|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **南京大学软件工程硕士学位论文中期检查报告** | | | | | |
| 导师1姓名 | 张瑾玉 | 研究生姓名  （学号） | mf1332067  王颖倩 | 方向 | 应用软件工程 |
| 导师2姓名 |  |
| 论文题目 | eBay云平台PaaS Provisioning项目自动化测试框架的设计与实现 | | | | |
| 论文选题来源及研究的目的和意义（500字左右）：  论文选题来源于对公司云平台PaaS Provisioning 项目的实际测试需要。PaaS Provisioning 项目主要为eBay的云平台提供PaaS服务，其产品以RESTful API的形式发布，API的内容包括请求的路径、http 方法、http headers以及具体的报文内容。对于同一个API，由于调用它的模型的标识、模型所处的环境、模型本身的属性具有差异，造成了实际发送的请求在总体上具有相似性，在具体字段上却千差万别的情况。如果通过浏览器的RESTful 插件进行手动测试，不仅工作繁琐、工作量大、容易出错，而且效率低下，在一周的scrum里无法完成测试工作。此外，对于API中相同的部分没有进行很好的复用，测试结果也没有系统的清晰的记录，使得测试工作结果难以维护，缺陷复现、与开发人员的沟通也相当困难，这就达不到测试的目的。因此对于该项目的测试需要一个RESTful的自动化API测试框架。  为了解决上述问题，该自动化测试框架应能够根据不同的模型渲染不同的请求报文并进行发送，调用IaaS层服务，完成PaaS层服务，接收返回报文后，对模型的正确性、数据中心的物理机器及openstack的虚拟机器的配置正确性等内容进行检查，并以清晰、准确的方式展现出测试结果，同时自动保存测试日志，从而帮助完成对API的多样化、复杂、大量的自动化测试，在测试人员和开发人员合作发现缺陷、修正缺陷、提高产品质量的过程中起到重要作用。  开发和使用该自动化测试框架，大大缩短了测试的时间，提高了测试的效率，从而降低了PaaS Provisioning项目的测试成本。在提高产品质量方面，能够更快更多地发现缺陷，测试人员能够及时反馈给开发人员，而且通过清晰的测试结果沟通也能够很好地降低沟通的成本，最终使得产品的质量更高、更为健壮。 | | | | | |
| 该方向的研究现状或技术进展综述（2000字左右）  **自动化测试整体发展现状：**  由于人力成本、时间成本等非常有限，因此采用良好的自动化测试可以大幅减少在进行足够测试的前提下所需要耗费的人力和物力，也可以在有限的时间里尽可能多的进行测试。在整个测试的V-模型中，有许多工具能够帮助测试人员提高测试的自动化程度，例如测试设计工具能够帮助进行测试输入数据的设计，静态分析工具可以在不执行代码的情况下对代码进行分析，帮助发现一些漏洞，性能测试工具能够帮助模拟大量用户与系统并发交互，并监控与测试这种情况下的系统性能和负载情况等[Fewster et al.,1999]。但是如果测试自动化要取得一个良好的效果，并且真正起到提高效率降低成本的作用，需要考虑到自动化测试结果、用例的维护和管理等。  在软件开发生命周期的不同阶段进行的测试具有不同的测试目的，测试方法也有所不同。例如单元测试主要是为了保证各个方法或者模块单元的功能性的正确，以及对于异常极端情况的处理能够正确合理，而系统集成测试，主要测试各个模块的接口是否正确，整个系统是否能够正确运作，压力负载测试，则又是为了测试系统在高负荷情况下持续正常运作的能力及其容错性。测试可以在代码级别上进行白盒测试，或是在模块接口级别上进行黑盒测试等。基于这些不同的测试目的和方法，自动化测试工具的设计也有所不同。目前主流的自动化测试工具和框架有可以录制和模拟用户行为进行功能与性能测试的Winrunner、QTP、loadrunner等，还有使用与C/S结构应用和网站应用的JMeter工具，以及Java人员熟悉的Junit等。  **RESTful API测试工具：**  在目前的主流web服务实现方案中，SOAP和XML-RPC都相对复杂。以SOAP为例，在xml形式的请求消息中，会以Envelope命名空间来标识SOAP消息，并用<soapEnv>元素组织头部信息、调用和响应信息以及相关错误处理等内容。一旦请求复杂，xml消息将变得非常长且可读性低，层次结构很难分清，进行维护时也很容易引入错误。而Roy Fielding在2000年的文献[Fielding,2000]中提出的Representational State Transfer（REST）更为简洁，且应用越来越广。采用REST设计风格实现的web service中，资源由URL进行定位与标识，而对资源的不同操作，则通过HTTP的GET、POST、PUT和DELETE等方法实现，表达清晰而且易于解释。请求的头部信息和调用信息分离开来，调用信息可以采用xml格式或是更简洁的json格式。请求的执行结果，则可以通过解析响应报文及其状态码来获取。这种方法，也大大降低了维护时的成本，减少了可能发生的错误。  目前对REST web service进行测试的主流工具有以下几种：  cURL，这是一个以URL语法形式发送和传输数据的命令行工具，支持的传输协议较多，如FTP，HTTP，HTTPS等都支持，并且该工具的应用场景也很广泛，即使在路由器、打印机、音响设备等器材上也能够得到应用[Stenberg et al,2014]。熟悉命令行操作的人员使用cURL进行测试可获得灵活高效的效果，但是同样的，要真正获得这样的效果，需要一定的培训与学习成本。  SOAP UI是SmartBear公司开发的开源web service的功能/负载测试工具[SmartBear,2014]，它提供免费的版本和功能更强大的专业版本。SOAP UI可以实现SOAP形式和REST形式的测试，借助图形界面，测试者可以定制请求模板，设计测试用例和测试套装，模拟大量用户的并发访问行为，并获取返回信息来评估测试结果。对于测试者来说，SOAP UI的学习和应用比cURL更为容易和快速。  而基于一些具体的使用广泛的平台，也有相应的测试框架或是测试代码库，如Rest Assured和frisby.js。Rest Assured是为Java平台设计的REST测试与验证代码库，帮助Java开发人员对REST API进行测试，而frisby.js是一个基于node.js和Jasmine框架的REST API测试框架，可以直接使用javascript语言编写测试和预期返回结果等，非常适合开发人员快速地进行单元测试。  最为直接、容易获取的工具则是浏览器插件，如PostMan、RESTClient等，可以直接在浏览器中打开，输入URL、HTTP方法、headers和body内容等，通过浏览器直接发送请求并接受返回消息。这一类工具，学习时间极短，只要熟悉浏览器使用的人都可以很快上手。  但是以上提到的工具，在测试团队进行大量自动化REST API测试时都有一定的缺陷，例如不够灵活、用例维护困难成本高等，并且也很难直接应用于PaaS项目，需要基于项目进行定制。  **云平台自动化测试：**  随着云技术的发展，云服务逐渐在大范围内得到了推广，越来越多的公司开始利用云平台来构建自己的产品与服务。对云平台进行测试有许多异于传统软件测试的特点，例如对于高安全性的要求，对于异构数据、异构操作系统和分布式存储等的处理[Mohammad,2011]。尤其在如何取得高覆盖率这一点上，这对于云平台实现自动化测试而言是有一定的困难的，因为“云”的状态以及服务的应用实例所处的环境都是多变不确定的。为了解决这一难点，建立带状态的云模型的想法被提出，在实际测试时，可以在这些模型的基础上，灵活地模拟出不同环境和状态的云[Zhang et al.,2012]，但是其难点就在于如何正确对云进行建模，使这些模型能够最大程度抽象出各种云环境的共同特点，并且在实际生成测试时，能够方便地完成环境的模拟并得到最高的覆盖率。这种抽象出共性进行建模或者是分组的思想，在文献[Yu, 2010]中也有体现，Testing as a Service的概念被提出，对云平台的测试也可以像云本身一样，进行建模和聚簇分组。  2008年诞生的Open Cirrus为云测试提供了一个统一的平台，由惠普、因特尔、雅虎等大型企业和研究所提供支持，能够模拟真实的全球性互联网环境来进行云平台和服务的测试[Avetisyan et al.,2010]。 | | | | | |
| 论文的主要技术路线、研究思路和实现方法；相关项目应用前景：（重点说明变更部分）：  该测试框架采用直接调用API发送请求的方式进行测试，因此没有图形用户界面。框架主要应包含以下几个部分：   1. TestCase模块，即具体的测试类以及抽象出这些测试类的共同部分的父类，包含了共同的PaaS服务调用的和验证的方法及接口； 2. Validator模块，包含具体的验证方法； 3. Client模块，包括ServiceClient和ResourceClient，为调用相关服务和对资源进行操作提供了接口； 4. Framework模块，对上述三个模块提供方法支持，包含了许多操作的真正实现语句，较为偏向底层，例如模板渲染、断言、异常等内容的实现； 5. 配置模块，包含了运行必要的配置文件。   根据公司的实际情况，需要分为Stratus和GPaaS两部分，对应于eBay和PayPal两条路径，其中GPaaS部分是依赖于Stratus部分的。整体的测试框架在TestNG框架技术的基础上进行实现与扩展，TestNG是类似于junit的测试框架技术，测试用例可以单独运行也可以成套运行，可以定制运行前准备工作和运行后的整理工作，同时也可以为测试用例分组或定义用例之间的依赖关系。涉及到虚拟机器的部分，eBay云采用了Openstack的技术，因此在验证过程中，需要使用Openstack相关的技术来获取虚拟机器和簇的信息等。为了简化框架的装配及字段的初始化及管理，可在框架中引入部分spring框架的技术，如Autowired注解。  对于测试结果的打印和保存，除了使用TestNG所提供的测试结果报表生成功能以外，可采用Simple Logging Facade for Java (SLF4J) 帮助结果的输出打印，这一技术简化了日志的输出和管理，它包装了一系列日志管理的方法，为客户提供了简单的使用接口，也非常易于配置。项目的编译则采用maven技术，maven技术通过维护一个pom文件声明整个项目的结构及依赖关系，在项目编译时能够自动解析项目结构并自动根据依赖关系从在线包仓库中下载安装依赖的jar包，方便地完成整个项目的编译。  整个框架的运行入口在于TestCase模块，这部分的测试类将Client模块和Validator模块的类的引用作为自己的属性，从而可以调用Client和Validator的方法。而Framework模块中的实用方法均以静态方法的形式实现，从而在其他类的实现中，可以直接通过类名调用这些实用方法而无需实例化。  实现时四部分可以同时进行开发，只需为测试类提供设计良好的Client模块和Validator模块接口即可。在开发的过程中不断完善maven在构建项目时所必需的pom.xml文件，明确项目的结构和依赖。 | | | | | |
| 本人在相关项目中的扮演的角色和承担的工作（重点说明变更部分）：  自动化测试框架的需求分析，架构与设计，代码实现，测试，并利用该框架进行自动化测试。 | | | | | |
| 论文的主要工作（500字左右）：  在整个项目开发和论文撰写过程中，主要有以下工作：   1. 需求分析，确定项目范围和初步的项目需求，并进行文档化。   该项目的非功能需求较少，主要都集中在功能需求分析上。   1. 查找相关资料，调查类似项目的发展情况，确定使用的技术等。 2. 学习相关技术，如TestNG，maven，Openstack等。 3. 制定项目计划，细化前期的项目需求，编写测试用例文档等。   需求文档化可以帮助确定项目范围，需求文档不局限于编写用例表，用例图、复杂需求的过程图、目标用户、项目约束和依赖等都可包含在内。   1. 根据需求进行概要设计和详细设计，确定模块划分和业务流程等。在云平台中，有一些业务流程虽然不难，但是步骤繁多，细而且杂，需要仔细分析。 2. 进行实际的代码编写和系统实现，包括各模块内部细节的实现以及模块与模块之间的接口和关系的实现等。 3. 制定测试计划，编写测试用例，对模块进行单元测试、接口测试等，并对整个系统进行功能测试。 4. 实际投入PaaS Provisioning项目使用，持续后期的维护。进一步优化项目的结构，提高可扩展性和可维护性等。   论文主要分为五章，从项目的应用背景、用户背景、技术背景等入手，按照需求分析、架构和模块设计以及具体实现的顺序对项目开发过程进行说明，并在最后对工作进行总结，对未来进行展望。各章内容如下：   1. 引言部分，主要介绍项目的背景，包括项目的必要性、国内外发展情况等。 2. 技术综述，对项目中主要使用到的技术进行了介绍。 3. 测试框架的分析和设计，这一章包含了需求分析和项目设计两个部分，项目设计又包括概要设计和详细设计两部分。 4. 框架的具体实现，在设计的基础上，展示关键代码，描述具体实现。 5. 总结与展望，总结论文工作，并提出待改进之处和对未来工作的展望。 | | | | | |
| 论文三级大纲：   1. 引言    1. 项目背景    2. 国内外发展现状       1. 自动化测试整体发展现状       2. RESTful API测试工具       3. 云平台自动化测试    3. 本文主要研究的工作    4. 本文的组织结构 2. 技术综述    1. TestNG    2. REST    3. 模板化技术    4. OpenStack    5. 其他相关技术       1. Maven       2. spring    6. 本章小结 3. 自动化测试框架的分析与设计    1. 项目概述       1. 系统总体规划       2. 用户类型       3. 假设和依赖    2. 系统需求分析       1. 功能性需求       2. 流程需求       3. 非功能性需求    3. 系统总体设计       1. 总体结构       2. 系统模块及其关系设计    4. 模块详细设计       1. Client模块设计       2. TestCase模块设计       3. Framework模块设计       4. Validator模块设计    5. 本章小结 4. 自动化测试框架的实现    1. 配置初始化的实现    2. 请求渲染与发送的实现       1. RESTful请求模板文件       2. 请求渲染和发送的实现    3. TestBase类的实现    4. Validator的实现       1. Topology模型合法性验证的实现       2. 与预期结果对比验证的实现    5. 本章小结 5. 总结与展望    1. 总结    2. 进一步工作展望 | | | | | |
| 论文和相关项目的当前进度：  目前该自动化测试框架已在PaaS Provisioning项目的实际开发中得到了有效运用，但是在最初的使用过程中也暴露了许多的问题，在经过改进和优化之后，其质量已经有了很大的提高。  论文目前已完成初稿的撰写，下来将与导师沟通并进行进一步修改。 | | | | | |
| 论文和相关项目进展过程中遇到的困难和问题，以及解决的措施：   1. 需求复杂多变   **应对措施：**在进行需求分析时，分辨出主要和关键需求以及附加的需求，对需求进行系统分类，首先重点实现主要和关键的需求。  对于不明确的需求，通过与同事进行讨论和对需求进行更进一步分析，明确模糊的需求，分解过大的需求。  在设计实现时，尽量优化接口和内部的实现，使其高内聚低耦合，具有灵活性和可扩展性，减少需求变化带来的影响。   1. 技术不熟练   **应对措施：**在项目初期比较各种实现技术，确定最适合的实现方法，在进行需求分析、设计等工作的同时，研究技术的官方说明以及实际的代码例子，并进行简单的编程实践，快速地熟悉技术的大体框架并能够尝试使用。  在实现过程中碰到实际问题时，向小组其他成员求教，能够获得一些技巧和优化的实现方法。此外也可以借助搜索引擎和一些技术社区，查找相似的问题和可行的解决方案。最后，在官方的FAQ、帮助文档等当中再寻找是否有官方的解答和实现方案。   1. 模型复杂不理解   **应对措施：**借助内部的wiki查阅背景知识、模型架构和API使用等内容，也可求教于小组成员，对整个ebay云模型有一个从整体到细节，从大到小的认识和掌握。 | | | | | |
| 主要参考文献：   1. [Apache,2010]   <http://velocity.apache.org/>, The Apache Velocity Project, maintained by Apache Velocity Project, 2006-2010   1. [Apache,2014a]   <http://maven.apache.org/guides/introduction/introduction-to-the-pom.html>, Maven-Introduction to the POM, maintained by Apache Maven Project, 2002-2014   1. [Apache,2014b]   <http://maven.apache.org/maven-features.html>, Maven-Feature Summary, maintained by Apache Maven Project, 2002-2014   1. [Avetisyan et al.,2010]   A. I. Avetisyan, R. Campbell, I. Gupta, et al., Open cirrus: A global cloud computing testbed. Computer, 2010, 43(4): 35-43   1. [Beust,2004]   <http://beust.com/weblog/2004/08/25/testsetup-and-evil-static-methods/>, TestSetup and evil static methods, authored by Cedric Beust   1. [Beust,2007]   C. Beust, Next Generation Java Testing: TestNG and Advanced Concepts. Pearson Education, 2007   1. [Beust,2013]   <http://testng.org/doc/index.html>, TestNG-Welcom, maintained by Cedric Beust   1. [Costello,2007]   R. L. Costello, Building web services the rest way. URL: http://www.xfront.com/REST-Web-Services.html. Ultima Consulta, 2007, 11: 2007   1. [Fewster et al.,1999]   M. Fewster and D. Graham, Software test automation. Addison-Wesley Professional, 1999.p6-p8   1. [Fielding,2000]   R. T. Fielding, Architectural styles and the design of network-based software architectures, PhD thesis, University of California, Irvine, 2000   1. [Harris,2009]   T. Harris, Cloud Computing-An Overview, Whitepaper, Torry Harris Bussiness Solutins, 2010, 1   1. [Incki et al.,2012]   K. Incki and H. Sozer, A survey of software testing in the cloud, Software Security and Reliability Companion (SERE-C), 2012 IEEE Sixth International Conference on. IEEE, 2012   1. [McCarthy,2012]   <http://www.crn.com/news/cloud/240005076/ebay-builds-virtualized-test-network-with-nicira-openstack.htm>, eBay Builds Virtualized Test Network With Nicira, OpenStack, authored by Jack McCarthy   1. [Mohammad,2011]   A. F. Mohammad and H. Mcheick, Cloud services testing: An understanding, Procedia Computer Science, 2011, 5: 513-520   1. [OpenStack,2014]   <http://www.openstack.org/software/>, OpenStack: The Open Source Cloud Operating System, maintained by OpenStack Foundation   1. [Qian et al.,2009]   Qian L, Luo Z, Du Y, et al. Cloud computing: An overview, Cloud computing. Springer Berlin Heidelberg, 2009: 626-631   1. [Rodriguez,2008]   A. Rodriguez, Restful web services: The basics, IBM developerWorks, 2008   1. [SmartBear,2014]   <http://www.soapui.org/About-SoapUI/what-is-soapui.html>, What is soapUI? Maintained by SmartBear Software   1. [Stenberg et al,2014]   <http://curl.haxx.se/>, cURL Web Site, maintained by D. Stenberg, Dan Fandrich and Yang Tse   1. [Venkatraman,2014]   <http://www.computerweekly.com/news/2240222899/Case-study-How-eBay-uses-its-own-OpenStack-private-cloud>, Case study: How eBay uses its own OpenStack private cloud, authored by A. Venkatraman, 2014   1. [Yu, 2010]   L. Yu, W. T. Tsai, X. Chen, et al. Testing as a Service over Cloud[C], Service Oriented System Engineering (SOSE), 2010 Fifth IEEE International Symposium on. Ieee, 2010: 181-188   1. [Zhang et al.,2012]   L. Zhang, T. Xie, N. Tillmann, et al. Environment modeling for automated testing of cloud applications. IEEE Software, Special Issue on Software Engineering for Cloud Computing, 2012, 1   1. [阿里云,2014]   <http://www.aliyun.com/>, 阿里云官方网站, 阿里云计算有限公司   1. [陈康 等,2008]   陈康, 郑纬民, 云计算的三架马车: Google, 亚马逊和 IBM[J]. 计算机世界报, 2008, 17: 38-40. | | | | | |
| 导师意见： | | | | | |
| 学院备案意见：        年 月 日 | | | | | |