Aluno: Francisco Italo Silva do Nascimento
Matrícula: 20190000680

Aluno: Epitácio Bessa da Silva
Matrícula: 20210071827

Introdução

O propósito deste trabalho é analisar e comparar duas soluções para um mesmo problema: uma solução no paradigma sequencial e outra concorrente. O problema é a multiplicação de matrizes quadradas. Feito as implementações serão realizadas as comparações de performance entre ambas.

Metodologia

As soluções foram implementadas utilizando a linguagem de programação Python, versão 3.9. Foi adotada esta linguagem por ser mais familiar aos desenvolvedores. Para medição do tempo de execução das soluções, foi utilizado a biblioteca *timeit,* por ser simples e permitir a realização de N execuções e o formato de tempo de execução. Foram realizadas 20 execuções para cada solução e o tempo adotado foi em segundos. E para exportar os resultados no formato CSV foi utilizado a biblioteca *pandas* por também ser simples.

Os resultados são exportados para um arquivo CSV. O nome do arquivo possui o seguinte formato: {dimensões da matrizes}{tipo de solução (“S” ou “C”)}.csv. Exemplo, 2x2C.csv.

Durante o desenvolvimento dos algoritmos concorrentes, foi pensado de duas formas. Uma forma foi criando uma thread a cada linha/coluna das matrizes que fossem sendo lida. A outra foi limitar a quantidade de threads e quanto esse limite fosse alcançado teria que esperar alguma delas terminar a execução para outra poder começar. Porém, durante os testes, as diferenças entre essas duas formas foram irrelevantes. Sendo assim, a forma adotada para representação deste trabalho foi a primeira: criar uma thread para cada multiplicação.

Os testes realizados foram feitos numa máquina com os seguintes requisitos técnicos: processador i5-8250U; CPU 1.60Ghz; memória RAM 20Gb.

Resultados

Segue abaixo todas as tabelas com os resultados obtidos nas execuções de ambos os algoritmos. As tabelas estão separadas de acordo com cada dimensão das matrizes. A matriz 2048x2048 não foi possível a obtenção dos resultados, devido ao tamanho das dimensões, o código ficou executando por mais de 5 minutos sem saída.

- 2x2

	Sequencial	Concorrente
Máximo	0.039452199009247124	0.0046059539999987464
Mínimo	0.0004153600020799786	0.0008282129997496668
Médio	0.002463548950618133	0.0018727620499248588
Desvio Padrão	0.0084861305902	0.000875548302505

- 4x4

	Sequencial	Concorrente
Máximo	0.14522098299994468	0.006534076000207278
Mínimo	0.0010646519999681914	0.0008192440000129864
Médio	0.008529017150021901	0.002941744899999321
Desvio Padrão	0.0313600675377	0.00167996006285

- 8x8

	Sequencial	Concorrente
Máximo	0.12051688699284568	0.02738600698648952
Mínimo	0.0019876429869327694	0.002262800990138203
Médio	0.008034892243449577	0.00440988799527986
Desvio Padrão	0.0258052479819	0.00531714169353

- 16x16

	Sequencial	Concorrente
Máximo	0.005442739000045549	0.02535976600029244
Mínimo	0.00207452699987698	0.010758158000044205

Médio	0.0027526191500328425	0.012651339649983129
Desvio Padrão	0.000797774273785	0.00339501222193

- 32x32

	Sequencial	Concorrente
Máximo	0.052332010000100126	0.06524954699989394
Mínimo	0.01376185599974633	0.020117935000143916
Médio	0.016058250050014065	0.043548129000032534
Desvio Padrão	0.00833495945868	0.0195466581557

- 64x64

	Sequencial	Concorrente
Máximo	0.5627616860001581	0.3585999189999711
Mínimo	0.23393898099993748	0.1191731260000779
Médio	0.3142806432000043	0.1873501325999996
Desvio Padrão	0.0810106817105	0.0681395599777

- 128x128

	Sequencial	Concorrente
Máximo	2.776382824999928	2.3633025819999602
Mínimo	0.9304002879998734	0.7998439230000258
Médio	2.0616934266000273	1.1509737347500049
Desvio Padrão	0.359295615574	0.416388495912

- 256x256

	Sequencial	Concorrente
Máximo	19.450714663000326	19.425723946999824
Mínimo	12.645973329999833	8.66337308600032
Médio	16.751092631899997	12.813591695050059
Desvio Padrão	1.93486624952	3.12771954707

- 512x512

	Sequencial	Concorrente
Máximo	143.01906459100064	142.71017495600063
Mínimo	113.27459677200022	94.62757114400029
Médio	128.6331457611001	126.89006876725003
Desvio Padrão	9.46575261381	12.8570788537

- 1024x1024

	Sequencial	Concorrente
Máximo	1334.9197957299984	1326.432359650993
Mínimo	718.6097670119998	683.2641035109991
Médio	948.0974675604502	963.7704502475993
Desvio Padrão	177.845736816	176.64090944

Comparação

Dados todas as tabelas com os resultados obtidos. Segue abaixo alguns gráficos para melhor comparação entre as duas implementações.

Na Figura 1 pode-se notar que os resultados do algoritmo concorrente se saiu melhor em quase todas as dimensões das matrizes, exceto com 16x16 e 32x32. Já na Figura 2 mostra os resultados para as dimensões de 128x128 até 1024x1024. O conjunto das dimensões foram separados devido os resultados terem sido muito maiores que as outras. Note que os resultados se saíram muito próximos para essas dimensões. Mostrando que quase não houve diferença entre os resultados.

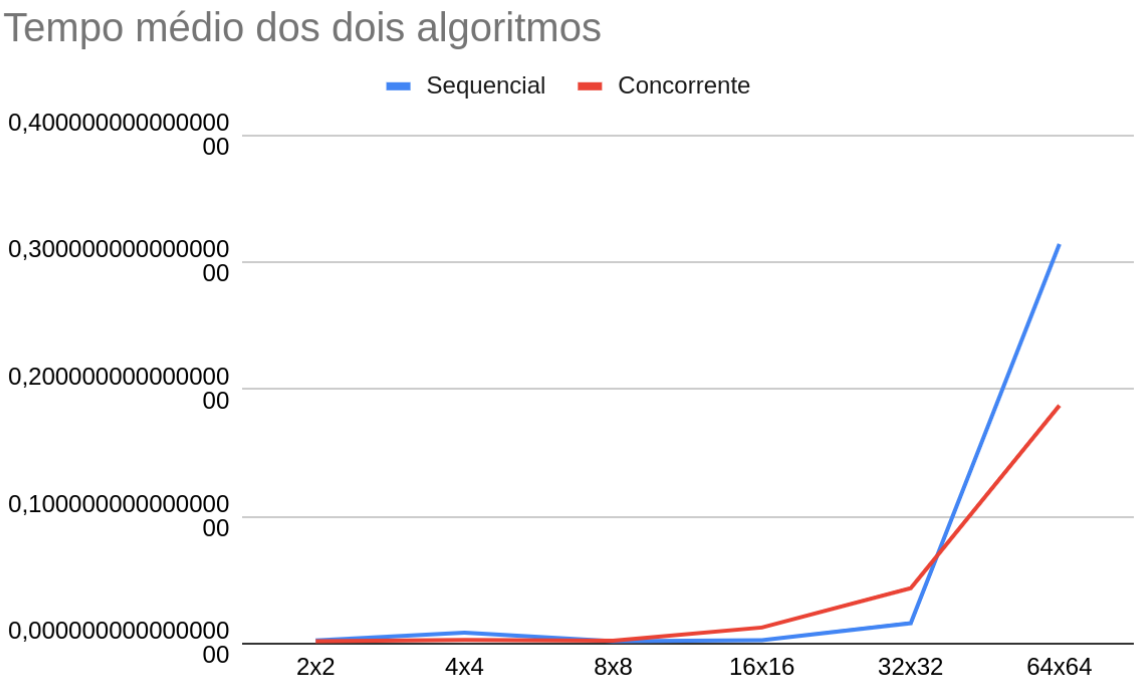

Figura 1: Gráfico de comparação do tempo máximo das dimensões 2x2 até 64x64

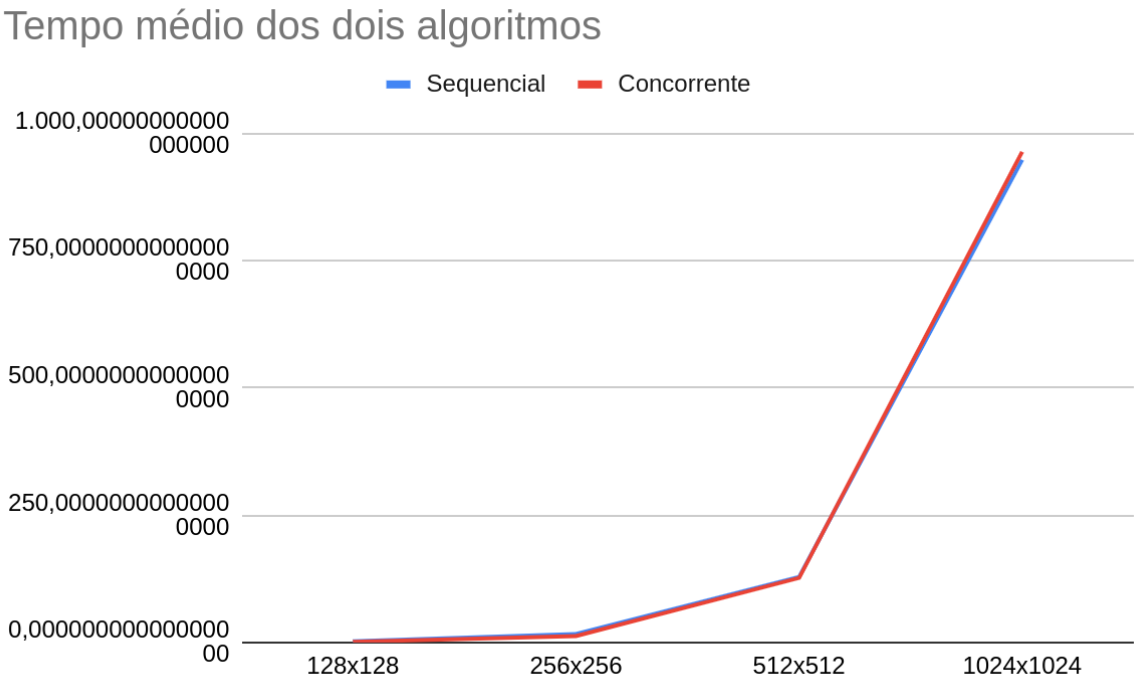

Figura 2: Gráfico de comparação do tempo máximo das dimensões 128x128 até 1024x1024

Todos os resultados, inclusive os mais detalhados podem ser vistos através deste link:<https://drive.google.com/drive/folders/1yOBC3J7UIiavdhOGXdIyTgk_R2v6oLR1?usp=sharing>.

Conclusão

Com base no que foi desenvolvido os resultados obtidos de acordo com os resultados das duas implementações. Conclui-se que os algoritmos concorrentes e sequenciais se saíram muito próximos uns dos outros. Mas vimos que para dimensões de matrizes menores, de 2x2 até 64x64 a forma concorrente se saiu melhor, mas para o restante das dimensões os resultados foram muito próximos.