

MEMORIA PRACTICA 1 DE TÉCNICAS DE OPTIMIZACIÓN

Ejercicio 1. Realizar una implementación del programa de multiplicación de 2 matrices de dimensiones 2048x2048

Se ha realizado la implementación del programa mediante código C++.

El código a implementar es el siguiente:

```
for(int i=0; i < 2048; i++){  
    for(int j=0; j < 2048; j++){  
        for(int k = 0; k < 2048; k++){  
            C[i][j] = C[i][j] + A[i][k] * B[k][j]  
        }  
    }  
}
```

Ejercicio 2. Caracterizar el equipo donde se ejecutará el programa

Las características del equipo empleado son las siguientes:

PROCESADOR	Intel Core i7-8565U
ENTORNO DE DESARROLLO	Ubuntu 20.04.3
HERRAMIENTA DE DESARROLLO	C++
COMPILADOR	G++ 9.3.0
MEMORIA CACHÉ L1	
DATA	4x32 KB
LINE SIZE	64 B
ASSOCIATIVITY	8-way set
NUMERO DE LINEAS POR VIA	$\frac{32 \text{ KB}}{8} = 4 \frac{\text{Kb}}{\text{via}} \rightarrow \frac{4 \frac{\text{KB}}{\text{via}}}{64 \frac{\text{B}}{\text{via}}} = 64 \frac{\text{lineas}}{\text{via}}$
MEMORIA CACHÉ L2	
DATA	4x256 KB
LINE SIZE	64 B
ASSOCIATIVITY	8-way set
NUMERO DE LINEAS POR VIA	$\frac{256 \text{ KB}}{8} = \frac{32 \text{ KB}}{\text{via}} \rightarrow \frac{32 \frac{\text{KB}}{\text{via}}}{64 \frac{\text{B}}{\text{via}}} = 5 \frac{\text{lineas}}{\text{via}}$
MEMORIA CACHÉ L3	
DATA	8 MB
LINE SIZE	64 B
ASSOCIATIVITY	16-way set

NUMERO DE LINEAS POR VIA	$\frac{8 MB}{16} = 500 \frac{KB}{via} \rightarrow \frac{500 \frac{KB}{via}}{64 \frac{B}{via}} = 800 \frac{lineas}{via}$
--------------------------	---

Ejercicio 3. Medir los tiempos del bucle de ejecución de las dos matrices, inicialmente con vector de iteraciones (i,j,k)

Se ha tardado un tiempo de ejecución de 124'533 ms

Ejercicio 4. Medir los tiempos de ejecución con vector de iteraciones (i,k,j)

Se ha tardado un tiempo de ejecución de 24'187.5 ms

Ejercicio 5. Realizar comparativa de tiempos para la compilación con las opciones -O0 y -O2. Identifica la opción por defecto e indica vuestras conclusiones generales sobre el compilador

	-O0	-O2
Vector (i,j,k)	124'616 ms	56'763.1 ms
Vector (i,k,j)	24'402.3 ms	5'579.78 ms

Por defecto, se ejecuta la opción -O0.

Sobre el compilador, al ejecutar con la opción -O0 no se aplica una optimización, hecho que cambia para la opción -O2. Esto demuestra que el tiempo de ejecución de la opción -O2 sea inferior al tiempo de ejecución de la opción -O0.

Ejercicio 6. Si se obtienen tiempos distintos con ambos vectores de iteraciones, realizar una justificación exhaustiva del resultado explicando a tu parecer lo que está sucediendo internamente en el procesador y la jerarquía de memoria

Para realizar un estudio de lo que está sucediendo en la jerarquía de memoria, debemos de calcular el stride cometido tanto con el vector de iteraciones (i,j,k) como con el vector de iteraciones (i,k,j).

Se conoce el stride como la ubicación en palabras de memoria consecutivas a través de la jerarquía de memoria.

Si calculamos el stride para el vector de iteraciones (i,j,k), obtenemos que dicho valor es 249. Por tanto, se ubican palabras en la jerarquía de memoria caché cada 249 líneas.

Si calculamos el stride para el vector de iteraciones (i,k,j), obtenemos que dicho valor es 1. Por tanto, se ubican palabras en la jerarquía de memoria caché cada 1 línea.

Con estos datos, podemos observar que la velocidad en memoria caché del vector de iteraciones (i,k,j) es mayor que el vector de iteraciones (i,j,k).