

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

Práctica 1

Quim Picó Mora, Ian Palacín Aliana
PraLab1

Professorat : F. Cores
Data : 18 d'Octubre 2019

Índex

1	Introducción	1
2	Concurrente vs Sequential	1
3	Diferencias entre multiples hilos	2
4	Diseño de la Solución	3

1 Introducción

En este documento se compara la eficiencia en tiempo que supone ejecutar la aplicación calcArboles de forma concurrente respecto de forma secuencial. También se muestra a continuación la mejora 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 Secuencial

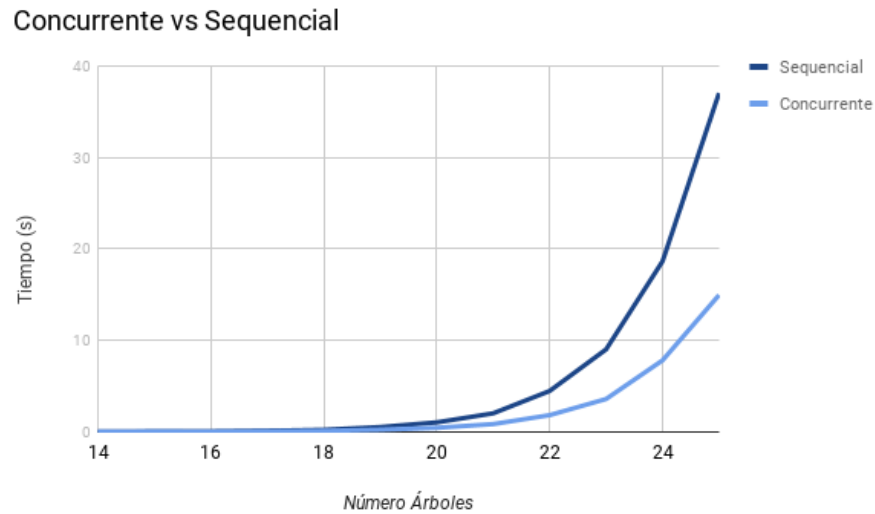


Figura 1: Concurrente vs Secuencial

El objetivo de la práctica es que la versión concurrente sea más rápida que la versión secuencial. Aunque esto se pueda discernir claramente al mirar la gráfica, cabe destacar algunas peculiaridades sobre los resultados obtenidos.

Como podemos ver, tanto la versión secuencial como la concurrente crece de forma exponencial en función del número de árboles que tiene el problema. Aunque con la versión concurrente consigamos un menor tiempo, esta también tenderá a crecer exponencialmente, puesto que el número de combinaciones en función de n árboles crece también de forma exponencial (2 elevado a n).

También se puede ver como la diferencia de tiempo es mayor cuanto mayor es el número de árboles. Esto es debido a que, cuanto más árboles hay, mayor es el cómputo que se hace de forma concurrente y por tanto, mayor el impacto en la diferencia de tiempo. De la misma forma, cuanto menos árboles hay, mayor es la proporción de código que se ejecuta de forma secuencial, como por ejemplo, la creación de hilos o el tratamiento de argumentos, y sale menos a cuenta que se ejecute concurrentemente.

3 Diferencias entre multiples hilos

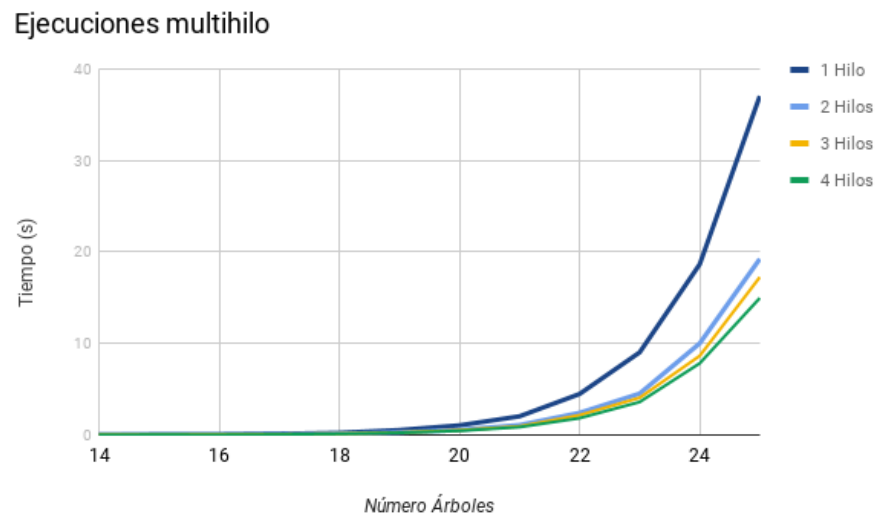


Figura 2: Ejecucion con múltiples hilos

Como se puede ver en la Figura 2, conforme mas hilos ejecutándose de forma paralela más eficiente es la ejecución del programa en términos de tiempo.

Se puede observar también, que a medida que va aumentando el número de threads la mejora de eficiencia es menor. En la gráfica se puede ver que el cambio de ejecutar la aplicación de forma secuencial a paralela con dos hilos, mejora casi al doble la eficiencia. Mientras que al cambiar de una ejecución paralela de dos hilos a una de tres, no supone un cambio tan drástico.

4 Diseño de la Solución