

Respuestas a preguntas

¿El programa podría ser más eficiente con otro lenguaje de programación que no es concurrente?

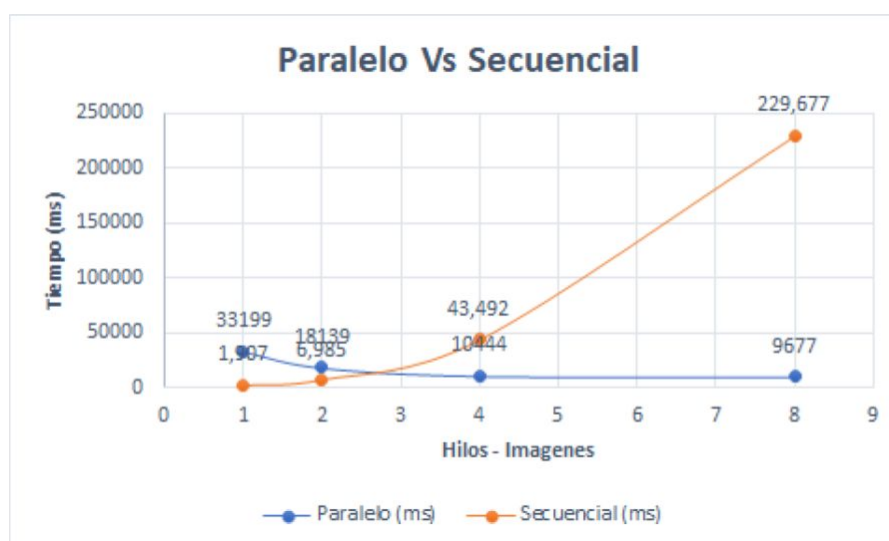
Al realizar la implementación secuencial lo que logramos percibir fue que los ciclos que calculan el color de cada uno de los píxeles se podría llegar a repartir entre los procesadores, por lo que la API de OpenMP nos serviría por completo para poder hacer más eficiente la creación del conjunto. Pero la parte gráfica se convertiría en un problema que tener que resolver, y al ser arreglos bidimensionales los que se tendrían que utilizar, realmente el tiempo de acceso y la cantidad de memoria utilizada podrían hacer más ineficiente esa versión.

¿Es posible paralelizar el algoritmo usando cláusulas de OpenMP como `single` para asignar un proceso al hilo principal o `sections` para asignar procesos?

Llegamos a pensar acerca de eso, y realmente es viable si se encuentra un método tan amigable como Java, para la creación de las imágenes, debido a que en C/C++, se deben de considerar como arreglos, o crear tu propia biblioteca. Fuera de la creación de las imágenes del conjunto, es posible el paralelizar por medio de OpenMP los ciclos `for`, y las instrucciones contenidas dentro de esto debido a que el valor que tiene cada uno de los píxeles es independiente de todos los demás.

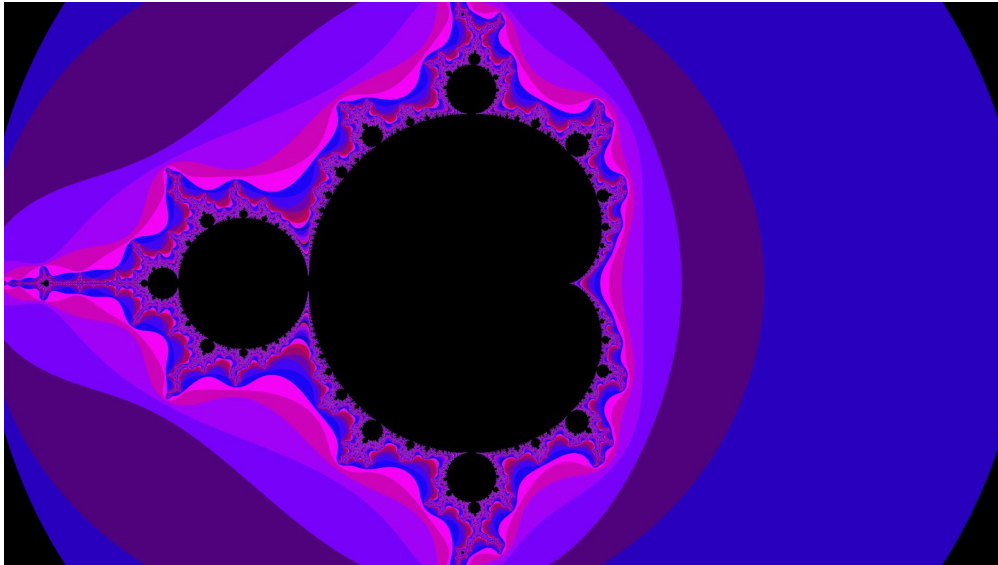
¿Cómo fue el desempeño de la versión paralela contra la secuencial, según las otras métricas de desempeño?

La versión paralela fue claramente superior en todas las ocasiones, pues se calculan de forma más rápida los píxeles que tendrá cada imagen a la vez, por consiguiente, es más eficiente.



¿Cuál es la cantidad mínima de iteraciones para conseguir un fractal que se puede identificar que pertenece al conjunto de Mandelbrot?

Esta respuesta es algo relativa, pues el número de iteraciones ayuda a definir más el conjunto, por lo que dependerá de cada quien de cuánto conoce al fractal para poder definirlo. De forma general, podríamos decir que con cerca de 100 iteraciones una persona no tan familiarizada con el fractal podría intuir que se trata del conjunto de Mandelbrot.



¿Tuvieron dificultades en aterrizar los conceptos teóricos sobre el conjunto de Mandelbrot a la hora de escribir el código?

El principal problema fue el limitar estos cálculos, pues cuando existe un pixel que no importa cuántas veces se calcule siempre se quede en nuestro radio de convergencia el programa llega a ciclarse, para resolver esto fue que se implementó el número de iteraciones, el cual también sirvió para realizar el “degradado” de colores para mostrar cuántas veces iteró el programa sobre un píxel antes de que divergiera.

¿La implementación del algoritmo tiene alguna aplicación?

La aplicación más práctica es la creación de estas imágenes, fractales, pues estas muestran formas muy peculiares cada vez que se acerca la imagen. Este hecho de demostrar que estas imágenes eventualmente son formadas (o incluso aunque no se produzcan, sabemos que existen) las ha llevado a ser objeto de estudio para conocer más acerca de los fenómenos naturales no solo en nuestro planeta, sino en el espacio.

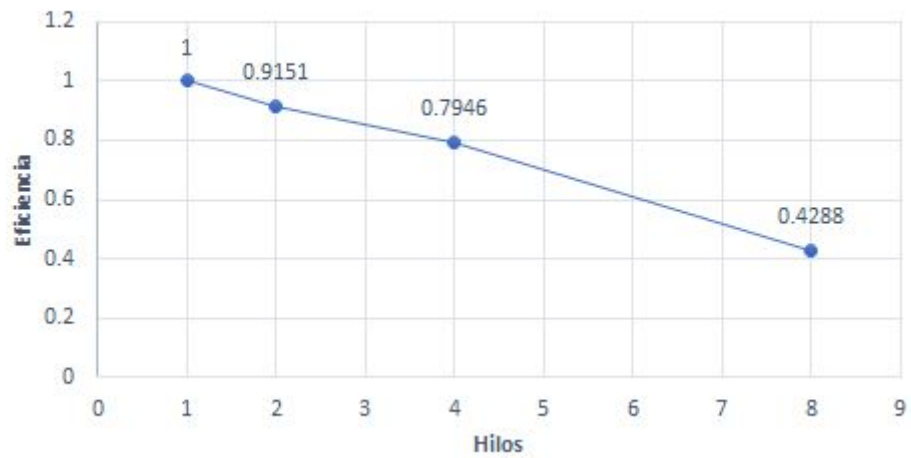
Un ejemplo de esto, es en la astrofísica en donde los fractales se utilizan para analizar la formación de las estrellas, ya que las nubes de partículas se forman siguiendo el principio de autosimilitud, con patrones irregulares pero recurrentes, de forma similar a como ocurren en el conjunto de Mandelbrot.

¿Cómo afecta el tamaño de imagen o resolución al desempeño del programa?

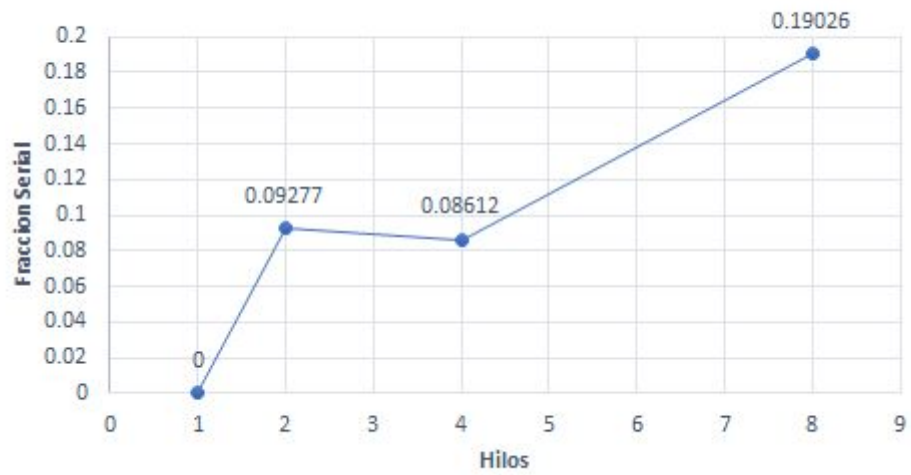
El aumentar el tamaño de la imagen implica un uso mayor de los recursos del programa, por lo tanto, al aumentar la resolución, también aumenta el tiempo de ejecución, pues se tienen que calcular los colores que tendrá cada pixel nuevo (en cuanto al aumento de tamaño).

¿Cuáles son las métricas de desempeño?

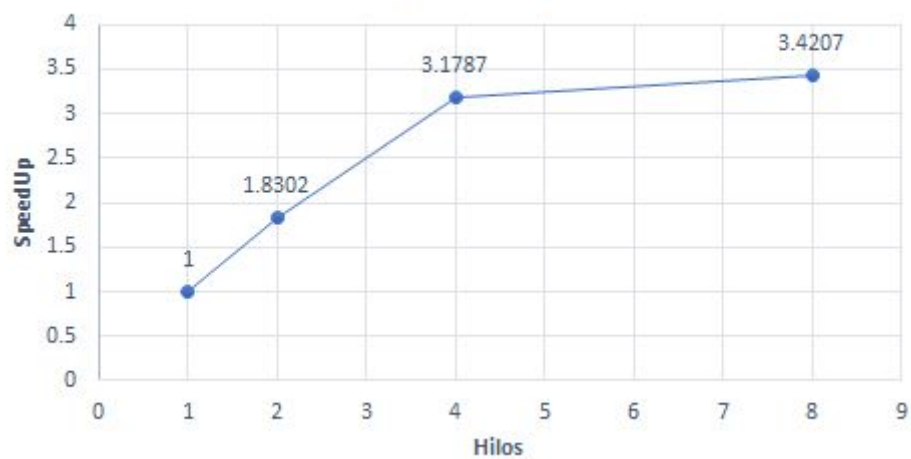
Eficiencia



Fraccion Serial



SpeedUp





¿El código se puede replicar en algún otro lenguaje o sólo se puede en Java?

Dado que el código implica realizar cálculos sencillos y la creación de imágenes mediante hilos, se puede implementar el código en cualquier otro lenguaje que permita, principalmente, la creación de imágenes a color a partir de una paleta de colores, así como el uso de hilos si se deseara realizar la versión paralela del programa.