



UNIVERSIDADE FEDERAL DA GRANDE DOURADOS
Faculdade de Ciências Exatas e Tecnologias - FACET

GEILSO FARIA RODRIGUES DA SILVA

RELATÓRIO SOBRE O TRABALHO DE THREADS
SISTEMAS OPERACIONAIS

Dourados - MS
2023

Introdução

O conceito de threads, também conhecido como "linhas de execução", refere-se à capacidade de dividir um processo de um programa em várias tarefas que podem ser executadas simultaneamente em diferentes núcleos de processamento. Com o uso de threads, um processo pode executar múltiplas tarefas ao mesmo tempo, compartilhando um mesmo espaço de memória e recursos do sistema.

O uso de threads é uma técnica eficaz para melhorar o desempenho de um programa, pois permite que várias partes do processo sejam executadas em paralelo, sem a necessidade de criar novos processos. Neste relatório, será apresentado o uso de threads na linguagem de programação C++, utilizando matrizes e submatrizes de diferentes tamanhos.

Para isso, será explorado um programa que utiliza threads para realizar operações em matrizes e submatrizes. Serão variados os tamanhos das matrizes e submatrizes, bem como o número de threads utilizadas em cada operação. O objetivo é avaliar o desempenho do programa em diferentes configurações e comparar os resultados.

Em resumo, este relatório irá explorar em detalhes o uso de threads na linguagem de programação C++, demonstrando suas vantagens e desvantagens em diferentes cenários.

Experimentos

Máquina de Teste:

Processador: Intel(R) Core(TM) i3-7020 U CPU @ 2.30GHz

Memória RAM instalada: 4 GB

Sistema Operacional: Windows 10 Home Single Language

Número de núcleos: 2

Nº de threads: 4 (dois núcleos físicos e dois virtuais)

Testes

No âmbito deste trabalho, foi estabelecido um tamanho máximo de 15.000 x 15.000 para as submatrizes utilizadas, uma vez que foi usado da memória virtual para tamanhos ligeiramente maiores, o que não se enquadra nos objetivos deste relatório. Para facilitar a compreensão dos experimentos realizados, os resultados serão apresentados em formato de gráficos neste relatório. Tais resultados incluem:

Matriz 1000X1000

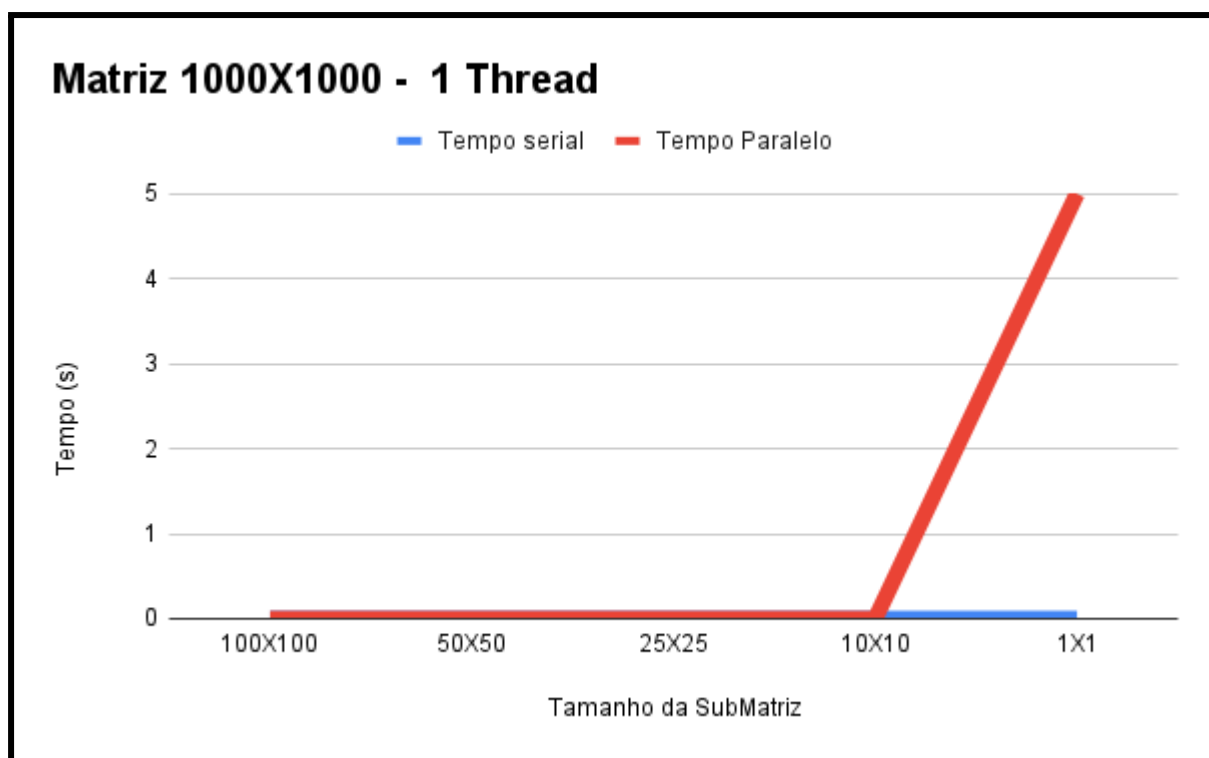


Figura 1 - Fonte: Geilso Faria Rodrigues da Silva

Na figura em questão, é possível observar que, devido ao tamanho da matriz ser considerado pequeno para o computador, os tempos de processamento em série e em paralelo são iguais. No entanto, no caso paralelo, quando o número de submatrizes aumenta demasiadamente, o tempo de processamento tende a aumentar também, como pode ser visto quando o tamanho das submatrizes é de 1x1. Esse fato pode ser explicado pelo fato de que, no processamento paralelo, é necessário fazer uma gestão correta dos recursos, e se o número de submatrizes for muito pequeno, essa gestão terá que ser feita muitas vezes, o que pode aumentar o tempo de processamento.

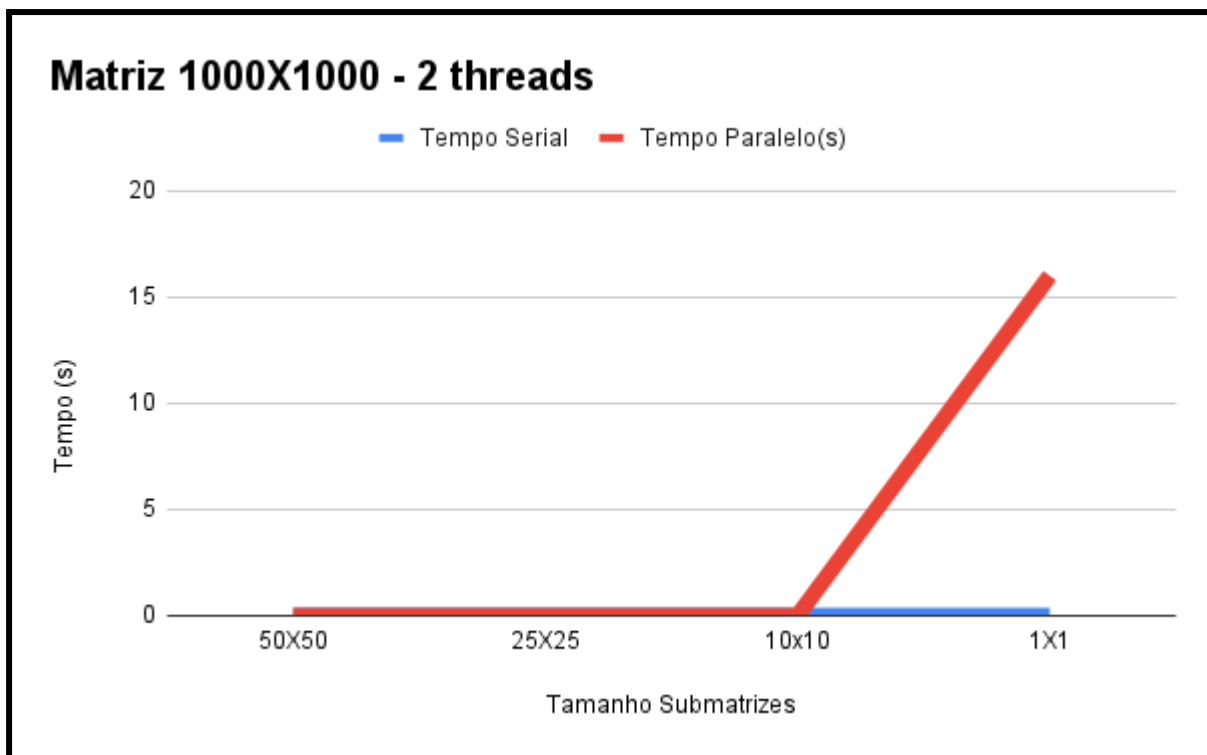


Figura 2 - Fonte: Geilso Faria Rodrigues da Silva

Nesta figura, pode-se observar um comportamento similar ao da figura anterior. Além disso, é possível notar que um aumento na quantidade de threads resulta em um desempenho inferior ao caso serial (16 segundos). Isso ocorre porque, devido à comunicação entre as threads e ao compartilhamento de recursos entre elas, há um tempo considerável envolvido nesse processo.

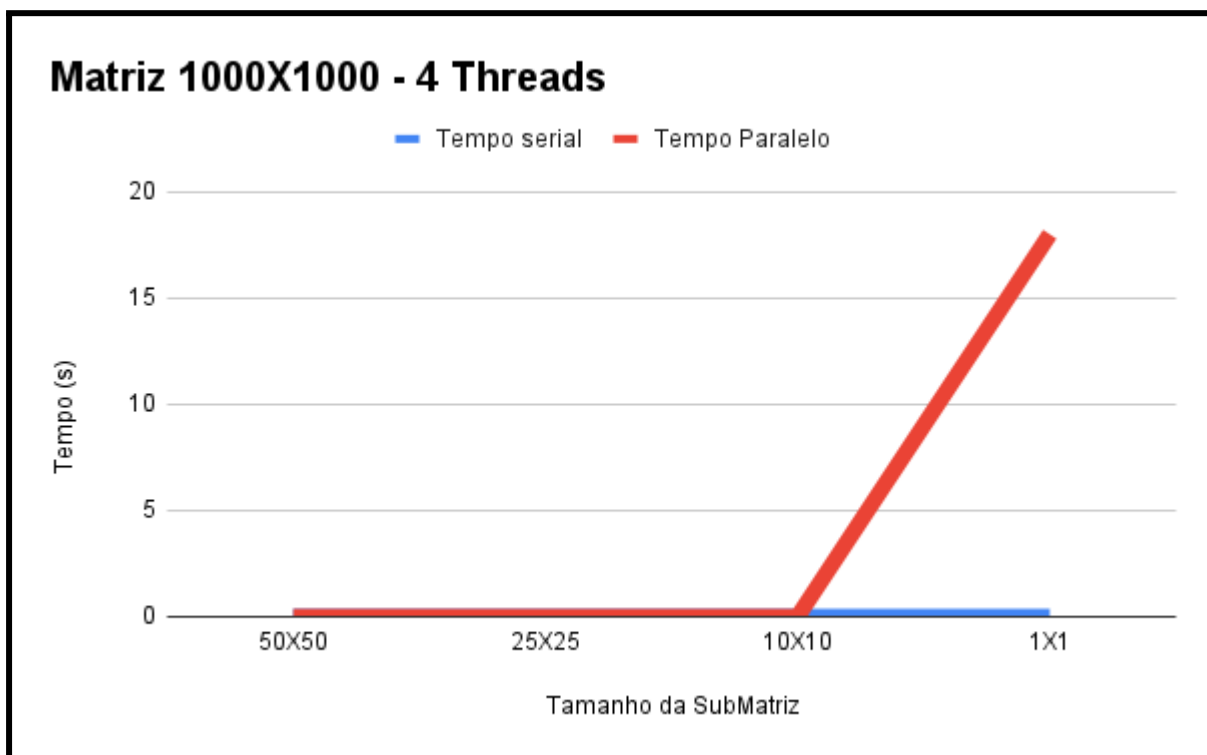


Figura 3 - Fonte: Geilso Faria Rodrigues da Silva

Matriz 10000X10000

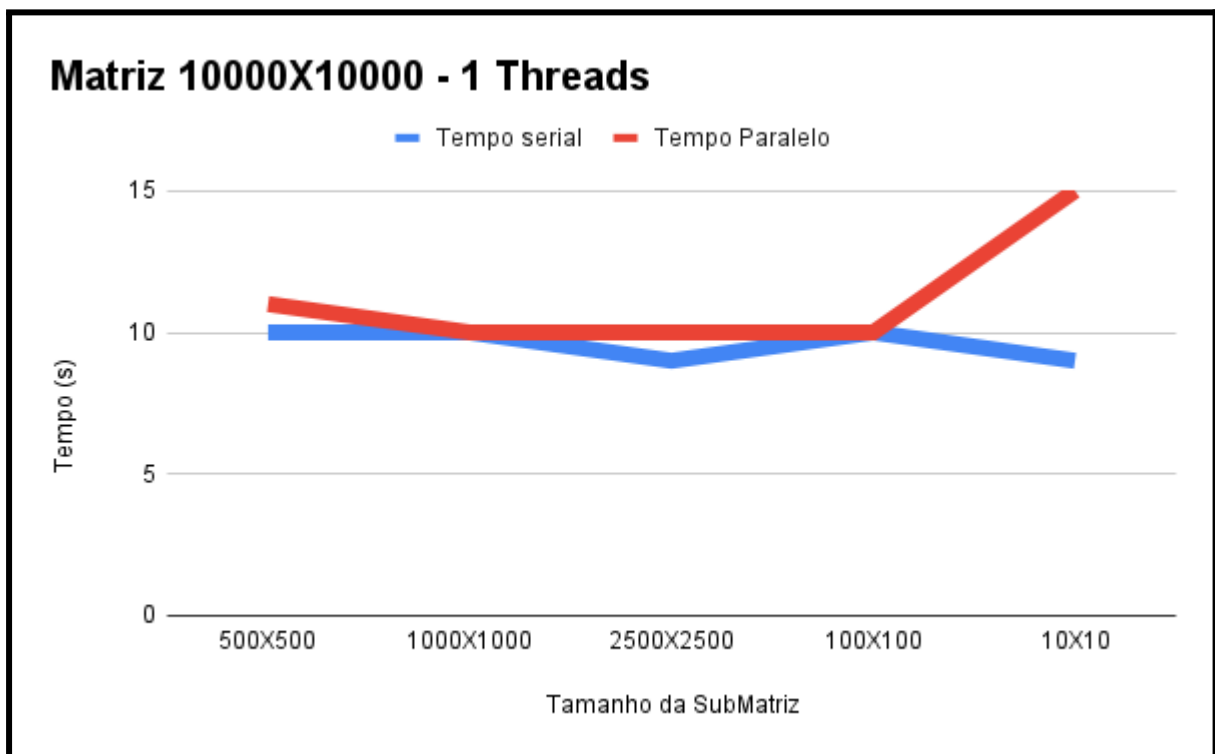


Figura 4 - Fonte: Geilso Faria Rodrigues da Silva

Com o aumento do número de elementos na matriz, foi possível perceber que os tempos de processamento em série e em paralelo eram praticamente os mesmos, exceto pelo aumento abrupto no número de submatrizes, como no caso de 10x10

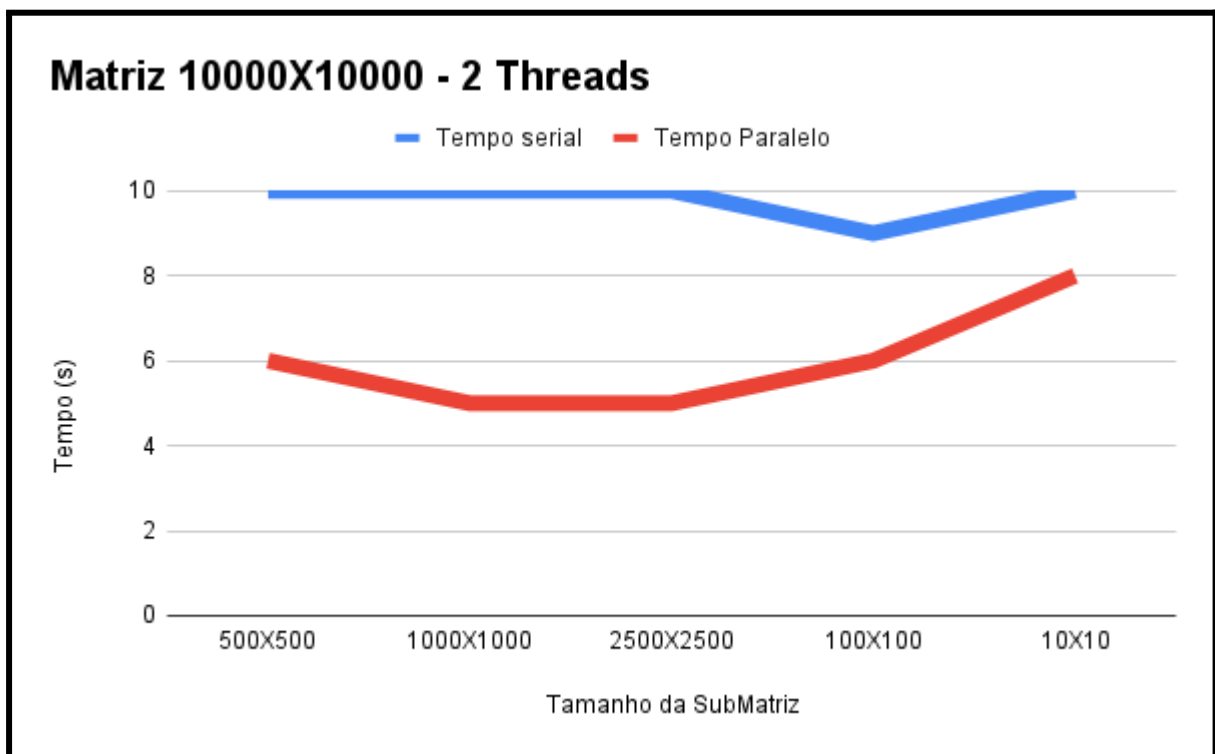


Figura 5 - Fonte: Geilso Faria Rodrigues da Silva

Nesta figura, já é possível observar os benefícios do uso de threads para a agilidade do processamento. Comparando com a figura anterior, é notável que o tempo de processamento em paralelo é consideravelmente menor em relação ao tempo em série. Como mencionado anteriormente, o tempo de processamento somente aumenta quando a quantidade de submatrizes é muito grande.

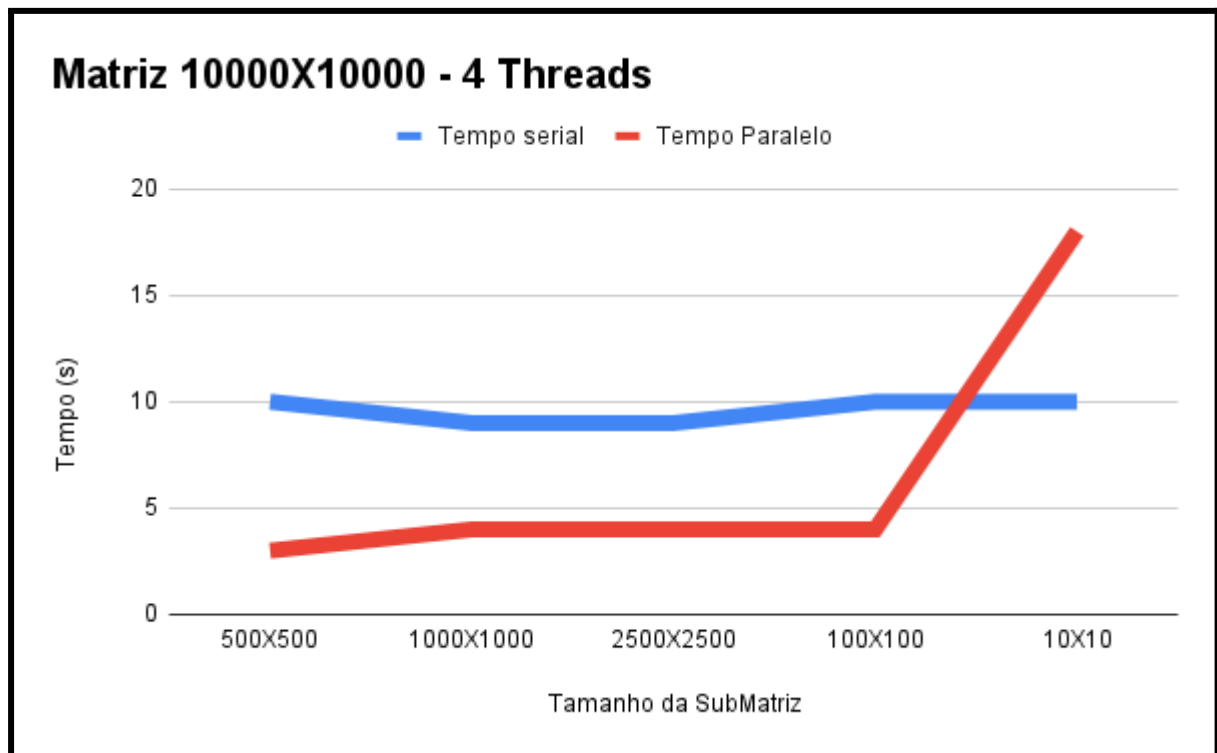


Figura 6 - Fonte: Geilso Faria Rodrigues da Silva

Neste experimento, foi possível observar claramente a diferença de tempo entre os processamentos em paralelo e sequencial. Na maior parte do tempo, o tempo de processamento em paralelo foi cerca de duas vezes menor em relação ao uso sequencial. O tempo em paralelo só aumentou novamente quando o número de submatrizes cresceu.

Matriz 15000 X 15000

De modo geral, neste experimento, foi observado o mesmo comportamento das figuras anteriores. Quanto menor o número de submatrizes, melhor para o uso de threads. Por outro lado, quanto maior o número de submatrizes, pior, devido à gestão de recursos e sincronização das threads.

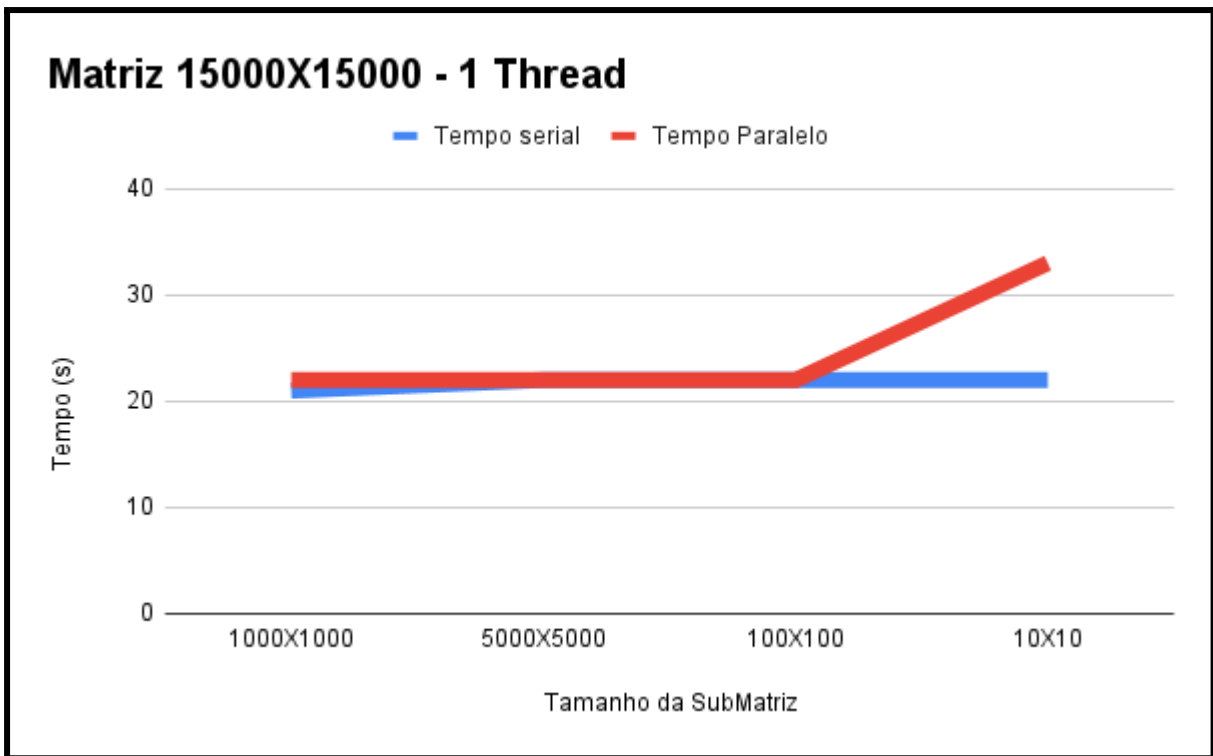


Figura 7 - Fonte: Geilso Faria Rodrigues da Silva

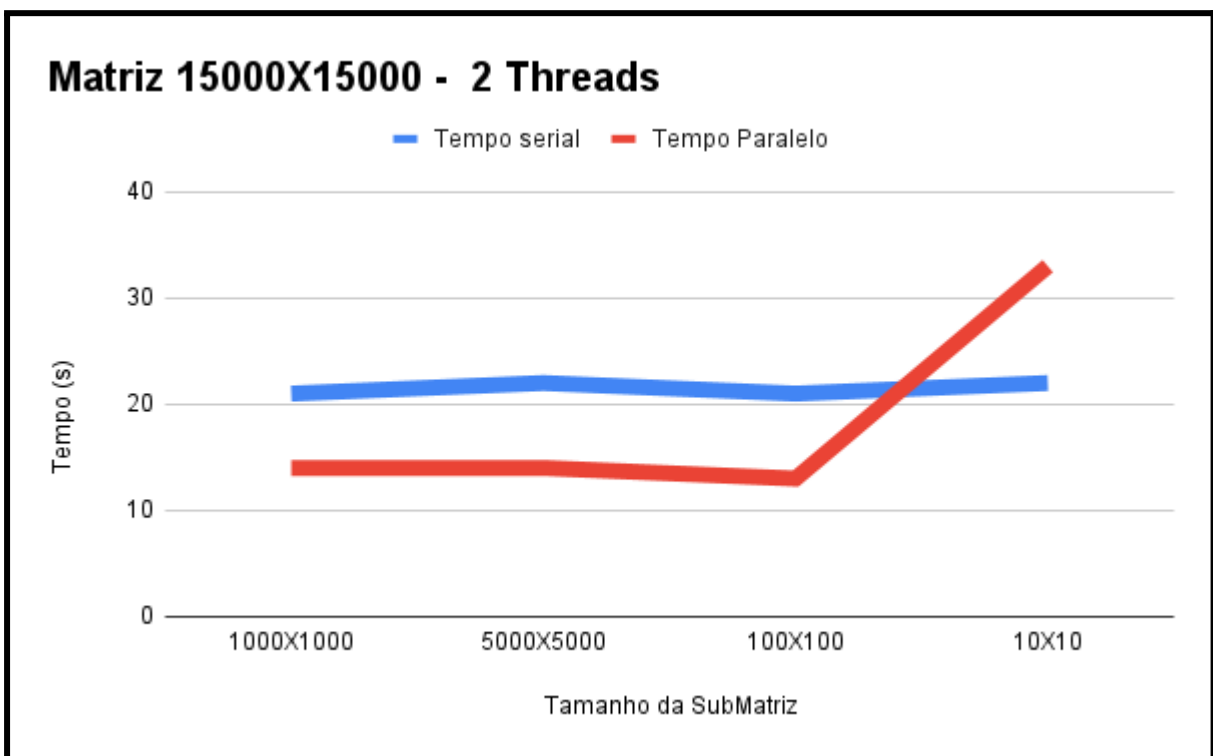


Figura 8 - Fonte: Geilso Faria Rodrigues da Silva

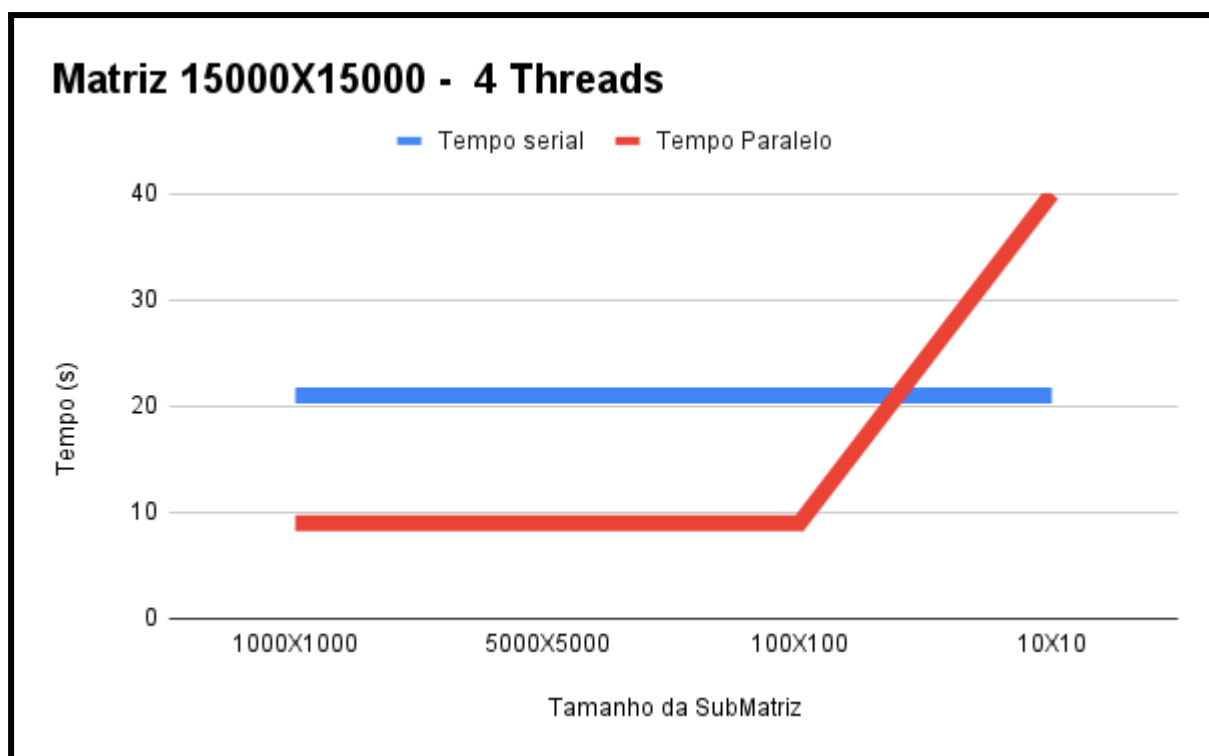


Figura 9 - Fonte: Geilso Faria Rodrigues da Silva

Caso especial - Submatrizes 1 X 1

Durante os testes, foi observado um fator interessante sobre o uso de submatrizes muito pequenas, como já ressaltado anteriormente. No entanto, um fato especial que comprovou isso foi o caso das matrizes 10000 X 10000 divididas em submatrizes de 1X1. O tempo aumentou grandemente, chegando a 461 segundos para 1 thread, 1640 segundos para 2 threads e 1790 segundos para 4 threads. Isso pode ser explicado devido à existência de muitas submatrizes, o que torna os cálculos em paralelo lentos. Além disso, aumentando o número de threads, é necessário gerenciar os recursos compartilhados entre elas, o que leva ainda mais tempo. Para valores maiores, foi usado memória virtual, e portanto, os testes não foram contabilizados.

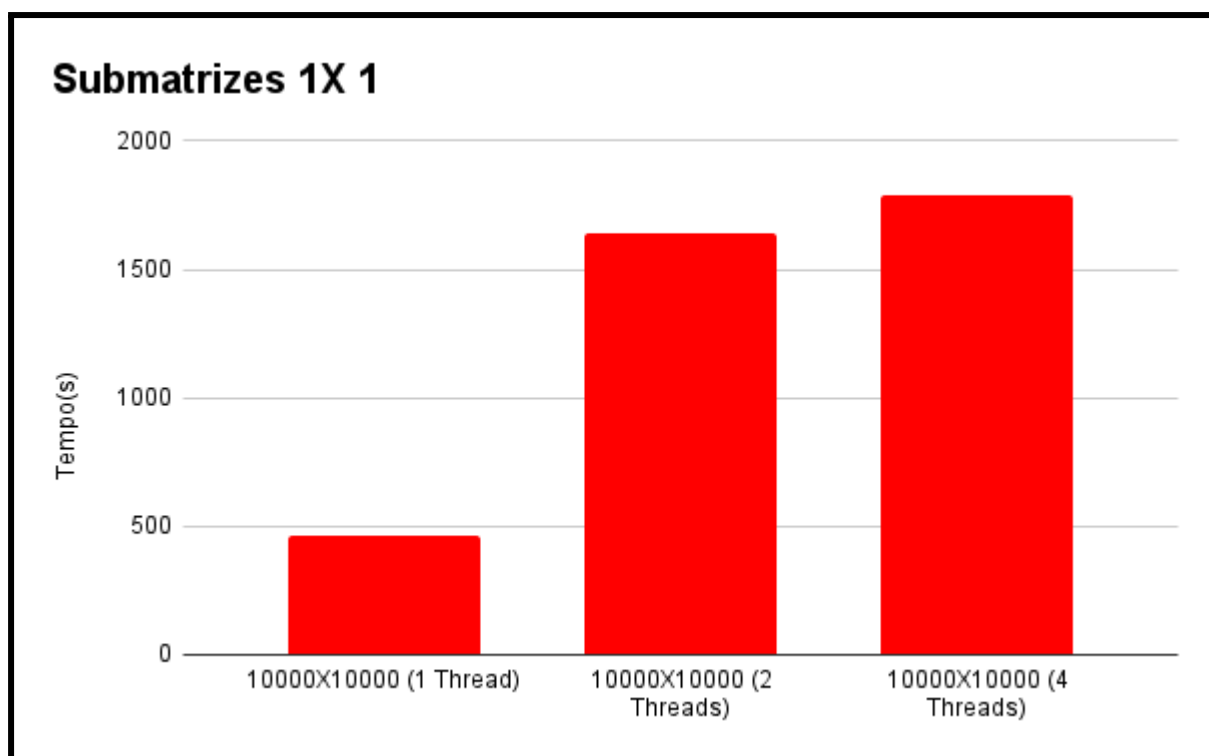


Figura 10 - Fonte: Geilso Faria Rodrigues da Silva

Conclusão

Portanto, nos experimentos realizados foi observado que, para matrizes pequenas, o tempo de execução em serial e em paralelo, usando threads, é praticamente o mesmo. No entanto, para matrizes maiores se as submatrizes forem muito pequenas, o processamento das threads é mais lento, devido à sobrecarga de comunicação entre elas. Para submatrizes grandes e um número ideal de threads por processador, e uma divisão ideal de matrizes por submatrizes, o desempenho em paralelo é muito mais rápido. No entanto, se a quantidade de submatrizes for muito grande, o desempenho será afetado, especialmente se a quantidade de threads for muito elevada, devido à sobrecarga de comunicação e ao compartilhamento de recursos entre as threads.

Conclui-se que o uso de threads é excelente, pois proporciona eficiência e agilidade aos processos executados em mais de um núcleo. No entanto, é preciso ter atenção a alguns fatores que podem diminuir seu desempenho, como submatrizes muito pequenas ou uma quantidade excessiva de threads, o que pode causar gargalos de comunicação. Sendo assim, é recomendável fazer uma análise cuidadosa e otimizar o uso de threads para obter o melhor desempenho possível.