

Etapas para o Desenvolvimento e Análise de um Programa Paralelo

I - Desenvolvimento de um Algoritmo Paralelo

- Abordagem do Algoritmo
- Identificação do Algoritmo e Divisão dos Processos
- Organização do Trabalho

II - Desenvolvimento do Programa Paralelo

III - Mapeamento de Processos

IV - Teste e Depuração

V - Avaliação de Desempenho

Desenvolvimento do Algoritmo Paralelo

Abordagem do Algoritmo
Identificação do Algoritmo e Divisão dos Processos
Organização do Trabalho

Organização do Trabalho

- Especifica o modelo de concorrência a ser utilizado;
- Depende da arquitetura considerada:
 - SIMD (*Single Instruction Multiple Data*)
 - MIMD (*Multiple Instruction Multiple Data*)
 - **Memória compartilhada**
 - **Memória distribuída**
- Duas Abordagens:
 - Paralelismo por Dado;
 - Paralelismo por Controle.

Organização do Trabalho

- Paralelismo por Dado:
 - Executa as mesmas instruções simultaneamente em um conjunto de dados distintos.
- Paralelismo por Controle:
 - Executa instruções diferentes sobre dados diferentes.
- Exemplo: *Sieve of Eratosthenes*
 - Algoritmo para procurar números primos em um conjunto de números naturais.

Organização do Trabalho

■ Algoritmo *Sieve of Eratosthenes*

- Marcar os múltiplos de um conjunto de números até n ;
- Termina a execução quando for atingir um número maior que \sqrt{n}

$$\sqrt{n}$$



2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
---	---	--------------	--------------	--------------	--------------	--------------	--------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------



2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
---	---	--------------	--------------	--------------	--------------	--------------	--------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------

2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
---	---	--------------	--------------	--------------	--------------	--------------	--------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------

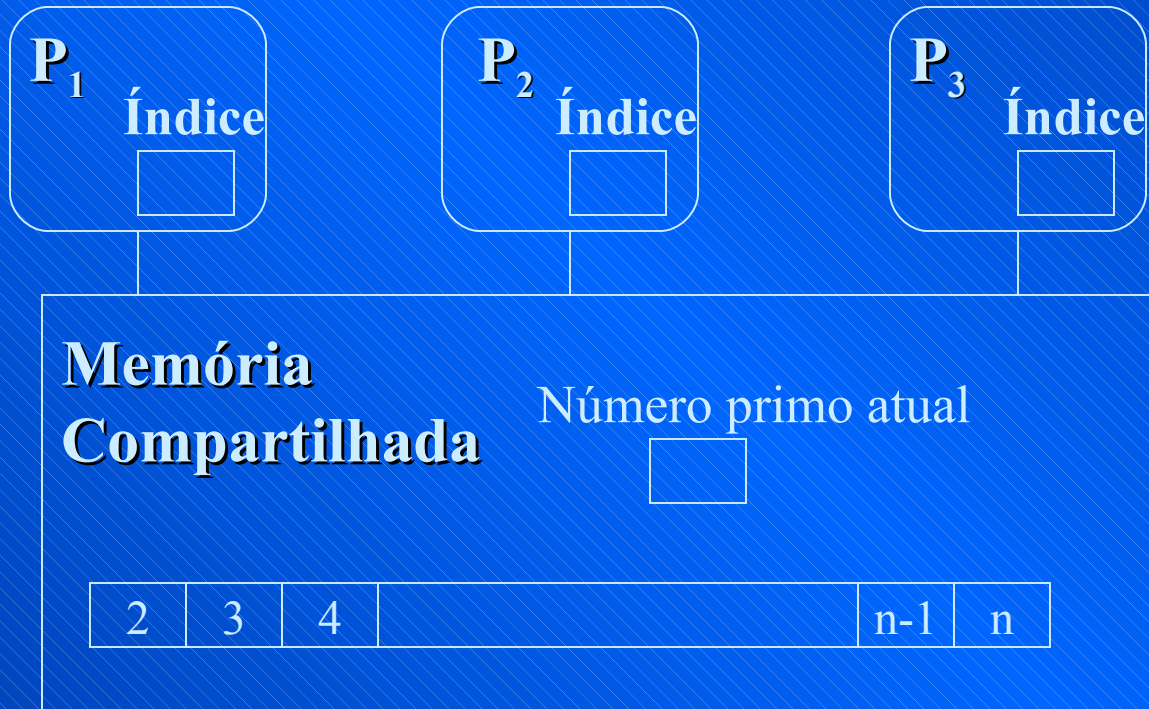


2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
---	---	--------------	--------------	--------------	--------------	--------------	--------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------

2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
---	---	--------------	--------------	--------------	--------------	--------------	--------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------

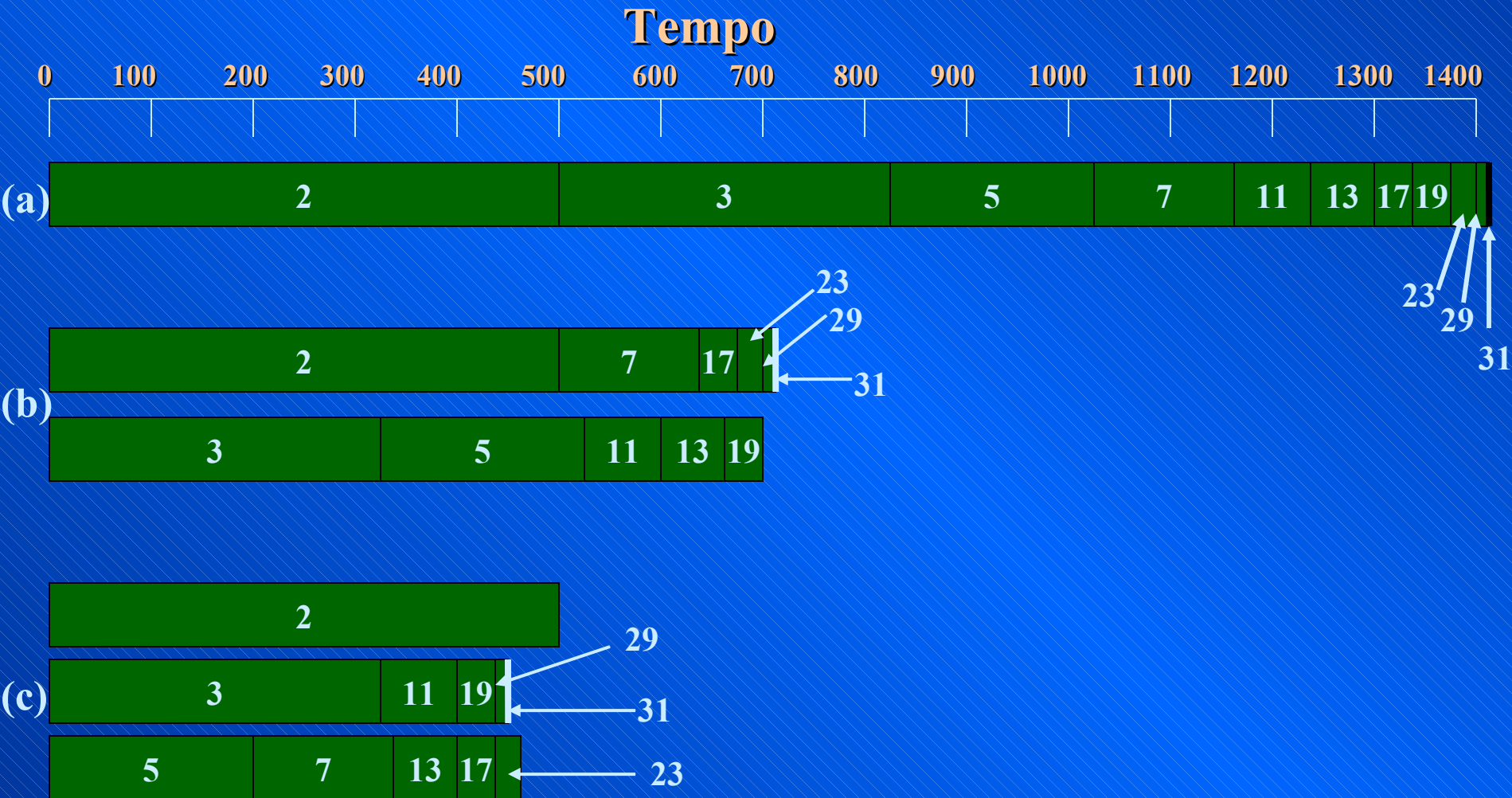
Organização do Trabalho

■ *Silve of Eratosthenes*: Paralelismo por Controle



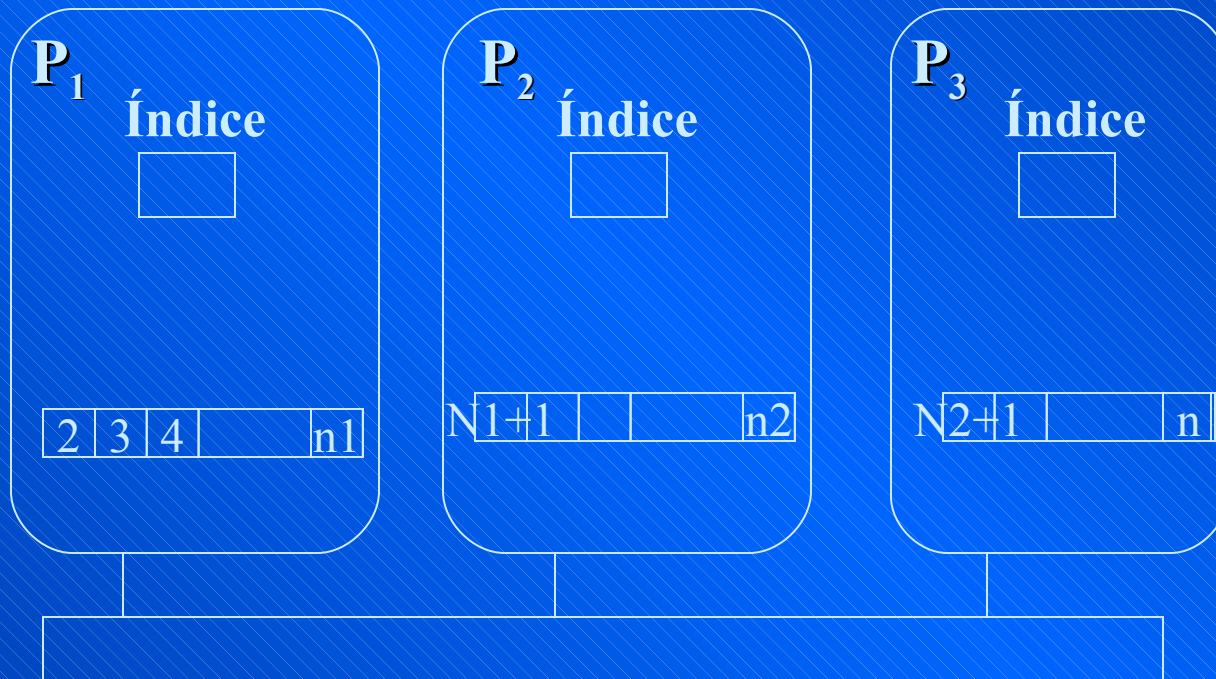
Organização do Trabalho

■ *Sieve of Eratosthenes*: Paralelismo por Controle



Organização do Trabalho

- *Sieve of Eratosthenes*: Paralelismo por Dado



Organização do Trabalho

Modelos de concorrência:

- Dados X Controle
- Depende da arquitetura
- Diversas classificações:
 - Cada autor - uma proposta
 - Diferentes nomes - mesmo modelos

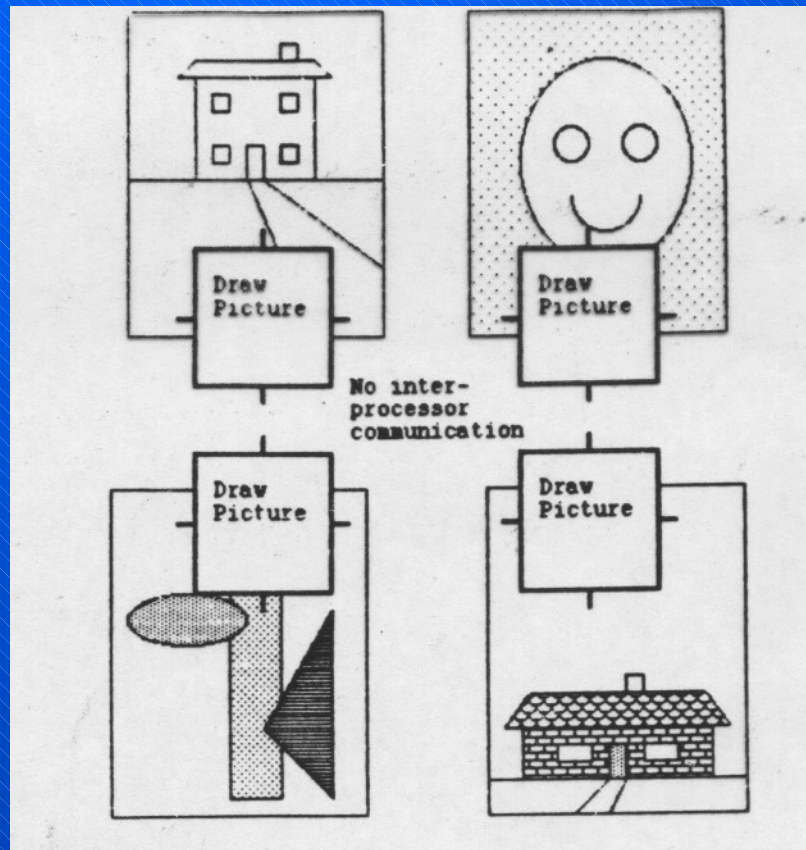
Organização do Trabalho

- Classificação de Hey:
 - Job Farm;
 - Abordagem *Processor Farm*;
 - Abordagem *Pipeline* ou Especialista;
 - Abordagem Geométrica ou Resultado.

Organização do Trabalho

■ Abordagem *Job Farm*:

- Diversos processadores executando diversas tarefas em paralelo sem nenhuma forma de colaboração



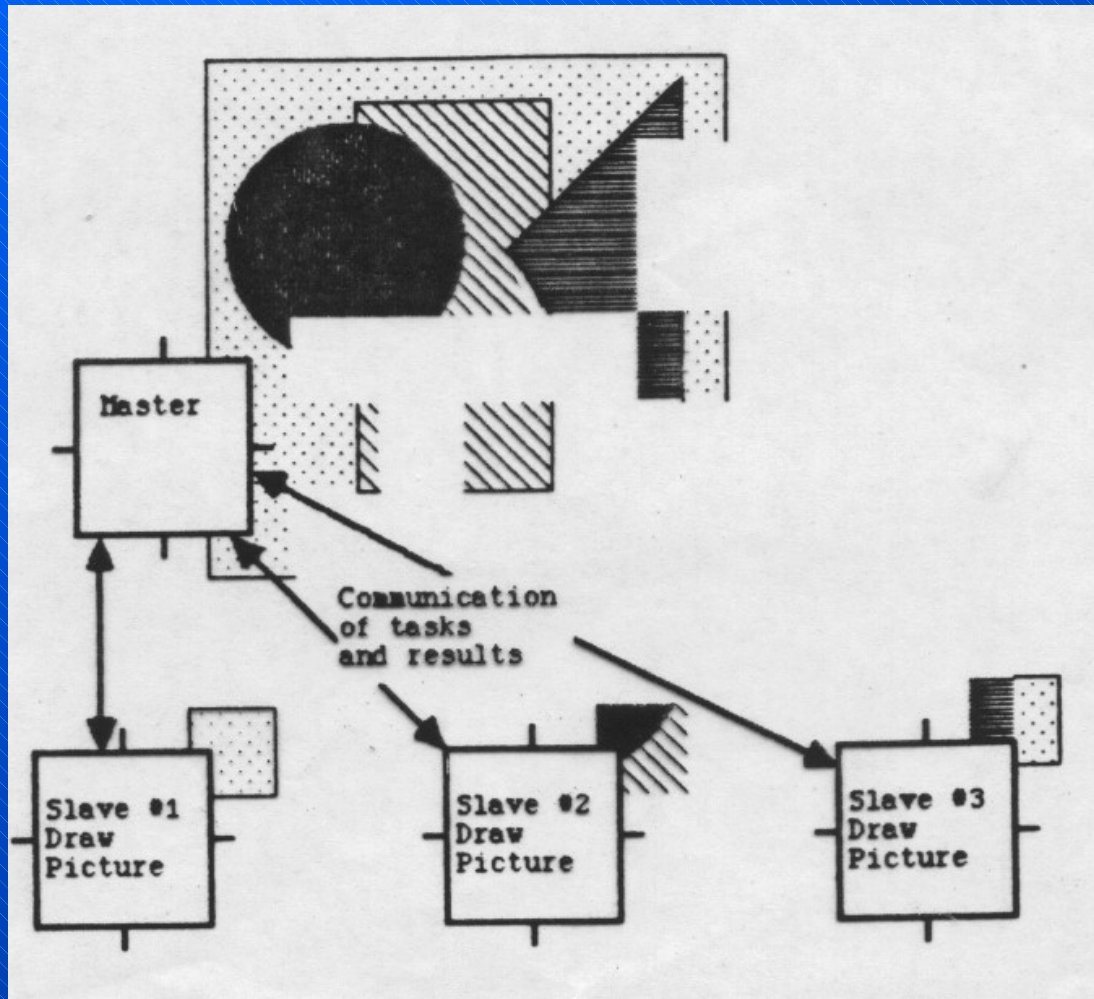
Organização do Trabalho

■ Abordagem *Processor Farm*:

- Cada processador é designado para ajudar no atual item da pauta;
- Geralmente, há um processador mestre que envia o item para cada processador participante;
- Problema: Sobrecarga para o processador mestre;
- Exemplos:
 - Gerar uma tela
 - Multiplicação de Matrizes $A[n][k] * B[k][m]$.

Organização do Trabalho

■ Abordagem *Processor Farm*:



Organização do Trabalho

■ Abordagem *Processor Farm*:

- Mestre: envia aos escravos ociosos a próxima posição da matriz produto a ser calculada;
- Escravo: solicita posição ao mestre, determina produto, envia resposta ao mestre.



– Vantagens:

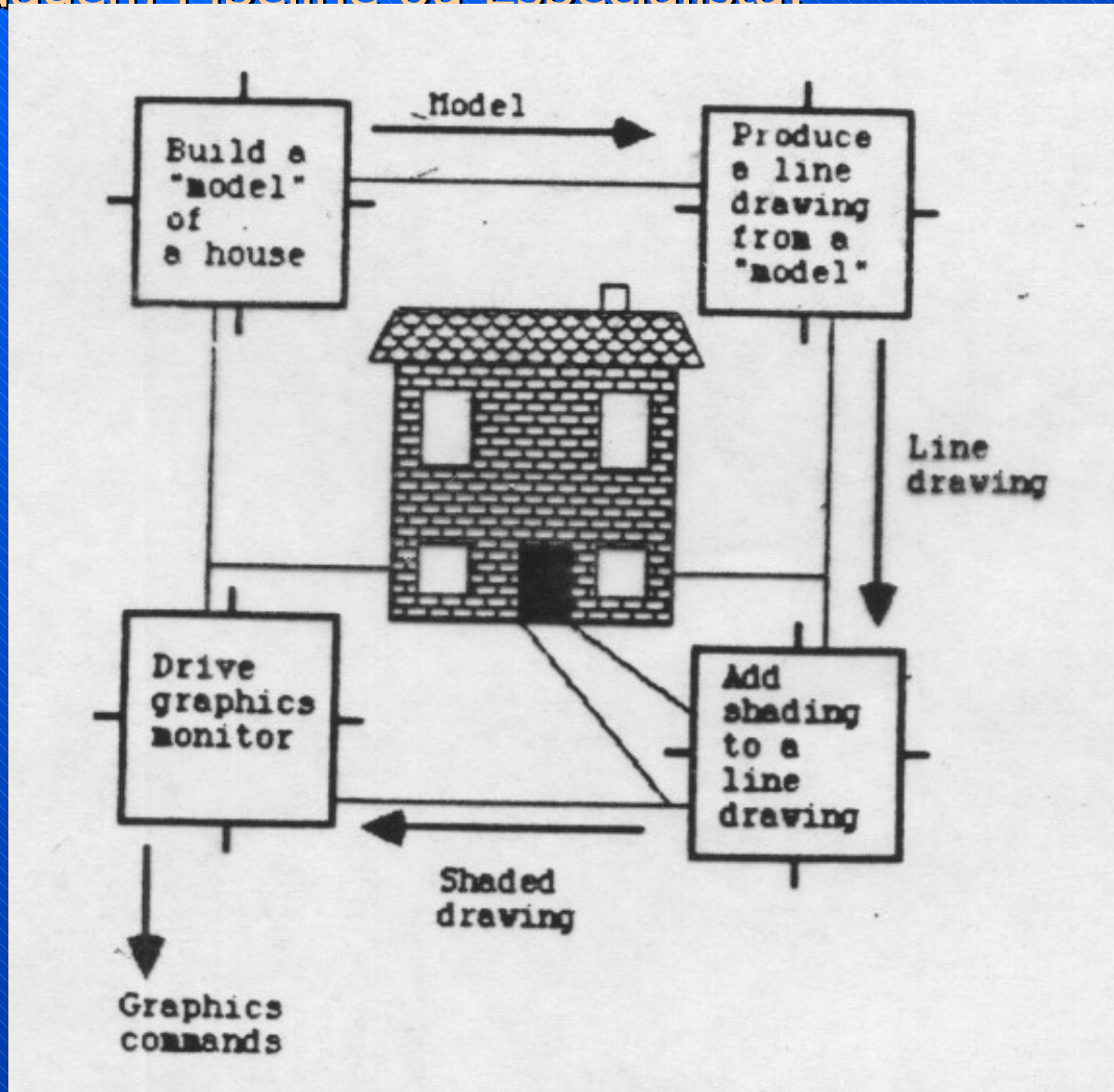
- Flexibilidade quanto ao número de processadores;
- Balanceamento de carga automático.

Organização do Trabalho

- Abordagem Pipeline ou Especialista:
 - Cada processador é responsável por um tipo específico de trabalho;
 - Para uma tarefa → Nenhum paralelismo!
 - Sincronismo é essencial, pois tarefas posteriores dependem das anteriores;
 - Eficiência: Depende do tamanho da tarefa;
 - Exemplo:
 - Gerar uma tela
 - Regra do Trapézio
 - Resolução de N Integrais → N tarefas;

Organização do Trabalho

■ Abordagem Pipeline ou Especialista:



Organização do Trabalho

- Abordagem Pipeline ou Especialista:
 - As integrais da Regra do Trapézio são formadas pela equação:

$$\int_a^b f(x) dx \cong \frac{(b-a) [f(a) + f(b)]}{2}$$

Diagram illustrating the stages of the Trapezoidal Rule formula:

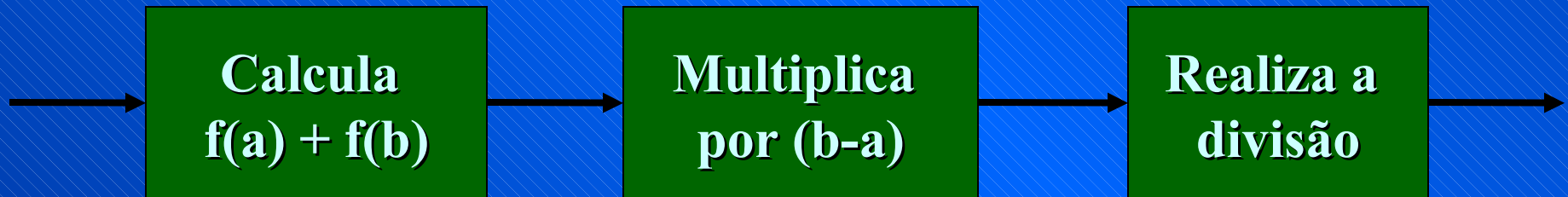
- 2º Estágio**: Points to the entire right-hand side of the equation.
- 1º Estágio**: Points to the term $[f(a) + f(b)]$.
- 3º Estágio**: Points to the entire fraction $\frac{(b-a) [f(a) + f(b)]}{2}$.

Organização do Trabalho

- Abordagem Pipeline ou Especialista:

$$\int_a^b f(x) dx \cong \frac{b-a [f(a) + f(b)]}{2}$$

- Com isso, o *pipeline* pode ser organizado conforme:



- Problemas:

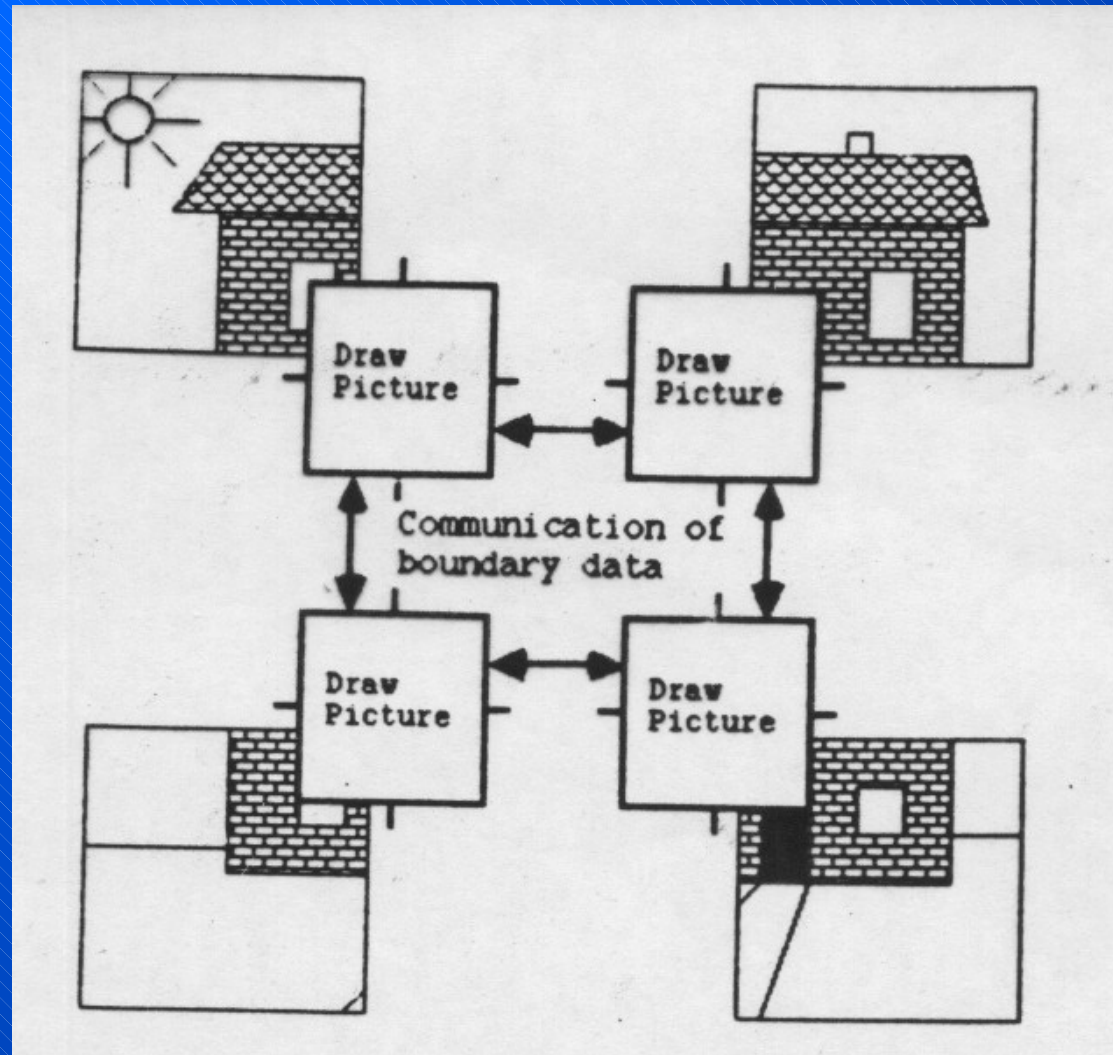
- Pouco flexível;
- Tempo de latência.

Organização do Trabalho

- Abordagem Geométrica ou pelo Resultado:
 - Cada processador é designado para produzir um pedaço do produto final;
 - Deve-se ponderar: Comunicação, Sincronismo e Balanceamento;
 - Exemplo:
 - Geração de uma Tela
 - Multiplicação das Matrizes: $A[n][k] * B[k][m]$
 - Cada processador pode ser responsável por um elemento da matriz resultante;
 - Os valores utilizados na multiplicação são enviados para outros processadores.

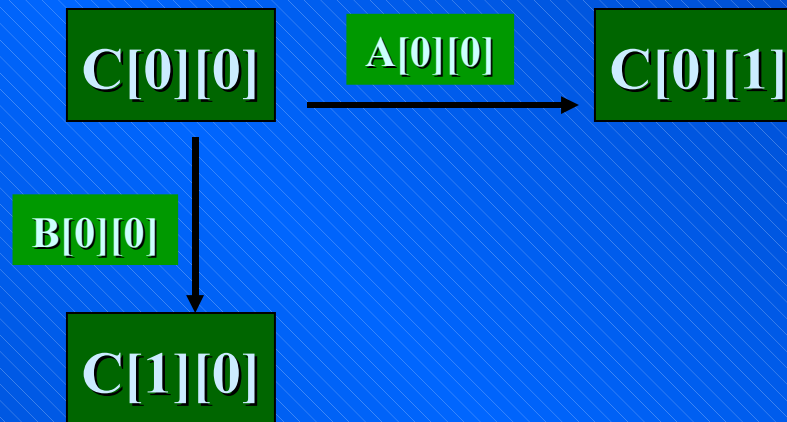
Organização do Trabalho

- Abordagem Geométrica ou pelo Resultado:



Organização do Trabalho

- Abordagem Geométrica ou pelo Resultado:



- Obtenção de um alto grau de paralelismo;
- São necessários NxM processadores;
- Sobrecarga na comunicação;
- Granulação fina.

Organização do Trabalho

- Classificação de Kung:
 - Computação Local
 - Pipeline
 - Pipeline Multifunção
 - Ring
 - Dividir e Conquistar
 - Filas de Tarefas
 - Particionamento de domínio
 - Recursiva

Problema - Muitos itens sem hierarquia

Subdivisões da classificação de Hey

Organização do Trabalho

- Subdivisões da classificação de Hey:
 - Pipeline e Pipeline Multifunção - Pipeline ou Especialista
 - Filas de Tarefas e Computação Local - Mestre e Escravo
 - Particionamento de domínio - Geométrico

Organização do Trabalho

■ Classificação de Kung:

- Computação Local
- Pipeline
- Pipeline Multifunção
- Ring

— **Dividir e Conquistar**

- Filas de Tarefas
- Particionamento de domínio
- Recursiva

Organização do Trabalho

- **Dividir e Conquistar**
- Repete Recursivamente:
 - dividir o problema
 - resolve em paralelo
- Muito Utilizado
- Sistemas lineares, computação gráfica, simulação
- Ex: Determinação do valor máximo ou mínimo de um vetor