

Sistemas Distribuídos

AULA 1

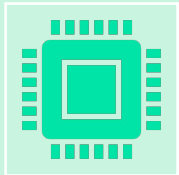
Prof. Mauro César Lopes

maurocl@terra.com.br

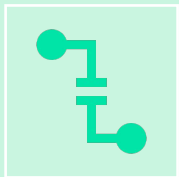
Motivação



Crescente dependência por parte dos usuários



Demanda maior que avanços combinados de *hardware* e *software* centralizados



Características inexistentes em sistemas centralizados como tolerância a falhas (*fault tolerance*)

Breve Histórico

Apareceu na década de 60 dentro do contexto de Sistemas Operacionais.

A motivação foi a criação de unidades de *hardware* denominadas canais ou dispositivos de controle.

Estes dispositivos funcionam independente de um processador de controle e podem fazer operações de E/S concorrentemente com a execução de um programa.

Breve Histórico (2)



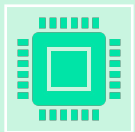
Um canal comunica-se com o processador central através de uma interrupção.



Com a introdução dos canais, partes de um programa poderiam funcionar de forma imprevisível.



Logo após o aparecimento dos canais, foram desenvolvidos as máquinas multiprocessadas.

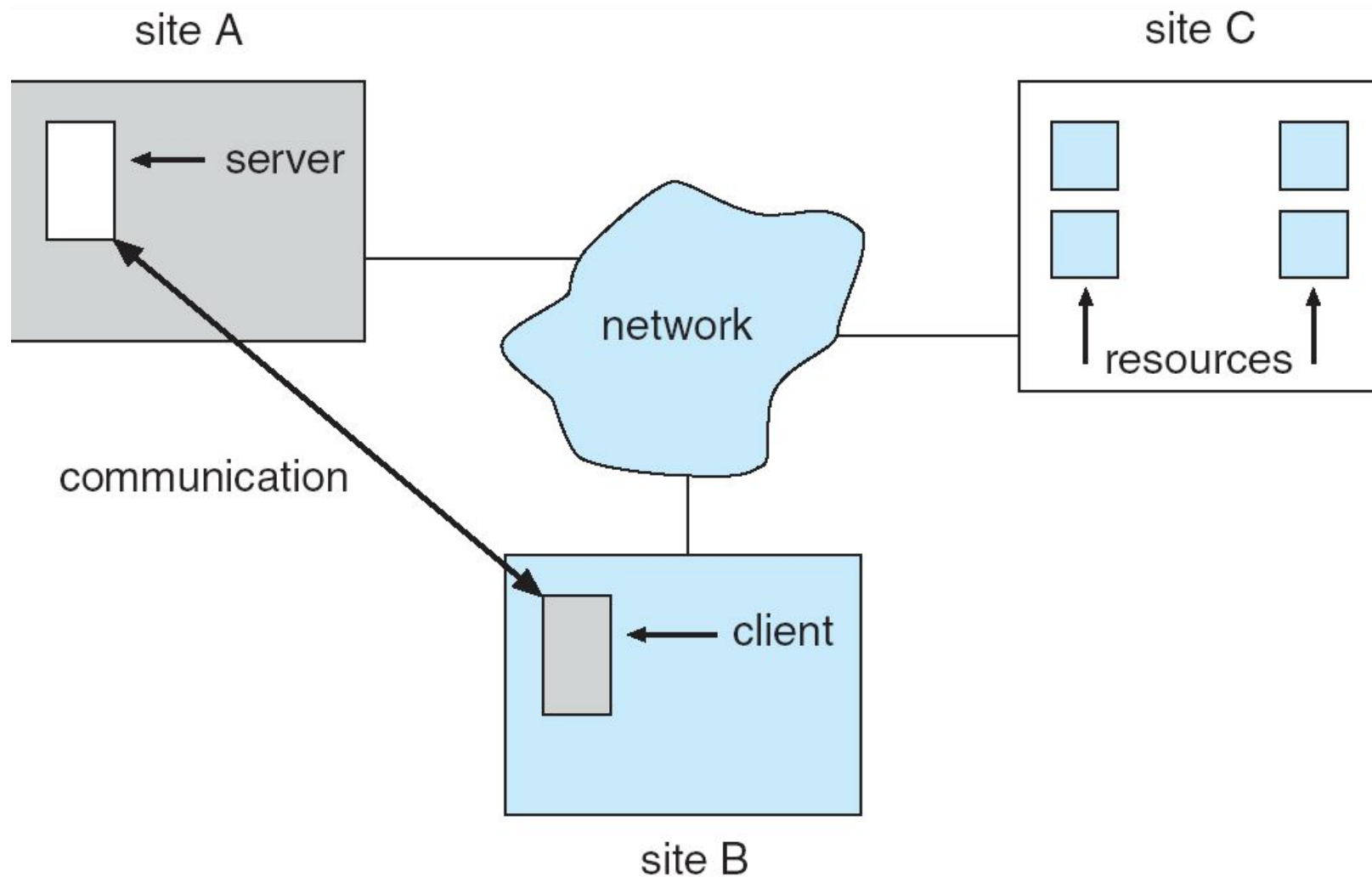


Estas máquinas permitem que aplicações diferentes sejam executadas em processadores diferentes ao mesmo tempo.

Breve Histórico (3)

- Permite também que uma aplicação possa ser executada mais rapidamente se puder ser reescrita de forma a utilizar múltiplos processadores.

Um sistema distribuído



Definição

- **Sloman, 1987**

“Um sistema de processamento distribuído é tal que, vários processadores e dispositivos de armazenamento de dados, comportando processos e/ou bases de dados, interagem cooperativamente para alcançar um objetivo comum. Os processos coordenam suas atividades e trocam informações por passagem de mensagens através de uma rede de comunicação”

Definição

- Andrew **Tanenbaum**
“Coleção de computadores independentes que se apresenta ao usuário como um sistema único e consistente (coerente)”
- **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”

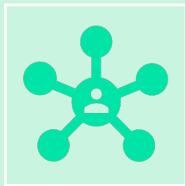
Definição

- **Computação Distribuída** é um método de processamento computacional na qual diferentes partes de um programa rodam simultaneamente em um ou mais computadores através de uma rede de computadores.
- É um tipo de **processamento paralelo**.
- **Sistema de processamento distribuído ou paralelo:** é um sistema que interliga vários nós de processamento (computadores individuais), não necessariamente homogêneos de maneira que um processo de grande consumo seja executado no nó “mais disponível”, ou mesmo subdividido por vários nós

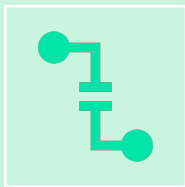
Definição

- **Paralelismo:** divisão de uma tarefa em sub-tarefas coordenadas e que são executadas simultaneamente em processadores distintos

Redes de Computadores x Sistemas Distribuídos



Redes de Computadores: é uma coleção de computadores separados interconectados que trocam mensagens baseadas em um protocolo específico. Computadores são endereçados pelo endereço IP.



Sistema Distribuído: vários computadores em rede trabalhando juntos como um sistema. A separação espacial dos computadores e aspectos de comunicação são escondidos dos usuários

Atrativos

Velocidade de processamento

Compartilhamento de recursos

Confiabilidade

Custo/desempenho

Independência de fornecedor

Características (1)

- Compartilhamento de recursos
 - Compartilhamento de equipamentos e dados
 - recursos de hardware: discos, impressoras, ...
 - recursos de software: arquivos, banco de dados, ...
 - outros recursos: poder de processamento, memória, largura de banda, ...
 - Uso da arquitetura cliente-servidor
 - Servidores que agem como clientes e servidores

Características (2)

- Abertura (*Openness*)
 - Determina o grau e facilidade de extensão dos serviços
 - Requisitos
 - Interfaces padronizadas
 - Comunicação entre processos uniforme
 - Possibilidade de integração de HW e SW heterogêneos

Características (3)

- Paralelismo e Concorrência
 - Ocorrem naturalmente em um Sistema Distribuído pela disponibilidade de múltiplas unidades funcionais
 - Atividades separadas => processamento em unidades diferentes => velocidade
 - Sincronização necessária

Características (4)

- Escalabilidade
 - Resposta a demanda crescente
 - Replicação de dados
 - Distribuição de carga entre servidores

Características (5)

- Tolerância a Falhas (*fault tolerance*)
 - Vital para sistemas críticos
 - Aumento de disponibilidade e confiabilidade
 - Redundância de partes vitais do sistema garantem boa tolerância a falhas parciais
 - Partes replicadas podem servir de redundância para mais de uma parte diferente

Características (6)

- Transparência
 - Acesso
 - Operações de acesso a objetos de informação são idênticas para objetos locais e remotos
 - Localidade
 - Acesso a um objeto ocorre sem que seja necessário o conhecimento de sua localização
- Aplicações inerentemente Distribuídas
 - Correio eletrônico
 - Contas bancárias
- Modularidade e Flexibilidade
- Especialização Funcional

Computação Concorrente

Cenário de Exemplo

Vários carros desejam ir de um ponto *A* a um ponto *B*. Eles podem competir por espaços em uma mesma estrada ou acabam seguindo uns aos outros ou competindo por posições. Ou poderiam andar em vias paralelas, desta forma chegando quase que ao mesmo tempo sem invadir a via do outro. Ou poderiam trafegar em rotas diferentes usando estradas separadas.

Computação Concorrente

Existem múltiplas tarefas a serem feitas. (carros em movimento)

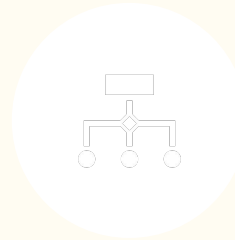
Cada tarefa pode ser executada:

- uma de cada vez em um único processador (uma única estrada);
- em paralelo em múltiplos processadores (pistas em uma estrada); ou,
- em processadores distribuídos (estradas separadas).

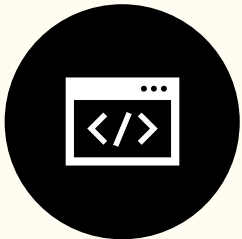
Características



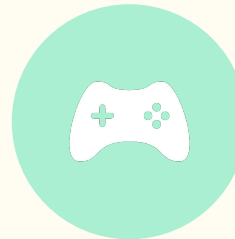
Um programa concorrente contém dois ou mais processos que trabalham juntos para executar uma tarefa.



Cada processo é um programa seqüencial.



Programa seqüencial: único *thread* de controle.



Programa concorrente: múltiplos *threads* de controle.

Comunicação



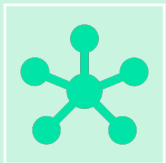
Os processos em um programa concorrente trabalham juntos comunicando-se entre si.



A comunicação pode ser feita através de:

variáveis compartilhadas
(*shared memory*)

troca de mensagens
(*messages*)



Independente da forma de comunicação, os processos precisam sincronizar-se.

Algoritmos Distribuídos

Algoritmos que foram desenvolvidos para serem executadas em muitos processadores “distribuídos” em uma grande área geográfica.

Atualmente, o termo cobre algoritmos que são executadas em redes locais e em multiprocessadores de memória compartilhada (*shared memory*).

Aplicações

processamento
de informações
distribuídas;


computação
científica;

controle de
processos de
tempo real

Tipos de Algoritmos Distribuídos


Método de comunicação entre processos: memória compartilhada, mensagens ponto-a-ponto, difusão de mensagens (*broadcast*) e chamadas remotas a procedimentos (RPC).

Modelo de Execução (*Timing model*): completamente síncronos, completamente assíncronos, parcialmente síncronos.



Tipos de Algoritmos Distribuídos (2)

- Modelo de falha: *hardware* completamente confiável ou pode-se admitir alguma falha. Na presença de falha: o processo pode parar com ou sem aviso; pode falhar brevemente; ou pode apresentar falhas graves e o sistema funcionar de forma arbitrária. Também podem ocorrer falhas na comunicação: perda ou duplicação de mensagens.



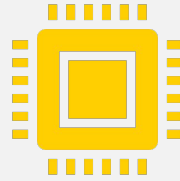
Tipos de Algoritmos Distribuídos (3)

- Problemas abordados: alocação de recursos, comunicação, consenso entre processadores distribuídos, controle de concorrência em bancos de dados, detecção de *deadlock*, sincronização, e implementação de vários tipos de objetos.

Características

- Apresentam um alto grau de incerteza e mais independência de atividades, entre elas:
 - número de processadores desconhecido;
 - topologia de rede desconhecida;
 - entradas independentes em diferentes locais;
 - vários programas sendo executados de uma só vez, começando em tempos diferentes, e operando a velocidades diferentes.
 - não-determinismo dos processadores;
 - tempos de envio de mensagens incertos;
 - desconhecimento da ordenação das mensagens;
 - falhas de processadores e comunicação

Arquitetura de Sistemas de Computação



Sistemas de
Processador Único
(Single-Processor
Systems)



Sistemas
Multiprocessadores
(Multiprocessor
Systems)



Sistemas de Clusters
(Clustered Systems)

Sistemas Monoprocessados x Multiprocessados

- Sistemas de Processador Único (Single-Processor Systems)
- Sistemas com Multiprocessadores (Multiprocessor Systems) □ também conhecidos por Sistemas Paralelos (Parallel Systems) ou Sistemas Fortemente Acoplados (Tightly Coupled Systems)
 - Esses sistemas possuem dois ou mais processadores em comunicação (close communication) compartilhando o barramento do computador e as vezes o relógio (*clock*), memória e dispositivos periféricos

Vantagens de um sistema com multiprocessadores

- Aumento de *throughput* □ "mais trabalho em menos tempo"
 - O uso de n processadores não implica no aumento de n vezes na velocidade de processamento. Há o overhead causado pelo sincronismo e pela disputa do uso de periféricos compartilhados
- Economia de escala
 - Múltiplos processadores podem compartilhar periféricos, armazenamento de dados, e fontes de alimentação. Se vários programas operam nos mesmos dados, é mais barato armazenar esses dados em apenas um disco e ter todos os processadores compartilhando do que ter vários discos locais e várias cópias dos dados
- Aumento da confiabilidade (*reliability*)
 - Se as funções podem ser distribuídas entre os vários processadores, então a falha de um processador não irá parar o sistema, apenas irá deixá-lo mais lento

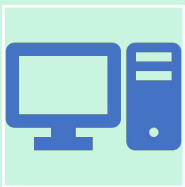
Sistemas de Múltiplos-Processadores

- Dois tipos:
 - Multiprocessamento Assimétrico (Asymmetric multiprocessing)
 - Multiprocessamento Simétrico (Symmetric Multiprocessing (SMP))

Sistemas de Processador Único (Single-Processor Systems)



Somente um processador de uso geral



Processadores de propósitos especiais □ processadores de dispositivos específicos como disco, teclado, controladores gráficos

Executam conjunto de instruções limitadas (*run limited instructions-set*)
Não executam processos do usuário

Multiprocessadores Assimétrico

- Para cada processador é atribuída uma tarefa específica
 - Processador *master* (mestre) □ controla o sistema (escalona e aloca trabalho aos escravos)
 - Processador *slave* (escravo) □ procura por instruções no master ou possuem tarefas específicas
 - Define uma relação *master-slave* (mestre-escravo)



Multiprocessadores Simétrico (SMP)

É mais comum que o assimétrico

Cada processador executa todas tarefas dentro do sistema operacional

Significa que todos os processos são *peers* (pares) (não há relação *master-slave*)

Diferenças entre multiprocessadores assimétrico e simétrico

- A diferença entre multiprocessadores assimétrico e simétrico pode resultar tanto de hardware quanto de software (exemplo: um mesmo hardware pode implementar multiprocessamento simétrico ou assimétrico dependendo do software usado)
- Hardwares especiais podem diferenciar múltiplos processadores ou o software pode ser escritos para permitir um único mestre e vários escravos

Múltiplos Núcleos de Computação (Multiple Compute Cores)

- Múltiplos núcleos em um único chip
 - Chips de dois núcleos (*Two-way chips*)
 - Chips de N núcleos (*N-way chips*)
- Aparecem ao sistema operacional como se fossem N processadores padrão

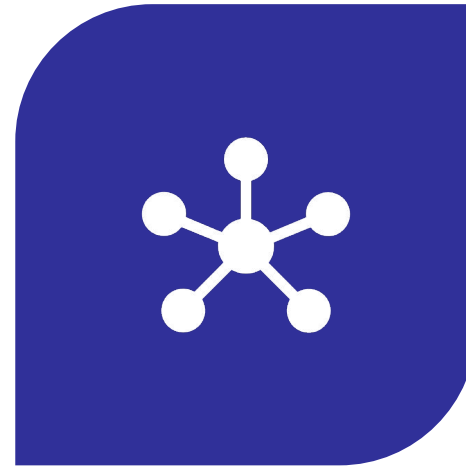
Sistemas Agrupados (*Clustered Systems*)

- Sistemas de múltiplas CPUs
- É composto de dois ou mais sistemas individuais acoplados juntos (*coupled together*)
- Definição geralmente aceita:
 - Computadores em *cluster* compartilham armazenamento, são ligados por uma LAN ou por interconexões mais rápidas como InfiniBand.
 - *Clustering* é geralmente usado para prover serviços de alta-disponibilidade (*high-availability* ou *HA*), isto é, serviços que irão continuar funcionando mesmo que um ou mais nós do *cluster* falhem

Sistemas Agrupados (*Clustered Systems*) (2)



ALTA DISPONIBILIDADE É ALCANÇADA
GERALMENTE ADICIONANDO UM
NÍVEL AO SISTEMA



CLUSTER NODES (NÓS DO CLUSTER)

Clustering

- Clusters podem ser estruturados assimetricamente ou simetricamente
 - *Assimetric clustering* □ uma máquina encontra-se em modo *hot-standby* enquanto a outra está executando as aplicações
 - *Symmetric mode* □ duas ou mais máquinas (hosts) estão executando aplicações e estão monitorando um ao outro (uso de mecanismos de *heartbeat*)

Outras forma de Cluster

- Parallel cluster (cluster paralelo)
- Clustering over a WAN (wide-area network) ou rede de longa distância

Cluster Paralelo (Parallel Cluster)

Permite que múltiplos hosts acessem os mesmos dados no *storage* (dispositivo de armazenamento de dados) compartilhado



Geralmente obtido pelo uso de versões especiais de software e liberações especiais de aplicações



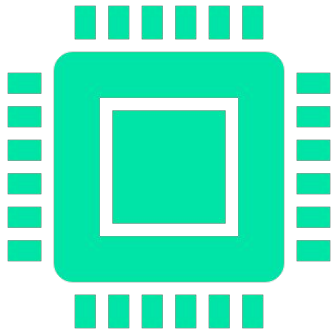
Exemplo:

Oracle Parallel Server □ é uma versão do Oracle database liberada para uso em clusters paralelos

Processador Dual-core

- Um processador dual-core é um único chip que contém dois processadores distintos ou “dois núcleos de execução” no mesmo circuito integrado

Processadores



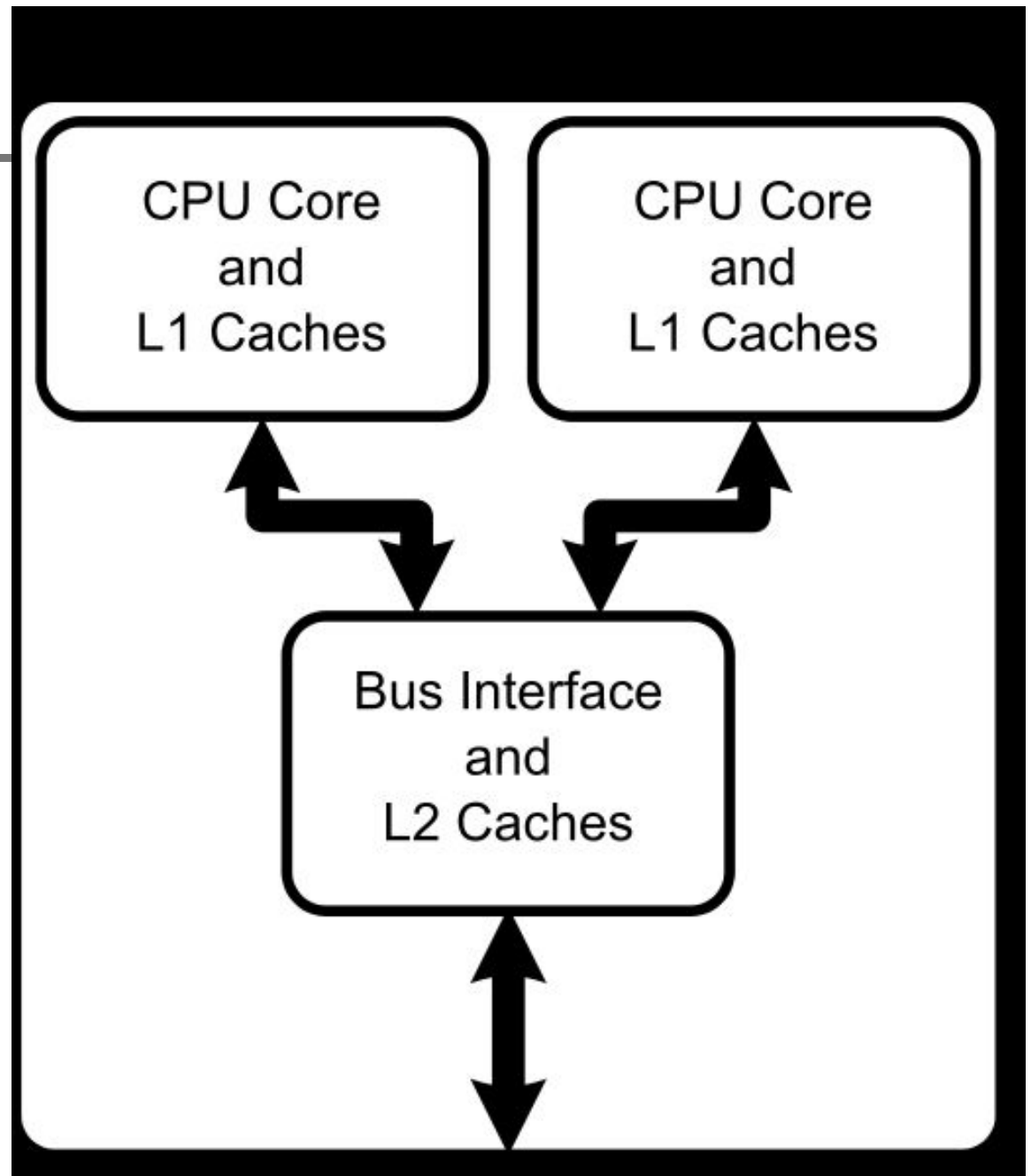
Processador **dual-core** contém
dois núcleos



Processador **quad-core** contém
quatro núcleos

Exemplo: Intel Core 2 dual core processor

- Diagrama de um Intel Core 2 dual core processor, with CPU-local Level 1 caches, and a shared, on-die Level 2 cache.



Sistemas de Computação

Composto por “grande”
quantidade de
computadores
conectados por uma
rede de alta velocidade

Redes de
computadores ou
Sistemas Distribuídos

Metas

Acesso a recursos

Transparência da distribuição

Abertura

Escalabilidade



Tipos de Sistemas Distribuídos

Sistemas de Computação Distribuídos

- Sistemas de computação em cluster
- Sistemas de computação em grade (GRID)

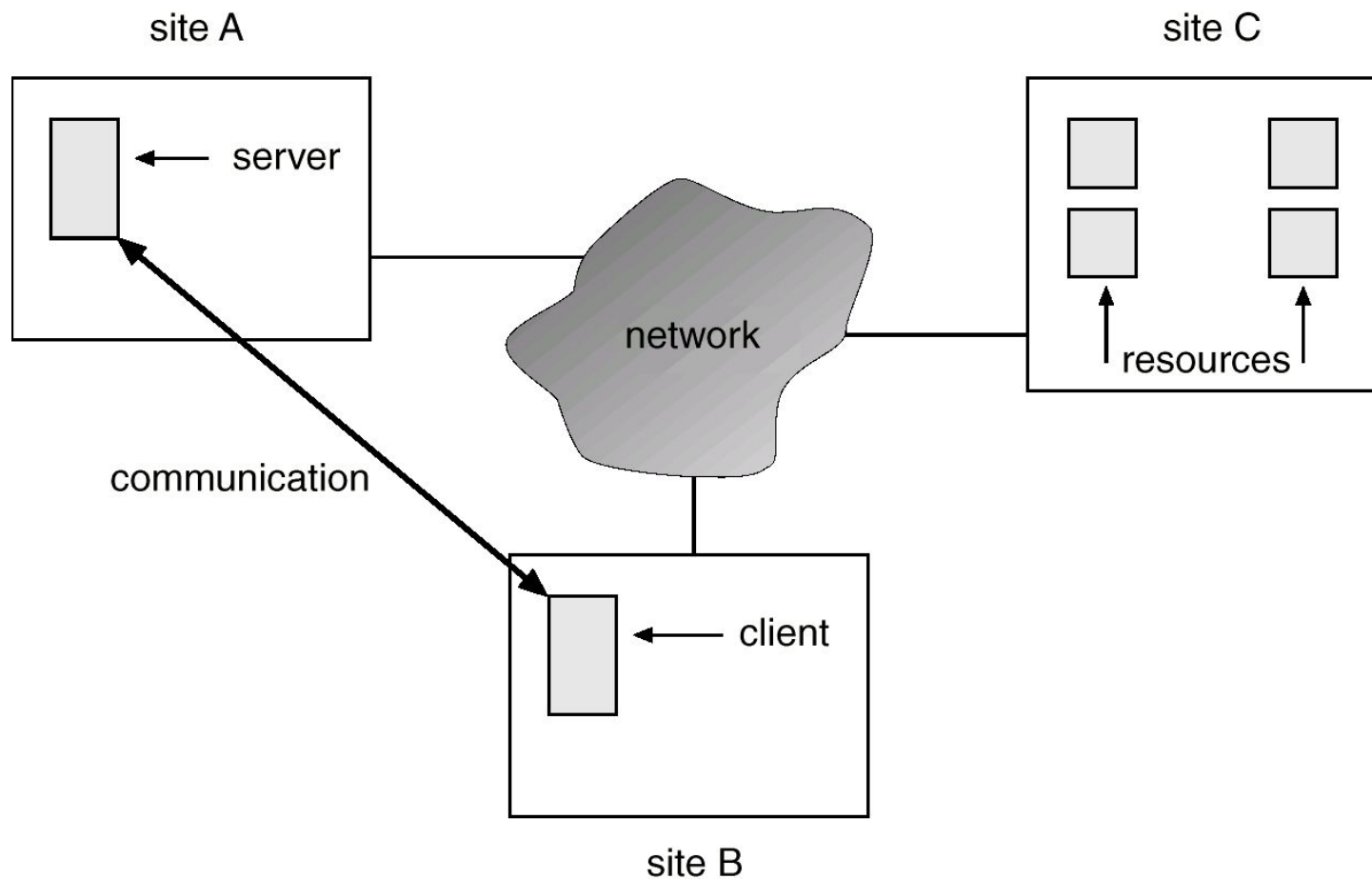
Sistemas de Informação Distribuídos

- Sistema de processamento de informações
- Integração de aplicações Empresariais (EIA)

Sistemas Distribuídos Pervasivos

- Sistemas domésticos
- Sistemas Eletrônicos para tratamento de saúde
- Redes de sensores

Um sistema distribuído



Graceful Degradation



- É a habilidade de continuar provendo serviços proporcional ao nível de hardwares que permaneceram funcionando.

Tolerância a falha (Fault tolerant)

São sistemas que podem sofrer uma falha em qualquer componente único e permanecer em operação (funcionamento)

A tolerância a falha requer mecanismos que permitam identificar a ocorrência de falhas, diagnosticá-la e se possível, corrigi-la

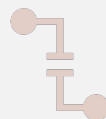
Tipos de Transparências (1)



Transparência de Acesso (*Access transparency*): enables local and remote resources to be accessed using identical operations.



Transparência de Localização (*Location transparency*): enables resources to be accessed without knowledge of their physical or network location (for example, which building or IP address).



Transparência de Concorrência (*Concurrency transparency*): enables several processes to operate concurrently using shared resources without interference between them.



Transparência de Replicação (*Replication transparency*): enables multiple instances of resources to be used to increase reliability and performance without knowledge of the replicas by users or application programmers.

Tipos de Transparências (2)

Transparência de Falhas (*Failure transparency*): enables the concealment of faults, allowing users and application programs to complete their tasks despite the failure of hardware or software components.

Transparência de Mobilidade (*Mobility transparency*): allows the movement of resources and clients within a system without affecting the operation of users or programs.

Transparência de Performance (*Performance transparency*): allows the system to be reconfigured to improve performance as loads vary.

Transparência de Escala (*Scaling transparency*): permite ao sistema e aplicação expandir em escala sem modificar a estrutura do sistema ou o algoritmo de aplicação

Sistemas Distribuídos

- Um sistema distribuído é uma coleção de sistemas de computadores, fisicamente separados, possivelmente heterogêneos, ligados em rede para prover aos usuários acesso a vários recursos que ele mantém.
- O acesso a recursos compartilhados aumenta a velocidade de computação, funcionalidade, disponibilidade de dados e confiabilidade

Sistemas Distribuídos

Em termos simples, uma rede é um caminho de comunicação (*communication path*) entre dois ou mais sistemas

Sistemas distribuídos dependem da rede para sua funcionalidade

A redes variam segundo os protocolos utilizados, a distância entre os nós e o meio de transporte

Redes são caracterizadas baseadas na distância entre seus nós: LAN, WAN, MAN

Network Operating System (NOS)

É um sistema operacional que provê características (*features*) como compartilhamento de arquivos através da rede e inclui um sistema de comunicação que permite diferentes processos em diferentes computadores a trocarem mensagens

Um computador rodando um sistema operacional de rede (NOS) age autonomamente de todos os outros computadores da rede embora ele esteja consciente da rede e seja capaz de comunicar com outros computadores da rede



Distributed Operating System

- Um sistema operacional distribuído provê um ambiente menos autônomo: os diferentes sistemas operacionais comunicam-se mais próximo suficientemente a fim de prover a ilusão de que apenas um sistema operacional controla a rede