Tolerância a falhas

Ricardo Costa – rcosta@estg.ipp.pt

Tolerância a falhas

Conceitos Básicos

- Técnica fundamental para tratamento de falhas: redundância;
- Tolerância a falhas está relacionada à confiabilidade (dependability) de um sistema;
- Requisitos para confiabilidade:
  - Disponibilidade (availability)
  - Confiabilidade (reliability)
  - Segurança (safety)
  - Capacidade de manutenção (maintainability)

Tolerância a falhas

Disponibilidade (availability)

- Indica que um sistema está pronto para uso imediato;
- Refere-se à probabilidade de um sistema estar a funcionar corretamente num dado instante e disponível para executar suas funções;
- Alta disponibilidade indica que um sistema, muito provavelmente, estará funcionando a qualquer dado instante;

Tolerância a falhas

Confiabilidade (reliability)

- Refere-se à propriedade que um sistema irá funcionar continuamente sem falha;
- É definida em termos de um intervalo de tempo;
- Um sistema confiável é aquele que muito provavelmente irá funcionar sem interrupção durante um período relativamente longo de tempo;

Tolerância a falhas

Segurança (safety)

- Refere-se às consequências da falha de um sistema que não funcione corretamente;
- Sistemas críticos devem ter um alto grau de segurança;

Tolerância a falhas

Capacidade de manutenção (maintainability)

- Indica a facilidade que um sistema que falhou seja reparado;

- Sistema com alta capacidade de manutenção normalmente também apresenta alto grau de disponibilidade, especialmente se as falhas poderem ser detectadas e reparadas automaticamente;

Tolerância a falhas

DISPONIBILIDADE X CONFIABILIDADE

- Sistema que falha por 1 milissegundo a cada hora:
  - Disponibilidade alta, acima de 99.9999 %;
  - Confiabilidade baixa;
- Sistema que nunca crasha mas é desligado por 2 semanas no ano:
  - Alta confiabilidade;
  - Apenas 96 % de disponibilidade;

Tolerância a falhas

Conceitos Básicos

- Defeito: se um sistema não pode cumprir com a suas especificação, apresenta defeito.
  - Ex.: Não consegue garantir as consistências especificadas.
- Erro: parte do estado de um sistema causado por uma falha.
  - Ex.: Pacotes danificados.
- Falha: é a causa de um erro.
  - Ex.: Um meio de transmissão errado pode danificar pacotes.

Tolerância a falhas

Conceitos Básicos

- Sistema falha quando não pode manter a sua especificação;
- Um erro é uma parte do estado de um sistema que pode levar a uma falha;
  - Causa do erro é chamada de falta(fault);
- A construção de sistemas confiáveis (dependable) está relacionada ao controle das faltas (faults);
  - Faltas podem ser prevenidas, removidas e previstas.

Tolerância a falhas

Conceitos Básicos

- Sistemas de uma máquina (não distribuídos): uma falha é quase sempre total;
- Sistemas distribuídos: pode ocorrer uma falha parcial, quando um componente do sistema falha.
  - Objetivo: recuperar automaticamente de falhas parciais sem afetar seriamente o desempenho global

Tolerância a falhas

Tolerância a Faltas

- Significa que um sistema pode continuar a fornecer os seus serviços mesmo na presença de faltas;
- NÃO significa que falhas não vão ocorrer!!!

Tolerância a falhas

Tipos de Faltas

- Transiente:
  - Ocorre uma vez e desaparece;
- Intermitente:
  - Ocorre por um período indeterminado, desaparece, reaparece, e assim por diante;
- Permanente:
  - Continua a existir até que o componente faltoso seja substituído;

Tolerância a falhas

Como tratar Faltas

- Fault prevention:
  - Prevenir a ocorrência de faltas
- Fault tolerance:
  - Construir componente que possa mascarar a presença de faltas;
- Fault removal:
  - Reduzir a presença, o número e a gravidade (seriousness) de faltas;
- Fault forecasting:
  - Estimar a situação atual e futura de faltas e as consequências das suas ocorrências;

Tolerância a falhas

Modelos de Faltas

- Para melhor identificar as falhas, foram desenvolvidos diversos esquemas de classificação, entre eles, o quadro abaixo:

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

Tolerância a falhas

Mascaramento de falhas por redundância

- Para o sistema ser tolerante a falhas, as ocorrências destas devem ser ocultas de outros processos e dos utilizadores;

- A técnica fundamental para mascarar falhas é usar redundância;

Tolerância a falhas

Mascaramento de falhas por redundância

- Redundância de informação:
  - bits extras para recuperação de pacotes (Hamming);
- Redundância de tempo:
  - executar novamente uma ação, se for preciso;
- Redundância física:
  - adicionar equipamentos ou processos extras para possibilitar tolerância a perda ou o mau funcionamento de alguns componentes;

Tolerância a falhas

Resiliência de Processo

- Resiliência:
  - The ability to recover quickly from illness, change, or misfortune;
  - The property of a material that enables it to resume its original shape or position after being bent, stretched, or compressed; elasticity;
- Proteção contra falha de processos: obtida com replicação de processos em grupos;

Tolerância a falhas

Resiliência de Processo

- Diversos processos idênticos são organizados num grupo;
- Quando uma mensagem é enviada para um grupo, todos os processos membros deste grupo a recebem;
- Se um processo no grupo falha, outro pode assumir;
- Grupos podem ser dinâmicos, criados e destruídos a pedido;

Tolerância a falhas

Resiliência de Processo

- Processos podem entrar e sair de grupos em tempo de execução e podem pertencer a vários grupos simultaneamente;
- Devem existir mecanismos para gestão de grupos e processos;
- Os grupos permitem tratar vários processos usando abstração;
- As mensagens podem ser enviadas para os grupos;

Tolerância a falhas

Resiliência de Processo

- A abordagem fundamental para tolerar um processo em falta é organizar vários processos idênticos num grupo.
  - Quando uma mensagem é enviada a um grupo, todos membros do grupo a recebem;
  - Se um processo falhar, espera-se que algum outro se encarregue da mensagem no seu lugar;
  - Grupos podem ser dinâmicos;

Tolerância a falhas

Deteção de Falha

- O mascaramento de falhas requer sua identificação prévia;
- Membros do grupo que não têm falta devem ser capazes de detectar quem ainda é membro do grupo e quem apresentou falta e deve ser removido;
- Mecanismos:
  - Envio de msg: “are you alive?” entre os nós, e espera por resposta – ping ativo;
  - Espera passiva até que mensagens cheguem dos diferentes processos (viável apenas se há comunicação suficiente (frequente) entre os nós;

Tolerância a falhas

Deteção de Falha

- Mecanismo de timeout é usado para verificar processos com faltas;
- Problema: ausência de resposta pode indicar erroneamente que o processo tem falta;
- Implementação pode considerar mecanismo de gossiping, em que nó propaga informações do seu estado aos vizinhos;
- Necessidade de distinguir falha da rede e falha de um processo;

Tolerância a falhas

Deteção de Falha

- Detecção de faltas também pode ser feita como um efeito colateral do uso de gossiping para troca de informações entre os vizinhos, propagando informações sobre os estados dos nós;

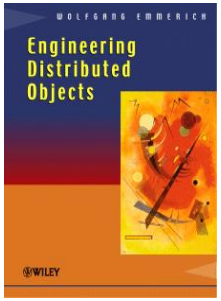
- Eventualmente, todos os processos terão informações suficientes para decidir sobre o estado dos restantes e se há processos com falta;

Tolerância a falhas

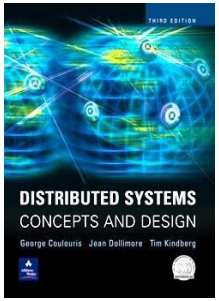
Recuperação

- Ocorrida uma falha, é necessário não apenas identificá-la, mas recuperar da mesma e voltar para um estado correto.
- Essencialmente existem 2 maneiras de se recuperar:
  - Recuperação retroativa – volta para um estado anterior à falha;
  - Recuperação para a frente – tenta levar o sistema para um novo estado correto para que possa continuar a executar.

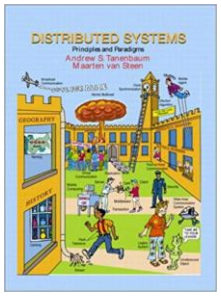
Bibliografia



From: Wolfgang Emmerich
Engineering Distributed Objects
John Wiley & Sons, Ltd 2000



From: Coulouris, Dollimore and Kindberg
Distributed Systems: Concepts and Design
Edition 4 © Addison-Wesley 2005



From: Andrew S., Tanembaum and Van Steen, Maarten
Distributed Systems: Principles and Paradigms
Edition 2 © Pearson 2013

Questões?