

Projeto de Sistemas Distribuídos

Considerações

Projeto de Sistemas Distribuídos

- Problemas
- Objetivos
- Requisitos de usuário

Como são estruturados?

Problemas-chave

- Nomeação
- Alocação de carga
- Manutenção de consistência
- Comunicação
- Estrutura de software

Problema-chave

- Nomeação
 - *Nome*: interpretado por usuários ou programas
 - *Identificador*: interpretado apenas por programas
 - Nome resolvido (*identificador de comunicação*): traduzido para uma forma que pode ser usada para invocar um recurso ou objeto
 - Nomes dão transparência de localização

Problema-chave

- Alocação de carga
 - Ambiente: cargas mutantes
 - Objetivo: bom desempenho

Problema-chave

- Manutenção de consistência
 - Razões para inconsistência
 - Distribuição (separação) de recursos
 - Concorrência
 - Problemas
 - Dados (atualização, replicação, cache)
 - Falha
 - Relógio
 - Interface de usuário (atrasos na interação)

Consistência de atualização

- Perde-se quando a escrita concorrente em dados compartilhados não se realiza como uma única transação atômica
- Solução: transações (ACID)

Exemplo: reserva de trecho de viagem

```

BEGIN_TRANSACTION;
  Reserva (Recife, Salvador);
  Reserva (Salvador, Rio);
  Reserva (Rio, Recife);
END_TRANSACTION;
                
```

ERRO!!!
ABORT_TRANSACTION

© 2002-2003 Carlos A. G. Ferraz 7

Consistência de réplica

- Quando um conjunto de dados deve se manter replicado em várias estações

Quando há modificação em um deles → Multicast

Se não chega a algum deles → INCONSISTÊNCIA

Ex: jogo multi-usuário em rede

© 2002-2003 Carlos A. G. Ferraz 8

Consistência de *cache*

- Para agilizar o acesso a dados compartilhados

↓

Memória *cache*

- Quando um cliente modifica a sua *cache*

As cópias dos outros clientes ficam desatualizadas → INCONSISTÊNCIA

© 2002-2003 Carlos A. G. Ferraz 9

Consistência de falha

- Uma falha em um sistema centralizado

↓

Todos os programas falham → um único comportamento

- Uma falha em um sistema distribuído

↓

Apenas os componentes que executam no sistema com problema falham → os componentes cooperantes falham posteriormente → comportamentos diversos

© 2002-2003 Carlos A. G. Ferraz 10

Consistência de relógio

- Muitos algoritmos dependem de tempo (*timestamps*)
- Há que sincronizar relógios para a realização consistente de operações (eventos) distribuídas

© 2002-2003 Carlos A. G. Ferraz 11

Consistência de interface de usuário

- Exemplo: em uma aplicação distribuída interativa, o usuário aperta um botão e o resultado na tela não aparece de imediato, como se esperaria

- Pode ter havido um retardo (de comunicação) maior do que o usuário suportaria para ter a impressão de dispor de um sistema dedicado

© 2002-2003 Carlos A. G. Ferraz 12

Problema-chave

- Comunicação
 - Desempenho
 - Atraso
 - Confiabilidade
 - Dado não chega
 - Dado deturpado

© 2002-2003 Carlos A. G. Ferraz 13

Problema-chave

■ Estrutura de software

```

    graph TD
      subgraph Aplicação
        A1[Suporte à Prog. Distribuída]
        A2[Serviços Abertos]
      end
      A1 --- A2
      A1 --- N["Núcleo do S.O."]
      A2 --- N
      N --- HW["HW de rede e do computador"]
    
```

© 2002-2003 Carlos A. G. Ferraz 14

Objetivos

- Desempenho
- Confiabilidade
 - Disponibilidade
 - Segurança
 - Tolerância a falhas
- Escalabilidade
- Consistência
- Transparência

Trade-offs: o projeto de qualquer sistema de computação (SD em particular) envolve compensações

- negociação envolvendo os objetivos
 - Ex: desempenho da comunicação versus confiabilidade / segurança (o custo da comunicação)

© 2002-2003 Carlos A. G. Ferraz 15

Requisitos de usuários

- Funcionalidade
- Reconfigurabilidade
- Qualidade de serviço

© 2002-2003 Carlos A. G. Ferraz 16

Requisitos de Usuários

- *Trade-offs* não devem ser uma preocupação dos usuários
- usuários devem receber garantias com respeito aos objetivos do projeto e aos requisitos impostos por eles próprios (usuários)
 - Ex: garantias de *consistência* de um serviço de arquivos com respeito a atualizações concorrentes

© 2002-2003 Carlos A. G. Ferraz 17

Requisitos (cont.)

- Funcionalidade: o que o sistema deve fazer pelos usuários - em geral espera-se que um SD traga melhoramentos sobre serviços fornecidos por outros sistemas
- Lembrando do apelo de SD:
 - economia, em função do compartilhamento de recursos
 - melhor desempenho e disponibilidade

© 2002-2003 Carlos A. G. Ferraz 18

Requisitos (cont.)

- Reconfigurabilidade: acomodação de mudanças sem causar interrupções no provimento de serviços existentes
- Qualidade de serviço: envolvendo tópicos como
 - desempenho
 - confiabilidade e disponibilidade
 - segurança

© 2002-2003 Carlos A. G. Ferraz 19