第 36 卷 第 1 期 计 算 机 工 程 2010 年 1 月
Vol.36 No.1 Computer Engineering January 2010
·软件技术与数据库· 文章编号：1000—3428(2010)01—0070—03 文献标识码：A 中图分类号：TP393

基于服务替换的 Web Services 容错方法

刘 超，杨金民，张大方
(湖南大学软件学院，长沙 410082)

摘 要：现有网络服务(Web Services)容错方案存在资源开销大、与 Web Services 应用系统特性不一致等不足。针对该问题，根据 Web Services 应用系统特点提出基于服务替换的 Web Services 应用系统容错方法，其主要思想是在失效发生时使用网络中的等价服务替换失效服务。实验表明了该方法的可行性。

关键词：网络服务；容错；服务替换

Service Replacing-based Fault-tolerant Method for Web Services

LIU Chao, YANG Jin-min, ZHANG Da-fang
(Software School, Hunan University, Changsha 410082)

【Abstract】 Current Web Services fault-tolerant technologies cost too much resource due to their static configuration characteristics, and are inconsistent with the idiosyncrasy of Web Services systems. This paper proposes a fault-tolerant method employing service replacement strategy, which counts on the specialty of Web Services systems. The main idea of the strategy is replacing the invalid service with the equivalent service on Internet. The feasibility of this strategy is proved by experiment.

【Key words】 Web Services; fault tolerant; service replacing

1 概述

由于网络服务(Web Services)技术的平台无关性，它为实现各种不同信息系统之间的互操作提供了解决方案，为实现 Internet 上资源共享和充分利用提供了有效途径。Internet 上资源都通过 Web 服务方式提供给客户。随着人们对 Web 服务依赖程度的提高，其高可靠性和高可用性已成为人们关注的焦点。

Web Services 系统具有自包含(self-contained)、自描述(self-describing)、模块化和松耦合等特点，其提供的服务通过网络发布、查找和调用。Web Services 的这种动态和自适应特性[1]给其可用性带来了一些问题：Web Services 系统中各服务组件是自治的，使客户无法把握其提供的服务是否可用。当客户正要调用某一 Web 服务时，该服务可能在脱网维护。因此，客户调用服务是否成功除了受服务本身制约外，也受通信链路、中间代理和系统配置等因素的制约。

建立容错的 Web Services 系统是提高其可靠性和可用性的关键。已有 Web Services 容错方法主要集中在服务器容错方面，通过服务器冗余实现[2-3]。当服务失效发生时，使用冗余服务器响应客户请求，达到容错目的。例如文献[4]借鉴容错 CORBA 思想，提出基于服务法的容错 Web 服务体系结构。该体系结构扩展了 WSDL，引入了服务组的概念进行容错。在该体系结构中，客户端根据 WSDL 中注册的服务组信息依次访问服务组中的成员，直到发现不失效服务为止。文献[5]使用代理保证服务高可用，用户通过代理访问某个服务实例。代理能捕获服务不可用的异常，并将该信息转发给备份服务实例。

然而通过服务器冗余实施容错存在 3 个问题：资源开销大，无法容纳通信链路或中间代理等因素导致的失效，容错方式与 Web Services 系统特性不相适配。本文提出基于服务替换的 Web Services 容错方法。在服务执行前查询网络中与主服务等价的可用服务作为备份，当主服务失效时，使用该等价服务替换失效服务达到容错目的。

2 系统模型

实施服务替换需要解决 2 个问题：等价服务的获取和服务替换的实现。对于第 1 个问题，当前服务发现方面的研究成果已有很多，故在本文中不作赘述。对于第 2 个问题，Web Services 自身的互操作架构以及现有的 Web 服务组合技术为实施服务替换提供了基础。

在 Web Services 系统运行时，客户与服务器端会形成关联依赖关系。若服务在执行过程中失效，系统需要解除与该服务的关联依赖关系。对于备份服务，要求其与客户端建立连接时，状态能与系统达成一致并继续响应用户请求。因此，服务替换需要解决的问题包括以下 3 点：关联依赖关系的转化，执行的重定向和全局状态一致性的维护。本文通过建立容错代理、服务组和日志记录与恢复机制解决上述问题。

基于服务替换的容错 Web Services 框架(Service Replacing-based Fault-Tolerant Web Services framework, SRFTWS)模型如图 1 所示。

SRFTWS 框架包含 3 个模块：服务管理，失效管理以及日志和恢复管理。

(1)服务管理的主要部分是容错代理(Fault-Tolerant Agent, FT-Agent)。FT-Agent 提供一个接口用于接收服务组成员状态信息，根据该信息创建或删除成员，同时能够通过定位到不同主机的本地工厂(Local Factory, LF)创建服务。

作者简介：刘 超(1983－)，男，硕士研究生，主研方向：网络服务，软件容错；杨金民，副教授、博士；张大方，教授、博士、博士生导师

收稿日期：2009-11-05 **E-mail：**ilmyliu@163.com

(2)失效管理包括失效检测器(Fault Detector, FD)和失效通告器(Fault Notifier, FN)。FD 负责监听服务状态，FN 负责失效事件的传输。

(3)日志和恢复管理包括日志与恢复机制(Logging and Recovery Mechanism, L&RM)和请求监测器(Request Detector, RD)。RD 负责系统运行时请求和应答消息的监听，并将其转发给日志与恢复机制。系统正常运行时，L&RMS 将从 RD 获取的请求和应答消息记录到日志中。当失效发生时，L&RMS 调用日志对新成员状态进行恢复。

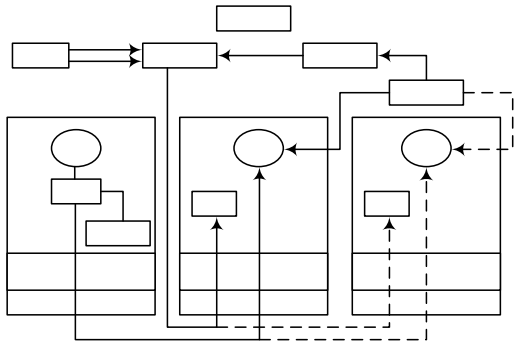

图 1　SRFTWS 容错框架

3　关键实现

3.1　服务部署

服务管理负责容错域中服务组成员的管理和服务失效状况的处理。服务部署过程主要由 FT-Agent 完成，流程如下：

(1)FT-Agent 通过用户应用(Application, AP)获取服务信息并创建服务组。

(2)FT-Agent 通过服务接口获取服务组成员信息，并向服务的 LF 请求创建服务。LF 在创建服务后将服务的 URL 反馈给 FT-Agent，由 FT-Agent 记录在服务组中。

(3)FT-Agent 向 FD 注册服务组成员，通过 FD 检测服务状态。

由于 WSDL 不支持 Web 服务组的描述，因此借鉴容错 CORBA 思想扩展了 WSDL，在 WSDL 中增加 WSG 标签以支持 Web 服务组的描述，例如：

```
< WSG name = sayhello>
< PRIMARY version = “1.0” location = “http://localhost: 8080/HelloService/SayHello”/>
<REPLICA version = “1.0” location = “http://210.43.109.76:8080/HelloService/SayHello”/>
</WSG>
```

该例定义了一个名为“SayHello”的 Web 服务组，该服务组包含 2 个部署在不同 Web 服务器上的服务，其中部署在 localhost 上的为主服务，另一个为备份服务，并提供了各服务的详细信息。

3.2　日志记录

由于通过网络获取的备份服务可能与主服务存在操作平台或编程语言等差异，在失效恢复时无法直接将主服务失效前的检查点状态覆盖到备份服务上，因此建立了一个服务状态信息组保存请求和应答序列。服务状态的定义如下：

定义　服务状态表示服务执行过程中请求与应答消息状态。表示为

service_state = {msg_requested,msg_state,responce}

其中，msg_requested 表示客户端请求；msg_state 表示请求的处理状态；response 表示服务器端响应的内容。

在系统正常运行时，L&RMS 将 RD 监听的客户端请求消息记录为 msg_requested，msg_state 为 FALSE；RD 收到服务端响应后，L&RMS 将 msg_state 改为 TRUE，并将响应内容记录为 response。在服务失效发生时，恢复机制向备份服务重播该服务状态信息组中的请求序列使其状态与失效前的主服务达成一致。

3.3　服务替换

在服务替换过程中，FT-Agent 需要处理 3 种类型的消息：FN 的服务失效通知，备份服务创建完成的响应，L&RMS 对备份服务进行状态一致性恢复完成的响应。为避免 RD 重复记录重播过程中的请求，把向 RD 更新服务信息设置在 L&RMS 对备份服务进行状态一致性恢复完成之后，使重播请求消息序列的过程对 RD 透明。服务替换流程如下：

(1)FT-Agent 监听是否有消息抵达。

(2)当有消息抵达时，判断是否 FN 服务失效通知。若是，则：

1)解析 WSG，获取下一个备份服务信息；

2)向备份服务 LF 发送创建服务请求。

(3)若否，则判断是否备份服务 LF 返回的应答消息。若是，则：

1)向 FD 更新服务信息，此后 FD 监测备份服务状态。

2)向 L&RMS 发送 state_recovery 消息告知备份服务信息。

3)L&RMS 从 state_recovery 中解析备份服务的信息：

①从 service_state 中获取请求序列 msg_requested；

②向备份服务重播 service_state 中所有 msg_state 为 TRUE 的请求；

③向 FT-Agent 返回应答；

④向备份服务发送 service_state 中 msg_state 为 FALSE 的请求。

(4)若否，则判断是否 L&RMS 返回的应答消息。若是，则向 RD 更新服务信息，RD 继续监听备份服务应答消息。

(5)FT-Agent 继续监听消息。

4　系统性能分析

为了对该容错系统的性能进行评估，使用了 2 台电脑作为主机，其配置如下：Intel Pentium4 2.4 GHz 主频，512 MB 内存，分别安装了 Windows2000 和 Linux 操作系统。在测试过程中，以简单的 hello 程序为例。用户向服务端发送 hello 请求，并附带自己的名字作为参数。服务端返回“hello”+用户名的响应。以上 2 台主机都提供该服务，使用 Windows2000 系统提供的服务作为主服务，Linux 系统提供的服务作为备份。

首先对服务失效情况做了测试。用户在 30 min 内对系统访问了 2 000 次。在该过程中，使失效按任意次序发生了 20 次。结果显示用户请求全部得到了正确的响应，证明了该方法的可行性。

由于绝大部分情况下系统处于无失效运行状态，失效带来的时间开销对请求的响应时间不会有很大影响，因此也对无失效的情况进行了测试。相对无容错的 Web Services 系统，容错的额外开销主要来自失效检测过程。所以对失效检测模块进行了测试，结果如图 2 所示。测试结果表明，失效检测对 CPU 使用率随检测时间间隔变化而改变，当检测时间间隔大于 2 s 时，CPU 使用率不再随着时间间隔增加而显著变化，可以说在一定程度上达到了恒定状态。也测试了失效检测时

间间隔的变化对失效处理时间的影响，结果如图 3 所示。测试结果表明，失效检测时间间隔对失效处理时间有显著影响，因此选择一个适当的检测时间间隔对提高容错系统性能有一定帮助。

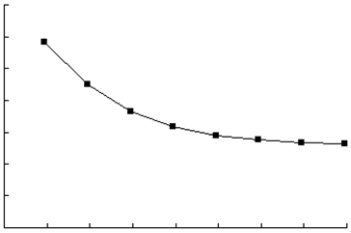

图 2　失效检测开销

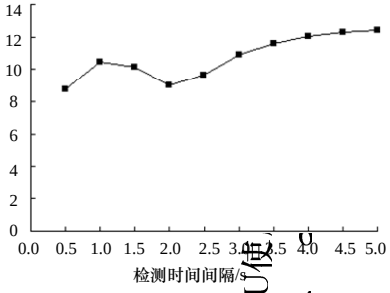

图 3　失效处理开销

由以上测试可以看出，容错模块给应用系统带来的开销是可以接受的。

5　结束语

本文提出基于服务替换的 Web Services 系统容错方法，通过建立与 Web Services 系统相一致的容错模型，达到降低系统资源开销、增大容错粒度的目的。在后续的工作中，将对日志与恢复模块进行改进，以进一步降低容错模块带来的时间开销。

参考文献

[1] Sycara K, Paolucci M, Ankolekar A. Automated Discovery, Interaction and Composition of Semantic Web Services[J]. Journal of Web Semantics, 2003, 1(1): 27-46.

[2] Diego Z, Maria B. A Fault Tolerant Web Service Architecture[C]// Proc. of Latin American Web Conference. Washington D. C., USA: IEEE Computer Society, 2007: 42-49.

[3] Liang D, Fang C, Chen C. Fault Tolerant Web Service[C]//Proc. of the 10th Asia-Pacific Software Engineering Conference. Washington D. C., USA: IEEE Computer Society, 2003: 310-319.

[4] Giuliana T, Lau C, Carlos M. FTWeb: A Fault Tolerant Infrastructure for Web Services[C]//Proc. of the 9th IEEE International EDOC Enterprise Computing Conference. Washington D. C., USA: IEEE Computer Society, 2005: 95-105.

[5] Alwagait E, Ghandeharizadeh S. DeW: A Dependable Web Services Framework[C]//Proc. of the 14th International Workshop on Research Issues on Data Engineering. Washington D. C., USA: IEEE Computer Society, 2004: 111-118.

编辑　任吉慧

（上接第 50 页）

的结果是一样的。以区分精度为启发函数的唯一约简结果是 110101111，并且该约简可以保证汽车数据集的近似精度和决策依赖区分精度不变。但是对于近似精度和决策依赖区分精度都乘以小于 1 的相同容错率系数，其约简结果不一样，见表 1。

表 1　约简结果比较

容错率系数	约简集 1	约简集 2
1.00	7	7
0.99	7	22
0.98	7	26
0.97	7	31
0.96	7	46
0.95	7	48
0.94	7	55
0.93	7	55
0.92	7	62
0.91	7	63
0.90	16	57

表 1 中，第 2 列表示以近似精度为启发函数对应的属性约简个数；第 3 列表示以决策依赖区分精度为启发函数对应的属性约简个数。从表中可以看出，当乘以不同容错率系数后以近似精度为启发函数的属性约简集变化具有明显的区间性，或者说乘以该区间内的不同容错率系数得到的属性约简不能够反映分类能力的变化，然而以决策依赖区分精度为启发函数的属性约简集随容错率系数的变化更明显、细致地反映了分类能力变化，因此，乘以小于 1 的容错率系数决策依赖区分精度能够更好地描述分类能力的变化。受篇幅所限，本文给出的区分精度、决策依赖区分精度和近似精度的其他一些比较结论在这里不一一验证。

6　结束语

基于对粗糙集属性约简理论方法的研究，本文提出了一个区分精度新概念模型，从而使区分能力有了一个定性定量的描述。通过决策表对区分能力和分类能力两者的关系进行研究，建立了相应的数学关系式，提出了决策依赖区分精度概念，为指导决策表的相对属性约简提供了一个新的判据，通过实验分析表明决策依赖区分精度与近似精度相比能够更细致客观地描述分类能力，同时给出了区分精度、近似精度和决策依赖区分精度在属性约简过程中相互关系的研究结论，从而为决策分析提供了新思路。本文的概念模型为粗糙集理论的进一步研究和实际应用提供了基础。

参考文献

[1] 徐　燕，怀进鹏，王兆其. 基于区分能力大小的启发式约简算法及其应用[J]. 计算机学报, 2003, 26(1): 97-103.

[2] Pawlak Z. Rough Set Approach to Multi-attribute Decision Analysis[J]. European Journal of Operational Research, 1994, 72(5): 443-459.

[3] Ruskey F. Combinatorial Generation(Working Version)[D]. Victoria, Canada: University of Victoria, 2001.

[4] Guan J W, Bell D A. Rough Computational Methods for Information Systems[J]. Artificial Intelligence, 1998, 105(1/2): 77-103.

[5] 王　珏. Rough Set 约简与数据浓缩[J]. 高技术通讯, 1997, 7(11): 40-45.

编辑　张正兴