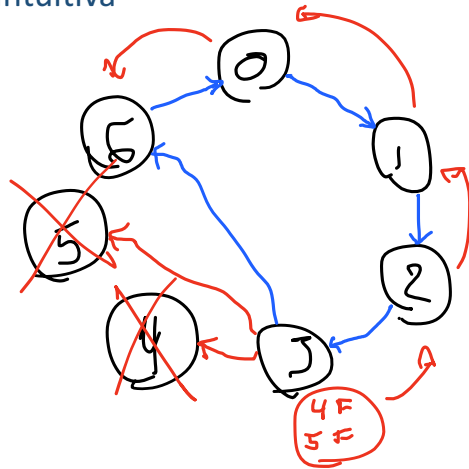


Aula 5 - Algoritmo Adaptive-DSD

Wednesday, March 16, 2016 13:46

Descrição intuitiva



0	200.17.212.176	1501
1	180.15.155.12	1600
...

Algoritmo Adaptive-DSD executado pelo nodo i

```

AdaptiveDSD(i)
  j ← i
  REPITA
    REPITA
      j ← (j+1) MOD N
      teste o nodo j
      se j é sem-falha
        então obtenha TESTE_Upj[]
    ATÉ j ser testado sem-falha
    para todo nodo k não testado por i nesse intervalo de testes
      TESTED_UPi[k] ← TESTED_UPj[k]
  Durma
  ATE o próximo intervalo de testes
    
```

$TESTED_UP_a[1] = c$

TESTED_UP

0	1	2	3	4	5	6
*	*	*	*	*	*	*

a considera que b
testou c sem-falha

1	2	3	6	*	*	0
---	---	---	---	---	---	---

Mostre, para intervalos de testes consecutivos, a partir do início, os vetores $TESTED_UP_{0,2,5}$

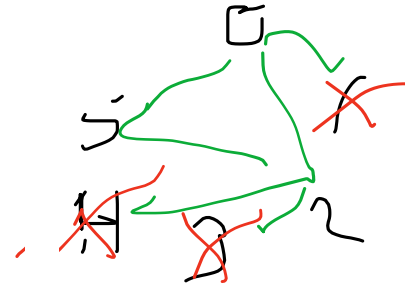
Intervalo de testes 1

0->2 obtém info sobre 2, 3, 4, 5

2->5 obtém info sobre 5, 0, 1

5->0 obtém info sobre 0, 1, 2, 3, 4

	0	1	2	3	4	5
$TESTED_UP_0$	2	*	*	*	*	*
$TESTED_UP_2$	*	*	5	*	*	*
$TESTED_UP_5$	*	*	*	*	*	0



Intervalo de testes 2

	0	1	2	3	4	5
$TESTED_UP_0$	2	*	5	*	*	*
$TESTED_UP_2$	*	*	5	*	*	0
$TESTED_UP_5$	2	*	*	*	*	0

Intervalo de testes 3

	0	1	2	3	4	5
$TESTED_UP_0$	2	*	5	*	*	0
$TESTED_UP_2$	2	*	5	*	*	0
$TESTED_UP_5$	2	*	5	*	*	0

O algoritmo diagnose usa o vetor $TESTED_UP$ para descobrir quais nodos estão {falhos, sem falha}

O algoritmo diagnose executado pelo nodo i:

```

Diagnose(i)
  Para k<-0 até N-1 faça
    STATEi[k] <- FALHO
  fim Para

```

```

aux ← i
Repita
    STATEi[aux] ← SEM-FALHA
    aux ← TESTE_UPi[aux]
Até aux = i

```

```

STATE0 = [F, F, F, F, F, F]
aux=0
STATE0 = [S, F, F, F, F, F]
aux=2
STATE0 = [S, F, S, F, F, F]
aux=5
STATE0 = [S, F, S, F, F, S]
aux=0

```

Prova

Provas de correção do algoritmo são baseados na **latência**.

Rodada de testes é o intervalo de tempo até que todos os nodos sem-falha executaram testes em pelo menos 1 nodo sem-falha.

Premissa: diagnóstico estático: o estado dos nodos não muda durante o diagnóstico.

Teorema 1

Considere o sistema S executando o algoritmo AdaptiveDSD por uma rodada de testes. O grafo de testes $T(S)$, cujos vértices correspondem aos nodos do sistema e cujas arestas direcionadas correspondem aos testes executados, contém um ciclo - um caminho direcionado de qualquer nodo sem-falha para qualquer nodo sem-falha.

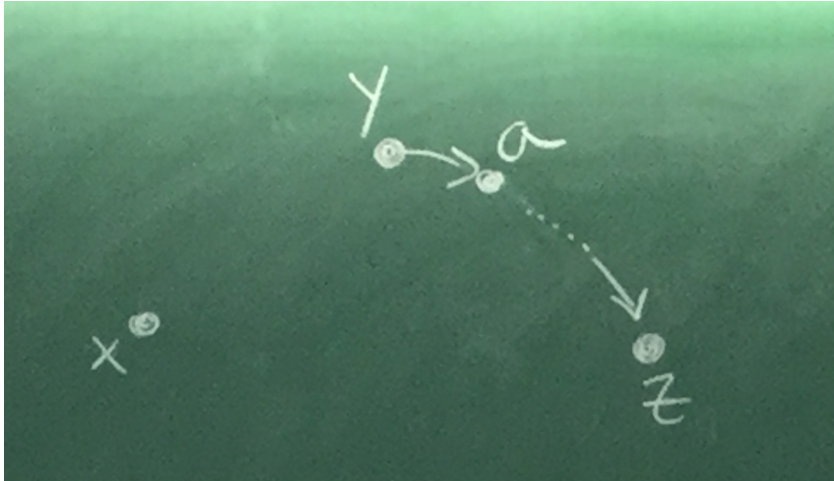
Prova por contradição

Escolha 2 nodos sem-falha x e z tais que **não** existe um caminho direcionado entre eles e $d(z, x)$ é o mínimo para todos os pares sem caminho direcionado.

Identifique o maior y , $y < z$, tal que y é um nodo sem-falha e existe um caminho de y para z . Após uma rodada de testes, y deve ter testado algum nodo sem-falha, digamos a . Como $y < z$ e y é o maior nodo para o qual existe um caminho para z , então a não pode estar entre y e z , pois caso contrário a seria y !

Entretanto, pela definição do algoritmo, y deve ter testado z falho antes de testar o a !

Contradição: z é um nodo sem-falha!



Corolário

Após uma rodada de testes do algoritmo AdaptiveDSD, todos os nós sem-falha formam um ciclo no grafo de testes $T(S)$.

Pelo Teorema 1, existe um caminho direcionado de qualquer nó sem-falha de $T(S)$ para qualquer nó sem-falha de $T(S)$. Além disso, há apenas uma aresta direcionada de cada nó sem-falha para um outro nó sem-falha.

O ciclo direcionado é a única estrutura que satisfaz a estas duas condições.

Latência de um algoritmo de diagnóstico distribuído é o número de rodadas de testes necessários para completar o diagnóstico do sistema.

AdaptiveDSD com N nós tem latência de N rodadas de teste, no pior caso, e 1 rodada, no melhor caso.