



Imagine - Filtros de imagen

11/06/2022

Joaquin Nuñez y Galo Bello Ponce

Ins. Ind. Luis A. Huergo

Programación sobre redes

- **Considerando un filtro particular, comparar la performance de ejecución en su versión Single-thread y Multi-thread**

En este caso utilizamos los filtros de blanco y negro

- ¿Qué se puede decir sobre la performance del filtro en función de la cantidad de threads utilizados?

En el caso de el filtro de blanco y negro, la diferencia entre el uso del multithreading fue notoria. Los resultados fueron:

Threads	Tiempo
1	0.024808
2	0.012858
3	0.008696
4	0.007383
5	0.010354

Como podemos apreciar en la tabla el uso de un mayor número de threads mejoró la velocidad a medida que se aumentaban los threads pero a partir del número 5 la velocidad no disminuye sino que aumenta

- ¿Qué impacto tiene considerar imágenes "grandes" en lugar de imágenes "chicas"?

En el caso del filtro black and white los resultados fueron estos:

Dimensiones Imagen	Tiempo
987x707	0.025646
800x500	0.014702
740x490	0.013271

Como podemos ver a medida que el tamaño de la imagen disminuye el tiempo que tarda en generar el filtro también disminuye

- – ¿Cuán determinante es la configuración de hardware donde se corren los experimentos y cómo puede relacionarse con lo observado?

Es determinante ya que el nivel de hardware que utilicemos va variar el tiempo de respuesta, aunque sean milésimas . Por una lado utilizando un i5 de segunda generación conseguimos los siguientes resultados:

Threads	Tiempo
1	0.024808
2	0.012858
3	0.008696
4	0.007383
5	0.010354

Por otro lado con un i3 de 4ta generación se consiguieron estos resultados:

Threads	Tiempo
1	0.022190
2	0.012168
3	0.00965
4	0.02185
5	0.02514

Como podemos ver aunque la variación sea mínima el hardware utilizado afecta

- ¿Hay diferencias de performance para los distintos tipos de filtros Multi-thread?
- Existen instrucciones de Assembler (SSE en X86) que permiten vectorizar algoritmos. De esta forma, con una sola instruccione SSE podría aplicarse una operación a varios elementos de una matriz en paralelo. Desde gcc es posible activar flags para que el código compile con optimizaciones y se fuerce el uso de este tipo de instrucciones. Indagar sobre esto y complementar los expermientos anteriores con este nuevo aspecto.
- En base a lo visto, ¿siempre es conveniente paralelizar? ¿De qué factores de la entrada depende esto?
- Considerar un análisis similar, pero para el caso del loader single-thread y Multi-thread, teniendo en cuenta también el tamaño de los lotes a experimentar.