



Instituto Tecnológico y de Estudios Superiores Monterrey

CAMPUS QUERÉTARO

Implementación de métodos computacionales

Profesor:

Pedro Oscar Pérez Murueta

Grupo 601

Actividad Integradora 5.3: Resaltador de Sintaxis Paralelo

PRESENTA

Iván Ricardo Paredes Avilez

A01705083

Fecha:

26/05/2023

Reflexión

En esta entrega realizamos un programa de resaltador de sintaxis que tiene muchas similitudes con el de la entrega pasada, con la excepción de que está escrito en otro lenguaje de programación (en este caso C++) y que además, ahora aplica procesamiento paralelo usando diferentes hilos de trabajo. A pesar de haber implementado esta tecnología, la complejidad del algoritmo permanece como $O(n)$, igual que el resaltador pasado codificado en Racket ya que, en realidad, los pasos del algoritmo no cambian mucho y siguen dependiendo en gran medida de la cantidad de información que tengan los archivos que deben ser procesados (a más información, más demora). Sin embargo, al utilizar hilos de trabajo podemos imaginar la complejidad como $O(n/h)$ donde h equivale al número de hilos de trabajo implementados, pues el trabajo de procesamiento se dividirá según la cantidad de estos. Como tal, esta notación no es la oficial, pero nos ayuda un poco a entender por qué es más eficiente el programa cuando usamos hilos de trabajo.

Para asegurarnos de que la versión con hilos tiene un mejor rendimiento, ejecutamos en múltiples ocasiones el código en ambas versiones (con/sin hilos) y medimos el tiempo que tardaban en finalizar el procesamiento. A partir de ello, pudimos calcular el speedup entre versiones y pudimos descubrir que en efecto, la versión con hilos es más rápida que aquella versión que no los implementa. Sin embargo, algo a tener en cuenta es que los tiempos de ejecución nunca van a ser exactamente los mismos debido a pequeñas fluctuaciones en el funcionamiento del procesador o el tamaño de la entrada, por ejemplo.

Por lo que se refiere a las implicaciones éticas, cuando diseñamos un resaltador de texto tenemos la responsabilidad de que este funcione correctamente ya que es una herramienta en la que usualmente los usuarios confían demasiado y pocas veces ponen a prueba. Cualquier mal funcionamiento podría ocasionar que los usuarios cometan errores y además les sea más difícil y tardado encontrarlos. Además, muchos programadores están tan acostumbrados a una aplicación tan simple como esta, que sin ella, se les dificultaría programar.