**MULTIPLICAÇÃO DE MATRIZES – CAIO VINICIUS CORSINI FILHO – 10342005**

**Compilações dos códigos:**

```
caio_corsini@LAPTOP-DJJBCNPI:~/CompParalel$ cp "/mnt/c/Users/caiof/Documentos disco loca
l/aaaComputacao_faculdade/Computação paralela/Labs/Lab3 - MultiplicandoMatrizes/ex1Matri
zesLinhaColuna.cpp" ~/CompParalel/ && \
g++ ~/CompParalel/ex1MatrizesLinhaColuna.cpp -o ~/CompParalel/ex1MatrizesLinhaColuna
caio_corsini@LAPTOP-DJJBCNPI:~/CompParalel$ cp "/mnt/c/Users/caiof/Documentos disco loca
l/aaaComputacao_faculdade/Computação paralela/Labs/Lab3 - MultiplicandoMatrizes/ex1Matri
zesColunaLinha.cpp" ~/CompParalel/ && \
g++ ~/CompParalel/ex1MatrizesColunaLinha.cpp -o ~/CompParalel/ex1MatrizesColunaLinha
```

Figura 1: compilação dos códigos do exercício 1

```
caio_corsini@LAPTOP-DJJBCNPI:~/CompParalel$ cp "/mnt/c/Users/caiof/Documentos disco local/aaaCom
putacao_faculdade/Computação paralela/Labs/Lab3 - MultiplicandoMatrizes/ex2MatrizesBlocagem.cpp"
 ~/CompParalel/ && \
g++ ~/CompParalel/ex2MatrizesBlocagem.cpp -O0 -fno-tree-vectorize -fno-inline -o ~/CompParalel/e
x2MatrizesBlocagem
```

Figura 2: Compilação não otimizada da blocagem

```
caio_corsini@LAPTOP-DJJBCNPI:~/CompParalel$ cp "/mnt/c/Users/caiof/Documentos disco local/aaaCom
putacao_faculdade/Computação paralela/Labs/Lab3 - MultiplicandoMatrizes/ex2MatrizesBlocagem.cpp"
 ~/CompParalel/ && \
g++ ~/CompParalel/ex2MatrizesBlocagem.cpp -O3 -fno-tree-vectorize -fno-inline -o ~/CompParalel/e
x2MatrizesBlocagem
```

Figura 2: Compilação otimizada da blocagem

g++ ~/CompParalel/ex1MatrizesLinhaColuna.cpp -o ~/CompParalel/ex1MatrizesLinhaColuna

g++ ~/CompParalel/ex1MatrizesColunaLinha.cpp -o ~/CompParalel/ex1MatrizesColunaLinha

g++ ~/CompParalel/ex2MatrizesBlocagem.cpp -O0 -fno-tree-vectorize -fno-inline -o ~/CompParalel/ex2MatrizesBlocagem

g++ ~/CompParalel/ex2MatrizesBlocagem.cpp -O3 -fno-tree-vectorize -fno-inline -o ~/CompParalel/ex2MatrizesBlocagem

Pelo que entendi pelo enunciado, era o exercício 2 que tinha que executar sem as otimizações.

**Exercício 1:**

```
caio_corsini@LAPTOP-DJJBCNPI:~/CompParalel$ ./ex1MatrizesColunaLinha

Tamanho da matriz quadrada: 96
Tamanho escolhido: 96


Duracao (milissegundos): 126
caio_corsini@LAPTOP-DJJBCNPI:~/CompParalel$ ./ex1MatrizesLinhaColuna

Tamanho da matriz quadrada: 96
Tamanho escolhido: 96


Duracao (milissegundos): 127
```

Figura 3: Execuções dos coluna linha e linha coluna sem o valgrind

Exercício 2:

```
caio_corsini@LAPTOP-DJJBCNPI:~/CompParalel$ ./ex2MatrizesBlocagem

Tamanho da matriz quadrada: 480
Tamanho escolhido: 480


Duracao (milissegundos): 7203
```

Figura 4: Figura: Execução com blocagem não otimizada sem o valgrind

```
caio_corsini@LAPTOP-DJJBCNPI:~/CompParalel$ ./ex2MatrizesBlocagem

Tamanho da matriz quadrada: 480
Tamanho escolhido: 480


Duracao (milissegundos): 814
```

Figura 5: Execução com blocagem otimizada sem o valgrind

**Exercício 3:**

Com base na análise do valgrind é possível observar que a versão Coluna linha é a menos eficiente de todas em termos de uso de cache pois ela teve quase o dobro de misses comparado às outras duas versões. Linha coluna e blocagem tiveram resultados mais eficientes e muito parecidos, sendo que linha coluna teve um pouco menos de misses e blocagem teve menos D refs e demorou menos tempo. Isso se deve ao fato que, na memória, as matrizes são organizadas em linhas, logo, pelo principio da localidade, é mais eficiente iterar linhas e colunas do que colunas linhas. Pode ser que a remoção das otimizações na compilação do código com blocagem seja o que causou a menor eficiência. Com otimização, teve uma redução significativa de tempo e D refs no blocagem.

```
caio_corsini@LAPTOP-DJJBCNPI:~/CompParalel$ valgrind --tool=cachegrind ./ex1MatrizesColunaLinha
==45574== Cachegrind, a cache and branch-prediction profiler
==45574== Copyright (C) 2002-2017, and GNU GPL'd, by Nicholas Nethercote et al.
==45574== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==45574== Command: ./ex1MatrizesColunaLinha
==45574==
--45574-- warning: L3 cache found, using its data for the LL simulation.

Tamanho da matriz quadrada: 96
Tamanho escolhido: 96


Duracao (milissegundos): 1269
==45574==
==45574== I   refs:      260,258,500
==45574== I1  misses:          2,830
==45574== LLi misses:          2,508
==45574== I1  miss rate:        0.00%
==45574== LLi miss rate:        0.00%
==45574==
==45574== D   refs:      130,204,567  (81,237,906 rd   + 48,966,661 wr)
==45574== D1  misses:         47,965  (    43,472 rd   +      4,493 wr)
==45574== LLd misses:         11,308  (     7,683 rd   +      3,625 wr)
==45574== D1  miss rate:         0.0% (       0.1%     +        0.0%  )
==45574== LLd miss rate:         0.0% (       0.0%     +        0.0%  )
==45574==
==45574== LL refs:            50,795  (    46,302 rd   +      4,493 wr)
==45574== LL misses:          13,816  (    10,191 rd   +      3,625 wr)
==45574== LL miss rate:          0.0% (       0.0%     +        0.0%  )
```

Figura 6: Valgrind do coluna linha

```
caio_corsini@LAPTOP-DJJBCNPI:~/CompParalel$ valgrind --tool=cachegrind ./ex1MatrizesLinhaColuna
==45648== Cachegrind, a cache and branch-prediction profiler
==45648== Copyright (C) 2002-2017, and GNU GPL'd, by Nicholas Nethercote et al.
==45648== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==45648== Command: ./ex1MatrizesLinhaColuna
==45648==
--45648-- warning: L3 cache found, using its data for the LL simulation.

Tamanho da matriz quadrada: 96
Tamanho escolhido: 96


Duracao (milissegundos): 1362
==45648==
==45648== I   refs:      260,258,447
==45648== I1  misses:          2,825
==45648== LLi misses:          2,505
==45648== I1  miss rate:        0.00%
==45648== LLi miss rate:        0.00%
==45648==
==45648== D   refs:      130,204,546  (81,237,893 rd   + 48,966,653 wr)
==45648== D1  misses:         20,427  (    15,933 rd   +      4,494 wr)
==45648== LLd misses:         11,308  (     7,683 rd   +      3,625 wr)
==45648== D1  miss rate:         0.0% (       0.0%     +        0.0%  )
==45648== LLd miss rate:         0.0% (       0.0%     +        0.0%  )
==45648==
==45648== LL refs:            23,252  (    18,758 rd   +      4,494 wr)
==45648== LL misses:          13,813  (    10,188 rd   +      3,625 wr)
==45648== LL miss rate:          0.0% (       0.0%     +        0.0%  )
```

Figura 7: Valgrind do linha coluna

```
caio_corsini@LAPTOP-DJJBCNPI:~/CompParalel$ valgrind --tool=cachegrind ./ex2MatrizesBlocagem
==45034== Cachegrind, a cache and branch-prediction profiler
==45034== Copyright (C) 2002-2017, and GNU GPL'd, by Nicholas Nethercote et al.
==45034== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==45034== Command: ./ex2MatrizesBlocagem
==45034==
--45034-- warning: L3 cache found, using its data for the LL simulation.

Tamanho da matriz quadrada: 96
Tamanho escolhido: 96


Duracao (milissegundos): 892
==45034==
==45034== I   refs:      179,974,202
==45034== I1  misses:          2,814
==45034== LLi misses:          2,505
==45034== I1  miss rate:        0.00%
==45034== LLi miss rate:        0.00%
==45034==
==45034== D   refs:       97,955,677  (63,161,030 rd   + 34,794,647 wr)
==45034== D1  misses:         23,738  (    19,254 rd   +      4,484 wr)
==45034== LLd misses:         11,297  (     7,678 rd   +      3,619 wr)
==45034== D1  miss rate:         0.0% (       0.0%     +        0.0%  )
==45034== LLd miss rate:         0.0% (       0.0%     +        0.0%  )
==45034==
==45034== LL refs:            26,552  (    22,068 rd   +      4,484 wr)
==45034== LL misses:          13,802  (    10,183 rd   +      3,619 wr)
==45034== LL miss rate:          0.0% (       0.0%     +        0.0%  )
caio_corsini@LAPTOP-DJJBCNPI:~/CompParalel$
```

Figura 8: Valgrind do blocagem sem otimização

```
caio_corsini@LAPTOP-DJJBCNPI:~/CompParalel$ valgrind --tool=cachegrind ./ex2MatrizesBlocagem
==49228== Cachegrind, a cache and branch-prediction profiler
==49228== Copyright (C) 2002-2017, and GNU GPL'd, by Nicholas Nethercote et al.
==49228== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==49228== Command: ./ex2MatrizesBlocagem
==49228==
--49228-- warning: L3 cache found, using its data for the LL simulation.

Tamanho da matriz quadrada: 96
Tamanho escolhido: 96


Duracao (milissegundos): 123
==49228==
==49228== I   refs:      14,614,132
==49228== I1  misses:         2,717
==49228== LLi misses:         2,460
==49228== I1  miss rate:       0.02%
==49228== LLi miss rate:       0.02%
==49228==
==49228== D   refs:      10,687,062  (8,762,943 rd   + 1,924,119 wr)
==49228== D1  misses:        23,732  (   19,240 rd   +     4,492 wr)
==49228== LLd misses:        11,299  (    7,681 rd   +     3,618 wr)
==49228== D1  miss rate:        0.2% (      0.2%     +       0.2%  )
==49228== LLd miss rate:        0.1% (      0.1%     +       0.2%  )
==49228==
==49228== LL refs:           26,449  (   21,957 rd   +     4,492 wr)
==49228== LL misses:         13,759  (   10,141 rd   +     3,618 wr)
==49228== LL miss rate:         0.1% (      0.0%     +       0.2%  )
```

Figura 9: Valgrind do blocagem com otimização