

Tarea 3

Curso: Computación Paralela

Estudiante: Javier Cordero Quirós

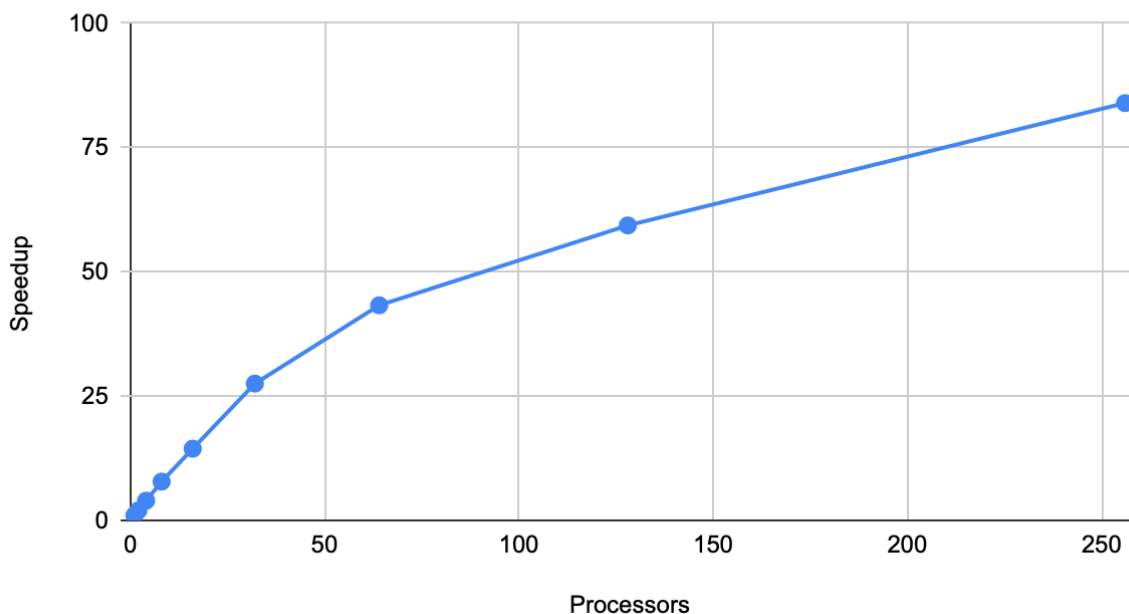
Fecha: 25/03/2022

Una estrategia general del esfuerzo de paralelización. ¿Qué patrones de programación paralela usó en la solución? ¿Qué operaciones de OpenMP parecieron las más apropiadas para el programa?

En la implementación paralela del algoritmo de Shear Sort se utilizaron los patrones: **Map** para la operación sobre matrices tales como liberar memoria y **Fork-join** para operaciones sobre diferentes filas y columnas. En cuanto a operaciones de OpenMP, se aplicaron pragmas de `omp parallel`, `omp for`, con especificaciones de variables `shared`, fueron especialmente útiles para ordenar filas y columnas a la vez y para crear valores random de la matriz al inicio.

Análisis de Speedup

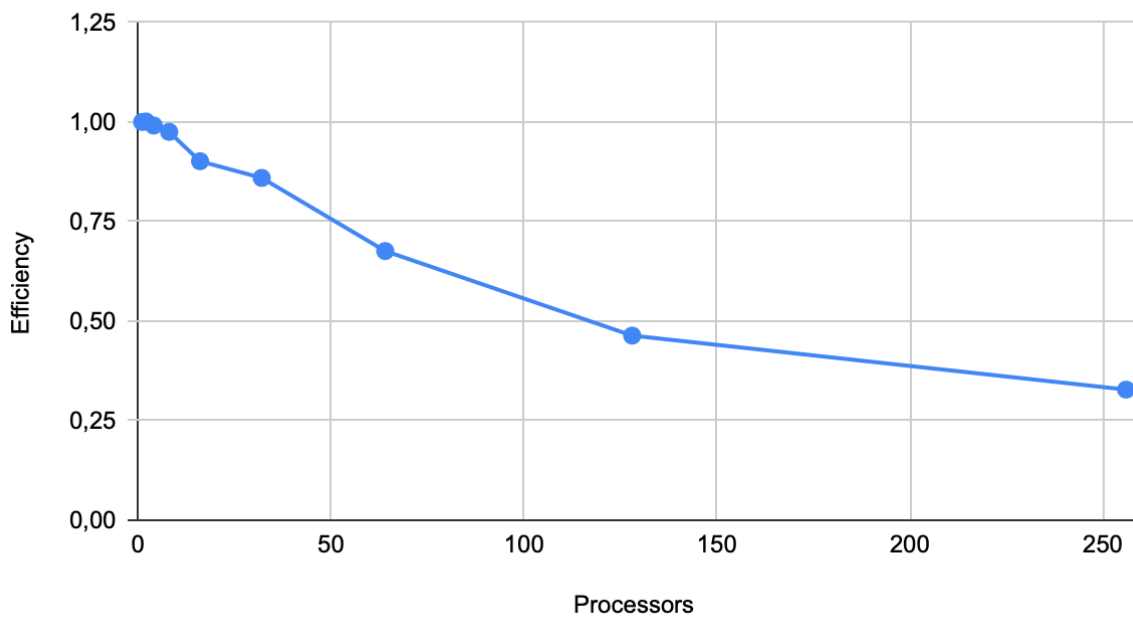
Gráfico de Speedup vs Processors



Como se puede observar en el gráfico, la versión paralela del programa logra aumentar la velocidad de ejecución hasta casi 32 veces para el caso de 256 workers. La tendencia del speedup es creciente pero no lineal, empieza a decaer a partir de los 32 workers. Una posible explicación sería que para valores más altos de cantidad de procesadores, el overhead de la distribución de tareas es mayor resultando en un peor speedup.

Análisis de Eficiencia

Gráfico de Eficiencia vs Processors



En cuanto al gráfico de eficiencia, como es de esperar, esta decae conforme aumenta la cantidad de entidades de procesamiento, esto debido a que en el programa existe una gran parte no paralelizable y cada vez se hace más difícil cumplir con la expectativa de speedup según la ecuación de eficiencia.