Students Identification

Number - Name

109682 - Sofia Aleixo

109862 - Lara Franco

110271 - Sara Português

2- First Steps

Please justify all your answers with values from the experiments.

1- What is the cache capacity of the computer you used (please write the workstation name)?

Data	Test1	Test2	Test3	Test4	Test5	Test6
Array Size	4kiB	8KiB	16KiB	32KiB	64KiB	128KiB
t2-t1	0.000213	0.000356	0.000688	0.001302	0.005898	0.012353
#accesses a[i]	8192	16384	32768	65536	131072	262144
#mean access time	2.6E-8	2.17E-8	2.10E-8	1.99E-8	4.50E-8	4.71E-8

Justification/Calculations

Computador usado -> 0-27 i5-100

Stride usado para os cálculos -> 64

Pelos dados conseguimos perceber que o tamanho da cache é 32KiB, uma vez que o mean access time se mantém constante até ao array size de 32KiB, seguindo-se uma diferença significativa entre o mean access time entre o 32KiB e o 64KiB.

Cálculos:

#accesses a[i] = n_iterations (spark.c) *2

#mean access time = t2-t1 ÷ #accesses a[i]

2- What is the cache capacity?

Da figura 1 podemos retirar que a cacpacidade da cache é 64KiB, uma vez que vemos uma clara mudança dos tempos da linha dos 64 KiB para os 128 KiB, o que nos indica que passamos do nível L1 de cache para o nivel L2 logo a capacidade da cache L1 é 64KiB.

3- What is the size of each cache block?

A dimensão do bloco de cache pode ser determinada ao identificar o primeiro stride em que o tempo de acesso para o grupo de arrays com tempos de acesso mais altos se torna estável. Essa estabilização ocorre quando o stride é igual ou superior ao tamanho do bloco de cache. Nesse ponto, cada elemento acessado do array mapeia para um bloco de cache diferente, levando a uma taxa de misses próxima de 100%, e o tempo de acesso permanece constante para strides seguintes. Neste caso, a estabilização ocorre para um stride de 16 bytes, indicando que o bloco de cache tem 16 bytes.

4- What is the L1 cache miss penalty time?

Para calcularmos o miss penalty time, olhamos para o stride que determinou o tamanho do nosso cache block que é 16, O menor tempo de acesso observado é considerado o tempo de hit (acesso à cache), enquanto o maior tempo, para esse stride, é o tempo de miss (acesso à memória principal). A diferença entre esses dois tempos dá-nos o valor do miss penalty.

1000-360 = 640.

3- Assignment

3.1.1 Modeling the L1 Data Cache

a) What are the processor events that will be analyzed during its execution? Explain their meaning.

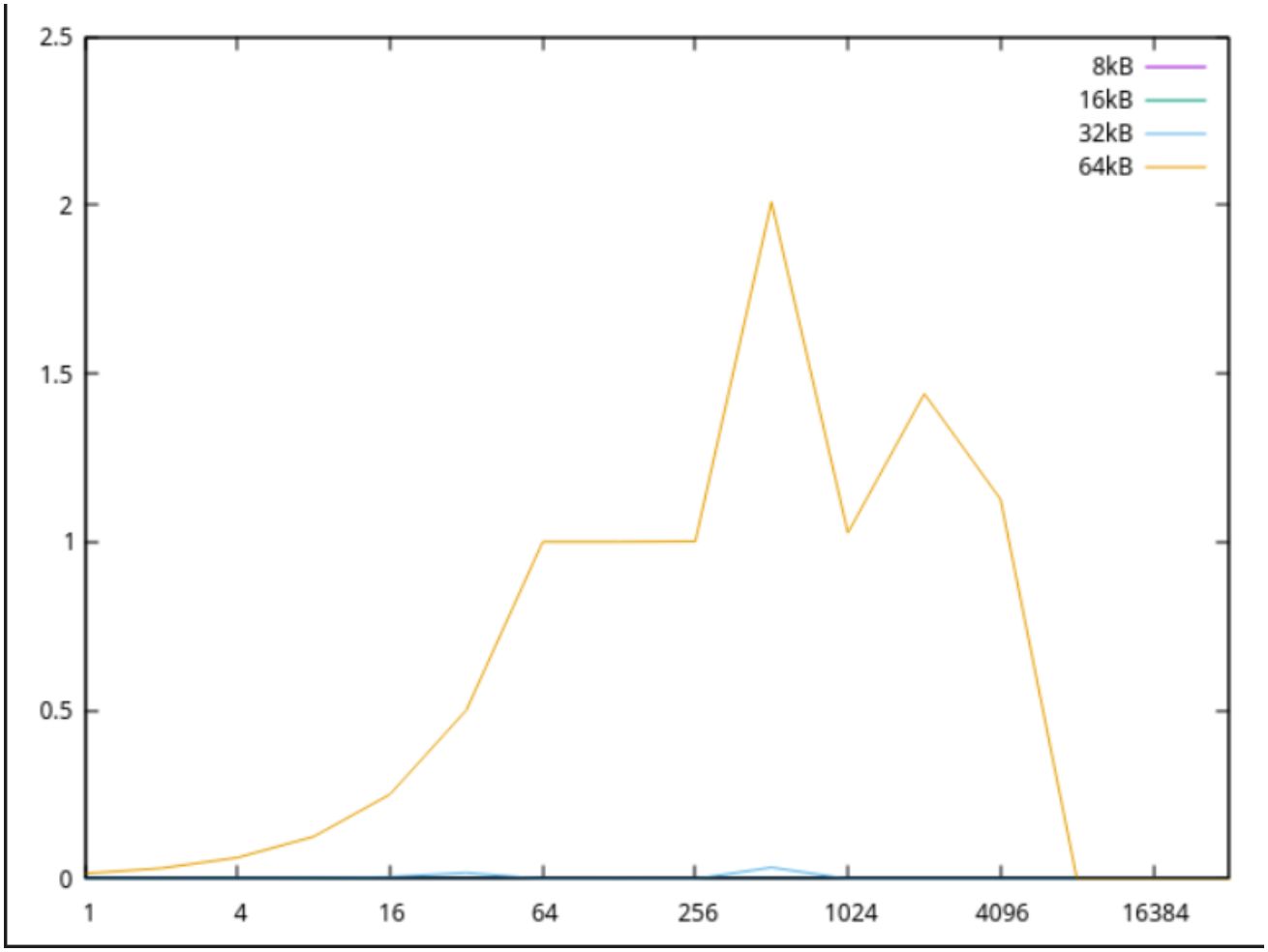
Os eventos a serem avaliados no ficheiro cm1 são "L1 Data Caches Misses", analisados pelo PAPI_L1_DCM. Isto permite-nos avaliar quantas vezes o programa tenta acessar dados que não estão disponiveis na cache L1 durante a execução do programa.

b) Plot the variation of the average number of misses (Avg Misses) with the stride size, for each considered dimension of the L1 data cache (8kB, 16kB, 32kB and 64kB). Note that, you may fill these tables and graphics (as well as the following ones in this report) on your computer and submit the printed version.

Array Size	Stride	Avg Misses	Avg Cycle Time
	1	0.000106	0.000703
	2	0.000104	0.000775
	4	0.000015	0.000810
	8	0.000013	0.000811
	16	0.000022	0.000828
	32	0.000018	0.000828
8k Bytes	64	0.000023	0.000872

Array Size	Stride	Avg Misses	Avg Cycle Time
	128	0.000009	0.001005
	256	0.000006	0.001182
	512	0.000017	0.000853
	1024	0.000002	0.000878
	2048	0.000008	0.001027
	4096	0.000004	0.001504
---	---	---	---
	1	0.000099	0.000810
	2	0.000047	0.000808
	4	0.000019	0.000758
	8	0.000046	0.000680
	16	0.000021	0.000660
	32	0.000040	0.000584
16k Bytes	64	0.000023	0.000594
	128	0.000022	0.000576
	256	0.000007	0.000634
	512	0.000008	0.000759
	1024	0.000002	0.000573
	2048	0.000003	0.000622
	4096	0.000002	0.000601
	8192	0.000003	0.000753
---	---	---	---
	1	0.000925	0.000401
	2	0.001526	0.000398
	4	0.002578	0.000402
	8	0.003648	0.000395
	16	0.004616	0.000400
	32	0.017508	0.000400
32k Bytes	64	0.000159	0.000402
	128	0.000109	0.000409
	256	0.000113	0.000427
	512	0.037020	0.000551
	1024	0.000010	0.000634
	2048	0.000005	0.000636
	4096	0.000004	0.000887
	8192	0.000002	0.000578
	16384	0.000002	0.000734

Array Size	Stride	Avg Misses	Avg Cycle Time
	1	0.015673	0.000398
	2	0.031340	0.000396
	4	0.062661	0.000396
	8	0.125274	0.000396
	16	0.250563	0.000421
	32	0.500739	0.000437
64k Bytes	64	1.000157	0.000826
	128	1.000190	0.001244
	256	1.001146	0.001264
	512	1.008848	0.001771
	1024	1.134926	0.002243
	2048	1.244382	0.001873
	4096	1.186060	0.003139
	8192	0.000005	0.000885
	16384	0.000001	0.000576
	32768	0.000001	0.000734



c) By analyzing the obtained results:

• Determine the size of the L1 data cache. Justify your answer.

O tamanho da cache L1 é 32 KiB pois conseguimos verificar que é o maior array cujo número de misses está mais próximo de 0.

• Determine the block size adopted in this cache. Justify your answer.

O block size é 64 bytes. Olhando para o maior array, com 64 KiB, existe uma subida do número de misses até ao stride 64 a partir do qual se mantém constante. Esta subida deve-se ao facto de, para strides pequeno, o bloco que é alocado depois de um miss permitir guardar sucessivos valores. Portanto, o número de misses é menor e vai aumentando à medida que o stride aumenta. Quando o tamanho do stride é igual ou superior ao tamanho do bloco, o valor vai incidir noutro bloco, resultando num miss para cada valor, e num gráfico constante.

• Characterize the associativity set size adopted in this cache. Justify your answer.

Para calcularmos o tamanho do associativity set, analisamos o maior array, neste caso será a linha amarela (64kB), e vemos o maior valor de stride que se aproxima de zero, neste caso será o stride 8192. Em seguida será só dividir o array size pelo stride. Logo, o cache tem 8 vias, sendo um 8-way set.

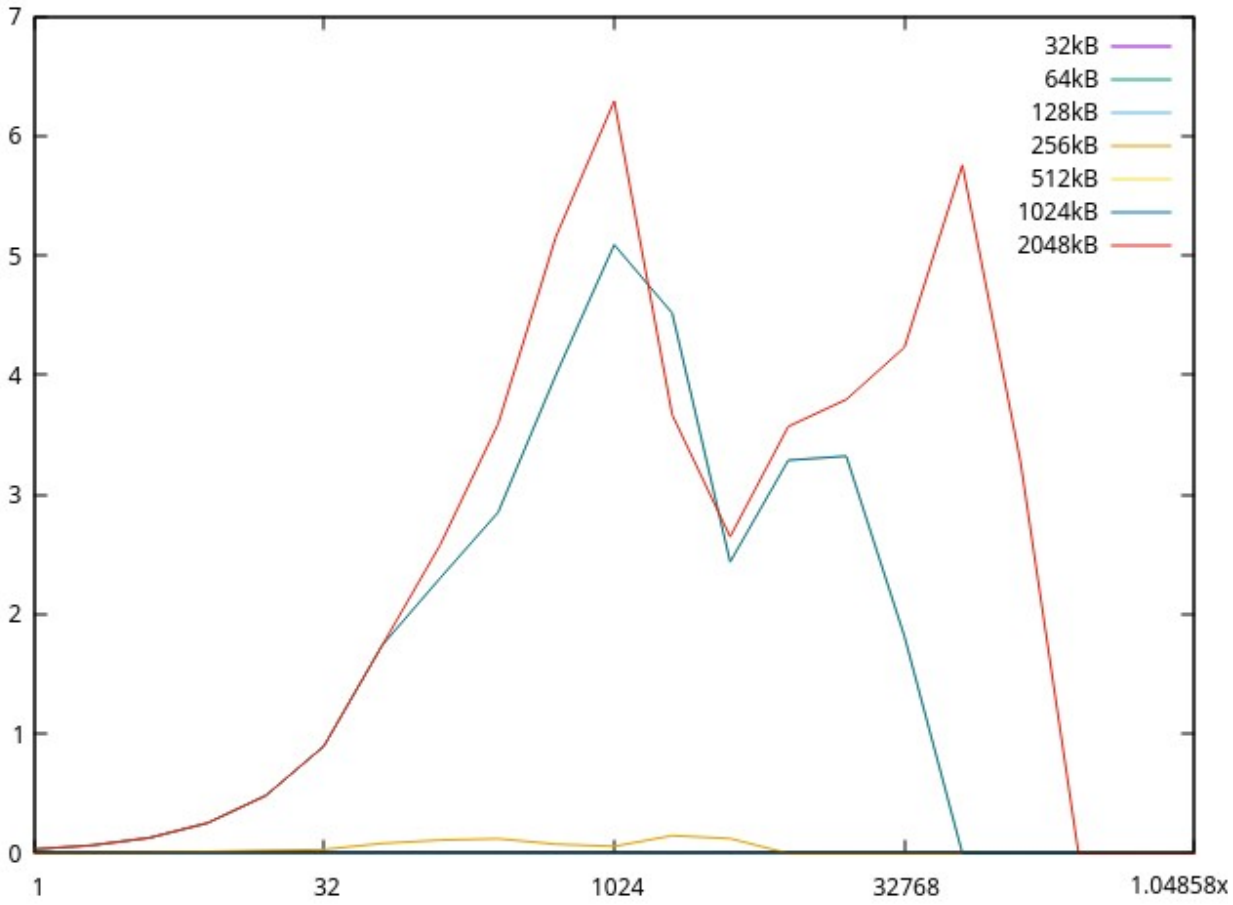
65536 ÷ 8192 = 8

3.1.2 Modeling the L2 Cache

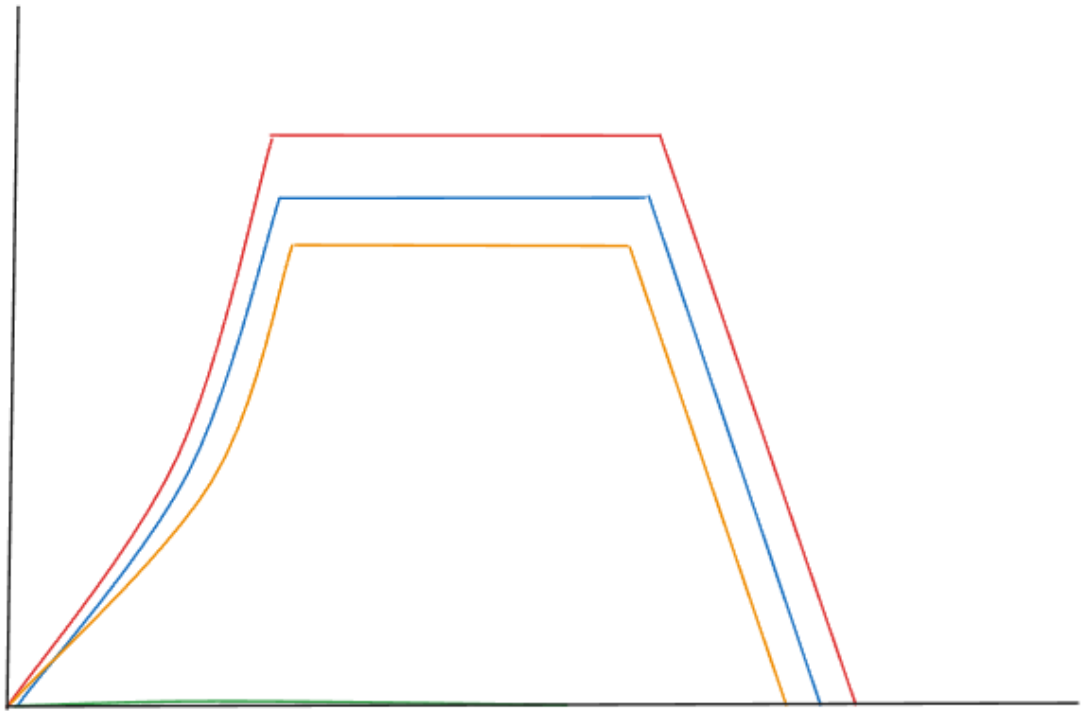
a) Describe and justify the changes introduced in this program.

Fizemos duas mudanças. Alterámos o evento PAPI de PAPI_L1_DCM para PAPI_L2_DCM para focar na análise da taxa de falhas da cache L2. Alterámos também o valor de CACHE_MAX, aumentando-o para 2MiB. Desta forma, acomoda o maior tamanho da cache L2 e ultrapassa-o, permitindo-nos deduzir o seu tamanho.

b) Plot the variation of the average number of misses (Avg Misses) with the stride considered dimension of the L2 cache.



O gráfico acima foi obtido no computador do laboratório, e apartir dele não conseguimos retirar nada do que nos é pedido abaixo. No entanto fizemos o desenho de como deveria ser o nosso gráfico.



c) By analyzing the obtained results:

• Determine the size of the L2 data cache. Justify your answer.

Para determinar o tamanho da cache L2 olharíamos para o maior array cujo número de misses está mais próximo de 0.

• Determine the block size adopted in this cache. Justify your answer.

Com o objectivo de determinar o block size da L2, temos duas situações. Se o tamanho for igual ao do L1, o gráfico vai ter uma única curva, pois as variações de L1 e L2 ocorrem nos mesmos strides. Se o tamanho for superior, vão existir duas curvas, a primeira de L1 e a segunda de L2, sendo o stride em que a segunda se estabiliza o tamanho do bloco de L2.

• Characterize the associativity set size adopted in this cache. Justify your answer.

Para calcular o tamanho do associativity set, analisaríamos o maior array e procuraríamos o maior valor de stride a aproximar-se de zero. Por fim, dividiríamos o array size pelo stride.

3.2 Profiling and Optimizing Data Cache Accesses

3.2.1 Straightforward implementation

a) What is the total amount of memory that is required to accommodate each of these matrices?

Cada matriz ocupa cerca de 512KiB. Sendo o N = 512, com cada elemento da matriz a ocupar 2 bytes.

512 * 512 * 2B = 512KiB

b) Fill the following table with the obtained data.

Info	Data
Total number of L1 data cache misses	135.36x10^6
Total number of load / store instructions completed	402.92x10^6
Total number of clock cycles	655.35x10^6
Elapsed time	0.218 seconds

c) Evaluate the resulting L1 data cache Hit-Rate:

O Hit-Rate é 66.4%.

3.2.2 First Optimization: Matrix transpose before multiplication

a) Fill the following table with the obtained data.

Info	Data
Total number of L1 data cache misses	4.21x10^6
Total number of load / store instructions completed	402.92x10^6
Total number of clock cycles	264.60x10^6
Elapsed time	0.0804 seconds

b) Evaluate the resulting L1 data cache Hit-Rate:

O Hit-Rate é 98.96%.

c) Fill the following table with the obtained data.

Info	Data
Total number of L1 data cache misses	4.22x10^6
Total number of load / store instructions completed	403.44x10^6
Total number of clock cycles	316.65x10^6
Elapsed time	0.1055 seconds

Info	Data

Comment on the obtained results when including the matrix transposition in the execution time:

Podemos observar que praticamente não afecta o tempo de execução, tem um aumento ligeiro, mas desprezável em comparação à otimização.

d) Compare the obtained results with those that were obtained for the straightforward implementation, by calculating the difference of the resulting hit-rates (ΔHitRate) and the obtained speedups.

ΔHitRate = HitRate mm2 − HitRate mm1 = 98.96 - 66.4 = 32.56%

Speedup(#Clocks) = #Clocks mm1 ÷ #Clocks mm2 = 655.35 ÷ 264.60 = 2.5

Speedup(Time) = Time mm1 ÷ Time mm2 = 0.218 ÷ 0.0804 = 2.711

A implementação otimizada (mm2) obteve uma melhoria significativa tanto no desempenho da cache quanto na velocidade de execução quando comparada com a implementação direta (mm1). A taxa de acertos na cache aumentou 32.56% (de 66,4% para 99,96%), evidenciando uma menor recorrência à memória principal e, consequentemente, uma redução no tempo de acesso aos dados.

3.2.3 Second Optimization: Blocked (tiled) matrix multiply

a) How many matrix elements can be accommodated in each cache line?

256 elementos da matrix. (l1 block size * l1 associativity set size ÷ 2) sendo o block size 64 e o tamnanho do associativity set 8.

b) Fill the following table with the obtained data.

Info	Data
Total number of L1 data cache misses	2.22x10^6
Total number of load / store instructions completed	536.94x10^6
Total number of clock cycles	248.38x10^6
Elapsed time	0.083 seconds

c) Evaluate the resulting L1 data cache Hit-Rate:

O Hit-Rate é 99.59%

d) Compare the obtained results with those that were obtained for the straightforward implementation, by calculating the difference of the resulting hit-rates (ΔHitRate) and the obtained speedup.

ΔHitRate = HitRatemm3 − HitRatemm1 = 99.59 - 66.4 = 33.19%

Speedup(#Clocks) = #Clocksmm1 ÷ #Clocksmm3 = 2.6

Aqui apenas medimos o speedup dos ciclos de relógio pois é uma métrica mais fiável para comparar o desempenho puro das diferentes implementações, uma vez que está diretamente relacionado com o processamento efetivo do programa.

A implementação mm3 apresentou um aumento claro na taxa de acertos da cache em relação à implementação direta mm1, com um ΔHitRate de 33,19% (de 66,4% para 99,59%). Este resultado demonstra que a otimização realizada melhora significativamente a eficiência do acesso à memória, reduzindo o número de acessos à memória principal e aumentando o desempenho global do programa.O speedup calculado com base no número de ciclos de relógio foi de 2,6, indicando que a implementação mm3 executa aproximadamente 2,6 vezes mais rápido do que a implementação direta.

e) Compare the obtained results with those that were obtained for the matrix transpose implementation by calculating the difference of the resulting hit-rates (ΔHitRate) and the obtained speedup. If the obtained speedup is positive, but the difference of the resulting hit-rates is negative, how do you explain the performance improvement? (Hint: study the hit-rates of the L2 cache for both implementations;)

ΔHitRate = HitRatemm3 − HitRatemm2 = 99.59 - 98.96 = 0.63%

Speedup(#Clocks) = #Clocksmm2 ÷ #Clocksmm3 = 264.60 ÷ 248.38 = 1.065

A diferença na taxa de acertos da cache entre o mm3 e o mm2 foi de apenas 0,63% , o que indica que a otimização adicional feita na implementação mm3 teve um impacto muito pequeno na eficiência do uso da cache em relação à versão mm2.

O speedup calculado com base no número de ciclos de relógio é de 1,065, ou seja, a implementação mm3 é apenas cerca de 6,5% mais rápida do que mm2. Este pequeno ganho demonstra que, apesar da ligeira melhoria na taxa de acertos da cache, o benefício em termos de desempenho é bastante limitado.

3.2.3 Comparing results against the CPU specifications

Now that you have characterized the cache on your lab computer, you are going to compare it against the manufacturer's specification. For this you can check the device's datasheet, or make use of the command lscpu. Comment on the results.

Os resultados obtidos após correr o comando lscpu estão de acordo com os resultados analisados através do gráfico da cache L1. Do segundo gráfico, não conseguimos retirar informação logo não conseguimos confirmar os dados obtidos após correr o comando.