



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Nome: Luís Fernando de Mesquita Pereira

RA: 10410686

1- Crie dois executáveis para a parte de multiplicação de matrizes: um que percorre em ordem de linha e outro que percorre em ordem de coluna.

a) Implementação para percorrer a matriz em ordem de linha

```
luisf@Luis MINGW64 ~/OneDrive/Área de Trabalho/Faculdade/threads
• $ g++ linhas_matriz.cpp -o linhas_matriz

luisf@Luis MINGW64 ~/OneDrive/Área de Trabalho/Faculdade/threads
• $ ./linhas_matriz.exe
Tempo de execução: 14534762 microsegundos
```

b) Implementação para percorrer a matriz em ordem de coluna

```
luisf@Luis MINGW64 ~/OneDrive/Área de Trabalho/Faculdade/threads
• $ g++ colunas_matriz.cpp -o colunas_matriz

luisf@Luis MINGW64 ~/OneDrive/Área de Trabalho/Faculdade/threads
• $ ./colunas_matriz.exe
Tempo de execucao: 21414643 microsegundos
```

2 - Crie um terceiro executável para utilizar corretamente o cache (hierarquia de memória):

a) desligando todas as otimizações do compilador, como indicado acima.

i) Com block size de 32

```
luisf@Luis MINGW64 ~/OneDrive/Email attachments/Documentos/Linguagem C/threads
• $ g++ -O0 -fno-tree-vectorize -fno-inline blocking.cpp -o blocking

luisf@Luis MINGW64 ~/OneDrive/Email attachments/Documentos/Linguagem C/threads
• $ ./blocking
Tempo de execucao: 14005504 microsegundos
```

b) ligando a otimização máxima.

```
luisf@Luis MINGW64 ~/OneDrive/Email attachments/Documentos/Linguagem C/threads
• $ g++ -O3 -funroll-loops -march=native -o blocking_otm blocking.cpp

luisf@Luis MINGW64 ~/OneDrive/Email attachments/Documentos/Linguagem C/threads
• $ ./blocking_otm
Tempo de execucao: 725843 microsegundos
```



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática



3 - Valgrind

- Percorrendo matriz por linha

```
luis@Ubuntu-2-25:/media/sf_Linguagem_C/threads$ valgrind --tool=cachegrind ./linhas_matriz
==19594== Cachegrind, a cache and branch-prediction profiler
==19594== Copyright (C) 2002-2017, and GNU GPL'd, by Nicholas Nethercote et al.
==19594== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==19594== Command: ./linhas_matriz
==19594==
--19594-- Warning: Cannot auto-detect cache config, using defaults.
--19594-- Run with -v to see.
==19594== brk segment overflow in thread #1: can't grow to 0x485b000
==19594== (see section Limitations in user manual)
==19594== NOTE: further instances of this message will not be shown
Tempo de execucao: 325632599 microsegundos
==19594==
==19594== I refs:      126,015,566,827
==19594== I1 misses:      2,426
==19594== L1i misses:      2,265
==19594== I1 miss rate:      0.00%
==19594== L1i miss rate:      0.00%
==19594==
==19594== D refs:      66,008,376,403 (41,005,499,146 rd + 25,002,877,257 wr)
==19594== D1 misses:      1,154,066,945 ( 1,153,874,760 rd +      192,185 wr)
==19594== L1d misses:      63,081,850 ( 62,890,597 rd +      191,253 wr)
==19594== D1 miss rate:      1.7% (      2.8% +      0.0% )
==19594== L1d miss rate:      0.1% (      0.2% +      0.0% )
==19594==
==19594== LL refs:      1,154,069,371 ( 1,153,877,186 rd +      192,185 wr)
==19594== LL misses:      63,084,115 ( 62,892,862 rd +      191,253 wr)
==19594== LL miss rate:      0.0% (      0.0% +      0.0% )
```



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática



- Percorrendo matriz por coluna:

```
luis@Ubuntu-2-25:/media/sf_Linguagem_C/threads$ valgrind --tool=cachegrind ./colunas_matriz
==19890== Cachegrind, a cache and branch-prediction profiler
==19890== Copyright (C) 2002-2017, and GNU GPL'd, by Nicholas Nethercote et al.
==19890== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==19890== Command: ./colunas_matriz
==19890==
--19890-- Warning: Cannot auto-detect cache config, using defaults.
--19890-- Run with -v to see.
==19890== brk segment overflow in thread #1: can't grow to 0x485b000
==19890== (see section Limitations in user manual)
==19890== NOTE: further instances of this message will not be shown
Tempo de execucao: 362812323 microsegundos
==19890==
==19890== I refs:      126,015,566,913
==19890== I1 misses:      2,429
==19890== L1i misses:      2,268
==19890== I1 miss rate:      0.00%
==19890== L1i miss rate:      0.00%
==19890==
==19890== D refs:      66,008,376,410 (41,005,499,153 rd + 25,002,877,257 wr)
==19890== D1 misses:      2,734,504,724 ( 2,734,312,539 rd +      192,185 wr)
==19890== L1d misses:      148,410,593 ( 148,219,341 rd +      191,252 wr)
==19890== D1 miss rate:      4.1% (      6.7% +      0.0% )
==19890== L1d miss rate:      0.2% (      0.4% +      0.0% )
==19890==
==19890== LL refs:      2,734,507,153 ( 2,734,314,968 rd +      192,185 wr)
==19890== LL misses:      148,412,861 ( 148,221,609 rd +      191,252 wr)
==19890== LL miss rate:      0.1% (      0.1% +      0.0% )
```



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática



- Utilizando localidade espacial e BLOCK SIZE
 - Sem otimizações

```
luis@Ubuntu-2.25:/media/sf_Linguagem_C/threads$ valgrind --tool=cachegrind ./blocking
==20617== Cachegrind, a cache and branch-prediction profiler
==20617== Copyright (C) 2002-2017, and GNU GPL'd, by Nicholas Nethercote et al.
==20617== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==20617== Command: ./blocking
==20617==
--20617-- Warning: Cannot auto-detect cache config, using defaults.
--20617-- Run with -v to see.
==20617== brk segment overflow in thread #1: can't grow to 0x485b000
==20617== (see section Limitations in user manual)
==20617== NOTE: further instances of this message will not be shown
Tempo de execucao: 339617096 microsegundos
==20617==
==20617== I refs: 131,539,411,596
==20617== I1 misses: 2,358
==20617== L1i misses: 2,254
==20617== I1 miss rate: 0.00%
==20617== L1i miss rate: 0.00%
==20617==
==20617== D refs: 69,335,912,856 (44,300,978,774 rd + 25,034,934,082 wr)
==20617== D1 misses: 5,529,680 ( 5,337,497 rd + 192,183 wr)
==20617== L1d misses: 3,138,789 ( 2,947,536 rd + 191,253 wr)
==20617== D1 miss rate: 0.0% ( 0.0% + 0.0% )
==20617== L1d miss rate: 0.0% ( 0.0% + 0.0% )
==20617==
==20617== LL refs: 5,532,038 ( 5,339,855 rd + 192,183 wr)
==20617== LL misses: 3,141,043 ( 2,949,790 rd + 191,253 wr)
==20617== LL miss rate: 0.0% ( 0.0% + 0.0% )
luis@Ubuntu-2.25:/media/sf_Linguagem_C/threads$
```



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática



- Com otimizações

```
luis@Ubuntu-2-25:/media/sf_Linguagem_C/threads$ valgrind --tool=cachegrind ./blocking_otm
==20443== Cachegrind, a cache and branch-prediction profiler
==20443== Copyright (C) 2002-2017, and GNU GPL'd, by Nicholas Nethercote et al.
==20443== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==20443== Command: ./blocking_otm
==20443==
--20443-- Warning: Cannot auto-detect cache config, using defaults.
--20443-- Run with -v to see.
==20443== brk segment overflow in thread #1: can't grow to 0x485b000
==20443== (see section Limitations in user manual)
==20443== NOTE: further instances of this message will not be shown
Tempo de execucao: 18934146 microsegundos
==20443==
==20443== I refs: 9,340,600,437
==20443== I1 misses: 2,153
==20443== L1i misses: 2,104
==20443== I1 miss rate: 0.00%
==20443== L1i miss rate: 0.00%
==20443==
==20443== D refs: 3,039,988,565 (3,038,828,822 rd + 1,159,743 wr)
==20443== D1 misses: 4,801,727 ( 4,609,530 rd + 192,197 wr)
==20443== L1d misses: 3,138,236 ( 2,946,959 rd + 191,277 wr)
==20443== D1 miss rate: 0.2% ( 0.2% + 16.6% )
==20443== L1d miss rate: 0.1% ( 0.1% + 16.5% )
==20443==
==20443== LL refs: 4,803,880 ( 4,611,683 rd + 192,197 wr)
==20443== LL misses: 3,140,340 ( 2,949,063 rd + 191,277 wr)
==20443== LL miss rate: 0.0% ( 0.0% + 16.5% )
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Análise

Estratégia	I refs (Total de instruções)	I1 misses (Quantidade de falhas no cache L1 de instruções)	LLi misses (Quantidade de falhas que chegaram até o último nível de cache)	D refs (Total de acessos a dados leitura/escrita)	D1 misses (Falhas no cache L1 de dados)	LLd misses (Falhas que chegaram até o último nível de cache)	LL miss rate (Taxa de falhas de cache)
Matriz por linha	> 126.015B	2.426	2.265	> 66.008B	> 1.154B	0.1%	~0.0%
Matriz por Coluna	> 126.015B	2.429	2.268	> 66.008B	> 2.734B	0.2%	~0.1%
Blocagem sem otimizar	> 131.539B	2.358	2.254	> 69.335B	> 5.529B	0.0%	~0.0%
Blocagem otimizado	> 9.340B	2.153	2.104	> 3.039B	> 4.801B	0.1%	~0.0%

É possível observar a quantidade drástica da redução de instruções utilizando a blocagem com otimizações no compilador, como também, nessa abordagem há uma queda no L1 miss, ou seja, os dados não encontrados no cache de nível 1 foram menores se comparados com os demais.

Em suma, pelo princípio da localidade os dados blocados puderam ser melhores aproveitados.