

# Web 服务测试的研究

李 乔, 郑 啸, 秦 锋

(安徽工业大学 计算机学院, 安徽 马鞍山 243002)

**摘 要:** Web 服务是一种全新的分布式计算技术, 它具有真正意义上的系统平台异构性和语言的独立性。随着 Web 服务技术的不断发展和广泛应用, 需要运用测试技术来保障 Web 服务的正确有效运行。然而由于 Web 服务采用了新的体系结构和核心协议, 其测试方法有别于以往的传统软件测试或网络协议测试, 所以有必要对其测试方法和技术进行研究。文中对 Web 服务的测试进行了分析, 针对不同的测试目的提出了对 Web 服务进行测试的方法, 并给出了一种 Web 服务的测试执行框架。

**关键词:** Web 服务; 概念论证测试; 功能性测试; 回归测试; 压力测试

**中图分类号:** TP311.5

**文献标识码:** A

**文章编号:** 1673-629X(2006)09-0093-04

## The Research of Web Services Testing

LI Qiao, ZHENG Xiao, QIN Feng

(School of Computer Science, Anhui Univ. of Tech., Ma'an Shan 243002, China)

**Abstract:** Web services (WS) is a new distributed computing technology. The technology allows applications to be run in different platform and programming language - independent manner. With the development of technology, WS is applied extensively and new testing technology is needed to make WS highly reliable and run effectively. However because the WS adopted new system structure and the core protocol, WS testing is different from in the former traditional software testing or the network protocol testing, so it is necessary to study its method and technology. The paper analyzed WS testing and classified testing technologies on WS according to different purposes. Furthermore, it showed a kind of performance process of WS testing.

**Key words:** Web services; proof of concept testing; functional testing; regression testing; stress testing

### 0 引言

随着网络技术的发展, 网络应用正朝着 SOA(Service Objected Architecture) 软件体系结构方向转变。Web 服务实现了“基于 Web 无缝集成”的目标, 它是一种可透过网络存取、由多个应用程序组件组合所构筑的交互使用的环境。它描述了一种新出现的、重要的分布式计算范式, 是利用 Internet 协议(如 HTTP 和 XML) 实现远程调用的应用程序组件。Web 服务以其跨平台的可互操作性, 得到了广泛的应用。新应用技术出现的同时, 也需要新的测试技术。测试技术能为服务软件的性能、质量、可靠性等方面提供有力的支持。

Web 服务处于分布式计算的核心位置, 对其进行测试很困难。主要有以下一些原因:

(1) Web 服务是一种松散连接的结构, 但是仍需要很高的保证。Internet 上的任何人都可能使用它, 必须要考虑额外的可测量性和安全性。

(2) Web 服务没有固定的展示可供用户测试的接口, 虽然一些开发工具(Studio. Net)会构建一个关于 Web 服务的页面, 但是这并不是 Web 服务本身的一部分。在这种情况下, 手动测试是很困难的, 理想的是自动测试。

(3) 由于在运行时间中涉及包括中间构件和其他 Web 服务多个参与者的发现、绑定等, 因此 Web 服务是一种动态的运行时间行为。

(4) Web 服务的运算可能要涉及到并发线程和对象共享, 并且对并发进程的测试是困难的。

(5) Internet 协议如 WSDL 和 UDDI 在设计时主要考虑功能性的特征而很少考虑验证方面的问题。

因此对 Web 服务程序模型来说, 如何有效测试 Web 服务很重要。

### 1 Web 服务核心技术及体系结构

#### 1.1 Web 的核心技术

XML 是 Web 服务平台中表示数据的基本格式。SOAP<sup>[1]</sup>是调用 Web 服务的协议, 提供了应用程序和 Web 服务之间的通信手段。WSDL 是描述 Web 服务的格式, Web 服务采用 WSDL<sup>[2]</sup>语言来描述, WSDL 采用 XML 语法, 可将其看成 XML 的一个子集。UDDI<sup>[3]</sup>是 Web 服务

收稿日期: 2006-01-09

基金项目: 安徽省教育厅自然科学研究项目(2005KJ070)

作者简介: 李 乔(1980-), 男, 安徽马鞍山人, 硕士研究生, 研究方向为 Web 服务的测试; 秦 锋, 教授, 研究方向为计算机网络。

登记、查找和利用的组合。它提供了一个开放平台独立的技术框架,来使企业之间能在互联网上找到对方的服务,定义它们在 Internet 上的交互活动,以及这些信息的共享方式。

1.2 Web 服务体系结构

Web 服务采用了面向服务(SOA)的体系结构<sup>[4]</sup>,通过服务提供者、请求者和注册中心实体之间的交互实现服务调用。参与者之间的 3 种操作都采用 SOAP 完成。如图 1 所示,Web 服务采用的体系结构中包含 3 种角色。

- (1) 服务提供者。按一定规则使用的应用程序,将其描述信息和访问规则发布到服务注册库。
- (2) 服务请求者。需要特定功能的企业或组织,从体系结构上看是查找和调用服务的客户端应用程序。
- (3) 服务注册库。存储服务描述信息的信息库,服务提供方在此发布他们的服务,服务请求方在此查找服务,获取服务的绑定信息。

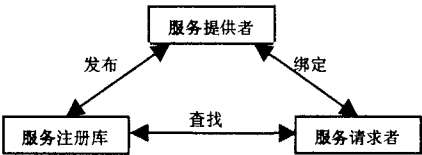


图 1 Web Services 体系结构

2 传统软件测试与 Web 服务测试

从表 1 中可见,传统软件测试经常是静态、离线的测试,尽管这些传统的测试技术也可以应用到 Web 服务的测试当中,然而相当有意义的一部分 Web 服务测试都必须是动态的和在运行时间内的即时的测试。而 Web 服务这种动态和及时的测试对传统的测试理论提出了挑战和新要求。

表 1 传统软件测试和 Web 服务测试

验证和确认	传统软件测试	面向对象的 Web 服务测试
测试人员	测试验证小组独立于开发小组以确保客观和完整性。测试仍然由软件提供者来进行	测试验证由 Web 服务的提供者、使用客户和独立的服务中间者共同协作完成。它的重点是即时测试,在测试当中,对其评估动态地使用数据
测试位置	集中的多态测试	分布的、远程的、多个代理和多态的测试
操作性测试	离线领域的测试	在线的即时测试
回归测试	离线的回归测试	在线的动态地使用收集数据的测试
完整性测试	静态单元和完整性的测试	动态配置和完整性测试
测试覆盖	结构的(白盒)或功能性(黑盒)覆盖	服务提供者拥有传统的覆盖范围,而服务中间者和客户端使用者可能只有黑盒覆盖范围
测试脚本集合	静态维护的集合	动态扩展的集合

3 Web 服务的测试

3.1 概念论证测试

在开发 Web 服务的初期,有许多种不同的选择来构建一个 Web 服务,到底使用哪种开发工具,使用哪种语言,以及使用哪种数据库作为后台在一开始就要考虑清楚。例如,必须了解所选择的 Web 服务的这种框架是否

正确。如果能够在开发周期的早期解决这些问题,那么在以后的整个开发中将节约大量的时间和资金。概念论证测试的目标是回答像“你朝着正确的方向开发之类的问题。”在某种程度上,概念论证测试是一种通常的缩减的负载测试。

3.2 功能性测试

功能性测试<sup>[5]</sup>是一个宽广的测试范畴。它包括一系列的测试方法,常见的面向任务的功能测试(TOFT)、边界测试、强制性错误测试(FET)、探测性(exploratory)测试技术等也可用于 Web 服务测试。各自的测试功能如下:

- (1) TOFT:使用的 Web 服务都能像期望的那样做一些有用的工作吗?它是积极的测试,通过将已执行任务的结果同需求文档进行比较来验证 Web 服务的功能。
- (2) 边界测试:在输入数字的应用边界会发生什么?例如当输入一个不符合规范要求的字符的时候,会有什么样的响应。
- (3) FET:当一个出错条件发生时会出现什么样的事情?其目的就是找到未检测到的和/或错误处理了的任何出错条件。
- (4) 探测性测试:在应用中,关于应用中的潜在问题域,经验会告诉我们什么?检测其行为,以及假设它会有哪些行为过程。

当然功能测试中还包括 Web 服务是否执行安全和授权的,是否 Web 服务支持所有期望的通讯协议,以及 Web 服务怎样处理不期望客户端的请求。在这个过程中,可以检查服务是否能够正确执行它的各个功能,对产品源代码了解有限的测试者进行测试以确认产品或服务的核心功能。如果这些测试失败,通常就意味着检测到了产品的一个基本问题(这个问题通常可以直接修复)。

3.3 回归测试

一个回归测试通常的目标是确保 Web 服务在已发布的版本与后来升级的版本之间仍然能运行正常,即测试计算机程序的改变,以确保旧的程序仍然能够随着新的改变运行正常。回归测试是一种程序开发处理过程中的通常的一部分。在一个新的软件产品发布之前,老的测试例被运行在新的软件版本中,以便保证所有旧的性能仍然运行良好。如果这些旧的功能出现问题 and 错误,往往是加入了新的代码或是对原有代码的改变导致不期望的代码错误。很显然,在某种程度上,回归测试可以被认为是一种缩减的功能测试。实际上这也就是为什么有些关于功能测试和回归测试的产品通常被融合为单一的程序了。由于回归测试本质上是一种重复的测试任务,因此它通常是自动进行的测试。

3.4 压力测试

压力测试不但可以测试服务对并发请求的响应,而且可以确认测试压力是否导致产生功能上的问题。对 Web 服务进行压力测试时,目的是要弄清楚被测试的 Web 服务在被施加了某些高强度压力的情况下是否仍然继续正

常运行。为了解随着越来越多的用户使用 Web 服务的响应情况,在压力测试的过程中需要考虑一些综合因素,这对得到比较客观和真实的结果是非常有用的。

(1) 重复。不断执行某个 Web 服务功能。虽然功能性测试可以用来验证一个操作能否正常工作,但并不能保证每次执行都正常运行。压力测试则弥补了功能测试的不足之处,对于某个 Web 服务的功能进行多次重复性的测试以确保其正确性,重复就可以发现许多隐蔽的代码错误。

(2) 并发。很多软件具有某个并发表为或多线程行为,通过执行多个代码示例能够看出这一点。Web 服务常会访问多线程的共享数据,由此可能会因并发而带来许多事先没有预料到的错误。所以引入并发性会使得一个线程中的代码被其他线程中的代码中断而引发错误。

(3) 随机性。压力测试过程中总会有一些错误,在允许的范围内测试时间越长,就可能会发现多一些的错误,这样就会多遍历一些代码路径。通常由于某一代码的错误会在运行一段时间后出现,所以系统必须保证当出现错误时所有可用的调试信息都被生成,否则可能就必须花费同样多的时间来重现这个错误。

压力测试的最终目标是确定如果 Web 服务的响应可以接受的话,它究竟能承受多少客户端每秒多少次的请求要求。为了执行压力测试,通常是模拟真实的用户和真正的活动来运行,执行性能测试以确定在 Web 服务响应失败以前的最大负载能力。性能测试揭示了 Web 服务的最终限制,同时,也可执行可测量性测试,以确定随着不断增加的负载,Web 服务是多么有效地适应扩展。例如图 2 的一个例子,Web 服务 1 运行良好,直到有 120 请求/秒的时候,Web 服务 1 性能就开始下降。随着请求的增加,Web 服务 1 性能持续下降,响应时间不断增加。而对于 Web 服务 2,从开始请求的不断加,它的性能就一直在下降。显然,Web 服务 2 的服务质量是比较差的。

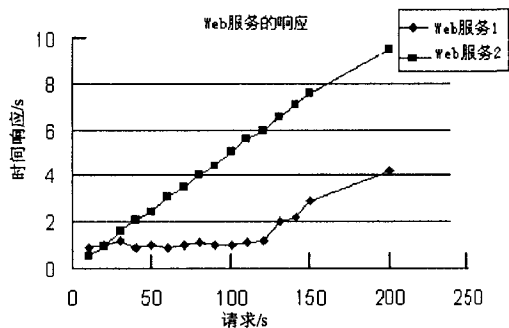


图2 不同 Web 服务的时间响应

### 3.5 监测

一旦 Web 服务开始被真正的客户使用,就很有必要监测 Web 服务。尽管进行了以上的各种测试,如果 Web 服务系统放到一个真实的环境中被成千上万的用户使用时,没有预见的问题仍然可能出现。由于这些问题,一些公司执行早期的警告系统来不断地监控关键的事务和商

务功能。由软件商提供的监测系统是一种压力测试工具的补充扩展。它允许分析者和系统管理员设置事务使用和响应时间可以接受的阈值。虽然这种监测工具的使用被证明是有益的,但是在运行环境内必须平衡已有的资源限制,以便确定监测工具的使用不会影响到响应时间。

## 4 Web 服务的测试执行框架

下面以一种 Web 服务的执行框架为例,说明 Web 服务的测试执行,它由 3 部分组成(如图 3 所示):测试主管(Test Master)、测试代理(Test Agents)和测试监视器(Test Monitor)。这 3 部分可以同处一地,但是经常是分布式的,位于不同的地方,并且它们彼此之间使用 SOAP 进行通讯。

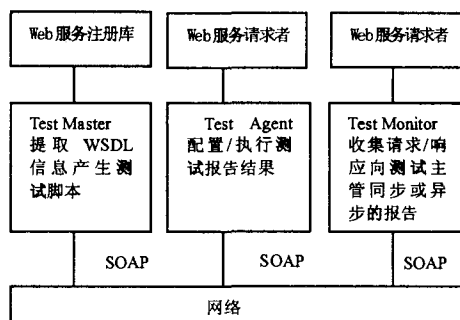


图3 Web 服务测试的执行

(1) 测试监视器经常驻留在与被测的 Web 服务同样的地方。测试监视器捕获在 Web 服务和测试代理客户端的信息,并且记录状态的变化。

(2) 测试主管通过从数据库中创建案例管理测试,并且通过使用 SOAP 并行地发送命令到一个或更多测试代理去运行。它也同步或异步地接受来自于一个或更多测试监视器的数据。

(3) 测试代理担当在 Web 服务方面测试主管的代理,它执行测试,验证结果,并且报告响应结果给测试主管。如果一个测试代理和被测的 Web 服务在同一位置的时候,它也能担当一个测试监视器的角色。

## 5 Web 服务测试发展及相关工作

在过去的几年中,Web 服务使用的目的是为了降低企业防火墙内部的融合开销,测试主要集中在测试 SOAP 消息、测试 WSDL 文件,并使用 SOAP 消息和 WSDL 文件产生测试计划,最终用来模拟 Web 服务提供者 and 使用者进行测试。随着 Web 服务技术的不断成熟,解决了诸如安全、管理、事务等方面的问题,不同的公司以不受限制的、松散连接的方式互相通讯。所以此时更多的是测试 SOA 结构的发布、发现、绑定的能力,测试 Web 服务的异步能力,测试 SOAP 消息中介能力,以及 Web 服务质量的监测。在目前 Web 服务测试的研究领域中,有的对 WSDL 文件进行了扩展,例如文献[6]从输入和输出的依赖关系、调用顺序、层次功能描述和并发顺序规范四个方面进行了扩展。有的则提出了一个支持测试执行和测试脚本

管理的测试框架,如文献[7]。这些都满足了软件易测试性分析技术的要求,也将为改进和提高软件测试的过程乃至软件开发的过程提供帮助。此外,测试数据的生成也是 Web 服务测试的重要内容,文献[8]实现了使用 TTCN-3 描述的测试数据的自动生成,文献[9]则是基于合约变异的技术,实现一种有效测试数据的自动生成方法。文献[10]对 Web 服务测试报告的管理方法进行了研究,提出了一种实现方法和框架。

## 6 结束语

由于 Web 服务是一种包含大量运行行为的分布式应用,因而不可能完全沿用传统软件测试技术对其进行测试。文中基于 Web 服务测试的一些新特点,进行了一些比较和归纳,并给出了一种测试的框架,说明整个测试的过程。

目前,随着 SOA 的不断成熟,Web 服务处理自动化工具变得流行起来,Web 服务在运行的时候能够被动态描述。以前 Web 服务使用者只能动态地发现服务并使用它们,但是现在这些服务在被使用时候可能时刻都在变化。一个服务提供者只提供其他使用者一些类似产品目录的服务,这种 Web 服务实际上是大量的、动态的其他 Web 服务的集合。这时需要一种标准语言来描述不同的 Web 服务是如何集成在一起的,即描述 Web 服务流的语言。当前有两种常见的 Web 服务流描述语言,即 WSFL 和 XLANG。如果让用户自己来描述 Web 服务流,很可能会导致错误。在发现错误之前,流中的许多 Web 服务已经被调用了,其后果会导致事务回滚困难、引起网络拥塞,从而造成资源浪费。文献[11]和[12]对 Web 服务流进行了研究。因此在以后的研究工作中,更多研究将是 Web 服务组的测试、Web 服务版本测试。

## 参考文献:

- [1] W3C note. SOAP(Simple object access protocol)[EB/OL].

(上接第 92 页)

```
TCAG_VAR.PACK.global_fit = f1 + f2 + f3; \ 将适值评价函数返回给遗传算法包
end TEST_Unit;
```

通过以上的实例可以看出,运用“分支函数正相叠加法”生成的适值评价函数既能反映测试数据的优劣,又能引导遗传算法最终找到目标参数值。从而对于寻找覆盖测试路径的最优值起到极大的促进作用,也使得面向路径测试数据的生成更加方便、可靠。

## 3 结论

着重分析了面向路径软件测试中测试数据生成问题,遗传算法凭借其优秀的强壮性和高效性,在解决这一问题上体现了其独具的优势。从整个系统模型的构建到最后

http://www.w3.org/TR/SOAP/,2003-06.

- [2] W3C note. WSDL (Web Service Description language)[EB/OL].http://www.w3.org/TR/wsdl.html,2001-03.
- [3] OASIS. Universal Description, Discovery, and Integration[EB/OL].http://www.uddi.org,2005-02.
- [4] Sprott D, Wilkes L. Understanding Service-Oriented Architecture[EB/OL].http://msdn.microsoft.com/architecture/soa/default.aspx,2004-02.
- [5] Nguyen H Q, Johnson B, Hackett M. Web 应用测试: test planning for mobile and internet-based systems[M].北京:电子工业出版社,2005.183-191.
- [6] Tsai W T, Paul R, Wang Y, et al. Extending WSDL to Facilitate Web Services Testing[A]. Proceedings of the 7th IEEE International Symposium on High Assurance Systems Engineering[C]. Tokyo:[s. n.],2002.171-172.
- [7] Tsai W T, Paul R, Song W, et al. Coyote: An XML-based Framework for Web Services Testing[A]. Proceedings of the 7th International Symposium on High Assurance Systems Engineering[C]. Tokyo:[s. n.],2002.173-174.
- [8] Schieferdecker I, Stepien B. Automated Testing of XML/SOAP based Web Services[A]. Proceedings of the GIFachtagung "Kommunikation in Verteilten Systemen", KIVS[C]. Leipzig, Germany:[s. n.],2003.43-54.
- [9] 姜 瑛,辛国茂,单锦辉,等.一种 Web 服务的测试数据的自动生成方法[J].计算机学报,2005,28(4):568-577.
- [10] Luo ling, BAI Xiaoying. Web services-Based Test Report Generation[J]. Tsinghua science and technology,2005,10(3):282-287.
- [11] Nakajima S. On Verifying Web Service Flows[A]. Proceedings of the Symposium on Applications and the Internet Workshops[C]. Nara:[s. n.],2002.223-224.
- [12] Nakajima S. Verification of Web Service Flows with Model2 Checking Techniques[A]. Proceedings of the 1st International Symposium on Cyber Worlds[C]. Tokyo:[s. n.],2002.378-385.

算法的应用都很好证明了该方法的实用性和有效性。

## 参考文献:

- [1] 郑人杰. 计算机软件测试技术[M].北京:清华大学出版社,1992.
- [2] 英 伟,高仲仪.基于遗传算法的软件结构测试数据生成技术研究[J].北京航空航天大学学报,1997,23(1):36-40.
- [3] 傅 博.基于模拟退火遗传算法的软件测试数据自动生成[J].计算机工程与应用,2005(12):81-83.
- [4] Korel B. Automated software test data generator[J]. IEEE Trans on Software Eng,1990,16(8):870-879.
- [5] Huang J C. Detection of Data Flow Anomaly Through Program Instrumentation[J]. TSE,1979,5(3):226-236.