

Isadora Coelho
Leonardo Calsavara
Pedro Hernesto

INTRODUÇÃO A PROGRAMAÇÃO PARALELA

SO: Linux pop-os
Processador: core i7 8th Gen;
8 threads;
RAM: 8 GB

Para a realização da atividade, foram criados dois arquivos que computam a mesma função: a diferença do quadrado de duas matrizes, sendo um deles serial, que pode ser compilado por: `gcc MatrizConcorrente.c -o matrizc`, e um paralelizado, compilado por: `gcc -fopenmp MatrizParalela.c -o matrizp`.

O intuito da atividade era perceber a diferença do tempo de execução entre as duas modalidades do código.

Os testes no código concorrente foram realizados para matrizes de ordem 1000, 5000, 10000, 12000 a 15000, na qual a última ultrapassou o tempo médio de execução de 10 segundos.

Almejando valores consistentes, para cada ordem de matriz, foram executadas 10 vezes, e ao obter suas médias, tínhamos o tempo total de execução.

No código paralelo, o mesmo foi feito, com apenas a matriz de 15000 linhas e colunas, mas ao invés de ordem de matriz, o parâmetro era o número de threads utilizados. Todos esses valores estão armazenados na planilha “execução.csv” que pode ser encontrada na atual pasta.

Por fim, foram calculados os speedups e eficiência de cada um dos números de threads, que nos forneceram os gráficos abaixo:

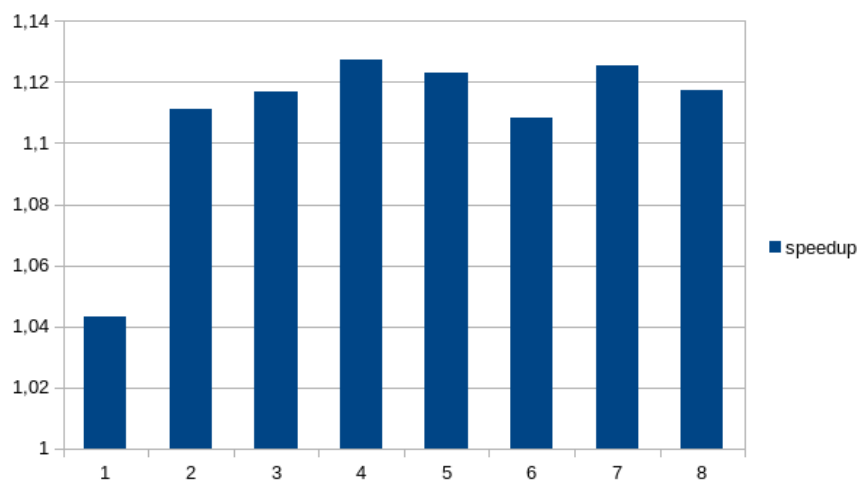


gráfico 1: speedup de 2,4,6,8,10,12,14 e 16 threads

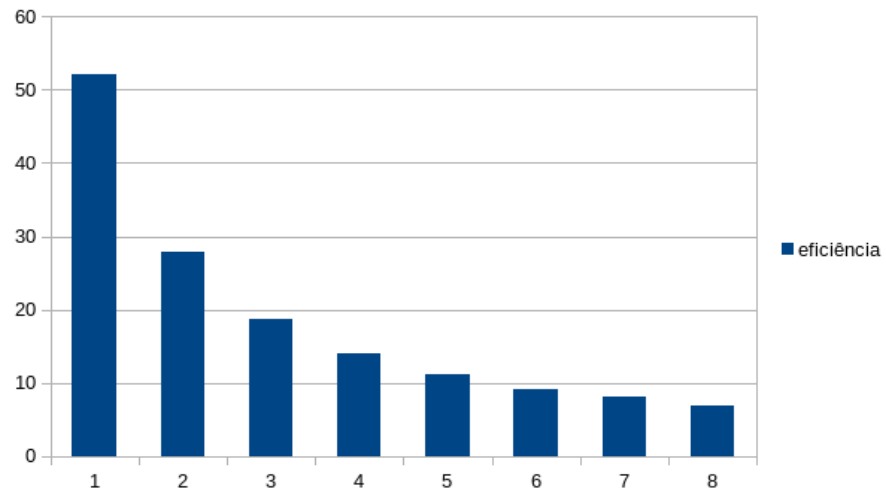


gráfico 1: eficiência de 2,4,6,8,10,12,14 e 16 threads

Observação: o número de threads não aumenta de 1 em 1 como mostra nos gráficos, e sim de 2 em 2.