1997年度日本オペレーションズ・リサーチ学会
春季研究発表会

ハードウェア／ソフトウェアシステムの性能評価のための アベイラビリティモデル

01307475　鳥取大学　*得能貢一　TOKUNO Koichi
01702425　鳥取大学　山田茂　YAMADA Shigeru

1　はじめに

最近，大規模かつ複雑なコンピュータシステムを，ハードウェアおよびソフトウェアのトレードオフを考慮して最適設計を行う，**ハードウェア／ソフトウェア協調設計（コデザイン）**(hardware/software co-design) [1] という概念が注目されるようになった．この概念は目新しい概念ではないが，現在のコデザインの研究はシステム設計を定量的な方法に基づいて組織的に行う手法の確立を目指しており，システムの信頼性／性能を計測・評価する上でもコデザインの概念が重要となってきた．

本研究では，一つのハードウェアサブシステムと一つのソフトウェアサブシステムから構成されるハードウェア／ソフトウェアシステム（以下システムと略す）のアベイラビリティ評価モデルを構築する．このとき，ソフトウェアサブシステムは修復作業によって信頼性が向上することを考慮に入れる．動作状態および不動作状態を交互に繰り返すシステムの時間的挙動をマルコフ過程 (Markov process [2]) を用いて記述する．また，本モデルに基づいて，システムの信頼性／性能評価尺度を導出する．最後に，本モデルの数値例を示す．

2　モデルの記述

時刻 t におけるシステムの状態を表す確率過程 $\{X(t),\ t \geq 0\}$ を考える．確率過程 $\{X(t),\ t \geq 0\}$ の状態空間を以下のように定義する．

W_n: システムは正常に稼働している．
R_n^{S}: ソフトウェア故障が発生し，システムは修復状態にある．
R_n^{H}: ハードウェア故障が発生し，システムは修復状態にある．

ここで，$n = 0,\ 1,\ 2,\ \cdots$は完全に修正されたフォールト数の累積値を表す．また，モデルを記述するに当たり以下の仮定を設ける．

A1. ハードウェア故障あるいはソフトウェア故障が発生したらシステムはダウンし，すぐに修復作業に入る．修復作業が完了するまでシステムは稼働しない．

A2. ソフトウェアに対する修復作業にはデバッグ作業が含まれる．デバッグ作業が実施されると，高々一つのフォールトが修正される．各フォールトは確率 a $(0 < a \leq 1)$ で確実かつ完全に修正され，確率 $b(= 1-a)$ でフォールトは修正されない（すなわち，不完全デバッグとなる）．確率 a を完全デバッグ率と呼ぶことにする．

A3. ハードウェア故障とソフトウェア故障は，それぞれ独立であるものとする．ソフトウェアに対する故障発生時間および修復時間は，それぞれ平均 $1/\lambda_n$および$1/\mu_n$の指数分布に従う．また，ハードウェアに対する故障発生時間および修復時間は，それぞれ平均$1/\theta$および$1/\eta$の指数分布に従う．

A4. 2個以上のハードウェア故障あるいはソフトウェア故障が同時に発生することはない．

A2より，任意の時刻 t で $\{X(t) = R_n^{\mathrm{S}}\}$ のとき，ソフトウェアに対する修復作業が完了すると，

$$X(t) = \begin{cases} W_n & (\text{確率 } b) \\ W_{n+1} & (\text{確率 } a), \end{cases} \tag{1}$$

となる．また，ソフトウェア故障発生時間に対するハザードレートを次のように表す [3].

$$\lambda_n = Dk^n$$
$$(n = 0,\ 1,\ 2,\ \cdots;\ D > 0,\ 0 < k < 1). \tag{2}$$

ここで，Dおよびkは，それぞれ初期ハザードレートおよびハザードレートの減少率を表す．式 (2) は，システム運用の初期段階ではソフトウェア故障の発生頻度は高く，その後次第に減少していくようなソフトウェア故障発生現象を記述している．次に，ソフトウェアの修復時間について言及する．一般に，後で発見されるフォールトほど複雑度が高くなると言われる．すなわち，後で発生するソフトウェア故障の原因であるフォールトを認知したりフォールト修正作業を実施するのにより多くの時間が費やされる．したがって，除去されたフォールト数の累積値が大きくなると，ソフトウェアの平均修復時間も大きくなると考えるのが適当であろう．上述のことを考慮して，μ_nを次のように仮定する．

$$\mu_n = Er^n$$
$$(n = 0,\ 1,\ 2,\ \cdots;\ E > 0,\ 0 < r < 1). \tag{3}$$

ここで，Eおよびrは，それぞれ初期ソフトウェア修復率およびソフトウェア修復率の減少係数を意味する．マルコフ過程を形成する $X(t)$ の状態遷移図を図1に示す．

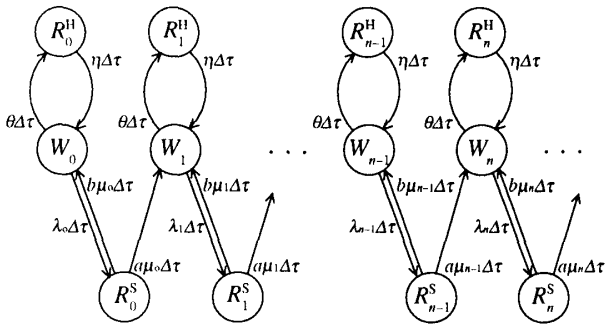

図1　$X(t)$ の状態遷移図

3　システムの性能評価尺度

n 個のフォールトを修正するのに要する時間 S_n の分布関数は，

$$G_n(t) \equiv \Pr\{S_n \le t\} = 1 - \sum_{i=0}^{n-1}(A^1_{n,i}e^{-x_i t} + A^2_{n,i}e^{-y_i t} + A^3_{n,i}e^{-z_i t}) \quad (t \ge 0;\ n = 0,\ 1,\ 2,\ \cdots), \tag{4}$$

で与えられる．ここで，$-x_i$，$-y_i$ および $-z_i$ は，以下の s に関する3次方程式

$$s^3 + (\theta + \eta + \lambda_i + \mu_i)s^2 + (\theta\mu_i + \eta\lambda_i + \eta\mu_i + a\lambda_i\mu_i)s + a\eta\lambda_i\mu_i = 0, \tag{5}$$

の解を表す．また，係数 $A^1_{n,i}$，$A^2_{n,i}$ および $A^3_{n,i}$ は，それぞれ

$$A^1_{n,i} = \frac{[a(\eta - x_i)]^n \prod_{j=0}^{n-1}\lambda_j\mu_j}{x_i \prod_{\substack{j=0\\ j\ne i}}^{n-1}(x_j - x_i)\prod_{j=0}^{n-1}(y_j - x_i)(z_j - x_i)} \quad (i = 0,\ 1,\ 2,\ \cdots,\ n-1), \tag{6}$$

$$A^2_{n,i} = \frac{[a(\eta - y_i)]^n \prod_{j=0}^{n-1}\lambda_j\mu_j}{y_i \prod_{\substack{j=0\\ j\ne i}}^{n-1}(y_j - y_i)\prod_{j=0}^{n-1}(z_j - y_i)(x_j - y_i)} \quad (i = 0,\ 1,\ 2,\ \cdots,\ n-1), \tag{7}$$

$$A^3_{n,i} = \frac{[a(\eta - z_i)]^n \prod_{j=0}^{n-1}\lambda_j\mu_j}{z_i \prod_{\substack{j=0\\ j\ne i}}^{n-1}(z_j - z_i)\prod_{j=0}^{n-1}(x_j - z_i)(y_j - z_i)} \quad (i = 0,\ 1,\ 2,\ \cdots,\ n-1), \tag{8}$$

となる．

また，時刻 t でシステムが正常に稼働している確率を表す**瞬間アベイラビリティ** (instantaneous availability, [4] 参照) は，

$$A(t) = \sum_{n=0}^{\infty}\left(\frac{g_{n+1}(t)}{a\lambda_n} + \frac{g'_{n+1}(t)}{a\lambda_n\mu_n}\right), \tag{9}$$

となる．ここで，$g_n(t)$ は S_n の確率密度関数を表し，$g'_n(t) \equiv \mathrm{d}g_n(t)/\mathrm{d}t$ である．

4　数値例

完全デバッグ率 a と式 (9) のシステムの瞬間アベイラビリティ $A(t)$ の関係を図2に示す．この図より，稼働開始直後にシステムアベイラビリティは低下しており，システム性能が不安定であることが定量的に示される．また，完全デバッグ率が高いほど，すなわちソフトウェアに対する修復作業の完全性が高いほどシステム全体の可用性を向上させることがわかる．

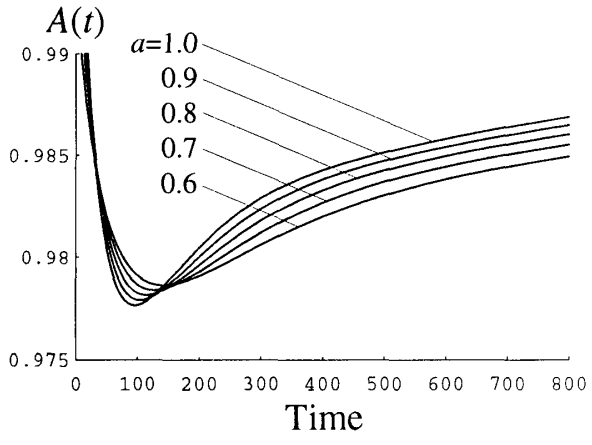

図2　完全デバッグ率 a と $A(t)$ の関係 ($\theta = 0.01$, $\eta = 1.0$, $D = 0.01$, $k = 0.8$, $E = 0.5$, $r = 0.9$)

参考文献

[1] G. De Micheli, "ハードウェア／ソフトウェア協調設計のコンピュータ支援における問題および方法について", 情報処理, Vol. 36, No. 7, pp. 605–613, 1995年7月.

[2] S. M. Ross, "*Stochastic Processes*", John Wiley & Sons, New York, 1983.

[3] P. B. Moranda, "Event-altered rate models for general reliability analysis", *IEEE Trans. Reliability*, Vol. R-28, No. 5, pp. 376–381, 1979.

[4] K. Tokuno and S. Yamada, "A Markovian software availability measurement with a geometrically decreasing failure-occurrence rate", *IEICE Trans. Fundamentals*, Vol. E78-A, No. 6, pp. 737–741, 1995.