

随机高级 Petri 网的 层次模型和分层性能评价*

林 闯

吴建平 王鼎兴

(国家信息中心信息科学研究所, 北京 100045)

(清华大学计算机科学与技术系, 北京 100084)

摘要 这篇论文提供了一种将大型随机高级 Petri 网(SHLPN)模型构造成为独立子网的方法和技術. 这些子网可以独立地进行性能评价, 在原模型中子网可由非基本变迁进行性能等效替代, 以达简化状态空间的目的. 这种技术依赖于 SHLPN 模型的层次设计和一个标记穿越子网时平均延时时间的估算. 在这篇文章中, 四种子网界面结构的性能等效分解和合并以及子网平均延时时间的计算做了着重的讨论.

关键词 随机高级 Petri 网, 性能评价, 层次模型, 网络分解, 网络合并.

80 年代初, 随机 Petri 网 (SPN) 的提出^[1], 以及后来各种随机网的不断发展^[2], 和随机高级 Petri 网 (SHLPN) 的研究和应用, 为系统的性能评价提供了一种新的数学描述工具. 随机高级 Petri 网是高级 Petri 网的一种扩充, 它既保持了高级 Petri 网的良好模型特性, 又具有随机网与马尔可夫链 (MC) 同构的特性. 在 SHLPN 模型中, 可将多个同结构的子系统压缩成一个子系统来描述, 而且可将多个同类 (或说同行) 的标记 (token) 所形成的多个标识 (Marking) 合并成一个标识变量来表示, 以达到显著地简化系统状态空间的目的. 有关 SHLPN 的基本概念和分析方法, 可见文献^[4]. 从 SPN 到 SHLPN, 主要是从标记语义的表达上寻找更有抽象能力的性能模型描述工具. 如何从语法上或网的结构上寻找分解和简化网络分析复杂性的方法, 是网络分析方法研究的重要内容. 研究 SHLPN (或 SPN) 的分解和合并, 主要着眼于系统模型的性能一致性的代换. 系统模型的性能影响主要体现在系统实体操作运行及操作与环境的关系. 模型中的实体操作运行可分为串行操作和并行操作, 操作之间的关系又分为异步操作和同步操作. 在 SHLPN 中, 实体操作运行是由标记的流动来表示的, 操作关系是由变迁和位置的关系来描述的. 如果能对标记、变迁和位置的关系进行分解, 对标记流动过程进行抽象, 就可以简化随机网的性能分析, 达到缓解状态空间的指数性增长问题. 本文着重研究的是 SHLPN 模型 (当然也包括 SPN 模型, 后者是前者的特例) 的层次

* 本文 1993-11-03 收到, 1994-03-11 定稿

本课题得到国家自然科学基金的资助. 作者林闯, 1948 年生, 研究员, 主要研究领域为系统性能评价, 随机 Petri 网理论和应用, 人工智能, 计算机网络和并行处理系统. 吴建平, 1953 年生, 教授, 主要研究领域为计算机体系结构, 计算机网络, 协议工程学. 王鼎兴, 1938 年生, 教授, 博士生导师, 主要研究领域为计算机系统结构, 并行计算机, 分布式系统, 智能计算机.

本文通讯联系人: 林闯, 北京 100045, 国家信息中心信息科学研究所

化模拟和逐步分解的技术方法,进行分层性能评价逐步合并、抽象的计算方法。

1 模型的层次化模拟方法

SHLPN 适合描述复杂、大型系统,尤其是分布处理系统。对这些系统进行描述时,采用自顶向下逐步分解和自底向上逐步综合的分层模拟的方法。另外,在系统实体(进程、任务)的描述时注意实体类型的划分和确定。在相同类型的进程中,不同的进程有相同的特性。进程类型的划分便于系统模型的描述和系统模型的分析。

1.1 非基本变迁与层次模型设计

假定讨论的系统是分布处理系统,它有许多处理结点,它们松散地由一个通讯网络连接在一起,具有不可预测的延时。除了硬件和设置的物理分布外,数据和控制的分布是这类系统的基本特征。这些处理结点在服务类型、资源的保护以及有效性和可靠性方面具有高度的自主性,充分地表现出非集中控制的特性。

为了能有效地分层描述复杂系统,在 SHLPN 中采用两种变迁:基本变迁和非基本变迁。基本变迁不可再分解,同一般意义随机 Petri 网中的变迁相同,非基本变迁可以表示成一个子网,它有内部结构,它的实施表示子网的整体执行。在非基本变迁所表示的子网中还可以包括非基本变迁。这样,利用非基本变迁就可以实现自顶向下逐步分解的模型描述。当把一个子网作为一个非基本变迁时,亦即自底向上逐步综合分析时,非基本变迁的实施条件和实施过程同基本变迁相同。正因为如此,并不是任何一个子网都可以表示为一个非基本变迁,对子网的界面结构和通过标记的语义连接都有相应的要求。这一点在下一节将详细讨论。有关非基本变迁在分布软件系统和分布进程控制系统模型中的应用,可见参考文献[6, 7]。在使用 SHLPN 对分布处理系统进行描述时,首先可以考虑将每一个处理结点使用一个 SHLPN 子网来描述,即使用一个非基本变迁来模拟,尽量简化子网的界面结构,亦即简化非基本变迁之间的连接关系。所采用的技术和方法如同软件系统层次模块设计一样,使得模块层次之间仅有单方向的连接为最好。当某一个处理结点具有双向输入输出处理,可以考虑使用两个非基本变迁来描述不同方向的处理功能。然后,可以将非基本变迁展开成一个 SHLPN 的子网,精细地描述处理结点的模型。当处理结点内部功能复杂,不易描述清楚时,可以考虑按功能划分层次,将每个不同的处理功能用一个非基本变迁来描述,按层次将它们组织成该结点的子网。我们可以继续进行这种自顶向下的模型设计,直至完成。这种 SHLPN 模型的设计方法可以带来很多好处:①可以隐蔽子网内部结构和详细描述,使模型设计者集中在相应抽象层次的设计。②每个子网模型的描述可以并行进行,相同的子网不必重复设计。③使模型具有良好的层次结构,可为进一步的系统模型的分析奠定基础。

1.2 标记类型的划分与模型构造

在 SHLPN 模型中,标记类型的划分特别是同类标记的标定具有重要意义,不但有益于标识空间和不变量计算的简化,而且方便于系统模型结构的设计和系统性能的评价。在进行系统模型时,每种系统活动的对象或系统资源都可抽象描述成一种标记。这种标记可以有类型、标志和其它属性,这些属性可以用来表示活动对象所涉及的数据。模型中还有另一类标记,表示为系统环境信息,主要用于描述前一类标记活动之间的同步和相互影响信息。这类标记可以没有属性或仅有类型和标志属性。在很多情况下,例如表示活动对象与系统资源之

间分配情况,系统模型需要用复合标记表示.复合标记是用原子标记构成的向量来表示.

在 SHLPN 模型中,相同类型的标记具有下列两条性质:(1)它们在由包含它们颜色的位置组成的子网中运行,且总是由相同变量(弧标变量)表示.(2)它们具有相同的移动规则和相同的运行速率.

在 SHLPN 模型中,原子标记可以包括类型变量和标志变量.标志变量一般为旋转变量(如“模”变量)、置换变量(如集合元素置换)和标志函数变量.同一类型标记的标志变量具有相同的值域,同一时刻不同时标志变量取不同的值.

由于同类标记具有相同的行为,在构造模型可达树时,可以忽略同类标记之间的区别,仅描述类型标记状态,进而得到复合状态简化系统模型的状态空间.由于同类标记具有相同的 S-不变量,可以简化系统模型 S-不变量的计算.关于 SHLPN 模型的可达树和不变量的算法,可见文献[4].在系统模型实现中,标记的类型同系统进程的类型之间存在着对应关系^[7],标记类型的描述可提供系统模型实现和模拟的程序框架.在系统模型的设计中,可按标记类型划分模型的结构,确定子网的执行语义.

2 层次模型的性能评价和抽象合并

在 SHLPN 模型中,不是任何网结构都可以分割出一个子网用一个非基本变迁进行性能等效替代.在下列 4 种子网同外界接口的结构形式中可以进行层次模型的性能等效和子网的抽象合并:

(1)子网与外界接口仅由一个输入变迁和一个输出变迁组成,如图 1 所示.如有多个输入位置或多个输出位置和子网相连,则要求所有的输入位置都是子网输入变迁的输入位置,所有的输出位置都是子网输出变迁的输出位置.在图 1 中,可使用非基本变迁 t 替代包括 t_1 和 t_2 在内的子网,位置 P_{i1}, \dots, P_{in} 都是 t_1 的输入位置,位置 P_{j1}, \dots, P_{jm} 都是 t_2 输出位置.在替代中,子网同外界相连接的位置颜色不变,相应的弧标变量也保持不变.标记变量系数在子网中传递的结果要保证进出数量的平衡,亦即从 t_1 输入标记数量与从 t_2 输出标记的数量要满足相应的此例关系.替代变迁 t 的颜色应是子网中标记所经过变迁的颜色传递的结果.

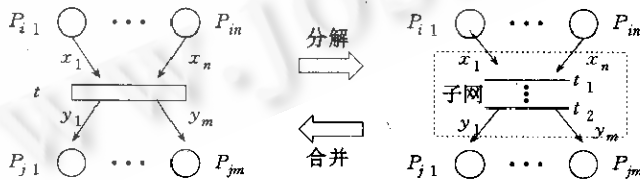


图1 情况(1)的非基本变迁与子网的代换

(2)子网同外界相连接仅有一个输入位置和一个输出位置.此时,输入位置和输出位置同子网内变迁的连接没有结构上的特殊要求.如图 2 所示的代换,输入、输出位置同子网变迁可有多种连接. P_1 到 t 的弧标代表原 P_1 到子网弧标的合并,同样 t 到 P_2 的弧标是原弧标的合并.在形式(2)中,如果位置 P_1 和 P_2 包含不同类型的标记,它们通过子网时有不同的行为和实施速率,可以考虑使用不同的非基本变迁来描述不同类型标记子网的替代,如图 3 所

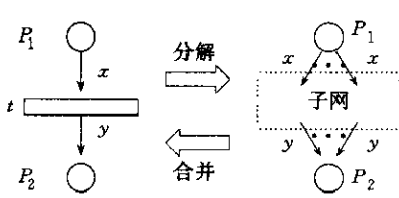


图2 情况(2)的一种代换

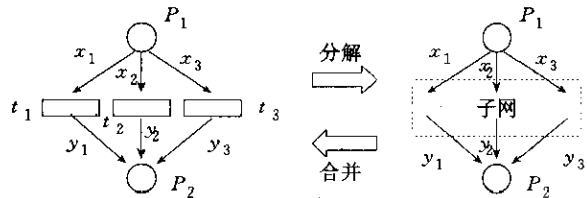


图3 情况(2)的另一种代换

示. 从本质上讲, 图 3 所示的子网替代可分割成多个图 2 所示的替代, 而多个图 2 所示的替代在整个网络结构中又可合成图 3 所示的替代结构. 在 SHLPN 模型的性能分析中, 我们着重考察每种实体(进程、任务)类型标记的动态行为. 因此, 网络结构的分解要充分注意实体标记类型的区分. 在情况(2)的替代中与情况(1)的替代要求一样, 要注意子网同外界连接位置颜色的一致性, 弧标变量和系数的一致和平衡, 以及替代变迁 t 的颜色传递要求.

(3) 子网同外界相连接仅有一个输入变迁和一个输出位置时, 可以进行如图 4 所示的替代.

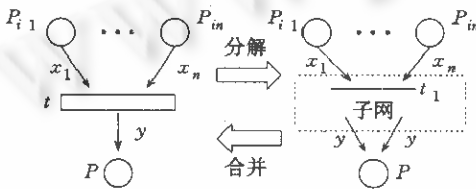


图4 情况(3)的代换方法

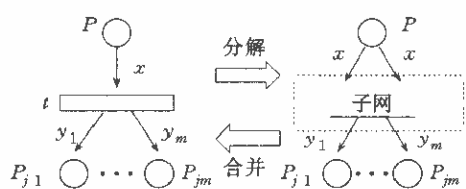


图5 情况(4)的代换方法

(4) 子网同外界相连接仅有一个输入位置和一个输出变迁时, 可进行如图 5 所示的替代.

在情况(3)和(4)的替代中, 位置和变迁的颜色以及弧标的要求和替代同情况(1)和(2)相同, 要注意保持一致性. 如果能在复杂的 SHLPN 模型中, 发现如情况(1)~(4)所示的子网, 而且系统性能评价中并不考察子网执行的详细状态, 那么就可将此类子网分离出来, 计算一个标记通过此子网的平均延时的时间随机变量, 将此随机变量作为替代变迁的实施时间, 就可得到如图 1~5 所示的性能等效模型, 简化原系统模型性能评价的复杂性. 此类子网的平均延时时间可在如图 6 所示的测试网模型中进行计算, 可附加一个测试变迁 t_1 , 构成一个封闭标记回路, 以使此系统保持有界性和历遍性.

在上述模型的初始标识设置中, 要在 P_{j1}, \dots, P_{jm} 和 P_2 中设置适当数量的标记, 这个数量与子网的缓冲量相关. 子网的缓冲量指保证子网中变迁能充分实施不出现死变迁的最少标记容量. 例如, 图 7(1)中的子网, 在 P_2 中初始设置一个标记就能使子网充分运行不出现死变迁, 因此其子网缓冲量为 '1'. 而在图 7(2)中的子网, 在 P_2 中至少设置 3 个标记才能使子网不出现死变迁, 其缓冲量为 '3'.

在测试网中, 主要是测试所设置的实体标记通过子网的平均延时时间, 缓冲量是指实体标记的缓冲量, 而不是控制标记(或说环境标记)的缓冲量. 另外要注意在测试网中, 每个变迁的实施速率都与输入位置中的实体标记的含量相关, 成正比. 由于随机 Petri 网模型的状态

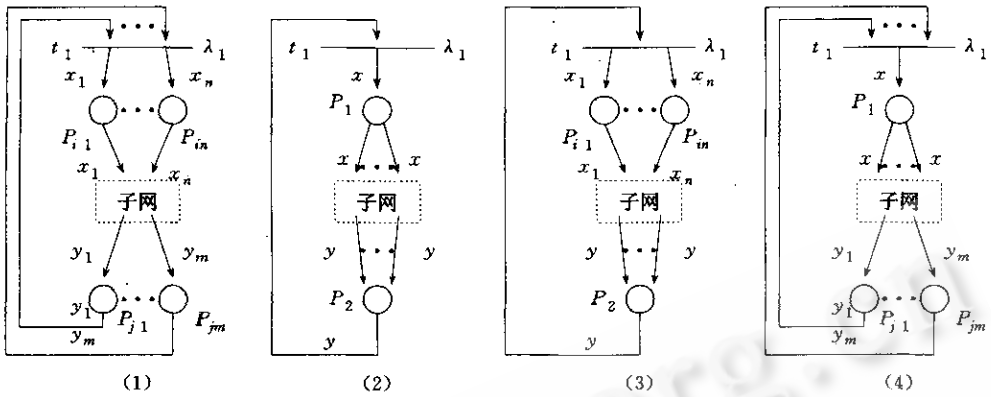


图6 求情况(1)–(4)子网平均延时的测试网

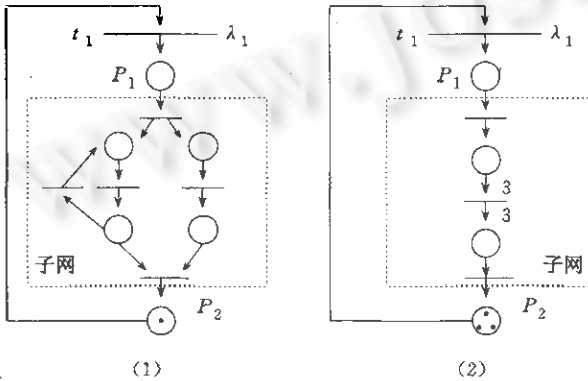


图7 子网初始标记设置与缓冲置

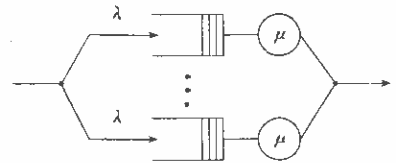


图8 排队网表示的随机网的子网

态空间复杂性会随着系统所含标记数量的增加而快速(指数性)的增长,因此为了减少测试网计算的复杂性,在测试网的初始标识的设置中仅设置与子网缓冲量相应的标记数量,而不设置大于子网缓冲量的标记数.这样计算出来的子网平均延时时间同样也适合子网输入位置含有大于缓冲量的标记的情况.这一点是由变迁的实施速率与输入实体标记数量成正比来保证的.换句话说,每个实体标记通过子网平均延时时间是不变的,它并不随着输入实体标记数量改变而变化.随机网的变迁实施速率与输入实体标记相关的关系可由图8的排队网来表示.

在随机网的结构中,多个相同的队列可由一个子网来表示.多个队列的执行可由变迁实施速率与输入实体标记数量成正比来表示多个队列执行的可能性,由变迁的实施延时时间来表示一个队列的服务延时时间.对于测试网,可求出其稳定状态概率,从而可获得测试变迁 t_1 的输入位置平均标记密度函数,使用 r_i 表示其输入位置中含有 i 个标识的概率,那么可得 t_1 的标识平均输入速率 f_1 ,亦即子网的吞吐率(throughput):

$$f_1 = \lambda_1 \sum_{i=1}^n i r_i, i \geq 1 \tag{1}$$

其中 n 是测试网 t_1 输入位置所含标记的最大数量(亦即缓冲量), λ_1 是 t_1 实施速率.

测试网所含平均标记数量 \bar{N} ,也可由其稳定状态概率求出.但由于测试网要求保证历

遍性和标记的有界性,有时是可借助 S—不变量计算平均标记数量.

根据 Little 规则,可求出测试网(包括 t_1)的每个标记平均延时时间

$$T' = \frac{\bar{N}}{f_1} \tag{2}$$

子网的每个标记平均延时时间等于 T' 减去变迁 t_1 的实施时间.

$$T = T' - \frac{1}{\lambda_1} \tag{3}$$

在原 SHLPN 模型中,经分割和合并后,可将子网性能等效替换为非基本变迁 t, t 的指数分布的随机实施速率变量为 $\lambda = 1/T$.

3 顺序子网的性能等效分解和合并

下面对最简单情况的子网——顺序子网的分解和合并做进一步的探讨,也可以说是一个例子,具体显示一个子网性能分解和合并的特性.在子网同外界连接同时满足接口情况(1)和(2),而且子网是由一系列顺序变迁组成时,这种性能等效代换的非基本变迁的平均延时时间的计算就相当简单,可以不使用公式(1)–(3),而直接计算.以图 9 中的随机网模型的性能等效代换为例,加以说明.

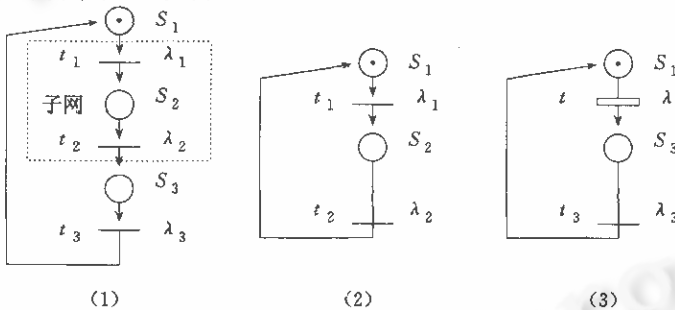
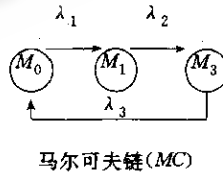


图9 顺序子网的等效代换

图 9(1)的随机网模型直接求解,结果如下:

状态	S_1	S_2	S_3	稳定概率
M_0	1	0	0	P_1
M_1	0	1	0	P_2
M_2	0	0	1	P_3



所得稳定状态方程

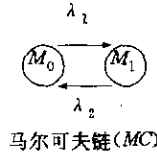
$$\begin{cases} \lambda_1 p_1 = \lambda_3 p_3 \\ \lambda_2 p_2 = \lambda_1 p_1 \\ p_1 + p_2 + p_3 = 1 \end{cases}$$

求解稳定状态概率

$$p_3 = \frac{\lambda_1 \lambda_2}{\lambda_1 \lambda_2 + \lambda_2 \lambda_3 + \lambda_1 \lambda_3}$$

如果将由 t_1 和 t_2 组成的子网分割出来,由于子网比较简单不用测试变迁就可组成测试子网,如图 9(2).解此测试网,可有如下结果:

状态	S_1	S_2	稳定概率
M_0	1	0	P_1
M_1	0	1	P_2



所得稳定状态方程

$$\begin{cases} \lambda_1 p_1 = \lambda_2 p_2 \\ p_1 + p_2 = 1 \end{cases}$$

求得稳定状态概率

$$p_1 = \frac{\lambda_2}{\lambda_1 + \lambda_2}$$

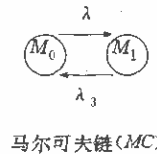
求得子网图 9(2) 的标记流平均输入速率(见公式(1)):

$$\lambda = \lambda_1 \cdot p_1 = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2}$$

由于此子网中有一 S-不变量包括所有位置,子网所含标记数量就是初始标识中的标记数量, $\bar{N}=1$. 根据公式(2), λ 就是替代变迁 t 的平均实施速率. 经非基本变迁替代化简后的等效性能模型,如图 9(3)所示.

图 9(3) 的随机网模型,结果如下:

状态	S_1	S_3	稳定概率
M_0	1	0	P_1
M_1	0	1	P_3



所得稳定状态方程

$$\begin{cases} \lambda p_1 = \lambda_3 p_3 \\ p_1 + p_3 = 1 \end{cases}$$

求得稳定状态概率

$$p_3 = \frac{\lambda}{\lambda + \lambda_3} = \frac{\lambda_1 \lambda_2}{\lambda_1 \lambda_2 + \lambda_1 \lambda_3 + \lambda_2 \lambda_3}$$

所得的表示位置 S_3 中包含一个标记而其它位置中不包含标记状态的概率 p_3 , 两种解法结果完全相同,后一种解法状态空间简化.

通过这个例子,可以看到,当子网是由一系列顺序变迁 $t_i (1 \leq i \leq k)$ 组成时,它的等效非基本变迁 t 的平均实施速率 λ 可由下列公式给出:

$$\frac{1}{\lambda} = \sum_{i=1}^k \frac{1}{\lambda_i} \tag{4}$$

其中 λ_i 是子网中变迁 t_i 相联的实施速率.

在上述例子中,已证明: $k=2$ 时,确有 $\lambda = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2}$, 表明公式(4)正确. 使用数学归纳法对子网中的变迁数量进行归纳证明. 假定 $k=n-1$ 时公式(4)成立,亦即有

$$\frac{1}{\lambda'} = \sum_{i=1}^{n-1} \frac{1}{\lambda_i}$$

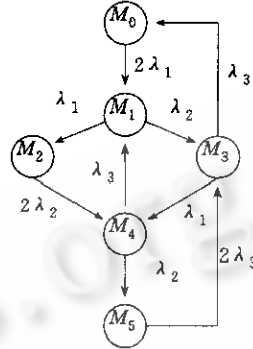
可将 $n-1$ 个变迁使用一个非基本变迁 t' 替代, t' 的实施速率为 λ' . 当 $k=n$ 时,可将子网看成由变迁 t' 和 t_n 组成,因已证明 $k=2$ 时成立,所以有

$$\frac{1}{\lambda} = \frac{1}{\lambda'} + \frac{1}{\lambda_n} = \sum_{i=1}^n \frac{1}{\lambda_i}$$

亦即,公式(4)成立.

公式(4)不但在子网缓冲量等于 1 时成立,而且在子网初始标识中标记数为任意有限正整数时也成立. 理论上的解释可见图 8 及其说明. 下面就图 9 随机网模型,做一个具体的演示,设位置 S_1 中包括 2 个标记. 图 9(1)的随机网模型的直接求解结果如下:

状态	S_1	S_2	S_3	稳定概率
M_0	2	0	0	P_1
M_1	1	1	0	P_2
M_2	0	2	0	P_3
M_3	1	0	1	P_4
M_4	0	1	1	P_5
M_5	0	0	2	P_6



马尔可夫链(MC)

得稳定状态概率方程

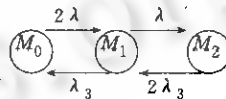
$$\begin{cases} 2\lambda_1 p_1 = \lambda_3 p_4 \\ (\lambda_1 + \lambda_2) p_2 = 2\lambda_1 p_1 + \lambda_3 p_5 \\ 2\lambda_2 p_3 = \lambda_1 p_2 \\ (\lambda_3 + \lambda_1) p_4 = \lambda_2 p_2 + 2\lambda_3 p_6 \\ (\lambda_3 + \lambda_2) p_5 = 2\lambda_2 p_3 + \lambda_1 p_4 \\ 2\lambda_6 p_6 = 2\lambda_2 p_5 \\ p_1 + p_2 + p_3 + p_4 + p_5 + p_6 = 1 \end{cases}$$

求解稳定状态概率

$$p_6 = \frac{\lambda_1^2 \lambda_2^2}{(\lambda_1 \lambda_2 + \lambda_3 (\lambda_1 + \lambda_2))^2}$$

如果直接从简化等效模型图 9(3)求解,可得

状态	S_1	S_3	稳定概率
M_0	2	0	P_1
M_1	1	1	P_2
M_2	0	2	P_6



马尔可夫链(MC)

求解稳定状态概率

$$p_6 = \frac{\lambda^2}{(\lambda + \lambda_3)^2}$$

根据公式(4)将 $\lambda = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2}$ 代入上式得

$$p_6 = \frac{\lambda_1^2 \lambda_2^2}{(\lambda_1 \lambda_2 + \lambda_3 (\lambda_1 + \lambda_2))^2}$$

结果和意义同上面复杂模型的解相同,可见子网的平均延时时间与子网初始标识的标记数量(大于等于缓冲量)的设置无关.

4 结论

本文提出的 SHLPN 模型的设计方法和层次模型的性能等效代换方法对大型、复杂系统的随机网模型的性能评价的化简有实际意义,它是解决随机网复杂性随实际系统增大而

指数性增长的一种有效途径。

这种方法已在计算机网络协议的 SHLPN 模型性能评价中应用,取得了实际效果^[8]。当然,仅有层次模型的性能等效代换方法是不够的。如何解决多进程资源共享和多任务异步操作 SHLPN 模型的分解和合并,进行性能等效代换是有待我们研究的下一个问题。

参考文献

- 1 Molloy M K. Performance analysis using stochastic Petri nets. IEEE Trans. on Comp. 1982,C-31:913-917.
- 2 Pagnoni A. Stochastic nets and performance evaluation. LNCS 254, 1986.
- 3 Lin Chuang, Marinescu D C. Stochastic high-level Petri nets and applications. IEEE Trans. on Comp. , 1988,37(7), Also in Jensen K, Rozenberg G eds. , High-Level Petri Nets, Springer-Verlag, 1991.
- 4 Lin Chuang, Marinescu D C. On the analysis of stochastic high-level Petri net models. International Journal, Microelectronics and Reliability, 1991,31(4).
- 5 Berthlot G. Transformations and decompositions of nets. LNCS 254, 1986.
- 6 Yau S S, Caglayan M U. Distributed software system design representation using modified Petri nets. IEEE Trans. on Software Eng. , 1983,6:733-745.
- 7 Bruno G, Movchetto G. Process-translatable Petri nets for the rapid prototyping of process control systems. IEEE Trans. on Software Eng. , 1986,SE-12(2):346-357.
- 8 Lin Chuang, Marinescu D C. Translation of modified predicate transition net models of communication protocols into simulation programs. Proc. of 1986 Winter Simulation Conference, IEEE Computer Society, USA, Dec. 1986.

HIERARCHICAL MODELING AND PERFORMANCE EVALUATION OF STOCHASTIC HIGH-LEVEL PETRE NETS

Lin Chuang

(Information Science Institute, State Information Center, Beijing 100045)

Wu Jianping Wang Dingxing

(Department of Computer Science and Technology, Tsinghua University, Beijing 100084)

Abstract A methodology is presented for the performance evaluation of large stochastic high-level Petri net (SHLPN) models that are structured into independent submodels. These subnets are independently evaluated and then are substituted by the nonprimitive transitions as performance equivalence in the original model to achieve a state space reduction. This is based on the hierarchical design of SHLPN models and on the estimated mean delay time for a token traffic process in subnets. The paper investigates four subnet interfaces for decomposition and composition as performance and computation of the mean response time for subnets.

Key words Stochastic high-level Petri nets, performance evaluation, hierarchical modeling, net decomposition, net composition.