

Relatório do Projeto

TT304 – Sistemas Operacionais
Professor André Leon S. Gradvohl

GRUPO DevDuo

Beatriz Galvão Pinheiro - 233733
Karen Utino Sartori - 260697

Limeira – SP
2024

Introdução

O projeto consiste no desenvolvimento de um programa em linguagem C, que realiza operações com matrizes e utiliza múltiplas threads para otimizar o desempenho.

Descrição do problema

O projeto envolve três matrizes de entrada (A, B e C) de dimensões $N \times N$ (100x100 ou 1000x1000).

O programa realiza as seguintes operações com as matrizes:

- Soma de matrizes ($A + B = D$)
- Multiplicação de matrizes ($C \times D = E$)
- Redução da matriz E em um único valor
- Cálculo dos tempos de execução

Os testes foram realizados com diferentes números de threads e tamanho de matrizes:

- Número de threads: 1, 2 e 4
- Tamanho das matrizes: 100 x 100 e 1000 x 1000

Instruções para compilação

Para a compilação do programa, é necessário o sistema Linux, as instruções dadas são referentes a esse ambiente.

1. Acesse o repositório disponível no seguinte endereço no GitHub:
<https://github.com/KarenSartori/Projeto_Sistemas_Operacionais.git>
2. Clone o repositório do Github ou baixe os arquivos postados
3. Acesse o terminal do seu computador e faça os seguintes comandos, modelo disponível no arquivo "shell.txt". Podendo ser T (1, 2 ou 4) e N(100 ou 1000)

make

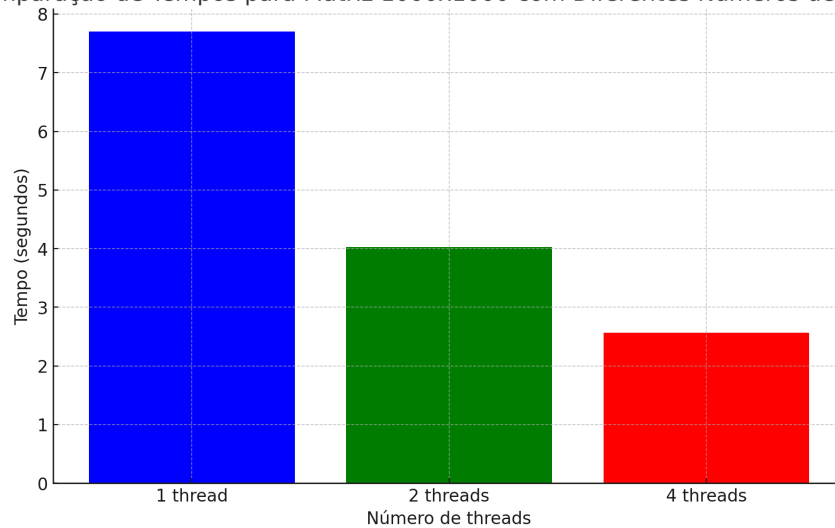
./projeto T N arqA.dat arqB.dat arqC.dat arqD.dat arqE.dat

Gráfico com os tempos de execução

Matriz 1000 x 1000

- **1 thread:** 7.700543 segundos.
- **2 threads:** 4.027885 segundos.
- **4 threads:** 2.561983 segundos.

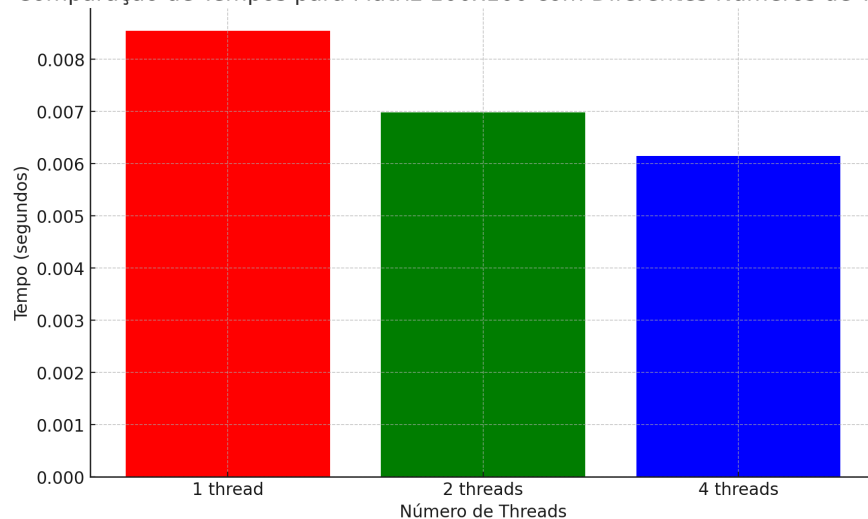
Comparação de Tempos para Matriz 1000x1000 com Diferentes Números de Threads



Matriz 100 x 100

- **1 thread:** 0.008544 segundos.
- **2 threads:** 0.006979 segundos.
- **4 threads:** 0.006150 segundos.

Comparação de Tempos para Matriz 100x100 com Diferentes Números de Threads



Conclusão

Os resultados mostram que o uso de múltiplos threads melhora significativamente o tempo de execução das operações com matrizes. À medida que o número de threads aumenta, os tempos para a soma e a multiplicação das matrizes diminuem, demonstrando a eficácia da paralelização.

Por outro lado, a operação de redução não mostra uma variação significativa no tempo de execução, pois é uma operação relativamente simples e rápida. O tempo total de execução do programa também diminui com o aumento do número de threads, confirmando a eficiência da abordagem paralela. Essa eficácia é evidenciada ainda mais em testes feitos com a matriz maior (1000x1000).

Esses resultados destacam a importância de utilizar técnicas de programação com múltiplos threads para otimizar a performance de aplicações que envolvem operações intensivas em computação.

Vídeo disponível em: <https://youtu.be/YiuwNFdoQ9Q>