大学生创业平台的开发与系统优化策略研究

武斌

电子系 中国海洋大学

Department electronic engineering
Ocean University Of China

2016 年 9 月 8 日

概述

1. 选题背景
2. 研究内容
3. 研究方案
4. 进展情况

选题背景

课题来源

- 创业过程中负责创游记平台的系统开发、维护和 WEB 开发
- 海信实习过程中参与小微系统开发、环境部署与优化的工作

课题目的

- 基于 WEB 技术开发一款服务于大学生创新创业的网络平台
- 研究系统开发过程中的代码、持续集成、负载等方面的优化策略

研究内容

1. 创游记平台的开发
2. 系统优化策略研究

创游记平台的开发

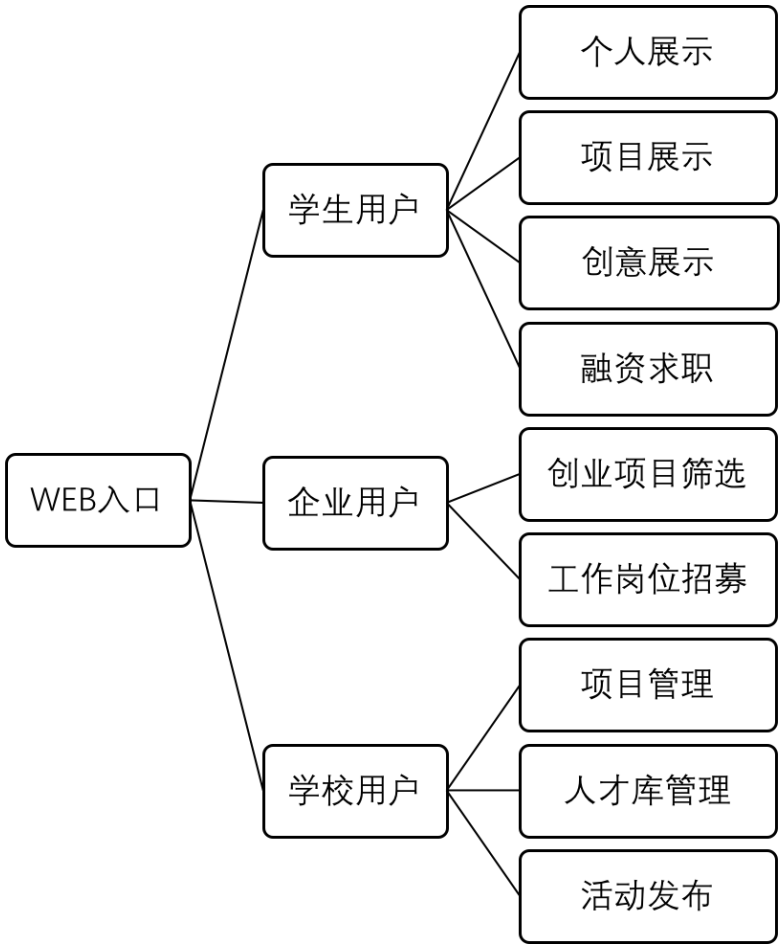
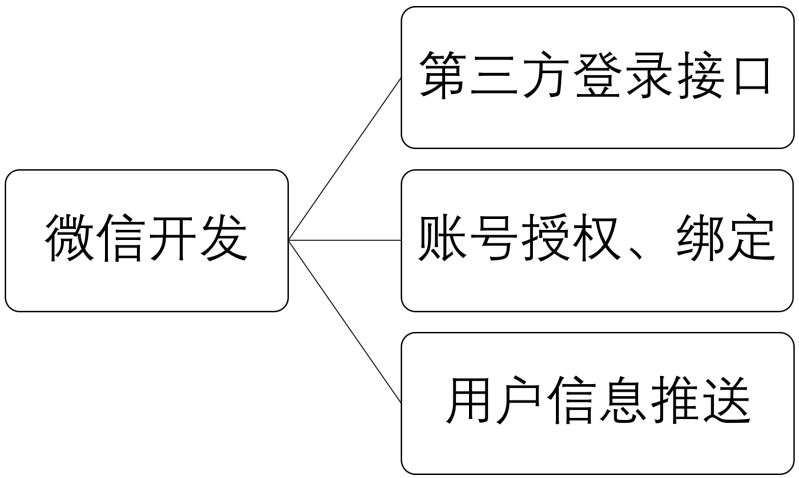


图: 平台功能

创游记平台的开发



开发目标

1. 针对不同用户完成不同功能模块的开发
2. 实现微信端的浏览、访问、授权
3. 实现针对定向用户的个性化推荐

针对平台的优化策略研究

推荐系统开发

- 探索基于平台用户数据的推荐系统开发，实现学生用户的融资求职机会推荐和企业用户的创业项目推荐

针对系统的优化策略研究

持续集成环境构建

- 通过境搭建一个持续集成环境，研究系统自动构建过程，包括自动编译、分发、部署和测试等功能

系统缓存开发

- 探索系统缓存对于 WEB 系统性能的影响
- 基于本系统研究系统的缓存机制和策略
- 基于本系统进行缓存机制的开发和性能测试

搜索优化

- 研究在保证搜索的实时、稳定、可靠和快速基础上，降低系统负载的方式
- 探索基于分布式搜索的系统接口开发和性能测试

研究方案

1. 创游记平台的开发
2. 推荐系统研究与开发
3. 自动化测试部署方案研究
4. 系统缓存和优化方案研究

一、创游记平台的开发

基本环境搭建

- 搭建基于 Linux+Apache+MySQL+PHP/Python 的 WEB 开发环境
- 搭建本地的代码版本控制工具 GitLab

一、创游记平台的开发

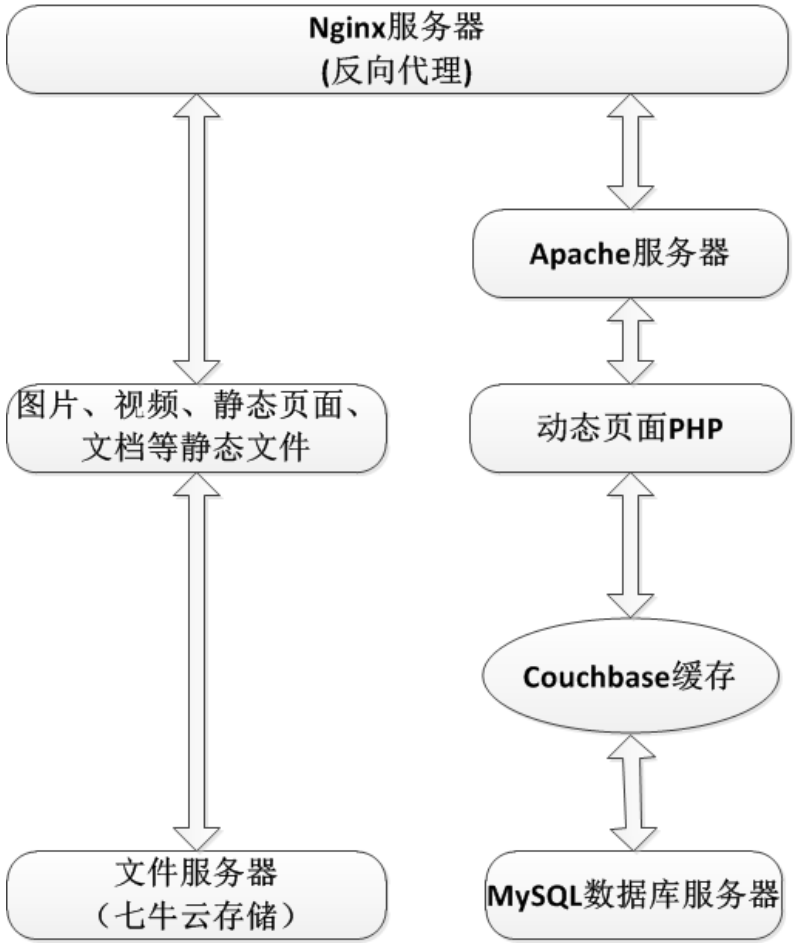


图: 平台基本框架

二、[4] 推荐系统研究与开发

本课题使用的推荐算法和框架

- 本课题主要使用协同过滤推荐算法中的基于用户的协同过滤算法
- [6] 本课题使用的框架是基于 Python 的 Crab 推荐系统引擎

协同过滤推荐算法 (Collaborative Filtering Recommendation)

- 基于用户的协同过滤 (User-Based CF)
  找到和目标用户兴趣相似的用户集合，然后给目标用户推荐这个集合的用户喜欢的物品
- 基于物品的协同过滤 (Item-Based CF)
  给目标用户推荐与他喜欢的物品相似度较高高的物品

基于用户的协同过滤算法的主要公式和步骤

- [1] 基于用户的协同过滤的关键在于计算用户与用户之间的兴趣相似度，主要使用余弦相似度来计算：

$$\omega_{\mu\nu} = \frac{|\mathrm{N}(\mu) \cap \mathrm{N}(\nu)|}{\sqrt{|\mathrm{N}(\mu) \parallel \mathrm{N}(\nu)|}}$$

$\omega_{\mu\nu}$ 代表用户 μ 与 ν 之间的兴趣相似度，$\mathrm{N}(\mu)$ 表示用户 μ 曾经喜欢过的物品集合，$\mathrm{N}(\nu)$ 表示用户 ν 曾经喜欢过的物品集合

- 根据上述描述，可以有如下算法步骤
  1. 建立物品-用户的倒排表 (Inverted Index)
  2. 计算用户与用户之间的共享矩阵 $\mathrm{C}[\mu][\nu]$，表示用户 μ 与 ν 喜欢相同物品的个数
  3. 计算用户与用户之间的相似度矩阵 $\omega[\mu][\nu]$，根据上述相似度计算公式计算
  4. 用上面的相似度矩阵来给用户推荐和他兴趣相似的用户喜欢的物品

二、推荐系统研究与开发

Crab 推荐系统引擎

- Crab 是基于 Python 开发的开源推荐软件，其中实现的方法有 item 和 user 的协同过滤
- 基于 scikit-learn 库 (Python 下的一个机器学习库)，建立在 NumPy，SciPy 和 matplotlib 模块之上，为用户提供各种机器学习算法接口

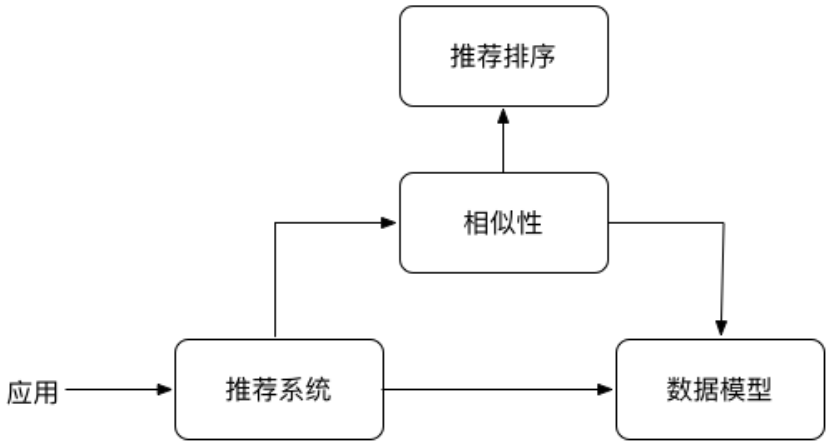
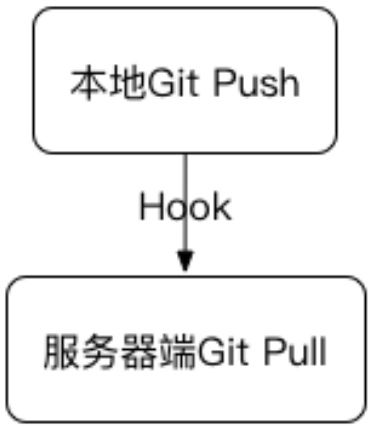


图: crab 框架

三、持续集成方案研究

本课题搭建持续集成的目的和研究

- 在使用持续集成方案之前，系统的部署流程是直接将代码通过 GitLab 的钩子 (hook) 将代码同步到服务器端的 WEB 目录下，没有进行自动化的构建、测试和部署。



图：原部署流程

三、持续集成方案研究

本课题搭建持续集成的目的和研究

- 使用持续集成方案之后，通过 GitLab 的钩子 (hook) 触发系统的持续集成工具，对代码进行合并、构建、测试和部署，出现重大问题时可以回滚。

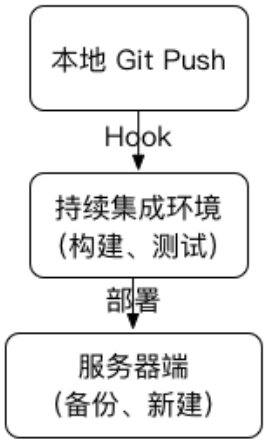


图: 现部署流程

三、持续集成方案研究

关于持续集成

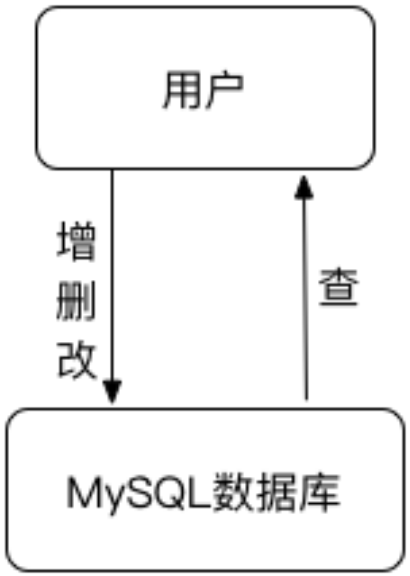
持续集成是一种软件开发实践，通过频繁地（一天多次）将代码集成到主干。每次集成都通过自动化的构建（包括编译，发布，自动化测试）来验证，从而尽早地发现集成错误[2]。

集成环境的搭建和配置

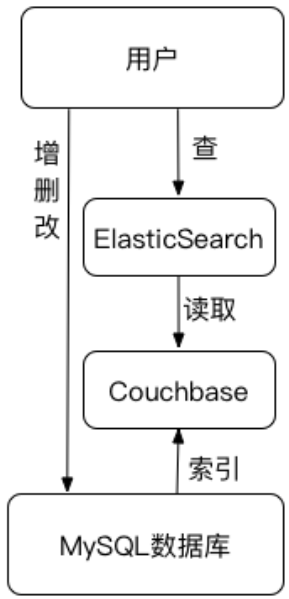
- 本地搭建 Jenkins 持续集成服务器，并且安装针对 PHP 的构建、测试插件
- 开发 Hook 触发脚本和持续部署脚本，实现服务器端的的代码更新和文件备份

四、系统缓存和优化方案研究

- 系统之前版本和现在版本使用的缓存机制对比



图：之前数据操作模型



图：当前数据操作模型

四、系统缓存和优化方案研究

Couchbase 缓存[5]

- Couchbase 是一个高性能以及可用性强的 NoSQL 数据库引擎
- 使用 Couchbase 将系统的业务数据在系统启动时加载到缓存中，降低系统的丢包率并加快数据读取速度

ElasticSearch 搜索引擎[3]

- Elasticsearch 是一个基于 Apache Lucene(TM) 的开源搜索引擎
- 实时分析搜索引擎，每个字段都被索引并可被搜索，能够达到搜索实时、稳定、可靠和快速，支持通过 HTTP 请求，使用 JSON 进行数据索引[7]

四、系统缓存和优化方案研究

七牛云存储

- 将平台的图片和视频等媒体数据上传到七牛云存储
- 通过七牛的 CND 加速加快系统图片和视频的加载速度

WEB 缓存的意义和作用

- 减少网络带宽消耗
- 降低服务器压力
- 减少网络延迟，加快页面打开速度

进展情况

已完成

- 平台功能开发
- Jenkins 自动化测试、部署工具的配置和部署脚本开发
- Couchbase 缓存机制的开发
- ElasticSearch 分布式全文搜索引擎的部署和测试

进行中

- 缓存和搜索接口的开发
- 推荐系统的开发

References I

[1] Jark. 推荐系统学习：协同过滤实现.

[2] 卞孟春. 基于 jenkins 的持续集成方案设计与实现. Master's thesis, 中国科学院大学, 2014.

[3] 唐志贤 王超姜康, 冯钧. 基于 elasticsearch 的元数据搜索与共享平台. 计算机与现代化, 2:117–121, 2015.

[4] 刘贺平王国霞. 个性化推荐系统综述. 计算机工程与应用, 07, 2012.

[5] 王旭铭. 负载均衡集群中的会话保持研究. Master's thesis, 中山大学, 2013.

[6] 胡新明. 基于商品属性的电子商务推荐系统研究. Master's thesis, 华中科技大学, 2012.

[7] 许大宏. Elasticsearch 在车牌识别系统中的应用研究. 计算机时代, 12, 2014.