

ARCA: 一种应用可靠性及补偿代价敏感的 Web 服务选取方法

陈旺虎^{1,2,3} 李婧¹

(西北师范大学数学与信息科学学院 兰州 730070)¹

(中国科学院计算技术研究所网格与服务计算中心 北京 100190)² (中国科学院研究生院 北京 100039)³

摘要 提出了一种应用整体可靠性和补偿代价敏感的 Web 服务选取方法。该方法基于应用的逻辑特性来评估应用的交互点对于应用整体可靠性和补偿代价的影响,并获取相应的影响因子。在此基础上,结合反映用户偏好的服务质量向量的规约完成 Web 服务的选取。分析及实验结果表明,该方法从服务使用者和应用系统两个角度出发,在保证服务使用者偏好的基础上,可弥补现有方法在保证应用整体可靠性和降低系统补偿代价方面的欠缺。

关键词 服务选取, 可靠性, 补偿代价

ARCA: Approach to Application Reliabilities and Compensation Costs Aware Service Selection

CHEN Wang-hu^{1,2,3} LI Jing¹

(College of Mathematics and Information, Northwest Normal University, Lanzhou 730070, China)¹

(Research Center for Grid and Service Computing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China)²

(Graduate School of Chinese Academy of Sciences, Beijing 100039, China)³

Abstract An approach to service selection based on application reliabilities and compensation costs was proposed. The approach assesses the compact of each node in the application process to reliabilities and compensation costs of the application. The service selection is based on such compact factors and preferences of consumers to services expected. Analysis and experiments show that the proposed approach can sustain reliabilities and reduce compensation costs of the application when taking advantages of current service selection approaches in reflecting preferences of service consumers.

Keywords Service selection, Service reliability, Compensation costs

1 引言

随着服务计算^[1]技术的发展,通过 Web 服务装配应用的方法越来越受到关注。由于服务计算环境的开放性和动态性,为提高应用系统对于资源环境的适应能力,人们通常依赖业务过程描述语言来首先定义应用的业务逻辑,而在应用的执行过程中动态选取 Web 服务并绑定到应用流程,如图 1 所示。在服务的动态选取过程中,服务质量(QoSs)是非常重要的依据之一。当前,基于服务的 QoSs,从众多功能相同的 Web 服务中选取期望的服务已成为一个研究热点。

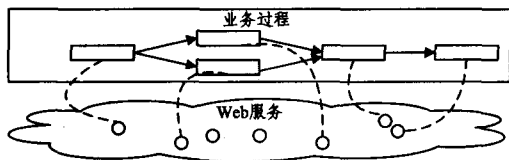


图 1 应用装配和 Web 服务选取

服务的 QoSs 通常指服务的价格、执行时间、有效性和可靠性等非功能属性^[2]。文献[3]认为服务的 QoSs 也可以包

含一些与领域相关的指标。当前,围绕服务的 QoSs,国内外在多个方面展开了研究。首先,一些研究关注服务 QoSs 的获取和管理方法^[4,5];另外一些研究从服务 QoSs 的表示和计算等方面来探讨服务 QoSs 的模型,例如文献[3,6]等提出的服务 QoSs 的表示和计算模型,以及文献[7]等提出的服务 QoS 的本体模型。另外,相当一部分研究关注基于 QoSs 的服务优化选取和个性化的服务选取方法^[8-10]。最后,文献[11-14]等提出了组合服务的 QoSs 的估算方法,并在此基础上探讨了 QoSs 相关的服务组合方法。

由于服务的 QoSs 对于服务使用者的感受往往有着直接的影响,在基于 QoSs 的 Web 服务优化选取方面的相关研究中,当前的方法多从服务使用者的偏好出发,对 Web 服务 QoSs 的各指标进行综合评估,以决定 Web 服务的合理选取方案^[8-10]。然而,应用系统的整体可靠性和应用系统的执行代价是构建应用必须关注的两个重要问题。事实上,上述两个方面也会间接影响到应用使用者的使用感受。

在基于 Web 服务装配应用的过程中,Web 服务的个体可靠性对于整个应用系统的整体可靠性有多大影响? 当该 Web 服务不能可靠地完成预期的功能时,可能导致系统付出

到稿日期:2008-07-15 返修日期:2008-09-26 本研究得到国家重点基础研究发展计划(2007CB310805)资助。

陈旺虎(1973—),男,博士生,讲师,主要研究方向为服务计算与软件集成,E-mail:chenwanghu@software.ict.ac.cn;李婧(1975—),女,硕士,讲师,主要研究方向为软件工程和服务工程。

的非完成业务功能的额外代价(我们称之为补偿代价)有多大?如何在反映服务使用者的偏好的同时,提高应用系统的整体可靠性并降低系统的补偿代价?这些问题在服务选取的过程中显然应该受到关注。然而,当前基于 QoSs 的 Web 服务选取方面的研究对于这些方面的关注程度还不多见。本文正是从上述角度出发,试图研究一种能够 Web 服务的选取方法,以弥补现有方法在保证应用整体可靠性以及降低系统的补偿代价方面的不足。

本文的主要贡献是针对目前 Web 服务选取主要关注请求偏好的不足,提出了一种新的服务选取方法,使得服务的选取既能保证服务使用者的偏好,又能保证应用整体可靠性的要求并降低系统的补偿代价。具体贡献包括:(1)提出了一种基于业务流程的逻辑结构衡量流程交互点(见第 2 节),对于应用系统整体可靠性的影响程度的策略,定义了应用可靠性的影响因子,并给出了其计算方法;(2)提出了基于业务流程的特性,衡量流程交互点对于应用系统补偿代价(见第 2 节)的影响程度的策略,定义了流程交互点对于应用系统补偿代价的影响因子,并给出了一种计算方法;(3)基于上述两点,提出了一种兼顾服务使用者的偏好,并且能够保证应用整体可靠性及降低系统补偿代价的 Web 服务选取方法。

本文第 2 节对所关心的问题及其相关的概念进行了精确的定义;第 3 节就交互点对应用整体可靠性和系统补偿代价的影响的评估方法,以及基于应用可靠性和补偿代价的服务选取方法进行了详细的论述;在第 4 节中,对所提出的方法进行了实验验证和分析;在第 5 节中,分析了当前基于 QoSs 的 Web 服务选取方面的相关研究工作;最后得出了本文提出的方法可保证应用整体可靠性及降低系统补偿代价的结论。

2 问题定义

为方便问题的讨论,首先给出一些主要概念的严格定义。

定义 1 将用以描述应用的流程逻辑但还没有绑定具体 Web 服务的业务过程称为应用装配模板,其每个待绑定到具体 Web 服务的结点称为交互点。

应用的装配模板可表示为一个二元组 $AT(G, T)$,其中, $G(V, E)$ 为一个有向图, T 为该应用的事务集合。 $G(V, E)$ 的结点集合 V 的每个元素对应一个交互点,而边集 E 的每个元素表示两个交互点之间的交互关系。 T 的每个元素对应一个包含在某个事务之中的交互点的序列。

定义 2 将交互点 I 不可靠时整个应用的不可靠概率称之为当前交互点 I 的可靠性对应用整体可靠性的影响因子,简称为交互点 I 的应用可靠性影响因子,记作 δ 。 δ 表示了当交互点选取的服务不可靠时,对应用整体可靠性的影响程度。

当交互点 I 所绑定的 Web 服务不可靠时,其行为将导致系统付出一定的非完成业务功能的额外代价,例如事务的撤销、送出的货物的收回等。

定义 3 当交互点 I 所绑定到的服务不可靠时,系统要付出的非处理正常业务功能的额外代价称为交互点补偿代价,记作 ϵ 。一个交互点的补偿代价相对整个系统的补偿代价的比值称为交互点的补偿代价影响因子,记作 λ 。

规则 1 假设服务 s 的质量向量表示为 $q = (q_1, q_2, \dots, q_n)$,如果根据应用使用者对于各服务质量指标 q_i 的偏好以及该服务所在交互点对于应用系统的可靠性和补偿代价的影

响,将服务 s 的质量规约为一个整数 d ,那么 d 的取值越大,则认为在当前环境下选取服务 s 的方案更优。

规则 1 表明,尽管服务的可靠性对于应用的整体可靠性和补偿代价具有很大的影响,但不能简单地选取可靠性最高的服务。因为,首先要考虑服务使用者对于各种 QoSs 指标的偏好,同时服务的质量指标之间往往会存在相互的制约,例如可靠性越高可能会导致服务的价格越高,而响应速度又可能会相对较慢。

3 应用可靠性和补偿代价敏感的服务选取方法

当前基于 QoSs 的服务选取方法广泛采用的一种思想是从服务使用者的角度或者 QoSs 指标全局优化的角度出发,却没有考虑选取的服务能否保证应用的整体可靠性,以及当服务不可靠时系统需要付出的补偿代价。本文提出的方法正是在现有研究的基础上,使得服务选取的过程对于应用可靠性和补偿代价具有敏感性。

3.1 估算交互点可靠性影响因子

因为服务绑定的滞后性,准确计算一个交互点的应用可靠性影响因子显然不够现实。但是,一个交互点在装配模板中的上下文环境,可以在一定程度上反映该交互点的可靠性对于应用整体可靠性的影响程度。

应用的执行过程可看作是从初始交互点开始,沿着一定的路径向终止交互点逼近的状态迁移过程。假设在应用装配模板 $AT(G, T)$ 中, G 的初始结点为 V_i ,终止结点为 V_e 。给定一个交互点 I, V_i 到 I 的所有可能路径数为 m, I 到 V_e 的所有可能路径数为 n, V_i 到 V_e 的所有可能路径数为 t ,可定义交互点 I 的可靠性影响因子为 $m * n / t$ 。根据定义,有 $0 < m * n / t \leq 1$,否则会有从 V_i 到 V_e 的路径数大于 t 的矛盾出现。实际上, m/n 反映了执行节点 I 的概率, n 反映了到达 I 后可能影响到的路径数。需要说明的是,该定义所基于的一个假设是对于每条路径的选择是完全随机的。

当由于应用的业务特性或者用户的使用兴趣导致应用的路径的执行概率可能不相等时,交互点的可靠性影响因子可表示为

$$\left(\sum_{i=1}^m p_i \sum_{j=1}^n q_j \right) / t$$

其中, p_i 表示到达 I 的每条路径的概率, q_j 表示 I 到达 V_e 的每条路径的执行概率。

算法 1 给出了在各路径访问概率相等的情况下,计算交互点 I 的可靠性影响因子的简要过程。其中函数 CountPath(X, Y) 计算从装配模板的 X 节点开始到 Y 节点结束的路径数。各访问路径非等概率情况下的可靠性影响因子的计算方法与算法 1 类似,在此不再赘述。

算法 1 AssessReliabilityFactor (I)

- 1) Pre = CountPath(S, I)
 - 2) Post = CountPath(I, E)
 - 3) RETURN (Pre * Post) / (Pre + Post)
- Function CountPath(S, V)
- 1) Count = 0
 - 2) If (S, V) in G, E THEN
 Count = Count + 1
 - 3) FOR EACH (V_1, V) IN G DO
 Count = Count + CountPath(S, V_1)

4) RETURN Count

在获取各交互点的可靠性影响因子之后,可以得出辅助服务选取的一条规则如下,以保证服务选取过程中对于应用系统整体可靠性的要求。

规则 2 假设当前交互点 I 的可靠性影响因子是 δ ,为当前交互点选取的服务的可靠性为 r ,应用的整体可靠性为 R ,则有 $(1-r)\delta=1-R$ 。

证明:因为 I 的不可靠的概率是 $(1-r)$,根据可靠性影响因子的定义,当 I 不可靠时应用不可靠的概率是 δ ,所以有当前应用的不可靠性是 $(1-r)\delta$,所以 $(1-r)\delta=1-R$ 。

规则 2 表明,当设交互点 I 的可靠性为 r ,应用的可靠性为 R ,由于 $(1-r)\delta=1-R$,因此如果需要保证应用的可靠性高于 v ,即 $R=1-(1-r)\delta \geq v$,则必须满足当前交互点所选取的服务的可靠性 $r \geq 1-(1-v)/\delta$ 。

3.2 评估交互点补偿代价影响因子

交互点的补偿代价反映了应用执行过程中该交互点出错使得系统需要额外付出的非完成业务功能的代价的程度。因此,交互点的补偿代价是一个相对的度量值。为系统的每个交互点定义这样一个度量值显然是比较困难的。然而,每个应用的特性决定了一些特殊的交互点的出错对于系统的补偿代价有着很大的影响,我们不妨称这些交互点为关键交互点,而为关键交互点定义补偿代价显然更为可行。另外,一些交互点从表面来看似乎不是关键点,但是我们发现交互点的补偿代价实际上会有一种传播效应。事务处理可以很好地解释这种传播效应的存在,尤其是在服务计算这种松耦合的环境下,对于长事务的处理更是如此。

在本文中,我们的一个基本假设是:如果某个服务没有包含在任何事务之中,而该交互点又不是关键点,那么此服务的失效从系统的本身来讲是不需要付出补偿代价的,所以其补偿代价为 0。

因此,在对系统中的关键交互点进行补偿代价的标记后,可根据应用的逻辑特性,对相关的交互点的补偿代价以及交互点的补偿代价影响因子进行如下计算。

给定一个应用的装配模板 $AT(G, T)$,如果 $t=(t_1, t_2, \dots, t_n) \in T$ 表示了该应用中所存在的一个事务。那么当 t_i 出现错误时,系统需要对 t_1 到 t_{i-1} 进行撤销,从而付出相应的补偿代价 $\sum_{j=1}^{i-1} \epsilon_j$,其中 ϵ_j 表示 t_j 失效时的系统补偿代价。显然,对于事务 $t=(t_1, t_2, \dots, t_n)$,当 $i < j$ 时, $\epsilon_i < \epsilon_j$,这从一个侧面反映了补偿代价的传播效应。

当然,一个交互点可能出现在多个事务中,假设交互点 i 出现在 n 个事务当中,那么该交互点的补偿代价为 $\epsilon_i = \sum_{j=1}^n p_j \epsilon_j$,其中 p_j 表示执行各事务的概率,在随机情况下 $p_i = 1/n$ 。

在此基础上,交互点 t_i 的补偿代价影响因子可表示为

$$\lambda_i = \epsilon_i / \sum_{j=1}^n \epsilon_j$$

那么,交互点 i 的当前服务的补偿代价影响因子可表示为

$$\lambda_i' = (1-r_i)\lambda_i \quad (1)$$

其中, r_i 表示该交互点当前候选服务的可靠性, λ_i' 则反映了该服务不可靠是系统可能要付出的补偿代价的程度。

需要说明的是,由于补偿代价因子与可靠性影响因子一样,均是针对应用的转配模板进行定义的,对于一个应用只需要一次计算,因此这种方法在实际应用中是可行的。

综上所述,通过应用装配模板的特性的分析,可以评估出一个交互点对于应用的整体可靠性和补偿代价的影响因子,从而为服务的选取策略提供了新的参考因素。

3.3 应用可靠性和补偿代价敏感的 Web 服务选取方法

图 2 表示了采用本文所提出的方法 (ARCA) 进行 Web 服务选取的过程。首先,为保证应用的可靠性,需要根据规则 2 对不满足可靠性要求的服务进行过滤。然后,结合服务的补偿代价影响因子和服务使用者的偏好对服务的 QoS 向量进行规约,以实现服务 QoS 的定量比较。最后,根据 QoS 的最终规约结果对候选服务进行排序。

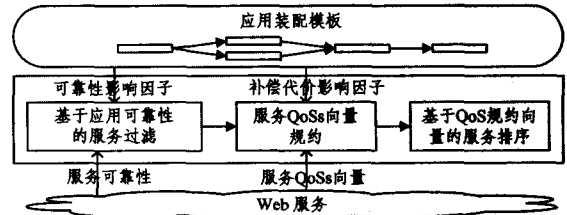


图 2 ARCA 的服务选取过程

在服务的选取过程中,在服务 QoS 向量结合服务使用者的偏好的规约方法上,我们借鉴了文献[3]提出的方法。同时,本文从服务的可靠性对于应用的整体可靠性以及系统补偿代价的影响角度,对服务 QoS 的规约方法进行了完善。基于该方法,服务 QoS 不但可以反映服务对于使用者的影响,同时能反映其对于应用系统本身的影响。

令矩阵 Q 表示 n 个 Web 服务在 m 个 QoS 指标上的取值;向量 N 表示某个 QoS 指标的取值的增加对于用户利益的影响, N 的分量取值为 1 表示用户受益,取值为 0 表示用户的利益会受到负面的影响;向量 C 表示对某个 QoS 指标进行规约时的最大值。则

$$Q = \begin{pmatrix} q_{1,1} & q_{1,2} & \dots & q_{1,m} \\ q_{2,1} & q_{2,2} & \dots & q_{2,m} \\ \dots & \dots & \dots & \dots \\ q_{n,1} & q_{n,2} & \dots & q_{n,m} \end{pmatrix}$$

$$N = (n_1, n_2, \dots, n_m)$$

$$C = (c_1, c_2, \dots, c_m)$$

令 $(A = \sum_{i=1}^n q_{i,j})/n$ 表示各服务的第 j 个 QoS 指标当前值的平均值,可根据下面的表达式对 Q 进行规约^[3]:

$$v_{i,j} = \begin{cases} q_{i,j}/A & A \neq 0 \text{ and } q_{i,j}/A < c_j \text{ and } n_j = 1 \\ c_j & A = 0 \text{ and } n_j = 1 \text{ or } q_{i,j}/A \geq c_j \\ A/q_{i,j} & q_{i,j} \neq 0 \text{ and } n_j = 0 \text{ and } A/q_{i,j} < c_j \\ c_j & q_{i,j} = 0 \text{ and } n_j = 0 \text{ or } A/q_{i,j} \geq c_j \end{cases}$$

Q' 表示对服务 Q 进行第一步规约后的结果矩阵, Q' 对 Web 服务 QoS 的取值对于用户的影响进行了一种定量的表示。

$$Q' = \begin{pmatrix} v_{1,1} & v_{1,2} & \dots & v_{1,m} \\ v_{2,1} & v_{2,2} & \dots & v_{2,m} \\ \dots & \dots & \dots & \dots \\ v_{n,1} & v_{n,2} & \dots & v_{n,m} \end{pmatrix}$$

假设向量 $F = (f_1, f_2, \dots, f_m)$ 表示服务 QoS 指标 i 对于服务使用者的影响权重(从另一个角度反映了用户的使用偏好),则借鉴文献[3]中的方法,服务 i 的 QoS 最终可规约为

$$QoS(s_i) = \sum_{j=1}^m v_{i,j} \times f_j \quad (2)$$

在实际应用中,可以对服务 QoS 指标进行分组,通过考虑各分组内的 QoS 指标对于服务使用者的影响程度对 QoS 向量进行两次规约。这样做的好处是能够适应服务 QoS 指标的动态变化^[3]。本文中,在不影响问题讨论的前提下,我们对该方法进行了简化,没有考虑 QoS 的分组情况。

然而,考虑到服务 QoS 对于应用的补偿代价的影响,本文对文献[3]中服务 i 的 QoS 值的规约方法进行如下改进:

$$QoS(s_i) = (\sum_{j=1}^n v_{i,j} \times f_j) \times W_1 \times \lambda_i \times W_2$$

结合上面的分析,设 G 为当前应用的装配模板, S 为所发布服务的集合,为交互点 I 选取服务的算法表示如下:

算法 2 ARCA (G, I, S)

- 1) $\delta = \text{AssessReliableFactor}(G, I)$
- 2) $\Lambda = \text{AssessCompensationFactors}(I, S)$, 其中 $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_n)$
- 3) FOR EACH $s \in S$ DO BEGIN
 - IF $s.r < 1 - (1 - v) / I.\delta$
 - THEN BEGIN
 - $S = S - \{s\}$
 - Update(Λ)
- END
- 4) $QoS = \text{NormalizingQoSs}(S)$, 其中 $QoS = \{q_1, q_2, \dots, q_n\}$
- 5) IF $\text{HasInitialized}(W_1, W_2) = \text{false}$
- THEN $W_1 = W_2 = 1/2$
- 6) FOR EACH $s_i \in S$ DO
 - $q_i = q_i * W_1 - \lambda_i * W_2$
- 7) $S = \text{RankingDescend}(S, QoS)$
- 8) RETURN s_1

在算法 2 中,语句 1 和语句 2 分别计算交互点 I 的应用可靠性影响因子 δ 以及各服务的应用补偿代价影响因子,具体计算方法见算法 1 和 3.2 节中的式(1)。语句 3 根据规则 2 对不满足应用整体可靠性要求的候选服务进行过滤。语句 4 产生服务 QoS 的规约向量,具体方法见 3.3 节中的式(2),在此过程中充分考虑了用户对服务质量的偏好。语句 5 和语句 6 结合应用的补偿代价影响因子对服务的规约向量进行修正。语句 7 和语句 8 根据最终的 QoS 规约向量,对候选服务以降序进行排序,并返回 QoS 值最大的服务作为当前交互的最优选取方案。

图 2 和算法 2 表明,ARCA 方法在反映服务选取的质量指标偏好的同时,可保证应用在整体上的可靠性要求,并在此基础上尽可能降低系统的补偿代价。ARCA 算法的时间复杂度是多项式。对于一个应用,应用可靠性和系统补偿代价影响因子均只需计算一次。

4 实验仿真和分析

本文以一个简化的旅游预定应用为例,对 ARCA 方法进行仿真验证。其应用装配模板共有 5 个交互点,分别对应天气查询、旅游线路查询、车票预定、宾馆预定和车票查询业务,如图 3 所示。

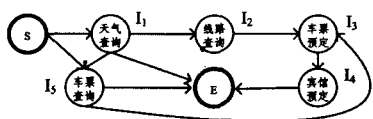


图 3 购物应用的装配模板

由于当前在服务选取方面尚没有公认测试集,因此在

实验过程中为每个交互点提供 100 个模拟 Web 服务以供选取。所考虑的 QoS 指标包括可靠性、价格和响应时间。为增加试验的真实性, QoS 指标均为随机产生。

假设系统的可靠性要求保证在 0.95 以上,则在等概率访问情况下,各交互点的可靠性影响因子 δ 、交互点的候选服务的最低可靠性要求以及按照可靠性过滤的候选服务的个数 filtered 如表 1 所列。表 2 表示了 6 条访问路径的概率分别为 0.05, 0.05, 0.1, 0.2, 0.3 和 0.3 时各交互点的对应情况。这 6 条路径分别为 $I_1, I_1 I_5, I_5, I_5 I_3 I_4, I_1 I_2 I_3 I_4, I_1 I_5 I_3 I_4$ 。在访问路径不等概率的情况下,为简化实际计算过程,交互点的可靠性影响因子采用 $(\sum_{i=1}^n p_i')$, 其中 p_i' 表示包含交互点 I 的第 i 条路径的执行概率。

表 1 等概率情况下的交互点情况分析

	I_1	I_2	I_3	I_4	I_5
δ	0.670	0.170	0.50	0.50	0.670
$\min(r)$	0.925	0.706	0.900	0.900	0.925
Filtered	70	21	45	74	55

表 2 非等概率情况下的交互点情况分析

	I_1	I_2	I_3	I_4	I_5
δ	0.800	0.300	0.800	0.800	0.600
$\min(r)$	0.938	0.833	0.938	0.938	0.917
Filtered	72	18	69	94	53

表 1 和表 2 反映出,如果考虑到候选服务对于应用整体可靠性有影响,在服务选取的过程中,可以通过过滤不满足要求的服务来保持系统性能,同时能提高服务选取过程的效率。另外,对比表 1 和表 2 可以发现,考虑不同路径的访问概率时,同一个交互点的候选服务的可靠性要求可能出现比较大的差异。这点对于实际应用是非常有用的,因为一个应用系统对应的业务往往决定了一些功能会是系统的关键部分。需要说明的是,由于可靠性的影响因子仅由装配模板确定,因此在实际应用中该方法是可行的。

图 4 和图 5 是对两种情况下可靠性影响因子和最低可靠性要求的对比分析。图 6 是对两种情况下,由于候选服务的可靠性不满足应用的整体可靠性的要求而过滤掉的服务数量的对比分析。

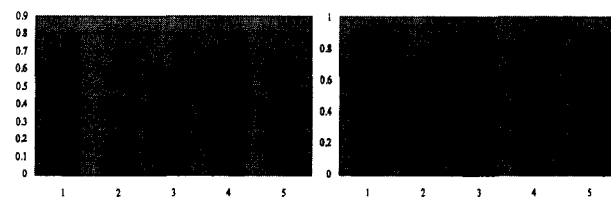


图 4 交互点可靠性影响因子分析

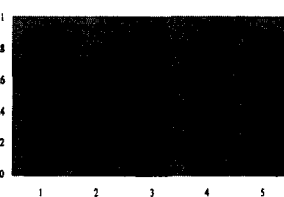


图 5 交互点候选服务可靠性分析

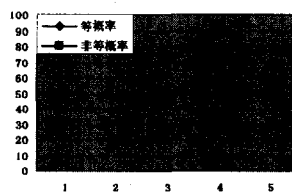


图 6 基于可靠性影响因子的服务过滤结果

在该应用中,由于 I_3 和 I_4 两个服务的出错会导致退票和退宾馆的两种补救措施的执行,因此从系统补偿代价的角

度来看这两个交互点均是系统的关键点。同时 I_4 对应操作的撤销往往会导致 I_3 对应操作的撤销,所以 I_3 和 I_4 处于一个事务中。我们指定 I_3 和 I_4 的撤销将分别导致系统付出的补偿代价的度量值为 4 和 5(因为在本实验中, QoS 的规约的最大值选取为 5,而该度量值主要反映各交互点的补偿代价之间的对比程度),那么根据补偿代价影响因子的计算方法,可得 $\lambda_3 = 4/9, \lambda_4 = (4+5)/9 = 1$ 。

表 3 选取了交互点 I_4 的候选服务中满足应用整体可靠性要求的 5 个候选服务,以说明考虑应用补偿代价影响因子对于服务选取结果的影响。Q 为按照传统 QoS 规约方法得到的服务质量值。服务质量向量被规约为小于等于 5 的整数。R 表示当前所选服务的可靠性。SQ 表示按照 ARCA 算法所得到的服务质量值。P₀ 和 P 分别表示按照 Q 和 SQ 的服务选取优先级。结果表明,尽管 S_2 的 Q 值高于 S_3 和 S_4 ,但是因为该交互点的不可靠可能导致较高的补偿代价,所以最终的优先级却是 S_3, S_4, S_2 ,其中 $W_1 = 0.4, W_2 = 0.6$,而且 I_4 的补偿代价因子为 5(补偿代价被规约为不大于 5 的整数)。各交互点因系统补偿代价因子而需对原服务选取结果的优先级进行改变的次数如图 7 所示。

表 3 可靠性满足要求的 I_4 的候选服务

	Q	R	SQ	P ₀	P
S_1	4.75	0.95	1.75	1	1
S_2	4.5	0.95	1.65	2	4
S_3	4.3	0.99	1.69	3	2
S_4	4.3	0.98	1.66	4	3
S_5	3.28	0.97	1.22	5	5

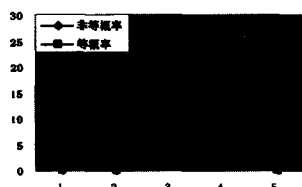


图 7 补偿代价对服务选取结果的影响

实验结果表明,本文所提出的服务选取方法在反映了请求者对于各服务质量指标的偏好的同时,能够满足应用的可靠性对于候选服务可靠性的要求,并降低应用的补偿代价。

5 相关工作分析

文献[6,13]从不同程度上讨论了 Web 服务 QoS 模型应该关心的因素,并分别就一些因素的计算方法进行了探讨。文献[4]提出了一种分为系统层、服务层、业务层的 QoS 模型。文献[12]提出了一种支持组合服务选取的 QoS 模型。文献[3]提出了一种用 Web 服务选取的开放、公平和动态的 QoS 计算模型,以及一种 QoS 向量的规约方法。其规约方法在本文中进行了借鉴,但是该研究不关注服务 QoS 对于应用系统的影响。文献[8-10,13]等研究了如何通过综合评估服务的可靠性、价格、响应速度等多种服务质量指标来进行服务的最优选取,其整体研究思路与文献[3]的思路相似。文献[15]将服务动态选择全局最优化问题转化为一个带 QoS 约束的多目标服务组合优化问题,利用多目标遗传算法的智能优化原理,通过同时优化多个目标函数,最终产生一组满足约束条件的优化服务聚合流程集。文献[16]综合了服务的功能和质量属性来进行服务的选取。文献[17]提出了一种不确定性敏感的服务 QoS 匹配模型,描述了 QoS 属性在不同条件

下的不确定性,并提出 QoS 的判定模型。文献[11,12,14,18,19]提出了计算组合服务 QoS 的方法,以及服务选取的方法和在此基础上服务组合的方法。文献[20]提出一种基于 QoS 的服务选取方法,基于 WSMO 定义了一种 QoS 本体,并提出了一个公平的动态服务选取机制。与上述研究相比,本文提出的方法从应用的逻辑特性出发,探讨了个体的可靠性和补偿代价对于应用的整体,并在此基础上提出了一种服务选取方法,该方法具有传统方法反映用户偏好的能力。同时从应用系统的角度出发,关注了在服务选取的过程中如何保证应用系统的整体可靠性以及降低应用系统的补偿代价。

结束语 实验和分析表明,本文提出的方法在反映服务使用者的使用偏好的同时,可以弥补现有服务选取方法在保证应用整体可靠性,以及降低系统的补偿代价方面的欠缺。将来,我们将研究基于统计的可靠性影响因子评估方法,并对在应用访问路径不等概率的情况下的服务选取策略进行更为深入的探讨。

参考文献

- [1] Zhang L, Zhang J, Cai H. Service Computing[M]. 1st edition. New York: Springer, 2007
- [2] Shuping R. A Model for Web Services Discovery With QoS [J]. ACM SIGecom Exchanges, 2003,4(1): 1-10
- [3] Liu Y, Anne H H, Zeng L. QoS Computation and Policing in Dynamic Web Service Selection[C]// WWW2004. New York, USA, 2004:66-73
- [4] Yeom G, Yun T, Min D. A QoS Model and Testing Mechanism for Quality-driven Web Services Selection [C]// The Fourth IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems. Gyeongju, Korea, 2006:199-200
- [5] Sharma A, Adarkar H, Sengupta S. Managing QoS through Prioritization in Web Services [C]// 4th International Conference on Web Information Systems Engineering Workshops. Rome, 2003:140-148
- [6] Menascé D A. QoS Issues in Web Services [J]. IEEE Internet Computing, 2002,6(6):72-75
- [7] Tsesmetzis D T, Roussaki I G, Papaioannou I V, et al. QoS awareness support in Web-Service semantics [C]// AICT-ICIW06, 2006:128-128
- [8] Wang X, Vitvar T, Kerrigan M. A QoS-Aware Selection Model for Semantic Web Services [C]// ICSOC2006. Chicago, 2006: 390-401
- [9] Mou Y, Cao J, Zhang S, et al. Interactive Web Service Choice-making Based on Extended QoS Model [C]// CIT. 2005: 1130-1134
- [10] Wang H, Tong P, Thompson P. QoS-based Web Services Selection [C]// IEEE International Conference on e-Business Engineering. Hong Kong, 2007:631-637
- [11] Canfora G, Penta M D, Esposito R, et al. An Approach for QoS-aware Service Composition: Based on Genetic Algorithms [C]// GECCO'05. Washington, USA, 2005:1069-1075
- [12] 代钰,杨雷,张斌,等.支持组合服务选取的 QoS 模型及优化求解[J].计算机学报,2006,29(7): 1167-1178
- [13] Yu T, Lin W. Service selection algorithms for Web services with end-to-end QoS constraints [C]// International Conference

- [14] Menascé D A. Composing Web Services: A QoS View[J]. IEEE Internet Computing, 2004, 8(6):88-95
- [15] 刘书雷,刘云翔,张帆,等.一种服务聚合中 QoS 全局最优服务动态选择算法[J].软件学报,2007,18(3):646-656
- [16] Demian A D, Ananthanarayana V S. Quality Driven Web Service Selection and Ranking[C]//Fifth International Conference on Information Technology. New Generations, 2008
- [17] Wan C, Wang H. Uncertainty-aware QoS Description and Selection Model for Web Services[C]//IEEE International Conference. 2007:154-161

- [18] Cardoso J, Bussler C. Semantic Web Services and Processes: Semantic Composition and Quality of Service[C]//On the Move to Meaningful Internet Computing and Ubiquitous Computer. Irvine CA, 2002
- [19] Jaeger M C, Rojec G, Muhl G. QoS aggregation for Web service composition using workflow patterns[C]//The 8th IEEE International Conference on Enterprise Distributed Object Computing. California, 2004:149-159
- [20] Wang X, Tomas V, Mick K, et al. A QoS-aware Selection Model for Semantic Web Services[C]//ICSOC 2006, LNCS 4294, 2006:390-401

(上接第 113 页)

5. TNC 客户端告诉 IMC 开始了一个新的网络连接,这个网络连接需要一个完整性握手协议。IMC 返回所需信息。TNC 服务端将这些信息交给 IMV。

6. 在这个过程里面,TNC 客户端和 TNC 服务端需要交换一次或多次数据,直到 TNC 服务器端满意为止。

7. 当 TNC 服务器完成了对客户端的完整性握手,它将发送一个推荐信(Action Recommendation)给网络访问授权(NAA),要求允许访问。这里需要特别注意,如果还有另外的安全考虑,此时 NAA 仍旧可以不允许网络访问者(NA)的访问。

8. NAA 传递访问决定(final decision)给 PEP,PEP 将最终执行这个决定,来控制 NA 的访问。

TNC 的安全接入控制系统主要基于“可信代理”的安全机制,结合终端系统的认证、评估子系统来实现对接入可信网络的终端系统、用户进行认证/授权控制,能够有效地避免可信网络中因不可信终端系统接入所带来的潜在风险。但一些关键技术点:网络安全策略的定义,客户端完整性的度量值和网络传输协议的安全性和效率都是需要进一步的研究。

4 网络的可管、可控和生存性研究

互联网络发展至今,已成为一个庞大的非线性复杂系统,如系统规模和用户数量巨大且不断增长,协议体系庞杂,业务种类繁多,异质网络融合发展等等。这远远超过了当初设计的考虑,现有的一些控制手段相对显得很薄弱,产生了许多安全隐患。“边缘论”和面向非连接的设计思想保障了网络的高效互通,逐跳存储转发的分组传送方式简单灵活,无需在中间节点维护过多的状态信息,核心网络的工作集中于路由转发。这些机制的优点是设计简单,可扩展性强等,然而却造成了分组传输路径的不可控,网络中间节点对传输数据包的来源不验证、不审计,导致地址假冒、垃圾信息泛滥,大量的入侵和攻击行为无法跟踪^[7]。所以对网络可管和可控目前依然是一个研究难题。

可信网络的可控可管性主要是指网络具有对用户行为、网络运行状态和网络资源的有效控制和管理的能力^[8]。为了实现真正的可信和安全,网络必须具有对用户行为高度的控制和管理能力。对用户行为的可控可以采用可信计算技术和密码技术进行管理和控制。对于网络状态的管理可以采用网络用户数据和管理数据分离;将网络状态数据的管理功能纳入到核心路由器的处理中;将状态管理功能模块化和普适化。

可生存性(Survivability)是指网络在遭受攻击、失效或者意外后能够及时地完成的能力^[9]。可生存性研究包括:定量评估问题,涉及建立包括网络的脆弱性分析、用户攻击行为描述在内的合理故障模型理论和定量评估的方法;保证可生存性的机制和策略问题,从单纯容错到同时考虑容错、容侵;从同构网络环境下的单种技术到异构网络下的层次化、协同可生存性技术。目前这些问题都是没有解决的热点问题。

结束语 可信网络是国际上将要出现的研究方向,借鉴了系统可信性的概念,将传统孤立的研究内容融合到网络可信这一目标下,面向用户提供系统的安全服务。本文对可信网络的几个关键问题进行了分析、总结和研究,包括它的架构、信任模型、接入控制、网络的可控可管和可生存性方面。

参考文献

- [1] 林闯,彭雪海.可信网络研究[J].计算机学报,2005,28(5):751-758
- [2] Forsberg D, Ohba Y, Patil B, et al. Protocol for Carrying Authentication for Network Access (PANA) [EB/OL]. [2007-12-03]. <http://www.ietf.org/internet-drafts/draft-ietf-pana-pana-18.txt>
- [3] Casadem, Garfinel T, Akellaa, et al. SANE: A protection architecture for enterprise networks[C]//Proceedings of 15th USENIX Security Symposium. July31-Aug. 4. 2006, Vancouver, Canada, 2006:137-151
- [4] Nicol D M, Sanders W H, Trivedi K S. Model-based evaluation: From dependability to security[J]. IEEE Transactions on Dependable and Secure Computing, 2004, 1(1)
- [5] IEEE. Std. 802. 1x IEEE Standard for Local and Metropolitan Area Network Port Based Network Access Control[s]. 2001
- [6] TCG Trusted Network Connect TNC Architecture for Interoperability[OL]. www.trustedcomputinggroup.org April 2008
- [7] Lin Chang, Peng Xuehai. Research on network architecture with trustworthiness and controllability[J]. Journal of Computer Science and Technology, 2006, 21(5): 732-739
- [8] Shenker S, Allman M, Paxson V. Architectural support for network trouble-shooting [EB/OL]. NSF FING' 2006. <http://www.nets-fing.net/Funded/ArchtSupportNet.php>
- [9] Ellison R, Fischer D, Linger R, et al. Survivable network systems: An emerging discipline [R]. CMU/SEI-2001-TN-001. Pittsburgh, PA, USA: Software Engineering Institute, Carnegie Mellon University, 2001