




Plano de Ensino



- **Introdução: Conceitos e Definições.**
- Processos, Comunicação e Sincronização de Sistemas Distribuídos.
- Processamento Paralelo e Multiprocessadores.
- Tolerância a falhas.
- Aplicações Distribuídas: Socket.
- Objetos Distribuídos: RMI.



Livro-Texto



- **Bibliografia Básica:**
 - » TANENBAUM, Andrew S; STEEN, Maarten Van. Sistemas Distribuídos : Princípios e Paradigmas. 2ª Ed. São Paulo: Pearson Prentice Hall, 2007.
- **Bibliografia Complementar:**
 - » DANTAS, Mario. Computação Distribuída de Alto Desempenho : redes, clusters e grids computacionais. 1ª Ed. Rio de Janeiro: Axcel Books, 2005.
 - » SILBERSCHATZ, Abraham; GALVIN, Peter Baer; GAGNE, Greg. Sistemas Operacionais - Conceitos e Aplicações. 1ª Ed. Rio de Janeiro: Campus, 2001.

1. Introdução – Definições



- Um Sistema Distribuído é aquele no qual os componentes localizados em computadores interligados em rede se comunicam e coordenam suas ações com troca de mensagens.
- Algumas características:
 - » Concorrência entre componentes.
 - » Falta de relógio global.
 - » Falhas de componentes independentes.
 - » Compartilhamento de recursos.
 - » Heterogeneidade: arquiteturas, SOs, interfaces.
 - » Sistema aberto: segurança.
 - » Escalabilidade: transparência.
 - » Tratamento de falhas.

1. Introdução – Definições



- Segundo Tanenbaum:
 - » "Um sistema distribuído é uma coleção de computadores independentes que aparece para os usuários do sistema como um único e coerente sistema".
- Segundo Coulouris:
 - » "Coleção de computadores autônomos interligados através de uma rede de computadores e equipados com *software* que permita o compartilhamento dos recursos do sistema: *hardware*, *software* e dados".

1. Introdução – Definições



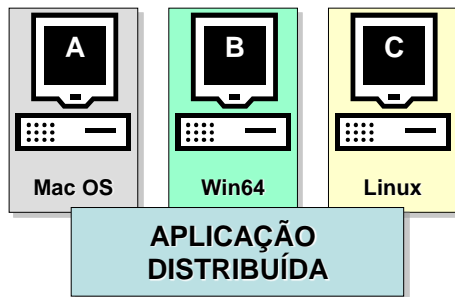
- As definições anteriores focam 2 aspectos importantes:
 - » Hardware: as máquinas são autônomas.
 - » Software: os usuários pensam que estão trabalhando em um sistema único.

1. Introdução – Definições



- Desta forma podemos pensar em alguns pontos importantes em Sistemas Distribuídos:
 - » Compartilhamento de recursos.
 - » Processamento descentralizado.
 - » Expansão ou escalonamento fácil.
 - » Organização em camadas de *software* logicamente alocadas → uma camada de alto nível (usuários e aplicações) e uma camada de baixo nível (SO).
 - » Permitem a integração de vários sistemas, através da comunicação entre componentes.

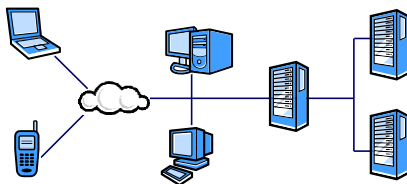
1. Introdução – Definições



1. Introdução – Exemplos



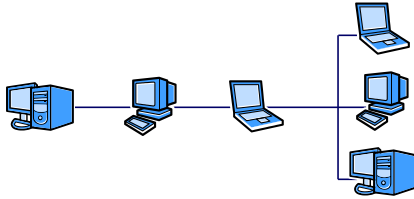
- Exemplo 1: Internet



1. Introdução – Exemplos



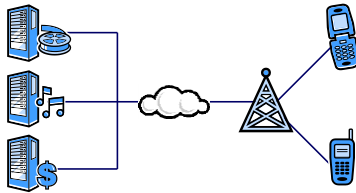
▪ Exemplo 2: LAN



1. Introdução – Exemplos



▪ Exemplo 3: Computação Móvel e Ubíqua.



1. Introdução – Objetivos



- Compartilhamento de recursos
- Abertura
- Concorrência
- Escalabilidade
- Robustez
 - » Tolerância a falhas
 - » Disponibilidade
- Transparência

1. Introdução – Objetivos



▪ Compartilhamento de recursos

- » É o principal objetivo de um sistema distribuído: usuários ↔ recursos remotos.
- » Recurso: caracteriza o conjunto de elementos que podem ser compartilhados de forma útil em um SD:
 - Hardware: impressoras, discos, ...
 - Software: arquivos, bancos de dados, compiladores, ...
- » Facilita a colaboração e troca de informações.
- » O compartilhamento reduz custos.
- » Necessário para dar suporte a grupos de usuários.
- » Segurança → ponto importante.

1. Introdução – Objetivos



- Recursos fisicamente encapsulados em um dos computadores de um SD só podem ser acessados por outros computadores através de **comunicação**.
- Cada conjunto de recursos de um tipo particular deve ser gerenciado por um programa (Gerenciador de Recursos) que oferece uma **interface de comunicação**.
- Uma interface de comunicação em um Gerenciador de Recursos permite que os mesmos sejam:
 - » Consultados:
 - Deve haver um esquema de nomeação para permitir que recursos individuais sejam acessados a partir de qualquer localização.
 - Mapeamento de nomes de recursos em endereços de comunicação.
 - » Atualizados: há necessidade de sincronização de acesso concorrente para garantir consistência da informação.

1. Introdução – Objetivos



▪ Abertura

- » Determina se um SD pode ser estendido de várias maneiras:
 - extensão por hardware
 - extensão por software
- » Sem a interrupção ou duplicação de serviços existentes.
- » Conseguida através da publicação de interfaces, tornando-as disponíveis para desenvolvedores de software.
- » Unix e Linux são um exemplo de sistemas abertos.
- Um sistema distribuído aberto oferece serviços de acordo com regras padronizadas que descrevem a sintaxe e semântica dos serviços.
 - » Serviços em SD são descritos como interfaces que especificam os nomes das funções, tipos de parâmetros, valores de retorno, exceções, etc.

1. Introdução – Objetivos



- » Outro ponto importante da abertura em SD é a *Interoperabilidade* que permite que duas implementações diferentes de dois fabricantes distintos coexistam.
- » Além disso, a *Portabilidade* caracteriza-se como uma aplicação desenvolvida para um sistema distribuído A que pode ser executado, sem modificações em um sistema distribuído B, com a mesma interface.

1. Introdução – Objetivos



▪ Concorrência

- » Concorrência e execução paralela existem em um SD devido:
 - às atividades separadas de usuários,
 - à independência de recursos e,
 - à localização de processos em computadores separados.

1. Introdução – Objetivos



▪ Escalabilidade

- » Pode ser mensurado em 3 dimensões:
 1. Tamanho: deve-se facilmente adicionar mais usuários ou recursos ao sistema.
 2. Localização: recursos e usuários não necessitam estar próximos.
 3. Administração: deve-se ter administração fácil, mesmo que existam muitas peças de administração independentes.
- » Operação eficiente em diferentes escalas.
- » Processamento deve ser independente do tamanho da rede.

1. Introdução – Objetivos



▪ Tolerância a Falha

- » Em um SD, o hardware essencial para a operação contínua de aplicações críticas pode ser replicado.
 - este hardware redundante pode ser usado para atividades não-críticas → quando não há falhas.
- » O software pode ser projetado para recuperar o estado de dados permanentes quando uma falha é detectada.
- » Tipos de falha
 - Transiente: pode acontecer uma vez.
 - Intermitente: acontece de tempos em tempos.
 - Permanente: acontece sempre.

1. Introdução – Objetivos



▪ Disponibilidade

- » Quando um componente falha em um SD, apenas a parte que usa este componente é afetada; além disso, o componente pode ser reinicializado em outro computador.
- » Um SD tem mais partes disponíveis por mais tempo.

1. Introdução – Objetivos



▪ Transparência

- » Um importante objetivo de um sistema distribuído é esconder que seus processos e recursos estão fisicamente distribuídos sobre vários computadores.
- » O conceito de transparência pode ser aplicado sobre vários aspectos.

1. Introdução – Objetivos



Tipos de Transparência

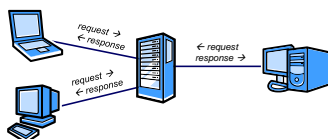
- » Acesso: operações idênticas para acesso local ou remoto.
- » Localização: esconde onde o recurso está localizado.
- » Mobilidade: movimento de recursos sem afetar o sistema.
- » Concorrência: compartilhamento de recursos sem afetar processos concorrentes.
- » Falha: término das tarefas mesmo com falhas previstas ou imprevistas.
- » Replicação: oculta a réplica de recursos de usuários ou desenvolvedores.
- » Desempenho: reconfiguração de desempenho conforme carga de trabalho.
- » Escalabilidade: expansão sem afetar arquitetura existente ou lógica da aplicação.

1. Introdução – Arquiteturas



Cliente-Servidor

- » Neste tipo de arquitetura, uma aplicação cliente conecta-se a uma aplicação servidor criando com isso um canal para troca de informações.
- » Protocolos para transporte de informações: orientado à conexão (TCP - *Transmission Control Protocol*) e não orientada à conexão (UDP - *User Datagram Protocol*).

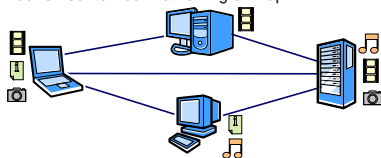


1. Introdução – Arquiteturas



Peer-to-Peer

- » Neste tipo de arquitetura não existe o papel de cliente/servidor, trabalha-se com o conceito de cooperação com seus pares (peers), sem distinção de papéis, nem em quais computadores são executados.
- » Inicialmente os protocolos envolvidos em tais ambientes eram particulares, mas já existem protocolos abertos como: JXTA, Windows Peer-to-Peer Networking e XNap.



1. Introdução – Arquiteturas



- Diversas variações dos modelos anteriores podem ser extraídas a partir da consideração dos seguintes fatores:
 - » o uso de vários servidores e de caches para aumentar o desempenho e a resiliência;
 - » o uso de código móvel e agentes móveis;
 - » a necessidade dos usuários possuírem computadores de baixo custo, com recursos de hardware limitados e simples de gerenciar;
 - » o requisito de adicionar e remover dispositivos móveis de maneira conveniente.

25

1. Introdução – Arquiteturas



- Serviços fornecidos por vários servidores:
 - » Os serviços podem ser implementados como vários processos servidores localizados em diferentes computadores hospedeiros, interagindo conforme necessário para fornecer um serviço para seus clientes.
 - Um serviço em Cluster é um serviço que pode ser particionado por vários servidores garantindo escalabilidade e robustez.
- Servidores proxies e caches:
 - » Armazena os objetos recentemente utilizados mais próximos que os locais reais de origem.
- Código móvel:
 - » Um código é migrado para ser executado localmente pelo seu requisitante.
 - Os applets são os exemplos mais comuns de código móvel.

1. Introdução – Arquiteturas



- Agente móvel:
 - » É um programa em execução (código e dados) que passa de um computador para outro em um ambiente de rede, realizando uma tarefa em nome de alguém, como uma coleta de informações, e finalmente retornando com os resultados obtidos.
- Computadores de rede:
 - » Nesta arquitetura, tanto o SO, quanto os softwares aplicativos sejam carregados a partir de um servidor de arquivos remoto.
- Objetos distribuídos:
 - » Nesta arquitetura vários objetos ficam distribuídos sobre uma arquitetura de computadores em rede interligada por um canal de objetos (ORB - *Object Request Broker*) e podem ser requisitadas (através de uma chamada) por um cliente qualquer.
 - » O protocolo envolvido nesta comunicação é o IIOP (*Internet Inter-ORB Protocol*).

1. Introdução – Arquiteturas



▪ SOA:

- » A arquitetura é uma arquitetura Orientada a Serviços e fornece serviços de **software** fracamente acoplados, para acesso a serviços de negócio por clientes.
- » Esta arquitetura não é direcionada a nenhuma tecnologia e pode ser implementada por qualquer uma existente.
- » O protocolo envolvido nesta tecnologia é o SOAP.

▪ Clientes leves:

- » Refere-se a uma camada de software, em um computador local, que oferece ao usuário uma interface baseada em janelas para que se possa executar programas aplicativos em um computador remoto.
 - WTS (Windows Terminal Server).

1. Introdução – Vantagens



- Economia: microprocessadores oferecem um preço/performance mais atrativo do que mainframes.
- Velocidade: um sistema distribuído pode ter mais poder de processamento que um mainframe.
- Distribuição Física: algumas aplicações envolvem máquinas separadas espacialmente.
- Confiabilidade: caso alguma máquina quebre, o sistema como um todo pode sobreviver.
- Crescimento gradativo: o poder computacional pode ser aumentado de forma gradativa.

1. Introdução – Vantagens



- Compartilhamento de Dados: permite a múltiplos usuários o acesso a um BD comum.
- Compartilhamento de Dispositivos: permite a múltiplos usuários o acesso a dispositivos caros.
- Comunicação: permite a comunicação entre usuários mais fácil.
- Flexibilidade: permite a distribuição de carga de trabalho mais eficiente entre os computadores disponíveis e com um melhor custo efetivo.

1. Introdução – Desvantagens



- Software: os softwares que trabalham em tais ambientes são complexos e ainda em menor número.
- Rede: a rede pode ficar saturada com o tráfego de informações.
- Segurança: a facilidade de acesso a dados compartilhados pode comprometer a segurança dos mesmos.



Sistemas Distribuídos – Aula 01

clayton.valdo@aedu.com