

Sistemas Computacionais Distribuídos

**Prof. Marcos José Santana
SSC-ICMC-USP**

São Carlos, 2008

Grupo de Sistemas Distribuídos e Programação Concorrente

**Departamento de Sistemas
de Computação - SSC**

Sistemas Computacionais Distribuídos

3a. Aula

Projeto de S.D.

Conteúdo

Sistemas Distribuídos

- ⇒ Características
- ⇒ Como atingir essas características
- ⇒ Itens a serem considerados no projeto
 - Nomeação
 - Comunicação
 - Estrutura de software
 - Alocação de carga
 - Manutenção de consistência

Características

- ◆ **Compartilhamento de recursos;**
- ◆ **Abertura;**
- ◆ **Concorrência;**
- ◆ **Escalabilidade;**
- ◆ **Tolerância a falhas;**
- ◆ **Transparência.**

Características

◆ **Compartilhamento de recursos:**

- **Fundamental em sistemas distribuídos;**
- **Elementos de hardware:**
 - **Discos**
 - **Impressoras**
- **Elementos lógicos:**
 - **Dados**
 - **Arquivos**
 - **mensagens**

Características

◆ Abertura:

- Facilidade de extensão;
- Facilidade de adaptação a evolução:
 - Física;
 - Lógica.
- Obtém-se com:
 - Interfaces bem documentadas e disponibilizadas;
 - Adoção de padrões, sempre que possível;
- Historicamente sistemas fechados!
- Sistemas distribuídos ⇒ mudança de paradigma!

Características

◆ Abertura:

— Hardware:

- Introdução de novos elementos;
- Adaptação a novas tecnologias;
- Etc.

— Software:

- Atualizações de software;
- Atualizações de protocolos de comunicação;
- Inserção de serviços compartilhados;
- Etc.

Características

- ◆ **Concorrência:**
 - **Vários usuários**
 - **Vários processos clientes**
 - **Vários processos servidores**
 - **Trabalho concorrente**
 - **Recursos compartilhados**
 - **Acesso a dados compartilhados**

Características

♦ Escalabilidade:

- Capacidade de crescer (e)
- Manutenção do desempenho
- Limitação da rede de comunicação
- Limitação do modelo arquitetural
- Duas estações e um servidor (até)
- Milhares de estações e dezenas de servidores

Características

- ◆ **Tolerância a falhas:**
 - **Redundância em nível de hardware**
 - Existência de múltiplos recursos capazes de oferecer o mesmo serviço.
 - **Recuperação em nível de software**
 - Existência de procedimentos para recuperar (consistentemente) após falhas.

Características

- ♦ **Tolerância a falhas:**
 - **Granulosidade fina:**
 - **Replicação (física) de servidores.**
 - **Alta disponibilidade**
 - **Aumento de complexidade**

Características

- ♦ **Transparência:**
 - **Característica (conceito) fundamental**
 - **Visão uniforme e centralizada do sistema**
 - **Mobilidade dos usuários**
 - **Diversas formas de transparência**

Características

♦ Transparência:

– Formas de transparência:

- **Acesso** ⇒ acesso local e remoto semelhantes;
- **Localização** ⇒ objetos remotos sem saber onde estão;
- **Concorrência** ⇒ concorrência a dados consistência;

Características

♦ Transparência:

– Formas de transparência:

- **Replicação** ⇒ várias instâncias aumentando confiabilidade, desempenho, disponibilidade sem conhecimento dos usuários e/ou programas aplicativos;
- **Falhas** ⇒ esconder falhas de hardware e de software

Características

♦ Transparência:

– Formas de transparência:

- **Migração** ⇒ migração de dados sem consequência para usuários e/ou programas aplicativos;
- **Desempenho** ⇒ reconfiguração de acordo com a carga;
- **Escala** ⇒ crescimento em escala sem alterar estrutura do sistema ou dos aplicativos

Como atingir estas características

♦ **Objetivos para um bom projeto:**

- ⇒ **Desempenho;**
- ⇒ **Confiabilidade;**
- ⇒ **Escalabilidade;**
- ⇒ **Consistência;**
- ⇒ **Segurança.**

Itens a serem considerados no projeto:

- ⇒ Nomeação;
- ⇒ Comunicação;
- ⇒ Estrutura de software;
- ⇒ Alocação de carga;
- ⇒ Manutenção de consistência.

Itens a serem considerados no projeto:

Nomeação

- ⇒ Compartilhamento de recursos;
- ⇒ Transparência.

✓ Recursos ou objetos devem receber nomes:

- Significado global.
- Independente da localização.

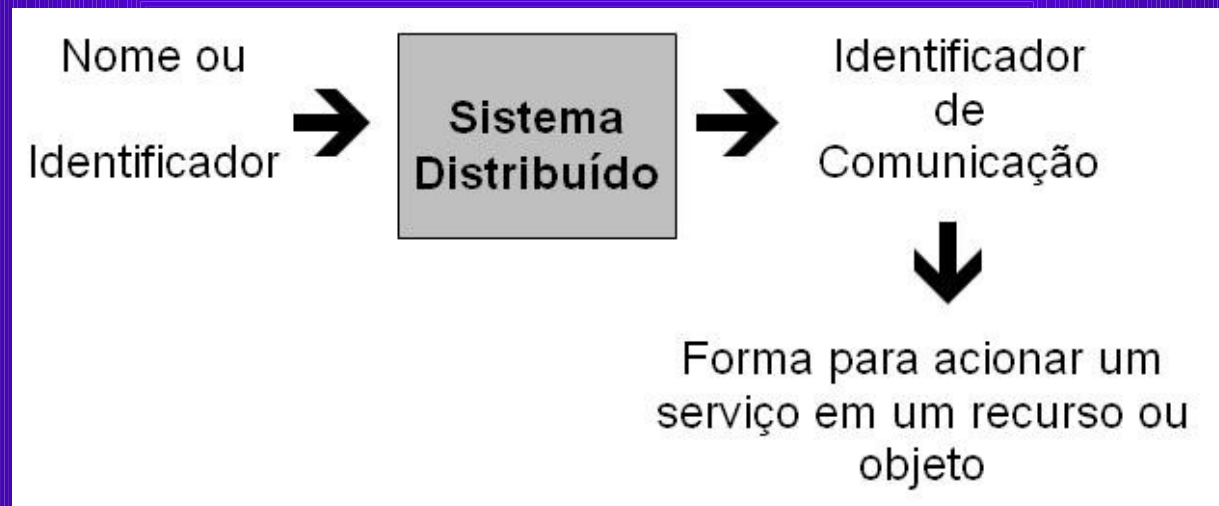
✓ Sistema de interpretação de nomes:

- Escalabilidade.
- Desempenho.

Itens a serem considerados no projeto:

Nomeação

* Em um Sistema Distribuído:



* Importância do contexto:

⇒ Hierarquia - Domínios

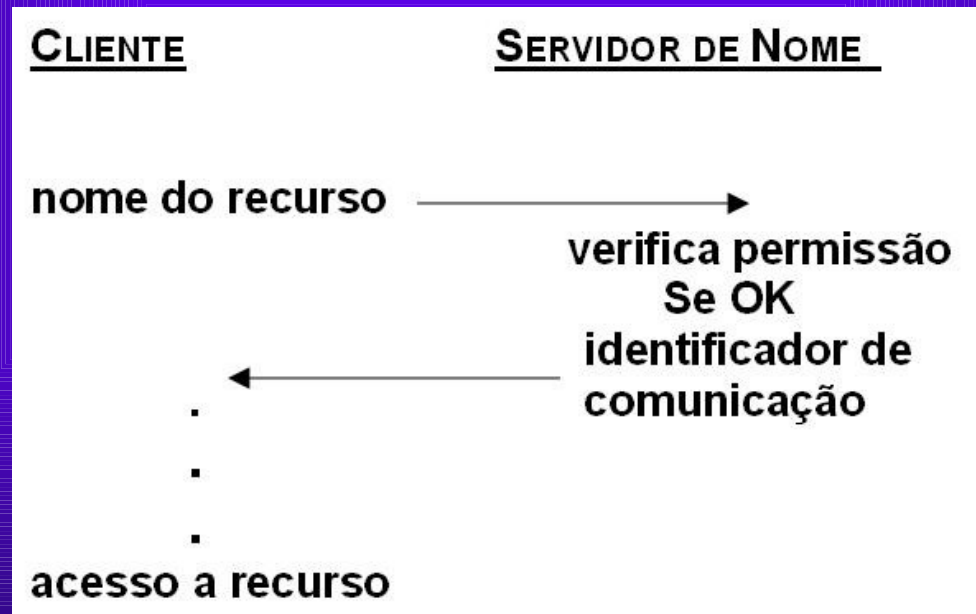
Itens a serem considerados no projeto:

Nomeação

* Proteção:

⇒ Servidor de Nome

- Protege recurso de uso não autorizado!



✓ Para primeiro acesso ao recurso.

Itens a serem considerados no projeto:

Comunicação

⇒ Desempenho;

⇒ Confiabilidade.

* S.D.: vários processos que requisitam ou fornecem serviços.

⇒ Comunicação entre processos

✓ Transferência de dados;

✓ Sincronização.

Itens a serem considerados no projeto:

Comunicação

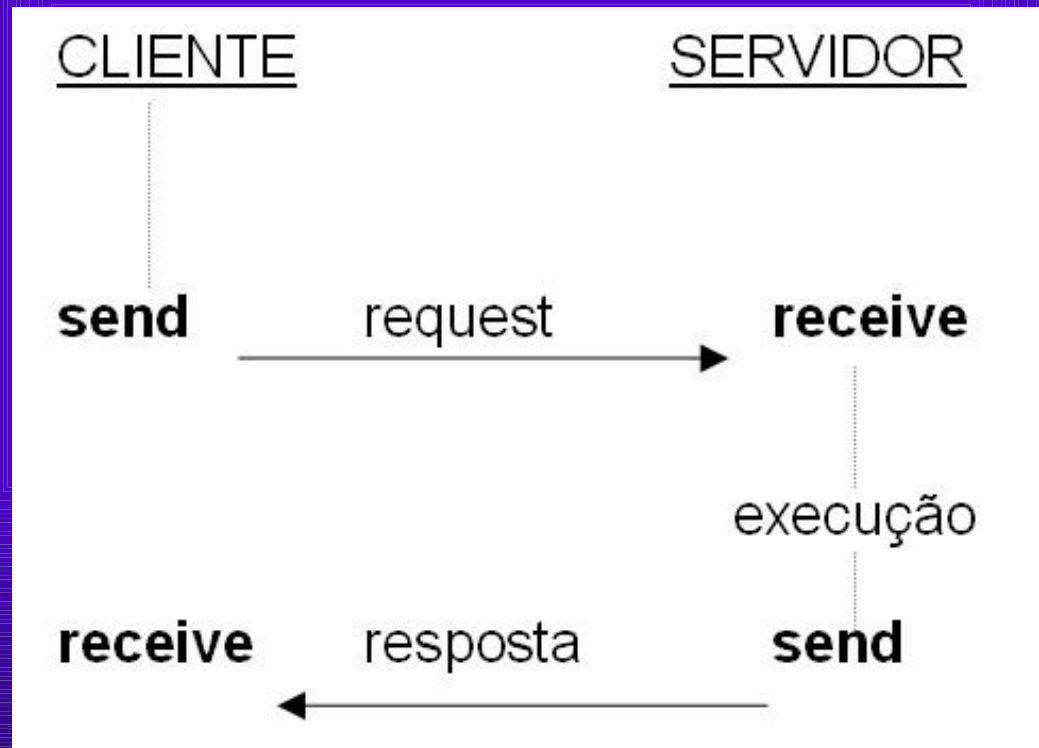
⇒ Troca de mensagens

- ✓ Send / Receive
- ✓ Bloqueante / Não bloqueante
- ✓ Cliente / Servidor
- ✓ Multicast

Itens a serem considerados no projeto:

Comunicação

* Comunicação Cliente / Servidor



Itens a serem considerados no projeto:

Comunicação

* **RPC - Chamada a Procedimento Remoto**

* **MULTICAST:**

1 proc	⇒	n procs
1 group Send	⇒	n receives

Ex.:

- Atualização - relógio.
- Tolerância a falha.
- Localização de objetos

Itens a serem considerados no projeto:

Estrutura do Software

- ✓ **Abertura.**
- ✓ **Componentes de software com interfaces bem definidas.**
 - **Serviços: gerenciadores de objetos de um dado tipo.**
 - **Interface: conjunto de operações.**
- ✓ **Introdução de novos serviços.**

Itens a serem considerados no projeto:

Estrutura do Software

* Sistema Centralizado

✓ Organização:

- Aplicações
- Suporte para linguagem de programação
- Sistema Operacional
- Hardware

✓ Gerenciamento de recursos:

- memória;
- criação de processos;
- periféricos

✓ Serviços para usuários:

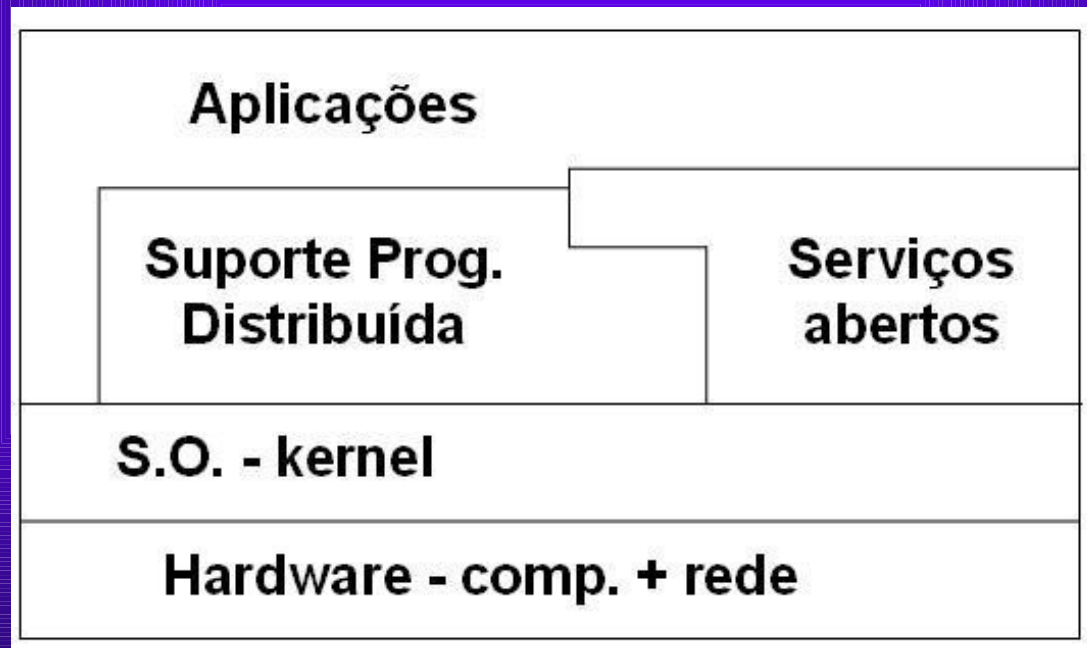
- autenticação;
- controle de acesso;
- gerenciamento de arquivos;
- clock.

Itens a serem considerados no projeto:

Estrutura do Software

* Sistemas Distribuídos

✓ Organização:



Itens a serem considerados no projeto:

Estrutura do Software

* Sistemas Distribuídos

- ✓ S.O.:
 - memória;
 - criação de processos;
 - comunicação entre processos;
 - periféricos.

- ✓ Serviços abertos:
 - serviços compartilhados.

- ✓ Suporte para programação distribuída:
 - RPC;
 - Multicast.

Itens a serem considerados no projeto:

Alocação de Carga

- * **Como utilizar**

- ⇒ Processamento e comunicação

- ⇒ Recursos

para obter o melhor desempenho!

- * **Escalonamento de tarefas em S.D.**

- ⇒ escalonamento local

- ⇒ escalonamento global

Itens a serem considerados no projeto:

Alocação de Carga

* Modelos Arquiteturais

- ⇒ Modelo E.T. / Servidor
- ⇒ Modelo E.T. / Servidor + E.T. ociosas
- ⇒ Pool de processadores
- ⇒ outras organizações

Itens a serem considerados no projeto:

Manutenção de consistência

⇒ **Confiabilidade**

⇒ **Disponibilidade**

⇒ **Desempenho**

(Cache, replicação, falha, clock, etc.)

* **Consistência de manutenção de dados**

⇒ **Modificação Atômica.**

* **Consistência em Replicação**

⇒ **Sincronismo entre múltiplas cópias.**

Itens a serem considerados no projeto:

Manutenção de consistência

- * **Consistência em cache**
 - ⇒ Cache nas E.T.
- * **Consistência em falha**
 - ⇒ Dados em diferentes componentes ⇒ Compatíveis.
- * **Consistência de clock**
 - ⇒ Timestamp;
 - ⇒ Tempo de transmissão;
 - ⇒ Clock Lógico.

Fim!