Actividad 5.2 Computación Paralela

Oscar Jahir Valdés Caballero – A01638923 – ITC

Implementación de Métodos Computacionales

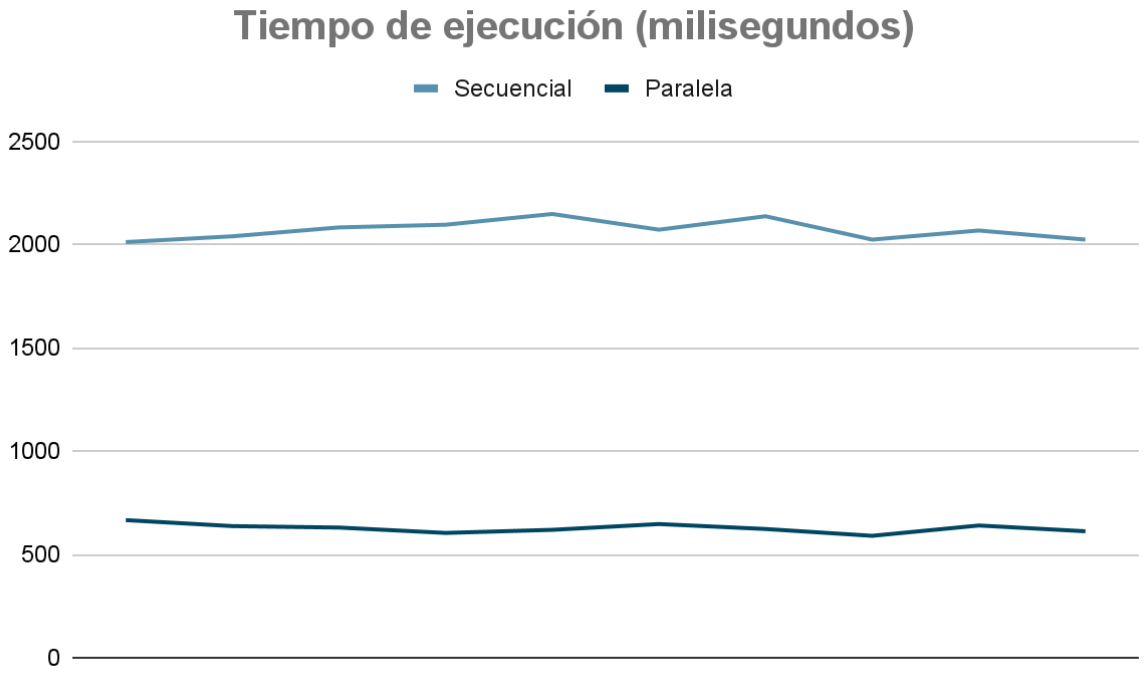
Group 2

Luis Ricardo Peña Llamas

06 de Mayo de 2022

Instrucciones:
Reemplazar la cadena directory dentro de los archivos parSynt.java y seqSynt.java que contienen el valor:
"C:\\Users\\oscar\\Documentos\\IMC_Proyecto\\"
por el path de la carpeta que contenga el proyecto. Posteriormente compilar ambos archivos con javac *nombreDelArchivo*.java y ejecutarlos con java *nombreDeLaClase*. Si se desea cambiar las expresiones regulares se puede hacer en el archivo patterns.txt dentro de la carpeta Scanner siguiendo las instrucciones del archivo readme.md que se encuentra en la misma carpeta.

Tiempo de ejecución:
Para comparar los tiempos de ejecución ejecute ambos programas diez veces y obtuve el promedio. El tiempo lo medí en milisegundos con la función *System.currentTimeMillis()* restando *tiempo final - tiempo inicial* dentro del método main. En ambos casos se tomaban los mismos tres archivos como inputs.



	Mínimo	Máximo	Promedio
Secuencial	86 ms	106 ms	96.1 ms
Paralela	77 ms	95 ms	86.2 ms

Speedup:

Mide el tiempo en que tarda en ejecutar cada versión del programa y calcula el speedup obtenido usando la siguiente f órmula:

$$S_p = \frac{T_1}{T_p}$$

En dónde:

- S_p es el speedup obtenido usando p procesadores.
- p es el número de procesadores o núcleos.
- T_1 es el tiempo que tarda en ejecutarse la versión secuencial del programa.
- T_p es el tiempo que tarda en ejecutarse la versión paralela del programa utilizando p procesadores.

Reemplazando los datos en la fórmula obtenemos

$$S_8 = \frac{96.1}{86.2} = 1.11$$

El trabajo fue desafiante ya que para resolverlo fue necesario aprender mucho sobre java y sobre procesos. Creo que fue una actividad muy enriquecedora y que la solución es muy inteligente y eficiente porque emplea el scanner léxico que cree en una actividad anterior. La parte difícil no fue trabajar con los hilos, ya que en este caso no compartían recursos de escritura. En realidad lo difícil fue alimentar el archivo .txt al archivo .exe con la clase ProcessBuilder ya que no tenía experiencia con ella.

Los algoritmos de la solución son muy veloces y tienen una complejidad lineal ya que leen el archivo carácter por carácter sin repetir o saltar carácteres. La versión paralela también es de complejidad O(n), pero es más rápida que la versión secuencial por un factor constante de 1.11. Si la cantidad de archivos leídos fuese mayor este factor podría aumentar, y alcanzaría su máximo cuando la cantidad de archivos sea igual o mayor que la cantidad de núcleos.

Conclusión:
Al realizar las pruebas con pocos archivos la diferencia no fue sustancial, ya que no se logró aprovechar todos los núcleos. Esta actividad me demuestra que es posible leer información de muchos archivos a la vez de manera muy veloz. Una posible implicación ética sería la de un programa malicioso que pueda encriptar los archivos de una víctima y cobrar para que ésta los pueda recuperar, como ocurrió con el ataque de WannaCry. También podría ser que un programa se ponga a leer los archivos de una víctima en busca de contraseñas o información sensible.

Creo que debido al poder que tiene esta tecnología y a su capacidad de causar daño es importante que se eduque a las personas para evitar esta

clase de ataques y que las empresas o asociaciones que desarrollan estas tecnologías sean responsables y publiquen actualizaciones con regularidad para incrementar la seguridad de los usuarios.