

Multiprocessor systems – Tipo de sistema de computação que possui mais de um processador a trabalhar em conjunto dentro de um único computador. Existe partilha de memória.

Multicomputer systems – Classe de sistemas de computação onde várias unidades de processamento independentes, cada uma com a sua própria memória e recursos, são interligadas numa rede para realizar tarefas de processamento distribuído.

Paralelismo – Técnica de realizar várias tarefas ou operações em paralelo, ou seja, simultaneamente.

O que é um **Sistema Distribuído**?

Um **SD** é aquele em que os seus componentes estão distribuídos por uma rede de computadores, comunicam e coordenam as suas ações através de troca de mensagens, e funciona para o utilizador como se de um sistema elementar (único) se tratasse.

Quais as **características de um SD**?

Pode haver concorrência (cárias tarefas sejam executadas de forma simultânea), não existe um relógio global (cada computador do SD tem o seu próprio relógio) e podem existir falhas isoladas (um disco de um dos computadores falhar).

O que é uma **intranet**?

É uma porção da internet com administração própria, com limites que permitem a aplicação de políticas locais de segurança (Firewall).

O que é um **Sistema Aberto**?

Um sistema aberto é um sistema que pode ser estendido em termos de componentes ou funcionalidades sem perturbação das operações existentes.

O que são **aplicações distribuídas sobre o Middleware**?

Refere-se a programas de software ou sistemas que são projetados para funcionar num SD e fazem uso de uma camada intermédia de software chamada “middleware” para facilitar a comunicação e a coordenação entre diferentes componentes distribuídos.

Características a ter em conta num SD?

Heterogeneidade, abertura, segurança, escalabilidade, resolução de falhas ou problemas, concorrência, transparência.

Cite tipos de recursos de hardware e recursos de dados ou software que possam ser compartilhados com sucesso. Dê exemplos práticos de seu compartilhamento em sistemas distribuídos.

Processadores (podem ser compartilhados para executar tarefas paralelas), memória (para armazenar dados e permitir o acesso a partir de diferentes nós do sistema distribuído), impressoras (para que vários utilizadores consigam aceder à impressora).

Como os relógios de dois computadores ligados por uma rede local podem ser sincronizados sem referência a uma fonte de hora externa? Quais fatores limitam a precisão do procedimento que você descreveu?

É possível sincronizar dois relógios com uso de protocolos de sincronização de tempo internos (protocolo NTP – Um computador é designado como um “servidor de tempo”). A precisão é limitada por fatores como a latência, jitter, sobrecarga de sistemas.

Explique como o **Middleware simplifica a tarefa** de desenvolvimento de aplicações distribuídas.

Ao oferecer uma camada intermediária de software que abstrai muitos dos detalhes complexos associadas à comunicação e à coordenação entre componentes distribuídos.

O que é um **Modelo Físico**?

Representação da camada de hardware onde assenta todo o sistema distribuído. Identifica os diferentes computadores e outros dispositivos pertencentes ao sistema e o modo como estes estão interligados.

Aspetos a considerar no **Desenho de Arquiteturas Distribuídas**?

Desempenho, qualidade do serviço, cache e replicação, segurança e tolerância a falhas.

O que é um **Modelo de Arquitetura**?

Descreve o sistema em termos das tarefas computacionais e comunicação desempenhadas por cada elemento.

Quais os **modelos na arquitetura de SD**?

Cliente-servidor, variantes do modelo cliente-servidor, peer processes, variantes multicamada (two-tier e three-tier).

Variante do **Modelo Cliente-Servidor: Código Móvel** - Refere-se a uma arquitetura em que parte da lógica de processamento do aplicativo é movida para o cliente, permitindo que ele seja executado localmente no dispositivo do cliente.

Arquiteturas multicamada: **Two-tier e Three-tier** - Divide a aplicação em duas partes principais: o cliente (interface do usuário) e o servidor (lógica de negócios e acesso a dados). Divide a aplicação em três partes distintas: o cliente, uma camada intermediária (geralmente chamada de camada de aplicação ou lógica de negócios) e um servidor de dados.

Spontaneous Networking – A forma de interação automática entre dispositivos móveis e outros dispositivos inseridos numa rede ou ambiente distribuído.

O que é um **Modelo de Análise Fundamental**?

Tem o propósito de realçar aspetos de desenho, dificuldades e ameaças a considerar no desenvolvimento de SD.

Diferentes **níveis** dos Modelos de análise fundamental?

Interação: análise de aspetos relacionados com o desempenho e a dificuldade de estabelecer limites temporais num SD. Falhas: especificação exata das falhas que poderão surgir em processos, dispositivos ou canais de comunicação/rede. Segurança: análise das possíveis ameaças a processos e canais de comunicação/rede.

Diferenças entre comunicação **Síncrona** e **Assíncrona**?

Em sistemas **Síncronos** existem limites para o tempo de execução de cada passo de um processo, há um tempo definido até a receção de uma mensagem enviada, garantido que seja entregue em um tempo previsível. E a taxa de derivação de relógio em cada máquina é conhecida. O send e o receive são operações bloqueantes. Em sistemas **Assíncronos** não há limites definidos ou garantias para o tempo de execução de um processo. O tempo de transmissão de uma mensagem pode variar, e pode haver delays. O clock drift rate é arbitrário e não é controlado. O send não é bloqueante e o receive pode ser bloqueante ou não bloqueante.

O que é o **Modelo de Falhas**?

Modelo que enumera as formas em que o sistema pode falhar, facilitando a compreensão dos efeitos de cada falha. Falhas de omissão, falhas arbitrárias, falhas temporais

O que é o **Modelo de Segurança**?

Identifica possíveis ameaças num sistema distribuído aberto.

Camada “inferior” do Middleware está relacionada com detalhes de comunicação, representação externa de dados e marshalling.

Preocupações com o envio das mensagens: **fiabilidade** (garantia de entrega das mensagens, ainda que alguns pacotes sejam perdidos) e **ordenação correta** (garantia de que as mensagens são entregues pela ordem com que foram enviadas).

No **UDP** o send é não bloqueante e o receive é bloqueante. Existe um **modelo de falhas**: falhas por omissão e por ordenação.

No **TCP** é bloqueante para receive e bloqueante no send apenas se o controlo de fluxo atuar. Existe um **modelo de falhas**: não é fiável, um processo não distingue a causa da falha, um processo não sabe se uma mensagem recente já foi ou não recebida.

Representação de Dados Externa – Formato padronizado para representar dados em sistemas distribuídos de maneira independente da plataforma. Define uma forma específica de codificar dados de maneira que eles possam ser transmitidos e interpretados corretamente por SD que podem estar em computadores de arquiteturas diferentes. **Marshalling** é o processo de transformar dados em uma representação específica. Vantagens: Permitem que SD heterogêneos se comuniquem de forma eficaz, pode-se mover aplicações para diferentes ambientes, simplifica a codificação e decodificação de dados. Desvantagens: As conversões de dados podem adicionar sobrecarga de processamento e consumo de largura de banda, implementação complexa.

Comunicação Cliente-Servidor – Normalmente baseada no protocolo **Request-Reply** síncrona. UDP ou TCP. Modelo de falhas para o **Request-Reply**: falhas de comunicação se for implementado sobre UDP (falhas de omissão e ordenação), falhas relacionadas com processos (paragem, crash).

Request-Reply sobre TCP: facilita a transmissão de argumentos e resultado de tamanho arbitrário, mais fiável, evita a necessidade de filtragem de duplicados e reenvios.

Comunicação em Grupo – Multicast operation: operação que envia uma mensagem de um processo para cada um dos membros de um grupo de processos. IP Multicast: pacotes IP destinados a endereços sem portos.