SEGUNDO TALLER

Presenta: Manuel José Parra Malagón y David Santiago Beltrán Benavides
Instructor: Oscar Alexander Mendez Aguirre
Fecha: 6 de Abril de 2024

I. Introducción

Este taller se enfoca en profundizar el entendimiento y la aplicación de estructuras de datos complejas, como **matrices y tensores**, se utilizó Python en Kaggle, también, se experimentó con diferentes entornos de ejecución ***(CPU, GPU y TPU)*** para analizar y comparar el rendimiento. [1] [2][3]

II. Resultados de la práctica

El taller se desarrolló en dos secciones, es decir por cada sección se evalua un producto correspondiente a una estructura de datos. Cabe destacar que el tamaño de las matrices y tensores es el mismo podemos resumirlo a $\boxed{M_{15000\times15000}}$

Puede consultar los resultados de las mediciones en este enlace ***[consultar mediciones por algoritmo]***

A partir de los resultados a través de los diferentes tipos de acerlaradores se obtiene la siguiente tabla:

Hardware	**Algoritmo**	
	1	2
CPU	(104.058 & 109.326 & 19980)	(0.785 & 2.353 & 3.18)
GPU	(0.001 & 1.082 & 1.11)	(0.001 & 42.382 & 42.4)
TPU	(0.120 & 1.566 & 1.7)	(0.000 & 0.453 & 0.465)

Table I
NÓTESE QUE EXISTE TRES VARIABLES α, β, λ EXPRESADAS EN SEGUNDOS PARA CADA MEDICIÓN

Donde cada letra del alfabeto griego corresponde a una herramienta de medición para el mismo algoritmo con el mismo tipo de acelerador donde:

- α: Corresponde a la herramienta de **cProfile**
- β: Corresponde a la herramienta de **PyInstrument**
- λ: Corresponde a la herramienta de **Magic Line Jupyter**

Aclarados los conceptos, procedemos a narrar lo que sucedió en cada parte del experimento:

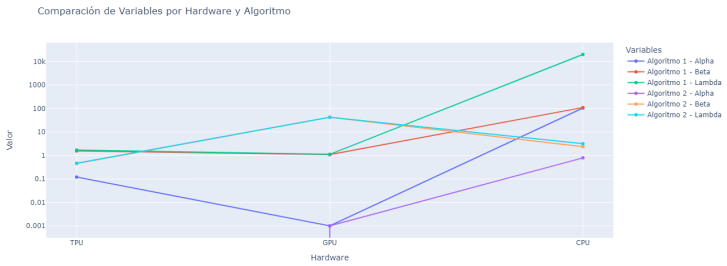
A. Producto de *MATRICES (Algoritmo 1)*

- Para la CPU se utiliza numpy, se obtiene los resultados de α = 104.058 segundos & β = 109.326 segundos & λ = 19980 segundos
- Para la GPU se utiliza cupy, se obtiene los resultados de α = 0.001 segundos & β = 1.082 segundos & λ = 1.11 segundos
- Para la TPU se utiliza tensorflow, se obtiene los resultados de α = 0.120 segundos & β = 1.566 segundos & λ = 1.7 segundos

B. Producto de *TENSORES (Algoritmo 2)*

- Para la CPU se utiliza numpy, se obtiene los resultados de α = 0.785 segundos & β = 2.353 segundos & λ = 3.18 segundos
- Para la GPU se utiliza cupy, se obtiene los resultados de α = 0.001 segundos & β = 42.382 segundos & λ = 42.4 segundos
- Para la TPU se utiliza tensorflow, se obtiene los resultados de α = 0.000 segundos & β = 0.453 segundos & λ = 0.465 segundos

En el siguiente gráfico se representa el rendimiento de los algoritmos usando los tipos de aceleradores:



III. Conclusiones

- Para el algoritmo 1, el acelerador más óptimo es el de la GPU, esto se debe a que el algoritmo se planteó con la estructura de matrices desde cupy, el acelador de TPU no es la pieza de hardware más eficiente para esta estrutura, la CPU es lo que no se debe uasr para procesar productos de matrices grandes.
- Para el algoritmo 2, el acelerador más óptimo es de la TPU, esto se debe a que el algoritmo se planteó con la estructura de tensores desde tensorflow, luego, se puede observar que la CPU es una pieza de hardware efectiva pero no la mejor para procesar tensores, finalmente la GPU presenta menor rendimiento con respecto al cálculo de producto de tensores.

IV. Referencias

[1] CuPy Development Team. *API Reference — CuPy 13.0.0*.
[2] NumPy Development Team. *NumPy user guide*.
[3] TensorFlow Development Team. *TensorFlow Core v2.4.1*.