



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN
IIC3524 - Tópicos avanzados de sistemas distribuidos

Tarea 1

16 de mayo de 2017

Nicolás Gebauer - 13634941

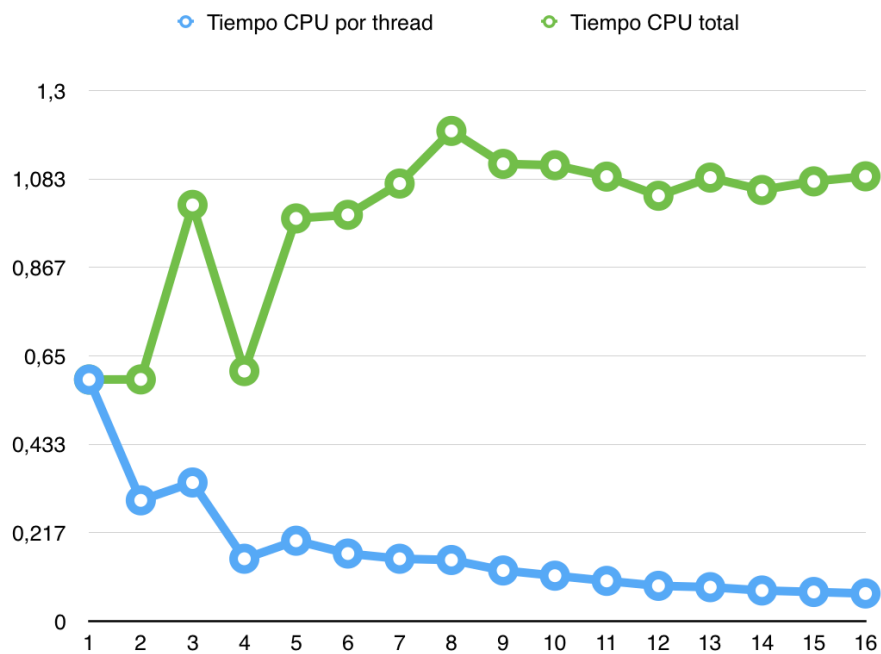
@negebauer - repo T1

Análisis de rendimiento

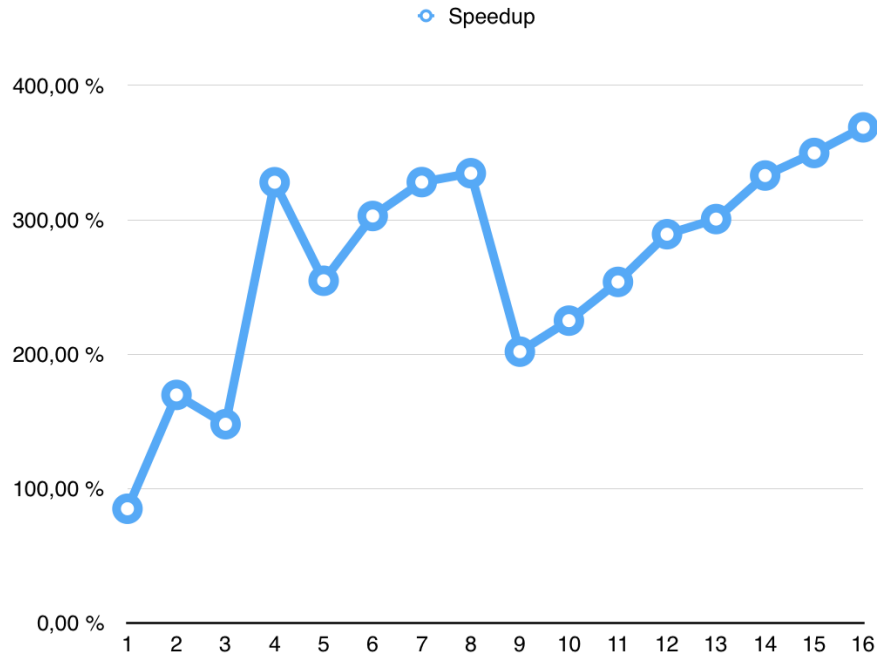
Se ejecutó el programa en su versión secuencial y paralela con distinto número de threads (de 1 a 16) con las imágenes *big.png*. Los resultados de tiempo se presentan a continuación, junto con los speedups que representan. Importante notar que la versión secuencial tomó 0,502388 segundos (los tiempos se miden solo en el proceso de aplicar el filtro). El comando usado para generar estos datos es:

```
./run.sh test/img/big.png test/kernel/blur_3.txt 3
```

Antecedido por `OMP_NUM_THREADS=n` con n de 1 a 16.



Tiempos CPU con 1 a 16 threads. Comando usado:



Speedup con 1 a 16 threads. Comando usado:

Discusión

Se aprecia que al aumentar el número de threads disminuye el tiempo que ocupa la CPU cada thread, esto es esperable ya que al dividir la tarea en más procesos cada proceso tendrá que trabajar menos. También el tiempo total de uso de la CPU aumenta al aumentar el número de threads. Esto también es esperable ya que al tener más procesos que corren sobre la CPU el uso total es mayor. Es importante destacar que este tiempo no es cuanto demora el programa, sino más bien cuanto tiempo de cómputo usa de la CPU (en su totalidad). A partir de los 8 threads este valor se estabiliza. Esto puede deberse a que a partir de los 8 threads la CPU se utiliza por completo. Por lo tanto, el tiempo total de uso ya no se puede disminuir. Tener más threads solo hace que cada thread trabaje menos, pero el uso total de la CPU se mantiene.

Lo importante es el speedup, el cual tiene a disminuir al aumentar la cantidad de threads. Se observa que al pasar de 2 a 3 y de 4 a 5 el speedup disminuye. Esto puede deberse a que el paralelismo funciona mejor con potencias de 2.