

基于粒子群算法的 Web 服务组合研究

刘莉平^{1,2}, 陈志刚^{1,2}, 刘爱心²

(1. 中南大学软件学院, 长沙 410083; 2. 中南大学信息科学与工程学院, 长沙 410083)

摘 要: 针对现有服务组合中 QoS 优化的不足, 该文提出一种基于粒子群算法的解决 QoS 动态服务组合算法。通过对服务组合的业务逻辑与服务实例进行合理编码, 重新定义粒子的位置、速度与“加”运算, 利用粒子群算法的智能优化原理以及局部与全局优化信息加快粒子群的搜索速度, 使其能够快速得到一组满足约束条件的 Pareto 优化的服务组合。实验结果证明了算法的可行性和有效性。

关键词: Web 服务组合; 服务选取; 粒子群算法; Pareto 优化

Research on Web Services Composition Based on Particle Swarm Optimization

LIU Li-ping^{1,2}, CHEN Zhi-gang^{1,2}, LIU Ai-xin²

(1. School of Software, Central South University, Changsha 410083;

2. School of Information Science and Engineering, Central South University, Changsha 410083)

【Abstract】 This paper presents an improved algorithm based on particle swarm, which is to resolve dynamic Web Services selection with QoS optimal in Web Services composition. The essence of the algorithm is that the problem of dynamic Web Service selection with QoS optimal is transformed into a multi-objective services composition optimization with QoS constraints. The theory of intelligent optimization of particle swarm optimization algorithm is utilized to produce a set of optimal Pareto services composition process with constraint principle by accelerating global and detail searching speed based on deciding PSO state. Experimental results indicate the feasibility and efficiency of this algorithm.

【Key words】 Web Services composition; services selection; particle swarm optimization; Pareto optimal

1 概述

近年来 Web Services 作为一种新技术广受关注。Web Services 中的接口定义语言 WSDL^[1] 和内容传输格式 SOAP^[2] 已经成为 W3C 的草案和建议标准。然而, 在实际应用中, 单个 Web 服务通常无法满足复杂应用的需求。因此, 如何集成单一服务形成功能更强大的服务组合从而满足不同用户的复杂应用已成为一个新的研究热点。服务组合广义上可以分为手工组合、半自动组合和自动组合^[3]。在 Web 环境中, 服务是经常变化的, 手工组合的方式并不能满足实际的应用需求, 而完全智能化的自动组合方式是一个非常复杂的过程, 因此, 很多关于服务组合的应用和研究工作都侧重于半自动方式^[4-5]。半自动服务组合方式首先由业务人员根据特定的行业背景, 建立适合具体应用需求的通用服务组合业务模型, 服务组合业务模型由多个服务节点组成, 各服务节点包含功能需求描述, 但不指定具体的服务调用实例, 如图 1 所示。

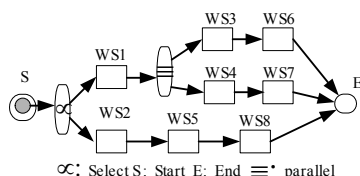


图 1 服务组合业务结构

在 Web 环境中, 满足相同功能需求而具有不同 QoS 参数 (如执行时间、费用、可靠性等) 的 Web 服务实例存在多个, 如何从中选择满足各服务节点功能需求的具体服务, 形成一个可执行的服务链来完成用户的需求称为服务动态组合问

题。目前, 服务动态组合的基本目标是针对不同的应用, 从候选服务集合中选择一组服务, 使得其 QoS 达到 Pareto 最优。

2 服务组合基本模型

本文所采用的服务组合模型为通用的服务覆盖网^[6], 其对应互联网的实际场景如图 2 所示。在互联网中每一个服务提供节点提供一个或多个服务, 如图 2 的节点 V₂ 提供 S₅ 和 S₇ 两个服务, 这样形成服务覆盖网。

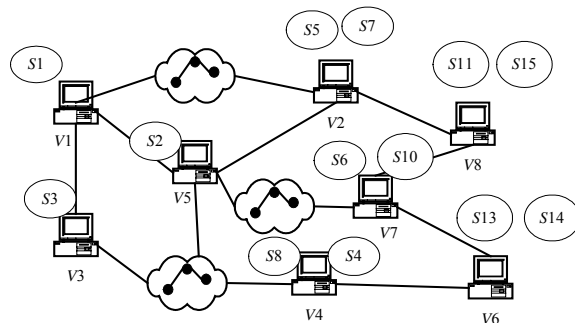


图 2 服务覆盖网

服务覆盖网中的每个提供服务的节点具有不同的资源能力, 可作如下定义:

定义 1 系统中每个提供服务的节点 V 拥有的有效多维资源为 $N_v = (r_v^1, r_v^2, \dots, r_v^m)$, $V = \{1 \dots \text{提供服务的节点数}\}$, 表

基金项目: 国家自然科学基金资助项目(60573127)

作者简介: 刘莉平(1971 -), 女, 博士研究生, 主研方向: 网络计算, 面向服务计算; 陈志刚, 教授、博士; 刘爱心, 高级工程师

收稿日期: 2007-03-10 **E-mail:** csuliu@163.com

示节点 V 提供的资源维数为 m 维,第 i 维的资源为 r^i_v ,如CPU大小、内存、I/O等节点处理能力的资源量。

定义 2 一个 Web 服务可以用下面的表达式(1)来描述:

$$WS_i = (I_i, O_i, Res_i, QoS_i) \quad (1)$$

其中, WS_i 是Web服务的名字; I_i 是该服务的输入集合, O_i 是该服务的输出集合。 QoS_i 是表示服务 WS_i 在服务质量参数上的约束,对于服务 WS_i 的共有服务质量维数 $Q_i = \{q_{i,1}, q_{i,2}, \dots, q_{i,k}\}$,对于服务 WS_i 存在最小可接受的服务质量级别 $q_i^{\min} = (q_{i,1}^{\min}, q_{i,2}^{\min}, \dots, q_{i,k}^{\min})$, Res_i 是表征服务执行需要系统资源的情况,对于服务 WS_i 言,假定要使其具有服务质量 Q_j ,有映射函数 f 是服务 WS_i 得到服务质量 Q_j 时所需系统资源大小, $f(Q_j) = f(R(q_{j,1}, q_{j,2}, \dots, q_{j,k})) = r_i^j = \{r^1_i, r^2_i, \dots, r^m_i\}$,其中, r_i^j 表示服务 WS_i 得到服务质量 Q_j 所需的资源向量, r^j_i 表示服务 WS_i 在得到服务质量为 Q_j 在第 i 维资源上所需的资源量。

定义 3 服务组合业务图(Services Composition Business Graph, SCBG): 依据服务组合的业务逻辑关系确定的各服务类型间相互关系的组合图,如图 2 所示。在服务组合业务逻辑图中的服务是指一种服务的类型,而不是具体的服务实例。用二元组可以描述为 $WCL = (W_{model}, CC)$, 其中:

$W_{model} = \{W_1, W_2, \dots, W_N\}$ 为服务类型的集合, W_1, W_2, \dots, W_N 表示服务组合的 N 个参与的服务类型;

$CC \subseteq W_{model} \times \varphi \times W_{model}$ 为组合控制的集合, 控制结构运算符 φ 定义为从 W_{model} 到 W_{model} 的控制转换关系, WfMC 定义的 4 种基本模型^[5]为: 顺序, 选择, 并行, 循环。

定义 4 服务组合实例图: 将服务覆盖网中的实际服务实例依据定义 3 的服务组合业务逻辑图组成如图 3 所示的服务组合实例图。

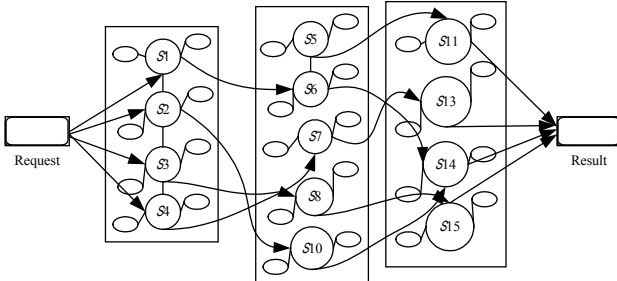


图 3 服务组合实例

定义 5 服务组合的多约束条件下多目标最优: 对于服务组合业务逻辑图, 存在多个从源到终点的组合路径, 依次从服务组合实例图中选取合适的服务形成所有可能的服务组合, 存在 $k(k \geq 2)$ 个约束条件 $C_i (i=1, 2, \dots, k)$, 如执行时间、执行代价不高于(低于)某个值, 同时, 每一条从源点 S 到目的点 T 的服务组合 P 具有 $m(m \geq 2)$ 个非负的性能度量准则 f_1, f_2, \dots, f_m , P 为多约束条件下多目标最优非劣路径, 当且仅当 $\forall P^* (P^* \neq P)$, 在 P 和 P^* 均满足约束 C_i 的条件下, 对于所有的度量准则均使得 $f_i(P) \geq f_i(P^*)$, 且至少存在一个目标 i 满足 $f_i(P) > f_i(P^*)$ 。其中, \geq 和 $>$ 分别表示度量准则之间的不劣于和优于关系。

3 基于粒子群算法的 Web 服务组合

3.1 基本 PSO

粒子群算法由 Kennedy J 等在 1995 年提出, 是一种基于迭代的进化计算方法^[7]。基本 PSO 公式如式(2)所示。设搜索空间是 D 维的, 粒子群中第 i 个粒子的位置用 $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$ 表示, 第 i 个粒子的速度表示为 $v_i = (v_{i1}, v_{i2}, \dots, v_{id})$ 。第 i 个粒子

迄今为止搜索的最好位置记为 $p_i = (p_{i1}, p_{i2}, \dots, p_{id})$, 整个粒子迄今为止搜索到的最好位置记为 $p_g = (p_{g1}, p_{g2}, \dots, p_{gd})$ 。对于每一个粒子, 其第 d 维($1 \leq d \leq D$)根据如下等式变化:

$$\begin{cases} V_{id} = w \times V_{id} + c_1 r_1 \times (P_{id} - X_{id}) + c_2 r_2 \times (p_{gd} - X_{id}) \\ V_{id} = V_{\max} & \text{if } V_{id} > V_{\max} \\ V_{id} = -V_{\max} & \text{if } V_{id} < -V_{\max} \\ X_{id} = X_{id} + V_{id} \end{cases} \quad (2)$$

其中, r_1, r_2 是介于 $[0, 1]$ 之间的随机数; C_1, C_2 是加速度系数; w 是惯量因子; V_{\max} 是常数, 限制了速度的最大值, 由用户设定。

3.2 粒子群思想和模型

基于粒子群思想的服务组合算法从多目标优化的角度出发, 搜索服务集合中 S 和 E 之间满足所有约束条件的一组多目标 Pareto 非劣解。基本思想是: 首先产生一定数目的粒子群, 每一个服务组合流程编码为一个粒子, 每一个粒子利用本身信息、局部较优信息和全局较优信息 3 个信息, 产生具有更高目标值的新粒子。这一过程不断进行, 实现在解空间的并行全局搜索; 算法停止时, 得到一组粒子集合, 对应了多条收敛于 Pareto 优化或近似优化路径的组合集。算法流程见算法 1。

算法 1

- (1) Initialize particle_swarm PS //初始化粒子群
- (2) DO
- (3) for $i=1$ to particle_swarm_Size
- (4) calculate the fitness value $f(x_i)$ of the i th particle
- (5) if $f(x_i) > f(p_i)$ then $p_i = x_i$
- (6) $p_g = \max(p_i)$
- (7) for $d=1$ to DimensionSize
- (8) $V_{id} = w \times V_{id} + c_1 \times r_1 \times (p_{id} - V_{id}) + c_2 \times r_2 \times (p_{gd} - V_{id})$
- (9) $X_{id} = X_{id} + V_{id}$
- (10) $V_{id} = V_{\max}$ if $V_{id} > V_{\max}$
- (11) $V_{id} = -V_{\max}$ if $V_{id} < -V_{\max}$
- (12) Next d
- (13) Next i
- (14) While termination criterion is met

3.3 服务组合算法的设计与分析

3.3.1 编码策略

首先对图 2 中的各个服务节点按顺序编号; 其次, 增加虚拟服务 S 和 E 作为服务组合的起点和终点, SCBG 中节点的个数为粒子的 D 维, 因为每一个给定的服务组合所包含的服务节点数目相同, 所以采用整数定长编码的方式, 将组合路径映射为粒子空间中的个体; 粒子体中第 1 个和最后一个粒子位置总是 SCBG 图中的顶点 S 和 E , 取值为-1, 粒子位置的每一维的值是 SCBG 图中相应位置对应的某一服务实例的编号, 如果对应位置的服务不在组合路径中, 则取 0。

假设粒子 $X_i = (x_{i1}, x_{i2}, \dots, x_{id})$ 表示服务组合路径的一个解, 其中, D 为SCBG图中顶点的个数, 其编号是从上到下, 从左到右编号; x_{i1} 和 x_{id} 的取值是-1, 其他 x_{ij} 的取值为 0 或者服务实例的编号(正的整数)。

例如, 对于图 2 中的服务业务逻辑图对应的粒子编码如下所示: $X = (-1, 0, 2, 0, 0, 5, 0, 0, 9, -1)$ 表示一条经过第 2 号、第 5 号、第 9 号服务实例的组合路径。

3.3.2 初始组合粒子群集的生成

本文利用随机方法产生一组满足约束条件的从 S 到 T 的

初始组合路径集。具体过程见算法 2。设初始粒子群规模为 N ， $Constr(PS)$ 表示从集合 PS 中选取满足约束的组合路径； $RandS(S,T)$ 表示利用随机方法选择 S 到 T 的一条路径。

算法 2 particle_swarm PS

```

Begin
(1) PS ← 
(2) while (|PS| < N)
(3) PS ← PS  Constr(RandS(S,T))
(4) Goto (2)
(5) Output PS
End

```

3.3.3 粒子适应值计算

本文中采用加权综合的方法来求适应值。

对于每一条组合路径用 $f_1(x), f_2(x), \dots, f_p(x)$ 分别代表对 QoS 因素 $QoS(time)$, $QoS(cost)$, $QoS(cap)$, $QoS(rep)$ 等进行 QoS 计算的函数。希望在 QoS 上得到满足以下目标的优化目标：

$$f(x) = F(\max f_1(x), \min f_2(x), \min f_3(x), \max f_4(x), \max f_5(x)) \quad (3)$$

由于式(3)中的 \max 函数可以通过转化为 \min 函数，因此用加权法求得每一条组合路径的综合服务质量函数为

$$f(x) = \min \sum w_i f_i(x) \quad \partial_i, \quad x \in X \quad (4)$$

3.3.4 粒子的运算规则

至此，算法 1 中只有第 8 行和第 9 行还没有定义。第 8 行、第 9 行的作用分别是计算粒子的飞行速度和更新粒子的位置，更新粒子位置采用“+”运算。本文的策略是将粒子的飞行速度 V_{id} 定义为服务质量的值。粒子位置的更新就是将服务替换为与此服务质量最接近的服务实例，因此“加法” (+) 的定义为将粒子 X_i 的第 D 维位置更换为与 V_{id} 服务质量最接近的服务实例。下面通过一个实例来说明。

在图 1 中，假设在计算前 V_{id} 的值为 0.35，并取 $w, c_1, r_1, p_{id}, c_2, r_2$ 分别是 0.2, 0.4, 0.5, 0.23, 0.12, 0.25，则计算出新的 V_{id} 的值为 0.36，则说明粒子在对第 D 类服务的选取 QoS 高的服务实例。 $x_{id} = x_{id} + v_{id}$ 的计算用式(5)完成：

$$x_{id} = \bar{f}((f(x_{id}) + v_{id})) \quad (5)$$

式(5)中的 f 函数是计算出 x_{id} 所对应的服务实例的 QoS，计算出值为 5.36， \bar{f} 函数是求得与 QoS 值最接近的服务实例，为 S_8 ，与 f 是反函数。

4 实验分析

4.1 实验场景设置

实验环境为 100 Mb/s 局域网，微机配置为 Pentium 4，2.52 GHz/512 MB，Windows 2003 Server/XP，用 Java 实现。实验所用服务组合业务模型如图 1 所示，模型中每一类服务节点对应实际中的一组服务实例集合，共有 9 种服务 (WS_1, WS_2, \dots, WS_9)。每一个服务实例由一个线程来模拟。各个服务群中服务的 QoS 参数采用随机方法在一定范围内生成，参数取值范围设定为 $0 \leq Time \leq 60$ s, $0 < Cost \leq 100$, $0 < Repute \leq 10$, $0 < Reliable \leq 1$ ，最小信誉等级为 2，最小可靠性为 0.1。

4.2 算法参数设置

根据 3.3.1 节算法编码，得到如下所示的服务组合模型的粒子编码实例的形式：

-1	Sc(1i)	Sc(2i)	Sc(3i)	Sc(4i)	Sc(5i)	Sc(6i)	Sc(7i)	Sc(8i)	-1
----	--------	--------	--------	--------	--------	--------	--------	--------	----

其中， $Sc(Ii)$ 表示服务节点 1 对应的服务集合 $SCi(i \in \{1/1, j/8\})$ 中所选具体服务的编号为 i 。

对于图 1 所示的组合业务模型，共有 2 条组合路径。

第 1 条是选择 ($S \rightarrow WS_2 \rightarrow WS_5 \rightarrow WS_8 \rightarrow E$)，其 QoS 指标的计算公式如式 (6) 所示，第 2 条是 ($S \rightarrow WS_1 \rightarrow ((WS_3 \rightarrow WS_6 \rightarrow E), (WS_4 \rightarrow WS_7 \rightarrow E)))$ ，其计算公式如式 (7) 所示。

$$\begin{cases} f_1(x) = t_2 + t_5 + t_8 \\ f_2(x) = C_2 + C_5 + C_8 \\ f_3(x) = Rep_2 \times Rep_5 \times Rep_8 \\ f_4(x) = Rel_2 \times Rel_5 \times Rel_8 \end{cases} \quad (6)$$

$$\begin{cases} f_1(x) = t_1 + \max((t_3 + t_6), (t_4 + t_7)) \\ f_2(x) = C_1 + C_3 + C_4 + C_6 + C_7 \\ f_3(x) = Rep_1 \times \min((Rep_3 \times Rep_6), (Rep_4 \times Rep_7)) \\ f_4(x) = Rel_1 \times \min((Rel_3 \times Rel_6), (Rel_4 \times Rel_7)) \end{cases} \quad (7)$$

最后通过加权综合适应度式(4)求得综合适应度参数，对于式(4)中的权重 w 均取值为 0.25。

4.3 有效性实验

通过算法执行的时间开销来验证本文提出的算法有效性。分别考虑了粒子服务群规模为 100、150 和 200、迭代数取 200、300、400、500 和 600 的情况下的 CPU 时间开销。对于每一种情况，算法分别运行 15 次取平均值。由图 4 可以看出：随着服务实例数目的增加，在不同的迭代次数下，CPU 运行时间并没有大量增加。

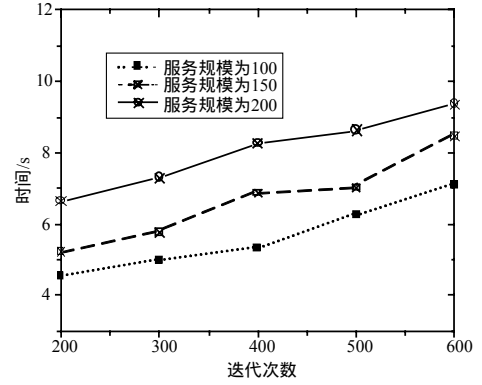


图 4 算法的执行时间

实验还对线性规划法、采用服务个数为粒子维数的编码算法以及本的算法的效率进行了比较。分别给定有 100、150、200、250、300 个服务实例情况。它们达到最优解 QoS 指标相差在 2% 以内所需要的时间如图 5 所示。

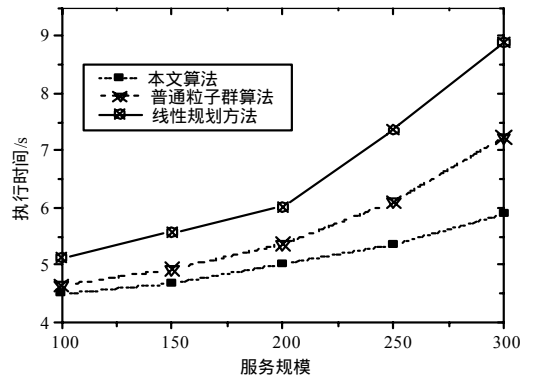


图 5 不同算法执行时间的比较

5 结束语

本文提出的基于粒子群算法的服务组合算法，对于构建基于互联网的复杂组合服务的应用具有一定意义。进一步的工作是根据服务组合的具体实际，进行基于 PSO 算法的优化与改进。

(下转第 112 页)