CONVOLUCIÓN EN UNA DIMENSIÓN:

SECUENCIAL, BÁSICA, CON CATCHING Y CON TILING

HIGH PERFORMANCE COMPUTING (HPC)

ESTUDIANTES:
Diego Alejandro Ramirez Ramirez
Carlos Eduardo Zuleta Uribe

PROFESOR:
John H. Osorio

UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍAS
INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

21/10/2015

ESPECIFICACIONES DEL SISTEMA

La máquina en la cual se corrió el algoritmo tiene las siguientes características:

CPU:

processor : 1
vendor_id : GenuineIntel
cpu family : 6
model : 58
model name : Intel® Core™ i7-3770K CPU @ 3.50GHz
stepping : 9
cpu MHz : 1600.000
cache size : 8192 KB
cpu cores : 4

GPU:

Tesla K40c: 3.5
Global memory: 11519mb
Shared memory: 48kb
Constant memory: 64kb
Block registers: 65536
Warp size: 32
Threads per block: 1024
Max block dimensions: [1024, 1024, 64]
Max grid dimensions: [2147483647, 65535, 65535]

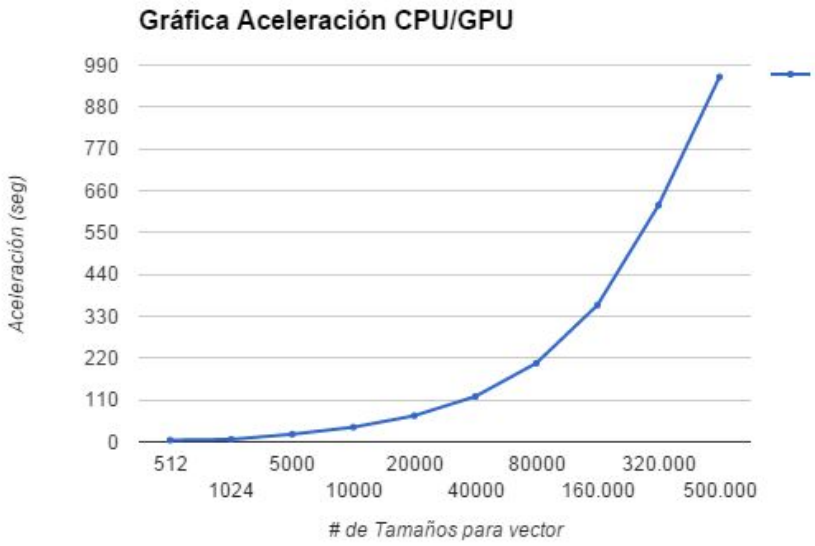
PROCESO APLICADO

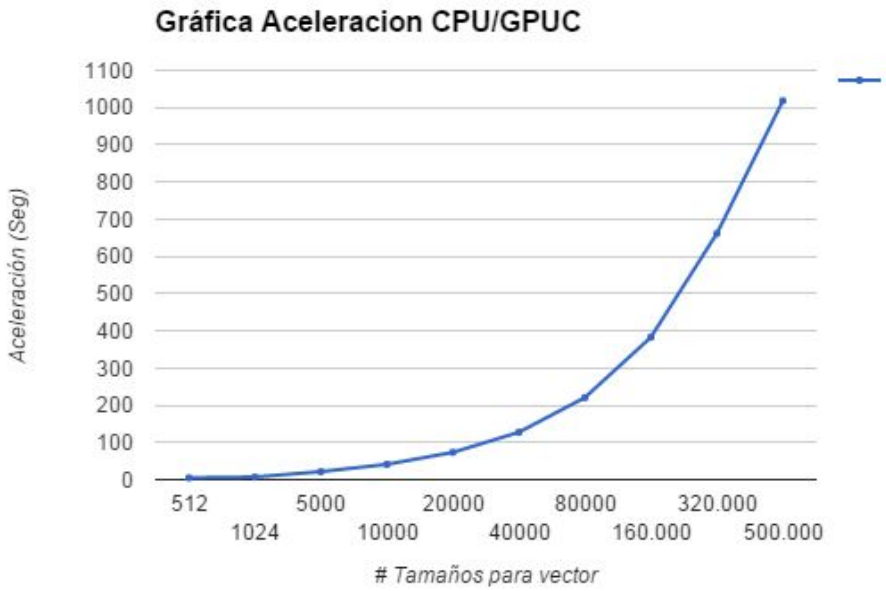
Se tomaron 10 datos para cada uno de los tiempos con las cuatro diferentes implementaciones del algoritmo: Convolución secuencial, convolución básica, convolución con catching y convolución con tiling, teniendo en cuenta que se debía ir incrementando el tamaño del vector y dejando un valor constante para la Mask, para este caso un valor muy grande, luego se calculó una media más precisa. De la misma manera se halló la aceleración entre Convolución secuencial respecto a cada una de las demás y la aceleración de la implementación con catching con respecto a la implementación con tiling tomando el valor medio de cada tiempo Secuencial y dividiéndolo entre cada valor de tiempo medio de los otros tres respectivamente, del mismo modo para la Aceleración de convolución con Tiling en este caso, tiempo de Convolución con Catching dividido entre tiempo de Convolución con tiling.

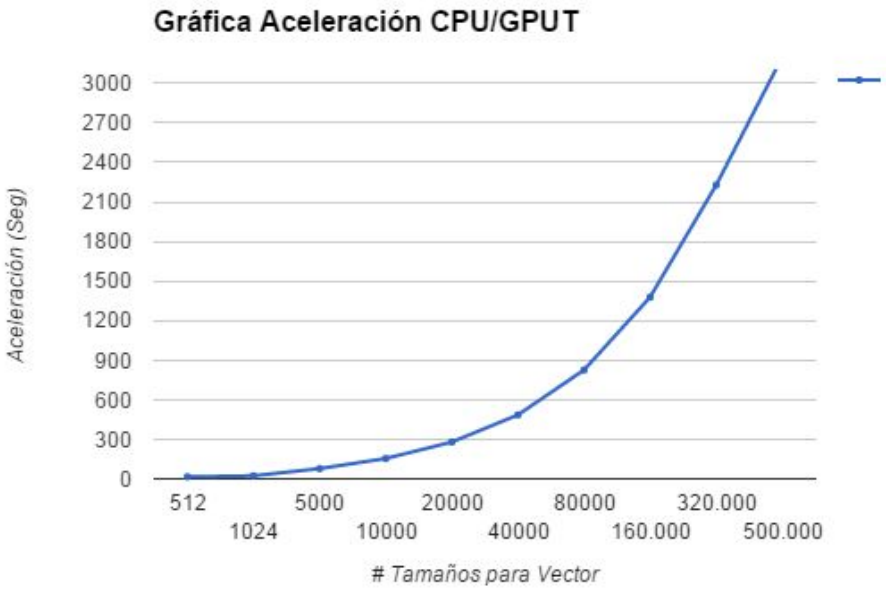
En la siguiente tabla se muestra con más detalle lo mencionado anteriormente.

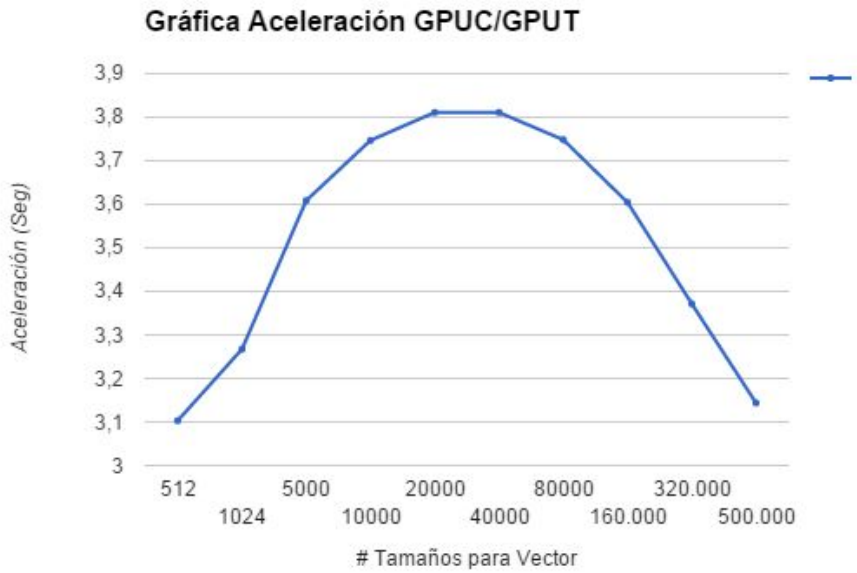
Mask	Dim vectores		Time Convolution				Aceleraciones			
M	n	m	Secuencial	Básica	Convolution Catching	Convolution Tiling	CPU /GPU	CPU/GPUC	CPU/GPUT	GPUC/GPUT
5000	1	512	0,00505150	0,00113700	0,00104200	0,00033583	4,442832014	4,847888676	15,04183664	3,102760325
5000	1	1024	0,01546400	0,00237717	0,00219300	0,00067133	6,505214183	7,051527588	23,03487108	3,266649785
5000	1	5000	0,07877683	0,00393617	0,00366250	0,00101533	20,01357411	21,50903208	77,58741493	3,607201599
5000	1	10000	0,21145618	0,00550467	0,00514467	0,00137367	38,41396124	41,10199099	153,9352101	3,745200812
5000	1	20000	0,48654366	0,00708950	0,00663900	0,00174283	68,62876931	73,28568459	279,1687428	3,809321621
5000	1	40000	1,04002607	0,00872083	0,00816817	0,00214433	119,2576934	127,3266925	485,0121343	3,80919448
5000	1	80000	2,15108585	0,01039700	0,00975717	0,00260417	206,8948591	220,4620653	826,0159091	3,746748484
5000	1	160.000	4,38503981	0,01220633	0,01146350	0,00318150	359,2430985	382,5219008	1378,293198	3,603174603
5000	1	320.000	8,84495163	0,01422900	0,01337400	0,00396767	621,6144234	661,3542418	2229,255868	3,370744039
5000	1	500.000	15,75605679	0,01642983	0,01547266	0,00492233	958,9908593	1018,315971	3200,934677	3,143360969

A continuación se mostrarán las gráficas para cada una de las aceleraciones obtenidas en las cuatro diferentes implementaciones de Convolución.

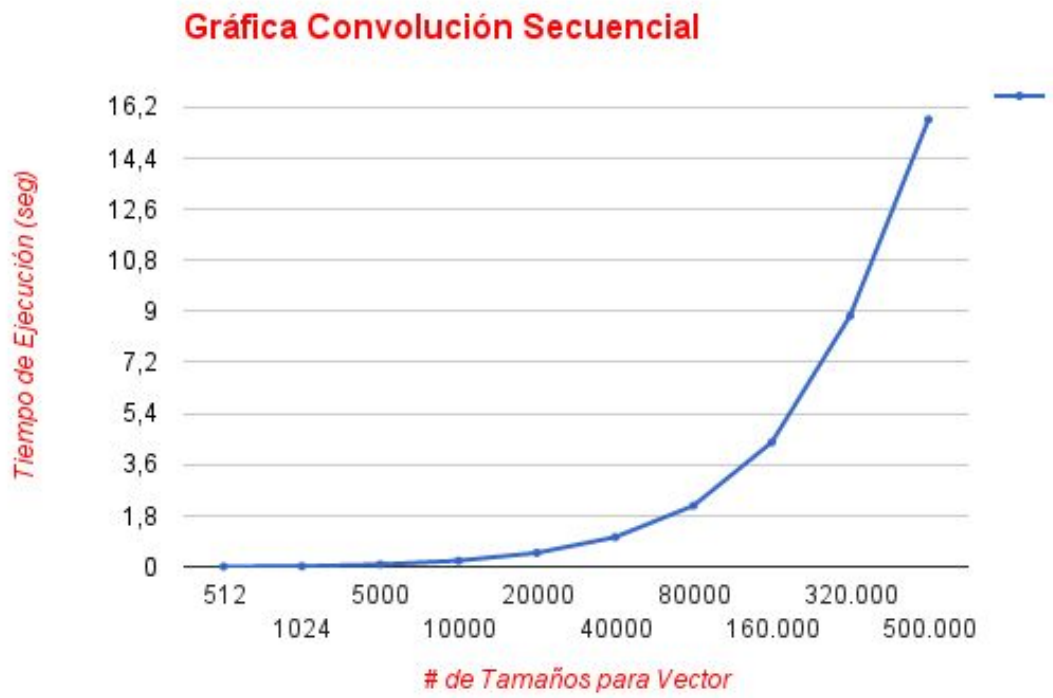


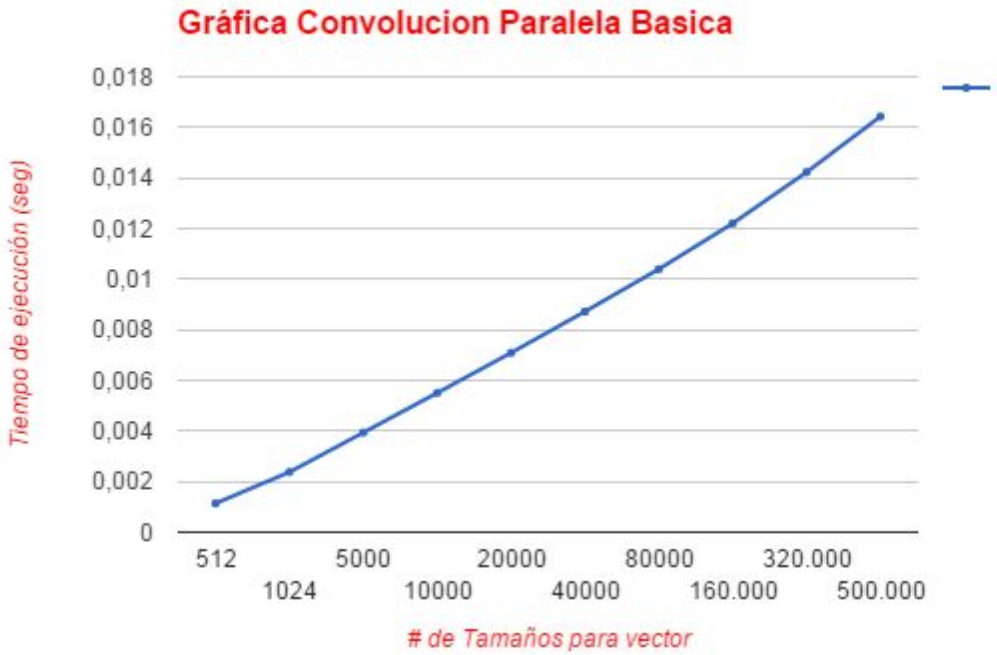



Gráfica Aceleración CPU/GPUT
Aceleración (Seg)
3000
2700
2400
2100
1800
1500
1200
900
600
300
0
512
1024
5000
10000
20000
40000
80000
160.000
320.000
500.000
Tamaños para Vector


Gráfica Aceleración GPUC/GPUT
Aceleración (Seg)
3,9
3,8
3,7
3,6
3,5
3,4
3,3
3,2
3,1
3
512
1024
5000
10000
20000
40000
80000
160.000
320.000
500.000
Tamaños para Vector

Gráficos de tiempos con cada una de las implementaciones.





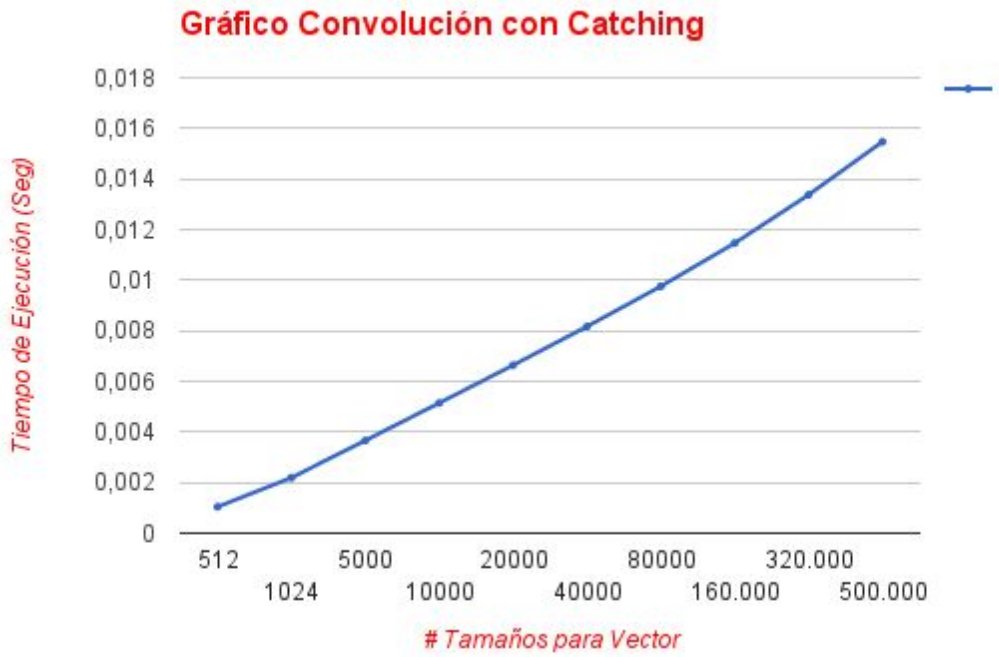


Gráfico Convolución con Catching
0,018
0,016
0,014
0,012
0,01
0,008
0,006
0,004
0,002
0
Tiempo de Ejecución (Seg)
512
1024
5000
10000
20000
40000
80000
160.000
320.000
500.000
Tamaños para Vector


Gráfica Convolución con Tiling
0,0054
0,0048
0,0042
0,0036
0,003
0,0024
0,0018
0,0012
0,0006
0
Tiempo de ejecución (seg)
512
1024
5000
10000
20000
40000
80000
160.000
320.000
500.000
de Tamaños para vector

Variando el valor de la Mask a un valor bastante pequeño (50) hicimos el mismo proceso para la toma de datos y para hallar las respectivas aceleraciones.

Mask	Dim vectores		Time Convolution				Aceleraciones			
M	n	m	Secuencial	Basica	Convolution Catching	Convolution Tiling	CPU /GPU	CPU/GPUC	CPU/GPUT	GPUC/GPUT
50	1	512	0,00019117	0,00005933	0,00005400	0,00003300	3,222147312	3,540185185	5,793030303	1,636363636
50	1	1024	0,00034100	0,00011683	0,00011150	0,00006917	2,918770864	3,058295964	4,929882897	1,611970507
50	1	5000	0,00107033	0,00018267	0,00017717	0,00011400	5,85936388	6,041259807	9,388859649	1,554122807
50	1	10000	0,00253567	0,00025883	0,00025333	0,00017267	9,796661902	10,00935539	14,68506399	1,467133839
50	1	20000	0,00546200	0,00034700	0,00033767	0,00023783	15,74063401	16,17555602	22,96598411	1,419795652
50	1	40000	0,01131500	0,00046983	0,00045750	0,00034600	24,08317902	24,73224044	32,70231214	1,322254335
50	1	80000	0,02313250	0,00065083	0,00063683	0,00050367	35,54307576	36,3244508	45,92788929	1,264379455
50	1	160.000	0,04652467	0,00094867	0,00093567	0,00077817	49,04199564	49,72337469	59,78728298	1,202397934
50	1	320.000	0,09390933	0,00146467	0,00144133	0,00126467	64,11637434	65,15463496	74,25599564	1,139688614
50	1	500.000	0,16730799	0,00214700	0,00211533	0,00191667	77,92640429	79,09309186	87,29097341	1,103648515

A Continuación se mostrarán las gráficas para cada uno de los tiempos de ejecución.

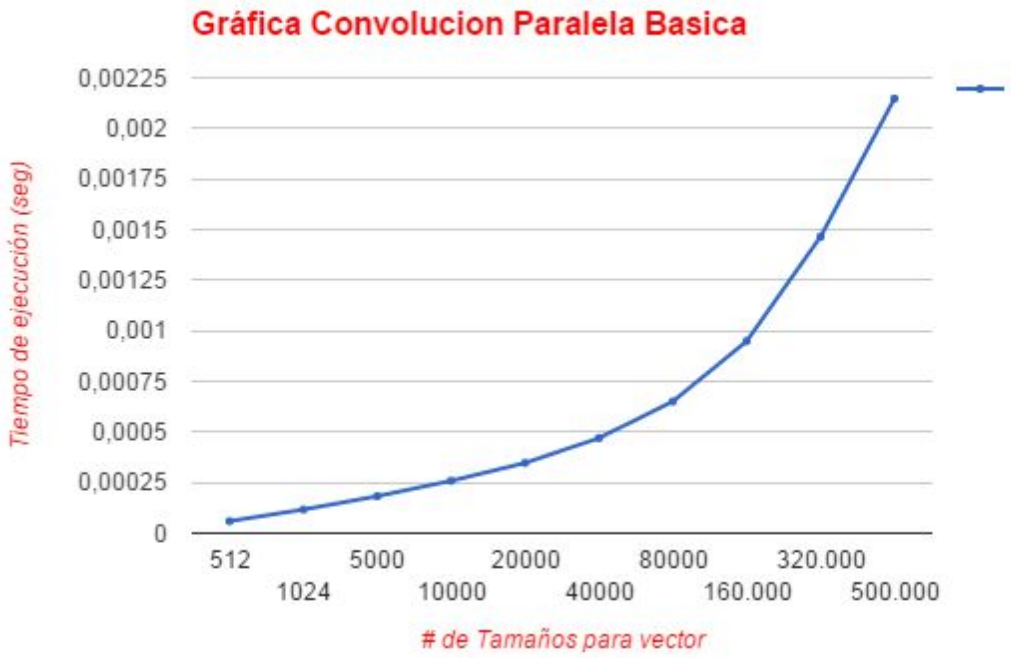
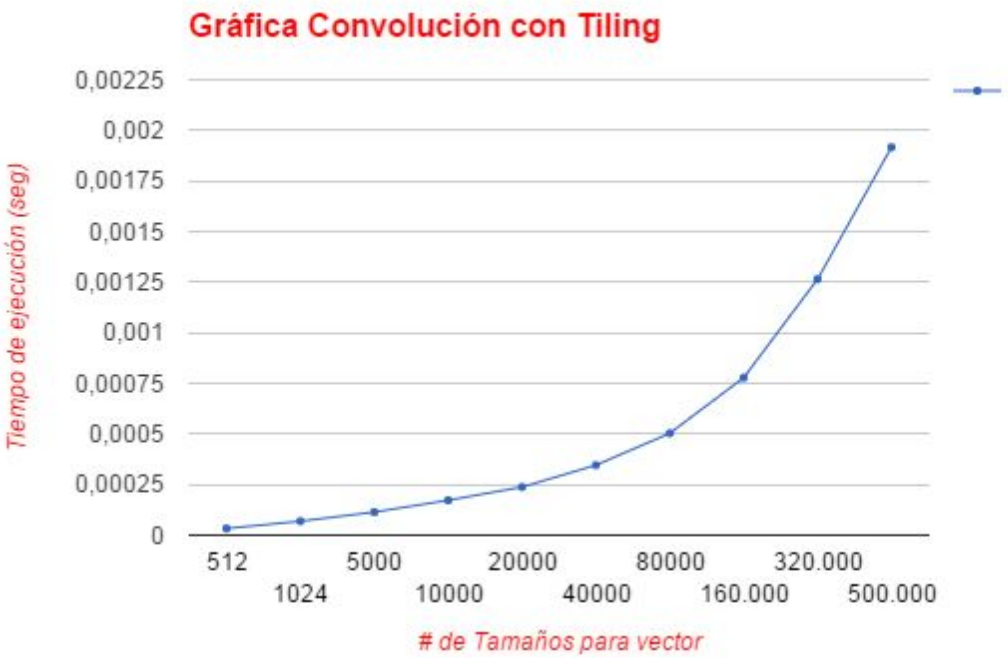
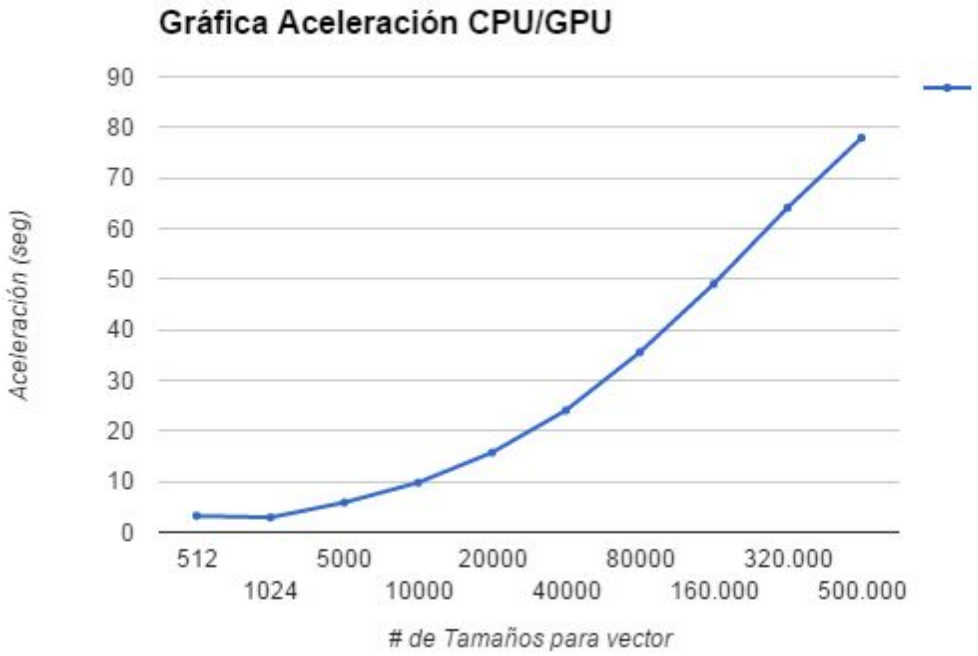
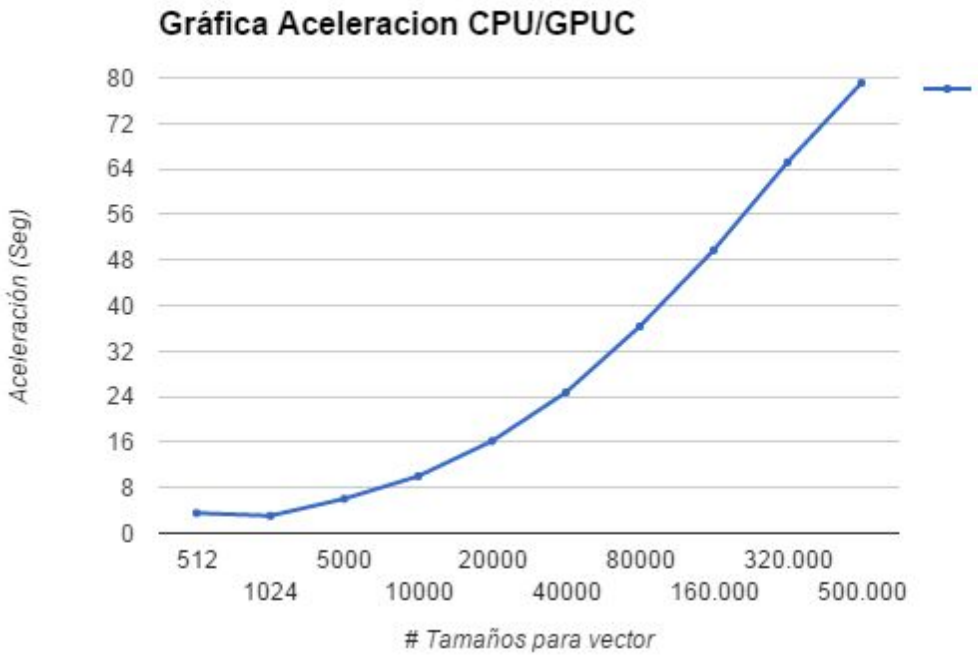


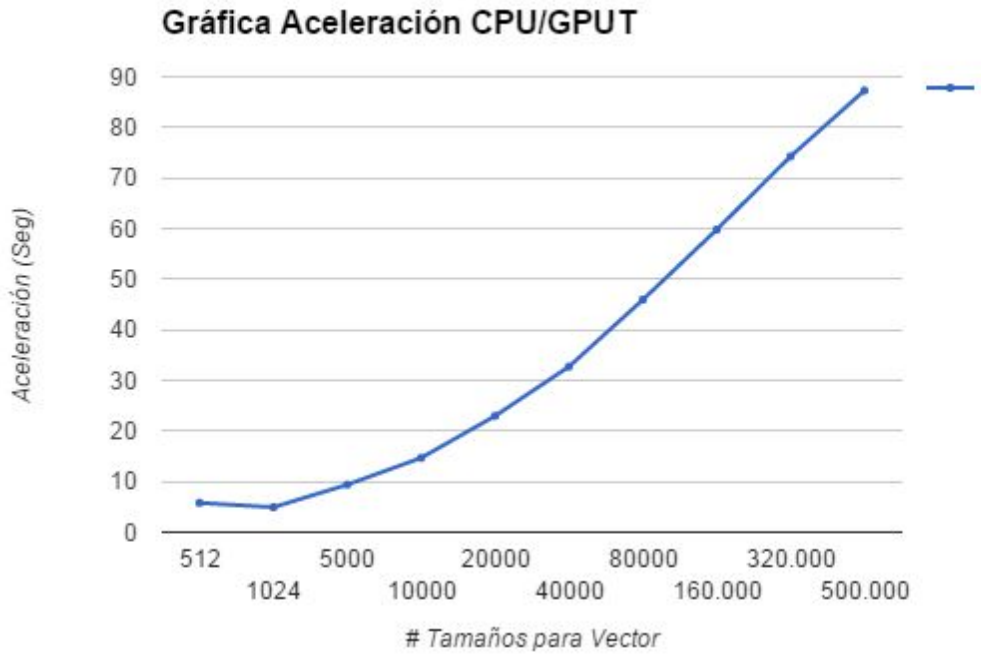

Gráfico Convolución con Catching
0,00225
0,002
0,00175
0,0015
0,00125
0,001
0,00075
0,0005
0,00025
0
Tiempo de Ejecución (Seg)
512
1024
5000
10000
20000
40000
80000
160.000
320.000
500.000
Tamaños para Vector

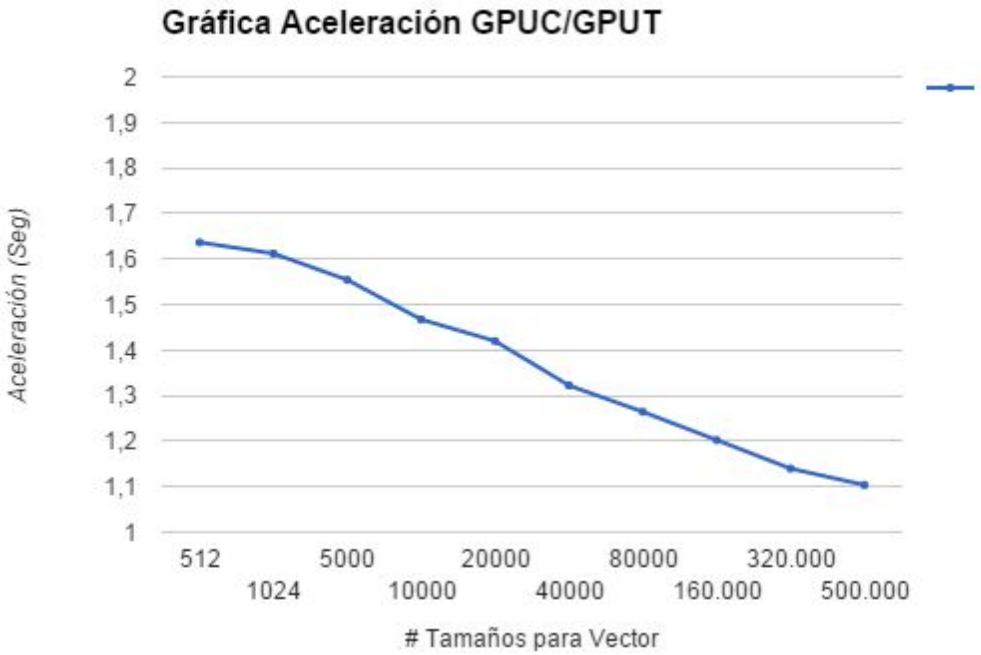

Gráfica Convolución con Tiling
0,00225
0,002
0,00175
0,0015
0,00125
0,001
0,00075
0,0005
0,00025
0
Tiempo de ejecución (seg)
512
1024
5000
10000
20000
40000
80000
160.000
320.000
500.000
de Tamaños para vector

A continuación podemos ver las gráficas de cada una de las aceleraciones y las implementaciones con una Mask mucho más pequeña.






Gráfica Aceleración CPU/GPUT
90
80
70
60
50
40
30
20
10
0
Aceleración (Seg)
512
1024
5000
10000
20000
40000
80000
160.000
320.000
500.000
Tamaños para Vector


Gráfica Aceleración GPUC/GPUT
2
1,9
1,8
1,7
1,6
1,5
1,4
1,3
1,2
1,1
1
Aceleración (Seg)
512
1024
5000
10000
20000
40000
80000
160.000
320.000
500.000
Tamaños para Vector

CONCLUSIONES

Analizando el trabajo realizado se encontró que:

- La implementación secuencial es eficiente solamente con vectores relativamente pequeños.
- Pudimos notar que al hacer una comparación entre las gráficas realizadas con cada una de las implementaciones, obviamente la implementación secuencial tarda muchísimo más a medida que se aumenta el valor de las dimensiones en los vectores.
- El tiempo de ejecución aumenta o disminuye dependiendo también del tamaño del vector máscara, entre mas grande el vector máscara mas tarde en ejecutarse.
- La aceleración a también depende del tamaño de la máscara, entre mas grande la máscara más alta es la aceleración.
- Se puede notar en las gráficas que en las aceleraciones de CPU/GPU,CPU/GPC y CPU/GPUT que la aceleración aumenta entre más datos hayan, pero en GPUC/GPUT cuando la memoria compartida llega a su máxima capacidad la aceleración disminuye.