



Laboratorio 3 - Paralelismo con OpenMP

Nombre: John Serrano Carrasco

Curso: Sistemas Distribuidos y Paralelos

Sección: 13329-0-A-1

Profesor: Fernando Rannou Fuentes

19 de Noviembre de 2023

Tabla de contenidos

1. Resultados	1
1.1. Imagen resultante	1
1.2. Rendimiento Computacional	1
1.2.1. Tiempos de ejecución	1
1.2.2. Speedup	2

1. Resultados

Los siguientes resultantes fueron alcanzados utilizando los siguientes datos de entrada: Archivo de entrada = **hltau_completo_uv.csv**, $N = 2048$ y $d = 0.003$.

1.1. Imagen resultante

Una vez que se tienen los archivos de salida de la parte real y la parte imaginaria, se puede crear un código de MATLAB para comprobar la solución. Tras cargar los archivos en MATLAB y ejecutar el código, la imagen resultante tras aplicar ambas soluciones es la siguiente:

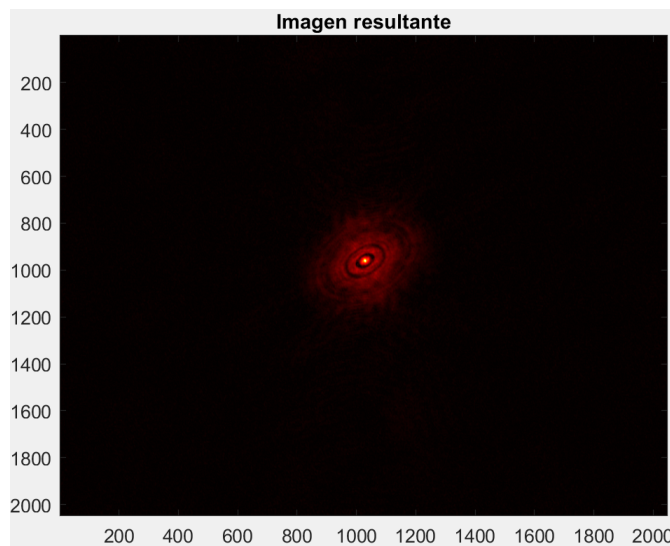


Figura N°1: Imagen resultante en MATLAB.

1.2. Rendimiento Computacional

Se realizaron pruebas con ambas soluciones en un computador de los laboratorios del DIINF.

1.2.1. Tiempos de ejecución

Se realizaron 10 iteraciones con ambas soluciones y luego se obtuvo el promedio, utilizando un chunk de 10000 y cambiando el número de tareas desde 2 hasta 16. Cabe

destacar, de que para poder comparar los resultados con un tiempo secuencial, se implementó una solución secuencial para el proceso de gridding, el cual no usa OpenMP, hebras ni tareas.

A continuación, se presenta una tabla con los tiempos de ejecución de las soluciones. Dada la naturaleza de la solución secuencial, se considera el mismo tiempo secuencial para ambas soluciones.

N° Tasks	Tiempo Matriz Compartida [s]	Tiempo Matriz Propia [s]
1	3,930471	3,930471
2	2,302032	2,276684
3	1,674599	1,663639
4	1,369062	1,429237
5	1,360925	1,450596
6	1,335057	1,503803
7	1,323506	1,519045
8	1,309548	1,560055
9	1,313798	1,614342
10	1,315954	1,663935
11	1,310474	1,719175
12	1,308001	1,772219
13	1,31049	1,824375
14	1,310657	1,879091
15	1,313216	1,933812
16	1,313828	1,985539

Cuadro 1: Tiempos de ejecución del proceso de gridding diferentes cantidades de tareas, utilizando un chunk de 10000.

1.2.2. Speedup

De lo anterior se puede obtener el speedup, considerando la siguiente formula:

$$\frac{T_s}{T_p} \quad (1)$$

Donde T_s corresponde al tiempo de la solución secuencial mientras que T_P corresponde al tiempo de la solución paralela. Aplicando esto, se pueden obtener los speedups al ejecutar la solución paralela con una cierta cantidad de tareas.

N° Tasks	Speedup Matriz Compartida	Speedup Matriz Propia
2	1,707391991	1,726401644
3	2,347111756	2,362574453
4	2,870922573	2,750048452
5	2,888087881	2,709555934
6	2,944047333	2,613687431
7	2,969741731	2,587461859
8	3,001395138	2,519443866
9	2,991685936	2,434720152
10	2,986784492	2,362154171
11	2,999274308	2,286254163
12	3,00494495	2,217824659
13	2,99923769	2,154420555
14	2,998855536	2,091687417
15	2,993011812	2,032499023
16	2,991617624	1,979548626

Cuadro 2: Speedups utilizando los tiempos de ejecución de la tabla 1.

Para poder ver mejor el comportamiento del speedup, se realizan gráficos con el apoyo de Python y Matplotlib. En primer lugar, el gráfico resultante para la solución con matriz compartida es el siguiente:

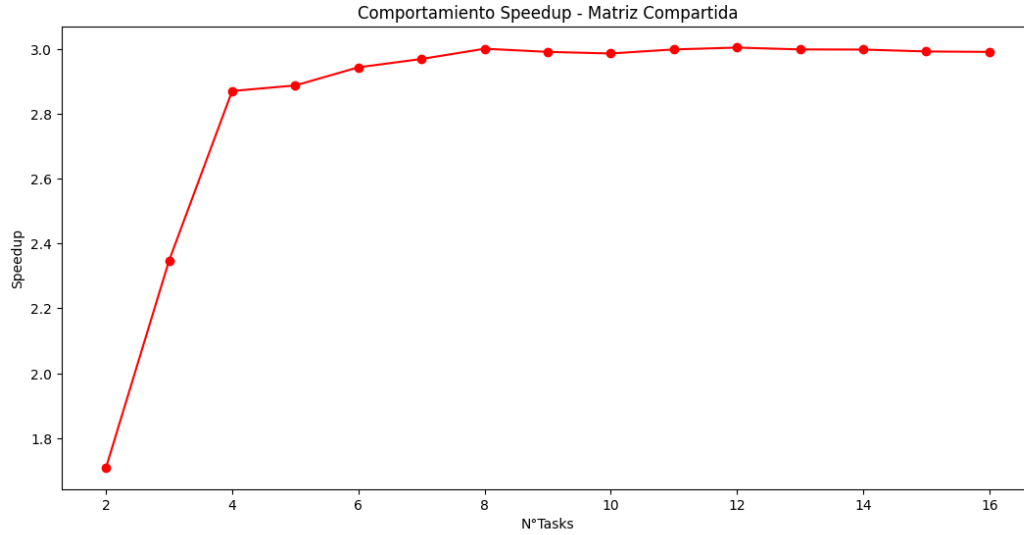


Figura 1: Gráfico de Speedup - Matriz Compartida.

Se puede ver como en esta solución, el speedup comienza a mejorar, hasta que aproximadamente llega a un límite luego de que el número de tareas supera a 8. El mejor speedup se obtiene cuando se utilizan 8 tareas, probablemente dado al hecho de que el computador en el que se hicieron las pruebas para obtener los tiempos de ejecución, tiene 8 procesadores. No se puede apreciar un momento claro donde el speedup pasa a ser negativo, dado que el speedup entre 9 a 16 tareas tiene un valor muy similar. Sin embargo, dada la ley de Amdahl, es claro que en algún punto el speedup dejará de estar cercano al límite y comenzará a empeorar.

En el caso de la matriz privada, el speedup se comporta de la siguiente manera a medida que aumenta el número de tareas:

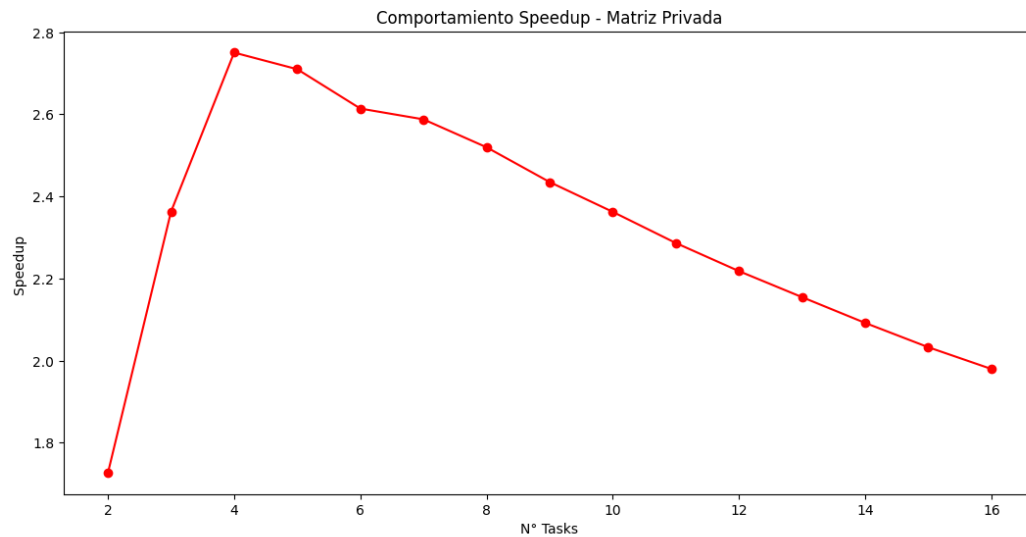


Figura 2: Gráfico de Speedup - Matriz Privada.

Se puede ver claramente que el speedup llega a un peak con 4 tareas y luego comienza a disminuir cada vez más. Esto se puede deber dado que tener más tareas implica que se utiliza más tiempo para la creación de matrices y las operaciones entre matrices.

Esto permite concluir que la mejor solución es la solución con matriz global, dado que se pueden obtener menores tiempos de ejecución a medida que aumentan las tareas, a diferencia de la solución con matriz privada donde el speedup comienza a bajar y los tiempos de ejecución comienzan a aumentar luego de cierto punto.