

Informe Lab 5: Explotando el paralelismo con Python en CPUs

Aspectos Positivos

Este laboratorio me ha parecido especialmente interesante porque permite profundizar de forma práctica en distintas estrategias de paralelización en Python, comparando enfoques muy diferentes como multiprocessing y Numba, tanto en modo secuencial como paralelo. Los ejercicios están bien planteados y permiten observar de manera muy clara cómo el rendimiento depende no solo del número de núcleos disponibles, sino también del tipo de paralelismo utilizado (procesos frente a hilos) y del tamaño del problema.

Además, en este laboratorio he tenido más tiempo para desarrollar una actividad extra distinta a la propuesta originalmente por el profesor, optando por diseñar un ejercicio relacionado con el análisis de datos biológicos, concretamente un análisis simulado de expresión génica. Esto me ha parecido especialmente adecuado, ya que conecta directamente los conceptos de paralelización con problemas reales del ámbito de la Biología y la Bioinformática, haciendo el laboratorio más aplicable a mi formación.

Aspectos a Mejorar

Aunque el laboratorio está bien estructurado, en algunos ejercicios resulta complicado determinar desde el principio qué enfoque de paralelización es el más adecuado y qué resultados cabe esperar. En particular, no siempre es evidente cuánto tiempo debería tardar en completarse un *job*, lo que obliga a realizar varias ejecuciones de prueba y a experimentar con distintos parámetros. Por ello, considero que sería útil incluir alguna orientación aproximada sobre los tiempos de ejecución esperables para los distintos valores de N, de modo que se pueda trabajar con mayor seguridad y previsión.

Dificultad y Tiempo

La dificultad del laboratorio es alta, especialmente a nivel conceptual. Más allá de escribir el código, lo que más tiempo me ha llevado ha sido entender el comportamiento de cada método y justificar adecuadamente los resultados obtenidos. La parte técnica relacionada con la ejecución en SLURM, el uso de scripts sbatch y la gestión del número de núcleos me ha resultado relativamente sencilla en comparación con el diseño y la optimización del código Python.