

Disciplina: Computacao Concorrente

Prof.: Silvana Rossetto

Módulo 1 - Laboratório: 2

Aluno: Gabriel Silva Pereira

- **Objetivo**

Projetar e implementar uma versão concorrente para o problema de multiplicação de matrizes; e avaliar o desempenho da aplicação em termos de tempo de execução. Usaremos a linguagem C e a biblioteca Pthreads.

- **Informações relevantes obtido do comando `lscpu` no linux:**

CPU(s): 4

Thread(s) per núcleo: 2

Núcleo(s) por soquete: 2

Nome do modelo: Intel(R) Core(TM) i5-3337U CPU @ 1.80GHz

CPU MHz máx.: 2700,0000

CPU MHz mín.: 800,0000

cache de L1d: 64 KiB

cache de L1i: 64 KiB

cache de L2: 512 KiB

cache de L3: 3 MiB

- **Dados puros obtidos**

```
$ ./mul_mat.out 500 1
Tempo inicializacao:0.003204
Tempo multiplicacao:1.681533
Tempo finalizacao:0.000351
```

total = 1.685088

```
$ ./mul_mat.out 500 2
Tempo inicializacao:0.003153
Tempo multiplicacao:0.837688
Tempo finalizacao:0.000473
```

total = 0.841314

```
$ ./mul_mat.out 500 4
Tempo inicializacao:0.003705
Tempo multiplicacao:0.776699
Tempo finalizacao:0.000250
```

total = 0.780654

```
$ ./mul_mat.out 1000 1
```

```
Tempo inicializacao:0.013521
Tempo multiplicacao:17.811788
Tempo finalizacao:0.000782
```

total = 17.826091

```
$ ./mul_mat.out 1000 2
Tempo inicializacao:0.013768
Tempo multiplicacao:8.842847
Tempo finalizacao:0.001473
```

total = 8.858088

```
$ ./mul_mat.out 1000 4
Tempo inicializacao:0.014161
Tempo multiplicacao:7.909425
Tempo finalizacao:0.000875
```

total = 7.924461

```
$ ./mul_mat.out 2000 1
Tempo inicializacao:0.054755
Tempo multiplicacao:141.023186
Tempo finalizacao:0.003438
```

total = 141.081379

```
$ ./mul_mat.out 2000 2
Tempo inicializacao:0.052742
Tempo multiplicacao:69.732474
Tempo finalizacao:0.003349
```

total = 69.788565

```
$ ./mul_mat.out 2000 4
Tempo inicializacao:0.051573
Tempo multiplicacao:62.376708
Tempo finalizacao:0.004117
```

total = 62.432398

- Dados tabelados para comparação

- Comparação com 2 threads

Tamanho da Matriz	1 Thread	2 Threads	Ganho de desempenho
500x500	1.685088	0.841314	2.158559
1000x1000	17.826091	8.858088	2.012408
2000x2000	141.081379	69.788565	2.021554

- Comparação com 4 threads

Tamanho da Matriz	1 Thread	4 Threads	Ganho de desempenho
500x500	1.685088	0.780654	2.158559
1000x1000	17.826091	7.924461	2.249502
2000x2000	141.081379	62.432398	2.259746

- Comparação entre 2 e 4 threads

Tamanho da Matriz	2 Threads	4 Threads	Ganho de desempenho
500x500	0.841314	0.780654	1.077704
1000x1000	8.858088	7.924461	1.117815
2000x2000	69.788565	62.432398	1.117826

- Comentários sobre os resultados

Os resultados obtidos estão dentro da expectativa de melhora no desempenho. Comparando o desempenho de 2 a 4 threads, houve um ganho de 7 a 11%. O processador utilizado tem na prática 2 núcleos de processamento, e 2 núcleos virtualizados. A tecnologia de *Hyper Threading* (<https://www.hp.com/us-en/shop/tech-takes/hyper-threading-everything-to-know>), em média tal tecnologia aumenta de 15 a 30% de ganho no processamento. Esse processador tem as caches de nível 1, 2 e 3 com 64 KiB, 512KiB e 3MiB respectivamente. Dado a configuração do computador e o quanto a tecnologia de *Hyper Threading* ajuda na melhora do desempenho, os resultados são bem satisfatórios nessas condições apresentadas acima.