Universidade de São Paulo
Instituto de Física de São Carlos

Prática 2: Cache

Stefan Taiguara Couperus Leal 10414866

03 de Setembro de 2019

Contents

1 Primeiro Método

N	tempo (s)
100	0.0013819
200	0.0128335
500	0.518457
1000	4.03742
1500	31.1972

Table 1:

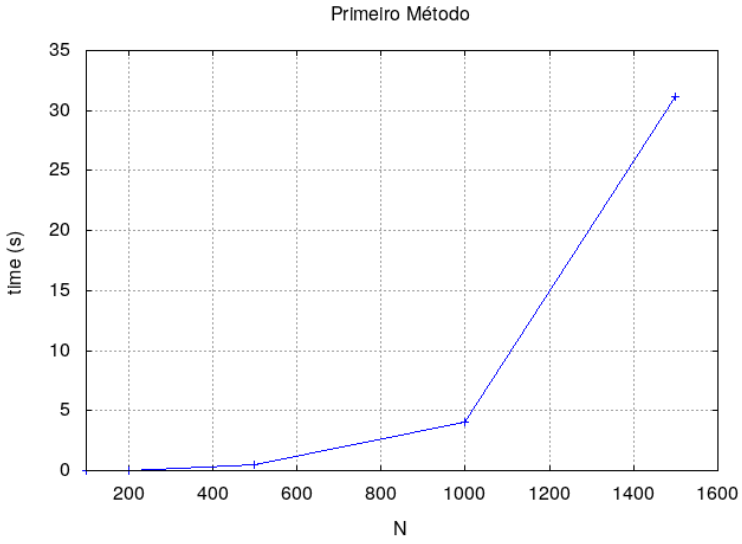

Figure 1:
Tempo de execução para dados valores de N

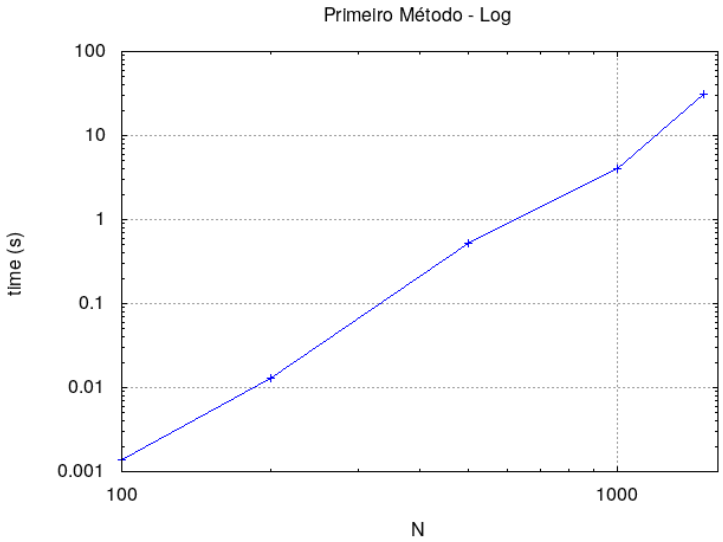

Figure 2:
Tempo de execução para dados valores de N

1.1 Análise da Cache

1.1.1 N = 100

I refs: 11,620,342
I1 misses: 2,051
LLi misses: 2,006
I1 miss rate: 0.02%
LLi miss rate: 0.02%

D refs: 4,336,811 (3,978,854 rd + 357,957 wr)
D1 misses: 147,518 (141,246 rd + 6,272 wr)
LLd misses: 13,391 (8,014 rd + 5,377 wr
D1 miss rate: 3.4% (3.5% + 1.8%)
LLd miss rate: 0.3% (0.2% + 1.5%)

LL refs: 149,569 (143,297 rd + 6,272 wr
LL misses: 15,397 (10,020 rd + 5,377 wr)
LL miss rate: 0.1% (0.1% + 1.5%)

1.1.2 N = 1000

I refs: 7,240,270,789

I1 misses: 2,059
LLi misses: 2,033
I1 miss rate: 0.00%
LLi miss rate: 0.00%

D refs: 3,062,721,948 (3,043,551,288 rd + 19,170,660 wr)
D1 misses: 1,253,268,733 (1,252,015,869 rd + 1,252,864 wr)
LLd misses: 125,514,392 (125,137,346 rd + 377,046 wr)
D1 miss rate: 40.9% (41.1% + 6.5%)
LLd miss rate: 4.1% (4.1% + 2.0%)

LL refs: 1,253,270,792 (1,252,017,928 rd + 1,252,864 wr)
LL misses: 125,516,425 (125,139,379 rd + 377,046 wr)
LL miss rate: 1.2% (1.2% + 2.0%)

2 Segundo Método

N	tempo (s)
100	0.0017117
200	0.0139805
500	0.51274
1000	4.37631
1500	30.4692

Table 2:

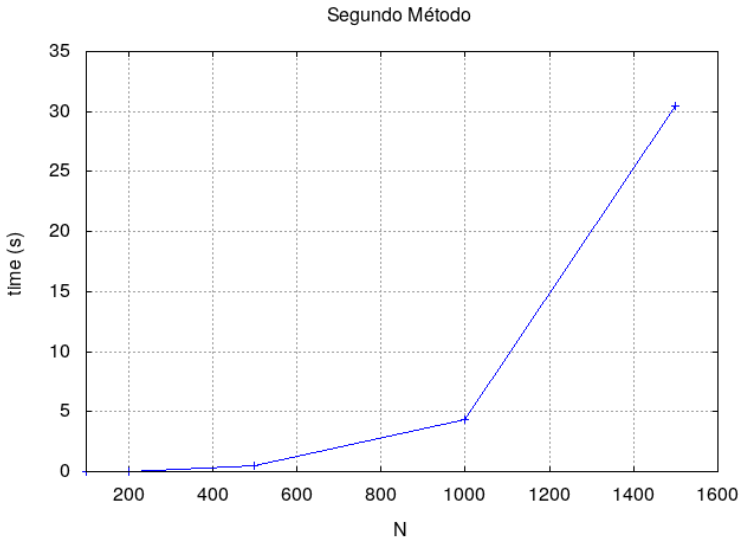

Figure 3:
Tempo de execução para dados valores de N

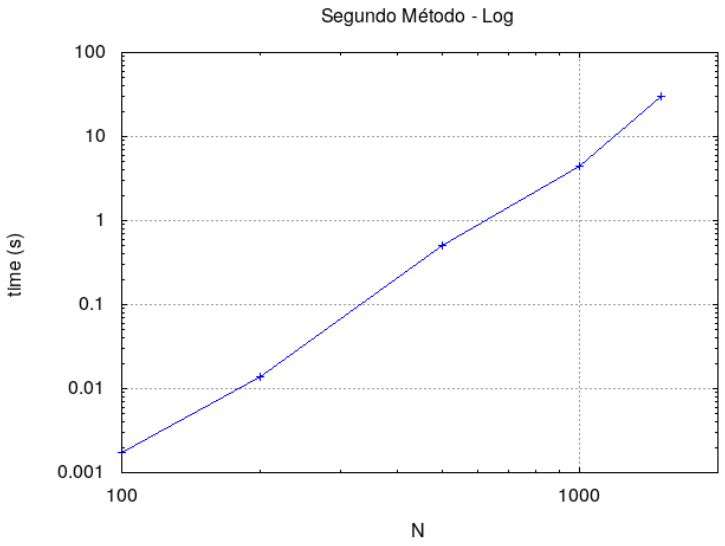

Figure 4:
Tempo de execução para dados valores de N

2.1 Análise da Cache

2.1.1 Para N = 100

I refs: 11,641,536
I1 misses: 2,053
LLi misses: 2,008
I1 miss rate: 0.02%
LLi miss rate: 0.02%

D refs: 4,356,615 (3,998,657 rd + 357,958 wr)
D1 misses: 158,574 (143,553 rd + 15,021 wr)
LLd misses: 13,391 (8,014 rd + 5,377 wr)
D1 miss rate: 3.6% (3.6% + 4.2%)
LLd miss rate: 0.3% (0.2% + 1.5%)

LL refs: 160,627 (145,606 rd + 15,021 wr)
LL misses: 15,399 (10,022 rd + 5,377 wr)
LL miss rate: 0.1% (0.1% + 1.5%)

2.1.2 Para N = 1000

I refs: 11,641,536
I1 misses: 2,053
LLi misses: 2,008
I1 miss rate: 0.02%
LLi miss rate: 0.02%

D refs: 4,356,615 (3,998,657 rd + 357,958 wr)
D1 misses: 158,574 (143,553 rd + 15,021 wr)
LLd misses: 13,391 (8,014 rd + 5,377 wr)
D1 miss rate: 3.6% (3.6% + 4.2%)
LLd miss rate: 0.3% (0.2% + 1.5%)

LL refs: 160,627 (145,606 rd + 15,021 wr
LL misses: 15,399 (10,022 rd + 5,377 wr)
LL miss rate: 0.1% (0.1% + 1.5%)

3 Terceito Método

N	tempo (s)
100	0.0014775
200	0.0132012
500	0.256493
1000	2.12084
1500	6.89262

Table 3:

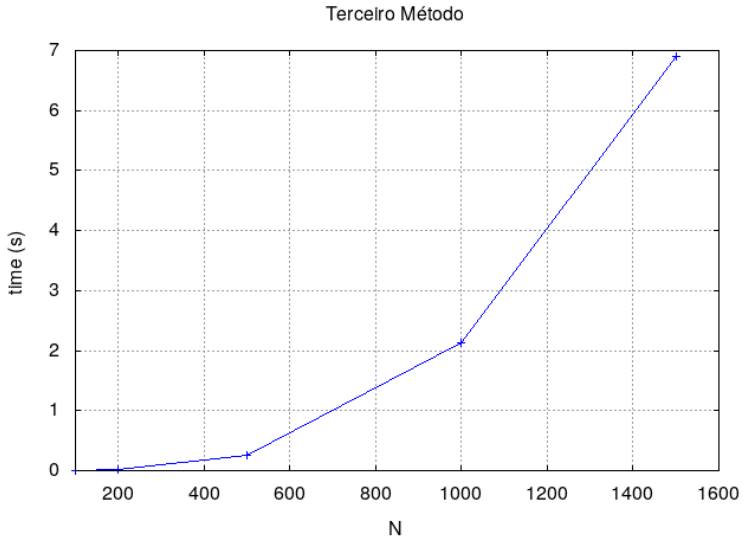

Figure 5:
Tempo de execução para dados valores de N

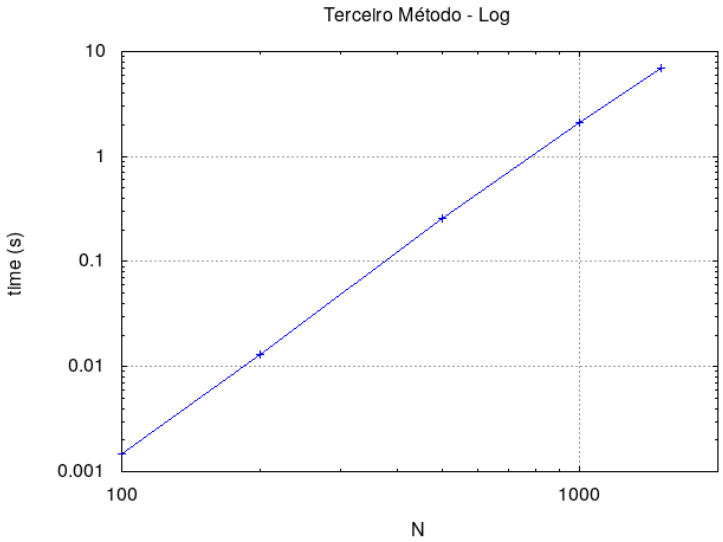

Figure 6:
Tempo de execução para dados valores de N

3.1 Análise da Cache

3.1.1 Para N = 100

I refs: 11,661,234
I1 misses: 2,052
LLi misses: 2,006
I1 miss rate: 0.02%
LLi miss rate: 0.02%

D refs: 5,346,915 (3,988,956 rd + 1,357,959 wr)
D1 misses: 149,987 (143,715 rd + 6,272 wr)
LLd misses: 13,392 (8,015 rd + 5,377 wr)
D1 miss rate: 2.8% (3.6% + 0.5%)
LLd miss rate: 0.3% (0.2% + 0.4%)

Total:
LL refs: 152,039 (145,767 rd + 6,272 wr)
LL misses: 15,398 (10,021 rd + 5,377 wr)
LL miss rate: 0.1% (0.1% + 0.4%)

3.1.2 Para N = 1000

I refs: 7,244,279,903
I1 misses: 2,055
LLi misses: 2,029
I1 miss rate: 0.00%
LLi miss rate: 0.00%

D refs: 4,063,723,087 (3,044,552,444 rd + 1,019,170,643 wr)
D1 misses: 125,771,870 (125,394,008 rd + 377,862 wr)
LLd misses: 125,764,936 (125,387,890 rd + 377,046 wr)
D1 miss rate: 3.1% (4.1% + 0.0%)
LLd miss rate: 3.1% (4.1% + 0.0%)

LL refs: 125,773,925 (125,396,063 rd + 377,862 wr)
LL misses: 125,766,965 (125,389,919 rd + 377,046 wr)
LL miss rate: 1.1% (1.2% + 0.0%)

4 Comparação dos métodos anteriores

N	Primeiro Método t(s)	Segundo Método t(s)	Terceiro Método t(s)
100	0.0013819	0.0014704	0.001236
200	0.0128335	0.0141117	0.0115682
500	0.518457	0.391628	0.296568
1000	4.03742	3.61946	2.49403
1500	31.1972	26.9204	8.12806

Table 4: Comparação de três diferentes métodos de multiplicação de matrizes

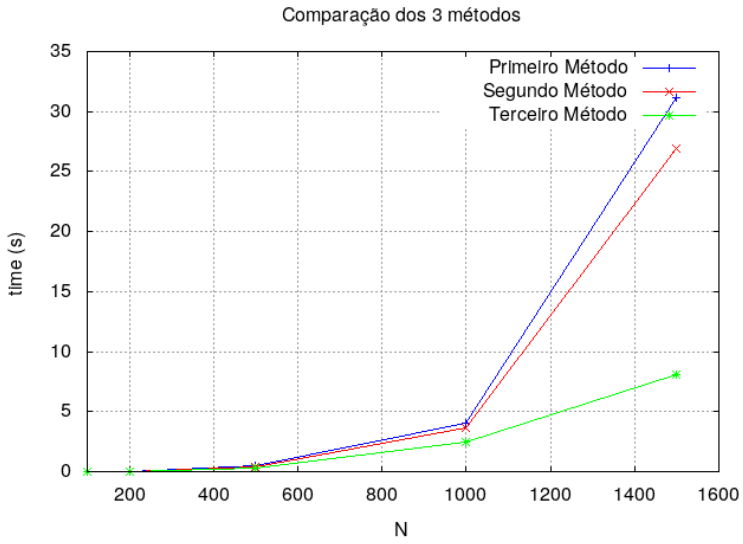

Figure 7:
Comparação de tempos de execuções para dados valores de N

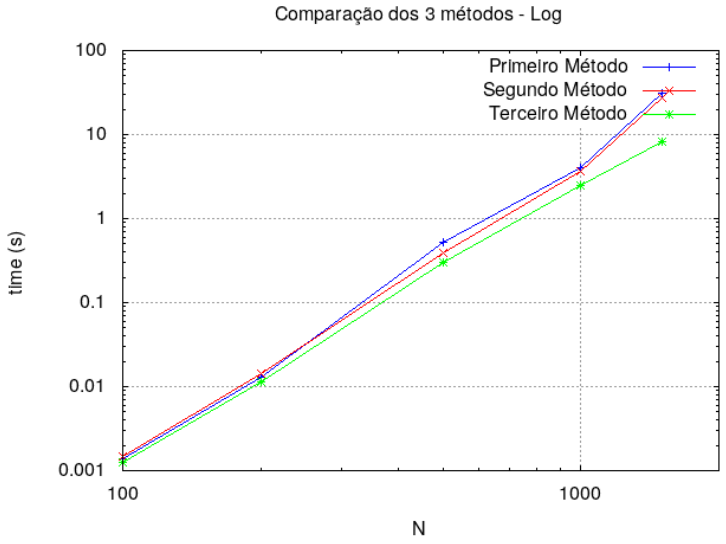

Figure 8:
Comparação de tempos de execuções para dados valores de N

É possível notar que para valores de $N \leq 200$, não há uma diferença tão significativa nos tempos de execução para os diferentes métodos, mas

para valores maiores é notado diferenças em relação a performance entre os métodos.

Analisando o funcionamento da cache com o valgrind nota-se que para $N = 100$ os valores de miss rates para os diferentes métodos não diferem muito.

	Primeiro Método		Segundo Método		Terceiro Método	
	rd	wr	rd	wr	rd	wr
D1 refs	3.5%	1.8%	3.6%	4.2%	3.6%	0.5%
Lld	0.2%	1.5%	0.2%	1.5%	0.2%	0.4%

Table 5:

Mas quando é analisado para $N = 1000$ nota-se valores mais destoantes.

	Primeiro Método		Segundo Método		Terceiro Método	
	rd	wr	rd	wr	rd	wr
D1 refs	41.1%	6.5%	3.6%	4.2%	4.1%	0.0%
Lld	4.1%	0.2%	0.2%	1.5%	4.1%	0.0%

Table 6:

Nota-se que para o primeiro método a porcentagem de cache miss atinge 41.1% na parte de leitura dos dados e 3.6% na parte de escrita (levando 4.03742 s) isso indica que a CPU passou uma parte considerável do tempo esperando a liberação da memória.

Pode ser observado que para o segundo método a quantidade de cache miss cai para 3.6% para escrita e 4.2% para a leitura, o que faz seu tempo ser significativamente menor (levou 3.61946 s). O segundo método apresenta essa melhora em performance já que percorre por (j,i,k), com isso há um aproveitamento maior da cache dado que a equação usada é : "soma += A[i][k] * B[k][j]".

Já no terceiro método é notado que ouve uma queda significativa no cache miss para a escrita 0.0% e um valor de leitura 4.1% (o que levou o programa a rodar em 2.46403 s). Já para com o terceiro método foi utilizado da ordem (i,k,j) e foi usado C[i][j] += A[i][k] * B[k][j], com isso ouve um aproveitamento melhor da cache tornando o programa mais eficiente que os demais métodos.