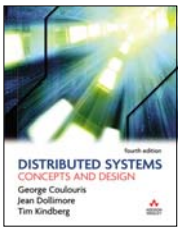
Material baseado no livro *Distributed Systems: Concepts and Design*, 4th Edition, Addison-Wesley, 2005.



1 - Introdução aos Sistemas Distribuídos

Agenda:

- Objetivo do curso
- Definição e motivação
- Exemplos de sistemas distribuídos
- Compartilhamento de recursos e a Web
- Desafios de projeto

Objetivo do curso

- Redes de computadores estão em todo lugar!
  - Redes de telefones celulares
  - Redes corporativas
  - Redes universitárias
  - Redes domésticas
  - Redes embarcadas (carros, aviões, trens)
- Este curso tem como objetivo:
  - Estudar as características das redes de computadores relevantes para projetistas e programadores de sistemas de software; e
  - Apresentas os principais conceitos e técnicas já desenvolvidos para ajudar na tarefa de projetar e implementar sistemas e aplicações baseados nelas (redes).

Definição de sistema distribuído

- Algumas definições encontradas na literatura nos últimos 20 anos:

  Um sistema composto por processadores que se comunicam através de várias linhas de comunicação como barramentos de alta velocidade ou linhas telefônicas. Cada processador possui sua memória local particular, inacessível aos outros processadores [Peterson 85]

  Um conjunto de elementos de computação que cooperam entre si através da troca de informações [Lages 86]

  Um sistema executando em uma coleção de computadores sem memória compartilhada, e que é percebido por seus usuários como um único computador [Tanenbaum 92]

  Uma coleção de computadores independentes que são percebidos por seus usuários como um único e coerente sistema [Tanenbaum & van Steen 02]

Definição de sistema distribuído

- "Definição" de Lamport:

  Um sistema distribuído é aquele onde eu não consigo fazer nada porque algum computador do qual eu nunca tinha ouvido falar falhou

  – Leslie Lamport é um famoso pesquisador da área de sistemas distribuídos, tendo feito diversas contribuições em temas como ordenação de mensagens, sincronização de relógios, tolerância a falhas e consenso

Definição de sistema distribuído

- Sistema distribuído vs. rede de computadores
  - Rede: um meio para interconectar computadores e trocar mensagens através de protocolos bem definidos. Entidades da rede são visíveis e endereçadas explicitamente (IP)
  - Sistema distribuído: a existência de múltiplos computadores autônomos é transparente
  - Muitos problemas (e.g., abertura, confiabilidade) são comuns a ambos, mas tratados em diferentes níveis
    - As redes tratam no nível de pacotes, roteamento, etc, enquanto os sistemas distribuídos tratam no nível das aplicações
    - Todo sistema distribuído depende dos serviços oferecidos por uma ou mais redes de computadores

Definição de sistema distribuído

- Definição adotada no curso:

  Um sistema no qual componentes de hardware e/ou software, localizados em diferentes computadores conectados em rede, se comunicam e coordenam suas ações apenas através da troca de mensagens [Coulouris et al. 05]
- Definição implica em três características:
  - Concorrência
  - Ausência de relógio global
  - Falhas independentes

Características dos sistemas distribuídos

- Concorrência
  - Execução autônoma de programas que podem ou não compartilhar recursos
- Ausência de relógio global
  - Impossibilidade de compartilhar a mesma noção de tempo em todo o sistema (Por quê?)
- Falhas independentes
  - Dificuldade de detectar e esconder a ocorrência de falhas

Motivação para utilizar sistemas distribuídos

- Razão principal: compartilhamento de recursos!
  - Ex.: hardware, software, dados, serviços, etc
- Outras motivações relevantes:
  - Maior desempenho (paralelismo, cache)
  - Maior confiabilidade (redundância, falhas parciais)
  - Aplicações intrinsecamente distribuídas
- Importante: distribuição implica em custos, complexidade e riscos adicionais que devem ser ponderados cuidadosamente em relação aos benefícios esperados

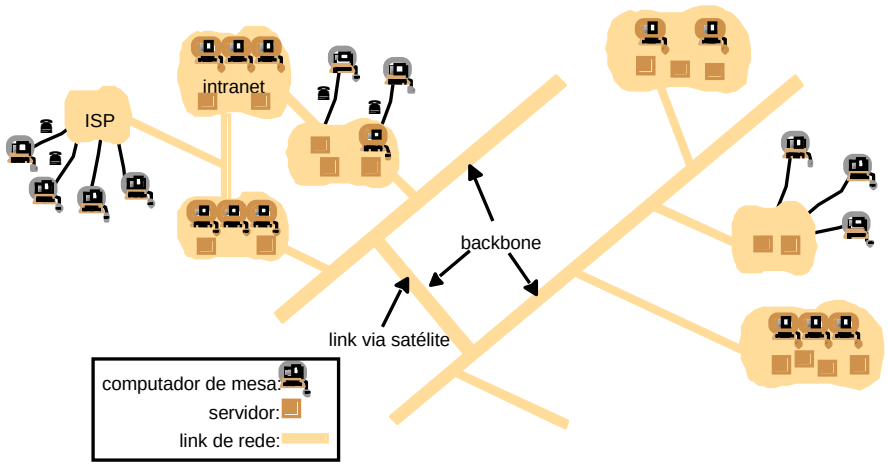
Agenda

- Definição e motivação
- Exemplos de sistemas distribuídos
- Compartilhamento de recursos e a Web
- Desafios de projeto

Exemplos de sistemas distribuídos

- Baseados em conhecidas redes de computadores largamente utilizadas nos dias de hoje:
  - Internet
  - Intranets
  - Ambientes de redes sem fio

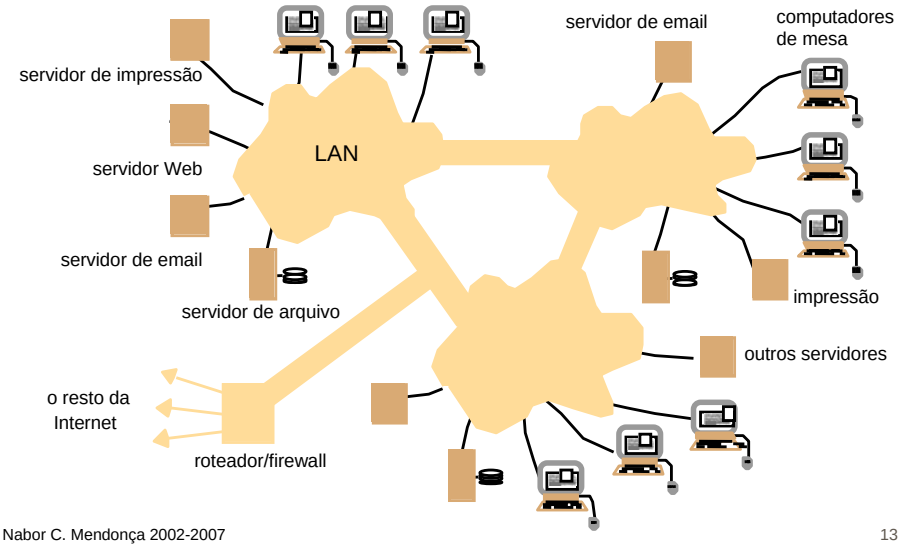
Internet



Internet

- Principais características
  - Vasta coleção de diferentes redes de computadores
  - Troca de informação entre programas e usuários conectados através de um meio comum de comunicação (protocolos da Internet)
  - Permite o acesso a serviços como WWW, email, e ftp, independentemente da sua localização
  - Conjunto aberto de serviços (extensível através da adição de novos servidores e de novos tipos de serviço)
  - Serviços de multimídia (áudio, vídeo, etc) disponíveis, mas capacidade de acesso ainda restrita devido às limitações de largura de banda de grande parte da infra-estrutura de comunicação atualmente em uso

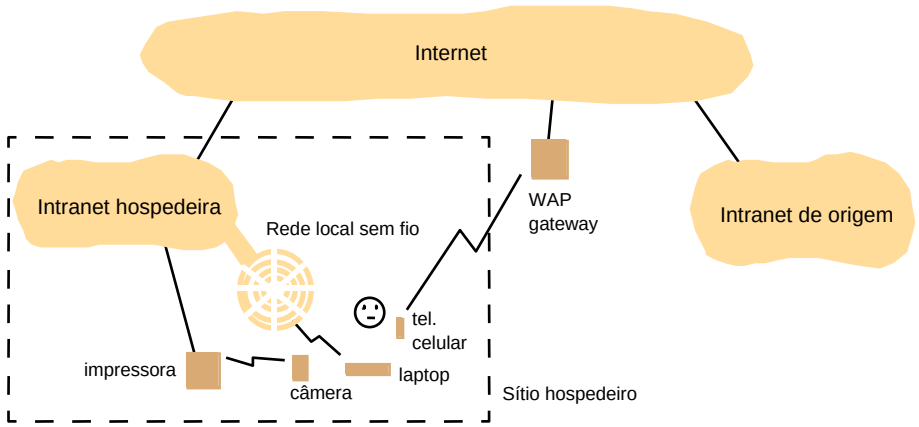
Intranet



Intranet

- Principais características
  - Parte da Internet administrada separadamente e cuja fronteira pode ser configurada de modo a impor políticas de segurança locais
  - Pode conter várias redes locais (LANs) conectadas através de backbones de alta velocidade
  - Roteador permite que usuários de dentro da intranet acessem serviços externos, como email e a Web, e que usuários externos acessem os serviços que a Intranet oferece
  - Firewall protege a intranet, prevenindo a entrada ou saída de mensagens não autorizadas
    - Para aumentar a segurança, algumas intranets podem ser completamente isoladas da Internet, dispensando a necessidade do Firewall

Ambiente de rede sem fio



Ambiente de rede sem fio

- Principais características
  - Infra-estrutura de rede sem fio para comunicação entre dispositivos móveis (laptops, PDAs, celulares, câmeras, etc) e/ou embutidos (automóveis, eletrodomésticos, máquinas de ponto de venda, etc)
  - Computação móvel:
    - Execução ininterrupta de serviços computacionais enquanto o usuário se desloca através de seu meio físico, ou visita outros ambientes diferentes do seu ambiente original
  - Computação consciente de localização (“location-aware”):
    - Acesso aos recursos convenientemente mais próximos
  - Computação ubíqua:
    - Recursos computacionais “despercebidos” na maioria dos dispositivos por estarem intimamente ligados à sua funcionalidade física
    - Acesso à rede tão comum quanto eletricidade?

Agenda

- Definição e motivação
- Exemplos de sistemas distribuídos
- Compartilhamento de recursos e a Web
- Desafios de projeto

Compartilhamento de recursos

- Um sistema distribuído oferece uma grande de variedade de recursos e de formas de compartilhá-los
  - Recursos físicos (impressoras, discos, etc)
  - Recursos lógicos (dados, textos, imagens, etc)
  - Ferramentas e aplicações de uso geral (busca de informação, previsão do tempo, conversão de moedas, etc)
- Recursos são disponibilizados na forma de serviços:
  - Acesso restrito a um conjunto de operações que o provedor do serviço (“servidor”) exporta através de uma interface de comunicação
  - Usuários e aplicações (“clientes”) utilizam o serviço invocando as operações definidas na interface de comunicação
  - Um mesmo programa pode assumir o papel de cliente ou servidor, dependendo se ele solicita ou oferece serviços, respectivamente

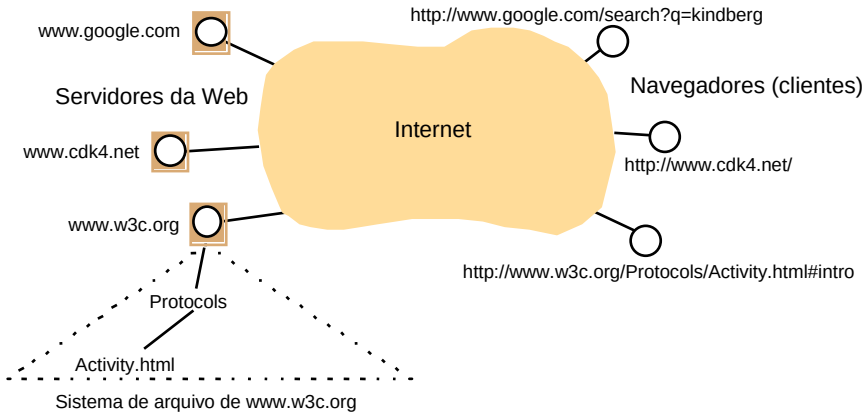
Compartilhamento de recursos na Internet: a Web

- Sistema distribuído para publicação e acesso a recursos e serviços através da Internet
- Criado por Tim Berners-Lee no início dos anos 90, como solução para facilitar o compartilhamento de recursos (documentos) entre os pesquisadores do Centro Europeu de Pesquisas Nucleares (CERN), na Suíça
- Baseado no conceito de hipertexto, proposto na década de 40, onde documentos são descritos contendo referências explícitas (*links*) para outros documentos, permitindo uma leitura (navegação) não linear do seu conteúdo

Compartilhamento de recursos na Internet: a Web

- Arquitetura cliente-servidor de natureza totalmente aberta, baseada em três importantes padrões tecnológicos:
  - HTML – descrição do conteúdo e do formato de visualização dos recursos (páginas) que são acessados pelos clientes
  - URL – identificação e localização dos recursos mantidos pelos servidores
  - HTTP – definição das regras de comunicação e interação entre clientes e servidores
- Contínua adição de novas tecnologias
  - Páginas dinâmicas (ASP, JSP, PHP, AJAX)
  - Mobilidade de código (Applets, Flash scripts)
  - XML (XMLSchema, XPath, XQuery, XSLT)
  - Web Services (SOAP, WSDL, UDDI)
  - Web Semântica (RDF)
  - ?

Compartilhamento de recursos na Web



A Web em discussão

- Sucesso fenomenal, jamais previsto por seus criadores ou por qualquer “guru de tecnologia” do passado
- Principais razões:
  - Relativa facilidade com que muitos indivíduos e organizações conseguem publicar e disponibilizar recursos na Internet
  - Conveniência da estrutura de hipertexto para organizar muitos tipos de informação e conteúdo
  - Grande abertura propiciada pela sua arquitetura cliente-servidor

A Web em discussão

- Apesar do sucesso, a Web está longe da perfeição e apresenta uma série de problemas de ordem prática
- Principais limitações:
  - Alto risco das referências ficarem desatualizadas (*broken links*)
  - Usuários “perdidos no (hiper)espaço”
  - Motores de busca ainda bastante imperfeitos para encontrar as informações que os usuários realmente desejam
  - Gargalos típicos da arquitetura cliente-servidor podem acarretar problemas de desempenho e escalabilidade
  - Páginas HTML insuficientes como interface de interação com o usuário (inclusão de imagens e scripts diminuem o tempo de acesso)

Agenda

- Definição e motivação
- Exemplos de sistemas distribuídos
- Compartilhamento de recursos e a Web
- Desafios de projeto

Desafios de projeto

- Questões importantes que devem ser consideradas cuidadosamente durante o projeto e desenvolvimento de um sistema ou aplicação distribuída
- Ilustrados aqui através de cenários hipotéticos do tipo “e se...” aplicados a um exemplo “real” de comércio eletrônico
- Exemplo: Livraria Online
  - Clientes podem se conectar de seus computadores pessoais ao computador responsável pela livraria (servidor web) para:
    - Consultar o catálogo de livros em estoque
    - Fazer pedidos de compra
    - …

Desafios de projeto

- E se…
  - Os clientes utilizarem diferentes arquiteturas de hardware (PC, MAC, …)?
  - ... ou diferentes sistemas operacionais (Windows, Unix,…)?
  - ... ou diferentes formatos de representação dos dados (ASCII, EBCDIC,…)?
  - **Heterogeneidade**
- E se…
  - Você quiser mudar o seu negócio e todos os seus computadores para Fernando de Noronha (por causa do clima)?
  - ... ou todos os seus clientes mudarem para Fernando de Noronha?
  - **Transparência de distribuição**

Desafios de projeto

- E se…
  - Dois clientes quiserem comprar o mesmo livro ao mesmo tempo?
  - **Concorrência**
- E se…
  - O banco de dados com todo os seus dados de estoque falhar?
  - … ou os computadores dos seus clientes falharem bem no meio de uma compra?
  - **Tolerância a falhas**

Desafios de projeto

- E se…
  - Alguém tentar invadir seus sistema para roubar informações importantes (e.g., dados dos cartões de crédito dos clientes)?
  - … ou os seus clientes fizerem pedidos e depois não aceitarem a entrega dos produtos dizendo que não pediram nada?
  - **Segurança**
- E se…
  - O seu negócio fizer tanto sucesso que milhões de usuários passam a visitar o sítio da sua livraria na web ao mesmo tempo?
  - **Escalabilidade**

Desafios de projeto

- Quando for construir o seu sistema…
  - Você vai querer escrever todo o software vocês mesmo (comunicação em rede, banco de dados, …)?
  - O que fazer quando precisar realizar atualizações e/ou adquirir novas tecnologias?
  - **Reuso** e **Abertura** (padrões)
- Ajudar os projetistas e programadores de aplicações distribuídas a enfrentar essa série de desafios é a principal motivação para o desenvolvimento das tecnologias de middleware atuais!

Heterogeneidade

- Variedade e diferença em termos de:
  - Hardware
  - Sistemas operacional
  - Rede
  - Linguagem de programação
  - Fabricante
- Exemplos de heterogeneidade na Internet
  - Diferentes implementações do mesmo conjunto de protocolos para diferentes tipos de rede: IP, TCP, UDP, SMTP
  - Diferentes padrões de representação de dados: IDL, XML
  - Diferentes padrões de bibliotecas: POSIX, DLL
  - Diferentes padrões de invocação de serviços: COM, CORBA, RMI, SOAP
  - Diferentes plataformas de execução: JVM (Java), CLR (.NET)

Abertura

- Facilidade de extensão e atualização
  - Adição de novos recursos e serviços
  - Re-implementação de serviços existentes
- Depende que as interfaces de acesso aos principais componentes do sistemas sejam conhecidas e estejam disponíveis para os programadores
- Exemplos de abertura na Internet
  - Especificações controladas e atualizadas por um Comitê Gestor
  - Novos produtos e serviços implementados de acordo com as especificações vigentes
  - Conformidade da implementação deve ser testada e verificada para garantir o correto funcionamento do sistema (Quem verifica?)

Segurança

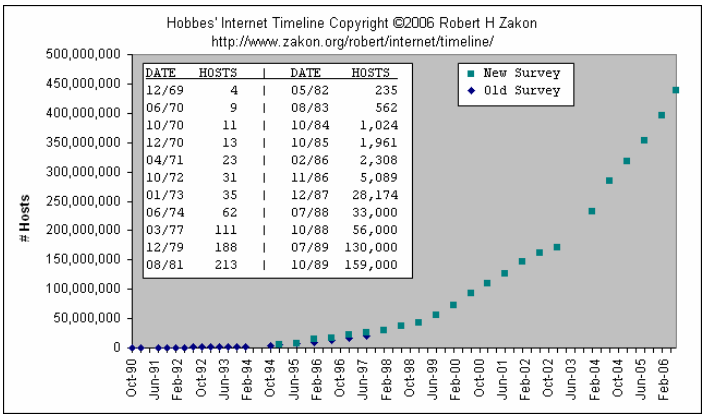
- Proteção para recursos compartilhados
  - Confidencialidade (proteção contra usuários não autorizados)
    - Ex.: Acesso a dados sobre salário, histórico médico, preferências sexuais
  - Integridade (proteção contra alteração e corrupção)
    - Ex.: Alteração indevida de dados usados em transações bancárias
  - Disponibilidade (proteção contra interferência ao meio de acesso)
    - Ex.: Queda ou sobrecarga do servidor ou do meio de comunicação
- Principais mecanismos de segurança na Internet
  - Firewall
  - Assinaturas digitais
  - Canais de seguros de comunicação
- Desafios recentes
  - Ataques de negação de serviço
  - Segurança para código móvel

Escalabilidade

- Capacidade do sistema permanecer operando de forma efetiva mesmo diante de um aumento significativo do número de usuários e/ou dos recursos disponíveis
- Principais desafios:
  - Controlar o custo dos recursos físicos
  - Controlar perdas de desempenho (ex.: DNS)
  - Prevenir o esgotamento dos recursos de software (ex.: endereços IP)
  - Evitar “gargalos” de desempenho na rede ou nos próprios servidores
- Principais técnicas:
  - Replicação
  - Caching
  - Concorrência e paralelismo

Escalabilidade da Internet

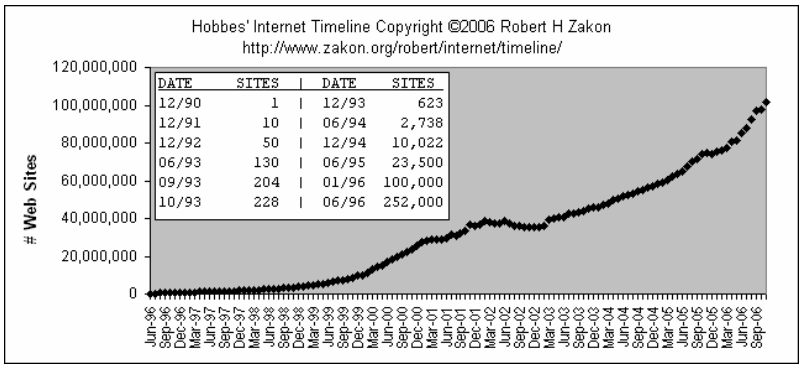
- Crescimento do número de computadores conectados



DATE	HOSTS	DATE	HOSTS
12/69	4	05/82	235
06/70	9	08/83	562
10/70	11	10/84	1,024
12/70	13	10/85	1,961
04/71	23	02/86	2,308
10/72	31	11/86	5,089
01/73	35	12/87	28,174
06/74	62	07/88	33,000
03/77	111	10/88	56,000
12/79	188	07/89	130,000
08/81	213	10/89	159,000

Escalabilidade na Internet

- Crescimento do número de servidores web



Escalabilidade na Internet

- Taxa de expansão da Web

Data	*Computadores*	*Servidores Web*	*Porcentagem*
1993, Julho	1,776,000	130	0.008
1995, Julho	6,642,000	23,500	0.4
1997, Julho	19,540,000	1,203,096	6
1999, Julho	56,218,000	6,598,697	12
2001, Julho	125,888,197	31,299,592	25
2003, Janeiro	171,638,297	35,424,956	21
2005, Julho	353,284,187	67,571,581	19
2006, Julho	439,286,364	88,166,395	20

Fonte: info.isoc.org

Tolerância a falhas

- Falhas são inevitáveis em sistemas computacionais
  - Resultados incorretos
  - Interrupção não planejada do serviço antes de sua conclusão
- Falhas em sistemas distribuídos são parciais (por quê?)
- Técnicas de tratamento de falhas mais comuns:
  - Detecção (ex. bits de paridade)
  - Ocultamento (ex. retransmissão de mensagens)
  - Tolerância (ex. informar o usuário do problema)
  - Recuperação (ex. transações em BD's)
  - Redundância (ex. replicação de tabelas no DNS)
- Sistemas distribuídos devem oferecer alta disponibilidade de recursos mesmo diante da ocorrência de falhas
  - Disponibilidade: medida da proporção do tempo que um recurso está disponível para uso

Concorrência

- Suporte para múltiplos acessos simultâneos a um ou mais recursos compartilhados
  - Possibilidade de inconsistências quando os recursos são alterados
- Serviços que representam recursos compartilhados devem ser responsáveis por garantir que as operações de acesso os mantenham em um estado consistente
  - Válido para servidores e objetos de aplicações
- Técnicas mais comuns:
  - Sincronização de acesso (ex.: exclusão mútua distribuída)
  - Protocolos de controle de concorrência (ex.: 2PC)

Transparência

- Abstração para os usuários e programadores de aplicação da separação física dos recursos em um sistema distribuído
  - Sistema percebido como um “todo” coerente ao invés de uma coleção de partes independentes
- Formas de transparência
  - *Transparência de acesso*: permite o acesso a componentes remotos e locais através das mesmas operações
  - *Transparência de localização*: permite o acesso a componentes sem conhecimento da sua localização física
  - *Transparência de concorrência*: permite a execução concorrente de múltipla operações sobre o mesmo conjunto de recursos sem causar interferência entre elas
  - *Transparência de replicação*: permite usar múltiplas instâncias de um mesmo recurso lógico sem conhecimento da existência de réplicas pelos usuários e programadores

Transparência

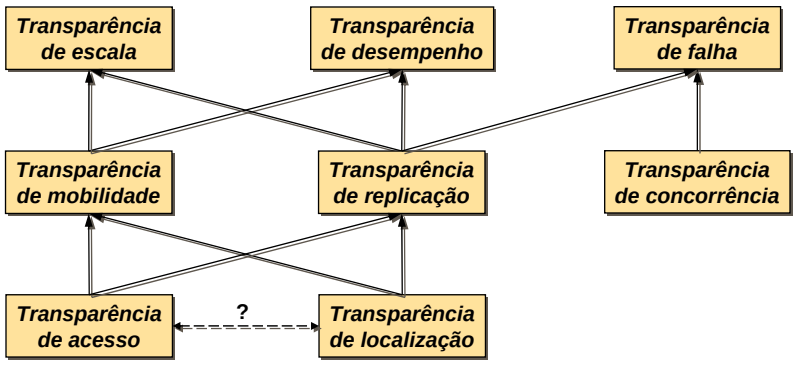
- Formas de transparência (cont.)
  - *Transparência de falha*: permite esconder a ocorrência de falhas dos usuários e programadores
  - *Transparência de mobilidade (migração)*: permite a re-alocação de recursos e aplicações sem afetar o seu uso
  - *Transparência de desempenho*: permite a re-configuração do sistema para aumentar o seu desempenho conforme varia a carga de trabalho
  - *Transparência de escala*: permite a expansão do sistema e de suas aplicações sem exigir mudanças significativas na infra-estrutura existente
- As duas formas mais importantes são *acesso* e *localização*!
  - Suas presenças (ou ausências) afetam profundamente a maneira como os recursos são utilizados em um sistema distribuído
  - Também conhecidas conjuntamente como *transparência de rede*

Transparência

- Exemplos de transparência:
  - Ferramenta para “exploração” de arquivos que mantêm as mesmas opções de navegação para pastas locais e remotas
  - API para acessar dados que utiliza as mesmas operações para dados locais e remotos
- Exemplos de falta de transparência:
  - Sistema distribuído onde só é possível acessar arquivos remotos via FTP
  - Serviço de jogos online que precisa ser tirado do ar para acrescentar ou trocar um servidor
- Classificação quanto à dificuldade de implementação (hierarquia de dependência) e nível (usuário ou programador)

Transparência

- Hierarquia de dependência



Fonte: Wolfgang Emmerich, *Engineering Distributed Objects*, Wiley, 2000.

Transparência

- Níveis de transparência
  - Nível do usuário: distribuição física dos recursos é imperceptível para os usuários das aplicações (ex.: navegador da Web)
  - Nível do programador: distribuição física dos recursos é imperceptível tanto para os usuários quanto para os programadores das aplicações (ex.: programação com middleware ou SO distribuído)
- Importante: transparência total pode ser indesejável ou até mesmo impossível na prática!! (Por quê?)

Exercícios

- Pesquise na literatura/Web definições para os seguintes tipos de sistema distribuídos:
  - Cluster
  - Grid
- Discuta sob quais aspectos as definições que você encontrou são semelhantes ou contrastantes, tanto entre si quanto em relação às definições de sistema distribuído vistas em sala
- Pesquise na literatura/Web quais tipos de transparência são oferecidos pelas seguintes tecnologias de middleware:
  - EJB
  - .NET
  - CORBA
- No livro: 1.1–1.13