UA

Universidad
de Alicante

ESCUELA POLITÉCNICA SUPERIOR

COMPARACIÓN ENTRE CPU Y GPU

Curso académico 2024 / 2025
Semestre 2
Grado en Ingeniería en Inteligencia Artificial
Grupo 1

Profesora: Ricardo Moreno Rodríguez
Alumno: Santiago Álvarez Geanta

Universitat d'Alacant
Universidad de Alicante

Febrero 2025

ÍNDICE

1.Resultados experimentales

1.1 Introducción a la Comparación

Las CPU (Unidad Central de Procesamiento) y GPU (Unidad de Procesamiento Gráfico) tienen arquitecturas distintas que influyen en su desempeño en distintas tareas computacionales. Mientras que las CPU están optimizadas para la ejecución de tareas secuenciales, las GPU están diseñadas para el procesamiento paralelo masivo.

1.2 Tabla de Resultados

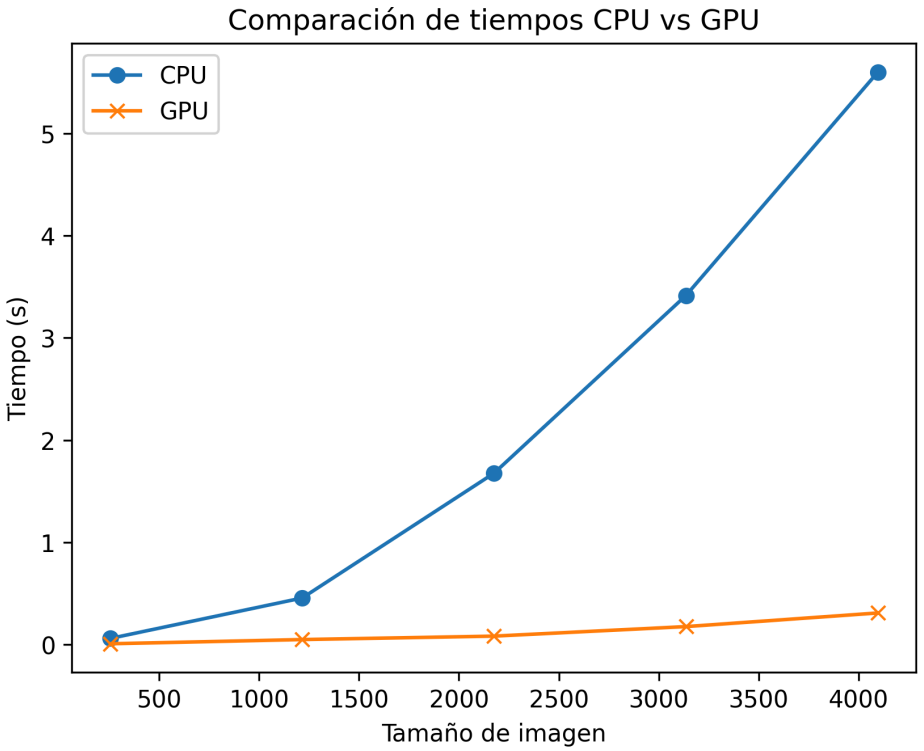
He realizado pruebas comparativas utilizando diferentes tamaños de imagen en una CPU y una GPU. Los tiempos de ejecución obtenidos para cada tamaño de imagen se presentan en la siguiente tabla:

Tamaño de Imagen	Tiempo en CPU (ms)	Tiempo en GPU (ms)
256 x 256	59.3	7.29
1216 x 1216	455	48.5
2176 x 2176	1677	82.1
3136 x 3136	3414	175
4096 x 4096	5599	309

Los resultados muestran una reducción significativa en los tiempos de ejecución al utilizar GPU, especialmente en imágenes de mayor tamaño. Además, podemos ver cómo el tiempo de ejecución en CPU crece de manera exponencial conforme aumenta el tamaño de la imagen, mientras que en la GPU este crecimiento se mantiene más cercano a un comportamiento lineal. Esto se debe a la arquitectura altamente paralelizada de las GPU, que permite dividir la carga de trabajo entre múltiples núcleos de procesamiento.

En la representación gráfica realizada con Matplotlib, se observa claramente cómo la curva de los tiempos de CPU aumenta de forma pronunciada a medida que el tamaño de la imagen se incrementa, mientras que la curva de la GPU presenta un crecimiento mucho más controlado. Este comportamiento confirma la eficiencia del paralelismo masivo de la GPU en la ejecución de tareas que requieren procesamiento intensivo.

Otro punto importante a destacar es que, en las imágenes de menor tamaño, la diferencia de tiempos entre CPU y GPU no es tan marcada, sin embargo, a partir de un cierto umbral (en nuestro caso, imágenes superiores a 1000x1000 píxeles), la ventaja de la GPU se vuelve evidente y la diferencia de tiempo de ejecución crece de manera considerable.



2. Análisis de los resultados

2.1 Diferencias entre CPU y GPU

- **Arquitectura**: Las CPU tienen pocos núcleos altamente optimizados para la ejecución de hilos secuenciales, mientras que las GPU poseen miles de núcleos más simples diseñados para el procesamiento masivo paralelo.
- **Paralelismo**: Las GPU pueden manejar múltiples operaciones simultáneamente, lo que resulta en una aceleración significativa para tareas computacionales intensivas (como puede ser el tratamiento de imágenes complejas como en nuestro caso).

2.2 Impacto del Paralelismo en el Rendimiento

El paralelismo ofrecido por las GPU permite una ejecución más rápida de algoritmos intensivos en cómputo, reduciendo la latencia y aumentando la eficiencia. En comparación, las CPU son más versátiles para tareas generales pero menos eficientes en cálculos masivos.

2.3 Justificación Técnica

- **Uso de Memoria**: Las GPU optimizan el acceso a memoria mediante la paralelización, mientras que las CPU dependen de estrategias de caché avanzadas para mejorar el rendimiento.
- **Tiempos de Ejecución**: En la tabla de resultados se observa que el aumento en el tamaño de la imagen afecta significativamente el rendimiento de la CPU, mientras que la GPU mantiene tiempos más bajos debido a su capacidad de procesamiento en paralelo.

3. Propuestas de optimización

3.1 Estrategias para Mejorar el Rendimiento

- **Modelos Ligeros**: Implementar algoritmos optimizados para reducir la carga computacional en CPU y GPU tal y como se nos menciona en la práctica.
- **Paralelismo Adicional**: Ajustar los algoritmos para explotar mejor el paralelismo de la GPU.
- **Gestión Eficiente de la GPU**: Mejorar la asignación de memoria y la distribución de cargas de trabajo.

3.2 Implementación de Técnicas Avanzadas

- **Uso de CUDA**: Para maximizar el desempeño en GPU.
- **Optimización de Kernels**: Ajustar los kernels de cómputo para aprovechar al máximo los recursos disponibles

4. Reflexión y práctica

4.1 Aprendizaje Adquirido

El análisis comparativo que hemos realizado demuestra la importancia del hardware en la ejecución eficiente de tareas intensivas. Hemos podido ver cómo el paralelismo impacta en el rendimiento y cómo la optimización de código puede mejorar la eficiencia computacional.

4.2 Aplicación en Escenarios Reales

- **Inteligencia Artificial**: Los modelos de redes neuronales dependen fuertemente de las GPU para el entrenamiento eficiente.
- **Simulaciones Científicas**: La computación paralela permite realizar simulaciones complejas en menos tiempo.
- **Procesamiento de Imágenes y Video**: Aplicaciones como reconocimiento facial y renderizado 3D se benefician de la aceleración por GPU, tal y como se explicó en clase, las GPU tienen memoria caché dedicada a las texturas.