基于 SOA 的 Web 服务可靠性预测

谢春丽[1] 俞析蒙[2] 王书芹[1]
(江苏师范大学计算机科学与技术学院 徐州 221116)[1] (东南大学计算机科学与工程学院 南京 211189)[2]

摘 要 在 SOA 架构下,服务发现的准确率是影响服务可靠性的因素之一。为了提高 Web 服务可靠性的预测精度,分析服务发现的失效是非常关键的。首先分析 Web 服务发现过程中可能出现的失效因素,构建 Web 服务发现的可靠性模型,并在此基础上提出新的 Web 服务的可靠性预测方法。最后通过实验验证了 Web 服务发现阶段的失效对 Web 服务可靠性的影响,并通过实例说明了 Web 服务的新的可靠性预测方法。实验结果表明,新的可靠性预测方法更加适用于 Web 服务。

关键词 SOA,服务可靠性,Web 服务
中图法分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2014.07.046

Reliability Prediction Approach for Web Service Based on SOA

XIE Chun-li[1] YU Xi-meng[2] WANG Shu-qin[1]
(School of Computer Science & Technology, Jiangsu Normal University, Xuzhou 221116, China)[1]
(School of Computer Science & Engineering, Southeast University, Nanjing 211189, China)[2]

Abstract In SOA, the precision ration of service discovery is the key factor in predicting service reliability. To improve the prediction accuracy of reliability for Web services, it is important to analyze the failure of service discovery. In this paper, firstly, the failure of service discovery was described and a discovery reliability model was built. Then, a new approach for predicting services reliability was proposed based on the discovery reliability. Lastly, an experiment was conducted and the influence of service discovery on service reliability was analyzed. The experimental results show that the new reliability approach is more appropriate for Web services.

Keywords SOA, Service reliability, Web service

1 引言

SOA(Service-Oriented Architecture)作为一种分布式应用范型,为 Web 服务的应用和组合提供了便利。Web 服务具有自描述性和自适应性,这为 Web 服务的自动发现和组合奠定了基础;而且 Web 服务采用松散耦合和动态绑定的方式,服务消费者和服务提供者之间通过服务代理如常用的 UDDI(Universal Description, Discovery and Integration)中心互操作,服务提供者将服务注册到 UDDI 中心,服务消费者根据自己的需要从 UDDI 中心发现想要的服务,并绑定到服务提供者。Web 服务所处的环境是一个动态的网络环境,很多服务可能经常发生变化,由于缺少自动更新机制,服务提供者可能经常忘记更新服务注册中心 UDDI 的信息[1,2],这导致服务注册中心的很多服务已经失效,服务消费者在不知道的情况下,组合、绑定这些服务将会引起服务的失效。

近年来对原子服务可靠性的研究比较少,多数文献都是在原子服务可靠性的基础上研究组合服务的可靠性,假设原子服务的可靠性是已知的[3-5]。由于原子服务是由服务提供者提供,其内部结构对用户来说不可见,因此只能使用黑盒方法定量分析基本服务的可靠性。Tsai 提出了对具有相同功能的基本服务进行群测试[6],实时计算 Web 服务的可靠性,并使用多数票决的思想选择参与执行的基本服务。该方法需要强大计算能力的 Web 服务开发环境来完成服务的测试和选择功能,集成和运行代价甚高。Li 等人提出了基于结构信息和运行日志的服务可靠性分析,该模型既分析了服务的静态结构信息也考虑了服务执行的动态信息[7]。Zheng 等人利用聚类的方法对相似服务或者相似服务用户进行聚类,通过相似服务或用户的可靠性来预测新服务或新用户的可靠性[8-10]。Silic 等人提出了利用数据挖掘的方法预测服务的可靠性[11],该方法对服务调用的数据利用 K-means 方法进行原子服务可靠性的预测。这些方法都需要大量的服务调用数据,也就是说是从服务使用者的角度考虑服务的可靠性。Bloomberg 认为 Web 服务的失效不仅仅包含基本功能的失效,还应包括 SOAP 消息,WSDL 文档,SOA 的发布、发现和绑定能力,Web 服务的异步能力,SOAP 的中介能力,服务的 QoS,动态运行能力的失效等[12]。因此服务的发布、发现以及绑定能力对 Web 服务的可靠性都会有一定的影响,本文立足于观察 Web 服务发现和服务可靠性的关系,分析服务发现过

到稿日期:2013-09-01 返修日期:2014-01-03 本文受徐州师范大学自然科学研究基金项目(10XLB12)资助。
谢春丽(1979-),女,博士,主要研究领域为软件可靠性分析、Web 服务可靠性预测和分析等,E-mail:xcl_bhb@163.com;俞析蒙(1990-),男,硕士生,主要研究领域为 Web 服务可靠性预测和分析;王书芹(1979-),女,博士生,主要研究领域为机器学习等智能信息处理。

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

程中可能出现的失效因素，通过实验观察服务发现的失效对Web服务可靠性的影响。

目前主要的服务发现技术有基于关键字的服务发现、基于语义的服务发现[13,14]。基于关键字服务发现完全依赖于关键词的准确与否，服务发现结果很大程度上受到关键词的影响，一般返回的结果集比较大，可能存在大量无关的、不能和服务请求匹配的服务，需要用户手工筛选。例如，返回结果集中会出现很多服务的输入参数和用户请求的参数不一致等现象，导致服务的查准率(Precision Ratio)降低，同时基于关键字的查询可能会遗漏一些能够很好进行匹配的服务，从而影响服务的查全率(Recall Ratio)。为了提高服务查准率，近几年研究者们开始致力于研究基于语义的服务发现，该方法将Web服务和语义Web技术结合，充分利用服务中存在的语义信息，针对服务请求和广告服务中描述的功能进行匹配，并通过语义相似度来衡量两者匹配的程度，在一定程度上提高了Web服务发现的查准率和查全率。本文提出的方法是对原子服务在执行之前的可靠性进行考虑，主要从服务提供者以及服务注册中心两方面的角度考虑服务的可靠性，不对服务执行信息进行考虑。

2 Web服务发现

2.1 Web服务发现的错误类型

服务具有自描述性，WSDL(Web Services Description Languages)是Web服务的描述语言，以文档的形式描述服务的功能和非功能信息，服务发布时把WSDL文档注册到UDDI中心。服务提供者用WSDL文档公布服务的信息，服务消费者通过服务的WSDL文档了解服务，因此WSDL文档的准确性、正确性直接关系到服务的查找效率[15-17]。

服务消费者根据搜索条件在UDDI中心搜索满足条件的服务，UDDI中心返回的服务结果集中有可能出现以下3类问题。

1)错误类型1：服务描述文件错误

此种错误产生的原因主要是服务提供者在服务中心UDDI注册服务时，未仔细检查测试，因而使得服务描述文件WSDL内部含有语法错误或逻辑错误。在此种情况下，对用户而言，该服务在尚未使用的时候是完全正确的一个服务，因此用户能够查询到相应服务并且使用，但是不会获得正确的服务响应与结果，发生服务失效。

2)错误类型2：返回的服务已被提供者修改或删除

由于服务中心UDDI与服务提供者相对独立，一般在物理距离上相隔也很远，同时UDDI中心里通常注册有大量的来自各种不同提供者注册的服务，庞大的数量使得UDDI中心无法实时根据服务提供者的修改来修正注册信息。因此从服务提供者到服务中心UDDI存在一个信息交流的延迟，短至几分钟，长至几天甚至几个月。这就会导致一种情况的发生：服务提供者更新、修改或是删除了某个服务，但是反映到UDDI中心却尚未修改。在这段二者同时发生的时间里，倘若有服务用户查询该服务，UDDI中心会反馈该服务的信息给用户，使得用户选择该服务。然而，实际上该服务已不是用户所需要的功能或是已被删除，这就导致服务失效。

由于UDDI中心里所提供的服务数量庞大，因此相对于列出来的其他错误种类，该种错误发生的概率是最高的。

3)错误类型3：服务参数不匹配

此种错误主要是由于用户输入的参数类型与查找到服务的参数类型不一致，使得查找到的服务不符合用户要求。如果仅仅是按照关键字进行查找，服务的输入输出参数并未列入查找条件中，只能由UDDI中心给出后由用户来判断参数正确与否。若不正确，那么该服务就出现失效情况。例如，一个查询某个城市的天气预报的服务，用户要求的服务输入参数是城市名称，但是返回的服务集中有的输入参数是城市名称，有的是城市ID，有的是邮政编码，因此随机选择服务调用，同样会出现失效。

2.2 Web服务的查准率

查准率是指Web服务在发现阶段查找到的有效服务数与查找到的总服务数的比例。与之相对应的概念为查全率，是指Web服务在服务发现阶段查找到的有效服务数与应当查找到的服务数的比例。Web服务的查准率和服务的WSDL文档，以及UDDI注册中心的情况有着直接的关系，本文中只计算在错误类型2存在的情况下服务的查准率。假设S是应当查找到的总服务集合即标准结果集，T是查找返回的服务集合，那么可以分别利用式(1)和式(2)来计算查准率η和查全率θ：

$$\eta=\frac{S\cap T}{T} \tag{1}$$

$$\theta=\frac{S\cap T}{S} \tag{2}$$

查准率和查全率是通过两个不同的角度来观察服务查找的结果，二者是相互关联的。本文中只考虑服务查找的准确率对可靠性的影响，因此只关心查准率的计算。

3 Web服务发现和可靠性的关系

3.1 服务发现的可靠度

Web服务发现的可靠度是指当服务请求者向UDDI中心请求服务时，UDDI中心能够正确返回请求服务的概率。服务发现的可靠度和服务的查准率有着直接的关系。查准率依赖于发现过程中可能出现的错误，接下来根据上节给出的服务发现时的错误类型，定义服务发现的可靠度。

1)错误服务数m：按照关键字查询到的服务中属于错误服务集的服务数量。

2)查询的服务总数M：按照关键字查询出来的所有服务的数量。

3)查准率η：查询结果中，满足该关键字的服务中属于给定的正确服务集的服务数量与满足关键字的服务的数量的比例，即

$$\eta=\frac{M-m}{M} \tag{3}$$

此处，查准率只包含错误类型2的情况。

4)服务发现的可靠度R_{dis}：综合考虑了3种错误类型的计算结果，因此其计算公式要综合考虑多个同名服务所产生的影响。将服务的选择分为3个阶段，第一个阶段是从多个服务中找出属于正确服务集的服务；第二个阶段是考虑选出的服务的语法、逻辑正确率对查准率的影响；第三个阶段是考虑错误类型3对可靠度的影响，因此服务发现的可靠度可按如下公式计算：

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

$$R_{dis}=\eta\beta\sum_{i=1}^{M-m}\frac{\alpha_i}{M-m} \tag{4}$$

其中，α_i 是查询到的 $M-m$ 个正确服务中第 i 个服务的语法逻辑正确率，β 反映错误类型 3 对可靠度带来的影响。当查询到的服务只有一个且为正确服务集里的服务时，服务的查准率的计算可以简化为：

$$R_{dis}=\alpha\beta\eta \tag{5}$$

由之前的分析可知，服务的可靠度与服务发现的可靠度有着直接的关系。而服务发现的可靠度 R_{dis} 则直接受到查准率的影响。

3.2 Web 服务的可靠性

Web 服务的可靠性指的是由服务提供者提供给用户的 Web 服务能否正确执行，它可以用可靠度来进行衡量。可靠度是可靠性的定量表示，一般用概率来表示。可靠度作为 Web 服务的重要属性之一，可以通过多种方法得出，或者由服务提供者给出，或者通过测试获取等等[18,19]，但是一般情况下给定的可靠度都是本地测试的结果，没有考虑网络环境的影响，也没有考虑在 SOA 架构下其它因素的影响，因此是不够准确的。我们需要考虑 Web 服务发布到 UDDI 中心之后，UDDI 中心对服务的可靠度的影响，重新计算服务的可靠度。

我们将服务提供者给定的可靠度或者本地测试得到的可靠度称为服务的初始可靠度，记为 R，在 UDDI 中心，Web 服务发现的可靠度记为 R_{dis}，则 Web 服务新的可靠度根据不同的服务匹配策略有不同的计算方法。如果我们在返回的服务结果集中随机选择一个服务进行绑定，则 Web 服务新的可靠度为：

$$R'=\frac{R_{dis}\sum_{i=1}^{N}R_i}{n} \tag{6}$$

其中，N 为返回的结果集中所有服务的个数，n 是返回的结果集中正确服务的个数。如果将返回结果集中所有服务按照可靠度高低排序，并按照顺序选择正确的服务进行绑定，则 Web 服务新的可靠度为：

$$R'=R_{dis}R \tag{7}$$

4 实验

4.1 实验准备

本实验主要是针对服务发现阶段的可靠度进行的实验。对于本实验，在某些环节做出一些分析与假设如下：

1）由于无法搭建一个实际的服务器并在异地的客户机上进行访问，实验平台是由 JUDDIv3 构建的。用于在平台上测试的服务由人为注册。服务的来源是网络提供的免费服务资源，总体样本共 3738 个。由于采用的是手动注册服务的方式，每个服务注册都需要花费一定的时间，因此只能抽取其中 150 个左右的服务用于实际测试环境的搭建。其服务描述文件 WSDL 文档是按照“服务 ID_服务提供国_服务名称”的方式存储，按照国家的不同从 Australia、Argentina、Belgium 中选出部分服务，分为数个商业实体（business）来注册，以此来区分不同的服务商提供的服务。

2）对于错误类型 1，即服务描述文件 WSDL 语法或逻辑错误，根据之前服务过程模型中的分析，我们按照不同的服务提供者，根据随机数分配来给出失效概率，即对同一个服务提供者提供的服务，它发生语法或逻辑错误的概率在某一个区间浮动，通过随机数的生成来指定该概率值；而对不同的提供者，这个区间范围是不同的。

3）对于错误类型 2，即服务已被提供者删除或修改但 UDDI 中心的信息未更新的错误，我们无法获取实际数据，因此只能考虑通过某种方式来进行模拟。由于 UDDI 中心注册的服务数量庞大，以及发生该类错误的随机性与不可预知性，考虑通过人为指定部分服务的方式来实现模拟。故而，对于该类错误，通过人为指定其中 2 个商业实体中的服务作为发生该类错误的服务集合来实现模拟。很明显，人为指定的概率会与实际发生的概率不一致，因此并不能用本实验中的概率来代替实际概率，而是通过不同概率的变化来观察对查准率以及可靠性的影响。

4）对于错误类型 3，即参数不匹配的问题，很难通过数据来进行定量分析。这是由于我们搭建的 JUDDI 中心最后服务选择的决定权是在用户手上，而人的模拟性思维难以通过仿真来模拟。尽管仍然属于服务发现阶段的问题，但是参数不匹配的问题绝大部分发生在中心给出查询结果、由用户选择的时候。此时语义匹配与否，会因用户对其的理解不同而产生很大的反差，这会导致最后选择与否的区别，从而引起服务失效。因此，我们仅通过实例分析此种错误类型导致的服务失效。

综合以上关于本实验的分析，这里给出本实验所应用的假设如下：

1）注册的服务都是从资源库里随机选取的；

2）UDDI 中心的服务对用户而言是黑匣状态；

3）通过指定概率的方式来模拟错误类型 1；

4）通过指定服务错误集的方式来模拟错误类型；

5）根据关键字进行服务查找和匹配。

4.2 实验

本实验是在开源平台 Juddi-portal-bundle-3.1.3 上进行的。实验的具体流程如下：

第 1 步　搭建测试平台并注册所需服务。

第 2 步　在 5 个商业实体、93 个服务（其中有 1 个商业实体、21 个服务是给定的作为错误 2 的错误服务集）的情况下进行查询测试，并记录查询数据，给出每个服务的语法错误概率等参数，计算这种情况下的查准率与可靠度。

第 3 步　增加 1 个商业实体、20 个服务至错误服务集中，再次进行测试并记录计算数据。

第 4 步　同第 3 步。

第 5 步　分析由参数不匹配引起的服务失效对查准率的影响，并综合分析查准率对服务可靠度的影响。

4.3 实验过程和分析

4.3.1 实验数据

如图 1 所示，在 JUDDI 中我们注册了 5 个商业实体 Argentina_business，Australia_business，Australia_business_2，Austria_business_3 和 Belgium_business_4，其中前面 4 个中的服务集合设为正确服务集，服务都是能被正确查找到的。Belgium_business_4 中的服务设置为错误服务集，其中的服务注册信息仍然在 JUDDI 中心，但实际服务已被服务提供者从服务器上删除了。这些服务的选取都是在资源集中，根据国家的不同，按照序号顺序挑选。

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

图 1　JUDDI 中心注册的服务列表

4.3.2　实验分析

实验 1　错误 1 和 2 对服务发现可靠度的影响

错误类型 1 主要是语法错误，这部分错误可以通过一些检查工具(例如 WSDL4J)检查出来，此处，我们采用随机数方法，任意指定一个[0,1]之间的概率值。对前 4 个商业实体中的服务设置了 α 的取值范围，分别为：

Argentina_business([0.7,0.85])，Australia_business([0.75,0.9])，Australia_business_2([0.8,0.95])，Austria_business_3([0.85,1.0])

具体每一个服务的 α 在对应的范围内随机给出。对于服务错误集中的服务需要分为两种情况考虑：一是与正确服务集中的服务有同名情况，则与其他同名服务数据相同；二是与正确服务集中的服务没有同名情况，则该服务必然查找失败，认为其语法、逻辑正确率为 0。由于数据量比较大，表 1 中仅给出了 2 个商业实体的实验结果，其它的结果在表中省略了。实验中我们对 5 个商业实体中共 93 个服务进行了查询。从我们的实验结果发现虽然 α 值增大了，但是查询的可靠性并没有随着 α 值显著增大，原因是错误类型 2 对服务的查准率造成较大的影响。

表 1　2 个商业实体的实验结果

服务名称	错误 1	错误 2			R_{dis}
	α	m	M	η	
Argentina_business	[0.7,0.85]				
BarcodeProfessionalWS	0.798	0	1	1	0.798
BuyerData	0.720	0	1	1	0.72
CinemaData	0.722	0	1	1	0.722
CinemaSinchronization	0.804	0	1	1	0.804
HotelDoInterface	0.804	0	1	1	0.804
Service	0.858	2	5	0.6	0.515
ServicioOnix	0.789	0	1	1	0.789
StadiumData	0.738	0	1	1	0.738
StadiumSinchronization	0.724	0	1	1	0.724
TiempoService	0.836	0	1	1	0.836
WSAcademico	0.756	0	1	1	0.756
WebService1	0.739	0	1	1	0.739
Belgium_business_4	—				
Authentication	0.828	3	10	0.7	0.58
BdcWebService	0.852	3	7	0.571	0.487
CatalogModService	0.000	1	1	0	0
Cijfers	0.000	2	2	0	0
LibelleService	0.000	1	1	0	0
Service	0.858	2	5	0.6	0.515
SessionManagement	0.000	1	1	0	0
SessionManagement-Service	0.000	1	1	0	0
ToucanSOAPService	0.000	1	1	0	0
VersionService	0.878	1	4	0.75	0.659
XmlSigner	0.000	1	1	0	0
content	0.000	1	1	0	0
iMobilWS	0.000	1	1	0	0
localService	0.000	1	1	0	0
sendsms	0.000	1	1	0	0

由于服务较多，我们以商业实体为单位进行分析，图 2 中给出了 5 个商业实体的查准率和服务发现可靠性的关系。

观察前 4 个商业实体可知，在查准率相差不大的情况下，语法、逻辑正确率较高的商业实体其服务发现的可靠度也会相应较高。比较其中的第 1 个和第 2 个商业实体可以发现，尽管在语法、逻辑正确率上第 2 个商业实体比第 1 个高一个等级，但是由于在查准率上第 1 个比第 2 个高不少(从0.875到0.707)，导致第 2 个商业实体的服务发现的可靠度要低于第 1 个的查准率。这也说明了错误类型 2，即由于服务提供者修改或删除但是未修改 UDDI 中心的信息所引起的错误，对服务服务发现的可靠度的影响是很大的。

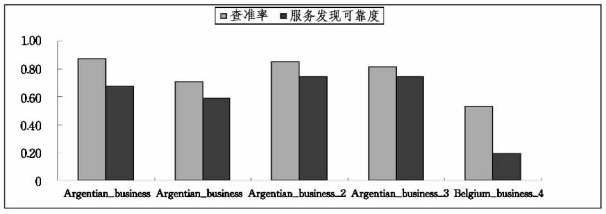

图 2　商业实体的查准率和服务发现的可靠度

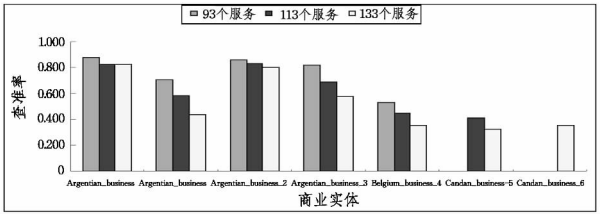

图 3　商业实体的查准率

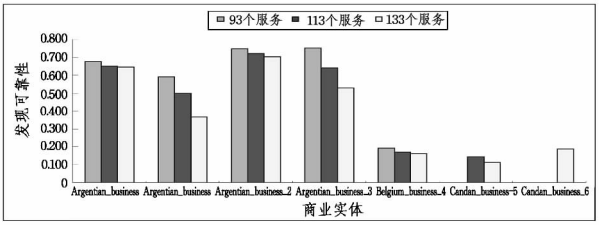

图 4　商业实体的服务发现的可靠度

据观察，第 5 个商业实体的实验结果与前 4 个商业实体相比，在查准率上明显低很多，这是由于该商业实体是作为错误服务集存在的。作为错误服务集，其服务均为被服务提供者删除的状态，查准率与可靠度应该均为 0。但实际上，其中有一部分服务的查找结果只会出现在错误集中，因此其查准率与可靠度确实为 0；同时由于同名服务的存在，使得在查找另一部分服务的时候会查找到正确的服务，因此总体可靠度的实际值不为 0。这使得第 5 个商业实体有了一定的可靠度，但依然比前 4 个商业实体低。

增加第 6 个和第 7 个商业实体，相应的服务个数由 93 依次增加到 113133。图 3 和图 4 中分别给出了各个商业实体查准率和服务发现可靠性的变化情况。从图中可以发现随着错误服务集的扩大，商业实体 Argentina_business 和 Austria_business_2 的可靠度并未有太大变化，而商业实体 Australia_business 和 Austria_business_3 的可靠度却大幅下降。这种情况的产生与和错误服务集有同名称的交集的服务的数量有关。前二者与错误服务集的交集较少，反映出实际情况即发生服务提供者未修改 UDDI 中心的服务信息的错误的情况较少，由此会使得服务的可靠度下降的幅度较小；而后二者则很明显地受到该种错误的影响，使得可靠度下降的幅度较大。

实验 2　错误 3 对服务发现可靠度的影响

虽然有时服务名称一样，但是其具体服务的内容有所区别，因而所需的输入参数会有差别。此时，由于其具体的所需

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

参数与用户传递给它的参数不一致，导致服务与用户的契合不一致，从而导致服务发现失败。

在 JUDDI 中心，根据用户需求查找名为 Service 的服务，结果搜寻到 5 个同名服务，其中有两个是服务发布者已删除或修改但未更新 UDDI 中心的信息的错误服务。这 5 个服务虽然拥有相同的名称，但是其具体所需要的参数以及具体的功能都是不同的。尽管在用户方会有相关的对服务的描述说明，但是其说明可获得的信息有限，使得用户的选择仍然未知。我们假设在 3 个正常服务中只有第 1 个服务符合用户传递参数的要求，而其他 2 个服务都不满足用户的参数需要。因此，尽管进行服务查询时会有 3 个正确服务，但是实际只有 1 个服务能够满足要求的服务响应，使得服务实际的匹配成功率下降。此时 λ 的值只有 0.33 甚至更低，使得查询该服务时的服务可靠度 R_{dis} 与不考虑参数不匹配的问题相比大幅度下降。

4.4 实例分析

在 Seekda 网站查找 weather 服务，返回 95 个服务，将前 30 个服务抓取下来(见表 2)，并注册到 JUDDI 中心。查询请求服务参数为城市名称(Cityname)，绑定支持 SOAP1.1 的服务，返回的 30 个服务都能够支持 SOAP1.1，但是服务的参数是多种多样的，有的是邮政编码(Zipcode)，有的是城市 ID(CityID)，有的是要求多个参数，比如城市名称、国家名称。与服务请求参数匹配的服务个数为 7(S3，S11，S12，S17，S18，S24，S25)，其中有 2 个服务是不可用的(S12，S18)，因此返回的服务中正确的查询个数为 5，故服务发现的可靠度 $R_{dis}=5/30=0.16667$。

表 2 返回的 weather 服务集

ID	服务名称	参数类型	$R_{provider}$	R
1	MeteorologyWS		0.997	0
2	GlobalWeather	Cityname，countryname	0.873	0
3	Weather	GetWeatherForCityState	0.993	0.99
4	GlobalWeather	Cityname，countryname	0.859	0
5	USWeather	ZipCode	0.875	0
6	USWeather	ZipCode	0.860	
7	GlobalWeatherService	GetLatestWeather	0.300	0
8	WeatherStationService	WMOGetAllWeather__StationsReques	0.997	0
9	MetarService		0.992	0
10	WeatherForecastService		0.995	0
11	Weather	Cityname	0.995	1
12	DASWorldWeather	Cityname	0.991	0
13	Weather	Zip	0.997	1
14	getweather	metar report	0.597	0
15	DASWorldWeather	Station	0.991	0
16	Weather	Zip	1.000	1
17	WeatherWebService	Cityname	0.954	1
18	WeatherService	City	0.967	0
19	Weather	CityID	0.890	0
20	WeatherService	CityID	0.913	0
21	WeatherWS	Type，time	0.997	1
22	WeatherService		0.989	0
23	WeatherBug_WebServices	Zipcode	1.000	1
24	WeatherWS	Cityname	0.936	1
25	Weather	Cityname	0.995	1
26	WeatherService		0.970	1
27	Weather	City	0.300	0
28	weather	Getweather	0.971	0
29	Weatherforcast	Citycode	0.995	0
30	WeatherService	Area	0.997	1

根据返回的结果集，假设可以采用不同的策略选择一个服务进行匹配，下面我们讨论不同选择策略下的 Web 服务的可靠性。

1. 基于关键字的服务发现

1)随机匹配：$R_{dis}=5/30=0.16667$，Web 服务的平均可靠性 $R=\frac{1}{30}\sum_{i=1}^{30}R_i=0.16239$。

2)按照参数匹配：$R_{dis}=5/7=0.714286$，Web 服务的可靠性 $R=\frac{1}{7}\sum_{i=1}^{7}R_i=0.69597$。

3)按照参数匹配并剔除无效的服务：$R_{dis}=1$，Web 服务的可靠性 $R=\frac{1}{5}\sum_{i=1}^{5}R_i=0.97436$。

2. 基于 QoS(可靠性)的服务发现

对返回的结果集，根据服务提供者提供的可靠性高低进行排序，进行服务选择时应用 3 种策略的 Web 服务发现的可靠度如下：

1)顺序匹配：前 8 个服务都是无效的，第 9 个服务与服务请求匹配。$R_{dis}=1/9=0.1111$，Web 服务的可靠性 $R=\frac{0.9945}{9}=0.1105$。

2)按照参数匹配：$R_{dis}=1$，Web 服务的可靠性 $R=0.9945$。

3)按照参数匹配并剔除无效的服务：$R_{dis}=1$，Web 服务的可靠性 $R=0.9945$。

从上述实验以及实例结果，我们可以看出提高 Web 服务的发现的可靠性，能够极大地提高 Web 服务的可靠性。

结束语 不同于一般的分布式系统，Web 服务通常都是位于不同的地址位置，服务使用者需要通过服务发现找到想要的服务，然后才能进行服务的调用。由于服务使用者和服务提供者往往不是同一个用户，而且服务使用者对查找到的服务不是非常了解，造成服务的失效要比一般分布式系统多的多，因此服务发现的过程在整个服务使用过程中显得非常重要，是影响服务可靠性的一个重要阶段。本文主要分析了服务发现过程中产生的失效因素对服务可靠性的影响，构建了新的服务可靠性模型，该模型和传统模型相比更加适用于 Web 服务。实验结果表明，在同一个环境的情况下，服务描述文件 WSDL 本身可能出现的错误会对服务的可靠度造成一定影响，根据假设为 0.75 至 0.95，而由于服务提供者未及时修改 UDDI 中心的信息所引起的错误，使得服务的查准率大幅下降，从而导致可靠度下降。同时，对于查询到的正确服务，会出现参数不匹配的情况，即用户输入的参数类型与服务要求的参数类型不一致或服务参数语义不匹配，导致服务发现失效，使得可靠度进一步下降。对于不同服务数时，服务错误集的扩大会导致各个商业实体中的服务的查准率进一步下降，从而导致服务可靠度也进一步下降，总体上从 0.584 降至 0.479 再降至 0.367。

利用 UDDI 中心的信息对服务预测的新的可靠度不仅可以对 UDDI 中心的可靠性做出评价，同时还可为用户选择服务以及评价服务提供者提供参考；而对每一个商业实体的可靠度预测则可以向 UDDI 中心提供有关服务提供者的可靠性信息，以便中心决定是否接受该提供者提供的服务。对这些信息进行分析，能让 UDDI 中心和服务提供者发现并改进自身的不足，从而得到长足的发展。

(下转第 231 页)

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

验证了本文提出的 MRQJ 算法在 SPARQL 查询语句查询变量多、连接复杂的情况下具备一定的优越性。本文在存储过程中尚未考虑 RDF 文件分割策略，在后续研究过程中重点研究文件分割方法，希望将关系紧密的 RDF 数据分割存储于同一个文件而提高检索效率。

参 考 文 献

[1] 李慧颖，瞿裕忠. 基于关键词的语义网数据查询研究综述[J]. 计算机科学，2011，38(7)：18-23

[2] 金强. 基于 Hase 的 RDF 存储系统的研究与设计[D]. 杭州：浙江大学，2011

[3] Li L，Song Y. Distributed Storage of Massive RDF Data Using HBase[J]. Journal of Communication and Computer，2011，8(5)：325-328

[4] Sun J，Jin Q. Scalable rdf store based on hbase and mapreduce[C]// 2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE). IEEE，2010：633-636

[5] Husain M F，Doshi P，Khan L，et al. Storage and retrieval of large rdf graph using hadoop and mapreduce[M]// Cloud Computing. Springer Berlin Heidelberg，2009：680-686

[6] Myung J，Yeon J，Lee S G. SPARQL Basie Graph Pattern Processing with Iterative MapReduce[C]// Proceedings of the Workshop on Massive Data Analytics on the Cloud(MDAC'10). 2010：6-12

[7] Husain M，McGlothlin J，Masud M M，et al. Heuristics-Based Query Processing for Large RDF Graphs Using Cloud Computing[J]. IEEE Transactions on Knowledge and Data Engineering，2011，23(9)：1312-1327

[8] Cheng J，Wang W，Gao R. Massive RDF Data Complicated Query Optimization Based on MapReduce[J]. Physics Procedia，2012，25：1414-1419

[9] Liu L，Yin J，Gao L. Efficient Social Network Data Query Processing on MapReduce[C]// Proc of the 5th ACM workshop. New York：ACM，2013：27-32

[10] 张伟奇，张坤龙. 基于关系型数据库的 RDF 存储引擎[D]. 天津：天津大学，2011

[11] 吴刚. RDF 图数据管理的关键技术研究[D]. 北京：清华大学，2008

[12] 刘翔宇，吴刚. 基于 Prüfer 序列的 RDF 数据索引与查询[J]. 计算机学报，2011，34(10)

[13] Dean J，Ghemawat S. MapReduce：simplified data processing on large clusters[J]. Communications of the ACM，2008，51(1)：107-113

(上接第 226 页)

参 考 文 献

[1] 郑啸，罗军周，曹玖新，等. 基于发布/订阅机制的 Web 服务 QoS 信息分发模型[J]. 计算机研究与发展，2010，47(6)：1088-1097

[2] AI-Masri E，Mahmoud Q H. Investigating Web services on the World Wide Web [C]// Proc of the 17th Int Conf on World Wide Web. New York：ACM，2008：795-803

[3] Wang Li-jun，Bai Xiao-ying，Zhou Li-zhu，et al. A Hierarchical Reliability Model of Service-Based Software System[C]// Proc of 33rd Annual International Computer Software and Applications Conference. Seattle，Washington，USA，2009：199-208

[4] Ren Ying-xin，Gu Qing，Qi Jing-xian，et al. Reliability Prediction of Web Service Composition Based on DTMC[C]// Proc of third IEEE International Conference on Secure Software Integration and Reliability Improvement. Shanghai，China，2009：369-375

[5] Ding Zuo-hua，Jiang Ming-yue，Abraham K. Port-Based Reliability Computing for Service Composition[J]. IEEE Transactions on Services Computing，2012，5(3)：422-436

[6] Tsai T，Zhang D，Chen Y，et al. A Software Reliability Model for Web Services[C]// Proc of 8th IASTED International Conference on Software Engineering and Applications. Cambridge，MA，2004：144-149

[7] Li Bi-xin，Fan Xiao-cong，Zhou Ying，et al. Evaluating the Reliability of Web Services Based on BPEL Code Structure Analysis and Runtime Information Capture[C]// Proc of of 17th Asia Pacific Software Engineering Conference. Sydney，Australia，2010：206-215

[8] Zheng Zi-bin，Wu Xin-miao，Zhang Yi-lei，et al. QoS Ranking Prediction for Cloud Services[J]. IEEE Transactions on Parallel and Distributed Systems，2013，24(6)：1213-1222

[9] Zheng Zi-bin，Lyu Michael R. Personalized Reliability Prediction of Web Services[J]. ACM Transaction on Software Engineering Methodol，2013，22(2)：1-25

[10] Zheng Zi-bin，Ma Hao，Lyu Michael R，et al. Collaborative Web Service QoS Prediction via Neighborhood Integrated Matrix Factorization[J]. IEEE Transaction on Services Computing，2013，6(3)：289-299

[11] Silic M，Delac G，Srbljic S. Prediction of atomic web services reliability based on K-means clustering [C]// Proc of 2013 9th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering. Saint Petersburg，Russian Federation，2013：70-80

[12] Bloomberg J. Web Services Testing：Beyond SOAP，Zap Think LLC，Sep[OL]. http://www.zapthink.com，2002

[13] Chaari S，Badr Y，Biennier F，et al. Framework for Web Service Selection Based on Non-Functional Properties[J]. International Journal of Web Services Practices，2008，3(1/2)：94-109

[14] Li Chang-bao，Cheng Bo，Chen Jun-liang，et al. A Discovery Approach for Web Services Composition Flows[C]// Proc of the 2010 IEEE Asia-Pacific Services Computing Conference. 2010：700-706

[15] Chan K S M，Bishop J，Steyn J，et al. A Fault Taxonomy for Web Service Composition[C]// Proc of the 3rd International Workshop on Engineering Service Oriented Applications (WESOA'07). Springer LNCS，2007：363-375

[16] Brüning S，Weißleder S，Malek M. A Fault Taxonomy for Service-Oriented Architecture[C]// Proc of the 10th IEEE High Assurance Systems Engineering Symposium. Plano，TX，2007：367-368

[17] Mallick S，Kushwaha D S. LWSDM：Layered Web Service Discovery Mechanism [J]. Advanced in Information Sciences and Service Sciences，2010，2(3)：25-31

[18] 陆文，徐峰，吕建. 一种开放环境下的软件可靠性评估方法[J]. 计算机学报，2010，33(3)：452-462

[19] Hwang S-Y，Lee C-H. Reliable Web service selection in choreographed environments[J]. Decision Support Systems，2013，54(3)：1463-1476

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net