

Universidade do Vale do Itajaí  
Curso de Bacharelado em Ciência da Computação  
Disciplina: Sistemas Operacionais  
Alunos: Luiz Alberto, Edson Perdoná  
Avaliação 02 – Threads e Paralelismo

Implementação em C/C++ de uma multiplicação entre matrizes utilizando o sistema single thread e multithread, no qual o último deve ser feito usando as bibliotecas Pthread, OpenMP e MPI. Análise comparativa no quesito tempo de processamento utilizando bibliotecas como time.h. A operação de multiplicação deve ser entre, no mínimo, uma matriz 2X3 e outra 3x2.

A solução foi implementada usando de um algoritmo ingênuo no qual a única real otimização é com relação as matrizes usadas que foram abstraídas em um vetor simples para tirar proveito da memória cache. Usamos da biblioteca pthread e da máquina virtual fornecida pelo professor modificada apenas para uso de 4 núcleos de processamento.

A solução consiste de 3 loops aninhados onde para o caso de single thread tudo é multiplicado e somado dentro do mesmo, tendo complexidade de  $O(n^3)$ .

Já a versão multithread abusa da não dependência de cada elemento da matriz resultado. É então instanciada uma thread, que calcula as multiplicações e as somas, para cada elemento da matriz resultado. Cada thread recebe todos os parâmetros necessários para o cálculo de um elemento da matriz final. Então executa seu loop e encerra devolvendo o elemento da matriz. A função pai espera que todas as threads tenham finalizado seus cálculos e monta a matriz resultado. Teoricamente a complexidade diminuiu de  $O(n^3)$  para  $O(n^2 + m)$ .

Por fim foi simulado várias matrizes de diferentes tamanhos e seu tempo de execução mensurado. Os resultados obtidos não são os esperados, o algoritmo simples ganha do paralelizado em uma ordem de 100 vezes menos tempo. Foram testadas matrizes esdrúxulas que beiravam o estouro de memória na sua alocação e mesmo assim o algoritmo simples venceu. Na implementação nem um dos dois casos foi levado em consideração o tempo de alocação das matrizes mas no caso do multithread foi considerado o tempo de criação de cada thread como parte do tempo mensurado.

Não sabemos porque o resultado não foi o esperado. Imaginamos que o algoritmo simples possa estar sendo otimizado pelo compilador ou pelo hardware da CPU mas esse é só um chute no escuro.

Código e screenshots disponível em: <https://github.com/lu1z/M1SO20192>.