Fontes principais

1. J. Jaja, An introduction to Parallel Algorithms, Addison Wesley, 92

   ▷ Algoritmos paralelos

2. E. Cáceres, H. Mongeli, S. Song: Algoritmos paralelos usando CGM/PVM/MPI: uma introdução
   `http://www.ime.usp.br/~song/papers/jai01.pdf`

Por que discutir Computação Paralela?

- Buscar execuções cada vez mais rápidas dos programas.
- Processamento sobre grande volume de dados

Sistemas de computação paralela

Um **sistema de computação paralela** é uma coleção de processadores interconectados de maneira a permitir a coordenação de suas atividades e a troca de dados.

Os processadores trabalham simultaneamente, de forma coordenada para resolver um problema.

Aplicações do Paralelismo

▷ Problemas computacionalmente custosos, que demandam muito tempo de processamento em computadores sequenciais.

▷ Problemas com entrada de dados muito grande

▷ Problema que não possuem algoritmos sequenciais eficientes também não terão algoritmos paralelos eficientes. Problemas que possuem algoritmos sequenciais eficientes podem ter ou não algoritmos paralelos eficientes

Dificuldades no paralelismo

- Limitações Físicas
- Dependência de arquitetura
- Dificuldade de programação

Diferentes áreas do processamento paralelo

- Arquitetura de Computadores
- Algoritmos
- Redes
- Compiladores e linguagens de programação
- Sistemas operacionais
- Engenharia de software
- Banco de dados

Classificação de Flynn (1966)

Os sistemas de computação paralela podem ser definidos de acordo com:

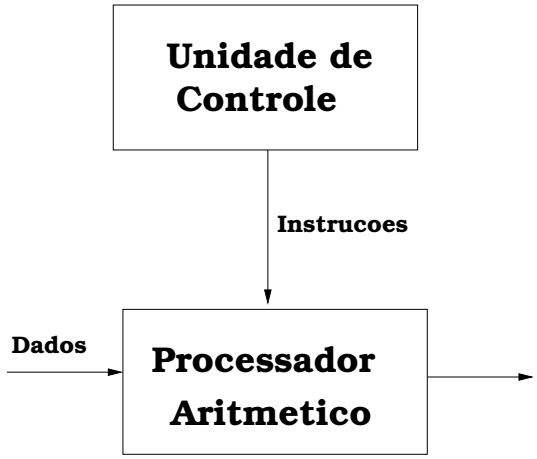
- ▷ Fluxo de instruções
- ▷ Fluxo de dados

	SD (Single Data)	MD (Multiple Data)
SI (Single Instruction)	SISD	SIMD
MI (Multiple Instruction)	MISD	MIMD

SISD - Single Instruction Single Data

Computador de Von Newman

- Computadores convencionais
- Uma CPU e uma unidade de controle conectadas por barramento.



SIMD - Single Instruction Multiple Data

Vários processadores homogêneos com acesso à memória compartilhada em tempo constante.

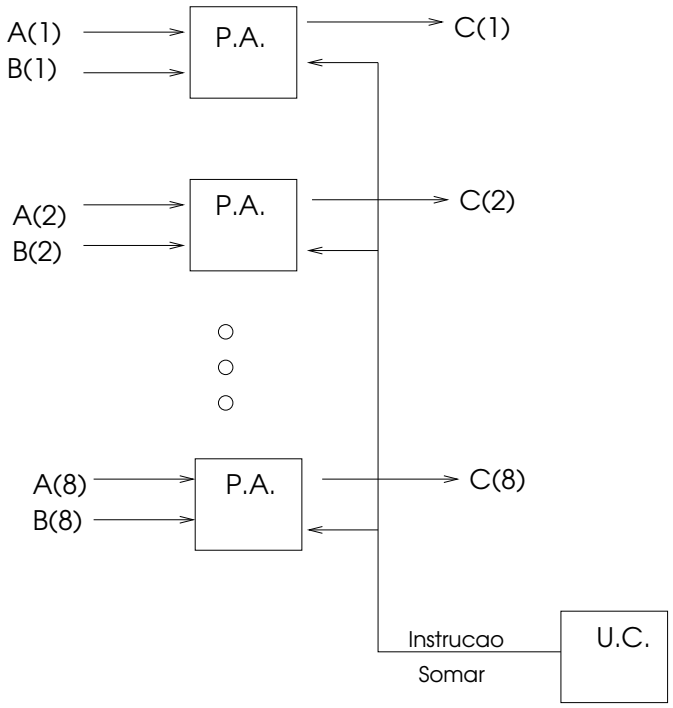
Mesma instrução executando sobre dados distintos

Ex.: Processador vetorial

Processador vetorial

Executa em 1 passo

C(1:8) = A(1:8) + B(1:8)

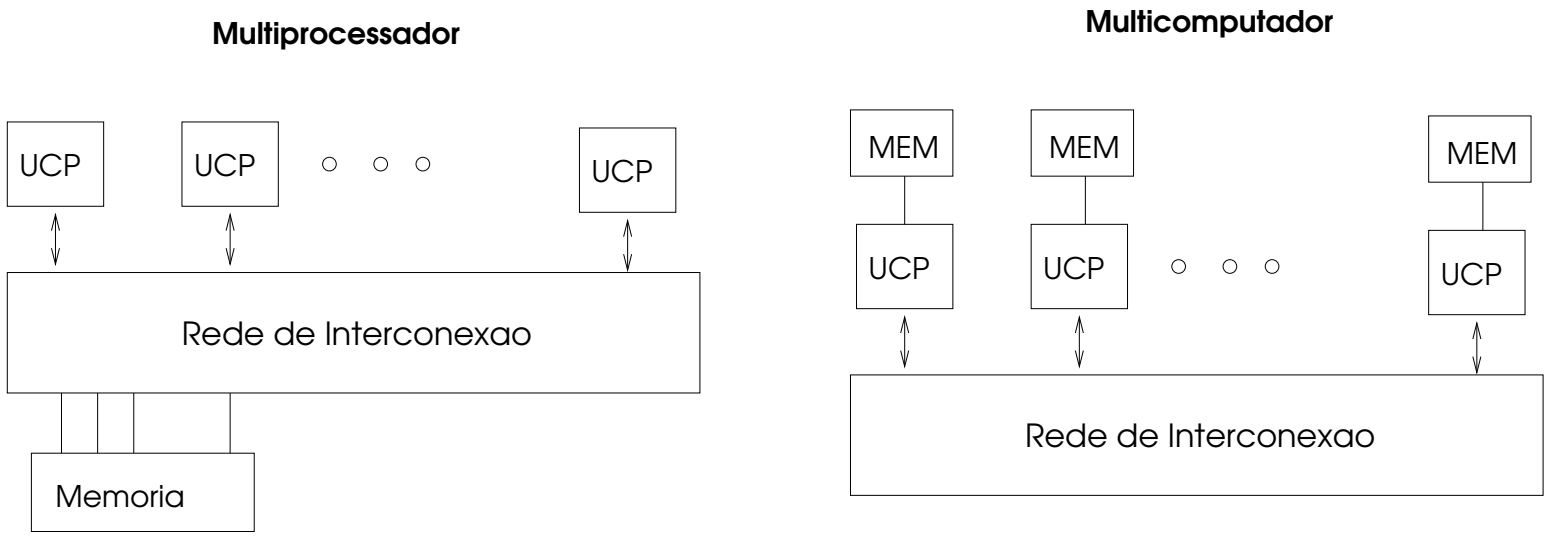


MIMD - Multiple Instruction Multiple Data

Cada processador executa o seu próprio fluxo de instruções.

- Multicomputadores
- Multiprocessadores

MIMD - Multiple Instruction Multiple Data



Sistemas de computação paralela

Existem outras formas de classificar os sistemas de computação paralela, entre eles, destacamos: **dispersão de computadores**, **estruturas de interconexão**, e **sincronismo**.

Modelos de computação paralela

Modelos de computação paralela

Existem vários tipos de arquiteturas paralelas que implementam diferentes sistemas de computação paralela. Para cada arquitetura paralela, temos modelos distintos de desenvolvimento de algoritmos paralelos.

Estes modelos, nos quais baseamos o desenvolvimento de algoritmos, são denominados **modelos de computação paralela e distribuída**.

Modelos de computação paralela

Memória compartilhada

- PRAM

Modelo de rede

- Array, Anel, Hipercubo, Malha, Torus, Estrela

Realístico

- BSP, CGM

Projeto de algoritmos paralelos

2 enfoques comuns:

▷ Paralelismo de dados: os dados são particionados, cada subconjunto de dados é associado a um processo, cada processo executa os mesmos comandos sobre o seu conjunto de dados.

▷ Paralelismo de controle: o problema é dividido em tarefas independentes, cada tarefa é associado com um processo, cada processo executa comandos diferentes.

Modelo PRAM

Modelo PRAM

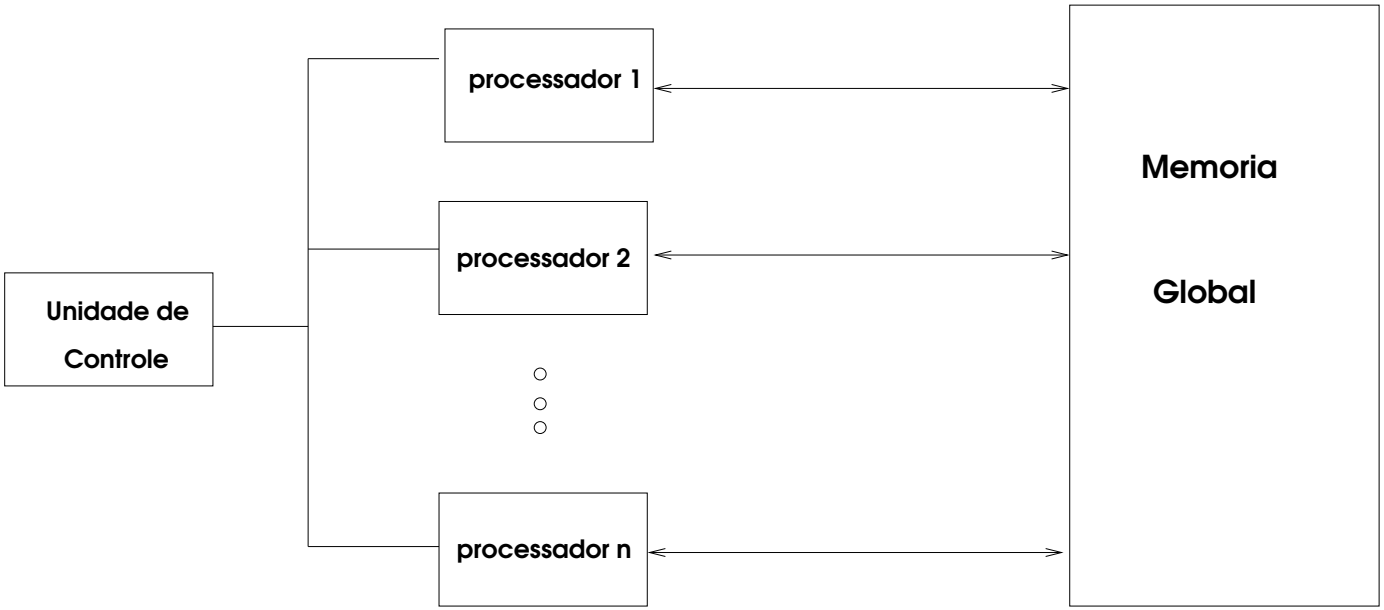
Nos algoritmos sequenciais o modelo **RAM (Random Access Machine)** é bastante próximo da forma de escrever algoritmos em máquinas Von Neuman.

O modelo **PRAM (Parallel Random Access Machine)** é uma extensão do modelo RAM.

Modelo PRAM

- O modelo PRAM consiste de um conjunto de processadores conectados a uma memória global.
- Existem n processadores ligados a uma memória global, e cada processador é identificado por um número inteiro.
- A memória global pode ser acessado por qualquer processador.

Modelo PRAM


processador 1
processador 2
processador n
Unidade de
Controle
Memoria
Global

Modelo PRAM

- Existe uma única unidade de controle, que passa a instrução a ser executada para todos os processadores (SIMD).
- Em cada instante, todos os processadores ativos executam a mesma instrução sobre dados possivelmente diferente.
- Existe um único relógio (clock) global compartilhado pelos processadores. Modelo síncrono.
- Processadores se comunicam através da memória compartilhada, utilizando variáveis compartilhadas.

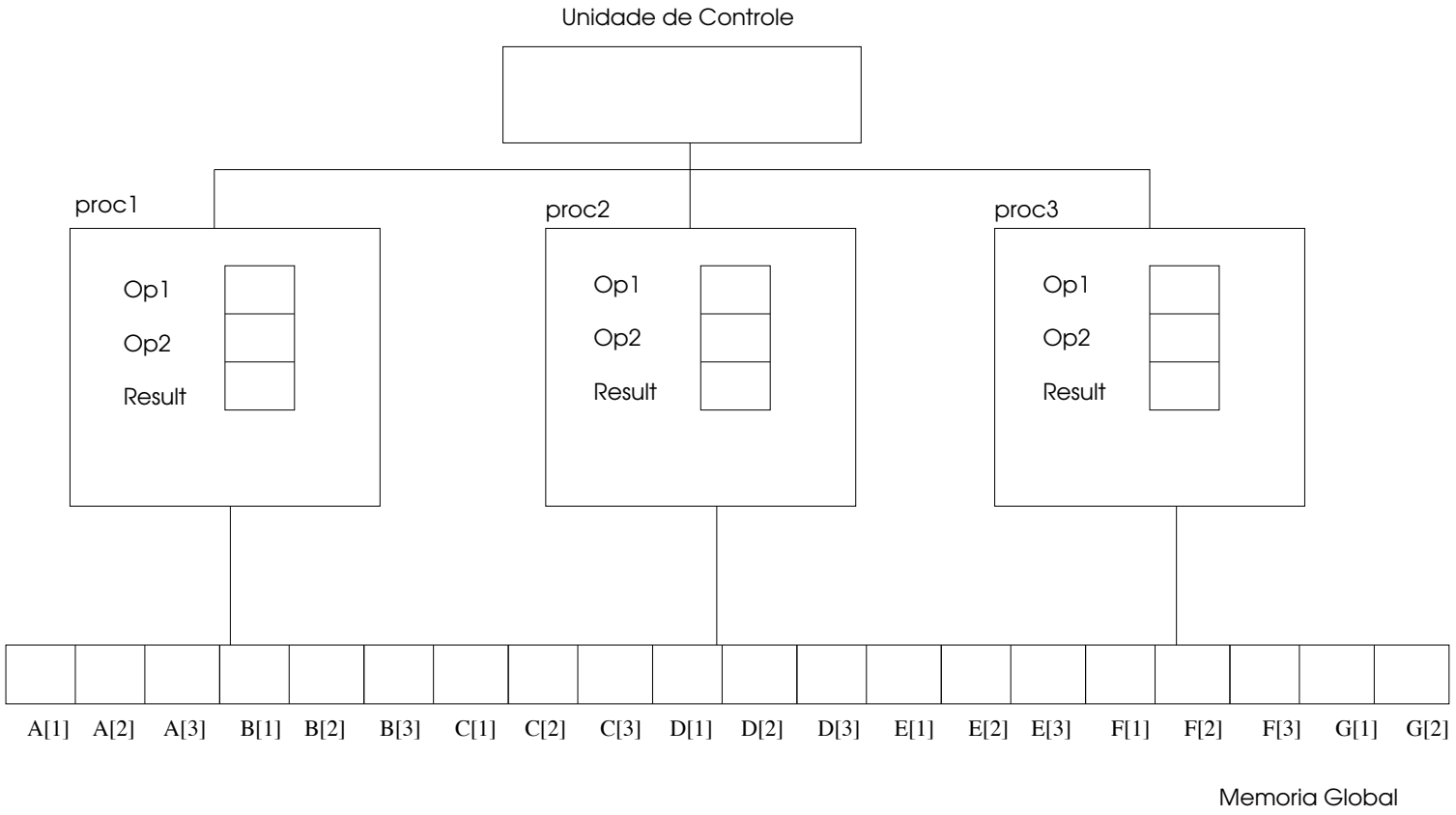
Exemplos

Algoritmos paralelos no modelo PRAM

para $1 \leq i \leq n-1$ faça em paralelo
  $C[i] := A[i] + B[i]$
  $F[i] := D[i] + E[i]$
fim para

para $1 \leq i \leq n-1$ faça em paralelo
  $G[i] := C[i+1]$
fim para

Modelo PRAM


Unidade de Controle
proc1
Op1
Op2
Result
proc2
Op1
Op2
Result
proc3
Op1
Op2
Result
A[1] A[2] A[3] B[1] B[2] B[3] C[1] C[2] C[3] D[1] D[2] D[3] E[1] E[2] E[3] F[1] F[2] F[3] G[1] G[2]
Memoria Global

Modelo de Computação Paralela e Distribuída

Modelo de Computação Paralela e Distribuída

- Modelo MIMD.
- Existem p processadores que executam em paralelo e estão interligados através de uma rede de interconexão.
- Cada processador é identificado por um número inteiro

Modelo de Computação Paralela e Distribuída

- Cada processador possui associado a ele uma memória local, acessível apenas a ele. Um processador não possui acesso à memória local associada aoutro processador (memória distribuída).

- Cada posição de memórias locais possui um endereço e pode ser acessada apenas por seu processador associado.

- A cada instante os processadores podem estar ativos ou inativos.

Modelo de Computação Paralela e Distribuída

- Cada processador possui sua unidade de controle, que passa a instrução a ser executada para ele.

- Em cada instante, cada processador ativo executa uma instrução, possivelmente diferente dos demais, sobre os dados possivelmente diferentes (MIMD) Rede de Interconexão

- Não existe relógio global. Cada processador possui o seu relógio local. Modelo assíncrono.

Modelo de Computação Paralela e Distribuída

- Mesmo que os processadores ativos estejam executando a mesma sequência de instruções, eles estarão executando a taxas diferentes.

- Processadores se comunicam através da rede de interconexão, usando troca de mensagens.

Modelo de Computação Paralela e Distribuída

- **envia(x, i)**: envia uma cópia de x ao processador P_i e passa a executar a próxima instrução (não bloqueante).

- **recebe(y, j)**: suspende a execução do seu programa até que dos dados do processador P_j sejam recebidos. O processador armazena os dados e continua a execução do programa (bloqueante).

Modelo de Computação Paralela e Distribuída

- Se um processador i precisar de um dado que o processador j calculou, para fazer um novo cálculo, o processador j envia este dado para o processador i (em uma mensagem), através da rede, e o processador i recebe este dado.

- Os termos de transmissão de mensagens através da rede são indeterminados (porém finitos).

Exemplos

Processador 1

para $1 \leq i \leq n-1$ faça
  $C[i] := A[i] + B[i]$
  se $i \neq 1$ então
    $envia(C[i], proc3)$
  fim se
fim para

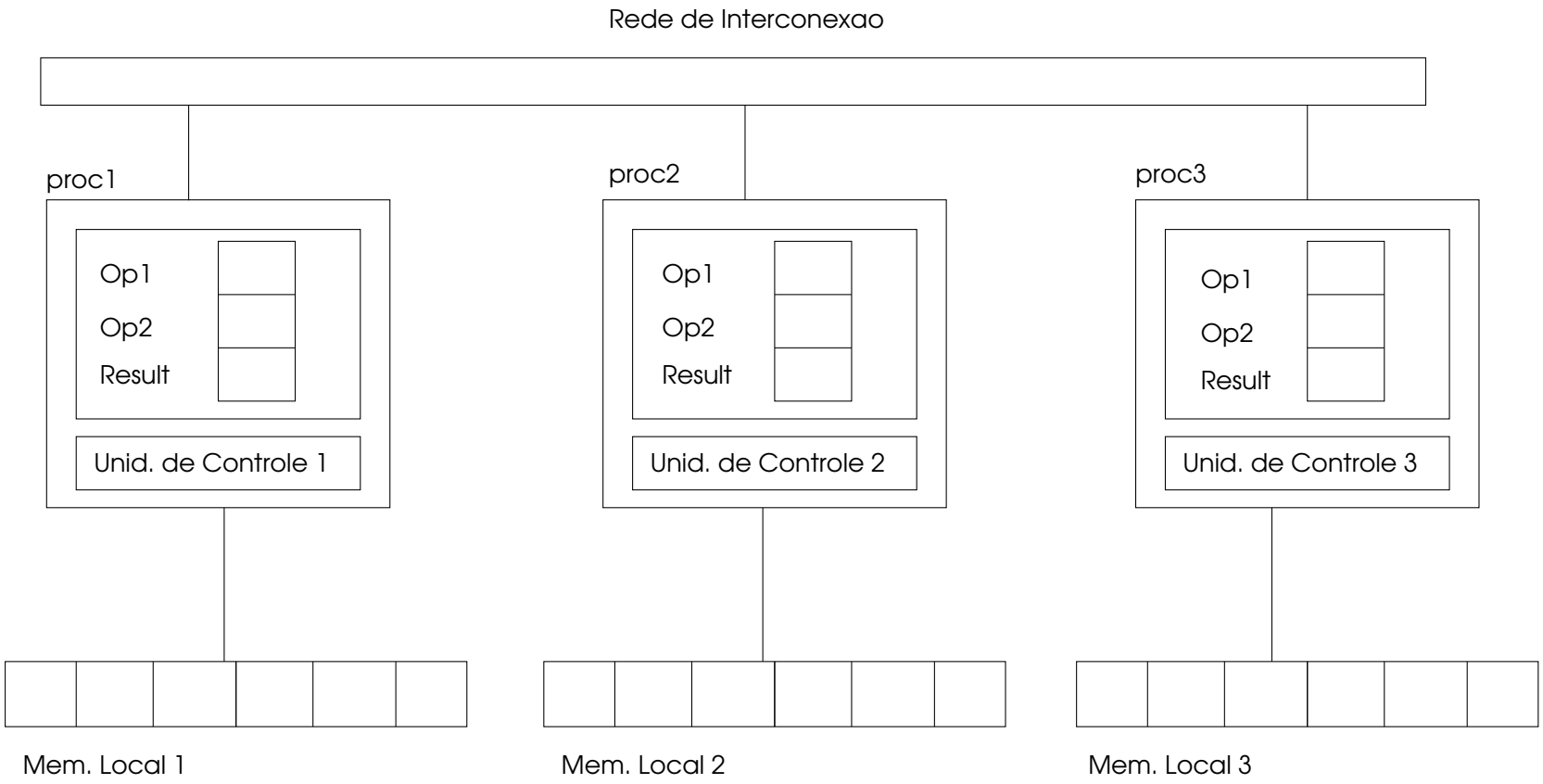
Processador 2

para $1 \leq i \leq n-1$ faça
  $F[i] := D[i] + E[i]$
fim para

Processador 3

para $1 \leq i \leq n-1$ faça
  $recebe(M, proc1)$
  $G[i] := M$
fim para

Modelo de Computação Paralela e Distribuída



Comparações entre modelos

Modelo PRAM

- Execução síncrona
- Memória compartilhada
- Comunicação através de memória compartilhada

Modelo Distribuído

- Execução assíncrona
- Memória distribuída
- Comunicação através de troca de mensagens

Fim