INFORME COMPARATIVO TILES vs GRANULARIDAD

Hardware Utilizado:

CPU	: AMD Athlon(tm) II X3 455 Processor × 3 3,300 Mhz
Memoria	: 8 GB
SO	: Ubuntu 16.04 x64
Dispositivo CUDA	: GeForce GTX 1050 – 640 CudaCores
Ancho de Banda Memoria (bits)	: 128
Peak Memory Bandwidth (GB/s)	: 112.128000
Memoria global	: 1997mb
Memoria compartida	: 48kb
Memoria constante	: 64kb
Registros por bloque	: 65536
Multi Processor Count	: 5
Tamaño Warp	: 32
Threads por block	: 1024
Dimension Max block	: [1024, 1024, 64]
Dimension Max grid	: [2147483647, 65535, 65535]

Pruebas:

Para las pruebas respectivas , se utilizaron matrices cuadradas, en bloques respectivos de 16 y 32 . Se tomo como maximo 10240 como tamaño de la matriz, debido al tamaño de memoria de la GPU.

- Multiplicacion de matrices utilizando memoria compartida en cuda

		bloques de 16	bloques de 32
Simulacion	Matriz cuadrada	GPU (seg.)	GPU (seg.)
1	1024	0.014546	0.01655
2	2048	0.100888	0.101018
3	4096	0.605461	0.599393
4	8192	4.50833	4.22293
5	10240	8.61183	7.71257

En general podemos apreciar que la multiplicacion en matrices con el uso de memoria compartida es mas rapido debido a que la memoria local tiene un acceso mas rapido y los datos no se estan leyendo constantemente de la memoria global

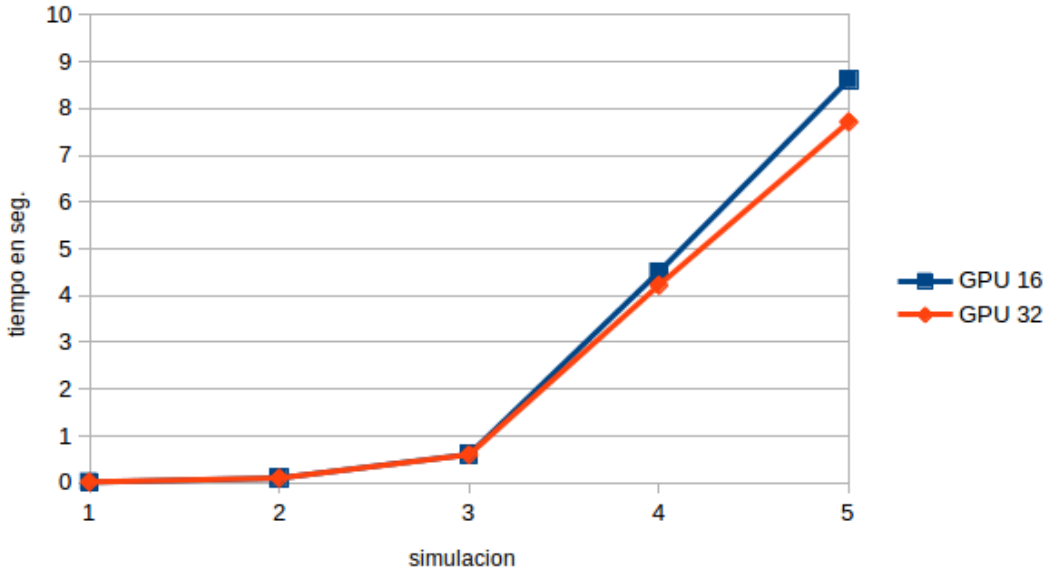

Fig 1. Multiplicacion usando tiles en bloques de 16 y 32

- Multiplicacion de matrices utilizando memoria compartida y granularidad de bloque.

Simulacion	Matriz cuadrada	GPU 16 - Granu	GPU 32- Granu
1	1024	0.016885	0.013992
2	2048	0.083965	0.070271
3	4096	0.524679	0.500023
4	8192	3.38834	3.37672
5	10240	6.50082	6.4411

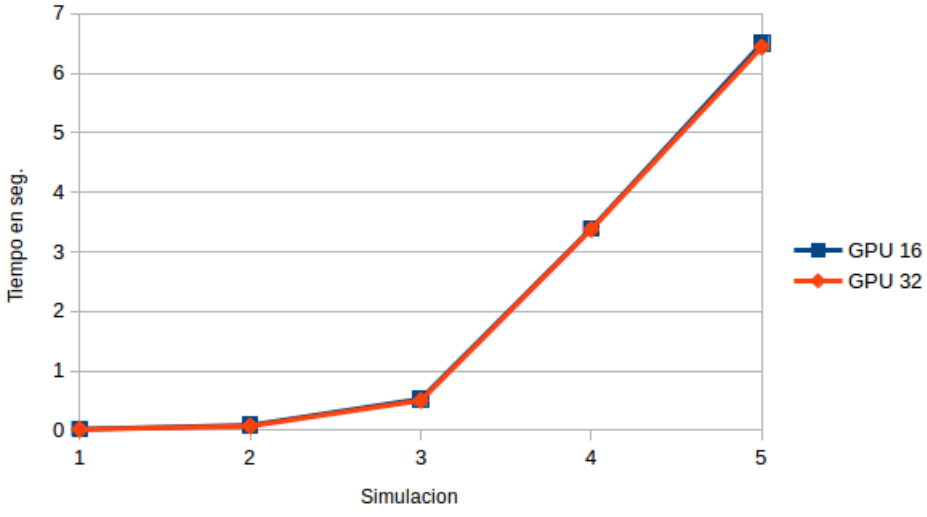

Fig 2. Multiplicacion usando granularidad de bloques de tamaño 16 y 32

Realizando una comparativa entre Multiplicacion Tiled y Mult. Con granularidad

En la grafica se aprecia que existe una variacion entre, la multiplicacion con tiles y la modificacion utilizando mas memoria compartida, se puede observar que se obtienen mejores resultados para este caso, las matrices de tamaño mayor a 3000 obtiene un rendimiento de mas de un cuarto para bloques de tamaño 32. En cuanto al uso de bloques de tamaño 16 tambien podemos observar una mejora muy parecida

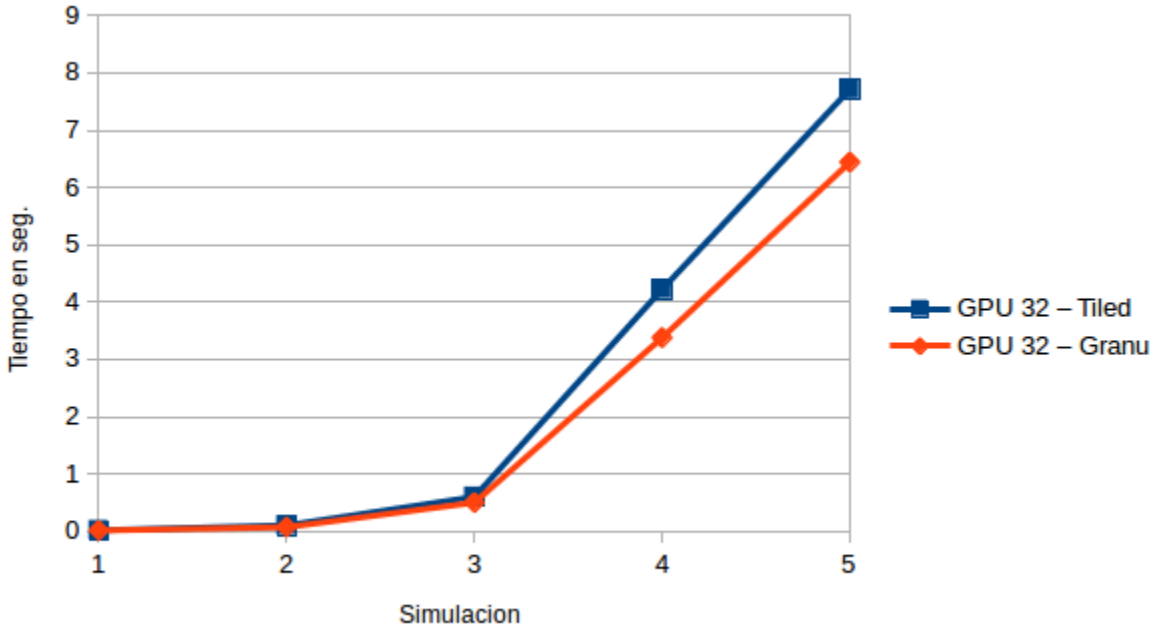

Fig 3. Diferencia entre Tiles y Granularidad en bloques de tamaño 32.

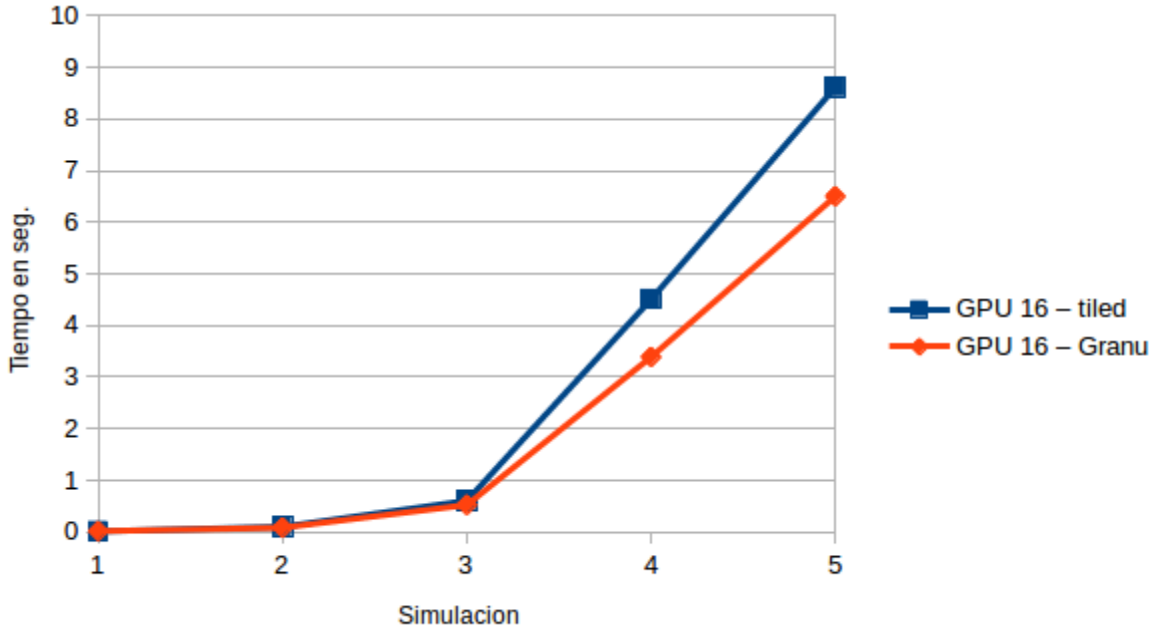

Fig 4.Diferencia entre Tiles y Granularidad en bloques de tamaño 16.

Conclusiones

- El uso de memoria local tiene un coste menor de lectura y escritura de datos, pero contiene un limite de memoria mucho menor a la memoria global, y esta depende de la arquitectura de la GPU utilizada.
- El uso adecuado de la memoria compartida, como en este caso, nos produjo un mejor rendimiento, a pesar de utilizar muchas veces menos bloques o menos cantidad de hilos.
- Para estas pruebas se observo un cambio mas rapido en la temperatura del dispositivo, a comparacion de los anteriores experimentos con multiplicaciones.