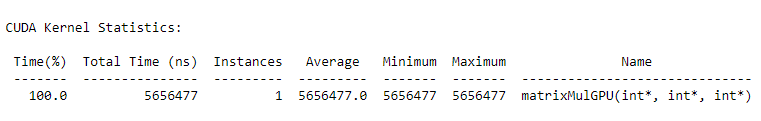
**MEMORIA DE EJERCICICIOS OPCIONALES DEL CURSO DE CUDA**

**PRODUCTO DE MATRICES**

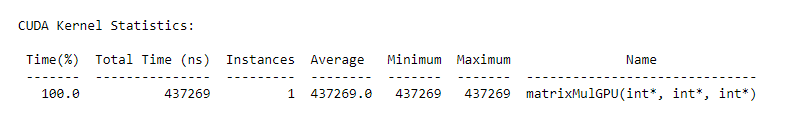
Para calcular el producto de dos matrices cuadradas de tamaño N, se necesita realizar un programa que mediante dos bucles de N iteraciones, se calcule cada elemento de la matriz resultado.

Para analizar dicha mejora, vamos a utilizar el profiler de CUDA para ver el tiempo de ejecución que tendríamos si lo ejecutásemos por defecto y lo compararemos con el tiempo de ejecución del mismo código, pero aprovechando el paralelismo que nos ofrece la GPU.

En primer lugar, si ejecutamos el código sin paralelizar obtenemos el siguiente resultado:

Por otro lado, para paralelizar el código, vamos a aprovechar cada hilo para calcular una celda de la matriz resultante, por lo que hay que calcular a qué celda corresponde cada identificador del hilo. El número de fila y de columna se calcula de la siguiente manera:

El número de bloques que utilizamos sería el número total de hilos que necesitamos, es decir, N x N, divido entre el número de hilos que vamos a utilizar por bloque, es decir, 256 hilos en este caso. Así, obtendríamos el siguiente resultado:

Podemos ver que pasamos de 5,656 ms a 0,437 ms, es decir, se obtiene un factor de mejora de 12.94, lo que viene a ser un 1294 % menos.

**OPTIMIZACIÓN Y ACELEACIÓN DE SAXPY**

En este ejercicio, se nos da un código que contiene algunos errores y que está sin optimizar, aunque sí que está paralelizado, por lo que se propone arreglar dichos fallos y optimizar el código de tal forma que se ejecute el kernel en menos de 200 us.

En primer lugar, hay que solucionar dos fallos que tiene el código, los cuales son:

* Error al calcular el número de hilo: En el código propuesto se calcula el índice multiplicando todas las componentes x de blockIdx, blockDim y threadIdx mientras que lo correcto es multiplicar las componentes x de blockIdx y blockDim y sumar la componente x de threadIdx.
* Falta cudaDeviceSynchronize(), debido a que si la cpu no espera a la finalización de la ejecución del kernel entonces muestra los resultados incorrectos.
* Los datos de las matrices son declarados como float en vez de int, por lo que hay que cambiarlo también.

Imagen que contiene Texto

Descripción generada automáticamenteUna vez solucionado esto, los resultados que obtendríamos con el profiler son los siguientes: