

# Trabajo Nro. 2

## Programación en GPU

1)

Compilacion: `nvcc -o entrega_2 entrega_2.cu`

Parametros para ejecutar: N, CUDA\_BLK

El algoritmo resuelve el problema con una granularidad de para cada hilo una matriz, los datos que se transmiten a la gpu son todas las matrices y un vector de N para guardar el determinante resultado de cada matriz.

Se usa un kernel el cual cada hilo resuelve el determinante de cada matriz y la suma se resuelve como una reducción, el cual otro kernel hace que cada hilo suma un resultado con otro resultado, para sumar todos los resultados se llama múltiples veces hasta terminar la reducción.

La suma también se resuelve con una reducción, al kernel se le asigna un bloque con matrices y cada hilo suma la una matriz con otra matriz, después la CPU sigue llamando al kernel hasta que todas las sumas se hayan completado.

El promedio de el determinante y la multiplicación con el determinante lo resuelve la CPU por ser operaciones que no dependen del tamaño del problema y la cantidad de operaciones no son lo suficiente para resolver con la GPU.

GPU GeForce 960(Maxwell) -----CPU Intel i5-4460

N	Tiempo Secuencial Intel i5-4460	GPU GeForce 960 (Maxwell)					
		128 HxB	Aceleración	256 HxB	Aceleración	512 HxB	Aceleración
4194304	0.368785	0.09261800	3.981785398	0.092766	3.975432809	0.094465	3.903932673
8388608	0.737912	-	-	0.18575000	3.972608345	0.18762600	3.932887766
16777216	1.477271	-	-	-	-	0.369128	4.002056197

GeForce 560 Ti(Fermin) ----- CPU Intel i5-2310

N	Tiempo Secuencial Intel i5-2310	GPU GeForce 960 (Maxwell)					
		128 HxB	Aceleración	256 HxB	Aceleración	512 HxB	Aceleración
2097152	0.224281	0.04661600	4.811245066	0.053117	4.222395843	0.070261	3.192112267
4194304	0.444845	0.094155	4.724603048	0.10116800	4.397091966	0.13702500	3.246451377
8388608	0.888365	-	-	0.204066	4.353321964	0.273836	3.244149783

Las mayores optimización se lograron al crear la menor cantidad de divergencia haciendo que siempre trabajan los hilos de se forma secuencial para que no estén en diferentes warps.

Teniendo acceso coalescente cuando van a buscar los datos a la memoria global.

Y por último desenrolando los for que no eran dinámicos.