

模拟退火算法在 Web 服务合成中的应用

佟 萍^{1,2}

(1. 东南大学 计算机科学与工程系, 江苏 南京 210096;

2. 江苏铜山工业职业高级中学, 江苏 徐州 221121)

摘 要: Web 服务使用基于标准的协议, 为在线 B2B 合作提供了前所未有的机会。特别地, 把现有的 Web 服务合成以提供增值的服务就成为研究的热点。面对众多可以提供相同或相似功能的 Web 服务, 根据每个服务提供的非功能属性即服务质量(QoS)进行服务选择就变得至关重要。针对这个问题, 提出把具有相同或相似功能的服务组成服务类, 设计 QoS 全局最优的问题模型, 并用模拟退火算法求解。

关键词: Web 服务合成; 模拟退火算法; QoS

中图分类号: TP393

文献标识码: A

文章编号: 1005-3751(2006)03-0220-03

Application of Simulated Annealing Algorithm in Web Services Composition

TONG Ping^{1,2}

(1. Department of Computer Science and Engineering, Southeast University, Nanjing 210096, China;

2. Jiangsu Tongshan High Vocational School of Industry, Xuzhou 221121, China)

Abstract: Web services provides unprecedented opportunities for the on-line B2B collaborations, using the standard protocol-based. Particularly, it has become a new hot issue to composite the existing ones in order to supply value-added services. It has been crucial to choose the right services according to the non-functional attributes(i.e. QoS) offered by a services when facing a lot of Web services with same or similar functionality. To this question, this paper presents to combine the Web services with same or similar functionality to form a services class, designs a QoS global optimal model and solves it with simulated annealing algorithm.

Key words: Web services composition; simulated annealing algorithm; quality of services

0 引言

在分布式计算和电子商务领域, Web 服务已成为一个流行的软件框架, 它将 Web 应用从信息交互的领域扩展到了服务交互的领域。Web 服务是自包含、自描述的模块应用, 利用基于标准的协议(比如 SOAP, WSDL 和 UDDI), 可以把不同提供商提供的 Web 服务有效地集成, 从而提供增值的服务, 这就是 Web 服务合成。在对服务进行合成时, 由于 Web 服务数量的逐渐增加, 提供相同或相似功能的服务也越来越多, 再加上 Web 服务本身具有的自治性、异构性、分布性等特征, 使得服务的选择成为实现服务合成的前提。那么, 如何选择服务, 选择的依据是什么就成了人们研究的热点。当前对这方面的研究有很多, 但大多数都只考虑根据单个服务的 QoS 属性(比如价格、响应时间等)选择服务, 是局部的选择, 没有从合成服务的全局出发, 即根据合成服务的 QoS 从整体进行考虑。文献[1]分别从局部和全局的角度设计了服务的选择算

法, 但对全局选择算法的设计太复杂, 需占用大量的计算资源。

文中把具有相同功能的服务看成一个服务类, 提出把服务的选择问题转换成在给定的约束条件下, 求目标函数最优值的组合优化问题, 然后利用模拟退火算法进行服务的选择, 使合成后的 QoS 满足用户的需求。本方法对 Web 服务合成过程中的服务选择问题提出了一种新的求解思路, 可以有效地解决合成过程中的关键问题。

1 Web 服务合成模型

1.1 单个服务的 QoS 参数

文中考虑如下参数作为 Web 服务的质量标准^[2]:

(1) 服务费用: 表示用户调用该服务时需要付的金额。该费用由服务提供者给出, 用户通过查询可以获得。

(2) 响应时间: 表示用户从调用该服务到该服务响应给用户的时间。

(3) 服务可用性: 表示服务在某个时间段内可用的概率。

(4) 服务可靠性: 表示在期望的时间内请求被正确响

收稿日期: 2005-06-07

作者简介: 佟 萍(1977—), 女, 江苏徐州人, 硕士研究生, 研究方向为 Web 服务合成。

应的概率。

1.2 合成服务的类型

在介绍合成服务的类型前,先介绍两个概念^[1]。

(1) 执行路径:合成服务中每个任务(或服务)的执行顺序,比如 $[t_1, t_2, \dots, t_n]$ 就是一条执行路径。

(2) 有向无环图:用 $G(E, V)$ 表示,其中 E 是图中结点的集合,每个结点表示一个任务(或完成该任务的服务), V 是图中的边集,表示任务(或服务)之间的控制依赖和数据依赖。

根据执行路径的数量把合成服务分为两种类型^[3]:管道型和有向无环图型,如图1所示。管道型合成服务是最简单的合成服务结构,它只有一条执行路径;有向无环图型有多条执行路径,每条路径可以看作是一个管道型的合成服务。

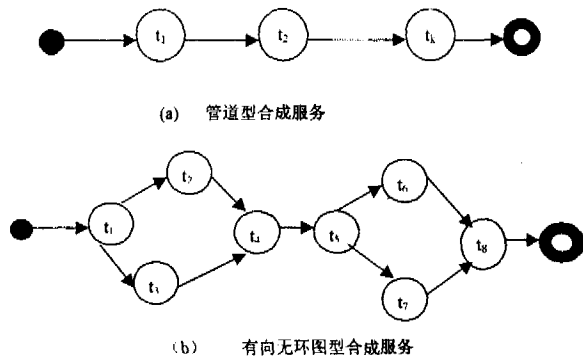


图1 合成服务的类型

2 问题模型

考虑管道型合成服务,假设一个管道型合成服务由 k 个任务组成,每个任务有 n 个服务提供者,现要为每一个任务选择一个服务提供者,则会产生 n^k 个可选方案,如何从这些方案中作出选择,以使合成服务满足用户的需求。

根据上面的描述,建立使合成服务全局最优的QoS数学模型。模型遵循以下原则:

- * 用户的期望最大,同时所付费用较少。
- * 满足用户提出的合成服务响应时间的约束。

模型符号定义如下:

(1) s :表示服务类,即具有相同功能,不同非功能属性的单个Web服务的集合。

(2) $p(s, i)$:表示调用服务类 s 中的第 i 个服务时需付的费用。

(3) $r(s, i)$:表示调用服务类 s 中的第 i 个服务的响应时间。

(4) R :表示合成服务总的响应时间。

(5) $C_{\max}(s, i)$ ^[3]:表示服务类 s 中的第 i 个服务的最大负载,即允许访问的最大用户数。

(6) $C_{\text{cur}}(s, i)$ ^[3]:表示服务类 s 中的第 i 个服务的当前负载,即当前正在访问该服务的用户数。

目标函数和约束条件:

首先对单个服务的期望值如下定义:

$$\text{考虑公式: } b(s, i) = 1 - \frac{1}{C_{\max}(s, i)} * C_{\text{cur}}(s, i) \quad (1)$$

由于 $C_{\text{cur}}(s, i)$ 是分布在 $[0, C_{\max}(s, i)]$ 之间的整数,是离散型随机变量,所以应考虑期望值。由泊松分布的特点^[4]可知,它符合泊松分布 $\pi(\lambda)$,其分布概率为:

$$P\{C_{\text{cur}}(s, i) = m\} = \frac{\lambda^m \cdot e^{-\lambda}}{m!} \quad (2)$$

$m = 0, 1, 2, \dots, C_{\max}(s, i)$, λ 为大于零的常数它的期望值为:

$$\begin{aligned} E(C_{\text{cur}}(s, i)) &= \sum_{m=0}^{C_{\max}(s, i)} P\{C_{\text{cur}}(s, i) = m\} \\ &= \sum_{m=0}^{C_{\max}(s, i)} \frac{\lambda^m \cdot e^{-\lambda}}{m!} \end{aligned} \quad (3)$$

则:

$$E(b(s, i)) = 1 - \frac{1}{C_{\max}(s, i)} * E(C_{\text{cur}}(s, i)) \quad (4)$$

然后根据期望值和费用,定义单个服务的效用函数:

$$F(s, i) = w_b * E(b(s, i)) - w_c * p(s, i) \quad (5)$$

为简单起见, $F(s, i)$ 用 F_s 表示。

其中: w_b :期望的权重。 $0 < w_b < 1$

w_c :费用的权重。 $0 < w_c < 1$ 且 $w_c = 1 - w_b$

$E(b(s, i))$:调用服务类 s 中的第 i 个服务所获取的期望值

$p(s, i)$:调用服务类 s 中的第 i 个服务所需的费用

则合成服务总的效用函数为:

$$W = \sum_{i=1}^k \sum_{j=1}^n F_{ij} x_{ij} \quad (6)$$

其中: $\sum x_{ij} = 1, 1 \leq i \leq k, 1 \leq j \leq n$;

$x_{ij} \in \{0, 1\}, 1 \leq i \leq k, 1 \leq j \leq n$;

$x_{ij} = \begin{cases} 0, & \text{第 } i \text{ 个服务类中第 } j \text{ 个服务未被选中} \\ 1, & \text{第 } i \text{ 个服务类中第 } j \text{ 个服务被选中} \end{cases}$

最后根据模型应遵循的原则,设计目标函数和约束如下:

$$\begin{aligned} \text{obj Max } W &= \sum_{i=1}^k \sum_{j=1}^n F_{ij} x_{ij} \\ \text{s. t. } &\sum_{i=1}^k \sum_{j=1}^n r_{ij} x_{ij} \leq R \\ &\sum x_{ij} = 1, 1 \leq i \leq k, 1 \leq j \leq n; \\ &x_{ij} \in \{0, 1\}, 1 \leq i \leq k, 1 \leq j \leq n; \end{aligned} \quad (7)$$

其中: F_{ij} :第 i 个服务类中第 j 个服务的效用值。

r_{ij} :第 i 个服务类中第 j 个服务的响应时间。

R :合成服务总的响应时间。

3 模型求解

上述问题可以看成组合优化问题,当采用穷举法时,虽然最终可以获得最优解,但却需要问题规模的指数阶时间,尤其是当规模很大时,其计算时间将是难以忍受的。因此,需要找出兼顾解的质量以及运算时间的较好算法。

模拟退火算法是Kirkpatrick等于1983年首先提出

的,是从自然界固体退火过程中得到启发并从中抽象得来,是解大规模优化问题的一种随机优化算法。该算法最显著的特征是以一定的概率接受使目标函数值增大的移动,即恶化解,所以能从局部最优解的“陷阱”中爬出来而不是简单地终止在局部最优解上,具有全局收敛性^[5]。

基于模拟退火算法的思想,设计适合文中模型的算法如下。

步骤 1 $X_{00} \in Q$ 是随机产生的初始解的矩阵,其中:

$$Q = \left\{ \begin{bmatrix} x_{11} \cdots x_{1n} \\ \vdots \\ x_{k1} \cdots x_{kn} \end{bmatrix}, x_{ij} \in \{0,1\} \text{ 且 } \sum x_{ij} = 1 \right\}$$

为可能解的集合, x_{ij} 表示第 i 个服务类中第 j 个服务是否被选中的状态, 计算相应的目标函数值 W_0 ; 给出控制参数初值 T_0 、马尔柯夫链长度 N 以及停止参数 ϵ (指当控制参数 T 递减到 ϵ 时算法停止)。

注:在计算目标函数的值时,因为问题的描述中带有约束条件,所以采用惩罚策略来修正目标函数:

$$F(x) = f(x) \pm R_j \sum_{j=1}^m \max[0, g_j(x)]^2 \pm R_k \sum_{k=1}^l [h_k(x)]^2 \quad (8)$$

式中: $g_j(x)$ 为不等式约束, $h_k(x)$ 为等式约束, R_j, R_k 为加权大数因子;在最大化问题中取“-”,在最小化问题中取“+”。因此,本问题中的目标函数修正为:

$$W = \sum_{i=1}^I \sum_{j=1}^n F_{ij} x_{ij} - R_j * \max[0, (\sum_{i=1}^I \sum_{j=1}^n r_{ij} x_{ij} - R)]^2 \quad (9)$$

步骤 2 产生新解并计算新解与当前解的目标函数值之差 Δf 。

新解的产生:

在 $1 \sim n$ 之间随机选取 i, j , 当前解中若每个服务类中第 i 个服务和第 j 个服务状态相同,则将其中一个和状态不同的服务交换状态;否则,两者交换其状态。即:

$$\begin{cases} x_{ii} = 1 \text{ 或 } x_{ij} = 1, \text{ 同时置状态为 1 的服务状态为 0} & x_{ii} = x_{ij} \\ x_{ii} = 1 - x_{ij} \text{ 且 } x_{ij} = 1 - x_{ii} & x_{ii} \neq x_{ij} \end{cases}$$

步骤 3 由 Metropolis 接受准则判断是否接受新解。

(1) 若 $\Delta f \geq 0$, 即新解可行且优于当前解,则接受。

(2) 若 $\Delta f < 0$, 则按新点接受概率:

$$p(\Delta f, T) = \exp\left(\frac{\Delta f}{T}\right)$$

取 $[0, 1]$ 区间上均匀分布的随机数 δ , 若 $p(\Delta f, T) \geq \delta$, 则接受新解, 否则放弃新解。

步骤 4 累计次数 n , 若 $n \leq N$ 转步骤 2, 否则转步骤 5。

步骤 5 判断停止准则是否满足。若不满足则令 $T_{k+1} = \lambda * T_k, n = 1$, 转步骤 3, 否则停止算法输出当前解。

式中: $\lambda = 0.8 \sim 0.999$

4 算法分析

从上述算法可以看出, 模拟退火算法由于采用了 Metropolis 接受准则, 在搜索解的过程中引入了新的随机因素, 使算法进程呈现跳跃性而可能跳离局部最优的“陷阱”, 从而保证了最终解并不依赖于随机选取的初始解, 提高了最终解质量的稳定性; 对于算法的时间复杂性, 可以借助算法的比较次数进行分析。由控制参数 T 的初值 T_0 和衰减函数 $T_{k+1} = \lambda * T_k$, 对 T 的每一取值所进行的 N 次迭代及停止准则所规定的控制参数的终值 ϵ 构成了控制算法有限时执行的冷却进度表, 由它的控制作用, 使算法的时间复杂性为问题规模的多项式时间^[5]。而采用穷尽搜索时, 当问题规模很大时需要指数级时间, 因此该算法大大提高了运算效率。

5 总结

文中针对 Web 服务合成中的研究热点问题, 即如何从合成服务的 QoS 角度选择满足用户需求的服务, 设计了 QoS 全局最优的数学模型, 并进行了适合本模型的模拟退火算法的设计。从算法分析的结果可以看出, 本算法对求解 Web 服务的选择问题是合适的, 是可以在实际中应用的。对于有向无环图型的合成服务, 只要计算每一条执行路径的最大效用值, 然后再从中选择效用值最大的那条路径作为最优执行路径即可。

参考文献:

- [1] Zeng L, Benatallah B, Ngu A, et al. QoS - Aware Middleware for Web Services Composition[J]. IEEE Transactions on Software Engineering, 2004, 30(5): 311 - 327.
- [2] Cardoso J, Bussler C, Sheth A. Semantic Web Services and Processes: Semantic Composition and Quality of Service. tutorial at Federated Conferences (CooPIS, DOA, ODBASE), 2002 [EB/OL]. <http://lsdis.cs.uga.edu/lib/presentations/SWSP-tutorial-resource.htm>, 2002.
- [3] Yu T, Lin K J. Service Selection Algorithms for Web Services with End - to - End QoS Constraints[A]. Proceedings of the International Conference on E - Commerce Technology[C]. San Diego, California: IEEE Computer Society Press, 2004. 129 - 136.
- [4] 李裕奇. 随机过程[M]. 北京: 国防工业出版社, 2003.
- [5] 康立山, 谢云, 尤矢勇, 等. 非数值并行算法——模拟退火算法[M]. 北京: 科学出版社, 1994.

(上接第 123 页)

- [4] Bigi B, De Mori R, El - Beze M, et al. Detecting topic shifts using a cache memory[A]. 5th International Conference on Spoken Language Processing[C]. Sydney, Australia: [s. n.],

1998. 2331 - 2334.

- [5] 杨力. 美国口语大观: 中英文对照[M]. 合肥: 中国科学技术大学出版社, 2001.