

文章编号: 2096-1472(2016)-08-09-05

# 基于CTI-TET和Selenium Webdriver的Web应用 自动化测试框架的设计与实现

叶 鹏<sup>1,2</sup>

(1.北京大学信息科学技术学院计算机应用技术,北京 100871;

2.甲骨文软件研究开发中心(北京)有限公司,北京 100094)

**摘要:** 基于Web应用的自动化测试随着互联网科技的高速发展已然成为了许多互联网公司不可或缺的技术领域。但价格高昂的商业测试工具以及其繁冗复杂的使用说明也为不少公司带来了不小的额外成本。本文将Solaris平台的Analytics Webui应用服务为例,介绍如何利用当前互联网上比较流行的开源软件和免费资源,开发出适合自身需求特点的Web应用自动化测试脚本。成功的Web应用服务的自动化测试脚本的开发不仅是测试团队的技术实力体现,更是对其应用服务产品本身开发的高质量监督和保证。

**关键词:** CTI/TET; Selenium; 测试框架; 开源; 自动化测试

**中图分类号:** TP311.5 **文献标识码:** A

## Design and Implementation of the Web Application Automated Testing Frame Based on CTI-TET and Selenium Webdriver

YE Peng<sup>1,2</sup>

(1.School of Electronic Engineering and Computer Science,Peking University,Computer Application Technology,Beijing 100871,China;

2.Oracle Software Research and Development Center Beijing Co.Ltd.,Beijing 100094,China)

**Abstract:**With the rapid development of internet technology,the automated testing based on web application has become the indispensable technical field in numerous internet companies.However,the expensive business testing tools and complicated instructions bring much extra cost to the companies.Taking Analytics Webui Application Server on Solaris OS platform as an example,the paper introduces how to utilize the popular open-source software and free resources to develop web application automated test scripts,which can meet their specific requirements.The successful development of automation testing scripts for the web application service products is not only the reflection of testing team's technical strength, but also the high-quality supervision and quality-assurance for the development of web application service products.

**Keywords:**CTI/TET;selenium;testing frame;open-source;automated testing

### 1 引言(Introduction)

随着信息技术的发展,互联网行业日益昌盛并在不知不觉中逐渐渗透到了各个领域。伴随着成千上万的互联网应用软件产品的开发与产生,人们对基于Web应用的软件产品的测试需求也与日俱增。

然而,由于Web服务技术规范的复杂性,应用部署的网络分布性和在线运行形态的多变瞬时性等特点,传统的软件测试技术已不能完全满足其测试工作需求。此外,Web服务提供了一种机器对机器的通讯机制,其软件测试策略主要基于服务接口进行设计与实现,与传统需要大量人工干预的测

试方法不同,难以通过手工测试方式完成,需采用自动化测试方法<sup>[1]</sup>。

因此,针对Web应用,业界涌现出了不少成熟且广为应用的自动化测试框架,一些知名公司也针对用户的普遍需求,开发出了不少优秀且功能强大的Web自动化测试工具软件,比如Parasoft公司的webking, MERCURY公司的LR和QTP, IBM公司的Rational Robot等。自动化测试不仅可以大大减少测试人员的工作量,降低工作难度,还可以避免不必要的人为疏忽与测试盲点,加快测试速度,保证产品质量和进度,提高效益<sup>[2]</sup>。

当然,不菲的价格以及缺少深层业务功能测试定制的灵活性也限制了这些商业化测试软件在更多公司的普及。不少公司(甚至是一些500强大公司),更倾向于使用开源项目的测试框架和测试平台,在降低测试工具的来源成本的同时依旧能够保持灵活的且高技术含量的测试内容及出色的测试质量。本文意在通过介绍Oracle公司的Solaris Core OS QE技术团队对某项Solaris Web应用服务(Analytics Webui)的测试,来介绍一组完全基于开源项目的自动化测试框架(CTI-TET+Selenium Webdriver)和Python语言脚本的设计及其实现。

## 2 CTI-TET概述(CTI-TET summary)

### 2.1 CTI和TET

CTI,即Common Test Infrastructure,通用测试框架;TET,即Test Environment Toolkit,也叫TETware。最早可以追溯到1989年,由OpenGroup的一群崇尚免费和自由的工程师开发并维护。后来,经过不断改进和整合,SUN公司于2005年开始使用基于CTI接口的TET,简称CTI-TET,并于2007年公布于OpenSolaris社群。它的出现,是为了取代早期的SUN公司的测试工具集Solaris Test Framework(STF)。因其丰富的接口和多语言支持的兼容性,在原SUN公司的Solairs和Solaris Cluster组织里广为使用和流传,是专门针对Unix操作系统核心组件和各种核心服务进行自动化测试的。在SUN被Oracle公司收购以后,由于Solaris业务依旧继续开展,使得这套测试框架和工具集得以继续沿用和发展。如今的Oracle Solais团队,依然在使用和维护这套测试工具集,并利用它强大的测试框架作为他们新项目的开发与测试的主流自动化测试工具。

### 2.2 特点

如今,TETware已经发展到了version 3.8,任何单位或个人,都可以免费下载获得并使用它们。当然,不少大公司为了更好地利用这套工具,愿意投入一定的金钱购买license以获得更好的技术支持和功能扩展。CTI-TET也随着原型测试工具TETware的更新得到了升级和发展。作为一款在SUN/Oracle公司使用近10年的成熟测试工具,其在业界尤其是\*unxi领域也享有很高的使用度。概括来说,其测试功能具备以下特点:

(1)支持执行脚本自定义,即可以在测试集合里定义各种非常实用的参数,已期达到不同的测试执行效果。例如:

- a.非分布式测试用例的本地化测试运行。
- b.非分布式测试用例的异地化测试运行,多节点(最多支持999)远端运行。
- c.分布式测试用例本地与异地同时运行。
- d.单一测试用例的随机选取运行。
- e.以上不同测试方式的组合并行测试执行。
- f.以上不同测试方式的自定义顺序测试执行。

(2)支持分布式测试。

(3)运行模式可以分为随机式,重复式和并行式。

(4)本质上来讲,通过TET的简单接口,来支持更多的基于CTI Library的工具集OpenGroup在TET里包含了大量的实用的免费测试工具,比如POSIX一致性测试集。

(5)在运行权限上既支持超级用户(root)也支持普通用户。

(6)在测试结果上,CTI-TET可以自动收集所有测试信息,包括journal日志、错误日志、debug日志、系统日志等。并提供基于Web格式的summary报告,通过报告里的超级链接,测试者可以很容易的定位每一个测试用例的journal日志。

## 3 Selenium概述(Selenium summary)

严格说来,Selenium是一套完整的Web应用程序测试系统。它是一种轻便的可以在多平台操作系统(Windows、Linux、Macintosh)执行的开源软件,测试用例可以使用HTML tables或者其他多种编程语言来编写,并且可以直接运行在大部分Web浏览器上<sup>[3]</sup>。它包含了测试录制(Selenium IDE)、编写及运行(Selenium Remote Control)和测试的并行处理(Selenium Grid)。早期的Selenium包括Selenium Remote Control(RC)和Selenium IDE,是2004年由ThoughtWorks公司的Jason Huggins用JavaScript语言开发的,被后人称为Selenium 1。它可以进行页面的交互操作,并且可以重复地在不同浏览器上进行各种测试操作。后来2006年Google公司的工程师Simon Stewart开启了一个叫Webdriver的项目,可以直接让测试工具调用浏览器和操作系统本身提供的内置方法,以此来绕过JavaScript的沙盒限制。2008年,Selenium和新开发的Webdriver整合到了一起,正式发布为Selenium 2(简称Webdriver)<sup>[4]</sup>。

### 3.1 Selenium IDE

Selenium IDE是一种Firefox浏览器插件,仅限于安装Firefox浏览器中,可实现网页操作步骤的录制和回放,使用

此插件可以执行简单的测试逻辑的自动化测试，可将Selenium IDE插件的脚本导出为Java、Python、C#等语言格式的程序代码，可将人为操作网页的各种动作直接转化为自动化测试的程序代码，便于编写更复杂的测试代码。

优点：小巧简单，无需编程经验可快速上手，可以使用列表方式选择操作命令。

缺点：录制脚本转化为其他语言脚本有可能出现一些错误，还需要自动化测试工程师做相应的调试和修改。学习此插件需要熟悉HTML、JavaScript和DOM相关。

### 3.2 Webdriver

#### (1)实现原理

Webdriver与Selenium 1的JavaScript注入实现不同，它直接利用浏览器的内部接口来操作浏览器。对于不同平台中的不同浏览器，必须依赖浏览器内部的Native Component(原生组件)来实现吧WebDriverAPI调用转化为浏览器内部接口的调用。它使用的是系统内部接口或函数，首先要找到这个元素的坐标位置，并在这个坐标点触发一个鼠标左键的单击操作。因此，它可以更好的模拟真实的环境，仅能测试那些可见的页面元素<sup>[4]</sup>。

#### (2)WebDriver的优点<sup>[2]</sup>：

a. 必须操作真实的浏览器，但Webdriver可以使用HtmlUnit进行测试，在不打开浏览器的情况下进行快速测试。

b. 基于浏览器的内部接口实现自动化测试，可以更接近用户的真实情况。

c. 提供了更简洁的面向对象API，提高了测试脚本的编写效率。

d. 使用过程中无需启动Selenium Server。

e. 相比较于Selenium 1，可以调用本机的键盘和鼠标事件。

#### (3)WebDriver+面向对象语言

WebDriver使Selenium用户在完成自动化测试任务中明显体会到强大的技术优势。这主要归功于它对面向对象语言的支持，比如对Java、Python语言的支持。编写代码简单、易懂，相比较用QTP录制脚本，Selenium/Webdriver脚本更易于维护，真个测试框架可以用面向对象思想进行设计，框架复用性高，不足之处就是它对测试步骤中的验证支持度不够，需要用户自己开发额外的功能代码来实现<sup>[2]</sup>。

## 4 测试需求与分析(Testing requirement and analysis)

### (1)测试目标

关于软件测试目标，人们在长期的实践中逐渐有了统一的认识，既首要目标是预防错误，如果能实现这一目标，那么就不需要修正错误和重新测试<sup>[3]</sup>。

### (2)Web服务与应用测试

Web服务(Web service)是针对因特网上分布计算提出的一种基于开放标准、松散耦合及跨平台的新型软件构件，已广泛应用于通信、金融、地理信息、电子政务、手持设备等领域，成为当前和未来网络分布式应用的主流软件开发工具<sup>[1]</sup>。

Web应用具备内容驱动、易于导航，平台无关性、分布式、动态性、交互性、美观性、即时性、持续演化性和安全性等特点，这就决定了针对Web应用的测试相比于桌面应用软件或系统服务/中间件软件来说有其自身特有的测试原则和基准。在设计测试内容的时候，根据特点可划分为：系统性能测试、浏览器兼容性测试、Web系统安全性测试，以及Web易用性测试<sup>[6]</sup>。

### (3)Solaris Analytics Web应用测试

Oracle Solaris Analytics是一个隶属于甲骨文公司即将于2017年发布的新的Oracle Solaris12原生态核心服务的系统检测和诊断工具。它是通过调用Oracle Solaris Analytics Statistics Store的接口，通过网页将系统性能检测和诊断的数据以图表的形式展现出来。即使是不具备资深Solaris技术背景的工程师，也可以轻松对系统的性能和故障进行诊断和分析。所以从结构上概括地说，它分为两个部分，后台部分就是Oracle Solaris Analytics Statistics Store(简称Sstore)，它为调用者提供数据源接口；前台就是Oracle Solaris Analytics BUI(简称Webui)，即通过对接口的调用，对操作系统的数据源以图表(Chart)的形式在Web页面里动态显示。关于接口的测试虽然也属于Web应用测试的一个范畴，但在本文的介绍范围之内。本文着重介绍基于浏览器交互操作的测试，即analytics BUI方面的测试。

该测试实际上是基于事物流的一种功能测试技术。事物由一系列操作组成，可用一个事物流表达之。事物流中的某些操作可能有系统执行，某些操作可能由用户或系统之外的设备执行，它们共同协作，完成用户的一项工作。可见，一个事物流是系统行为的一种表示方法，为功能测试建立了程

序的动作模式<sup>[5]</sup>。

### 5 测试运行机理与基本流程(Mechanism for testing execution and elementary procedure)

#### 5.1 自动化测试

软件自动化测试就是使用自动化测试工具来代替手工进行一系列测试动作，以及验证软件是否满足需求，包括测试活动的管理和实施。自动化测试主要是通过所开发的软件测试工具，脚本等来实现，其目的是减轻手工测试的工作量，已达到节约资源(包括人力、物力等)，保证软件质量，缩短测试周期，提高测试效率的目的<sup>[4]</sup>。自动化测试的优点：

- a. 程序的回归测试更方便。
- b. 可以运行更多更繁琐的测试。
- c. 执行手工测试很难或不可能进行的测试。
- d. 充分利用资源。
- e. 测试具有一致性和可重复性。
- f. 让产品更快面向市场。
- g. 增加软件信任度。

#### 5.2 CTI-TET/Webui-core

既然自动化测试具备那么多的优点，如何部署和开发一个针对本Web应用的自动化测试脚本使其可以代替人工测试来高效的完成回归测试任务，便成为了工程师们需要花精力去研究的工作。经过一段时间的探讨和研究，工程师们终于完成了基于甲骨文公司内部已有的测试框架的自动化脚本的设计与开发。这组测试脚本的Suite目前命名为webui-core。

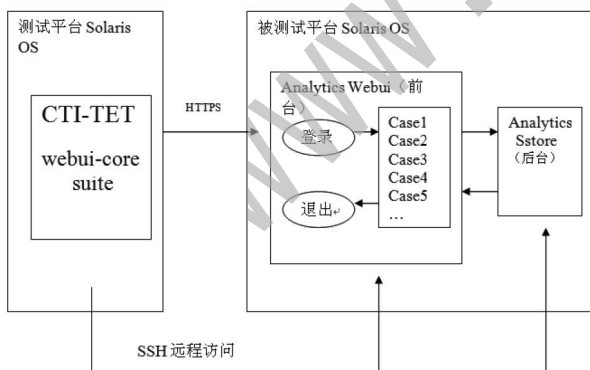


图1 Webui测试部署与运行

Fig.1 Webui testing deployment and execution

如图1所示，测试需要两台装有Solaris操作系统的机器。一台为测试机，负责运行基于CTI-TET测试框架的TEST SUITE(webui-core)；另一台为目标被测试机，装有Analytics Webui服务和Analytics Sstore服务，Webui需要根

据访问请求随时调用Sstore API以获得能动态汇报本机系统性能及状态的源数据。

测试机通过运行SUITE测试脚本，向远端的被测试主机的Analytics Webui发送Secure HTTP访问请求，通过登录验证建立session后，就可以进行一组测试了，比如點選按钮、创建/删除图表、暂停/启动动态曲线图页面等。在完成一组测试内容之后，通过點選logout按钮正常退出来终止session。

在测试运行过程中，SUITE测试机会经常以无密码SSH通道方式在后台访问被测试主机终端，进行一系列必要的后台操作工作，比如清理webui json配置文件、重启服务、通过CLI指令调用Sstore以获得进行测试结果校验对比的源数据等。

### 6 测试脚本设计与实现(Design and implementation for testing script)

#### 6.1 测试功能模块

基于Web页面的测试，要想达到Solaris软件合格的工业级质量检测和保证，需要对Analytics Webui应用有比较深入的理解。因此，在设计测试用例的时候，就需要针对其功能和业务内容，通过参阅该软件设计和文档说明，进行详实的测试需求分析。为了便于测试者快速理解设计思路，以及将来测试脚本的实现，我们将被测业务以模块的形式进行分类，每个模块可以理解为包含一组相同事务流的测试用例。

如图2所示，我们将被测业务按功能划分为八大模块。

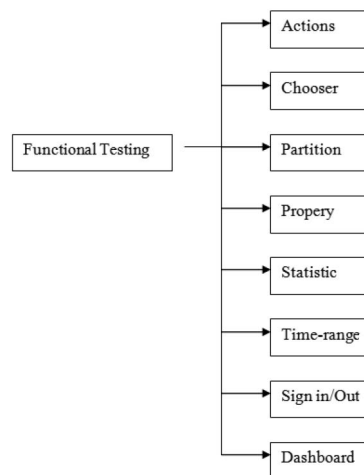


图2 Analytics Webui功能模块

Fig.2 Analytics Webui functional modules

#### 6.2 设计与实现

在完成被测模块的划分以后，接下来就可以考虑具体的软件设计和实现了。CTI-TET这套测试体系，支持多种编

程语言的实现,例如C、Java、Perl、Ksh、Python等。在综合考虑开发成本、可扩展性、可维护性后,我们决定采用Python语言作为主要的脚本实现语言。Python之所以能够在时下成为比较流行的广受大家欢迎的语言,与其语法简明,易学易用,以及对OO(面向对象)设计很好支持有很大的关系。通过对testcase类的设计,我们可以很容易的实现类继承以及方法的复用,加上Selenium的WebDriver为Python也提供了非常丰富强大的API接口,在脚本设计与实现上还是相对比较简单的。如图3所示,通过对某一事物模块的测试运行的顺序图示例,我们可以对它的测试脚本的设计特点及运行过程窥见一斑。

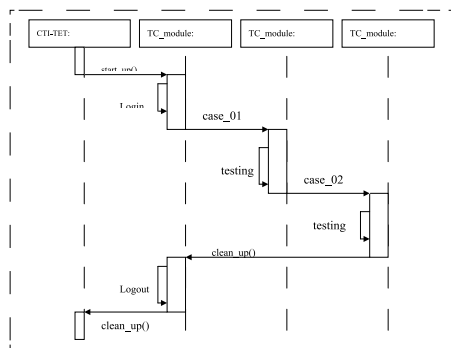


图3 单一模块测试顺序图

Fig.3 Single module testing sequence diagram

顺序图<sup>[3]</sup>是基于UML的用于建立面向对象分析的辅助模型,它是二维的,其中,垂直方向表示运行时间,水平方向代表不同的对象或参与者。在这个顺序图里,对象分别指run\_test、tc\_module、tp\_case\_01、tp\_case\_02。由于是单进程运行,所以我们可以看到这四个运行对象在时间上彼此衔接,但是不存在运行重叠(即不存在多线程/进程并行执行的情况)。

当然,图3介绍的只是suite中的通用运行环节,要完全实现测试的脚本自动化,还需要在并行测试、configuration/unconfiguration、x86/i386平台划分、3rdtools需求规划等方面进行考虑。此外,如何合理利用CTI的日志报告的接口函数cti\_report(),以及python自带的logging module来定义测试结果报告格式,以及在后期如何设计DEBUG接口以期高效快速的进行故障分析和维护,都需要进行合理的规划与设计。

## 7 结论(Conclusion)

自动化测试跟早期的传统的软件手工化测试在测试理念和测试方式上都有很大的不同。虽然它由于缺少测试灵活性而无法完全替代后者,并且也很不适合在产品开发初期就被

过早地使用,但在需要大量重复性工作的回归测试需求中,以及一些无法通过手动操作来进行的测试(比如性能测试、极限应力测试等),它就变得非常的不可或缺和重要。综合考虑时间成本和测试质量,它是产品开发后期及产品更新与维护过程中的最佳测试选择方式。

而基于Web页面的自动化测试由于其自身的特点,相对于CLI(交互式命令行)测试在自动化测试设计与实现上,由于涉及较广的技术领域,在部署和环境搭建上也相对比较繁冗复杂,因此存在一定的技术难度(尤其是在非Windows系统环境下),这为很多在该领域进行测试与测试脚本开发的工程师带来了不少挑战,无形中也为该领域Web相关产品的开发进度和质量监测带来了一定的影响。对于一个长期从事\*unix系统服务,以及中间件相关软件测试的团队(Solaris Core OS QE Team),测试工程师们需要开阔技术视野和思路,不被固有的设计理念和模式禁锢,应该学会利用已有的开源资源(CTI-TET、Python/Ksh、Selenium/Webdriver)和测试环境(Solaris OS/Desktop/Firefox),并发挥自身的硬件测试环境优势(强大的SPARC硬件性能),创造性的设计新的测试框架并付诸实现。这样才能丰富和完善团队的测试领域和测试技术水平,同时也能够更好的完成产品的质量评估与检测,协助公司开发部门顺利完成新产品的研发工作。

## 参考文献(References)

- [1] 马春燕,朱怡安,陆伟.Web服务自动化测试技术[J].计算机科学,2012,2(2):162.
- [2] 卢晨.基于Selenium进行Web应用测试研究[J].软件导刊,2015,14(1):154-155.
- [3] Christof Ebert.Web Application Tests with Selenium[J].IEEE Software.September/October 2009:88.
- [4] 吴晓华.Selenium WebDriver实战宝典[M].北京:电子工业出版社,2015.
- [5] 王立福,孙艳春,刘学洋.软件工程(第3版)[M].北京:北京大学出版社,2009.
- [6] 兰景英,王永恒.Web应用程序测试[M].北京:清华大学出版社,2015.

## 作者简介:

叶 鹏(1976-),男,硕士,高级工程师.研究领域:Linux分布式集群文件系统测试,Solaris操作系统install/analytics测试.