

USP - ICMC - SSC
SSC 0510 - Informática - 2o. Semestre 2009

Disciplina de Arquitetura de Computadores

Prof. Fernando Santos Osório

Email: fosorio [at] { icmc. usp. br , gmail. com }

Página Pessoal: <http://www.icmc.usp.br/~fosorio/>

Estagiário PAE Maurício Dias - Email: [acdias29 \[at\] yahoo.com.br](mailto:acdias29@yahoo.com.br)

Material on-line: COTEIA - <http://coteia.icmc.usp.br>

Aula 09 – Arquiteturas Especiais

Conteúdos Abordados:

- 1. Arquiteturas Paralelas:**
Processamento Paralelo do SISD ao MIMD
 - 1.1. SISD**
 - 1.2. MISD**
 - 1.3 SIMD**
 - 1.4 MIMD**

 - 2. Arquiteturas Avançadas:**
Processadores Super-Escalares
Processadores Vetoriais
- >> Discussão: Trabalho 2**

Níveis de paralelismo

- **Instrução** (granulosidade fina)
 - Paralelismo entre as instruções
 - Arquiteturas Pipeline, Superescalar, VLIW
- **Tarefas** (granulosidade média)
 - Paralelismo entre as *threads*
 - Arquiteturas SMT (*Simultaneous MultiThreading*)
- **Processos** (granulosidade grossa)
 - Paralelismo entre os processos
 - Computação Paralela
 - Arquiteturas multiprocessadores e multicomputadores

Computação Paralela – Conceitos

- Permite a execução das tarefas em menor tempo, através da execução em paralelo de diversas tarefas.
- O paralelismo pode ser obtido em diversos níveis, com ou sem o uso de linguagens de programação paralela.
- Arquiteturas de diversos tipos, elaboradas para aplicações específicas, podem ser utilizadas para acelerar a execução dessas aplicações.

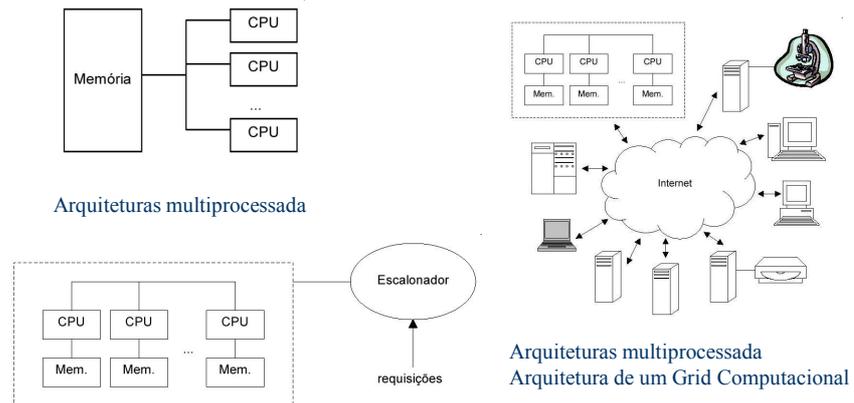
Computação Paralela – Conceitos

- Programação Seqüencial
- Programação Concorrente
 - Um servidor, atendendo vários clientes através de uma política de escalonamento no tempo
- Programação Paralela
 - Vários servidores, atendendo vários clientes simultaneamente no tempo

Computação Paralela – Aplicações

- Genoma Humano
- Biologia Estrutural
- Farmacêutica
- Simulação:
 - Turbulência dos Fluidos
 - Dinâmica de Fluidos Viscosos
 - Circulação de Oceanos (água, poluentes)
 - Dinâmica de veículos e de aeronaves
 - Simulação Física (partículas, eletromagnetismo, multiphysics)
- Meteorologia: Previsão do Tempo (maior antecedência)
- Modelagem de Supercondutores
- Processamento de Imagens e Visão por Computador
- Computação Gráfica, Realidade Virtual, Jogos de Computador

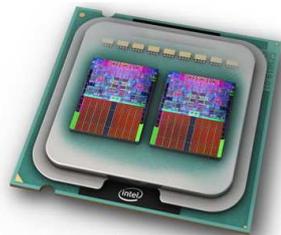
Computação Paralela – Classificação e Tipos



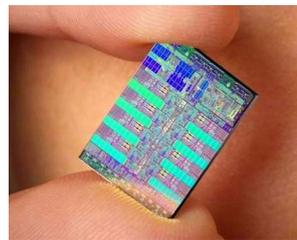
7

Out. 2009

Computação Paralela – Classificação e Tipos



Multicore Processor
Intel



Cell Processor
PlayStation 3

Mas também...
Pipeline, Pré-Fetch, Multi-Level Caches, ...

Parallel Distributed Processing (PDP) ;^)
MultiCore, MultiProcessador, Clusters => Processamento Neural



8

Out. 2009

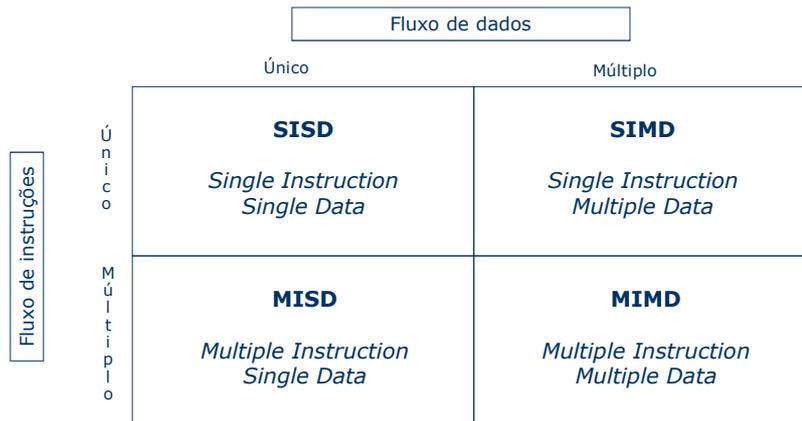
Computação Paralela – Classificação e Tipos

- Existem diversas classificações para as arquiteturas paralelas
- Devido a constante evolução, nenhuma classificação consegue abranger todas as arquiteturas existentes
- Classificação de Flynn (1972):
 - “Some computer organizations and their effectiveness”, *IEEE Transactions on Computers*, vol. C-21, pp. 948-960, 1972.
 - Mais conhecida
 - Baseia-se na unicidade e multiplicidade do fluxo de dados e instruções

Classificação de Duncan (1990)

- “A survey of parallel computer architectures”, *IEEE Computer*, pp. 5-16, Fevereiro, 1990
- classificação mais recente e abrangente
- menos conhecida

Classificação de Flynn (1972)



Classificação de Flynn (1972)

Flynn's taxonomy

| | Single Instruction | Multiple Instruction |
|---------------|--------------------|----------------------|
| Single Data | SISD | MISD |
| Multiple Data | SIMD | MIMD |

SISD:
 Single Instruction stream, Single Data stream

MISD:
 Multiple Instruction stream, Single Data stream

SIMD:
 Single Instruction stream, Multiple Data stream

MIMD:
 Multiple Instruction stream, Multiple Data stream

Arquiteturas Paralelas

SISD: único fluxo de instrução, único fluxo de dados

- classe que representa os computadores convencionais (seriais)
- as instruções são executadas serialmente, porém os estágios (busca da instrução, decodificação, busca do operando e execução) podem ser sobrepostos (*pipeline*)
- Pode-se saber o que está ocorrendo exatamente em cada instante de tempo e reproduzir o processo passo a passo mais tarde

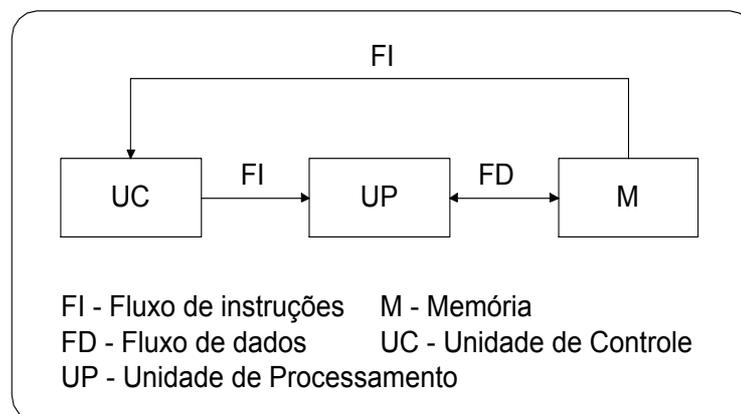
| Flynn's taxonomy | | |
|------------------|--------------------|----------------------|
| | Single Instruction | Multiple Instruction |
| Single Data | SISD | MISD |
| Multiple Data | SIMD | MIMD |

13

Out. 2009

Arquiteturas Paralelas

SISD: único fluxo de instrução, único fluxo de dados



14

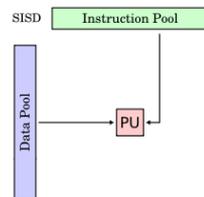
Out. 2009

SISD: único fluxo de instrução, único fluxo de dados

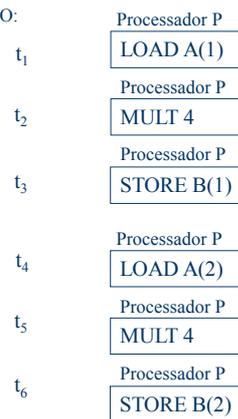
Exemplo (SISD)

For $l = 1$ to N

 LOAD A(l)
 B(l)=A(l)*4 => MULT 4
 STORE B(l)



TEMPO:

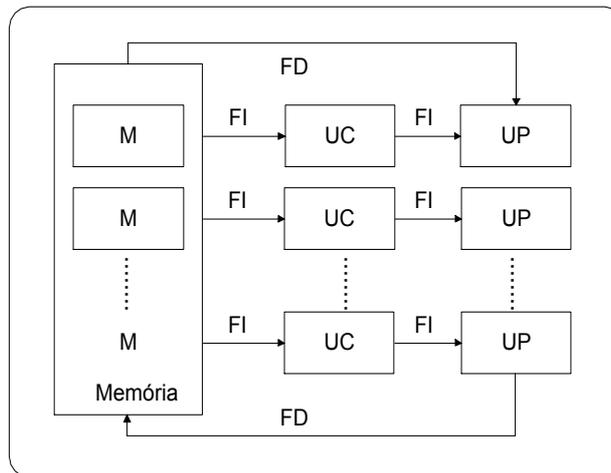


MISD: múltiplo fluxo de instruções, único fluxo de dados

- vários processadores, onde cada um recebe instruções distintas mas operam sobre o mesmo conjunto de dados
- poucos exemplos
 - Múltiplos filtros de frequência operando sobre um único fluxo de sinal
 - Múltiplos algoritmos de criptografia para decodificar uma mensagem

| Flynn's taxonomy | | |
|------------------|--------------------|----------------------|
| | Single Instruction | Multiple Instruction |
| Single Data | SISD | MISD |
| Multiple Data | SIMD | MIMD |

MISD: múltiplo fluxo de instruções, único fluxo de dados



SIMD: único fluxo de instruções, múltiplo fluxo de dados

- Classe que representa os processadores matriciais, paralelos e associativos
- Uma única unidade de controle que envia um fluxo de instruções para vários processadores
- Os processadores recebem a mesma instrução ao mesmo tempo e atuam sobre diferentes fluxos de dados
- IBM, Sony, Toshiba Group: Cell Processor
- GAPP Processor => Image Processing
- Modern Graphics Processing Units are often very wide SIMD implementations



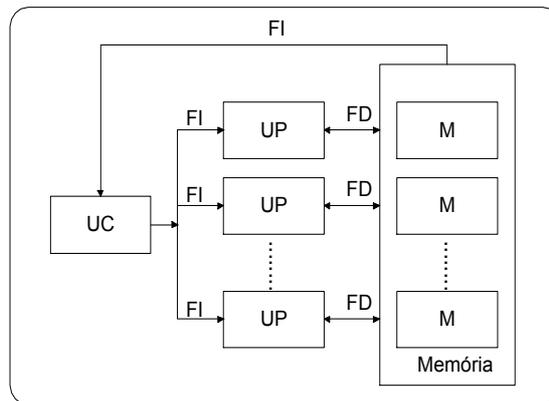
GeForce 6600GT (NV43) GPU

Flynn's taxonomy

| | Single Instruction | Multiple Instruction |
|---------------|--------------------|----------------------|
| Single Data | SISD | MISD |
| Multiple Data | SIMD | MIMD |

Arquiteturas Paralelas

SIMD: único fluxo de instruções, múltiplo fluxo de dados



- 1024 processadores em um array 32x32, ou 4096 em 64x64
- processador de controle armazena instruções e dados são armazenados na memória de cada processador

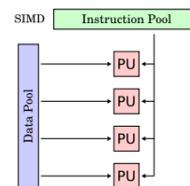
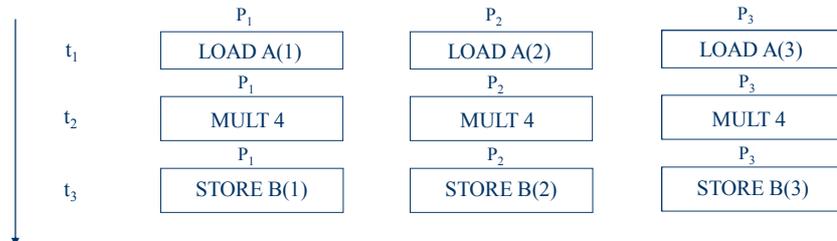
Arquiteturas Paralelas

SIMD: único fluxo de instruções, múltiplo fluxo de dados

Exemplo (SIMD)

$B(l) = A(l) * 4 \Rightarrow$
 LOAD A(l)
 MULT 4
 STORE B(l)

TEMPO:



Arquiteturas Paralelas

MIMD: múltiplo fluxo de instruções, múltiplo fluxo de dados

- vários processadores, cada um controlado por uma unidade de controle
- processadores recebem instruções diferentes e operam sob fluxo de dados diferentes
- podem ser síncronos ou assíncronos

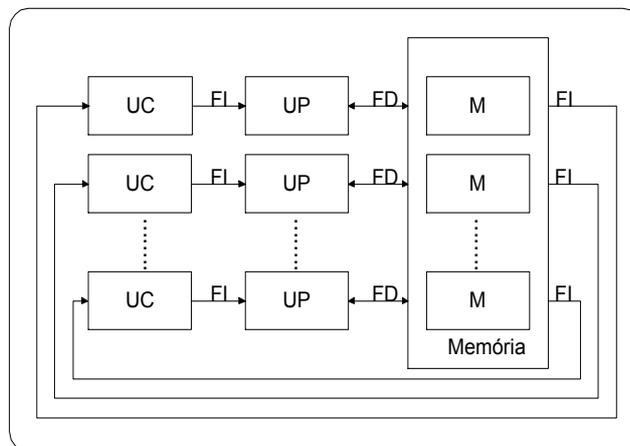
| Flynn's taxonomy | | |
|------------------|--------------------|----------------------|
| | Single Instruction | Multiple Instruction |
| Single Data | SISD | MISD |
| Multiple Data | SIMD | MIMD |

21

Out. 2009

Arquiteturas Paralelas

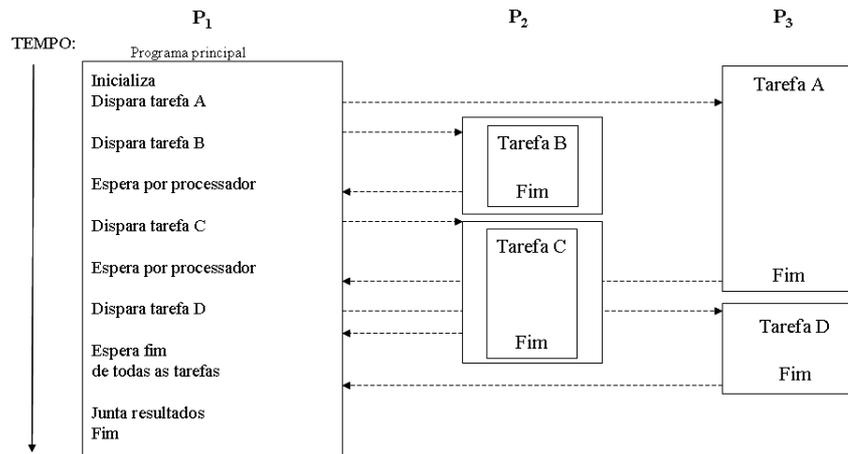
MIMD: múltiplo fluxo de instruções, múltiplo fluxo de dados



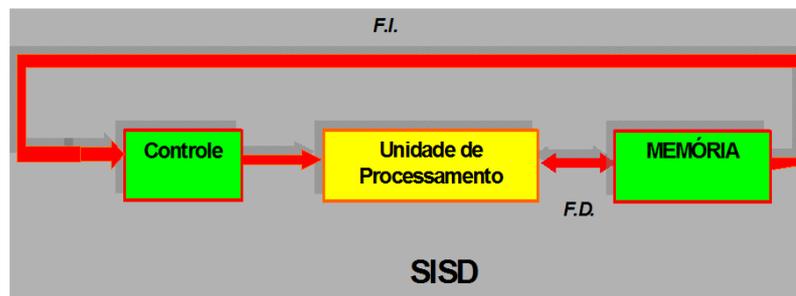
22

Out. 2009

MIMD: Exemplo (MIMD)



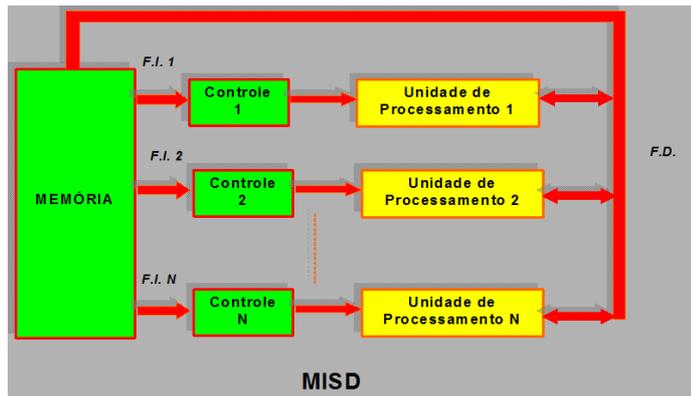
SISD: o computador consiste de uma unidade de processamento que recebe um fluxo simples de instruções e opera sobre um simples fluxo de dados;



Ex.: computadores von Neumann (o usual)

Arquiteturas Paralelas

MISD: N processadores, cada um com sua unidade de controle e unidade de processamento, dividem uma mesma memória e executam diferentes instruções sobre o mesmo dado

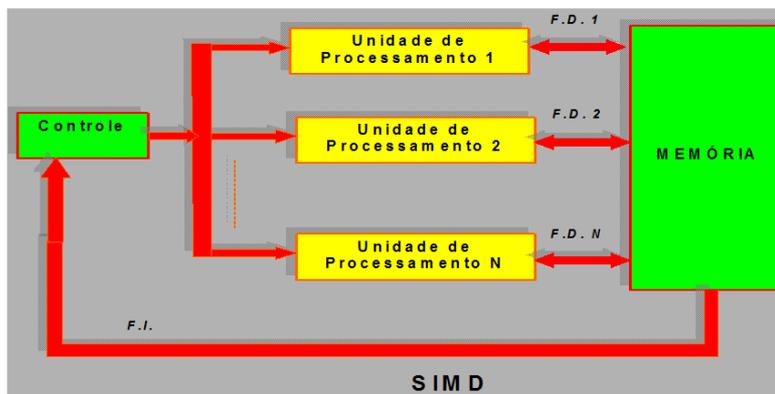


25
Out. 2009

Ex.: arquitetura difícil de ser encontrada. Poderia ser utilizada para aplicar diferentes algoritmos em um mesmo dado.

Arquiteturas Paralelas

SIMD: o computador consiste de uma unidade de controle e N unidades de processamento. Todo o processamento está sobre o controle de um único fluxo de instruções mas opera sobre N fluxos de dados;

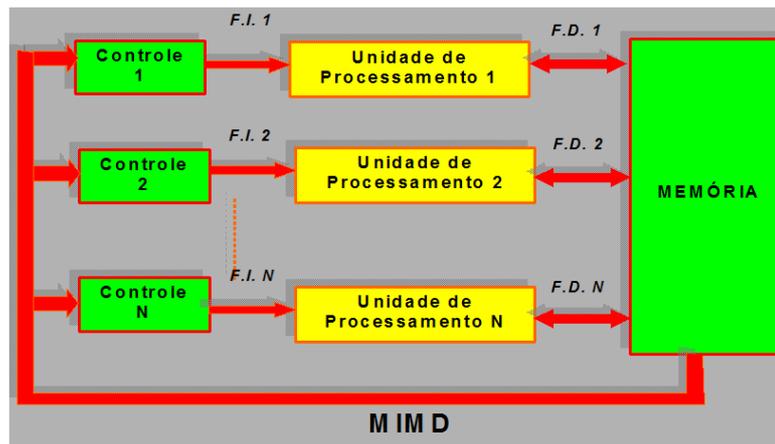


26
Out. 2009

Ex.: processadores vetoriais.

Arquiteturas Paralelas

MIMD: consiste de N processadores distintos, controlados por N fluxos de instruções e operando sobre N fluxos de dados.



27
Out. 2009

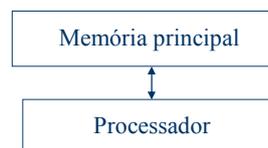
Ex.: multiprocessadores e multicomputadores.

Arquiteturas Paralelas

Modelos de acesso à memória

Um computador convencional consiste de um processador executando um programa armazenado na memória:

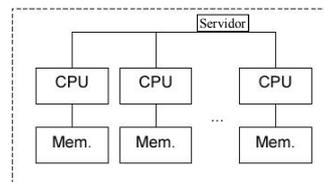
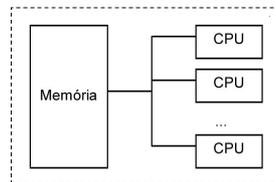
Instruções para o processador
Dados para ou do processador



28
Out. 2009

Modelos de acesso à memória

- Cada lugar da memória possui um endereço que inicia em 0 e vai até $2^n - 1$, onde n é o número de bits do endereço
- Em computação paralela, pode-se ter:
 - memória compartilhada (multiprocessadores)
 - memória distribuída (multicomputadores)

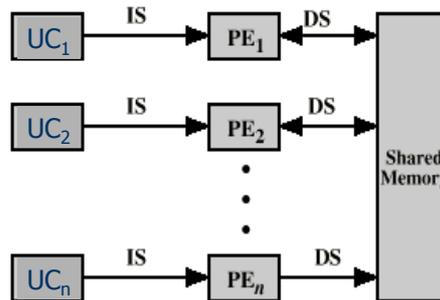


MIMD com Memória Compartilhada

- A mesma memória é acessada pelos múltiplos processadores
- **Sincronização entre tarefas** é feita por escrita/leitura na/da memória compartilhada e usuário é responsável por sua especificação
- Um lugar da memória não pode ser modificado por uma tarefa enquanto outra o estiver acessando
- Arquitetura "BlackBoard"

MIMD com Memória Compartilhada

- Comunicação entre tarefas é rápida
- Escalabilidade limitada pelo número de caminhos entre memória e processadores



31

Out. 2009

MIMD com Memória Compartilhada

- Usuário é responsável pela sincronização
- Programação:
 - Linguagens de programação paralela
 - Construções e instruções paralelas permitem declarações de variáveis compartilhadas e seções paralelas de código
 - Compilador responsável pela geração do código final executável
 - Threads
 - Sequências de código escritas em alto nível para processadores individuais que podem acessar localidades compartilhadas

Exemplos: SMP (Symetric MultiProcessors)
NUMA (NonUniform Memory Access)

32

Out. 2009

MIMD com Memória Distribuída

- Memória fisicamente distribuída entre os processadores e cada memória local só pode ser acessada pelo seu processador
- Tarefas se comunicam através de **troca de mensagens** e a sincronização entre as tarefas é feita através dessa troca

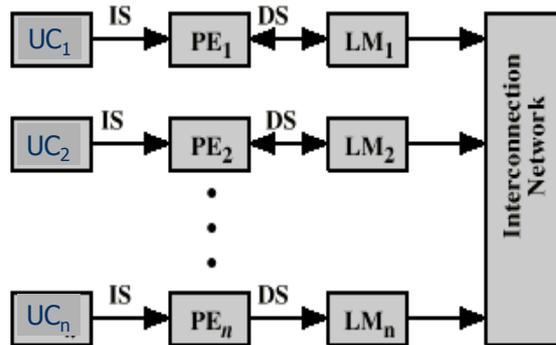
MIMD com Memória Distribuída

Programação:

- Bibliotecas com rotinas para passagem de mensagens que são ligadas a programas seqüenciais convencionais são bastante utilizadas;
- Problema dividido em um número de tarefas que se comunicam;
- MPI é um protocolo independente de linguagem usado para programar computadores paralelos.

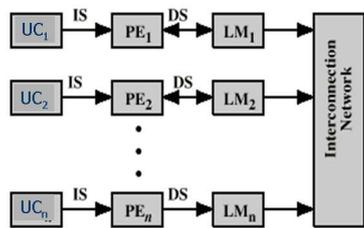
“Message Passing Interface (MPI) is a specification for an API that allows many computers to communicate with one another. It is used in computer clusters and supercomputers.”

MIMD com Memória Distribuída

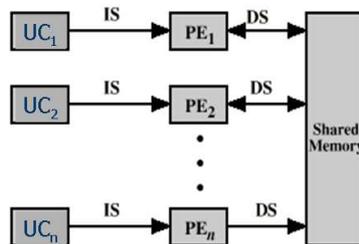


MIMD

Memória Distribuída



Memória Compartilhada



MIMD com Memória Distribuída

MPP (*Massively Parallel Processors*)

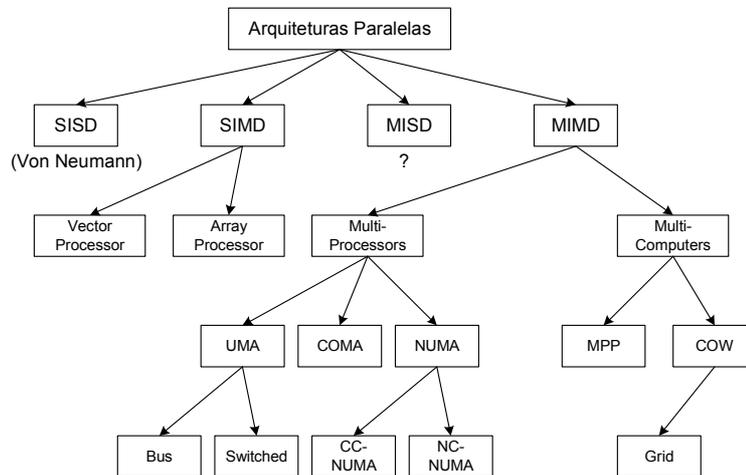
- interconectadas por rede de alta velocidade
- boa escalabilidade (podem ser formadas por uma grande quantidade de máquinas)
- complicadas de programar
- alto custo (em torno de US\$ 1.000.000)
- Exemplos:
 - Intel Paragon
 - Cray T3E
 - Thinking Machines CM-5

MIMD com Memória Distribuída

COW (*Cluster of Workstations*)

- Utilização de estações de trabalho em uma rede local de processamento
- Baixo custo
- Boa relação custo/benefício

Arquiteturas Paralelas



INFORMAÇÕES SOBRE A DISCIPLINA

USP - Universidade de São Paulo - São Carlos, SP
ICMC - Instituto de Ciências Matemáticas e de Computação
SSC - Departamento de Sistemas de Computação

Prof. Fernando Santos OSÓRIO

Web institucional: <http://www.icmc.usp.br/ssc/>

Página pessoal: <http://www.icmc.usp.br/~fosorio/>

E-mail: fosorio [at] icmc. usp. br ou fosorio [at] gmail. com

Disciplina de Arquitetura de Computadores / Informática

Estagiário PAE: Maurício A. Dias

Web disciplina: COTEIA - [Http://coteia.icmc.usp.br](http://coteia.icmc.usp.br)

> Programa, Material de Aulas, Critérios de Avaliação,

> Lista de Exercícios, Trabalhos Práticos, Datas das Provas