

# COMPUTACIÓN PARALELA Y CÁLCULO DISTRIBUIDO

## Entrega 3

Vojtěch Obhlídal

30 de junio de 2022

La **tercera entrega** se concentraba en la memoria distribuida. La primera parte era programar dos ejercicios sencillos y probar las topologías de estrella y anillo. Luego había que programar multiplicación de matrices por tercera vez - en este caso usándo la librería **MPI** con y sin operaciones colectivas. Al final alcancé solamente hacerlo sin operaciones colectivas. La segunda parte era evaluar el tiempo de computación de siguientes problemas:

1. Tiempo de ejecución de la versión del producto de matrices secuencial, sin OpenMP/MPI.
2. Tiempo de ejecución de la versión del producto que matrices que solo usa OpenMP con 2, 4 y 8 threads.
3. Tiempo de ejecución de la versión del producto que matrices que solo usa MPI con 4 nodos.
4. Tiempo de ejecución de la versión del producto que matrices que usa MPI+OpenMP con 4 nodos y 2, 4 y 8 threads.

Cuando se utiliza MPI (es decir, las tareas 3 y 4), el algoritmo de cálculo es el siguiente. Primero, el proceso 0 envía la parte correspondiente de las matrices de multiplicación a cada proceso. A continuación, cada proceso multiplicaba las matrices recibidas para obtener parte de la matriz producto resultante. Después, se pasó al proceso 0, que construyó la matriz resultante.

La parte interesante de esta tarea es especialmente la comparación del tiempo de cálculo entre diferentes combinaciones de MPI y OpenMP. Como ya hemos visto en la multiplicación con bloques, los diferentes valores de los parámetros pueden depender directamente unos de otros.

## 1. Evaluación

Para evaluar el tiempo de cálculo de cada tarea, elegimos un tamaño de matriz de 1500x1500. Resumimos los resultados en el Gráfico 1. Como esperamos, el cálculo es más lento sin utilizar OpenMP y MPI, es decir, con un solo hilo. El uso de OpenMP con 2, 4, 6 y 8 hilos hace que nuestro cálculo sea más rápido, como vimos en la Práctica 1.

Tercera tarea utilizando MPI con 4 procesos sin OpenMP, es decir, con un hilo. El tiempo resultante es muy similar al de la tarea dos utilizando 4 hilos. Así, sin más investigación, podríamos concluir que el aumento de velocidad usando OpenMP con 4 hilos se corresponde con el aumento de velocidad con MPI con 4 procesos. Sin embargo, se necesitaría una investigación más exhaustiva para confirmarlo.

La última tarea, la multiplicación de matrices utilizando MPI con 4 procesos y OpenMP con 2, 4, 6 y 8 hilos da los mejores resultados. De nuevo, es interesante observar un tiempo de cálculo similar con 2 hilos que en la tarea dos para 8 hilos. Bajo la hipótesis ya mencionada de que los procesos de MPI coinciden con los hilos de OpenMP este hecho tiene mucho sentido.

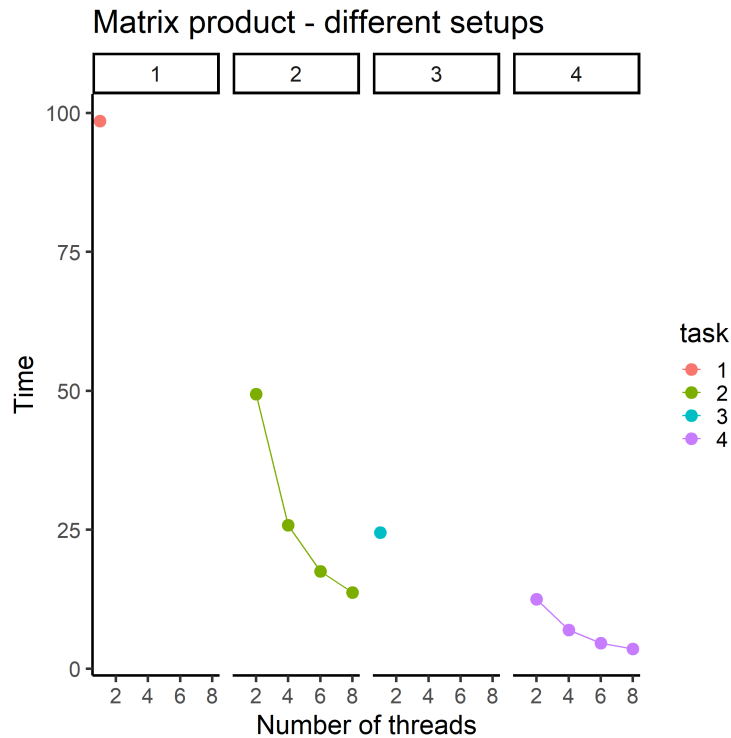


Figura 1: Evaluación de los problemas 1-4.