

PROGRAMACIÓN CONCURRENTE Y DE TIEMPO REAL

## Práctica 5 - Análisis1

*Nicolás Ruiz Requejo*



## Índice

1	Búsqueda paralela de números primos en un rango de naturales dados . . . . .	<b>2</b>
1.1	Tipología del problema . . . . .	2
1.2	Tipo de solución . . . . .	2
1.3	Aplicación de la ecuación de Subramaniam . . . . .	2
2	Descarga paralela de diversas páginas web . . . . .	<b>2</b>
2.1	Tipología del problema . . . . .	2
2.2	Tipo de solución . . . . .	2
2.3	Aplicación de la ecuación de Subramaniam . . . . .	2

## 1. Búsqueda paralela de números primos en un rango de naturales dados

### 1.1. Tipología del problema

La búsqueda de números primos tratada en esta práctica es una problema computacional intensivo que no se bloquea a la espera de operaciones de entrada/salida.

### 1.2. Tipo de solución

La solución planteada divide el rango de números enteros sobre el que producimos la búsqueda de primos en particiones disjuntas de enteros según el número de hilos que se ejecuten. Se dividen las tareas con su rango para cada hilo y se ejecutan en un pool de threads. Una vez añadidas todas las tareas al pool, el hilo principal espera a que finalicen todas y muestra los resultados de la ejecución.

### 1.3. Aplicación de la ecuación de Subramaniam

En este problema la ecuación de Subramaniam se aplica considerando un coeficiente de bloqueo igual a cero ( $Cb = 0$ ) porque es un problema no bloqueante en I/O. Por lo tanto el número de hilos que se lanzan a ejecución es igual al número de núcleos de la máquina donde se ejecuta. Finalmente el reparto de tareas se hace equitativamente entre el tamaño del problema (rango  $[0, \text{maximoenterobuscado}]$ ) y el número de núcleos.

## 2. Descarga paralela de diversas páginas web

### 2.1. Tipología del problema

El problema planteado consiste en la visita varias URL a páginas web para su descarga y guardado en disco. Es un problema de uso intensivo de operaciones de entrada/salida a red y disco, por lo que es bloqueante.

### 2.2. Tipo de solución

La solución planteada crea una tarea por cada URL leída de un fichero de texto que se encargará de conectar con el sitio web, descargar el contenido de la página y guardarlo en un fichero .html en disco. Se utiliza un pool de threads con un número fijo de hilos para poner en ejecución las tareas.

### 2.3. Aplicación de la ecuación de Subramaniam

Se aplica la ecuación para calcular el número de hilos según el coeficiente de bloqueo que proporcionaremos como parámetro en línea de comandos al programa.

De esta forma podemos probar distintos coeficientes de bloqueo, obteniendo los resultados de tiempo de ejecución para cada uno. Y así estimar experimentalmente el coeficiente de bloqueo del problema que será aquel con el que obtengamos el mínimo tiempo de ejecución.

Probamos en el rango  $Cb \in [0, 0,99]$  en incrementos de 0,05 y obtenemos los siguientes resultados:

Coeficiente de bloqueo	Tiempo paralelo en segundos
0	4,27
0,05	3,48
0,10	4,40
0,15	3,82
0,20	3,76
0,25	3,73
0,30	3,12
0,35	3,95
0,40	3,51
0,45	4,53
0,50	3,79
0,55	3,21
0,60	2,92
0,65	3,46
0,70	3,19
0,75	3,30
0,80	3,52
0,85	3,25
0,90	2,88
0,95	2,57
0,99	3,44

A la vista de los datos obtenidos estimamos el coeficiente de bloqueo entre 0,9 y 0,95