

INFORME LABORATORIO 5

PARALELISMO (NÚCLEOS Y GPUS) CON PYTHON EN GACOP

Esta práctica tuvo muchos puntos interesantes, aunque también algunos problemas importantes que me hizo invertir más tiempo del previsto.

El primer reto en el apartado 3.2, fue entender cómo dividir el array de manera óptima usando código Python, me resultó difícil de realizar y ver cómo usar la función map adecuadamente y poder aplicar el multiprocessing a la función reduce-operation. Una vez realizado este apartado, los 3 siguientes fueron mucho más sencillos en cuanto a desarrollo de código. A pesar de esto, hubo aspectos muy positivos. Usar multiprocessing me permitió experimentar con la paralelización en CPUs, y fue gratificante ver cómo los tiempos de ejecución disminuían al aumentar el número de núcleos utilizados.

En el siguiente apartado, fue interesante trabajar con CuPy en la GPU, ya que realmente pude apreciar el poder de la paralelización masiva en cálculos simples como la suma de arrays grandes. En este sentido, quedó muy claro que, aunque las GPUs requieren una inversión inicial de tiempo para configurar y transferir datos, si los datos son lo suficientemente grandes, merece la pena usar GPU.

Una de las mayores dificultades de la práctica fue intentar realizar el apartado 3.3 b), el cual considero que superaba mis capacidades de entendimiento y me retrasó mucho en la realización de la tarea.

Otro de los problemas que tuve fue al intentar usar la cola Bohr-gpu, esta no presentaba la librería cuda cargada y tuve que cargar el módulo cuda además de anaconda, pienso que podría ser un punto a añadir en la práctica para poder realizarla correctamente.

En general, fue una práctica enriquecedora que me permitió ver las diferencias de rendimiento entre CPUs y GPUs, y entender mejor cómo aprovechar estas tecnologías en Python. Sin embargo, algunos problemas técnicos y detalles operativos limitaron un poco el alcance de lo que pude explorar.