


1


Falhas aconcetem e
são inevitaveis!!!

2


Falhas inevitaveis, mas
suas consequências
podem ser amenizadas
ou evitadas

3


Falhas inevitaveis, mas
TOLERÂNCIA A FALHAS
podem ser amenizadas
ou evitadas

4

TOLERÂNCIA A FALHAS EM SISTEMAS DISTRIBUÍDOS

Marcelo Iury – marceloiury@gmail.com

Agenda

1. Introdução
2. Conceitos básicos e terminologia
   a. Falha, erro, defeito
   b. Tipos e modelos de falhas
3. Técnicas de tolerância a falhas
   a. Fases de Tolerância a Falhas
   b. Redundância (Hardware, Software, Informação e Tempo)
4. Conclusão
5. Exercícios

Introdução

7

A informática evoluiu, mas...

- Computadores e seus programas tornaram nossa vida mais fácil
  - Automatizaram e aceleraram a execução de tarefas enfadonhas e repetitivas
- Computadores e seus programas também falham
  - Dados perdidos
  - Sistema fora do ar
  - Computador travado
  - Em suma: deu pau! (Termo usado na tradução do Coulouris)

8

Luz no Fim do Tunel

- Aumentar confiabilidade dos componentes
  - Nem sempre é possível
- Tornar o sistema tolerante a falha
  - Capacidade de manter o serviço desejado mesmo na presença de falhas

9

Tolerância a Falhas

- Identificar os tipos de falhas e suas probabilidades;
- Reconhecer os problemas potencialmente provocados por falhas no sistema
- Dominar as soluções que existem para evitar falhas ou recuperar o sistema após a sua ocorrência
- Saber o custo associado das soluções

10

Dependabilidade

- Tolerância a falhas está fortemente relacionada a dependabilidade.
- É um termo que abrange uma série de requisitos:
  - **Disponibilidade** – O sistema está pronto para ser usado imediatamente?
  - **Confiabilidade** – O sistema funciona continuamente sem falhas?
  - **Segurança** – Se o sistema deixar de funcionar corretamente por um certo tempo acontecerá algo "catastrófico"?

Sistemas Tolerantes a Falhas

- Antigamente apenas sistemas críticos, tais como aviões, sondas espaciais, controles industrais e mainframes tinham como preocupação a tolerância a falhas.
- As redes e internet aumentaram a depedência tecnológica da população
  - Email, distribuição de conteúdos, relacionamentos, comunicação, comercio
  - Falhas nesses serviços podem implicar em perda de dinheiro ou reputação.
  - Serviços desse tipo são normalmente implementados através de sistemas distribuídos

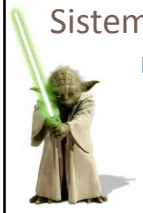
12

Sistemas Distribuídos

- Sistemas distribuídos são construídos por vários nós processadores independentes
- Toda a interação deve ser feita por troca de mensagens através de canais de comunicação
- São geralmente construídos com elementos não homogêneos

13

Sistemas Distribuídos e as Falhas



- Oferecem redundância natural
- Falhas parciais

- Mais componentes -> Mais falhas
- Processos e canal de comunicação podem falhar
- Heterogeneidade
- Múltiplos domínios administrativos

14

Sistemas Distribuídos e as Falhas

- Oferecem redundância natural
- Falhas parciais

TÉCNICAS DE TOLERÂNCIA A FALHAS TAMBÉM SE APLICAM A SISTEMAS DISTRIBUÍDOS

- Mais
- Proc
  podem falhar
- Heterogeneidade
- Múltiplos domínios administrativos

15

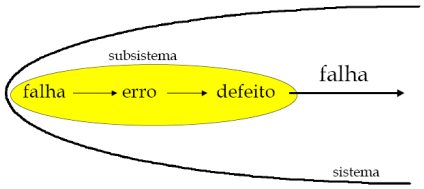
O que são e quais são as falhas?

16

Falhas, Erros ou Defeitos?

- Um **defeito** é um desvio do comportamento do sistema de algum conjunto de especificações pré-definidas.
  - Algum requisito não foi implementado ou não está sendo atendido de forma adequada
- Um sistema está em estado errôneo, ou em **erro,** se o processamento posterior a partir desse estado pode levar a um defeito.
- **Falha** ou falta (*fault*) como a causa física ou algorítmica do erro.

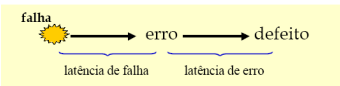
Falha -> Erro -> Defeito



18

Latência

- **latência de falha**
  - período de tempo desde a ocorrência da falha até a manifestação do erro devido aquela falha
- **latência de erro**
  - período de tempo desde a ocorrência do erro até a manifestação do defeito devido aquele erro



19

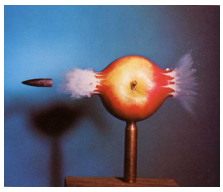
Tipos de Falhas

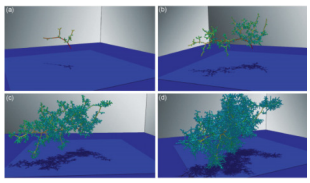
- Falhas físicas
  - Permanentes
  - Temporárias { Transientes, Intermitentes }
- Humanas
  - Falhas de projeto
  - Interação { Intencionais, Não intencionais }

20

Modelo de Falhas

- Modelos permitem obter um entendimento de como o sistema se comporta no mundo real
- Um modelo de falha define as formas com que podem ocorrer e como diagnosticá-las





21

Modelo de Falhas (Cristian)

- Falha por **queda**
  - Servidor pára de funcionar subtamente,
- Falha por **omissão**
  - A resposta de uma requisição não chega ao cliente
    - Recepção
    - Envio
    - Servidor
- Falha de **temporização**
  - A resposta de uma requisição ou chega cedo demais ou tarde demais.
- Falhas de **resposta**
  - Respostas está incorreta, mas dectável
- Falhas **arbitrárias** ou bizantinas
  - Respostas está incorreta, mas não dectável

22

Técnicas de Tolerância a Falhas

23

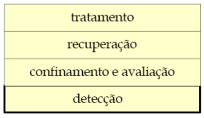
Técnicas de Tolerância a Falhas

- Prevenção
  - Minimizar, por construção, a probabilidade da ocorrência de faltas
  - Não é suficiente!!!!!!!
- Tolerância a Falhas
  - detecção de erros seguida de recuperação do sistema ou mascaramento

24

Detecção seguida de recuperação

- Fases (*Anderson & Lee):*
  - Detecção de erros
  - Confinamento e avaliação
  - Recuperação
  - Tratamento da falha
- Outra classificação
  - detecção, diagnóstico, confinamento, mascaramento, compensação



25

Detecção de erros

- Dedução de erro
  - Uma falha primeiro se manifesta como um erro, para então ser detectada
- Principais tipos de detetores de erros
  - duplicação e comparação
  - temporizadores (*watchdogs*)
  - códigos e redundância estrutural
  - testes de aceitação e diagnósticos

26

Consenso

- O Consenso é um problema fundamental em Sistemas Distribuídos.
  - Utilizado como componente básico de vários algoritmos onde os processos precisam ter uma visão idêntica (ou ação coordenada).
- Exemplos:
  - definição de uma ordem total entre eventos (ou mensagens)
  - visão uniforme sobre o conjunto de processos não falhos/ativos de um grupo
  - definição sobre a ação comum a ser executada

27

Confinamento e Avaliação

- Latência de falha
  - pode provocar espalhamento de dados inválidos
- Confinamento
  - estabelece limites para a propagação do dano ocorrência de falha até erro (detectado)
- Dependem de decisões de projeto do sistema facilitam detecção e recuperação,mas não são obrigatórias

28

Recuperação

- Abordagem pessimista (*backward recovery*)
  - pontos de salvaguarda (*checkpoint*)
  - implementação razoavelmente simples
  - custo alto
  - efeito "dominó"
- Abordagem otimista (*forward recovery*)
  - ações corretivas
  - considerações sobre a natureza dos erros
  - normalmente é dependente da aplicação ou do sistema

29

Tratamento

- Identificação de componentes com erros
- Localização do falha
- Reparo do sistema
  - *on-line*
  - reconfiguração
  - componentes suplentes (*spares*)
    - passivos (*cold-spares*) e ativos (*hot-spares*)

30

Mascaramento

- Busca tornar as ocorrências de falhas ocultas para outros processos e usuários.
- A técnica fundamental para mascarar falhas é usar redundância:
  - **Redundância de informação** – bits extras para recuperação de pacotes;
  - **Redundância de tempo** – executar novamente uma ação, se for preciso.
  - **Redundância física** – adicionar equipamentos ou processos extras para possibilitar tolerância a perda ou mal funcionamento de alguns componentes.

31

Conclusão

32

Conclusão

- Tolerância a Falhas incrementa a **complexidade do sistema.**
- Implementar sistemas distribuídos **como Tolerante a Falhas é uma** tarefa difícil.
  - Multicast confiável
  - Checkpoint distribuído
- Há necessidade por **ferramentas de suporte à** implementação de sistemas distribuídos Tolerância a falhas.