Programación Concurrente y de Tiempo Real

PRÁCTICA 3

Mª Teresa Supervielle Sánchez

1. Ejercicio 2: matVector.java y matVectorConcurrente.java

SECUENCIAL			**PARALELO**		
nº Ejecución	**tiempo (ms)**	**(%) CPU**	**nº Hebras**	**tiempo (ms)**	**(%) CPU**
1	72 ms	31 %	2	51 ms	44 %
2	74 ms	32 %	4	48 ms	42 %
3	65 ms	28 %	8	54 ms	41 %
4	73 ms	29 %	10	52 ms	41 %

Cuadro 1: Resultados en Windows con un tamaño de n = 10^4

2. Ejercicio 3: continuación del Ejercicio 2 en Linux

SECUENCIAL			**PARALELO**		
nº Ejecución	**tiempo (ms)**	**(%) CPU**	**nº Hebras**	**tiempo (ms)**	**(%) CPU**
1	73 ms	32 %	2	101 ms	48 %
2	75 ms	38 %	4	98 ms	35 %
3	101 ms	37 %	8	130 ms	48 %
4	92 ms	42 %	10	138 ms	52 %

Cuadro 2: Resultados en Linux con un tamaño de n = 10^4

Una vez realizada ambas pruebas (en mi caso, siendo Windows mi SO principal, y haciendo uso de Linux a partir de una Máquina Virtual), construí las dos tablas comparativas que se muestran a continuación. Partiendo de las mismas he llegado a la siguiente conclusión: aunque los resultados son muy similares a grandes rasgos si existen ciertos matices. En cuanto a tiempo de ejecución, Windows es considerablemente más rápido que Linux, siendo en este último el doble de tiempo en todos los casos; esto se puede deber a la versión donde están siendo ejecutados, ya que, como he comentado antes, Windows es mi sistema principal y lo mantengo actualizado a la última versión. Por otra parte, el uso de CPU es menor en Linux, en algunas ocasiones, esto es debido a que el propio sistema de Windows consume mucho % dde CPU. Finalmente si ambos programas se hubiesen ejecutado en igualdad de condiciones (es decir que Linux no se hubiese utilizado a partir de una Máquina Virtual), Linux hubiese obtenido mejores resultados.

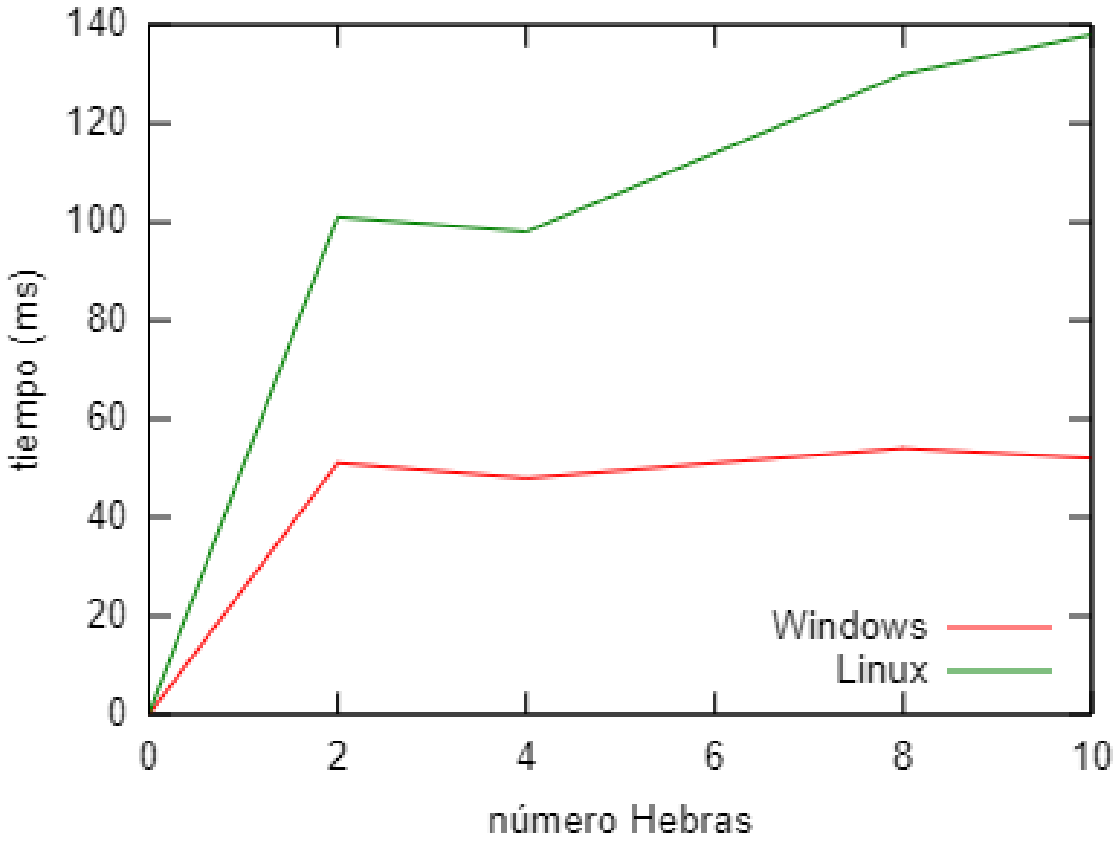

Figura 1: Curva de tiempo resultante

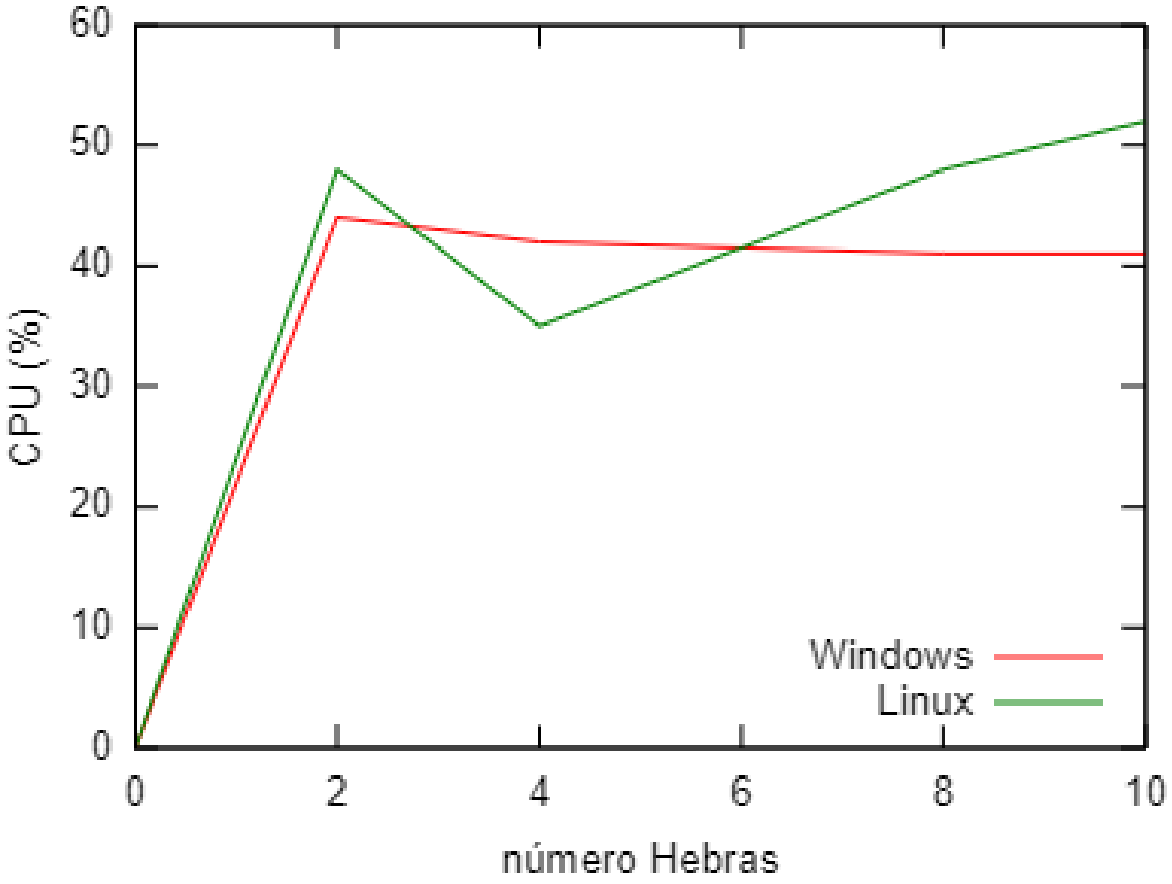

Figura 2: Curva del uso de CPU resultante