Universidade Estadual de Campinas - UNICAMP
Faculdade de Tecnologia - FT

LABORATÓRIO I - OpenMP

GRUPO: **OPTIMUS TECH**

Integrantes:
Carolina da Silva Sancho 214376
Gabriella Carlos do Nascimento 260587

LIMEIRA, SÃO PAULO
09 DE NOVEMBRO DE 2020

SUMÁRIO COM HYPERLINKS

1. Introdução

Este laboratório tem como objetivo resolver um problema utilizando biblioteca para programação de alto desempenho (PAD), onde neste trabalho será utilizado a biblioteca `<omp.h>`, cujo código foi realizado obrigatoriamente na Linguagem C e executado no MobaXterm (para Windows).

O problema a ser resolvido neste laboratório, e nos demais, consiste em calcular a multiplicação entre matrizes e a redução pela soma dos elementos da matriz resultante dessa multiplicação. Como requisitos importantes para a realização da multiplicação das matrizes temos:

- As matrizes precisam ser alocadas dinamicamente (com o comando malloc ou equivalente).
- Todas as matrizes necessárias para os programas precisam ser alocadas em uma única etapa.

2. Compilação e Execução

1) Para compilar e executar o programa em C **sem o script** no MobaXterm ou em outro terminal digite o seguinte comando:

```
gcc -o matrizOpenMP -fopenmp matrizOpenMP.c
./matrizOpenMP y w v arqA.dat arqB.dat arqC.dat arqD.dat
```

Em que, "`matrizOpenMP.c`" é o nome do arquivo em C, "`-fopenmp`" é o comando para executar a biblioteca `<omp.h>`, "`y w v`" são as variáveis do tamanho das matrizes (`matrizA[y][w], matrizB[w][v] e matrizC[v][1]`), e "`arqA.dat arqB.dat arqC.dat arqD.dat`" são os arquivos que serão criados dentro do código com valores aleatórios, conforme quantidade determinada pelas variáveis digitadas. Abaixo, estão os exemplos dos comando com os valores das variáveis determinadas pelas especificações dos laboratório:

Para y=10, w=10 e y=10, temos:

```
gcc -o matrizOpenMP -fopenmp matrizOpenMP.c
./matrizOpenMP 10 10 10 arqA.dat arqB.dat arqC.dat arqD.dat
```

Para y=100, w=100 e y=100, temos:

```
gcc -o matrizOpenMP -fopenmp matrizOpenMP.c
./matrizOpenMP 100 100 100 arqA.dat arqB.dat arqC.dat arqD.dat
```

Para y=1000, w=1000 e y=1000, temos:

```
gcc -o matrizOpenMP -fopenmp matrizOpenMP.c
./matrizOpenMP 1000 1000 1000 arqA.dat arqB.dat arqC.dat arqD.dat
```

Como atalho, caso algum dos comandos já foi digitado, basta escrever a segunda linha, com novos valores para as variáveis, visto a seguir:

```
gcc -o matrizOpenMP -fopenmp matrizOpenMP.c
./matrizOpenMP 10 10 10 arqA.dat arqB.dat arqC.dat arqD.dat
./matrizOpenMP 100 100 100 arqA.dat arqB.dat arqC.dat arqD.dat
./matrizOpenMP 1000 1000 1000 arqA.dat arqB.dat arqC.dat arqD.dat
```

2) Para compilar e executar o programa em C **com o script** digite o seguinte comando (em que novamente "`y w v`" são as variáveis do tamanho das matrizes):

```
chmod +x matriz.sh
./matriz.sh
./matrizOpenMP y w v arqA.dat arqB.dat arqC.dat arqD.dat
```

Exemplo:

```
chmod +x matriz.sh
./matriz.sh
./matrizOpenMP 10 10 10 arqA.dat arqB.dat arqC.dat arqD.dat
```

Como atalho, para os demais valores, caso algum dos comandos já foi digitado, basta digitar novos valores para "`y w v`", como:

```
chmod +x matriz.sh
./matriz.sh
./matrizOpenMP 10 10 10 arqA.dat arqB.dat arqC.dat arqD.dat
./matrizOpenMP 100 100 100 arqA.dat arqB.dat arqC.dat arqD.dat
./matrizOpenMP 1000 1000 1000 arqA.dat arqB.dat arqC.dat arqD.dat
```

3. Análise e Gráficos

A partir da execução do código em linguagem C disponível em: https://github.com/carolsancho/Laboratorios-PAD, os tempos de processamento da multiplicação e do cálculo da redução por soma da matriz D, descontadas as operações de entrada e saída, para os valores das seguintes variáveis: y = 10, w = 10, v = 10 **(1)**; y

=100, w = 100, v = 100 **(2)**; y =1000, w = 1000, v = 1000 **(3)**, em que **não foi utilizado a biblioteca <omp.h>** tiveram os seguintes resultados:

VARIÁVEIS	TEMPO DE PROCESSAMENTO
(1)	0.07 ms
(2)	5.09 ms
(3)	6242.26 ms

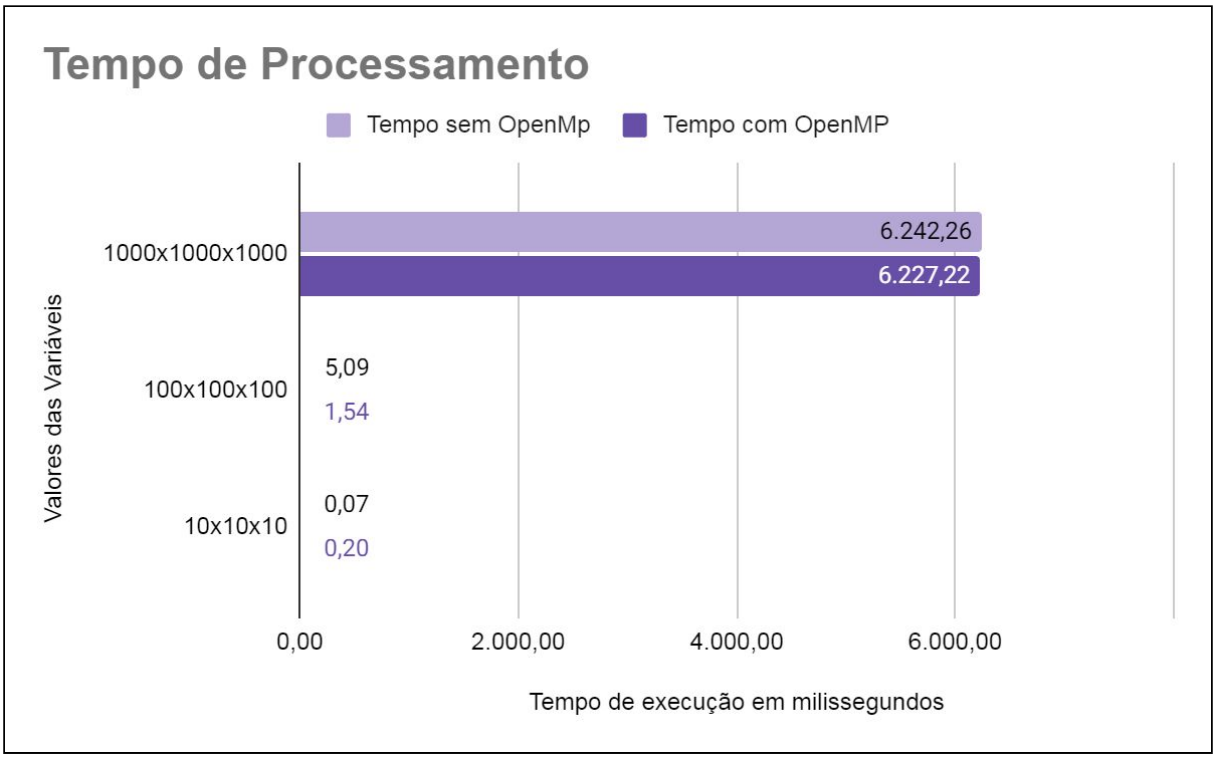
Agora, os tempos de processamento para as mesmas variáveis e **utilizando a biblioteca <omp.h>** foram obtidos os seguintes resultados:

VARIÁVEIS	TEMPO DE PROCESSAMENTO
(1)	0.20 ms
(2)	1.54ms
(3)	6227.22 ms

Assim, percebe-se que, para variáveis com valores menores quando não utilizado a biblioteca <omp.h>, como exemplificado em **(1)**, o tempo de processamento da multiplicação é inferior, se comparado com o tempo em **(1)** quando usado a biblioteca <omp.h>, o que não mostra dessa forma um bom desempenho. Porém para variáveis com valores maiores, e consequentemente com mais iterações, como observado em **(2)** e **(3)**, os tempos de processamento da multiplicação quando utilizado a biblioteca <omp.h>, apresentou significativa diminuição, se comparado com os tempos onde não foram utilizado a biblioteca <omp.h>, portanto é evidente que para operações mais complexas e com maior número de iterações, o uso dessa biblioteca melhora o desempenho da execução do código.

Abaixo apresenta-se o gráfico dos tempos de processamento da multiplicação e do cálculo da redução da soma da matriz D, utilizando ou não a biblioteca <omp.h>.

Gráfico 1 - Tempos de Processamento da multiplicação e cálculo da redução pela soma da matriz D.



Fonte: elaborado pelas autoras deste documento.

4. Conclusão

A partir do Laboratório I de OpenMP da disciplina de Tópicos Especiais em Telecomunicações I (Programação de Alto Desempenho), o grupo Optimus Tech foi capaz de ter como experiência a aplicação de conceitos estudados durante as vídeo aulas de OpenMP e pelos exemplos apresentados no repositório disponibilizado pelo professor no github, e assim correlacionando-os com a prática, sendo possível através da criação de um código em linguagem C que utilizou o OpenMP para a programação multi-processo de memória compartilhada, onde no código realizado pelo grupo utilizou-se regiões paralelas por meio de `#pragma omp parallel for firstprivate() private()`.

Portanto, através da prática foi possível ter uma melhor fixação dos conteúdos trabalhados na disciplina, além de ter explorado o emprego do OpenMP e utilizado a ferramenta MobaXterm como suporte.