

Relatório do Laboratório 3 de Programação Concorrente

Aluno: João David Jotta Mendonça Estorque
Dre: 121155853

Implementei a função concorrente me baseando nos códigos dados pela professora, percebi que era possível usar o id como uma posição na matriz de saída, ou seja, cada passo da thread era a multiplicação de uma linha por uma coluna.

Levei bastante tempo implementando, foi um laboratório realmente difícil!

Para testar, criei uma matriz 10x10 e uma 10x8 e multipliquei elas com os dois códigos, ao usar o código diff <saida1><saida2>, depois de várias adaptações no código do concorrente, consegui um código funcional que o diff retornasse nada.

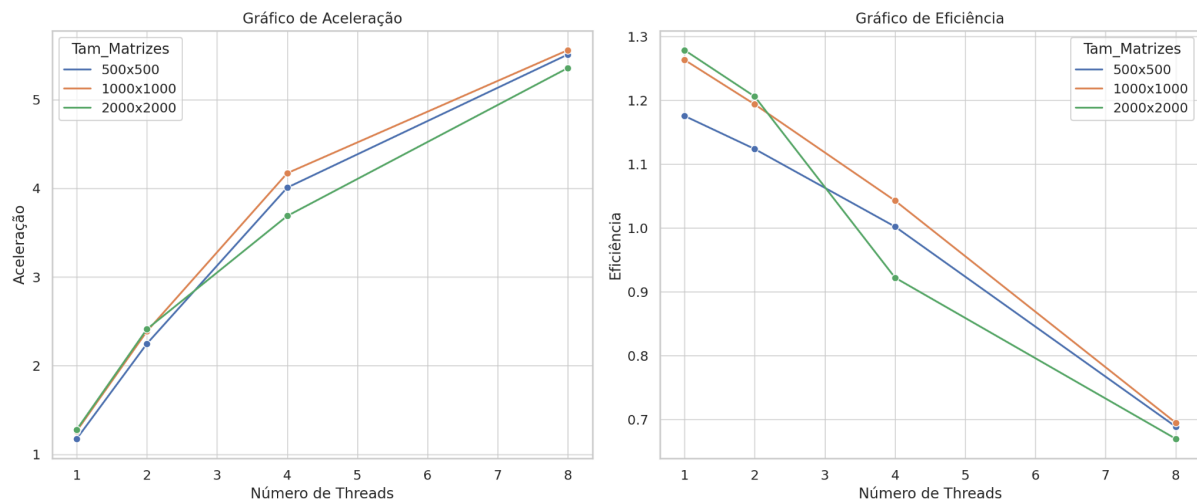
Agora vamos colher as informações e analisar tais dados.

Configuração da máquina:
cpu cores : 14

Tabela com informações:

	A	B	C	D	E	F
1	Programa	Tam_Matrizes	Tempo de Inicialização	Tempo de Multiplicação	Tempo de Finalização	Quant_Threads
2	Sequencial	500x500	0.001218	0.260888	0.000132	0
3	Sequencial	1000x1000	0.005434	2.229133	0.000437	0
4	Sequencial	2000x2000	0.018293	21.782771	0.001496	0
5	Concorrente	500x500	0.002359	0.221926	0.000079	1
6	Concorrente	500x500	0.002318	0.116079	0.000186	2
7	Concorrente	500x500	0.002406	0.065086	0.000133	4
8	Concorrente	500x500	0.002393	0.047350	0.000070	8
9	Concorrente	1000x1000	0.008838	1.764415	0.000416	1
10	Concorrente	1000x1000	0.005558	0.933706	0.000287	2
11	Concorrente	1000x1000	0.007023	0.534439	0.000280	4
12	Concorrente	1000x1000	0.009113	0.401010	0.000422	8
13	Concorrente	2000x2000	0.019108	17.035172	0.000876	1
14	Concorrente	2000x2000	0.017298	9.031286	0.000992	2
15	Concorrente	2000x2000	0.020285	5.905479	0.001601	4
16	Concorrente	2000x2000	0.022013	4.065897	0.001350	8

Gráficos:



Gerar essas informações foi um trabalho realmente difícil, como temos muitos valores na tabela, e queremos rodar 5 vezes para tirar uma média, isso toma um tempo enorme.

Utilizei o Chat Gpt para criar o código que lê a tabela csv e imprime os gráficos, como era uma parte secundária do laboratório, não vi problema em usar a ferramenta para tal.

Meu código sequencial perdeu em todos os valores para o concorrente, até no concorrente com 1 thread.

não sei o motivo, talvez eu tenha escrito um código mal otimizado?

As formas de multiplicação dos códigos são diferentes também, um usa 3 for aninhados (sequencial) e outro usa o método por id, talvez por id seja mais eficiente?

Enfim, não consegui tirar conclusões de tal informação.

Sobre os gráficos, a aceleração foi como prevista, crescente mas com a derivada diminuindo.

E a Eficiência primeiramente surpreendeu com um valor maior que 1 inicialmente, por motivos desconhecidos que discuti lá em cima. Ela teve uma queda linear, com a eficiência de cada thread 70% em média no final, o que ainda é um bom valor.

Acredito que o valor ideal de eficiência é entre 85% e 100%, não sei bem como otimizar meu código para chegar em tal métrica.