

Informe Lab 6: GPUs con Python

Aspectos Positivos

Este laboratorio me ha parecido especialmente interesante porque permite introducir de forma práctica el uso de GPUs para acelerar cálculos científicos en Python, comparando distintas herramientas como NumPy, CuPy, Numba CUDA y PyTorch. A lo largo de los ejercicios se observa de manera muy clara cómo el rendimiento no depende únicamente del hardware disponible, sino también del modelo de programación empleado y de si se minimizan correctamente las transferencias de datos entre CPU y GPU.

Además, he realizado una actividad extra utilizando PyTorch, adaptando los objetivos del laboratorio a un contexto más biológico. En esta actividad se ha simulado el cálculo de distancias entre perfiles celulares de alta dimensión y un perfil de referencia, una operación habitual en análisis de poblaciones celulares, clasificación de células o estudios de similitud fenotípica. Esta parte me ha resultado especialmente interesante, ya que demuestra cómo herramientas desarrolladas para aprendizaje automático pueden emplearse también para acelerar cálculos numéricos generales en GPU dentro del ámbito de la Biología y la Bioinformática.

Aspectos a Mejorar

Aunque el laboratorio está bien planteado, en algunos apartados resulta complicado interpretar los resultados. En particular, averiguar qué parte del proceso corresponde al cálculo real y cuál al overhead de lanzamiento y sincronización.

Dificultad y Tiempo

La dificultad del laboratorio es intermedia. Aunque muchas implementaciones pueden parecer sencillas a primera vista, entender por qué ciertos enfoques son más rápidos que otros y justificar correctamente los resultados requiere esfuerzo.

El tiempo invertido en el laboratorio ha sido significativo. Más allá de la escritura del código, gran parte del trabajo ha consistido en depurar errores, ejecutar jobs en la cola `bohr-gpu`, analizar los resultados obtenidos para distintos tamaños del problema y relacionarlos con los conceptos teóricos vistos en clase.