



# Reconstrucción de imagen en función de los bordes con MPI en C++

David Alsina y Nicolás Quintero

Abril 2021

[Github](#)

## 1. Resumen

Para la realización de este ejercicio se utilizó un algoritmo de reconstrucción de imagen en blanco y negro que recibe una imagen deconstruida con su contorno, el cual fue paralelizado utilizando la librería MPI que permite la comunicación entre distintos procesos corriendo en multiples computadores, de manera que se logrará mejorar el tiempo de ejecución de este algoritmo.

Los resultados de esta ejecución han sido bastante curiosos, al correr el algoritmo multiples veces dentro de la misma region paralela (*para calcular medias y desviaciones estándar*) MPI empieza a tener problemas con la comunicación (*aparentemente*) y nos da los resultados para las imagenes incorrectos, de manera que tuvimos que modificar la ejecucion para obtener los resultados correctos que se ven a continuación.

Las graficas de tiempo de ejecución y de los speedups se realizaron unicamente sobre la reconstrucción de la imagen del monstruo del lago ness debido a falta de tiempo para iterar adecuadamente sobre la reconstrucción de las demás imagenes

## 2. Resultados

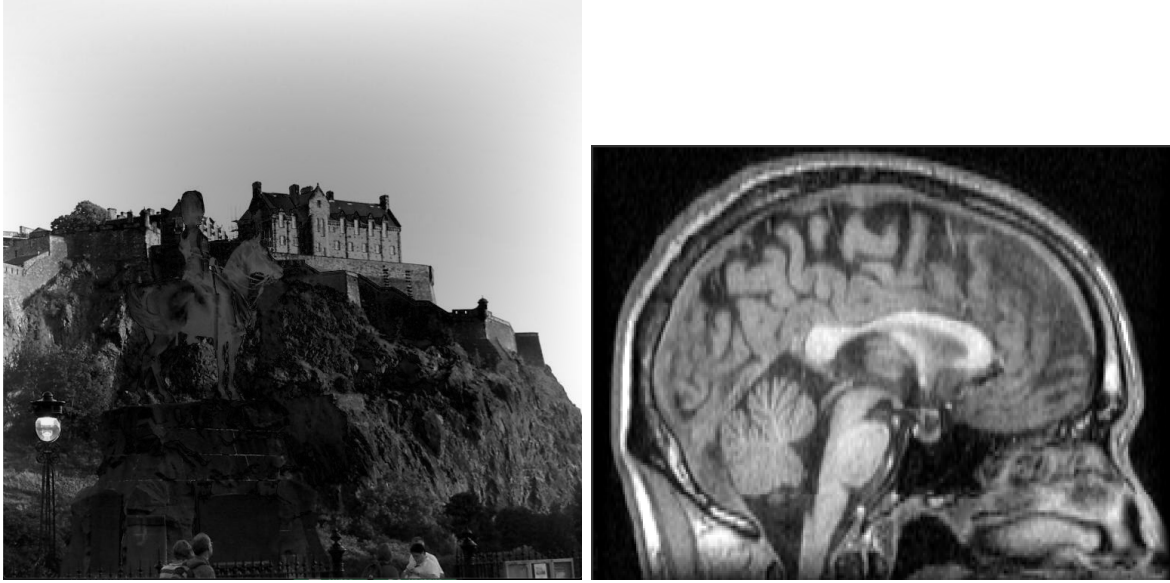


Figura 1: Reconstrucción de imagen de dimensiones 768x768 (*izquierda*) vs Reconstrucción de imagen de dimensiones 256x192 (*Derecha*)

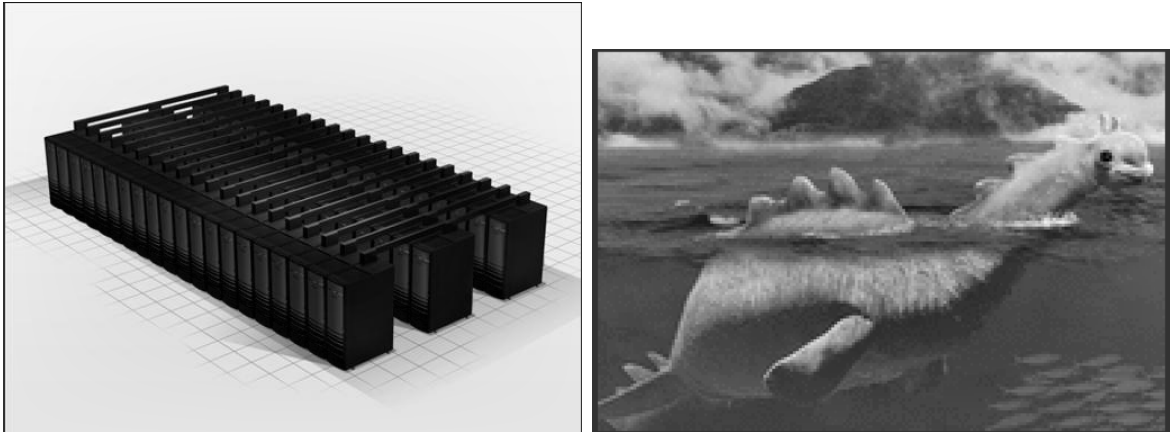


Figura 2: Reconstrucción de imagen de dimensiones 512x384 (*izquierda*) vs Reconstrucción de imagen de dimensiones 192x128 (*Derecha*)

### 2.1. Complejidades Computacionales y Speedups

*El número de repeticiones del procesamiento de la imagen usado para calcular la media y la desviación estándar fue de 10.*

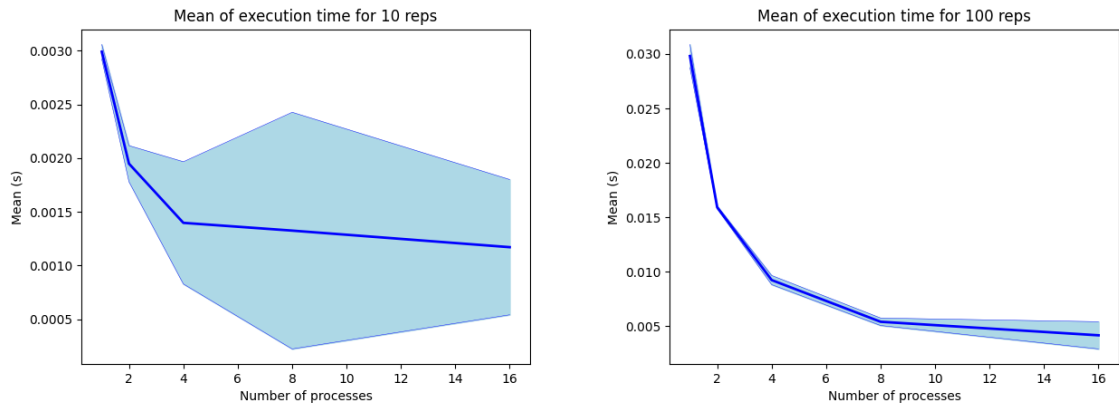


Figura 3: *Medias de tiempos de ejecución*

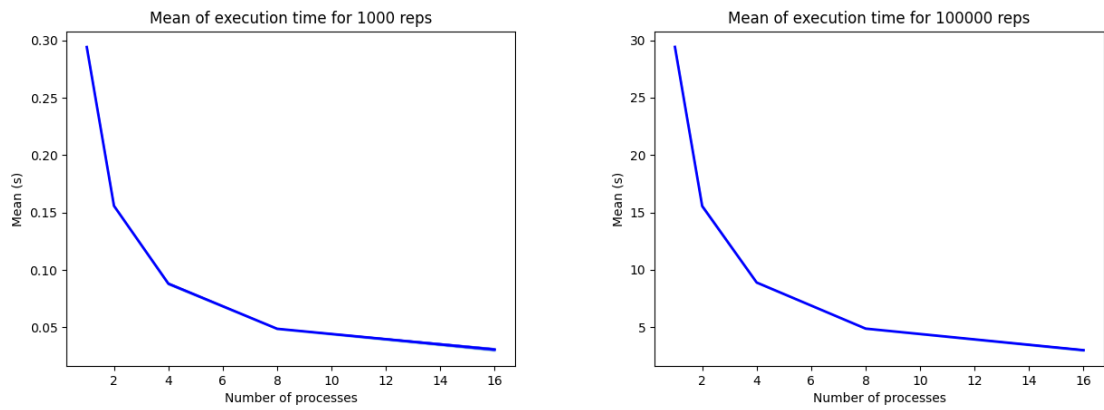


Figura 4: *Medias de tiempos de ejecución para una imagen de 24576 entradas*

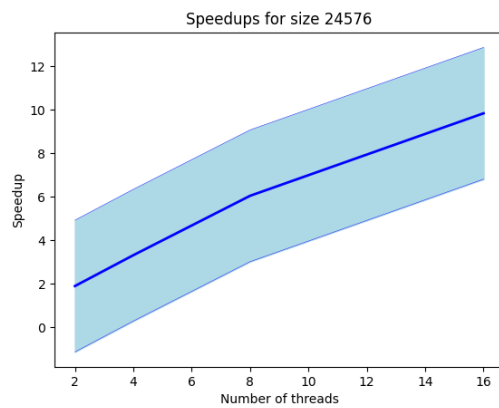


Figura 5: *Speed ups para una imagen de 24576 entradas*