

**本科生毕业设计（论文）开题报告**

**学生姓名： 卫亚峰**

**导师姓名、职称： 邹承明**

**所属学院： 计算机科学与技术学院**

**专业班级： 软件zy1501**

**设计（论文）题目： 基于 thrift rpc 的微服务自测平台**

2019年2月24日

**开题报告填写要求**

1．开题报告应根据教师下发的毕业设计（论文）任务书，在教师的指导下由学生独立撰写。

2．开题报告内容填写后，应及时打印提交指导教师审阅。

3．“设计的目的及意义”至少800汉字（外语至少500字），“基本内容和技术方案”至少400汉字（外语至少200字）。进度安排应尽可能详细。

4．指导教师意见：学生的调研是否充分？基本内容和技术方案是否已明确？是否已经具备开始设计（论文）的条件？能否达到预期的目标？是否同意进入设计（论文）阶段？

|  |
| --- |
| **撰写内容要求（可加页）：**   1. 目的及意义（含国内外的研究现状分析）   “微服务”架构是近期软件应用领域非常热门的概念。在这种架构风格中，一个大型复杂软件应用会被分解为多个微服务。系统中的每个微服务一般都通过一个进程实现，仅需关注于完成一件任务并很好地完成该任务。在这种架构中，一方面由于微服务之间是松耦合的，使得开发人员可以专注于某个微服务而不是整个系统，有效降低了开发成本，另外一方面由于每个微服务都体量较小，可独立部署，使得可以实现快速开发测试部署上线，从而达到随着业务变化产品快速迭代的目的。  要想合理利用微服务架构，微服务之间的有效可靠通信是个必须解决的问题。当不同的微服务位于不同的机器时，基于 TCP/IP 的网络通信是微服务之间进行有效可靠通信的唯一途径，但每一个微服务都独立实现基于 TCP 的应用层协议一是会增加开发难度，二是无法保证可靠性，因此目前的互联网公司在使用微服务架构时，都会使用一些成熟的 RPC 框架来用作微服务之间的通信机制。常用的 RPC 框架主要有 Facebook 开源的 thrift RPC 与 Google 开源的 grpc 。  Thrift RPC 是由 Facebook 开发的，后托管到 Apache 软件基金会的一个开源项目。 在实际的生产环境中，使用 Thrift RPC 作为通信机制开发完一个微服务之后，需要先进行本机自测，然后提交到 QA 进行统一测试，然后才开始逐步上线。在自测阶段，通常的解决方法是：对于每一个下游依赖服务，实现出一个服务器，并 mock 出每一个 RPC 调用的返回数据，作为一个测试用例；对于当前微服务，实现一个客户端，用来调用当前服务，并根据是否出错与返回数据对当前微服务进行测试。在自测阶段主要存在两个问题。问题一：是对于每一个微服务，都需要为所有的下游依赖服务实现一个服务器，这一方面增加了开发成本并无法保证可靠性，另一方面由于在同一个系统中两个不同的微服务有很大概率依赖同一个下游服务，所以这实际上会造成重复开发。问题二：由于自测阶段是开发人员自己 mock 出测试数据，因此很难进行测试用例的管理与维护，就会造成测试不够充分的问题，从而丧失了测试原本的意义。  为了解决自测阶段存在的上述问题，我会利用 C/S 架构实现一个基于thrift RPC 的微服务自测平台。这个测试平台要解决的核心用户需求如下：一是通过注册机制像平台注册特定服务的 thrift 服务器或客户端；二是帮助用户进行测试用例的管理与维护。     1. 研究（设计）的基本内容、目标、拟采用的技术方案及措施   **基本内容**：  1、通过阅读文献，理解 Thrift RPC 的实现机制，学习其各层协议栈的实现原理及各层协议之间的通信方式，学习 Thrift RPC 代码生成器根据 IDL 文件生成代码的过程与原理。根据以上学习研究动态解析 IDL 文件生成 Thrift 客户端或服务器并载入程序自动启动的解决方案。  2、在深刻理解需求的前提下，研究微服务维度的测试用例的维护与管理方案。包含用于展示给用户的视图模型，用于处理业务的逻辑模型，用于持久化的存储逻辑。并结合 Redis 提出可以显著提高性能的存储解决方案。  3、通过阅读文献，结合其实现原理，学习 Vue.js 与 Proxygen 的使用方法。并结合本课题的需求，实现出一个高性能的 B/S 架构应用程序。  **目标**：  1、针对待测试微服务所对应的客户端，系统会向用户提供已被当前用户注册的所有客户端列表，若列表不存在所需客户端，用户可以通过提供IDL文件的方式向系统注册客户端。  2、针对待测试微服务所以来的下游服务，系统会向用户提供已被当前用户注册的所有服务器列表，若存在某个下游服务不在所提供列表中，用户可以通过提供IDL文件的方式向系统注册服务器。  3、针对被测试微服务，用户需选定一组下游服务作为一组测试方案。用户需为每个下游服务接口提供测试数据，系统将一组测试方案中所有接口的测试数据作为一个测试用例，并进行测试用例的管理与维护。    **拟采用的技术方案及措施：**  技术方案：  1、后端技术方案：基于C++编程语言和 Linux 开发环境。Vim 作为代码编辑器，g++ 作为代码编译器，gdb 作为代码调试器，基于 makefile 的 make 工具作为代码构建工具。Proxygen  作为HTTP服务器框架。  2、前端技术方案：基于 HTML、CSS、Javascript。使用Bootstrap 样式库与 Vue.js 框架。  措施：  1、当系统接收来自用户的 IDL 文件时，系统调用脚本按需生成客户端或服务器代码，在按需修改相应生成代码后，再调用另一脚本将相应生成代码编译成动态链接库，再动态载入相应库文件后即可启动相应的服务器或客户端。  2、系统对服务器、客户端的管理不以源代码的维度持久化和维护，而以IDL文件的维度进行持久化与维护。当用户选定一个测试方案，系统会实例化测试方案包含的所有服务器与客户端。  3、针对测试用例的持久化问题，系统将采取关系型数据库 MySQL 与 非关系型数据库 Redis 相结合的存储方案。在保证可靠持久化的前提下，尽量保证高性能。   1. 进度安排   **第一阶段**（第1周—第3周）：阅读相关参考文献，完成外文资料翻译及文献摘要撰写，并交予指导教师检查。  **第二阶段**（第4周—第6周）：研究 Thrift 服务器与客户端的动态生成加载启动机制，实现系统的服务器与客户端管理维护模块。  **第三阶段**（第7周—第9周）：研究微服务维度的测试用例管理方案，实现系统的测试用例管理模块与用户管理模块。  **第四阶段**（第10周—第12周）：阅读 BootStrap、Vue.js 相关文档，完成系统需要的前端代码。  **第五阶段**（第13周—第14周）：撰写毕业论文，准备毕业答辩的有关文档及资料。   1. 阅读的参考文献不少于15篇（其中近五年外文文献不少于3篇）   [1]Slee, Mark & Agarwal, Aditya & Kwiatkowski, Marc. (2019). Thrift: Scalable cross-language services implementation.  [2]游双. Linux高性能服务器编程[M].北京：机械工业出版社，2013.06：123-145  [3]Thrift Document[EB]. Apache Thrift: <https://thrift.apache.org/docs/>, 2017年  [4]Thrift C++ Tutorial[EB]. Apache Thrift: <https://thrift.apache.org/tutorial/cpp>, 2017年  [5] N. Alshuqayran, N. Ali and R. Evans, "A Systematic Mapping Study in Microservice Architecture," *2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA)*, Macau, 2016, pp. 44-51.  [6]Developer guide[EB]. Developer Guide|Protocol Buffer: <https://developers.google.com/protocol-buffers/docs/overview>, 2018年8月23号  [7]Protocol Buffer Tutorial[EB]. Tutorial|Protol Buffer: <https://developers.google.com/protocol-buffers/docs/tutorials>, 2018年11月6日  [8] [[美] Stanley B. Lippman](https://book.douban.com/search/Stanley%20B.%20Lippman) / [[美] Josée Lajoie](https://book.douban.com/search/Jos%C3%A9e%20Lajoie) / [[美] Barbara E. Moo](https://book.douban.com/search/Barbara%20E.%20Moo). C++ Primer, 5th Edition[M].北京：电子工业出版社， 2013年9月1日  [9] [W.Richard Stevens](https://book.douban.com/search/W.Richard%20Stevens) / [Stephen A.Rago](https://book.douban.com/search/Stephen%20A.Rago). Advanced Programming in the UNIX Environment[M].北京：人民邮电出版社， 2006年  [10] W. Richard Stevens. Unix Network Programming[M].北京：清华大学出版社，2006年1月  [11]Vue.js Guide[EB]. Introduction----Vue: https://cn.vuejs.org/v2/guide/  [12] Thrift 源码[EB].apache/thrift: https://github.com/apache/thrift  [13]BootStrap Document[EB]. Introduction-Bootstrap: https://getbootstrap.com/docs/4.3/getting-started/introduction/#reboot  [14]Proxygen 源码[EB]. Facebook/Proxygen: https://github.com/facebook/proxygen  [15][侯捷](https://book.douban.com/author/104388/).STL源码剖析[M].武汉：华中科技大学出版社，2002年6月  5．指导教师意见  指导教师（签名）：  年 月 日 |