

Ejercicio 7

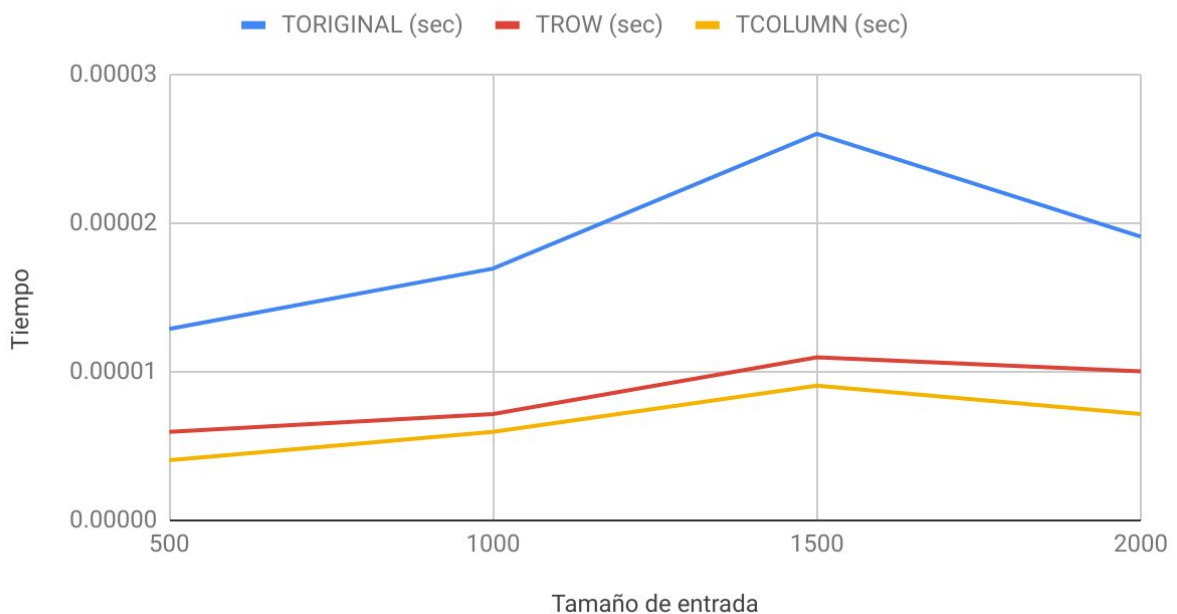
Tras la implementación de los 3 kernels en un mismo archivo CUDA se han tomado mediciones de tiempo con los diferentes tamaños de entrada y de bloque.

IMPORTANTE : Todas las mediciones se han realizado sobre una NVIDIA GTX 960M

Estos son los resultados obtenidos:

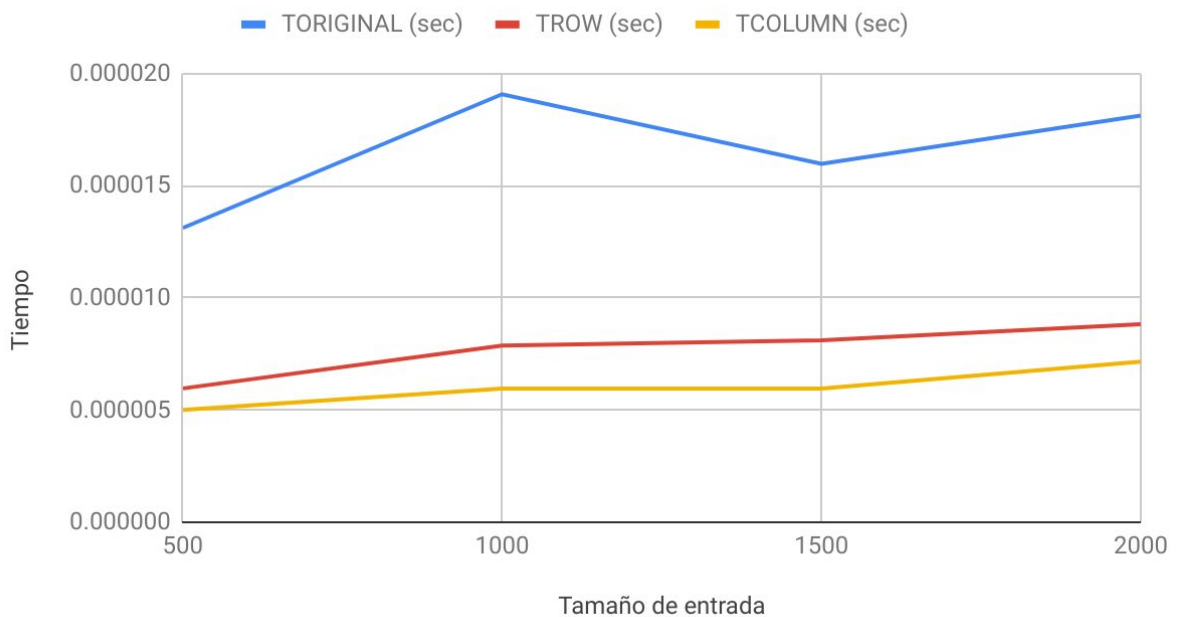
BSize : 64					
	TORIGINAL (sec)	TROW (sec)	SROW (sec)	TCOLUMN (sec)	SCOLUMN (sec)
500	0.00001287	5.96E-06	2.16	4.05E-06	3.17647
1000	1.69E-05	7.15E-06	2.37	5.96E-06	2.84
1500	2.60E-05	1.10E-05	2.36957	9.06E-06	2.86842
2000	1.91E-05	1.00E-05	1.90476	7.15E-06	2.66667

Tamaño de bloque: 64



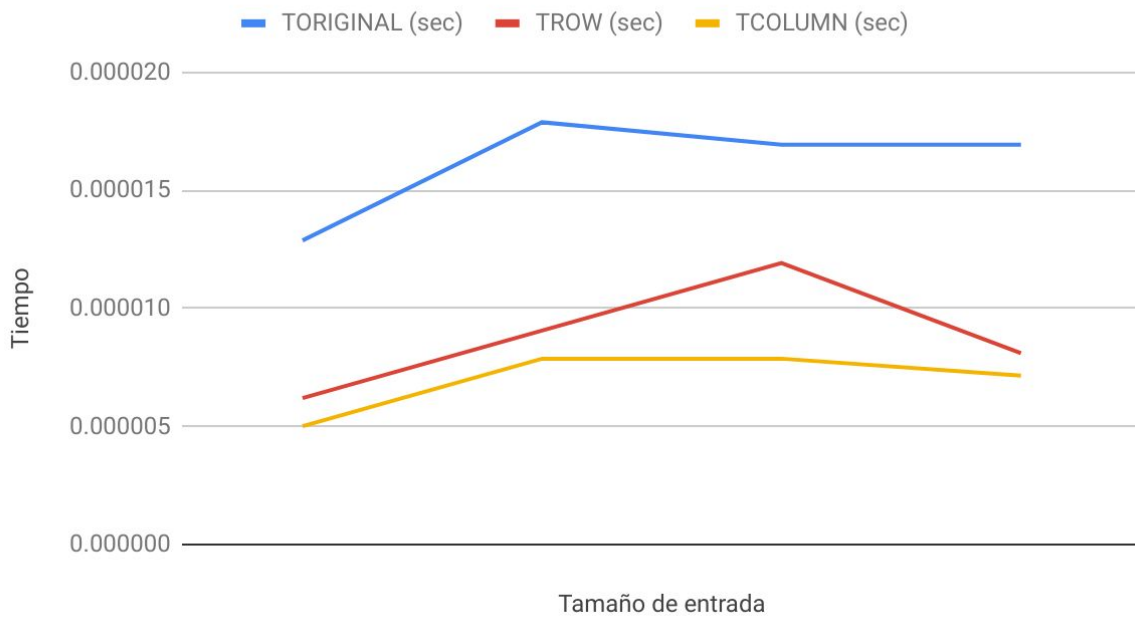
BSize : 256					
	TORIGINAL (sec)	TROW (sec)	SROW (sec)	TCOLUMN (sec)	SCOLUMN (sec)
500	0.00001311	5.96E-06	2.2	5.01E-06	2.61905
1000	1.91E-05	7.87E-06	2.42	5.96E-06	3.2
1500	1.60E-05	8.11E-06	1.97059	5.96E-06	2.68
2000	1.81E-05	8.82E-06	2.054E+00	7.15E-06	2.53333

Tamaño de bloque: 256



BSize : 1024					
	TORIGINAL (sec)	TROW (sec)	SROW (sec)	TCOLUMN (sec)	SCOLUMN (sec)
500	0.00001287	6.20E-06	2.07692	5.01E-06	2.57143
1000	1.79E-05	9.06E-06	1.97	7.87E-06	2.27273
1500	1.69E-05	1.19E-05	1.42	7.87E-06	2.15152
2000	1.69E-05	8.11E-06	2.08824	7.15E-06	2.36667

Tamaño de bloque: 1024



Conclusiones

Como se puede apreciar en las graficas la version del kernel de suma de matrices en la que cada hebra se encarga de una columna es mejor en todos los tamaños de bloque y de entrada. Esto se debe principalmente a que el acceso a los datos es coalescente.