摘要

Abstract

目录

Table of Contents

[目录 2](#_Toc405987991)

[第一章 引言 4](#_Toc405987992)

[**1.1** **项目背景** 4](#_Toc405987993)

[**1.2** **国内外发展现状** 5](#_Toc405987994)

[**1.2.1 自动化测试整体发展现状** 5](#_Toc405987995)

[**1.2.2 RESTful API测试工具** 5](#_Toc405987996)

[**1.2.3 云平台自动化测试** 6](#_Toc405987997)

[**1.3** **论文的主要工作和组织结构** 6](#_Toc405987998)

[第二章 相关技术概述 7](#_Toc405987999)

[**2.1** **testng** 7](#_Toc405988001)

[**2.2** **maven** 7](#_Toc405988002)

[**2.3** **openstack** 7](#_Toc405988003)

[**2.4** **spring** 7](#_Toc405988004)

[**2.5** **RESTful** 7](#_Toc405988005)

[**2.6** **模板化** 7](#_Toc405988006)

[**2.7** **小结** 7](#_Toc405988007)

[第三章 需求分析与概要设计 8](#_Toc405988008)

[**3.1** **整体概述** 8](#_Toc405988010)

[**3.2** **需求分析** 8](#_Toc405988011)

[**3.3** **小结** 8](#_Toc405988012)

[第四章 详细设计与实现 9](#_Toc405988013)

[**4.1** **模块划分** 9](#_Toc405988015)

[**4.2** **概要设计** 9](#_Toc405988016)

[**4.3** **详细设计** 9](#_Toc405988017)

[**4.4** **关键性代码** 9](#_Toc405988018)

[**4.5** **测试过程截图** 9](#_Toc405988019)

[**4.6** **小结** 9](#_Toc405988020)

[第五章 总结与展望 10](#_Toc405988021)

[**5.1** **总结** 10](#_Toc405988023)

[**5.2** **展望** 10](#_Toc405988024)

[参考文献 11](#_Toc405988025)

[致谢 12](#_Toc405988026)

图目录

表目录

第一章 引言

* 1. **项目背景**

在第一台计算机诞生半个多世纪后的今天，数据量的爆炸性增长、互联网的飞速发展、人们需求的复杂多变，使得单机软件的劣势越来越明显，有限的存储和资源也越来越无法满足人们的需求。尤其是在虚拟的网络与人们的生活紧密结合的情况下，分布式的、互联的、可伸缩的云平台和云服务成为了一种必需，以提供动态的应用、数据和文件存储设施，提高了计算机和网络完成任务的能力并降低了计算、存储、传输等成本[1]。研究者们提出了许多云计算的定义，如文献[2]中给出的定义是“一种由通过IP网络连接起来的一大组低成本计算单元提供IT服务的计算技术”。可以看出，云计算有这样几点特征：大规模的计算资源、高可伸缩性、共享的虚拟和物理资源池、动态的资源分配以及通用用途。

目前许多主流的互联网公司都已经提供云计算相关的服务。国内的几大互联网公司都有自己的云服务平台，例如百度公司的百度云，其得到广泛应用的一个服务就是云盘云存储；而阿里云则更侧重于为大数据应用提供一个稳定安全的平台，主要的产品有云服务其、负载均衡SLB、开发存储服务OSS[3]等 。在国外，具有代表性的云产品提供商有amazon、google和IBM等。Amazon的产品 EC2[[1]](#footnote-1)基于服务器虚拟化技术；google的GAE[[2]](#footnote-2)则是基于特定技术的sandbox来实现的[2]，是针对google的特定应用设计的私有云；IBM的蓝云计算平台，其易于传统分布式平台的地方在于它的机器虚拟化技术以及它对Hadoop文件系统的应用[4]。

在这种背景下，作为全球最大的网络拍卖商eBay公司也开始进军这一领域，利用Nicira和OpenStack技术开发自己的共有私有混合云，即其既有为内部应用而特别设计的私有平台，也有与外部云融合的能力，以适应对大数据大存储和弹性需求的要求[5]，并且有着极大的灵活性和高性价比的特点。eBay的云平台主要有三层，从下往上分别是IaaS、PaaS和高层应用服务如Altus等。PaaS作为中间层，其主要的职责是进一步抽象下层IaaS的服务，向上完全屏蔽物理机器，提供虚拟机器创建删除、添加网络别名、虚拟机器重镜像等功能，对于云的实现和应用起着必不可少的重要作用，因此，这一层的高可靠性、健壮性、可扩展性、可伸缩性等都必须得到良好的检测、维护和保障。同时，该项目的产品以RESTful的API形式发布，在使用时，客户发送的请求之间有一定的相似性，但又有许多细节的差异，这些差异可能导致完全不同的结果，这一特点使得对PaaS产品的测试也具有特殊性。为此，一个合适的自动化测试工具或框架，对于又多又快地发现代码缺陷、提高产品质量、减少测试成本、提高效率是非常必要的。

* 1. **国内外发展现状**

**1.2.1 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[6]。但是测试自动化如果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效果，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是否正确合理，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。而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。

**1.2.2 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，而Roy Fielding在2000年的文献[7]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[10]，表达清晰而且易于解释。而请求的执行结果，则可以通过解析响应报文及其状态码来获取。

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[8]。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[9]，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。

**1.2.3 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。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[13]。同时，各种基于云平台、云服务测试的研究也在进行。

* 1. **论文的主要工作和组织结构**

总分

第二章 相关技术概述

1. 1. **testng**
   2. **maven**
   3. **openstack**
   4. **spring**
   5. **RESTful**
   6. **模板化**
   7. **小结**

第三章 需求分析与概要设计

1. 1. **整体概述**

目的

假设和依赖

* 1. **需求分析**

功能性

流程

非功能性

* 1. **小结**

第四章 详细设计与实现

1. 1. **模块划分**
   2. **概要设计**
   3. **详细设计**
   4. **关键性代码**
   5. **测试过程截图**
   6. **小结**

第五章 总结与展望

1. 1. **总结**
   2. **展望**

参考文献

[1] Torry Harris' “Cloud Computing – An Overview”, p2

[2] Qian L, Luo Z, Du Y, et al. Cloud computing: An overview[M]//Cloud computing. Springer Berlin Heidelberg, 2009: 626-631.

[3] 阿里云官方网站http://www.aliyun.com/

[4] 云计算的三架马车:Google、亚马逊和IBMhttp://www.nscc-tj.gov.cn/show.asp?id=199

[5] eBay Builds Virtualized Test Network With Nicira, OpenStack http://www.crn.com/news/cloud/240005076/ebay-builds-virtualized-test-network-with-nicira-opopensta.htm

[6] Fewster M, Graham D. Software test automation[M]. Addison-Wesley Professional, 1999.p6-p8

[7] Fielding R T. Architectural styles and the design of network-based software architectures[D]. University of California, Irvine, 2000.

[8] cURL <http://curl.haxx.se/>

[9] What is soapUI? http://www.soapui.org/About-SoapUI/what-is-soapui.html

[10] Wikipedia REST条目 <http://zh.wikipedia.org/wiki/REST>

[11] Mohammad A F, Mcheick H. Cloud services testing: An understanding[J]. Procedia Computer Science, 2011, 5: 513-520

[12] Incki K, Ari I, Sozer H. A survey of software testing in the cloud[C]//Software Security and Reliability Companion (SERE-C), 2012 IEEE Sixth International Conference on. IEEE, 2012: 18-23.

[13] Avetisyan A I, Campbell R, Gupta I, et al. Open cirrus: A global cloud computing testbed[J]. Computer, 2010, 43(4): 35-43.

致谢

1. Amazon Elastic Compute Cloud [↑](#footnote-ref-1)
2. Google App Engine [↑](#footnote-ref-2)