Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

SISTEMAS DISTRIBUÍDOS

TOLERÂNCIA A FALHAS

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

TOLERÂNCIA A FALHAS

- Uma característica de sistemas distribuídos que os distingue de sistemas de uma máquina é a noção de falha parcial
- Uma falha parcial pode acontecer quando um componente de um SD falha
- Esta falha pode afetar a operação de alguns componentes e ao mesmo tempo deixar outros completamente ilesos
    - Uma falha em um sistema de uma máquina só é quase sempre total => afeta todos os componentes

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

TOLERÂNCIA A FALHAS

- Um objetivo importante no projeto de um sistema distribuído:
    - Construir o sistema de modo que ele possa se recuperar automaticamente de falhas parciais sem afetar seriamente o desempenho global.
    - Sempre que ocorrer uma falha o SD deve continuar a funcionar de maneira aceitável enquanto o sistema se recupera
        - O sistema deve ser tolerante a falhas

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

TOLERÂNCIA A FALHAS

Tolerância a falhas está fortemente relacionada a sistemas confiáveis. Um sistema confiável abrange uma série de requisitos:

- Disponibilidade
- Confiabilidade
- Segurança
- Capacidade de Manutenção

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

TOLERÂNCIA A FALHAS

Disponibilidade

- É a propriedade de um sistema estar pronto para ser usado imediatamente
- Um sistema de alta disponibilidade é aquele que mais provavelmente estará funcionando em dado instante

Confiabilidade

- É a propriedade do sistema funcionar continuamente sem falhas
- Um sistema de alta confiabilidade continuará a executar sem interrupção durante um período de tempo longo
- Tempo médio entre falhas

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

TOLERÂNCIA A FALHAS

Disponibilidade x Confiabilidade

- Disponibilidade é definida em termos de instante de tempo e confiabilidade em termos de intervalo de tempo
- Um sistema muito confiável é aquele que continua a trabalhar sem interrupção durante um período relativamente longo de tempo
- Se um sistema ficar fora do ar por um milissegundo a cada hora, terá uma disponibilidade de mais de 99.99999%, mas sua confiabilidade ainda será muito baixa
- Por outro lado, um sistema que nunca cai mas é desligado por duas semanas no ano, tem alta confiabilidade, mas somente 96% de disponibilidade

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

TOLERÂNCIA A FALHAS

Segurança

- Refere-se a situação que se o sistema deixar de funcionar corretamente por um certo tempo nada catastrófico acontecerá

Capacidade de Manutenção

- Refere-se a facilidade com que um sistema que falhou pode ser consertado
- Sistemas de alta capacidade de manutenção também podem mostrar alto grau de disponibilidade, em especial se as falhas puderem ser detectadas e reparadas automaticamente

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

DEFEITOS x ERROS x FALHAS

- **Defeito** => se um sistema não pode cumprir suas promessas, apresenta defeito

- **Erro** => parte do estado de um sistema que pode levar a um defeito
  - Exemplo: Pacotes danificados transmitidos

- **Falha** => é a causa de um erro
  - Um meio de transmissão errado ou ruim pode danificar pacotes, neste caso => fácil reconhecer a falha
  - Alguns erros de transmissão podem ser causados por más condições atmosféricas => difícil remover a falha

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

TIPOS DE FALHAS

- **Transiente**: Ocorre uma vez e depois desaparece
  - Se a operação for repetida, a falha não acontecerá novamente
  - Exemplo: Pássaro voando na frente de um feixe de micro-ondas interrompe a transmissão
- **Intermitente**: Ocorre e desaparece por "sua própria vontade"
  - Exemplo: conector com problemas (difícil de diagnosticar)
- **Permanente**: Continua a existir até que o componente faltoso seja substituído
  - Exemplo: bugs de software, chips queimados

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

MODELOS DE FALHAS

- Um sistema que apresenta defeito não fornece seus serviços adequadamente => Como encontrar o problema?
- Nem sempre o servidor que está funcionando mal é a falha que está se procurando
- Se o servidor depende de outros servidores, por exemplo, pode ser que a falha esteja em outro lugar
- Tais relações de dependência acontecem muito em sistemas distribuídos
- Um disco defeituoso em um servidor de arquivos que faz parte de um banco de dados distribuído => pode comprometer o funcionamento adequado de todo o banco

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

MODELOS DE FALHAS

Falha por queda

Um servidor para de funcionar, mas estava funcionando corretamente até sua parada.

- Exemplo: Um SO que para em um estado que somente um reboot possa fazer ele voltar a funcionar

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

MODELOS DE FALHAS

Falha por omissão

Ocorre quando o servidor falha em responder à solicitações dos clientes, falha em receber mensagens ou em enviar mensagens. Razões:

- Omissão-recebimento: A conexão entre cliente e servidor foi estabelecida corretamente, mas não tem thread para receber as mensagens.
- Omissão-envio: Um buffer de envio estoura e a mensagem não é enviada. O servidor tem que estar preparado porque um cliente pode solicitar
- Um loop infinito onde cada iteração cria um novo processo causando que o processo pare em algum momento

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

MODELOS DE FALHAS

Falha por temporização

- Uma resposta do servidor está fora de um intervalo de tempo específico
- Um site de e-commerce site pode definir que uma resposta ao usuário não deve ser dada em mais de 5 segundos
- Em uma aplicação de vídeo por demanda, um cliente tem que receber os frames em uma determinada frequência
- Difíceis de gerenciar

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

MODELOS DE FALHAS

Falha arbitrária

- Servidor está realizando respostas incorretas, mas que não podem ser detectadas como incorretas
- Um servidor faltoso pode estar trabalhando maliciosamente com outros servidores para produzir respostas erradas

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

MODELOS DE FALHAS

Tipo de falha	Descrição
Falha por queda	O servidor pára de funcionar, mas estava funcionando corretamente até parar.
Falha por omissão *Omissão de recebimento* *Omissão de envio*	O servidor não consegue responder a requisições que chegam O servidor não consegue receber mensagens que chegam O servidor não consegue enviar mensagens
Falha de temporização	A resposta do servidor se encontra fora do intervalo de tempo
Falha de resposta *Falha de valor* *Falha de transição de estado*	A resposta do servidor está incorreta O valor da resposta está errado O servidor se desvia do fluxo de controle correto
Falha arbitrária	Um servidor pode produzir respostas arbitrárias em momentos arbitrários

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

MASCARAMENTO DE FALHA

Para o sistema ser tolerante a falhas, as ocorrências das falhas devem ser ocultas de outros processos e usuários

A técnica fundamental para mascarar falhas é usar **redundância**:

- Redundância de Informação
- Redundância de Tempo
- Redundância de Física

MASCARAMENTO DE FALHA

Redundância de informação

Bits extras podem ser adicionados para recuperação de bits deteriorados.

- Exemplo: Código de Hamming (é um código de detecção, isto é, permite não apenas detectar erro de um bit, mas também a localização do bit errado)

https://www.ime.usp.br/~song/mac412/hamming.pdf

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

MASCARAMENTO DE FALHA

Redundância de Tempo

Uma ação é realizada e se necessário é realizada novamente

Exemplo: Se uma transação aborta, ela pode ser refeita sem prejuízo.

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

MASCARAMENTO DE FALHA

Redundância Física

Processos ou equipamentos extras são adicionados para possibilitar que o sistema possa como um todo tolerar a perda ou mau funcionamento de alguns componentes.

Redundância Física pode ser feita em hardware ou em software.

Exemplos:

- Aeronave: 747 tem 4 motores mas voa com 3.
- Aeronave espacial: Tem 5 computadores

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

ESTRATÉGIAS DE TOLERÂNCIA A FALHAS

Resiliência de Processos

- Replicação de processos em grupos
- Grupos Simples ou Hierárquicos

Comunicação Confiável Cliente-Servidor

- Falhas de Comunicação
- Canal de Comunicação pode exibir falhas por queda, por omissão, arbitrárias
- TCP (ponto-a-ponto)
- RPC

ESTRATÉGIAS DE TOLERÂNCIA A FALHAS

Comunicação Confiável de Grupo

- Como implementar entrega confiável de mensagens a todos os processos?

Comprometimento Distribuído

- Envolve a realização de uma operação por cada membro de um grupo de processos ou por absolutamente nenhum (entrega de mensagens).