Fabrício Mady **1215080268**

Multiplicação Paralela de Matrizes

1 – Código em C++:

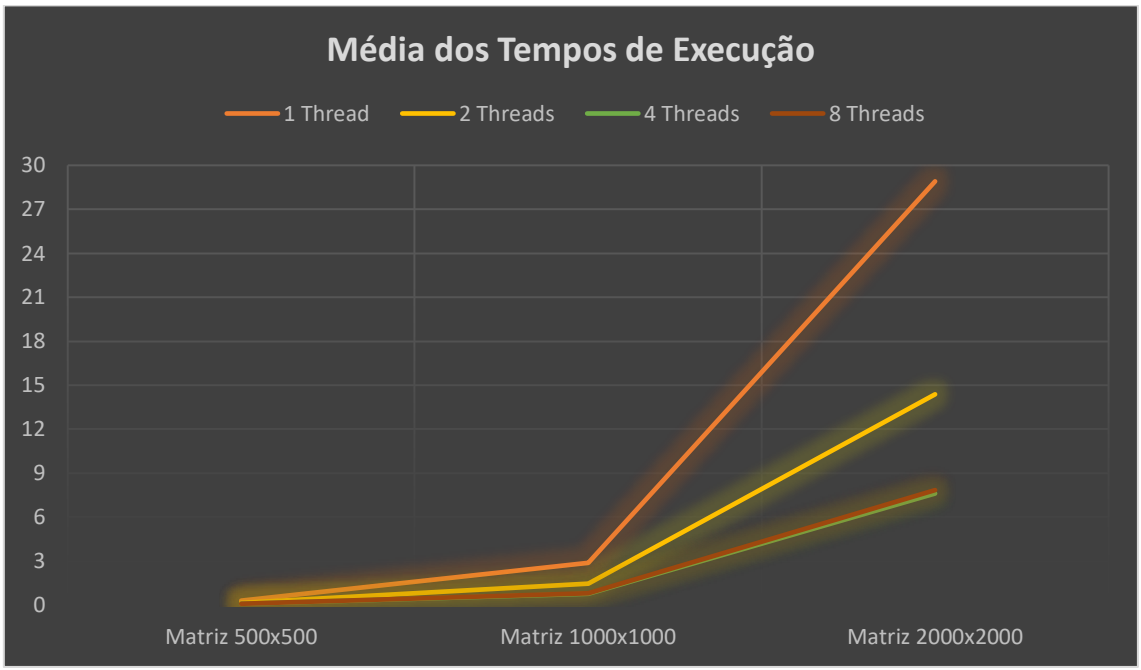
```cpp
#include <iostream>
#include <pthread.h>
#include <stdlib.h>
#include <time.h>

using namespace std;

// Tamanho da matriz !!
#define TAM 50

// Numero de Threads, aumentar ou diminuir para testar
#define THREADS 1

int m1[TAM][TAM];
int m2[TAM][TAM];
int m3[TAM][TAM];
int cont = 0;

void* multiplica(void* arg){
    int nucl = cont++;

    for (int i = nucl * TAM / THREADS; i < (nucl + 1) * TAM / THREADS; i++) {
        for (int j = 0; j < TAM; j++) {
            for (int k = 0; k < TAM; k++) {
                m3[i][j] += m1[i][k] * m2[k][j];
            }
        }
    }

}


int main(){

    clock_t tempo_inicial, tempo_final;



    for (int i = 0; i < TAM; i++) {
        for (int j = 0; j < TAM; j++) {
            m1[i][j] = rand() % 10;
            m2[i][j] = rand() % 10;
        }
    }


    pthread_t threads[THREADS];

    tempo_inicial = clock();

    for (int i = 0; i < THREADS; i++) {
        int* p;
        pthread_create(&threads[i], NULL, multiplica, (void*)(p));
    }


    for (int i = 0; i < THREADS; i++) {
        pthread_join(threads[i], NULL);
    }

    tempo_final = clock();
    cout << fixed;
    cout.precision(6);
    cout <<"execucao: "<< (tempo_final - tempo_inicial)/((double)CLOCKS_PER_SEC) ;

    return 0;
}
```

2 – Avaliação dos tempos de execução:

THREADS	TAMANHO MATRIZ	TEMPO DE EXECUÇÃO MULTIPLICAÇÃO (5X EXECUTADO)					MÉDIA
1	500	0.290000,	0.310000,	0.303000,	0.313000,	0.295000	0.3022
2	500	0.148000,	0.146000,	0.148000,	0.165000,	0.153000	0.152
4	500	0.081000,	0.098000,	0.079000,	0.090000,	0.100000	0.0896
8	500	0.085000,	0.092000,	0.096000,	0.092000,	0.089000	0.0908

THREADS	TAMANHO MATRIZ	TEMPO DE EXECUÇÃO MULTIPLICAÇÃO (5X EXECUTADO)					MÉDIA
1	1000	2.911000,	2.928000,	2.875000,	2.896000,	2.887000	2.8994
2	1000	1.441000,	1.481000,	1.456000,	1.458000,	1.446000	1.4564
4	1000	0.768000,	0.798000,	0.747000,	0.743000,	0.745000	0.7602
8	1000	0.787000,	0.821000,	0.789000,	0.870000,	0.792000	0.8118

THREADS	TAMANHO MATRIZ	TEMPO DE EXECUÇÃO MULTIPLICAÇÃO (5X EXECUTADO)					MÉDIA
1	2000	28.782000,	29.285000,	28.839000,	28.771000,	28.854000	28.9062
2	2000	14.316000,	14.289000,	14.224000,	14.845000,	14.220000	14.3788
4	2000	7.809000,	7.552000,	7.569000,	7.620000,	7.498000	7.6096
8	2000	8.146000,	7.656000,	7.929000,	7.828000,	7.657000	7.8432



Avaliando os Resultados:

Através da implementação e da análise dos tempos de execução, foi possível perceber a redução drástica dos tempos necessários para executar multiplicação de matrizes. Foram realizadas 5 execuções para cada quantidade de threads e, o resultado acima é calculado pela média das execuções realizadas.

Embora que, para valores pequenos (1 até 500), os resultados foram em média próximos para o número de threads testados.

A situação muda a partir de matriz 1000x1000, onde foi possível verificar a redução de, em média, aproximadamente 50% do tempo de cálculo da multiplicação das matrizes apenas dobrando a quantidade de threads.

Outro ponto ao se levar em consideração é que, no computador utilizado, o processador é um i5-7600K, um quad-core, onde é altamente perceptível que para se explorar os limites do mesmo, o necessário seria apenas utilizar 4 threads. Pode-se observar no gráfico que, os tempos de execução para 8 threads é na média, igual ao tempo de execução de 4 threads.

Por fim, é possível perceber a diferença que a utilização da multiplicação paralela adiciona em desempenho, tendo resultados melhores na casa de aproximadamente 50%.