



# Sistemas Distribuídos

## **VCube**: Um Algoritmo Distribuído Hierárquico

**Prof. Elias P. Duarte Jr.**

**Universidade Federal do Paraná (UFPR)**

**Departamento de Informática**

**[www.inf.ufpr.br/elias/sisdis](http://www.inf.ufpr.br/elias/sisdis)**

# Sumário

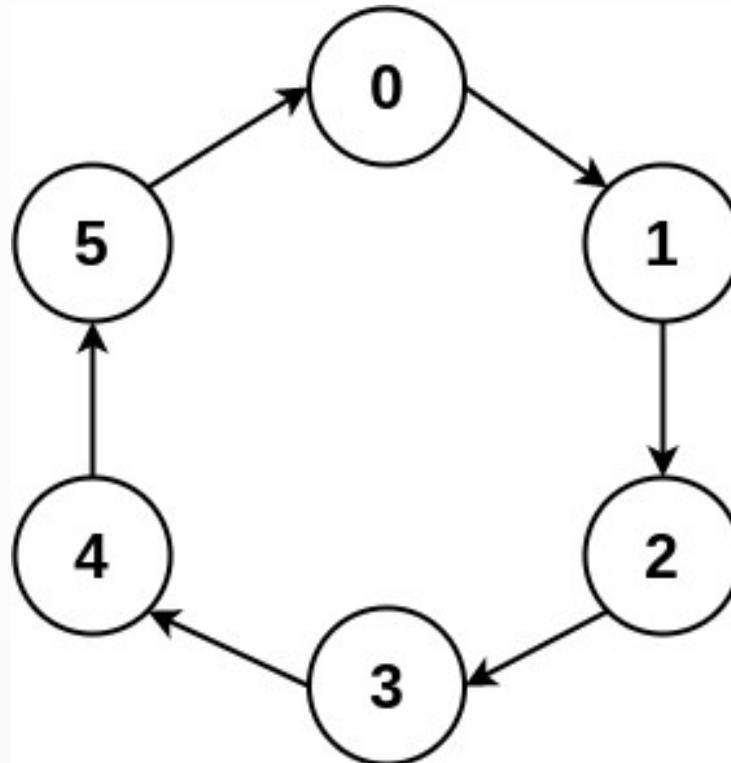
- A Latência do Algoritmo VRing: N rodadas de testes
- Como diminuir: Estratégias Orientadas a Evento
- Um Algoritmo Força-Bruta
- Um Algoritmo Hierárquico: VCube
- Hipercubo: Uma Topologia Importante
- VCube: Muito Além do Hipercubo
- A Função  $c(i,s)$ : vizinho do processo  $i$  no cluster  $s$
- Especificação do Algoritmo VCube
- VCube: Propriedades Escaláveis

# A Latência do Algoritmo VRing

- Latência: após a ocorrência de um evento...
  - Lembre-se: um evento corresponde a um processo correto se tornar falho ou vice-versa
- ...quantas rodadas de testes são necessárias até que todos os processos corretos saibam do evento?
  - Lembre-se: uma rodada de testes é o intervalo de tempo no qual todos os processos corretos testam pelo menos 1 outro processo correto
- Resposta:  $N$  rodadas de testes no pior caso

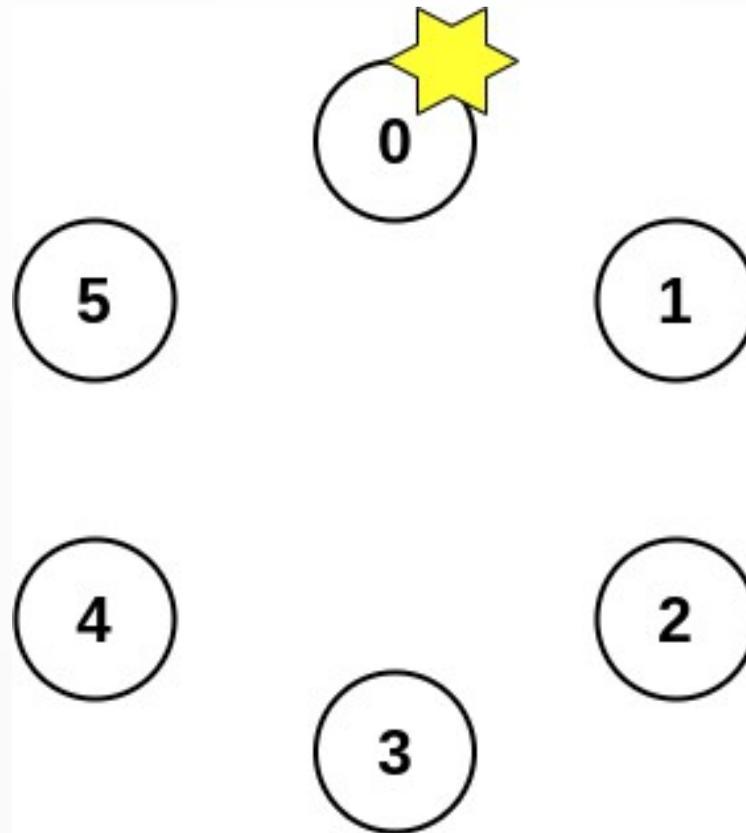
# VRing: Latência N Rodadas

- Acontece uma rodada de testes



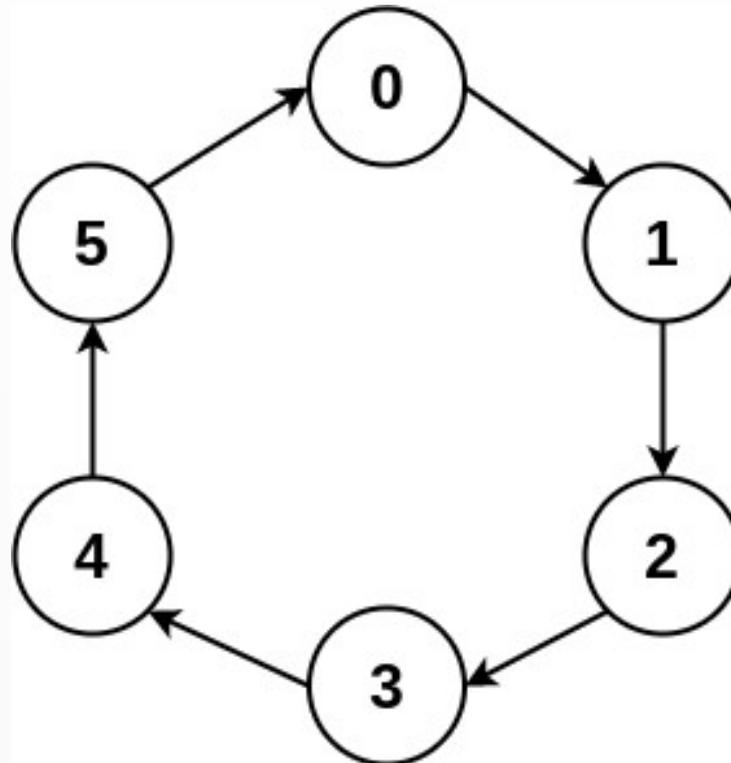
# VRing: Latência N Rodadas

- Um evento ocorre no processo 0



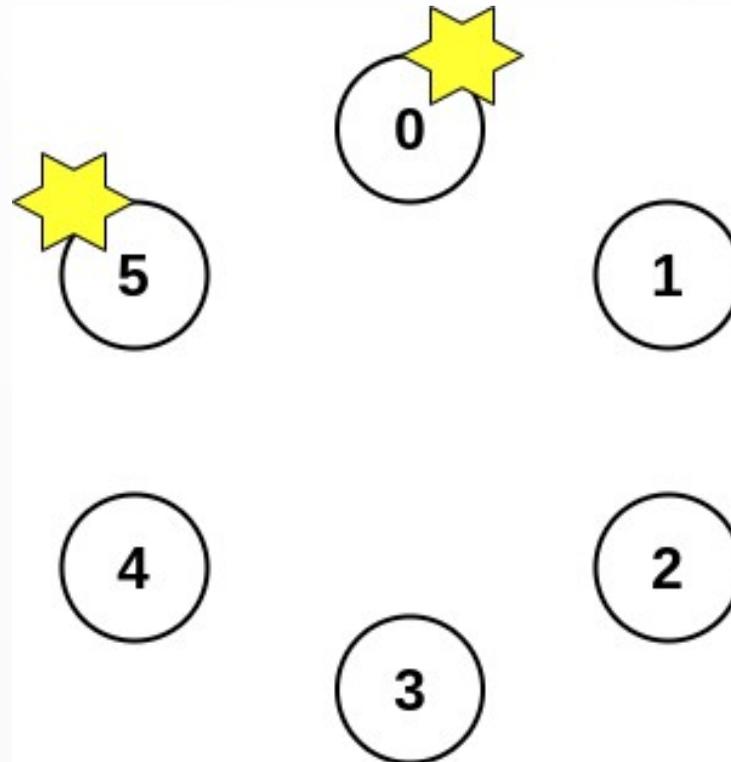
# VRing: Latência N Rodadas

- Acontece uma rodada de testes



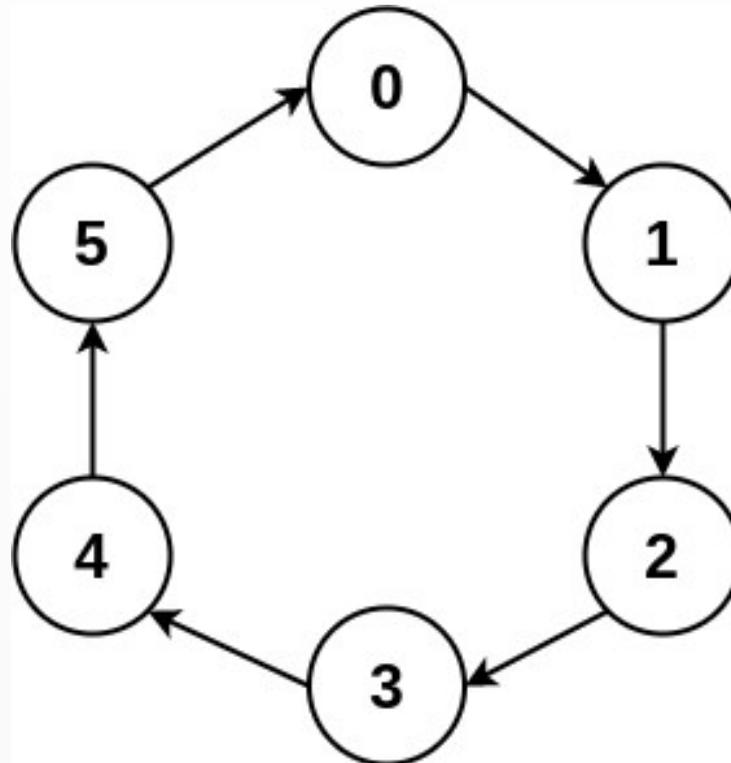
# VRing: Latência N Rodadas

- O processo 5 detecta o evento



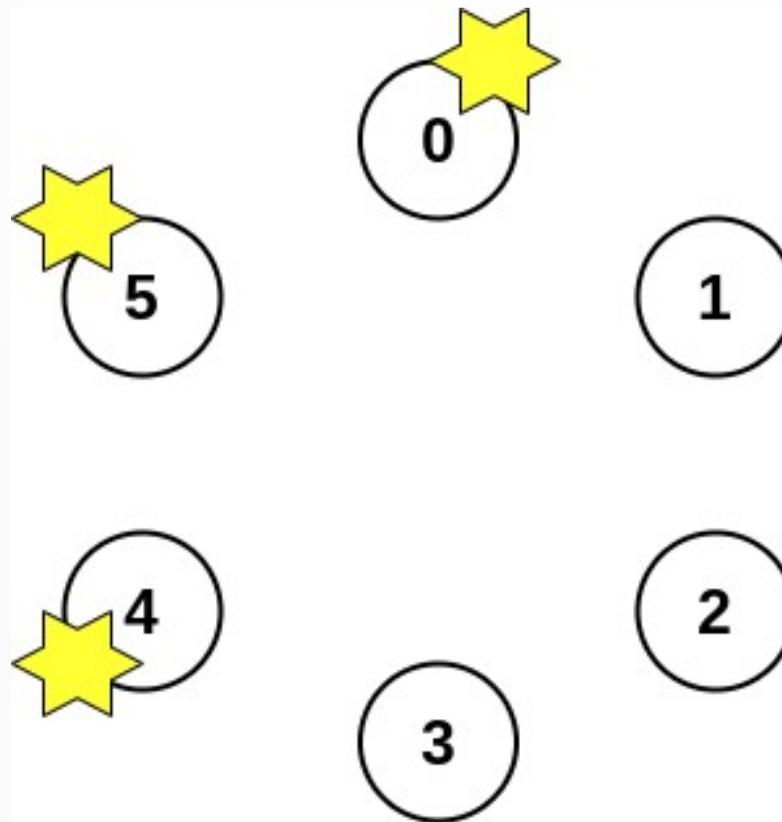
# VRing: Latência N Rodadas

- Acontece uma rodada de testes



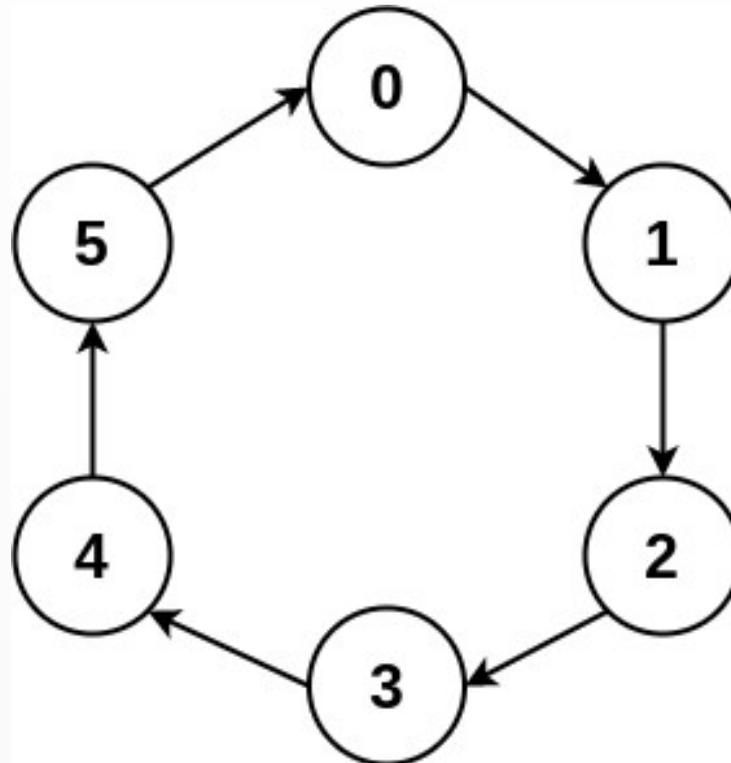
# VRing: Latência N Rodadas

- O processo 4 descobre o evento



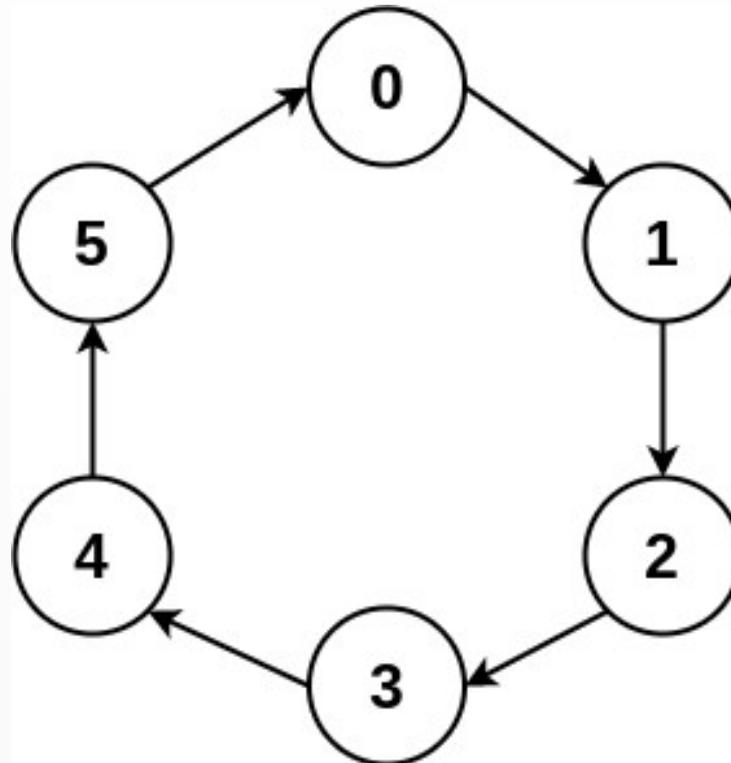
# VRing: Latência N Rodadas

- Acontece uma rodada de testes



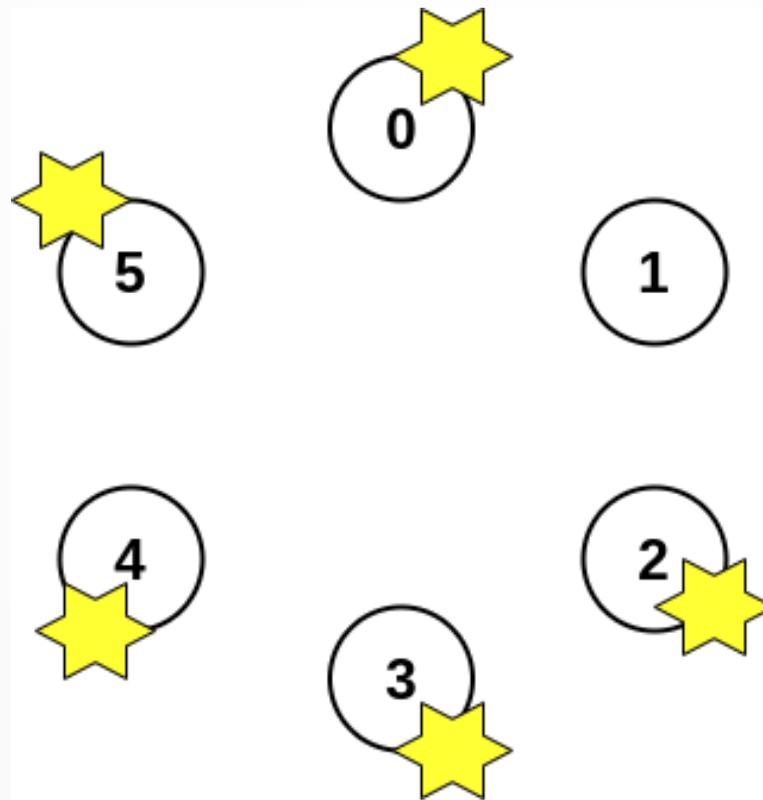
# VRing: Latência N Rodadas

- Acontece uma rodada de testes



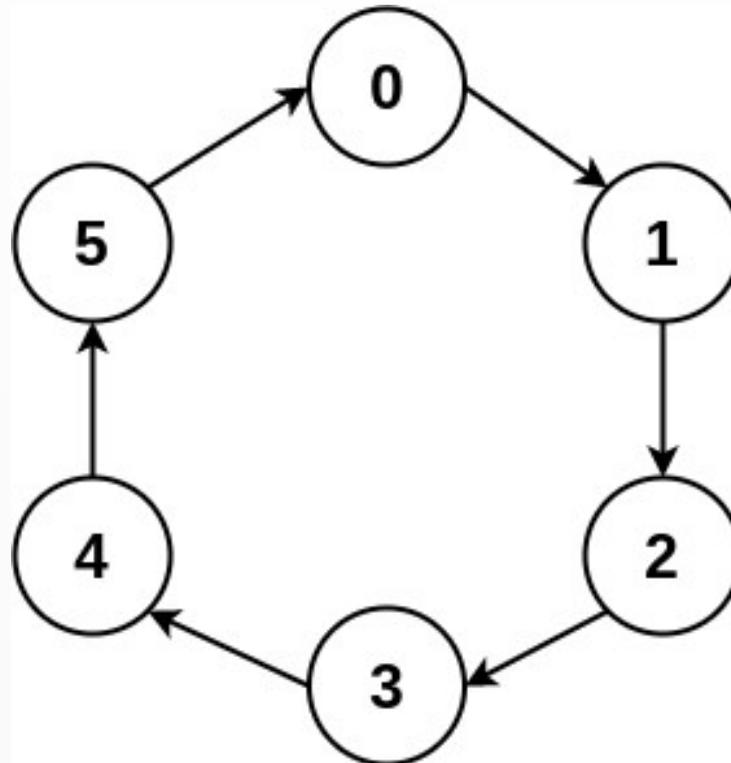
# VRing: Latência N Rodadas

- O processo 2 descobre o evento



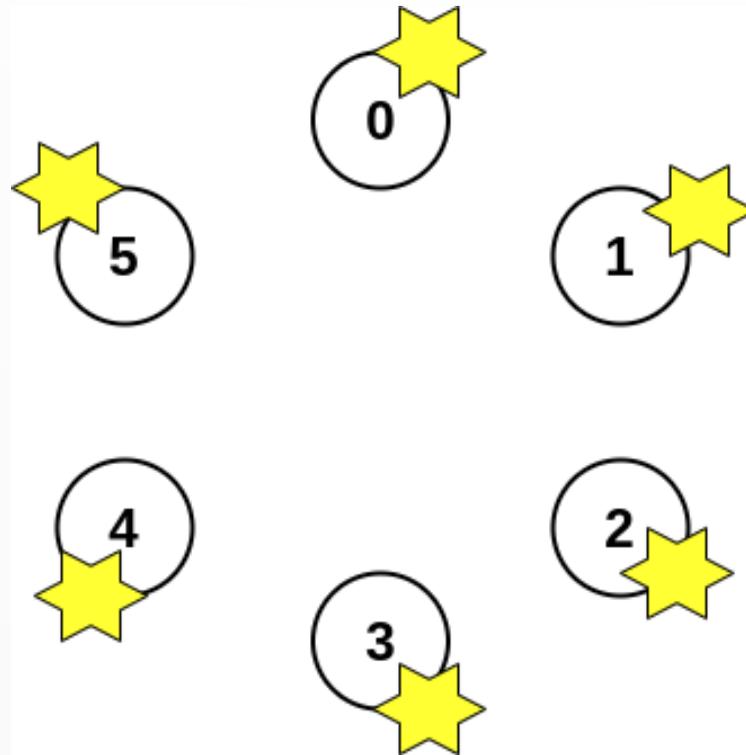
# VRing: Latência N Rodadas

- Acontece uma rodada de testes



# VRing: Latência N Rodadas

- Diagnóstico completo: todos os processos



# A Latência do Algoritmo VRing

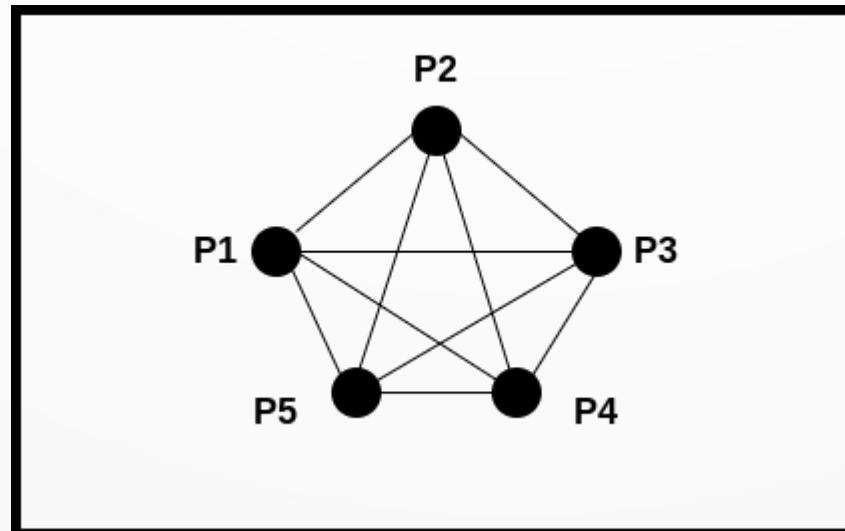
- A latência do algoritmo VRing é no pior caso de  $N$  rodadas de testes, sendo  $N$  o número total de processos
- Vamos supor que um intervalo de testes seja de  $t=30s$ , e que  $N=100$
- Latência pode chegar a  $30*100=3000s=50min$  :-0
- Como diminuir esta latência?

# Estratégias Orientadas a Evento

- Ao ser detectado um evento: dispara a disseminação de informação...
- ... que deve chegar a todos os processos corretos
- No artigo original estratégias assim são discutidas, mas...
- Transmitir 1 mensagem para ser entregue por todos os processos corretos, mesmo que haja falhas é *Reliable Broadcast* (Difusão Confiável)
- Custo  $O(N^2)$  mensagens

# O Algoritmo da Força Bruta

- No VRing os processos estão organizados em anel
- E como seria o algoritmo distribuído da força bruta?
- Todos os processos testam todos os demais em cada intervalo de testes



# O Algoritmo da Força Bruta

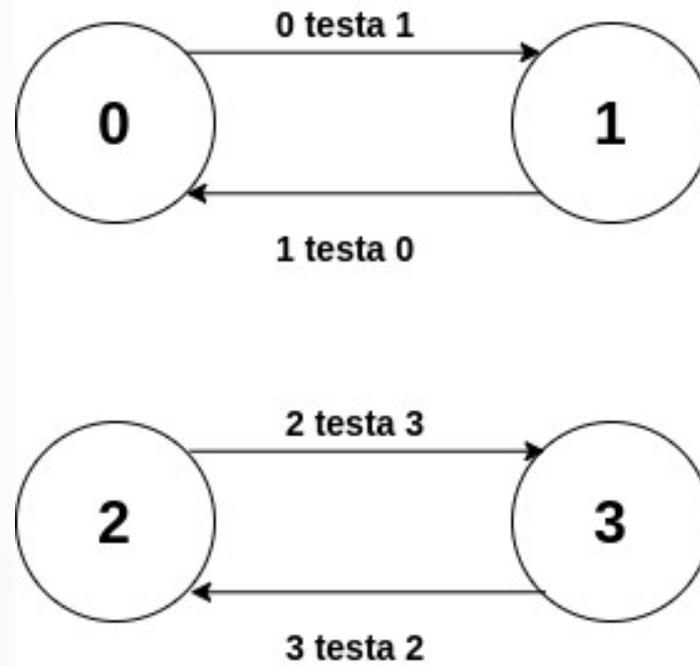
- Qual o número de testes executados?
- Resposta:  $O(N^2)$  testes por rodada
- Mas... a latência é de...
- Apenas 1 rodada de testes no pior caso!
- A latência é ótima (a melhor possível!) mas e o número de testes... é muito caro?
- Resposta com 1 exemplo: se  $N=100$  o número de mensagens (testes) é 10.000 (dez mil!)
- Super caro!

# O Algoritmo VCube

- Uma alternativa é usar um algoritmo distribuído que seja escalável
- “Entre” o VRing e a Força-Bruta:
  - VRing:  $N$  testes por rodada, latência  $N$  rodadas
  - Força-Bruta:  $N^2$  testes por rodada, latência 1 rodada
- O VCube é um algoritmo hierárquico, escalável por definição

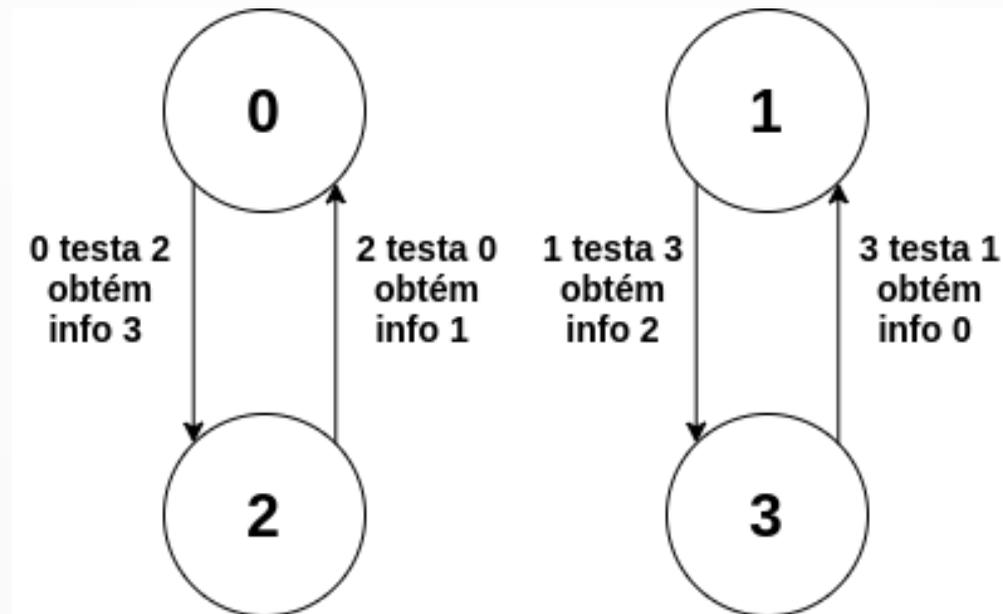
# VCube: Estratégia de Testes

- Na primeira rodada de testes



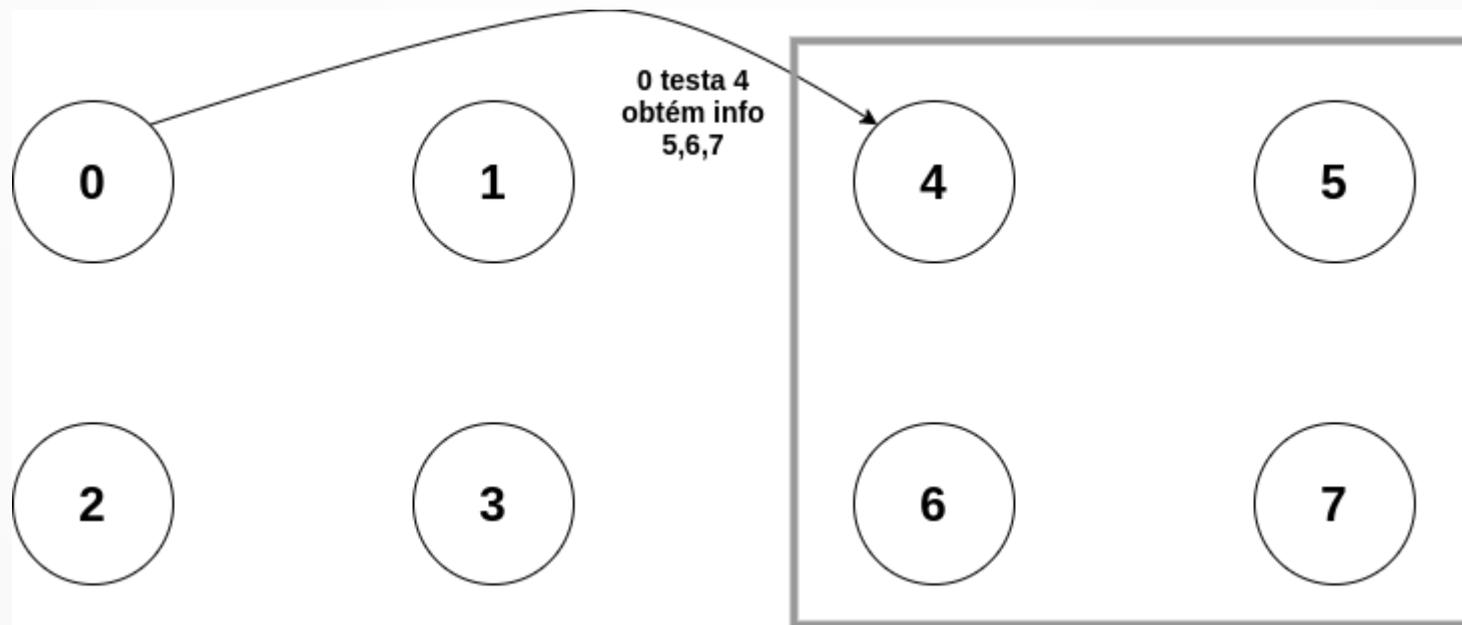
# VCube: Estratégia de Testes

- Neste exemplo, na segunda rodada de testes todos os processos corretos têm informações completas



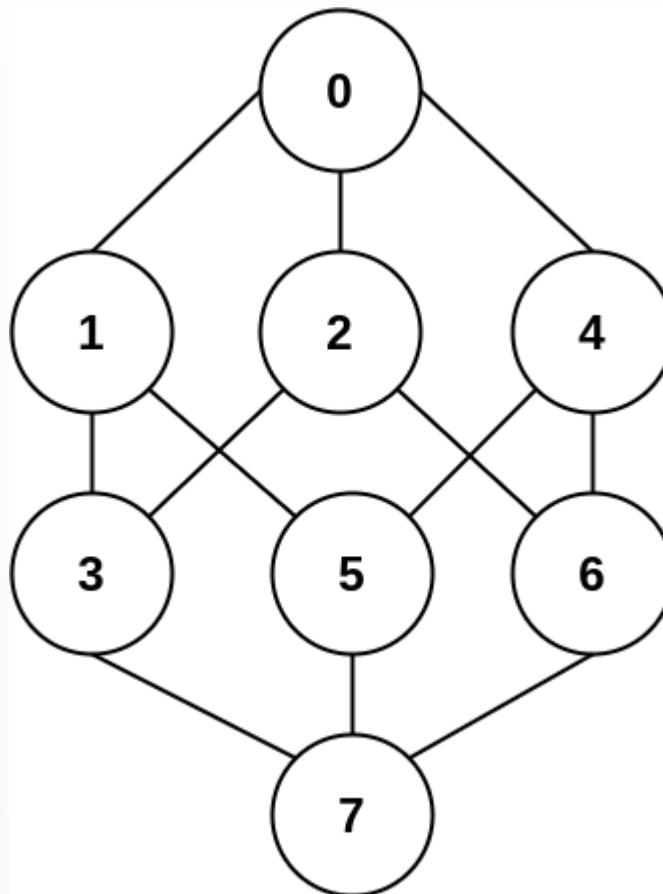
# VCube: Estratégia de Testes

- Na terceira rodada de testes: podemos dobrar os nodos para 8 e completam o diagnóstico



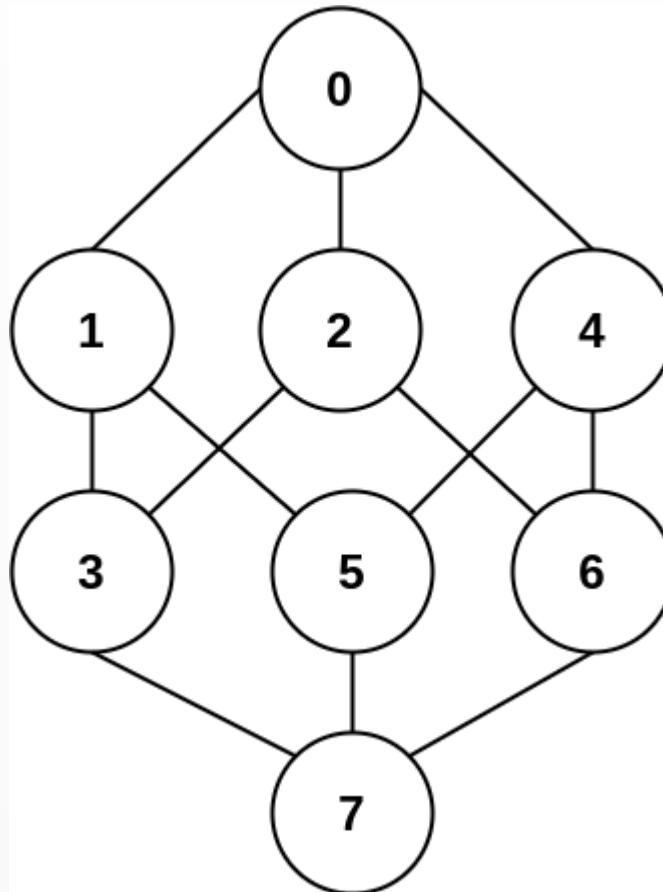
# VCube: Estratégia de Testes

- Na terceira rodada de testes: podemos dobrar os nodos para 8 e completam o diagnóstico



# Topologia Hipercubo

- Este é um hipercubo formado por 8 processos
- Consegue ver o cubo?



# O Hipercubo

- Em um hipercubo de  $d$  dimensões há  $N=2^d$  processos
- $d$  é o número de bits necessário para identificar os  $N$  processos
- Existe uma aresta entre dois vértices  $(i,j)$  se os identificadores de  $i$  e  $j$  diferem em exatamente 1 bit

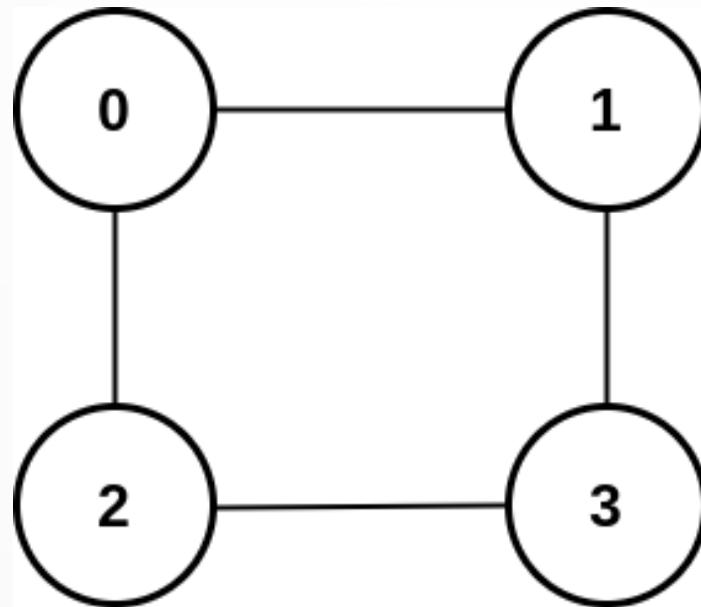
# Hipercubo de 1 Dimensão

- $N=2^1=2$  processos



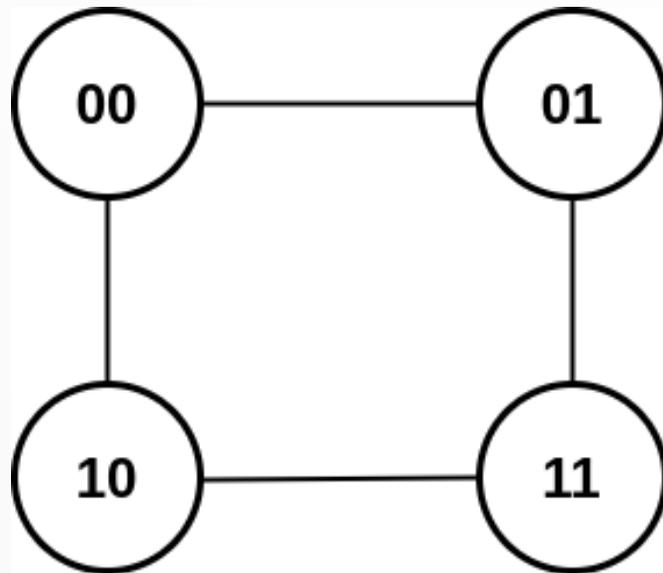
# Hipercubo de 2 Dimensões

- $N=2^2=4$  processos



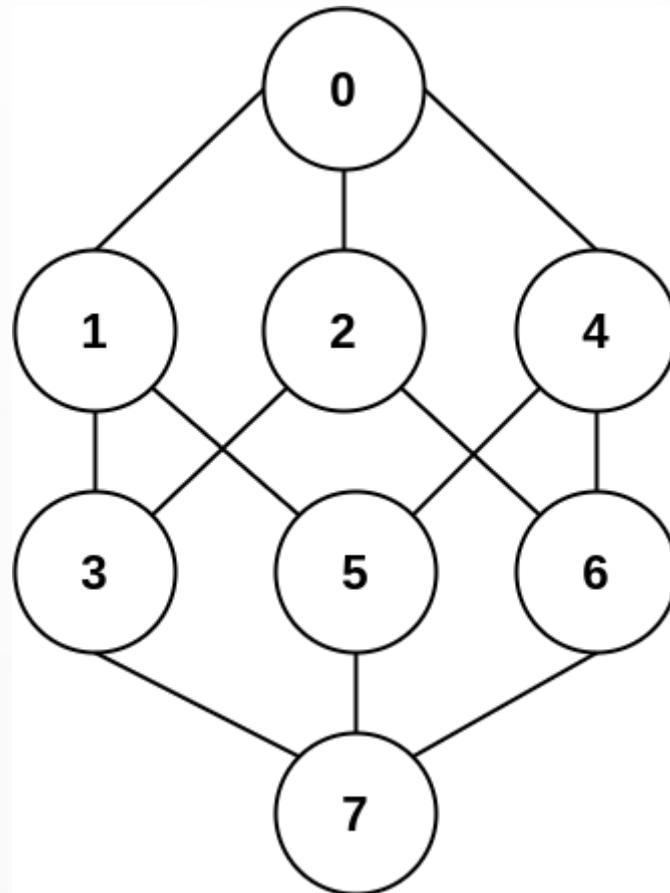
# Hipercubo de 2 Dimensões

- Agora mostrando os identificadores em binário
- Lembre-se só há aresta se os identificadores dos vértices diferem em exatamente 1 bit



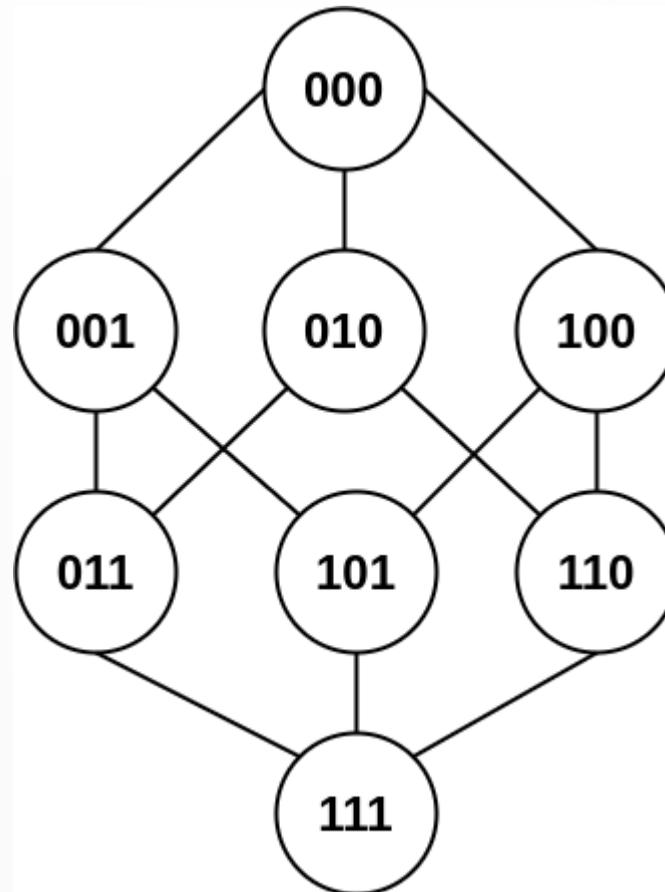
# Hipercubo de 3 Dimensões

- $N=2^3=8$  processos



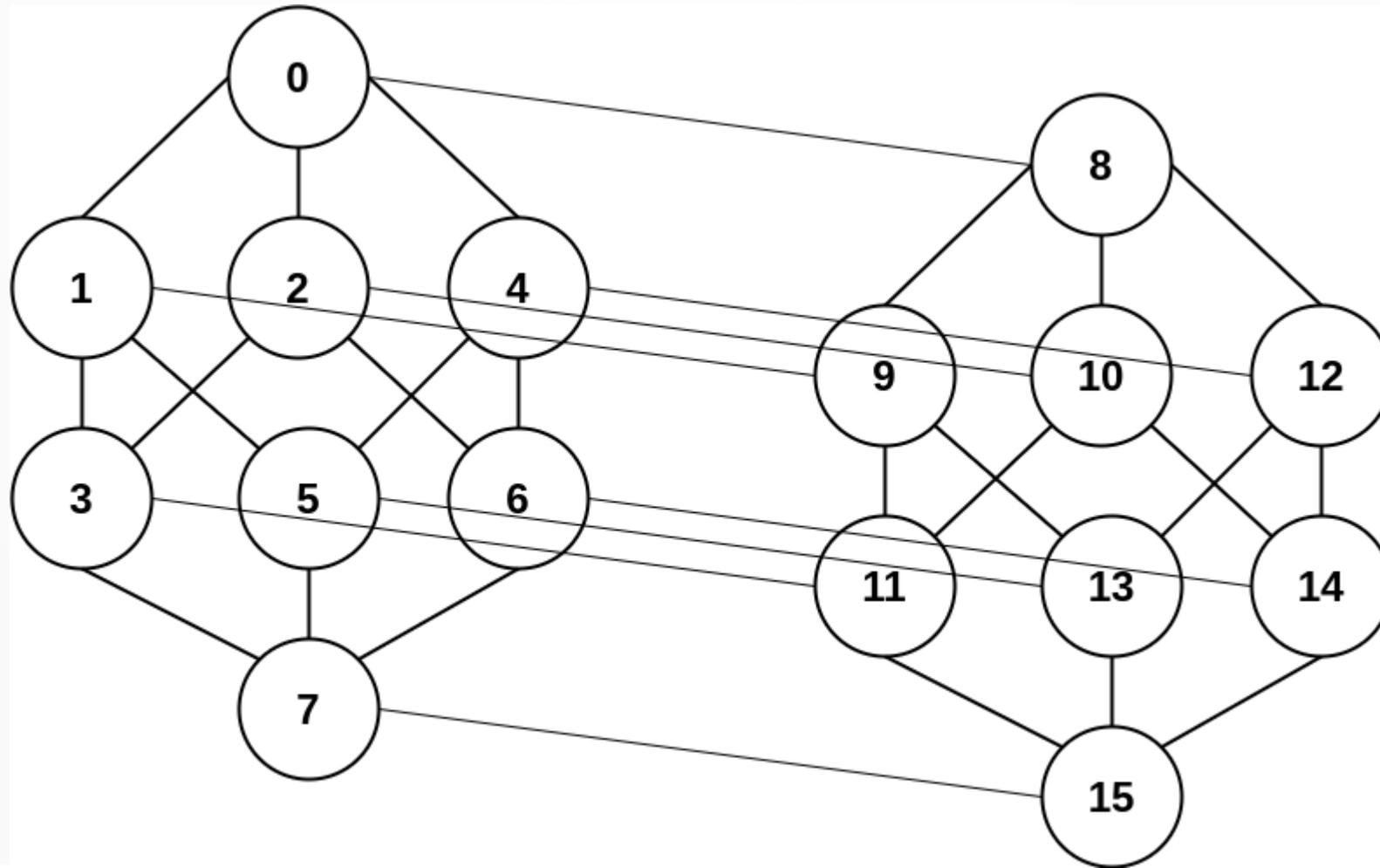
# Hipercubo de 3 Dimensões

- Agora mostrando os identificadores em binário



# Hipercubo de 4 Dimensões

- Veja que dobramos o número de nodos, são dois hipercubos de 8 nodos



# Hipercubo: Topologia Escalável

- O hipercubo têm diversas propriedades logarítmicas
  - $\log N$  – logaritmos base 2, sempre
- Grau de cada nodo, diâmetro (distância máxima)
- Quando dobra o número de nodos, o incremento é de 1 unidade
- Propriedades logarítmicas correspondem a escalabilidade: funciona bem quando N cresce

# VCube: Muito Mais que Hipercubo

- O hipercubo foi muito usado para a construção de arquiteturas paralelas, entre outras aplicações computacionais
- VCube: corresponde a um hipercubo virtual quando todos os processos estão corretos
- Topologia subjacente: *fully-connected*
- VCube: quando processos falham a topologia se reorganiza, mantendo diversas propriedades logarítmicas

# VCube: Clusters de Processos

- No VCube o processos são organizados em clusters sob o ponto de vista do testador
- O tamanho do cluster é sempre uma potência de 2, há  $\log N$  clusters
- A função  $C(i,s)$  retorna a lista de processos (em ordem) que devem ser testados pelo testador  $i$  nos clusters  $s=1,2,\dots,\log N$

# A Função $C(i,s)$ , $N=8 \rightarrow$ Completar!

$s$	$C(0,s)$	$C(1,s)$	$C(2,s)$	$C(3,s)$	$C(4,s)$	$C(5,s)$	$C(6,s)$	$C(7,s)$
1	1			2				6
2	2,3		0,1				4,5	
3	4,5,6,7				0,1,2,3			

# A Função $C(i,s)$ , $N=8$ processos

$s$	$C(0,s)$	$C(1,s)$	$C(2,s)$	$C(3,s)$	$C(4,s)$	$C(5,s)$	$C(6,s)$	$C(7,s)$
1	1	0	3	2	5	4	7	6
2	2,3	3,2	0,1	1,0	6,7	7,6	4,5	5,4
3	4,5,6,7	5,6,7,4	6,7,4,5	7,4,5,6	0,1,2,3	1,2,3,0	2,3,0,1	3,0,1,2

# O Algoritmo Distribuído VCube

- Processos executam testes periodicamente em intervalos de testes (p.ex. 30s ou 10ms)
- Cada processo  $i$  mantém o vetor local  $State_i[N]$
- Em cada intervalo, um processo testa um dos seus  $\log N$  clusters
- Testes são executados sequencialmente em cada cluster, até um processo correto ser testado
  - versão 1, vamos ter uma outra versão na próxima aula
- Ao testar um processo correto: testador obtém informações sobre todos os demais processos do cluster que não testou neste intervalo

Algoritmo VCube executado pelo processo i:

repita

para s de 1 até  $\log N$  e voltando a 1 faça

repita

execute um teste no próximo processo de  $C(i,s)$ ;

se o processo foi testado correto

então obtenha info cluster s e atualize  $State_i[N]$ ;

até ter testado um processo correto ou todos falhos;

durma até o próximo intervalo de testes;

fim-para;

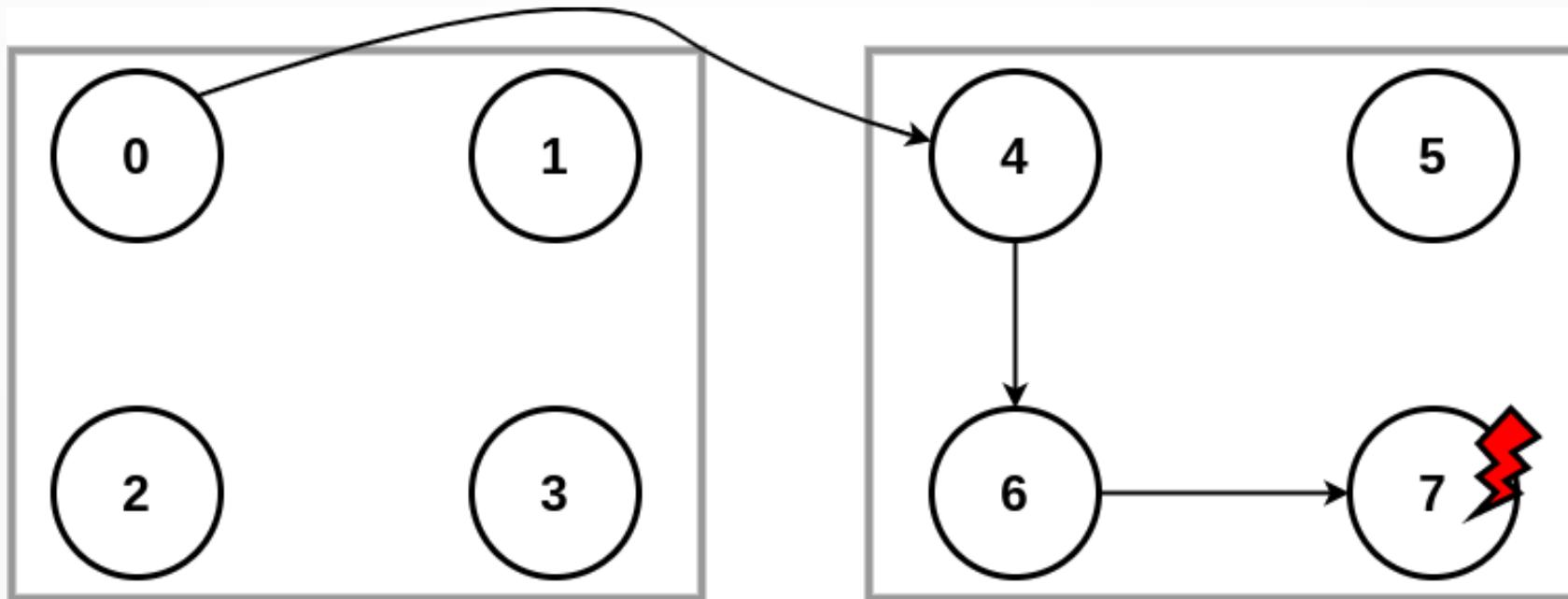
para-sempre;

# Propriedades do VCube: Latência e Número de Testes

- Qual a latência do VCube no pior caso?

# Propriedades do VCube: Latência e Número de Testes

- Qual a latência do VCube no pior caso?
- Não é  $\log N$ , mas  $\log^2 N$  rodadas de testes: próxima aula

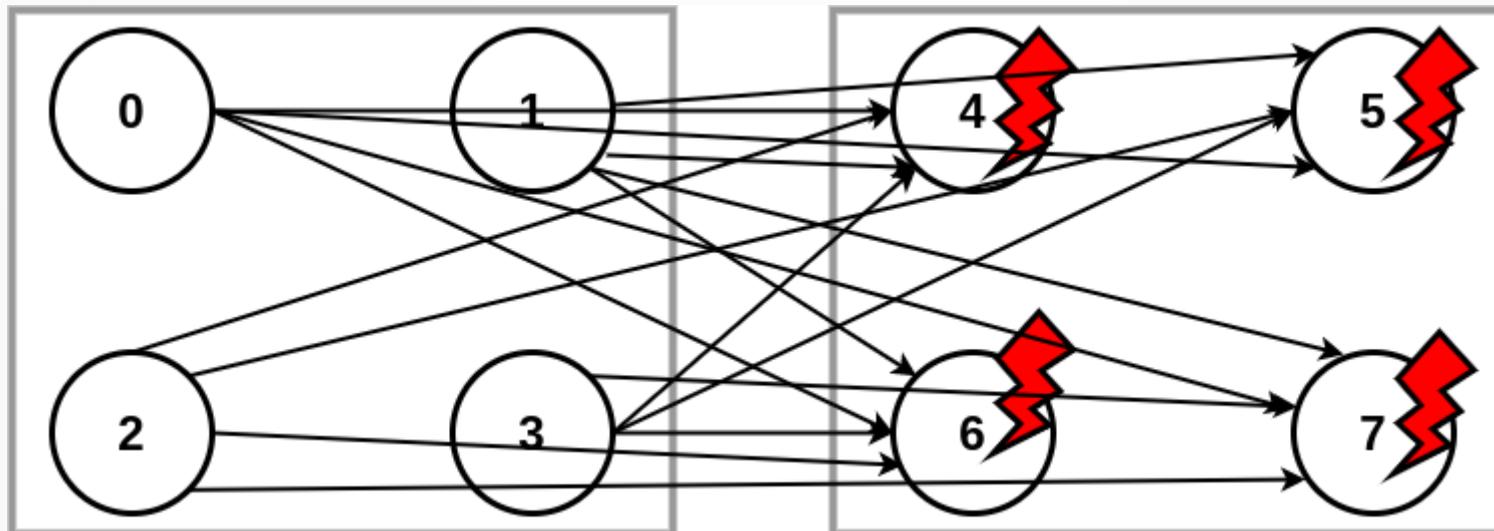


# Propriedades do VCube: Latência e Número de Testes

- Qual a latência do VCube no pior caso?
- Não é  $\log N$ , mas  $\log^2 N$  rodadas de testes: próxima aula
- Qual o número máximo de testes executados?

# Propriedades do VCube: Latência e Número de Testes

- Qual a latência do VCube no pior caso?
- Não é  $\log N$  e sim  $\log^2 N$ : próxima aula
- Qual o número máximo de testes executados?
- Considere que todos os processos do maior cluster (com  $N/2$  processos) estão falhos
- Quantos testes executa?



# Propriedades do VCube: Latência e Número de Testes

- Qual a latência do VCube no pior caso?
- Não é  $\log N$  e sim  $\log^2 N$ : próxima aula
- Qual o número máximo de testes executados?
- Considere que todos os processos do maior cluster (com  $N/2$  processos) estão falhos
- Quantos testes executa?
- Os  $N/2$  testadores testam todos os  $N/2$  falhos:  $N^2/4 = O(N^2)$
- Caso especial, mas este é o número de testes da força-bruta... por isso outra estratégia foi definida: próxima aula

## **Referência original do VCube:**

“A Hierarchical Adaptive Distributed System-Level Diagnosis Algorithm,” *IEEE Transactions on Computers*, Vol. 47, No. 1, 1998.

# Conclusão

- Nesta aula estudamos o algoritmo distribuído VCube, que tem menor latência no pior caso do que o VRing
- Vimos outra alternativa: força-bruta, não escalável, diversas propriedades quadráticas
- VCube: escalável, diversas propriedades logarítmicas
- Vimos também que há alternativas orientadas a evento, reliable broadcast, em janeiro!

**Obrigado!**  
**Página da Disciplina**  
**Sistemas Distribuídos:**  
**[www.inf.ufpr.br/elias/sisdis](http://www.inf.ufpr.br/elias/sisdis)**