



Mergesort Paralelizada en c++

David Alsina y Nicolás Quintero

Febrero 2021

[Github](#)

1. Resumen

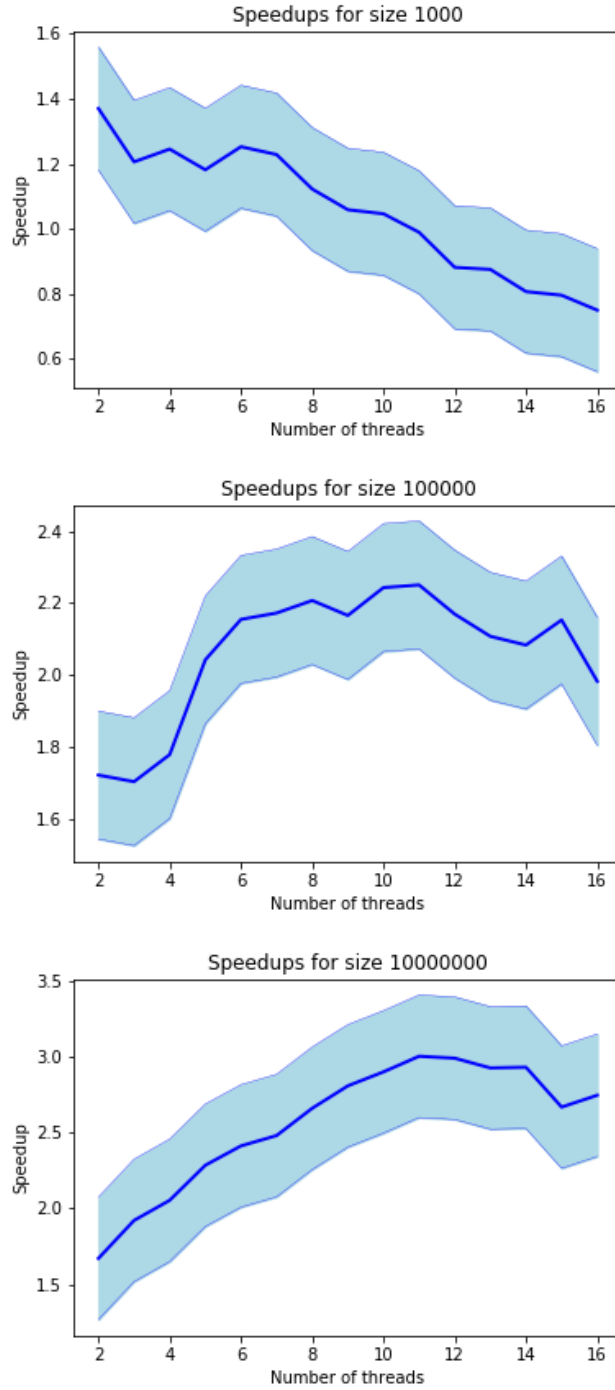
Para el desarrollo de este proyecto construimos el algoritmo *mergesort* y lo paralelizamos usando la librería *Open MP*^{1 2} y una modificación a este algoritmo para incrementar el rendimiento (*Se realiza un insertion sort cuando el tamaño del array llega a ser de 100 o menor*), adicionalmente para hacer posible el testeo de este algoritmo se usó computación en el cluster de la Universidad del Rosario³, se hicieron pruebas desde la versión serial (*un solo hilo*), hasta 16 hilos, usando un incremento de 1 hilo en cada iteración. *Los resultados se pueden ver a continuación.*

¹[página de la organización](#)

²[artículo introductorio a *open MP*](#)

³[Universidad del rosario](#)

2. Resultados de Speedups



En el caso de arreglos de tamaño 10^3 podemos observar que no se logra conseguir un speedup mayor a 1.4, y que a medida que el número de hilos aumenta, el speedup disminuye. Esto se debe a que el método utilizado para la paralelización del ordenamiento es tasking y al aumentar el número de hilos, el tiempo que tarda el hilo maestro en generar y asignar las tareas se vuelve mayor al tiempo que tarda el merge sort en ejecutarse sobre este tamaño del arreglo.

Para los arreglos de tamaño 10^5 y 10^7 logramos observar que el número de hilos con el que obtuvo un mejor speedup fue de 11.

3. Resultados de promedios y desviaciones estándar por hilo

