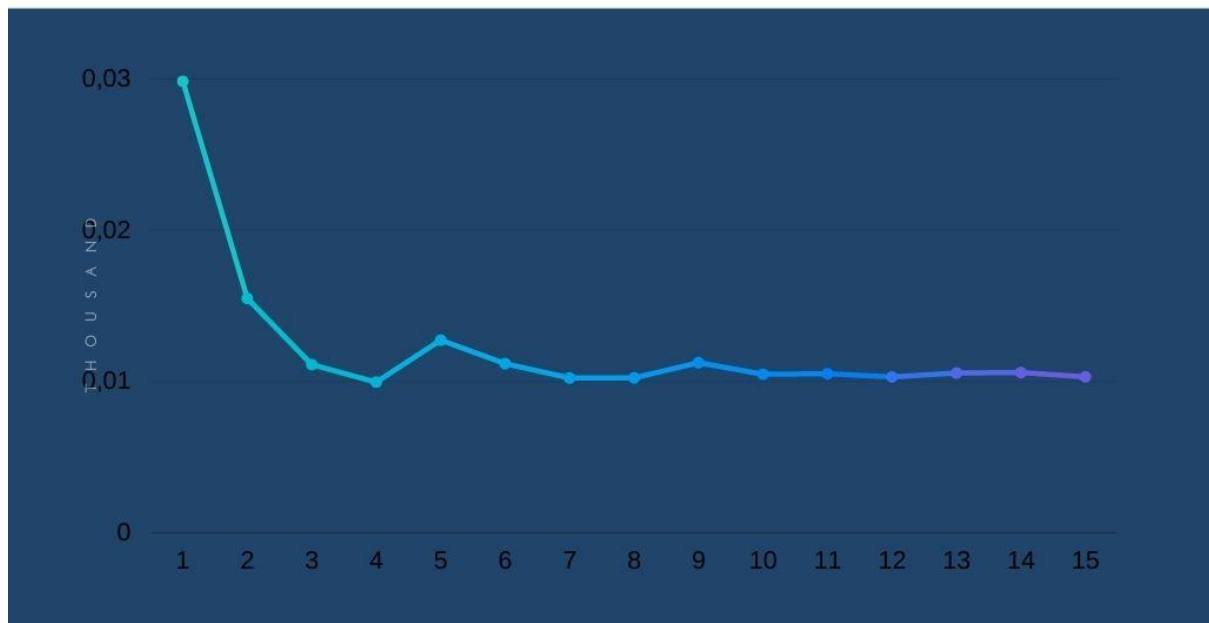


Experimentación

A la hora de realizar la experimentación utilizamos el filtro Black and White y lo probamos en una imagen de 1200 x 1200 píxeles y en otra de 5800 x 5100 píxeles.

Tanto en la imagen pequeña como en la grande los resultados obtenidos con el multi-thread son superiores al single-thread (hasta un 33% superiores), no obstante, la constante implementación de nuevos threads no implica una mejor performance en el filtro, como se puede observar en el siguiente gráfico, la cantidad de threads que dio el resultado más óptimo fue de 4.

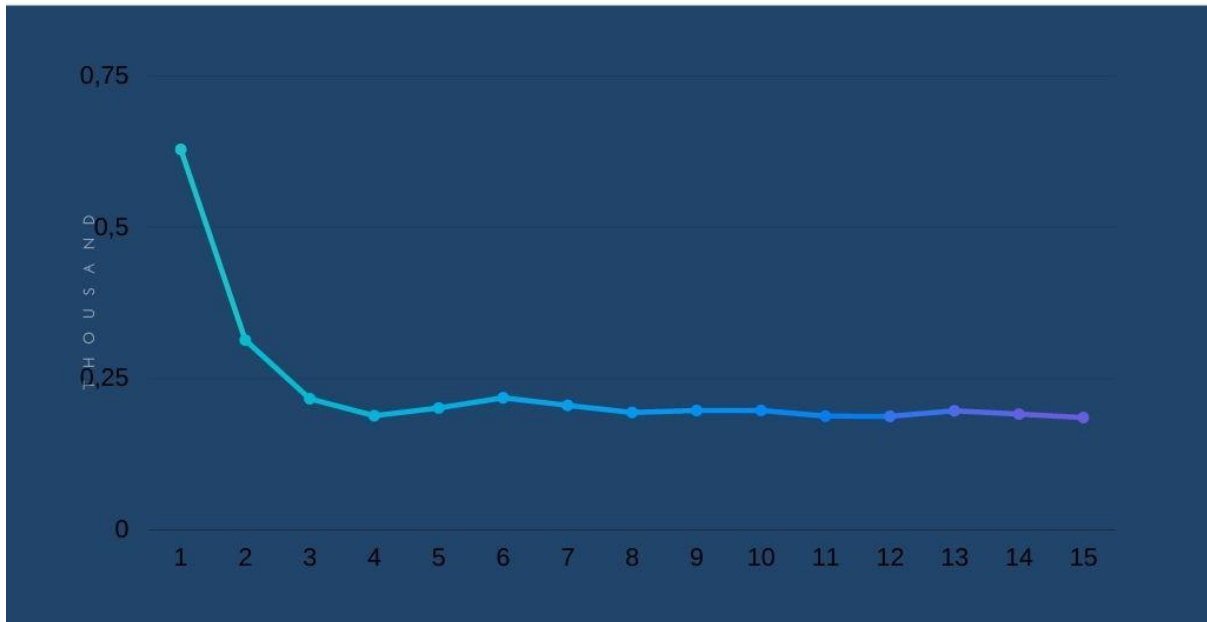
IMAGEN PEQUEÑA



Otra apreciación que podemos rescatar es que el tiempo de la tarea, comienza a estabilizarse a los 10 threads.

Respecto a la imagen grande las conclusiones son bastante similares, la mejora entre un solo thread y el máximo performance es del 29,2%, de forma similar a la imagen pequeña, con 4 threads se consiguieron resultados muy superiores a los obtenidos con menos threads, pero, a diferencia de la imagen pequeña, aquí sí se consiguieron mejores resultados con más threads, pero, no de forma inmediata.

IMAGEN GRANDE



Como se puede observar en el gráfico, una mejora casi imperceptible, pero, a partir del thread número 11 hasta el 13 los resultados son superiores, en el thread 14, vuelve a empeorar y a los 15 threads se consigue el mejor resultado, esto nos muestra que, tanto en el caso de la imagen pequeña como en el de la grande, añadir más threads puede dar mejores resultados en las primeras instancias, pero, habrá un punto de quiebre en el que los resultados irán empeorando o mejoraran de forma mucho menos drástica.

Respecto a la diferencia de de análisis entre imágenes, el resultado es obvio, a la hora de correr el filtro en una imagen 5 veces más grande que la otra, los resultados varian ampliamente siendo el tiempo más rápido de la imagen grande 6.16 veces más lento que el tiempo más lento de la imagen pequeña, lo que sí fue sorprendente fue ver que el porcentaje de mejora entre el peor y el mejor tiempo de las imágenes, era mayor en la imagen pequeña que en la grande.

La configuración del hardware juega un factor determinante puesto que permite paralelizar más procesos, puesto que uno tiene más hilos de los que hacer uso, nosotros usamos una computadora con 4 núcleos y 4 hilos y a la hora de hacer 4 tareas cada procesador hacia una, peor a la hora de añadir más, cada hilo realiza el exceso de tareas de forma intercalada, por ejemplo, de ser 5 tareas, el hilo 1 hace las tareas 1 y 5 de forma intercalada.

Performance de los filtros

A la hora de comparar los distintos filtros vemos que en la mayoría de casos, hay una mejora considerable en la mayoría de filtros, siendo la única excepción, el filtro marge, que si bien vio una mejora, no fue una tan drástica como la que sufrieron los otros filtros.

Filtro	Tiempo 5 threads	Tiempo 1 thread
Black and white	0.012519	0.0298435
Contrast	0.0167772	0.0406065
Sharpen	0.0603781	0.125203
Zoom	0.172255	0.218109
Frame	0.00271783	0.00519685
Merge	0.12068	0.131825

No siempre es conveniente paralelizar, puesto que la performance de la tarea dependerá del tamaño de la imagen, la complejidad del filtro y el tamaño de la tarea, por ende, no es rentable paralelizar cuando la tarea es muy pequeña y se aplica en una imagen muy pequeña, mientras que, si se aplica en una imagen más grande, la tarea será mayor, por lo que se maximizará la distribución de la tarea en los threads.