

基于非马尔可夫随机 Petri 网的软件再生建模与分析

孟海宁 齐 勇 侯 迪

(西安交通大学电子与信息工程学院 西安 710049)

摘 要 软件老化是影响软件系统可靠性的重要潜在因素,软件再生作为一种主动预防性的软件容错技术是解决软件老化问题的主要手段.以往的随机 Petri 网再生模型假定所有变迁的实施时间服从指数分布.针对变迁的实施时间服从确定性分布或一般性分布的情况,文中提出了一种用非马尔可夫随机 Petri 网建立软件再生模型的方法.该方法采用马尔可夫再生理论对模型进行分析,并给出模型的瞬态解和稳态解.仿真实验表明:选择合适的软件再生周期,可以有效地降低存在老化的软件系统的平均宕机成本,提高系统的可用性和可靠性.

关键词 软件老化;软件再生;软件可靠性;非马尔可夫随机 Petri 网;马尔可夫再生理论

中图法分类号 TP311

Modeling and Analysis of Software Rejuvenation Based on Non-Markovian Stochastic Petri Nets

MENG Hai-Ning QI Yong HOU Di

(School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049)

Abstract Software aging is an important potential factor that affects the software reliability. As a proactive and preventive software fault tolerant technique, software rejuvenation is a main method for counteracting software aging. Almost all developed software rejuvenation models based on stochastic Petri Nets assume that all the firing times submit to exponential distributions. Aiming at the firing times submit to determined or general distributions, a software rejuvenation modeling method is proposed using Non-Markovian Stochastic Petri Nets. In addition, model is solved for both steady and transient state via Markov regenerative theory. The numeric experiment results show that selecting optimal software rejuvenation schedule can improve systematic availability and reduce downtime cost.

Keywords software aging; software rejuvenation; software reliability; non-Markovian stochastic Petri nets; Markov regenerative theory

1 引 言

以往研究表明:系统在长期运行一段时间后会
出现系统性能下降或停机的现象,这种情况被称为
软件老化.许多高可用性和可靠性的应用软件都有

老化现象^[1].软件再生技术作为一种主动性的容错
技术成为解决软件老化的重要手段.软件再生的定
义于1995年由AT&T实验室的Huang^[1]等人正
式提出.软件再生是指在系统故障发生之前,通过周
期性地暂停软件的运行,清除持续运行系统的内部
状态,重新启动并恢复为干净的初始状态或中间状

态,以抢先停止将来可能发生的更严重的故障. 软件再生技术已成为系统可靠性研究方面的一个重要课题^[2]. 对于具有周期波动性的负载业务类型的长期运行软件系统(如教学视频点播系统,应用服务器中间件系统等),采用软件再生技术,周期性地暂停软件运行,恢复系统到初始健康状态,可以提高系统可靠性和可用性.

软件再生的研究有两种不同的方法,即基于模型的研究方法和基于测量的研究方法. 其中基于模型的方法主要是对软件系统的状态关系建立模型,主要有连续时间马尔可夫模型^[1]、半马尔可夫模型^[3]、马尔可夫决策过程模型^[4]、基于两级软件再生的模型^[5-6]及随机 Petri 网(Stochastic Petri Nets, SPNs)模型^[7]等. 软件再生技术所涉及的主要问题是选择合适的软件再生周期的问题,即对系统何时采取再生技术以及如何采取再生技术的问题. 过高频率的软件再生会增大系统平均宕机时间和宕机成本,过低频率的软件再生则不能保证起到应有的效用.

Petri 网^[8]是研究并行系统和分布式系统的一种强有效的形式化工具,能够很好地刻画系统的动态行为、分析系统的性能. 以往基于随机 Petri 网再生模型的研究方法,是将模型转化为具有马尔可夫特性的随机过程进行分析,但模型中变迁实施时间的随机变量仅考虑服从指数分布的情况. 本文提出一种用非马尔可夫随机 Petri 网对再生系统建模的方法,模型中变迁的实施时间可以不仅仅服从指数分布,可以服从确定性分布或一般性分布,然后应用马尔可夫再生理论对模型进行分析求解,并给出模型的瞬态解和稳态解. 仿真实验表明:选择合适的软件再生周期,可以有效地降低系统的平均宕机成本,提高系统的可用性和可靠性.

2 基本概念及分析方法

为了讨论用非马尔可夫随机 Petri 网对软件再生系统建模和分析的方法,先给出相关定义及应用软件再生理论对一般系统的非马尔可夫随机 Petri 网模型的分析方法.

定义 1. 假定随机过程 $\{Z_t, t \geq 0\}$ 的状态空间为 $\Omega = \{0, 1, 2, \dots\}$, $\tau_0 < \tau_1 < \tau_2 < \dots$ 是该过程的状态转移时刻,其中 $\tau_0 = 0$, X_n 是时刻 τ_n 发生后过程转移到的状态,若对所有整数 $n \in \Omega$ 和实数 $t \geq 0$ 有

$$p(X_{n+1}, \tau_{n+1} - \tau_n \leq t | X_0, \dots, X_n, \tau_0, \dots, \tau_n) = p(X_{n+1}, \tau_{n+1} - \tau_n \leq t | X_n) = p(X_1, \tau_1 \leq t | X_0) \quad (1)$$

则称过程 $\{(X_n, \tau_n), n \geq 0\}$ 是一个马尔可夫更新过程.

定义 2. 对于随机过程 $\{Z_t, t \geq 0\}$,若存在马尔可夫更新过程 $\{(X_n, \tau_n), n \geq 0\}$ 满足下式

$$p\{Z_{\tau_n+t} = j | Z_u, 0 \leq u \leq \tau_n, X_n = i\} = p\{Z_t = j | X_0 = i\} \quad (2)$$

则称随机过程 $\{Z_t, t \geq 0\}$ 为马尔可夫再生过程.

定义 3. 随机 Petri 网中,仅允许变迁的实施时间服从指数分布,由于指数分布的无记忆特性,模型可以转化为具有马尔可夫特性的随机过程进行分析,通常称这种可以转化为马尔可夫过程的随机 Petri 网为马尔可夫随机 Petri 网(Markovian Stochastic Petri Nets, MSPNs).

定义 4. 若随机 Petri 网中,允许变迁的实施时间为连续的非指数分布,则称该随机 Petri 网为非马尔可夫随机 Petri 网(Non-Markovian Stochastic Petri Nets, NMSPNs).

马尔可夫再生理论(Markov regenerative theory)的思想是将非马尔可夫问题转化为马尔可夫问题进行分析求解. 可将马尔可夫再生理论的思想用于 NMSPNs 模型中. 由定义 4 可知, NMSPNs 模型允许瞬时变迁、指数变迁以及一般变迁的存在. 如果模型中至少有一个瞬时变迁可实施,即系统处于消亡状态,在分析 NMSPNs 模型之前需要将消亡状态移去^[9]. 对于化简后的模型,可以利用马尔可夫再生理论进行分析. 首先选取模型中状态改变的时点为再生点(regenerative points),在再生点上系统的行为具有无记忆特性,从而形成嵌入马尔可夫链(Embedded Markov Chain, EMC). 然后用传统的分析嵌入马尔可夫链的方法,分析系统的整体性能. 下面给出这种分析方法的形式化描述.

对于一个系统的 NMSPNs 模型,对应的随机过程记为 $Z = (Z_t, t \geq 0)$, 其中 Z_t 表示系统在 t 时刻所处的状态,状态空间为可数集 Ω . 在随机过程 Z 的时点上采样,得到的再生状态序列为可数集合 $\Omega' = \{X_n, n \geq 0\}$ ($\Omega' \subseteq \Omega$), 对应的再生点是 τ_n ($\tau_0 < \tau_1 < \tau_2 < \dots$), 随机过程 $\{X_n; \tau_n \geq 0\}$ 构成嵌入马尔可夫链. 根据定义 1, 随机过程 $\{X_n; \tau_n \geq 0\}$ 是一个马尔可夫更新过程,其状态转移概率为

$$K_{ij}(t) = p(X_{n+1} = j, \tau_{n+1} - \tau_n \leq t | X_n = i) = p(X_1 = j, \tau_1 \leq t | X_0 = i), \quad \forall i, j \in \Omega', t \geq 0 \quad (3)$$

根据定义 2 可知,随机过程 $\{Z_t, t \geq 0\}$ 是马尔可夫再生过程,其状态转移概率为

$$\begin{aligned} V_{ij}(t) &= p(Z_t = j | Z_0 = i) = \\ &= p(Z_t = j, \tau_1 > t | Z_0 = i) + p(Z_t = j, \tau_1 \leq t | Z_0 = i) = \\ &= E_{ij}(t) + \sum_{k \in \Omega'} \int_0^t dK_{ik}(u) \cdot V_{kj}(t-u) \end{aligned} \tag{4}$$

其中

$$E_{ij}(t) = p(Z_1 = j, \tau_1 > t | Z_0 = i), \forall i \in \Omega', j \in \Omega, t \geq 0 \tag{5}$$

这里, $E_{ij}(t)$ 表示相邻两再生点之间系统的随机过程的行为, $K_{ij}(t)$ 表示再生点上系统的随机行为。

根据求出的 NMSPNs 的状态转移概率,可求得系统稳态概率解为

$$\pi_j = \frac{\sum_{k \in \Omega} D_k \alpha_{kj}}{\sum_{k \in \Omega} D_k \alpha_k} \tag{6}$$

$$\alpha_{ij} = \int_0^\infty E_{ij}(t) dt \tag{7}$$

$$\Phi_{ij} = \lim_{t \rightarrow \infty} K_{ij}(t) \tag{8}$$

其中 $D = D\Phi$, $\sum D_i = 1 (\forall i \in \Omega', j \in \Omega, t \geq 0)$, 基于系统的稳态概率解,可以对系统性能进行分析。

3 软件再生建模与分析

3.1 非马尔可夫随机 Petri 网的软件再生模型

软件老化引起的系统失效是一种突发性的事件,它造成的损失往往是巨大的。在软件运行过程中定时地对系统执行再生,即主动地停止软件运行,并清理运行环境,然后重启应用软件,可避免意外宕机。软件再生系统包括三个状态:工作状态、再生状态和失效状态。在工作状态时,系统可以处理外部资源请求,此时系统可用;再生状态时,系统执行再生,此时系统不可用;失效状态时,系统发生故障进行恢复,此时系统也不可用。系统采用再生技术后的非马尔可夫随机 Petri 网模型如图 1 所示。

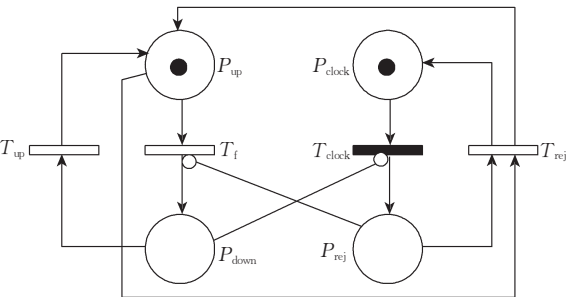


图 1 采用再生技术的系统非马尔可夫随机 Petri 网模型

位置 P_{up} 表示系统处于正常工作状态;位置 P_{down} 表示系统发生故障,处于失效状态;位置 P_{rej} 表示系统准备再生的状态。初始时刻,系统处于 P_{up} 状态。当变迁 T_f 实施后,系统进入 P_{down} 状态,然后变迁 T_{up} 实施,系统进行恢复后回到初始状态 P_{up} 。确定变迁 T_{clock} 表示再生周期,变迁 T_{clock} 实施后,系统进入 P_{rej} 状态,然后触发变迁 T_{rej} ,系统进行软件再生,再生结束后回到初始状态 P_{up} 。变迁 T_{rej} 和 T_{up} 的实施速率为 λ_r 和 λ_a ,实施时间服从指数分布,变迁 T_f 的实施速率为 λ_f ,实施时间服从一般分布,确定变迁 T_{clock} 实施速率为常量 δ 。从位置 P_{down} 到变迁 T_{clock} 的禁止弧表示系统失效时禁止变迁 T_{clock} 实施,从位置 P_{rej} 到变迁 T_f 的禁止弧表示系统再生时,系统不会失效。

求解图 1 中系统的稳态解,可以先将系统的非马尔可夫随机 Petri 网转换为系统状态可达图,如图 2 所示。

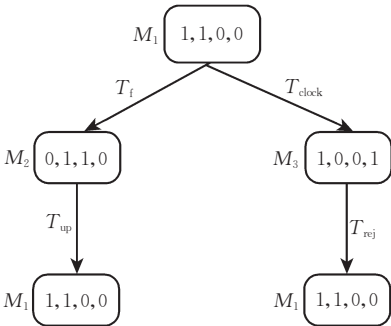


图 2 系统状态可达图

系统状态可达图中给出再生系统的标识过程,标识为 $M_1 = (1, 1, 0, 0)$, $M_2 = (0, 1, 1, 0)$, $M_3 = (1, 0, 0, 1)$, 其中四元组标识中的元素依次表示 P_{up} , P_{clock} , P_{down} 和 P_{rej} 中的 Token 数。标识过程 $M = (M_t; t \geq 0)$ 可表示任意时刻 t 系统的状态, M_t 的取值与相应标识的下标一致,其含义为

$$M_t = \begin{cases} 1, & \text{系统处于正常工作状态,再生时间还未到} \\ 2, & \text{系统处于失效状态,再生时间还未到} \\ 3, & \text{系统准备再生} \end{cases}$$

由于标识过程中不含有可实施的瞬时变迁,则标识过程中选取再生状态 1,2,3,对应的嵌入马尔可夫链由图 3 所示。

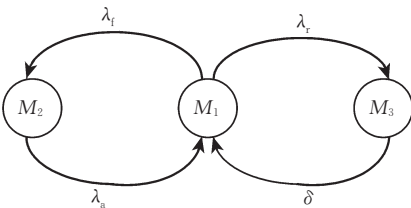


图 3 系统嵌入马尔可夫链

3.2 软件再生模型求解

由上面得到的系统的嵌入马尔可夫链,可得软件再生系统的嵌入马尔可夫链的状态转移矩阵,即相邻两再生点上的状态转移矩阵 $\mathbf{K}(t)$ 有下列形式:

$$\mathbf{K}(t)=\begin{bmatrix} 0 & K_{12}(t) & K_{13}(t) \\ K_{21}(t) & 0 & 0 \\ K_{31}(t) & 0 & 0 \end{bmatrix} \quad (9)$$

由式(3),计算矩阵 $\mathbf{K}(t)$ 中的元素值如下:

$$K_{12}(t)=\begin{cases} G_f(t), & 0 \leq t < \delta \\ G_f(\delta), & t \geq \delta \end{cases},$$
$$K_{13}(t)=\begin{cases} 0, & 0 \leq t < \delta \\ 1-G_f(\delta), & t \geq \delta \end{cases},$$
$$K_{21}(t)=1-e^{-\lambda_a t}, K_{31}(t)=1-e^{-\lambda_r t} \quad (10)$$

其中, $G_f(t)$ 可以是软件再生系统的故障时间 T_f 服从的任意故障概率分布函数, δ 是系统的软件再生周期. 对于相邻两再生点之间的状态转移矩阵 $\mathbf{E}(t)$, 有

$$\mathbf{E}(t)=\begin{bmatrix} E_{11}(t) & 0 & 0 \\ 0 & E_{22}(t) & 0 \\ 0 & 0 & E_{33}(t) \end{bmatrix} \quad (11)$$

由式(5),计算矩阵 $\mathbf{E}(t)$ 中的元素值如下:

$$E_{11}(t)=\begin{cases} 1-G_f(t), & 0 \leq t < \delta \\ 0, & t \geq \delta \end{cases}, E_{22}(t)=e^{-\lambda_a t},$$
$$E_{33}(t)=1-K_{31}(t)-K_{32}(t)=e^{-\lambda_r t} \quad (12)$$

根据系统的瞬态解 $\mathbf{E}(t)$ 和 $\mathbf{K}(t)$ 及式(6),可以求得系统的稳态解 π 为

$$\pi=\frac{[\alpha_{11}, \alpha_{22} G_f(\delta), \alpha_{33}(1-G_f(\delta))]}{\alpha_{11}+\alpha_{22} G_f(\delta)+\alpha_{33}(1-G_f(\delta))} \quad (13)$$

其中

$$\alpha=E(\infty)=\begin{bmatrix} \int_0^\delta (1-G_f(t))dt & 0 & 0 \\ 0 & \frac{1}{\lambda_a} & 0 \\ 0 & 0 & \frac{1}{\lambda_r} \end{bmatrix},$$
$$\Phi=K(\infty)=\begin{bmatrix} 0 & G_f(\delta) & 1-G_f(\delta) \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix},$$
$$D=\left[\frac{1}{2}, \frac{G_f(\delta)}{2}, \frac{(1-G_f(\delta))}{2}\right] \quad (14)$$

4 仿真实验结果和系统性能分析

由上面给出的模型可以看出,当 $M_i=1$,系统处

于可用状态,再由系统的稳态解可得,系统的可用性 P_A 为

$$P_A=\frac{\pi_1}{\pi_1+\pi_2+\pi_3} \quad (15)$$

若单位时间内系统恢复时的宕机成本为 C_f ,系统再生时的宕机成本为 C_r ,则系统单位时间内的平均总宕机成本 C 为

$$C=\pi_2 C_f+\pi_3 C_r \quad (16)$$

模型中对应的变迁实施参数如表 1 所示(单位:小时⁻¹).

表 1 变迁参数表

变迁	变迁参数服从的分布	变迁参数
T_f	一般	λ_f
T_{clock}	确定	δ
T_{rej}	指数	λ_r
T_{up}	指数	λ_a

仿真实验中,假定系统再生的变迁速率 $\lambda_r=1$ 小时⁻¹,系统恢复的变迁速率 $\lambda_a=0.1, 0.2, 0.3$ 小时⁻¹三种情况. 本文提出的再生系统的故障时间 T_f 服从任意一般的概率分布函数 $G_f(t)$, 实验中选取概率分布函数为 Weibull 分布和超指数分布为例,分别给出故障时间服从这两种故障分布函数时,再生时间间隔和系统可用性之间的关系图,如图 4(a)和图 4(b)所示. 其中,当系统故障时间服从 Weibull 分布时,有

$$G_f(t)=1-e^{-c \cdot t^\beta} \quad (17)$$

其中 $c=-7, \beta=2$.

当系统故障时间服从超指数分布时,有

$$G_f(t)=1-\frac{\lambda_2}{\lambda_2-\lambda_1} e^{-\lambda_1 t}+\frac{\lambda_1}{\lambda_2-\lambda_1} e^{-\lambda_2 t} \quad (18)$$

其中 $\lambda_1=0.0004, \lambda_2=0.0006$.

从图 4 可以看出,对于系统故障时间服从 Weibull 分布和超指数分布的两种情况,随着再生时间间隔 δ 的增加,系统可用性先增后减,最后趋于一个稳定值. 当 δ 较小,即再生频率较大时,系统处于不可用的状态,随着 δ 逐渐增加,系统可用性达到最大值,此时的 δ 为最优值,当 δ 继续增加时,系统可用性下降. 总体上看,当 λ_r 已固定, λ_a 增加,即 λ_a/λ_r 增大,再生时间间隔 δ 的最优值增大,再生频率减小.

图 5(a)和图 5(b)分别给出在故障率服从 Weibull 分布和超指数分布的条件下,再生时间间隔和系统的总的平均宕机成本之间的关系图. 假定系统恢复时宕机成本 $C_f=5$,系统再生时宕机成本

$C_r=1$. 从图中可以看出,两种情况下,当 δ 较小时,即再生频率较大时,系统处于不可用的状态,总的宕机成本相当大,随着 δ 逐渐增加,总的宕机成本达到最小值,此时的 δ 为最优值,当 δ 继续增加,再生频

率下降,系统失效影响增大,使得总的宕机成本 C 增加. 总体上看,当 λ_r 已固定, λ_a 增加,即 λ_a/λ_r 增大,再生时间间隔 δ 的最优值减小,再生频率增大.

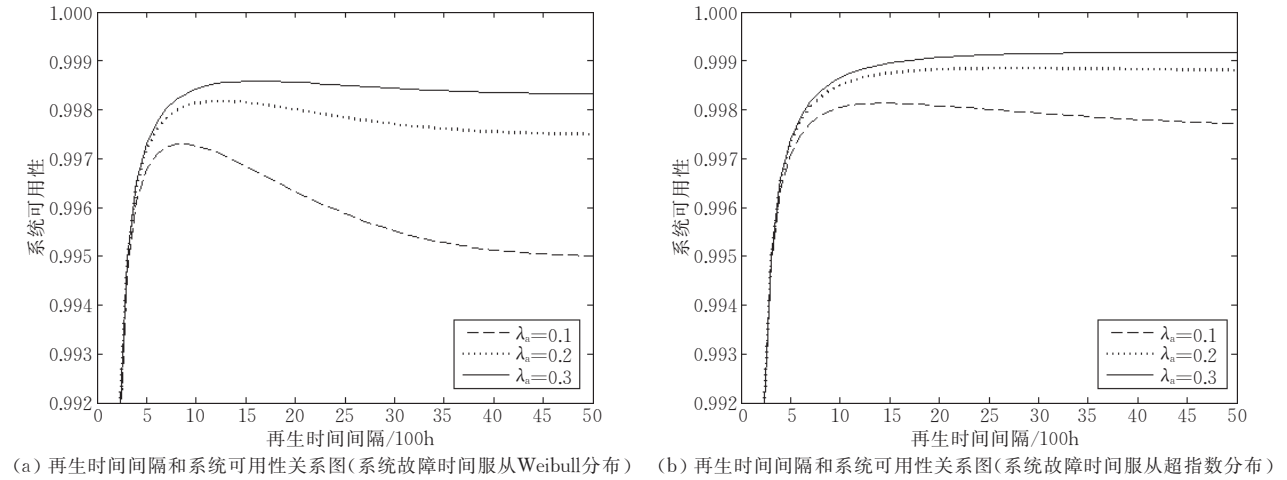


图 4

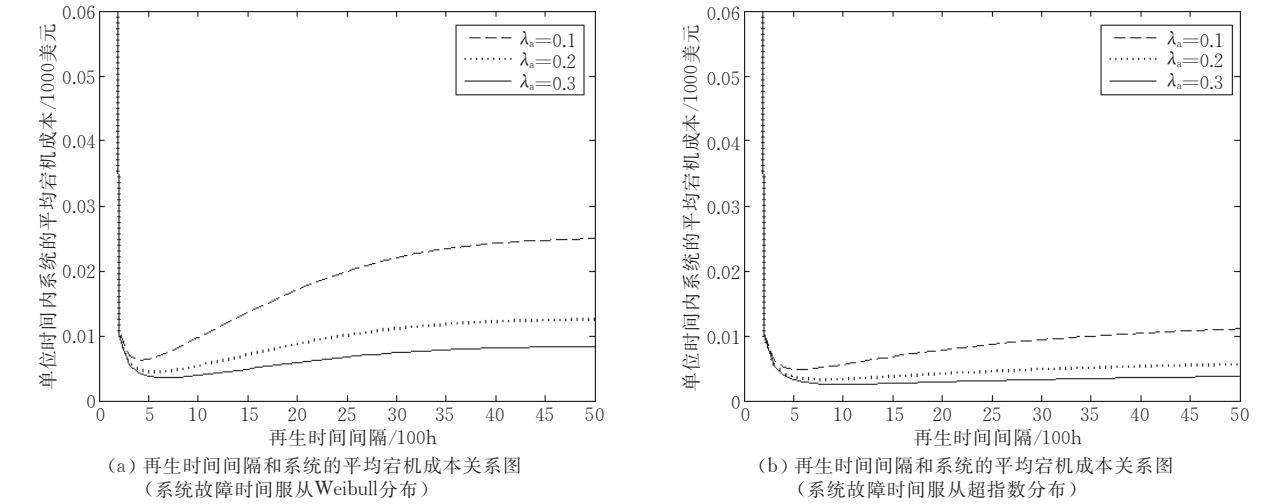


图 5

5 结 论

软件老化是影响软件系统可靠性的重要潜在因素,软件再生技术作为一种预反应式的软件容错机制,是一种成本可控的有计划的预维护手段,在解决老化问题中已日益显现出其重要价值. 以往研究的随机 Petri 网软件再生模型中,假设所有变迁的实施时间服从指数分布. 针对变迁的实施时间呈确定性分布或一般性分布的情况,本文提出了一种基于非马尔可夫随机 Petri 网建立软件再生模型的方法,方法中采用马尔可夫再生理论对模型进行分析求解,并给出模型的瞬态解和稳态解. 由仿真实验结

果可知:选择合适的软件再生周期,可以有效地降低存在老化的软件系统的平均宕机成本,提高系统的可用性和可靠性.

参 考 文 献

[1] Huang Y, Kintala C, Kolettis N, Fulton N. Software rejuvenation: Analysis, module and applications//Proceedings of the IEEE International Symposium on Fault Tolerant Computing. Canada, 1995: 381-390

[2] Avritzer A, Weyuke J. Monitoring smoothly degrading systems for increased dependability. Empirical Software Engineering, 1997, 2(1): 59-77

[3] Dohi T, Goseva-Popstojanova K, Trivedi K S. Statistical non-parametric algorithms to estimate the optimal software

rejuvenation schedule//Proceedings of the Pacific Rim International Symposium on Dependable Computing (PRDC 2000). Los Angeles, USA, 2000: 77-84

[4] Pfening A, Garg S, Puliafito A, Telek M, Trivedi K S. Optimal software rejuvenation for toleration software failures//Proceedings of the 18th International Symposium on Performance Evluation. Lausanne, Switzerland, 1996: 491-506

[5] Xie Wei, Hong Yi-Guang, Trivedi Kishor. Analysis of a two-level software rejuvenation policy. Reliability Engineering and System Safety, 2005, 87(1): 13-22

[6] Castelli V et al. Proactive management of software aging. IBM Journal of Research & Development, 2001, 45(2): 311-332

[7] Garg S, Huang Y, Kintala C, Trivedi K S. Time and load based software rejuvenation: Policy, evaluation and optimality//Proceedings of the 1st Fault Tolerance Symposium, FTS-95. Madras, India, 1995: 22-25

[8] Yuan Chong-Yi. The Principle of Petri Net. Beijing: Peking University Press, 1999(in Chinese)
(袁崇义. Petri 网原理. 北京:北京大学出版社, 1999)

[9] Lin Chuang. Stochastic Petri Net and Systematic Performance Evaluation. Beijing: Tsinghua University Press, 2000 (in Chinese)
(林闯. 随机 Petri 网和系统性能评价. 北京:清华大学出版社, 2000)

[10] Wang Rong-Xin. Stochastic Process. Xi'an: Xi'an Jiaotong University Press, 1987(in Chinese)
(汪荣鑫. 随机过程. 西安:西安交通大学出版社, 1987)



MENG Hai-Ning, born in 1979, Ph. D. candidate. Her research interests include distributed system, software reliability.

QI Yong, professor, Ph. D. supervisor. His research interests include distributed system, Internet Technology.

HOU Di, associate professor. His research interests focus on database.

Background

Recent studies have reported the phenomenon of software aging in which the state of system performance degrades with time. In order to enhance software reliability and prevent system degradation or crash, such a preventive maintenance technique called software rejuvenation was introduced which provides the theory basis and the method for fault-re-

cover and self-healing.

Under the National "863" Foundation Project (2001AA113040), the National Natural Foundation of China (60473098) and IBM Joint Research, we developed the technical achievement and conduct the research on software rejuvenation.