

Computação Concorrente (DCC/UFRJ)

Módulo 1 - Semana 2 - Aula 1: Projeto e avaliação de uma aplicação concorrente (visando ganho de desempenho)

Prof. Silvana Rossetto

Dezembro 2020

Dada uma implementação sequencial para o problema de **multiplicação de matrizes** na forma $A * B = C$, sendo A , B e C matrizes quadradas de dimensão $N \times N$, projetar, implementar e avaliar uma versão concorrente

Considerando uma solução sequencial existente:

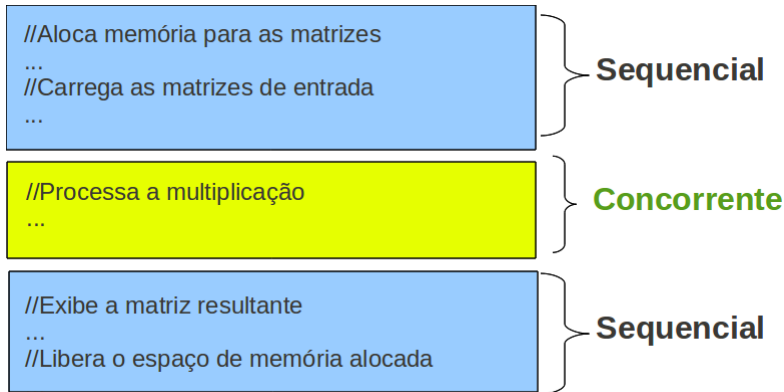
- 1 PASSO 1: análise do algoritmo sequencial
- 2 PASSO 2: cálculo do ganho de desempenho previsto
- 3 PASSO 3: projeto e implementação do algoritmo concorrente
- 4 PASSO 4: avaliação dos resultados

Algoritmo sequencial para multiplicação de matrizes

```
//Aloca memória para as matrizes
...
//Carrega as matrizes de entrada
...
//Processa a multiplicação
...
//Exibe a matriz resultante
...
//Libera o espaço de memória alocada
...
```

PASSO 1: análise do algoritmo básico

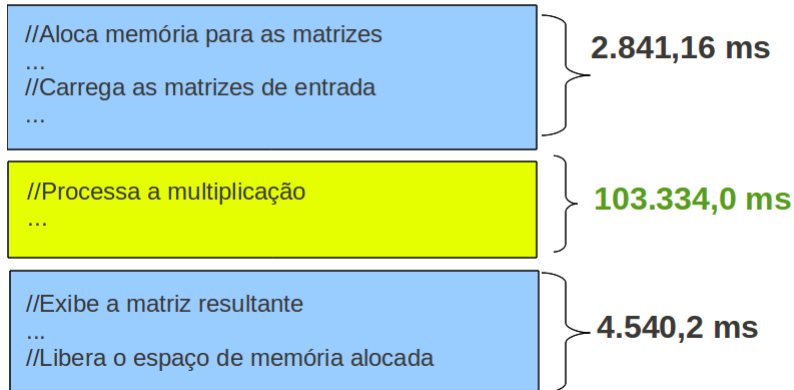
Quais são as etapas principais do programa?



PASSO 1: análise do algoritmo básico

Qual é o tempo de processamento de cada uma das etapas principais do programa?

Matrizes de entrada: A2048x2048.txt



Como estimar o ganho de velocidade de execução de um programa sequencial usando vários processadores?

Tempo sequencial

- $T_{\text{sequencial}}$: tempo total de execução do programa sequencial

Tempo concorrente

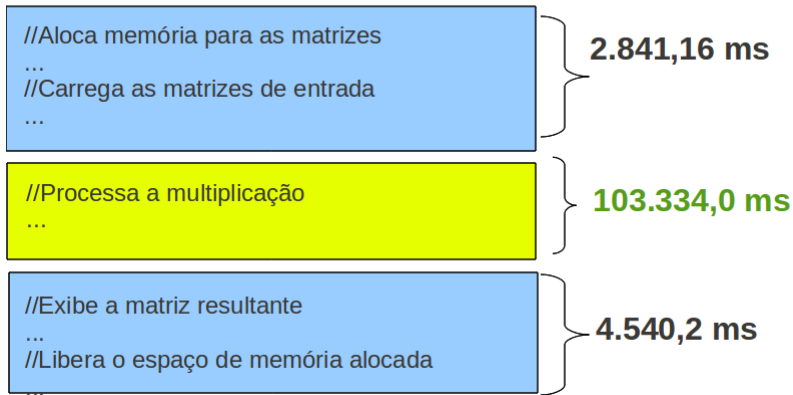
- t_s : tempo da parte sequencial do programa (que não será dividida entre threads)
- $t_p(P)$: tempo da parte paralela do programa usando P processadores
- $T(P)$: tempo total de execução previsto do programa usando P processadores ($t_s + t_p(P)$)

O ganho de velocidade da execução é dado por:

$$T_{sequencial} / t_s + t_p(P)$$

Voltando ao problema de multiplicação de matrizes

Matrizes de entrada: A2048x2048.txt

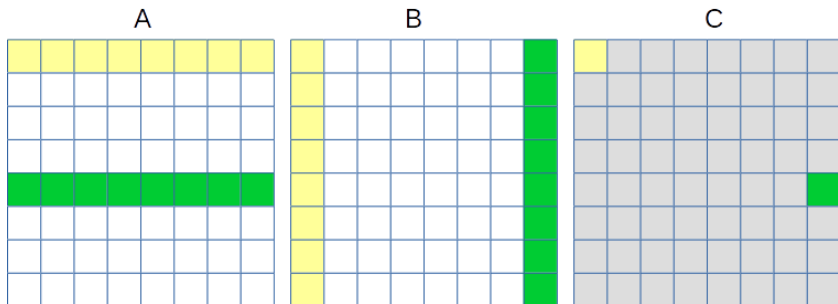


Qual será o ganho estimado de execução em uma máquina com 2 processadores?

Problema de multiplicação de matrizes

- O tempo total sequencial ($T_{sequencial}$) é igual a **110.715,36**
- Com 2-processadores, a parte paralela gastará $p/n = 103334/2 = \mathbf{51.667,0}$
- A parte sequencial (entrada + saída) continuará igual a **7.381,36**
- O ganho estimado será $S = \frac{110715.36}{(7381.36+51667)} = \mathbf{1,87}$

Divisão da parte paralela em subtarefas



Divisão da parte paralela em subtarefas

uma thread por elemento

T1	T2	T3	...				
						...	T64

uma thread por linha

							T1
							T2
							T3
							T4
							T5
							T6
							T7
							T8

uma thread por linha
(consecutivo)

							T1
							T2
							T3
							T4

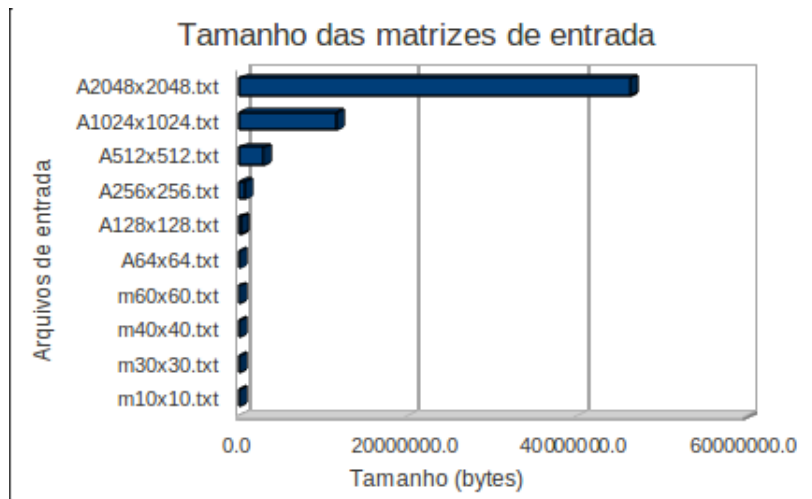
uma thread por linha
(alternado)

							T1
							T2
							T3
							T4
							T1
							T2
							T3
							T4

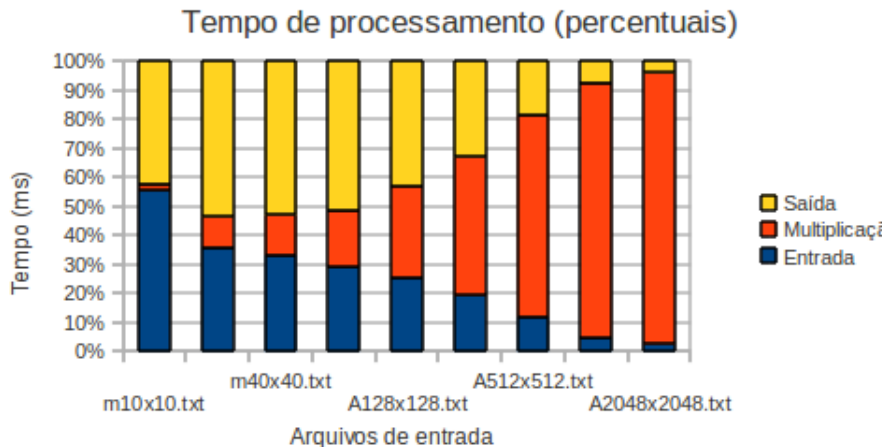
Alternativa escolhida para **divisão das tarefas entre as threads**:

- cada thread calcula um subconjunto de linhas (alternadas) da matriz de saída
- o número de threads pode ser determinado pelo número de processadores da máquina, independente da dimensão das matrizes
- não é necessário que o número de linhas seja divisível pelo número de threads

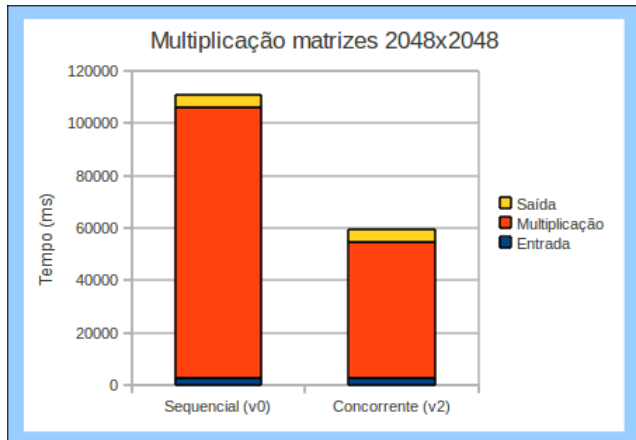
PASSO 4: Avaliação do ganho de desempenho



PASSO 4: Avaliação do ganho de desempenho



PASSO 4: Avaliação do ganho de desempenho



Ganho real: $110715.36 / 59283.36 = 1,867$

A aplicação concorrente **pode gastar mais tempo de execução** do que a aplicação sequencial dependendo:

- do **projeto da solução**
- do **conjunto de dados de entrada**
- do **ambiente computacional disponível**