

重配置与周期再生相结合的软件再生模型

杜小智¹, 齐勇¹, 侯迪¹, 陈滢²

(1. 西安交通大学电子与信息工程学院, 710049, 西安; 2. IBM 中国研究院, 100094, 北京)

摘要: 针对持续运行软件系统中的老化问题, 提出一种重配置与周期再生相结合的软件再生方法. 采用连续时间马尔可夫链对两阶段衰退软件系统的工作模式进行建模, 给出系统可用性和单位时间平均成本的形式化定义与分析, 并以可用性和成本作为评价标准计算相应的最佳再生周期, 推导出了采用重配置的临界条件. 仿真结果表明, 随着重配置成功率的提高, 系统的可用性增加, 成本降低, 当重配置成功率为 1 时, 与纯周期再生方法相比, 采用结合重配置的软件再生方法可使系统单位时间宕机成本降低 48.9%.

关键词: 软件再生; 重配置; 软件老化

中图分类号: TP393 **文献标志码:** A **文章编号:** 0253-987X(2010)01-0091-05

Software Rejuvenation Model Based on Reconfiguration and Periodical Rejuvenation

DU Xiaozhi¹, QI Yong¹, HOU Di¹, CHEN Ying²

(1. School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China;

2. IBM China Research Lab, Beijing 100094, China)

Abstract: To counteract the phenomenon of software aging and system failures, a software rejuvenation model based on reconfiguration is proposed. The continuous time Markov chain is adopted to describe the rejuvenation model for software systems with two degradation states. The formal definition and analysis of the system availability and of the expected total cost per unit time are given. The optimal rejuvenation interval is obtained by either maximizing the availability or minimizing the expected cost. A reconfiguration threshold is derived by comparing the rejuvenation method based on reconfiguration with the periodical rejuvenation method. Simulation results show that the system availability increases and the downtime cost decreases as the success probability of the reconfiguration increases. A comparison with the periodical rejuvenation method shows that when the success probability of the reconfiguration is 1, the system downtime cost is reduced by 48.9%.

Keywords: software rejuvenation; reconfiguration; software aging

近年来对于软件可靠性的研究发现, 很多大型软件系统在持续平稳运行一段时间之后, 往往会出现系统资源大量消耗、服务性能和质量下降, 甚至出现挂起或停机, 这种现象称为“软件老化”^[1]. 软件老化是由于软件系统随运行时间的增加, 使得系统资

源逐渐消耗或运行时错误逐渐积累导致的, 例如在操作系统^[2]、Java 虚拟机^[3]、Web 服务器^[4]、SOA 服务器^[5]、OLTP 服务器^[6]、电信计费软件^[1]、应用服务器^[7]、通信交换软件、Netscape 等系统中, 都普遍发现了软件老化现象. 为了消除或者减缓系统软件

老化现象,有学者提出了一种称为软件再生(software rejuvenation)^[1]的“前摄式”(proactive)容错技术,通过排除与软件老化相关的不良因素,使系统恢复到健康状态。

周期再生是一种常用的软件再生方法,其主要思想是周期性地暂停软件的运行,清除持续运行系统的内部状态,重新启动并恢复到干净的初始或中间状态,从而达到预先防止将来可能发生的更严重故障的目的^[1,8-9]。然而,有时当系统发生性能衰退时,通过对系统资源进行重新配置,使资源使用更加合理,就可以恢复其性能^[10],而不需要停止系统的运行进行再生操作。本文结合重配置思想,采用连续时间马尔可夫链描述基于重配置的软件再生模型,并进行理论分析和仿真实验。

1 软件系统行为模型

针对两阶段衰退软件系统,结合重配置的软件再生模型如图1所示。一开始软件系统运行于鲁棒态 S_0 ,在该状态下,系统具有很高的服务能力且不会发生故障;随着系统的持续运行,软件系统进入衰退态 S_p ,在该状态下,系统出现性能衰退,服务能力较差,且可能发生失效引起系统宕机;如果不进行重配置或再生操作,系统继续运行就会发生故障进入失效态 S_f ,此时系统处于宕机状态,不能提供服务;在对系统故障进行修复后,系统又恢复到初始的鲁棒态 S_0 。

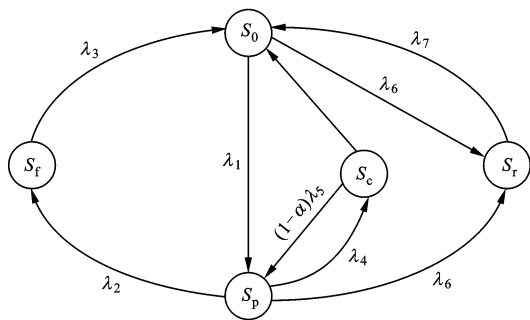


图1 结合重配置的软件再生模型

由于系统故障发生的随机性,系统从衰退态变为失效态将引起非预期的系统停机,往往造成较大的损失。为了减少非预期宕机带来的损失,可以在系统运行一段时间后,主动进行软件再生操作,使系统从鲁棒态或衰退态进入再生态 S_r 。在该状态下,系统处于停机状态,但是由于是有计划的停机,所以造成的损失较少且恢复时间较短。当再生操作完成后,系

统恢复到鲁棒态。

当发现系统出现性能衰退时,有时候进行重配置操作(如将CPU、内存、网络等系统资源在各个任务间进行重新分配),也能使系统恢复到高服务能力状态,而不需要进行再生操作。当系统在衰退态运行一段时间后进行重配置操作,则系统进入重配置态 S_c ,在该状态下系统不可用,但进行重配置的时间比进行再生操作的时间要短。当重配置操作完成后,如果重配置成功,即系统的性能得到恢复,则系统返回到鲁棒态,如果重配置失败,则系统继续运行于衰退态。 α 为重配置成功率。

2 系统性能分析

设系统各状态之间的转移时间服从指数分布,其参数如图1所示,则结合重配置的软件再生模型为连续时间马尔可夫过程,系统处于各个状态的概率满足如下方程组

$$\left. \begin{aligned} P_0 + P_p + P_f + P_r + P_c &= 1 \\ P_p(\lambda_2 + \lambda_4 + \lambda_6) &= P_0\lambda_1 + P_c(1-\alpha)\lambda_5 \\ P_f\lambda_3 &= P_p\lambda_2 \\ P_r\lambda_7 &= P_p\lambda_6 + P_0\lambda_6 \\ P_c\lambda_5 &= P_p\lambda_4 \end{aligned} \right\} \quad (1)$$

式中: P_0 、 P_p 、 P_f 、 P_r 、 P_c 分别为系统处于鲁棒态、衰退态、失效态、再生态和重配置态的概率。

对式(1)求解得

$$\left. \begin{aligned} P_p &= \left[1 + \frac{\lambda_2}{\lambda_3} + \frac{\lambda_4}{\lambda_5} + \frac{\lambda_2 + \lambda_6 + \alpha\lambda_4}{\lambda_1} + \frac{(\lambda_1 + \lambda_2 + \lambda_6 + \alpha\lambda_4)\lambda_6}{\lambda_1\lambda_7} \right]^{-1} \\ P_0 &= \frac{\lambda_2 + \lambda_6 + \alpha\lambda_4}{\lambda_1} P_p \\ P_f &= \frac{\lambda_2}{\lambda_3} P_p \\ P_r &= \frac{(\lambda_1 + \lambda_2 + \lambda_6 + \alpha\lambda_4)\lambda_6}{\lambda_1\lambda_7} P_p \\ P_c &= \frac{\lambda_4}{\lambda_5} P_p \end{aligned} \right\} \quad (2)$$

2.1 可用性分析

从上面的分析可知,当软件系统处于鲁棒态和衰退态时,系统处于运行状态,能够提供服务,而当系统处于失效态、再生态和重配置态时,系统处于停机状态,不能够提供服务。因此,系统的可用性

$$P_{\text{avail}} = P_0 + P_p \quad (3)$$

对可用性 P_{avail} 进行求导,并令 $P'_{\text{avail}}=0$,可得最佳再生频率

$$\lambda_6^* = \left[\left(\frac{\lambda_2}{\lambda_3} + \frac{\lambda_4}{\lambda_5} \right) \lambda_1 \lambda_7 \right]^{1/2} - (\lambda_1 + \lambda_2 + \alpha \lambda_4) \quad (4)$$

此时系统具有最大的可用性

$$P_{\text{avail}}^* = \lambda_7 / \left\{ 2 \left[\left(\frac{\lambda_2}{\lambda_3} + \frac{\lambda_4}{\lambda_5} \right) \lambda_1 \lambda_7 \right]^{1/2} + \lambda_7 - \lambda_1 - \lambda_2 - \alpha \lambda_4 \right\} \quad (5)$$

从式(5)可以看出,重配置成功率 α 越大,系统的可用性越高。

为了判断在软件系统运行过程中采用重配置方法是否合理有效,需要将其与纯周期再生方法进行比较。令图 1 所示系统再生模型中 $\lambda_4 = 0$, 即重配置周期为 ∞ , 此时系统模型变为纯周期再生模型, 其系统最大可用性

$$P_{\text{avail_std}}^* = \lambda_7 / \left[2 \left(\frac{\lambda_1 \lambda_2 \lambda_7}{\lambda_3} \right)^{1/2} + \lambda_7 - \lambda_1 - \lambda_2 \right] \quad (6)$$

当 $P_{\text{avail}}^* \geq P_{\text{avail_std}}^*$ 时, 采用重配置方法更有效, 此时

$$\lambda_4 \geq \frac{4}{\alpha^2} \left[\frac{\lambda_1 \lambda_7}{\lambda_5} - \alpha \left(\frac{\lambda_1 \lambda_2 \lambda_7}{\lambda_3} \right)^{1/2} \right] \quad (7)$$

只要重配置频率 λ_4 不小于式(7)的右半部分, 则在系统运行过程中采用重配置操作就可以提高系统的可用性。由式(7)可以看出, α 越大, 则满足条件的 λ_4 越小, 即当重配置成功率足够高时, 可适当减小重配置频率, 增大重配置周期。

2.2 成本分析

当软件系统处于失效态、再生态和重配置态时, 由于系统宕机不可用, 将会导致任务丢失, 造成损失。设系统处于失效态、再生态和重配置态时的单位时间宕机成本分别为 c_f 、 c_r 和 c_c 。由于系统失效引起的宕机是非预期的, 而由再生和重配置引起的系统停机是主动的, 通常非预期的系统宕机比有计划的停机所造成的损失更大, 因此一般取 $c_f \gg c_r$ 和 $c_f \gg c_c$, 而取 c_r 与 c_c 相等或相近。

系统的单位时间宕机成本

$$C_{\text{cost}} = c_f P_f + c_r P_r + c_c P_c \quad (8)$$

与系统可用性分析相似, 对系统的单位时间宕机成本进行求导, 并令 $(C_{\text{cost}})' = 0$, 可得最佳再生频率

$$\lambda_6^* = \frac{-b + (b^2 - 4ac)^{1/2}}{2a} \quad (9)$$

式中

$$a = \frac{c_r}{\lambda_1^2 \lambda_7}; \quad b = \frac{2}{\lambda_1 \lambda_7} \left(\frac{c_r(\lambda_2 + \alpha \lambda_4)}{\lambda_1} + \frac{c_f \lambda_2}{\lambda_3} + \frac{c_c \lambda_4}{\lambda_5} \right)$$

$$c = \frac{c_r(\lambda_1 + \lambda_2 + \alpha \lambda_4)}{\lambda_1 \lambda_7} \left(1 + \frac{\lambda_2}{\lambda_3} + \frac{\lambda_4}{\lambda_5} + \frac{\lambda_2 + \alpha \lambda_4}{\lambda_1} \right) - \left(\frac{c_f \lambda_2}{\lambda_3} + \frac{c_c \lambda_4}{\lambda_5} \right) \left(\frac{1}{\lambda_1} + \frac{\lambda_1 + \lambda_2 + \lambda_4}{\lambda_1 \lambda_7} \right)$$

将式(9)代入式(8), 可得到系统的最小单位时间宕机成本 C_{cost}^* 。令 $\lambda_4 = 0$, 即重配置周期为 ∞ , 此时系统模型变为纯周期再生模型, 重新计算相应的最佳再生频率和最小单位时间宕机成本 $C_{\text{cost_std}}^*$ 。令 $C_{\text{cost}}^* \geq C_{\text{cost_std}}^*$, 可得到采取重配置操作的频率边界条件, 与系统可用性分析相同, 当重配置成功率足够高时, 可适当减小重配置频率, 增大重配置周期。

3 仿真实验

基于上面的理论分析, 进行仿真实验来检验模型的有效性与合理性。模型的默认参数如表 1 所示。

表 1 默认参数

参数	名称	取值
λ_1/h^{-1}	衰退速率	1/120
λ_2/h^{-1}	失效速率	1/360
λ_3^{-1}/h	失效后修复时间	1
λ_4/h^{-1}	重配置频率	1/20
λ_5^{-1}/h	重配置完成时间	1/6
λ_6/h^{-1}	再生频率	可变
λ_7^{-1}/h	再生完成时间	1/4
α	重配置成功率	可变
c_f	失效单位时间成本	50
c_r	再生单位时间成本	2
c_c	重配置单位时间成本	2

图 2 是 α 分别为 0.2、0.5、0.8 时系统可用性与再生周期的关系图。为与纯周期再生方法进行比较, 图中也给出了采用纯周期再生的系统可用性与再生周期的关系。从图 2 中可以看出: 随着再生周期的增加, 系统的可用性增加; 对应固定的再生周期, 随着重配置成功率 α 的增加, 系统的可用性增加。从图 2 中未发现最佳的再生周期及最大系统可用性, 这表明对于当前的参数, 从可用性角度来说, 不进行再生操作最好。

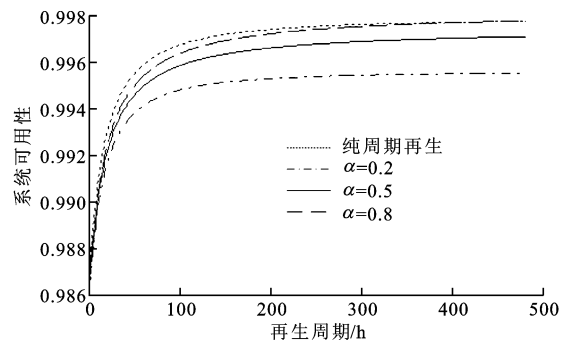


图 2 系统可用性与再生周期的关系

图 3 是 α 分别为 0.2、0.5、0.8 时系统单位时间平均宕机成本与再生周期的关系图, 为与纯周期再生方法进行比较, 图中也给出了采用纯周期再生的系统单位时间平均宕机成本与再生周期的关系.

从图 3 可以看出: 随着再生周期的增加, 软件系统的单位时间宕机成本逐渐减小, 当再生周期达到某个值时, 系统的平均成本达到最小值, 然后随着再生周期的增加, 系统的单位时间宕机成本逐渐增加; 随着重配置成功率的增加, 系统的宕机成本逐渐减小; 采用重配置再生方法时的宕机成本小于纯周期

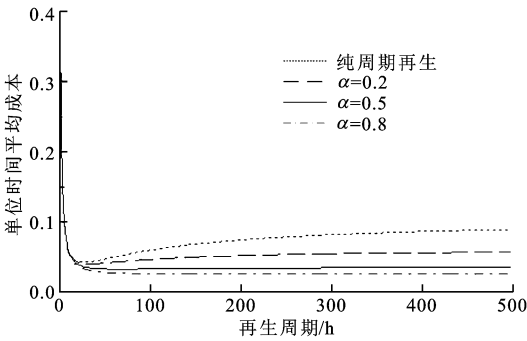


图 3 系统单位时间宕机成本与再生周期的关系

表 2 结合重配置再生方法的系统可用性

α	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
系统可用性	0.992 5	0.994 4	0.995 5	0.996 2	0.996 7	0.997 1	0.997 4	0.997 6	0.997 8	0.997 9	0.998 0
可用性增幅/%	-0.52	-0.33	-0.22	-0.15	-0.10	-0.06	-0.03	-0.01	0.01	0.02	0.03

表 3 结合重配置再生方法的单位时间宕机成本

α	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
平均成本	0.044 8	0.042 4	0.040 0	0.037 6	0.035 1	0.032 7	0.030 2	0.027 8	0.025 4	0.023 3	0.021 5
成本降幅/%	-6.4	-0.7	5.0	10.7	16.6	22.3	28.3	34.0	40.0	44.7	48.9

4 结 论

本文将重配置思想应用到软件老化与再生的研究中, 针对两阶段衰退软件系统, 提出一种重配置与周期再生相结合的软件再生模型, 并从系统可用性和单位时间宕机成本两方面进行了理论分析与实验验证. 结果表明: 当重配置成功率较高时, 基于重配置的再生方法能够比纯周期再生方法得到更高的系统可用性和更低的宕机成本; 当重配置成功率较低时, 尽管基于重配置的再生方法在可用性方面不如纯周期再生方法, 但系统可用性下降很少, 而其单位

再生的宕机成本.

表 2 为 α 变化时系统可用性与纯周期再生的系统可用性的对比关系. 由于系统可用性随再生周期的增加而增加, 因此选择再生周期为 500 h 时的可用性作为比较, 此时纯周期再生方法的系统可用性为 0.997 7. 从表 2 中可以看出, 当 $\alpha \geq 0.8$ 时, 基于重配置的方法优于纯周期再生方法, 即使重配置成功率很低, 与纯周期再生方法相比, 系统的可用性也仅仅有轻微的下降.

表 3 为 α 变化时系统单位时间宕机成本与纯周期再生宕机成本的对比关系. 取最佳再生周期情况下的宕机成本作为比较, 此时纯周期再生方法的宕机成本为 0.042 1. 从表 3 中可以看出: 当 $\alpha \geq 0.2$ 时, 基于重配置的方法优于纯周期再生方法, 即使 α 处于较低的水平, 基于重配置的方法依然合理有效; 随着 α 的增大, 系统平均宕机成本降低; 当 α 较大时, 系统的宕机成本降幅增大; 即使当 α 为 0, 即系统每次重配置都失败时, 系统的宕机成本也增加很少.

时间宕机成本则低于纯周期再生方法.

参考文献:

[1] HUANG Y, KINTALA C, KOLETTIS N, et al. Software rejuvenation: analysis, module and applications [C]// Proceedings of the 25th Symposium on Fault Tolerant Computing. Piscataway, NJ, USA: IEEE Press, 1995: 381-390.

[2] GARG S, MOORSEL A V, VAIDYANATHAN K, et al. A methodology for detection and estimation of software aging [C]// Proceedings of the Ninth International Symposium on Software Reliability Engineer-

- ing. Piscataway, NJ, USA: IEEE Press, 1998: 283-292.
- [3] DOMENICO C, SALVATORE O, STEFANO R. Characterizing aging phenomena of the Java virtual machine [C]//Proceedings of 26th IEEE International Symposium on Reliable Distributed Systems. Piscataway, NJ, USA: IEEE Press, 2007: 127-136.
- [4] GROTTKE M, LI L, VAIDYANATHAN K, et al. Analysis of software aging in a Web server [J]. IEEE Trans on Reliability, 2006, 55(3): 411-420.
- [5] SILVA L, MADEIRA H, SILVA J G. Software aging and rejuvenation in a soap-based server [C]//Proceedings of Fifth IEEE International Symposium on Network Computer and Applications. Piscataway, NJ, USA: IEEE Press, 2006: 56-65.
- [6] CASSIDY K J, GROSS K C, MALEKPOUR A. Advanced pattern recognition for detection of complex software aging phenomena in online transaction processing servers [C]//Proceedings of the International Conference on Dependable Systems and Networks. Piscataway, NJ, USA: IEEE Press, 2002: 478-482.
- [7] 徐文彬, 齐勇, 侯迪, 等. 基于时间序列分析的应用服务器性能衰退模型[J]. 西安交通大学学报, 2007, 41(2): 426-429.
- XU Wenbin, QI Yong, HOU Di, et al. Performance degradation model of application server system based on time series analysis [J]. Journal of Xi'an Jiaotong University, 2007, 41(2): 426-429.
- [8] 孟海宁, 齐勇, 侯迪. 基于非马尔可夫随机 Petri 网的软件再生建模与分析[J]. 计算机学报, 2007, 30(12): 2212-2217.
- MENG Haining, QI Yong, HOU Di. Modeling and analysis of software rejuvenation based on non-Markovian stochastic Petri nets [J]. Journal of Computer, 2007, 30(12): 2212-2217.
- [9] KOUTRAS V P, PLATIS A N. Applying software rejuvenation in a two-node cluster system for high availability [C] // Proceedings of the International Conference on Dependability of Computer Systems. Piscataway, NJ, USA: IEEE Press, 2006: 175-182.
- [10] YOUNG C H, KIM B, SHIM E B. Resource reconfiguration scheme based on temporal quorum status estimation for grid management [J]. IEICE Trans Commun, 2005, E88-B(11): 4378-4381.
- (编辑 葛赵青 苗凌)

我校在无铅压电材料研究领域取得重大突破

我校多学科材料研究中心主任、金属材料强度国家重点实验室首席科学家、国家“千人计划”入选者、日本国立材料研究院铁性物理研究室主任任晓兵博士率领的研究小组,近日报道了一种对环境无害的无铅压电材料——锆钛酸钡钙,其压电性能优于全世界使用了长达半个世纪、对人体和环境有害的核心压电材料——锆钛酸铅(PZT)陶瓷。该成果近日将在物理学界的权威学术刊物《Physical Review Letters》上发表,我校多学科材料研究中心、电力设备电气绝缘国家重点实验室博士研究生刘文凤为该文第一作者。

迄今为止,这是无铅压电材料首次在压电性能上超越锆钛酸铅陶瓷。同时,任晓兵研究小组提出了有效提高压电性能的理论,并提供了一套有效方法,对开发高性能无铅压电材料具有重要意义。这项研究成果不仅是无铅压电材料领域的突破,而且可能由此引发长期依赖有害锆钛酸铅的行业向无铅压电材料的变革。

任晓兵教授提出了一种形成大压电效应的新理论。这一理论表明:大压电性能与铅并无必要联系,锆钛酸铅只是满足了该理论的要求的一个体系,所有满足该理论要求的体系都可以产生大的压电性。基于这一理论,任小兵研究小组设计开发了这种新型无铅压电材料锆钛酸钡钙。他们的研究结果显示:该无铅压电材料具有令人惊异的压电性能,压电系数高达 620 pC/N,超过了锆钛酸铅的压电系数(250~590 pC/N)。

(撰稿:新宣 来源:交大新闻网 <http://xjtunews.xjtu.edu.cn>)