

## Práctica 7. Comparativa MPI-CUDA

### 1. Introducción

En esta práctica se pide realizar una comparativa del algoritmo elegible con MPI de la práctica 3 con otro código con la misma funcionalidad utilizando CUDA. No obstante, al utilizar una GPU para la ejecución del algoritmo, es imposible tratar con la estructura de datos elegida, puesto que en la GPU no se pueden realizar multiplicaciones con struct; por ello, ha sido necesario modificar el TDA a una matriz de enteros con la salvedad de que el resultado obtenido no será el mismo que en la práctica 3, puesto que en este caso tenemos un único dato (**r**, **g** o **b**) por celda de la matriz contra los tres (**r**, **g** y **b**) que teníamos en el código MPI. Sin embargo, esto no afecta en su totalidad a la realización de la comparativa entre los dos códigos puesto que se está realizando el mismo cálculo en ambos casos.

### 2. Comparativa

Para realizar la comparativa es necesario volver a tomar tiempos de la práctica anterior, puesto que en el nuevo código se realiza una tercera parte del número de operaciones que se realizaban en la primera al modificar la estructura de datos. Por ello, se han escogido para el código CUDA 300x300, 702x702 y 1020x1020 y para MPI las dimensiones 100x100, 234x234 y 340x340, que resultan ser la tercera parte de las anteriores, de manera que coincida el número de operaciones realizadas.

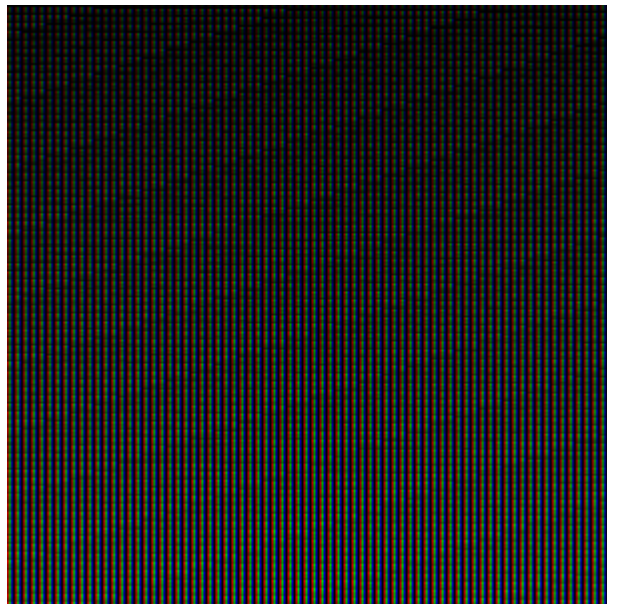
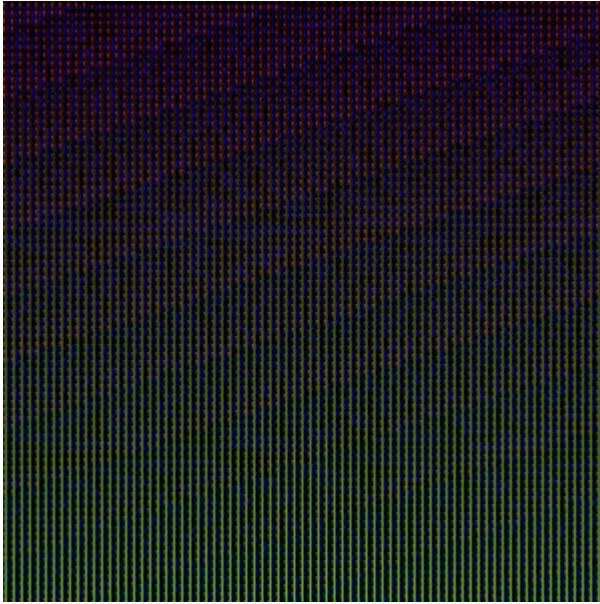
Los tiempos tomados en la versión secuencial, MPI y CUDA son los siguientes:

Tecnología	n = 100	n = 234	n = 340
MPI	5,43771E-05	4,39122E-04	6,31182E-04
CPU	3,68941E-05	1,17851E-04	2,52933E-04

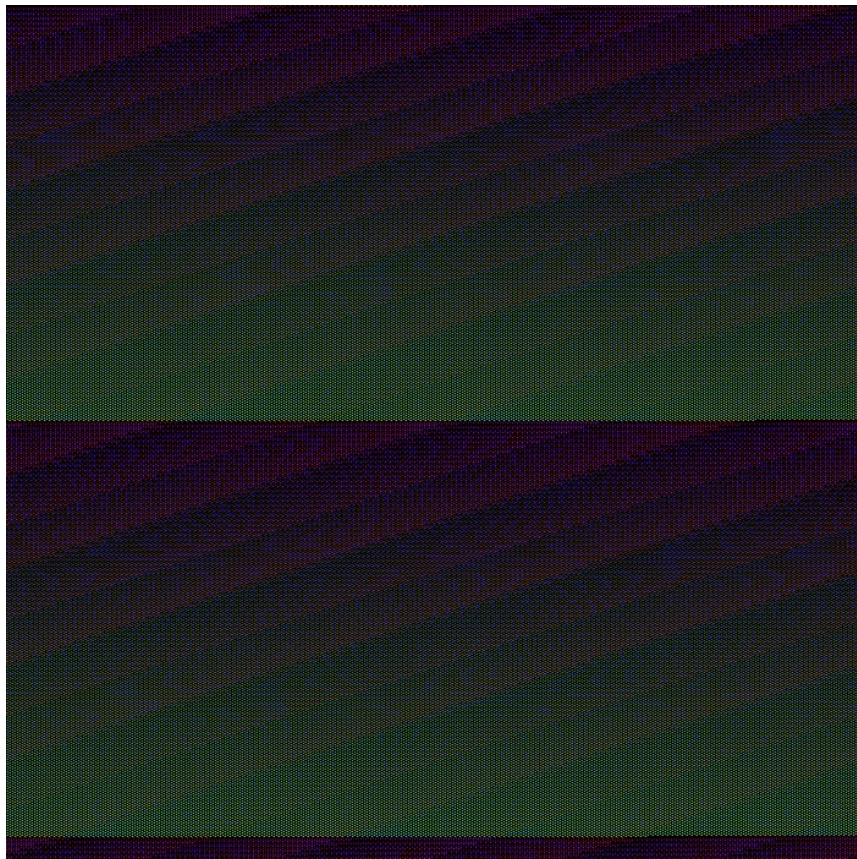
Tecnología	n = 300	n = 702	n = 1020
GPU	2,87920E-05	3,36750E-05	4,70180E-05

### 3. Resultados obtenidos en formato PPM

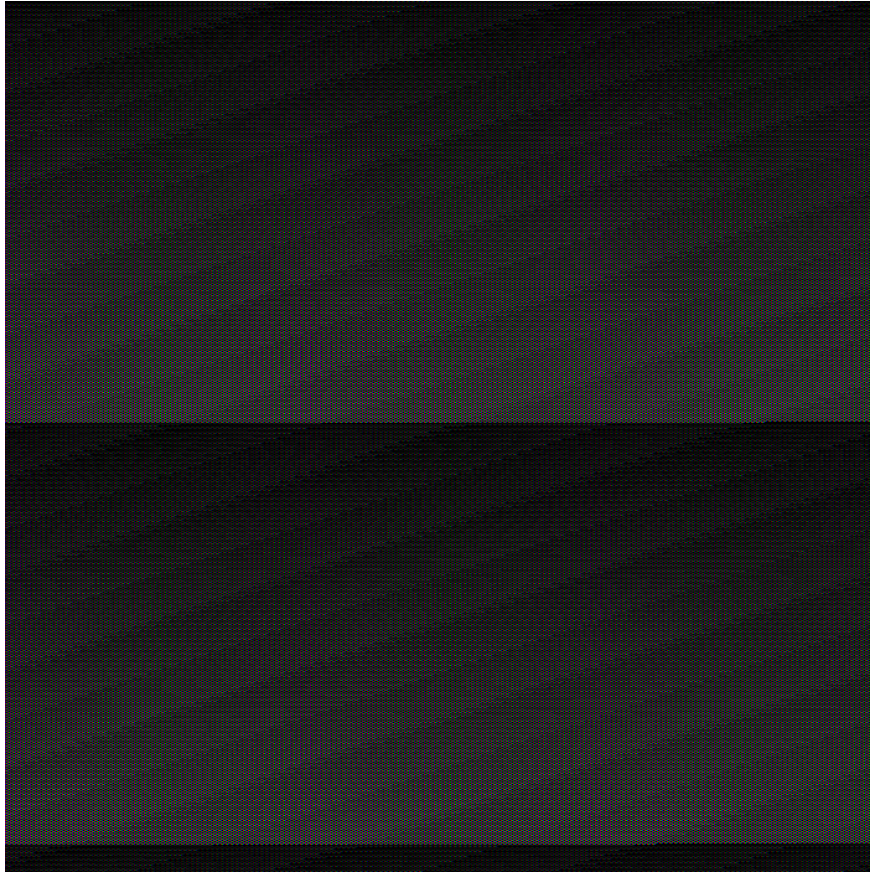
Para la carga de 300x300 obtenemos:



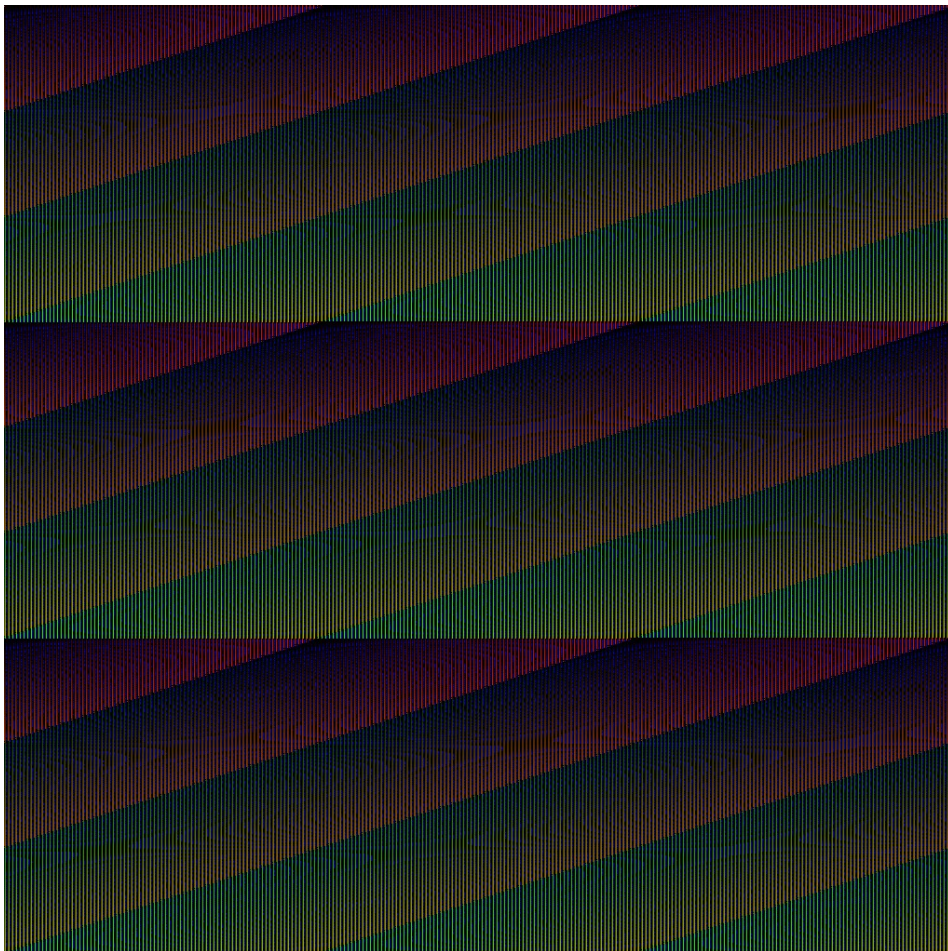
Para la carga de 702x702:



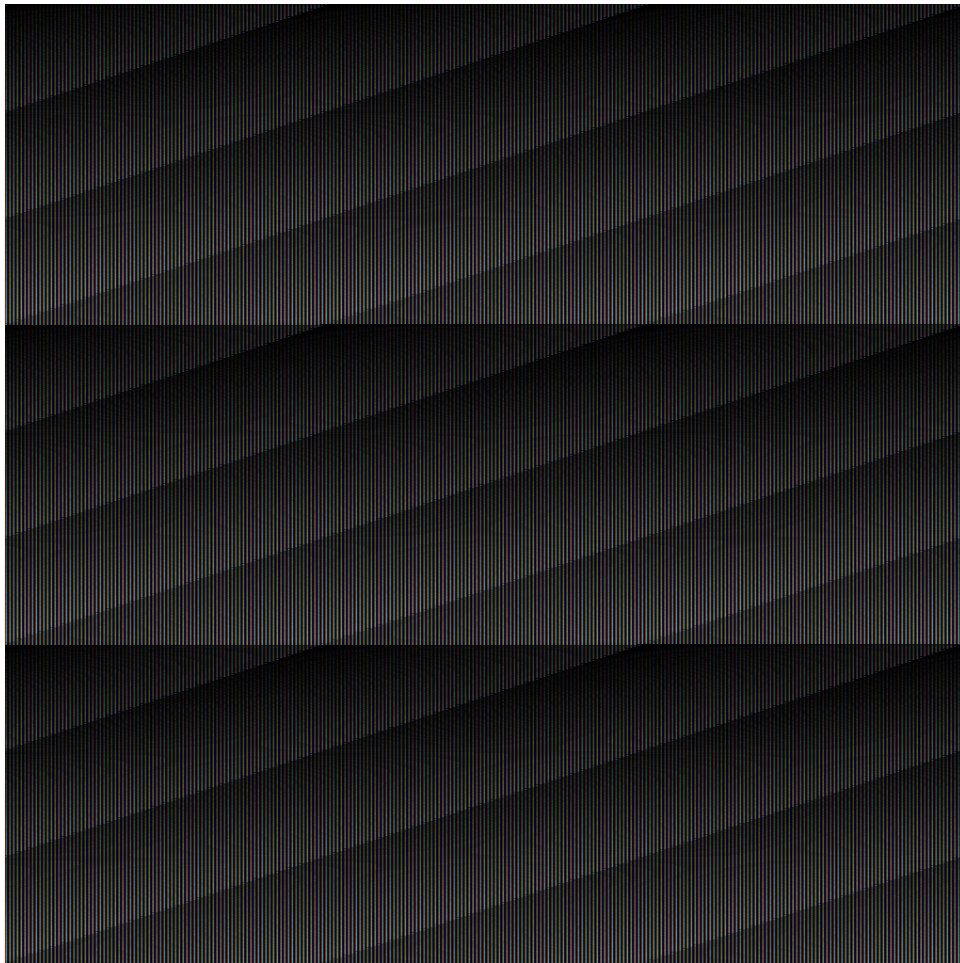




Y por último, para la carga de 1020x1020:







#### 4. Conclusión

Comenzamos aclarando que, como ya se dijo en la práctica 3, el tiempo de CPU es menor que el empleado en el código MPI puesto que se utiliza el mismo código para ambos, y el envío y recepción de datos para la ejecución secuencial resulta ser mucho menor que para MPI.

Vemos claramente que el código ejecutado en la GPU mejora el rendimiento de la CPU (secuencial y MPI), esto es debido a que la GPU está diseñada para manejar grandes cantidades de datos y realizar con ellos operaciones muy específicas de forma paralela gracias a los muchos núcleos pequeños de los que consta. Sin embargo la CPU no es capaz de paralelizar muchas tareas (más genéricas), relativamente, ya que consta de muy pocos núcleos que en realidad están optimizados para la ejecución secuencial.

Por lo tanto si queremos ejecutar un algoritmo que realice siempre las mismas operaciones con un elevado grado de paralelismo siempre será más eficiente hacerlo en una GPU, ya que la CPU puede tardar unas 10 veces más.