Uma Nova Estratégia para o Diagnóstico de Falhas Baseado em Comparações

Roverli Pereira Ziwich, Elias Procópio Duarte Jr.

Universidade Federal do Paraná, Departamento de Informática
Caixa Postal 19018 – 81531-990, Curitiba, PR – Brasil
Fone: +55-41-3361-3656, Fax: +55-41-3361-3205

{roverli,elias}@inf.ufpr.br

Resumo. *O diagnóstico baseado em comparações determina o estado das unidades do sistema a partir da comparação do resultados de tarefas produzidos por pares de unidades. Qualquer diferença na comparação indica que uma ou ambas as unidades estão falhas. O diagnóstico completo do sistema é baseado no resultado de todas as comparações. Este trabalho descreve uma nova solução para a identificação de unidades falhas em sistemas complexos. Estes sistemas são dos mais diversos tipos, incluindo hardware, software, além de redes de interconexão de computadores e processadores. Os algoritmos existentes para a realização do diagnóstico baseado em comparações em sistemas gerais envolvem, em alguma fase, técnicas de diagnóstico tradicional. A estratégia proposta neste trabalho resolve o diagnóstico utilizando exclusivamente a síndrome de comparações, sem necessidade de convertê-la para outro modelo. Um esboço da prova de corretude do algoritmo proposto também é apresentado.*

Abstract. *Comparison-based diagnosis is based on the comparison of tasks outputs produced by pairs of units. If the comparison results in a mismatch one or both units are faulty. The complete system diagnosis is performed based on the results of all comparisons. This work describes a new strategy to identify faulty units in hardware and software based systems. Previously published comparison-based diagnosis algorithms for general systems involve, at some phase, some technique inherited from traditional diagnosis models and algorithms. The strategy proposed in this work is based on the comparison syndrome alone and does not require any conversion to other models. An outline of the algorithm correctness proof is also presented.*

1. Introdução

Os sistemas computacionais estão cada vez maiores e mais complexos: processadores *many-core* já combinam dezenas de núcleos em um único chip, redes contemplam dezenas de milhares de unidades, softwares executam de forma distribuída em diversas máquinas [1, 2, 3] . É muito provável que unidades destes sistemas irão, em algum momento, funcionar de forma anormal, produzindo resultados diferente do esperado. Se a falha de uma unidade causar a falha completa do sistema, usuários podem ser fortemente prejudicados. Com isso, a preocupação com a monitoração de sistemas, visando a detecção de ataques, violações ou simplesmente comportamento anormal, têm crescido constantemente.

O diagnóstico baseado em comparações é uma forma realista para o diagnóstico de falhas em redes e sistemas distribuídos. Este paradigma de diagnóstico foi proposto há vários anos e diversas aplicações já foram apresentadas. Este trabalho propõe a utilização do framework de diagnóstico baseado em comparações para a identificação eficiente de unidades falhas em sistemas distribuídos compostos de múltiplos processadores.

O primeiro modelo de diagnóstico em nível de sistema foi o modelo PMC (Preparata, Metze, e Chien) [4]. O objetivo do diagnóstico é identificar o estado das unidades do sistema: determinando quais estão falhas e quais estão sem-falha [5]. No modelo PMC, as unidades realizam testes sobre as outras unidades e comunicam resultados de testes visando obter o diagnóstico completo do sistema. Neste modelo, um teste envolve a aplicação controlada de estímulos e a observação da resposta correspondente. Já o diagnóstico baseado em comparações [6, 7] utiliza a comparação do resultado de tarefas produzidos por pares de unidades do sistema. Qualquer diferença na comparação indica que uma ou ambas as unidades estão falhas.

Nos modelos de diagnóstico baseado em comparações assume-se que em um sistema de N unidades, a comparação dos resultados produzidos pela execução de tarefas de unidades é possível. O modelo MM – proposto por Maeng e Malek [8] assume que os resultados das comparações são enviados a um observador central que realiza o diagnóstico completo do sistema, mas permite que as comparações das saídas das tarefas sejam feitas pelas próprias unidades. A única restrição é que a unidade que realiza a comparação deve ser diferente das duas unidades que executam as tarefas. Maeng e Malek também apresentam um caso especial do modelo MM, chamado MM*, no qual uma unidade testadora executa comparações para todas as suas unidades vizinhas. Sengupta e Dahbura em [9] propuseram uma generalização do modelo MM onde a própria unidade testadora pode ser uma das unidades que são comparadas. Sengupta e Dahbura em [9] também apresentaram as condições necessárias para um sistema ser diagnosticável com base no modelo MM.

Alguns algoritmos de diagnóstico foram construídos com o objetivo de determinar quais são as unidades falhas a partir da síndrome do sistema. Sengupta e Dahbura apresentaram um algoritmo de diagnóstico de ordem de complexidade $O(N^5)$ para identificar as unidades falhas baseado no modelo MM*, onde N é o número de unidades do sistema. Além disso, Sengupta e Dahbura mostraram que a diagnosticabilidade de sistemas gerais sobre este modelo é NP-completa. Mais recentemente, um algoritmo de diagnóstico de ordem de complexidade $O(N \times \Delta^3 \times \delta)$ – onde Δ e δ são respectivamente o máximo e o mínimo grau de uma unidade do sistema – foi apresentado por Yang e Tang em [10].

Este trabalho apresenta uma nova solução para o diagnóstico baseado em comparações baseado no modelo MM*. O algoritmo apresentado é uma alternativa aos algoritmos de diagnóstico para o modelo MM* apresentados por Sengupta e Dahbura [9] e por Yang e Tang [10]. Ambos os algoritmos existentes para a realização do diagnóstico baseado em comparações envolvem técnicas do diagnóstico tradicional baseado no modelo PMC. O algoritmo de [10] faz uma conversão da síndrome de comparações para uma síndrome do modelo PMC; o algoritmo de [9] também aplica na sua última fase uma técnica de diagnóstico baseado no modelo PMC. A estratégia proposta neste trabalho resolve o diagnóstico utilizando exclusivamente a síndrome de comparações, sem necessidade de conversão para outro modelo. Além da especificação e descrição do algoritmo, um esboço das provas de corretude é apresentado neste trabalho.

O restante deste trabalho está organizado da seguinte forma. A seção 2 apresenta os modelos MM e MM* de diagnóstico baseado em comparações. Na sequência, a seção 3 descreve algoritmos existentes para o diagnóstico baseado em comparações. Em seguida, a seção 4 apresenta o novo algoritmo proposto para o modelo MM*. Um esboço da prova de corretude do algoritmo é apresentada na seção 5. Por fim, a seção 6 conclui o trabalho.

2. Os Modelos MM e MM* de Diagnóstico Baseado em Comparações

O modelo MM de diagnóstico baseado em comparações foi proposto por Maeng e Malek [8] para sistemas compostos de múltiplos processadores. O sistema é representado por um grafo $G = (V, E)$, onde V representa o conjunto de unidades e E representa o conjunto de links de comunicação. No modelo MM, o estado das unidades é determinado através da comparação da saída de uma tarefa executada por uma unidade com a saída da mesma tarefa executa por outra unidade. O conjunto de todas as comparações realizadas no sistema é representado por C. O modelo MM permite que a comparação das saídas das tarefas seja realizada pelas próprias unidades do sistema, isto é, as unidades são também comparadoras. Uma unidade k é uma unidade comparadora das unidades i e j somente se $(k, i) \in E$ e $(k, j) \in E$; além disso $k \neq i$ e $k \neq j$. Os resultados das comparações são também enviados a um observador central que realiza o diagnóstico completo do sistema. O conjunto com todos os resultados das comparações é chamado de síndrome do sistema – e representado por σ. Todos os possíveis resultados de uma comparação é mostrado na tabela 1.

Comparador	Unidade 1	Unidade 2	Resultado da Comparação
sem-falha	sem-falha	sem-falha	0 (igualdade)
sem-falha	sem-falha	falha	1 (diferença)
sem-falha	falha	sem-falha	1 (diferença)
sem-falha	falha	falha	1 (diferença)
falha	sem-falha	sem-falha	0 ou 1
falha	sem-falha	falha	0 ou 1
falha	falha	sem-falha	0 ou 1
falha	falha	falha	0 ou 1

Tabela 1. Possíveis resultados de comparações para o modelo MM.

A notação $r((i, j)_k)$ é usada para representar o resultado da comparação das saídas das unidades i e j pela unidade k. O resultado é 0 quando a comparação indicar igualdade e o resultado é 1 quando a comparação indicar diferença. Se $r((i, j)_k) = 1$, pelo menos uma das unidades i, j ou k está falha. Se $r((i, j)_k) = 0$ e a unidade testadora k é sem-falha então i e j são também sem-falha. Mas se a unidade testadora k é falha, o resultado da comparação não é confiável e não se pode obter nenhuma conclusão sobre o estado das unidades i e j.

As principais asserções do modelo MM são:

— Toda falha é permanente, isto é, as unidades não se recuperam das falhas;
— A comparação realizada por qualquer unidade falha não é confiável;
— Duas unidades falhas que executam a mesma tarefa sempre retornam saídas diferentes;
— Toda unidade falha sempre gera resultados incorretos para cada tarefa de entrada, isto é, a comparação das saídas de tarefas pro-

duzidas por uma unidade falha e qualquer outra unidade (falha ou sem-falha) sempre resulta em diferença;

— Existe um limite t, que é o número máximo de unidades que podem estar falhas para o diagnóstico do sistema ainda ser possível.

Maeng e Malek também apresentaram um caso especial do modelo MM, chamado de modelo MM*. A diferença é que no modelo MM* cada unidade executa comparações para todas as unidades vizinhas às quais está conectada.

Considere um sistema t-diagnosticável, isto é, um sistema no qual o diagnóstico é possível se a quantidade de unidades falhas for menor ou igual a t. Além de apresentar o modelo MM, Maeng e Malek também mostram em [8] que para o diagnóstico de uma falha ser possível, ou seja, $t = 1$, o número total de unidades do sistema (N) deve ser maior que 3. Para $t \geq 2$, N deve ser maior ou igual a $2t + 1$. As condições necessárias e suficientes para o diagnóstico de t unidades ser possível neste sistema são: (1) o grau de toda unidade no sistema deve ser maior ou igual a t; (2) todo par de unidades $\{i, j\}$ cuja distância é 1 ou 2, um número maior ou igual a t de unidades de um conjunto chamado $W_{i,j}$ devem ser removidas para que o par de unidades e seus vizinhos sejam desconectados do restante do grafo; além disso (3) não existe $W^*_{i,j}$ se $W_{i,j}$ que possui exatamente t unidades. $W^*_{i,j}$ é definido como um $W_{i,j}$ que possui pelo menos um par de vértices r, s no qual $W_{i,r} = (W_{i,j} - r) \cup j$ e $W_{j,s} = (W_{i,j} - s) \cup i$.

Na sequência, Sengupta e Dahbura em [9] caracterizaram a t-diagnosticabilidade do sistema para o modelo MM. Seja S_1 e S_2 conjuntos de unidades. Um par (S_1, S_2) tal que $S_1, S_2 \subset V$ e $|S_1|, |S_2| \leq t$ é definido como *distinguível* ou *indistinguível* como segue. Seja $\sigma(F)$ um conjunto de síndromes que pode ser produzido se F é o conjunto de unidades falhas. O par de conjuntos $S_1, S_2 \mid S_1 \neq S_2$ é dito indistinguível se e somente se $\sigma(S_1) \cap \sigma(S_2) \neq \emptyset$; caso contrário o par é distinguível.

Sengupta e Dahbura [9] também provam que todo sistema S com N unidades é t-diagnosticável se e somente se para todo conjunto $S_1, S_2 \in V \mid S_1 \neq S_2$ e $|S_1|, |S_2| \leq t$, (S_1, S_2) é um par distinguível. Em outras palavras, considerando o conjunto $\sigma(S_1)$, que é conjunto de síndromes que pode ser produzido se S_1 é o conjunto de unidades falhas, e considerando analogamente o conjunto $\sigma(S_2)$, $\sigma(S_1) \cap \sigma(S_2) = \emptyset$.

Os autores também provam que para um sistema com N unidades ser t-diagnosticável , $N \geq 2t + 1$ e cada unidade possui ordem maior ou igual a t, isto é, a saída do teste de cada unidade deve ser comparado com a saída de pelo menos outras $t - 1$ unidades.

3. Algoritmos de Diagnóstico de Tempo Polinomial

Dois algoritmos de diagnóstico para sistemas gerais com base no modelo MM* foram propostos na literatura. Sengupta e Dahbura em [9] apresentaram um algoritmo de tempo polinomial para identificar falhas em processadores onde os processadores executam comparações para todo par de vizinhos. Mais importante ainda, eles mostram que a diagnosticabilidade de sistemas gerais sobre este modelo é NP-completa. Recentemente, um algoritmo de diagnóstico de ordem de complexidade $O(N \times \Delta^3 \times \delta)$ – onde Δ e δ são respectivamente o máximo e o mínimo grau de uma unidade do sistema – foi apresentado por Yang e Tang em [10] também para o modelo MM*. Esta sessão apresenta uma breve descrição destes dois algoritmos de diagnóstico.

3.1. O Algoritmo de Sengupta e Dahbura

O algoritmo de diagnóstico apresentado por Sengupta e Dahbura – chamado *DIAGNOSIS* – é mostrado na figura 1. O algoritmo recebe como entrada o conjunto de todas as comparações, isto é, a síndrome do sistema (σ). Algumas definições são necessárias para entender o algoritmo e são apresentadas abaixo.

```
Algorithm DIAGNOSIS

/* Initialization Phase */
F ← ∅;
compute S(σ);

for each node ∈ S(σ) such that |N(i)| = t + 1 do
   /* First Step */
   for each k ∈ N(i) do
      if N(i) − {k} is an AFS (Allowable Fault Set) then
         F ← N(i) − k;
         stop the algorithm;
      end if
   end for
end for

for each node ∈ S(σ) such that |N(i)| = t do
   /* Second Step */
   for each k ∈ N(i) do
      if N(i) is an AFS then
         F ← N(i);
         stop the algorithm;
      end if
   end for

   /* Third Step */
   compute H(s);
   for each k ∈ N(i) do
      for each h ∈ H(σ) do
         if N(i) − k + h is a vertex cover of hypergraph Z = (V, H(σ)) then
            F ← N(i) − k + h;
            stop the algorithm;
         end if
      end for
   end for
end for

/* Fourth (and Final) Step */
begin
   construct graph Y = (V, M(σ));
   remove all self-loops in Y;
   compute the maximum matching of Y;
   F ← the minimum vertex cover set Y;
end
```

Figura 1. O algoritmo *DIAGNOSIS* proposto por Sengupta e Dahbura.

Dado um grafo $G' = (V', E')$, $K \subseteq V'$ é um *vertex cover set* de G' se toda aresta em E' for incidente a ao menos um vértice em K. Um vertex cover set de menor cardinalidade é chamado de um vertex cover set mínimo. $N(i) = \{j \mid (i, j) \in E\}$ é o conjunto de vizinhos da unidade i. Um conjunto $U \subset V$ é chamado de AFS (*Allowable Fault Set* – possível conjunto de unidades falhas) do sistema S para a síndrome σ, se para quaisquer três unidades i, j, k tal que $(i, j)_k$ tal que $(i, j)_k \in C$:

— se $k \in V - U$ e $i, j \in V - U$ então $r((i, j)_k) = 0$
— se $k \in V - U$ e $\{i, j\} \cap U \neq \emptyset$ então $r((i, j)_k) = 1$

Na fase de inicialização, o conjunto de unidades falhas (F) é atribuído como vazio, e $S(\sigma)$ é calculado. $S(\sigma)$ é o conjunto de comparadores que não retornaram igualdade para todas as comparações executadas.

Agora toda unidade i em $S(\sigma)$ tal que $|N(i)| = t + 1$ é examinada. Se removendo uma unidade k de $N(i)$ resultar em um AFS, então o conjunto de unidades falhas $F = N(i) - k$. Se ocorrer este caso o algoritmo então termina.

Caso F não for determinado pelo passo anterior, então toda unidade i em $S(\sigma)$, tal que $|N(i)| = t$, é examinada. Primeiramente o algoritmo verifica se $N(i)$ é um AFS: se ocorrer este caso, $F = N(i)$, e o algoritmo termina. Caso contrário, ainda existem unidades fora de $N(i)$ que podem estar falhas. Para verificar estas unidades, o hiper-grafo $Z = (V, H(\sigma))$ é criado, onde o conjunto $H(\sigma)$ é construído da seguinte forma. Inicialmente $H(\sigma) = \{\{i, j, k\} \mid (i, j)_k \in C$ e $r((j, k)_i) = 1\}$. Então o seguinte passo é executado até que $H(\sigma)$ não mude mais: se $\{i, j, k\} \in H(\sigma)$ e m testou k como sem-falha e $\{i, j, m\} \notin H(\sigma)$ então $\{i, j, m\}$ é adicionado ao $H(\sigma)$.

No próximo passo do algoritmo, cada unidade $h \in H(\sigma)$ substitui, uma a uma, cada unidade k em $N(i)$. O algoritmo verifica se o conjunto resultante é um *vertex cover set* do hiper-grafo $Z = (V, H(\sigma))$. Se ocorrer tal caso, o conjunto de unidades falhas foi encontrando, $F = N(i) - k + h$. E o algoritmo então termina.

Finalmente, se F ainda não foi encontrado nos passos anteriores, um novo grafo $Y = (V, M(\sigma))$ é construído, com $M(\sigma)$ construído em cinco passos: (1) Para qualquer $i \notin S(\sigma)$, se $r((j, k)_i) = 1$ e i testou ambos j e k como sem-falha, então $(i, i) \in M(\sigma)$. (2) Para qualquer $i \notin S(\sigma)$, se $r((j, k)_i) = 1$ e i testou j como falho, então $(i, j) \in M(\sigma)$, e se $r((j, k)_i) = 1$ e i testou k como falho, então $(i, k) \in M(\sigma)$. (3) Para qualquer $i \in S(\sigma)$, se existe $j \in N(i)$ tal que $j \in S(\sigma)$ então $(i, j) \in M(\sigma)$. (4) Para qualquer $i \in S(\sigma)$, se existe $j \in N(i)$ tal que $j \notin S(\sigma)$ e, se i testou j como sem-falha, então $(j, p) \in M(\sigma)$ para todo $p \in N(i) - \{j\}$, enquanto se i testou j como falho, então $(i, j) \in M(\sigma)$. (5) Para qualquer $(p, q) \in M(\sigma)$, se p testou α como sem-falha e q testou β como sem-falha o $(\alpha, \beta) \in M(\sigma)$ e $(p, \beta) \in M(\sigma)$.

Todos os loops são removidos e um algoritmo para o cálculo de matching em grafos, como o algoritmo apresentado em [11], é executado sobre Y. No passo final, o vertex cover set mínimo F de Y é encontrado usando [12] onde uma técnica de diagnóstico tradicional baseado no modelo PMC é apresentada.

3.2. O algoritmo de Yang e Tang

Yang e Tang em [10] apresentaram um algoritmo de diagnóstico para o modelo MM* como uma alternativa ao algoritmo proposto por Sengupta e Dahbura. As seguintes definições são necessárias para entender o algoritmo.

Seja σ a síndrome de comparação do sistema. Seja $N(i)$ o conjunto de vizinhos da unidade i e $|N(i)|$ o grau de i. Para duas unidades adjacentes u e v, v é um filho σ-0 de u se existe $w \in N(u)$ tal que $r((v, w)_u) = 0$, ou seja, v é um filho σ-0 de u se a unidade u avaliar a unidade v como sem-falha. Uma unidade é um comparador σ-0 se ela possuir ao menos um filho σ-0; caso contrário, v é um filho σ-1 de u. COMP_1 representa o conjunto de todos os comparadores σ-1. COMP_{10} representa o conjunto de todos os comparadores σ-1 com grau t. COMP_{11} representa o conjunto de todos os comparadores σ-1 com grau $t + 1$. COMP_{12} representa o conjunto de todos os comparadores σ-1 com grau $\geq t + 2$. $\text{COMP}_1 = \text{COMP}_{10} \cup \text{COMP}_{11} \cup \text{COMP}_{12}$. $\text{SON}_0(u)$ representa o conjunto de todos os filhos σ-0 da unidade u. A unidade u é σ-conflitante se u tem dois filhos σ-0 v e w tal que $r((v, w)_u) = 1$. CONF representa o conjunto de todas as unidades σ-conflitantes.

Uma unidade u é um σ-0 pai de v se existe uma unidade w tal que $r((v, w)_u) = 0$.

$\text{PARENT}_0(v)$ representa o conjunto de todas as unidades σ-0 pai da unidade v, e $\text{PARENT}_0(U) = \bigcup_{x \in U} \text{PARENT}_0(x)$. Uma unidade u é um σ-0 predecessor de v se existe a sequência de unidades $w_0 = u, w_1, \ldots, w_p, w_{p+1} = v$ tal que $w_i \in \text{PARENT}_0(w_{i+1})$, para $i = 0, 1, \ldots, p$. $\text{PRED}_0(u)$ representa o conjunto de todos os σ-0 predecessores da unidade u. $\text{PRED}_0(U) = \bigcup_{x \in U} \text{PRED}_0(u)$ e $\text{PRED}_0[U] = \text{PRED}_0(U) \cup U$.

A mesma definição de AFS (possível conjunto de unidades falha) também é utilizada neste algoritmo. Um t-AFS de σ é definido como um AFS de σ com no máximo t unidades. Seja K um conjunto de unidades tal que $K \subseteq V$, um K^{+1} AFS de σ é um AFS de σ que possui a forma $K \cup \{u\}$ para algum $u \in V - K$.

Seja σ a síndrome de comparação do sistema G e H um subsistema de G, $\sigma|_H$ é uma síndrome de comparação sobre H definida por $(v, w)_u$ para todo $u, v, w \in V(H)$, $v, w \in N(u)$, e $v \neq w$, onde $V(H)$ é o conjunto de vértices do grafo H. A síndrome de testes σ induzida sobre o modelo PMC, denotada por $t[\sigma]$, é definida desta forma: para quaisquer duas unidades adjacentes u e v, seja $t[\sigma](u, v) = 0$ ou 1 se, respectivamente, u é um σ-0 pai de v ou não.

O algoritmo de diagnóstico apresentado por Yang e Tang – chamado MM*_DIAG – é mostrado na figura 2. O algoritmo recebe como entradas o grafo $G(V, E)$ que representa o sistema t-diagnosticável e a síndrome de comparação do sistema. O algoritmo produz como saída o conjunto de unidades falhas.

```
Algorithm: MM*_DIAG
/* Input:   An MM* t-diagnosable system G(V, E) and a comparison syndrome σ */
/* Output:  The faulty set */

begin
   /* First Phase */
   for every node u of G do determine PARENT0(u) and SON0(u);
   calculate COMP1, COMP10 and COMP11;
   if there exist u ∈ COMP11 and v ∈ N(u) such that
            N(u) − {v} is an AFS of σ then
            return (N(u) − {v});
   end if
   if there exist u ∈ COMP10 such that N(u) is an AFS of σ then
            return (N(u));
   end if
   if there exist u ∈ COMP10 and v ∈ N(u) such that
            N(u) − {v} is an AFS of σ then
            return (N(u) − {v});
   end if
   if there exist u ∈ COMP10 and v ∈ N(u) such that
            CHECK_IF( G, N(u) − {v}, σ) ≠ "No." )
            return (CHECK_IF( G, N(u) − {v}, σ));
   end if
   /* Second Phase */
   determine CONF;
   U ← PRED0[COMP1 ∪ CONF];
   /* Third Phase */
   build the subgraph H = G − U, build the test syndrome t[σ|H];
   find out the minimum AFS U' of t[σ|H] by calling Sullivan's algorithm
            return (U ∪ U');
end
```

Figura 2. O algoritmo MM*_DIAG.

O algoritmo é dividido em três fases. Na primeira fase o algoritmo define os conjuntos COMP_{10}, COMP_{11} e identifica o conjunto COMP_{12}, ou seja, define todos os comparadores σ-1. Os autores provam que todos os comparadores no conjunto COMP_{12}

são falhos. Para todo comparador x nos conjuntos COMP_{11} ou COMP_{10}, o algoritmo verifica todos os possíveis candidatos para um t-AFS desde que x seja sem-falha.

Neste contexto o procedimento CHECK_IF é importante. Este procedimento recebe como entrada um sistema $G(V, E)$, a síndrome de comparação correspondente σ, e um conjunto $K \subset V$ que não é um AFS. O procedimento retorna um outro conjunto $K^{+1} = K \cup \{u\} \mid u \in V - K$, se existir tal conjunto. Este novo conjunto precisa ser um AFS e precisa ter $|K| + 1$ unidades. Caso contrário, o procedimento CHECK_IF retorna o valor "No". Se um t-AFS for encontrado em um destes passos, o diagnóstico está completo e o conjunto AFS encontrado é retornado como o conjunto de unidades falhas. Caso contrário, todos os comparadores COMP_1 são considerados falhos e o algoritmo passa para a segunda fase.

Na fase 2, o conjunto CONF é calculado. Os autores provam que todas as unidades do conjunto CONF são falhas. Então, o algoritmo define um novo conjunto PRED_0 com base nos conjuntos COMP_1 e CONF. O conjunto $U = \text{PRED}_0[\text{COMP}_1 \cup \text{CONF}]$ representa todos os predecessores que testaram diretamente ou indiretamente alguma unidade nos conjuntos COMP_1 e CONF. Os autores também provam que todas as unidades em $\text{PRED}_0[\text{COMP}_1 \cup \text{CONF}]$ são falhas.

Na terceira e última fase do algoritmo, a tarefa de diagnóstico é convertida para uma tarefa do modelo PMC, ao invés do modelo MM*. Um subconjunto $H = G - U$ composto de todas as unidades que ainda não foram identificadas como falhas nas duas etapas anteriores é construído. Em um passo chave do algoritmo, uma síndrome de testes do modelo PMC $t[\sigma|_H]$ baseada nas unidades de H é construída a partir da síndrome de comparação original σ. Os autores então provam que H é $(t - |U|)$-diagnosticável sobre o modelo PMC e que $F - U$ é o único conjunto $(t - |U|)$-AFS possível dada a síndrome de testes $t[\sigma|_H]$. Portanto a localização das unidades falhas remanescentes é equivalente a encontrar o AFS mínimo da síndrome de testes. Então é possível encontrar o conjunto AFS mínimo aplicando o algoritmo $O(\delta^3 + |E|)$ apresentado por Sullivan em [13].

4. O Algoritmo Proposto

Esta sessão apresenta um novo algoritmo de diagnóstico baseado em comparações para sistemas gerais com base no modelo MM*. Este novo algoritmo é uma alternativa ao algoritmo proposto por Sengupta e Dahbura [9] e ao algoritmo proposto por Yang e Tang [10]. Os algoritmos para o diagnóstico baseado em comparações em sistemas gerais de [10] e [9] envolvem, em alguma fase, técnicas de diagnóstico tradicional baseadas no modelo PMC. A estratégia do algoritmo proposto nesta seção realiza o diagnóstico utilizando exclusivamente a síndrome de comparações.

O novo algoritmo de diagnóstico apresentado é mostrado na figura 3. O algoritmo recebe como entradas o gravo $G(V, E)$ e o conjunto de todas as comparações, isto é, a síndrome do sistema (σ). Como resultado o conjunto F de unidades falhas é retornado. Algumas definições necessárias para entender o algoritmo são apresentadas abaixo.

No modelo MM* cada unidade do sistema executa comparações para todas as unidades vizinhas. Se uma comparação resultar em igualdade, as três unidades são sem-falha ou então a unidade comparadora – unidade testadora, ou simplesmente testador – é falha; se uma comparação resultar em diferença, pelo menos uma das três unidades é falha [9]. Cada uma das N unidades do sistema recebe um identificador único i, onde $0 \leq i \leq N - 1$. Para o sistema de N unidades ser t-diagnosticável, as seguintes duas condições devem ser satisfeitas [8]: $N \geq 2t + 1$, e cada unidade tem ordem maior ou igual a t.

Seja σ a síndrome de comparação do sistema $G(V, E)$ e F o conjunto de unidades falhas. $N(i) = \{j \mid (i, j) \in E\}$ é o conjunto de vizinhos da unidade i. Um conjunto $U \subset V$ também é chamado de um AFS (*Allowable Fault Set* - possível conjunto de unidades falhas) do sistema S para a síndrome σ, se para quaisquer três unidades i, j, k tal que $(j, k)_i \in C$, $j, k \in N(i)$ e $j \neq k$, as seguintes duas condições são satisfeitas:

— se $i \in V - U$ e $j, k \in V - U$ então $r((j, k)_i) = 0$
— se $i \in V - U$ e $\{j, k\} \cap U \neq \emptyset$ então $r((j, k)_i) = 1$

Seja F o conjunto de unidades falhas do sistema. O conjunto F deve ser um AFS de σ. Um t-AFS de σ é um AFS de σ que possui no máximo t unidades. Um sistema é MM* t-diagnosticável se cada síndrome de comparação deste sistema com até t unidades falhas possui um único conjunto t-AFS [9].

Seja P um conjunto que contém N subconjuntos P_i onde $0 \leq i \leq N - 1$. A construção do conjunto P é baseada em todas as comparações da síndrome e $P_i = \{\{j, k \in (j, k)_i \mid r((j, k)_i) = 1\} - \{p, q \in (p, q)_i \mid r((p, q)_i) = 0\}\}$, ou seja, P_i é o conjunto de unidades identificadas exclusivamente como falhas nas comparações realizadas pela unidade i.

Seja CFF um conjunto que também contém N subconjuntos CFF_i onde $0 \leq i \leq N - 1$. A construção do conjunto CFF é baseada em todas as comparações da síndrome que indicaram igualdade e $CFF_i = \{j, k \in (j, k)_i \mid r((j, k)_i) = 0\}$. Em outras palavras, CFF_i é o conjunto formado por todas as unidades identificadas como sem-falha em qualquer teste da unidade i.

Sejam CF e CR conjuntos com também N subconjuntos cada, respectivamente CF_i e CR_i, onde $0 \leq i \leq N - 1$. A construção dos conjuntos CF e CR são baseadas em todas as comparações de σ que resultaram em diferença. Cada conjunto CF_i é um conjunto de unidades; por outro lado, cada conjunto CR_i contem um conjunto de pares (j, k). Além disso, o conjunto CF é baseado no conjunto CFF e é construído da seguinte forma: $CF_i = \{j \in \exists (j, k)_i \mid r((j, k)_i) = 1$ e $k \in CFF_i\}$. Em outras palavras, CF_i é o conjunto de unidades j das comparações $(j, k)_i$ que resultaram diferença onde a unidade k possui ao menos uma comparação com qualquer outra unidade indicando igualdade. O conjunto CR é construído da seguinte forma: $CR_i = \{(j, k) \in (j, k)_i \mid r((j, k)_i) = 1$ e $j, k \notin CFF_i\}$.

Durante a fase de inicialização do algoritmo, o conjunto de unidades falhas F é atribuído como vazio, e o conjunto P é calculado. Logo após a fase de inicialização, a primeira fase tem o objetivo de encontrar F ou parte de F através da análise das unidades dos conjuntos P_i testadas exclusivamente como falhas. A primeira fase se inicia com um teste trivial: se não existir nenhuma unidade testada como falha, $|P_i| = 0$, onde $0 \leq i \leq N - 1$, o algoritmo termina e o conjunto $F = \emptyset$.

Caso o algoritmo continue, o algoritmo então classifica como falhas – atribui ao conjunto F – todas as unidades que identificaram mais de $t + 1$ unidades como falhas. Na sequência, o seguinte teste é realizado: se o conjunto F for um AFS, o algoritmo identificou o conjunto de unidades falhas. O algoritmo então termina.

Caso o algoritmo continue, todas as unidades i que testaram exatamente $t + 1$ unidades como falhas ($|P_i| = t + 1$) passam pela seguinte verificação: se existir algum j

Algoritmo:
/* *Input*: Um sistema MM* t-diagnosticável $G(V, E)$ e a síndrome de comparação σ */
/* *Output*: O conjunto de unidades falhas F */

/* *Fase de Inicialização* */
$F \leftarrow \emptyset$;
calcular P;

/* *Primeira Fase* */
if $\nexists i$ tal que $|P_i| > 0$ **then**
  return F;
end if

$F \leftarrow i$ tal que $|P_i| > t + 1$;
if F é um AFS **then** return F; **end if**

for each $i \in V$ tal que $|P_i| = t + 1$ **loop**
  for each $j \in P_i$ **loop**
    if $P_i - j$ é um AFS **then** return $P_i - j$; **end if**
  end loop
  $F \leftarrow F \cup i$;
end loop

for each $i \in V$ tal que $|P_i| = t$ **loop**
  if P_i é um AFS **then** return P_i; **end if**

  for each $j \in P_i$ **loop**
    if $P_i - j$ é um AFS **then** return $P_i - j$; **end if**

    for each $k \in V - P_i - F$ **loop**
      if $P_i - j + k$ é um AFS **then** return $P_i - j + k$; **end if**
    end loop
  end loop
  $F \leftarrow F \cup i$;
end loop
if F é um AFS **then** return F; **end if**

/* *Segunda Fase* */
calcular CFF, CF, CR;
for $i \in V - F$ **loop**
  if $CFF_i \cap \{F \cup CF_i\} \neq \emptyset$ **then**
    $F \leftarrow i$;
  else for any $j \in CFF_i$:
    $\{CFF_i \cup CFF_j\} \cap \{F \cup CF_i \cup CF_j\} \neq \emptyset$ **or**
    $\{CFF_i \cup CFF_j\} \cap \{p \mid (p, q) \in \{CR_i \cup CR_j\}$ and $q \in \{CFF_i \cup CFF_j\}\} \neq \emptyset$
   then
    $F \leftarrow i$;
  end if
end loop
if F é um AFS **then** return F; **end if**

/* *Terceira Fase* */
$FF \leftarrow \emptyset$;
while os conjuntos F ou FF aumentarem **loop**
  $FF \leftarrow FF \cup j \in V - \{F \cup FF\} \mid \exists$ mais de $t - |F|$ testadores i tal que $\exists\, r((j, k)_i) = 0$ e $i \notin F$;
  while o conjunto FF aumentar **loop**
    $FF \leftarrow FF \cup j \in V - \{F \cup FF\} \mid \exists r((j, k)_i) = 0$ onde $i \in FF$;
  end loop
  $F \leftarrow F \cup j \in V - \{F \cup FF\} \mid \exists r((j, k)_i) = 1$ onde $i, k \in FF$;
end loop
return F;

Figura 3. O algoritmo de diagnóstico apresentado para o modelo MM* de diagnostico baseado em comparação.

tal que $P_i - j$ é um AFS então o algoritmo termina e F é o conjunto AFS encontrado. Se nenhum AFS for encontrado, todas estas unidades i são classificadas como falhas.

Ainda no fim da primeira fase, uma verificação similar ocorre com cada uma das unidades i que testaram exatamente t unidades como falhas, ou seja, $|P_i| = t$.

— Se P_i for um AFS, o algoritmo identifica P_i como o conjunto F e termina.
— Se continuar, o algoritmo testa todos os subconjuntos possíveis de P_i com $t - 1$ unidades. Se algum subconjunto for um AFS, o algoritmo retorna o AFS encontrado e termina.
— Caso continue, o algoritmo testa todas as outras combinações possíveis de P_i com t unidades, substituindo uma a uma, cada unidade de P_i por cada uma das outras unidades de V. Se alguma combinação encontrada for um AFS, o algoritmo retorna o AFS encontrado e termina.
— Se nenhum AFS foi encontrado nos passos anteriores, todas as unidades i testadas são classificadas como falhas.

A segunda fase tem com objetivo identificar unidades falhas através da análise de possíveis contradições existentes na síndrome de comparação. Para cada uma das unidades do sistema ainda não identificadas como falha, dois testes são realizados:

— Se a interseção de CFF_i e $\{CF_i \cup F\}$ for diferente de nula, i é considerado falho. Em outras palavras, a unidade i é considerada falha se em suas comparações existir qualquer unidade identificada como falha e ao mesmo tempo como sem-falha.
— De forma análoga, o conjunto F, os testes da unidade i e os testes das unidades testadas por i como sem-falha agora são considerados. A unidade i será considerada falha se existir qualquer unidade identificada como falha e ao mesmo tempo sem-falha, ou seja: $F \leftarrow i$ tal que $\{CFF_i \cup CFF_j\} \cap \{p \mid (p, q) \in \{CR_i \cup CR_j\}$ e $q \in \{CFF_i \cup CFF_j\}\} \neq \emptyset$.

No final da segunda fase, o algoritmo verifica se o conjunto F corrente é um AFS. Caso isso ocorra, o algoritmo termina e retorna F.

Na terceira fase, o algoritmo realiza uma análise das unidades ainda não identificados como falhas. O objetivo desta fase é identificar o estado das demais unidades do sistema através das unidades sem-falha. O algoritmo então inicia um *loop* que é executado até que os conjuntos F e FF não se modifiquem mais. O loop realiza os seguintes três passos:

(1) Identifica como sem-falha – atribui ao conjunto FF – todas as unidades testadas como sem-falha por pelo menos outras $t - |F| + 1$ unidades ainda não identificadas como falhas.
(2) Um segundo loop é iniciado e executado até que o conjunto FF não mude mais. Este novo loop considera como unidades sem-falha – atribui ao conjunto FF – todas as unidades ainda não identificadas como falha ou sem-falha, tal que que exista qualquer comparação que resultou em igualdade realizada por qualquer unidade do conjunto FF.

(3) Todas as unidades $j \in V - \{F \cup FF\}$, tal que existe $r((j,k)_i) = 1$ são considerados falhas, se e somente se, $i, k \in FF$. Em outras palavras, este terceiro passo identifica as unidades falhas a partir das comparações que resultaram em diferença desde que a unidade testadora e uma das unidades testadas estejam sem-falha.

Após o fim da terceira fase, o algoritmo retorna o conjunto de unidades falhas F.

5. Esboço da Prova de Corretude

A corretude do algoritmo é definida em função de cada uma das três fases realizadas. Dessa forma, os argumentos também serão divididos de acordo com cada fase separadamente.

Primeira fase. A primeira fase analisa e identifica unidades falhas de acordo com a quantidade de testes que resultaram em diferença. Em seu primeiro teste, caso não exista a incidência de nenhuma comparação na síndrome que resultou em diferença, obviamente pode-se concluir que não existem falhas no sistema.

Se os resultados das comparações de uma unidade testadora i indicar a presença de um número maior que $t+1$ unidades falhas, estas unidades i são unidades falha. Esta afirmação decorre diretamente da t-diagnosticabilidade: o número máximo de unidades falhas para que o sistema seja diagnosticável é t. Portanto, em um conjunto maior ou igual a $t+2$ unidades existem ao menos 2 unidades sem-falha. A comparação destas duas unidades certamente resultaria em igualdade.

Caso ocorra a incidência de testadores i que indiquem – através dos resultados das suas comparações – a existência de um conjunto P_i com exatamente $t+1$ unidades, duas hipóteses devem ser consideradas:

(1) Como devido à t-diagnosticabilidade o possível conjunto de unidades falhas (AFS) não pode ser de tamanho $t+1$, deve existir um subconjunto de P_i de tamanho t que corresponde exatamente ao conjunto das unidades falhas do sistema. Ao retirar, uma a uma, cada unidade j do conjunto P_i, os conjuntos resultantes $P_i - j$ são testado com o objetivo de identificar se um AFS é encontrado. Se um AFS for encontrado, este é o conjunto das unidades falhas do sistema.

(2) Se nenhum AFS for encontrado no passo anterior, o testador i é falho. Por contradição, se o testador i fosse sem-falha e não existir um subconjunto de P_i de tamanho t que seja um AFS, devem existir ao menos duas unidades entre as $t+1$ unidades de P_i que são sem-falha. Neste caso, a comparação destas duas unidades resultaria em igualdade, o que contradiz a definição de P_i.

De forma similar ao caso anterior, se ocorrer a incidência de testadores i que indiquem a existência de um conjunto P_i com exatamente t unidades, uma das quatro alternativas abaixo é verdadeira:

(1) O conjunto P_i de tamanho t corresponde exatamente ao conjunto de unidades falhas do sistema;

(2) Existe um subconjunto de P_i de tamanho $t-1$ que corresponde exatamente ao conjunto de unidades falhas do sistema;

(3) Existe um outro conjunto de tamanho t formado por um subconjunto de P_i de tamanho $t-1$ e alguma outra unidade do sistema, que corresponde exatamente ao conjunto de unidades falhas do sistema; ou

(4) O testador i é falho.

Segunda fase. A segunda fase identifica unidades falhas através da observação de contradições nos resultados de comparações. Considere os conjuntos CFF, CF, CR e F. Se nos resultados das comparações realizadas por um testador i existir a identificação de qualquer unidade sem-falha (em CFF_i) que também esteja identificado como falho (em CF_i) ou então que já tenha sido identificado previamente como falha (em F), pode-se concluir que a unidade i é falha.

Agora considere as comparações realizadas por cada unidade $j \in CFF_i$, ou seja, cada unidade considerada como sem-falha pelo testador i. De forma análoga ao teste anterior, duas novas verificações podem ser realizadas sobre os testes reportados por cada uma das unidades j. Primeiramente, entre as unidades consideradas sem-falha pelo testador i (CFF_i) e entre as unidades consideradas sem-falha por j (CFF_j) não pode existir nenhuma unidade que também é considerada falha por i e j (CF_i e CF_j), ou já identificada como falha (em F). Caso exista tal contradição, a unidade i é uma unidade falha.

Por fim, também de forma análoga, entre as unidades consideradas sem-falha pelo testador i (CFF_i) e as unidades consideradas sem-falha por j (CFF_j) não podem existir outras duas unidades p, q tal que $(p, q) \in R$ e existe uma comparação $r((p, q)_i) = 1$ ou $r((p, q)_j) = 1$. Caso exista tal contradição, a unidade i é uma unidade falha. Ainda no fim da segunda fase, um último teste é realizado: se o conjunto F for um AFS o algoritmo encontrou o conjunto de unidades falhas.

Terceira fase. Se um AFS ainda não foi encontrado em uma das fases anteriores, ainda podem existir unidades falhas que ainda não foram identificadas. Considere agora um novo grafo X formado pela remoção dos vértices já identificados como falhos – nas duas fases anteriores – sobre o gravo G. O grafo resultante X é em um grafo conexo pois o grau mínimo de cada vértice ser t, mas foram removidos apenas número menor que t de vértices.

Considere agora apenas os resultados das comparações na síndrome σ das unidades em X. É possível identificar um conjunto FF inicial – não vazio – de unidades sem-falha. Esta identificação é realizada através da análise de comparações que resultaram em igualdade. Caso um número maior que $t - |F|$ de diferentes unidades testaram uma unidade p como sem-falha, todas estas unidades p são atribuídas a FF. Mesmo que o F ainda esteja vazio, esta condição será satisfeita; caso contrário, deveriam existir unidades falhas que já haveriam sido identificadas nas fases anteriores.

Neste momento, as unidades sem-falha e as unidades falhas são identificadas de forma recursiva. As unidades testadas em comparações que resultaram em igualdade realizadas pelas unidades de FF são unidades sem-falha. As unidades j de comparações $r((j, k)_i) = 1$ tal que $i, k \in FF$ são unidades falhas.

6. Conclusão

Este trabalho apresentou uma nova estratégia para a identificação de unidades falhas utilizando o modelo MM* de diagnóstico baseado em comparação. O diagnóstico baseado em comparações utiliza a comparação do resultado de tarefas atribuídas a pares de unidades para a identificação de falhas no sistema. Qualquer diferença na comparação dos resultados das tarefas indica que uma ou ambas as unidades estão falhas. Caso não ocorra nenhuma diferença na comparação das tarefas, e caso a unidade testadora esteja sem falha, ambas as unidades testadas estão sem-falha. O diagnóstico completo do sistema é obtido através do conjunto de todos os resultados das comparações.

O algoritmo proposto neste trabalho é uma alternativa ao algoritmo proposto por Sengupta e Dahbura e ao algoritmo proposto por Yang e Tang. Estes dois algoritmos existentes de diagnóstico baseado em comparações para sistemas gerais, envolvem, em alguma fase, técnicas de diagnóstico tradicional baseadas no modelo PMC. A estratégia proposta neste trabalho realiza o diagnóstico do sistema utilizando exclusivamente a síndrome de comparações, sem a necessidade de convertê-la para outro modelo. Um esboço da prova de corretude do algoritmo proposto também foi apresentado. Entre os próximos trabalhos, estão previstas a implementação do algoritmo e a realização de experimentos práticos.

Referências

[1] H. Wang, D. M. Blough, and L. Alkalaj, "Analysis and Experimental Evaluation of Comparison-Based System-Level Diagnosis for Multiprocessor Systems," *Twenty-Fourth Intl. Symp. on Fault-Tolerant Computing*, pp. 55–64, June 1994.

[2] S. Rangarajan, D. Fussell, and M. Malek, "Built-in Testing of Integrated Circuits Wafers," *IEEE Transactions on Computers.*, vol. 39, pp. 195–205, Feb. 1990.

[3] D. Fussell, M. Malek, and S. Rangarajan, *Wafer-Scale Testing/Design for Testability*, ch. 9, pp. 413–472. Kluwer, 1989.

[4] F. Preparata, G. Metze, and R. T. Chien, "On the Connection Assignment Problem of Diagnosable Systems," *IEEE Transactions on Computers.*, vol. 16, pp. 848–854, 1968.

[5] G. Masson, D. Blough, and G. Sullivan, *System Diagnosis*. Prentice-Hall, 1996.

[6] M. Malek, "A Comparison Connection Assignment for Diagnosis of Multiprocessor Systems," *Proc. 7th International Symp. Computer Architecture*, pp. 31–36, 1980.

[7] K. Y. Chwa and S. L. Hakimi, "Schemes for Fault-Tolerant Computing: A Comparison of Modularly Redundant and t-Diagnosable Systems," *Information and Control.*, vol. 49, pp. 212–238, 1981.

[8] J. Maeng and M. Malek, "A Comparison Connection Assignment for Self-Diagnosis of Multiprocessor Systems," *Digest 11th International Symp. Fault Tolerant Computing*, pp. 173–175, 1981.

[9] A. Sengupta and A. T. Dahbura, "On Self-Diagnosable Multiprocessor Systems: Diagnosis by Comparison Approach," *IEEE Transactions on Computers.*, vol. 41, pp. 1386–1396, Nov. 1992.

[10] X. Yang and Y. Y. Tang, "Efficient Fault Identification of Diagnosable Systems under the Comparison Model," *IEEE Transactions on Computers.*, vol. 56, pp. 1612–1618, Dec. 2007.

[11] S. Micali and V. V. Vazirani, "An $O(\sqrt{|V|}|E|)$ Algorithm for Maximum Matching in General Graphs," *In Proc. 16th Annu. Symp. Foundations of Comput. Science*, pp. 17–27, Oct. 1980.

[12] A. T. Dahbura and G. M. Masson, "An $O(n^{2.5})$ Fault Identification Algorithm for Diagnosable Systems," *IEEE Transactions on Computers.*, vol. C-33, pp. 486–492, June 1984.

[13] G. Sullivan, "An $O(t^3 + |E|)$ Fault Identification Algorithm for Diagnosable Systems," *IEEE Transactions on Computers.*, vol. 37, pp. 388–397, Apr. 1988.