一种多纬度服务质量评估技术

张燕生[1] 白晓颖[2] 黄冠利[3] 严 群[4]

(中国科学院研究生院工程教育学院 北京 100049)[1] (清华大学计算机科学技术系 北京 100084)[2]

(北京电子科技职业学院 北京 100029)[3] (中国煤炭地质总局信息中心 涿州 100029)[4]

摘 要 面向服务体系架构因其基于标准、松散耦合等特点给整个IT业带来巨大的变革,然而由于缺少合理有效的服务质量评估机制制约了其进一步的发展。根据贝叶斯加权均值的思想,提出了一种面向服务的多维度评估技术——在"测试中介"模型的基础上,利用协同测试共享数据,结合用户信誉度和测试用例信誉度权值对服务的测试通过率进行加权评估,为参与服务的各方提供服务排行参考。设计思想充分地体现了集体智慧和数据为核心的Web2.0的技术特点。

关键词 面向服务体系架构,测试中介,协同测试,Web2.0,质量评估

Multi-dimensional Evaluation Technique of Software Services

ZHANG Yan-sheng[1] BAI Xiao-ying[2] HUANG Guan-li[3] YAN Qun[4]

(College of Engineering, Graduate University of Chinese Academy of Sciences, Beijing 100049, China)[1]

(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)[2]

(Beijing Vocational College of Electronic Science, Beijing 100029, China)[3]

(Information Center of Coal Resource, China National Administration of Coal Geology, Zhuozhou 100029, China)[4]

Abstract Service-Oriented Architecture (SOA) is characterized by standard-based interfacing and loose coupling and has been the major paradigm for future software development. Service evaluation is critical to service selection and composition. The paper proposed a multi-dimensional evaluation technique based on testing results, tester credit and test case credit using the idea of weighted average of Bayesian method. The work is an extension of our previous work of test broker architecture which can support online collaborative testing and data collection. The technique presented is an application of Web 2.0 principle for testing with collective wisdom.

Keywords Services-oriented architecture, Testing agent, Collaborative test, Web 2.0, Evaluating quality

1 引言

Web服务作为构建面向服务架构(Service-Oriented Architecture,SOA)的主流分布式计算技术,相应的技术标准不断地推陈出新,这些都为下一代的软件技术发展带来了无限的机遇和挑战。一方面,人们从面向服务架构的Web服务应用中获取它的开放性、灵活性、可重用性和松散耦合等特点,从而得到巨大的好处;另一方面,也充分体会到了制约其进一步发展的瓶颈问题——即服务用户很难利用传统的UDDI模式在众多相同定义的服务中找到真正符合实际需要的优质服务。在传统的UDDI模式下,服务所有属性均由服务提供方利用WDSL所定义,出于商业或其他目的,服务提供方可能会过分地夸大其质量属性以吸引和迎合用户。因此,如何建立一个公正可信的、有效的服务质量评估机制显得尤为重要。

Web服务像传统软件一样需要经过严格有效的测试,并且这种测试应该是建立在参与服务的多方协同合作的基础上,即采用协同服务测试[1-3]。而传统上主要以可靠性因素度量软件质量的单维度评估方法,已难以适应SOA局部自治、自主协同、动态演化的新特点。多维度服务评估应运而生,其研究正处于起步阶段,需要进一步深入探讨其技术背景、数学模型和评估过程,从而建立一个真正有效的服务评估模型。

2 研究背景

虽然多维度的服务评估在软件工程学科内还少有涉及,但我们可以借鉴其它学科的已有成果。一般在提到多维度评估时,往往暗含着要对参与评估的多种因素进行权衡和折衷,以使系统整体最优,这属于系统工程中效能评估方法的研究范畴。比如,Levis的SEA(System Effectiveness Analysis)方法[4]就曾被应用于某些关键任务的大规模分布式系统的效能评估。这与我们所面对的服务评估问题有许多类似的地方。

在Web服务测试评估时,一方面要综合考虑服务的功能、性能、可用性、可维护性以及连接性和成本等多种与服务

到稿日期:2008-05-13 本文受国家自然科学基金项目(编号:60603035),国家863项目(项目号:2006AA01Z157),北京市自然基金项目(项目号:4072014)资助。

张燕生 高级工程师,硕士研究生,研究方向为软件测试,E-mail:yqzys@163.com;**白晓颖** 女,博士,副教授,硕士生导师,研究方向为软件工程和软件测试;**黄冠利** 硕士,讲师,研究方向为网络信息技术、软件测试技术、数据挖掘等;**严 群** 双学士,高级工程师,研究方向为计算机应用。

质量相关的因素作为评价指标；另一方面，由于 Web 服务测试的特点要求，对于 Web 服务的测试评估又应该是建立在协同测试服务的基础上的。由于各方测试资源的可信程度各不相同，必须从测试者、测试用例等各种资源的可信度方面对测试结果真实性进行多维度、综合考虑。

为了弥补传统 UDDI 在服务测试验证方面的不足，我们建立了一种可信的服务测试中介模型，它是对传统的 UDDI 模型的一种扩展，是一种具备协同测试与验证的服务测试功能的服务中介。该中介的服务器可以分布在各地，针对不同的问题承担不同的测试验证任务，同时还能彼此协作，实时地为用户提供满足需要的、可信的、具有质量保证的服务。其框图如图 1 所示。

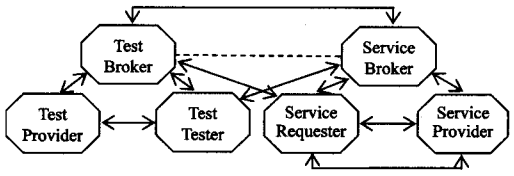

图 1　可信的服务测试中介

该中介提供一个多方参与的协同测试的管理平台，参与测试的各方用户可在 Web 浏览器中实时地发布各种测试资产，包括测试用例、测试脚本、测试结果；测试者可以利用已发布的各种测试资源进行测试并发布相应的测试结果，系统根据测试的结果自动地对服务进行评估排行。

在收集了大量的服务测试结果数据之后，对相同类型的服务进行服务质量排行是我们主要需要解决的问题，因为排行算法的优劣直接影响到协同测试系统的性能和质量。在文献[4]中我们提出过一种基于改进的贝叶斯投票算法的服务评估技术，利用协同服务测试结果数据，结合测试者的信誉度权值对服务评估进行排行评估的方法。该算法借鉴了类似“民主投票”的思想，测试者通过发布测试结果的方式为自身感兴趣的服务进行投票，如果某一服务的测试执行结果是“通过测试”，就相当于测试者对此服务投了一个赞成票，反之则投了反对票。通过对每个服务测试通过率在同类服务的平均通过率所进行的加权计算，最终进行同类服务的质量排行。本文是在此基础上进一步深入地分析影响服务评估的各种因子之间的相互关系，提出一种结合测试者信誉度和测试用例权值等多维度因子对服务进行排行评估的方法，并阐述其具体的矩阵表达式和运算关系。

3　服务的多维度评估算法

如果仅仅根据测试通过率的加权平均值进行服务评估的话，某服务的加权测试通过率计算公式如下[5]：

$$WR(s)=\frac{\frac{1}{n}\sum_{i=1}^{n}(N_i\times R_i)+N_s\times R_s}{\frac{1}{n}\sum_{i=1}^{n}N_i+N_s}$$

其中，$WR(s)$ 是服务“s”的加权通过率，N 是服务的测试次数。R 是服务的通过率。n 是此类服务的服务个数。

然而，由于每个测试者所处的地位不同、拥有的资源和测试经验不同，对于一个服务来说，即使测试的结果相同，这些测试的结果对服务质量评估的准确度也是不同的，每个不同的测试者对于服务评估的影响的信誉度是不同的；同样地，不同的测试用例由于其设计者的水平不同、测试覆盖率和针对性的不同，其信誉度也各不相同。如果只考虑服务测试次数的权重影响，而忽略了测试者和测试用例在每次测试中所产生的影响的话，这样计算出的服务排行的可信性就会存在着某些缺陷。因此，我们把影响服务质量排行的人为因素——如测试者信誉度、测试用例的信誉度等作为权值，多维度地综合考虑，在计算时尽量平抑掉人为干扰噪声，从而更加真实地体现出服务质量的原本属性。

3.1　结合用户信誉度权值的评估算法

在用户集合{$U1\cdots\cdots Um$}中的各个元素“用户”对服务集合{$S1\cdots\cdots Sn$}中的某些元素“服务”进行测试活动中，如果考虑到用户的信誉度权值随测试次数的增加而增长的因素的话，就存在着三元组(U_i，T_k，S_j)的关系。其中 U_i 是用户集合{$U_1\cdots\cdots U_m$}中的元素，T_k 是所有测试执行的集合{$T_1\cdots\cdots T_l$}中的元素，S_j 是服务集合{$S_1\cdots\cdots S_n$}中的元素，它们之间的关系如下：一个用户可能测试多个服务，而且针对同一个服务也可能有多次测试执行，同一个用户在不同的测试执行中的信誉度权值不同，一个服务也可能被多个用户所测试。用矩阵可以表示如下：

$$P=\begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \\ \vdots \\ S_n \end{array}\begin{array}{c} \begin{array}{cccccc} T_1 & T_2 & T_3 & \cdots & T_l \end{array} \\ \begin{pmatrix} p_{11} & p_{12} & p_{13} & \cdots & p_{1l} \\ p_{21} & p_{22} & p_{23} & & p_{2l} \\ p_{31} & p_{32} & p_{33} & & p_{3l} \\ \vdots & & & \ddots & \vdots \\ p_{n1} & p_{n2} & p_{n3} & \cdots & p_{nl} \end{pmatrix} \end{array}$$

矩阵 P 表示服务集合{$S_1\cdots\cdots S_n$}与测试执行的集合{$T_1\cdots\cdots T_l$}一对多的关系，即一次测试执行只针对一个服务，而一个服务可以在不同的测试执行中被测试。矩阵中各元素表示每个服务在各次测试执行中是否测试通过，通过为 1，没通过或没有参与测试为 0。例如：

$$P=\begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \\ \vdots \\ S_n \end{array}\begin{array}{c} \begin{array}{ccccc} T_1 & T_2 & T_3 & \cdots & T_l \end{array} \\ \begin{pmatrix} 1 & 0 & 1 & \cdots & 0 \\ 0 & 1 & 0 & & 1 \\ 0 & 0 & 0 & & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix} \end{array}$$

在这个矩阵中可见的元素表示服务 S_1 在第 1 次和第 3 次的两次测试执行中通过测试，S_3 在第 2 次和第 l 次测试执行中通过测试，其余的服务在各次测试执行中没有参与测试或没有通过测试。

$$Z=\begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \\ \vdots \\ S_n \end{array}\begin{array}{c} \begin{array}{cccccc} T_1 & T_2 & T_3 & \cdots & T_l \end{array} \\ \begin{pmatrix} z_{11} & z_{12} & z_{13} & \cdots & z_{1l} \\ z_{21} & z_{22} & z_{23} & & z_{2l} \\ z_{31} & z_{32} & z_{33} & & z_{3l} \\ \vdots & & & \ddots & \vdots \\ z_{n1} & z_{n2} & z_{n3} & \cdots & z_{nl} \end{pmatrix} \end{array}$$

矩阵 Z 表示服务集合{$S_1\cdots\cdots S_n$}与测试执行的集合{$T_1\cdots\cdots T_l$}一对多的关系，即一次测试执行只针对一个服务，而

一个服务可以在不同的测试执行中被测试。与矩阵 B 不同的是，矩阵中各元素表示每个服务在各次测试执行中是否参与，参与测试为 1，没有参与测试为 0。例如：

$$Z=\begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \\ \vdots \\ S_n \end{array}\begin{array}{c} \begin{array}{ccccc} T_1 & T_2 & T_3 & \cdots & T_l \end{array} \\ \begin{pmatrix} 1 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 0 & & 0 \\ 0 & 0 & 1 & & 1 \\ \vdots & & & \ddots & \vdots \\ 0 & 1 & 0 & \cdots & 0 \end{pmatrix} \end{array}$$

在这个矩阵中可见的元素表示服务 S_1 参与了第 1 次和第 3 次测试，S_3 参与了第 3 次和第 l 次测试，S_n 参与了第 2 次测试，其余的服务在各次测试执行中没有参与测试。

$$Q=\begin{array}{c} \\ T_1 \\ T_2 \\ T_3 \\ \vdots \\ T_l \end{array}\begin{array}{c} \begin{array}{ccccc} U_1 & U_2 & U_3 & \cdots & U_m \end{array} \\ \begin{pmatrix} q_{11} & q_{12} & q_{13} & \cdots & q_{1m} \\ q_{21} & q_{22} & q_{23} & & q_{2m} \\ q_{31} & q_{32} & q_{33} & & q_{3m} \\ \vdots & & & \ddots & \vdots \\ q_{l1} & q_{l2} & q_{l3} & \cdots & q_{lm} \end{pmatrix} \end{array}$$

矩阵 Q 表示测试执行的集合$\{T_1\cdots\cdots T_l\}$与用户集合$\{U_1\cdots\cdots U_m\}$多对一的关系，即一个测试执行只能被一个用户所执行，而一个用户可以参与多次测试执行。矩阵中各元素表示同一用户在不同测试中的不同信誉度权值，不同类型的用户有不同的初始信誉度权值，而且每个用户每增加一次测试其权值加 1，元素值为 0 表示没有参与此次测试。例如：

$$Q=\begin{array}{c} \\ T_1 \\ T_2 \\ T_3 \\ \vdots \\ T_l \end{array}\begin{array}{c} \begin{array}{ccccc} U_1 & U_2 & U_3 & \cdots & U_m \end{array} \\ \begin{pmatrix} 11 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & & 0 \\ 12 & 0 & 0 & & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 30 & \cdots & 0 \end{pmatrix} \end{array}$$

在这个矩阵中可见的元素表示 U_1 在 T_1 测试时的权值为 11，在 T_2 测试时的权值为 12；U_3 在 T_2 测试时权值为 1、在 T_3 测试时权值为 30。

$$A=P\times Q=\begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \\ \vdots \\ S_n \end{array}\begin{array}{c} \begin{array}{ccccc} T_1 & T_2 & T_3 & \cdots & T_l \end{array} \\ \begin{pmatrix} p_{11} & p_{12} & p_{13} & \cdots & p_{1l} \\ p_{21} & p_{22} & p_{23} & & p_{2l} \\ p_{31} & p_{32} & p_{33} & & p_{3l} \\ \vdots & & & \ddots & \vdots \\ p_{n1} & p_{n2} & p_{n3} & \cdots & p_{nl} \end{pmatrix} \end{array}\times\begin{array}{c} \\ T_1 \\ T_2 \\ T_3 \\ \vdots \\ T_l \end{array}\begin{array}{c} \begin{array}{ccccc} U_1 & U_2 & U_3 & \cdots & U_m \end{array} \\ \begin{pmatrix} q_{11} & q_{12} & q_{13} & \cdots & q_{1m} \\ q_{21} & q_{22} & q_{23} & & q_{2m} \\ q_{31} & q_{32} & q_{33} & & q_{3m} \\ \vdots & & & \ddots & \vdots \\ q_{l1} & q_{l2} & q_{l3} & \cdots & q_{lm} \end{pmatrix} \end{array}$$

$$=\begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \\ \vdots \\ S_n \end{array}\begin{array}{c} \begin{array}{ccccc} U_1 & U_2 & U_3 & \cdots & U_m \end{array} \\ \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1m} \\ a_{21} & a_{22} & a_{23} & & a_{2m} \\ a_{31} & a_{32} & a_{33} & & a_{3m} \\ \vdots & & & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nm} \end{pmatrix} \end{array}$$

矩阵 A 是由矩阵 P 乘以矩阵 Q 得到的，表示服务集合$\{S_1\cdots\cdots S_n\}$与用户集合$\{U_1\cdots\cdots U_m\}$多对多的关系，即一个用户可以测试多个服务，一个服务也可以被多个用户测试。还有一个隐含的关系就是一个用户可以多次测试同一个服务，用户信誉度动态变化对测试通过的影响就被表示在矩阵 A 的元素中了，如 a_{ij} 就是表示由 j 用户在测试 i 服务时，几次测试通过所带的不同信誉度权值之和，$a_{ij}=\sum_{k=1}^{l} p_{ik}\cdot q_{kj}$。例如：

$$A=P\times Q=\begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \\ \vdots \\ S_n \end{array}\begin{array}{c} \begin{array}{ccccc} U_1 & U_2 & U_3 & \cdots & U_m \end{array} \\ \begin{pmatrix} 23 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 31 & & 0 \\ 0 & 0 & 0 & & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix} \end{array}$$

在这个矩阵中可见的元素表示用户 U_1 在不同次测试通过服务 S_1 时所带的权值之和为 11+12=23，U_3 在不同次测试通过服务 S_2 时所带的不同权值之和为 1+30=31。

同理，可以算出矩阵 B，它表示服务集合$\{S_1\cdots\cdots S_n\}$与用户集合$\{U_1\cdots\cdots U_m\}$另一种多对多的关系，其中 b_{ij} 是由 j 用户在测试 i 服务时，各次测试时所带的不同信誉度权值之和，$b_{ij}=\sum_{k=1}^{l} z_{ik}\cdot q_{kj}$。

$$B=Z\times Q=\begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \\ \vdots \\ S_n \end{array}\begin{array}{c} \begin{array}{ccccc} T_1 & T_2 & T_3 & \cdots & T_l \end{array} \\ \begin{pmatrix} z_{11} & z_{12} & z_{13} & \cdots & z_{1l} \\ z_{21} & z_{22} & z_{23} & & z_{2l} \\ z_{31} & z_{32} & z_{33} & & z_{3l} \\ \vdots & & & \ddots & \vdots \\ z_{n1} & z_{n2} & z_{n3} & \cdots & z_{nl} \end{pmatrix} \end{array}\times\begin{array}{c} \\ T_1 \\ T_2 \\ T_3 \\ \vdots \\ T_l \end{array}\begin{array}{c} \begin{array}{ccccc} U_1 & U_2 & U_3 & \cdots & U_m \end{array} \\ \begin{pmatrix} q_{11} & q_{12} & q_{13} & \cdots & q_{1m} \\ q_{21} & q_{22} & q_{23} & & q_{2m} \\ q_{31} & q_{32} & q_{33} & & q_{3m} \\ \vdots & & & \ddots & \vdots \\ q_{l1} & q_{l2} & q_{l3} & \cdots & q_{lm} \end{pmatrix} \end{array}$$

$$=\begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \\ \vdots \\ S_n \end{array}\begin{array}{c} \begin{array}{ccccc} U_1 & U_2 & U_3 & \cdots & U_m \end{array} \\ \begin{pmatrix} b_{11} & b_{12} & b_{13} & \cdots & b_{1m} \\ b_{21} & b_{22} & b_{23} & & b_{2m} \\ b_{31} & b_{32} & b_{33} & & b_{3m} \\ \vdots & & & \ddots & \vdots \\ b_{n1} & b_{n2} & b_{n3} & \cdots & b_{nm} \end{pmatrix} \end{array}$$

通过计算矩阵 A 和 B，可以得出各个用户在不同次测试服务时，所具有不同的用户信誉度权值之和，这样就可以充分地体现用户信誉度权值的动态变化，使改进后的算法能够真实地反映出用户信誉度权值动态变化对加权通过率的影响。改进后的算法表达如下：

$$WR(s)=\frac{\frac{1}{n}\sum_{\substack{i=1\sim n\\ j=1\sim m}} a_{ij}+\sum_{j=1\sim m} a_{sj}}{\frac{1}{n}\sum_{\substack{i=1\sim n\\ j=1\sim m}} b_{ij}+\sum_{j=1\sim m} b_{sj}}$$

$$=\frac{\frac{1}{n}\sum_{\substack{i=1\sim n\\ j=1\sim m}}\left(\sum_{k=1\sim l}(p_{ik}\times q_{kj})\right)+\sum_{j=1\sim m}\left(\sum_{k=1\sim l}(p_{sk}\times q_{kj})\right)}{\frac{1}{n}\sum_{\substack{i=1\sim n\\ j=1\sim m}}\left(\sum_{k=1\sim l}(z_{ik}\times q_{kj})\right)+\sum_{j=1\sim m}\left(\sum_{k=1\sim l}(z_{sk}\times q_{kj})\right)}$$

3.2 结合用户和测试用例信誉度权值的评估算法

在考虑用户信誉度权值的影响的情况下，如果再考虑到测试用例因素的影响，那么就存在着四元组(U_i, C_g, T_k, S_j)关系，C_g是测试用例集合$\{C_1 \cdots\cdots C_h\}$中的元素，这样就存在以下几个矩阵关系：

$$R=\begin{array}{c} \\ U_1 \\ U_2 \\ U_3 \\ \vdots \\ U_m \end{array}\begin{array}{c} \begin{array}{ccccc} C_1 & C_2 & C_3 & \cdots & C_h \end{array} \\ \begin{pmatrix} r_1 & r_2 & r_3 & \cdots & r_h \\ r_1 & r_2 & r_3 & & r_h \\ r_1 & r_2 & r_3 & & r_h \\ \vdots & \vdots & & \ddots & \vdots \\ r_1 & r_2 & r_3 & \cdots & r_h \end{pmatrix} \end{array}$$

矩阵R表示用户集合$\{U_1 \cdots\cdots U_m\}$与测试用例集合$\{C_1 \cdots\cdots C_h\}$多对多的关系，即一个测试用例可以被多个用户所采用，同时一个用户也可以采用多个测试用例。矩阵中各元素表示不同用户采用不同测试用例所具有的不同的测试用例权值。对于用户而言，同一次计算时的测试用例权值是固定的，所以矩阵具有相同的行值。

$$D=Q\times R=\begin{array}{c} \\ T_1 \\ T_2 \\ T_3 \\ \vdots \\ T_l \end{array}\begin{array}{c} \begin{array}{ccccc} U_1 & U_2 & U_3 & \cdots & U_m \end{array} \\ \begin{pmatrix} q_{11} & q_{12} & q_{13} & \cdots & q_{1m} \\ q_{21} & q_{22} & q_{23} & & q_{2m} \\ q_{31} & q_{32} & q_{33} & & q_{3m} \\ \vdots & \vdots & & \ddots & \vdots \\ q_{l1} & q_{l2} & q_{l3} & \cdots & q_{lm} \end{pmatrix} \end{array}\times\begin{array}{c} \\ U_1 \\ U_2 \\ U_3 \\ \vdots \\ U_m \end{array}\begin{array}{c} \begin{array}{ccccc} C_1 & C_2 & C_3 & \cdots & C_h \end{array} \\ \begin{pmatrix} r_1 & r_2 & r_3 & \cdots & r_h \\ r_1 & r_2 & r_3 & & r_h \\ r_1 & r_2 & r_3 & & r_h \\ \vdots & \vdots & & \ddots & \vdots \\ r_1 & r_2 & r_3 & \cdots & r_h \end{pmatrix} \end{array}$$

$$=\begin{array}{c} \\ T_1 \\ T_2 \\ T_3 \\ \vdots \\ T_l \end{array}\begin{array}{c} \begin{array}{ccccc} C_1 & C_2 & C_3 & \cdots & C_h \end{array} \\ \begin{pmatrix} d_{11} & d_{12} & d_{13} & \cdots & d_{1h} \\ d_{21} & d_{22} & d_{23} & & d_{2h} \\ d_{31} & d_{32} & d_{33} & & d_{3h} \\ \vdots & \vdots & & \ddots & \vdots \\ d_{l1} & d_{l2} & d_{l3} & \cdots & d_{lh} \end{pmatrix} \end{array}$$

矩阵D是由矩阵Q乘以矩阵R得到的，表示测试执行集合$\{T_1 \cdots\cdots T_n\}$与测试用例集合$\{C_1 \cdots\cdots C_h\}$多对多的关系，即一次测试执行可以采用多个测试用例，一个测试用例也可以被多次测试执行所采用。其中，$d_{kg}=\sum_{j=1}^{m} q_{kj} \cdot r_{jg}$。由于矩阵$Q$中每次测试执行可能被一个用户执行，因此$d_{kg}$就是表示第$g$个测试用例被第$j$个用户采用在第$k$次测试执行时的用户信誉度权值与测试用例权值的乘积。

然后计算出矩阵E和F。其意义是矩阵A, B基础上加上了测试用例的权值因素。矩阵表示如下：

$$E=P\times D=\begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \\ \vdots \\ S_n \end{array}\begin{array}{c} \begin{array}{ccccc} T_1 & T_2 & T_3 & \cdots & T_l \end{array} \\ \begin{pmatrix} p_{11} & p_{12} & p_{13} & \cdots & p_{1l} \\ p_{21} & p_{22} & p_{23} & & p_{2l} \\ p_{31} & p_{32} & p_{33} & & p_{3l} \\ \vdots & \vdots & & \ddots & \vdots \\ p_{n1} & p_{n2} & p_{n3} & \cdots & p_{nl} \end{pmatrix} \end{array}\times\begin{array}{c} \\ T_1 \\ T_2 \\ T_3 \\ \vdots \\ T_l \end{array}\begin{array}{c} \begin{array}{ccccc} C_1 & C_2 & C_3 & \cdots & C_h \end{array} \\ \begin{pmatrix} d_{11} & d_{12} & d_{13} & \cdots & d_{1h} \\ d_{21} & d_{22} & d_{23} & & d_{2h} \\ d_{31} & d_{32} & d_{33} & & d_{3h} \\ \vdots & \vdots & & \ddots & \vdots \\ d_{l1} & d_{l2} & d_{l3} & \cdots & d_{lh} \end{pmatrix} \end{array}$$

$$=\begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \\ \vdots \\ S_n \end{array}\begin{array}{c} \begin{array}{ccccc} C_1 & C_2 & C_3 & \cdots & C_h \end{array} \\ \begin{pmatrix} e_{11} & e_{12} & e_{13} & \cdots & e_{1h} \\ e_{21} & e_{22} & e_{23} & & e_{2h} \\ e_{31} & e_{32} & e_{33} & & e_{3h} \\ \vdots & \vdots & & \ddots & \vdots \\ e_{n1} & e_{n2} & e_{n3} & \cdots & e_{nh} \end{pmatrix} \end{array}$$

$$F=Z\times D=\begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \\ \vdots \\ S_n \end{array}\begin{array}{c} \begin{array}{ccccc} T_1 & T_2 & T_3 & \cdots & T_l \end{array} \\ \begin{pmatrix} z_{11} & z_{12} & z_{13} & \cdots & z_{1l} \\ z_{21} & z_{22} & z_{23} & & z_{2l} \\ z_{31} & z_{32} & z_{33} & & z_{3l} \\ \vdots & \vdots & & \ddots & \vdots \\ z_{n1} & z_{n2} & z_{n3} & \cdots & z_{nl} \end{pmatrix} \end{array}\times\begin{array}{c} \\ T_1 \\ T_2 \\ T_3 \\ \vdots \\ T_l \end{array}\begin{array}{c} \begin{array}{ccccc} C_1 & C_2 & C_3 & \cdots & C_h \end{array} \\ \begin{pmatrix} d_{11} & d_{12} & d_{13} & \cdots & d_{1h} \\ d_{21} & d_{22} & d_{23} & & d_{2h} \\ d_{31} & d_{32} & d_{33} & & d_{3h} \\ \vdots & \vdots & & \ddots & \vdots \\ d_{l1} & d_{l2} & d_{l3} & \cdots & d_{lh} \end{pmatrix} \end{array}$$

$$=\begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \\ \vdots \\ S_n \end{array}\begin{array}{c} \begin{array}{ccccc} C_1 & C_2 & C_3 & \cdots & C_h \end{array} \\ \begin{pmatrix} f_{11} & f_{12} & f_{13} & \cdots & f_{1h} \\ f_{21} & f_{22} & f_{23} & & f_{2h} \\ f_{31} & f_{32} & f_{33} & & f_{3h} \\ \vdots & \vdots & & \ddots & \vdots \\ f_{n1} & f_{n2} & f_{n3} & \cdots & f_{nh} \end{pmatrix} \end{array}$$

因此，综合考虑到用户信誉度权值和测试用例权值对服务加权通过率的影响的情况下，算法可以被进一步改进为：

$$WR(s)=\frac{\frac{1}{n}\sum_{\substack{i=1\sim n\\ g=1\sim h}}\left(\sum_{k=1\sim l} p_{ik}\times\left(\sum_{j=1\sim m}(q_{kj}\times r_{jg})\right)\right)+\sum_{g=1\sim h}\left(\sum_{k=1\sim l} p_{sk}\times\left(\sum_{j=1\sim m}(q_{kj}\times r_{jg})\right)\right)}{\frac{1}{n}\sum_{\substack{i=1\sim n\\ g=1\sim h}}\left(\sum_{k=1\sim l} z_{ik}\times\left(\sum_{j=1\sim m}(q_{kj}\times r_{jg})\right)\right)+\sum_{g=1\sim h}\left(\sum_{k=1\sim l} z_{sk}\times\left(\sum_{j=1\sim m}(q_{kj}\times r_{jg})\right)\right)}$$

为了能够得到较为真实的同类服务的平均通过率，系统还可以采用以下两个策略来限制实际计算时的数据取样范围：1)设定用户在单位时间内发布测试的次数，以便剔除那些为了拉动排行恶意频繁发布测试结果的行为；2)设定最低的信誉度权值，只有拥有高于此权值的用户所发布的测试结果才会被选入取样范围。通过这些措施，系统可以自动地、有效地抑制不良干扰，提高排名的可信度。

为实时、动态地实现服务排行，每当用户提交了一个新的测试资源时，系统就立即更新相关的测试者的信誉度权值、同类服务的平均加权通过率和该服务的加权通过率，并刷新排

行。由于每次计算都是在上次更新数据的基础上迭代进行的，因此排行计算速度很快。

结束语 本文在"服务测试中介"模型基础上，提出一种基于贝叶斯投票算法的、多维度的服务评估技术，测试服务的各方在同一个协同测试的平台上共享和重用测试资产，系统根据协同测试数据实时地为同类服务进行动态排行，由于采用了带有用户信誉度和测试用例信誉度的加权平均的方法，设计思想具备用户交互和数据为核心的 Web2.0 的特点，服务排行的可信度会随着测试者、测试用例参与度和测试次数的增多而不断地提高。

参 考 文 献

[1] Bai X, Cao Z, Chen Y. Design of a Trustworthy Service Broker and Dependence-Based Progressive Group Testing. International Journal of Simulation and Process Modeling (IJSPM), 2006

[2] Tsai W T, Paul R, Cao Z, et al. Verification of Web Services Using an Enhanced UDDI Server// Proc. of IEEE WORDS. 2003: 131-138

[3] Tsai W T, Chen Y, Paul R, et al. Adaptive Testing, Oracle Generation, and Test Script Ranking for Web Services// 29th Annual International Computer Software and Applications Conference (COMPSAC). Edinburgh, Scotland, July 2005: 101-106

[4] Levis, Alexander H. Modeling and Measuring Effectiveness of C3 Systems. ADA173694, Sep. 1986

[5] 张燕生，白晓颖，蒋长征. 一种基于改进的贝叶斯投票算法的服务评估技术. 计算机科学，2008，35(4)

(上接第 277 页)

策略。这种逻辑分级结构有利于管理角色分工不同、各司其职，对属于不同级别的策略分别进行控制和管理。

5.2 策略定义和策略实施的分离

策略的实施是通过以安全实施管理器为中心，再辅以保护区、操作栈和策略映像 3 种数据结构的 SoftMan 授权服务完成的。我们可以通过外部 GUI 工具定义策略，动态更新策略，不必更改系统实现就可以实现对 SoftMan 行为的控制。这种分离的结构明显优于传统的策略定义和策略实施混合硬编码的解决方案，具有更强的可用性。

5.3 系统资源的高度抽象

系统内不仅定义具有普遍性的权限，如 File 读写、Socket 连接，而且抽象出特定于 SoftMan 系统的权限，SoftManPermission，SoftManServerPermission 和 MessagePermission。Soft-Man 扩展了资源的范围和类型，为系统的全面安全奠定了良好的基础。而且针对特定应用的需要，我们还可以扩展定义新的权限类型。

5.4 权限的不可传递性

在没有权限委托的默认情形下，权限是闭合的，即不可传递。如图 2 中的操作栈，操作 OP_n 调用 OP_{n-1}，OP_{n-1} 调用 $OP_{n-2}\cdots$，OP_2 调用 OP_1，按照调用次序依次进栈，OP_1 位于栈顶。设 $SoftMan_B$ 拥有比 $SoftMan_A$ 更高的权限，$SoftMan_A$ 的操作 OP_i 调用 $SoftMan_B$ 的操作 OP_{i-1}，那么 OP_i 仍然只被赋予了 $SoftMan_A$ 而非 $SoftMan_B$ 的权限。当安全实施管理器查询操作栈时自顶向下检查，遇到不具有权限的操作项，拒绝授予此权限。系统通过操作栈的这种数据结构可以有效地避免恶意代码的越权资源访问。

5.5 SoftMan 授权的分类管理

模型摒弃了直接将每个 SoftMan 和策略映像联系起来的方法，而是采用了更为灵活的设计，把具有相同特征(如代码源)的 SoftMan 分类到相应的保护区，每个保护区与一定的权限集合相关联，集中进行管理。

SoftMan 授权模型将策略定义和策略实施分离，这样我们就可以动态更新策略定义而不需要重构系统，增强系统的可扩展性和可用性。在策略的组织上，借鉴了网络中域的思想，对策略分层，把域的基本策略和特定于某 SoftMan 服务器的策略彻底分离，细化了策略的定义[9,10]。

结束语 SoftMan 自保护机制从系统架构入手，综合利用现有各种自保护技术，能够为各种应用提供合适的灵活的自保护安全服务，在一定程度上解决了如何选择移动代码保护方法的问题。

SoftMan 自保护机制从宏观上构建了一种安全框架并提出一些实现上的具体问题。如 SoftMan 提出安全申请前携带的安全需求规格说明的格式和描述语言、安全实施中心验证 SoftMan 申请合法性的依据和手段以及使用何种有效策略综合利用检测技术库、预防技术库和综合技术库中的保护技术等，都将成为下一步的研究重点。

参 考 文 献

[1] 曾广平，涂序彦. 软件人-网络世界中的虚拟人工生命[A]. 中国人工智能进展 2003[C]. 北京：北京邮电大学出版社，2003，12：677-682

[2] 曾广平，涂序彦. "软件人"的概念模型与构造特征[J]. 计算机科学，2005，32(5)：135-136，143

[3] 王洪泊，班晓娟，曾广平，等. 网络环境下虚拟机器人——"SoftMan"系统平台总体设计[J]. 计算机科学，2005，32(8)：152-154

[4] Bellavista G, et al. Security in Distributed Computing [M]. Prentice Hall, 1996: 21-27

[5] KarJoth G, Lange D B, Oshima M. A Security Model forAglets [J]. IEEE Internet Computing, 1997: 68-77

[6] Damianou N, et al. Ponder: A Language for Specifying Security and Management Policies for Distributed Systems, v2·0 [R]. Imperial College Research Report DoC 2000/1. 2000: 12-30

[7] Montanar R, et al. Flexible Security Policies for Mobile Agent Systems[J]. Microprocessors and Microsystems, 2001, 25(2): 93-99

[8] Wang Hongbo, Wang Zongijie, Zeng Guangping, et al. The Research on SoftMan Coordination and Its Application in Digital Gas Fields [R]// 2005 IEEE International Conference on Neural Networks and Brain. Beijing, China, Oct. 2005: 1429-1433

[9] 王洪泊，曾广平，涂序彦. 软件人系统研究及其应用[J]. 智能系统学报，2006，1(1)：24-28

[10] 曾广平，涂序彦，王洪泊. "软件人"研究及应用[M]. 科学出版社，2007