UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
DEPARTAMENTO DE INFORMÁTICA E MATEMÁTICA APLICADA

Disciplina: DIM0612 — Programação Concorrente
Docente: Everton Ranielly de Sousa Cavalcante
Discente: Felipe Cortez de Sá

Multiplicação de matrizes

1 Introdução

Este relatório descreve a implementação de algoritmos para multiplicação de matrizes de forma sequencial e usando múltiplas threads, detalhando as estratégias utilizadas e comparando resultados.

2 Detalhes da implementação

O problema foi resolvido utilizando a linguagem de programação C++11 com a biblioteca `std::thread`. No Mac foi selecionado o compilador `clang` e em Ubuntu `g++`, ambos utilizando a flag `-O2` para otimizações. As matrizes A, B e C foram codificadas como vetores (arrays estilo C) de tamanho n^2.

Na implementação com t *threads*, cada thread é responsável pelo cálculo de $\frac{n^2}{t}$ elementos da matriz C. Caso a divisão não seja exata, a última thread calcula o resto, ou seja, $\frac{n^2}{t} + (n^2 \mod t)$ elementos. As threads são criadas dentro do método `emplace_back`

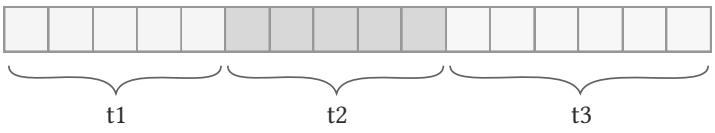

Figura 1: Matriz C $4x4$ com três threads

2.1 Medição de desempenho

Foi utilizada a biblioteca `std::chrono` para medir o tempo de execução. Na implementação sequencial, o *timer* inicia antes dos laços de multiplicação e para depois dos laços. Na implementação com *threads*, o *timer* inicia no momento anterior à criação da primeira *thread* e o posterior ao último `join()`.

2.2 Testes

Os testes foram automatizados com um script programado em *Python* que executa os programas sequencial e concorrente com valores diferentes para número de threads utilizadas e retorna o mínimo, médio e máximo, bem como desenha gráficos comparando as velocidades de execução para cada configuração.

3 Resultados

Caso	mínimo	média	máximo
Sequencial	2	3	4
2 threads	5	6	4
3 threads	8	9	4
4 threads	8	9	4

Tabela 1: Para n = .22l2l2

4 Discussão

A partir da realização desse trabalho foi possível concluir que o uso de threads pode aumentar o desempenho na resolução de problemas paralelos.