

基于控制结构的 Web 服务可靠性预测模型

谢春丽^{1,2} 王喜凤^{2,3}

(徐州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
(安徽工业大学计算机学院 马鞍山 243002)³

摘要 在 Web 服务开发过程中,为了创建满足特定应用需求的新服务,需要将不同地理位置、不同服务提供者提供的 Web 服务按照一定的结构关系组合起来。在服务组合过程中,首先要保证服务执行的可靠性,组合服务的可靠性是 Web 服务最重要的 QoS(Quality of Service)属性,是决定 Web 服务能否成功应用的关键。组合服务的可靠性不仅和服务之间的结构有关,而且和组合的条件也有一定的关系,因此传统的可靠性模型已经不再适用于 Web 服务。为了提高 Web 服务的可靠性预测精度,提出了一种基于控制结构的组合服务的可靠性模型,最后通过案例分析说明了该可靠性模型能够比较精确地预测服务可靠性。

关键词 控制结构,组合条件,可靠性预测,Web 服务

中图分类号 TP311 **文献标识码** A

Reliability Prediction Model for Web Services Based on Control-structure

XIE Chun-li^{1,2} WANG Xi-feng^{2,3}

(School of Computer Science & Technology, Xuzhou Normal University, Xuzhou 221116, China)¹

(School of Computer Science & Engineering, Southeast University, Nanjing 211189, China)²

(School of Computer Science, Anhui University of Technology, Ma'anshan 243002, China)³

Abstract In the development phase of Web services, the services located in different locations and provided by different organizations must be composed together in order to satisfy specific business requirements. The reliability must be ensured for composite services first. The reliability of composite services is the most important QoS(Quality of Service) which can determine the popularity of the services. The reliability of composite services is affected by the structure of services and by the composite conditions. So, the traditional models to predict the reliability are inapplicable to Web services. In this paper, to improve the accuracy of reliability prediction, a reliability prediction model for composite service was presented based on control-structure. Lastly, a case study was designed to support our reliability model. The experimental results show that our model can obtain better reliability prediction accuracy.

Keywords Control-structure, Composite condition, Reliability prediction, Web service

1 引言

Web 服务作为一种分布式应用程序,具有松耦合、平台无关等特性,可以通过网络将位于不同地理位置的服务组合起来,形成具有新业务含义的大粒度服务。Web 服务组合不仅有效降低了用户使用系统的复杂性,还大大提高了 Web 服务的复用性,因此成为开发分布式应用软件的核心方法和技术^[1]。在服务组合过程中,首先要保证组合服务的可靠性,服务的可靠性是决定 Web 服务能否成功应用的关键,但遗憾的是到目前为止对 Web 服务可靠性的研究还处于起始阶段。

目前,组合服务可靠性的预测主要有 3 种方法:Markov 模型^[2-5]、Petri 网^[6,7]和数据分析方法^[8]。Markov 模型使用控制流程图来描述软件的结构,并假设组件之间的转移具有 Markov 性,软件的结构可以采用离散时间马尔可夫链(DTMC)^[9-12]、连续时间马尔可夫链(CTMC)或者半马尔可夫过程(SMP)^[13]描述。Tsai 和 Grassi 等扩展了传统的 Markov 模型,使 Markov 链可以描述服务之间的结构关系^[2,3,14],Li

基于服务之间的结构关系建立模型,从而对组合服务进行可靠性分析^[15,16]。这些方法通常只考虑组合服务里的基本服务的失效以及服务间的结构,而 Web 服务一般由不同的服务提供者构建在不同的平台上,相互之间以一种松耦合的方式集成,服务之间按照一定的条件进行组合,我们称之为组合条件。条件的可靠性也会对组合服务造成一定的影响^[8],现有的可靠性模型基本都未考虑这一点,本文在组合条件的基础上提出了新的 Web 服务可靠性预测模型,不仅考虑服务之间的结构关系,而且考虑了组合条件的可靠性,使 Web 服务可靠性模型更加精确,更加符合实际应用环境。

2 组合服务的可靠性模型

2.1 基于控制结构的组合服务可靠性

BP EL 是 Web 服务常用的服务组合语言,用基本活动和结构化活动分别表示原子服务和组合服务。结构化活动用来表示活动之间的逻辑关系,规定了活动之间的执行顺序。WS-BPEL 2.0 定义了几种不同的控制流模式的结构化活动,

本文受徐州师范大学自然科学基金项目(10XLA12),安徽工业大学青年教师基金(QZ201015)资助。

谢春丽(1979—),女,讲师,主要研究方向为软件可靠性分析、Web 服务可靠性分析等,E-mail: xiechunti@xznu.edu.cn。

按一定的顺序执行的有 Sequence、If/Switch、While、Repeat-Until 和 ForEach 结构;并发和同步的有 Flow 结构。Li 简要讨论了这几种结构化活动,但没有考虑条件对可靠性的影响^[20]。本文选择用可靠性框图(RBD)表示组合服务之间的可靠性关系。下面详细讨论如何计算组合服务的可靠性,首先把条件也看成是一个原子服务;然后将结构化活动图转换为扩展的可靠性框图(ERBD);最后根据 ERBD 计算组合服务的可靠性。其中每个图中的上面一个图表示服务间的控制关系,下面一个图是经过转换的 ERBD。

①Sequence 结构

活动之间按顺序执行,因此当活动中任何一个活动出现故障时,Sequence 活动就失效了,其结构如图 1 所示,其可靠度表示为:

$$Rel_{sequence} = \prod_{i=1}^n Rel_i$$

式中, n 是 Sequence 活动包含的活动个数,可以是基本活动,也可以是包含其它活动的结构化活动。

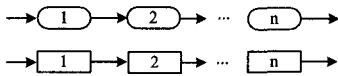


图 1 Sequence 结构和 ERBD

②If/Switch 结构

如图 2 所示,If/Switch 结构根据条件表达式的取值情况进行不同活动的选择。If 结构的可靠度可以表示为:

$$Rel_{if} = Rel_{con} * (P_{true} * Rel_{true} + P_{false} * Rel_{false})$$

式中, Rel_{con} 为条件的可靠度, Rel_{true} 与 Rel_{false} 分别是条件为真/假时执行的活动的可靠度,可以是基本活动也可以是结构化活动, P_{true} 与 P_{false} 分别是条件取真/假值的概率,且 $P_{true} + P_{false} = 1$ 。

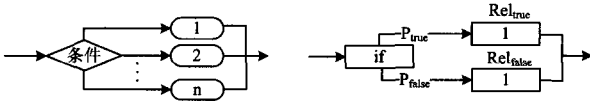


图 2 If 结构和 ERBD

Switch 结构的可靠度可以表示为:

$$Rel_{switch} = Rel_{con} * \sum (P_i * Rel_i)$$

式中, P_i 是每个条件取真值的概率,且 $\sum P_i = 1$ 。

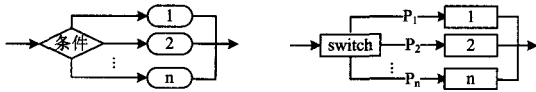


图 3 Switch 结构和 ERBD

③While 结构

While 结构在条件表达式为真值的情况下,反复执行某活动,直到条件不再满足为止。如图 4 所示,其可靠度可以表示为:

$$Rel_{while} = \sum_{i=0}^{\infty} (P * Rel_{con} * Rel_{activity})^i * (1-P) * Rel_{con} = \frac{(1-P) * Rel_{con}}{1 - P * Rel_{con} * Rel_{activity}}$$

式中, P 是条件取真值的概率。

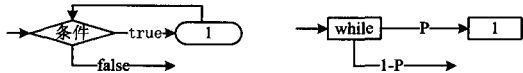


图 4 While 结构和 ERBD

④RepeatUntil 结构

RepeatUntil 结构反复执行某一活动,直到指定的条件为

真为止。RepeatUntil 与 While 之不同在于 RepeatUntil 循环至少执行包含的活动一次,而 While 结构有可能活动一次也未执行,如图 5 所示,其可靠度可以表示为:

$$Rel_{repeatuntil} = \sum_{i=0}^{\infty} ((1-P) * Rel_{con} * Rel_{activity})^i * P * Rel_{con} * Rel_{activity} = \frac{P * Rel_{con} * Rel_{activity}}{1 - (1-P) * Rel_{con} * Rel_{activity}}$$

式中, P 是条件为真的概率。



图 5 RepeatUntil 结构和 ERBD

⑤ForEach 结构

ForEach 结构执行它包含的 Scope 活动 N 次,如图 6 所示,其可靠度可以表示为:

$$Rel_{foreach} = Rel^N$$

式中, N 是整数,表示活动执行次数。

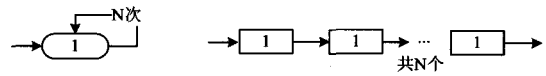


图 6 ForEach 结构和 ERBD

⑥Flow 结构

Flow 结构定义了一组可以并发执行的活动,当这些活动都执行完后,该并行活动就结束了。Flow 结构也可以包含非常复杂的结构,下面分两种情况讨论,一种是 Flow 定义的活动之间完全没有依赖关系,这种情形下,Flow 结构表示一般的并行关系,可靠度可以表示为:

$$Rel_{flow} = \prod Rel_i$$

第二种情况,Flow 语句用 Link 定义元素间的控制流关系,包含以下控制流关系:与分支(AND-SPLIT)、异或分支(XOR-SPLIT)、与归并(AND-JOIN)、异或归并(XOR-JOIN),如图 7 所示。

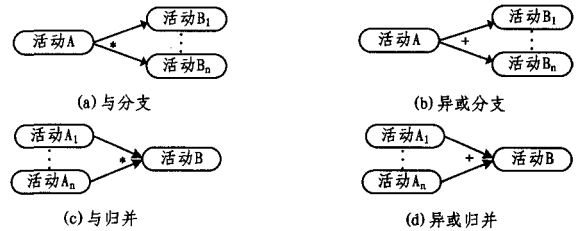


图 7 活动之间的控制结构

由 Link 链接的活动需要考虑以下两因素,首先考虑是否有链接,由 source 元素端的 TransitionCondition 属性(true/false)决定;其次考虑是否被激活,由目标端的 JoinCondition 属性(true)决定,BEPL 规定如果活动图中的某个活动无法被激活,则用一个默认的空活动代替该活动执行,这样保证了控制流不会因为某个无法激活的活动而终止运行。图 7 中首先检查活动 A 和 B_i 之间是否有链接,如果有链接再检查活动 B_i 是否可以被激活,如果 B_i 可以被激活,才执行活动 B_i ,否则 B_i 不被执行,只有被执行的 B_i 才对 Flow 结构的可靠性有影响。因此含有控制流关系的活动的可靠性估计在所有被激活的 B_i 中进行计算,没有激活的活动 B_i 则不参加计算,默认没有被激活的活动 B_i 的可靠度为 1。在讨论包含各种基本控制流关系的 Flow 结构之前,首先定义一些函数:

$Link(A, B)$: 布尔函数, 表示活动 A 和 B 之间的链接状态是肯定还是否定, 若链接处于肯定状态, 则 $Link(A, B) = 1$, 否则为 0。

$Active(B_i)$: 布尔函数, 表示活动 B 是否能被激活, 若 $Active(B) = 1$ 表示可以被激活, 否则表示不能被激活。

$$V(x) = \begin{cases} x, & x \neq 0 \\ 1, & x = 0 \end{cases}$$

式中, $V(x)$ 函数用来表示当某一活动无法被激活时, 其可靠度设为 1, 如果可以被激活, 为其本身的可靠度。

a) 与分支 (AND-SPLIT)

如图 7(a) 所示, 活动 A 执行完成后, 并行执行活动 $B_1, \dots, B_n$, 所有 $B_1, \dots, B_n$ 活动都执行完后, Flow 结构活动才完成, 与分支的可靠度表示为:

$$Rel_{flow_AND_SPLIT} = Rel_A * \prod_{i=1}^n V(Rel(B_i))$$

式中, $Rel(B_i) = Link(A, B_i) * Active(B_i) * Rel_{B_i}$, $Rel(B_i)$ 和 Rel_{B_i} 是两个不同的可靠度表达式, Rel_{B_i} 是活动 B_i 固有的可靠度, 而 $Rel(B_i)$ 是在与分支中 B_i 的可靠度, 受到 Link 元素定义的影响。

b) 异或分支 (XOR-SPLIT)

如图 7(b) 所示, 当活动 A 执行完成后, 根据条件选择执行 $B_1, \dots, B_n$, 只要有一个满足条件的活动被执行, 其它活动就不再被执行, 满足条件的活动执行结束, Flow 结构活动就完成了。首先检查是否被激活, 不能被激活的活动不在考虑范围内, 异或分支的可靠度表示为:

$$Rel_{flow_XOR_SPLIT} = Rel_A * \sum_{i=1}^n (P_i' * Rel_{B_i} * Rel_{con_i})$$

式中,

$$P_i' = \alpha * P_i,$$

$$\alpha = 1 / \sum_{i=1}^n (P_i * Link(i) * Active(B_i))$$

P_i 是分支 i 被执行的概率, 这里 α 是一个概率更改因子, 确保了修改后的 P_i' 依然满足 $\sum_{i=1}^K P_i' = 1$, $K = \sum (Link(j) * Active(B_j))$ 的性质, Rel_{con_i} 是或分支中条件的可靠度。

c) 与归并 (AND-JOIN)

如图 7(c) 所示, 所有经过与归并的活动都完成后转移到下一活动, 与归并通常用来表示同步。当 $A_1, \dots, A_n$ 都执行结束后, 才执行活动 B 。

$$Rel_{flow_AND_JOIN} = \prod_{i=1}^n V(Rel_{A_i} * Link(i) * V(Active'(B))) * Rel_B$$

式中, $Active'(B) = \prod_{i=1}^n Link(i)$ 用来判断所有和 B 相连的链接是否有效。

d) 异或归并 (XOR-JOIN)

如图 7(d) 所示, 当 $A_1, \dots, A_n$ 中有一个活动执行结束后, 立即触发活动 B 的执行, 所有其它的活动不再触发 B 的执行。异或归并的可靠度可以表示为:

$$Rel_{flow_XOR_JOIN} = \sum_{i=1}^n (P_i' * Rel_{A_i} * Rel_{con_i}) * V(Active''(B)) * Rel_B$$

$$P_i' = \alpha * P_i, \alpha = 1 / \sum_{i=1}^n (P_i * Link(i))$$

$$Active''(B) = \prod_{i=1}^n Link(i)$$

2.2 Web 服务系统的可靠性

Web 服务系统可以看作是很多原子服务或者组合服务

的工作流过程, 任何一个 Web 服务系统都可以分解成很多组合服务, 由这些组合服务按顺序结构组成。因此 Web 服务系统的可靠性分为以下几步:

- 1) 按控制结构将 Web 服务系统的工作流程划分为小粒度的组合服务;
- 2) 按照上节的计算模型计算划分出的组合服务的可靠度;
- 3) 根据顺序结构计算整个服务系统的可靠性。

3 实验分析

3.1 实验对象模型

以一个在线商品的购买服务系统为例来说明本文提出的可靠性模型。首先用户登录网站, 选择自己所需要的商品, 可以重复选择直到满意为止, 用户选择好预购买商品后, 列出商品列表, 生成订单同时进行在线支付, 然后生成包含用户信息的订单明细并根据客户要求选择快递或者邮寄方式送货。将一些具有简单结构关系的服务合在一起, 比如, 具有顺序关系的查询商品和选择商品根据顺序结构的可靠性计算原理组合成一个活动, 如图 8 中的活动 2, 在此直接给出服务系统的 ERBD。

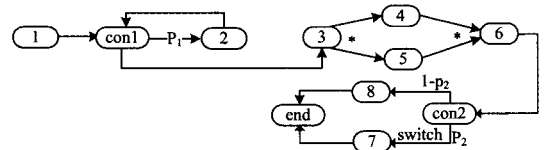


图 8 在线商品的购买服务

根据本文提出的 Web 服务系统可靠度模型, 本案例的可靠度公式可以表示为:

$$Rel_{pre} = R_1 \frac{(1-P_1)R_{con1}}{1-P_1R_{con1}R_2} R_3 R_4 R_5 R_6 R_{con2} [P_2 R_7 + (1-P_2) R_8]$$

如果不考虑条件可靠性, 上述公式可以表示为:

$$Rel_{pre} = R_1 \frac{(1-P_1)}{1-P_1R_2} R_3 R_4 R_5 R_6 [P_2 R_7 + (1-P_2) R_8]$$

3.2 实验设置

对上述案例进行仿真实验, 仿真中用变量 flag 记录组合服务是否失效, 实验中共仿真 $N = 1000$ 次, 变量 $failure_count$ 用来记录失效次数, 则可可靠度为:

$$Rel_{sim} = 1 - failure_count / N$$

为了提高仿真精度, 共仿真得到 100 个 Rel_{sim} , 然后取其均值作为实验结果。案例中所用的原子服务的可靠度和条件的可靠度见表 1 和表 2。

表 1 原子服务的可靠度

Service	Rel	Service	Rel
R ₁	0.95	R ₅	0.95
R ₂	0.98	R ₆	0.98
R ₃	0.9	R ₇	0.99
R ₄	0.97	R ₈	0.98

表 2 条件的可靠度

Condition	Rel	Probability
CON1	0.98	0.6
CON2	0.98	0.5

3.3 实验结果分析

实验一 预测精度

实验中采用绝对误差(AE)和相对误差(RE)作为预测精度的度量标准,AE和RE分别被定义为:

$$AE = |Rel_{pre} - Rel_{sim}|$$

$$RE = \frac{|Rel_{pre} - Rel_{sim}|}{Rel_{sim}}$$

在表1的设置下,由仿真实验得到的案例的可靠度均值 $Rel_{sim}=0.6912$,根据公式得到的可靠度预测值为 $Rel_{pre}=0.689$,绝对误差 $AE=0.0022$,相对误差 $RE=0.00318$ 。表3中分别给出了条件1和条件2在不同的可靠度下案例预测的结果和仿真结果,从表中可以看出,可靠性预测结果和仿真结果之间的误差是比较小的,由此可以证明本文可靠性模型的有效性,从结果可以看出,条件的可靠性对整个组合服务的可靠性有较大影响,是必须要考虑的因素。

表3(a) 预测结果比较(CON2=1)

CON1	Rel _{sim}	Rel _{pre}	AE	RE
0.1	0.031	0.032	0.001	0.043
0.2	0.072	0.069	0.003	0.042
0.3	0.110	0.111	0.001	0.007
0.4	0.161	0.159	0.002	0.012
0.5	0.207	0.215	0.008	0.041
0.6	0.265	0.282	0.017	0.064
0.7	0.352	0.362	0.010	0.028
0.8	0.473	0.460	0.013	0.050
0.9	0.612	0.582	0.030	0.013

表3(b) 预测结果比较(CON1=1)

CON2	Rel _{sim}	Rel _{pre}	AE	RE
0.1	0.077	0.074	0.003	0.041
0.2	0.143	0.148	0.005	0.033
0.3	0.224	0.222	0.002	0.011
0.4	0.322	0.295	0.027	0.083
0.5	0.378	0.369	0.009	0.023
0.6	0.457	0.443	0.014	0.031
0.7	0.627	0.517	0.110	0.176
0.8	0.581	0.591	0.014	0.017
0.9	0.679	0.665	0.014	0.021

实验二 条件可靠性的灵敏度分析

主要分析条件可靠性取不同值时对整个组合服务的影响, $Rel_{con} \in [0,1]$,初值取0,以0.05的间隔递增,结果如图9所示。从图9可以看出条件2的可靠性比条件1对组合服务可靠性的影响更大,因此提高条件2的可靠性对提高整个组合服务的可靠性至关重要。

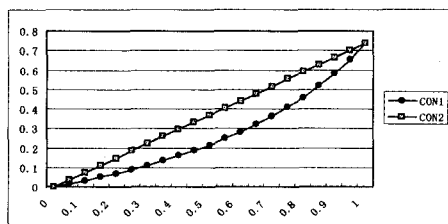


图9 条件可靠性的灵敏度分析

结束语 本文扩展了基于控制结构的组合服务的可靠性模型,不仅考虑了组合之间的结构关系,而且考虑了组合的条件对于可靠性的影响,因此可靠性模型更加精确。最后通过案例证明了组合条件对组合服务可靠性的影响,并做了灵敏度分析。从实验角度证明了我们提出的可靠性模型的有效性。下一步工作主要是对本文提出的可靠性模型在实际环境中进行验证,进一步分析影响Web服务可靠性的其它因素,

提高模型的精度,使其更好地适用于Web服务应用环境。

参考文献

- [1] 岳昆,王晓玲,周傲英. Web服务核心支撑技术:研究综述[J]. 软件学报,2004,15(3):428-442
- [2] Grassi V. Architecture-Based Reliability Prediction for Service-Oriented Computing[C]//Proceedings of the WADS workshop. 2004;279-299
- [3] Grassi V, Patella S. Reliability Prediction for Service-oriented Computing Environments[J]. IEEE Internet Computing, 2006, 10(3):43-49
- [4] 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,2009;199-208
- [5] 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. 2009;369-375
- [6] Guo Feng, Zhang Meng. Description and Analyzing the Reliability of Web Services Composition Based on Petri Nets[C]//Proc. of 1st International Conference on Information Science and Engineering (ICISE). 2009;5329-5332
- [7] Zhong Du-hang, Qi Zhi-chang. A Petri Net Based Approach for Reliability Prediction of Web Services[C]//Proc. of OTM 2006. Montpellier, France, 2006, LNCS 277. 2006;116-125
- [8] Zheng Zi-bin, Lyu M R. Collaborative Reliability Prediction for Service-Oriented Systems[C]//Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering. Cape Town, 2010;35-44
- [9] Gokhale S S. Architecture-Based Software Reliability Analysis: Overview and Limitations[J]. IEEE Transactions on Dependable and Secure Computing, 2007, 4(1):32-40
- [10] Cheung R C. A user-oriented software reliability model [J]. IEEE Trans. Softw. Eng. ,1980,6(2),118-125
- [11] Wang W L, Pan D, Chen M H. Architecture-based software reliability modeling[J]. The Journal of Systems and Software, 2006,79(1):132-146
- [12] Gokhale S S, Trivedi K S. Analytical models for architecture-based software reliability prediction: A unification framework [J]. IEEE Transactions on Reliability, 2006,55(4):578-590
- [13] Littlewood B. A Semi-Markov Model for Software Reliability with Failure Costs[C]//Proc. of Symp. Computer Software Eng. 1976;281-300
- [14] Tsai T, Zhang D, Chen Y, et al. A Software Reliability Model for Web Services[C]//Proceedings of 8th IASTED International Conference on Software Engineering and Applications. Cambridge, MA, 2004;144-149
- [15] Hansen K M, Brønsted J. Modeling Service Composition Reliability in Pervasive Computing, Science Institute, University of Iceland[R]. RH-02-2010. March 2010
- [16] Li Bi-xin, Su Zhi-yong, Zhou Ying, et al. User-oriented Web service reliability model[C]//Proceedings of IEEE International Conference on Systems, Man and Cybernetics, 2008;3612-3617