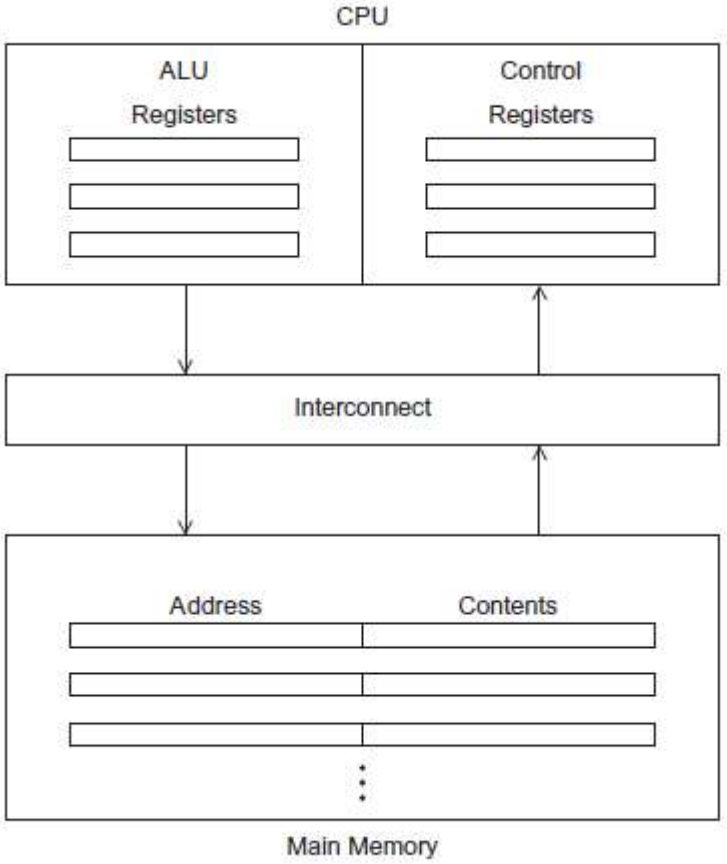
Introdução Programação Paralela - Hardware

Aula 3

Arquitetura Convencional

- A máquina de Van Neuman

Unidade Lógica Aritmética



Barramento de dados e instruções

Periféricos

Arquitetura Van Neuman

- Embora bastante eficiente em sua origem, é um problema
  - Acesso sequencial
- Uma alternativa para acelerar os acessos aos dados e instruções e execução deles
  - Memória Cache
  - Pipeline

Cache

- Uma memória de acesso muito rápido
- Normalmente localizada no mesmo chip do processador
- Realiza o armazenamento intermediário de dados e instruções
- Princípio de Localidade

Princípio de Localidade

- Lembranças mais recentes são armazenadas em memórias menores de curta duração
- Enquanto as mais antigas em memórias de longa duração e maior capacidade
- Pode ser
  - Temporal
  - Espacial

Temporal

- Um dado acessado recentemente tem mais chance de ser usado novamente do que um dado usado a mais tempo

- Sendo assim, o sistema de memória tende a manter os dados e instruções recentemente acessados no topo da hierarquia da memória

Espacial

- Há uma probabilidade de acesso maior para dados e instruções em endereços próximos àqueles acessados recentemente

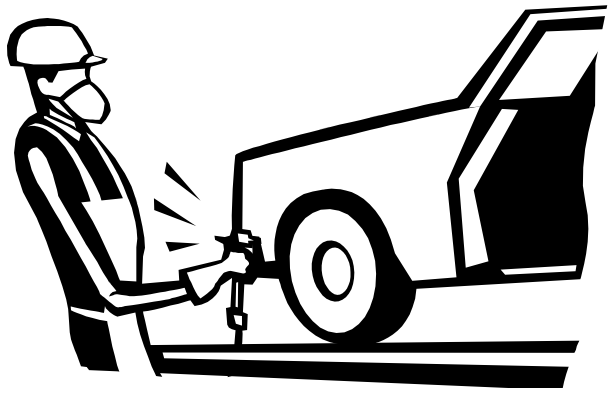
- Sendo assim, quando uma instrução é acessada, a instrução com maior probabilidade de ser executada em seguida é a instrução logo a seguir dela

Pipelines

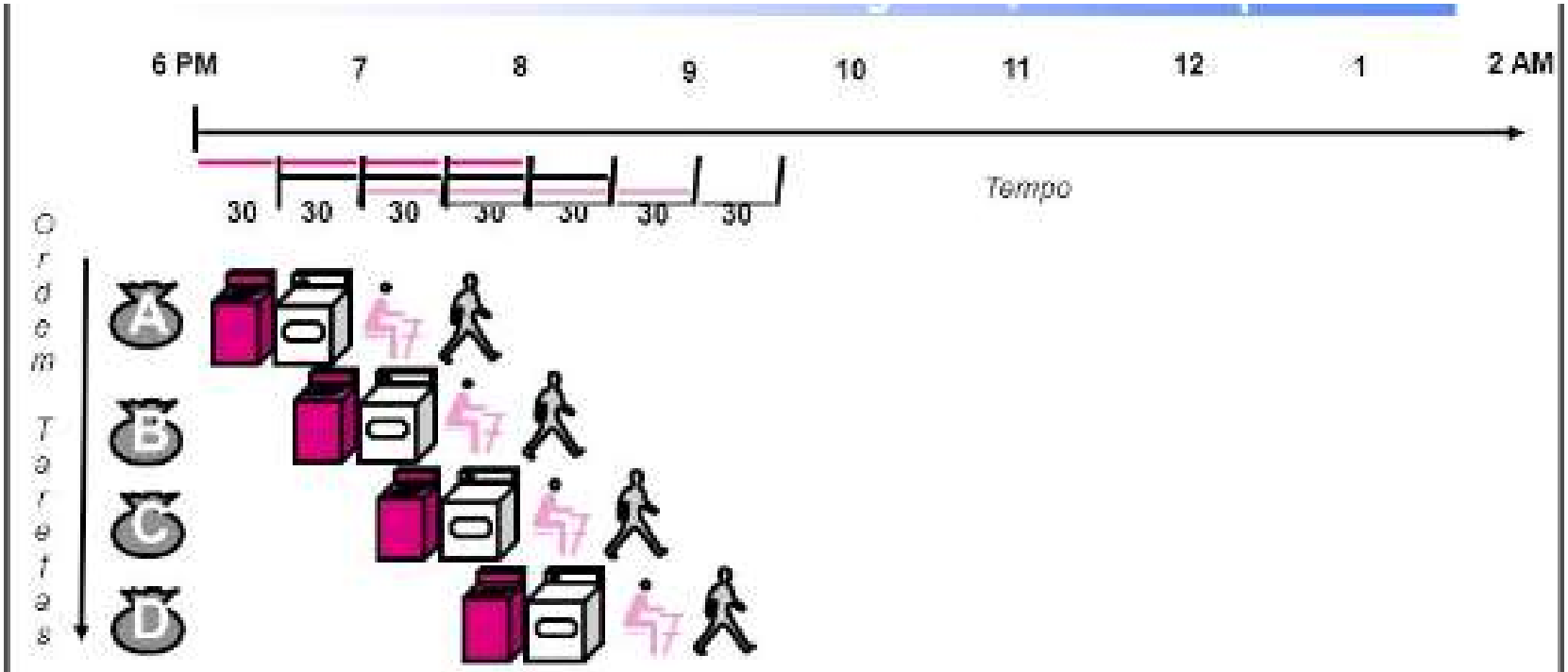
- É uma técnica de Hardware
- Unidades especializadas
- Próxima instrução geralmente está armazenada nos registradores da CPU
- Ganho substancial de velocidade

Pipeline - Problemas

- Dependência de instruções
- Desvios
  - Ex: um retorno do programa

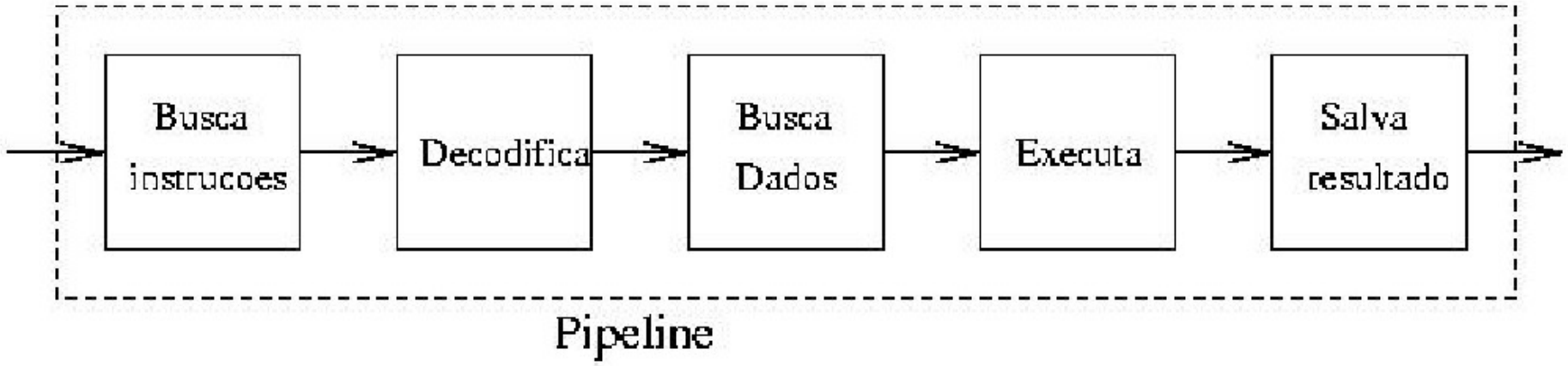


Pipeline



- **Utilizando-se a técnica de pipeline consome-se 3,5 horas no processo de lavagem !**

Pipeline


Busca
instrucoes
Decodifica
Busca
Dados
Executa
Salva
resultado
Pipeline

Superescalar

- Consiste em aumentar o número de pipeline
- Vantagens
- Paralelismo real
  - 2 ou mais instruções executando simultaneamente
- Desvantagens
  - Aumento de complexidade
  - Problemas de dependência e desvios

Outra alternativa

- Uso de arquiteturas paralelas

- A forma da estruturação de hardware paralelo pode variar bastante

Classificação de Flyn

- SISD
  - Sistemas convencionais
- SIMD
  - Computadores vetoriais
- MISD
  - Arrays sistólicos
- MIMD
  - Paralelismo massivo

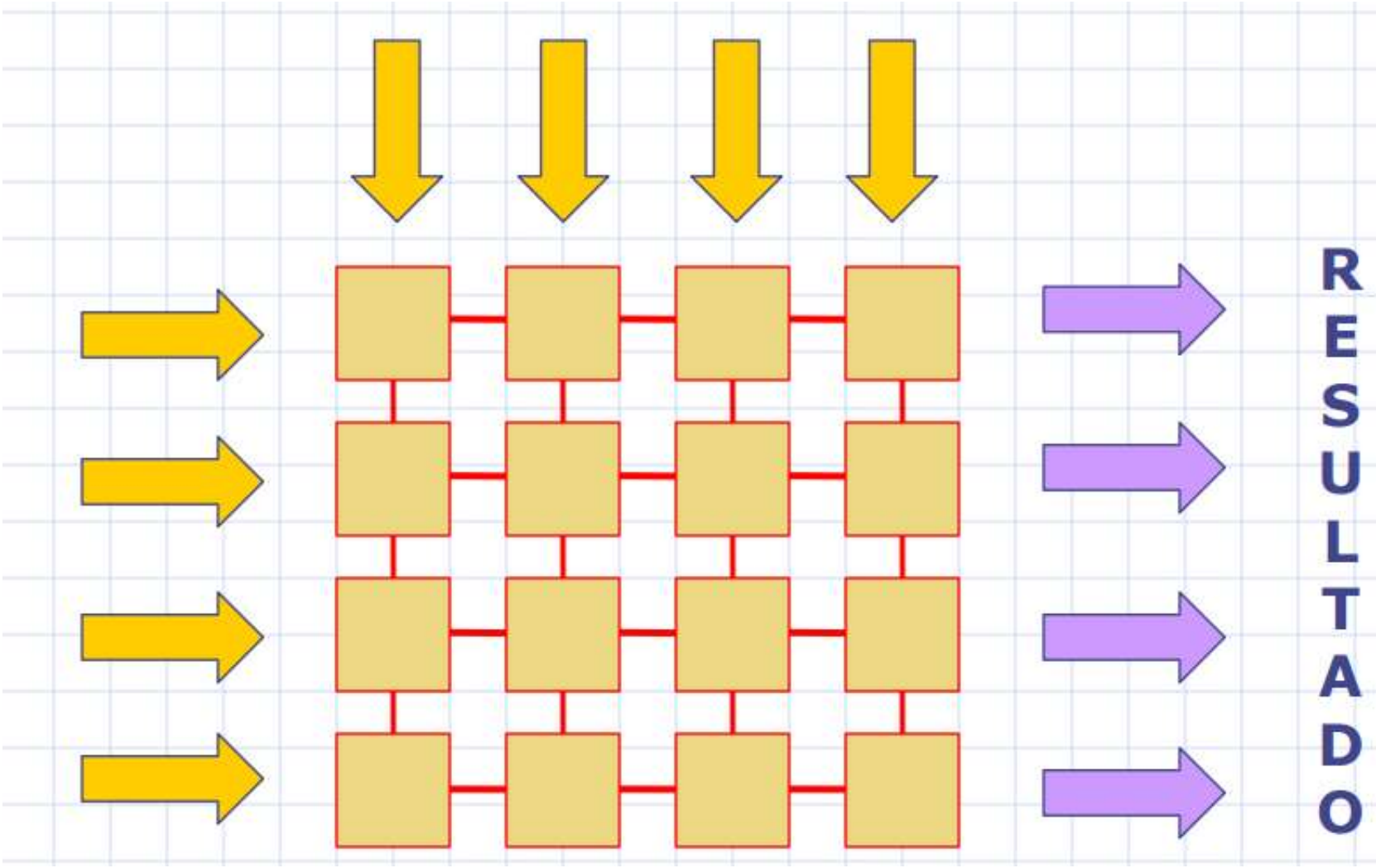
Array Sistólicos

- Sistema composto por elementos dispostos matricialmente
- Cada elemento computa uma função simples e única
- Cada elemento está conectado apenas aos seus vizinhos mais próximos através de um caminho unidirecional

Array Sistólicos

- Um único sinal de controle enviado através do arranjo é um pulso de relógio, utilizado para sincronizar as operações
- Os dados de entradas são inseridos nos dois lados da matriz e passam através do arranjo em duas direções ortogonais
- Os dados são extraídos como uma série de elementos de um dos dois lados restantes da matriz
- EX: Multiplicação de matriz

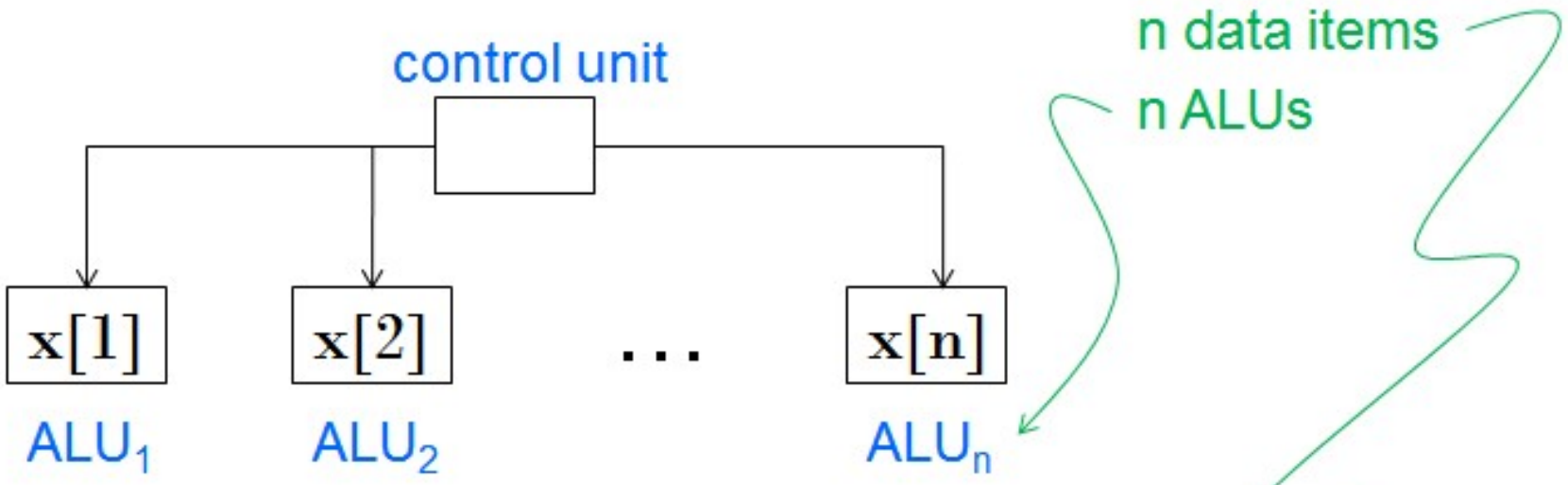
Array Sistólicos



SIMD

- Paralelismo alcançado dividindo os dados entre os processadores
- Aplica-se a mesmo instrução para múltiplos dados
- Chamado de paralelismo de dados

SIMD




```
for (i = 0; i < n; i++)
    x[i] += y[i];
```

SIMD

- E se tivéssemos mais dados do que ALU?
- Dividia-se o trabalho de forma iterativa

Round3	ALU_1	ALU_2	ALU_3	ALU_4
1	X[0]	X[1]	X[2]	X[3]
2	X[4]	X[5]	X[6]	X[7]
3	X[8]	X[9]	X[10]	X[11]
4	X[12]	X[13]	X[14]	

Inconvenientes

- Todas as ALUs são necessárias para executar a mesmo instrução ou permanecem inativas
- Normalmente, devem operar de forma síncrona
- As ALUs não possuem instruções de armazenamento
- Não eficiente para paralelismo mais complexo

Processadores vetoriais

- Implementa conjunto de instruções que operam em matrizes ou vetores de dados
- Cada instrução é um loop
- Registradores vetoriais

Processadores vetoriais

- Possui unidades funcionais vetorizadas e de pipeline SIMD
- Memória intercalada
  - Múltiplos banco de memória, que podem ser acessados independentemente
  - Distribuí elementos de um vetor em vários bancos

Processadores vetoriais

- Contras
  - Escalabilidade
  - Incapacidade de lidar com estrutura de dados irregulares
- Prós
  - Rápido
  - Fácil de usar
  - Largura de banda (taxa de transferência) da memória é alto
  - Excelente para aceleradores gráficos

Processadores Vetoriais - Exemplo

- Imagine mudar o brilho da tela
- Perca de tempo utilizando um processador tradicional (escalar)
- Já utilizando processadores vetoriais
- Ex: Playstation 3 usada o processador CELL
  - Composto por um escalar e 8 vetoriais

Perspectivas

- São ainda equipamentos preferidos no setor automobilísticos , aeroespaciais e jogos

- Na arquitetura das GPU
  - núcleos vetoriais(SIMD) e
  - Superescalares (MIMD)

MIMD

- Suporta múltiplos fluxos de instrução simultânea que operam em múltiplos fluxos de dados

- Consistem de um conjunto de unidades ou núcleos de processamento totalmente independentes

Como obter Paralelismo?

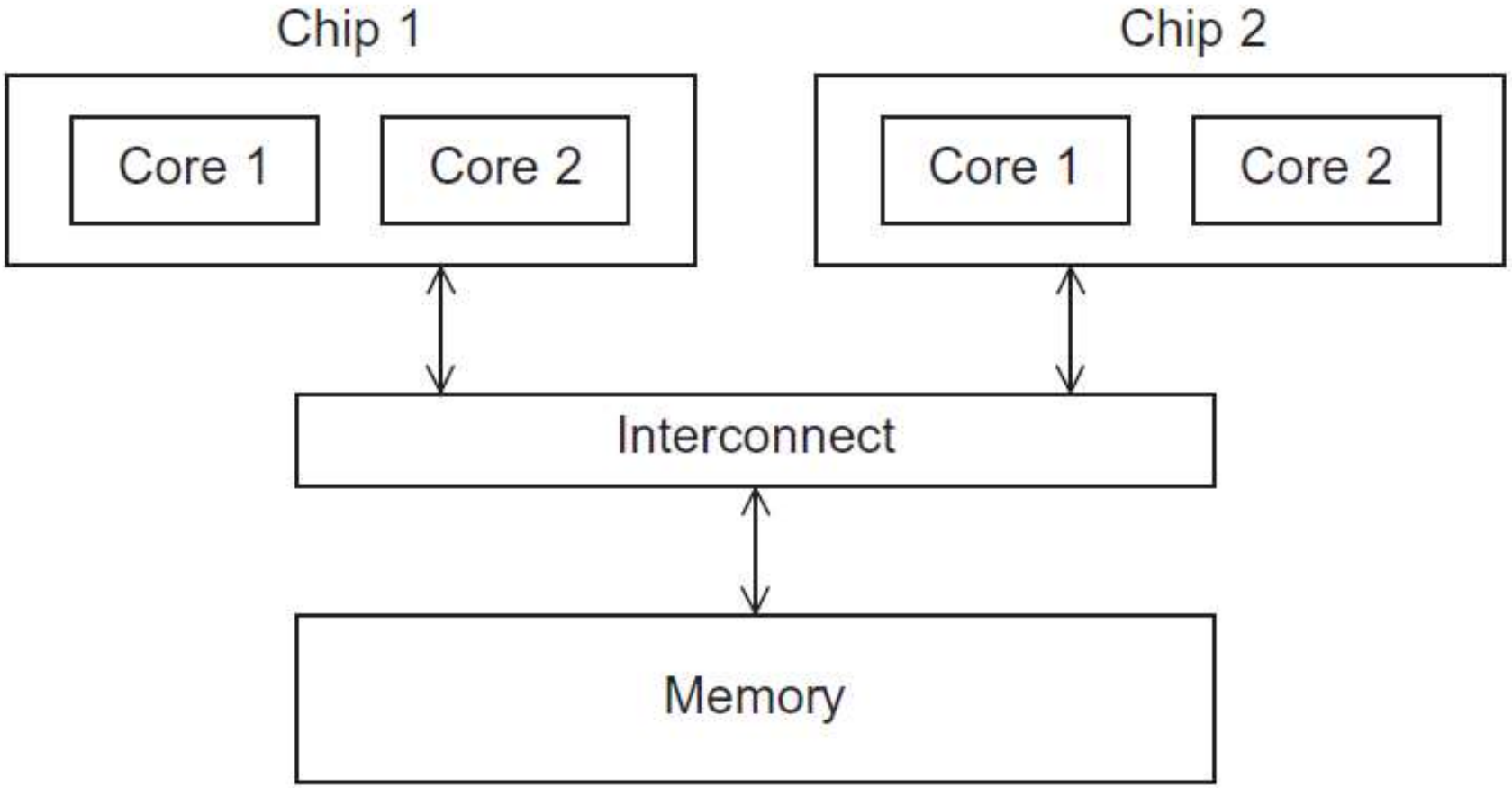
- Basicamente 2 sistemas
  - Os multiprocessadores
  - Os multicomputadores
- A diferença é a forma do acoplamento entre os elementos de processamento
- O que muda nos processadores?
  - Nada, exceto a quantidade de elementos de processamento
- Na memória?
  - Quase tudo

Memória compartilhada – Diferenças em termo de Hardware

- Arquitetura UMA (Uniform Memory Acess) ou conhecido como SMP(Multiprocessadores Simétricos)

- Arquitetura NUMA (Non- Uniform Memory Acess)

UMA

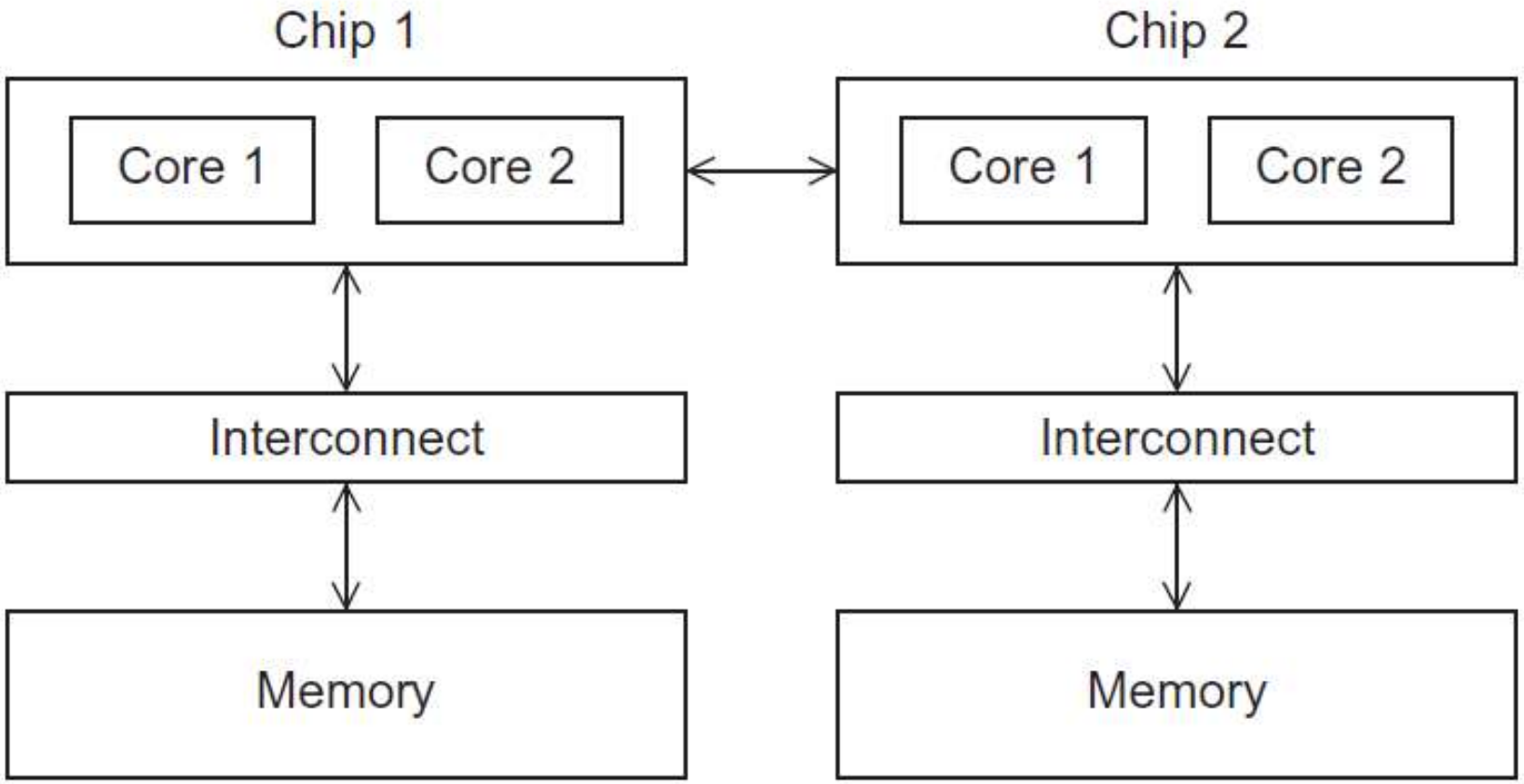


UMA

- Problemas de contenção

- Escalabilidade reduzida

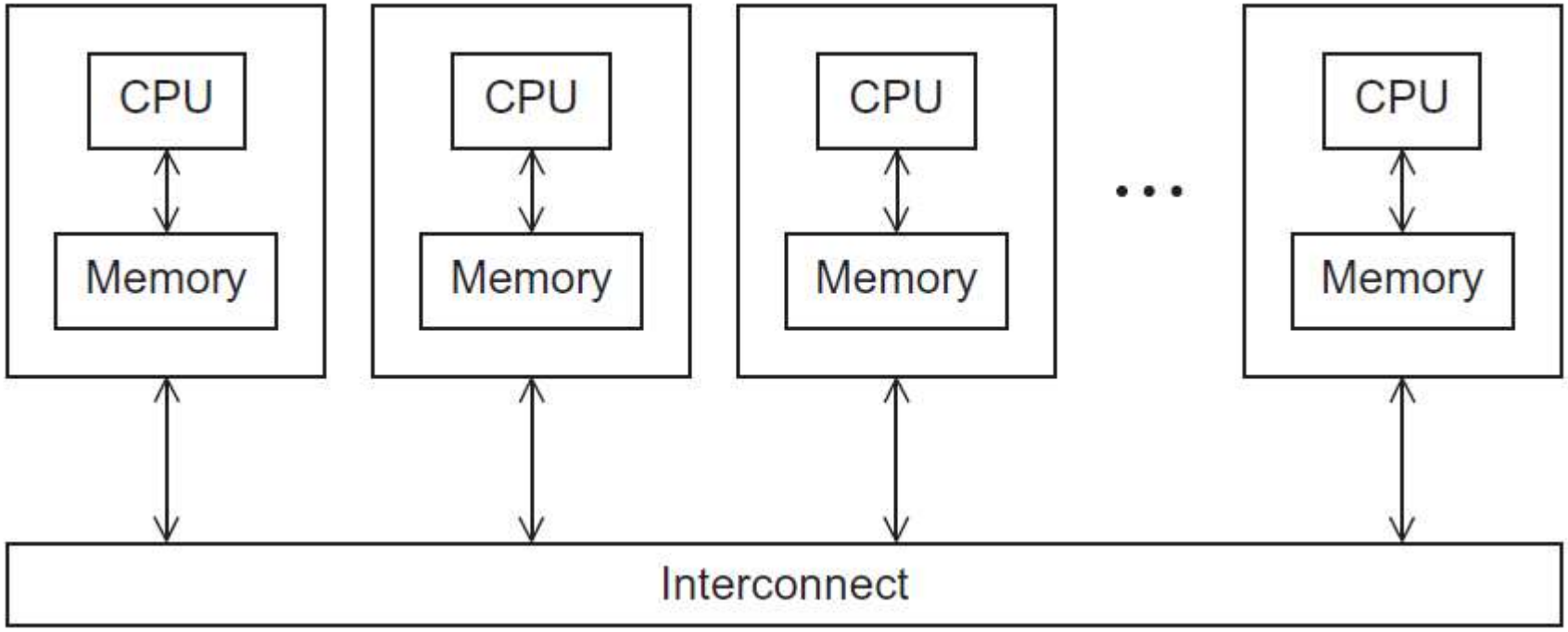
NUMA


Chip 1
Chip 2
Core 1
Core 2
Core 1
Core 2
Interconnect
Interconnect
Memory
Memory

MultiComputadores - Distribuído

- O padrão de acesso é feito através de troca de mensagem
- MPI, RMI, RPC, etc...

Sistema de memória distribuída



Metodologia de Foster

- A maior parte dos problemas têm várias soluções paralelas
- PCAM ou Metodologia de Foster

Metodologia de Foster

- Consiste em 4 etapas
  - Particionamento
  - Comunicação
  - Agrupamento
  - Mapeamento

Metodologia de Foster

- Particionamento
  - Divide os cálculos a ser realizados juntamente com os dados a serem operados em pequenas tarefas
- Comunicação
  - Determina qual comunicação deve ser realizada entre as tarefas identificadas da etapa anterior

Metodologia de Foster

- Aglomeração ou agregação
  - Combina tarefas e comunicações identificadas na primeira etapa em tarefas maiores

Metodologia de Foster

- Mapeamento
  - Atribuir as tarefas compostas identificadas no passo anterior para as threads/processos.
  - Isso deve ser feito para minimizar a comunicação
  - E para cada processo/thread tenham aproximadamente a mesma quantidade de trabalho

Métricas - Objetivos

- Desempenho
- Escalabilidade

Métricas

- Fatores que condicionam tais métricas
- Limites Arquiteturais
  - Latência e largura de banda
  - Coerência dos dados
  - Capacidade de memória
- Limites Algorítmicos
  - Falta de paralelismo (código sequencial/concorrência)
  - Frequência de comunicação e Sincronização
  - Escalonamento Deficiente (granularidade das tarefas/ balanceamento de carga)

Métricas de Desempenho

- Há métricas para 2 classes distintos
  - Métricas de desempenho para processadores
  - Métricas de desempenho para aplicações paralelas

Métricas de Desempenho para Aplicações Paralelas

- Há várias medidas que permitem medir/ avaliar o desempenho em uma aplicação paralela
  - Speedup
  - Eficiência
  - Redundância
  - Utilização
  - Qualidade

Métricas de Desempenho para Aplicações Paralelas

- Existe várias leis/métricas que permitem direcionar o comportamento de uma aplicação paralela para o seu potencial desempenho
  - Lei de Amdahl
  - Lei de Gustafson-Baris
  - Métrica de Karp-Flatt
  - Métricas de Isoeficiencia

Speedup

- É uma medida do grau do desempenho

$$S(p) = \frac{T(1)}{T(p)}$$

$T(1)$ é o tempo de execução com um processador

$T(p)$ é o tempo de execução com p processadores

	1 CPU	2 CPUs	4 CPUs	8 CPUs	16 CPUs
$T(p)$	1000	520	280	160	100
$S(p)$	1	1,92	3,57	6,25	10,00

Eficiência

- É uma medida do grau de aproveitamento dos recursos computacionais

$$E(p) = \frac{S(p)}{p} = \frac{T(1)}{p \times T(p)}$$

$S(p)$ é o speedup para p processadores

	1 CPU	2 CPUs	4 CPUs	8 CPUs	16 CPUs
$S(p)$	1	1,92	3,57	6,25	10,00
$E(p)$	1	0,96	0,89	0,78	0,63

Redundância

- É uma medida do grau do aumento da computação

$$R(p) = \frac{O(p)}{O(1)}$$

$O(1)$ é o número total de operações realizadas com 1 processador

$O(p)$ é o número total de operações realizadas com p processadores

	1 CPU	2 CPUs	4 CPUs	8 CPUs	16 CPUs
$O(p)$	10000	10250	11000	12250	15000
$R(p)$	1	1,03	1,10	1,23	1,50

Utilização

- É uma medida do grau de aproveitamento da capacidade computacional

$$U(p) = R(p) \times E(p)$$

	1 CPU	2 CPUs	4 CPUs	8 CPUs	16 CPUs
R(p)	1	1,03	1,10	1,23	1,50
E(p)	1	0,96	0,89	0,78	0,63
U(p)	1	0,99	0,98	0,96	0,95

Qualidade

- É uma medida do grau de importância de utilizar programação paralela

$$Q(p) = \frac{S(p) \times E(p)}{R(p)}$$

	1 CPU	2 CPUs	4 CPUs	8 CPUs	16 CPUs
S(p)	1	1,92	3,57	6,25	10,00
E(p)	1	0,96	0,89	0,78	0,63
R(p)	1	1,03	1,10	1,23	1,50
Q(p)	1	1,79	2,89	3,96	4,20

Lei Amdahl

- A computação realizada por uma aplicação paralela pode ser dividida em 3 classes
  - C(seq)
  - C(par)
  - C(com)

Lei Amdahl

- Se f for a fração da computação que só pode ser realizada sequencialmente, e
- 0<=f<=1, então o speedup máximo de uma aplicação paralela com p processadores é:

$$S(p) \leq \frac{1}{f + \frac{1-f}{p}}$$

Lei de Amdahl

- No entanto, pode ser utilizada também para determinar o limite máximo de speedup que uma determinada aplicação poderá alcançar independentemente do número de processadores a utilizar

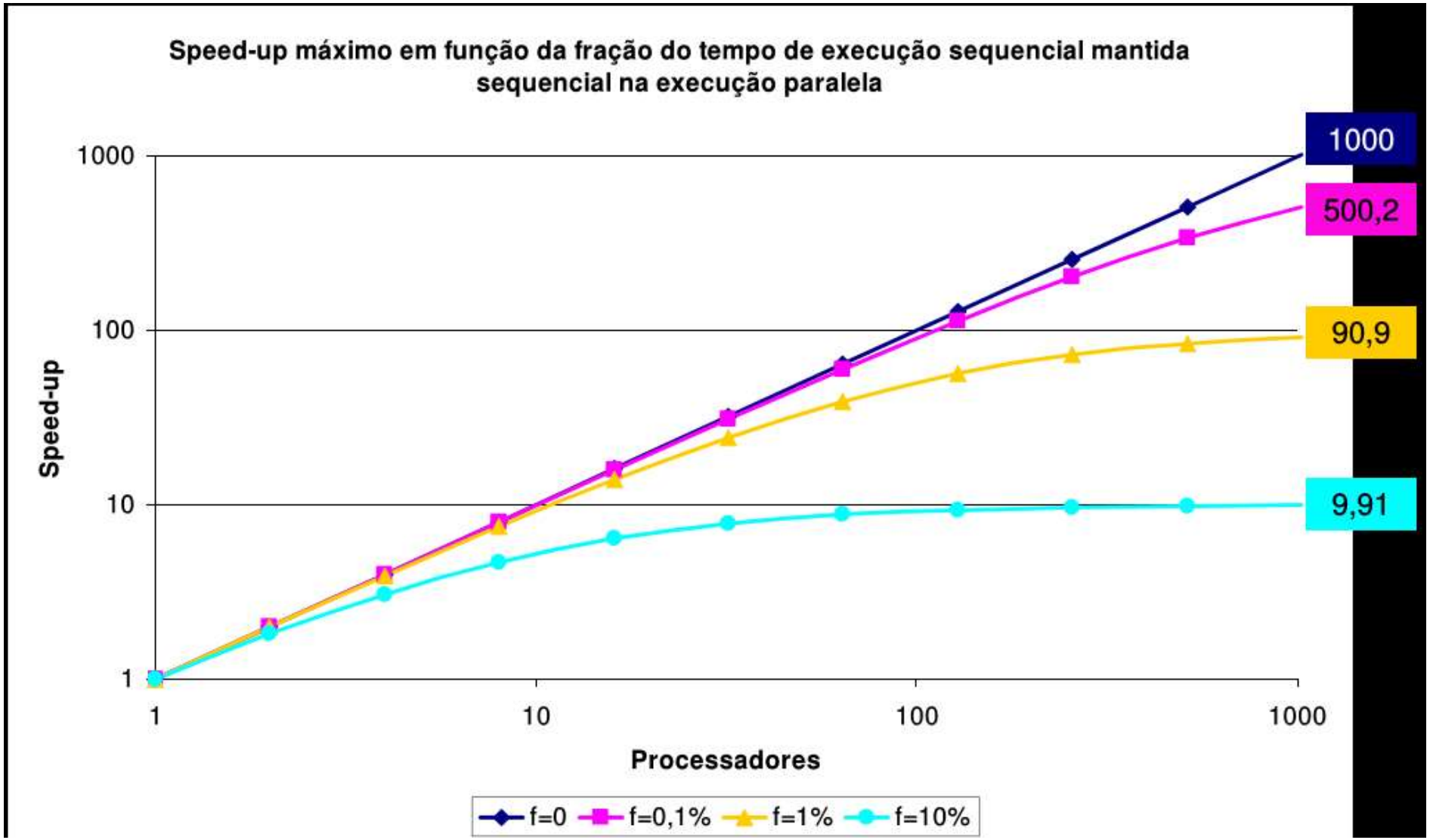
$$\lim_{p \to \infty} \frac{1}{0{,}1 + \frac{1 - 0{,}1}{p}} = 10$$

Lei de Amdahl

$$S(p) \leq \frac{1}{0{,}1 + \frac{1-0{,}1}{8}} \approx 4{,}71$$

- Suponha que pretende determinar se é vantajoso desenvolver uma versão paralela de uma determinada aplicação sequencial. Por experimentação, verificou-se que 90% do tempo de execução é passado em procedimentos que se julga ser possível paralelizar. Qual é o speedup máximo que se pode alcançar com uma versão paralela do problema executando em 8 processadores?

Lei de Amdahl



Escalabilidade

- Capacidade de manter a mesma eficiência à medida que o número de processadores e a complexidade do problema aumentam proporcionalmente

- Capacidade de utilizar mais recursos computacionais de forma efetiva

Conclusões

- Sistema serial
  - O padrão de hardware tem sido a arquitetura proposta por Von Neuman
- Hardware Paralelo
  - Taxonomia de Flynn
- Software Paralelo
  - Preocupar-se em desenvolver software para sistemas MIMD
- Design para Programas Paralelos
  - Metodologia de Foster

Conclusões

- Desempenho
  - Métricas
  - Lei de Amdahl

Rede

- Afeta o desempenho da memória compartilhada e distribuída
- Possui 2 categorias
- Rede de memória compartilhada
- Rede de memória distribuída

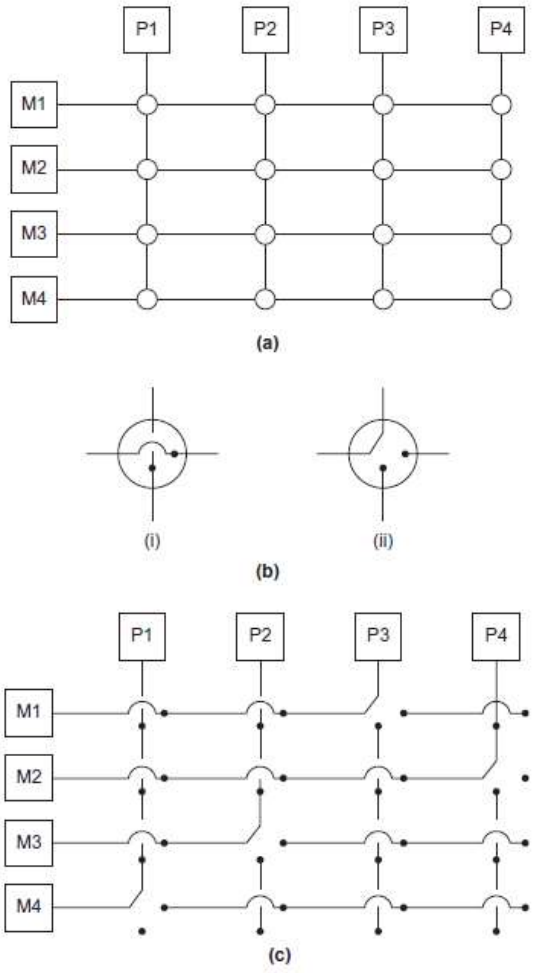
Rede de Memória Compartilhada

- Rede de Barramento
- Uma coleção de fio para comunicação paralela que junto com algum hardware controla o acesso ao barramento
- A comunicação pelos fios são compartilhadas pelos dispositivos que são conectados a ele
- Como o número de dispositivos conectado ao barramento aumenta
- A disputa pelo seu uso também
- Diminuindo o desempenho

Rede de Memória Compartilhada

- Interconexão Switched (comutada)
  - Utiliza chaves para controlar a rota dos dados entre os dispositivos conectados
  - Comutação CrossBar
    - Permite comunicação simultânea entre diferentes dispositivos
    - Mais rápido do que barramentos
    - Porém o custo dos aparelhos são relativamente maiores
    - Utiliza relês para controlar o fluxo das informações

- Uma rede crossbar com 4 Processadores e 4 módulos de memória
- A configuração interna de um crossbar
- Acesso simultâneo à memória pelos processadores



Rede de Conexão de Memória Distribuída

- 2 Grupos
- Interconexão direta
  - Cada switch é diretamente conectado a um par de processador e memória
  - E os switchs são ligados uns aos outros
- Interconexão indireta
  - Switchs não podem ser conectados diretamente a um processador