UNIVERSITAT DE LLEIDA
Escola Politècnica Superior
Grau en Enginyeria Informàtica
Sistemes Concurrents i Paral·lels

Práctica 2

Joaquim Picó Mora, Ian Palacín Aliana
PraLab1

Professorat : F. Cores
Data : 3 de Desembre 2019

Índex

1 Introducción

En este documento se compara la eficiencia en tiempo que supone ejecutar Indexing i Query de forma concurrente respecto de forma secuencial. También se muestra a continuación la diferencia en tiempo que supone el paralelismo de hilos durante la ejecución del programa.

Los datos que se utilizaran para hacer el estudio saldrán de ejecutar de forma secuencial y concurrente los mismos ejemplos, calculando así el tiempo en que tarda en realizarse cada una de las ejecuciones. Este tiempo será el que luego se usará para comparar y sacar conclusiones.
De mismo modo se ejecutará varias veces el programa con distinto número de hilos de ejecución y se procederá a la comparación de los resultados.

2 Concurrente vs Sequencial

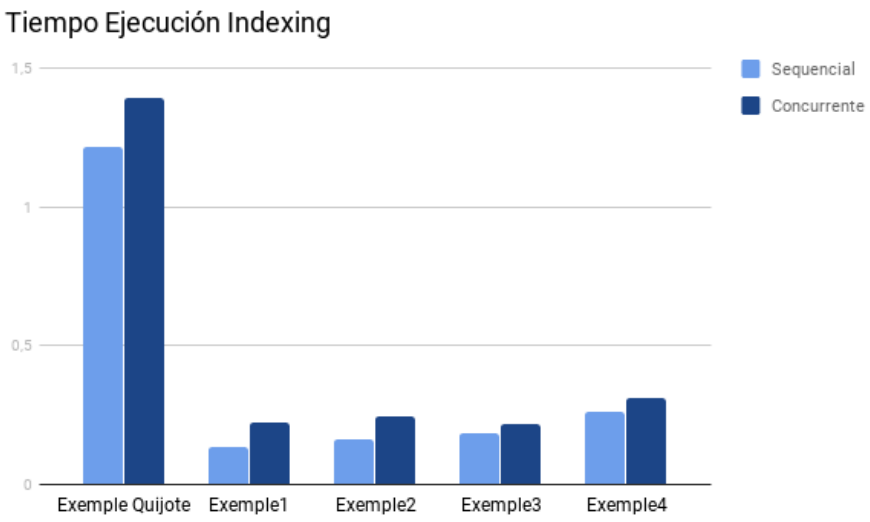

Figura 1: Indexing

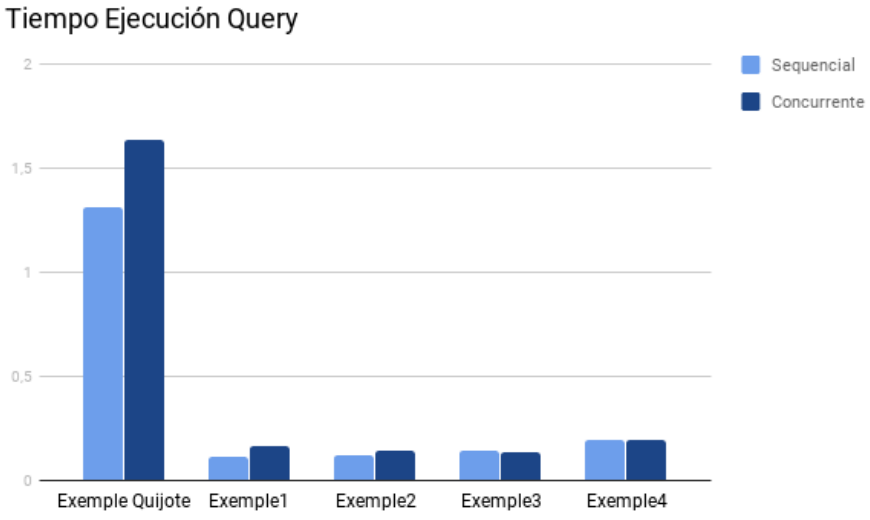

Figura 2: Query

El objetivo de la práctica ha sido implementar la versión concurrente de Query y Indexing, así como estudiar su comportamiento y tiempos de ejecución.

En las gráficas se pueden discernir los tiempos de ejecución de cada uno de los ejemplos en su versión secuencial y concurrente, sin embargo, cabe destacar algunas peculiaridades sobre los resultados obtenidos.

El resultado más destacable es que en algunos casos la aplicación concurrente nos da un tiempo superior a la implementación secuencial, y esto se debe por lo siguiente: Nuestra implementación tanto en Query como en Indexing se basa en una estrategia Master/Worker que consiste en dividir la construcción del HashMap de Inverted Index en diferentes hilos, haciendo que cada hilo construya su propio HashMap. Posteriormente unimos los HashMaps de cada uno de los hilos para generar el definitivo. Esta última operación genera un cuello de botella haciéndola bastante costosa en tiempo, cosa que explica porque a veces la aplicación concurrente es menos eficiente que la secuencial.

3 Diferencias entre multiples hilos

Como se puede ver en la Figura 3, en el caso tanto de Query como de Indexing conforme mas hilos ejecutándose de forma paralela, más coste en el tiempo.
Esto es debido a lo ya mencionado en el apartado anterior, a más hilos más veces se tiene que realizar la operación putAll alargando más el cuello de botella.

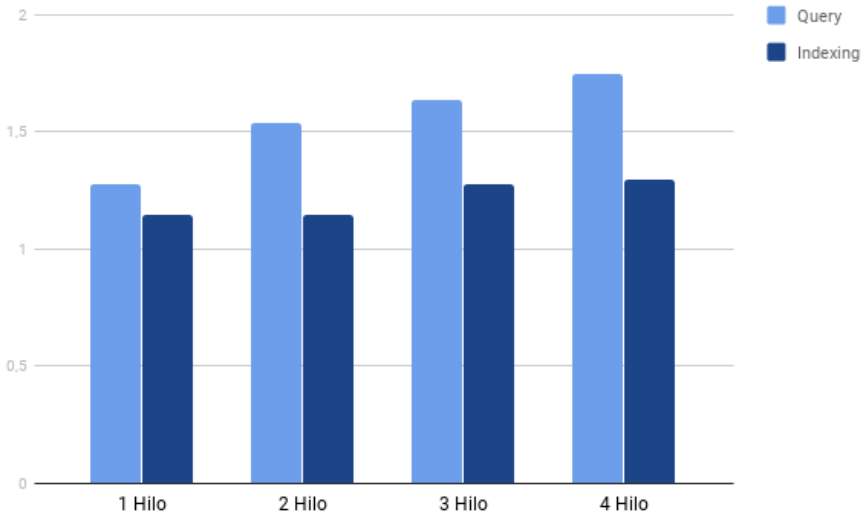

Figura 3: Ejecucion con múltiples hilos

4 Diseño de la Solución

La solución presentada en esta práctica consiste en lo siguiente.

4.1 Indexing

Se reciben por parámetro el número de hilos con los que se desea ejecutar la aplicación. Con este se realiza un balanceo de carga asignando a cada hilo un número determinado de caracteres a procesar del fichero. Una vez asignado a cada hilo la carga de trabajo a realizar, estos empiezan a procesar carácter a carácter las claves i a construir su propio HashMap. Una vez realizado el join de todos los hilos, se juntan los HashMaps parciales en uno global i se continua con la ejecución del programa.

4.2 Query

Utiliza la misma estructura que Indexing, pero esta vez el balanceo de carga se realiza respecto a los ficheros que deben leer cada uno de los hilos. Entonces cada thread leerà un número de fitxeros y creará su propio hashmap así continuando con el mismo proceso que Indexing.