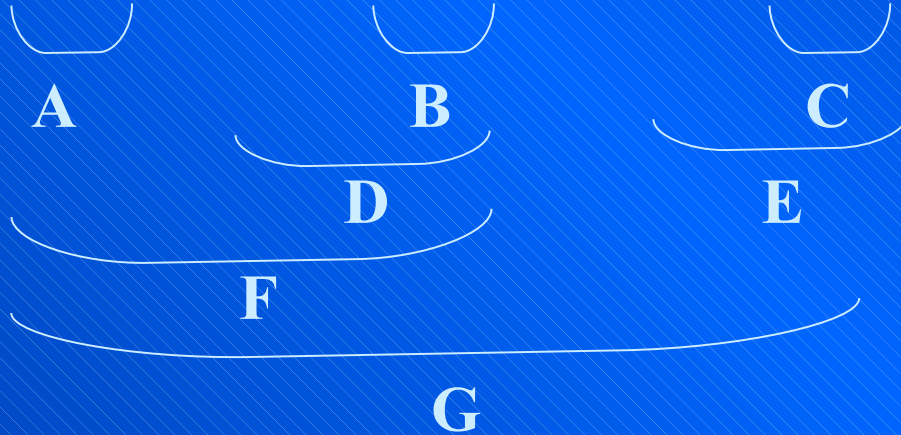


Identificação do paralelismo e divisão dos processos

■ $(a * b + c * d^2) * (g + f * h)$



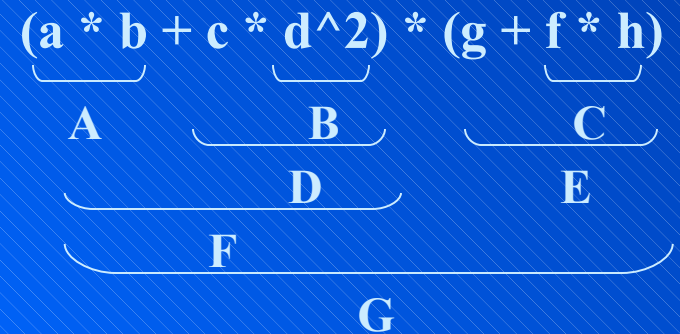
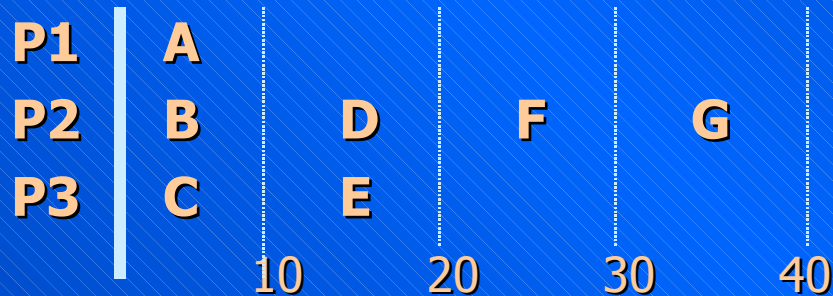
■ Considerando Toperação = 10 ut, tem-se que:

$$T_{seq} = 70 \text{ ut}$$

$$T_{par} = ?$$

Identificação do paralelismo e divisão dos processos

- Utilizando três processadores:



- Ao final da execução, verifica-se que:

$T_{par} = 40 \text{ ut}$

(sem considerar comunicação!)

- Calculando o *speedup* e a eficiência...

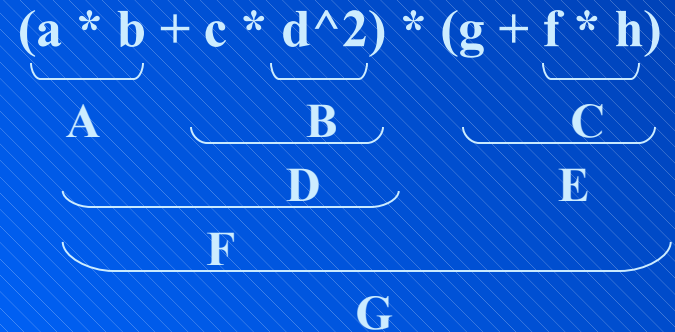
$Sp = 70 / 40 = 1.75$

$Ef = 1.75 / 3 = 58\%$

Identificação do paralelismo e divisão dos processos

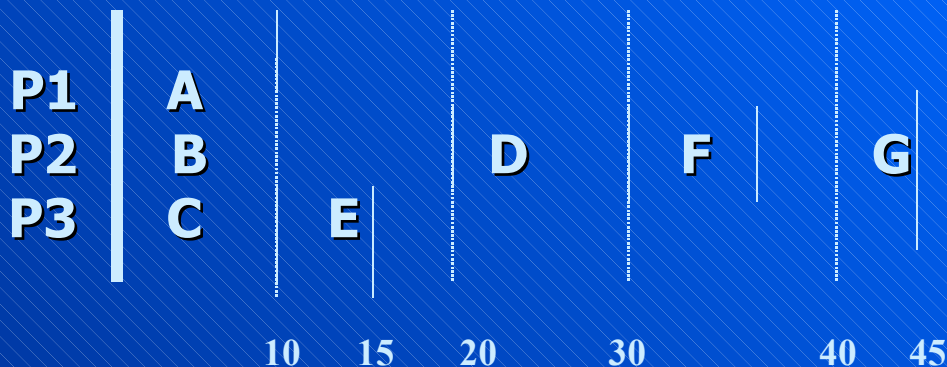
■ Agora, considerando:

- $T^{\wedge} = 20 \text{ ut}$
- $T^* = 10 \text{ ut}$
- $T^+ = 5 \text{ ut}$



■ $T_{\text{seq}} = 70 \text{ ut}$

■ $T_{\text{par}} = 45 \text{ ut}$

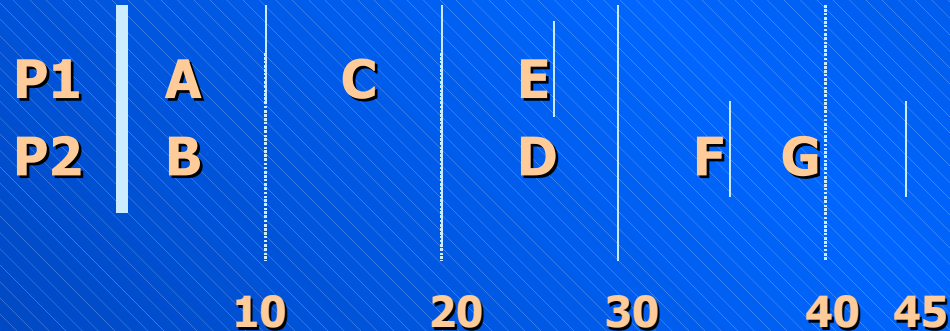


$$Sp = 70 / 45 = 1,55$$

$$Ef = 1,55 / 3 = 52\%$$

Identificação do paralelismo e divisão dos processos

- Analisando o diagrama, nota-se que A pode ser executado por P3 ou C e E por P1



O tempo de execução
ainda continua sendo 45 ut

- Assim:

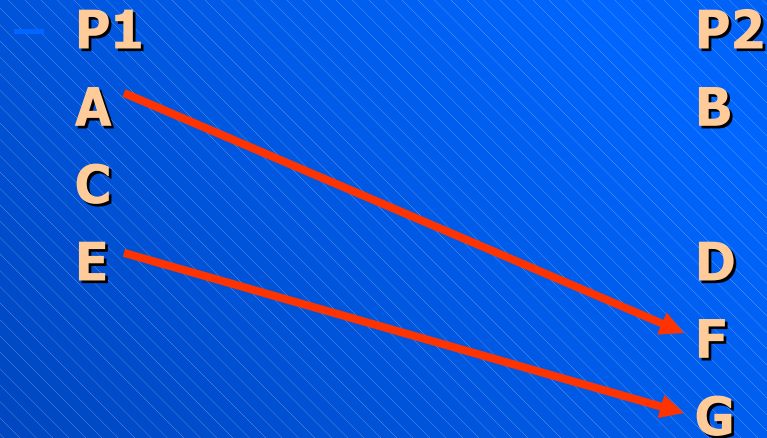
$$S_p = 70/45 = 1,55$$

$$E_f = 1,55 / 2 = 77\%$$



Identificação do paralelismo e divisão dos processos

■ Considerando comunicação...



$$\underbrace{(a * b + c * d^2)}_A * \underbrace{(g + f * h)}_E$$

$\underbrace{\hspace{10em}}_D$

$\underbrace{\hspace{15em}}_F$

$\underbrace{\hspace{20em}}_G$

■ Assim: $T_{par} = 45 + 2 * T_{comunicação}$

Comunicação = sobrecarga!

Abordagem do Algoritmo

■ Exemplo 2 - Soma dos Elementos de um Vetor

$$- a_1 + a_2 + a_3 + \dots a_{(n/2)-1} + a_{(n/2)} + \dots + a_{n-1} + a_n$$

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

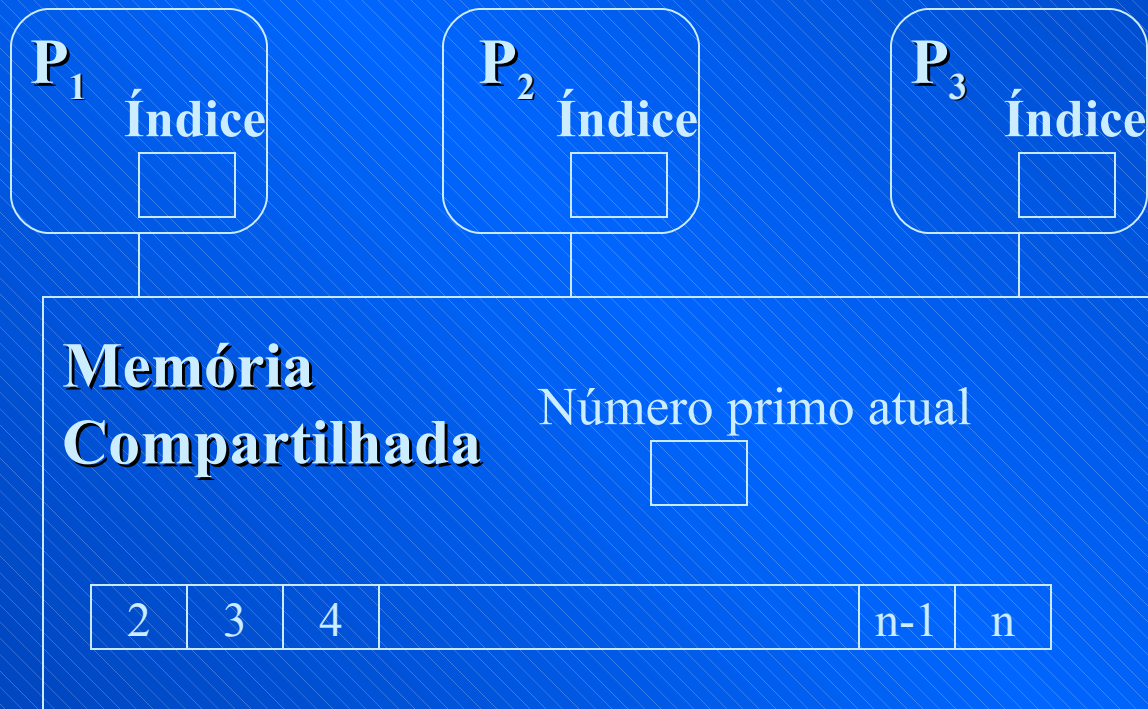
■ Algoritmo *Sieve of Eratosthenes*

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



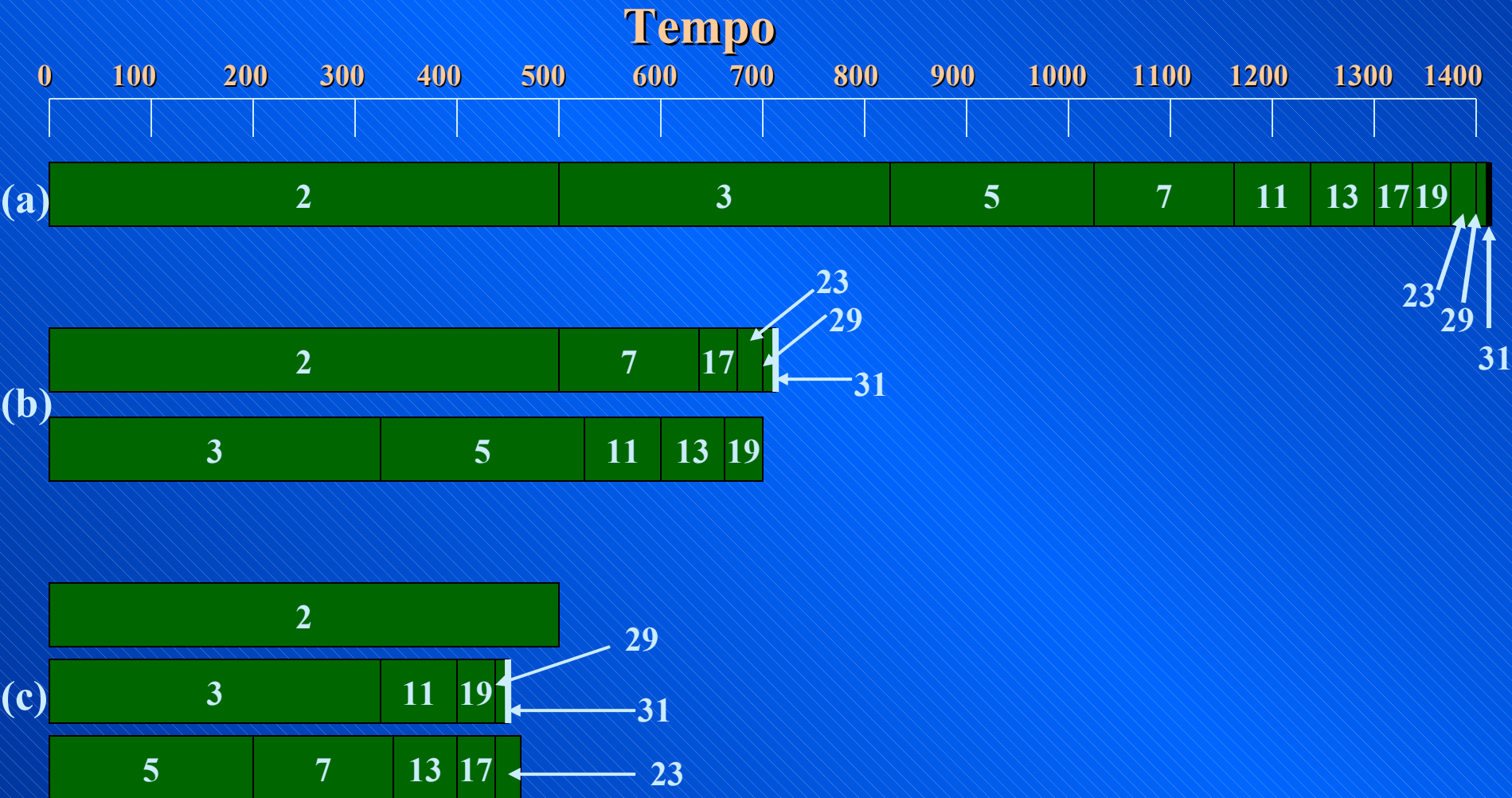
Organização do Trabalho

- *Sieve of Eratosthenes*: Paralelismo por Controle



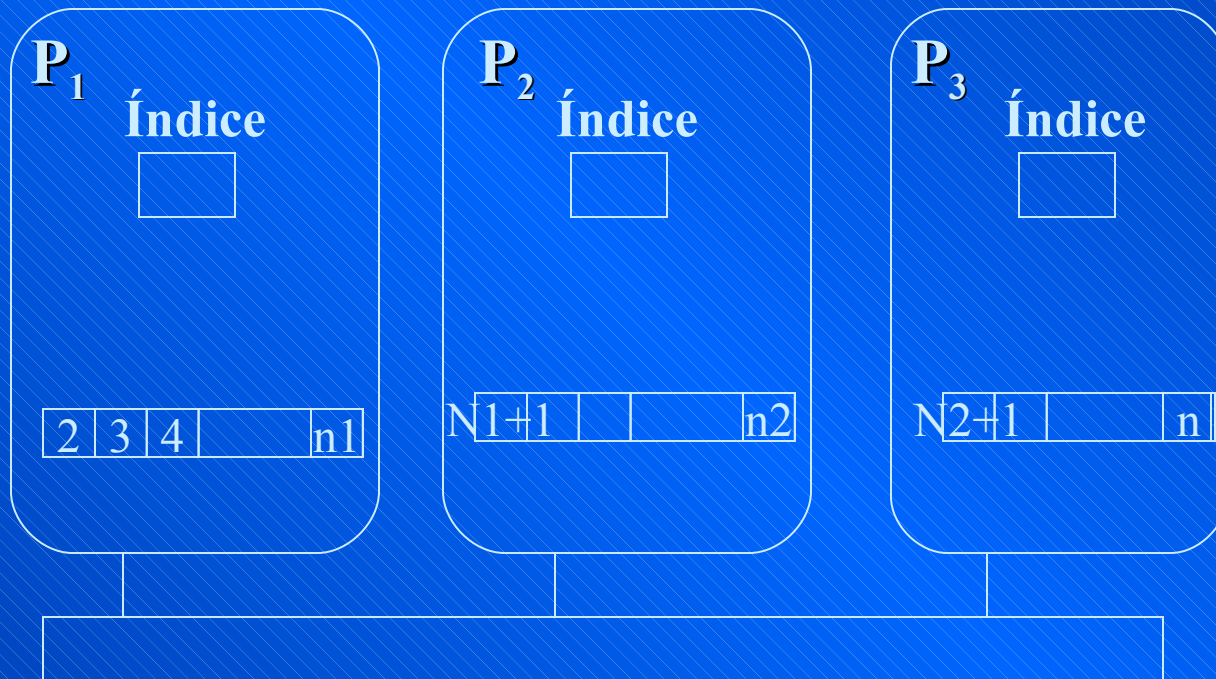
Organização do Trabalho

■ *Sieve of Eratosthenes*: Paralelismo por Controle



Organização do Trabalho

- *Sieve of Eratosthenes*: Paralelismo por Dado



Organização do Trabalho

- *Silve of Eratosthenes*: Paralelismo por Dado

Todos Processadores executam as mesmas instruções em conjunto de dados diferentes

Balanceamento facilitado

Exercício

$$X_i = \sum a_i * c_i + d_{i-1} * e_{i-1}, i = 1 \text{ até } n$$

1. Determinar:

Algoritmo que explore paralelismo por dados

Algoritmo que explore paralelismo por controle

2. Para cada um deles

Determinar os Processos

Considerar $T(*) = 2$; $T(+) = 1$; $T(\text{comunicação}) = 4$

Estimar Speedup

Organização do Trabalho

- Paralelismo por Dado e Controle podem ser divididos em:
 - Abordagem *Processor Farm*;
 - Abordagem *Pipeline* ou Especialista;
 - Abordagem Geométrica ou Resultado.

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;
- Exemplo: Multiplicação de Matrizes $A[n][k] * B[k][m]$.

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: Regra do Trapézio
 - Resolução de N Integrais → N tarefas;

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:

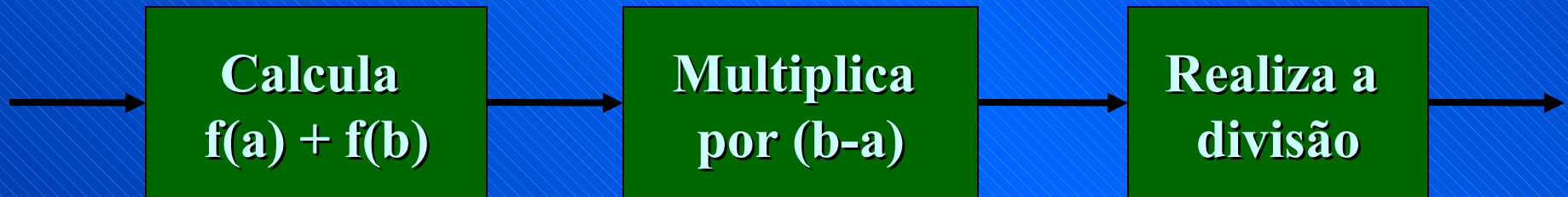
- 2º Estágio**: Points to the entire right-hand side of the equation.
- 1º Estágio**: Points to the term $[f(a) + f(b)]$.
- 3º Estágio**: Points to the entire fraction $\frac{(b-a) [f(a) + f(b)]}{2}$.

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:

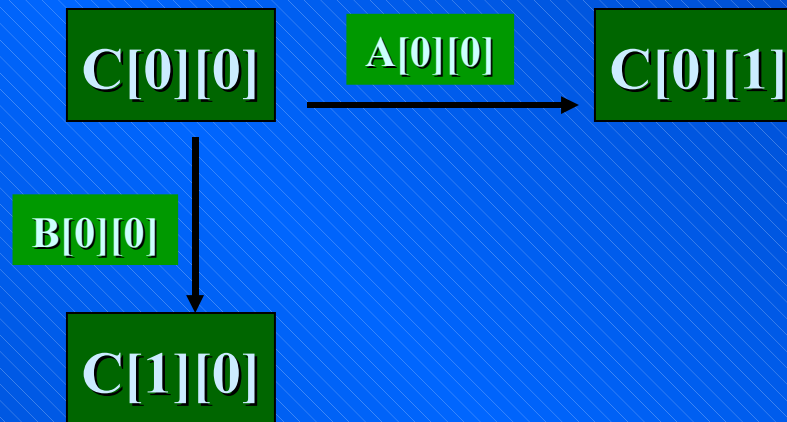
- Pouco flexível;
- Tempo de latência.

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: 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:



- Obtenção de um alto grau de paralelismo;
- São necessários $N \times M$ processadores;
- Sobrecarga na comunicação;
- Granulação fina.