

## Análise de Desempenho

Os gráficos foram gerados a partir dos dados no arquivo data/data.md, sendo que o valor considerado é a média dos 5 testes, aqui analisando apenas o tempo de Processamento, onde tivemos mudanças devido a escolha da maneira com que fizemos a multiplicação.

### Gráfico de Tempo de execução do Processamento por número de threads

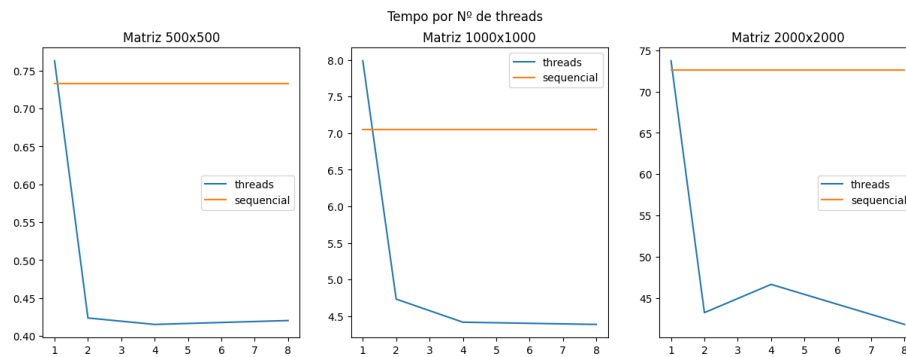
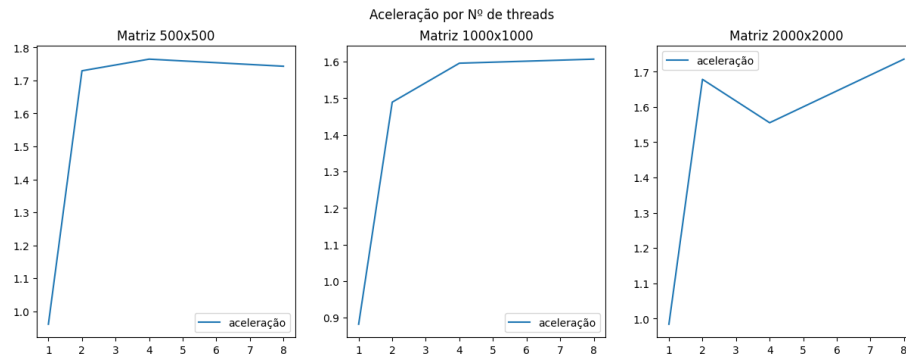


Figure 1: grafico TxT

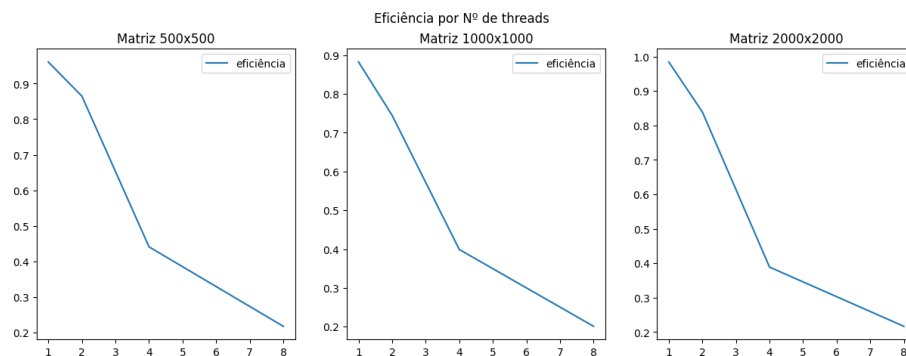
Neste gráfico temos a linha laranja indicando o tempo de processamento gasto para executar o programa sequencial e a curva azul em função do aumento do número de threads. Em todos os gráficos é possível perceber como a mudança para concorrência diminui consideravelmente o tempo de processamento, mas na terceira há um pico no caso das 4 threads, o que parece ter mais relação com outros programas do computador do que especificamente o código, na tabela de dados podemos ver que houve um aumento no tempo das 4 threads que não parece muito coerente visto que com 8 threads tivemos uma queda novamente.

## Gráfico de Tempo de execução do Processamento por número de threads



Nesse gráfico vemos como o aumento das threads aumenta a aceleração. Nessa fórmula o tempo com as threads está no denominador, por isso esperamos um gráfico crescente, indicando que o denominador está diminuindo. Não temos o valor sequencial neste caso, mas consideremos o valor com 1 thread (que seria um uso incorreto, já que aumentamos o tempo com a criação de threads e continuamos usando apenas um fluxo) para notar que, nos três gráficos, já com duas threads acontece um grande crescimento da curva, indicando como o uso correto de threads melhora nosso programa. No primeiro gráfico vemos um pico em 4, em seguida a aceleração decresce, indicando que aumentar o número de threads para além de 4 pode não compensar tanto. No segundo e terceiro o pico está em 8 threads, mas no segundo a diferença é pouca, enquanto no terceiro, temos uma queda em 4 threads, o que é esperado devido ao gráfico anterior, que já apresentava essa anormalidade.

## Gráfico de Tempo de execução do Processamento por número de threads



Por fim a eficiência, que eu acredito não ter os resultados desejados, pois as três curvas tem um decrescimento muito rápido, com pico próximo a 1 utilizando 1 thread. Ao meu ver isso indica que o ganho em aceleração não foi

tão relevante/bom para a quantidade de threads que criamos. Ou seja, apesar de ganharmos tempo, o custo benefício não é tão grande assim ao usar as threads.