

Trabalho Prático

Programação com Threads

Introdução

O propósito deste trabalho visa realizar a comparação entre as soluções concorrente e sequencial para a problemática da multiplicação de matrizes. Para realizar essa comparação foi desenvolvida uma aplicação que realiza a multiplicação entre matrizes quadradas de mesma dimensão e foram observados os tempos de execução de cada solução em cenários distintos, a fim de se identificar quais dos métodos utilizados é o mais vantajoso.

Metodologia

Para realização deste projeto foi utilizado um computador Intel Core I5 5200U 2.20 Ghz (2 núcleos), com 500 GB de memória interna e 4GB de memória RAM. O sistema operacional utilizado para a simulação foi o Windows 7 Home Premium 64 Bits. O projeto foi desenvolvido na linguagem de programação Java, junto a IDE Eclipse. A versão do Java utilizada junto ao projeto foi a versão 11 (JDK 11.0.10).

Para execução da atividade foram fornecidos pelo docente um conjunto de matrizes quadradas, em pares, com dimensões de 4 x 4 até 2048 x 2048 (As matrizes fornecidas possuem dimensões que seguem uma sequência de potências de 2). Estes pares de matrizes foram aplicados no processo de multiplicação nos modos sequencial e concorrente. Para cada par de matrizes foram realizadas 40 repetições, 20 no modo concorrente e 20 no modo sequencial, resultando em 400 repetições para todo o conjunto de matrizes fornecidas.

Na solução desenvolvida a aplicação recebe 2 argumentos via linha de comando, um argumento que representa a dimensão das matrizes e outro argumento que representa a solução a ser utilizada. Neste caso, "S" para a solução sequencial e "C" para a solução concorrente. Em seguida, de posse dos dados informados pelo usuário, a aplicação faz a consulta dos arquivos a serem utilizados na operação: Denominados arquivos A e B. Por fim, a aplicação realiza o processo de multiplicação entre as matrizes e salva o resultado final junto a um terceiro arquivo C, localizado nos diretórios do projeto. (Caminhos: bin/resultados/concorrente e bin/resultados/sequencial)

Os tempos de execução encontrados foram calculados considerando apenas o tempo de execução da multiplicação de matrizes. Estes tempos são computados e armazenados pela aplicação em arquivos .txt presentes nos diretórios: bin/tempos/concorrente e bin/tempos/sequencial.

Resultados

De posse dos tempos de solução de cada metodologia, foram criados gráficos e tabelas que representam o comparativo entre os métodos desejados. Neste caso, entre os métodos sequencial e concorrente.

Abaixo segue a tabela de representação da solução sequencial junto das informações estatísticas necessárias. Como ponto interessante observado, percebe-se que a solução no método sequencial apresenta vantagens no cálculo de matrizes com dimensão menor (a nível de tempo), e perde eficiência no cálculo de matrizes com dimensão mais elevada (A partir das matrizes com dimensões 512 x 512).

Dimensão da matriz	Tempo mínimo (ms)	Tempo médio (ms)	Tempo máximo (ms)	Desvio Padrão
4 x 4	0	0,35	1	0,48
8 x 8	0	0,7	1	0,46
16 x 16	0	0,8	1	0,40
32 x 32	1	1,8	3	0,60
64 x 64	4	5,7	11	1,82
128 x 128	14	18,75	31	4,75
256 x 256	48	63,85	78	8,90
512 x 512	335	347,20	357	9,11
1024 x 1024	3088	3185,15	3345	62,21
2048 x 2048	61008	61490,55	61851	252,11

Tabela 1 - Tempos da solução sequencial

Já abaixo, seguem dados estatísticos da solução concorrente. Esta solução apresenta maior eficiência em relação a solução sequencial nas matrizes de grande tamanho (A partir das matrizes com dimensões 512 x 512). Isto se deve, ao melhor processamento realizado em concorrência pelas threads criadas pela aplicação.

Dimensão da matriz	Tempo mínimo (ms)	Tempo médio (ms)	Tempo máximo (ms)	Desvio Padrão
4 x 4	1	1,45	2	0,50
8 x 8	2	2,15	3	0,36
16 x 16	3	4,1	5	0,44
32 x 32	7	7,95	9	0,59
64 x 64	18	19,95	22	1,47
128 x 128	34	41,7	47	3,55
256 x 256	92	101,5	114	7,28
512 x 512	242	263,4	280	12,62
1024 x 1024	1370	1395,95	1414	14,77
2048 x 2048	8999	9098,35	9497	123,55

Tabela 2 - Tempos da solução concorrente

Segue também a tabela comparativa entre os tempos médios das duas soluções propostas. Também é exibido o resultado do tempo do speed-up entre as soluções apresentadas.

Dimensão da matriz	Tempo médio sequencial (ms)	Tempo médio concorrente (ms)	Speed Up (Tempo Sequencial/ Tempo Concorrente)
4 x 4	0,35	1,45	0,24
8 x 8	0,7	2,15	0,33
16 x 16	0,8	4,1	0,20
32 x 32	1,8	7,95	0,23
64 x 64	5,7	19,95	0,29
128 x 128	18,75	41,7	0,45
256 x 256	63,85	101,5	0,63
512 x 512	347,20	263,4	1,32
1024 x 1024	3185,15	1395,95	2,28
2048 x 2048	61490,55	9098,35	6,76

Tabela 3 - Speed Up (Sequencial x concorrente)

Segue abaixo o gráfico em linhas dos tempos da solução sequencial e concorrente. Podemos observar aí o crescimento enorme no modelo sequencial para execução do cálculo para o último par de matrizes. (Com duração em torno de 60 segundos). Já no modelo concorrente, o crescimento a nível de tempo é mais “linear”.

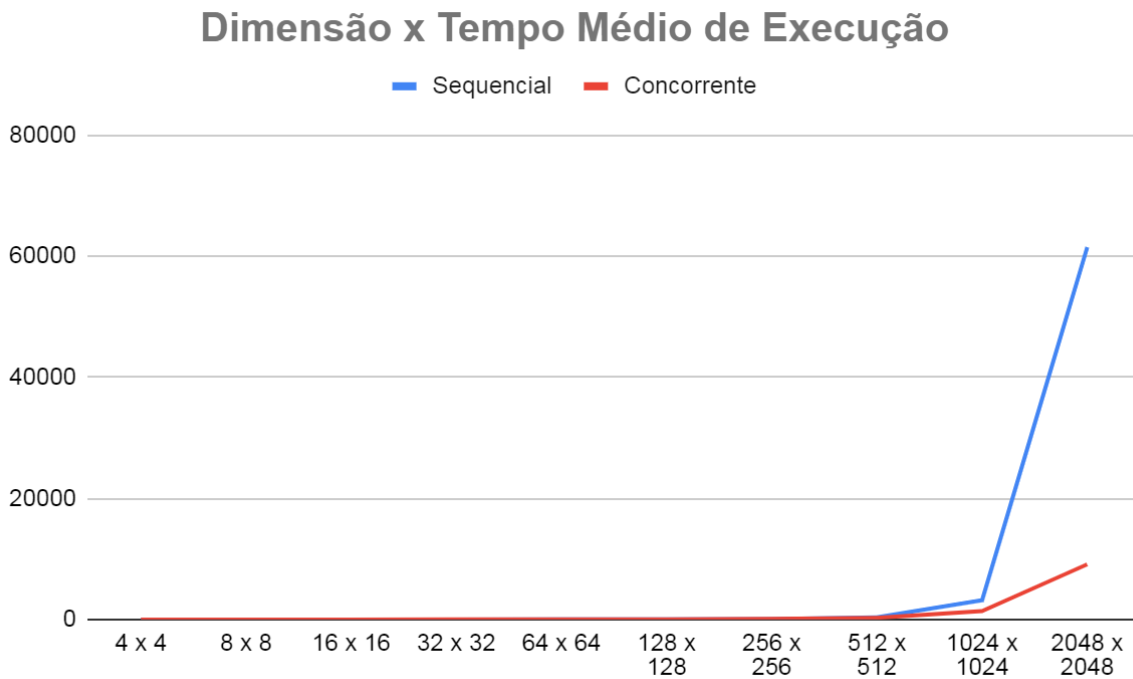


Gráfico 1 - Dimensão x Tempo Médio de Execução

Por fim, segue também um gráfico comparativo entre os speed-ups encontrados:

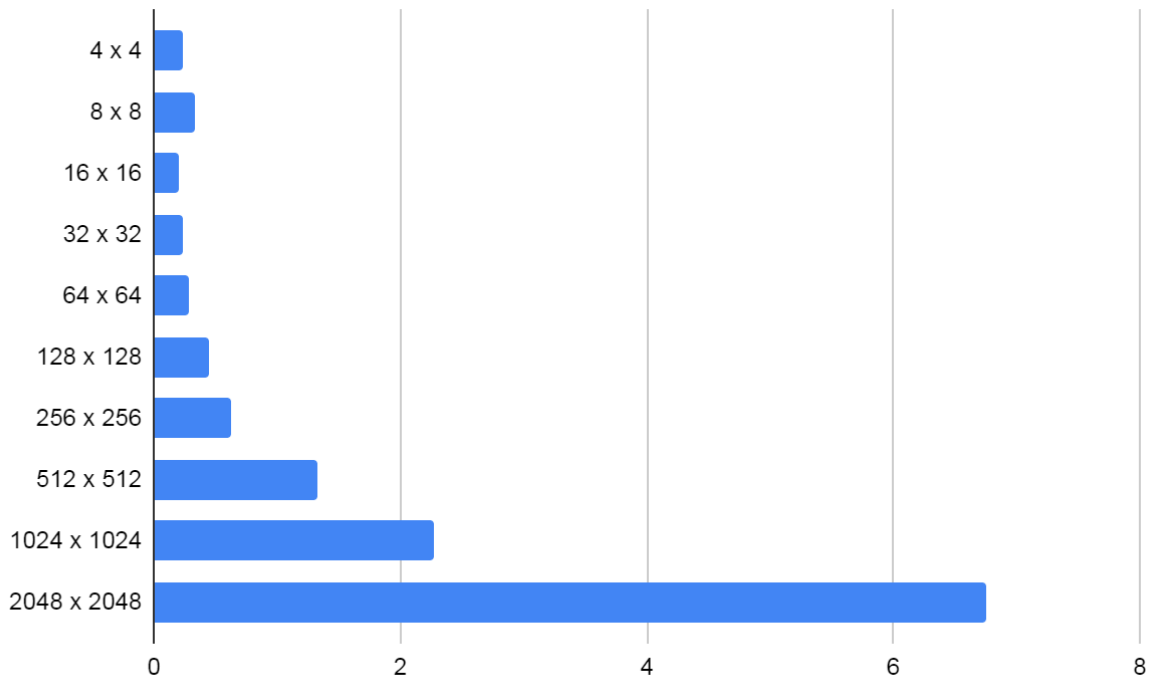


Gráfico 2 - Dimensão x Speed Up

Como bem evidenciado pelo gráfico, os valores acima de 1 retratam as situações em que o modo concorrente é mais vantajoso.

Conclusão

Em razão dos resultados obtidos, podemos afirmar que o desempenho da solução concorrente é semelhante ou, dependendo da situação, inferior à solução sequencial. Nas situações que envolvem poucos processos, como no cálculo das matrizes com dimensões menores, a solução sequencial apresenta vantagens. Todavia, nas matrizes de tamanho mais elevado, o desempenho da solução concorrente é melhor. Isso se deve ao melhor gerenciamento do alto número de processos realizados pelas threads. Nesta situação, no modo concorrente, como o trabalho é realizado em concorrência pelas threads, o processo de solução do problema acaba sendo mais rápido. Em contrapartida, no modelo sequencial, onde os cálculos são realizados em sequência, um após ao outro, o modelo acaba apresentando maior desvantagem.