第33卷 第21期 计 算 机 工 程 2007年11月
Vol.33 No.21 Computer Engineering November 2007
·博士论文· 文章编号：1000—3428(2007)21—0015—03 文献标识码：A 中图分类号：TP311

管道-过滤器风格的并行软件可靠性建模

吴 震，孟洛明
(北京邮电大学网络与交换国家重点实验室，北京 100876)

摘 要：当前基于体系结构的软件可靠性评估技术存在2个问题：不适合具有并行性质的软件和难以分析部件级的失效恢复行为。针对这2个问题，以管道-过滤器风格的并行软件为研究对象，该文使用基于时间的Petri网分析了管道-过滤器风格的并行软件运行阶段可靠性，并根据Petri网模型的特点，提出了一种分解模型和近似求解方法。数字实例证明了该方法的有效性和简便性。

关键词：软件可靠性；管道-过滤器风格；并行软件；Petri网

Reliability Modeling for Parallel Software with Pipe-filter Style

WU Zhen, MENG Luo-ming
(State Key Laboratory of Networking and Switching, Beijing University of Posts and Telecommunications, Beijing 100876)

【Abstract】 Current architecture-based software reliability evaluation is suitable for sequential software but not parallel software, and it is difficult to analyze the failure and restart behavior of component. To solve these problems, parallel software with pipe-filter style is selected as the research object and a new model is proposed to analyze the reliability of it by using time-based Petri nets. According to the characteristics of its model, a specifical solution to decompose and compute approximately the Petri net model is presented and it can avoid the explosion of state space. Numerical examples show the method is valid and convenient.

【Key words】 software reliability; pipe-filter style(PFS); parallel software; Petri net

当前基于体系结构的可靠性评估技术[1~4]主要有2种：(1)路径法，列举软件的可能执行路径，计算路径上各模块可靠度的乘积来获得路径可靠性，最后利用路径执行的概率求解整个软件的可靠性；(2)状态法，把软件的运行看作是一个模块间的转移过程，利用某些状态空间模型(如CTMC)来建模求解。

上述方法存在2个问题：(1)只考虑了顺序性质的软件[1]。这意味着系统在一个时间只有一个部件被执行，执行完毕后交出控制权给下一个部件，失效只能在部件被执行时发生。顺序性质是当前方法的基础，如在状态法中，往往用在部件执行，作为状态空间划分的依据。但在现实中还有不少软件不符合顺序性质，例如管道-过滤器风格(pipe-filter style, PFS)的软件，过滤器启动后处于idle状态，当接收到外界消息后转入busy状态进行处理，之后再转入idle状态等待新消息。即在软件运行时，若非因为失效，过滤器将始终处于运行中，而且由于管道的缓冲作用，在一个时间会出现多个过滤器同时处理消息的可能，从而具有某些并行特征，对此当前方法难以解决。(2)难以描绘某些可靠性技术。例如在运行阶段，部件可靠性已比较高，但也可能因为某些异常情况失效，这时系统诊断到部件崩溃，会采取自动重启进行恢复。对这种部件级的失效恢复行为，当前方法难以分析。

软件体系结构存在多种风格(style)，如CS风格等。不同风格在实现并行上各有特点。本文以PFS为研究对象，提出一种综合部件级失效恢复行为求解并行软件可靠性的方法。

1 PFS的特征

PFS由管道和过滤器组成。代表数据加工的过滤器是具有一组输入和一组输出的部件，通过管道协同工作。过滤器从数据源接收消息，经过内部处理后输送到缓冲消息队列，并被传送到下一个过滤器。从系统的角度来看，各个过滤器可以对消息进行局部变换，产生部分计算结果。管道是过滤器之间的连接器，是一个服务规则符合FCFS的数据缓冲区，存放待处理的消息。PFS有以下特征：

(1)过滤器是独立运行的部件，除了输入输出外，过滤器不受任何其他过滤器运行的影响。即使对于多次处理，过滤器自身也是无状态的。

(2)PFS中结果的正确性不依赖于各个过滤器运行的次序。对于原始输入，尽管其输出获得具有顺序要求，但在系统工作时，过滤器在输入后独立地完成自己的计算，完整计算过程包含在各个过滤器的拓扑结构中。

(3)PFS具有自然并行性，最小并行单位是过滤器。通过管道缓冲，各个过滤器可以同时运行，但在过滤器内部，则按处理逻辑顺序执行。

(4)过滤器并行形成了任务的并行。任务可被定义为在每一个输入向量上进行的运算和交互过程。与过滤器类似，任务间并行、任务内串行。

PFS最简单的类型是管线，它把过滤器严格地限制为单输入单输出，系统拓扑只能是线性序列。如果过滤器输出多于一个，系统就成为可能包含回馈的复杂拓扑。在这种情况下，消息在过滤器之间的转移可通过转移概率矩阵 $\boldsymbol{Q}$ 来描述。

基金项目：国家自然科学基金资助项目(90604020, 90604021)
作者简介：吴 震(1976－)，男，博士，主研方向：通信软件与网络管理；孟洛明，教授、博士生导师
收稿日期：2006-12-08 **E-mail：**wuzhen76@126.com

为了增强模型的普适性，本文重点对复杂拓扑 PFS 系统建模。为了便于分析，做如下假定：(1)任一过滤器失效，意味着系统失效；(2)输入管道的缓冲空间无限大，意味着不会因缓冲区不够而损失消息。

2 PFS 运行过程建模

2.1 过滤器运行过程建模

近年来，时间 Petri 网得到了迅速的发展，如 GSPN, DSPN, SHLPN 等。它保留了传统 Petri 网的优点，擅长用具体形象的方式描述复杂场景，也为复杂系统性能、可靠性分析提供了建模工具。详细理论和应用可参考文献[5~6]。

过滤器运行过程 Petri 网模型如图 1 所示。其中，圆圈表示位置；黑点表示 Token；中空宽竖条表示负指数分布的变迁；中间灰色的宽竖条表示任意分布的变迁；实心窄竖条表示瞬时变迁。

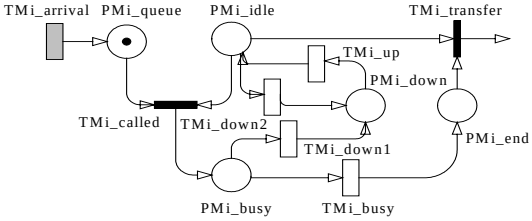

图 1　过滤器运行过程的 Petri 网模型

各位置和变迁的含义如下：

(1)P_{Mi_idle}：表示过滤器i处于idle状态，等待来自外界或其他过滤器产生的输入；

(2)P_{Mi_busy}：表示过滤器i从管道接收到数据消息，转入消息处理状态；

(3)P_{Mi_end}：表示过滤器i处理完消息后的瞬时状态；

(4)P_{Mi_down}：表示过滤器i在idle状态或busy状态时因为某些故障而失效；

(5)P_{Mi_queue}：表示过滤器i的输入管道，是一个符合FCFS规则的缓冲队列；

(6)$T_{Mi\_arrival}$：建模数据消息的到达过程，概率分布将在第 3 节中进行分析；

(7)T_{Mi_called}：建模当输入管道存在待处理消息时，过滤器i从idle向busy状态的转化；

(8)T_{Mi_busy}：建模过滤器i的处理过程，假定符合参数为u_i的负指数分布；

(9)T_{Mi_down1}：建模过滤器i从busy状态转为失效状态的过程，假定符合参数为λ_{i1}的负指数分布；

(10)T_{Mi_down2}：建模过滤器i从idle状态转为失效状态的过程，假定符合参数为λ_{i2}的负指数分布；

(11)T_{Mi_up}：建模过滤器i的失效恢复行为，重新转入idle状态，假定符合参数为λ_{i3}的负指数分布；

(12)$T_{Mi\_transfer}$：建模处理完毕后，过滤器i向其他过滤器传送消息，可能存在 0 个或多个transfer变迁。当为多个变迁时，实施规则由随机开关控制，参数取决于转移概率矩阵$\boldsymbol{Q}$；当为 0 个变迁时，表示该过滤器属于系统输出端。

2.2 系统运行过程建模

系统运行过程模型重点建模多个过滤器之间的协作。由于PFS具有任务间并行、任务内串行的特点，因此可以借鉴状态法，通过转移矩阵$\boldsymbol{Q}$来描述过滤器之间的数据传送。图 2 给出了一个含有 3 个过滤器的系统运行过程的Petri网模型。其中，虚线方框内的子图分别对应过滤器 1, 2, 3 的Petri网模型，为了便于观察，只把过滤器子模型中 3 个与消息传送相关的位置P_{Mi_queue}，P_{Mi_end}和P_{Mi_idle}绘制出来。$T_{arrival}$表示系统运行的外部数据，其产生假定服从速率θ的泊松过程。由于系统输入端口可能有多个，因此外部数据可以以不同的概率从多个过滤器进入系统，这通过多个瞬时变迁T_{toMi}来建模，其实施概率分布通过随机开关表达。按照泊松过程的分流特性，可以得到T_{toMi}的速率θ_i。T_{MitoMj}等多个变迁把图 1 的$T_{transfer}$具体化，用于表示过滤器之间的转移概率。值得注意的是，瞬时变迁$T_{endinM2}$表示当M2 处理完消息后，整个任务可能就在M2 完成，这时模块返回idle状态，且不再向其他过滤器传送消息。这时根据Petri网的分析方法，系统的可靠性可以表达为

$$R_{sys} = 1 - \sum_{S_i \in \Omega'} Pr(S_i) \tag{1}$$

其中，Ω'表示潜在标记过程状态空间的子集，每个元素都表示假设 1 约定的系统失效状态；$Pr(S_i)$表示该状态出现的稳态概率。

3 PFS 模型的近似求解

图 2 模型有 2 种分析方法：(1)根据模型构造相应的马尔科夫链(MC)直接求解；(2)对模型使用分解、化简等技术进行求解。一般情况下，图 2 模型是一个多维的 MC，随着过滤器总数 n 的增大，MC 的状态空间呈指数增长，可能会超出一般计算机的存储和计算能力。因此，可行解法是分解模型和分析子模型之间相互关系进行近似求解。文献[7~8]提出了几种化简分解方法。

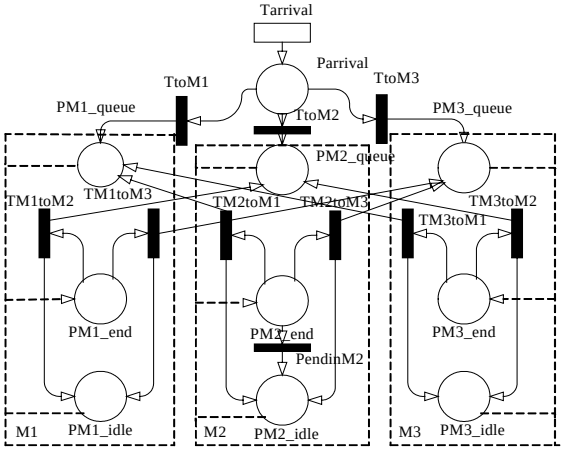

图 2　系统运行过程的 Petri 网模型

对PFS而言，过滤器运行相当独立，外界对过滤器i运行的影响主要体现在消息到达过程上。因此，可以通过分析消息到达过程，把图 2 分解成多个过滤器运行的Petri网子模型，这将极大减少计算消耗。当软件处于运行阶段时，各过滤器可靠度比较高，如果忽略过滤器失效带来的消息损失，那么软件运行符合开环Jackson排队网络的 3 个条件：(1)过滤器处理时间服从负指数分布；(2)马尔科夫路由；(3)外界消息到达服从泊松过程。因此，可把图 2 模型近似为一个开环Jackson排队网络。这种忽略会导致模块可靠性估计偏于保守。若系统内存在n个过滤器，每个过滤器的消息到达速率为λ_{i0}，按Jackson排队网络，各过滤器的消息到达速率可按如下方程求解：

$$\Lambda = \Psi (\boldsymbol{I} - \boldsymbol{Q})^{-1} \tag{2}$$

其中，$\boldsymbol{I}$为恒等矩阵；$\Lambda = (\lambda_{10}, \lambda_{20}, \cdots, \lambda_{n0})$；$\Psi = (\theta_1, \theta_2, \cdots, \theta_n)$，$\boldsymbol{Q}=[q_{ij}]_{n\times n}$。

在得到消息到达速率后，就可逐个分析图 1 的 Petri 网子模型。对于过滤器 i，以 X 表示过滤器本身的状态，以 Y 表示管道中消息数。X 的取值 k=0,1,2，分别表示 idle, busy 和失效状态，Y 的取值 $j\in[0, +\infty]$。则(X,Y)可用来表示过滤器 i 运行的所有状态。这时图 1 子模型可以用图 3 的 MC 表示。

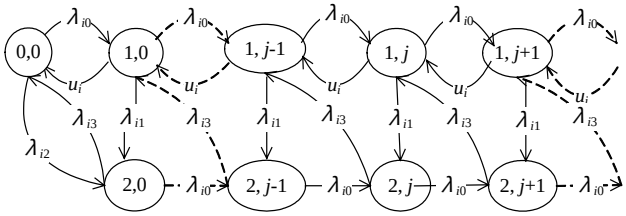

图 3　过滤器 i 运行过程 MC

根据图 3，列出稳态平衡方程如下：

$$\begin{cases}(\lambda_{i0}+\lambda_{i2})P_{0,0}=u_iP_{1,0}+\lambda_{i3}P_{2,0} & k=0 \quad j=0\\ (\lambda_{i0}+\lambda_{i1}+u_i)P_{1,0}=\lambda_{i0}P_{0,0}+u_iP_{1,1}+\lambda_{i3}P_{2,1} & k=1 \quad j=0\\ (\lambda_{i3}+\lambda_{i0})P_{2,0}=\lambda_{i2}P_{0,0}+\lambda_{i1}P_{1,0} & k=2 \quad j=0\\ (\lambda_{i0}+\lambda_{i1}+u_i)P_{1,j}=\lambda_{i0}P_{1,j-1}+u_iP_{1,j+1}+\lambda_{i3}P_{2,j+1} & k=1 \quad j>0\\ (\lambda_{i3}+\lambda_{i0})P_{2,j}=\lambda_{i0}P_{2,j-1}+\lambda_{i1}P_{1,j} & k=2 \quad j>0\end{cases} \tag{3}$$

通过母函数分析方法，可得过滤器 i 的稳态失效概率为

$$G_2(1)=\frac{\lambda_{i1}\lambda_{i2}+\lambda_{i2}u_i+\lambda_{i0}\lambda_{i1}-\lambda_{i0}\lambda_{i2}}{(\lambda_{i1}+u_i)(\lambda_{i2}+\lambda_{i3})} \tag{4}$$

则系统内存在 n 个过滤器的情况下，系统的可靠度为

$$R_{\text{sys}}=\prod_{i=1}^{n}[1-G_2(1)] \tag{5}$$

4　模型的分析应用

过滤器模型对各个参数敏感度的重点在于：(1)到达速率和处理速率的变化对过滤器可靠性的影响；(2)过滤器本身质量对可靠性的影响。图 4 为当λ_{i0}=100, λ_{i1}=0.1, λ_{i2}=0.001, λ_{i3}= 10 时，过滤器可靠性随处理速率u的变化曲线。可以看出，过滤器运行过程中的可靠性与工作负荷紧密相关，这与其他基于体系结构的软件可靠性模型是一致的，它们通常也要考虑部件使用强度，如访问次数、使用频率等。开始时，过滤器可靠性随u的增长快速上升，当u高于 80~120 后，过滤器可靠性开始缓慢上升，系统进入平稳状态。

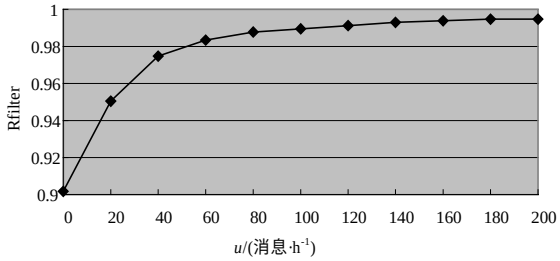

图 4　过滤器可靠性随处理速率的变化曲线

图 5 为λ_{i0}=100, λ_{i2}=0.001, λ_{i3}=10, u_i=100 时，过滤器可靠性随busy状态失效速率λ_{i1}的变化曲线。可以看出，过滤器可靠性与自身的质量紧密相关，在一般情况下，由于λ_{i2}比较小，因此，当处理速率u_i与到达速率λ_{i1}接近的情况下，过滤器可靠性与自身质量基本呈线性关系。

以图 2 系统为例讨论模型应用。外部数据按照速率为 100task/h的泊松过程到达，并可从多个过滤器进入系统，概率分布为q_1=0.5, q_2=0.2, q_3=0.3。系统有一个出口M_2。转移矩阵为

$$\boldsymbol{Q}=\begin{bmatrix}0 & 0.2 & 0.8\\ 0.1 & 0.2 & 0.2\\ 0.3 & 0.7 & 0\end{bmatrix}$$

以上参数代入式(2)，可得各过滤器的到达速率λ_{10}=119.7, λ_{20}=200, λ_{30}=165.8。表 1 给出了各过滤器参数。代入式(4)，可得各过滤器的可靠度R_{M1}=0.992032, R_{M2}=0.993370, R_{M3}=0.978 035。代入式(5)，则R_{sys}=0.9638。

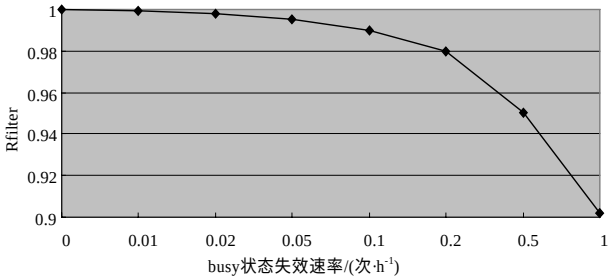

图 5　过滤器可靠性随 busy 状态失效速率的变化曲线

各过滤器参数如表 1 所示。

表 1　各过滤器参数

	M1	M2	M3
busy状态失效速率/(次·h^{-1})	0.1	0.05	0.1
idle状态失效速率/(次·h^{-1})	0.001	0.001	0.005
恢复速率/(次·h^{-1})	10	20	5
处理速率/(消息数·h^{-1})	150	200	150

5　结束语

当前基于体系结构的软件可靠性评估技术得到了很大发展，但既不适合对并行软件的分析，也不易解决部件级的失效恢复行为。本文以 PFS 并行软件为例，使用 Petri 网进行了可靠性建模，并根据其在运行阶段的特点提出了一种分解求解方法，有效地简化了计算。然而体系结构有多种风格，在实行并行上各有特点，例如还存在大量以信息交互方式实现的并行软件，对这些软件进行可靠性建模还有待进一步研究。

参考文献

1 Gokhale S S, Lyu M R. A Simulation Approach to Structure-based Software Reliability Analysis[J]. IEEE Trans. on Software Eng., 2005, 31(8): 643-656.

2 Yacoub S. A Scenario-based Reliability Analysis Approach for Component-based Software[J]. IEEE Trans. on Reli., 2004, 54(3).

3 蔡开元，白成刚，钟小军. 构件软件系统的可靠性评估模型简介[J]. 西安交通大学学报, 2003, 37(6): 551-554.

4 Gokhale S, Trivedi K S. Reliability Prediction and Sensitivity Analysis Based on Software Architecture[C]//Proc. of the 13th Int'l Symp. on Software Reli. Eng.. 2002.

5 林　闯. 随机 Petri 网和系统性能评价[M]. 北京: 清华大学出版社, 2005.

6 Horvath A, Telek M. Time Domain Analysis of Non-markovian Stochastic Petri Nets with PRI Transitions[J]. IEEE Trans. on Software Eng., 2002, 28(10): 933-939.

7 李雅娟，林　闯. 随机高级 Petri 网在异构系统中的应用[J]. 电子学报, 2004, 32(11): 1839-1842.

8 Mura I, Bondavalli A. Markov Regenerative Stochastic Petri Nets to Model and Evaluate Phased Mission Systems Dependability[J]. IEEE Trans. on Computer, 2001, 50(12): 1339-1347.