1996 年 5 月　　重庆大学学报 （自然科学版）　　Vol. 19, №. 3
第 19 卷第 3 期　　Journal of Chongqing University (Natural Science Edition)　　May. 1996

(10) 61-66

一种 PMC 模型下的自适应顺序诊断算法*

A New Adaptive Sequential Diagnosis Algorithm in the PMC Model

TP306.3

汪雪琴**　　周六丁　　杨晓帆
Wang Xueqin　　Zhou Liuding　　Yang Xiaofan
（重庆大学计算机研究所，重庆，630044；第一作者 25 岁，女，硕士）

摘　要　在 PMC 模型下提出一种能实现顺序诊断目标的自适应诊断算法。其特点是可以直接诊断出一个故障处理机，而不需要先寻找一个无故障处理机。

关键词　故障诊断；诊断算法；多处理机系统
中国图书资料分类法分类号　TP306.3

PMC 模型　自适应诊断，计算机

ABSTRACT　We present an adaptive algorithm for sequential diagnosis in the PMC model, which can directly identify a faulty processor without first identifying a fault-free processor.

KEYWORDS　fault diapnosis; diagnosis algorithm; multiprocessor system

0 引　言

随着多处理机系统规模的日益增大，如何才能有效地对系统中可能发生的故障进行定位，这已成为一个重要的研究课题。系统级故障诊断[1]的基本思想是：让系统中的处理机互相测试，然后通过对测试结果的分析确定出现故障的处理机。

按照测试关系的确定程度，可以将系统级诊断分成两大类：非自适应诊断[1~3]和自适应诊断[4~6]。所谓非自适应诊断，就是全部测试是预先确定的，但测试量很大[1~3]。自适应诊断则是将测试分成若干步来 完成，并且后面 测试是根据前面测试的结果来选定的，其优点是测试量较小[5]。

按照要求定位的故障处理机数目，又可以将系统级诊断分成一步诊断和顺序诊断。一步诊断要求诊断出全部故障处理机，而顺序诊断只要求识别一个故障处理机。顺序诊断的优点是诊断开销小。

Hakimi 和 Nakajima[6]提出了一个自适应顺序诊断算法，其思想是先找出一个无故障处理机，然后用它去确定一个故障处理机。笔者在 PM(模型下提出了一个新的自适应顺序诊断算法。与 Hakimi-Nakajima 算法相比较，我们的算法有许多优点。

* 收文日期　1995-03-18
** 作者现在广东机械学院工作

1 基本概念

PMC 模型[1]是系统级故障诊断理论中的一种经典模型。在 PMC 模型中，多处理机系统 S 中处理机之间的测试关系用一个有向图 $G(V,A)$表示，图中的顶点代表处理机，从一个顶点指向另一个顶点的弧表示前一个处理机对后者的测试。若前者认为后者是正常的，则将测试结果记为 0；否则将测试结果记为 1. PMC 假定只有当测试单元是正常的时候，测试结果才是可信的。如果用"+"表示正常的处理机，"−"表示故障处理机，则 PMC 模型可简洁刻划如下：

图 $G(V,A)$ 又称为系统 S 的测试图。而 S 的拓扑结构可以用无向图 $S(V_S,E_S)$ 来刻划(顶点对应于处理机，边对应于处理机间的通信链路)。$G(V,A)$ 是 $S(V_S,E_S)$ 的定向子图。

2 ADA 算法

在 PMC 模型下，Hakimi 和 Nakajima[2]证明对于由 n 个处理机组成的系统($n\geqslant 2t+1$)，在顺序诊断目标下，至多需要 n 次测试可诊断出一个故障处理机。Hakimi 的自适应算法可分为两步：

1) 找到一个无故障处理机 u；

2) 从 u 出发，诊断出一个故障处理机。

能否不用寻找一个正常处理机而直接诊断出故障处理机？而这样所需的测试数是否会增多？下面笔者提出的 ADA 算法回答了这一问题。ADA 算法出是一种自适应算法，但它可以直接诊断出一个故障处理机。

在描述算法之前，首先介绍一个算法中用到的关键技术。

1) 如果出现图 1 的情形，那么可以断定 v_0 是故障处理机。

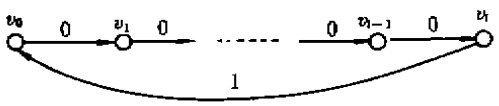

图 1 可以断定有向图上有一个故障处理机

若 v_0 是正常的，那么根据 PMC 模型的假定，$v_1,v_2,\cdots,v_l$ 都是正常的。而 v_l 对 v_0 的测试值为 1，即 v_l 认为 v_0 是故障的，矛盾。

2) 如果出现图 2 的情形，可以断定所有 $v_i(i=0,1,\cdots,l)$ 要么都是无故障的，要么都是

故障的。

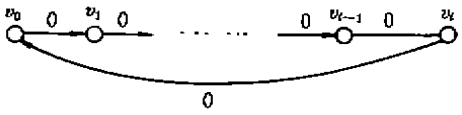

图 2　可以断定有向圈上所有处理机状态相同

v_0 只有两种状态，要么是正常的，要么是故障的。若 v_0 是正常的，显然 0 链上的所有处理机也都是正常的；若 v_0 是故障的，而 v_t 对 v_0 的测试值为 0，那么 v_t 一定是故障的，依次类推，v_{t-1}，…，v_1 也是故障的。

ADA 算法的基本思想是：任意选择一个处理机 u_0，让 u_0 去测试任一其它处理机 v。若测试结果为 0，则把 v 作为测试者，继续往下测试，若结果还是 0，重复刚才的步骤，直到出现测试结果为 1 时测试才暂时停止。在这个过程中，把 u_0 以及所有被测试为正常(0) 的处理机放入集合 V_0 中；把被测试为故障的处理机放入集合 V_1 中。让最后一个被测试为 0 的处理机去测试 u_0，若测试结果为 1，说明 u_0 是故障的(见图 1)，算法停止；若测试结果为 0，说明 V_0 中的所有元素都是无故障的，或都是故障的(见图 2)，之后，又重新从 u_0 开始去测试未被测试的处理机，重复上面的过程。$|V|$ 表示集合 V 中的元素数目。当在某次测试后，满足 $|V_1| \geqslant t+1$，说明 u_0 一定是故障的，算法停止。若 $|V_0| \geqslant t+1$ 时，就不再往下测试，而让最后一个被测试为 0 的处理机，去测试 u_0，若结果为 1，说明 u_0 是故障的；若结果为 0，说明 u_0 是无故障的(否则会有多于 t 个的处理机出故障)，如果这时 $|V_1| \neq 0$，那么 V_1 中的所有元素都是故障的；如果 $|V_1| = 0$，那么 u_0 至多再测试 $n-t-1$ 次，就可以找到一个发生故障的处理机。该算法可以形式地描述如下：

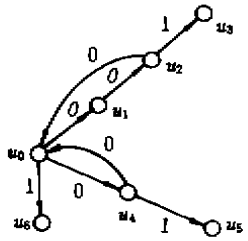

图 3　ADA 算法示例

$V_0 = \{u_0, u_1, u_2, u_4\}$

$V_1 = \{u_3, u_5, u_6\}$

PROCEDURE ADA (V：处理机集合，t：正整数)

{V 是多处理机系统的所有处理机的集合，t 是系统中发生故障的处理机数目的上限，系统应满足 $|V| \geqslant 2t+1$}

VAR

V_0, V_1：处理机子集合

BEGIN

$V_0 := \varnothing; V_1 := \varnothing;$

任选一处理机 $u_0 \in V$;

$V_0 := V_0 \cup \{u_0\}; V := V - \{u_0\};$

$u := u_0;$

```
        REPEAT
            任选一处理机      v ∈ V;V1 := V - {v};
            IF σ(u,v) = 1 THEN
                BEGIN
                    V1 := V1 ∪ {v};
                    IF |V1| = t + 1 THEN u0 出故障,算法停止
                    ELSE IF u<>u0THEN
                    BEGIN
                        u 测试 u0;
                        IF σ(u,u0) = 1 THEN u0 出故障,算法停止
                        ELSE u := u0
                    END
                END;
            ELSE BEGIN
                    V0 = V0 ∪ {v};
                    IF |V0| = t + 1 THEN
                        BEGIN
                            v 测试 u0;
                            IF σ(v,u0) = 1 THEN u0 出故障,算法停止;
                            ELSE test
                        END
                    END
            UNTIL |V| = 0
END


PROCEDURE σ(u,v)
        BEGIN
            u 对 v 进行测试;
            测试结果为 σ(u,v)
        END
PROCEDURE test
    BEGIN
        IF |V1|<>0 THEN V1 中的所有处理机出故障,算法停止
        ELSE BEGIN
            WHILE |V|<>0 DO
                BEGIN
                    任选 v0 ∈ V;
                    IF σ(u0,v0) = 1 THEN v0 出故障,算法停止
```

END;
打印无处理机出故障，算法停止
END
END

3　算法的正确性

定理　对于一个由 n 个处理机组成的多处理机系统 s（其发生故障的处理机数目不超过 t 且 $n \geqslant 2t+1$），在 PMC 模型下，ADA 算法至多进行 $\max\{n, 3t\}$ 次测试和至多考察 n 个处理机就会诊断出一个故障处理机。

证明　在 ADA 算法中，一旦能够诊断出一个故障处理机，算法立即停止。而只要满足下述条件之一就会使 ADA 算法停止：

1）对 u_0 的测试结果为 1；

2）$|V_1| \geqslant t+1$；

3）$|V_0| \geqslant t+1$.

若满足 1）或 2），已经可以诊断出至少一个故障处理机 u_0，算法立即停止。

在满足 3）的情形下，按算法要对 u_0 进行测试，若测试结果满足 1），算法就停止；若测试结果为 0，那么 V_0 中所有元素都是无故障的。这样，当 $|V_1| \neq 0$ 时，可以断定 V_1 中所有元素都是故障的；当 $|V_1| = 0$ 时，u_0 至多还要对未被测试过的处理机逐一施加测试，就能诊断出一个故障处理机，算法也会停止。

下面证明 ADA 算法一定会满足上述三个条件之一。

假设对 u_0 的测试值都为 0，而 $|V_1| \leqslant t$ 且 $|V_0| \leqslant t$. 这时考察过的处理机数目为 $|V_1| + |V_0| \leqslant 2t$. 而 $n \geqslant 2t+1$，所以 $n > |V_1| + |V_0|$. 这就是说，只要 1）、2）和 3）皆不满足，那么总是存在没有被考察过处理机，因而算法还可以继续进行。每一个被考察的处理机在考察后，在 V_1 中，或在 V_0 中。故每考察一个处理机，$|V_1| + |V_0|$ 就要增加 1，这样，即使不出现 1）的情形，2）或 3）总是会被满足的。不难分析，在最坏情形下，ADA 算法需考察全部处理机。（如图 4）

下面分析算法至多需要的测试数目。

在图 4 这种特殊情形下，算法至多需要 n 次测试。

除此情形外，按照算法的设计，每考察一个处理机，就要用到一次测试，而算法结束时所用测试的总数目为所考察到的处理机数与对 u_0 的回测次数之和。如果考察到的处理机最多且对 u_0 的测试数目也最多，则所需的测试数目也最多。易见最坏情形是图 5 的情形，所需的测试数为 $3(t-1)+1+2=3t$. 这时 $|V_0| + |V_1| = 2t+1$，考察到的处理机最多，且对 u_0 的测试数目最多。

所以，ADA 算法至多需要 $\max\{n, 3t\}$ 次测试。

4　结束语

ADA 算法在理论上解决了能否直接诊断出一个故障处理机的问题，而且 ADA 算法至多

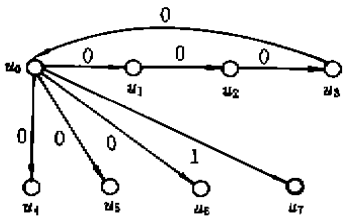

图 4 $|V_1|=0$ 且 $|V_0|=t+1$ 的情形

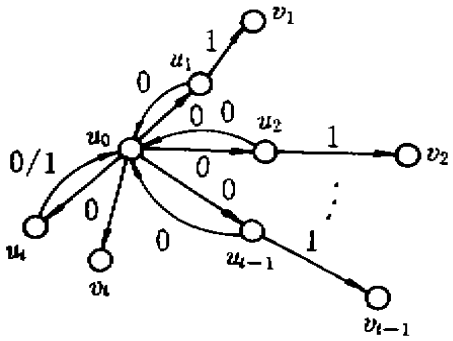

图 5 所需测试数最多的情形

$n=3$

$t=3$

$V_1=\varnothing$

$V_0=\{u_0,u_1,u_2,u_3\}$

需 $\max\{n,3t\}$ 次测试，只要 $n\geqslant 3t$，ADA 算法不比 Hakimi 和 Nakajima 算法在最坏情形下所需的测试数目多。(在实际系统中，允许发生故障的处理机数目不会很大，所以 $n\geqslant 3t$ 这个条件是合理的。) 此外，ADA 算法还有下面的优点：

1) ADA 算法往往不仅仅找到一个故障处理机，而是找到故障处理机的子集。见图 5，当 u_i 对 u_0 的测试结果为 0 时，$v_1,v_2,\cdots,v_t$ 全是故障处理机。

2) ADA 算法最好情形下只需要 3 次测试($t>1,t=1$ 时只需要 2 次)。而 Hakimi 和 Nakajima 的算法需 t 次[2].

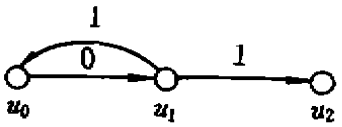

图 6 只需 3 次测试的情形结果：u_0 出故障

在自适应诊断中，由于是自适应地选择测试，所以可以利用较少的测试数目实现诊断目标。

参 考 文 献

1 Preparata F P, Metze G, Chien R T. On the connection assignment problem of diagnosable systems, IEEE Trans. Eloctron, Comput., 1967, c-16, 848～854

2 Dahbura A T, Masson G M. An $O(n^{2.5})$ fault identification algorithm for diagnosable systems, IEEE Trans. Comput., 1984, c-33, 486～492

3 Chen Tinghuai. Fault Diagnosis and Fault Tolerance-A Systematic Approach to Special Topics, Berlin: Springer-Verlag, 1992

4 Hakimi S L, Nakajima K. On adaptave system diagnosis, IEEE Trans, Comput., 1984, c-33, 234～240

5 杨晓帆，蔡兵，曹泽翰，陈廷槐．两个改进的自适应诊断算法及适用范围．第六届全国容错计算会议论文集(杨孝宗主编)，黑龙江教育出版社，1995，262～266

6 蔡兵，杨晓帆，陈廷槐．对称比较模型下的系统级故障诊断，计算机学报，1995，18，858～866