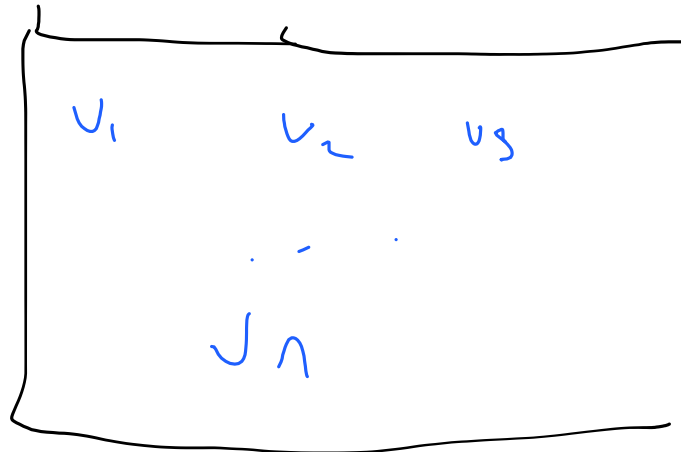


Aula 4 - Diagnósticos em nível de sistema (cont.)

Monday, March 14, 2016 13:37

Revisão



- Unidades executam testes umas nas outras
- Reportam os resultados de testes para um **observador central**
- Observador central determina, a partir da síndrome do sistema, quais unidades estão falhas e quais estão sem falhas.

Modelo PMC foi proposto em 1967.

Em 1970, dois pesquisadores (Hakimi & Amin) caracterizaram o modelo PMC:

- Se duas unidades não se testam mutuamente, as seguintes condições garantem que um sistema é t -diagnosticável:
 - a. $N \geq 2t + 1$
 - b. Cada unidade é testada por t outras.

New Self

No começo dos anos '80, Kuhl e Reddy propuseram o diagnóstico distribuído.

- As unidades sem-falha do sistema são capazes de realizar o diagnóstico, eliminando o observador central.
- Além de executar os testes, as unidades compartilham os resultados dos testes.

- Além de executar os testes, as unidades compartilham os resultados dos testes
- As próprias unidades completam o diagnóstico a partir da síndrome.

Propuseram o algoritmo **new-self**.

- Um assinalamento de testes é definido.
 - o Alguns nodos (unidades) testam seus vizinhos, repassando resultados de testes, e assim por diante.
 - o A premissa que permite esse diagnóstico é: um nodo sem-falha reporta resultados corretos de testes (teste perfeito).

Provam que se $N \geq 2t + 1$ e cada nodo é testado por $t + 1$ outros nodos: o sistema é t -diagnosticável.

O número de mensagens por este algoritmo é $N^2(t + 1)^2$.

Se $t = N - 1$ (hipotético!), gastaria $N^2(N - 1 + 1)^2 = N^4$.

Se $N = 100$, $N^2 = 10.000$, $N^4 = 100.000.000$.

Algoritmo **new self** é muito caro!

Diagnóstico Adaptativo

No começo dos anos '80, um outro grupo (Hakini & Nakajima), independentemente, propôs outra abordagem do diagnóstico: **diagnóstico adaptativo**.

Ambos papers foram publicados na mesma edição (Vol 33) da revista IEEE Transactions on Computers (1984).

No diagnóstico adaptativo, ao invés de determinar/executar um assinalamento de testes, os nodos executam testes em rodadas; após obter os resultados dos seus próprios testes, um nodo determina quais os próximos testes que vai executar. Em outras palavras: usando resultados dos testes executados em uma rodada, define os testes da rodada seguinte.

Diagnóstico Adaptativo & Distribuído

No começo dos anos 1990, Bianchini & Buskens, propuseram uma abordagem para um diagnóstico ao mesmo tempo adaptativo e distribuído.

- ★ Propuseram o algoritmo Adaptive-DSD (**Adaptive Distributed System-level Diagnosis algorithm**) (Artigo recomendado pelo Elias para leitura)

R.P. Bianchini & R.W. Buskens, "Implementation of Online Distributed System-level Diagnosis theory," IEEE Transactions on Computers, Vol. 41, No. 5, 1992.

O Algoritmo Adaptive-DSD

Premissas básicas:

- Assume que o sistema é representável por um grafo completo (fully-connected graph).

★ Outra dica do Elias: site [Planet Lab](#)

- Os enlaces de comunicação não falham (faz mais sentido do que parece - fazível em Ethernet).
- Cada nodo pode estar em 1 de 2 estados possíveis {falho, sem-falha}
- Até $N - 1$ nodos do sistema podem estar falhos.
- Premissa PMC: testes perfeitos.

Ideia básica:

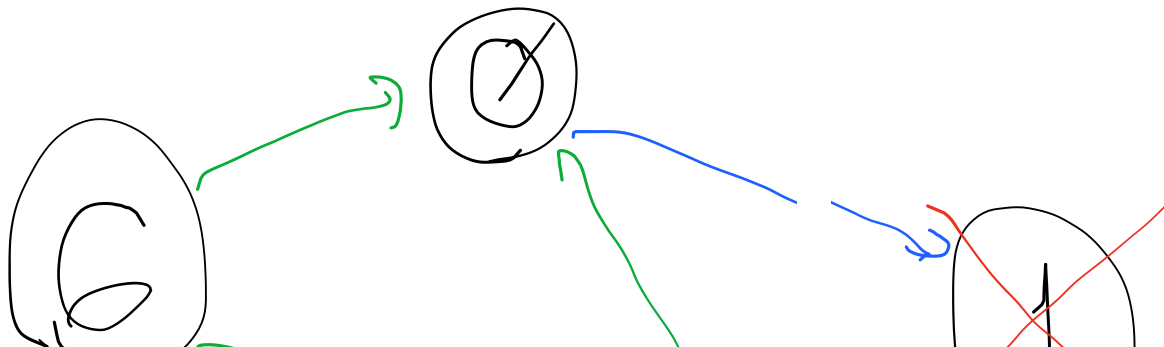


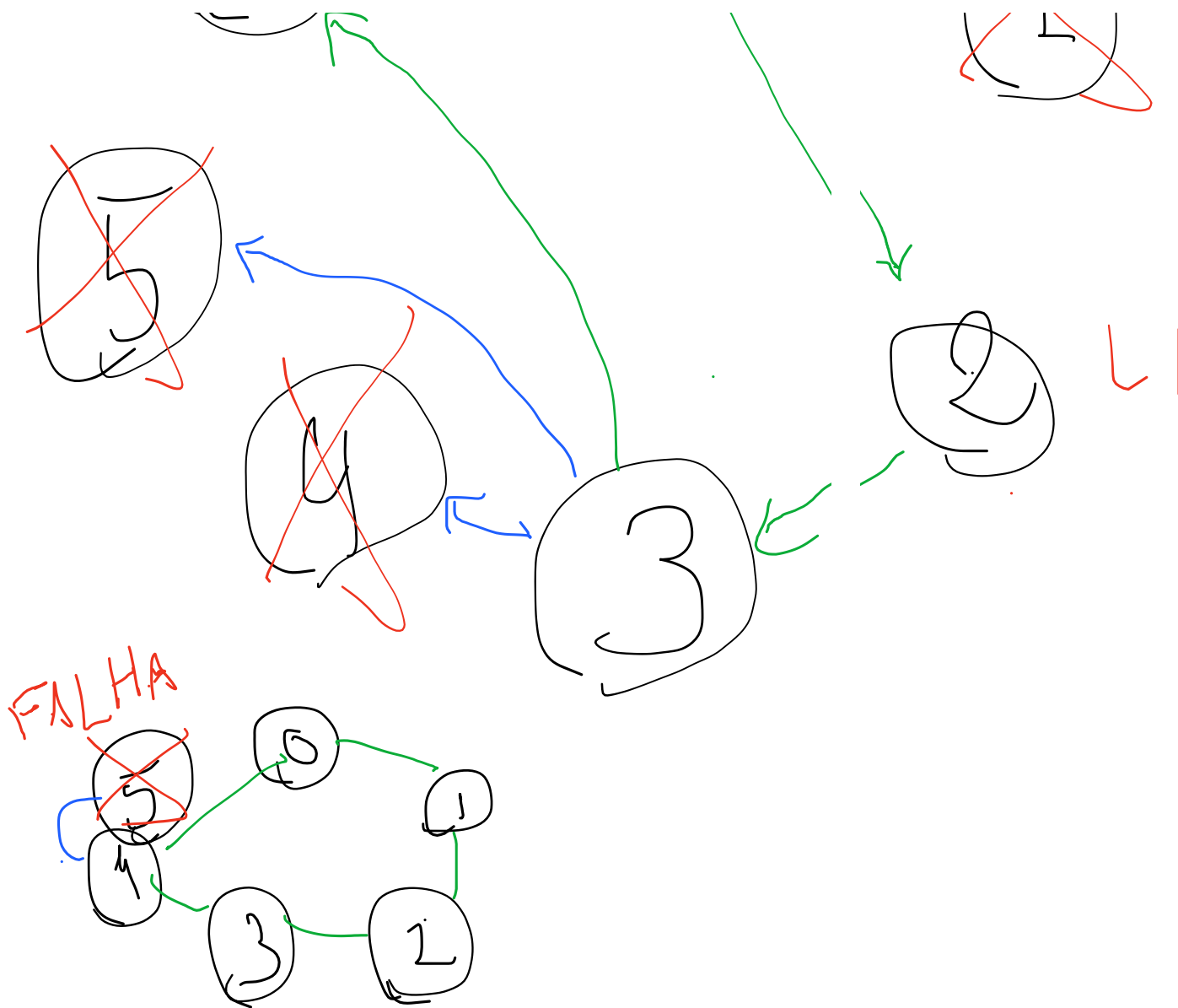
Sejam os nodos i e j sem-falha; j testou k .

Ao testar j , o nodo i pode "acreditar" na informação sobre k obtida a partir de j .

- Cada nodo mantém o vetor $TESTED_UP[1..N]$
- A entrada $TESTED_UP_x[y] = z$ significa que o nodo x tem a informação que o nodo y testou o nodo z sem falha.
- Nodos têm identificadores sequenciais $0..N-1$. Um nodo sem-falha executa testes sequencialmente, até encontrar outro nodo sem-falha (do qual obtém informações de diagnóstico) ou testar todos os outros nodos falhos.
- Se um nodo sem-falha foi testado, atualiza o vetor local $TESTED_UP$, exceto para os testes que executou.

Exemplo:





Hora do algoritmo!

O algoritmo executado no nó i :

O algoritmo executado no nó i .

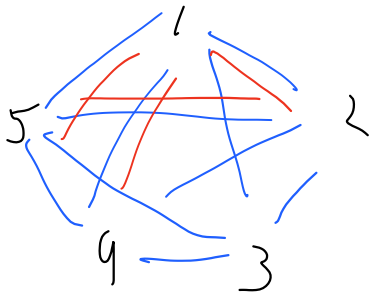
```
Atestar <- i
REPEAT
  Atestar <- (atestar+1) MOD N
  TESTE(atestar)
UNTIL (nodo atestar está sem falha)

TESTED_UPi[i] <- atestar
```

Atualiza posições não testadas com informações obtidas de atestar

$$N = 5$$

$$t = 2 - 1$$



$$N \neq t \rightarrow t \rightarrow N$$