08 – Tolerância a Falhas

Tolerância a falhas

- Modelos de falhas;
- Questões de projeto;

  Comunicação segura entre cliente-servidor;
- Tolerância a falhas em comunicação em grupo;
- Tolerância a falhas em comunicação ponto-a-ponto Semânticas RPC na presença de falhas;
- Recuperação de falhas;

08 – Tolerância a Falhas

- Uma das características essenciais em SD e que o distingue de Sistemas Centralizados é a sua capacidade de ter falhas parciais.

- Nesse sentido, o que significa dizer que um SD é tolerante a falhas?

08 – Tolerância a Falhas

- Um dos objetivos do projeto de SD é permitir a recuperação de falhas parciais sem afetar o desempenho do sistema
- Você consegue imaginar uma forma (simples ??) de fornecer tolerância a falhas?

08 – Tolerância a Falhas

- Imagine como é difícil prover tolerância a falhas em SD...
  - É necessário manter o desempenho...
  - É necessário manter a transparência...
  - Lembra dos protocolos de Rede?
- Eles já forneciam capacidades de tolerância a falhas?

08 – Tolerância a Falhas

- Conceitos básicos:
  - Disponibilidade:
    - É a capacidade do sistema estar pronto para ser utilizado imediatamente;
    - Em geral, refere-se a probabilidade do sistema estar operando corretamente em dado momento e estar disponível para realizar suas funções;

08 – Tolerância a Falhas

- Conceitos básicos:
  - Confiabilidade:
    - O sistema pode trabalhar continuamente sem falhas;
    - Ao contrario da disponibilidade, a confiabilidade é definida em termos de intervalos de tempo;
    - Um sistema confiável é aquele que funciona continuamente sem interrupção durante um longo período de tempo;

08 – Tolerância a Falhas

- Conceitos básicos:
  - Segurança:
    - Se o sistema falha temporariamente, nada catastrófico acontece;

08 – Tolerância a Falhas

- Conceitos básicos:
  - Capacidade de manutenção:
    - Refere-se a capacidade de um sistema falho ser reparado;

08 – Tolerância a Falhas

- Mas...qual a diferença entre falhas e erros?
  - Falhas: Ocorrem quando o sistema não consegue fornecer seus serviços;
  - São as causas de erros!
  - Ex: Um meio de transmissão ruim pode fazer com que pacotes sejam perdidos ou entregues com erros!

08 – Tolerância a Falhas

- E ainda...existem vários tipos de falhas:
  - Falhas transientes: Ocorrem uma vez e desaparecem;
  - Falhas intermitentes: Ocorrem, desaparecem, reaparecem;
  - Falhas permanentes: Existem até que o elemento seja reparado;
- Você pode encontrar exemplos dessas três falhas?

08 – Tolerância a Falhas

- Duvida:
  - Um Servidor Web que retorna uma pagina desatualizada ao invés de uma mais recente...está com falhas?
  - Em caso positivo, que tipo de falha?

08 – Tolerância a Falhas

Tipo de falha	Descrição
Falha de Crash	Um servidor é desligado, mas estava trabalhando corretamente até então
Falhas de Omissão *Omissão na Recepção* *Omissão de envio*	Um servidor falha ao responder requisições Um servidor falha para receber mensagens Um servidor falha ao enviar mensagens
Falha de Temporização	A resposta do servidor está fora do intervalo de tempo especificado
Falha de Resposta *Falha de valores* *Falha na transição de estados*	A resposta do servidor está incorreta O valor da resposta está errado O servidor desvia-se do fluxo de controle correto
Falhas arbitrárias	Um servidor pode produzir respostas arbitrárias em tempos arbitrários

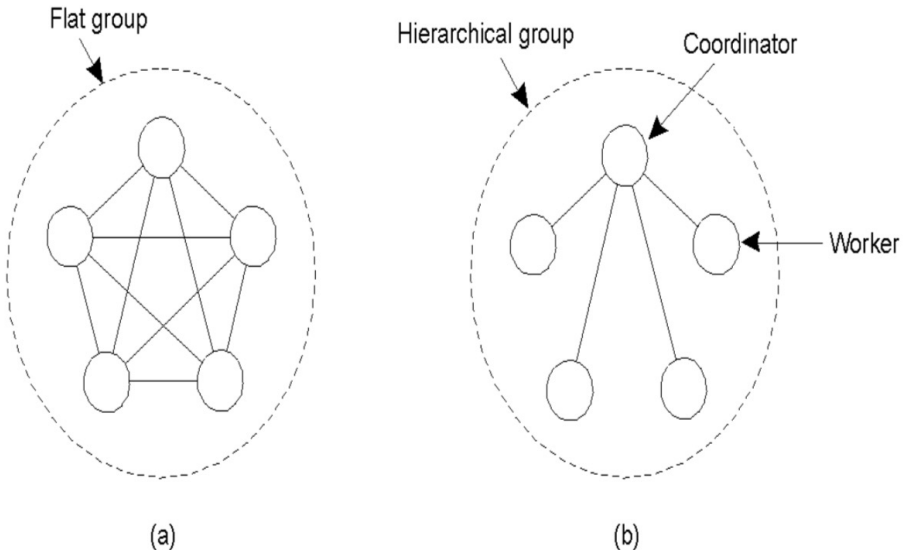
08 – Tolerância a Falhas

- O mecanismo básico para contemplar a tolerância a falhas é a replicação;
- No contexto de comunicação, a tolerância a falhas é amplamente relacionada com a utilização de grupos de processos (multicast);

08 – Tolerância a Falhas

a) Um grupo sem hierarquia (flat);

b) Um grupo hierárquico;



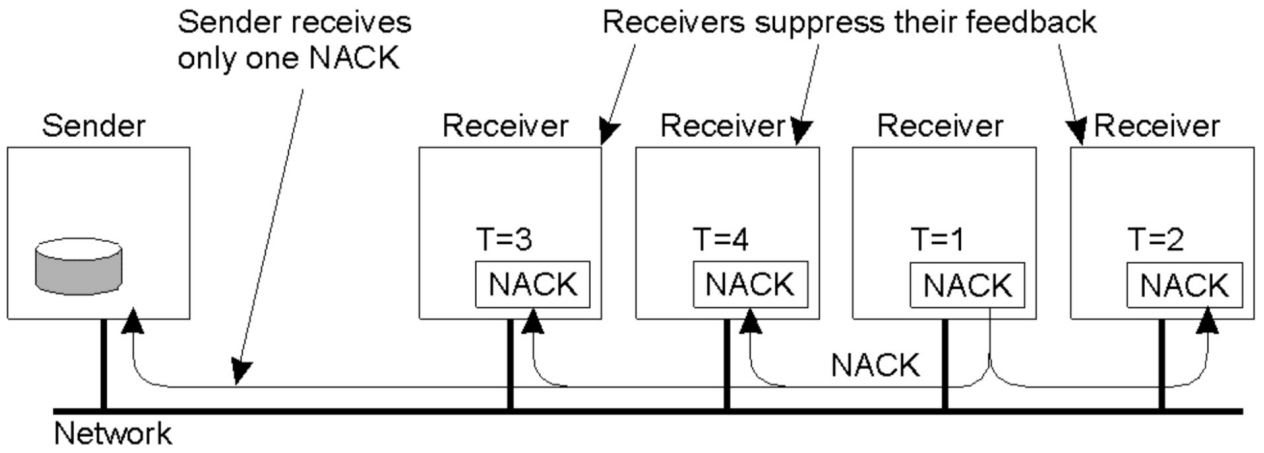
08 – Tolerância a Falhas

Esquemas de multicasting confiáveis

- Uma solução simples para multicasting confiável quando todos os receptores são conhecidos e assume-se que não falharão
  a) Transmissão de mensagens;
  b) Feedback;

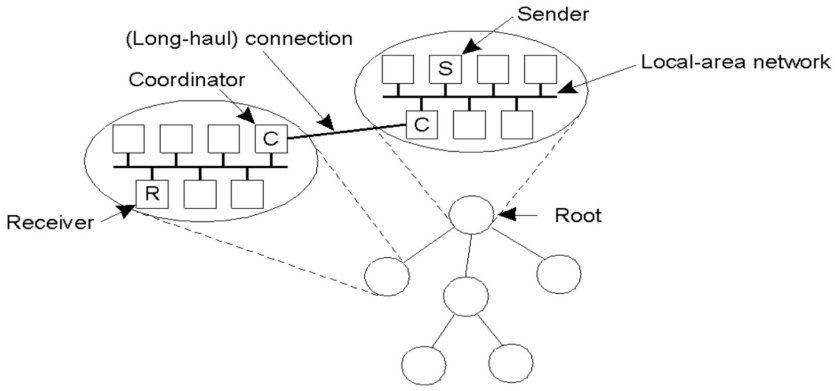
Controle de feedback não hierárquico

- Diversos receptores podem querer uma retransmissão, mas o primeiro pedido suprime as outras;



Controle de feedback hierárquico

- Multicasting hierárquico confiável
  - Cada coordenador local envia a mensagem para seus filhos;
  - Um coordenador local manipula requisições de retransmissão;



08 – Tolerância a Falhas

Comunicação Ponto a Ponto

- Comunicação ponto-a-ponto pode acontecer através do uso de protocolos confiáveis como o TCP;
- TCP mascara falhas de omissão através do uso de acks e retransmissões;
- No entanto, falhas de crash não são mascaradas! ;

Semânticas RPC na presença de falhas

- Aplicam-se para qualquer sistema de comunicação baseado em RPC:
  - Java RMI
  - CORBA
- É difícil mascarar falhas em comunicação entre processos de maneira que tais falhas pareçam as mesmas falhas de sistemas centralizados

08 – Tolerância a Falhas

Semânticas RPC na presença de falhas

- Há basicamente 05 classes de falhas que podem ocorrer em um sistema RPC:
    1. Cliente não consegue localizar o servidor;
    2. Mensagem de requisição do cliente é perdida;
    3. O servidor entra em crash depois de receber uma requisição;
    4. A mensagem de resposta do servidor para o cliente é perdida;
    5. O cliente entra em crash depois de enviar uma requisição;

Semânticas RPC na presença de falhas

- Cliente não pode localizar o servidor;
  - Servidor pode estar desligado;
  - Versões diferentes de stubs;
- Uma solução é fazer com que exceções sejam geradas!

08 – Tolerância a Falhas

Semânticas RPC na presença de falhas

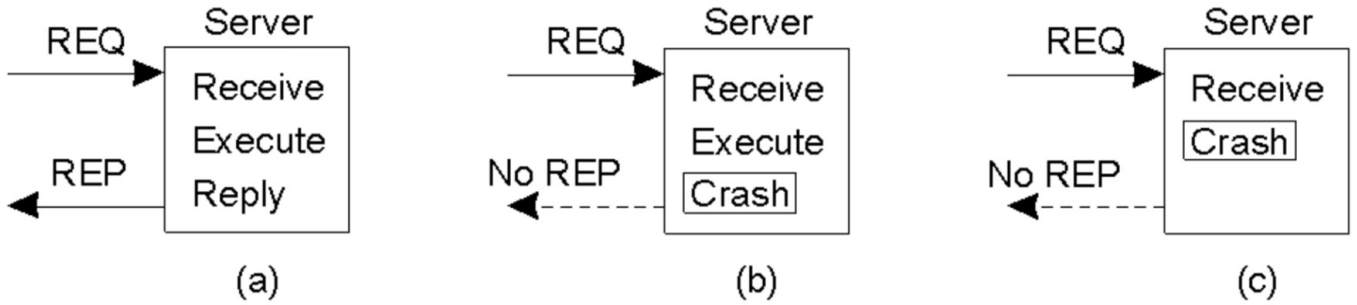
- Perda da mensagem de requisição
  - Cliente pode iniciar um timer;
  - Se uma resposta ou ack não chega, a mensagem é enviada novamente;
- Se as mensagens são sempre perdidas...
- Quais as conclusões do cliente?
  - O servidor pode diferenciar uma requisição de uma retransmissão?

08 – Tolerância a Falhas

Semânticas RPC na presença de falhas

- Crash do Servidor
  - O servidor pode falhar em diferentes momentos:
    - Após executar a requisição;
    - Antes de executar a requisição;
  - Dependendo do “momento” da falha, diferentes tratamentos devem ser dados;

Mensagens de Requisições perdidas e crashes do servidor



- Um servidor em comunicação cliente-servidor

a) Caso Normal;

b) Execução depois de um crash;

c) Execução antes de um crash;

Semânticas RPC na presença de falhas

- Crash do Servidor
  - Existência de semânticas para tratamento desse tipo de falha:
    - Pelo menos uma vez: Tentar até que, pelo menos uma execução completa seja realizada
      - ❑ Nesse modelo, a execução deve ocorrer pelo menos uma vez, possivelmente até mais que uma.
    - No máximo uma vez: A execução deve ocorrer no máximo uma vez mas, pode não ocorrer
    - O ideal seria uma semântica: exatamente uma vez!

Semânticas RPC na presença de falhas

- Perda de mensagens de resposta
  - Qual seria uma solução (simples?) para esse problema?

08 – Tolerância a Falhas

Semânticas RPC na presença de falhas

- Perda de mensagens de resposta
  - Uma possível solução seria basear-se em um temporizador
  - Questões que podem surgir:
    - A requisição foi perdida;
    - A resposta foi perdida?;
    - O servidor está lento?;

- Deve-se atentar para o caso de operações que não são idempotentes!
  - Ex: operações bancarias;

- Operações idempotentes são aquelas que podem ser repetidas sem causar inconsistências no sistema;

08 – Tolerância a Falhas

Semânticas RPC na presença de falhas

- Cliente entra em crash
  - O que acontece se um cliente envia uma requisição e entra em crash?
  - Quais seriam possíveis tratamentos para essa falha?

Semânticas RPC na presença de falhas

- Cliente entra em crash
  - As respostas do servidor ficam órfãs!
  - Soluções:
    - Antes de enviar a requisição, fazer um log em disco;
      - Órfãos são eliminados após um reboot do cliente;
    - Dividir o tempo em épocas que devem ser iniciadas após um reboot do cliente;
      - Órfãos são eliminados após um reboot do cliente;
      - Existem variantes dessa solução!
  - Quais são os problemas de eliminar uma requisição órfã?

08 – Tolerância a Falhas

Recuperação de Falhas

- A ideia geral é, passar de um estado de erro para um estado livre de erros;
- Duas técnicas básicas:
  - Recuperação para trás;
  - Recuperação para frente;

08 – Tolerância a Falhas

Recuperação de Falhas

- Exemplo: Recuperação de pacotes perdidos em uma rede
  - Retransmissão de pacotes
    - Recuperação para trás;
  - Reconstruir o pacote perdido a partir de outro
    - Recuperação para frente;

08 – Tolerância a Falhas

Recuperação de Falhas

- Para conseguir recuperar os dados para um estado seguro é necessário possuir um armazenamento estável;
  - Em termos práticos, consiste na adoção de uma matriz de discos que faz a redundância/distribuição dos dados;

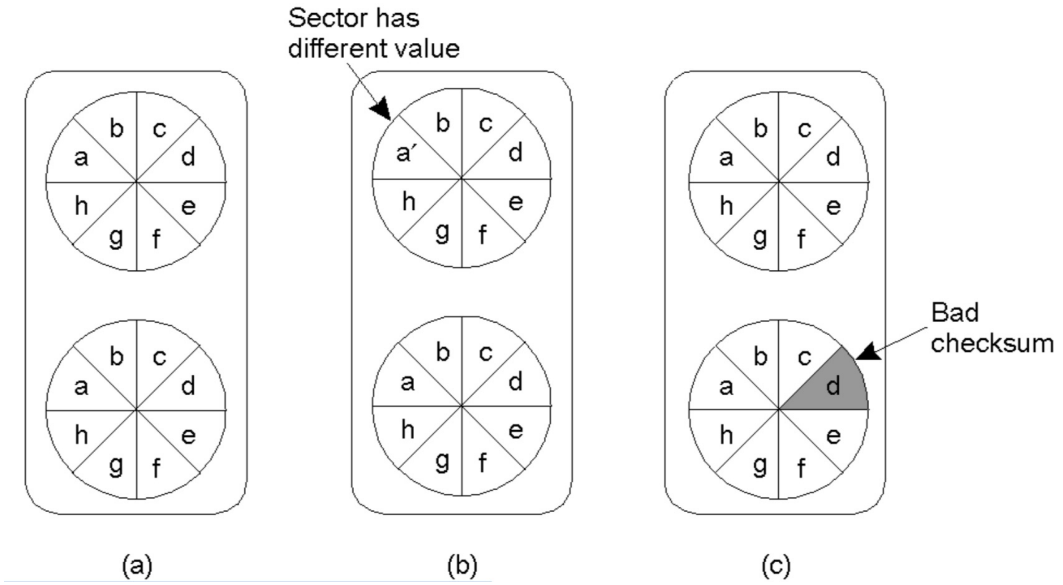
08 – Tolerância a Falhas

Recuperação de Armazenamento estável

a) Armazenamento estável;

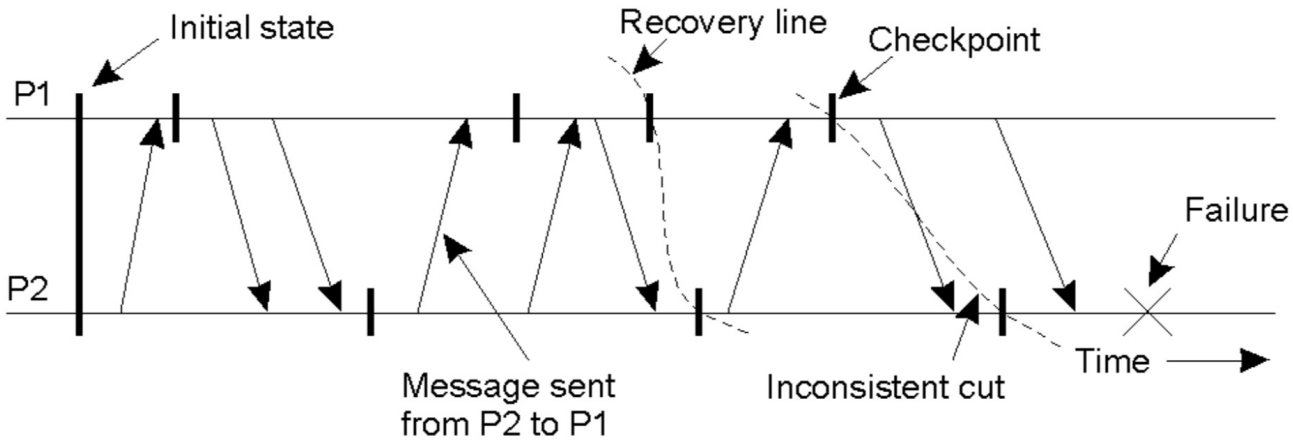
b) Crash de atualização no disco 2;

c) Soma errada no disco 2;



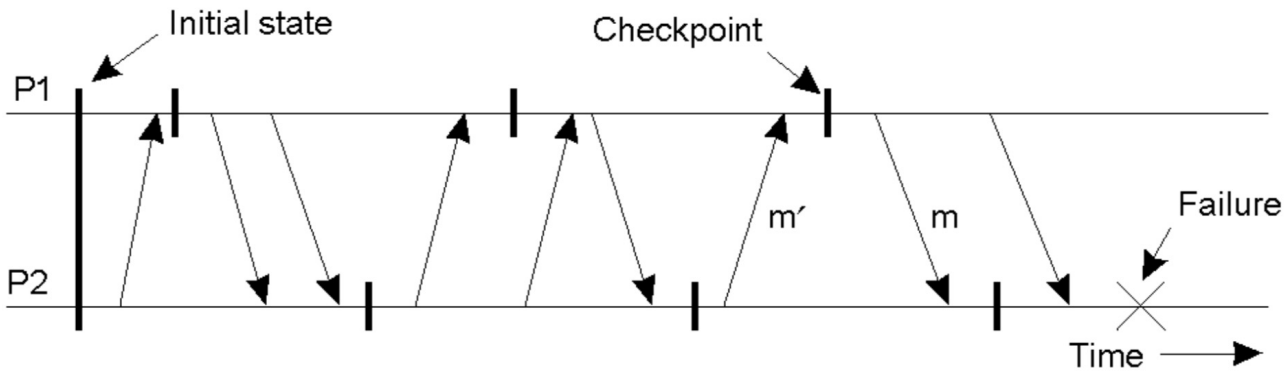
Checkpoint

- Salvar o estado dos processos de tempos em tempos



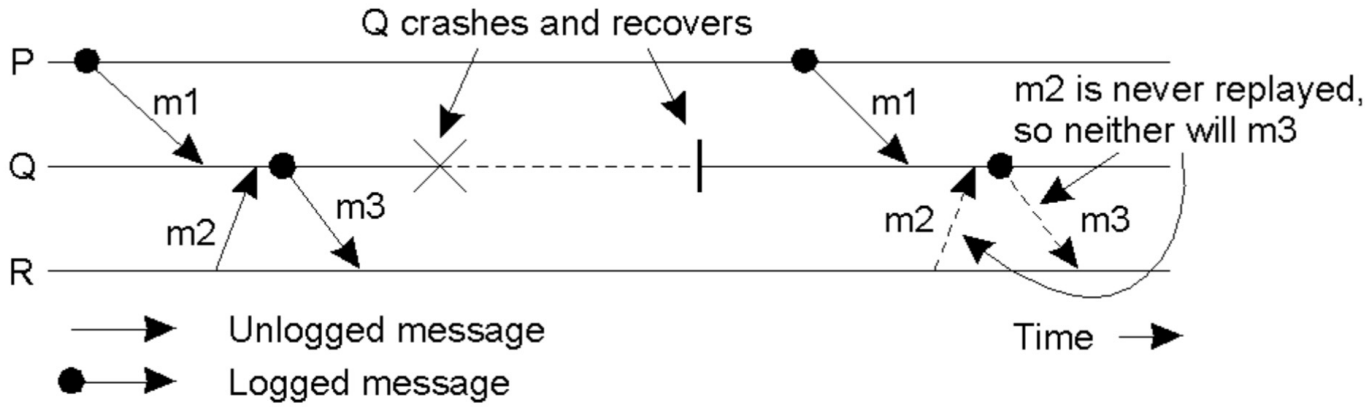
Checkpoint independente

- Pode causar um efeito dominó na recuperação para um estado livre de erros;
- Qual a solução?
- Como seria um checkpoint coordenado?



Looping de mensagens

- Repetição incorreta de mensagens depois da recuperação, gerando um processo órfão ;



08 – Fonte Bibliográfica

Material desenvolvido utilizando como base material do Prof. Ricardo Ribeiro dos Santos e o Livro Sistemas Distribuídos: Princípios e Paradigmas do Andrew S. Tanenbaum e Maarten Van Steen;