***Sebastian Duque Restrepo***

***1112783873***

***UTP***

**Tiempos de Ejecución (Matriz 1000X1000)**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **BlockSize** | **CPU** | **GPU** |
| **Prueba 1** | 4 | 0.003496 | 0.002015 |
| **Prueba 2** | 4 | 0.003426 | 0.001607 |
| **Prueba 1** | 8 | 0.003420 | 0.001390 |
| **Prueba 2** | 8 | 0.003426 | 0.001434 |
| **Prueba 1** | 16 | 0.003453 | 0.001412 |
| **Prueba 2** | 16 | 0.003459 | 0.001464 |
| **Prueba 1** | 32 | 0.003415 | 0.001396 |
| **Prueba 2** | 32 | 0.003470 | 0.001385 |

**Conclusiones**

* Se pudo apreciar que a medida que variábamos el BlockSize el tiempo de la **GPU** no variaba mucho.
* Esta prueba se hizo con una matriz de 1000X1000 y observamos que hubo una diferencia significativa entre el tiempo de la **CPU** y el tiempo de la **GPU** dejando ver que fue más rápido en la **GPU**, pero si las pruebas se hicieran para matrices más pequeñas como por ejemplo 10X10, entre otras veremos que la **CPU** será más rápida que la **GPU,** dejando ver que la **GPU** será mucho más eficiente que la **CPU** para matrices de tamaños grandes.

Por ejemplo esta es la tabla de ejecución para una matriz de 10X10:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **BlockSize** | **CPU** | **GPU** |
| **Prueba 1** | 4 | 0.000001 | 0.000037 |
| **Prueba 2** | 4 | 0.0000009 | 0.000037 |
| **Prueba 1** | 8 | 0.000001 | 0.000039 |
| **Prueba 2** | 8 | 0.000001 | 0.000042 |
| **Prueba 1** | 16 | 0.000001 | 0.000038 |
| **Prueba 2** | 16 | 0.000001 | 0.000036 |
| **Prueba 1** | 32 | 0.000001 | 0.000037 |
| **Prueba 2** | 32 | 0.000001 | 0.000038 |