INFORME DE LOS EJERCICIOS 2 Y 3

JON DORRONSORO MAIOZ

Enlace al git del código

BREVE EXPLICACIÓN

El ejercicio 2 constaba de dos partes, siendo la primera realizar el producto de dos matrices en serie y la segunda realizar la misma operación en paralelo. El ejercicio 3 constaba también de dos partes, en la primera había que escribir las matrices en un fichero txt, leerlas, realizar la multiplicación y finalmente escribir el resultado en un segundo txt. La segunda parte era realizar el mismo ejercicio pero con llamada MPI, **este apartado no ha sido realizado.**

El algoritmo implementado no es de los más eficientes ni por asomo, lo cual se observa en los resultados. La matriz A se divide equitativamente entre los procesadores mientras que la matriz B es mandada en su totalidad a cada uno de los procesadores. De esta forma cada procesador calcula X valores de la matriz resultado C.

Las mediciones de tiempo se realizan mediante el comando Linux de time: time ./programa, que calcula el tiempo total con exactitud de milisegundos.

RESULTADOS EN SERIE

Se han realizado experimentos con las matrices de los siguiente tamaños: 10X10, 100X100, 1.000X1.000, 2.000X2.000, 4.000X4.000, 5.000X5.000 y 10.000X10.000. Se puede observar que el tamaño máximo de una matriz es de 10.000X10.000, ya que este fue el valor en el que fue necesario abortar la ejecución del programa.

Los resultados de tiempo serán tratados en el apartado de COMPARACIÓN y CONCLUSIÓN.

RESULTADOS EN PARALELO

Se han realizado experimentos con las matrices de los siguiente tamaños y cantidad de procesadore (formato **TxT|P**, siendo T el tamaño y P el número de procesadores): 10X10|8_16_32, 100X100|8_16_32, 1.000X1.000|8_16_32, 2.000X2.000|8_16_32, 4.000X4.000|8_16_32, 5.000X5.000|8_16_32 y 10.000X10.000|8_16_32. Se puede observar que el tamaño máximo de una matriz es de 15.000X15.000, ya que las superiores a esta comienzan a dar errores con el scatter y segmentation faults o no terminan en tiempos mínimamente razonables, debido a esto el experimento se ha centrado en repetir y comparar resultados entre los distintos programas.

Los resultados de tiempo a comparar serán tomados de los apartados del ejercicio 2, ya que los del ejercicio 3 no optimizan el cálculo y su ejecución es más costosa. Sus resultados serán recogidos en una tabla aparte.

Los resultados de tiempo serán tratados en el apartado de COMPARACIÓN y CONCLUSIÓN.

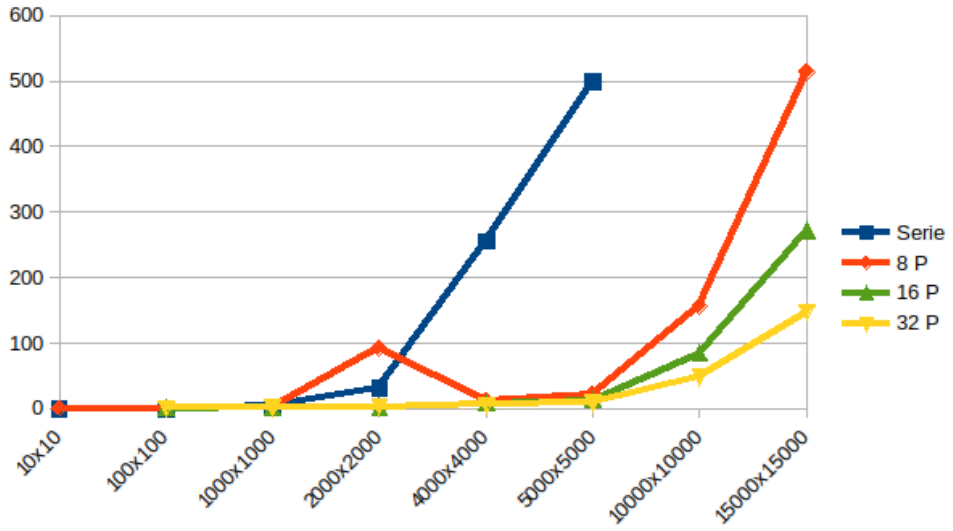
COMPARACIÓN Y CONCLUSIÓN

En la siguiente tabla podemos observar el tiempo de ejecución respecto del tamaño de las matrices y la cantidad de procesadores:

TAMAÑO	SERIE	8 P	16 P	32 P
10X10	0.001s	0.841s	-	-
100X100	0.006s	0.814s	0.916s	1.245s
1.000X1.000	4.005s	1.122s	1.531s	1.997s
2.000X2.000	31.689s	2.580s	2.341s	2.872s
4.000X4.000	4m 16.033s	11.961s	7.622s	6.061s
5.000X5.000	8m 18.899s	21.956s	12.933s	9.162s
10.000X10.000	**ABORTO***	2m36.517s	1m 23.673s	49.203s
15.000X15.000	-	8m33.216s	4m 31.548s	2m 27.362s

*A los 38 minutos de ejecución fue abortado

Observamos claramente que antes de las matrices de 1.000X1.000 no es rentable paralelizar los procesos. Además se ve que antes de los 2.000X2.000 es mejor utilizar 8 procesadores y que solo a partir de 4.000X4.000 empieza a ser útil usar 32.



Finalmente podemos observar la tabla de los resultados obtenidos con el ejercicio 3a. Como se ve los resultados son peores que con el ejercicio 2 en paralelo ya que este crea, lee y escribe las matrices en fichero, lo cual lastra su ejecución.

TAMAÑO	8 P	16 P	32 P
10X10	0.771s	-	-
100X100	0.794s	0.958s	1.153s
1.000X1.000	1.836s	2.197s	2.649s
2.000X2.000	5.161s	5.016s	5.611s
4.000X4.000	23.121s	18.076s	16.520s
5.000X5.000	40.489s	29.391s	25.629s
10.000X10.000	3m 40.305s	2m 25.517s	1m 49.359s
15.000X15.000	10m 50.387s	6m 44.318s	4m 44.259s