

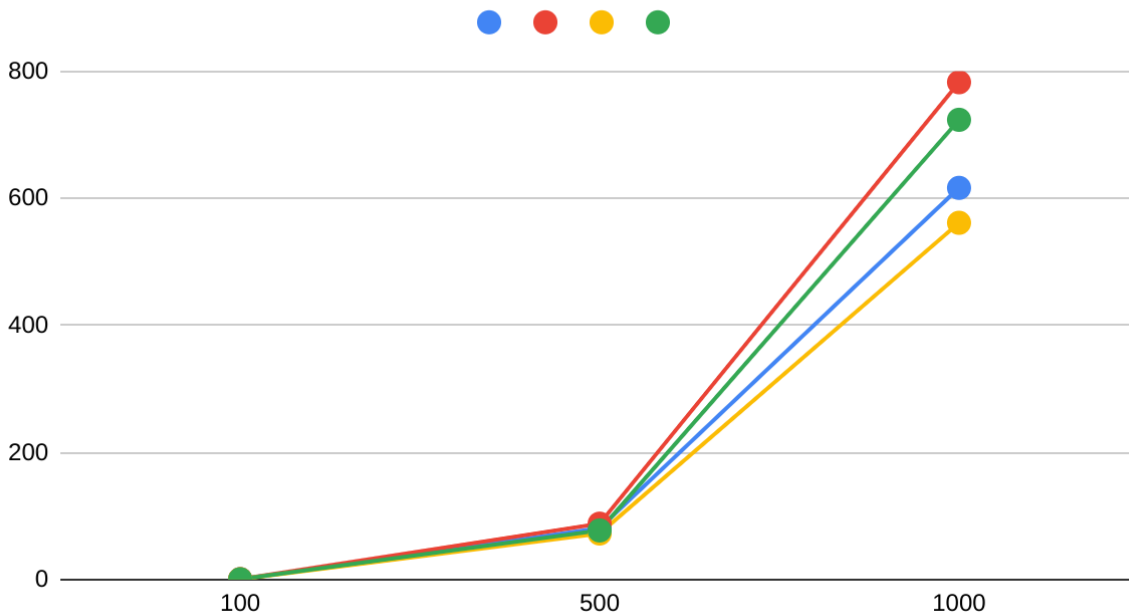
Práctica 1

En esta práctica he implementado la suma y el producto por matrices en forma flattened tanto por filas como por columnas. Además he utilizado la biblioteca multiprocessing de python para comparar los tiempos en computación secuencial y paralela. Como veremos a continuación, los tiempos del programa no son más rápidos utilizando multiprocessing. En el gráfico vemos como las líneas azul y naranja que se corresponden con el producto por filas y columnas sin multiprocessing son más rápidas que las líneas rojas y verdes que son los productos con multiprocessing. Lo que sí que podemos apreciar es que la forma escrita por columnas parece más rápida que por filas. Se han hecho pruebas con matrices de tamaños 100 x 100, 500 x 500 y 1000 x 1000. En el eje de ordenadas aparece el tiempo en segundos que ha tardado el programa en ejecutarse, donde la línea roja corresponde al producto por filas utilizando multiprocessing, la verde al producto por columnas utilizando multiprocessing, la azul al producto por filas sin multiprocessing y la amarilla al producto por columnas sin multiprocessing.

En la siguiente página de internet ocurre el mismo problema que a mí. En general las respuestas al problema que dan es que multiprocessing no implica que el programa sea más eficiente.

<https://stackoverflow.com/questions/50722291/using-multiprocessing-does-not-reduce-time-of-calculation>

Tiempo del programa en función del tamaño de las matrices

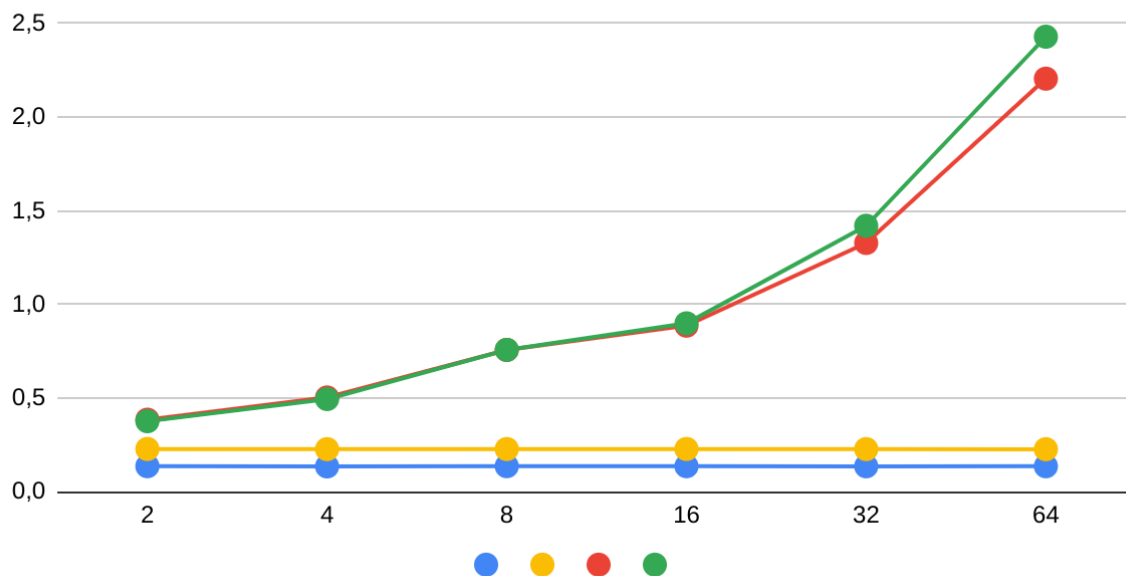


1

¹ Línea roja: multiplicación con notación por filas con multiprocessing.
Línea verde: multiplicación con notación por columnas con multiprocessing.
Línea azul: multiplicación con notación por filas sin multiprocessing.
Línea amarilla: multiplicación con notación por columnas sin multiprocessing.

Si ejecutamos el código con heracles (hasta 64 procesos de máximo) vemos que los tiempos no mejoran a medida que aumentamos el número de cpus. Para la siguiente tabla hemos cogido una matriz aleatoria de 5000*5000 elementos. Veamos qué pasa cuando aumentamos el número de cpus. Hemos probado con 2,4,8,16,32 y 64 cores. Vemos como en este caso, tal y como era de esperar el tiempo del programa no aumenta ni disminuye para el producto de matrices en forma secuencial. Sin embargo, en contra de lo que debería pasar, a medida que aumentamos los cpu, el tiempo en paralelo aumenta. En esta situación es ligeramente más rápido el producto de matrices expresadas por filas que por columnas.

Tiempo del programa para matrices de 5000x5000 en función del número de cores



2

² Linea roja: multiplicación con notación por filas con multiprocessing.
Linea verde: multiplicación con notación por columnas con multiprocessing.
Linea azul: multiplicación con notación por filas sin multiprocessing.
Linea amarilla: multiplicación con notación por columnas sin multiprocessing.