UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática

Comparar Threads

Nome do Integrante	RA
Jackson Barbosa dos Anjos	10418478

Código que percorre as linhas e faz multiplicação de matrizes 500x500:

```
@Jack-anjos ➜/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $ g++ linha.cpp -o linha
@Jack-anjos ➜/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $ ./linha

Tempo de execução percorrendo por linha: 446 ms
@Jack-anjos ➜/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $ ./linha

Tempo de execução percorrendo por linha: 438 ms
@Jack-anjos ➜/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $ ./linha

Tempo de execução percorrendo por linha: 472 ms
@Jack-anjos ➜/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $ ./linha

Tempo de execução percorrendo por linha: 453 ms
@Jack-anjos ➜/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $ ./linha

Tempo de execução percorrendo por linha: 444 ms
@Jack-anjos ➜/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $
```

Valgrind:

```
@Jack-anjos ➜/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $ valgrind --tool=cachegrind ./linha
==22075== Cachegrind, a cache and branch-prediction profiler
==22075== Copyright (C) 2002-2017, and GNU GPL'd, by Nicholas Nethercote et al.
==22075== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==22075== Command: ./linha
==22075==
--22075-- warning: L3 cache found, using its data for the LL simulation.

Tempo de execução percorrendo por linha: 18442 ms
==22075==
==22075== I   refs:      6,302,232,527
==22075== I1  misses:            1,892
==22075== LLi misses:            1,791
==22075== I1  miss rate:          0.00%
==22075== LLi miss rate:          0.00%
==22075==
==22075== D   refs:      2,770,123,988  (2,640,379,912 rd   + 129,744,076 wr)
==22075== D1  misses:      137,007,355  (  136,955,876 rd   +      51,479 wr)
==22075== LLd misses:           57,335  (        8,312 rd   +      49,023 wr)
==22075== D1  miss rate:           4.9% (          5.2%     +         0.0%  )
==22075== LLd miss rate:           0.0% (          0.0%     +         0.0%  )
==22075==
==22075== LL refs:         137,009,247  (  136,957,768 rd   +      51,479 wr)
==22075== LL misses:            59,126  (       10,103 rd   +      49,023 wr)
==22075== LL miss rate:            0.0% (          0.0%     +         0.0%  )
```

Código que percorre as colunas e faz multiplicação de matrizes 500x500

```
@Jack-anjos ➜/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $ g++ coluna.cpp -o coluna
@Jack-anjos ➜/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $ ./coluna

Tempo de execução percorrendo por coluna: 485 ms
@Jack-anjos ➜/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $ ./coluna

Tempo de execução percorrendo por coluna: 439 ms
@Jack-anjos ➜/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $ ./coluna

Tempo de execução percorrendo por coluna: 487 ms
@Jack-anjos ➜/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $ ./coluna

Tempo de execução percorrendo por coluna: 439 ms
@Jack-anjos ➜/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $ ./coluna

Tempo de execução percorrendo por coluna: 506 ms
@Jack-anjos ➜/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $
```

Valgrind:

```
@Jack-anjos ➜/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $ valgrind --tool=cachegrind ./coluna
==23025== Cachegrind, a cache and branch-prediction profiler
==23025== Copyright (C) 2002-2017, and GNU GPL'd, by Nicholas Nethercote et al.
==23025== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==23025== Command: ./coluna
==23025==
--23025-- warning: L3 cache found, using its data for the LL simulation.

Tempo de execução percorrendo por coluna: 18639 ms
==23025==
==23025== I   refs:      6,302,232,580
==23025== I1  misses:            1,892
==23025== LLi misses:            1,791
==23025== I1  miss rate:          0.00%
==23025== LLi miss rate:          0.00%
==23025==
==23025== D   refs:      2,770,124,008  (2,640,379,926 rd   + 129,744,082 wr)
==23025== D1  misses:      137,251,896  (  136,966,167 rd   +     285,729 wr)
==23025== LLd misses:           57,335  (        8,312 rd   +      49,023 wr)
==23025== D1  miss rate:           5.0% (          5.2%     +         0.2%  )
==23025== LLd miss rate:           0.0% (          0.0%     +         0.0%  )
==23025==
==23025== LL refs:         137,253,788  (  136,968,059 rd   +     285,729 wr)
==23025== LL misses:            59,126  (       10,103 rd   +      49,023 wr)
==23025== LL miss rate:            0.0% (          0.0%     +         0.0%  )
```

Código que percorre em blocos e faz multiplicação de matrizes 500x500 sem otimização:

```
@Jack-anjos ➜/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $ g++ -O0 -o bloco_sem_otimizar bloco.cpp
@Jack-anjos ➜/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $ ./bloco_sem_otimizar

Tempo de execução percorrendo por bloco: 490 ms
@Jack-anjos ➜/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $ ./bloco_sem_otimizar

Tempo de execução percorrendo por bloco: 508 ms
@Jack-anjos ➜/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $ ./bloco_sem_otimizar

Tempo de execução percorrendo por bloco: 518 ms
@Jack-anjos ➜/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $ ./bloco_sem_otimizar

Tempo de execução percorrendo por bloco: 489 ms
@Jack-anjos ➜/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $ ./bloco_sem_otimizar

Tempo de execução percorrendo por bloco: 490 ms
@Jack-anjos ➜/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $
```

Valgrind:

```
@Jack-anjos ➜/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $ valgrind --tool=cachegrind ./bloco_sem_otimizar
==26666== Cachegrind, a cache and branch-prediction profiler
==26666== Copyright (C) 2002-2017, and GNU GPL'd, by Nicholas Nethercote et al.
==26666== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==26666== Command: ./bloco_sem_otimizar
==26666==
--26666-- warning: L3 cache found, using its data for the LL simulation.

Tempo de execução percorrendo por bloco: 20839 ms
==26666==
==26666== I   refs:      6,801,236,051
==26666== I1  misses:            1,896
==26666== LLi misses:            1,794
==26666== I1  miss rate:          0.00%
==26666== LLi miss rate:          0.00%
==26666==
==26666== D   refs:      3,019,876,002  (2,890,381,923 rd   + 129,494,079 wr)
==26666== D1  misses:      137,007,358  (  136,971,629 rd   +      35,729 wr)
==26666== LLd misses:           57,333  (       23,561 rd   +      33,772 wr)
==26666== D1  miss rate:           4.5% (          4.7%     +         0.0%  )
==26666== LLd miss rate:           0.0% (          0.0%     +         0.0%  )
==26666==
==26666== LL refs:         137,009,254  (  136,973,525 rd   +      35,729 wr)
==26666== LL misses:            59,127  (       25,355 rd   +      33,772 wr)
==26666== LL miss rate:            0.0% (          0.0%     +         0.0%  )
```

Código que percorre em blocos e faz multiplicação de matrizes 500x500 com otimização máxima:

```
@Jack-anjos →/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $  g++ -O3 -o bloco_otimizado bloco.cpp
@Jack-anjos →/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $ ./bloco_otimizado

  Tempo de execução percorrendo por bloco: 101 ms
@Jack-anjos →/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $ ./bloco_otimizado

  Tempo de execução percorrendo por bloco: 104 ms
@Jack-anjos →/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $ ./bloco_otimizado

  Tempo de execução percorrendo por bloco: 102 ms
@Jack-anjos →/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $ ./bloco_otimizado

  Tempo de execução percorrendo por bloco: 101 ms
@Jack-anjos →/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $ ./bloco_otimizado

  Tempo de execução percorrendo por bloco: 120 ms
@Jack-anjos →/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $
```

Valgrind:

```
@Jack-anjos →/workspaces/comppar-05p-threads-cpp-Jack-anjos (main) $ valgrind --tool=cachegrind ./bloco_otimizado
==28882== Cachegrind, a cache and branch-prediction profiler
==28882== Copyright (C) 2002-2017, and GNU GPL'd, by Nicholas Nethercote et al.
==28882== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==28882== Command: ./bloco_otimizado
==28882==
--28882-- warning: L3 cache found, using its data for the LL simulation.

Tempo de execução percorrendo por bloco: 4829 ms
==28882==
==28882== I   refs:      1,042,454,292
==28882== I1  misses:            1,869
==28882== LLi misses:            1,772
==28882== I1  miss rate:          0.00%
==28882== LLi miss rate:          0.00%
==28882==
==28882== D   refs:        515,106,572  (385,864,482 rd   + 129,242,090 wr)
==28882== D1  misses:      135,518,429  (135,482,702 rd   +      35,727 wr)
==28882== LLd misses:           57,327  (     23,555 rd   +      33,772 wr)
==28882== D1  miss rate:          26.3% (       35.1%     +         0.0%  )
==28882== LLd miss rate:           0.0% (        0.0%     +         0.0%  )
==28882==
==28882== LL refs:         135,520,298  (135,484,571 rd   +      35,727 wr)
==28882== LL misses:            59,099  (     25,327 rd   +      33,772 wr)
==28882== LL miss rate:            0.0% (        0.0%     +         0.0%  )
```

Análise do Valgrind:

Multiplicação de matrizes percorrendo por linhas: alta taxa de miss e tempo de execução intermediário.

Multiplicação de matrizes percorrendo por colunas: aumentou taxa de miss, o que aumenta o tempo médio de execução, diminuindo eficiência do algoritmo, devido a acesso a memória menos facilitada.

Multiplicação de matrizes por bloco sem otimização: diminuiu a taxa de miss, porém sem ganho significativo no tempo de execução pela falta de otimização.

Multiplicação de matrizes por bloco com otimização máxima: menor taxa de miss entre os métodos, e tempo de execução caiu em 4x, indicando que tivemos menos acesso a memória principal, trabalhando mais os dados na memória cache.