



Grupo O enxuto III: A Vingança do Café - TT005 B – Programação de Alto Desempenho

Laboratório 01

Faculdade de Tecnologia da Universidade Estadual de Campinas, Limeira

Professor: Prof. Dr. André Leon S. Gradvohl

Alunos:

João Vitor D'Alkmin Basso

RA: 218927

Kauan da Silva

RA: 219594

Repositório GitHub: https://github.com/joao147/Programacao_alto_desempenho

1. Forma de compilação e execução de cada programa.

Para complicar o programa é necessário ter instalado na máquina o compilador gcc, se já estiver, abra um interpretador de linha de comando e utilize o comando "gcc -fopenmp programa.c -o programa.o"(sem aspas), "programa.c" é o código desenvolvido por nossa equipe, já o "programa.o" é

o executável que o compilador gerou, exemplo:

```
gcc -fopenmp programa.c -o programa.o
```

, deve ser executado da seguinte forma "./programa.o <valor_y> <valor_w> <valor_v> arqA.dat arqB.dat arqC.dat arqD.dat"(sem aspas), onde está "<valor_y>", "<valor_w>" e "<valor_v>" deve se inserir um valor inteiro positivo em cada um deles, de acordo com o requisito de execução que o professor solicitou, exemplo:

```
./programa.o 10 10 10 arqA.dat arqB.dat arqC.dat arqD.dat
```

2. Análises.

Agora vamos iniciar a análise a partir dos dados gerados nos testes. Todos os tempos então em Milisegundos.

Testes com os valores de $y=10$, $w=10$ e $v=10$.

O teste com matriz com poucas linhas e colunas, já era esperado que os dados seriam muito desfavoráveis para a utilização de threads, pois como foi citado nas aulas, é necessário um tempo para a criação das thread e esse tempo de criação faz um diferença muito grande, como vemos nos print dos testes e a tabela que geramos para facilitar a visualização dos dados, a média com threads é 0,135 milisegundos e a sem é 0,007 milisegundos.

Missão: Formar e aperfeiçoar cidadãos e prestar serviços atendendo às necessidades tecnológicas da sociedade com agilidade, dinâmica e qualidade

teste 0			
variáveis	valor	tempo	
y	10	com threads	sem threads
w	10	0,124	0,007
v	10	0,122	0,008
		0,141	0,008
		0,138	0,007
		0,148	0,007
		0,128	0,007
		0,14	0,007
		0,111	0,007
		0,144	0,007
		0,156	0,007
	Média	0,1352	0,0072

Tabela gerada a partir dos teste.

```
[k219594@ip-172-31-46-145 ~]$ ./p0.o 10 10 10 arqA10.dat arqB10.dat arqC10.dat arqD10.dat
Tempo para a conclusão da soma é de 0.124000, resultado 13963.456940
[k219594@ip-172-31-46-145 ~]$ ./p0.o 10 10 10 arqA10.dat arqB10.dat arqC10.dat arqD10.dat
Tempo para a conclusão da soma é de 0.122000, resultado 13963.456940
[k219594@ip-172-31-46-145 ~]$ ./p0.o 10 10 10 arqA10.dat arqB10.dat arqC10.dat arqD10.dat
Tempo para a conclusão da soma é de 0.141000, resultado 13963.456940
[k219594@ip-172-31-46-145 ~]$ ./p0.o 10 10 10 arqA10.dat arqB10.dat arqC10.dat arqD10.dat
Tempo para a conclusão da soma é de 0.138000, resultado 13963.456940
[k219594@ip-172-31-46-145 ~]$ ./p0.o 10 10 10 arqA10.dat arqB10.dat arqC10.dat arqD10.dat
Tempo para a conclusão da soma é de 0.148000, resultado 13963.456940
[k219594@ip-172-31-46-145 ~]$ ./p0.o 10 10 10 arqA10.dat arqB10.dat arqC10.dat arqD10.dat
Tempo para a conclusão da soma é de 0.128000, resultado 13963.456940
[k219594@ip-172-31-46-145 ~]$ ./p0.o 10 10 10 arqA10.dat arqB10.dat arqC10.dat arqD10.dat
Tempo para a conclusão da soma é de 0.140000, resultado 13963.456940
[k219594@ip-172-31-46-145 ~]$ ./p0.o 10 10 10 arqA10.dat arqB10.dat arqC10.dat arqD10.dat
Tempo para a conclusão da soma é de 0.111000, resultado 13963.456940
[k219594@ip-172-31-46-145 ~]$ ./p0.o 10 10 10 arqA10.dat arqB10.dat arqC10.dat arqD10.dat
Tempo para a conclusão da soma é de 0.144000, resultado 13963.456940
[k219594@ip-172-31-46-145 ~]$ ./p0.o 10 10 10 arqA10.dat arqB10.dat arqC10.dat arqD10.dat
Tempo para a conclusão da soma é de 0.156000, resultado 13963.456940
```

Teste com threads.

```
[k219594@ip-172-31-46-145 ~]$ ./p0.o 10 10 10 arqA10.dat arqB10.dat arqC10.dat arqD10.dat
Tempo para a conclusão da soma é de 0.007000, resultado 13963.456940
[k219594@ip-172-31-46-145 ~]$ ./p0.o 10 10 10 arqA10.dat arqB10.dat arqC10.dat arqD10.dat
Tempo para a conclusão da soma é de 0.008000, resultado 13963.456940
[k219594@ip-172-31-46-145 ~]$ ./p0.o 10 10 10 arqA10.dat arqB10.dat arqC10.dat arqD10.dat
Tempo para a conclusão da soma é de 0.008000, resultado 13963.456940
[k219594@ip-172-31-46-145 ~]$ ./p0.o 10 10 10 arqA10.dat arqB10.dat arqC10.dat arqD10.dat
Tempo para a conclusão da soma é de 0.007000, resultado 13963.456940
[k219594@ip-172-31-46-145 ~]$ ./p0.o 10 10 10 arqA10.dat arqB10.dat arqC10.dat arqD10.dat
Tempo para a conclusão da soma é de 0.007000, resultado 13963.456940
[k219594@ip-172-31-46-145 ~]$ ./p0.o 10 10 10 arqA10.dat arqB10.dat arqC10.dat arqD10.dat
Tempo para a conclusão da soma é de 0.007000, resultado 13963.456940
[k219594@ip-172-31-46-145 ~]$ ./p0.o 10 10 10 arqA10.dat arqB10.dat arqC10.dat arqD10.dat
Tempo para a conclusão da soma é de 0.007000, resultado 13963.456940
[k219594@ip-172-31-46-145 ~]$ ./p0.o 10 10 10 arqA10.dat arqB10.dat arqC10.dat arqD10.dat
Tempo para a conclusão da soma é de 0.007000, resultado 13963.456940
[k219594@ip-172-31-46-145 ~]$ ./p0.o 10 10 10 arqA10.dat arqB10.dat arqC10.dat arqD10.dat
Tempo para a conclusão da soma é de 0.007000, resultado 13963.456940
```

Teste sem threads.

Testes com os valores de $y=100$, $w=100$ e $v=100$

Agora os testes com matriz, com as variáveis com o valor de 100, já vemos uma diferença, o tempo de criação das thread já começa a compensar, há um ganho de aproximadamente 3 milissegundos, mas não é um ganho muito grande.

teste 1			
variáveis	valor	tempo	
y	100	com threads	sem threads
w	100	1,404	4,974
v	100	1,386	4,983
		1,705	4,976
		1,416	4,993
		1,416	4,981
		4,82	4,985
		1,438	4,985
		1,415	4,984
		1,386	4,978
		2,164	4,979
	Média	1,855	4,9818

Tabela gerada a partir dos teste.

```
[k219594@ip-172-31-46-145 ~]$ ./p0.o 100 100 100 arqA100.dat arqB100.dat arqC100.dat arqD100.dat
Tempo para a conclusão da soma é de 1.404000, resultado 192355.007812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 100 100 100 arqA100.dat arqB100.dat arqC100.dat arqD100.dat
Tempo para a conclusão da soma é de 1.386000, resultado 192355.007812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 100 100 100 arqA100.dat arqB100.dat arqC100.dat arqD100.dat
Tempo para a conclusão da soma é de 1.705000, resultado 192355.007812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 100 100 100 arqA100.dat arqB100.dat arqC100.dat arqD100.dat
Tempo para a conclusão da soma é de 1.416000, resultado 192355.007812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 100 100 100 arqA100.dat arqB100.dat arqC100.dat arqD100.dat
Tempo para a conclusão da soma é de 1.416000, resultado 192355.007812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 100 100 100 arqA100.dat arqB100.dat arqC100.dat arqD100.dat
Tempo para a conclusão da soma é de 4.820000, resultado 192355.007812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 100 100 100 arqA100.dat arqB100.dat arqC100.dat arqD100.dat
Tempo para a conclusão da soma é de 1.438000, resultado 192355.007812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 100 100 100 arqA100.dat arqB100.dat arqC100.dat arqD100.dat
Tempo para a conclusão da soma é de 1.415000, resultado 192355.007812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 100 100 100 arqA100.dat arqB100.dat arqC100.dat arqD100.dat
Tempo para a conclusão da soma é de 1.386000, resultado 192355.007812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 100 100 100 arqA100.dat arqB100.dat arqC100.dat arqD100.dat
Tempo para a conclusão da soma é de 2.164000, resultado 192355.007812
```

Teste com threads.


```
[k219594@ip-172-31-46-145 ~]$ ./p0.o 100 100 100 arqA100.dat arqB100.dat arqC100.dat arqD100.dat
Tempo para a conclusão da soma é de 4.976000, resultado 192355.007812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 100 100 100 arqA100.dat arqB100.dat arqC100.dat arqD100.dat
Tempo para a conclusão da soma é de 4.983000, resultado 192355.007812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 100 100 100 arqA100.dat arqB100.dat arqC100.dat arqD100.dat
Tempo para a conclusão da soma é de 4.976000, resultado 192355.007812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 100 100 100 arqA100.dat arqB100.dat arqC100.dat arqD100.dat
Tempo para a conclusão da soma é de 4.993000, resultado 192355.007812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 100 100 100 arqA100.dat arqB100.dat arqC100.dat arqD100.dat
Tempo para a conclusão da soma é de 4.981000, resultado 192355.007812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 100 100 100 arqA100.dat arqB100.dat arqC100.dat arqD100.dat
Tempo para a conclusão da soma é de 4.985000, resultado 192355.007812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 100 100 100 arqA100.dat arqB100.dat arqC100.dat arqD100.dat
Tempo para a conclusão da soma é de 4.985000, resultado 192355.007812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 100 100 100 arqA100.dat arqB100.dat arqC100.dat arqD100.dat
Tempo para a conclusão da soma é de 4.984000, resultado 192355.007812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 100 100 100 arqA100.dat arqB100.dat arqC100.dat arqD100.dat
Tempo para a conclusão da soma é de 4.978000, resultado 192355.007812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 100 100 100 arqA100.dat arqB100.dat arqC100.dat arqD100.dat
Tempo para a conclusão da soma é de 4.979000, resultado 192355.007812
```

Teste sem threads.

Testes com os valores de $y=1000$, $w=1000$ e $v=1000$

Agora falando de matriz com as variáveis com valor 1000, a um diferença razoável, na coleta de dados a função de tempo que usamos estava dando algum bug com as matrizes de 1000, não sabemos o porquê, mas contando com o auxílio do cronômetro de celular, vimos que demora aproximadamente 3 segundos para o programa executar com threads, já sem threads demorar aproximadamente 6 segundos exatamente como é visto nos dados. Com as threads constatado 3 segundos no cronômetro e é mostrado 6 segundos.

teste 2			
variáveis	valor	tempo	
y	1000	com threads	sem threads
w	1000	6.281,149	6247,266
v	1000	6264,828	6276,696
		6818,688	6313,122
		6770,386	6318,387
		6269,742	6.396,256
		6257,509	6.327,053
		6251,641	7295,985
		6275,443	7374,894
		6250,082	6975,675
		6263,791	6699,695
	Média	6370,3259	6622,5029

Tabela gerada a partir dos teste.

Missão: Formar e aperfeiçoar cidadãos e prestar serviços atendendo às necessidades tecnológicas da sociedade com agilidade, dinâmica e qualidade

```
[k219594@ip-172-31-46-145 ~]$ ./p0.o 1000 1000 1000 arqA1000.dat arqB1000.dat arqC1000.dat arqD1000.dat
Tempo para a conclusão da soma é de 6281.149000, resultado -522068.132812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 1000 1000 1000 arqA1000.dat arqB1000.dat arqC1000.dat arqD1000.dat
Tempo para a conclusão da soma é de 6264.828000, resultado -522068.132812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 1000 1000 1000 arqA1000.dat arqB1000.dat arqC1000.dat arqD1000.dat
Tempo para a conclusão da soma é de 6818.688000, resultado -522068.132812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 1000 1000 1000 arqA1000.dat arqB1000.dat arqC1000.dat arqD1000.dat
Tempo para a conclusão da soma é de 6770.386000, resultado -522068.132812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 1000 1000 1000 arqA1000.dat arqB1000.dat arqC1000.dat arqD1000.dat
Tempo para a conclusão da soma é de 6269.742000, resultado -522068.132812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 1000 1000 1000 arqA1000.dat arqB1000.dat arqC1000.dat arqD1000.dat
Tempo para a conclusão da soma é de 6257.509000, resultado -522068.132812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 1000 1000 1000 arqA1000.dat arqB1000.dat arqC1000.dat arqD1000.dat
Tempo para a conclusão da soma é de 6251.641000, resultado -522068.132812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 1000 1000 1000 arqA1000.dat arqB1000.dat arqC1000.dat arqD1000.dat
Tempo para a conclusão da soma é de 6275.443000, resultado -522068.132812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 1000 1000 1000 arqA1000.dat arqB1000.dat arqC1000.dat arqD1000.dat
Tempo para a conclusão da soma é de 6250.082000, resultado -522068.132812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 1000 1000 1000 arqA1000.dat arqB1000.dat arqC1000.dat arqD1000.dat
Tempo para a conclusão da soma é de 6263.791000, resultado -522068.132812
```

Teste com threads.

```
[k219594@ip-172-31-46-145 ~]$ ./p0.o 1000 1000 1000 arqA1000.dat arqB1000.dat arqC1000.dat arqD1000.dat
Tempo para a conclusão da soma é de 6247.266000, resultado -522068.132812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 1000 1000 1000 arqA1000.dat arqB1000.dat arqC1000.dat arqD1000.dat
Tempo para a conclusão da soma é de 6276.696000, resultado -522068.132812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 1000 1000 1000 arqA1000.dat arqB1000.dat arqC1000.dat arqD1000.dat
Tempo para a conclusão da soma é de 6313.122000, resultado -522068.132812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 1000 1000 1000 arqA1000.dat arqB1000.dat arqC1000.dat arqD1000.dat
Tempo para a conclusão da soma é de 6318.387000, resultado -522068.132812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 1000 1000 1000 arqA1000.dat arqB1000.dat arqC1000.dat arqD1000.dat
Tempo para a conclusão da soma é de 6396.256000, resultado -522068.132812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 1000 1000 1000 arqA1000.dat arqB1000.dat arqC1000.dat arqD1000.dat
Tempo para a conclusão da soma é de 6327.053000, resultado -522068.132812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 1000 1000 1000 arqA1000.dat arqB1000.dat arqC1000.dat arqD1000.dat
Tempo para a conclusão da soma é de 7295.985000, resultado -522068.132812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 1000 1000 1000 arqA1000.dat arqB1000.dat arqC1000.dat arqD1000.dat
Tempo para a conclusão da soma é de 7374.894000, resultado -522068.132812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 1000 1000 1000 arqA1000.dat arqB1000.dat arqC1000.dat arqD1000.dat
Tempo para a conclusão da soma é de 6975.675000, resultado -522068.132812
[k219594@ip-172-31-46-145 ~]$ ./p0.o 1000 1000 1000 arqA1000.dat arqB1000.dat arqC1000.dat arqD1000.dat
Tempo para a conclusão da soma é de 6699.695000, resultado -522068.132812
```

Teste sem threads.

3. Conclusão.

Com base nos testes e na experiência de desenvolvimento, desenvolver um programa sem thread é logicamente mais fácil, visto que threads só começam a gerar diferença no desempenho quando a aplicação irá lidar com uma grande quantidade de dados. Já falando sobre threads, a biblioteca OpenMP, facilita a utilização de threads, mas ainda é mais complicado do que criar um programa sem, só é necessário a utilização de threads caso o programa precise trabalhar com um grande quantidade de dados. Concluímos que dependendo da aplicação, é lógico a utilização de threads, porém quando se trata de programas simples e sem muitas operações é desnecessário a sua utilização.

4. Referências.

https://github.com/gradvohl/TT005_Laboratorios

<https://gist.github.com/giucortes/2a50ac91bbe49b20339b0de80b0831fd>

<https://www.clubedohardware.com.br/topic/515203-multipl%C3%A7%C3%A3o-de-matrizes-em-c/>

<https://github.com/kauan1/forasteiros>