DidadoY

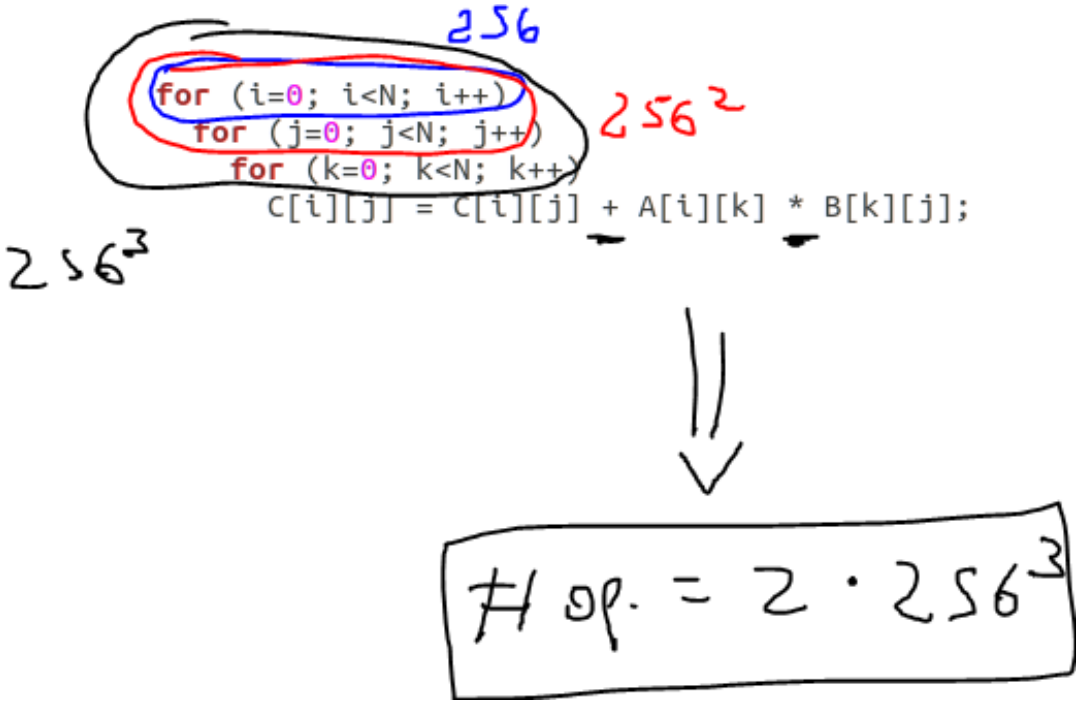
Sesión 8

1. Compilad y ejecutad los tres programas para un tamaño N=6. Comprobad que los 3 programas dan el mismo resultado.

Para los 3 códigos nos ha dado este mismo resultado:

```
8.345106   8.759226   9.173347   9.587466  10.001586  10.415707
8.633827   9.062347   9.490867   9.919386  10.347906  10.776426
8.922546   9.365466   9.808387  10.251307  10.694226  11.137147
9.211267   9.668587  10.125907  10.583227  11.040546  11.497867
9.499986   9.971707  10.443426  10.915147  11.386866  11.858586
9.788706  10.274826  10.760946  11.247066  11.733186  12.219306
```

2. Rellenad la siguiente tabla:



Tiempo de ejecución (sec)			MFLOPS		
Mm-ijk	Mm-jki	Mm-kij	Mm-ijk	Mm-jki	Mm-kij
0,21	0,15	0,09	161,58	217,49	389,62
1,49	1,41	1,11	180,45	190,84	242,17
14,45	41,17	6,30	148,57	52,16	340,62

3.Teniendo en cuenta lo que habéis hecho en los apartados anteriores y en el trabajo previo, explicad la razón de las diferencias de rendimiento en estos tres programas

Se debe al orden de los bucles del programa.. Dependiendo del orden, se accederá a las matrices horizontalmente (recorriendo todos los elementos de una fila, que es mucho más eficiente) o verticalmente (recorriendo todos los elementos de una columna, que es más ineficiente). Por este motivo, por ejemplo, conforme la N va siendo más grande, en el código mm-jki tarda muchísimo más que los demás, ya que a las 3 matrices se accede verticalmente.

4. Aplicad la optimización adicional a las otras dos aplicaciones. Compilad y ejecutad los tres programas para un tamaño N=6. Comprobad que los 3 programas dan el mismo resultado.

De nuevo, da el mismo resultado para los 3 códigos.

```
8.345106   8.759226   9.173347   9.587466  10.001586  10.415707
8.633827   9.062347   9.490867   9.919386  10.347906  10.776426
8.922546   9.365466   9.808387  10.251307  10.694226  11.137147
9.211267   9.668587  10.125907  10.583227  11.040546  11.497867
9.499986   9.971707  10.443426  10.915147  11.386866  11.858586
9.788706  10.274826  10.760946  11.247066  11.733186  12.219306
```

5. Rellenad la siguiente tabla:

Tiempo de ejecución (sec)			MFLOPS		
Mm-ijk	Mm-jki	Mm-kij	Mm-ijk	Mm-jki	Mm-kij
0,05	0,07	0,04	738,50	491,34	780,70
0,40	0,59	0,33	678,19	451,51	803,22
3,41	13,98	2,72	630,13	153,59	790,21

6. Comparad los resultados obtenidos con los obtenidos antes de optimizar los programas, y sacad conclusiones de dicha comparación

Debido a la optimización, los tiempos de ejecución en todos los casos se ha visto reducido. Creemos que es debido a que al utilizar una variable temporal que sustituye a la matriz que no se ve modificada en el último bucle, se disminuye los accesos a memoria, ya que se está accediendo a la variable local, y no a una posición de la matriz.