障碍。

12.%2 Nginx

Nginx是一个免费、高性能、稳定性高、开源的HTTP服务器，具有反向代理的功能，也可以作为IMAP/POP3代理服务器。与传统的HTTP服务器相比，Nginx不依赖于线程，而是采用了事件驱动的异步架构，能够承载上千甚至上万的并发连接。由于Nginx具有占用资源少的特点，即使应用场景不涉及高并发、不需要处理大量的请求，Nginx也是一个非常合适的选择。

7 系统架构与关键部分实现细节

13.%2 系统总体架构

系统架构见图1。

图1 系统架构

本系统基于微服务架构，支持集群、分布式部署。

前端使用Nginx作为HTTP服务器与反向代理服务器，负责用户获取静态页面以及AJAX请求的反向代理。

后端微服务主要分为两个部分：用于维护微服务架构的服务注册中心与配置服务、用于面向用户提供功能的资源服务器与评判器集群。其中，资源服务器与评判服务器依赖消息中间件RabbitMQ，资源服务器需要连接MySQL、Redis、Elasticsearch。

用于采集数据的爬虫为独立模块，可以任意部署在可以连接Elasticsearch服务器的机器，分布部署爬虫需要依赖同一个Redis实现任务队列与去重。

14.%2 数据库结构设计

8.%2.%3 Elasticsearch索引设计

本系统学习资源存储与索引基于Elasticsearch实现。本索引用于实现资源检索功能，详细功能包括：精确搜索、模糊搜索、基于前缀的搜索建议、拼音搜索、搜索关键词纠错等。学习资源索引结构见图2。

图2 学习资源索引结构

在Elasticsearch中，字符串对应的数据类型拥有两种：keyword、text。在Elasticsearch对文档进行索引时，使用keyword数据类型存储的字符串不会被分词器分析，整个字符串直接存储到Elasticsearch中；而使用text数据类型的字符串，在索引时会被默认或指定的分词器分析，并形成倒排索引。因此，使用text类型存储的字符串可以被用于搜索功能，但不可被作为term进行数据聚合；使用keyword类型存储的字符串查找时只能够使用term或者prefix，但在进行聚合时可以作为被聚合的字段。

9.%2.%3 MySQL表设计

用户表主要存储了用户名和加密后的用户密码。每个用户拥有唯一的用户名，登录时使用用户名、密码登录。为了提高系统的安全性，登录密码存储经过不可逆加密后的值。用户表结构如表1。

表1 用户表

列名	列含义	数据类型	长度	备注/默认值
id	用户编号	bigint	20	主键
username	用户名	varchar	255	值唯一
password	登录密码	varchar	255	BCrypt算法加密
sign_up_date	注册时间	timestamp		CURRENT_TIMESTAMP

由于本系统使用RBAC（基于角色的权限访问控制），每个用户拥有一定数量的角色，不同的角色能够使用不同的功能，例如发布新的编程题目需要具有题目管理员权限的用户执行，删除其他用户上传的资源需要用户具有资源管理员权限。用户角色表见表2。

表2 用户角色表

列名	列含义	数据类型	长度	备注/默认值
id	编号	bigint	20	主键
user_id	用户编号	bigint	20	外键
role_name	角色名称	varchar	255	
grant_time	角色授予时间	timestamp		CURRENT_TIMESTAMP

本系统会在为每位用户记录历史浏览记录，以便于用户回顾过去浏览的文章。由于本站资源来源分为站外和站内，站外资源可以直接通过url访问，站内资源需要经过资源服务器获取资源，于是使用external字区分站内与站外资源，以便于在前端采用不同的处理方式。用户浏览历史表结构如表3所示。

表3 用户浏览历史

列名	列含义	数据类型	长度	备注/默认值
id	历史记录编号	bigint	20	主键
user_id	用户编号	bigint	20	外键
title	资源标题	varchar	255	
url	资源链接	text		

external	是否外部资源	tinyint	1	
view_time	浏览时间	timestamp		CURRENT_TIMESTAMP

编程题目表用于存储本系统中拥有的编程题目以及该题目的发布者。编程题目表结构如表4。

表4 编程题目表

列名	列含义	数据类型	长度	备注
id	题目编号	bigint	20	主键
title	题目标题	varchar	255	
description	题目描述	text		
contributor_id	贡献者编号	bigint	20	外键

由于不同编程语言的语法不同，每个编程题目需要存储对应各编程语言的代码模板，如果代码中包含多个方法/函数，则需要显式指定入口方法/函数。代码模板表结构见表5。

表5 代码模板表

列名	列含义	数据类型	长度	备注
id	代码模板编号	bigint	20	主键
problem_id	题目编号	bigint	20	外键
template	代码模板	text		
entry_method	默认入口方法	varchar	255	
language	编程语言	varchar	255	

测试用例表用来存储验证提交代码所用的测试用例，每一行记录都对应一个问题的一个测试用例。输入值与期望值以JSON格式的文本存储，评判器在获取到JSON格式的字符串后，会把字符串解析成输入值、输出值对应类型的实例，以便于输入被测函数/方法。测试用例表结构见表6。

表6 测试用例表

列名	列含义	数据类型	长度	备注
id		bigint	20	主键
problem_id	题目编号	bigint	20	外键
input	输入值	text		
expect	期望值	text		
timeout	时间限制	bigint	20	

标签表用于存储编程题目相应的标签，通过该标签，编程题目可以和拥有相同标签的学习资源关联。标签表需要通过关联表与题目关联。标签表结构见表7。

表7 标签表

列名	列含义	数据类型	长度	备注
id		bigint	20	主键
tag_name	标签名	varchar	255	

题目与标签的关联，每一行记录包含了题目编号与标签编号，题目标签关联表结构如表8所示。

表8 题目标签关联表

列名	列含义	数据类型	长度	备注
id		bigint	20	主键
problem_id	题目编号	bigint	20	外键
tag_id	标签编号	bigint	20	外键

用户资源关联表用于记录系统内用户与其发布的资源的关联关系，其中，resource_id字段与Elasticsearch中每个文档自动生成的_id对应。用户资源关联表见表9。

表9 用户资源关联表

列名	列含义	数据类型	长度	备注
id		bigint	20	主键
user_id	用户编号	bigint	20	外键
resource_id	资源编号	varchar	32	与Elasticsearch资源索引对应

编程提交表用于记录用户的编程题解提交记录，包括了提交的题目与源码等数据。diagnosis字段用于记录编译出错时的提示消息或者运行抛出异常时的错误堆栈，以便于用户定位出错原因与相关代码行数。本表一般在用户提交代码时创建一条记录，等评判器评判结果返回时通过id对相应的记录进行更新。编程提交表结构如表10所示。

表10 编程提交表

列名	列含义	数据类型	长度	备注/默认值
id	提交编号	bigint	20	主键
problem_id	题目编号	bigint	20	外键
code	提交代码	text		
entry_method	入口方法	varchar	255	
language	编程语言	varchar	255	
compile_success	是否编译成功	tinyint	1	
message	错误消息	text		
diagnosis	错误诊断	text		
time	提交时间	timestamp		CURRENT_TIMESTAMP
judged	已评判	tinyint	1	
judge_time	评判时间	timestamp		
user_id	提交用户编号	bigint	20	外键
accepted	是否正确	tinyint	1	
used_testcase_count	运行次数	int	11	
accepted_testcase_count	正确次数	int	11	
error_case	错误输出	text		

15.%2 服务启动与注册流程

服务注册中心启动后，服务的注册与启动流程如图3所示。

图3 服务启动与注册流程

一般传统单体架构系统中各配置文件与代码集成，采用这种方式的系统在启动时流程比较简单，直接运行就可以连接数据库等其他服

务器。由于微服务架构的系统中有较多的服务节点需要部署，如果与数据库等相关的配置文件与代码集成，一旦需要更新配置内容，节点越多需要做的操作就越多，需要耗费较多时间、资源。因此，单体架构常用的配置方式不太适用于微服务架构系统。

本系统的配置方式是基于配置服务器。当服务注册中心启动完成后，接着启动配置服务，配置服务向服务注册中心注册并使用Git或SVN等工具或从本地拉取所有配置文件后，此时就可以启动系统中的其他服务了。

以资源服务器为例，资源服务启动时，会先向服务注册中心注册，并且从服务注册中心获取到配置服务的相关信息，例如Socket等信息。得到配置服务的地址与端口后，资源服务就向配置服务拉取相关配置。配置文件按照环境可以分为Development（开发环境）、Test（测试环境）、Production（生产环境）等，以便于在部署服务时无缝切换到不同环境。获取到配置文件后，资源服务开始与数据库等服务建立连接，完成服务启动流程。

16.%2 Scrapy爬虫实现细节

10.%2.%3 任务队列与去重

本系统采用Scrapy框架，在默认情况下，任务队列与去重都是在爬虫进程内存中实现的，支持持久化到本地。如果只有一个爬虫实例，默认配置能够满足需求。但在需要多个爬虫实例的情况下，默认配置的爬虫之间无法协作，因此需要将任务队列与去重数据转移到能够共用的区域。

图4 URL去重流程

URL去重流程如图4所示。

URL参数顺序是无关的。例如a=1&b=2等价于b=2&a=1，因此，去重逻辑第一步是对URL参数按照参数的键重新排序，避免因URL参数顺序问题被识别为不同URL。

根据文档RFC3986，在一个合法的URL中，只许包含英文字母、数字、特殊字符-_.~以及保留字符。因此，非ASCII编码的字符需要被编码成URL允许的字符。

URL中百分号编码大小写不敏感。例如百分号编码%2f等价于%2F，因此，需要对URL中的百分号编码进行大小写规整化，确保能够正确去重。

URL长度不固定，如果直接在去重集存储URL及相关的HTTP方法、请求体将会占用较大空间，故先拼接URL相关信息并做SHA1，将每个URL转换为占用20字节空间的哈希值，能够在有效去重的前提下节省大量空间。

Redis的SADD命令是将值放入指定的集合中，并返回成功放入的元素个数。如果被放入的值在集合中已存在，元素放入失败，返回成功放入的元素个数为0。

最后，根据成功放入去重集的元素个数，决定是否将目标URL放入爬虫任务队列中。

11.%2.%3 网站URL跟踪方式

网络爬虫跟踪URL的方式有许多种，常见的有如下：

（1） 分析URL规则并循环，适用于URL中有连续编号的ID或其他数据；

（2） 下载当前页面后解析页面中包含的所有URL，将符合规则的链接放入任务队列中，这种方式比较通用与常见；

（3） 解析网站sitemap，适用于提供了sitemap的网站。

本系统的爬虫模块采用了以上2、3两种方式。

12.%2.%3 配置正则表达式规则跟踪URL

爬取当前网页并解析当前页面中所有URL进行跟踪是一种比较通用的URL跟踪方式，这种方式需要选择合适的爬虫起点，例如首页或者一些专题页面。

本系统的爬虫，主要采用正则表达式配置URL跟踪规则，根据不同的匹配结果，决定该URL的是否需要被继续跟踪，是否需要解析页面内容。

13.%2.%3 读取sitemap跟踪URL

通过网站提供的sitemap爬取网站是一种比较高效且较容易覆盖全站的方式。sitemap是一种SEO（Search Engine Optimization，搜

索引擎优化）手段，sitemap文件中记录了网站大量URL，包括主页或其他重要页面，其目的是让搜索引擎能够尽量索引更多的网站内容(陈荣华,杨蓓，2018)。本系统的检索功能就是一个搜索引擎，爬虫通过sitemap爬取目标网站，能够使本系统的搜索功能覆盖更多学习资源。

以CSDN为例。CSDN博客在其robots.txt文件中（https://blog.csdn.net/robots.txt，该文件存储了对爬虫的规定，例如允许/禁止爬取的URL、允许/禁止的User-Agent等）提供了sitemap的URL（https://blog.csdn.net/s/sitemap/pcsitemapindex.xml）。

以下代码片段为sitemap文件节选。sitemap文件包含了URL、最后更新时间、修改频率、优先级等信息，爬虫根据sitemap爬取网站时可以更加高效、全面。

```
1. <urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
xmlns:mobile="http://www.google.com/schemas/sitemap-mobile/1.0">
2. <url>
3. <loc>https://blog.csdn.net/amone/article/details/148</loc>
4. <mobile:mobile type="pc,mobile"/>
5. <lastmod>2004-03-29</lastmod>
6. <changefreq>weekly</changefreq>
7. <priority>0.8</priority>
8. </url>
9. <url>
10. <loc>https://blog.csdn.net/amone/article/details/150</loc>
11. <mobile:mobile type="pc,mobile"/>
12. <lastmod>2004-03-29</lastmod>
13. <changefreq>weekly</changefreq>
14. <priority>0.8</priority>
15. </url>
16. <url>
17. <loc>https://blog.csdn.net/superyan/article/details/152</loc>
18. <mobile:mobile type="pc,mobile"/>
19. <lastmod>2004-03-29</lastmod>
20. <changefreq>weekly</changefreq>
21. <priority>0.8</priority>
22. </url>
23. </urlset>
```

14.%2.%3 页面内容的解析

爬虫发送HTTP请求并获取响应HTML后，需要从HTML页面中提取出所需信息。本系统的爬虫通过XPath的方式解析HTML。

XPath（XML Path Language，XML路径语言），是一种用于定位XML文档中某个或某群元素位置的语言。XPath基于XML的树结构，提供在树中找寻节点的能力。

对于常见的博客，同一网站下大多数博文都有相似的HTML页面结构，因此，对于一个网站，一般只需配置一次XPath即可解析该网站大部分页面。

以CSDN博客的博文为例：

文章标题的XPath："//*[@id="mainBox"]/main/div[1]/div/div/div[1]/h1/text()"；

正文内容的XPath："//*[@id="article_content"]//text()"。

使用XPath提取出页面各项所需内容后，即可通过Elasticsearch对资源进行索引。

17.%2 检索功能实现细节

17.1.%3 基于前缀的搜索建议

一般搜索引擎都会提供输入建议，在用户输入关键词时提供补全建议，优化用户体验。本系统检索功能的搜索框提供了基于前缀的搜索建议，使用Elasticsearch的数据类型completion实现。

在设计学习资源的索引结构时，考虑到需要提供搜索建议补全功能，于是在title与tag两个字段中加入了completion数据类型，以便使用Elasticsearch的Prefix Suggester。

17.2.%3 输入纠错

输入纠错功能基于Elasticsearch的Term Suggester与Phrase Suggester实现，底层核心算法是计算编辑距离。Phrase Suggester与Term Suggester的区别在于，前者会考虑多个Term之间的关系给出搜索建议，而后者只会计算当前Term的编辑距离并给出搜索建议。

在本系统的检索功能中，输入纠错会先基于Phrase Suggester计算出相关度更高的纠错建议，再使用Term Suggester计算出更模糊的纠错建议。

功能使用效果将在本文系统测试部分呈现。

17.3.%3 搜索功能实现

本系统搜索功能基于Elasticsearch的multi-match实现。搜索结果基于分数排序，分数的计算基于用户输入的搜索关键词与资源索引中的内容的匹配度，关键词匹配长度、词语命中数量越高，分数越高。在搜索时，系统会使用到资源索引结构中的3个字段，分别是title、tag.text、content，且三者权重不同，分别为3、2、1。其中，title的权重是3，即匹配时分数会乘3，其他字段同理。权重可以根据用户体验以及搜索效果进行调整。

18.%2 在线评判实现细节

本系统的在线评判功能支持评判Java源码。本文内容所指的评判器均为Java评判器。本系统所实现的评判器不依赖操作系统，不涉及文件读写，不涉及进程创建，具有运行速度快、计时精确、健壮性良好等优点，缺点是无法精确控制内存。因此，该评判器不适用于如竞赛等对内存方面要求较高的应用场景，比较适合于日常编码练习评判。

15.%2.%3 总体流程

代码评判总体流程如图5所示。

图5 代码评判总体流程

16.%2.%3 提交源码的编译

Java语言提供了一系列API，调用javax.tools.ToolProvider.getSystemJavaCompiler()方法，可以获取到当前运行环境的JDK编译器。javac工具本身主要由Java语言编写，因此与在命令行调用javac命令相比，本系统的评判器在Java代码中直接调用了javac代码的方法对源码进行编译。

默认情况下，javac命令编译Java源码后产生的字节码是以文件的形式存储的，但本系统的评判器重写了javax.tools.ForwardingJavaFileManager、javax.tools.SimpleJavaFileObject、java.lang.ClassLoader三个类，让调用编译方法产生的字节码存储在了堆内存中，并且使用自定义的ClassLoader加载用户编写的类。重写ClassLoader破坏了Java类加载的双亲委派模型，目的在于保护评判器原有的代码不被用户提交的代码影响。在自定义的类加载器完成加载类后，即可获得用户代码的Class的实例。后续评判操作涉及大量Java反射技术。

17.%2.%3 测试用例的描述与解析

本系统的测试用例的输入值与期望值都是以JSON（JavaScript Object Notation，JavaScript对象描述语言）格式字符串存储，因此在调用待评测代码之前，需要先把测试用例解析为对应的类的java.lang.Class实例。解析得到类型以后，使用JSON工具对字符串形

式的测试用例输入值做反序列化操作。基本类型、常用包装类型及对应数组类型的反序列化比较简单，难点在于如何序列化与反序列化拥有循环引用的对象。

拥有循环引用的对象，例如双向链表等，由于各个节点的指针会形成环，使用普通的JSON格式无法描述这类对象的结构，且在进行序列化操作的时候，由于循环引用的存在，会导致普通的JSON序列化工具发生无限循环调用，最后导致栈溢出并产生不可恢复的错误，例如java.lang.StackOverflowError。因此，本系统使用了特殊符号来代表JSON对象的引用。以下5条示例是能够表示循环引用的JSON语法的使用示例：

（1）{"$ref":"$"}代表引用根对象；

（2）{"$ref":"@"}代表引用自己；

（3）{"$ref":" "}代表引用父对象；

（4）{"$ref":" / "}代表引用父对象的父对象；

（5）{"$ref":"$.node[0].left"}代表基于路径的引用。

例如，如图6一个拥有4个节点的双向链表用遵循以上语法，可表示为以下JSON格式文本：

图6 拥有4个节点的双向链表

```
{
"value":1,
"pre":null,
"next":{
"value":2,
"pre":{
"$ref":" "
},
"next":{
"value":3,
"pre":{
"$ref":" "
},
"next":{
"value":4,
"pre":{
"$ref":" "
},
"next":null
}
}
}
```

在以上JSON格式文本中，每个节点的pre字段使用了{"$ref":" "}表示对上一节点的引用。在反序列化的过程中，每个对象被反序列化后的实例会被存储在一个数据结构中，当遇到{"$ref":" "}这类语法时，表示当前被反序列化的域是一个引用，指向其他对象实例。如果不使用{"$ref":" "}表示对父对象的引用，由于pre与next这两个域形成的环的存在，这个JSON将无法以有限长度的字符串表

示。

在进行输入值与期望值的反序列化操作前，需要确定反序列化操作的目标类型，因此需要先获取输入值、期望值的java.lang.Class的实例。这些参数的类型信息以字符串的格式存储，通过各种Java反射操作得到java.lang.Class的实例。

图7描述了测试用例输入值从字符串到对象实例的总体流程。

图7 参数类型解析与反序列化总体流程

18.%2.%3 被测代码运行信息

被测代码的运行信息包括了运行耗时、返回值、是否抛出异常、错误堆栈等信息。由于本系统对被测代码运行耗时的统计方式是记录调用方法前后的系统时间戳，不涉及进程调用或者其他耗时操作，因此一般情况下运行耗时统计非常精确。但若此时JVM发生了暂停线程垃圾回收操作，将会导致最后计算得到的运行时间包括了JVM垃圾回收操作暂停线程的时间。

Java语言的垃圾回收完全自动，无法人为干预，但JVM提供了一系列API，其中包括了获得垃圾回收次数与停机时间的方法(周志明，2013)。因此，在记录被测代码运行时间时，还需要记录被测代码运行前后JVM垃圾回收停机的时间。因此，运行时间的计算可以总结为以下公式：

运行时间 = 运行后时间戳 - 运行前时间戳 -（运行后GC耗时 - 运行前GC耗时）

其中，时间戳为以毫秒为单位的UNIX时间戳，GC耗时代表自JVM启动后因GC导致的累计停机毫秒数。

在评判代码时，抛出的异常有两种来源，一种是被测代码运行过程中产生的异常，另一种是被测代码运行超过当前测试用例所设置的timeout导致超时异常。

本系统运行评判功能的超时判断基于java.util.concurrent.FutureTask实现，当在所设置的timeout时间内无法获取到返回值时，即判断为代码运行超时，此时FutureTask的实例会抛出一个java.util.concurrent.TimeoutException异常。

当被测代码运行过程中抛出异常时，会被指定的异常处理代码块捕捉并终止后续测试用例的执行，并将执行结果返回。

19.%2.%3 运行结果评判

当被测代码运行正常并返回一个返回值时，本系统采用对比以JSON格式序列化后的返回值、JSON格式的测试用例期望值的方式，判断返回值是否与期望值一致。对返回值进行序列化操作时，仍然遵循能够表示循环引用的JSON语法，因此结构相同的对象实例在序列化后可以得到相同结构的JSON格式文本，此时即可对比返回值与期望值是否一致。

20.%2.%3 评判器健壮性

代码评判需要执行用户上传的代码，由于用户上传代码时自由度较高，为保证评判器、评判器所在运行环境不受恶意代码的影响，本系统采取3种措施：敏感代码检测、Java本身的Security机制、及时恢复恶意代码所产生影响的逻辑。

Java语言本身拥有完善的访问控制功能。在java.security包下，有一些类与Java访问控制机制相关，覆盖的方面包括了操作系统文件访问权限、Socket端口使用权限、线程相关操作、影响JVM运行状态的代码访问权限等。在默认情况下运行Java程序时，不会启用访问控制机制，因此使用Java进行文件读写、网络连接等操作时完全没有受到限制。

如果本系统代码评判器不做访问控制措施，用户将可以轻易攻击本系统。例如，用户在提交的代码中包含了System.exit(0);代码，评判器执行该代码后将会直接退出。

配置访问控制权限时，需要提供一个policy文件并创建java.lang.SecurityManager的实例，并将该实例设置为当前虚拟机进程的SecurityManager。

本系统所自定义的policy文件custom.policy内容节选：

```
grant codeBase "file:/-" {
permission java.security.AllPermission;
};
```

即对评判器所在运行环境的磁盘上的类授予所有权限。因用户上传的代码不会以文件形式存储，被测代码的将会以最小权限的状态运行，即不能涉及文件操作、不能使用网络相关API、不能调用JVM相关API、不能操作线程等。一旦用户的代码调用了权限不允许的方

法，代码会抛出异常。例如调用System.exit(0);时将会抛出异常java.security.AccessControlException，错误信息为：access denied ("java.lang.RuntimePermission" "exitVM.0")。

Java本身的Security机制能够防止用户调用敏感API，但无法防御计算逻辑上的攻击，比较典型的有无限循环创建强引用对象。

如果代码仅仅包含一个不会占用大量内存的无限循环，运行超时后即可停止线程并返回超时异常。但若无限循环的代码块中包含了创建强引用的对象，例如以下代码：

```
ArrayList<ArrayList<Integer>> l = new ArrayList<>();
for(;;) {
l.add(new ArrayList<>());
}
```

由于以上代码存在强引用，GC无法回收以上代码所产生的对象，将会不断地尝试垃圾回收操作，直接导致CPU占用率飙升、堆内存占用率飙升。且此时调用线程池的shutdownNow方法也无法停止无限循环创建强引用对象的线程。本系统在遇到这种极端情况时，使用了Thread.stop()方法强行杀死线程。线程被强行终止后，GC即可正常回收无用的实例，此时评判其的CPU占用率、堆内存占用率会恢复到正常水平。

21 系统测试

1.%2 功能测试

1.1.%3 资源检索功能测试

资源检索功能基本使用界面如下图8所示。在搜索框输入关键词二叉搜索树后，底部出现了搜索结果，并提供了其他搜索关键词二叉平衡树、二叉排序树

图8 资源检索功能基本界面

在页面底部有分页条，表示关键词二叉搜索树的相关结果有17552条。分页条界面如图9所示。

图9 检索界面底部分页条

在用户输入搜索关键词的过程中，搜索框会给出搜索补全建议。搜索补全建议功能效果如图10所示。

图10 搜索补全建议

当用户输入的搜索关键词中包含了错别字，例如数据结构写成了数剧结构、二叉排序树写成了二叉排序数、java写成了jave，搜索框下方会给出纠正建议。搜索关键词纠错效果如图11所示。

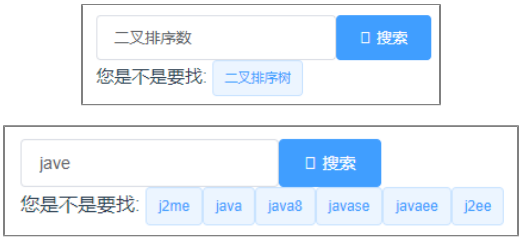

图11 搜索关键词纠错

1.2.%3 代码评判流程测试

本系统的在线编程主要界面如图12所示。

图12 在线编程基本界面

与一些在线评判系统不同，本系统所实现的评判功能不需要使用者编写处理输入、输出的代码或主函数/方法，只需专注于当前题目所需实现的计算逻辑。

在完成图12所示题目的计算逻辑后，如图13题所示，此时点击页面底部的提交按钮，代码将提交到评判器进行评判。

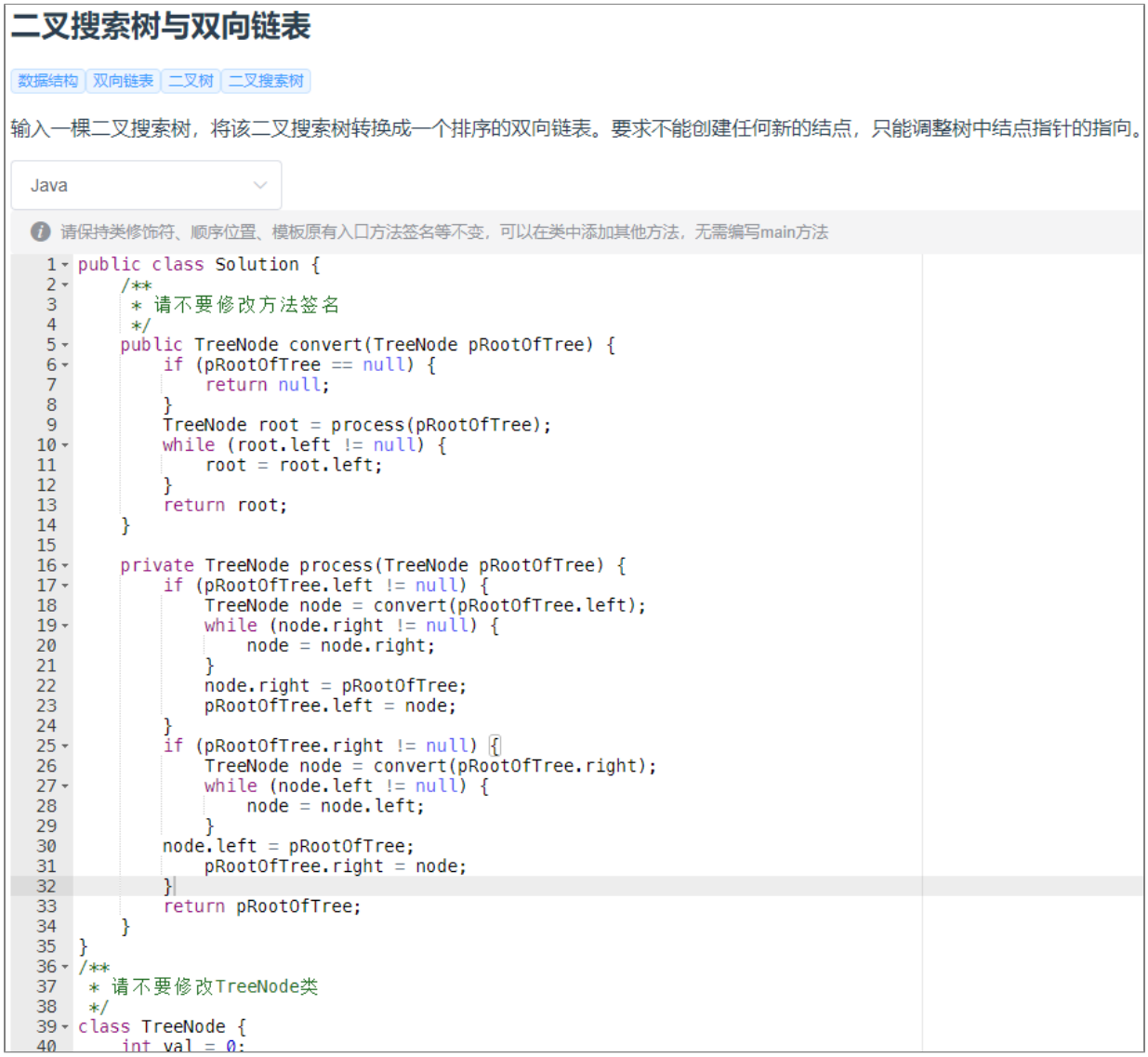

图13 完成代码编写后

代码提交后，用户即可在提交历史列表中查看到代码的提交记录，如图14。其中，提交时间是指用户在本系统前端界面点击提交代码的时间，评判时间是指评判器对本次提交进行评判的时间。运行次数即总共使用的测试用例数，通过用例数为运行返回值与期望值一致的次数，通过率为通过用例数量占总运行次数的比例。

待评判器评判完成后，用户即可查看包括用例通过率等详细信息，如图15所示。

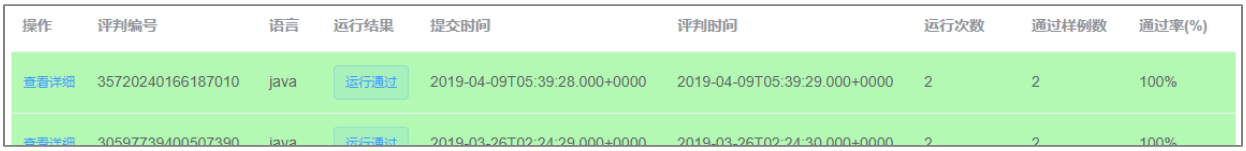

图14 提交历史列表界面

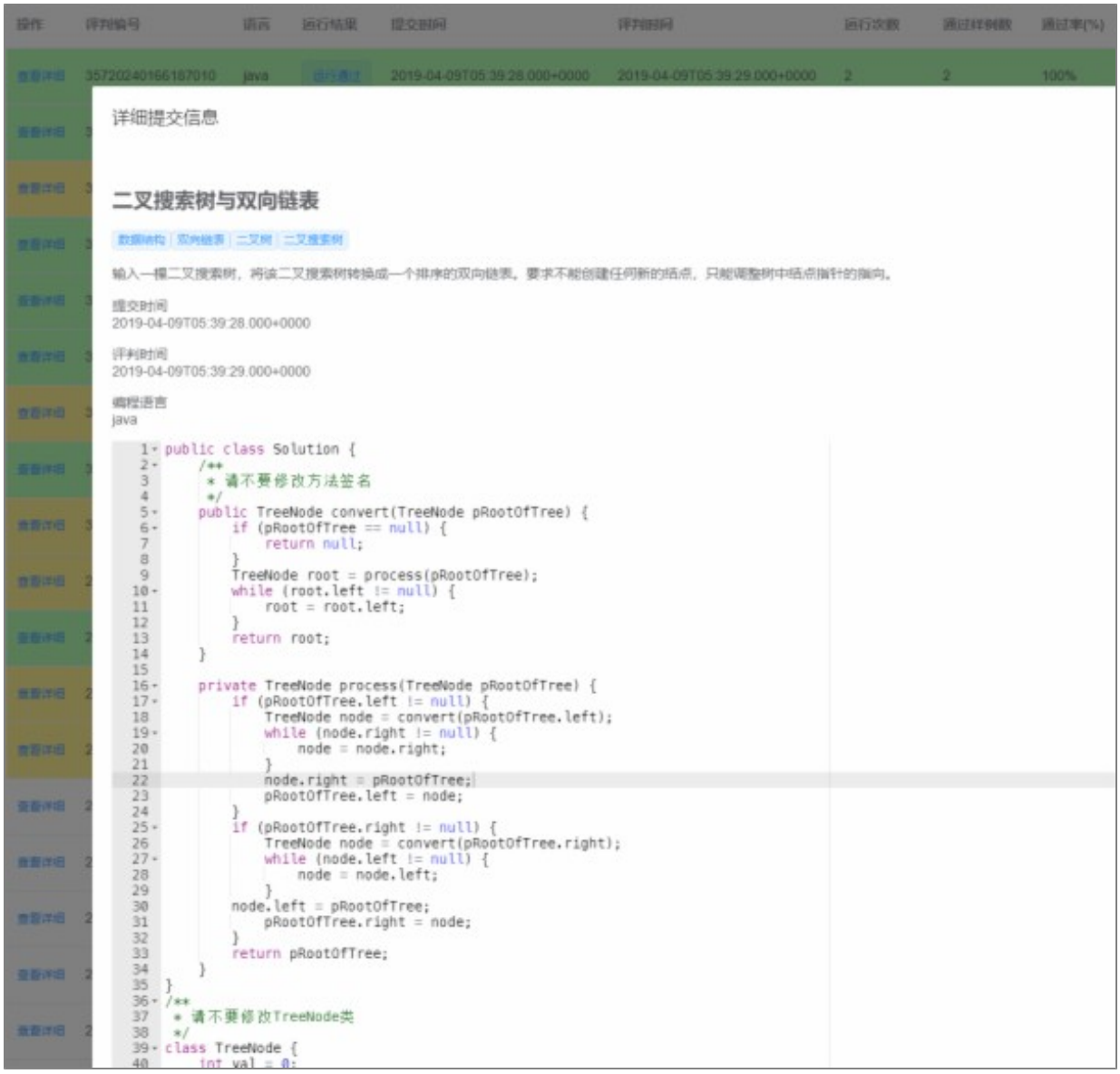

图15 提交历史详细信息界面

当用户提交的代码计算结果错误，用户能够在详细提交信息界面看到测试用例的输入值、期望值，以及代码运行实际返回值。在提交历史列表中，能够看到被测代码运行次数、通过的测试用例数，以及用例通过率。提交代码运行结果错误的效果如图16所示。

操作	评判编号	语言	运行结果	提交时间	评判时间	运行次数	通过样例数	通过率(%)
查看详细	35723083677806590	java	编译通过	2019-04-09T05:50:46.000+0000	2019-04-09T05:50:47.000+0000	2	1	50%

图16 提交代码运行结果错误

当提交的代码出现编译错误、运行过程中抛出异常等情况，在详细信息中可以看到输入值、期望值、实际值、错误堆栈。错误堆栈给出了抛出异常的代码行数，用户可以根据错误堆栈，在代码中快速定位到异常抛出的位置并分析原因。效果如图17所示。

图17 抛出异常、编译出错的情况

19.%2 评判器健壮性测试

当用户提交的代码出现了一些逻辑错误或存在用户尝试提交恶意代码，此时就要求评判器能够应对这些代码，保护自身不受这些问题代码的影响。

在一般情况下，在Java程序中调用代码System.exit(0);会直接导致虚拟机进程退出，现在尝试在本系统提交包含System.exit(0);语句的代码，运行结果如图18所示。虚拟机并没有正常退出，而是抛出了java.security.AccessControlException，即访问控制异常，用户提交的代码没有权限执行exit方法。类似的，在提交的代码中包含无限循环创建线程的代码，如图19所示，会发生访问拒绝的错误，不允许使用Thread相关操作。

详细提交信息

Sum

基础 Java

实现基本加法。

提交时间
2019-04-09T07:06:46.000+0000

评判时间
2019-04-09T07:06:47.000+0000

编程语言
java

测试用例
[1, 2]
期望结果
3
运行结果
java.security.AccessControlException: access denied ("java.lang.RuntimePermission" "exitVM.0")

错误信息
java.security.AccessControlException: access denied ("java.lang.RuntimePermission" "exitVM.0")

错误堆栈
java.security.AccessControlContext.checkPermission(AccessControlContext.java:472)
java.security.AccessController.checkPermission(AccessController.java:884)
java.lang.SecurityManager.checkPermission(SecurityManager.java:549)
java.lang.SecurityManager.checkExit(SecurityManager.java:761)
java.lang.Runtime.exit(Runtime.java:107)
java.lang.System.exit(System.java:971)
Solution.sum(Solution.java:7)
sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
java.lang.reflect.Method.invoke(Method.java:498)
ltd.scau.graduation.commons.entity.MethodMission.call(MethodMission.java:34)
ltd.scau.graduation.commons.entity.MethodMission.call(MethodMission.java:12)
java.util.concurrent.FutureTask.run$$$capture(FutureTask.java:266)
java.util.concurrent.FutureTask.run(FutureTask.java)
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
java.lang.Thread.run(Thread.java:748)

```java
public class Solution {
    /**
     * 请不要修改方法签名
     */
    public int sum(int a, int b) {
        // 在此处编辑
        System.exit(0);
        return a + b;
    }
}
```

图18 调用敏感代码（尝试退出JVM）

详细提交信息

Sum

基础 Java

实现基本加法。

提交时间
2019-04-09T07:46:20.000+0000

评判时间
2019-04-09T07:46:21.000+0000

编程语言
java

错误堆栈
Access denied: Thread

```java
public class Solution {
    /**
     * 请不要修改方法签名
     */
    public int sum(int a, int b) {
        // 在此处编辑
        for (;;) {
            new Thread(() -> {
                for (;;) {
                    new Thread(() -> {
                        for (;;) {
                            System.out.println("hhh");
                        }
                    }).start();
                }
            }).start();
        }
    }
}
```

图19 尝试无限循环创建线程

图20所示代码是一段无限循环创建对象的代码，由于循环创建的对象在List的实例内部的数组中有强引用，在虚拟机进行垃圾回收时，本段代码创建的对象将无法被释放，导致堆内存会快速耗尽，并不断触发垃圾回收，形成恶性循环。

为了监控评判器运行状况，测试时使用Java自带的工具Java VisualVM对虚拟机运行状况进行监控。

在空闲情况下，代码评判器运行状态如图21所示，此时CPU占用率几乎为0，堆内存使用率也较低。

当提交了无限循环创建对象的代码后，虚拟机状态如图22所示，堆内存占用迅速提高，由于当前的堆大小未达到最大限制，虚拟机的堆内存发生了扩容；CPU占用率因为无限循环的代码也瞬间出现了一个峰值。最后，被测代码因为抛出了因运行超时导致的异常java.util.concurrent.TimeoutException。评判器强制杀掉被测代码所在线程，并执行了一次垃圾回收之后，虚拟机运行状况恢复正常，此时CPU占用率恢复到几乎为0，堆内存占用几乎恢复到运行被测代码之前的状况。可见，虚拟机在遇到无限循环创建对象的代码后能够及时恢复正常。

图20 提交无限循环创建对象代码

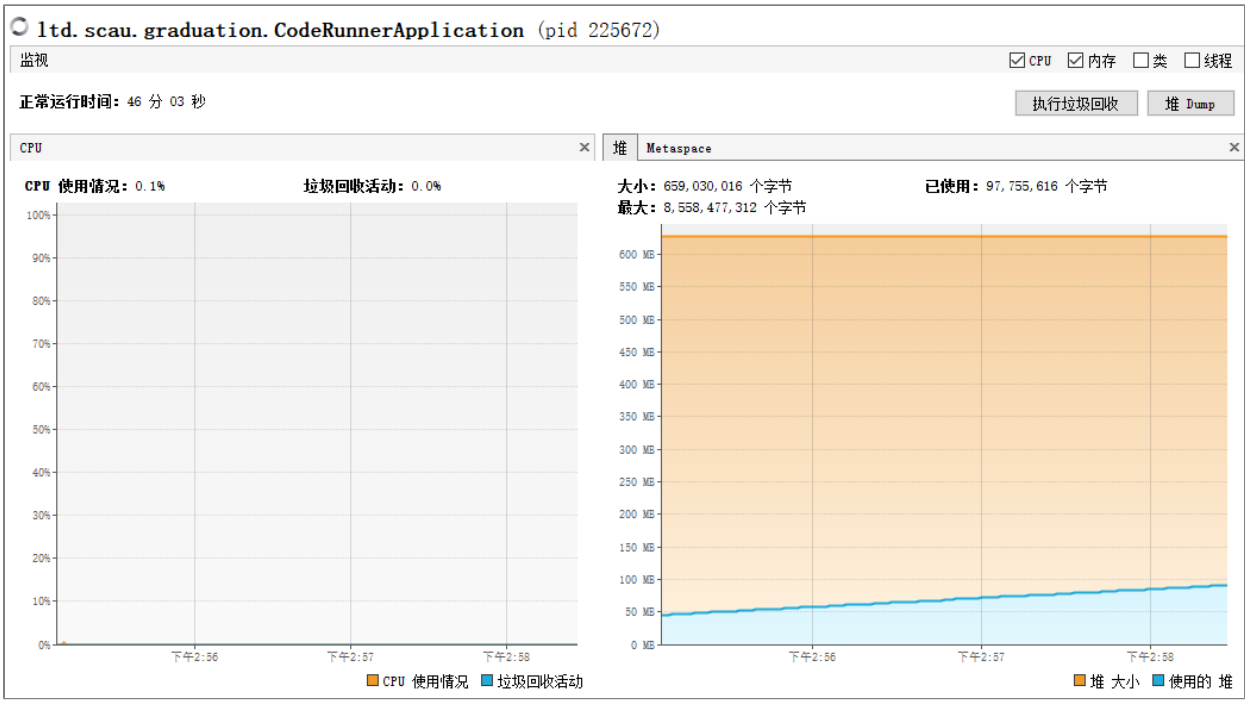

图21 空闲状况下的评判器运行状况

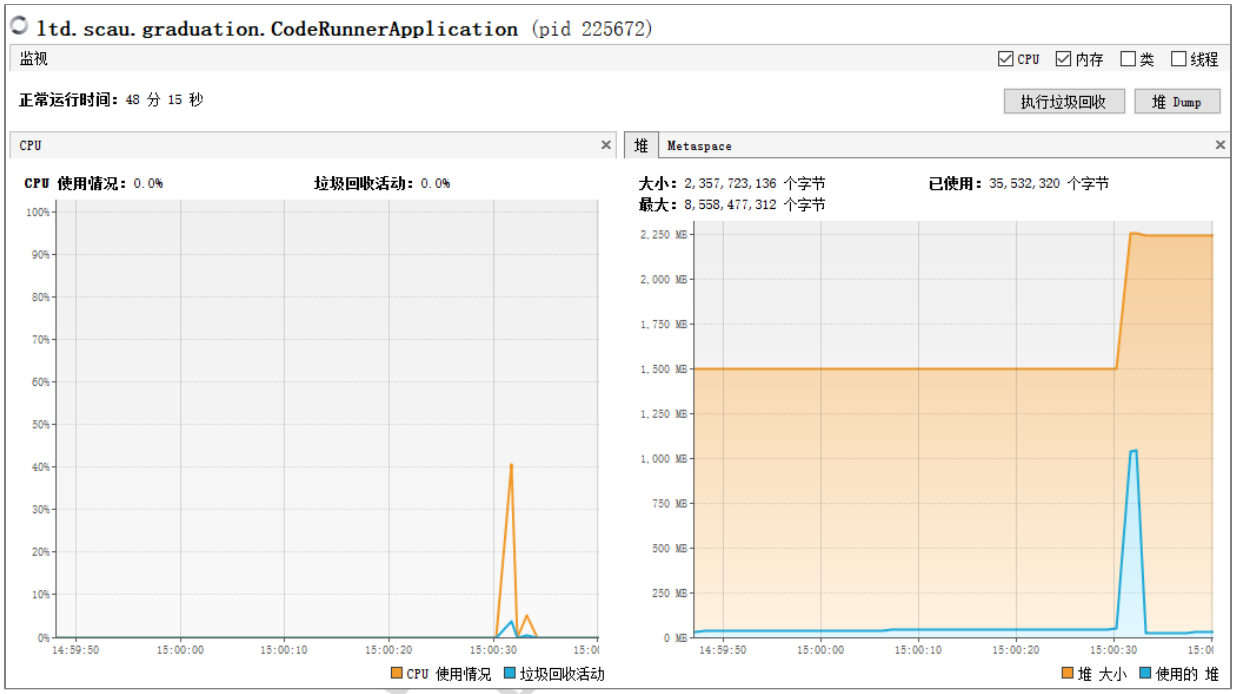

图22 提交无限循环创建对象的代码后虚拟机运行状况

由于无法穷举代码评判器会遇到的所有状况，本节通过重现部分常见的问题的方式，展示了本系统代码评判器所具有的基本的健壮性，能够应对一些日常遇到的异常情况。

22 总结与展望

本文阐述了计算机专业学习资源网站的研究背景和意义，分析了现有的学习资源网站、在线评判网站的局限性以及打破局限性的思路，从数据、功能上入手，设计并实现了基于Scrapy的计算机专业学习资源网站。本文主要进行了以下工作：

（1）探讨了现有通用搜索引擎、在线评判网站的局限性，以及如何打破这种局限性并将理论学习与实践操作结合。

（2）学习资源数据是本系统运行的基础，本文研究了现有技术资源网站页面结构，使用Scrapy框架开发了一个支持集群部署的爬虫，且在索引资源时对页面内容进行了分析，能够在用户检索资源时向用户提供与通用搜索引擎相比更详细的内容。

（3）本文从用户需求、系统架构、可扩展性等方面出发，设计并实现了一个基于微服务架构、支持分布式部署、部署难度低的计算机专业学习资源网站。

（4）在部署难度、健壮性方面，本文研究了多线程、代码编译运行、异常处理等方面，充分利用编程语言自身的特性，实现了一个不依赖操作系统、对运维比较友好、具有一定健壮性的代码评判器。

（5）本文对系统的功能和评判器的安全性进行了测试。根据测试结果，系统检索功能的使用流程与常见的通用搜索引擎相似，比较符合普通用户使用通用搜索引擎的操作习惯；在线编程的代码评判器具有基本的健壮性，能够应对常见代码的错误、异常状况。

由于时间与个人条件的限制，本文还存在一些需要在后续研究中需要改善、解决的问题。后续的工作将主要解决以下问题：

（1）增加在线评判功能所支持的编程语言数量

本文目前所实现的在线评判功能仅支持Java，对偏向于使用其他语言的用户不友好。

（2）研究对学习资源打标签的算法

本系统目前已索引的学习资源的标签均为文章作者所提供。部分资源作者没有提供标签，该资源在本系统中无法与编程题目相关联，无法实现理论与实践结合。在后续研究工作中，需要实现一种算法对没有标签的资源打上标签，如通过对文章分词等。

（3）完善本系统的前端界面，优化用户体验

由于本文工作研究偏向于后端实现，前端的用户体验有待提高。前端直接面向用户，是系统的门面，因此在后续工作中需要对系统前端进行完善与优化。

参考文献

陈荣华,杨蓓.门户网站搜索引擎优化策略研究[J].电脑知识与技术,2018,14(33):173-174.

胡二彪.基于SEO的教学资源网站设计与开发[D].河南师范大学,2012.

黄洪波.大规模编程题在线评判技术研究[D].华南农业大学,2016.

李嘉,赵凯强,李长云.Web前端开发技术的演化与MVVM设计模式研究[J].电脑知识与技术,2018,14(02):221-222+251.

刘思林.Scrapy分布式爬虫搜索引擎[J].电脑知识与技术,2018,14(34):186-188.

王伟,魏乐,刘文清,舒红平.基于ElasticSearch的分布式全文搜索系统[J].电子科技,2018,31(08):56-59+65.

郑彬彬. 基于微服务的OJ系统重构与优化[D].东华大学,2017.

中华人民共和国工业和信息化部. 2017年1-12月全国软件和信息技术服务业主要指标快报表[EB/OL].运行监测协调局.2018-01-26.
http://www.miit.gov.cn/n1146312/n1146904/n1648374/c6037922/content.html.

中华人民共和国工业和信息化部. 2018年全国软件和信息技术服务业主要指标快报表[EB/OL].运行监测协调局.2019-02-01.
http://www.miit.gov.cn/n1146312/n1146904/n1648374/c6633874/content.html.

周志明.深入理解Java虚拟机 JVM高级特性与最佳实践 第2版[M].北京:机械工业出版社,2013.

Elasticsearch B.V. Elasticsearch Reference [6.5] | Elastic[EB/OL]. Elasticsearch B.V, 2018-11-05
https://www.elastic.co/guide/en/elasticsearch/reference/6.5/index.html.

Evan You. Vue.js[EB/OL]. https://cn.vuejs.org.

Phillip Webb, Dave Syer, Josh Long, et al. Spring Boot Reference Guide[EB/OL]. Pivotal Software,
2018-10-30.https://docs.spring.io/spring-boot/docs/2.1.0.RELEASE/reference/htmlsingle.

Redis. Redis Documentation[EB/OL]. https://redis.io/documentation.

Scrapy. Scrapy 1.6 documentation[EB/OL].2019-01-30. https://doc.scrapy.org/en/latest.

致 谢

窗间过马，历时5个月，在老师的悉心指导与同学的帮助下，我顺利地完成了我的毕业论文。

感谢司国东老师在本次毕业设计中给我的指导。他是我计算机专业基础课程的授课教师，我所掌握的计算机基础离不开老师认真、负责地传道授业解惑。在本文研究过程中，有许多关键的技术问题阻碍着我的进度，老师非常耐心地帮助我分析问题的根本原因与设计解决方案，让我能够突破重重障碍，实现我所设计的系统，完成毕业论文。

感谢在我本科四年期间所有为我授课的老师，我现在所拥有的基础知识与技术积累都离不开每位老师以诲人不倦的态度，在每一堂课上耐心的教导。

感谢我身边的同学们，在课堂上齐心协力完成课堂任务、课程设计，在日常生活中进行技术交流，在遇到难题时大家互相帮助。

感谢父母多年的养育之恩、对我攻读计算机专业的全力支持。他们在我学习生涯提供了很多物质与精神上的帮助，为我创造了完美的学习条件。

感谢所有帮助我的人。

大家的恩情，本人铭记于心。

华南农业大学

本科生毕业论文成绩评定表

学号	201525050420	姓名	吴伟杰	专业	网络工程
毕业论文题目	基于Scrapy框架的计算机专业学习资源网站				
指导教师评语成绩（百分制）：			指导教师签名：		年 月

		评分项目		分值	得分
		1	专业培养目标	5	

评阅人评语及成绩评定	成绩评定标准	选题质量20%	2	课题难易度与工作量	10	
			3	理论意义或生产实践意义	5	
		能力水平40%	4	查阅文献资料与综合运用知识能力	10	
			5	研究方案的设计能力	10	
			6	研究方法和手段的运用能力	10	
			7	外文应用能力	10	
		成果质量40%	8	写作水平与写作规范	20	
			9	研究结果的理论或实际应用价值	20	
	评阅人评语成绩（百分制）：			评阅人签名：	年 月	

答辩小组评语及成绩评定	评价项目	具体要求（A级标准）
	论文质量	论文结构严谨，逻辑性强；有一定的学术价值或实用价值；文字表达准确流畅；论文格式规
	论文报告、讲解	思路清晰；概念清楚，重点（创新点）突出；语言表达准确；报告时间、节奏掌握好。
	答辩情况	答辩态度认真，能准确回答问题
	答辩小组评语是否同意通过论文答辩（打）1. 同意2. 不同意成绩（百分制）： 答辩小组 年 月 日	
成绩总评	论文总评分数： 教学院长签名： 学院盖章： 年 月 日	

续上表：

- 说明：

相似片段中"综合"包括：

《中文主要报纸全文数据库》 《中国专利特色数据库》 《中国主要会议论文特色数据库》 《港澳台文献资源》 《图书资源》 《维普优先出版论文全文数据库》 《年鉴资源》 《古籍文献资源》 《IPUB原创作品》

- 声明：

报告编号系送检论文检测报告在本系统中的唯一编号。

本报告为维普论文检测系统算法自动生成，仅对您所选择比对资源范围内检验结果负责，仅供参考。

关注微信公众号