

## Sistemas Distribuídos

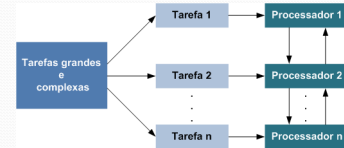
### 2 Computadores Distribuídos e Paralelos

- **Paralelismo**
- **Granularidade**
- **Multiprocessadores e Multicomputadores**
- **Ambientes fortemente acoplados e fracamente acoplados**
- **Sistemas Operacionais de Rede**
- **Sistemas Operacionais Distribuídos**

Profª Ana Cristina B. Kochem Vendramin  
DAINF/UTFPR

## Paralelismo

- Divisão de tarefas grandes e complexas em sub-tarefas que são distribuídas entre processadores distintos para serem executadas simultaneamente.
- Objetivo: processamento mais rápido.



Profª Ana Cristina B. Kochem Vendramin,  
DAINF/UTFPR

2

## Níveis de Paralelismo – Nível de Instrução

- **Nível de instrução**
  - Evolução dos processadores sequenciais para os **pipeline**.
  - **Pipeline**: técnica que divide o processamento de instruções em várias partes, cada qual tratada por uma unidade dedicada.
  - Objetivos:
    - Sobrepor no tempo diversas fases de execução das instruções;
    - Pipeline melhora o *throughput* de todo o trabalho, mas não reduz o tempo gasto para completar cada instrução individualmente.

Profª Ana Cristina B. Kochem Vendramin,  
DAINF/UTFPR

3

## Níveis de Paralelismo – Nível de Instrução

- Exemplo de **pipeline de cinco estágios**:
  - E1: busca a instrução na memória e armazena em um *buffer*.
  - E2: decodifica a instrução (determina seu tipo e os operandos necessários para a sua execução);
  - E3: localiza e busca os operandos que podem estar em registradores ou na memória;
  - E4: executa a instrução;
  - E5: escreve resultado do processamento em um registrador.
- Em um único ciclo de *clock*, é possível que o estágio E5 escreva o resultado de uma instrução *x* no registrador enquanto os outros estágios trabalham nas instruções subsequentes.

Profª Ana Cristina B. Kochem Vendramin,  
DAINF/UTFPR

4

## Níveis de Paralelismo – Nível de Instrução

- **Processadores Superescalares**
  - Primeiro processador superescalar: dois pipelines.
  - Dois ou mais pipelines de instruções de *n* estágios.
  - Executa duas ou mais instruções independentes em paralelo no mesmo ciclo de relógio.
  - Necessário verificar a existência de dependências entre as instruções.

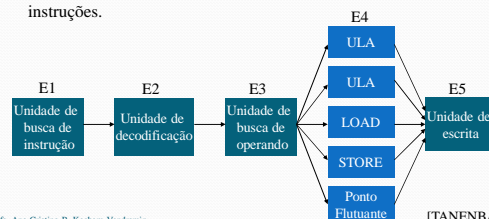
Profª Ana Cristina B. Kochem Vendramin,  
DAINF/UTFPR

5

## Níveis de Paralelismo – Nível de Instrução

- Ex.: processador Superescalar com cinco unidades funcionais

**Justificativa dos pipelines de execução:** estágios como o de busca de operandos são bem mais rápidos que os estágios de execução de instruções.



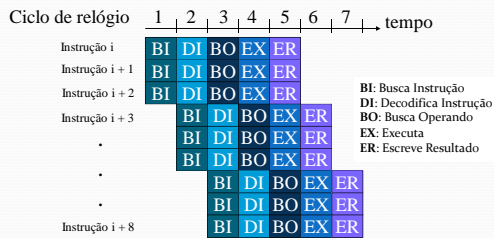
Profª Ana Cristina B. Kochem Vendramin,  
DAINF/UTFPR

[TANENBAUM 99]

6

## Níveis de Paralelismo – Nível de Instrução

### Processadores Superscalares



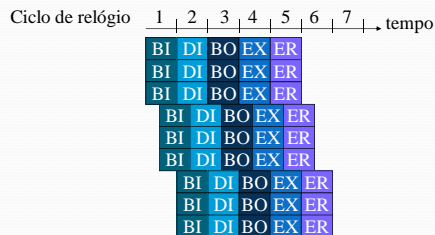
## Níveis de Paralelismo – Nível de Instrução

### Processadores *Superpipeline*

- Objetivo: reduzir tempo de ciclo de relógio
- A arquitetura *superpipeline* subdivide cada estágio do *pipeline* em subestágios e multiplica o *clock* internamente.
- Cada subestágio continua executando uma instrução por *clock*, mas como o *clock* interno é multiplicado, o *pipeline* pode aceitar duas ou mais instruções para cada *clock* externo.
- O grau de *superpipeline* é medido pelo número de subestágios em que é dividido um estágio.

## Níveis de Paralelismo – Nível de Instrução

### Processadores *Superpipeline* superscalares



## Níveis de Paralelismo – Nível de Instrução

### Processadores VLIW (*Very Large Instruction Word*)

- Reduzir número de instruções para executar uma aplicação
- Uma simples instrução pode especificar múltiplas operações.
  - Bits da instrução codificam as operações.
- Operações devem ser independentes para que possam ser enviadas para as unidades de execução ao mesmo tempo.
- Não precisa hardware especializado para escalonar ou verificar dependências entre instruções (função do compilador).

## Níveis de Paralelismo

- Ambiente Multithreading**
  - Várias linhas de execução lógicas em um mesmo processador.
- Nível do chip - Processadores *multicore***
  - Execução simultânea de mais de um fluxo de instruções.
- Nós multiprocessados - supercomputadores**
  - Vários chips que compartilham memória.
- Cluster**
- Grids**

## Níveis de Paralelismo

- Cluster**
  - Conjunto de máquinas interligadas através de uma rede que trabalham juntas, como se fossem um único computador de alto desempenho, com o intuito de concluir um processamento pesado.
- <http://www.top500.org>

## Níveis de Paralelismo

- **Cluster**

- Vantagens:
  - Relação custo x benefício se comparado aos supercomputadores.
  - Tolerância a falhas;
  - Independência de fornecedores:
    - Hardware aberto e Software livre.
  - Alto poder de processamento na resolução de problemas em aplicações paralelizáveis;
  - Alta escalabilidade
    - Mais carga de trabalho – novos componentes.
  - Heterogeneidade.

13

Profa. Ana Cristina B. Kochem Vendramin.  
DAINF/UTFPR

## Níveis de Paralelismo

- **Grade computacional (grid)**

- Nível continental ou nacional.
- Comunicação de clusters via Internet.
  - Coleção de recursos distribuídos geograficamente, tipicamente heterogêneos e compartilhados.
- Executar operações paralelas que demandam:
  - Capacidade de processamento e armazenamento não disponível em um conjunto de máquinas locais.

14

Profa. Ana Cristina B. Kochem Vendramin.  
DAINF/UTFPR

## Bibliotecas para Coordenar Processos em Cluster

- Para explorar ao máximo o potencial de paralelismo de um cluster:
  - Compiladores com diretivas "**OpenMP**": coordenar processos com memória compartilhada;
  - Biblioteca **MPI** (*Message Passing Interface*) ou **PVM** (*Parallel Virtual Machine*): coordenar processos que são distribuídos pelos nós do cluster, passando a se comportar como uma única máquina de alto desempenho.

15

Profa. Ana Cristina B. Kochem Vendramin.  
DAINF/UTFPR

## Granularidade

- Um sistema distribuído pode ser classificado em função da **granularidade** das interações entre seus componentes.
  - Interações: troca de dados e/ou sincronização entre os processadores que cooperam na execução de um programa.
  - Em alguns sistemas as interações são infrequentes mas envolvem grandes volumes de dados quando ocorrem. Em outros, as interações são muito frequentes e há pouca troca de dados e/ou sincronização.

16

Profa. Ana Cristina B. Kochem Vendramin.  
DAINF/UTFPR

## Granularidade de um programa

- $g = T_{proc}/T_{msg}$ , onde:
  - $T_{proc}$  → tempo de processamento;
  - $T_{msg}$  → overhead de comunicação entre os processos.
- Se o valor de **g é baixo**
  - Aumenta o potencial de paralelismo, mas aumenta peso de overhead de comunicação.
  - Paralelismo de granularidade fina (*fine-grained*)
- Se valor de **g é alto**
  - Overhead baixo;
  - Menor exploração do paralelismo;
  - Paralelismo de granularidade grossa (*coarsed-grained*).

17

Profa. Ana Cristina B. Kochem Vendramin.  
DAINF/UTFPR

## Tipos de Granularidade

- **Granularidade Grossa** (poucos grãos grossos).
  - Exemplo:
    - **Redes de computadores**
      - Cada processador é independente e geograficamente disperso dos demais. Dedicar apenas parte de seu tempo e recursos a tarefas comuns.
      - As interações entre os computadores, normalmente, são pouco frequentes mas podem envolver grandes volumes de dados.

18

Profa. Ana Cristina B. Kochem Vendramin.  
DAINF/UTFPR

## Tipos de Granularidade

- **Granularidade Fina** (muitos grãos finos)
  - Exemplo:
    - **Processadores vetoriais ou matriciais**
      - Máquinas projetadas para a solução de uma classe restrita de problemas. São construídos de forma que uma mesma operação possa ser executada simultaneamente sobre vários elementos de um vetor ou matriz.

19

Profa. Ana Cristina B. Kochem Vendramin,  
DAINF/UTFPR

## Computadores Distribuídos e Paralelos

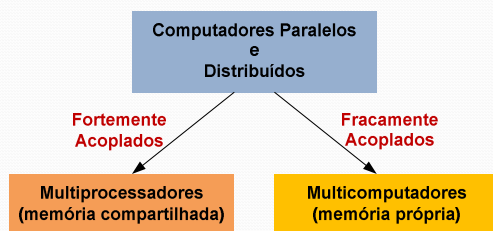
- Tanenbaum propõe a divisão de máquinas MIMD (Múltiplas Instruções e Múltiplos Dados) em **multiprocessadores** e **multicomputadores**.
- Outro ponto da taxonomia de Tanenbaum é a caracterização dos sistemas pelo grau de ligação entre os computadores, que podem ser **fortemente acoplados** ou **fracamente acoplados**.

20

Profa. Ana Cristina B. Kochem Vendramin,  
DAINF/UTFPR

## Computadores Distribuídos e Paralelos

MIMD (Multiple Instruction Multiple Data)



21

Profa. Ana Cristina B. Kochem Vendramin,  
DAINF/UTFPR

## Computadores Distribuídos e Paralelos

- **Ambientes Fortemente Acoplados**
  - Geralmente **multiprocessadores** são muito ligados;
  - Memória compartilhada;
  - Baixo retardo no envio de mensagens;
  - Alta taxa de transmissão;
  - Geralmente, sistemas muito ligados são usados como sistemas paralelos, trabalhando em um único problema.

22

Profa. Ana Cristina B. Kochem Vendramin,  
DAINF/UTFPR

## Computadores Distribuídos e Paralelos

- **Ambientes Fracamente Acoplados**
  - Normalmente **multicomputadores** são pouco ligados;
  - Cada computador possui sua memória;
  - Normalmente, sistemas pouco ligados são utilizados como um sistema distribuído, trabalhando em diversos problemas.

23

Profa. Ana Cristina B. Kochem Vendramin,  
DAINF/UTFPR

## Multiprocessadores

- Sistemas operacionais regulares:
  - Sistema de arquivos;
  - Chamadas no sistema;
  - Gerenciamento de Memória e Dispositivos de E/S;
- Duas ou mais CPUs;
- Memória compartilhada;
- Sincronização de Processos, escalonamento e gerenciamento de recursos.

24

Profa. Ana Cristina B. Kochem Vendramin,  
DAINF/UTFPR

## Sistemas Operacionais de Rede

- Exemplos: Windows, Linux.
- Possuem recursos de interligação em rede incorporado e, portanto, podem ser usados para acessar recursos remotos.
- Software fracamente acoplado
- Nós têm autonomia no gerenciamento de seus recursos.
- Comunicação é explicitamente solicitada pelo usuário e se dá através de troca de mensagens.

## Sistemas Operacionais de Rede

- Usuários tem visões diferentes do sistema: organização do sistema de arquivo local, dispositivos locais que cada máquina possui, etc.
- Máquinas podem executar diferentes SOs desde que os protocolos de comunicação e serviços sejam usados por todas as máquinas.

## Sistemas Operacionais Distribuídos

- Um sistema operacional que produz uma única imagem dos sistema para todos os recursos de um sistema distribuído é chamado de **Sistema Operacional Distribuído (SOD)**
- Software fortemente acoplado em um hardware fracamente acoplado.
- Objetivo: criar uma ilusão para os usuários que o sistema de multicomputadores funciona como uma grande máquina.
- O sistema operacional tem controle sobre todos os nós do sistema e dispara novos processos de modo transparente de acordo com sua política de escalonamento.

## Sistemas Operacionais Distribuídos

- Não adoção de SOD para uso geral.
  - Usuários já possuem softwares aplicativos que atendem suas necessidades;
  - Usuários querem ter autonomia em suas máquinas e não ter sua máquina compartilhada podendo ter seu desempenho prejudicado pelo uso de outros usuários.
- Equilíbrio entre requisitos de autonomia e acesso compartilhado aos recursos
  - *Middleware SD + Sistema Operacional de Rede*

## Referências Bibliográficas

- Coulouris, George; Dollimore, Jean; Kindberg, Tim. Distributed Systems Concepts and Design. Third Edition. Addison-Wesley 2001.
- Coulouris, George; Dollimore, Jean; Kindberg, Tim; tradução João Tortello. Sistemas Distribuídos: conceitos e projeto. 4. ed. Bookman 2007.
- Tanenbaum, A.S. Distributed Operating Systems. Prentice-Hall International, 1995.
- Tanenbaum, A.S. tradução Nery Machado Filho. Organização Estruturada de Computadores. 4. ed. LTC, 1999.