



Ciencia de la Computación

Computación Paralela y Distribuida

Multiplicación de Matrices Kernel

Alvarez Cardenas Andre Carlos

Semestre 8
2022-2

“Los alumnos declaran haber realizado el presente trabajo de acuerdo a las normas de la Universidad Católica San Pablo”

Análisis y Comparación:

Se pueden ver varias cosas que cambian entre el primer código y el segundo. La primera y más importante es que el primero código usan un solo thread para calcular y en el segundo se utiliza la memoria compartida. Esto se puede ver con el cambio de Mds y Nds en el segundo código, que remplazan lo que era el M y N. Ya que estas vienen a ser variables de memoria compartida, que cada bloque se creara un par de Mds y Nds. Esto es importante porque los bloques necesitan utilizar los valores que estaban en M y N.

Los valores de "threadIdx" y "blockIdx" son guardadas en variables globales para que se puedan utilizar al momento de determinar las filas y columnas que serán trabajadas por threads. Esto se da gracias a las variables de memoria compartida "TILE_WIDTH". De ahí se itera por todas las fases que se realizaron para obtener el producto punto. Y de ahí cada thread cargado con la operación vista en la línea 9-10 y se alinean los índices de fila y columna. Terminado esto se pasa a lo que es la sincronización de los threads, que se asegura que todos los thread terminaron de cargar los valores de M y N a Mds y Nds. Una vez estén sincronizados, se aplica un bucle que hace el producto punto. terminado esto, se vuelve a aplicar el syncthreads para que todos hayan terminado y los valores de Mds y Nds estén actualizados.

El segundo cambio fuerte que se nota entre el código 1 y el código 2 es que en el segundo, entre la línea 8-14 se puede ver la técnica Strip-mining, el cual toma un bucle largo y la rompe en fases. Cada una de estas fases consiste en un bucle interno que ejecuta una cantidad de iteraciones consecutivas.

Por último, el tercer cambio que se puede ver y el más importante. Es que este algoritmo reduce los accesos a memoria global gracias el TILE_WIDTH.

Link a Github: [Link](#)