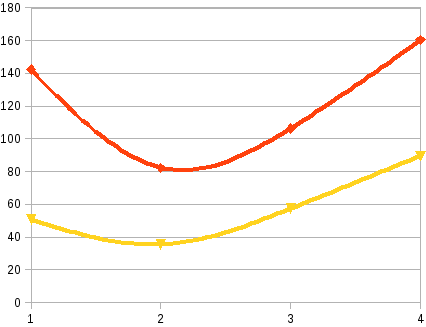
Los resultados de tiempo (en milisegundos) para diferentes tamaños de matrices cuadradas y un número de procesos variable en un procesador con 4 núcleos físicos y 8 lógicos son:

Ejercicio 1:

|  |  |  |
| --- | --- | --- |
| Tamaño de la matriz | Procesos | Tiempo |
| 200x200 | 1 | 50,760 |
| 200x200 | 2 | 35,381 |
| 200x200 | 4 | 57,317 |
| 200x200 | 8 | 89,612 |
| 300x300 | 1 | 142,416 |
| 300x300 | 2 | 82,361 |
| 300x300 | 4 | 106,347 |
| 300x300 | 8 | 160,431 |

Se puede observar que para ambos tamaños, a partir de 4 procesos se produce un empeoramiento de los tiempos de ejecución, debido a que es más el tiempo que se pierde en las comunicaciones entre procesos paralelos que el que se gana con la paralelización de la operación matemática.

A continuación podemos ver una gráfica en la que vemos la evolución del tiempo en función del número de procesos en una matriz de 300x300 y otra de 200x200.



Tiempo en ms matriz de 300x300

Tiempo en ms. matriz de 200x200

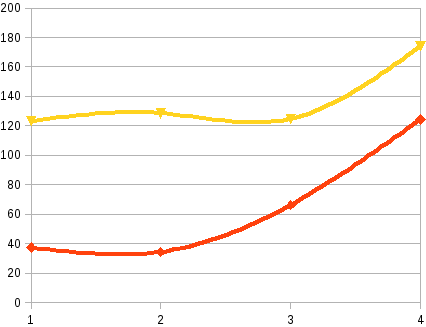
Número de procesos

Ejercicio 2:

|  |  |  |
| --- | --- | --- |
| Tamaño de la matriz | Procesos | Tiempo |
| 200x200 | 1 | 37,433 |
| 200x200 | 2 | 34,267 |
| 200x200 | 4 | 66,217 |
| 200x200 | 8 | 124,435 |
| 300x300 | 1 | 123,202 |
| 300x300 | 2 | 128,480 |
| 300x300 | 4 | 124,772 |
| 300x300 | 8 | 174,225 |

Llama la atención en este caso que se empeora el tiempo de ejecución en el caso de la matriz grande al aumentar el número de procesos (hay una pequeña mejora entre la ejecución con dos y cuatro procesos, pero de apenas 4 milisegundos), esto puede ser debido a que se utiliza una llamada MPI\_Barrier para sincronizar los procesos esclavos (se podrían haber utilizado también las llamadas MPI\_Wait o MPI\_Waitall, pero en algunas ejecuciones concretas se producían errores en tiempo de ejecución).

Se muestra a continuación una gráfica con los tiempos de ejecución mostrados en la tabla anterior



Tiempo en ms matriz 300x300.

Tiempo en ms matriz 200x200.

Número de procesos.