Journal of Computer Applications
计算机应用,2013,33(2):400-403
ISSN 1001-9081
CODEN JYIIDU
2013-02-01
http://www.joca.cn

文章编号:1001-9081(2013)02-0400-04
doi:10.3724/SP.J.1087.2013.00400

基于马尔可夫的 Web 应用生存性模型

秦志光,宋　旭*,耿　技,陈　伟
(电子科技大学 计算机科学与工程学院,成都 611731)
(*通信作者电子邮箱 yongyongtiyan@gmail.com)

摘　要:针对现有生存性模型缺乏实践指导意义及不能刻画 Web 应用特性的问题,对 Web 应用的特点进行讨论,尤其是对原子 Web 应用和组合 Web 应用的区别及特点进行探讨,重点考虑了如何对组合 Web 应用中各原子 Web 应用之间的调用关系进行分析和建模;同时通过将环境引入到生存性的分析中,分别构建了原子 Web 应用的生存性模型和基于马尔可夫过程模型的组合 Web 应用生存性模型。根据建立的 Web 应用生存性模型,提出一个在 Web 应用处于不利环境中时,部分或全部服务失效情况下的恢复方案。最后通过已建立的模型对一个案例进行了分析,给出了其恢复过程,在恢复过程中保证了较好的生存性。

关键词:原子 Web 应用;组合 Web 应用;生存性;运行环境;恢复;马尔可夫过程模型
中图分类号:TP309;TP393.09;TP311.522　　**文献标志码**:A

Markov-based survivability model for Web applications

QIN Zhiguang, SONG Xu*, GENG Ji, CHEN Wei
(*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu Sichuan* 611731, *China*)

Abstract: Current survivability models can hardly bring up a practical solution, nor reflect the properties of Web applications. Firstly, the properties of Web applications were analyzed, especially the differences between atomic Web applications and composite Web applications. Secondly, the mathematical model reflecting the invoking relationship of the atomic Web applications for the development of composite Web application was constructed. Lastly, a survivability model for atomic Web applications and a Markov-based survivability model for composite Web applications with regard to runtime environment were proposed. And on the basis of these models, a recovery approach for Web applications was given, when part or all of the functions failed in adverse environment. Besides, a case was analyzed using these models, and its recovery procedures were given, in which the high survivability was guaranteed.

Key words: atomic Web application; composite Web application; survivability; runtime environment; recovery; Markov process model

0　引言

随着科学技术的发展,尤其是互联网信息技术的发展,互联网已经成为了公共发布和信息获取的一个重要的平台,而 Web 应用也由于它的松散耦合、平台无关、位置透明等特性取得了较大的发展。

Web 应用采用典型的面向服务的架构,目前对于 Web 服务有着不同的定义[1],其中 Stencil Group 认为,Web 应用是一种松散耦合的、可复用的软件组件,这些组件是对具体功能的封装,这些功能可能并不相关,并且由程序通过标准的网络协议载入[2]。Web 应用典型的面向服务的设计方法,在方便了 Web 应用构建的同时,也使得设计人员常常对于采用的第三方提供的服务模块的具体构架并不了解,从而导致在安全性、可靠性和生存性上产生了新的问题,本文主要从生存性层面进行研究。

由于目前软件的结构越来越复杂,即使最优秀的系统分析和设计人员也无法对其进行完整的分析,因此传统的软件安全技术已经无法保证其对于可靠性的要求[3]。对于安全的关注点也由入侵阻止、入侵检测发展到了可生存性研究的层面[4]。一般而言,生存性可以定义为:系统遭遇攻击、失效或者事故时,系统可以及时恢复并完成其任务的能力[5-10]。

目前,对于生存性的研究工作主要侧重于如何使得系统服务可以经受住不利环境的影响,以及系统在受到攻击、自然灾害等不利影响时的恢复策略两方面,但是无论何种生存性策略,都定义系统能够正确运行所必需的服务和系统应该提供的核心服务为系统关键服务,并且要求在无论何种环境下,系统的关键服务都应该得到保证。

文献[11]通过对系统生存性进行数学分析,提出了一个形式化的系统生存性定义和评估模型,以及基于该模型制定系统服务恢复策略的方法。但是该模型没有考虑系统处于的具体操作环境以及系统架构对于生存性的影响,且在服务和服务操作集合的定义上也没有考虑到复用的存在。文献[12]从服务和服务操作集合复用的角度对文献[11]提出的系统生存性定义和评估模型进行了改进,但是依然没有考虑到环境因素对于生存性的影响。

文献[3]研究了环境对生存性的影响,指出同一系统在不同环境下所获得的生存性可能不相同,并进一步分析了环境可能对生存性产生的影响。

收稿日期:2012-08-01;**修回日期**:2012-09-20。
基金项目:国家自然科学基金资助项目(60973118);中央高校基本科研业务费资助项目(ZYGX2011J072)。
作者简介:秦志光(1956-),男,四川隆昌人,教授,博士生导师,主要研究方向:开放系统、中间件、信息安全;　宋旭(1986-),男,河北衡水人,硕士研究生,主要研究方向:软件确保、Web 应用安全;　耿技(1963-),男,安徽合肥人,教授,博士研究生,主要研究方向:系统软件、软件确保、信息安全;　陈伟(1978-),男,四川温江人,讲师,博士研究生,主要研究方向:无线网络路由、网络安全。

对于 Web 应用的生存性研究,目前相关文献较少。文献[13]提出了一个需求驱动的自适应的 Web 系统生存性确保框架,文献[14]则提出了一个光纤网中的生存性解决方案。但是其并未对 Web 应用的结构特征进行反映,也缺乏具体可行的实施方案。

本文中针对于 Web 服务的架构,以文献[11]中提出的服务生存性形式化模型为基础,提出了 Web 应用的生存性评估模型,并基于此提出了一个 Web 应用的生存性恢复策略。

1　Web 应用的生存性模型

Web 应用可以分为两类[15]:原子性 Web 应用(不依赖于其他 Web 应用实现的 Web 应用)和组合 Web 应用(由原子性 Web 应用组成的 Web 应用)。以下分别就两种形式的 Web 应用对其生存性进行考察。

1.1　原子 Web 应用的生存性

文献[11]中对服务的生存性以六元组 $(S,SO^{(i)},\varphi,\Lambda,F,\Delta)$ 的形式进行了形式化定义,其定义如式(1)所示。

$$\zeta^{(i)}=\begin{cases}0, & SO_a^{(i)}\in SO_{ai}^{(i)}\\ \dfrac{\sum_{j=1}^{|so_a^{(i)}|}\omega_j^{(i)}\,|\,so_j^{(i)}\in SO_a^{(i)}-SO_c^{(i)}}{\sum_{j=1}^{|SO^{(i)}|}\omega_j^{(i)}\,|\,so_j^{(i)}\in SO^{(i)}-SO_c^{(i)}}, & \text{其他}\end{cases}\tag{1}$$

对于原子 Web 应用,六元组中 S 即为自身。但是对于 Web 应用而言,其可能需要在多种环境下提供有区别的服务,这样造成在不同的环境中,其关键服务的定义可能会有所不同;另外由于可能会发生攻击、网络故障等情况,其处于的网络环境也是复杂多变的,但是式(1)并不能反映出环境对生存性的影响。因此在对生存性的定义中加入环境因素(用 E 表示),从而将该生存性定义六元组拓展为 $(S,SO^{(i)},\varphi,\Lambda,F,\Delta,E)$。

对于环境 $e_l\in E$,如式(1)可以定义原子 Web 应用的生存性 $\zeta_{e_l}^{(i)}$,在一段相当长的考察周期内,环境 e_l 出现的概率为 P_{e_l},如此可以如式(2)定义某一原子 Web 应用的生存性:

$$\zeta^{(i)}=E(\zeta_{e_l}^{(i)})=\sum_{i=1}^{k}P_{e_l}\cdot\zeta_{e_l}^{(i)}\tag{2}$$

1.2　组合 Web 应用的生存性

组合 Web 应用由已经发布的原子 Web 应用组合而成,组合 Web 应用现阶段实现的发布—查找—绑定模型如图 1[16]所示。

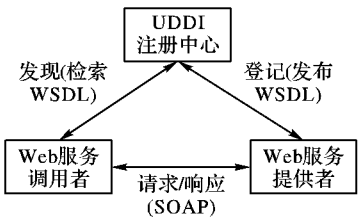

图 1　Web 服务发布—查找—绑定模型

组合 Web 应用功能的实现依赖于对原子 Web 应用的调用,因此可以通过对原子 Web 应用的生存性分析以及其调用关系分析组合 Web 应用的生存性。

通过对 Web 应用实现进行分析,可以对其调用关系通过马尔可夫状态迁移进行描述。每个 Web 应用被调用视为一个状态,其状态迁移概率为在一个可能的 Web 应用调用序列中,另一 Web 应用(包括自身)为此 Web 应用的后继的概率。整个软件服务间的调用关系可以使用一个有向图 G 来表示(如图 2),其中:顶点 i 表示 Web 应用 S_i 调用,有向边 (i,j) 表示 Web 应用 S_i 调用完成后调用应用 S_j,$P_{i,j}$ 为调用概率。

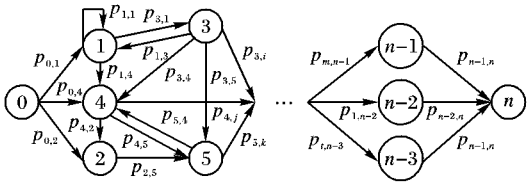

图 2　马尔可夫过程服务模型生成的有向图 G

图 2 所示的马尔可夫过程的状态迁移矩阵如式(3)所示:

$$\boldsymbol{P}=\begin{pmatrix}0 & p_{0,1} & p_{0,2} & \cdots & p_{0,n-1} & 0\\ 0 & p_{1,1} & p_{1,2} & \cdots & p_{1,n-1} & p_{1,n}\\ 0 & p_{2,1} & p_{2,2} & \cdots & p_{2,n-1} & p_{2,n}\\ \vdots & \vdots & \vdots & & \vdots & \vdots\\ 0 & p_{n-1,1} & p_{n-1,2} & \cdots & p_{n-1,n-1} & p_{n-1,n}\\ 0 & 0 & 0 & \cdots & 0 & 0\end{pmatrix}\tag{3}$$

对于无一致开始应用且无一致结束应用的组合 Web 应用,增加应用 S_0 和 S_n:应用 S_0 是一虚拟的不进行任何工作的 Web 应用,此应用仅仅用来标识开始状态;应用 S_n 也如 S_0 一样,用来标识结束状态。这样组合 Web 应用就有单一的输入和输出。由于应用 S_0 和 S_n 是虚拟的,故其生存性在任何环境下均为 1,据此有:

1) $p_{0,n}=0$。即系统开始便结束不提供任何服务的可能性是不存在的,而所有可能的应用调用序列为有向图 G 中从点 0 到点 n 的所有可能路径。

2) $\forall i(0\leqslant i\leqslant n)$,$p_{i,0}=0$,$p_{n,i}=0$。即在可能的应用调用序列中,$S_0$ 只能作为初始应用,S_n 只能作为结束应用。

3) 设 $\boldsymbol{P}=\begin{pmatrix}0 & \cdots & 0\\ \vdots & \boldsymbol{P}_{n-1,n-1}' & \vdots\\ 0 & \cdots & 0\end{pmatrix}$,则子矩阵 $\boldsymbol{P}_{n-1,n-1}'$ 中没有一行或者一列全为 0,即除了作为系统初始和结束标志的应用 S_0 和 S_n 外,其他应用(这里称为有效应用)完成后必然导致另外一个有效应用或者应用 S_n 发生,而这个应用必然由一个有效应用或应用 S_0 引发。

综合 2)、3),在有向图 G 中体现为节点 0 的入度为 0,出度不为 0;节点 n 的出度为 0,入度不为 0;其他节点的出、入度均不为 0。

4) 图 G 中除了 0、n 节点外其他节点允许有自环出现(如图 2 中节点 1),即允许应用完成后导致这个应用再次被重新调用的情况。但是,一个自环的概率为 1 则没有意义(这将导致死循环的发生),因此出度为 1 的节点不允许有自环存在,对于这样的节点 i,有 $p_{i,i}=0$。

5) 对于某非 S_0、S_n 应用 i,如果 $\zeta^{(i)}=1$,则其对于分析并无意义。为了进行简化,在状态迁移图中删去节点 i,将有向图中指向该节点的节点指向节点 i 的后继节点,转移概率为原转移概率相乘。

进一步假设应用之间的转移是可靠的,即应用间的转移过程中不会对系统的生存性造成影响,那么系统可靠地从应用 S_i 到 S_j 的迁移概率为:$p_{i,j}\cdot\zeta^{(i)}$,将其记为 $M_{i,j}$,这样可以进一步得到一步转移概率矩阵 $\boldsymbol{M}$ 和 k 步转移矩阵 $\boldsymbol{M}^k$。

进一步,系统是可生存的即为有向图 G 中存在节点 0 到 n 的路径,更进一步合成 Web 应用的生存性可由式(4)与(5)定义。

$$\zeta=s_{0,n}\tag{4}$$

其中 $s_{0,n}$ 表示矩阵

$$\boldsymbol{S}=\sum_{k=1}^{+\infty}\boldsymbol{M}^k\tag{5}$$

的第 0 行第 n 列的元素。

对于式(5)所示的正项矩阵级数，设矩阵 $\boldsymbol{M}=\begin{pmatrix}\boldsymbol{O}_{1\times1} & \boldsymbol{Q}_{1\times n}\\ \boldsymbol{O}_{n\times1} & \boldsymbol{N}_{n\times n}\end{pmatrix}$，则

$$\boldsymbol{M}^k=\begin{pmatrix}\boldsymbol{O}_{1\times1} & \boldsymbol{Q}_{1\times n}(\boldsymbol{N}_{n\times n})^{k-1}\\ \boldsymbol{O}_{n\times1} & (\boldsymbol{N}_{n\times n})^k\end{pmatrix};\ k>0 \tag{6}$$

由于 $0\leqslant\zeta^{(i)}<1(0<i<n)$，而对于矩阵 $\boldsymbol{P}$ 除最后一行全为 0 外，其他行相加均为 1，因此对于 $\forall i$ 有：

$$0\leqslant\sum_{j=0}^{n}M_{i,j}=\sum_{j=1}^{n}\zeta^{(i)}\cdot p_{i,j}=\zeta^{(i)}\cdot\sum_{j=1}^{n}p_{i,j}<1$$

而 $i=n$ 时有 $\sum_{j=0}^{n}M_{i,j}=0$，因此有

$$\|\boldsymbol{N}_{n\times n}\|_\infty=\max_{1\leqslant i\leqslant n}\sum_{j=0}^{n}|N_{i,j}|=\max_{1\leqslant i\leqslant n}\sum_{j=0}^{n}N_{i,j}<1$$

故有 $\lim\limits_{k\to+\infty}(\boldsymbol{N}_{n\times n})^k=\boldsymbol{O}_{(n+1)\times(n+1)}$，则根据式(6)有 $\lim\limits_{k\to+\infty}\boldsymbol{M}^k=\boldsymbol{O}_{(n+1)\times(n+1)}$，因此式(5)所示的正项矩阵级数收敛。

对于有同一入口或出口(或两者均有)的组合 Web 应用，则在状态迁移图中不用加入虚拟应用 S_0 或 S_n，则其如式(3)和由此生成的矩阵 $\boldsymbol{M}$ 中，与前述情况相比只是删去了第一行、第一列或者最后一行、最后一列(或者两行两列均删去)，故 $\lim\limits_{k\to+\infty}\boldsymbol{M}^k=\boldsymbol{O}$ 仍成立。下述不再区分两种情况。

进一步，由于矩阵级数收敛，则对于矩阵 $\boldsymbol{M}$ 的谱半径 $\rho(\boldsymbol{M})<1$，那么对于矩阵 $\boldsymbol{M}$ 的任意特征值 λ 有 $\lambda<1$。设 μ 为矩阵 $(\boldsymbol{E}-\boldsymbol{M})$ 的特征值($\boldsymbol{E}$ 为单位矩阵)，则有 $0=|\mu\boldsymbol{E}-(\boldsymbol{E}-\boldsymbol{M})|=|(\mu-1)\boldsymbol{E}+\boldsymbol{M}|=(-1)^{n+1}|(1-\mu)\boldsymbol{E}-\boldsymbol{M}|$，即 $1-\mu$ 是矩阵 $\boldsymbol{M}$ 的特征值。则对于矩阵 $(\boldsymbol{E}-\boldsymbol{M})$ 的任意特征值 μ，存在矩阵 $\boldsymbol{M}$ 的特征值 λ，使得 $\mu=1-\lambda$，而 $\lambda<1$，故 $\mu>0$。而 $|\boldsymbol{E}-\boldsymbol{M}|=\prod\mu^\alpha\neq0$($\alpha$ 为特征值重数)，故矩阵 $(\boldsymbol{E}-\boldsymbol{M})$ 可逆。而

$$(\boldsymbol{E}+\boldsymbol{M}+\boldsymbol{M}^2+\cdots+\boldsymbol{M}^k)=(\boldsymbol{E}-\boldsymbol{M}^{k+1})(\boldsymbol{E}-\boldsymbol{M})^{-1}$$

由于 $\lim\limits_{k\to+\infty}\boldsymbol{M}^k=\boldsymbol{O}_{(n+1)\times(n+1)}$，故

$$\boldsymbol{E}+\boldsymbol{M}+\boldsymbol{M}^2+\cdots+\boldsymbol{M}^k+\cdots=(\boldsymbol{E}-\boldsymbol{M})^{-1} \tag{7}$$

因此式(5)可化为：$\boldsymbol{S}=(\boldsymbol{E}-\boldsymbol{M})^{-1}-\boldsymbol{E}$，而根据式(4)和 $E_{0,n}=0$ 有合成 Web 应用的生存性为

$$\zeta=r_{0,n} \tag{8}$$

其中 $r_{0,n}$ 是矩阵 $\boldsymbol{R}=(\boldsymbol{E}-\boldsymbol{M})^{-1}$ 的第 0 行第 n 列的元素。

1.3 生存性恢复策略

当 Web 应用处于不利环境中时，可能会影响其某些服务，如果受影响的是非关键服务，Web 应用还可以提供有限的服务，而关键服务失效则会使得整个 Web 应用失效或者无法提供有效的服务。因此在恢复过程中应该先恢复关键服务，再恢复非关键服务，这也是生存性的要求之一。

$$\Delta_{f_k}^{(i)}:SO^{(i)}\times F\to\Lambda$$
$$(so_j^{(i)},f_k)\mapsto\Lambda_k^{i,j}=\begin{cases}\{\lambda_i\mid1\leqslant i\leqslant p\}, & f_k\text{ 对 }so_j^{(i)}\text{ 有影响}\\ \varnothing, & f_k\text{ 对 }so_j^{(i)}\text{ 无影响}\end{cases} \tag{9}$$

在 1.1 节中所述的七元组 $(S,SO^{(i)},\varphi,\Lambda,F,\Delta,\boldsymbol{E})$ 中，Λ 表示系统服务进行恢复的操作集合；F 表示在系统生存性说明中定义的一系列可能引起系统失效的事件；Δ 表示的系统服务失效后的恢复策略，即对于 Δ 定义了一个如式(9)所示的映射关系，$\{\lambda_i\mid1\leqslant i\leqslant p\}$ 中各个服务操作间是或的关系。

以下根据上述生存性模型来指导在不利环境中 Web 应用的恢复过程。假设 $f_k\in F$ 发生造成了 Web 应用失效，则以下分两步阐述恢复策略。

1.3.1 关键服务的恢复

对于 Web 应用的恢复工作首先需要对其关键服务进行恢复，以保证应用可以提供基本的核心服务。本步恢复过程如下：

1)构造待恢复关键服务集合 $R_k^c=\{so_j^{(i)}\mid\forall i,\forall j,so_j^{(i)}\in SO_c^{(i)}\text{ 且 }\Delta_{f_k}^{(i)}(so_j^{(i)},f_k)\neq\varnothing\}$，系统恢复操作集合 $RO=\varnothing$。

2)构造系统恢复操作集合 Γ_k^c，对于 $(\lambda_i,n_i)\in\Gamma_k^c$，有 $\lambda_i\in\bigcup\limits_{so_j^{(i)}\in R_k^c}\Delta_{f_k}^{(i)}(so_j^{(i)},f_k)$，$n_i$ 是恢复操作 λ_i 所影响的关键服务的计数。

3)选取具有最大 n_i 的二元组 $(\lambda_i,n_i)\in\Gamma_k^c$，执行恢复操作 λ_i。如果多个恢复操作具有相同的影响服务计数且最大，则计算这些恢复操作执行后系统在该环境下的生存性，并选取生存性最大的那个操作。事实上，这时由于系统关键服务还未恢复生存性依然为 0，这里计算生存性时假设关键服务均已恢复，这是考虑到恢复操作 λ_i 可能影响到非关键服务的操作，从而使得关键服务恢复完毕后非关键服务可以尽可能多地恢复。

4)$RO\cup=\{\lambda_i\}$，$\Gamma_k^c-=\{(\lambda_i,n_i)\}$，对于所有的 $so_j^{(i)}\in R_k^c$，如果 $\lambda_i\in\Delta_{f_k}^{(i)}(so_j^{(i)},f_k)$，$R_k^c-=\{so_j^{(i)}\}$，对于 $\forall\lambda_j\in\Delta_{f_k}^{(i)}(so_j^{(i)},f_k)$，$j\neq i$，$\Gamma$ 中的元素 (λ_j,n_j) 更新为 (λ_j,n_j-1)。如果 $n_j-1=0$，则 $\Gamma_k^c-=\{(\lambda_j,n_j)\}$。

5)如果 $R_k^c=\varnothing$，则终止本步骤，否则转到 3)。

1.3.2 非关键服务的恢复

在 Web 应用的关键服务恢复后，应用可以提供基本的服务。由于系统的生存性受到系统运行所处环境的影响，不同环境下，系统各个服务的重要程度以及用户偏好有可能不同，因此在进行非关键服务恢复时，需要考虑到系统恢复时所处的环境和系统在其他环境下的表现。

在非关键服务的恢复过程中，要综合考虑系统所处的环境和系统在各个环境下的期望，一种简单的办法是为软件在各个环境下的期望生存性与现环境下的生存性分配权重，考虑到这个权重应该与现环境持续时间有关系，故设在恢复过程中考虑的参数为 $\tau_A(t)\zeta_A^{(i)}+(1-\tau_A(t))\zeta^{(i)}$，其中 $\zeta_A^{(i)}$ 为服务 S_i 在环境 A 下的生存性，$\zeta^{(i)}$ 为服务 S_i 在各种环境下的期望生存性，$\tau_A(t)\in[0,1]$ 为环境 A 的权重，其具体数值和系统处于环境 A 下的持续时间有关。这样，对于上述两种特殊情况，简单地令 $\tau_A(t)=0$ 或 1 即可。

基于以上假设，对于系统非关键服务的恢复可以依照以下步骤进行：

1)构造待恢复的服务集合 $R_k^i=\{so_j^{(i)}\mid\forall i,\forall j,so_j^{(i)}\notin SO_c^{(i)}\text{ 且 }\forall\lambda_i\in RO,\lambda_i\notin\Delta_{f_k}^{(i)}(so_j^{(i)},f_k)\}$，以及恢复操作集合 $\Gamma_k^i=\bigcup\limits_{so^{(i)}\in R_k^i}\Delta_{f_k}^{(i)}(so_j^{(i)},f_k)$。

2)对于所有 $\lambda_i\in\Gamma_k^i$，计算采用恢复操作 λ_i 后的生存性指数 $\tau_A(t)\zeta_A^{(i)}+(1-\tau_A(t))\zeta^{(i)}$，其中 A 为系统现在所处的环境。选取具有最高生存性指数的恢复操作 λ_i。

3)对于 $\forall so^{(i)}\in R_k^i,\lambda_i\in\Delta_{f_k}^{(i)}(so_j^{(i)},f_k)$，$R_k^i-=\{so^{(i)}\}$，$RO\cup=\{\lambda_i\}$，$\Gamma_k^i-=\{\lambda_i\}$。

4)如果 $\Gamma_k^i=\varnothing$ 或者 $R_k^i=\varnothing$ 则终止本步骤，否则转到 2)。

基于以上两步恢复，系统可以在遭遇到不利因素影响时，保证可以满足用户基本需求的前提下，尽可能地为用户提供更加全面、可靠的服务。

2 案例分析

假设具有如图 3 所示的某合成 Web 应用，其中 Web 应用 S_s 和 S_e 分别表示开始和结束状态，该迁移图的迁移矩阵 $\boldsymbol{P}$ 如

式(10)所示。

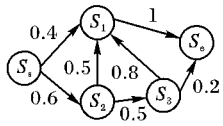

图 3　Web 应用状态迁移图

$$\boldsymbol{P}=\begin{pmatrix}0 & 0.4 & 0.6 & 0 & 0\\ 0 & 0 & 0 & 0 & 1\\ 0 & 0.5 & 0 & 0.5 & 0\\ 0 & 0.8 & 0 & 0 & 0.2\\ 0 & 0 & 0 & 0 & 0\end{pmatrix} \tag{10}$$

其中环境 $E=\{e_1,e_2,e_3\}$,在系统运行过程中,每种环境出现的概率为{0.2,0.6,0.2},系统在运行过程中可能受到的不利影响有 $F=\{f_1,f_2,f_3\}$,系统修复集合为 $\Lambda=\{\lambda_1,\lambda_2,\cdots,\lambda_9\}$。下面对于 f_1 发生时进行分析。原子 Web 应用 S_1、S_2、S_3 所包含的服务、权重以及 f_1 发生时的修复策略如表 1 ~ 3 所示。

表 1　服务 S_1 的权重映射和恢复策略

S_1	权重映射 $\varphi^1_{e_i}$			修复策略 $\Delta^{(i)}_{f_1}$
	e_1	e_2	e_3	
$so_1^{(1)}$	+∞	+∞	+∞	λ_2
$so_2^{(1)}$	1	2	5	λ_2, λ_3
$so_3^{(1)}$	3	1	2	λ_2, λ_3
$so_4^{(1)}$	4	4	4	λ_2, λ_3, λ_6
$so_5^{(1)}$	2	5	1	λ_6
$so_6^{(1)}$	+∞	+∞	+∞	—
$so_7^{(1)}$	5	3	3	λ_3

表 2　服务 S_2 的权重映射和恢复策略

S_2	权重映射 $\varphi^2_{e_i}$			修复策略 $\Delta^{(i)}_{f_1}$
	e_1	e_2	e_3	
$so_1^{(2)}$	+∞	+∞	+∞	λ_2,λ_3
$so_2^{(2)}$	1	3	2	λ_2, λ_3
$so_3^{(2)}$	2	1	1	λ_2, λ_3
$so_4^{(2)}$	+∞	+∞	+∞	—
$so_5^{(2)}$	3	2	3	λ_3

表 3　服务 S_3 的权重映射和恢复策略

S_3	权重映射 $\varphi^3_{e_i}$			修复策略 $\Delta^{(i)}_{f_1}$
	e_1	e_2	e_3	
$so_1^{(3)}$	+∞	+∞	+∞	λ_2, λ_3
$so_2^{(3)}$	+∞	+∞	+∞	λ_2, λ_3
$so_3^{(3)}$	4	2	3	λ_2, λ_3
$so_4^{(3)}$	1	3	4	λ_2, λ_3
$so_5^{(3)}$	+∞	+∞	+∞	—
$so_6^{(3)}$	3	4	2	—
$so_7^{(3)}$	2	1	1	λ_3

以下对在环境 e_1 下,f_1 发生时的恢复过程进行分析。

1)应用关键服务的恢复。产生实效的应用关键服务集合:$R^c_k=\{so_1^{(1)},so_1^{(2)},so_1^{(3)},so_2^{(3)}\}$,恢复操作:$\Gamma^c_k=\{(\lambda_2,4),(\lambda_3,3)\}$,故选取恢复操作 λ_2,即 $RO=\{\lambda_2\}$,此后 $R^c_k=\varnothing$,关键服务恢复完毕。

2)应用非关键服务的恢复。假设 $\tau_{e_1}(t)=(4-t)/4$,进行完每一步恢复操作后 t 加 1,所有恢复操作均耗时 1 个时间单位,初始 $t=1$。

待恢复的服务:$R^i_k=\{so_5^{(1)},so_7^{(1)},so_5^{(2)},so_7^{(3)}\}$,恢复操作集合 $\Gamma^i_k=\{\lambda_3,\lambda_6\}$,如果采用恢复操作 λ_3,则恢复后的生存性指数为:0.291 067,如果采用恢复操作 λ_6,则为 0.265 779,因此采用恢复操作 λ_3,此时 $R^i_k=\{so_5^{(1)}\}$,$\Gamma^i_k=\{\lambda_6\}$,进一步采用恢复操作 λ_6 后 $R^i_k=\varnothing$,恢复完成。

3　结语

本文对 Web 应用进行了研究,并在文献[11]工作的基础上分别针对原子 Web 应用和合成 Web 应用建立了生存性模型,在模型建立过程中,考虑了环境和应用调用关系对生存性的影响,并在这个模型基础上提出了 Web 应用在面临不利环境时的恢复策略。最后以一个案例对模型的应用进行了分析,并给出了恢复过程,根据这个恢复方案,Web 应用得以在恢复过程中保证较高的生存性。在未来的工作中,将进一步对环境对 Web 应用的影响以及高可生存性的 Web 应用的构建框架进行研究。

参考文献:

[1] ALONSO G, CASATI F, KUNO H, *et al*. Web services: concepts, architectures and applications [M]. Berlin: Springer-Verlag, 2004.

[2] MA JIANG, CHEN HAO-PENG. A reliability evaluation framework on composite Web service [C]// SOSE '08: Proceedings of the 2008 IEEE International Symposium on Service-Oriented System Engineering. Washington, DC: IEEE Computer Society, 2008: 123 - 128.

[3] KNIGHT J C, STRUNK E A. Achieving critical system survivability through software architectures [C]// ICSE 2003: Workshop on Software Architectures for Dependable Systems, LNCS 3069. Berlin: Springer-Verlag, 2004: 51 - 78.

[4] 张永,方滨兴,包秀国. 网络可生存性研究概述[J]. 计算机工程与应用, 2005, 41(7): 119 - 121.

[5] HILTUNEN M A, SCHLICHTING R D, UGARTE C A, *et al*. Survivability through customization and adaptability: the Cactus approach [C]// DISCEX '00: Proceedings of DARPA Information Survivability Conference and Exposition. Washington, DC: IEEE Computer Society, 2000, 1: 294 - 307.

[6] KNIGHT J C, STRUNK E A, SULLIVAN K J. Towards a rigorous definition of information system survivability [C]// Proceedings of DARPA Information Survivability Conference and Exposition. Piscataway: IEEE, 2003, 1: 78 - 89.

[7] ELLISON R J, FISHER D A, LINGER R C, *et al*. Survivability: protecting your critical systems [J]. IEEE Internet Computing, 1999, 3(6): 55 - 63.

[8] CALDERA J. Survivability requirements for the US health care industry [D]. Pittsburgh: Carnegie Mellon University, 2000.

[9] ELLISON R J, FISHER D A, LINGER R C, *et al*. Survivable network systems: an emerging discipline [R]. Pittsburgh: Carnegie Mellon University, Software Engineering Institute, 1997.

[10] BYON I. Survivability of the U. S. electric power industry [D]. Pittsburgh: Carnegie Mellon University, 2000.

[11] AYARA A, NAJJAR F. A formal specification model of survivability for pervasive systems [C]// ISPA 2008: IEEE International Symposium on Parallel and Distributed Processing with Applications. Piscataway: IEEE, 2008: 444 - 451.

[12] XIA Q, WANG Z. Survivability recovery of information system based on component availability [C]// Cyber Technology in Automation, Control, and Intelligent Systems. Piscataway: IEEE, 2011: 220 - 225.

[13] CHEN BIHUA, PENG XIN, YU YIJUN, *et al*. Survivability-oriented self-tuning of Web systems [C]// WWW '11: Proceedings of the 20th International Conference Companion on World Wide Web. New York: ACM, 2011: 23 - 24.

[14] ZHOU DONGYUN, SUBRAMANIAM S. Survivability in optical networks [J]. IEEE Network, 2000, 14(6): 16 - 23.

[15] ZHAO SHI, LU XIAOMING, ZHOU XIANZHONG, *et al*. A reliability model for Web services from the consumers' perspective [C]// International Conference on Computer Science and Service System. Piscataway: IEEE, 2011: 91 - 94.

[16] RAN S. A model for Web services discovery with QoS [J]. ACM SIGecom Exchanges, 2003, 4(1): 1 - 10.