

Nome: João Vitor de Freitas Barbosa
DRE: 117055449

Computação Concorrente

Laboratório 2 - Relatório de desempenho

Primeiramente vamos começar os testes com a aplicação sequencial para ver como o programa se comporta.

- **Multiplicação de duas matrizes 500x500:**

```
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1Sequencial 500 sequencial500
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.00520021
Tempo multiplicacao de duas matrizes 500x500 sequencial: 0.68701905
Tempo finalizacoes: 0.00000095
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1Sequencial 500 sequencial500
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.00395819
Tempo multiplicacao de duas matrizes 500x500 sequencial: 0.69249379
Tempo finalizacoes: 0.00000084
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1Sequencial 500 sequencial500
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.00457694
Tempo multiplicacao de duas matrizes 500x500 sequencial: 0.70156178
Tempo finalizacoes: 0.00000075
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1Sequencial 500 sequencial500
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.00749585
Tempo multiplicacao de duas matrizes 500x500 sequencial: 0.68622743
Tempo finalizacoes: 0.00000068
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$
```

Tomando o menor tempo como base, temos 0.686 o tempo da multiplicação de duas matrizes quadradas 500x500 feita de forma sequencial

- **Multiplicação sequencial de duas matrizes 1000x1000:**

```
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1Sequencial 1000 sequencial1000
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.01480943
Tempo multiplicacao de duas matrizes 1000x1000 sequencial: 9.86952860
Tempo finalizacoes: 0.00000091
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1Sequencial 1000 sequencial1000
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.01245498
Tempo multiplicacao de duas matrizes 1000x1000 sequencial: 9.92207804
Tempo finalizacoes: 0.00000105
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1Sequencial 1000 sequencial1000
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.01439177
Tempo multiplicacao de duas matrizes 1000x1000 sequencial: 10.29199552
Tempo finalizacoes: 0.00000089
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1Sequencial 1000 sequencial1000
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.00984764
Tempo multiplicacao de duas matrizes 1000x1000 sequencial: 9.60350746
Tempo finalizacoes: 0.00000123
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1Sequencial 1000 sequencial1000
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.01477606
Tempo multiplicacao de duas matrizes 1000x1000 sequencial: 8.37380064
Tempo finalizacoes: 0.00000090
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$
```

Tomando novamente o menor tempo como base, temos 8.37 o tempo da multiplicação de duas matrizes quadradas 1000x1000 feita de forma sequencial

- **Multiplicação sequencial de duas matrizes 2000x2000:**

```

Tempo inicializacoes: 0.00000090
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1Sequencial 2000 sequencial2000
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.03858942
Tempo multiplicacao de duas matrizes 2000x2000 sequencial: 103.54510413
Tempo finalizacoes: 0.00000110
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1Sequencial 2000 sequencial2000
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.04174939
Tempo multiplicacao de duas matrizes 2000x2000 sequencial: 104.63605165
Tempo finalizacoes: 0.00000099
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1Sequencial 2000 sequencial2000
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.04179068
Tempo multiplicacao de duas matrizes 2000x2000 sequencial: 106.83602179
Tempo finalizacoes: 0.00000117
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$

```

Tomando como base o menor tempo, temos 103.54 o tempo da multiplicação de duas matrizes quadradas 2000x2000 feita de forma concorrente

Mesmo sabendo que executar o programa com 1 thread seria equivalente a um programa sequencial, criei um que não precisasse ter como parâmetro threads no momento da execução. E os resultados foram bem semelhantes aos que foram obtidos com uma thread.

Agora, vamos analisar os resultados para a multiplicação feita de forma concorrente:

- **Multiplicação concorrente de duas matrizes 500x500:**

1 thread:

```

Tempo inicializacoes: 0.00000110
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 500 1 concorrente500
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.00223732
Tempo multiplicacao de duas matrizes 500x500 com 1 threads: 0.69060144
Tempo finalizacoes: 0.00000119
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 500 1 concorrente500
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.00318985
Tempo multiplicacao de duas matrizes 500x500 com 1 threads: 0.69480426
Tempo finalizacoes: 0.00000089
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 500 1 concorrente500
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.00466740
Tempo multiplicacao de duas matrizes 500x500 com 1 threads: 0.69075523
Tempo finalizacoes: 0.00000083
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 500 1 concorrente500
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.00553147
Tempo multiplicacao de duas matrizes 500x500 com 1 threads: 0.68385260
Tempo finalizacoes: 0.00000110
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$

```

Tomando como base o menor tempo obtido, temos 0.683 o tempo da multiplicação de duas matrizes quadradas 500x500 com 1 thread

2 threads:

```
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 500 2 concorrente500
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.00322050
Tempo multiplicacao de duas matrizes 500x500 com 2 threads: 0.36109487
Tempo finalizacoes: 0.00000100
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 500 2 concorrente500
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.00544022
Tempo multiplicacao de duas matrizes 500x500 com 2 threads: 0.35015155
Tempo finalizacoes: 0.00000094
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 500 2 concorrente500
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.00247107
Tempo multiplicacao de duas matrizes 500x500 com 2 threads: 0.36780114
Tempo finalizacoes: 0.00000120
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 500 2 concorrente500
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.00505672
Tempo multiplicacao de duas matrizes 500x500 com 2 threads: 0.35806667
Tempo finalizacoes: 0.00000112
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$
```

Tomando como base o menor tempo obtido, temos 0.3501 o tempo da multiplicação de duas matrizes quadradas 500x500 com 2 threads

4 threads:

```
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 500 4 concorrente500
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.00235389
Tempo multiplicacao de duas matrizes 500x500 com 4 threads: 0.31394754
Tempo finalizacoes: 0.00000095
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 500 4 concorrente500
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.00234583
Tempo multiplicacao de duas matrizes 500x500 com 4 threads: 0.27562313
Tempo finalizacoes: 0.00000106
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 500 4 concorrente500
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.00456299
Tempo multiplicacao de duas matrizes 500x500 com 4 threads: 0.20304801
Tempo finalizacoes: 0.00000110
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 500 4 concorrente500
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.00524751
Tempo multiplicacao de duas matrizes 500x500 com 4 threads: 0.20617949
Tempo finalizacoes: 0.00000128
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 500 4 concorrente500
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.00423527
Tempo multiplicacao de duas matrizes 500x500 com 4 threads: 0.20964173
Tempo finalizacoes: 0.00000102
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$
```

Tomando como base o menor tempo obtido, temos 0.275 o tempo da multiplicação de duas matrizes quadradas 500x500 com 4 threads

- Multiplicação concorrente de duas matrizes 1000x1000:

1 thread:

```
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 1000 1 concorrente1000
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.01449464
Tempo multiplicacao de duas matrizes 1000x1000 com 1 threads: 8.41695227
Tempo finalizacoes: 0.00000126
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 1000 1 concorrente1000
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.01567851
Tempo multiplicacao de duas matrizes 1000x1000 com 1 threads: 8.51267783
Tempo finalizacoes: 0.00000102
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 1000 1 concorrente1000
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.01642755
Tempo multiplicacao de duas matrizes 1000x1000 com 1 threads: 8.23920378
Tempo finalizacoes: 0.00000114
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 1000 1 concorrente1000
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.01618592
Tempo multiplicacao de duas matrizes 1000x1000 com 1 threads: 8.07363145
Tempo finalizacoes: 0.00000116
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$
```

Tomando como base o menor tempo obtido, temos 8.0736 o tempo da multiplicação de duas matrizes quadradas 1000x1000 com 1 thread

2 threads:

```
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 1000 2 concorrente1000
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.01267676
Tempo multiplicacao de duas matrizes 1000x1000 com 2 threads: 4.16350232
Tempo finalizacoes: 0.00000089
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 1000 2 concorrente1000
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.01420638
Tempo multiplicacao de duas matrizes 1000x1000 com 2 threads: 4.31490972
Tempo finalizacoes: 0.00000123
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 1000 2 concorrente1000
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.01536750
Tempo multiplicacao de duas matrizes 1000x1000 com 2 threads: 4.17769068
Tempo finalizacoes: 0.00000102
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 1000 2 concorrente1000
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.00937998
Tempo multiplicacao de duas matrizes 1000x1000 com 2 threads: 4.18966078
Tempo finalizacoes: 0.00000110
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$
```

Tomando como base o menor tempo obtido, temos 4.163 o tempo da multiplicação de duas matrizes quadradas 1000x1000 com 2 threads

4 threads:

```
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 1000 4 concorrente1000
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.00861039
Tempo multiplicacao de duas matrizes 1000x1000 com 4 threads: 2.49509004
Tempo finalizacoes: 0.00000109
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 1000 4 concorrente1000
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.01631734
Tempo multiplicacao de duas matrizes 1000x1000 com 4 threads: 2.42865007
Tempo finalizacoes: 0.00000114
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 1000 4 concorrente1000
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.00924899
Tempo multiplicacao de duas matrizes 1000x1000 com 4 threads: 2.66105259
Tempo finalizacoes: 0.00000117
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 1000 4 concorrente1000
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.00960613
Tempo multiplicacao de duas matrizes 1000x1000 com 4 threads: 2.43928150
Tempo finalizacoes: 0.00000117
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$
```

Tomando como base o menor tempo obtido, temos 2.439 o tempo da multiplicação de duas matrizes quadradas 1000x1000 com 4 threads

- **Multiplicação concorrente de duas matrizes 2000x2000:**

1 thread:

```
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 2000 1 concorrente2000
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.03740659
Tempo multiplicacao de duas matrizes 2000x2000 com 1 threads: 111.15151371
Tempo finalizacoes: 0.00000157
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 2000 1 concorrente2000
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.03720119
Tempo multiplicacao de duas matrizes 2000x2000 com 1 threads: 106.29361173
Tempo finalizacoes: 0.00000124
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 2000 1 concorrente2000
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.04056073
Tempo multiplicacao de duas matrizes 2000x2000 com 1 threads: 108.18262103
Tempo finalizacoes: 0.00000163
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$
```

Tomando como base o menor tempo obtido, temos 106.29 o tempo da multiplicação de duas matrizes quadradas 2000x2000 com 1 thread

2 threads:

```
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 2000 2 concorrente2000
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.04071311
Tempo multiplicacao de duas matrizes 2000x2000 com 2 threads: 51.67513354
Tempo finalizacoes: 0.00000185
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 2000 2 concorrente2000
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.04157891
Tempo multiplicacao de duas matrizes 2000x2000 com 2 threads: 52.36574581
Tempo finalizacoes: 0.00000118
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 2000 2 concorrente2000
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.03870699
Tempo multiplicacao de duas matrizes 2000x2000 com 2 threads: 52.64710785
Tempo finalizacoes: 0.00000136
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ █
```

Tomando como base o menor tempo obtido, temos 51.67 o tempo da multiplicação de duas matrizes quadradas 2000x2000 com 2 threads

4 threads:

```
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 2000 4 concorrente2000
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.04416370
Tempo multiplicacao de duas matrizes 2000x2000 com 4 threads: 27.77237682
Tempo finalizacoes: 0.00000132
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 2000 4 concorrente2000
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.04231704
Tempo multiplicacao de duas matrizes 2000x2000 com 4 threads: 26.46139237
Tempo finalizacoes: 0.00000113
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 2000 4 concorrente2000
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.03937139
Tempo multiplicacao de duas matrizes 2000x2000 com 4 threads: 26.70661646
Tempo finalizacoes: 0.00000109
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ ./atividade1 2000 4 concorrente2000
Numero de processadores: 4
Preenchimento das matrizes feito com sucesso
Tempo inicializacoes: 0.04034272
Tempo multiplicacao de duas matrizes 2000x2000 com 4 threads: 30.41507401
Tempo finalizacoes: 0.00000113
joao@joao-All-Series:~/Documentos/GitHub/comp_conc/laboratorio2/laboratorio2_atividade$ █
```

Tomando como base o menor tempo obtido, temos 26.461 o tempo da multiplicação de duas matrizes quadradas 2000x2000 com 4 threads

Sabemos que a parte das inicializações também devem ser consideradas, no entanto, como foram basicamente idênticas para cada caso, podemos considerar o tempo da multiplicação onde se observa a maior discrepância de tempo

Agora vamos calcular o ganho de desempenho (aceleração) obtido com a versão concorrente. Tomando como resultado os valores que foram obtidos na versão sequencial ao invés da executada com 1 thread pois no geral retornaram valores menores.

$$\frac{T_{sequencial}}{T_{concorrente}}$$

Temos então:

Tempos	Sequencial	1 thread	2 threads	4 threads
500x500	0.686	0.683	0.3501	0.275
1000x1000	8.37	8.0736	4.163	2.439
2000x2000	103.54	106.29	51.67	26.461

Ganhos	1 thread	2 threads	4 threads
500x500	1.0043	1.9594	2.4945
1000x1000	1.0367	2.0105	3.4317
2000x2000	0.9741	2.0038	3.9129