MATRIX MULTIPLICATIONS SEQUENTIAL/PARALLEL WITH CUDA C

Presentado por:
ANDERSON ALBERTO OCHOA ESTUPIÑÁN

Presentado a:
JOHN OSORIO RIOS

UNIVERSIDAD TECNOLÓGICA
Marzo 11 del 2015
PEREIRA

Introducción

En el siguiente trabajo se intenta mostrar el comportamiento de una multiplicación de matrices realizada de diferentes formas, de las cuales se busca analizar cual es el la versión óptima del algoritmo según sea la situacion presentada (cálculos usando distintos tamaños de matrices como eje central de la actividad).

Para resolver el parcial se utilizó un equipo con las siguientes características principales (Por medio de la plataforma dispuesta para la compilación y ejecución de algoritmos en CUDA C):

- Procesador Intel Core I7-3770k a 3.4 Ghz.
- 16 Gb de ram DDR3
- Tarjeta gráfica NVIDIA Tesla k40

Requerimientos del parcial

El trabajo propuesto por el docente cuenta con una serie de requerimientos que serán resumidos a continuación, además realizar los algoritmos para valores enteros y flotantes:

1. Codificar un algoritmo en lenguaje C que realice una multiplicación de matrices de tamaño NxM teniendo en cuenta las propiedades de dicha operación.
2. Codificar una versión paralela del algoritmo anterior usando CUDA y lenguaje C.
3. Modificar el algoritmo paralelo para que haga uso de la memoria compartida de la GPU.
4. Tomar y documentar tiempos de ejecución con matrices de diferentes tamaños y asignando distintos valores al Block Size para cada uno de los algoritmos anteriores.
5. Graficar los resultados.
6. Conclusiones.

Desarrollo

Los puntos 1, 2 y 3 se encuentran desarrollados en el siguiente repositorio:
https://github.com/aaochoa/Hpc
En la carpeta “Parcial”.

4. Para valores enteros.

Valores Tomados con un Block Size fijo de 32 y un tamaño del Tile de 32.

Matrix size	**N**	**M**	**O**	**Sequential Time**	**Parallel Time**	**Parallel with tiles time**	**Acceleration**	**Acceleration with tiles**
1	**4**	**8**	**10**	3.00E-006	0.101488	0.000271	2.96E-005	0.0110701
2	**32**	**16**	**32**	1.07E-004	0.09444	0.00026	0.00113299	0.411538
3	**64**	**32**	**128**	0.000675	0.093595	0.000606	0.00721192	1.11386
4	**128**	**128**	**256**	0.014058	0.089276	0.001032	0.157467	13.6221
5	**256**	**128**	**160**	0.015972	0.095483	0.00053	0.167276	30.1358
6	**320**	**256**	**240**	0.053884	0.09249	0.001459	0.582593	36.9321
7	**400**	**200**	**600**	0.139458	0.089195	0.002119	1.56352	65.8131
8	**512**	**360**	**600**	0.310999	0.096181	0.003566	3.23348	87.2123
9	**1024**	**512**	**960**	1.5835	0.099881	0.012153	15.8538	130.297
10	**2048**	**1024**	**1200**	9.81615	0.127337	0.055787	77.088	175.958
11	**2048**	**2048**	**1500**	29.9283	0.21006	0.136249	142.475	219.659

Valores Tomados con un Block Size fijo de 4 y un tamaño del Tile de 4.

M	N	O	Sequential Time	Parallel Time BS4	Parallel with tiles time BS4	Acceleration BS4	Acceleration with tiles BS4
1024	**512**	**960**	1.6084	0.14138	0.045366	11.3764	35.4539
2048	**1024**	**1200**	9.38493	0.316424	0.220333	29.6594	42.5943
2048	**2048**	**1500**	29.8121	0.649265	0.545313	45.9166	54.6697

Valores Tomados con un Block Size fijo de 16 y un tamaño del Tile de 16.

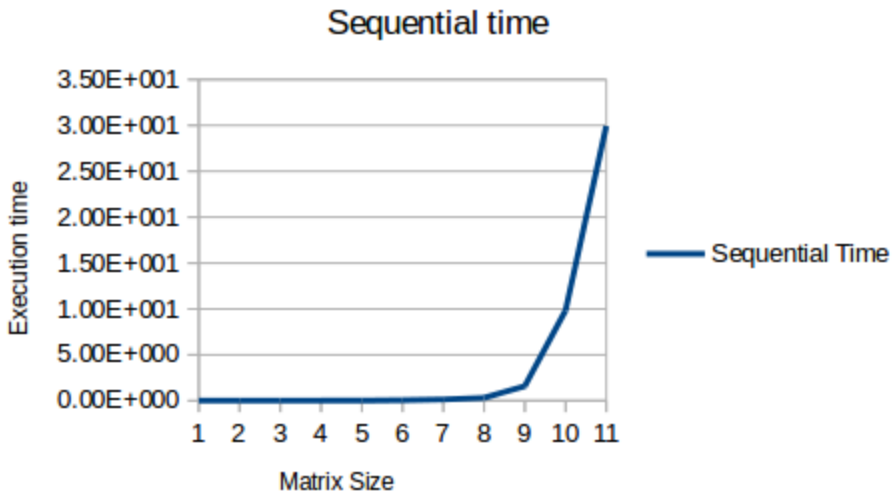
M	N	O	Sequential Time	Parallel Time BS16	Parallel with tiles time BS16	Acceleration BS16	Acceleration with tiles BS16
1024	**512**	**960**	1.68326	0.123109	0.012698	13.673	132.561
2048	**1024**	**1200**	7.59397	0.163004	0.057293	46.5876	132.546
2048	**2048**	**1500**	31.1042	0.245799	0.144055	126.543	215.919

Valores Tomados con un Block Size fijo de 32 y un tamaño del Tile de 32.

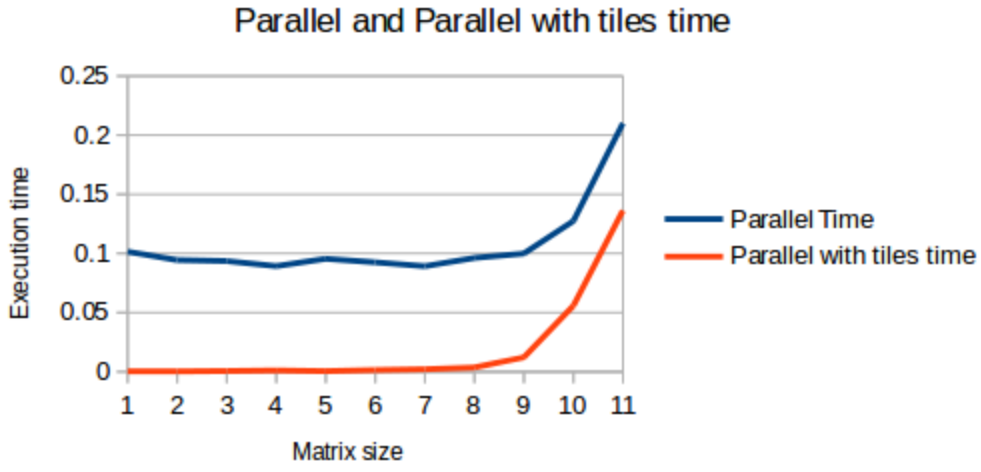
M	N	O	Sequential Time	Parallel Time BS32	Parallel with tiles time BS32	Acceleration BS32	Acceleration with tiles BS32
1024	**512**	**960**	1.71203	0.123737	0.012159	13.8361	140.804
2048	**1024**	**1200**	16.2845	0.165782	0.055878	98.2285	291.43
2048	**2048**	**1500**	30.8266	0.231585	0.136669	133.111	225.556

5. Gráficas

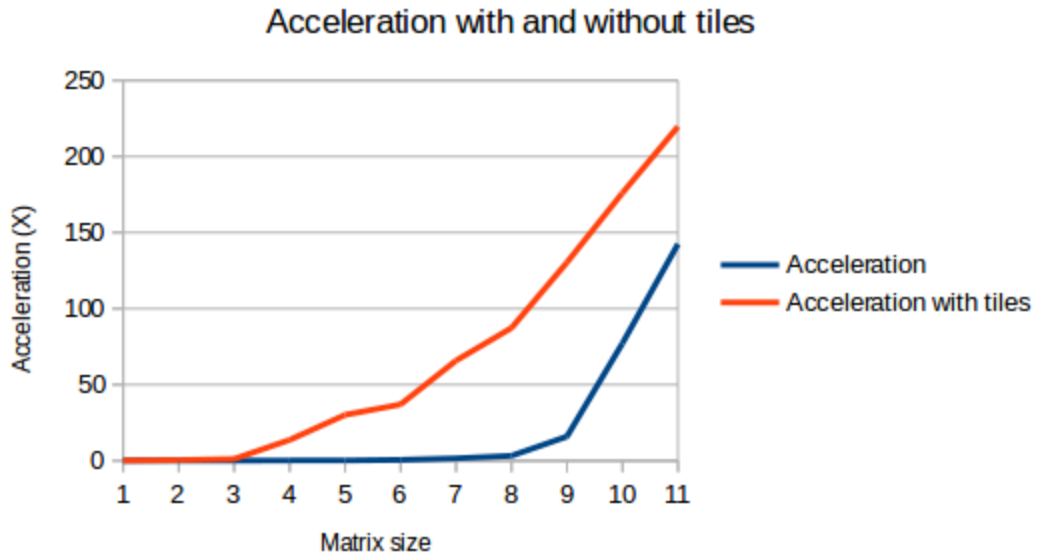
- En la siguiente gráfica se observa el comportamiento del algoritmo secuencial mientras variamos el tamaño de la matriz.



- En esta gráfica se realiza una comparación de los tiempos de ejecución del algoritmo paralelo con Tiles y sin ellos.

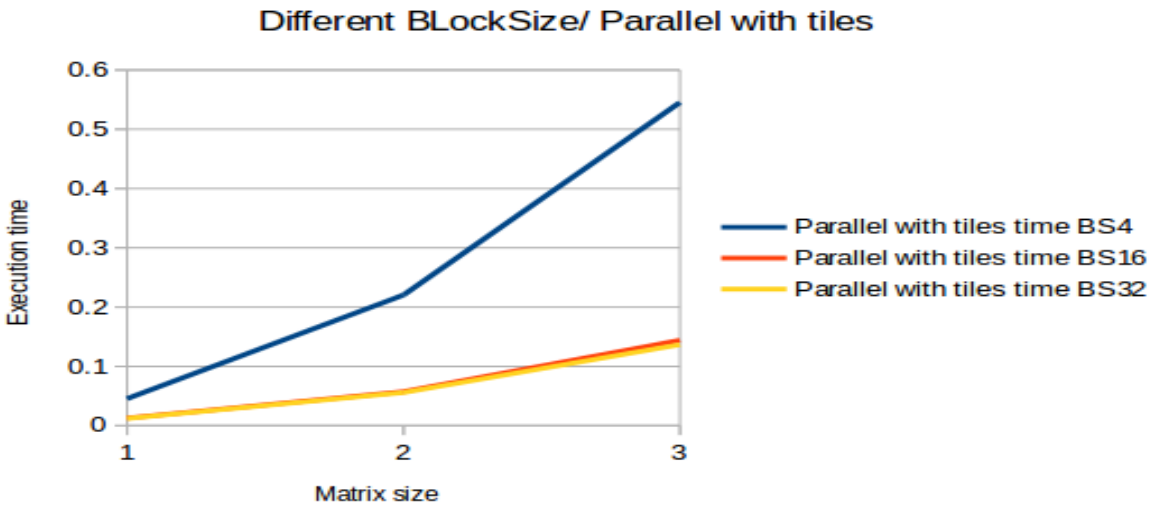


- Finalmente comparamos la aceleración obtenida en la ejecución de los algoritmos sin y con Tiles.

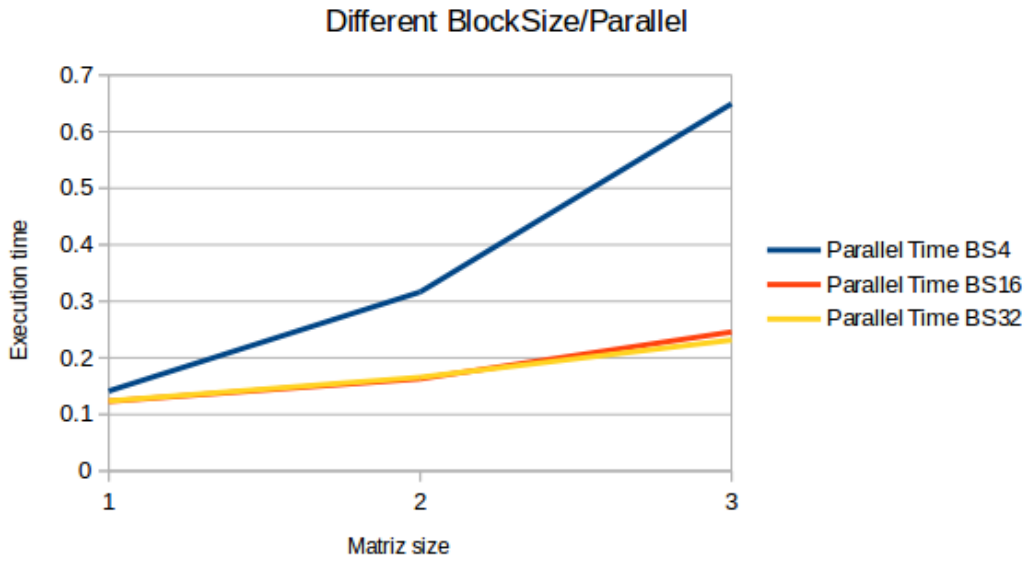


- Las siguientes gráficas son las encargadas de mostrar el comportamiento de los algoritmos modificando el tamaño del bloque y el Tile, tiempos de ejecución y aceleración.

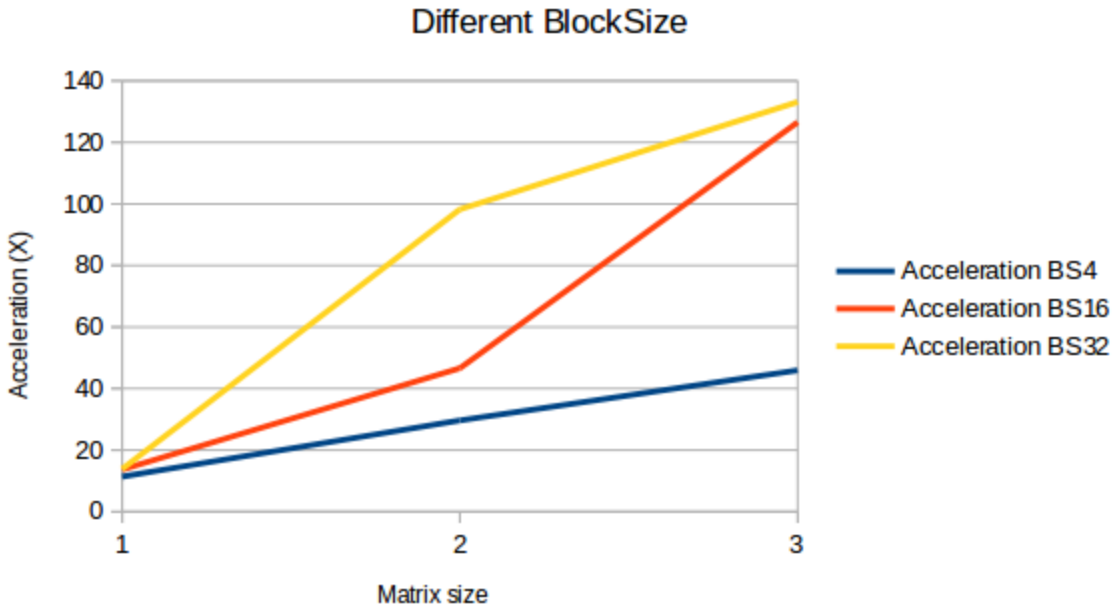
Parallel with tiles time BS4	**Parallel with tiles time BS16**	**Parallel with tiles time BS32**
0.045366	0.012698	0.012159
0.220333	0.057293	0.055878
0.545313	0.144055	0.136669



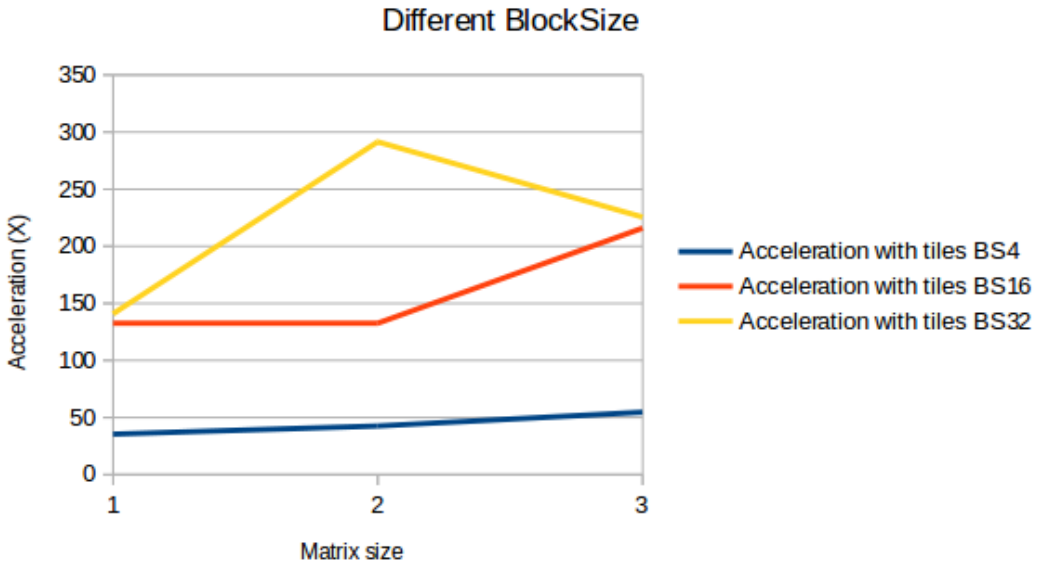
Parallel Time BS4	Parallel Time BS16	Parallel Time BS32
0.14138	0.123109	0.123737
0.316424	0.163004	0.165782
0.649265	0.245799	0.231585



Acceleration BS4	Acceleration BS16	Acceleration BS32
11.3764	13.673	13.8361
29.6594	46.5876	98.2285
45.9166	126.543	133.111



Acceleration with tiles BS4	Acceleration with tiles BS16	Acceleration with tiles BS32
35.4539	132.561	140.804
42.5943	132.546	291.43
54.6697	215.919	225.556



4. Para valores flotantes.

Valores Tomados con un Block Size fijo de 32 y un tamaño del Tile de 32.

Matrix size	N	M	O	Sequential Time	Parallel Time	Parallel with tiles time	Acceleration	Acceleration with tiles
1	**4**	**8**	**10**	4.00E-006	0.1023	0.000433	3.91E-005	0.00923788
2	**32**	**16**	**32**	1.07E-004	0.111088	0.000316	0.0009632	0.338608
3	**64**	**32**	**128**	0.000699	0.097373	0.000482	0.00717858	1.45021
4	**128**	**128**	**256**	0.019352	0.098004	0.000897	0.197461	21.5741
5	**256**	**128**	**160**	0.019704	0.091065	0.000459	0.216373	42.9281
6	**320**	**256**	**240**	0.053863	0.095368	0.001093	0.564791	49.28
7	**400**	**200**	**600**	0.141371	0.097386	0.001842	1.45166	76.7486
8	**512**	**360**	**600**	0.452076	0.107712	0.004245	4.19708	106.496
9	**1024**	**512**	**960**	1.61003	0.100213	0.010819	16.0661	148.815
10	**2048**	**1024**	**1200**	9.98214	0.154554	0.048837	64.5867	204.397
11	**2048**	**2048**	**1500**	30.4428	0.226919	0.13596	134.157	223.91

Valores Tomados con un Block Size fijo de 4 y un tamaño del Tile de 4.

M	N	O	Sequential Time	Parallel Time BS4	Parallel with tiles time BS4	Acceleration BS4	Acceleration with tiles BS4
1024	**512**	**960**	1.44125	0.138365	0.041168	10.4163	35.0091
2048	**1024**	**1200**	9.53144	0.302563	0.211464	31.5023	45.0736
2048	**2048**	**1500**	31.6592	0.614095	0.52575	51.5543	60.2173

Valores Tomados con un Block Size fijo de 16 y un tamaño del Tile de 16.

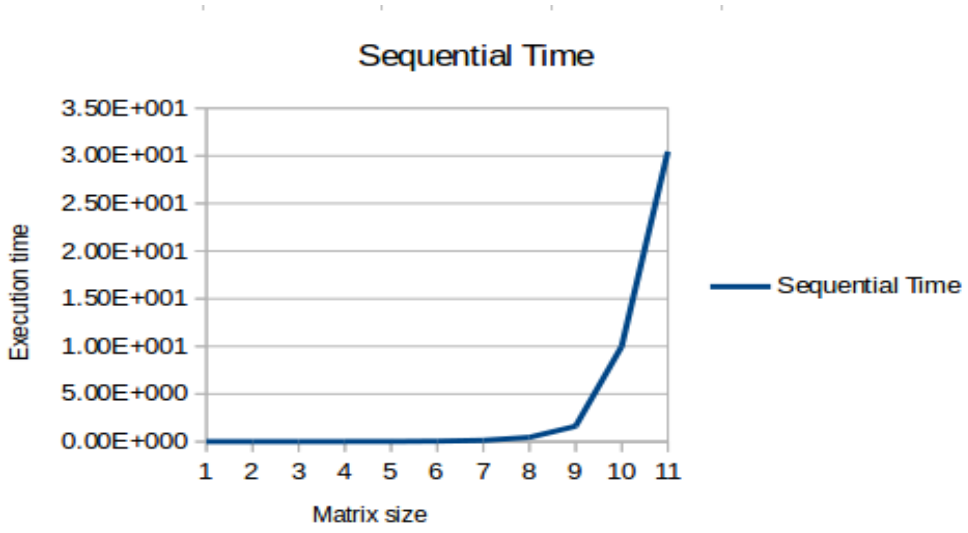
M	N	O	Sequential Time	Parallel Time BS16	Parallel with tiles time BS16	Acceleration BS16	Acceleration with tiles BS16
1024	**512**	**960**	1.44018	0.102143	0.012235	14.0996	117.71
2048	**1024**	**1200**	9.41678	0.14783	0.058189	63.7001	161.831
2048	**2048**	**1500**	30.1403	0.249477	0.155925	120.814	193.3

Valores Tomados con un Block Size fijo de 32 y un tamaño del Tile de 32.

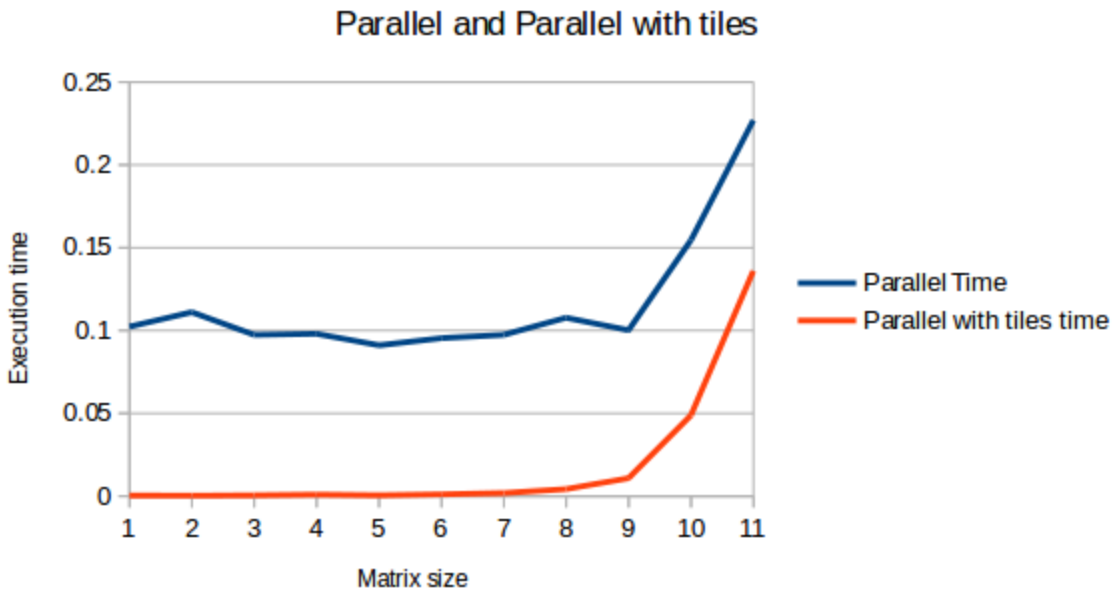
M	N	O	Sequential Time	Parallel Time BS32	Parallel with tiles time BS32	Acceleration BS32	Acceleration with tiles BS32
1024	**512**	**960**	1.45325	0.104785	0.010418	13.8688	139.494
2048	**1024**	**1200**	9.5693	0.142979	0.048123	66.928	198.851
2048	**2048**	**1500**	30.5385	0.228666	0.136944	133.551	223

5. Graficas

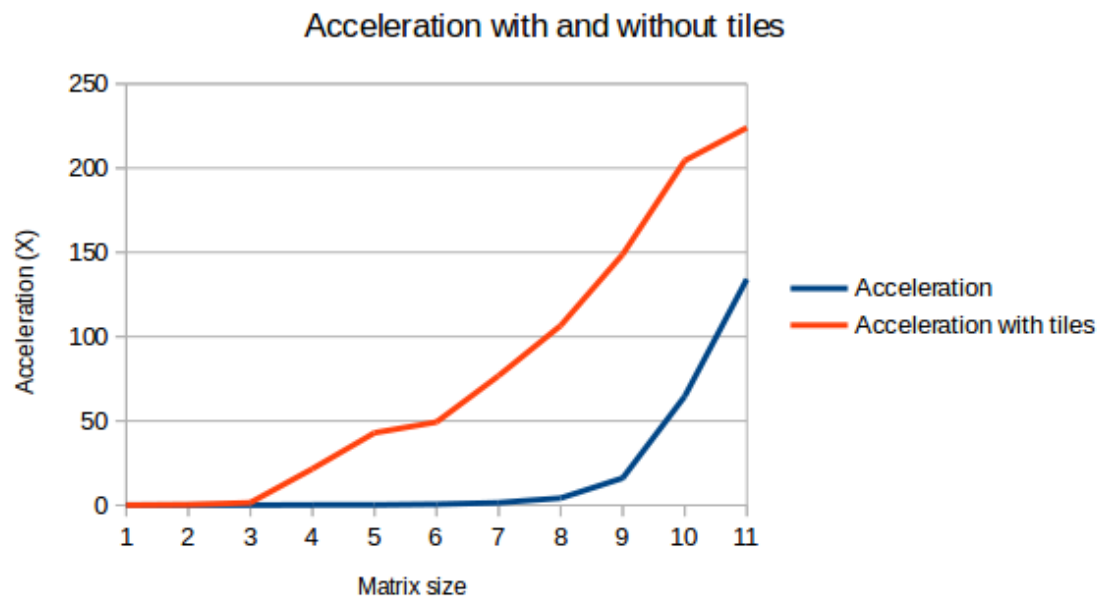
- En la siguiente gráfica se observa el comportamiento del algoritmo secuencial realizando cambios en el tamaño de la matriz relacionado con el tiempo de ejecución.



- En esta gráfica se realiza una comparación de los tiempos de ejecución del algoritmo paralelo con Tiles y sin ellos.

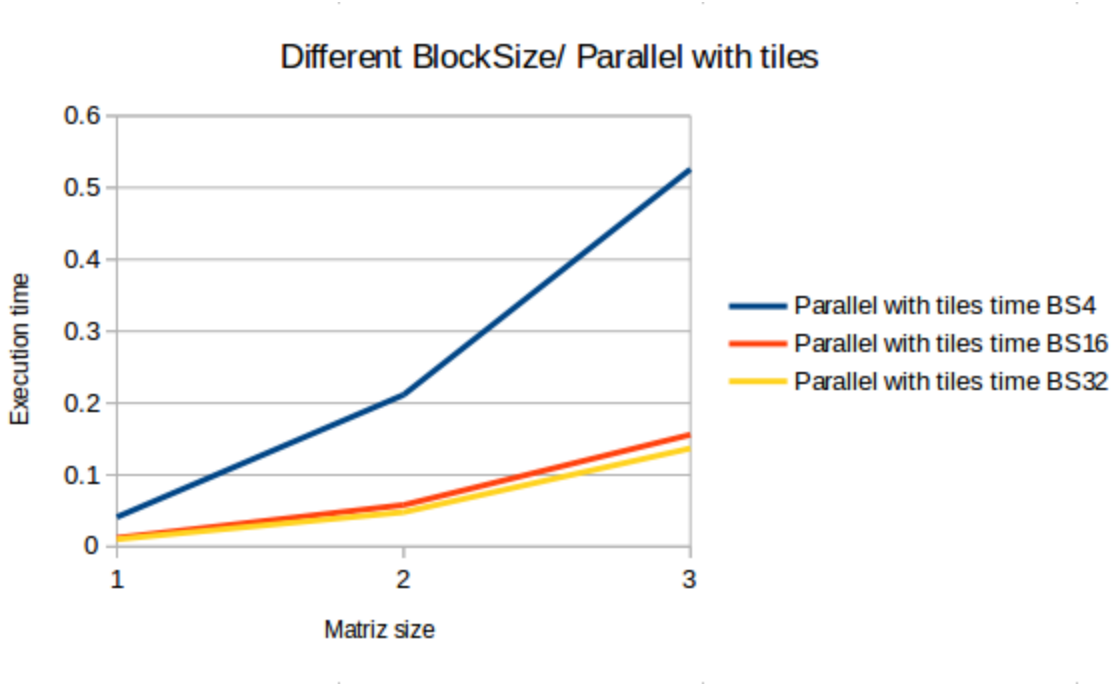


- Finalmente comparamos la aceleración obtenida en la ejecución de los algoritmos sin y con Tiles.

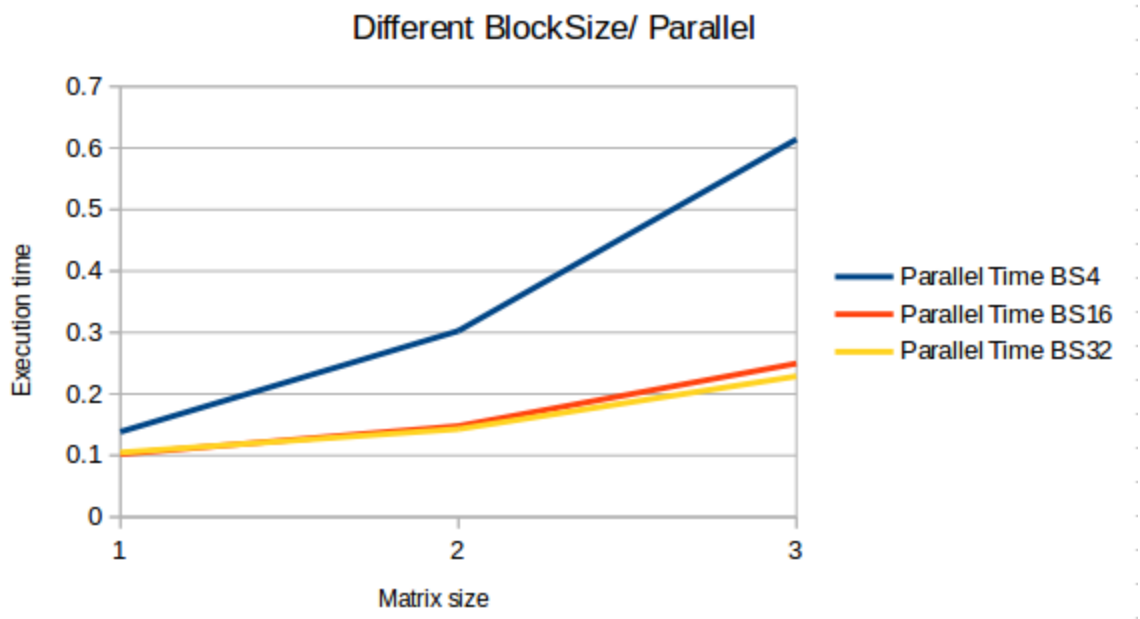


- Las siguientes gráficas son las encargadas de mostrar el comportamiento de los algoritmos modificando el tamaño del bloque y el Tile, tiempos de ejecución y aceleración.

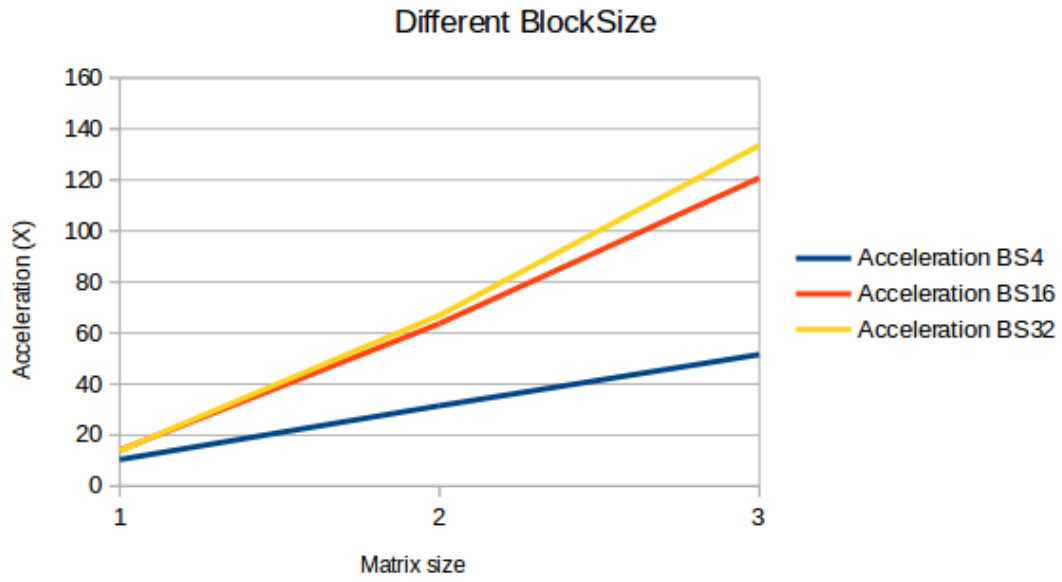
Parallel with tiles time BS4	Parallel with tiles time BS16	Parallel with tiles time BS32
0.041168	0.012235	0.010418
0.211464	0.058189	0.048123
0.52575	0.155925	0.136944



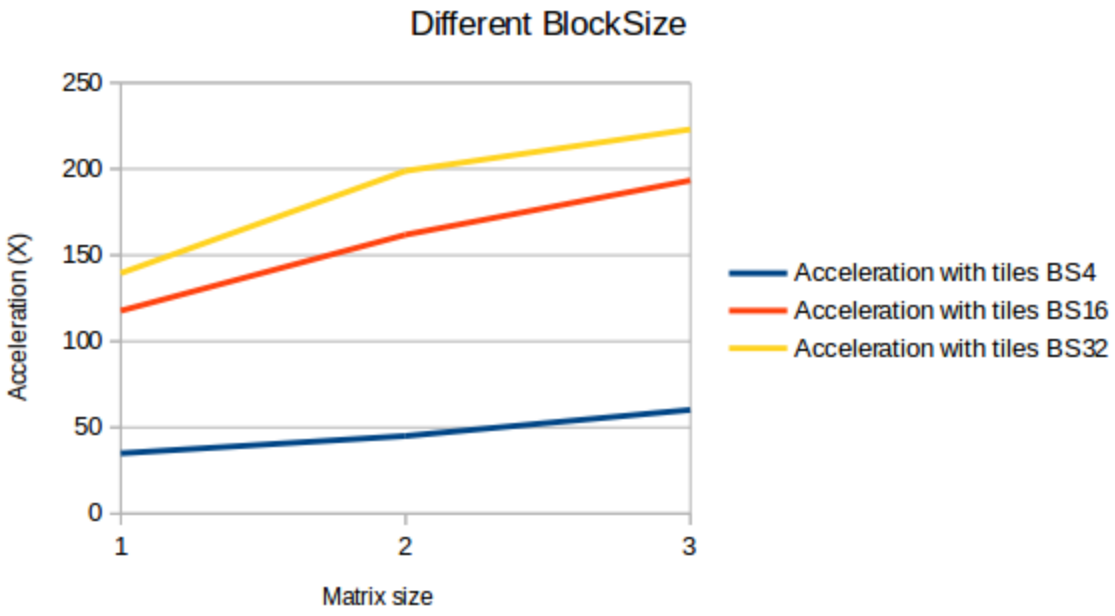
Parallel Time BS4	Parallel Time BS16	Parallel Time BS32
0.138365	0.102143	0.104785
0.302563	0.14783	0.142979
0.614095	0.249477	0.228666



Acceleration BS4	Acceleration BS16	Acceleration BS32
10.4163	14.0996	13.8688
31.5023	63.7001	66.928
51.5543	120.814	133.551



Acceleration with tiles BS4	Acceleration with tiles BS16	Acceleration with tiles BS32
35.0091	117.71	139.494
45.0736	161.831	198.851
60.2173	193.3	223



6. Conclusiones

De los valores recolectados y las gráficas elaboradas podemos concluir:

- Debido a las condiciones en las que se realizan las pruebas los resultados pueden presentar variaciones ya que el recurso utilizado es compartido.
- La manera en la que se inicializa el **dimgrid** cambia notoriamente con la utilizada para los algoritmos con matrices NxN ya que la variable que usamos para determinar los hilos por bloque tiene que ser calculada con respecto al tamaño de la matriz resultante.
- Los tiempos de ejecución del algoritmo secuencial son óptimos siempre y cuando los tamaños de las matrices sean del orden de decenas o centenas ya que se produce una pérdida de tiempo realizando la copia de los datos del host al device debido a la poca cantidad de los mismos.

- Cuando el tamaño de las matrices es considerablemente grande, osea, matrices de más de millones de datos, ejecutar las versiones en paralelo se convierte en una opción a tener en cuenta, destacando la versión con tiles la cual entregó tiempos de ejecución más bajos en todos los casos evaluados.
- Gracias a la reducción de tiempos de ejecución al usar memoria compartida la aceleración obtenida es mucho más alta que la que se obtuvo con el algoritmo en paralelo que se realizó primero.
- Al realizar los cambios del tamaño de bloque y de los Tiles se puede observar que la versión que entrega los mejores tiempos en paralelo es la que hace uso de un valor de de tamaño de bloque de 32 y un Tile de 32.
- Los tiempos de ejecución de los algoritmos con enteros no tienen mucha diferencia con los algoritmos que realizaban la misma operación con números en punto flotante, los tiempos varían aproximadamente en ±2 segundos con las matrices más grandes y ±0.02 segundos en las versiones paralelas.
- Las aceleraciones obtenidas con los algoritmos paralelos tienden a ser exponenciales conforme se aumente el tamaño de la matriz siempre y cuando los tamaños no superen las características de la tarjeta utilizada.