2003年日本オペレーションズ・リサーチ学会
春季研究発表会

アプレットの実行過程における信頼性解析

01014373　愛知学泉大学 *今泉　充啓　IMAIZUMI Mitsuhiro
岐阜市立女子短期大学　木村　充位　KIMURA Mitsutaka
01701193　愛知工業大学　安井　一民　YASUI Kazumi

1. はじめに

最近，WWW (World Wide Web) 上でのアプリケーション構築を実現する技術の一つとして，アプレット (Applet) が注目されている．アプレットは，WWWサーバからクライアントにダウンロードし，クライアントのセキュリティ保全を行いながら実行することを可能としたプログラムコードである．

アプレットはサーバからロードされるため，不良コードによりシステム故障を引き起こすことがある．この問題に対処するため，従来から様々な方策が提案されてきた[1]．その一つにアプレットによるシステム資源へのアクセス（システムコール）をセキュリティチェッカにより監視する方策がある．セキュリティチェッカは，ユーザが指定したセキュリティ管理方針に基づいてシステムコールの検査を行い，必要に応じてプログラムコード実行の制限を行うことができるモジュールである[2]．システムコールの内容をチェックするセキュリティ検査は，検査プログラムを実行させることにより行われる．このセキュリティ検査により，不良コードによるシステム故障は未然に防ぐことができるが，この検査によるオーバヘッドは無視できない問題となっている．

ここでは，アプレット実行過程における信頼性モデルを設定し，アプレット処理成功までの期待費用を最小にする最適方策を議論する．

2. モデル

(1) アプレットがサーバからクライアントにロードされたとき，JIT(Just-In-Time) コンパイラはクライアントにダウンロードされ，これによりアプレットはオブジェクトコードに変換され実行が行われる．アプレットのロードに要する時間は一定時間分布 $D(t)$ (平均 d) に従う．もし，サーバからの応答がない場合は，一定時間待ち再びアプレットのロードを要求する．この待ち時間は，一定時間分布 $W(t)$ (平均 w) に従う．また，アプレットの実行処理に要する時間は指数分布 $A(t)$ (平均 $1/a$) に従うものとする．

(2) オブジェクトコードから行われるシステム資源へのアクセス（システムコール）はセキュリティチェッカにより監視する．システムコールは指数分布 $B(t)$ (平均 $1/b$) に従い発生する．

(3) セキュリティチェッカは，システムコール発行ごとに検査が必要なシステムコールかどうかを判断する．検査は確率 p で行われ，この場合検査プログラムを実行する．検査プログラム実行に要する時間は一般分布 $U(t)$ (平均 u) に従う．

(4) セキュリティチェッカによる検査の結果，システムコールは確率 q で許可される．この場合，システムコールを再開してオペレーティングシステムに制御を渡す．システムコール処理に要する時間は，一般分布 $V(t)$ (平均 v) に従う．

(5) 逆に確率 $1-q$ で不許可の場合は，エラーコードをアプレットに返し，アプレットの実行を中断する．中断後，不良コードの除去を行いアプレットの実行を再開する．中断後，再開に要する時間は一定時間分布 $G(t)$ (平均 μ) に従う．

(6) セキュリティチェッカにより，検査を行わなかった場合は，指数分布 $F(t)$ (平均 $1/\lambda$) に従ってユーザーシステムへの不良アクセスによりシステム故障が発生する．この場合，保全を行い初期状態からやり直す．この時間は，一定時間分布 $Z(t)$ (平均 z) に従う．

以上の仮定のもとで，システムの各状態を次のように定義する．

状態 0: アプレットロード開始.

状態 1: アプレット実行開始.

状態 2: システムコール発生.

状態 3: アプレット実行中断，不良コード除去開始.

状態 F: 不良アクセスによるシステム故障発生.

状態 S: アプレット処理成功.

状態 E: アプレットロード待ち開始.

システムの状態を上のように定義すると，各状態は状態 S を吸収状態にもつマルコフ再生過程を形成する．

マルコフ再生過程における 1 ステップ推移確率時間分布を $Q_{ij}(t)(i=0,1,2,3,E,F; j=0,1,2,3,E,F,S)$ とし，そのラプラス・スチルチェス (LS) 変換を $q_{ij}(s)$ とすると，次式を得る．

$$q_{01}(s) = P_{00}(d)e^{-sd}, \quad (1)$$

$$q_{0E}(s) = P_{01}(d)e^{-sd}, \quad (2)$$

$$q_{EE}(s) = P_{11}(w)e^{-sw}, \quad (3)$$

$$q_{E1}(s) = P_{10}(w)e^{-sw}, \quad (4)$$

$$q_{12}(s) = \frac{b}{s+a+b}, \quad (5)$$

$$q_{1S}(s) = \frac{a}{s+a+b}, \quad (6)$$

$$q_{21}(s) = pqu(s)v(s) + (1-p)v(s+\lambda), \quad (7)$$

$$q_{23}(s) = p(1-q)u(s), \quad (8)$$

$$q_{2F}(s) = (1-p)\frac{\lambda}{s+\lambda}[1-v(s+\lambda)], \quad (9)$$

$$q_{31}(s) = g(s), \quad (10)$$

$$q_{F0}(s) = z(s). \quad (11)$$

ここで，$P_{ij}(t)$ はサーバが時刻 0 で状態 $i(i=0,1)$ から出発し，時刻 t で状態 $j(j=0,1)$ にある確率を表す．ただし，状態 0 は正常状態，状態 1 はビジー状態である．

次に，アプレット処理成功までの平均故障回数 M_F を求めよう．故障回数分布 $M_F(t)$ の LS 変換 $m_F(s)$ は次式で与えられる．

$$m_F(s) = \{q_{01}(s)+q_{0E}(s)\sum_{i=1}^{\infty}[q_{EE}(s)]^{i-1}q_{E1}(s)\} \sum_{j=1}^{\infty}\{q_{12}(s)[q_{21}(s)+q_{23}(s)q_{31}(s)]\}^{j-1} q_{12}(s)q_{2F}(s)[1+q_{F0}(s)m_F(s)]. \quad (12)$$

よって，平均故障回数は，

$$M_F \equiv \lim_{s\to 0} m_F(s) = \frac{a}{b}(1-p)[1-v(\lambda)], \quad (13)$$

として求められる．

同様に，アプレット処理開始から処理成功またはシステム故障までの平均処理中断回数 M_3 は，

$$M_3 = \frac{bp(1-p)}{a+b(1-p)[1-v(\lambda)]}, \quad (14)$$

となる．

3. 最適方策

ここでは，経済性を考慮して，最適方策を議論する．システム故障に伴う損失費用を c_1，中断に伴う損失費用を $c_2(<c_1)$，システムの通常的な運用に伴う固定費用を $c_3(<c_2)$ とし，アプレット処理成功までの期待費用 $C(p)$ を次のように定義する．

$$C(p) \equiv c_1 M_F + c_2 M_3 + c_3 = c_1\frac{b}{a}(1-p)[1-v(\lambda)] + c_2\frac{bp(1-q)}{a+b(1-p)[1-v(\lambda)]} + c_3. \quad (15)$$

このとき，$C(p)$ を最小にする最適なセキュリティ検査確率 p^* を求める．$C'(p)=0$ とおくと，

$$\frac{a(1-q)\{a+b(1-p)[1-v(\lambda)]\}}{[1-v(\lambda)]\{a+b(1-p)[1-v(\lambda)]\}^2} = \frac{c_1}{c_2}, \quad (16)$$

を得る．ここで，式 (16) の左辺を $L(p)$ とおくと，次式を得る．

$$L'(p) = \frac{2ab(1-q)\{a+b[1-v(\lambda)]\}}{\{a+b(1-p)[1-v(\lambda)]\}^3} > 0, \quad (17)$$

$$L(0) = \frac{a(1-q)}{[1-v(\lambda)]\{a+b[1-v(\lambda)]\}}, \quad (18)$$

$$L(1) = \frac{(1-q)\{a+b[1-v(\lambda)]\}}{a[1-v(\lambda)]}. \quad (19)$$

よって，$L(p)$ は $L(0)$ から $L(1)$ までの p の単調増加関数となる．以上から，次のような結論を得ることができる．

(i) もし，$L(0) \le c_1/c_2 \le L(1)$，すなわち，$a(1-q)/[1-v(\lambda)]\{a+b[1-v(\lambda)]\} \le c_1/c_2 \le (1-q)\{a+b[1-v(\lambda)]\}/a[1-v(\lambda)]$ ならば，式 (16) を満たす有限で唯一の $(0<)p^*(<1)$ が存在する．

(ii) もし，$L(0) > c_1/c_2$，すなわち，$c_1/c_2 < a(1-q)/[1-v(\lambda)]\{a+b[1-v(\lambda)]\}$ ならば，$p^*=0$ である．

(iii) もし，$L(1) < c_1/c_2$，すなわち，$c_1/c_2 > (1-q)\{a+b[1-v(\lambda)]\}/a[1-v(\lambda)]$ ならば，$p^*=1$ である．

参考文献

1 大山恵弘，加藤和彦，"SecurePot: システムコールフックを利用した安全なソフトウェア実行系"，日本ソフトウェア科学会第18回大会論文集，2001.

2 板橋一正，松原克弥，森山豊，染谷祐一，加藤和彦，関口龍郎，米澤明憲，"仮想機械独立なアプレットシステムの実現"，信学論(D-I), Vol.J84-D-I, No.6, pp.639–649, June 2001.