

---

**Desenvolvimento do Algoritmo Paralelo** 

**Desenvolvimento do Programa Paralelo** 

**Mapeamento de Processos** 

**Teste e Depuração de Prog. Paralelos**

**Avaliação de Desempenho de Prog.  
Paralelos**

# Mapeamento de Processos

## ■ Questões a serem tratadas:

- Onde cada processo deve ser executado?
- Como a comunicação entre processos será viabilizada no sistema de interconexão existente?

## — De forma geral, tem-se:



## — Em Arquiteturas Paralelas com memória distribuída



Necessidade de Mapeamento

# Mapeamento de Processos

- O mapeamento de processos paralelos visa minimizar o tempo de processamento;
- Duas estratégias são conflitantes:
  - Processos que podem executar concorrentemente



Mapeados em processadores diferentes

- Processos que se comunicam freqüentemente



Mapeados no mesmo processador

# Mapeamento de Processos

- Alguns fatores que devem ser verificados:
  - Processos com carga variável;
  - Comunicação não-uniforme;
  - Carga imposta por processo é dinâmica;
  - Comunicação entre tarefas diferentes durante execução.
- A partir desses fatores ...
  - O mapeamento fica mais complexo;
  - Necessidade de técnicas de escalonamento/balanceamento de carga.

# Mapeamento de Processos

## ■ É desejável:

- Garantir que todos os EPs (Elementos de Processamento) sejam bem aproveitados;
- Garantir trabalho suficiente e compatível com a capacidade global do sistema;
- Garantir que os EPs tenham uma carga uniforme.

## ■ Assim, deve-se distribuir os processos buscando:

- Minimizar tempo de execução;
- Minimizar atrasos por comunicação;
- Maximizar utilização dos recursos.

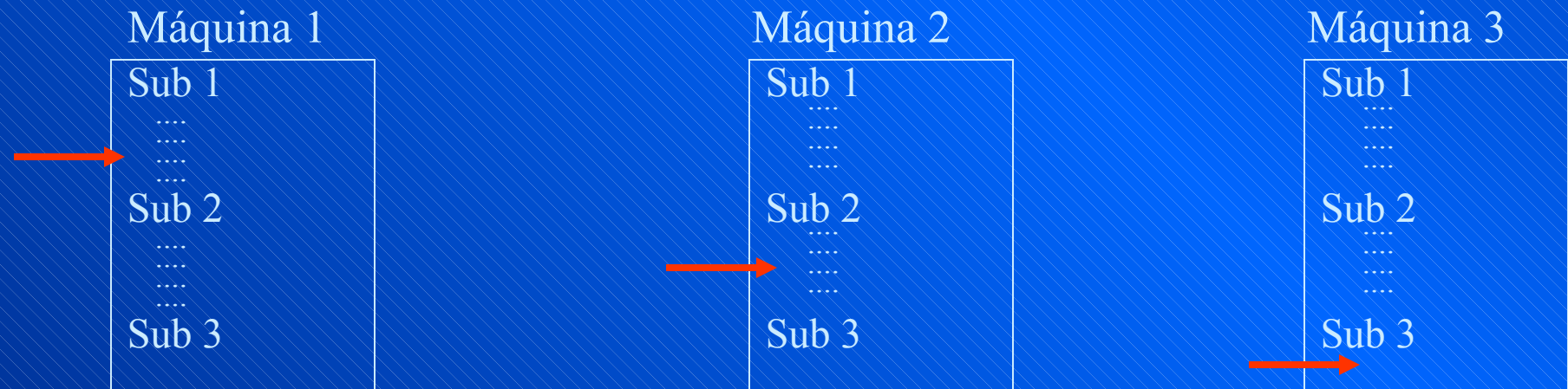
# Mapeamento de Processos

## ■ Dois conceitos importantes:

- Processos SPMD (*Single Program Multiple Data*);
- Processos MPMD (*Multiple Program Multiple Data*);

## ■ SPMD

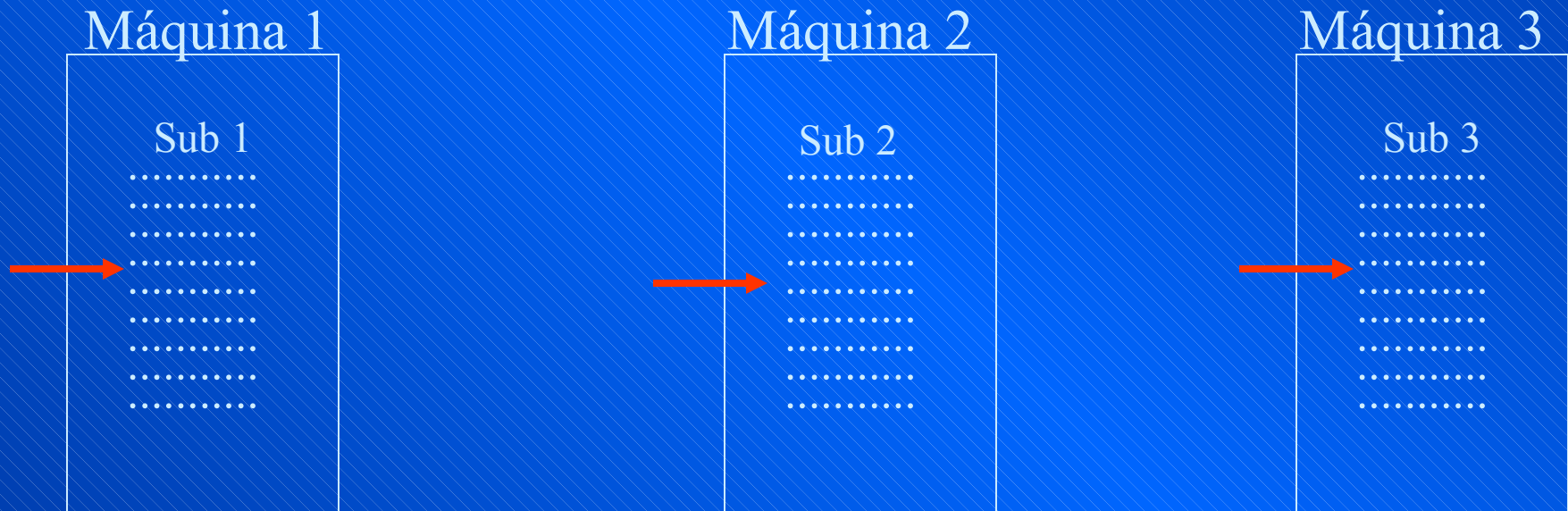
- Mesmo código em todas as máquinas;
- Parte diferente do código sendo executado em cada máquina;



# Mapeamento de Processos

## ■ MPMD

- Códigos distintos nas máquinas.



# Mapeamento de Processos

## ■ SPMD - Vantagens:

- Facilidade no desenvolvimento do algoritmo;
- Balanceamento de carga.

## ■ SPMD - Desvantagens:

- Gasto de memória.

## ■ MPMD - Vantagens:

- Apenas uma “parte” do programa reside em cada processador;

## ■ MPMD - Desvantagens:

- Necessidade de escalonamento.



# Escalonamento de Programas Paralelos

- Atribuir processos a processadores;
  - Influencia consideravelmente o desempenho em programas paralelos;
  - Pode ser realizado tanto no *kernel* (pelo sistema operacional) quanto no espaço do usuário (softwares escalonadores);
- Algumas dificuldades no estudo de escalonamento:
  - Diversas propostas de taxonomias;
  - Divergências quanto a nomenclatura;

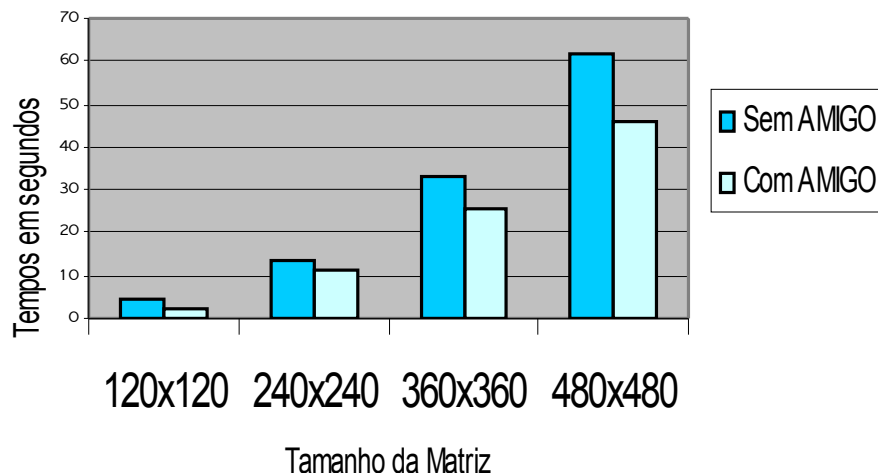
# Escalonamento de Programas Paralelos

- Em aplicações paralelas distribuídas há também a necessidade de gerenciar o escalonamento;
- Utilização de softwares para atuarem na gerência do escalonamento de aplicações;
  - Exemplos: **CONDOR, CODINE, LSF, AMIGO;**
- Balanceamento de carga: Um dos objetivos da atividade de escalonamento;
- Resultados obtidos com escalonamento fornecido pelo ambiente AMIGO...

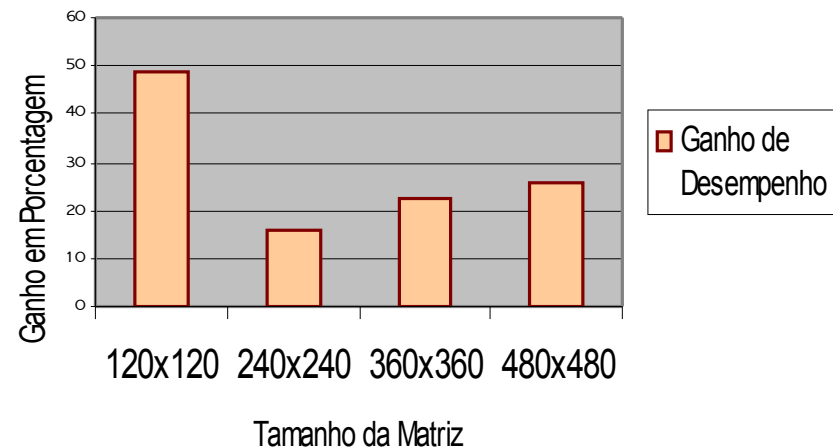
# Escalonamento de Programas Paralelos

- Resultados utilizando uma aplicação LAM-MPI com o escalonamento realizado pelo AMIGO;

Multiplicação de Matrizes



Ganho de Desempenho



- Experimentos realizados em 06 máquinas do tipo PC, formando uma arquitetura MIMD com memória distribuída;

---

**Desenvolvimento do Algoritmo Paralelo** 

**Desenvolvimento do Programa Paralelo** 

**Mapeamento de Processos** 

**Teste e Depuração de Prog. Paralelos** 

# Teste e Depuração de Programas Paralelos

- Todo programa não trivial contém erros;
- Erros = falha (*bug*);
- Teste = Conjunto de atividades que pode ser planejado antecipadamente e realizado sistematicamente;
- Depuração é uma consequência de um teste bem sucedido;
  - De maneira geral, a depuração de um programa paralelo é similar à depuração de um programa seqüencial.

# Teste e Depuração de Programas Paralelos

- Algumas características (inerentes) dificultam o desenvolvimento de programas paralelos:
  - Tipo do Processo;
  - Vários Processadores;
  - Tempo de Comunicação;
  - Tamanho do Sistema;
  - Baixa Visibilidade;
  - Estado Global.

# Teste e Depuração de Programas Paralelos

- Alguns erros que podem ocorrer em programas paralelos:
  - Condição de Disputa;
  - *Deadlock*;
  - *Livelock*;
  - Espera Infinita;
- Não-determinismo;
- Efeito de Intrusão.

# Teste e Depuração de Programas Paralelos

## ■ Testes em Programas Paralelos:

- Muitos algoritmos e métodos de teste para programas sequenciais falham quando utilizados em programas paralelos;
- Outro fator que dificulta o teste de programas paralelos é o não-determinismo;
- Teste de reprodução (*reproducible testing*).

## ■ Alguns testes para programas paralelos:

- Teste de Tarefas;
- Teste Comportamental;
- Teste Intertarefas;
- Teste do Sistema.



---

**Desenvolvimento do Algoritmo Paralelo** 

**Desenvolvimento do Programa Paralelo** 

**Mapeamento de Processos** 

**Teste e Depuração de Prog. Paralelos** 

**Avaliação de Desempenho de Prog.  
Paralelos** 

# Avaliação de Desempenho em Programas Paralelos

- Fatores que devem ser considerados:
  - Como coletar, analisar e entender os dados sobre o desempenho de um sistema paralelo?
  - Que funções devem ser consideradas para representar o desempenho?
  - Como comparar desempenho de programas em diferentes arquiteturas?
- Pode-se utilizar medidas como o *Speedup* e a Eficiência;

# Avaliação de Desempenho em Programas Paralelos

- Deve-se tomar cuidado ao tirar conclusões das medidas de desempenho;
- Exemplo:

|           | T (seq.) | T (p procs) | Speedup |
|-----------|----------|-------------|---------|
| Sistema A | 10       | 2           | 5       |
| Sistema B | 6        | 1,5         | 4       |

## Utilização da Eficiência $p=10$

|           | T(Seq) | T(p procs) | Speedup | Eficiência |
|-----------|--------|------------|---------|------------|
| Sistema A | 10     | 2          | 5       | 50%        |
| Sistema B | 6      | 1,5        | 4       | 40%        |

# Avaliação de Desempenho em Programas Paralelos

Eficiência – comparação com números diferentes de processadores

|           | T(Seq) | T(p procs) | Speedup | Eficiência |
|-----------|--------|------------|---------|------------|
| Sistema A | 10     | 2 (p=10)   | 5       | 50%        |
| Sistema B | 6      | 1,5 (p=5)  | 4       | 80%        |