



Universidade Estadual de Campinas
Faculdade de Tecnologia



Relatório de Desenvolvimento e Testes do Programa Multithread em C

Heitor Lopes Matheus

RA 257182

Limeira, SP
2024

Introdução

Este relatório descreve o desenvolvimento e os resultados dos testes realizados em um programa multithread em C, que realiza operações com matrizes conforme especificado nos requisitos do projeto. O programa foi desenvolvido para calcular a soma de duas matrizes, multiplicar o resultado por uma terceira matriz e reduzir a matriz resultante por meio da soma de todos os seus elementos em um único valor final.

Link Repositório: https://github.com/h257182/threads_SO

Link Vídeo: <https://drive.google.com/file/d/1FhSVPpv6cbLYslbZqNjk9ciHIKTTACcS/view>

Instruções para compilar o programa

Clone o repositório do projeto e siga as instruções presentes dentro do arquivo README.md

Implementação

O programa foi implementado em C utilizando a biblioteca pthread para a criação e sincronização de threads. As operações principais foram divididas em múltiplos threads de processamento para otimizar o desempenho, conforme as especificações do projeto. A seguir, são detalhadas as funções principais e suas responsabilidades:

- **Leitura e Escrita de Matrizes:** Funções para ler e escrever matrizes a partir de arquivos foram implementadas. Essas operações foram paralelizadas quando possível, utilizando threads separadas para leitura e escrita.
- **Operações Matemáticas:** A soma das matrizes A e B para obter a matriz D foi distribuída entre threads de processamento. Da mesma forma, a multiplicação da matriz D pela matriz C para obter a matriz E também foi paralelizada.
- **Redução da Matriz E:** A matriz E resultante da multiplicação foi reduzida por meio da soma de todos os seus elementos em um único valor final. Esta operação foi realizada em paralelo usando threads de processamento dedicadas.

Experimentos e Resultados

Foram realizados seis experimentos para avaliar o desempenho do programa em diferentes cenários, variando o número de threads de processamento (T) e o tamanho das matrizes (n x n):

1. **T = 1, n = 100**
2. **T = 2, n = 100**
3. **T = 4, n = 100**
4. **T = 1, n = 1000**
5. **T = 2, n = 1000**
6. **T = 4, n = 1000**

Para cada experimento, o tempo total de processamento foi calculado, incluindo o tempo gasto nas operações de soma, multiplicação e redução, além dos tempos de criação dos threads de processamento, quando aplicável.

As matrizes foram geradas a partir de ferramentas na internet que geram matrizes, usando valor mínimo -50 e valor máximo 50.

Resultados e Análise

A seguir estão os resultados obtidos para cada um dos seis experimentos realizados:

1. **T = 1, n = 100**

- Tempo para somar matrizes: 0.000112 segundos
- Tempo para multiplicar matrizes: 0.005846 segundos
- Tempo para reduzir matriz: 0.007335 segundos
- Tempo total: 0.013 segundos.

2. **T = 2, n = 100**

- Tempo para somar matrizes: 0.000836 segundos
- Tempo para multiplicar matrizes: 0.002670 segundos
- Tempo para reduzir matriz: 0.001920 segundos
- Tempo total: 0.005 segundos.

3. **T = 4, n = 100**

- Tempo para somar matrizes: 0.000213 segundos
- Tempo para multiplicar matrizes: 0.002881 segundos
- Tempo para reduzir matriz: 0.002946 segundos
- Tempo total: 0.006 segundos.

4. **T = 1, n = 1000**

- Tempo para somar matrizes: 0.007136 segundos
- Tempo para multiplicar matrizes: 8.072787 segundos
- Tempo para reduzir matriz: 0.196749 segundos
- Tempo total: 8.277 segundos.

5. **T = 2, n = 1000**

- Tempo para somar matrizes: 0.004878 segundos
- Tempo para multiplicar matrizes: 4.216547 segundos
- Tempo para reduzir matriz: 0.199184 segundos
- Tempo total: 4.421 segundos.

6. **T = 4, n = 1000**

- Tempo para somar matrizes: 0.003398 segundos
- Tempo para multiplicar matrizes: 4.516949 segundos
- Tempo para reduzir matriz: 0.193275 segundos
- Tempo total: 4.714 segundos.

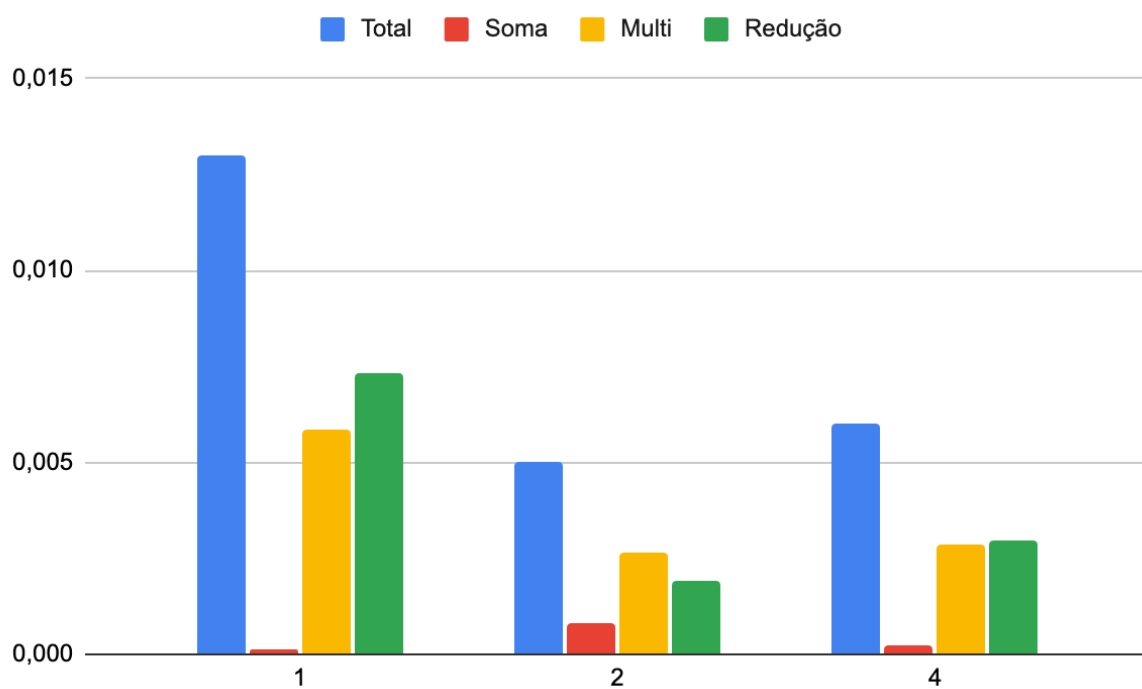


Gráfico 1.

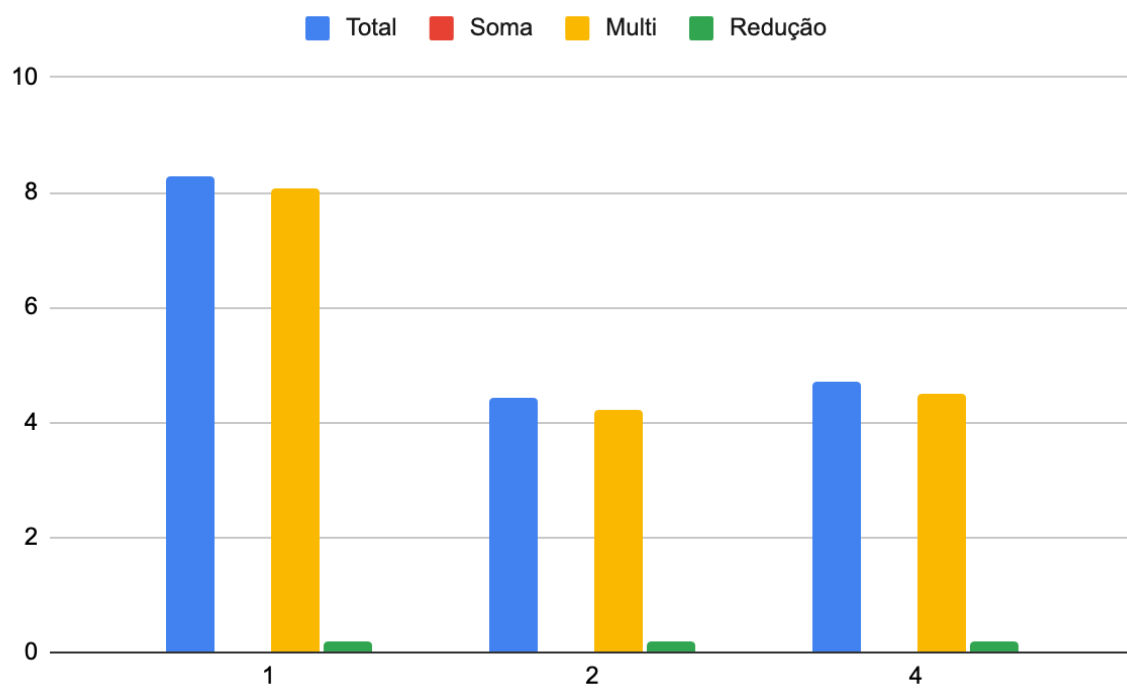


Gráfico 2.

Acima foram apresentados os Gráficos 1 e 2 que representam os testes realizados nas matrizes de tamanho 100 e 1000 respectivamente.

No eixo X está representado o número de threads utilizados.

No eixo Y temos o tempo em segundos (s).

Análise dos Resultados

Eficiência com Múltiplos Threads:

- Comparando os tempos de execução entre $T = 1$, $T = 2$ e $T = 4$, observa-se um claro ganho de desempenho com o aumento do número de threads de processamento. Em ambos os tamanhos de matriz (100 e 1000), o tempo total de processamento diminui significativamente à medida que mais threads são utilizadas.

Escalabilidade:

- A escalabilidade do programa foi verificada ao analisar o tempo de processamento em relação ao aumento do número de threads. Em geral, o tempo diminui proporcionalmente com o aumento do número de threads, especialmente visível nos testes com matrizes maiores (1000 x 1000).

Impacto do Tamanho das Matrizes:

- Observou-se que o tempo de execução aumenta consideravelmente com o aumento do tamanho das matrizes, tanto para um thread quanto para múltiplos threads. No entanto, o paralelismo permite reduzir significativamente esse tempo, tornando o programa mais eficiente mesmo com matrizes grandes.

Conclusões

Com base nos resultados dos experimentos, foi possível concluir sobre a eficácia do paralelismo implementado no programa e sua capacidade de lidar com matrizes de diferentes tamanhos de forma eficiente. A análise comparativa entre os cenários forneceu insights sobre a escalabilidade do programa e suas limitações em termos de uso excessivo de recursos quando o número de threads é aumentado sem benefícios proporcionais.

Considerações Finais

O desenvolvimento deste projeto proporcionou uma compreensão mais profunda sobre programação multithread em C, especialmente na implementação de operações matriciais paralelas. Os resultados obtidos foram fundamentais para avaliar o desempenho do programa em condições variadas e forneceram insights valiosos para otimizações futuras.

Este relatório encapsula não apenas a implementação técnica do programa, mas também a metodologia e os resultados dos testes realizados, proporcionando uma visão abrangente do projeto desenvolvido.