

一组编制模式下组合 Web 服务模型的 约简规则与算法

杨怀洲^{1,2}, 李增智¹

(1. 西安交通大学计算机系统结构与网络研究所, 710049, 西安; 2. 北京邮电大学
网络与交换技术国家重点实验室, 100876, 北京)

摘要: 为了解决 Web 服务组合过程中状态空间庞大而导致性能求解的效率难以满足实时性要求的问题, 提出了一种过程约简算法. 在利用广义随机 Petri 网进行过程建模后, 保留系统时间性能特征, 通过对可约简子网的自动探索, 剔除部分规则结构(序列、选择、并行、循环), 孤立出可单独求解的子结构, 由此在比较满意的时间内完成大规模模型约简. 多种模型规模下的实验性能表明, 所提算法可在动态服务组合与组合过程中自适应、快速地获得系统响应时间和吞吐量等性能指标, 适用于多数服务组合过程的在线响应时间分析.

关键词: 约简规则; 约简算法; 广义随机 Petri 网; 响应时间

中图分类号: TP311 **文献标志码:** A **文章编号:** 0253-987X(2009)06-0020-04

A Set of Model Reduction Rules and Algorithms for Web Services Composition under Orchestration Pattern

YANG Huaizhou^{1,2}, LI Zengzhi¹

(1. Institute of Computer Architecture & Networks, Xi'an Jiaotong University, Xi'an 710049, China; 2. State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: The performance calculation of Web services composition process in large state space is time-consuming, and is difficult to fulfill the real-time performance analysis. Therefore, a process reduction algorithm is presented to accelerate the calculation of system performance. The Web services composition process is modeled using the generalized stochastic Petri net. Through auto-detecting the reductive subnet, some regular structures, such as sequence, choice, concurrent and loop structures, can be eliminated under the precondition of preserving the timing constraints. The sub processes which can be calculated independently are separated. A large scale model can thus be reduced within a certain period. Experimental results with multiform models indicate that the system response time and throughput are obtained rapidly by using the algorithm in the dynamic and adaptive composition process. The algorithm can be applied to online performance analysis of most Web services composition process.

Keywords: reduction rule; reduction algorithm; generalized stochastic Petri net; response time

近年来,很多理论和技术已应用于 Web 服务组合的研究^[1-3],但多数研究着重于建模和模型检查,忽略了性能方面的量化问题.为了能够快速获取系

统性能指标,需要对模型进行约简.本文研究主要针对编制模式下的组合服务模型,该模式是目前较为实用的主流模式.编排模式下的组合服务模型包含

了大量同步、复杂的分支和竞争结构,在保留系统时间特性的条件下难以进行模型约简。

1 组合过程的广义随机 Petri 网模型

本文采用广义随机 Petri 网(GSPN)对组合过程进行建模,用位置来描述组合过程中的执行条件和局部状态,用时间变迁来描述服务实体的活动,用瞬时变迁来描述系统内部控制或逻辑上的选择,用弧描述因果关系,用标记描述组合过程所处理的实例。Web 服务功能一般不具有“记忆性”,其延时分布可用指数分布来描述。组合脚本语言中的组合结构和过程控制部分都具有各种相似的流程模板,可自然、直接、无歧义地映射为相应的 GSPN 结构。

2 模型结构的约简

基本 Petri 网约简方法由于不能保持系统时间属性,因而在对 GSPN 的约简中基本无法使用。对 Petri 网模型进行自动化约简的研究更为缺乏,针对工作流的一些研究^[4]给予了我们很大的启发,本文就是借鉴了其基于距离标定的子网探索思想。为了保持约简后模型时间特性不变,我们选择了如图 1 所示组合服务 GSPN 模型中常见的 4 种基本结构作为约简对象,文献^[5]给出了这 4 种结构的性能等价公式,可直接使用。在提出约简规则之前,首先做如下定义。

定义 2 GSPN 可约简子网。

令 Σ' 是 Σ 的 GSPN 子网, $N = S' \cup T'$ 为 Σ' 的结点集, Σ' 是 Σ 的 GSPN 可约简子网,条件是: $\exists i, o \in S'$, 使得对于 Σ 以下条件成立。

(1) $|T'| \geq 2$, 即子网至少包含 2 个变迁。

(2) $(i \cdot \cup o) \subseteq N$ 。

(3) $i \nsubseteq T', o \nsubseteq T'$ 。

(4) $\forall n \in N \setminus \{i, o\}: n \cup n' \subseteq N$ 。除 i 和 o 外, Σ' 的结点在 Σ 中不与任何除 Σ' 结点外的其他结点相联。条件(3)、(4)说明 i 和 o 分别是 Σ' 惟一的输入和输出位置。

(5) $\bigcup_{n \in N \setminus \{i, o\}} (n \cup n') = N$ 。条件(3)、(5)说明子网不孤立,子网是全连通的。

(6) $\forall s \in S' \setminus \{i, o\}, M_0(s) = 0$ 。除 i, o 外, Σ' 中其他位置没有初始标记。

令 A 是一个原子性网,它由一个时间性变迁 t 和惟一的输入位置 i 、输出位置 o 构成,如图 1 中虚框部分所示。GSPN 可约简子网 Σ' 可以被替换为 A , 时间变迁 t 的实施速率 λ 可根据相应性能的等价公式^[5]计算, A 中 i, o 位置的标记情况与 Σ' 中 i, o 位置的标记情况相同。当 Σ' 为图 1 所示的规则结构时,可通过文献^[5]中的相应公式得到变迁 t 的 λ 值,其相应的约简规则如下。

规则 1 串联型约简。

一个 GSPN 可约简子网 Σ' 能被替换为一个原子性网 A , 但需满足: ① $\forall n \in N \setminus \{i, o\}: |n| = |n'| = 1$; ② $|i \cdot| = |o| = 1$ 。

规则 2 选择型约简。

一个 GSPN 可约简子网 Σ' 能被替换为一个原子性网 A , 但需满足: ① $\forall t_i \in i \cdot \cup o, \lambda_{t_i} = \infty$; ② $|i \cdot| = |o| = |T'|$; ③ $\forall n \in N \setminus \{i, o\}: |n| = 1, |n'| = 1$, 每一个选择分支均为串行结构; ④ $\forall t_i \in T'_i: (t_i) \in i \cdot, (t_i)' \in o$ (每一个分支中仅包含一个原子性网 A)。

规则 3 并联型约简。

一个 GSPN 可约简子网 Σ' 能被替换为一个原子性网 A , 但需满足: ① $\forall t_i \in i \cdot \cup o, \lambda_{t_i} = \infty$; ② $|i \cdot| = |o| = 1$; ③ $|(i \cdot)'| = |(o)| = |T'_i|$; ④ $\forall n \in N \setminus \{i, o, i', o'\}: |n| = 1, |n'| = 1$, 每一个并行分支均为串行结构; ⑤ $\forall t_i \in T'_i: (t_i) = i \cdot, (t_i)' = o$ (每一个分支中仅包含一个原子性网 A)。

规则 4 循环型约简。

一个 GSPN 可约简子网 Σ' 能被替换为一个原子性网 A , 但需满足: ① $|i \cdot| = |o| = 1$; ② $|(o)| =$

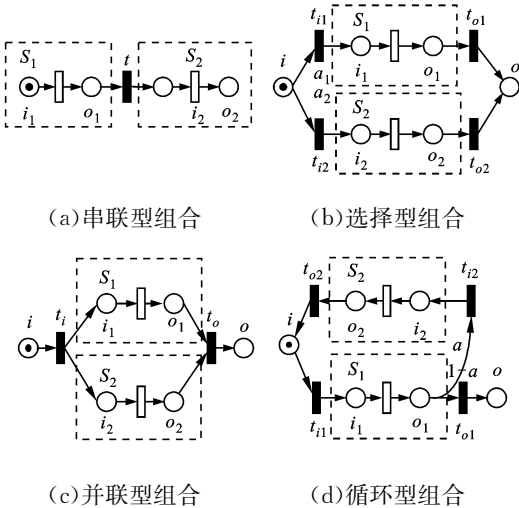


图 1 组合服务 GSPN 模型可约简子网结构

定义 1 GSPN 子网。

令 $\Sigma' = (S', T', T'_i; F', W', M'_0, \lambda')$, $\Sigma = (S, T_i, T_i; F, W, M_0, \lambda)$ 是 2 个 GSPN, Σ' 是 Σ 的子网, 条件是: $S' \subseteq S, T'_i \subseteq T_i, T'_i \subseteq T_i$, 且 $W' = W' \cap W, \lambda' = \lambda' \cap \lambda, F' = F \cap ((S' \times T') \cup (T' \times S'))$ 。

1, 令 $o_1 = \cdot(b)$, $|o_1| = 2$, 且 $\forall t_1 \in o_1, \lambda_{t_1} = \infty$ (Σ' 的输出部分具有一个二元分支冲突结构); ③ $|\cdot i \cap T'| = 1$ (Σ' 中具有一个回环); ④ $\forall n \in N \setminus \{i, o, o_1\}: |\cdot n| = 1, |n \cdot| = 1$ (前向和后向分支均为串行结构); ⑤ $|T'| = 2$, 令 T' 元素分别为 t_{t_1} 和 t_{t_2} , 如果 $t_{t_1} = \{o_1\}$, 则 $\cdot(\cdot(t_{t_2})) = \{o_1\}$, 反之如果 $t_{t_2} = \{o_1\}$, 则 $\cdot(\cdot(t_{t_1})) = \{o_1\}$ (Σ' 中具有 2 个时间变迁, 且分别位于前向和后向分支之中).

系统模型中可能存在若干瞬时变迁, 消除后并不影响系统时间特性, 同时消除瞬时变迁也可减小系统状态空间. 我们将其作为规则 1~规则 4 的重要辅助约简规则.

规则 5 瞬时变迁的消除.

Σ 中的一个瞬时变迁 t_i 可以消除, 但需满足: ① $(\cdot t_i) \cdot = \cdot t_i$ (t_i 是其前向和后向位置的惟一连接变迁); ② $|\cdot t_i| = |t_i \cdot| = 1$.

消除 t_i 后, 与 t_i 连接的弧同时被消除, t_i 的前向和后向位置融合为一个位置结点.

规则 6 多余位置的消除.

Σ 中的一个位置 s 可以消除, 但需满足: ① $|\cdot s| = 0, \cdot(s) = \{s\}$ (s 没有输入变迁, 并且是其输出变迁的惟一输入位置); ② $M_0(s) = 0$ (s 没有初始标记).

s 消除后, 其所有输出变迁和与它们相连的弧同时被消除.

3 约简算法

约简规则的应用条件已通过形式化定义详细地给出, 很容易实现, 所以算法中不再体现, 而是重点描述单输入、输出的 GSPN 可约简子网的探索方法. 串联型子网的探索算法(算法 1)如图 2 所示. 可约简子网的探索算法(算法 2)如图 3 所示.

由于串联型是模型中最常见的结构, 因此独立设计算法 1 先于算法 2 连续执行, 消除算法 2 执行前模型中可约简的所有串型结构以及运行后可能产生的局部串型结构. 算法 2 的主要设计原理: 分别以模型输入、输出端为起始点对模型结点进行前后向距离标定后, 任意一个可约简子网的输入位置距模型输入端距离是所有子网结点中最小的一个, 子网的输出位置是距模型输出端距离最小的一个. 约简算法采用广度优先搜索, 整体约简算法的复杂度由算法 2 决定, 其最坏情况下(网中不存在 GSPN 可约简子网)为 $O(n^2)$, $n = |S| + |T| + |F|$.

```

Input: 初始GSPN模型Σ
Output: Σ的一个可串联约简子网Σ'或Σ不可约简
Begin
  NotFound = True
  Do While (S ≠ ∅)
    选取 ∀s ∈ S
    If (|s| = |s'| = 1) ∧ ((s') = {s}) ∧ (M0(s) = 0)
      令 t1 = s, t2 = s'
      If |t1| = |t2| = 1 ∧ λt1 ≠ 0 ∧ λt2 ≠ 0 ∧ t1 ≠ t2
        N = {s, t1, t2, t1, t2}
        令 i = t1, o = t2; i, o 和 N 构成 Σ'
        NotFound = False
        Exit Do
      End If
    End If
    S = S \ {s}
  Loop
  根据NotFound输出Σ'或Σ不可串联约简
End

```

图 2 串联型子网的探索算法

```

Input: 初始GSPN模型Σ
Output: Σ的一个可约简的子网Σ'或Σ不可约简
Begin
  '开始标记距离, 令初始位置为b, 结束位置为c
  M = {b} '待标记结点集
  K = ∅ '已标记结点集
  d = 0
  Do While (M ≠ ∅)
    For Each m ∈ M
      dis(m) = d
    Next
    K = K ∪ M
    M = M \ (K ∩ M) '后向搜索
    d = d + 1
  Loop '标记距离完成
  '同样以c为起点, 前向搜索标记各结点到c的距离
  NotFound = True
  Do While (S ≠ ∅)
    选取 ∀o ∈ S '假设子网输出位置o
    S = S \ {o}
    令 P = {o}, T = o '子网位置与变迁集
    令 X = {o}, Y = o '临时位置与变迁集
    I = ∅ '子网输入位置集
    Do While (Y ≠ ∅)
      X = X ∪ Y
      X = X \ (P ∩ X)
      If X = ∅ Then Exit Do
      令 X = X ∪ I, P = P ∪ I
      Q = {q ∈ X | dq(q) = miny ∈ X (∪ do(y))}
      'db, dc 为结点到 b, c 的距离
      If |Q| = 1 Then I = Q Else I = ∅
      If M0(x) ≠ 0 | x ∈ X \ I Then Exit Do
      If |T'| < 2 ∧ I ≠ {o} Then I = ∅
      Y = (X \ I) ∪ X '输入点不再前向搜索
      If ∀dc(y) | y ∈ Y ≤ dc(c) Then
        I = ∅
        Exit Do 'o 为输出不成立
      End If
      令 Y = Y \ (T' ∩ Y), P = P ∪ X
      T' = T' ∪ Y
    Loop
    If I ≠ ∅ ∧ |P'| < |P| ∧ |t ∈ T | λt ≠ 0 | ≥ 2 Then
      NotFound = False
      Exit Do 'i 以及 P', T' 和 o 构成 Σ'
    End If
  Loop
  根据NotFound输出Σ'或Σ不可约简
End

```

图 3 GSPN 可约简子网的探索算法

4 实验与验证

实验以图 4 所示模型为基本模型结构, 操作系

统为 Windows2000 Professional,处理器为 Phenom X3 8400,主频为 2.1 GHz,内存为 1 GB. 每一项实验中子网探索 and 约简连续运行,直到模型不可约简为止. 各实验项模型为不同数量、图 4 所示的基本模型,去除变迁 t_{17} 后串联首尾,以此来判定算法在不同网络规模下的性能表现.

实验 1 基本模型 1 完全由 4 种规则结构构成,模型约简前后的情况见表 1,其中模型规模可用 $(n \times \text{结点数})$ 表示, n 为基本模型数量,最终约简后的模型均为一个包含 2 个位置、2 个时间变迁的简单模型. 由表 2 实验结果可以看出,算法可以在比较满意的时间内完成大规模模型约简,在一定的模型规模前提下,可满足在线获取系统响应时间的需求.

表 1 模型约简前后结果对比

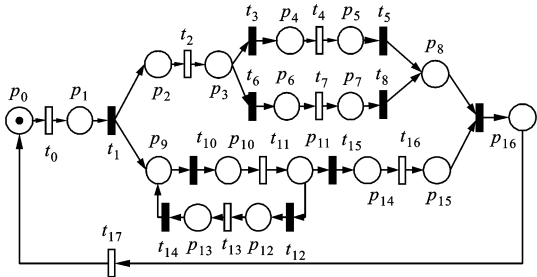
	模型规模	性能分析的	
		最大规模	性能分析数
模型 1	约简前	$n \times 35$	$n \times 35$
	约简后	2 个位置、2 个时间变迁	β
模型 2	约简前	$n \times 38$	$n \times 38$
	约简后	n 个 $p_1 \sim p_{16}$ 子网	n

注: β 表示直接获得响应时间.

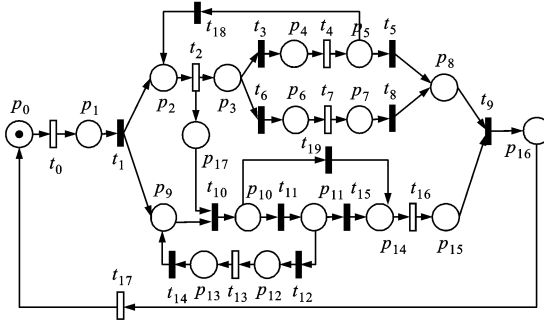
实验 2 基本模型 2 是在基本模型 1 的基础上增加了 2 个分支活动和一个同步位置,破坏了原有的规则子网结构,增加了模型的复杂度. 不规则结构使约简效率大幅降低,算法性能表现见表 2,但在一定的模型规模下算法仍具有实用性. 将原有模型分解为多个可独立进行性能分析的子网,便解决了性能分析所面临的状态空间爆炸问题. 模型约简的实验结果如图 5 所示.

表 2 模型规模与约简时间

模型数	初始结点数				约简时间/s	
	模型 1		模型 2		模型 1	模型 2
	S	T	S	T		
1	17	18	18	20	0.10	0.10
5	85	90	90	100	0.38	1.08
10	170	180	180	200	0.85	5.00
15	255	270	270	300	1.44	14.22
20	340	360	360	400	1.80	26.90
30	510	540	540	600	3.43	91.03
50	850	900	900	1 000	7.32	
70	1 190	1 260	1 260	1 400	13.48	>400
80	1 360	1 440	1 440	1 600	17.60	
100	1 700	1 800	1 800	2 000	26.69	



(a)基本模型 1



(b)基本模型 2

图 4 约简实验基本模型

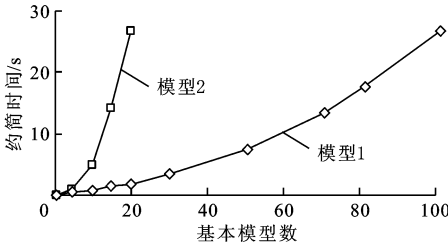


图 5 模型约简的实验结果

5 结束语

本文通过对 Web 服务组合系统实施 GSPN 建模,并保留时间特性条件下的模型约简,对约简模型性能进行求解,实现了编制模式下 Web 服务组合系统时间性能的快速获取. 今后的工作是,将本文的研究应用于构建原型系统之中,并进一步研究编排模式下定量性能的快速获取方法.

参考文献:

[1] 雷丽晖,段振华. 一种基于扩展有限自动机验证组合 Web 服务的方法[J]. 软件学报,2007,18(12): 2980-2990.
LEI Lihui, DUAN Zhenhua. An extended deterministic finite automata based method for the verification of composite Web services [J]. Journal of Software, 2007, 18(12): 2980-2990.
[2] 门鹏,段振华. 广义随机 Petri 网下的组合 Web 服务

- 建模与评价[J]. 西安交通大学学报, 2008, 42(8): 967-971.
- MEN Peng, DUAN Zhenhua. Modeling and evaluation of composite Web services based on generalized stochastic Petri net [J]. Journal of Xi'an Jiaotong University, 2008, 42(8): 967-971.
- [3] 郭玉彬, 杜玉越, 奚建清. Web 服务组合的有色网模型及运算性质[J]. 计算机学报, 2006, 29(7): 1067-1075.
- GUO Yubin, DU Yuyue, XI Jianqing. A CP-net model and operation properties for Web service composition [J]. Chinese Journal of Computers, 2006, 29(7): 1067-1075.
- [4] ZERGUINI L. Analysis of large workflow models [C] //Proceedings of 2002 IEEE International Conference on Systems, Man and Cybernetics. Los Alemitos, CA, USA: IEEE Computer Society, 2002: 188-194.
- [5] 田立勤, 林闯, 周文江. 随机 Petri 网模型中变迁的串、并联性能等价化简技术[J]. 电子学报, 2002, 30(8): 1134-1137.
- TIAN Liqin, LIN Chuang, ZHOU Wenjiang. Performance-equivalent simplification of sequence models and parallel models in stochastic Petri nets [J]. Journal of Electronics, 2002, 30 (8): 1134-1137.

(编辑 苗凌)