

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. No se a qué se deba, si los tiempos deberían ser más cortos. Aquí 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 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>

