基于任务模块的实时软件可靠性模型*

雷 航** 熊光泽 刘锦德
(电子科技大学微机所 成都 610054)

【摘要】 根据实时多任软件的特征和各任务所占系统时间的非均匀分布,以任务模块为基本测试单元,提出一种实时多任务软件可靠性评价模型。由于任务周期、任务切换方式、切换频率以及任务故障率等基本参数易于分析和测试,因此该模型具有良好的可实现性和可扩充性,为实时多任务软件的可靠性评价开辟了一条新途径。

关 键 词 可靠性; 实时系统; 任务切换; 故障率; 评价模型
中图分类号 TP311.1

日趋复杂的软件系统对整个计算机系统可靠性的影响已变得十分突出,因此,软件可靠性评价已受到人们广泛重视。国内外已提出 100多种软件可靠性评价模型,所有这些模型可以分为随机过程模型和非随机过程模型两大类。前者以 J-M模型[1]、G-O模型[2]和 Musa执行时间模型[3]为代表,后者则以 L-V模型[4]、Seeding模型[5]等为代表。由于没有形成成熟的软件可靠性理论,对各种不同模型的优劣存在很大争议,但是有一点已达成共识,即可靠性模型中所需的各种参数要易于分析和测试,对假设参数要尽可能合理,否则即使模型在理论上作得十分完善,但因所需参数不易得到而失去现实意义。

现有的软件可靠性模型基本都建立在如下假设基础上,即故障率正比于软件中的剩余故障数。这一假设的局限性在于:如果一个模块中有较少的剩余故障参数但被系统频繁调用,则由此表出较高的故障率;而另一模块中有较多的剩余故障数但很少被系统调用,表现出较低的故障率。现有模型的另一弱点是故障数据测试是在软件设计完成并组成系统以后进行,因此软件的扩充、版本升级等都会导致对整个软件系统的重新测试。

实时软件的应用范围多为工业控制等单位和系统,其可靠性评价尤为重要。但现有的各类软件可靠性评价模型基本上不针对实时软件。目前实时软件已向多任务、模块化方向发展,以便于系统测试、集成和扩充。早在 70年代末期,软件可靠性模型的开拓者之一 B. Littlewood就提出了按结构化、模块化的方式进行软件可靠性建模的思想[6]。但对于非实时系统,模块之间的转换方式和转换频率以及模块间的接口故障较难确定,使这一建模思想未得到进一步发展。

基于上述原因,本文根据 FCFS并结合任务优先级静态调度算法,以任务模块为基本测试单元,提出一种实时多任务软件可靠性评价模型,用以估测系统故障率。

1 模型的建立

对一个实时多任务系统,假设有 n 个周期任务 $T_1, T_2, \cdots, T_n$,周期分别为 $t_1, t_2, \cdots, t_n$,每一任

1996年 3月 22日收稿
* 国防科工委"九五"重点科研项目
** 男 35岁 博士生 讲师

务的执行时间分别为 $C_1, C_2, \cdots, C_n$，n 个实时任务执行时，任务之间的切换按照 FCFS结合任务优先级来进行。优先级的确定采用短周期任务优先。软件故障发生在每一个任务的执行过程中和从一个任务到另一个任务的切换过程中，前者的故障数据测试可在系统集成之前进行，后者则在系统集成之后。

设：
- 每个任务模块中故障的发生是泊松过程；
- 每个任务有各自的故障率；
- 系统任一时刻只运行一个任务；
- 系统执行按 FCFS并结合优先级调度算法在任务之间切换；
- 一个切换的任务对之间有各自的切换故障率。

记：
R_i 为任务 T_i 的执行占系统运行的时间比例；
$E_i(t)$为任务 T_i 的故障率；
S_j为任务 T_i 切换到任务 T_j 的次数；
B_{ij}是 T_i 切换到 T_j 的频率；
$F_i(t)$为任务 T_i 切换到任务 T_j 的切换故障率；
$P(t)$为系统故障率。

根据上述假设，项 $R_i E_i(t)$表示在系统运行时间 $[0,t]$内任务 T_i 的故障率，和式$\sum_{i=1}^{n} R_i E_i(t)$ 表示系统中所有任务在运行时间 $[0,t]$内的故障率（不包括切换时的故障）。项 $B_{ij} F_{ij}(t)$表示系统运行时间 $[0,t]$内任务 T_i 切换到任务 T_j 的切换故障率，则和式$\sum_{i=1}^{n}\sum_{j=1}^{n} B_{ij} F_{ij}(t)$表示在系统运行时间 $[0,i]$内所有任务的切换故障率，由此可得系统故障率为

$$P(t) = \sum_{i=1}^{n} R_i E_i(t) + \sum_{i=1}^{n}\sum_{j=1}^{n} B_{ij} F_{ij}(t)$$

其中 $B_{kk} = 0 \quad (1 \leqslant k \leqslant n)$。

将系统故障率分为任务执行故障率和任务切换故障率两部分，使模型中第一项参数的分析或测试可在系统集成之前进行，使错误可以得到尽早修正，并且单个任务模块的故障测试和分析比整个软件系统的测试和分析要容易得多。

2 模型参数的确定

2.1 参数 R_i

R_i 是 T_i 的执行所占系统总运行时间的比例。设系统从时刻 0运行到 $t(t \gg \max(t_1, t_2, \cdots, t_n))$时刻，则有

$$R_i = \frac{|t/t_i| C_i}{t}$$

其中 $|t/t_i|$表示在 $[0,t]$内，T_i 执行的次数，C_i 是 T_i 的执行时间。

2.2 参数 E_i

参数 E_i 的确定在系统集成之前通过对任务 T_i 的测试和分析计算得到。设 $m_i(t)$表示在测试区间 $[0,t]$中的期望错误数，a_i 是最终查出的期望错误个数，b_i 是到时刻 t时 T_i 中每个错误的错误查出率，在 $(t, t+\Delta t)$中错误出现的个数与剩余故障成正比，并设边界条件 $m_i(0) = 0, t, m_i(\infty) = a_i$，则有

$$m_i(t+\Delta t) - m_i(t) = b_i(a_i - m_i(t))\Delta t$$

令 $\Delta t \to 0$,得

$$m'_i(t) = a_i b_i - b_i m_i(t)$$

$$m_i(t) = \exp\left(\int_0^t -b_i \mathrm{d}t\right)\left[\int_0^t a_i b_i \exp\left(-\int_0^t -b_i \mathrm{d}t\right)\mathrm{d}t + C\right]$$

利用边界条件 $m_i(t) = 0$可得

$$m_i(t) = a_i[1 - \exp(-b_i t)]$$

到时刻 t的剩余故障数的期望值为

$$E_i(t) = b_i E\{\tilde{N}_i(t)\} = a_i b_i \exp(-b_i t)$$

对未知参数 a_i和 b_i的估计,假设在时刻 $T_k(k = 1, 2, \cdots, N)$在 T_i中查出累积软件故障数为 x_k,则关于 a_i和 b_i的似然函数是

$$P_r\{N(t_1) = x_1, N(t_2) = x_2, \cdots, N(t_N) = x_N)\} =$$

$$\prod_{k=1} \frac{\{a_i[\exp(-b_i t_{k-1}) - \exp(-b_i t_k)]\}^{x_k - x_{k-1}}}{(x_k - x_{k-1})!} \exp\{-a_i[1 - \exp(-b_i t_N)]\}$$

对上式的对数似然函数中参数 a_i和 b_i求偏导,并令

$$\frac{\ln P_r}{a_i} = \frac{\ln P_r}{b_i} = 0$$

可得

$$a_i[1 - \exp(-b_i t_N)] = X_N$$

$$a_i t_N \exp(-b_i t_N) = \sum_{k=1} \frac{(x_k - x_{k-1})[t_k \exp(-b_i t_k) - t_{k-1}\exp(-b_i t_{k-1})]}{t\exp(-b_i t_{k-1}) - \exp(-b_i t_k)]}$$

采用 Newton-Raphson迭代算法[7]求解上述方程,即可得 a_i和 b_i的估计 $\hat{a}_i$和 $\hat{b}_i$,可将故障率函数写成为

$$E_i(t) = \hat{a}_i \hat{b}_i \exp(-\hat{b}_i)$$

2.3 参数 S_{ij}

当任务周期 $t_1, t_2, \cdots, t_n$和任务执行时间 $C_1, C_2, \cdots, C_n$以及任务调度算法确定之后,可以利用这些先验知识计算 S_{ij}。在此提出一种计算 S_{ij}的算法,并计算出 B_{ij}。

设 PR_i是 T_i的优先级,将 n个任务按周期长度从小到大排列,设排列为 $T_1, T_2, \cdots, T_n$,于是当 $i < j$时,$PR_i > PR_j$。该算法的思想是从任务 T_1开始,计算 $(S_{11}, S_{12}, \cdots, S_{1n})$,然后计算 $(S_{21}, S_{22}, \cdots, S_{2n})$直到 $(S_{n1}, S_{n2}, \cdots, S_{nn})$。任务 T_i执行完成后,要切换到的任务是从 T_i到达时刻起,其余 $n-1$个任务中最早到达的任务。若多个任务同时到达则切换到优先级最高的任务。设系统运行时间范围 $[0, t]$内,算法如下:

STEP1. (i) (j) $(S_{ij} \leftarrow 0)$, $(1 \leqslant i \leqslant n, 1 \leqslant j \leqslant n)$

STEP2. $t \leftarrow 1$,

STEP3. $j \leftarrow 1, k \leftarrow 1$,

STEP4. *IF* $l_{tj} \geqslant t$ THEN STEP14 ELSE STEP5.

STEP5. $K_j \leftarrow 1 l(t_i / t_j)\rfloor$,

STEP6. $j \leftarrow j + 1$, IF $j \leqslant n$ THEN STEP5 ELSE STEP7.

STEP7. $j \leftarrow 1$,

STEP8. $U_j \leftarrow (K_j + 1)t_j - l_{ti}$,

STEP9. $j \leftarrow j + 1$, *IF* $j \leqslant n$ THEN STEP8 ELSE STEP10.

STEP10. $U_{\min} \leftarrow \min(U_1, U_2, \cdots, U_n)$.

STEP11. $j \leftarrow 1$,

STEP12. IF($U_{min} = U_j$) $\wedge$ ($i \leqslant j$) THEN $S_{ij} \leftarrow S_{ij} + 1$, GOTO STEP13
ELSE $j \leftarrow j + 1$, GOTO STEP12.

STEP13. $k \leftarrow k + 1$, GOTO STEP4.

STEP14. $i \leftarrow i + 1$, IF $i > n$ THEN STEP15 ELSE STEP3.

STEP15. $\forall (i) \forall (j) B_{ij} = S_j \Big/ \left(\sum_{i=1}^{n} \sum_{k=1}^{n} S_{ij} \right)$.

上述算法中,K_j 表示当 T_i 到达时 T_j 已到达的次数。在 STEP8 STEP9 STEP10计算并选择 T_i 之后下一次最早到达的任务,U_j 表示 T_i 到达时刻距下一个 T_j 到达之间的时间,因此,U_{min}是最早到达的任务。由于多个任务可能同时到达,使($U_1, U_2, \cdots, U_n$)中多项等于 U_{min},所以在 STEP12选择并切换到优先级最高的任务,但不能切换到与 T_i 同时到达且优先级更高的任务,因为该任务已于 T_i 之前执行完成,所以在 STEP12中还必须满足条件 $i \leqslant j$。

2.4 参数 $F_{ij}(t)$

$F_{ij}(t)$参数的确定应在系统集成之后,根据多任务软件开发和调试的知识,任务 T_i 切换到任务 T_j 的接口故障可能会传播到从 T_j 到 T_i 的切换,即关系到 T_i 切换到 T_j 的故障可能会使 T_j 到 T_i 的切换发生故障,反之变然。因此 $F_{ij}(t)$正比于 $T_i \rightarrow T_j$ 接口上的剩余故障数,同时也正比于 $T_j \rightarrow T_i$ 接口上的剩余故障数乘以一个比例常数。设 $N_{ij}(t)$和 $m_{ij}(t)$分别表示在区间 $[0, t]$内 $T_i \rightarrow T_j$ 接口上的累积错误数和期望数,A_{ij}和 A_{ji}是最终查出的期望错误数,D_{ij}和 D_{ji}是在时刻 t 每个错误查出率,Q_j和 Q_i是比例常数。并设边界条件 $m_{ij}(0) = m_{ji}(0) = 0, m_{ji}(\infty) = A_{ij}, m_{ji}(\infty) = A_{ji}$。根据上的错误面的描述和假设可得

$$m_{ij}(t + \Delta t) - m_{ij}(t) D_{ij}(A_{ij} - m_{ij}(t))\Delta t + Q_{ij} D_{ji}(A_{ji} - m_{ji}(t))\Delta t m_{ji}(t + \Delta t) - m_{ji}(t) = D_{ji}(A_{ji} - m_{ji}(t))\Delta t + Q_{ji} D_{ij}(A_{ij} - m_{ij}(t))\Delta t$$

令 $\Delta t \rightarrow 0$,可得方程

$$m'_{ij}(t) = A_{ij} D_{ij} m'_{ij}(t) + A_{ji} D_{ji} Q_j - D_{ji} Q_{ij} m_{ji}(t)$$
$$m'_{ji}(t) = A_{ji} D_{ji} m'_{ji}(t) + A_{ij} D_{ij} Q_{ji} - D_{ij} Q_{ji} m_{ij}(t)$$

上述方程属常系数非齐次线性微分方程组,将方程组写成如下形式

$$M'(t) = AM(t) + f$$

其中

$$A = \begin{bmatrix} -D_{ij} & -D_{ji} Q_j \\ -D_{ij} Q_{ji} & -D_{ji} \end{bmatrix} \qquad f = \begin{bmatrix} A_{ij} D_{ij} + A_{ji} D_{ji} Q_{ij} \\ A_{ji} D_{ji} + A_{ij} D_{ij} Q_{ji} \end{bmatrix}$$

则

$$M(t) = H(t) H^{-1}(0) M(0) + H(t) \int H^{-1}(s) f \, ds$$

其中

$$H(t) = \begin{bmatrix} \exp(r_1, t) u_{11} & \exp(r_2 t) u_{21} \\ \exp(r_2, t) u_{12} & \exp(r_2 t) u_{22} \end{bmatrix} \qquad M(0) = \begin{bmatrix} M_{ij}(0) \\ M_{ji}(0) \end{bmatrix}$$

式中 r_i 是 A 的特征根;u_{ij}是对应于特征根的特征向量。

求解得均值函数 $m_{ij}(t)$和 $m_{ji}(t)$后,根据

$$F_{ij}(t) = D_{ij}(A_{ij} - m_{ij}(t)) + Q_{ij} D_{ji}(A_{ji} - m_{ji}(t))$$

可得 $T_i \rightarrow T_j$ 的接口故障率函数。对 $F_{ij}(t)$中未知参数的确定,可利用已查出的累积错误数和关于 A_{ij}、D_{ij}、A_{ji}、D_{ji}、Q_j和 Q_i的似然函数求出上述 6个参数的估计值。至此,已确定出模型 $P(t)$中的全部参数。

3 结　论

不同的任务在系统运行时间内所占的时间比例是非均匀的,从系统的角度,单个任务所表现出的故障率除了与该任务的剩余故障数有关外,还与该任务所占系统运行时间有关。因此用时间比例和切换频率作故障率系数更适合多任务、多模块的软件结构。另外,参数 R_i 和 S_i 的确立独立于故障数据测试,参数 $E_i(t)$ 可在系统集成之前单独测试并确定,删减或增加任务模块都不需要对整个软件系统进行重新测试,因此该模型具有良好的可扩充性和可实现性。

参　考　文　献

1 Jelinski Z, Moranda P B. Statistical computer performance evaluation. New York: Academic press, 1972

2 Goel A L, Okumoto K. Time dependent error detection rate model for software reliability and other performance measures. IEEE Trans Reliability, 1979, (3): 206~211

3 Musa J D. A theory of software reliability and its application. IEEE Trans Software Engineering, 1975, SE-1, (3): 312~327

4 Littlewood B, Verrall J L. A bayesian relibility growth model for computer software. Applied statistics, 1973, 22: 332~346

5 Sandoh H, Fujii S. Reliability growth analysis for discrete type software by quansi-error seeding. Osaki S, Jao J H Eds Reliability Theory and Applications, Singapore: World Scientific, 1987: 319~327

6 Littlewood B Software reliability model for modular programm structure. IEEE Trans Reliability, 1979(8), R-28, (3): 241~246

7 Lawless J F. Statistical model and methods for lifetime date. New York: John Wiley and Sons, 1982

Real-time Software Reliability Model Based on Task Modular

Lei Hang　　Xiong Guangze　　Liu Jinde

(Research Institute of Microcomputers, UEST of China　Chengdu　610054)

Abstract　According to the feature of real-time multi-tasks software systems and nonhomogeneous distributed that different task consume system time, taking task modular as basic testing unit, this paper presents a evaluation model of real-time multi-task software systems. Since the parameters, for example task period, task switch method and switch frequence are easy to analyse and test, the model possesses good extensibility and feasibility. This modeling method provides a new way for the reliability evaluation of real-time multi-task software.

Key words　reliability;　real-time system;　task switch;　fault rate;　evaluation model

编辑　徐安玉