

Modelos Arquiteturais

Carlos Ferraz
cagf@cin.ufpe.br

Tópicos da Aula

- Modelos arquiteturais
 - Cliente-servidor
 - *Peer process*
 - Objetos distribuídos

O que é um modelo arquitetural?

- Estrutura em termos de componentes especificados separadamente
- Alocação de componentes em uma rede de computadores
- Interrelações de componentes
- Divisão de responsabilidades entre componentes

Terminologia

- Servidor – processo que aceita pedidos de outros processos
- Serviço – provido por um ou mais servidores
- Plataforma – hardware e sistema operacional
- *Middleware* – camada de software que esconde heterogeneidade e fornece um modelo de programação

Camadas de software e hardware em SDs

Applications, services

Middleware

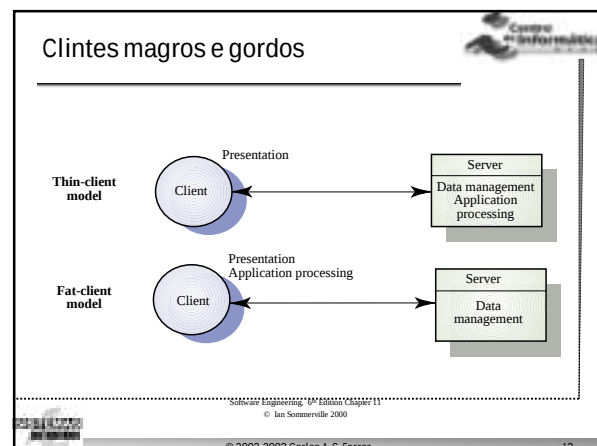
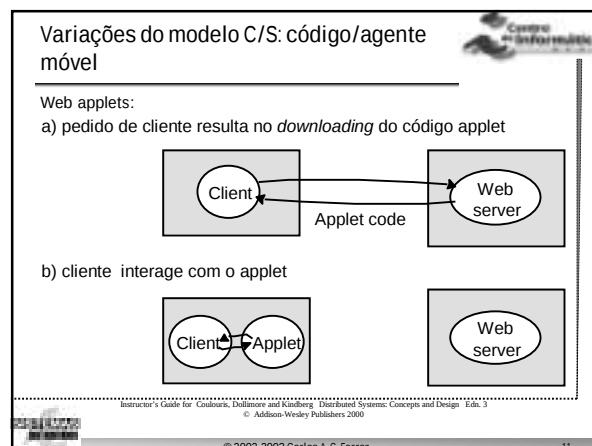
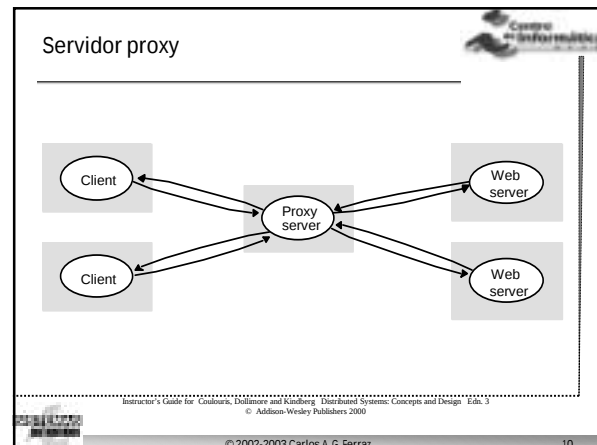
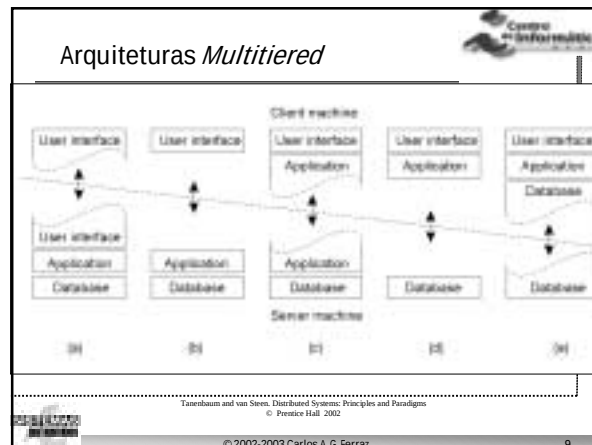
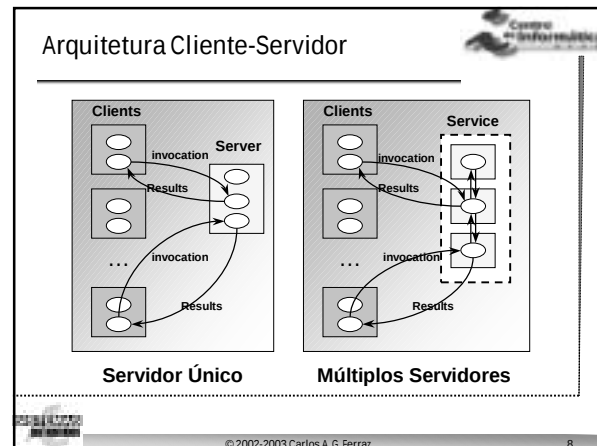
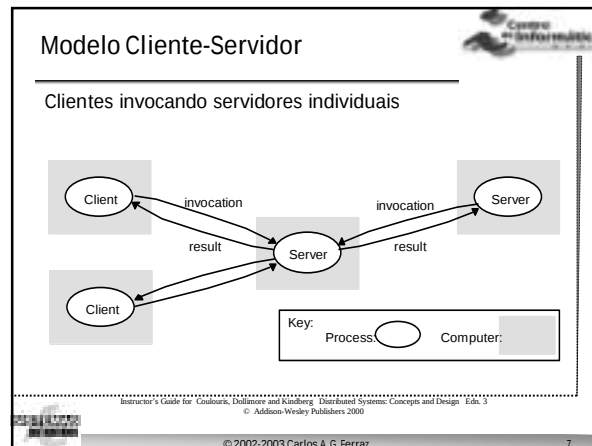
Operating system

Computer and network hardware

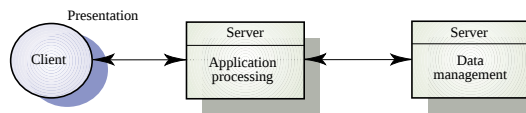
Platform

Illustration's Castle for Coulson, Dollimore and Kindberg. Distributed Systems: Concepts and Design. Edn. 3. © Addison-Wesley Publishers 2000

Arquiteturas

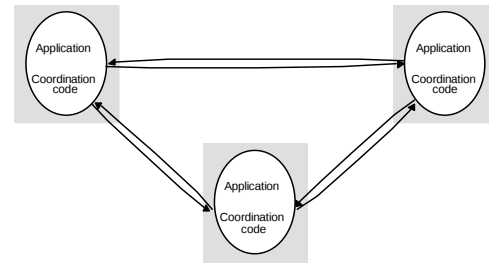


Arquitetura C/S 3-tier



© 2002-2003 Carlos A. G. Ferraz

13

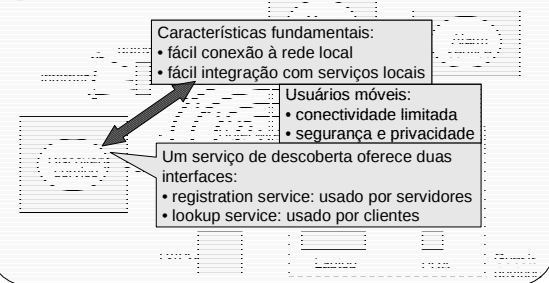
Arquitetura de Peer processes
(Peer-to-Peer)

© 2002-2003 Carlos A. G. Ferraz

14

Exemplo de um SD em um hotel

Spontaneous Networking



© 2002-2003 Carlos A. G. Ferraz

15

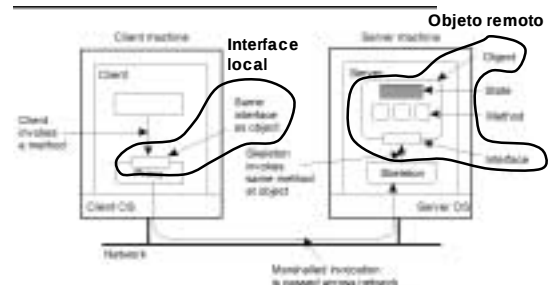
Objetos Distribuídos

- Uma aplicação distribuída pode ser vista como um conjunto de objetos
- **Objetos:**
 - Consistem de dados + código
 - Podem ser clientes, servidores ou ambos
 - Interface esconde detalhes de implementação
 - Modelar com objetos não implica no uso de programação orientada a objetos
- **Middleware:**
 - *Object brokers*: permitem que objetos se encontrem em um sistema distribuído, e interajam uns com os outros
 - *Object services*: permitem criar, nomear, mover, copiar, armazenar, deletar, restaurar e gerenciar objetos

© 2002-2003 Carlos A. G. Ferraz

16

Objetos Distribuídos

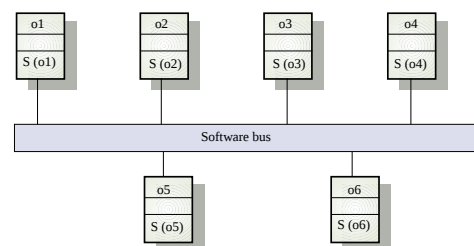


- Observe a separação entre interface e objeto

© 2002-2003 Carlos A. G. Ferraz

17

Arquitetura de Objetos Distribuídos



© 2002-2003 Carlos A. G. Ferraz

18

Ligação (Binding) Cliente-Objeto

■ Ligação implícita

```
Distr_object* obj_ref;           // Declare a systemwide object reference
obj_ref = ...;                  // Initialize the reference to a distributed object
obj_ref->do_something();         // Implicitly bind and invoke a method
```

■ Ligação explícita

```
Distr_object objPref;           // Declare a system-wide object reference
Local_object* obj_ptr;          // Declare a pointer to local objects
obj_ref = ...;                  // Initialize the reference to a distributed object
obj_ptr = bind(obj_ref);        // Explicitly bind and obtain pointer to local proxy
obj_ptr->do_something();         // Invoke a method on the local proxy
```

Tanenbaum and van Steen, Distributed Systems: Principles and Paradigms
© Prentice Hall, 2002

© 2002-2003 Carlos A. G. Ferraz

19

Comentários finais (1/2)

- O uso de objetos distribuídos melhora a capacidade de manutenção e adaptabilidade de um sistema
- Arquiteturas cliente-servidor fornecem uma infra-estrutura versátil que suporta a inserção de novas tecnologias mais rapidamente
- Arquiteturas de software cliente-servidor têm sido usadas desde os anos 80 → maturidade

© 2002-2003 Carlos A. G. Ferraz

20

Comentários finais (2/2)

- Um número de *tradeoffs* deve ser considerado para selecionar a arquitetura cliente-servidor apropriada, incluindo:
 - O crescimento potencial do número de usuários,
 - Custo e
 - Homogeneidade do ambiente computacional futuro e do momento

© 2002-2003 Carlos A. G. Ferraz

21