

Fundamentos de TF

Redundância

Taisy Silva Weber
UFRGS

Redundância

- redundância
 - de hardware
 - de software
 - de informação
 - no tempo
- impacto no sistema (custo):
 - desempenho, ou área (tamanho, peso), ou potência consumida, ...

toda redundância tem custo

Redundância

- de hardware { passiva
ativa
híbrida
- de software
- de informação
- no tempo

Johnson, B.W. *Fault Tolerance*, The Electrical Engineering Handbook, Ed. Richard C. Dorf, Boca Raton: CRC Press LLC, 2000

Redundância de hardware

- redundância **passiva** (ou **estática**)

- marcar falhas
- não requer ação do sistema
- não indica falha

- redundância **ativa** (ou **dinâmica**)

envolve detecção e recuperação

- redundância **híbrida**

- combinação das anteriores
 - mascaramento e longa vida
 - geralmente de alto custo

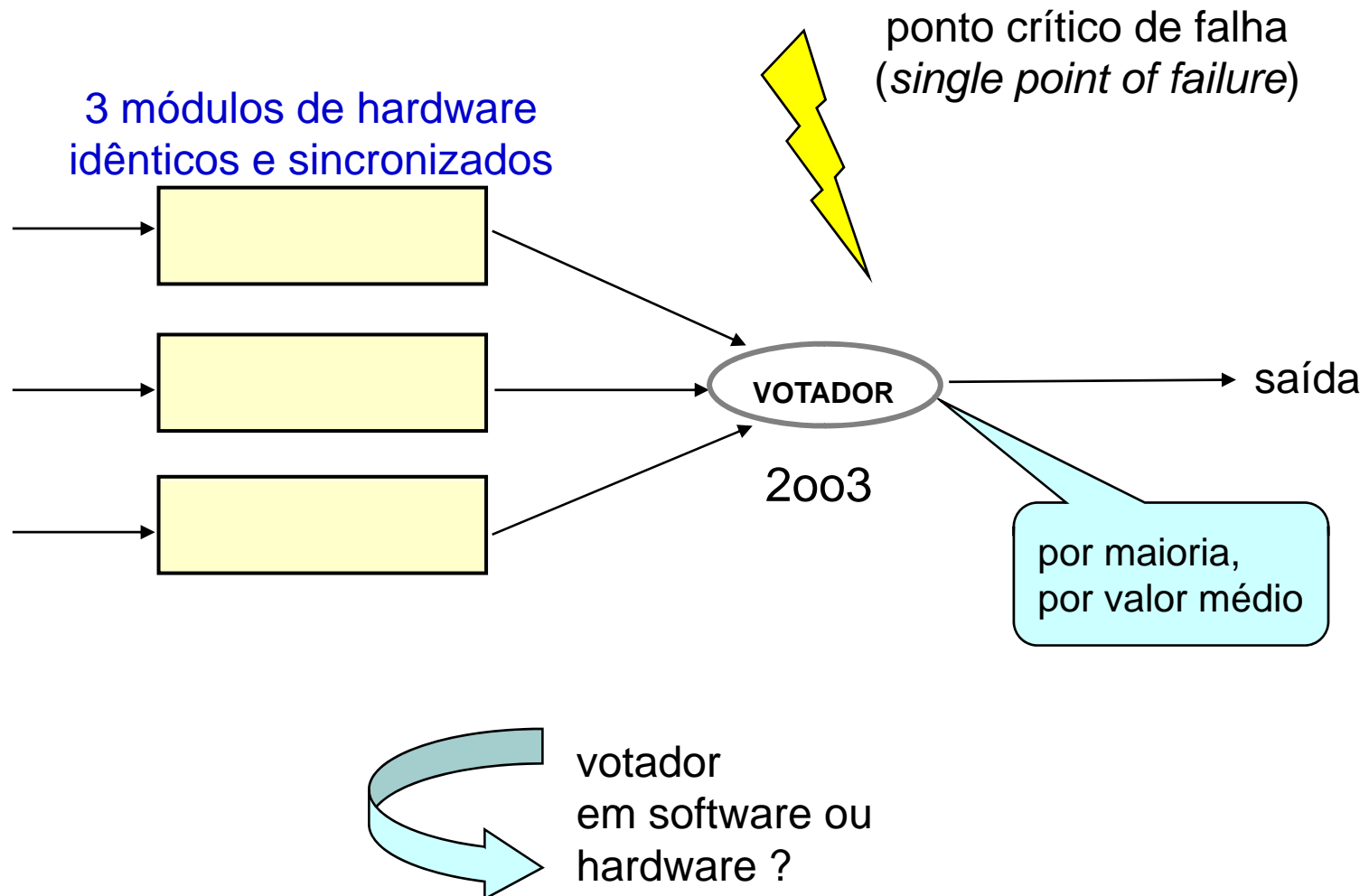
TMR

- redundância modular tripla
 - redundância de hardware **passiva**
 - mascara falha em **um** componente
 - votação
 - votação por maioria (2 em 1)
 - votação por seleção do valor médio
 - sinal selecionado reside no meio dos outros dois

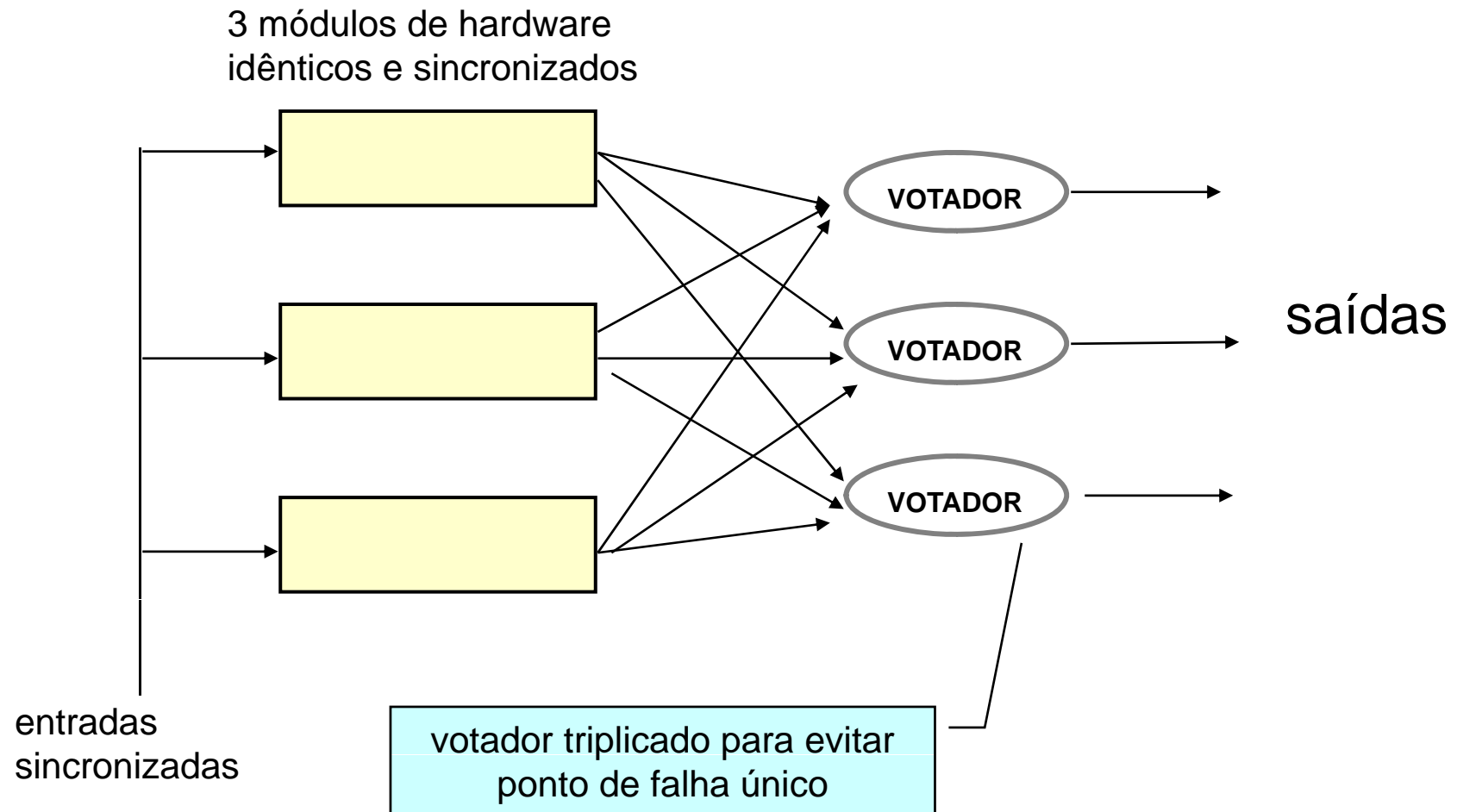


mais usada em sistemas críticos,
mas aparece também em sistema de alta
disponibilidade como alguns servidores de rede

TMR: redundância modular tripla



TMR com 3 votadores

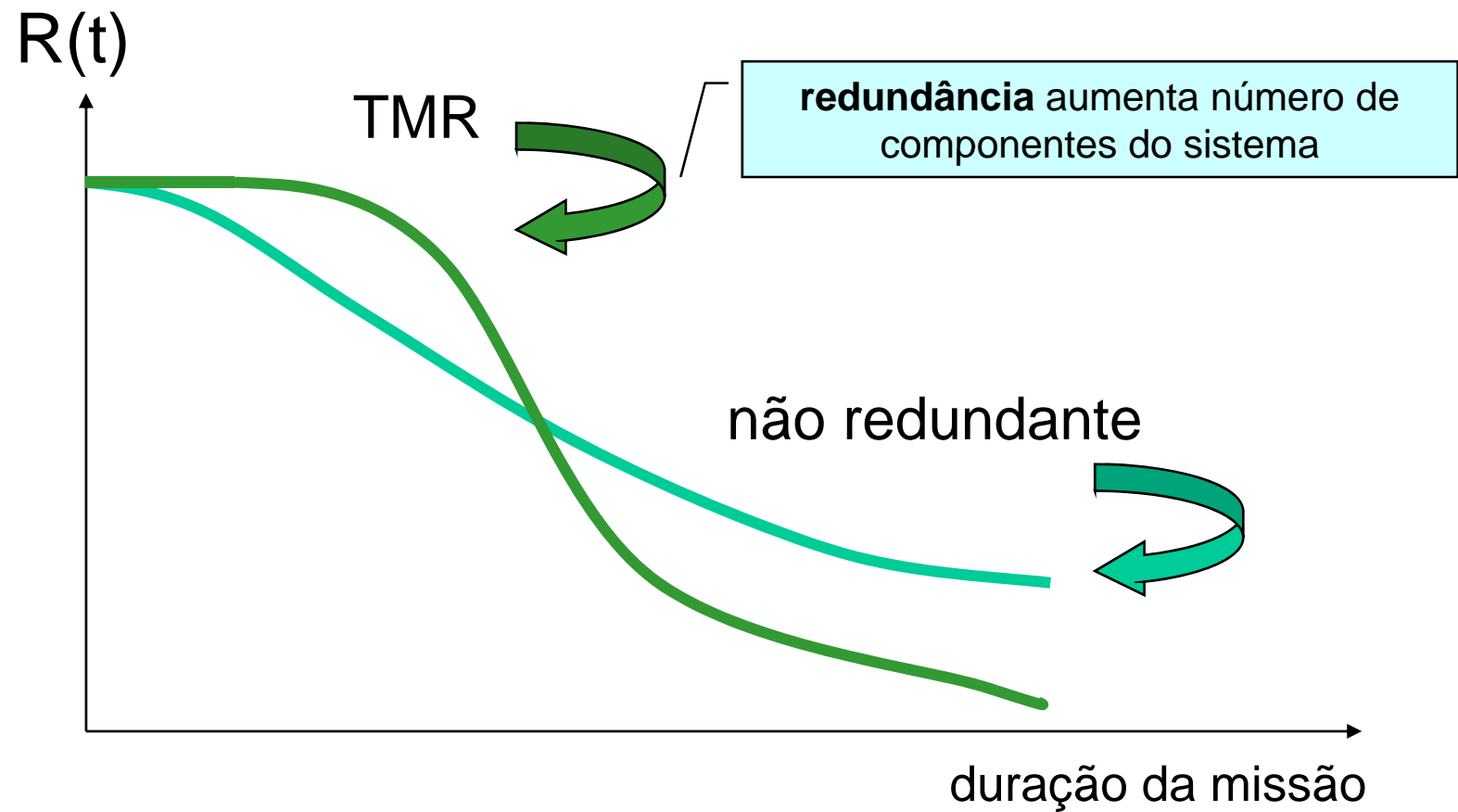


Confiabilidade de TMR

- redundância aumenta número de componentes do sistema
 - maior número de componentes, maior probabilidade de falha
- ideal para períodos não muito longos de missão
 - suporta falhas permanentes em um dos componentes (exceto votador) mas é ideal para falhas temporárias

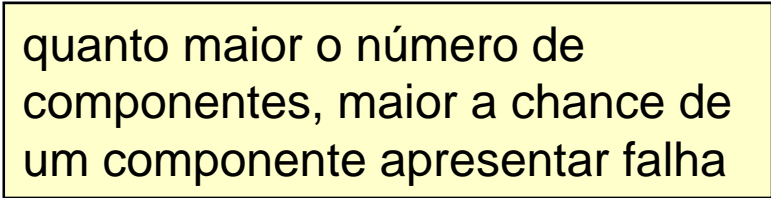
pergunta: faz sentido usar TMR para detectar erros de software (triplicando um processo)?

Confiabilidade de TMR



NMR

- redundância modular múltipla - NMR
 - são usados N elementos
 - TMR = caso especial de NMR
 - idealmente número ímpar de componentes
 - N é ímpar para evitar empate
 - dificuldade:
 - determinação do N
 - custo, área, potência e até confiabilidade dos componentes



quanto maior o número de componentes, maior a chance de um componente apresentar falha

Redundância dinâmica (ou ativa)

- detecção e recuperação

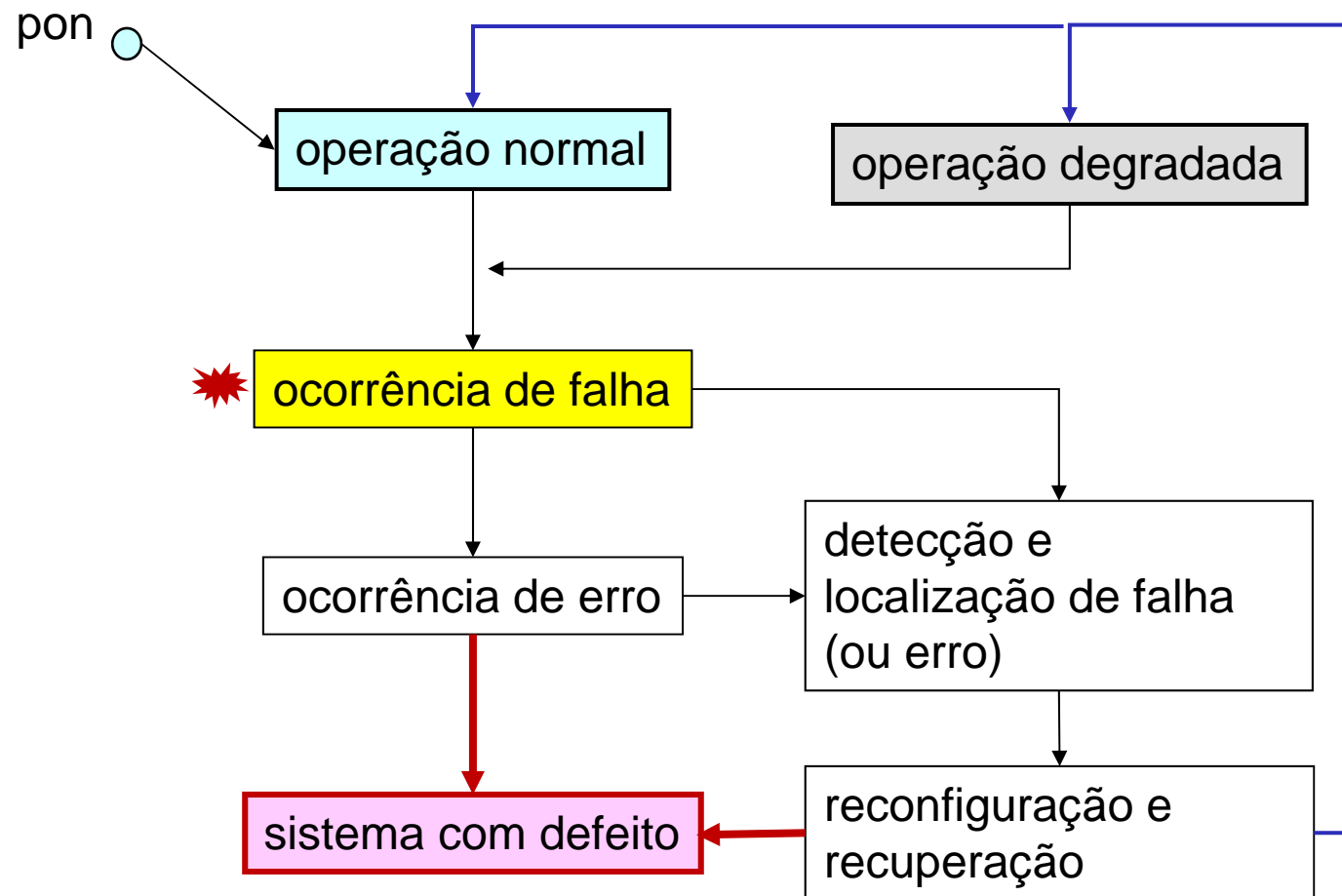
- sem mascaramento

detectar e recuperar leva um certo tempo

- aplicações

- sistemas que toleram resultados errados durante um curto período de tempo
 - exemplos:
 - manutenção adiada (satélites)
 - alta disponibilidade

Estados na redundância ativa



Johnson, Barry. An introduction to the design and analysis of the fault-tolerant systems, cap 1. **Fault-Tolerant System Design**. Prentice Hall, New Jersey, 1996

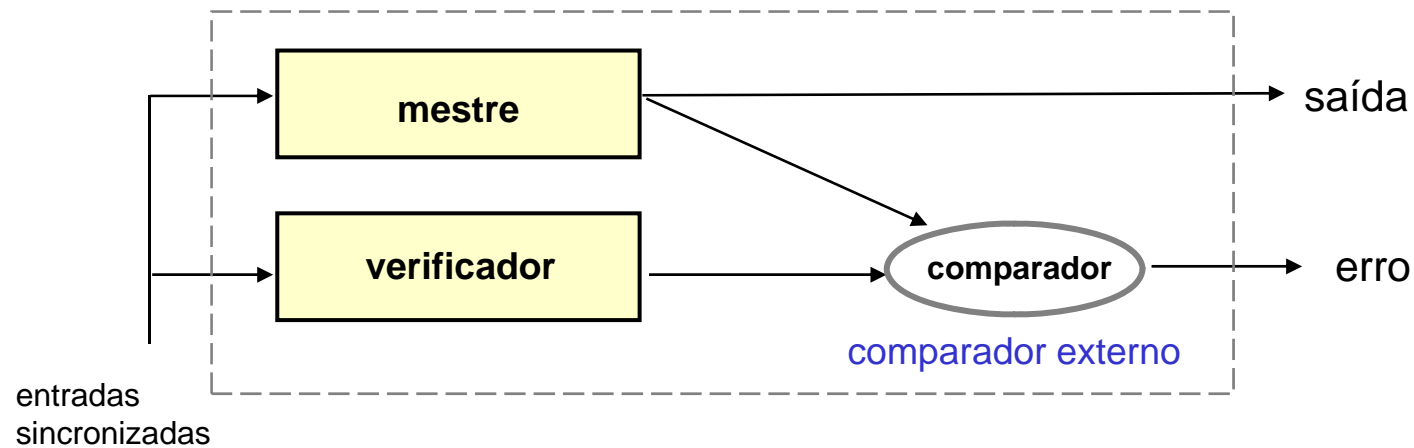
Exemplos de redundância dinâmica

- *standby sparing*
 - módulo operacional vs. módulos estepe
- 2 tipos:

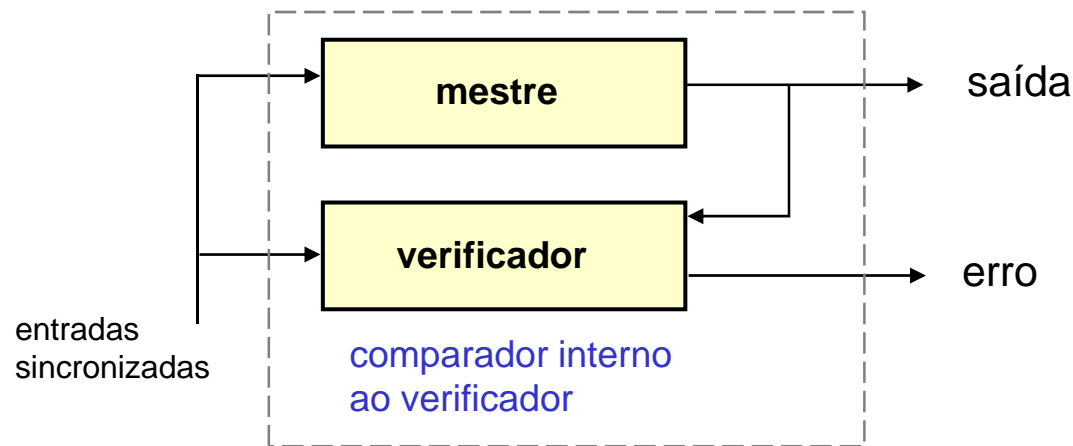
alimentados = ligados (com energia elétrica)

 - alimentados (*hot standby*)
 - minimiza a descontinuidade do processamento durante chaveamento entre os módulos
 - todos os componentes sofrem envelhecimento
 - não alimentados (*cold standby*)
 - aumenta a vida útil dos módulos
 - estepes (backups) só começam a operar após conectados e inicializados

Componentes auto verificadores



Nelson, V. – Fault Tolerant Computing: Fundamental Concepts, IEEE Computer, 1990, pp 19-25

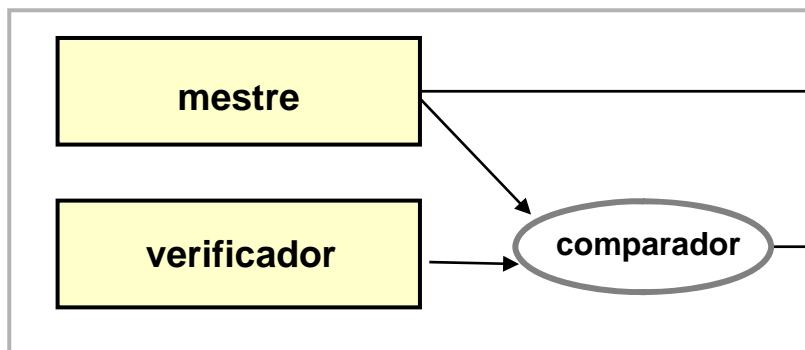


signal de erro
permite
implementar
redundância ativa

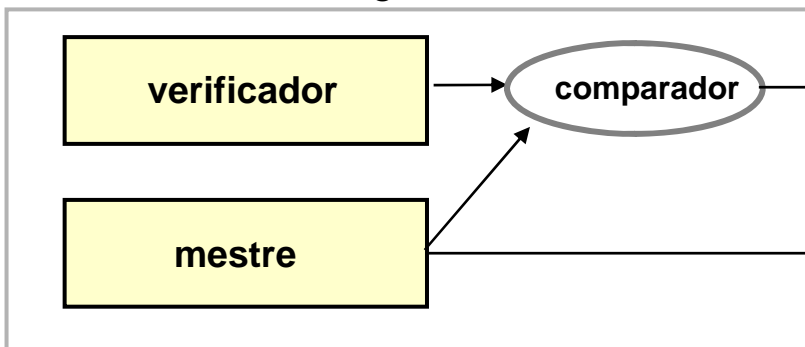
Operação contínua

com componentes auto verificadores

módulo self-cheking A



módulo self-cheking B



saída

erro

chave

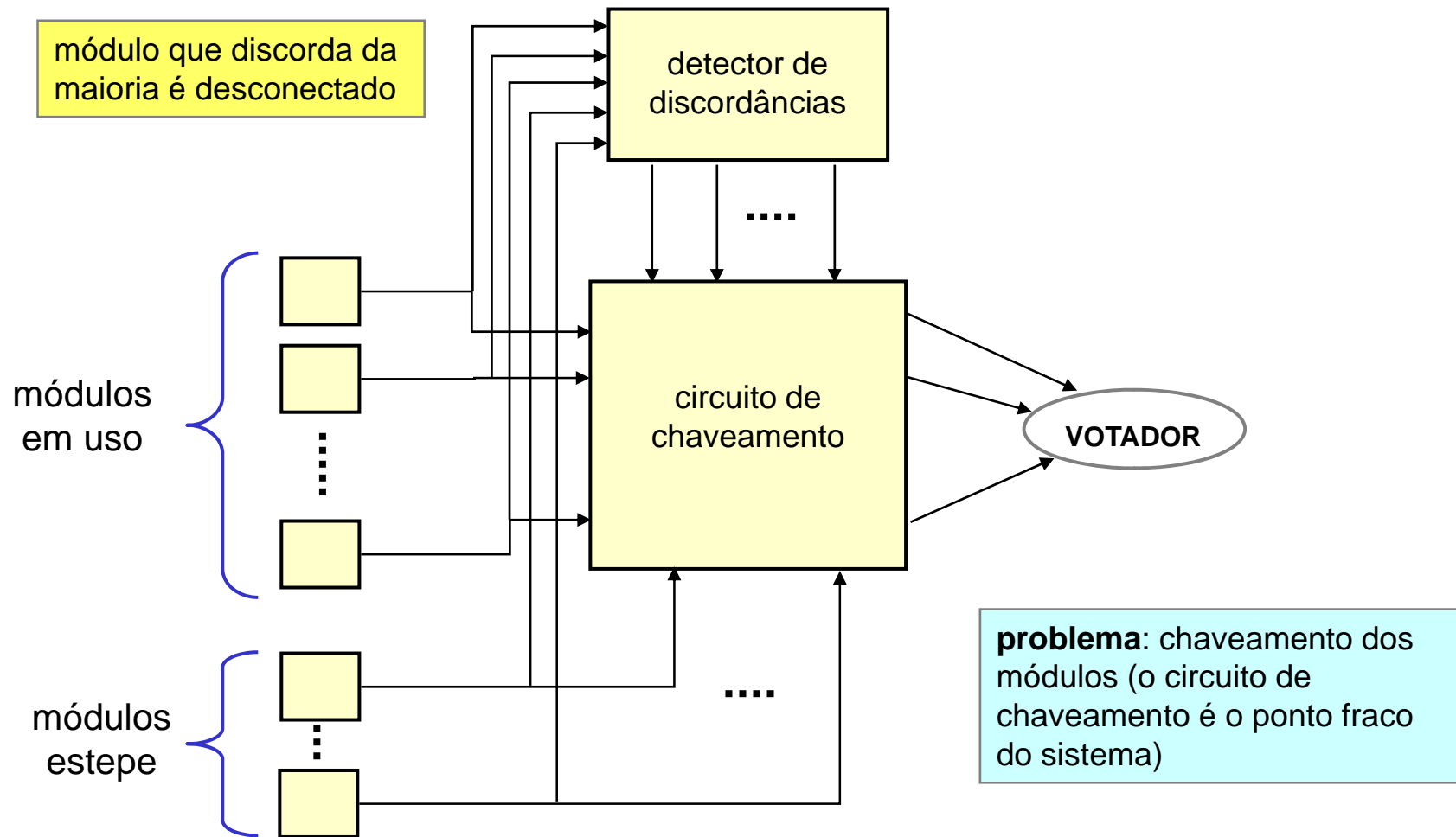
saída
contínua

erro

saída

Nelson, V. – Fault Tolerant Computing:
Fundamental Concepts, IEEE Computer,
1990, pp 19-25

Redundância híbrida



estepe substitui módulo com erro


Redundância auto-eliminadora

- caso especial de redundância híbrida
 - também baseada em votação
 - mas módulo que discorda da maioria é isolado
 - não existe estepe
 - todos os módulos estão desde o início da operação conectados ao sistema

todos os módulos sofrem igualmente os efeitos do envelhecimento

Redundância

- de hardware
- de software
- de informação
- no tempo



também são formas
comuns de redundância

sem componentes físicos replicados
(menor número de componentes, menor a
probabilidade de defeito devido a falhas
em um componente)

Redundância de software

para **tolerar erros** de software

replicação de componentes de software idênticos
é útil apenas para falhas evasivas

- exemplos:


- diversidade
 - de versões de programas
 - de projeto
 - de especificação
 - de implementação
- blocos de recuperação
- verificação de consistência

programação n-versões

programação
diversitária

Programação n-versões

- versões diferentes de um mesmo programa
 - votação na saída
 - erros devem se manifestar de forma diferente nas versões



correlação não permitida

- vantagens
 - facilidade de reconhecer erros na fase de teste
 - tolerância a falhas permanentes e intermitentes
 - tolerância a falhas externas
 - do compilador e do sistema operacional

Desvantagens

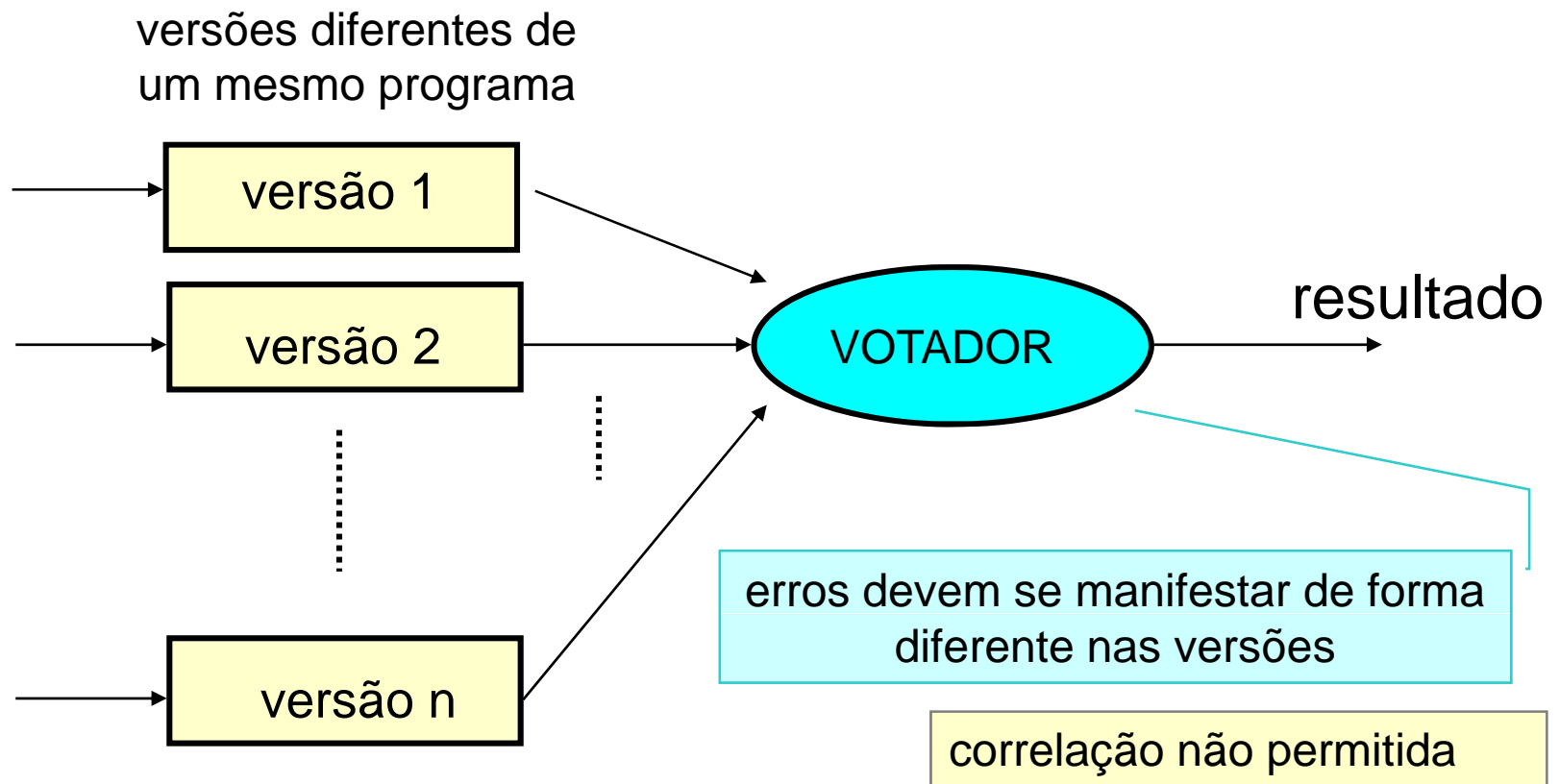
- aumento dos custos de desenvolvimento
- complexidade de sincronização
- dificuldade de evitar correlação
 - não há garantias que cópias diferentes não apresentem os mesmos erros
- falta prova formal de aumentar confiabilidade

experimentos comprovam o aumento de confiabilidade, por essa razão diversidade tem sido usada em projetos críticos

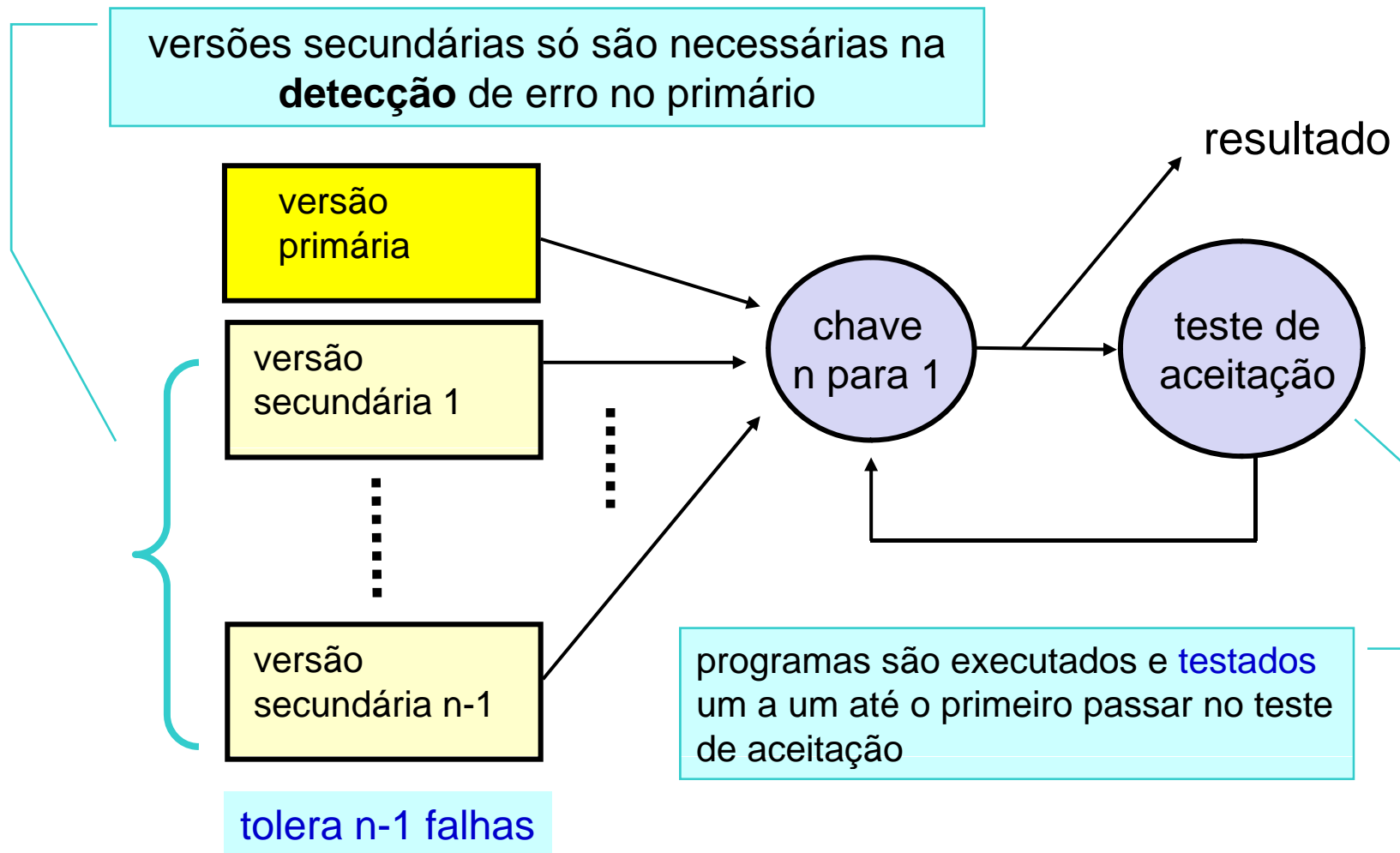
- dificuldade de determinar o número de réplicas

Modelo

Avizienis



Blocos de recuperação



Atenção

se o software fosse correto,
então não seriam necessárias técnicas de tolerância a falhas para software


tolerância a falhas não substitui
boas e tradicionais técnicas de
programação e desenvolvimento de
software

Redundância temporal

- repete a computação no tempo
 - a computação é realizada duas ou mais vezes no mesmo equipamento
- evita custo de hardware adicional
 - mas há custo de desempenho
- aplicações usuais:
 - detecção de falhas transientes
 - falhas transientes provavelmente não afetam igualmente duas computações separadas no tempo
 - detecção de falhas permanentes

Detecção de falhas permanentes

- tempo t_0
 - computação normal
 - armazena resultado
- tempo t_1
 - codifica dado
 - computação
 - decodifica dado
 - armazena resultado
 - compara resultados



codificação força uso
diferente do hardware



indicação de erro

Redundância de informação

- códigos de detecção de erros
 - capacidade de indicar que a informação está incorreta (ou inconsistente)
- códigos de correção de erros
 - capacidade de correção e detecção
 - correção realizada por hardware
- qualquer código implica no aumento do número de bits

Exemplo: ECC

redundância

Bibliografia para redundância

- capítulo de livro
 - Johnson, Barry. An introduction to the design na analysis of the fault-tolerante systems, cap 1. **Fault-Tolerant System Design**. Prentice Hall, New Jersey, 1996
- artigos
 - Nelson, V. – **Fault Tolerant Computing: Fundamental Concepts**, IEEE Computer, 1990
 - biblioteca digital da IEEE
 - BEV LITTLEWOOD, PETER POPOV, LORENZO STRIGINI. **Modeling Software Design Diversity – A Review**. ACM Computing Surveys, Vol. 33, No. 2, June 2001, pp. 177–208.
 - biblioteca digital da ACM
 - VINCENZO DE FLORIO & CHRIS BLONDIA. **A Survey of Linguistic Structures for Application-Level Fault Tolerance**. ACM Computing Surveys, Vol. 40, No. 2, April 2008.
 - biblioteca digital da ACM