Multiplicación de matrices con tiles

Reporte técnico

Iván Aram González Su

A01022584

Octubre 2018

Resumen

La multiplicación de matrices ha sido un problema trivial en programación durante mucho tiempo pero fue hasta que la programación en GPU fue viable que dicho problema se volvió a cuestionar, se volió a cuestionar para poder entender y explorar las posibilidades de la programación multithreading. No solo eso, sino que teniendo la programación en GPU nacieron nuevos paradigmas como el de tiles que mejora la eficiencia del programa, este paradigma se basa en usar memoria compartida para disminuir el número de operaciones realizadas en memoria global ya que el acceso a la memoria compartida es más veloz. Dicho paradigma se presenta en este proyecto.

1. Especificaciones del proyecto

El programa tiene 2 ejecutables, el primero es para ver al comparación entre la multiplicación de matrices en CPU, la multiplicación de matrices en GPU sin tiles y multiplicación de matrices con tiles. El segundo ejecutable es solo la multiplicación de matrices con tiles para poder observar mejor el tiempo que toma dicha multiplicación. El programa en GPU sin tiles lo hice con un grid de cierto tamaño para que el número de bloques fuera el óptimo y bloques de 512x1 hilos. El programa en GPU con tiles lo hice con un grid de cierto tamaño de tal forma que cupieran N tiles los cuales abarcaran todas las filas y columnas, y cada bloque tenía TSxTS hilos (TS es el tamaño de los tiles), o sea cada bloque tenia el mismo numero de hilos que numero de espacios dentro de cada tile. El programa se probó con 3 configuración diferentes para los tiles, primero con tiles de 8x8, después con tiles de 16x16, y al final con tiles de 32x32.

2. Desarrollo

Desarrollé 2 códigos, el primero llamado tilledMatrixMult.cu el cual tiene la multiplicación de matrices con tiles nada más (para poder observar la eficiencia de usar tiles), y el segundo programa se llama tilledMatrixMultComparison.cu en el cual se encuentra las 3 multiplicaciones (CPU, GPU sin tiles y GPU con tiles) para poder ver la comparación de los tiempos. Para cambiar el tamaño de los tiles solo se debe cambiar la variable ‘TS’ definida al inicio del código (igual para cambiar el tamaño de las matrices a multiplicar).

Las siguientes tablas muestran la comparación de los tiempos tomados en aplicar la multiplicación de matrices en CPU, en GPU sin tiles, y en GPU con tiles (el promedio de 20 tiempos):

Versión	Tiempo
CPU	60001.48046 ms
GPU sin tiles	697.42843 ms
GPU con tiles (8x8)	164.85272 ms
GPU con tiles (16x16)	87.72264 ms
GPU con tiles (32x32)	84.08228 ms

El Speedup lo obtuve con las siguiente fórmula y se muestra en la siguiente tabla:

$$Speedup = \frac{Tiempo\,secuencial}{Tiempo\ paralelo}$$

Versión	Speedup
CPU vs GPU con tiles (8x8)	363.97021
CPU vs GPU con tiles (16x16)	683.99082
CPU vs GPU con tiles (32x32)	713.60435
GPU sin tiles vs GPU con tiles (8x8)	4.23061
GPU sin tiles vs GPU con tiles (16x16)	7.95038
GPU sin tiles vs GPU con tiles (32x32)	8.29459

Hice un Makefile para compilar los 2 códigos sin problema el cual tiene la regla ‘all’ para compilar todos, la regla ‘rebuild’ para borrar los ejecutables y volver a compilar los códigos y la regla ‘clean’ para borrar los ejecutables.