ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.21, No.2, February 2010, pp.344−358
doi: 10.3724/SP.J.1001.2010.03784

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一种关键任务系统自律可信性模型与量化分析*

王慧强, 吕宏武+, 赵 倩, 董玺坤, 冯光升

(哈尔滨工程大学 计算机科学与技术学院,黑龙江 哈尔滨 150001)

Model and Quantification of Autonomic Dependability of Mission-Critical Systems

WANG Hui-Qiang, LÜ Hong-Wu+, ZHAO Qian, DONG Xi-Kun, FENG Guang-Sheng

(College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China)

+ Corresponding author: E-mail: lvhongwu@hrbeu.edu.cn

Wang HQ, Lü HW, Zhao Q, Dong XK, Feng GS. Model and quantification of autonomic dependability of mission-critical systems. *Journal of Software*, 2010,21(2):344−358. http://www.jos.org.cn/1000-9825/3784.htm

Abstract: In this paper, the existing intrusion tolerance and self-destruction technology are integrated into autonomic computing in order to construct an autonomic dependability model based on SM-PEPA (semi-Markov performance evaluation process algebra) which is capable of formal analysis and verification. It can hierarchically anticipate Threats to dependability (TtD) at different levels in a self-management manner to satisfy the special requirements for dependability of mission-critical systems. Based on this model, a quantification approach is proposed on the view of steady-state probability to evaluate autonomic dependability. Finally, this paper analyzes the impacts of parameters of the model on autonomic dependability in a case study, and the experimental results demonstrate that improving the detection rate of TtD as well as the successful rate of self-healing will greatly increase the autonomic dependability.

Key words: dependability; autonomic computing; PEPA (performance evaluation process algebra); self-tolerance; self-healing; self-destruction

摘 要: 将现有入侵容忍、自毁技术与自律计算相结合,提出了一种基于 SM-PEPA(semi-Markov performance evaluation process algebra)的关键任务系统自律可信性模型以支持形式化分析和推理.该模型具有一定程度的自管理能力,采用分级处理的方式应对各种程度的可信性威胁,满足了关键任务系统对可信性的特殊需求.在此基础上,从稳态概率角度提出了一种自律可信性度量方法.最后,结合具体实例对模型参数对自律可信性的影响进行了初步分析.实验结果表明,增大关键任务系统可信性威胁检测率和自恢复成功率,可在较大范围内提高系统的自律可信特性.

关键词: 可信性;自律计算;PEPA(performance evaluation process algebra);自容忍;自恢复;自毁

中图法分类号: TP311 **文献标识码**: A

当前,异构、复杂的分布式系统每天面临数以百万计的入侵攻击、系统随机故障、人为操作失误等可信性

* Supported by the National Natural Science Foundation of China under Grant Nos.90718003, 60373000, 60973027 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2007AA01Z401 (国家高技术研究发展计划(863))
Received 2009-06-15; Revised 2009-09-11; Accepted 2009-12-07

威胁(threats to dependability,简称TtD),导致软件失效、偏离使命、中断运行,甚至崩溃死机等现象层出不穷,可信性问题空前严峻.而某些应用于金融、交通、军队、核电、政府机关等国民经济或国防领域的分布式系统(通常也称为关键任务系统)面临的可信性威胁尤为严重.它们除了面临常见的可信性威胁之外,还会面临由此造成的失泄密或生命财产危害,这使得可信性成为其最为重要的属性之一.然而,由于规模的扩大和复杂性的提高,系统的配置、管理、维护变得越发困难,复杂性本身已经成为一项巨大的挑战,从而导致传统的使用复杂结构来实现可信的被动式方式不再满足当前的需求.受人体自律神经系统的启发,2001 年,IBM提出了自律计算技术(autonomic computing)[1],具有自感知、上下文感知、自配置、自恢复、自优化和自保护等属性和“以技术管理技术”的特征,使得在隐藏系统管理复杂性的同时实现可信性成为可能.一方面,自律计算的各属性蕴含可信特性,例如,利用自保护属性阻止恶意攻击和病毒入侵,或者通过自配置和自恢复属性使系统从故障中恢复,从而保持和提高系统的可信性[2];另一方面,它又克服了传统可信性实现复杂的缺点,通过“以技术管理技术”使软件系统具有自管理的能力,在向用户提供服务的同时屏蔽底层的复杂性,被认为是一种新的解决可信问题的有效途径.

目前,对于自律可信性模型的研究还处于起步阶段,主要采用自然语言[3]或框图[4,5]等非形式化的描述方法.这些方法虽然比较简单、易用,直观、明了,但缺乏精确的语义,不能满足自律可信性研究的推理验证需求,因此迫切需要建立一种形式化的自律可信性模型.由于从不可信状态恢复的动作延迟时间可能服从一般分布而不是指数分布[6],本文采用高级形式化语言半马尔可夫性能评价进程代数(semi-Markov performance evaluation process algebra,简称SM-PEPA)建立了一种自律可信性模型.该模型采用自律计算的“自管理”思想,综合现有入侵容忍和自毁技术,根据可信性威胁的不同进行分级处理,在减少或免于人为干预的同时实现关键任务系统可信性的保持与增长.此外,本文在建立的SM-PEPA模型基础上,尝试从稳态概率的角度提出一种自律可信性的量化方法,对系统的自律可信性进行分析,以确定影响关键任务系统自律可信性的关键因素,为后续自律可信性的度量提供参考.

本文首先简单介绍模型使用的形式化描述语言 SM-PEPA,然后利用 SM-PEPA 构建关键任务系统的自律可信性模型;接着,通过求解模型对应的半马尔可夫过程(semi-Markov process,简称 SMP)设计一种自律可信性量化方法;最后,结合具体实例简单分析模型参数取值对自律可信性的影响.

1 模型的形式化基础——SM-PEPA 语言

性能评价进程代数(PEPA)是一种典型的进程代数,其基本组成元素包括组件和活动[7].由于PEPA推理简便、易于修改,而且对应于一个隐含的马尔可夫链,除了语义验证功能以外还具有性能分析的能力,因此被广泛应用于各种形式化建模过程.通过聚类、模型分解等手段,PEPA能够更好地处理空间爆炸问题,对多攻击、多故障的大型分布式关键任务系统而言,可以更为快速和精确地实现形式化建模与量化分析.

但是,PEPA的动作延迟时间严格遵循指数分布,在实际的建模过程中具有一定的限制性,许多研究者都尝试对其改进以适应包含非指数分布的情形.Bradley[8]在完整地继承纯PEPA原有语义规则的基础上,对PEPA语言进行一定的扩充以支持动作延迟时间服从一般分布的情况,建立了SM-PEPA.

SM-PEPA的语法如下[8]:

$$P ::= (a^{[n]}, D).P \mid P + P \mid P \underset{L}{\triangleright\!\triangleleft} P \mid P/L \mid A \tag{1}$$

$$D ::= \lambda \mid S \tag{2}$$

$$S ::= \top \mid \omega.L(s) \tag{3}$$

其中,各表达式的含义如下:

(1) 前缀操作$(a^{[n]},D).P$. 组件$(a^{[n]},D).P$执行活动$(a^{[n]},D)$后将变成组件P.其中:a是活动中动作的类型;n为动作a的优先级;D是延迟时间参数,又称为驻留时间(sojourn time)参数.不同于传统的标准PEPA,延迟参数D既可以代表通常的指数分布延迟参数,又可以代表一般分布延迟参数,仅当D的延迟参数服从一般分布时n有效.λ是标准PEPA的分布参数,$\lambda \in R^+ \cup \{n\top \mid n \in Q, n>0\}$,其中,$\top$是被动动作延迟参数,当$D \equiv \lambda$时,SM-PEPA就退化为标

准 PEPA.若动作 a 的延迟时间服从一般分布,则对其概率密度函数进行拉普拉斯变换 $L(s)$,并赋予权值 ω(用于延迟时间服从一般分布动作的合作操作).

(2) 选择操作 P_1+P_2. P_1+P_2 表示系统选择性地执行 P_1 或 P_2,且只能执行二者之一.

(3) 合作操作 $P \underset{L}{\triangleright\!\triangleleft} Q$. $P \underset{L}{\triangleright\!\triangleleft} Q$ 代表组件 P 与组件 Q 并行执行,并通过动作集合 L 进行同步.对于集合 L 中的动作类型而言,它们独立、并发触发交互的动作.如果一个组件在合作中是被动的,则其活动的延迟参数记为 $\top$,此时,合作活动的时间延迟参数等于另一个组件中动作的时间延迟参数.通常当 $L\equiv\phi$ 时,记合作操作为 $P\|Q$.

(4) 隐藏操作 P/L. L 代表在组件 P 中可被视为内部或私有动作的全体,集合 L 中的动作称为隐藏动作,这为进程抽象时忽略与目标无关的活动提供了一种可能.

(5) 常量定义 A. 这是对具体组件含义进行定义的公式.

SM-PEPA扩展了标准PEPA的语义,除了具有纯PEPA的操作语义以外,还遵循如图 1 所示的特有规则[8,9].

Competitive choice:

$$\frac{P \xrightarrow{(a^{[n]},D)} P'}{P+Q \xrightarrow{(a^{[n]},D)} P'} \quad \text{if } Q \xrightarrow{(b^{[m]},D)} \text{ where } m>n;$$

Cooperation:

$$\frac{P \xrightarrow{(a^{[n]},\omega:L(s))} P'}{P \underset{S}{\triangleright\!\triangleleft} Q \xrightarrow{(a^{[n]},\omega:L(s))} P' \underset{S}{\triangleright\!\triangleleft} Q} \quad \text{if } a\notin S \text{ and } Q \xrightarrow{(b^{[m]},D)} \text{ where } m>n;$$

$$\frac{P \xrightarrow{(a^{[n]},\omega:L(s))} P', Q \xrightarrow{(a^{[n]},\tau)} Q'}{P \underset{S}{\triangleright\!\triangleleft} Q \xrightarrow{(a^{[n]},\omega:L(s))} P' \underset{S}{\triangleright\!\triangleleft} Q'} \quad \text{if } a\in S$$

$$\frac{P \xrightarrow{(a^{[n]},\omega:L(s))} P', Q \xrightarrow{(a^{[n]},\varphi:N(s))} Q'}{P \underset{S}{\triangleright\!\triangleleft} Q \xrightarrow{(a^{[n]},\omega':R(s))} P' \underset{S}{\triangleright\!\triangleleft} Q'} \quad \text{if } a\in S \text{ where } \omega'=f(\omega,\varphi) \text{ and } R(s)=g(\omega,\varphi,L(s),N(s))$$

are user-definable synchronization

Equivalence:

$$P+Q\equiv Q+P$$

$$P \underset{S}{\triangleright\!\triangleleft} Q \equiv Q \underset{S}{\triangleright\!\triangleleft} P$$

Fig.1 Special operational semantics for SM-PEPA

图 1 SM-PEPA 特有操作语义

由于一般分布的存在,使得SM-PEPA动作延迟时间不再遵循简单的叠加,其隐含的数学过程为一个半马尔可夫过程.需要指明的是,延迟时间服从一般分布的动作其优先级通常高于其他动作[8],本文将其设定为 2.例如 $P=(action1,\lambda).P1+(action2^{[2]},1:L(s)).P2$,$action2$ 动作的延迟时间服从一般分布,优先级为 2,因此将执行 $(action2^{[2]},1:L(s)).P2$.

此外,与单纯的半马尔可夫过程相比,利用SM-PEPA语言建模具有诸多优点:一方面,SM-PEPA建模过程简便、易于修改,适用于任务目标、应用环境、资源用户动态更改的分布式系统;另一方面,在表达交互过程时,SM-PEPA直观、简洁,含义清晰.此外,SM-PEPA有自动化的求解工具ipc[9],可以通过把SM-PEPA转化为其下层支撑工具DNAmaca/Hydra支持的语言,实现隐含半马尔可夫过程的自动化分析,适宜于规模化、快速化的模型求解,已应用于多种建模过程.

2 基于 SM-PEPA 的关键任务系统自律可信性模型

2.1 相关概念与属性

自律可信性是指系统在对内外部可信性威胁感知的基础上,主动利用自身具有的可信增强措施保持和增长系统可信性的一种能力.关键任务系统对于可信性的特殊需要决定了它必须能够分级应对可信性威胁,根据事件的威胁程度采取不同的应对措施,屏蔽、容忍、恢复和消除这些不可信事件,甚至在必要时能够毁掉核心数据或软件自身来保障包括私密性(privacy)和防危性(safety)在内的可信属性[10].因此,关键任务系统自律可信性的模型至少包含自容忍(self-tolerance)、自恢复(self-healing)和自毁(selfdestruction)这 3 个属性.它们在一个具

有自管理能力的自省模块控制下,处理威胁程度依次递增的可信性威胁.

定义 1(自省). 自省是指在自律反馈结构[11]控制下,依据可信策略对系统内部状态和环境改变进行感知和分析,进而指导关键任务系统自适应的一个过程.当发现系统遭受可信性威胁时,自省模块依据威胁程度决定调用自容忍、自恢复或自毁机制来保障系统的可信性,并对执行结果进行反馈检测.

自省作为自律计算实现的前提,包括自感知和上下文感知[1]两部分,通过对内、外部环境信息的收集分析,为其余属性的实现提供了基础.自省模块是自省功能实现的实体,由监测器、分析器、计划器和执行器这 4 个部分组成,监测器通过部署在系统各个部分的感知器收集内外部环境信息,经过筛选和规格化处理后提交给分析器;分析器对收集的信息进行初步的约减、聚类或融合,生成简单的可信性事件;针对这些事件,计划器依照存储在知识库中的可信策略,制定一系列可信性保障计划;执行器将调用对应的效应器完成对可信性的保持和增长.以上 4 个部分和可信策略库共同构成了一个自律反馈控制结构,在实现可信性的同时减少了人为干预.

定义 2(自容忍). 自容忍是指当外部入侵或系统故障发生后,根据自省模块的感知结果,在一定限度和一定范围内主动阻止入侵并防止失效蔓延的机制.它主要用于处理非紧急的、不伤及系统核心能力和关键机密、不造成灾难的不可信事件.

定义 3(自恢复). 自恢复是指系统在自省模块指导下,采用微重启、热插拔或悔改等快速恢复技术[12],使系统可信性得以保持或增长的机制.它主要处理不适合自容忍或自容忍失效的情况.

定义 4(自毁). 自毁是指当系统遭受严重入侵或致命故障,核心机密开始外泄,或生命财产安全即将受到极大损害时所采取的销毁系统部分或全部内容以保障可信性的一种机制.它是关键任务系统可信性保障的最后手段.

同时,还可以根据应用环境和任务目标的特异性,向关键任务系统自律可信性模型添加自优化、自保护等其他 self-*扩展属性.它们都将根据事件的可信性威胁程度,在自省模块控制下分级应对关键任务系统面临的可信性威胁,以保障系统的可信性,如图 2 所示.此外,在上述过程中,自省模块也将根据面临的可信性威胁不断学习,对可信策略库进行扩充,以提高检测的命中率和效率.

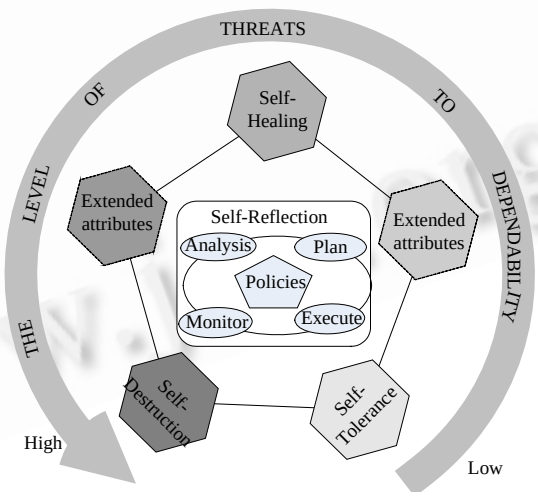


Fig.2 Attributes of autonomic dependability

图 2 自律可信性的属性

2.2 模型的形式化描述

关键任务自律可信性模型的建立有利于从全局角度处理可信性威胁,以最有效的手段和最小开销,及时保障系统的可信性.而通过 SM-PEPA 形式化的描述可以使自律可信性模型具有精确的语义和推理验证的能力,便于模型的检验以及性能分析,为确定影响关键任务系统自律可信性的因素和潜在可信缺陷提供基础.由于自

律可信性表现了关键任务系统应对可信性威胁并保障可信性的一种能力,因此本文主要对关键任务系统遭受可信性威胁时的演化过程进行刻画.换言之,通过可信性威胁与系统的交互来描述自律可信性模型.

可信性威胁的多样性和不确定性导致关键任务系统动作延迟时间的概率分布复杂化.近年来,包括DARPA在内的诸多实验研究表明[6],系统从失效态(或不可控状态)恢复时,动作延迟时间的分布不同于传统上假设的指数分布,而应该是一般分布.因此从概率统计的角度来看,系统自恢复过程应该对应于一个SMP.然而,SMP建模过程复杂、修改开销较大,不能直观地表示交互.因此,本文尝试根据SM-PEPA对应于一个隐含半马尔可夫过程的特性,利用高级形式化语言SM-PEPA易于修改、表达直观的优点,建立关键任务系统自律可信性模型.此外,还可以采用其他数学理论替代SMP对SMP-PEPA进行分析,便于后续模型求解的改进.

2.2.1 可信性威胁的描述

针对可信性的威胁是可信性下降的直接诱因,对关键任务系统的可信性造成了巨大的危害,也是关键任务系统自律可信性模型必须描述的关键要素.由于通常暴露于复杂环境中,关键任务系统面临可信性威胁的数量巨大,然而目前对这些可信性威胁尚没有统一的分类.本文按照文献[13]中的分类方法,把可能面临的可信性威胁除硬件故障外分为 3 类:蓄意攻击、故障失效和偶发事故.

对这3类可信性威胁进行抽象,然后利用SM-PEPA对其进行描述.当蓄意攻击发生时,入侵者首先执行动作 search 搜索潜在的系统漏洞或脆弱点.若发现漏洞,则执行攻击准备动作 start_attack,以确定攻击的途径或方法,为后续攻击动作 attack 做准备;若未发现漏洞、不满足攻击条件或由于自身原因中止,则返回初始状态查找其余漏洞.系统随机故障可以分为两个步骤:故障发生阶段 Failure、可信性影响阶段 Failure0.其中,第 1 阶段为第 2 阶段提供了基础,而第 2 阶段是造成系统可信性下降的直接执行者;同理,偶发事故也可以抽象为两个阶段:事故发生阶段 Accident 和可信性影响阶段 Accident0.以上 3 种威胁可以并发执行,相互之间独立.因此,关键任务系统面临的可信性威胁可以用 SM-PEPA 描述如图 3 所示.

Intruder:=(*search*,*h*).*Attack*;
Attack:=(*start_attack*,*p*).(*attack*,*k*).*Attack*+(*start_attack*,*g*).*Intruder*;
Failure:=(*start_fail*,*i*).*Failure*0;
*Failure*0:=(*failing*,*q*1).*Failure*;
Accident:=(*operate*,*j*).*Accident*0;
*Accident*0:=(*error*,*q*2).*Accident*;

Fig.3 SM-PEPA model of TtD

图 3 可信性威胁的 SM-PEPA 模型

2.2.2 系统应对可信性威胁过程的描述

关键任务系统自律可信性模型的核心就是系统在遭受可信性威胁时,通过自省模块调控,采用自容忍、自恢复、自毁等机制保障系统可信性的过程.对关键任务系统应对可信性威胁的过程进行抽象,忽略与之无关的状态和活动,形成状态集合 $S=\{G,V,D,UC,ST,SH,SD,DG,UD\}$,其中,各状态含义及其之间的相互关系如图 4 所示.

假设系统从不可信状态恢复到正常态的恢复动作延迟时间服从一般分布F_{delta},其优先级为 2,且概率密度函数经拉普拉斯变换后为*delta*,而其他系统动作的延迟时间均服从指数分布,优先级为 1.在此条件下,根据图 4 采用SM-PEPA对自律可信系统应对可信性威胁的处理过程进行描述,如图 5 所示.

不妨设系统在初始时刻处于正常态 General,此时系统是可信的.当遭受到可信性威胁时,系统可能因产生相应的漏洞或弱点而变得脆弱,进入脆弱态 Vulnerable.由于这一过程中系统处于被动地位,因此可信性威胁 attack/failing/error 动作延迟参数为⊤.当进入脆弱态后,可信性开始下降,表现为动作 start_dependability_drop,如果可信性威胁通过动作 mask 被屏蔽,则系统重新恢复到正常态 General;如果漏洞或弱点被利用,则可信性会继续下降,系统中的感知器将收集这些信息表现为 probe 动作,此时进入检测态 Detection.

当处于 Detection 状态时,自省模块首先对各传感器收集的信息进行检测,表示为 start_detect.如果自省模块未感知到可信性威胁,则转入未检测状态 Uncover,此时可信性不可判定;若自省模块检测到可信性下降,则采取分级应对的策略.可信性分级保障策略可以表述为:首先判断核心机密是否即将外泄或生命财产安全即将受到重大损害,若是,则通过动作 enmergency1 进入自毁触发态 SelfDestruction;否则,判断是否符合容忍条件,如果符

合容忍条件,则通过动作 tolerance 转入容忍触发态 SelfTolerance;若不符合以上条件,则通过动作 healing 进入自恢复触发态 SelfHealing.

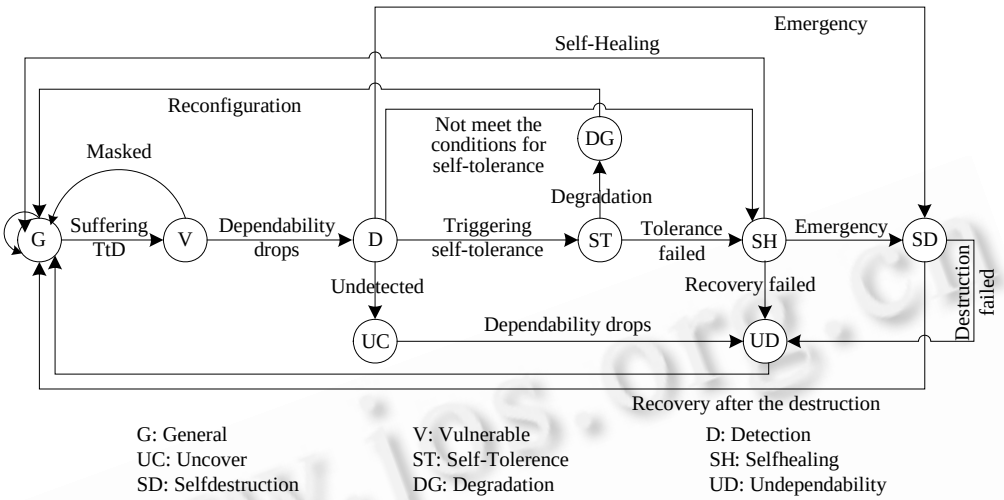


Fig.4 State transition diagram for system's response to TtD

图 4 系统应对可信性威胁过程的状态转移图

General:=(*attack*,⊤).*Vulnerable*+(*failing*,⊤).*Vulnerable*+(*error*,⊤).*Vulnerable*;
Vulnerable:=(*start_dependability_drop*,*L*1).(*probe*,*w*1).*Detection*+(*start_dependability_drop*,*L*2).(*mask*,*w*2).*General*;
Detection:=(*start_detect*,*L*3).*Uncover*+(*start_detect*,*L*4).(*enmergency*1,*p*1).*SelfDestruction*+
(*start_detect*,*L*5).(*tolerance*,*t*1).*SelfTolerance*+(*start_detect*,*L*6).(*healing*,*h*1).*SelfHealing*;
Uncover:=(*drop*,*d*1).*Undependability*;
SelfTolerance:=(*sart_tolerance*,*L*8).(*degrade*,*d*2).*Degradation*+(*sart_tolerance*,*L*9).(*fail_tolerate*,*h*2).*SelfHealing*;
Degradation:=(*degradation*,*L*10).(*reconfiguring*,*r*1).*General*;
SelfHealing:=(*start_heal*,*L*11).(*selfhealing*,*s*1).*General*+(*start_heal*,*L*12).*Undependability*+
(*start_heal*,*L*13).(*enmergency*2,*p*2).*SelfDestruction*;
SelfDestruction:=(*start_destroy*,*L*14).(*destroy*,*s*2).(*backup*,*b*).*General*+(*start_destroy*,*L*15).*Undependability*;
Undependability:=(*dependability_drop*,*L*16).(*recovery*[2],1:*delta*).*General*+
(*dependability_drop*,*L*17).(*fail*,*f*).*Undependability*;

Fig.5 SM-PEPA model of system's response to TtD

图 5 系统应对可信性威胁过程的 SM-PEPA 模型

进入 SelfTolerance 状态后,首先执行动作 sart_tolerance,通过门限函数或诱骗子网等机制对可信性威胁进行甄别,若成功,则通过 degrade 动作转入服务降级态 Degradation;若失败,则对可信性处理措施升级,系统通过 fail_tolerate 动作进入自恢复触发态 SelfHealing.当处于 Degradation 态后,首先执行服务降级动作 degradation,然后通过重配置 reconfigure 回到正常态 General.当处于 SelfHealing 态后,首先执行自恢复准备操作 start_heal,若执行自恢复动作 selfhealing 成功,则回到初始状态 General;若失败,则首先判断是否符合自毁条件,若是,则通过 enmergency2 动作转入自毁触发态;否则,转入非自律可信态 Undependability.当处于 SelfDestruction 状态时,首先执行自毁准备动作 start_destroy,若自毁触发失败将进入 Undependability 态,否则按照可信性威胁的程度进行不同范围自毁 destroy 动作.如果自毁成功,则因为保护了关键任务系统的核心机密或防止了生命财产损害,实现了私密性和防危性,本文认为也是自律可信的.从 SelfDestruction 状态可以通过 backup 动作实现备份恢复或同功能异构模块替换,再次恢复到正常态 General.当系统处于非自律可信性状态时,可信性继续下降,表现为动作 dependability_drop,此时,系统不能直接自律恢复,需要通过特定恢复措施回到正常态 General,例如根据人为检查,从备份中恢复;如果得不到恢复或恢复失败,表示为动作 fail,则继续保留在非自律可信性状态.

在该模型中,对某个延迟时间服从指数分布的动作.当具有两个或两个以上可能的执行结果时,每个执行结果发生的可能性(likelihood)为[14]

$$Likelihood_i = \frac{\lambda_i}{\sum_i \lambda_i},\ i = 1,2,... \tag{4}$$

其中,λ_i为结果i出现时动作的延迟时间参数.例如,当处于Detection状态时,start_detect动作延迟时间服从指数分布,有 4 种可能的结果,转到Uncover状态的可能性为$L3/(L4+L5+L6+L3)$.

如果系统面临 m 种蓄意攻击、n 种故障失效和 l 种偶发事故,则可信性威胁集合可以记为$threats=\{Intruder[m],Failure[n],Accident[l]\}$,其中,$Intruder[m]$,$Failure[n]$,$Accident[l]$分别代表蓄意攻击、系统故障失效和偶发事故这 3 种可信性威胁的序列.若$k\in\{m,n,l\}$,$k=0$,则表示其中相应的可信性威胁序列不存在.因此,关键任务系统自律可信性模型可以表示为

$$General \underset{\{attark,\ failing,error\}}{\triangleright\!\triangleleft} (Intruder[m] \triangleright\!\triangleleft Failure[n] \triangleright\!\triangleleft Accident[l]).$$

3 模型求解与量化分析

SM-PEPA 形式化模型的建立为自律可信性的量化分析提供了条件,下面将对 SM-PEPA 模型求解的原理和过程进行介绍,并利用求解结果提出一种自律可信性的量化方法.

3.1 模型的求解

SM-PEPA模型的求解与纯PEPA类似,可以通过分析对应的半马尔可夫过程来实现[8].由于后续对于自律可信性的量化用到了该SMP各状态的稳态概率,因此,本文将对SM-PEPA模型对应SMP各状态的稳态概率进行求解.

对于 SM-PEPA 中的任意组件 P,如果 $P \xrightarrow{(a_1,r_1)} ... \xrightarrow{(a_n,r_n)} P'$,则称 P'为 P 的派生,P 的派生的全体记为派生集$ds(P)$.各个派生之间通过活动进行连接建立派生图.派生图是一个标号有向多图(multigraph)[14].把SM-PEPA模型对应派生图中的各节点(local derivation)集合记为状态空间X_S.例如,当本文模型中$m=n=l=1$ 时,$\forall x\in X_S$,具有 $General' \underset{\{attark,\ fail,erro\}}{\triangleright\!\triangleleft} (Intruder' \triangleright\!\triangleleft Failure' \triangleright\!\triangleleft Accident')$ 的形式,其中,$General'$,$Intruder'$,$Failure'$,$Accident'$分别为组件$General$,$Intruder$,$Failure$和$Accident$的派生.

在派生图的基础上构建SM-PEPA对应的半马尔可夫过程$\{X,T\}=\{X_n,T_n,n=0,1,2,\ldots\}$,其中,$\forall n$,随机变量$X_n\in X_S$,$T_n\in[0,+\infty]$,$0=T_0\le T_1\le T_2\le\ldots$,且满足[9]:

$$P\{X_{n+1}=j,T_{n+1}-T_n\le t \mid X_0=i_0,...,X_n=i_n;T_0=t_0,...,T_n=t_n\} = P\{X_{n+1}=j,T_{n+1}-T_n\le t \mid X_n=i_n\} \tag{5}$$

该半马尔可夫过程的核 $Q(t)$,满足:

$$Q_{ij}(t) = P\{X_{n+1}=j,T_{n+1}-T_n\le t \mid X_n=i\} = p_{ij}H_{ij} \tag{6}$$

其中,$p_{ij}=P\{X_{n+1}=j \mid X_n=i\}$ 代表状态 i 和 j 之间的转移概率,$H_{ij}=P\{T_{n+1}-T_n\le t \mid X_{n+1}=j,X_n=i\}$ 代表状态 i 到 j 转移动作的驻留时间分布概率.

半马尔可夫过程的稳态概率公式为[15,16]

$$\pi_i = \frac{v_i E[T_i]}{\sum_j v_j E[T_j]},\ i,j\in X_S \tag{7}$$

其中,v_i代表半马尔可夫过程对应的嵌入马尔可夫链(embedded Markov chain,简称EMC)各状态的稳态概率,$E[T_i]$为在半马尔可夫过程中状态i的平均驻留时间.

半马尔可夫过程对应的EMC为$X=\{X_n,n=0,1,2,\ldots\}$,是一个状态空间为X_S、转移概率矩阵为$P=(p_{ij})$的齐次马尔可夫链[15],且

$$p_{ij} = \lim_{t\to+\infty} Q_{ij}(t) \tag{8}$$

设状态i可直接到达状态的集合为k,即从状态i经过一步转移即可到达的集合,记从状态i到集合k的所有动作的集合为$ACT(C_{ij})$.对本文对应模型约减后,对于某一确定的状态a,当$ACT(C_{aj})$中动作的延迟时间均服从指数分布时,从状态a到状态l的转移概率为 $p_{al}=r_{al}\Big/\sum_j r_{aj}$,其中,$r_{aj}$为动作延迟参数;当$ACT(C_{aj})$中存在延迟时

间参数为一般分布的动作recovery时,由于recovery优先级高于其余动作,通过(*recovery*[2],*delta*:*L*(*s*))活动到达确定的状态 q 的转移概率 $p_{aq}=\lim_{t\to+\infty}F_{delta}(t)=1$,到其余状态的转移概率为 0.

嵌入式半马尔可夫链的稳态概率满足[16]:

$$\begin{cases}\vec{v}=\vec{v}P\\ \sum_i v_i=1\end{cases} \tag{9}$$

其中,$\vec{v}$ 是嵌入的半马尔可夫链的稳态概率向量,可由公式(7)求得.

半马尔可夫过程状态i的平均驻留时间$E[T_i]$满足:

$$E[T_i]=\sum_{j=1}^{N}p_{ij}E[\tau_{ij}] \tag{10}$$

其中,$E[\tau_{ij}]$代表从状态i到状态j的过程中的平均时间延迟,可以通过H_{ij}对应的分布函数求得.

当恢复动作的延迟时间服从指数分布时,上述 SM-PEPA 模型退化为纯 PEPA 模型.此时,PEPA 对应一个连续时间马尔可夫过程,可以通过分析对应马尔可夫链(continuous-time Markov chain,简称 CTMC)来对模型求解.假设稳态概率分布是$\pi(\cdot)$,那么,

$$\begin{cases}\pi Q=0\\ \sum_{i=1}^{n}\pi_i=1\end{cases} \tag{11}$$

其中,$\pi=\{\pi_1,\pi_2,\ldots,\pi_n\}$是稳态概率向量.

综上即可求得 SM-PEPA 模型对应的半马尔可夫过程的稳态概率.为了降低上述求解过程的复杂性,本文在后续分析中利用伦敦帝国理工大学开发的 PEPA 工具 ipc 进行辅助求解,以简化计算过程.ipc 具体用法可参见文献[9,17].

3.2 自律可信性的量化分析

在自律能力量化中,通常用系统在无人为干预下正常运行的概率来表示系统的自律性.由于添加了自律计算的特性,传统可信性评价手段难以适用于自律可信性的度量,本文尝试借鉴自律计算和系统容侵能力的量化方法,从稳态概率角度对自律可信性进行量化,用系统在无干预的情况下保持可信运行的概率来表征关键任务系统的自律可信性,并记为自律可信指数.

借鉴入侵容侵研究领域Trivedi等人的研究成果[18,19],把自律可信性SM-PEPA模型的状态空间X_S分成两个集合:自律可信状态集X_1和非自律可信状态集X_2.X_2中的每一个状态(local derivation)包含不自律可信的状态和不可判定的状态,具有如下的形式:$X_2=\{x|x=Undependability \triangleright\!\triangleleft \ldots$或$Uncover \triangleright\!\triangleleft \ldots\}$.同理,稳态概率集合$\pi=\{\pi_1,\pi_2,\ldots,\pi_n\}$也可以分成两部分,对应于$X_1$的子集记为$C_D$,另一子集记为$C_{UD}$.

定义 5(自律可信指数). 设$\pi_1,\pi_2,\ldots,\pi_s$分别为各自律可信状态的稳态概率,则自律可信指数记为Γ_{AD},$\Gamma_{AD}\in(0,1)$,满足:

$$\Gamma_{AD}=\sum_{\pi_i\in C_D}\pi_i \tag{12}$$

即系统处于自律可信态的稳态概率之和.

下面结合一个实例对Γ_{AD}进行举例说明.为了问题的简化,假设某关键任务系统对应的自律可信性模型中$m=n=l=1$,可信性威胁动作的延迟时间均服从指数分布,并参照文献[8]设恢复动作*recovery*的延迟服从伽玛分布*gamma*(2,2),SM-PEPA模型其余参数的取值见表 1.

首先把表 1 中的参数带入到自律可信性 SM-PEPA 模型中,然后根据公式(5)构建对应的半马尔可夫过程,得到 276 个状态,最后利用公式(9)解得隐含 SMP 的各状态的稳态概率见表 2.

把表 2 中的稳态概率带入公式(12),可以求得该实例的自律可信指数为 0.996 112.

Table 1 Parameters of SM-PEPA model

表 1 SM-PEPA 模型的参数取值

Parameters	Values	Parameters	Values	Parameters	Values	Parameters	Values
h	0.001 8	$L2$	$1-L1$	$L8$	0.7	$p2$	0.1
p	0.6	$w2$	1.0	$d2$	0.75	$L14$	0.95
k	0.3	σ	0.999 5	$L9$	$1-L8$	$s2$	0.3
g	0.4	$L3$	$1-\sigma$	$h2$	1.0	b	0.2
i	0.3	$L4$	0.000 1	$L10$	1.0	$L15$	$1-L14$
$q1$	0.2	$p1$	0.1	$r1$	1.0	$L16$	0.9
j	0.9	$L5$	$0.7\times\sigma$	$L11$	0.995	$h1$	0.5
$q2$	0.3	$t1$	0.5	$s1$	0.95	$L17$	$1-L16$
$L1$	0.75	$L6$	$\sigma-L4-L5$	$L13$	0.000 1	f	0.05
$w1$	1.0	$d1$	0.01	$L12$	$1-L11-L13$		

Table 2 Steady-State probability of SM-PEPA model

表 2 SM-PEPA 模型的稳态概率

No.	State	Steady-State probability π_i
1	{*General*,*Intruder*,*Failure*,*Accident*}	2.698842923978399E−4
2	{*General*,*Attack*,*Failure*,*Accident*}	5.320305273151165E−7
3	{*General*,*Intruder*,*Failure*0,*Accident*}	0.004 142 361 688 472 932
4	{*General*,*Intruder*,*Failure*,*Accident*0}	0.017 473 375 636 472 086
5	{*General*,(*attack*,0.3).*Attack*,*Failure*,*Accident*}	7.2678632970561935E−6
6	{*General*,*Attack*,*Failure*0,*Accident*}	8.539402898307569E−6
7	{*General*,*Attack*,*Failure*,*Accident*0}	3.71630841735485E−5
8	{*Vulnerable*,*Intruder*,*Failure*,*Accident*}	0.002 758 191 146 901 304 6
9	{*General*,*Intruder*,*Failure*0,*Accident*0}	0.195 533 161 146 041 44
…	…	…
274	{(*reconfiguring*,1.0).*General*,*Attack*,*Failure*0,*Accident*0}	6.791031194137566E−5
275	{*Degradation*,(*attack*,0.3).*Attack*,*Failure*0,*Accident*0}	0.002 320 195 467 861 16
276	{(*reconfiguring*,1.0).*General*,(*attack*,0.3).*Attack*,*Failure*0,*Accident*0}	0.002 386 548 477 079 751

4 模型参数效应分析

关键任务系统自律可信性模型包含许多参数.这些参数的取值对模型的稳定性和合理性存在着一定的影响.本节以自律可信指数为指标,对可信性威胁构成、自律可信属性以及可信性威胁检测率等参数的取值对自律可信性的影响进行简要分析,并求出模型的首次通过时间概率密度函数,为确定影响自律可信性的关键因素提供参考.

(1) 可信性威胁构成对自律可信性的影响

可信性威胁的数量与构成通常是表征可信性威胁程度最为重要的参数.关键任务系统的使命性决定了它必须能够应对一定量的可信性威胁,避免因可信性威胁数量的激增或构成的变化而使自律可信性急剧下降或发生较大的波动.

为使问题求解简便,首先假设可信性威胁序列 $threats$ 中的每种可信性威胁的延迟时间服从相同的分布,若$threats^i$为蓄意攻击、偶发事故、故障失效中任意一类,则$\forall threats^i \subset threats, \forall\ threats_a^i, threats_b^i \in threats^i$, $threats_a^i = threats_b^i$.换言之,$threats^i$集合中的元素是相同的.然后,通过改变自律可信性模型中参数m,n,l的取值,利用第 3 节中的求解方法得到表 3.

由表 3 可以看出,随着可信性威胁构成的改变,自律可信性维持在较高水平,且变化较小,说明可信性威胁构成的改变对系统的自律可信性影响较弱,系统具有良好的稳定性.

(2) 自律可信属性对自律可信性的影响

自容忍、自恢复和自毁等属性是关键任务系统自律可信性的核心保障措施.下面首先来分析自容忍、自恢复和自毁动作延迟时间参数的改变对自律可信指数的影响.

在一个纯 PEPA 模型中,平均延迟时间的值恒为动作延迟时间参数λ的倒数 $1/\lambda$.由于本文模型的特殊性,仅

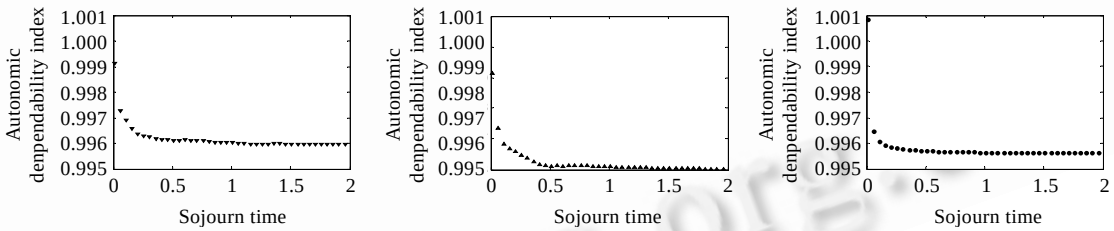
有 1 个动作 *recovry* 延迟时间服从一般分布,而自容忍、自恢复、自毁的动作延迟时间均服从指数分布,因此根据 SM-PEPA 的特性,这 3 种动作不与 *recovry* 并发执行,且平均延迟时间是 $1/\lambda$.下面以动作延迟时间为自变量,以自律可信指数为因变量,通过改变平均延迟时间的值,利用公式(10)和公式(12)分析自容忍、自恢复和自毁动作延迟时间的改变对自律可信指数的影响.

Table 3 Impact of the composition of TtD on autonomic dependability

表 3 可信性威胁的构成对自律可信性的影响

m	*n*	*l*	Number of states in state space	Autonomic dependability index
1	—	—	69	0.999 986
—	1	—	46	0.995 246
—	—	1	46	0.995 246
10	—	—	1 386	0.998 950
—	10	—	253	0.995 246
—	—	10	253	0.995 246
1	1	1	276	0.996 112
3	3	3	3 680	0.995 082
5	1	1	1 392	0.995 475
1	5	5	2 484	0.994 988
5	5	5	17 388	0.994 673

图 6(a)~图 6(c)分别显示了自容忍、自恢复和自毁动作的延迟时间对自律可信指数的影响.以图 6(b)为例,随着自恢复的延迟时间的增大,自律可信性逐渐降低;但当延迟时间趋近于 0.5 时,自律可信性趋于平稳,保持在 0.996 左右.其主要原因是,随着自恢复的平均延迟时间 $1/\lambda$的增大,在$[0,t]$时间内自恢复变迁概率 $1-e^{-\lambda t}$减小,系统有可能得不到及时恢复,自律可信性降低;然而,当自恢复延迟时间大于一定值时,由于仍能完成自恢复,因此自律可信性将保持在一定的范围内.同理,图 6(a)和图 6(c)中自容忍和自毁延迟时间对可信性影响曲线也呈现出先降低后趋于平缓的趋势.



(a) Self-Tolerance (a) 自容忍 (b) Self-Healing (b) 自恢复 (c) Self-Destruction (c) 自毁

Fig.6 Impact of the sojourn time of self-tolerance, self-healing and self-destruction on Γ_{AD}

图 6 自容忍、自恢复和自毁动作延迟时间对Γ_{AD}的影响

接下来,以自恢复成功率为例分析自恢复对保障自律可信性的影响.根据提出的模型,由于自恢复准备动作 start_heal 服从指数分布,优先级为 1,它具有 3 种可能的执行结果,由公式(4)得到自恢复成功率 $v11=L11/(L11+L12+L13)$,带入表 1 中的数据可得 $v11=L11$.图 7 的横轴为自恢复驻留时间,纵轴为自律可信指数,4 条曲线分别描述了自恢复成功率为 0.5,0.75,0.9 和 0.995 时对自律可信性的影响.可见,在自恢复延迟时间相同的条件下,自恢复成功率越高,系统自律可信性越强;当等于 0.995 时,自律可信性趋近于 1.这表明,自恢复成功率的增大,提高了系统从不可信状态恢复的概率,有助于保障系统的自律可信性,是自律可信性增长的关键因素之一.

(3) 可信性威胁检测率对自律可信性的影响

可信性威胁检测率是可信性保障措施实施的基础,直接决定着是否可以采用自容忍、自恢复和自毁来处理系统面对的可信性威胁.

根据公式(4),未检测到可信性威胁(即转移到uncover态)的概率为$Likelihood_{uc}=L3/(L4+L5+L6+L3)$,那么可

信性威胁的检测概率为 1–$Likelihood_{uc}$,带入表 1 的数据可知,1–$Likelihood_{uc}$=σ.因此,以动作延迟时间为横轴,以自律可信指数为纵轴作图,通过改变σ的取值得到如图 8 所示的结果.

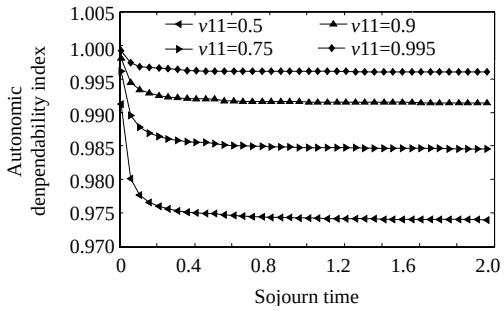


Fig.7 Impact of the successful rate of self-healing on Γ_{AD}

图 7 自恢复成功率对Γ_{AD}的影响

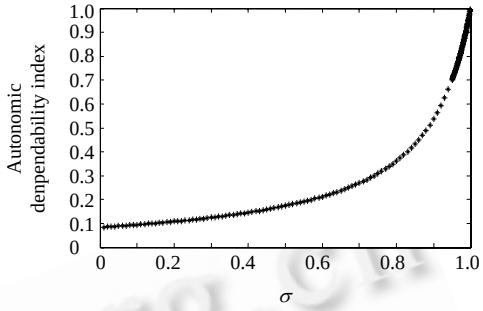


Fig.8 Impact of detection rate of TtD on Γ_{AD}

图 8 可信性威胁检测率对Γ_{AD}的影响

由图 8 可以看出,当可信性威胁检测率很低时,系统的自律可信指数很小.这是由于存在大量未检测出的可信性威胁,导致系统的自律可信性急剧下降.随着检测概率的增大,自律可信性呈上凹趋势($\Gamma''_{AD}>0$),快速提高;当检测概率趋近于 1 时,自律可信性也逼近于 1.这主要是因为各种可信性威胁被检测出后已经被分级处理,导致系统的自律可信性提高.综上所述,不可信检测概率对自律可信性的影响巨大,也是自律可信性研究的关键点之一.

(4) 首次通过时间概率密度函数

首次通过时间(first passage time)是马尔可夫或半马尔可夫过程分析中一个常用的参数,与驻留时间等参数紧密相关.下面将对从可信性威胁出现到系统从不可信状态恢复的首次通过时间概率密度函数(probability density function,简称 PDF)和积累分布函数(cumulative distribution function,简称 CDF)进行求解,以分析系统的可信性.

对于一个有限不可约简的半马尔可夫过程,设其具有 N 个状态{1,2,…,N},令 $Z(t)$表示 t 时刻(t>0)半马尔可夫过程所处的状态,$N(t)$代表t时刻发生的状态转移数目,由状态i到达非空目标集 $\vec{j}$ 的首次通过时间满足[20]:

$$P_{i\vec{j}}(t)=inf\{u>0:Z(t+u)\in \vec{j},N(t+u)>N(t),Z(t)=i\} \tag{13}$$

设 $P_{i\vec{j}}(t)$ 的概率密度函数为 $f_{i\vec{j}}(t)$,对 $f_{i\vec{j}}(t)$ 进行 Laplace 变换得到 $L_{i\vec{j}}(s)$,

$$L_{i\vec{j}}(s)=\sum_{k\notin\vec{j}} r^*_{ik}(s)L_{k\vec{j}}(s)+\sum_{k\in\vec{j}} r^*_{ik}(s),\ 1\le i\le N \tag{14}$$

其中,k 为由状态 i 一步到达的状态集, $r^*_{ik}(s)=\int_0^\infty e^{-st}dR(i,k,t),R(i,j,t)=P(Z_{n+1}=j,T_{n+1}-T_n\le t\mid Z_n=i),\forall n\ge 0.$

当有多个初始状态时,i记为 $\vec{i}$,那么[17,20],

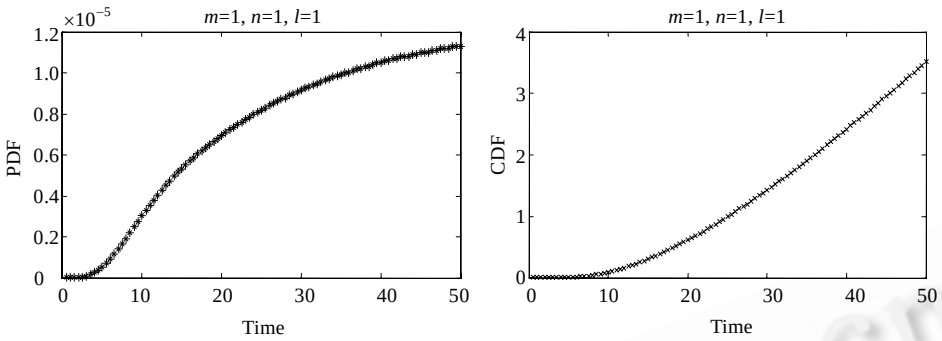
$$L_{i\vec{j}}(s)=\sum_{k\in\vec{i}}\alpha_k L_{k\vec{j}}(s) \tag{15}$$

其中,α_k是在初始时刻状态k的稳态概率,

$$\alpha_k=\begin{cases}\pi_k/\sum_{j\in\vec{i}}\pi_j, & \text{if } k\in\vec{i}\\ 0, & \text{其他}\end{cases} \tag{16}$$

其中,π是对应嵌入式马尔可夫链的稳态概率向量.

再对 $L_{i\vec{j}}(s)$ 进行拉普拉斯逆变换即可得到 $f_{i\vec{j}}(t)$.根据公式(13),由表 1 中的数据可以解得当 m=n=l=1 时,SM-PEPA 模型从可信性威胁出现到系统从不可信状态恢复的首次通过时间 PDF 和 CDF,如图 9 所示.由图 9 中可以看出,在 50 个时间单位内,PDF 和 CDF 都较小,这说明本文提出的模型在一定程度上会降低系统进入不可信状态的可能性.



(a) PDF (b) CDF

Fig.9 PDF and CDF for the time taken from the appearances of TtD to the first recovery from undependability state

图 9 从可信性威胁出现到系统从不可信状态恢复的首次通过时间 PDF 和 CDF

5 相关工作与比较

Autonomic一词最早来源于人体自律神经系统autonomic nerve system,主要是指在无意识控制下进行的动作或反应[21].根据这一思想,IBM提出了“自律计算”一词,其目标是设计一种能够自动运行、动态适应环境改变的计算系统,具有有效调配资源和预期下一步需求的能力,在向用户提供优质服务的同时隐藏其自身的复杂性.自律计算由于更加自动化地运行和高效利用可用资源,极大地降低了服务开销,已广泛应用于网络控制[22]、数据库管理[23]、太空探索[24]等多种领域.在国内,autonomic computing一词由于翻译方法和惯用领域的不同又称为自主计算,少数情况下也译为自治计算.但是通常情况下,自主与自治计算对应于英文autonomous一词.虽然,“自律”与“自治”含义有重叠之处,但是按照IBM的解释,自主与自治计算主要强调“系统自动或独立的执行,不受无外界的干预”[21],而与自律计算“无意识、自发地做某事”[21]有较大的区别.

由于自律计算的自保护、自恢复等属性所表现出的特点,Murch认为,它虽然与微软提出的可信计算(trustworthy computing)无直接关联,但是二者具有相似的功效[21].本文正是基于这些特性,尝试利用自律计算来处理关键任务系统面临的可信性威胁,进而提高系统的可信性.

目前,对自律计算技术与可信性相结合的研究还处于初步阶段,但是已受到诸多研究机构和学者的关注.2003 年,Sterritt首次提出利用自律计算实现可信结构[2],主要探讨了如何利用自律计算技术消除和避免系统错误从而达到可信的目的.日本金泽科技学会的Kunii在文献[25]中从方法学层面讨论了不变量对可信信息系统的作用,并建议通过减少变量并保留不变量来实现自律,最终达到可信的目的.上述研究工作还主要集中在概念研究上,但是并没有给出自律可信性的明确定义,尚未形成系统的理论体系.

除此之外,许多学者对自律与可信相结合的模型进行了探索.2003 年,Tohma等人提出把自律计算技术应用于容错系统构建可信自律计算系统[3],重点探讨了反常情况通告(notification of abnormity)、计算单元协商等关键性因素,以及可以采用的实现方法.与之相似,Baldini等人通过添加自恢复特性提高系统的可信性,设计了一个用于测试复杂异构系统的可信自律计算环境[4].清华大学管晓宏教授等人在文献[5]中充分总结和分析了现有各种利用自律计算实现可信的途径,并且基于模型驱动的方法建立了一个可信的自律管理系统架构,以提高系统的可靠性和安全性,然而缺少对具体实现的说明.针对服务分发过程中动态变更管理对自律计算系统可信性的影响,卡内基梅隆大学的Dumitras等人提出了一种变更管理框架[26],通过减少变更管理对服务的影响,提高系统的可信性.此外,针对卫星系统的错误诊断与恢复,NASA开发了一种以实现可信航天计算为目的的自律管理器,用来消除环境条件改变和错误发生带来的影响[27].但是,该方法具有很强的平台相关性,需要进一步修改才能应用于普通的关键任务系统.综上所述,现有模型普遍存在如下的不足:

(1) 主要采用自然语言或框图作为建模手段,缺乏精确的语义和推理分析能力,不能满足系统对自律可信性形式化验证的需要.

(2) 通过添加某些现有可信技术构建可信的自律计算系统,主要关注于自律计算本身所蕴含的可信特性,还未充分利用“以技术管理技术”的主动式管理特点.

(3) 大多针对可信性的某一方面且实现手段单一,难以处理不同程度的可信性威胁,尤其是难以满足关键任务系统对私密性和防危性等特殊可信性的需求.

此外,对自律可信性量化与评价的研究尚未见诸报端.

与以往的工作相比,本文在总结现有研究成果的基础上,首先提出了一种基于SM-PEPA的关键任务系统自律可信性模型,与Baldini和Tohma的模型相比抛弃了传统的基于框图与自然语言的描述方式,利用形式化语言进行描述,具有更为严格的语义,且支持模型的推理与验证;其次,在包含Baldini和Tohma可信性保障方法的前提下,引入自毁技术,形成可信性威胁的分级应对措施,有针对性地满足了关键任务系统对可信性的特殊需求;再次,本文模型在自我管理能力方面比现有模型有所改进,与以往模型只注重上下文感知不同,自省属性兼顾了系统内部状态和外部环境改变带来的可信性变化,且贯穿于模型演化的整个过程,对用户屏蔽了下层结构的变更,更好地隐藏了系统的复杂性.此外,依托建立的自律可信性SM-PEPA模型,从稳态概率角度提出了一种关键任务系统自律可信性量化方法,为后续自律可信性的量化与评价提供了参考依据.

6 结论与下一步工作

系统管理复杂性的增长,导致传统可信实现方式难以满足当前系统需求.由于自律计算不但蕴含可信特性而且具有“以技术管理技术”的特征,被认为是一种解决可信问题的新的有效途径.本文在对国内外研究现状分析的基础上,针对现有自律可信性模型缺乏推理验证能力的缺点提出了一种基于SM-PEPA的关键任务系统自律可信性模型,以满足自律可信性研究对形式化推理和量化分析的需要.由于采用了自律反馈控制结构,该模型具有一定的自管理能力,克服了管理复杂性对可信性实现方式的制约.同时,自容忍、自恢复、自毁三级处理模式可以分级应对不同程度的可信性威胁,满足了关键任务系统对于诸如私密性和防危性等特殊可信性需求.在此基础上,从稳态概率角度提出了一种自律可信性量化方法,尝试利用自律可信指数Γ_{AD}对系统的自律可信性进行度量.最后,结合具体实例,以Γ_{AD}为指标分析了部分模型参数对关键任务系统自律可信性的影响.分析结果表明,增大关键任务系统可信性威胁检测概率和自恢复成功率,将在较大范围内提高系统的自律可信特性.本文的工作为后续自律可信性模型的建立和量化评价提供了借鉴.

本文下一步研究工作的重点是建立实际的具有自律可信性的关键任务系统来代替仿真实验,为自律可信性的进一步研究提供基础.此外,对于大型分布式的关键任务系统而言,由于同一时刻可能面临大量可信性威胁,导致对应隐含SMP状态空间巨大,PEPA工具ipc求解困难且时间开销较大,需要设计更为有效的自动化求解工具.

致谢 在此,我们向对本文的工作给予支持和建议的各位评审专家、编辑和在NASAC 2009会议上提出宝贵修改意见的毛新军教授表示感谢.

References:

[1] Kephart JO, Chess DM. The vision of autonomic computing. Computer, 2003,1(36):41−45.

[2] Sterritt R, Bustard D. Autonomic computing: A means of achieving dependability? In: Proc. of the 10th IEEE Int'l Conf. and Workshop on the Engineering of Computer-Based Systems. Huntsville: IEEE Computer Society Press, 2003. 247−251.

[3] Tohma Y. Fault tolerance in autonomic computing environment. In: Proc. of the 2002 Pacific Rim Int'l Symp. on Dependable Computing. Tsukuba: IEEE Computer Society Press, 2002. 2503−2507.

[4] Baldini A, Benso A, Prinetto P. A dependable autonomic computing environment for self-testing of complex heterogeneous systems. Electronic Notes in Theoretical Computer Science, 2005,116(19):45−57.

[5] Dai YS, Marshall T, Guan XH. Autonomic and dependable computing moving towards a model-driven approach. Journal of Computer Science, 2006,6(2):496−504.

[6] Madan BB, Goševa-PopstojanovaP K, Vaidyanathan K, Trivedi KS. Modeling and quantification of security attributes of software systems. In: Proc. of the 2002 Int'l Conf. on Dependable Systems and Networks. Bethesda: IEEE Computer Society Press, 2002. 505−514.

[7] Hillston J. Tuning systems: From composition to performance. The Computer Journal, 2005,48(4):385−400.

[8] Bradley JT. Semi-Markov PEPA: Compositional modelling and analysis with generally distributed actions. In: Proc. of the 20th Annual UK Performance Engineering Workshop. Irfan Awan: University of Bradford, 2004. 266−275.

[9] Bradley JT. Semi-Markov PEPA: Modelling with generally distributed actions. Int'l Journal of Simulation, 2005,6(3):43−51.

[10] Xing XJ, Lin C, Jiang YX. A survey of computer vulnerability assessment. Chinese Journal of Computers, 2004,27(1):1−11 (in Chinese with English abstract).

[11] Liao BS, Li SJ, Yao Y, Gao J. Conceptual model and realization methods of autonomic computing. Journal of Software, 2008, 19(4):779−802 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/19/779.htm

[12] Candea G, Brown AB, Fox A, Patterson D. Recovery oriented computing: Building multi-tier dependability. Computer, 2004, 11(37):60−67.

[13] Ellison RJ, Fisher DA, Linge RC, Lipson HF, Longstaff T, Mead NR. Survivable network systems: An emerging discipline. Technical Report, CMU/SEI-97-TR-013, Pittsburgh: Carnegie Mellon University, 1999. 1−33.

[14] Hillston J. A Compositional Approach to Performance Modelling. Cambridge: Cambridge University Press, 1996. 17−43.

[15] Dingle NJ. Parallel computation of response time densities and quantiles in large Markov and semi-Markov models [Ph.D. Thesis]. London: Imperial College, University of London, 2004.

[16] Gokhale SS, Wong WE, Horgan JR, Trivedi KS. An analytical approach to architecture-based software performance and reliability prediction. Performance Evaluation, 2004,58(4):391−412.

[17] Clark A. ipc: Imperial PEPA compiler. 2009. http://www.doc.ic.ac.uk/ipc/

[18] Sharma VS, Trivedi KS. Quantifying software performance, reliability and security: An architecture-based approach. The Journal of Systems and Software, 2007,80(4):493−509.

[19] Yin LH, Fang BX. Security attributes analysis for intrusion tolerant systems. Chinese Journal of Computers, 2006,29(8): 1505−1512 (in Chinese with English abstract).

[20] Bradley JT, Dingle NJ, Harrison PG, Knottenbelt WJ. Distributed computation of transient state distributions and passage time quantiles in large semi-Markov models. Future Generation Computer Systems, 2006,22(7):828−837.

[21] Murch R. Autonomic Computing. Upper Saddle River: Pearson Education Inc., 2004. 223, 289−292.

[22] Dong XD, Hariri S, Xue LZ, Chen HP, Zhang M, Pavuluri S, Rao S. Autonomia: An autonomic computing environment. In: Proc. of the 2003 IEEE Int'l Conf. on Performance, Computing, and Communications. Phoenix: IEEE Computer Society Press, 2003. 61−68.

[23] Elnaffar S, Powley W, Benoit D, Martin P. Today's DBMSs: How autonomic are they? In: Proc. of the 14th Int'l Workshop on Database and Expert Systems Applications. Prague: Republic IEEE Computer Society Press, 2003. 651−655.

[24] Hinchey M, Dai YS, Rouff CA, Rash JL, Qi MR. Modeling for NASA autonomous nano-technology swarm missions and model-driven autonomic computing. In: Proc. of the 21st Int'l Conf. on Advanced Networking and Applications (AINA 2007). Niagara Falls: IEEE Computer Society Press, 2007. 250−257.

[25] Kunii TL. Autonomic and trusted computing for ubiquitous intelligence. In: Proc. of the 4th Int'l Conf. on Autonomic and Trusted Computing (ATC 2007). Berlin, Heidelberg: Springer-Verlag, 2007. 1−12.

[26] Dumitras T, Rosu D, Dan A, Narasimhan P. Dynamic change management for minimal impact on dependability and performance in autonomic service-oriented architectures. Technical Report, CMU-CyLab-06-003, Pittsburgh: CyLab, Carnegie Mellon University, 2006. 1−20.

[27] Troxel IA, George AD. Adaptable and autonomic mission manager for dependable aerospace computing. In: Proc. of the Symp. on Dependable, Autonomic and Secure Computing. Indianapolis: IEEE Computer Society Press, 2006. 11−18.

附中文参考文献:

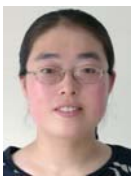
[10] 邢栩嘉,林闯,蒋屹新.计算机系统脆弱性评估测评研究.计算机学报,2004,27(1):1−11.

[11] 廖备水,李石坚,姚远,高济.自主计算概念模型与实现方法.软件学报,2008,19(4):779−802. http://www.jos.org.cn/1000-9825/19/779.htm

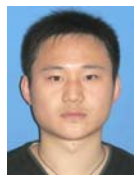
[19] 殷丽华,方滨兴.入侵容忍系统安全属性分析.计算机学报,2009,29(8):1505−1512.

王慧强(1960－),男,河南周口人,博士,教授,博士生导师,CCF 高级会员,主要研究领域为信息安全,自律计算与可信计算,认知网络.

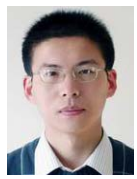
吕宏武(1983－),男,博士生,主要研究领域为自律计算与可信计算,自省模型研究.



赵倩(1980－),女,博士生,CCF 学生会员,主要研究领域为软件可信性增长及评价.



董玺坤(1984－),男,博士生,主要研究领域为自律计算,软件快速恢复技术.



冯光升(1980－),男,博士,讲师,主要研究领域为软件演化,认知网络.