

硕士专业学位论文

回放对比平台的设计与实现

作者：吴艳丽

导师：刘铎

北京交通大学

2018年3月

学位论文版权使用授权书

本学位论文作者完全了解北京交通大学有关保留、使用学位论文的规定。特授权北京交通大学可以将学位论文的全部或部分内容编入有关数据库进行检索，提供阅览服务，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构送交论文的复印件和磁盘。学校可以为存在馆际合作关系的兄弟高校用户提供文献传递服务和交换服务。

（保密的学位论文在解密后适用本授权说明）

学位论文作者签名： 导师签名：

签字日期： 年 月 日 签字日期： 年 月 日

学校代码：10004

密级：

北京交通大学

**硕士专业学位论文**

作者姓名：吴艳丽 学 号：16126222

导师姓名：刘铎 职 称：副教授

工程硕士专业领域：软件工程 学位级别：硕士

北京交通大学

2018年3月

致谢

本论文的工作是在我的导师刘铎老师的悉心指导下完成的，从论文选题开始到论文写作结束，刘铎老师都给我提出了很多宝贵的意见，正是因为刘铎老师的悉心指导，我才能顺利完成毕业论文的撰写。刘铎老师严谨的治学态度和科学的工作方法给了我极大的影响，此外刘铎老师在学习和生活上也给予了我很大的关心和帮助，在此衷心感谢两年来刘铎老师对我的教育和指导。

感谢公司提供给我宝贵的实习机会，让我参与到公司的项目中去，从中学到了很多在学校学不到的东西，使我在实践中得到了极大的锻炼。

感谢企业导师对我的耐心指导，使我能够快速提高技术水平，并且感谢他们可以指出我在工作中的不足，使我可以及时纠正以便更好的融入团队进行工作。

另外，感谢家人以及所有在我论文撰写期间帮助过我的同学，你们对我的理解和帮助使我能够在学校专心完成我的学业。

最后，感谢所有参与我的论文评审和答辩的老师，你们给了我一个审视我两年来学习成果的机会，我希望可以得到你们的肯定。毕业并不意味着学习的终点，我将会以同样严谨的学习态度面对接下来的学习、生活和工作。

摘要

一般就是三部分：大环境和你所做的东西的重要性啊它好重要；这个东西都怎么做得啊你个人承担的工作是啥；这东西现在用的怎么样？

一般750字左右，不要超过一页。样式为“毕业论文正文”

**关键词：**

ABSTRACT

一般为1000个左右实词。一般尽量不要超过一页。

**KEYWORDS：**

目录

摘要 iii

ABSTRACT iv

1 引言 1

1.1项目背景 1

1.2项目开发目的及意义 2

1.3本人工作 3

1.4项目应用现状 4

1.5论文组织结构 5

1.6本章小结 6

2相关理论和关键技术 1

2.1服务端架构技术 1

2.1.1Yaf框架 2

2.1.2MVC 2

2.1.3Mirror框架 2

2.1.4 2

2.1.5 2

2.1.6 2

2.2前端架构技术 2

2.2.1 2

2.2.2 3

2.2.3 3

2.2.4 3

2.2.5 3

2.2.6 3

2.3Nginx服务器 3

2.3.1 3

2.3.2 3

2.3.3 4

2.3.4 4

2.3.5 4

2.3.6 4

2.4前端对比技术 4

2.4.1 4

2.4.2 4

2.4.3 4

2.4.4 5

2.4.5 5

2.4.6 5

2.5并发处理 5

2.5.1 5

2.5.2 5

2.5.3 5

2.5.4 5

2.5.5 6

2.5.6 6

2.6 6

2.6.1 6

2.6.2 6

2.6.3 6

2.6.4 6

2.6.5 6

2.6.6 7

2.7 7

2.7.1 7

2.7.2 7

2.7.3 7

2.7.4 7

2.7.5 7

2.7.6 7

2.8 8

2.8.1 8

2.8.2 8

2.8.3 8

2.8.4 8

2.8.5 8

2.8.6 8

2.9 本章小结 8

3需求分析 1

3.1功能性需求分析 1

3.1.1 2

3.1.2 2

3.1.3 2

3.1.4 2

3.1.5 2

3.1.6 2

3.2非功能性需求 2

3.2.1 2

3.2.2 3

3.2.3 3

3.2.4 3

3.2.5 3

3.2.6 3

3.3 3

3.3.1 3

3.3.2 3

3.3.3 4

3.3.4 4

3.3.5 4

3.3.6 4

3.4 4

3.4.1 4

3.4.2 4

3.4.3 4

3.4.4 5

3.4.5 5

3.4.6 5

3.5 5

3.5.1 5

3.5.2 5

3.5.3 5

3.5.4 5

3.5.5 6

3.5.6 6

3.6 6

3.6.1 6

3.6.2 6

3.6.3 6

3.6.4 6

3.6.5 6

3.6.6 7

3.7 7

3.7.1 7

3.7.2 7

3.7.3 7

3.7.4 7

3.7.5 7

3.7.6 7

3.8 8

3.8.1 8

3.8.2 8

3.8.3 8

3.8.4 8

3.8.5 8

3.8.6 8

3.9 本章小结 8

4系统概要设计 1

4.1系统架构设计 1

4.1.1 1

4.1.2 1

4.1.3 2

4.1.4 2

4.1.5 2

4.1.6 2

4.2功能模块设计 2

4.2.1 2

4.2.2 2

4.2.3 2

4.2.4 3

4.2.5 3

4.2.6 3

4.3数据库设计 3

4.3.1数据库逻辑设计 3

4.3.2数据库表结构设计 3

4.3.3 3

4.3.4 3

4.3.5 4

4.3.6 4

4.4 4

4.4.1 4

4.4.2 4

4.4.3 4

4.4.4 4

4.4.5 4

4.4.6 5

4.5 5

4.5.1 5

4.5.2 5

4.5.3 5

4.5.4 5

4.5.5 5

4.5.6 5

4.6 6

4.6.1 6

4.6.2 6

4.6.3 6

4.6.4 6

4.6.5 6

4.6.6 6

4.7 6

4.7.1 7

4.7.2 7

4.7.3 7

4.7.4 7

4.7.5 7

4.7.6 7

4.8 7

4.8.1 7

4.8.2 8

4.8.3 8

4.8.4 8

4.8.5 8

4.8.6 8

4.9 本章小结 8

5系统详细设计与实现 1

5.1服务端主要模块设计 1

5.1.1 1

5.1.2 2

5.1.3 2

5.1.4 2

5.1.5 2

5.1.6 2

5.2前端主要模块设计 2

5.2.1 2

5.2.2 2

5.2.3 3

5.2.4 3

5.2.5 3

5.2.6 3

5.3重难点部分设计 3

5.3.1 3

5.3.2 3

5.3.3 3

5.3.4 4

5.3.5 4

5.3.6 4

5.4 4

5.4.1 4

5.4.2 4

5.4.3 4

5.4.4 4

5.4.5 5

5.4.6 5

5.5 5

5.5.1 5

5.5.2 5

5.5.3 5

5.5.4 5

5.5.5 5

5.5.6 6

5.6 6

5.6.1 6

5.6.2 6

5.6.3 6

5.6.4 6

5.6.5 6

5.6.6 6

5.7 7

5.7.1 7

5.7.2 7

5.7.3 7

5.7.4 7

5.7.5 7

5.7.6 7

5.8 7

5.8.1 8

5.8.2 8

5.8.3 8

5.8.4 8

5.8.5 8

5.8.6 8

5.9 本章小结 8

6实现效果展示 9

6.1 9

6.1.1 10

6.1.2 10

6.1.3 10

6.1.4 10

6.1.5 10

6.1.6 10

6.2 10

6.2.1 10

6.2.2 11

6.2.3 11

6.2.4 11

6.2.5 11

6.2.6 11

6.3 11

6.3.1 11

6.3.2 11

6.3.3 12

6.3.4 12

6.3.5 12

6.3.6 12

6.4 12

6.4.1 12

6.4.2 12

6.4.3 12

6.4.4 13

6.4.5 13

6.4.6 13

6.5 13

6.5.1 13

6.5.2 13

6.5.3 13

6.5.4 13

6.5.5 14

6.5.6 14

6.6 14

6.6.1 14

6.6.2 14

6.6.3 14

6.6.4 14

6.6.5 14

6.6.6 15

6.7 15

6.7.1 15

6.7.2 15

6.7.3 15

6.7.4 15

6.7.5 15

6.7.6 15

6.8 16

6.8.1 16

6.8.2 16

6.8.3 16

6.8.4 16

6.8.5 16

6.8.6 16

6.9 本章小结 16

7系统测试概述 17

7.1功能测试 17

7.1.1 17

7.1.2 18

7.1.3 18

7.1.4 18

7.1.5 18

7.1.6 18

7.2接口测试 18

7.2.1 18

7.2.2 18

7.2.3 19

7.2.4 19

7.2.5 19

7.2.6 19

7.3性能测试 19

7.3.1 19

7.3.2 19

7.3.3 19

7.3.4 20

7.3.5 20

7.3.6 20

7.4 20

7.4.1 20

7.4.2 20

7.4.3 20

7.4.4 20

7.4.5 21

7.4.6 21

7.5 21

7.5.1 21

7.5.2 21

7.5.3 21

7.5.4 21

7.5.5 21

7.5.6 22

7.6 22

7.6.1 22

7.6.2 22

7.6.3 22

7.6.4 22

7.6.5 22

7.6.6 22

7.7 23

7.7.1 23

7.7.2 23

7.7.3 23

7.7.4 23

7.7.5 23

7.7.6 23

7.8 23

7.8.1 24

7.8.2 24

7.8.3 24

7.8.4 24

7.8.5 24

7.8.6 24

7.9 本章小结 24

8结论与展望 25

8.1 25

8.1.1 26

8.1.2 26

8.1.3 26

8.1.4 26

8.1.5 26

8.1.6 26

8.2 26

8.2.1 26

8.2.2 27

8.2.3 27

8.2.4 27

8.2.5 27

8.2.6 27

8.3 27

8.3.1 27

8.3.2 27

8.3.3 28

8.3.4 28

8.3.5 28

8.3.6 28

8.4 28

8.4.1 28

8.4.2 28

8.4.3 28

8.4.4 29

8.4.5 29

8.4.6 29

8.5 29

8.5.1 29

8.5.2 29

8.5.3 29

8.5.4 29

8.5.5 30

8.5.6 30

8.6 30

8.6.1 30

8.6.2 30

8.6.3 30

8.6.4 30

8.6.5 30

8.6.6 31

8.7 31

8.7.1 31

8.7.2 31

8.7.3 31

8.7.4 31

8.7.5 31

8.7.6 31

8.8 111 32

8.8.1 32

8.8.2 32

8.8.3 32

8.8.4 32

8.8.5 32

8.8.6 123 32

8.9 本章小结 32

9 结论 33

参考文献 34

附录 A 35

索引 36

作者简历及攻读硕士/博士学位期间取得的研究成果 37

独创性声明 38

学位论文数据集 39

1. 引言

回放对比平台系统源于作者在北京嘀嘀无限科技发展有限公司实习时候公司的项目，目的是开发一个平台，这个平台能直观看到接口代码改动对返回结果造成的区别，并区分出这些区别是否是bug，提高测试效率及测试覆盖率。出本章着重介绍项目的背景、实现目的及意义，并概述了论文的主要工作及结构安排。

* 1. 项目背景

在公司里，本人所在部门的服务还处于探索阶段，提供给用户的服务形态还不稳定，需要不断的探索给用户提供新的服务，来查看用户真实的需要，并根据用户的反馈不断优化服务，在这个背景下，部门内上线的频率非常高，根据公司数据统计，在2017年9月，部门上线次数共129次，这么频繁的上线的情况下，测试人员需要同时关注很多项目，长期加班来赶进度也无法完成所有的测试，在高压且多项目的情况下，测试人员因为跟进不同的项目，精力会被不同的人分散，导致效率特别低下，这时开发会和测试人员一起定义一些认为改动牵扯很少的项目，但是这些被认为是简单的改动，往往会导致线上问题，影响到用户的体验，造成非常严重的后果；同时，在提测的时候，开发对提测返回定义不清楚，接口之间存在耦合，会导致测试返回覆盖不全面，对改动影响的范围定义不清楚，导致漏测，把问题带入线上，在129次上线过程中，有问题共回滚13次，这个回滚率已经达到了10%，对部门和公司都有很不好的影响。同时，开发在这种情况下开发出来的项目质量低下，提测给测试的项目需要不断返工，这时测试就需要一遍一遍的做同样的事情。在这种高频上线的节奏下，非常需要一个能直观看到每一次改动对所有接口的返回结果造成了什么样的影响，同时能知道这些影响的确是因为代码改动造成的合理影响，还是出现了bug，在开发提测之前，他们能确保提测质量；测试时，测试人员能快速定位问题，减少不必要的人力资源浪费，同时，能保证所有项目都被测试过，不会出现测试人员无人力而放过一些项目而导致了线上问题。

* 1. 项目意义和目的

当前所处的背景，迫切要求一个能提高工作效率和工作质量的工具出现。此项目能直观的看到代码改动对接口返回结果的影响，同时在开发提测的时候能及时通知到测试人员，并给测试人员提供相关的测试信息，例如提测工程，分支，注意事项等等，让测试人员对即将测试的项目基础信息有初步的认识。

由于项目的改动很大可能会影响接口返回的结果，直接通过普通的文本对比不能直接地、直观地找到代码的bug，回放对比平台能让用户提供验证条件，告诉平台某个字段变成某个形式属于正常现象，当这个字段出现区别的时候，平台会验证是否满足设定的验证条件，满足则能通过，不能则直观的标识不同，也能验证返回结果字段的属性，如可以验证字段是否是字符串等。

同时，平台能将整个流程控制在半小时内，是一个轻量级，不会耽误上线时间。

可以直观的看到新改动代码对接口返回结果的影响，提高开发的提测质量，提高项目测试的效率，将一些低级的错误在项目的前期被修改；能及时的召回覆盖率的问题；作为准出环节的把关，于所需时间不超过半小时，也能满足轻量级的需求。可以节约开发和测试人员的大量时间。

* 1. 本人主要工作

整个回放对比平台是由本人独立完成的，包括需求收集，需求分析，概要设计，详细设计，项目的开发实现（包括前端和后端），测试及后期的维护工作。

* 1. 项目应用现状

目前，项目已经上线并投入使用，开发在提测之前，必须在系统上跑一遍回放对比，通过之后再提测；在项目上线之前，必须在系统上跑一遍回放对比，经过两层的把关，项目问题得到了大大的改善。开发提测质量变高，测试人员效率提高了，能有更多的时间来做别的提高测试质量的事情，提高了大家的工作效率，同时测试的范围变完善了，大大减少了由于代码耦合导致的线上问题，整个开发-测试-上线的流程有了很高的效率提升，在效率提升的同时，也保证了质量。

* 1. 论文组织结构

本文以作者实习期间的工作内容为基础，论述了回放对比平台的设计与实现过程。论文分为以下七个章节：

第一章：第一章：引言。主要对项目选题背景、意义与目的、本人的工作内容以及论文的组织结构进行介绍。

第二章：系统关键技术介绍。对实现整个回放，对比及前后端交互所设计到的关键技术进行简单的介绍。

第三章：系统需求分析。从功能性需求和非功能性需求两个方面，阐述回放对比平台的需求。

第四章：系统概要设计。对项目的整体架构设计、数据库设计以及接口设计进行说明。

第五章：系统详细设计与实现。详细介绍了项目中的各个模块的设计与实现的具体方法和步骤，包括模块逻辑设计、类设计和实现效果等。

第六章：系统测试概述。简要介绍作者参与回放对比平台测试的情况。

第七章：结论。总结在设计和实现过程中的优点和不足，并针对这些不足提出进一步的改进方案。

* 1. 本章小节

本章首先阐述了项目的背景，介绍了项目对个人，对部门，对公司的重要意义。然后对本人的工作进行了简要的说明。最后，对本论文的组织结构进行了详细的介绍。

1. 相关理论和关键技术

本章对回放对比平台在开发过程中所涉及的相关理论以及所采用的技术进行详细的分析和介绍。系统分为前端和后台两个部分，前端主要采用angularjs+bootstrap框架进行系统界面的开发，后台采用PHP和Go语言进行开发。PHP使用了Yaf框架，Go语言整合了Martini、Gorilla框架，并结合自己公司的业务进行了修改，取名为Mirror框架，后端选择两种语言是因为PHP对初学者来说简单易上手，同时易扩展，而回放对比里“回放”这个过程，对系统的性能要求比较高，遂选择了性能比较好的Go语言，同时，Go语言对数据的并发处理非常简单好用，很适合回放这个步骤。数据库为MySQL数据库。

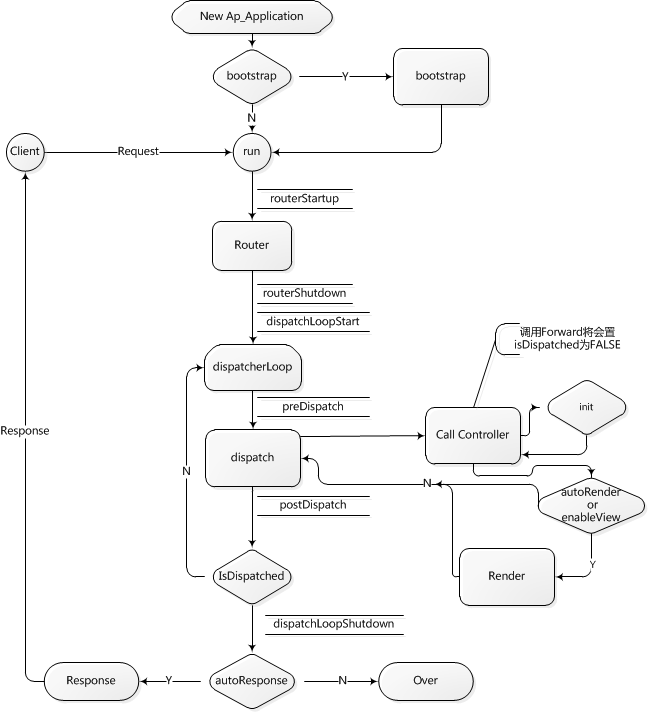
* 1. 服务端架构技术

本项目服务端开发主要使用了Yaf框架和Martini、Gorilla框架整合成的Mirror框架及MVC架构模式，使得服务端接口在开发和维护过程更加清晰，与前端的交互也更加简单容易，实现了数据，业务和处理的分离。

* + 1. Yaf框架

由于本项目服务端主要流程使用PHP语言进行开发，因此选择了PHP语言中性能和易用性比较高的Yaf框架进行开发。Yaf，全称 Yet Another Framework，是一个高性能的PHP开发框架，采用PHP扩展实现(c语言)。相比于一般的PHP框架, 它更快，更轻便. 它提供了Bootstrap, 路由, 分发, 视图, 插件, 是一个全功能的PHP框架。它有如下的几个特点：无需编译，修改之后即时生效，无效重启服务，在PHP启动的时候加载, 并常驻内存；用C语言开发的PHP框架, 相比原生的PHP, 几乎不会带来额外的性能开销；更短的内存周转周期, 提高内存利用率, 降低内存占用率；同时能灵巧的自动加载. 支持全局和局部两种加载规则, 方便类库共享；高度灵活可扩展的框架, 支持自定义视图引擎, 支持插件, 支持自定义路由等等；强大而又高度灵活的配置文件支持. 并支持缓存配置文件, 避免复杂的配置结构带来的性能损失；由Yaf作者进行的官方测试得知，Yaf框架和原生的PHP相比，在同样的功能下，性能损失少于10%。

因此，在如上所述的框架优势支持下，项目能轻易的完成与客户端的交互，开发时仅仅需要考虑模块的业务逻辑，减轻了开发的压力，节约大量的人力物力和开发时间。如图 2-3是Yaf框架运行的流程图



1. Yaf框架运行流程图
2. Yaf Farmework Running Process

在application目录下有个Bootstrap.php文件，这个就是图中的第一个环节，如果存在Bootstrap()就会先执行该文件，该文件包含了一系列的初始化环节，并返回一个Yaf\_Application对象，紧接着调用了它的run方法，run里面包含了图中所有环节，run首先是调用路由，路由的主要目的其实就是找到controllers文件，然后执行里面的init和action方法，或者找到所有actions的地址然后加载，在去执行对应的execute方法，如果设置了autoRender在返回的时候会执行render方法，就是view自动渲染，图中有六个双横线标出的环节，就是六个插件方法，用户可以自定义实现这几个方法，然后Yaf框架会在图中相应的步骤处调用对应的HOOK方法。

如图 2-2是Yaf框架的目录结构图。



1. Yaf框架目录结构图
2. Yaf框架目录结构图

<http://yaf.laruence.com/manual/yaf.sequence.html>

https://www.jianshu.com/p/130389235abc

* + 1. MVC框架

MVC框架是一个将数据处理，业务和展示分离的一个框架，全名是Model View Controller，是模型(model)－视图(view)－控制器(controller)的缩写，本项目在应用的过程中，将代码整个流程分为了几个层次：

Controller层：是前端界面与PHP逻辑交互之间的桥梁，定义和前端的交互（数据验证、数据组装等）。

api层：定义具体业务相关的逻辑代码；

Model层：定义数据的‘增删改查’；

如图是MVC架构的框架图



1. MVC框架图
2. MCV framework
   * 1. Mirror框架

Mirror是公司根据Martini和Gorilla框架整合成的一个框架。Martini是使用Go语言作为开发语言的一个强力的快速构建模块化web应用与服务的开发框架。Martini是一个专门用来处理Web相关内容的框架，Martini功能比较全面，能够支持一些常用特性，特别是它的反射功能，能够动态插入数据到操作（handler）函数中，是一个轻量级的框架；Gorilla 一个工具套件，可能不算框架。但是，它作为一个运行最大并且最长的Gi网络模块，常常被基础的Web网络应用所采用。Gorilla有很多组件，根据业务需要灵活地引入你所需要的组件。结合这两个框架，能简单轻易的搭建一个web服务器，同时保证服务器的性能。

* 1. 前端架构技术

本项目的前端使用angularjs+bootstrap框架，由于angularjs支持自动化双向数据绑定，能是整个前端的静态页面和动态操作有紧密联系，解决html构建WEB应用乏力的现象

* + 1. Angularjs框架

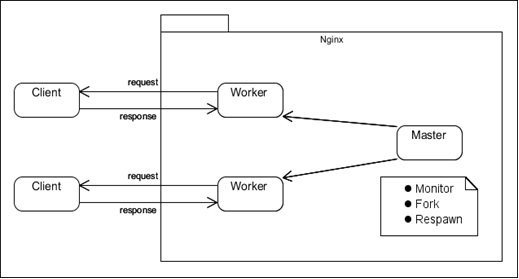
Angularjs是一款优秀的前端JS框架。AngularJS有着诸多特性，最为核心的是：MVW（Model-View-Whatever）、模块化、自动化双向数据绑定。它能使用DOM控制结构来实现迭代或者隐藏DOM片段；支持表单和表单的验证；能将逻辑代码关联到相关的DOM元素上；能将HTML分组成可重用的组件；模板功能强大丰富，并且是声明式的，自带了丰富的Angular指令；是一个比较完善的前端MV\*框架，包含模板，数据双向绑定，路由，模块化，服务，过滤器，依赖注入等所有功能；自定义Directive，比jQuery插件还灵活；ng模块化比较大胆的引入了Java的一些东西（依赖注入），能够很容易的写出可复用的代码，对于敏捷开发的团队来说非常有帮助，这样即使前端静态页面变化很大，js代码几乎不用修改。

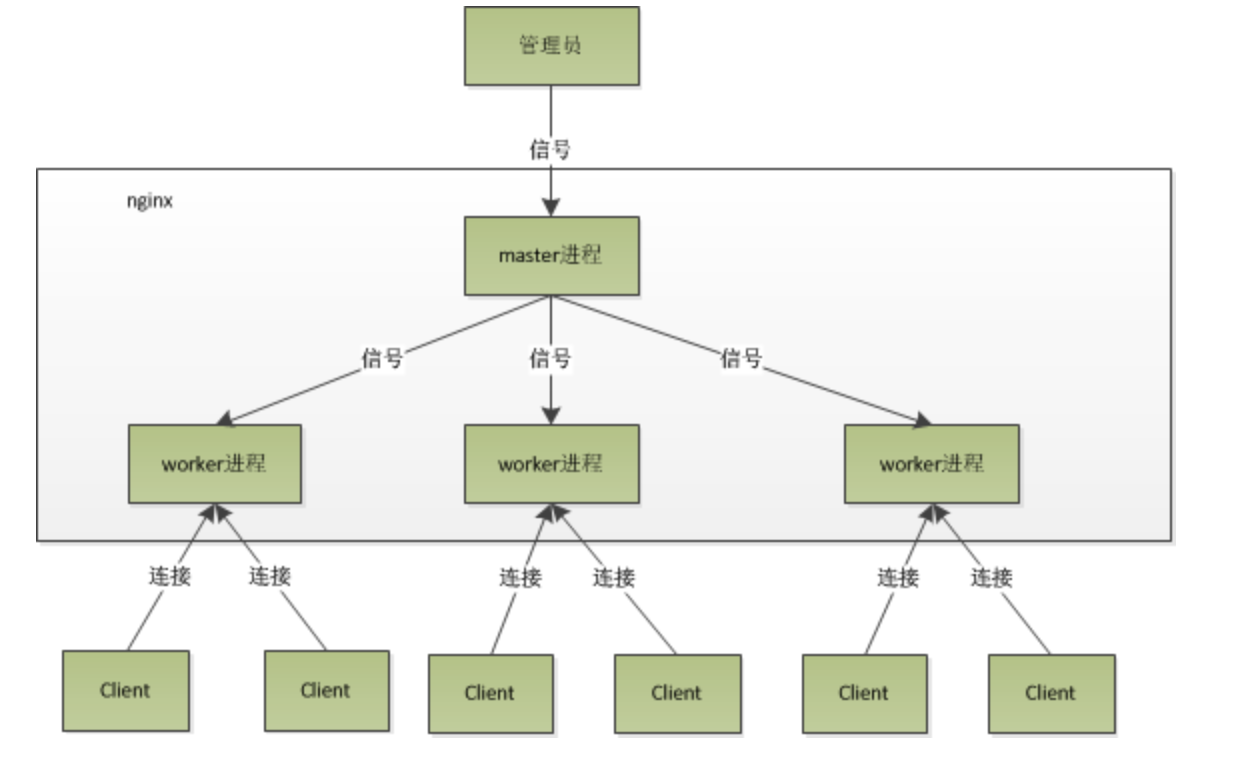
* + 1. Bootstrap框架

Bootstrap，是目前很受欢迎的前端框架。是基于 HTML、CSS、JavaScript 的一个框架，它简洁灵活，使得 Web 开发更加快捷。 它由Twitter的设计师Mark Otto和Jacob Thornton合作开发，是一个CSS/HTML框架。Bootstrap提供了优雅的HTML和CSS规范，它即是由动态CSS语言Less写成。它Bootstrap 提供了一个带有网格系统、链接样式、背景的基本结构。这将在Bootstrap 基本结构部分详细讲解。Bootstrap 自带以下特性：全局的 CSS设置、定义基本的 HTML 元素样式、可扩展的 class，以及一个先进的网格系统；它包含了十几个可重用的组件，用于创建图像、下拉菜单、导航、警告框、弹出框等等；同时，它具有很强的复用性和扩展性，使用者可以对Bootstrap中所有的CSS变量进行修改，依据自己的需求裁剪代码，能快速简单的搭建一个漂亮的web网站。

* 1. Nginx服务

nginx，也称"engine x"，它是一个HTTP的反向代理服务器，也是一个IMAP/POP3/SMTP服务器。它是一个轻量级的web反向服务器，也可以用于电子邮件，包括IMAP/POP3代理服务器，它十分的稳定，有着丰富的功能集、丰富的示例配制文件，只会消耗很低的系统资源，占用很少的内存，同时并发能力很强，表现良好，所以在大型的网站项目里十分的受欢迎。能有效的支持负载均衡。如图 2-5是nginx服务器的进程模型。





1. Nginx服务器的进程模型
2. Nginx

Nginx 做为 HTTP 服务器，有以下几项基本特性：

1. 处理静态文件，索引文件以及自动索引；打开文件描述符缓冲；
2. 无缓存的反向代理加速，简单的负载均衡和容错；
3. 简单的负载均衡和容错；
4. 模块化的结构。包括 gzipping, byte ranges, chunked responses,以及 SSI-filter 等 filter。如果由 FastCGI 或其它代理服务器处理单页中存在的多个 SSI，则这项处理可以并行运行，而不需要相互等待；
5. 支持 SSL 和 TLSSNI．
   1. Go并发任务处理

在整个回放对比过程中，回放过程需要保证对两个环境中的回放过程是同时进行的，控制住时间这个变量，所以需要存在并发处理，这是使用Go语言的重要原因。Go语言对并发编程的支持是天生的、自然的和高效的。Go语言为此专门创造出了一个关键字“go”。使用这个关键字，我们就可以很容易的使一个函数被并发的执行。Go语言这种并发模式的内核只需要协程goroutines和通道channel就够了。协程负责执行代码，通道负责在协程之间传递事件。goroutine之间通过 channel进行通信，程序会不停的从channel之间收集元素值，并进行处理，直到通道满了被关闭。这样简单的代码，就能实现高效的并发程序。

虽然我们可以使用其他语言的代码实现这样的并发，但是它们不会像Go语言这样为此提供语言级别的原生支持。也正是由于这个原因，那些代码会看起来复杂得多。它们不得不使用若干个类库或辅助工具来满足并发的要求，例如要想编写一个良好的并发程序，我们不得不了解线程，锁，semaphore，barrier甚至CPU更新高速缓存的方式。另一方面，Go语言先进、高效的并发编程模型及其实现系统会使得程序对系统资源的消耗大大减少，并且会在很大程度上提高对这些资源的使用效率。这一优势是很多其他语言望尘莫及的。

* 1. 本章小结

本章对项目开发过程中所用到的相关理论知识和关键的技术进行了详细的介绍，主要包括服务端使用的Yaf和MVC框架，基于Martini、Gorilla框架的Mirror框架，和前端使用的angularjs及bootstrap框架，前后端的通信：Nginx；以及回放的核心过程：并发操作。重点介绍了这些技术的原理、概念、特点和选择这些技术的原因。

1. 需求分析

引言（或绪论）简要说明研究工作的目的、范围、相关领域的前人工作和知识空白、理论基础和分析、研究设想、研究方法和实验设计、预期结果和意义等。应言简意赅，不要与摘要雷同，不要成为摘要的注释。一般教科书中有的知识，在引言中不必赘述。





1. （使用样式：第三章中文图题）
2. （使用样式：第三章英文图题）
3. （使用样式：第三章中文表题）
4. （使用样式：第三章英文表题）
5. （使用样式：第三章中文图题）
6. （使用样式：第三章英文图题）
7. （使用样式：第三章中文表题）
8. （使用样式：第三章英文表题）








   8. 本章小结





1. （使用样式：第四章中文图题）
2. （使用样式：第四章英文图题）
3. （使用样式：第四章中文表题）
4. （使用样式：第四章英文表题）
5. （使用样式：第四章中文图题）
6. （使用样式：第四章英文图题）
7. （使用样式：第四章中文表题）
8. （使用样式：第四章英文表题）


   2. 数据库设计

本小节将从数据库逻辑设计和数据库表结构设计两个方面来介绍回放对比平台的数据库设计，数据库的逻辑设计将阐述表和表之间的关系，数据库表结构设计将阐述表的用途和每一个表每一个字段的涵义。

* + 1. 数据库逻辑设计

本项目使用了mysql数据库，前端通过对服务端的请求对数据库进行增删改查操作，同时，存在脚本直接对数据库进行操作。在本项目中一共有10个数据表，如表 4-3为数据库中各个表的大致说明。在整个数据库设计的过程中，通过建立关系模型对数据库进行建模，如图 4-3 是本项目的实体关系图（ER图），图中只列出重要的属性。

根据ER图，对表和表之间的关系进行结构设计，可以得出如图 4-4 所示的数据库逻辑模型

1. 数据库表说明
2. Database Table Description

|  |  |
| --- | --- |
| 表名 | 说明 |
| api\_rule\_info | 每个接口具体字段的规则，包括接口规则， |
| check\_rule | 规则类型 |
| diff\_case\_info | case实例 |
| diff\_info | 请求结果及参数 |
| diff\_instance\_info | 接口字段生成历史 |
| diff\_product\_info | 产品线数据 |
| diff\_show | Diff 统计结果 (废弃) |
| diff\_task | 任务细节 |
| input\_rule\_info | 各个接口返回字段存储 |
| interface\_info | 接口回放规则 |
|  |  |
|  |  |

1. 数据库关系实体图
2. 数据库关系实体图
3. 数据库逻辑图
4. 数据库逻辑图
   * 1. 数据库物理设计

根据数据库的实体图和逻辑模型，得出数据库的物理数据模型和本系统的详细表设计，本小节将给出数据库的物理模型，给出每张表的详细设计，表内所有字段的用途，字段类型，字段类型及长度，字段要求，字段说明等。如图 4-5 是数据库的逻辑模型。

1. 数据库物理模型
2. 数据库物理图、模型
3. 验证规则表
4. rule table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段要求 | 默认值 | 描述 |
| id | int(11) | not null | 自增 | 唯一识别标志，自增 |
| product | varchar(64) | not null | 无 | 产品线 |
| url | varchar(128) | not null | 无 | 接口名称 |
| input\_rule | varchar(1000) | 无 | 无 | 输入参数规则 |
| output\_rule | varchar(1000) | 无 | 无 | 输出参数规则 |
| update\_time | timestamp | not null | 当前时间 | 规则更新时间 |
| type | varchar(30) | not null | diff | 规则类型,diff/api |
| field\_type | varchar(30) | not null | 无 | 参数类型 |
| field\_name | varchar(100) | not null | 无 | 参数名称 |
| isnew | int(11) | 0或1 | 0 | 是否是新增规则 |
| output\_condition | varchar(128) | 无 | 无 | 输出规则 |
| enable | int(11) | 0或1 | 1 | 是否可用 |

1. 验证规则表

验证规则表的表名api\_rule\_info，用于存储各个接口的不同字段的验证规则。字段id是整个表的唯一识别标志，自增id，用于区别不同的验证规则；字段product是这个验证规则属于哪个产品线，不同的产品线有不同的验证规则；url是接口的名称，表明这个规则属于哪个接口；input\_rule是指输入条件验证规则，格式如{"rule\_id":"1","fields":"name","params":"3"}，代表输入的参数name必须满足rule1，这个rule1需要一个输入值，这个值为3，例如name不等于3这样的验证规则；output\_rule只输出字段验证规则，格式同input\_rule：{"rule\_id":"2","fields":"mode","params":"3"}区别在于output\_rule验证的是接口返回的结果，input\_rule验证的是接口接收的参数；output\_condition，输出验证规则的条件，格式同input\_rule：{"rule\_id":"1","fields":"price","params":"3"}，代表的是当接口输入的字段满足规则时，验证output\_rule。当三种条件同时存在，涵义为当存在输入参数name满足规则1，且输出结果中字段price满足规则1的时候，验证输出字段mode是否满足规则2；字段update\_time代表的是这个规则被更新的时间；type字段代表的是规则的类型，有diff验证规则和api验证规则两种；field\_type字段代表的是待验证的字段是什么类型的; field\_name字段是指待验证的字段的名称；isnew字段代表的这个规则是否是最新生成的；enable字段是指这个验证规则的有效性，enable为0的时候，此规则不可用，enable为1时，此规则可用。

* 6. 本章小结



引言（或绪论）简要说明研究工作的目的、范围、相关领域的前人工作和知识空白、理论基础和分析、研究设想、研究方法和实验设计、预期结果和意义等。应言简意赅，不要与摘要雷同，不要成为摘要的注释。一般教科书中有的知识，在引言中不必赘述。





1. （使用样式：第五章中文图题）
2. （使用样式：第五章英文图题）
3. （使用样式：第五章中文表题）
4. （使用样式：第五章英文表题）
5. （使用样式：第五章中文图题）
6. （使用样式：第五章英文图题）
7. （使用样式：第五章中文表题）
8. （使用样式：第五章英文表题）



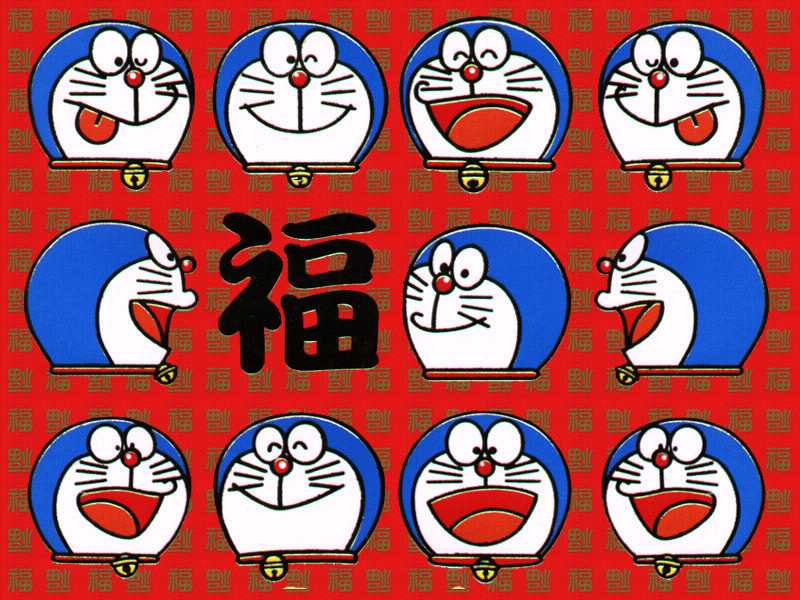





   8. 本章小结

引言（或绪论）简要说明研究工作的目的、范围、相关领域的前人工作和知识空白、理论基础和分析、研究设想、研究方法和实验设计、预期结果和意义等。应言简意赅，不要与摘要雷同，不要成为摘要的注释。一般教科书中有的知识，在引言中不必赘述。





1. （使用样式：第六章中文图题）
2. （使用样式：第六章英文图题）
3. （使用样式：第六章中文表题）
4. （使用样式：第六章英文表题）
5. （使用样式：第六章中文图题）
6. （使用样式：第六章英文图题）
7. （使用样式：第六章中文表题）
8. （使用样式：第六章英文表题）








   8. 本章小结





1. （使用样式：第七章中文图题）
2. （使用样式：第七章英文图题）
3. （使用样式：第七章中文表题）
4. （使用样式：第七章英文表题）
5. （使用样式：第七章中文图题）
6. （使用样式：第七章英文图题）
7. （使用样式：第七章中文表题）
8. （使用样式：第七章英文表题）








   8. 本章小结



引言（或绪论）简要说明研究工作的目的、范围、相关领域的前人工作和知识空白、理论基础和分析、研究设想、研究方法和实验设计、预期结果和意义等。应言简意赅，不要与摘要雷同，不要成为摘要的注释。一般教科书中有的知识，在引言中不必赘述。





1. （使用样式：第八章中文图题）
2. （使用样式：第八章英文图题）
3. （使用样式：第八章中文表题）
4. （使用样式：第八章英文表题）
5. （使用样式：第八章中文图题）
6. （使用样式：第八章英文图题）
7. （使用样式：第八章中文表题）
8. （使用样式：第八章英文表题）







   7. 111





      6. 123
   8. 本章小结
9. 结论

论文的结论是最终的、总体的结论，不是正文中各段的小结的简单重复。结论应该准确、完整、明确、精练。如果不可能导出应有的结论，也可以没有结论而进行必要的讨论。可以在结论或讨论中提出建议、研究设想、仪器设备改进意见以及尚待解决的问题等。

参考文献

参考文献是文中引用的有具体文字来源的文献集合。按照GB 7714《文后参考文献著录规则》的规定执行。

参考文献以文献在整个论文中出现的次序用[1]、[2]、[3]……形式统一排序、依次列出。

参考文献的表示格式为:

著作：[序号]作者.译者.书名[M].版本(第一版不著录).出版地:出版社,出版时间:引用部分起止页.

期刊：[序号]作者.译者.文章题目[J].期刊名,年份,卷号(期数):引用部分起止页.

会议论文集：[序号]作者.译者.文章名[C]. //编者.论文集名,会议地址，会议时间.出版地:出版者，出版年.引用部分起止页.

学位论文：[序号]作者.题名[D].保存地点:保存单位,年份.引用部分起止页.

专利：[序号]专利申请者.专利文献题名[P].国别,专利文献种类,专利号.发布日期:引用部分起止页.

技术标准：[序号]起草责任者.标准代号.标准顺序号——发布年.标准名称.出版地.出版者.出版年份:引用部分起止页.

报纸: [序号]作者.题名[N].报纸名，出版日期(版次)

附录 A

附录是作为论文主体的补充项目，并不是必须的。

论文的附录依序用大写正体英文字母A、B、C……编序号，如：附录A。

索引

按照需要编排分类索引、著者索引、关键词索引等。

作者简历及攻读硕士/博士学位期间取得的研究成果

包括教育经历、工作经历、攻读学位期间发表的论文和完成的工作等。行距16磅，段前后各为0磅。

一、作者简历

姓名：吴艳丽

性别：女

教育经历：

本科：2012.09-2016.07 北京交通大学软件学院 软件工程专业

硕士：2016.09-2018.07 北京交通大学软件学院 软件工程专业

实习经历：

2016.03-2016.07 北京寰宇惠旅科技有限公司

2017.03-2018.03 北京嘀嘀无限科技发展有限公司

二、发表论文

[1]

[2]

[3]

.

.

.

三、参与科研项目

[1]

[2]

[3]

.

.

.

四、专利

[1]

[2]

[3]

.

.

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作和取得的研究成果，除了文中特别加以标注和致谢之处外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京交通大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名： 签字日期： 年 月 日

学位论文数据集

表1.1： 数据集页

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 关键词\* | 密级\* | 中图分类号 | | UDC | 论文资助 |
|  |  |  | |  |  |
| 学位授予单位名称\* | | 学位授予单位代码\* | | 学位类别\* | 学位级别\* |
| 北京交通大学 | | 10004 | |  |  |
| 论文题名\* | | 并列题名 | | | 论文语种\* |
|  | |  | | |  |
| 作者姓名\* |  | | | 学号\* |  |
| 培养单位名称\* | | | 培养单位代码\* | 培养单位地址 | 邮编 |
| 北京交通大学 | | | 10004 | 北京市海淀区西直门外上园村3号 | 100044 |
| 工程领域\* | | | 研究方向\* | 学制\* | 学位授予年\* |
|  | | |  |  |  |
| 论文提交日期\* |  | | | | |
| 导师姓名\* |  | | | 职称\* |  |
| 评阅人 | 答辩委员会主席\* | | | 答辩委员会成员 | |
|  |  | | |  | |
| 电子版论文提交格式 文本（ ） 图像（ ） 视频（ ） 音频（ ） 多媒体（ ） 其他（ ） 推荐格式：application/msword；application/pdf | | | | | |
| 电子版论文出版（发布）者 | | | 电子版论文出版（发布）地 | | 权限声明 |
|  | | |  | |  |
| 论文总页数\* |  | | | | |
| 共33项，其中带\*为必填数据，为21项。 | | | | | |