

PARCIAL I HPC

ALEJANDRO SUÁREZ GARCÍA

UNIVERSIDAD TECNOLÓGICA DE PEREIRA  
INGENIERÍA DE SISTEMAS Y COMPUTACIÓN  
PEIRA RISARALDA  
11 DE MARZO DE 2015

## INTRODUCCIÓN

En el siguiente documento se ilustrara la aceleración del algoritmo de multiplicación de matrices utilizando tiles y bloques de tamaños 4x4, 16x16, 32x32, utilizando matrices de diferentes tamaños que en el caso de los enteros, contienen datos estáticos, en este caso solo el número 1 en la primera y el 2 en la segunda y en el caso de los flotantes solo el número 1.5 en la primera y 1.75 en la segunda.

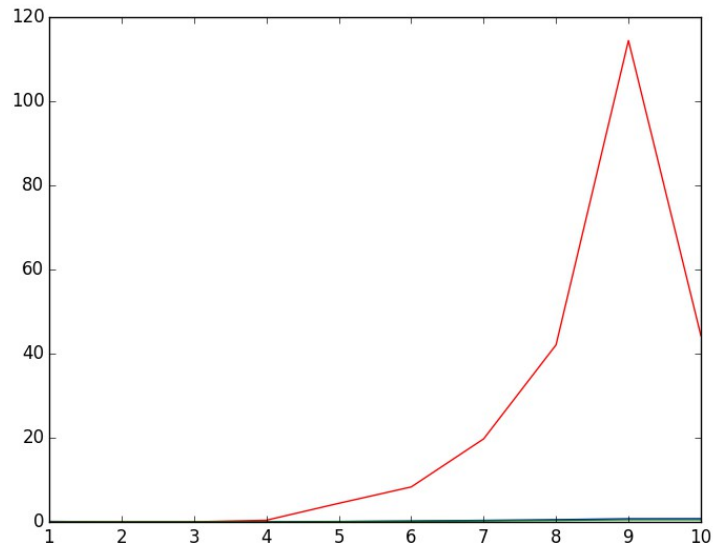
Los tamaños de las matrices fueron:

Filas de la primera matriz	Tamaño común para las matrices	Columnas de las segunda matriz
32	32	32
32	63	37
127	129	128
513	512	500
1024	1024	1024
1500	1000	1613
2000	1032	1993
2001	1500	2002
2048	2048	2048
2050	2051	2052

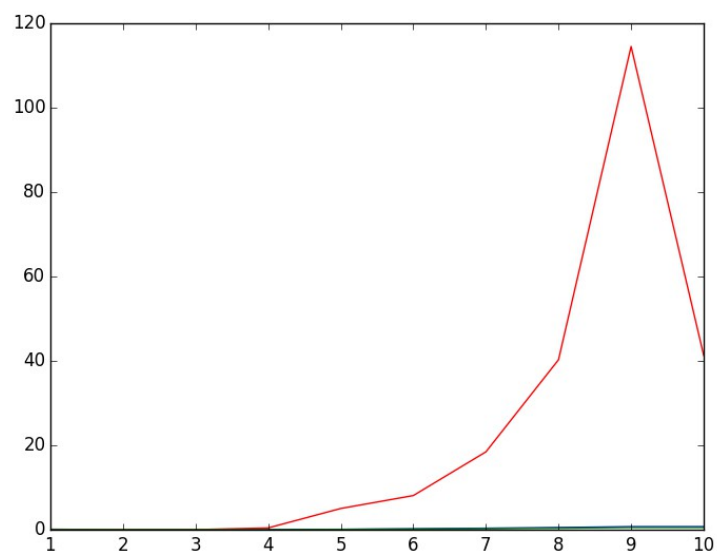
## RESULTADOS

### Algoritmo secuencial vs algoritmo paralelo vs algoritmo paralelo con tiles:

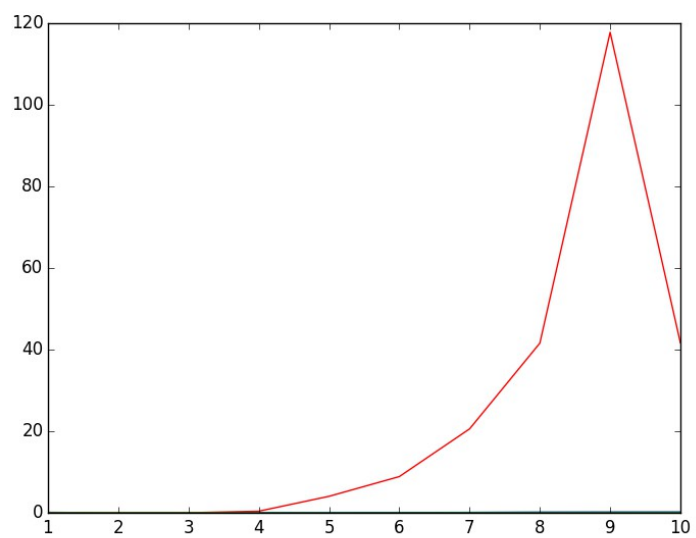
Se utilizo el color rojo para representar los tiempos del algoritmo secuencial, el azul para el algoritmo paralelo y el verde para el algoritmo paralelo utilizando tiles.



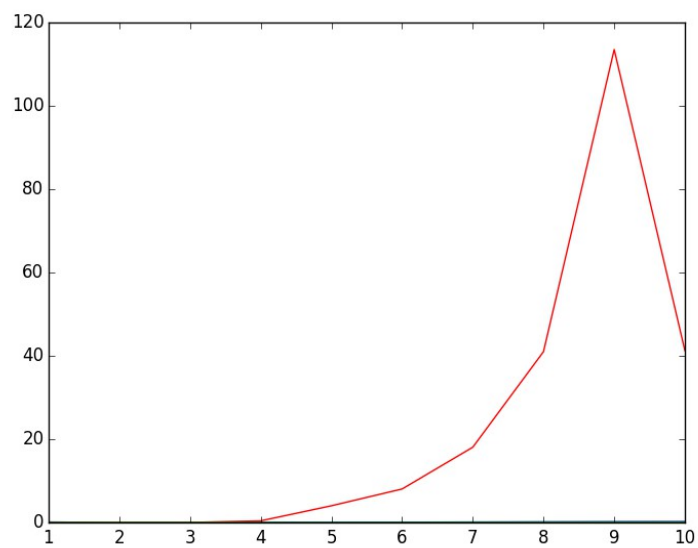
Gráfica 1: Enteros con tile 4x4



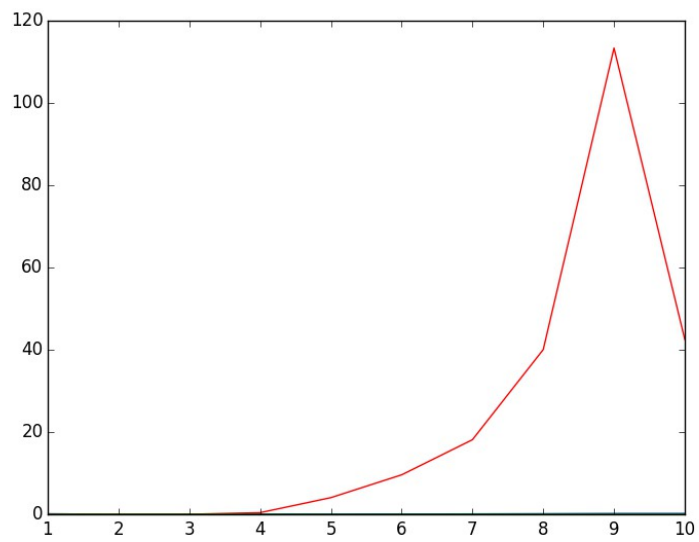
Gráfica 2: Flotantes con tile 4x4



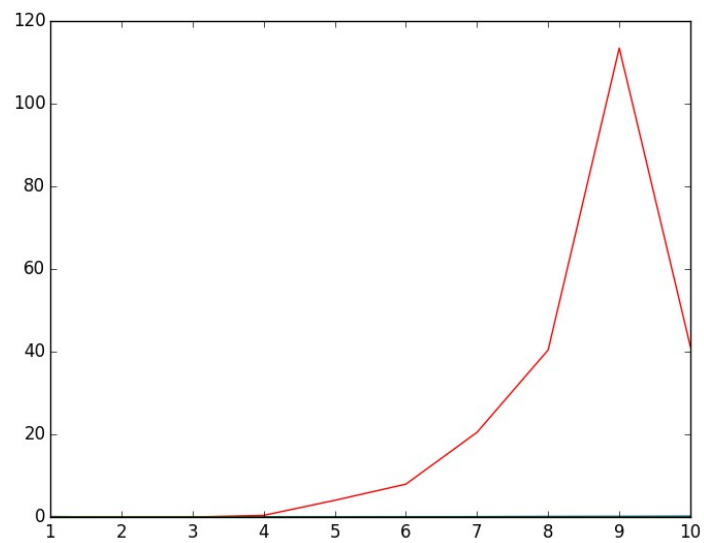
*Gráfica 3: Enteros con tile 16x16*



*Gráfica 4: Flotantes con tile 16x16*



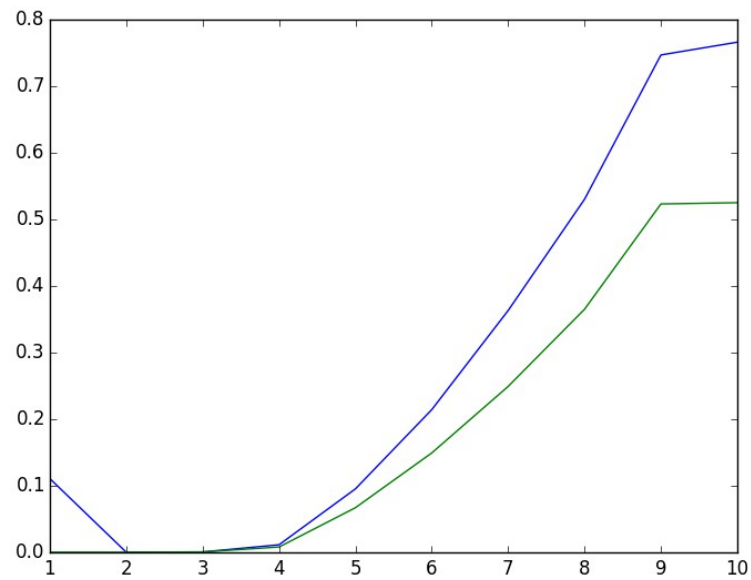
*Gráfica 5: Enteros con tile 32x32*



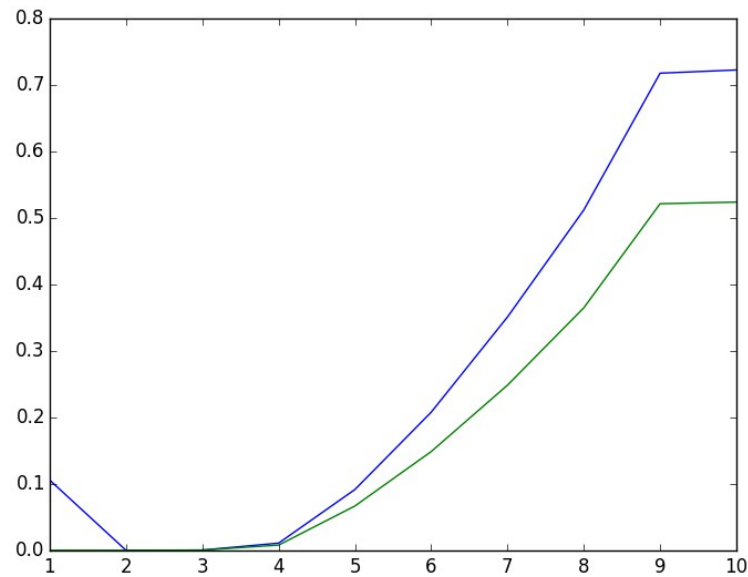
*Gráfica 6: Flotantes con tile 32x32*

### Algoritmo paralelo vs algoritmo paralelo con tiles:

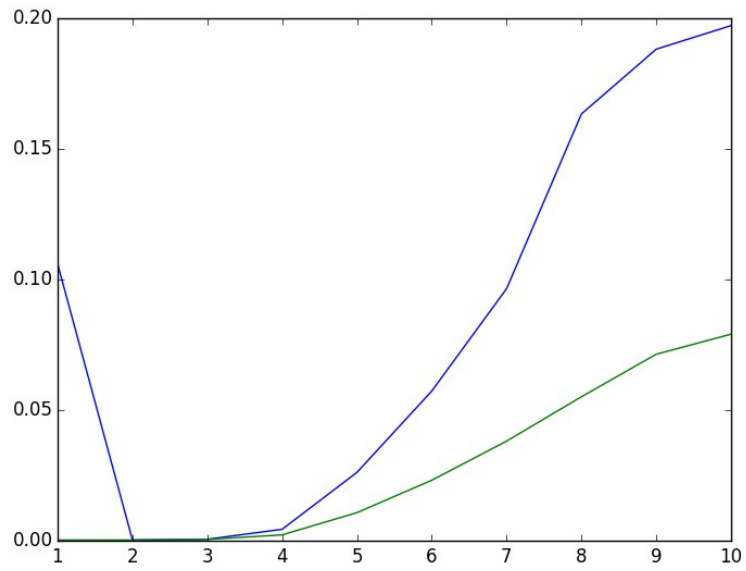
Se utilizo el color azul para el algoritmo paralelo y el verde para el algoritmo paralelo con tiles



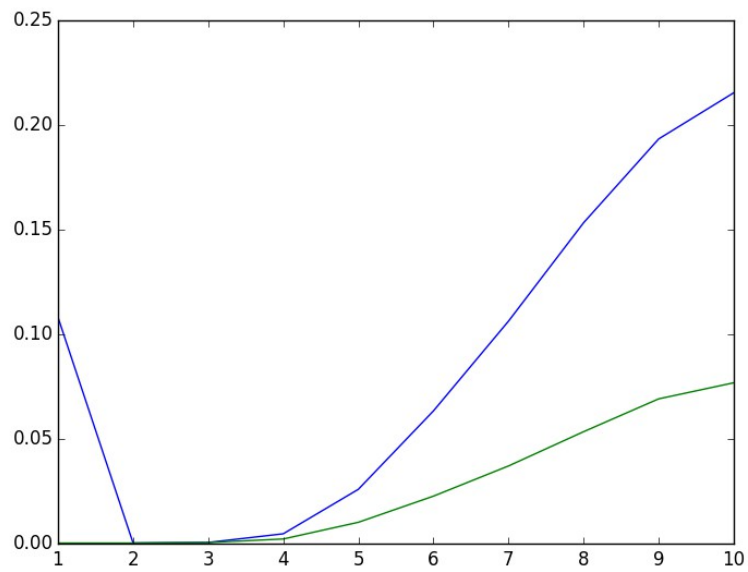
Gráfica 7: Enteros con tile 4x4 (solo paralelas)



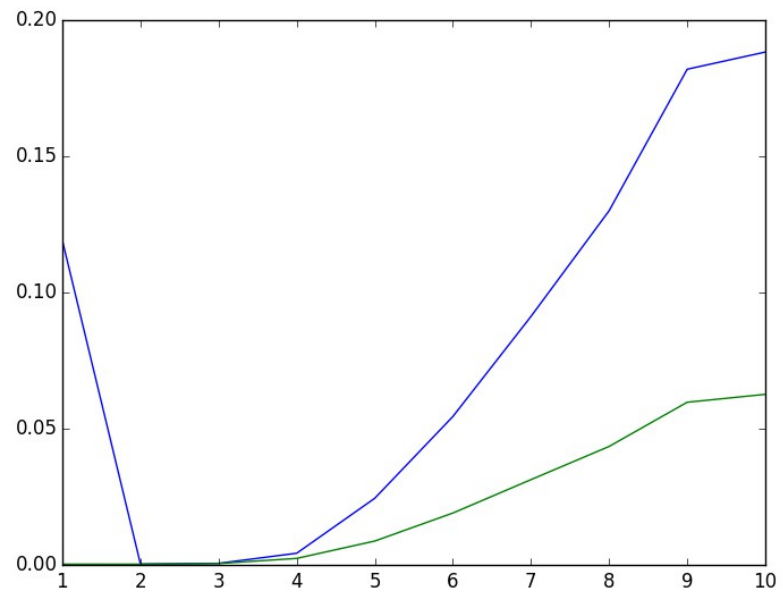
Gráfica 8: Flotantes con tile 4x4 (solo paralelas)



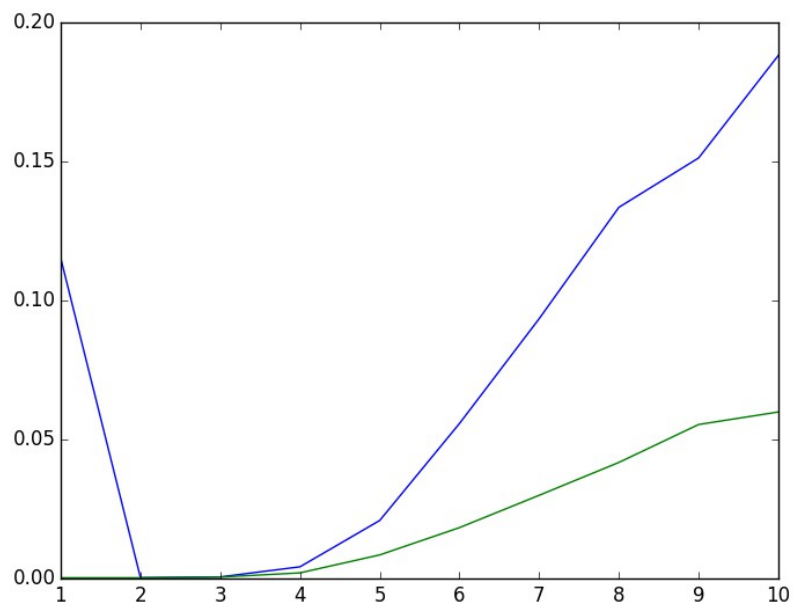
*Gráfica 9: Enteros con tile 16x16 (solo paralelas)*



*Gráfica 10: Flotantes con tile 16x16 (solo paralelas)*



Gráfica 11: Enteros con tile 32x32 (solo paralelas)

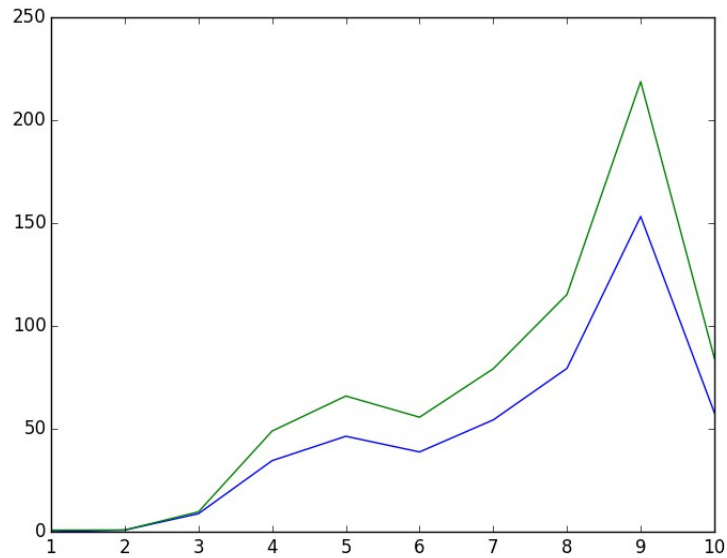


Gráfica 12: Flotantes con tile 32x32 (solo paralelas)

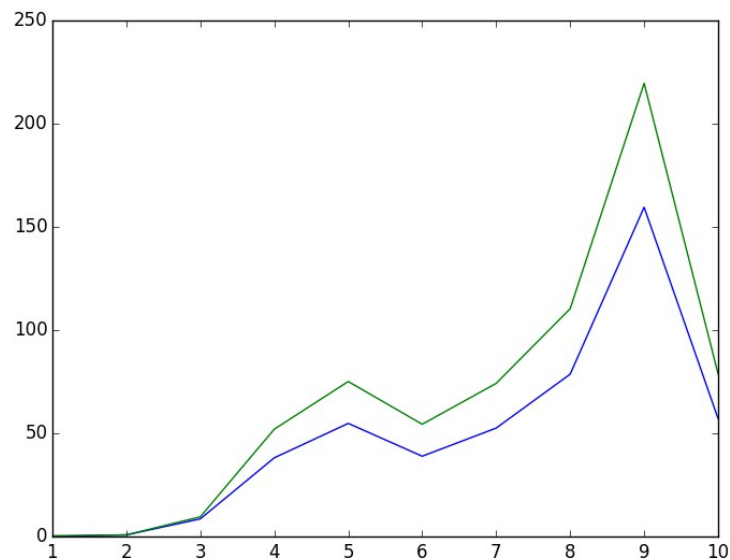


### Aceleración del algoritmo paralelo vs aceleración del algoritmo paralelo con tiles:

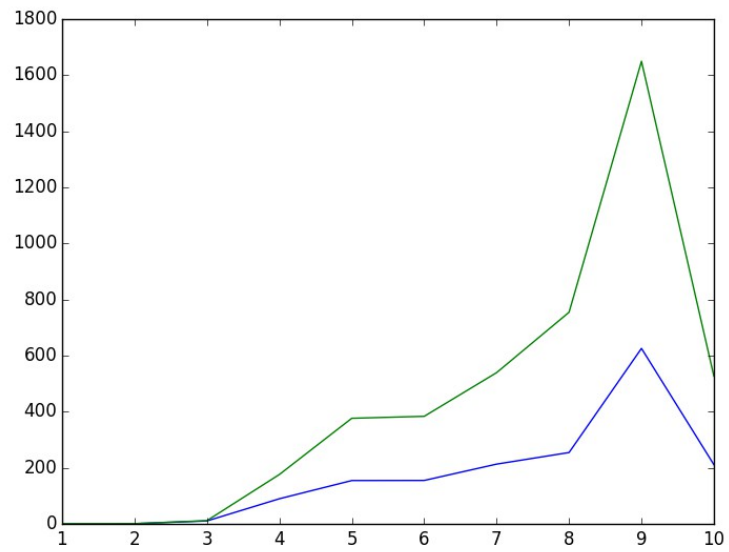
Se utilizo el color azul para la aceleración con el algoritmo paralelo y el color verde para el algoritmo paralelo con tiles



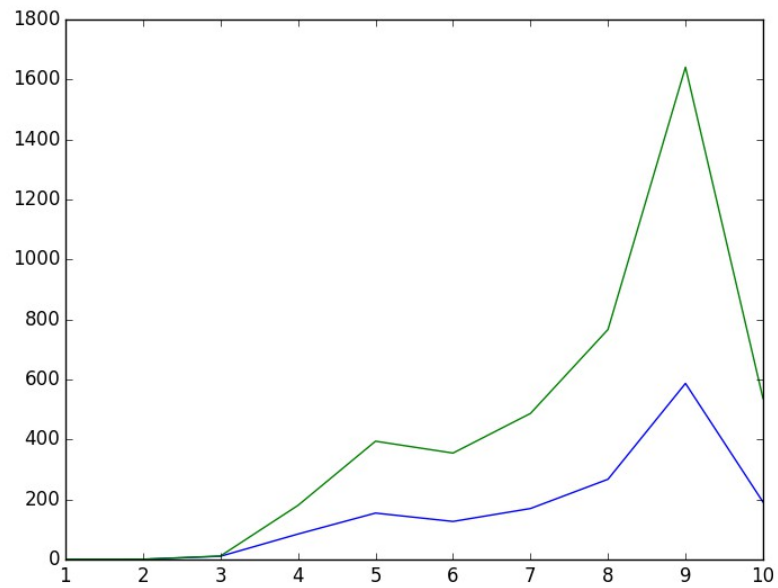
Gráfica 13: Enteros con tile 4x4 (aceleración)



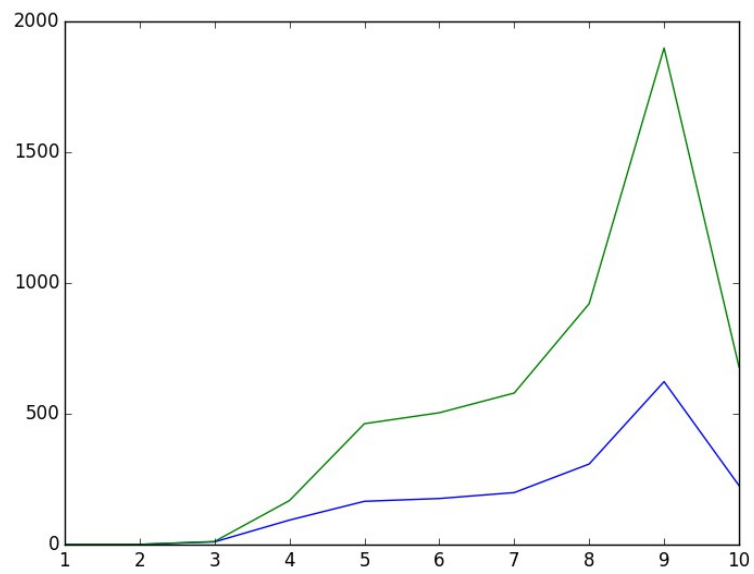
Gráfica 14: Flotantes con tile 4x4 (aceleración)



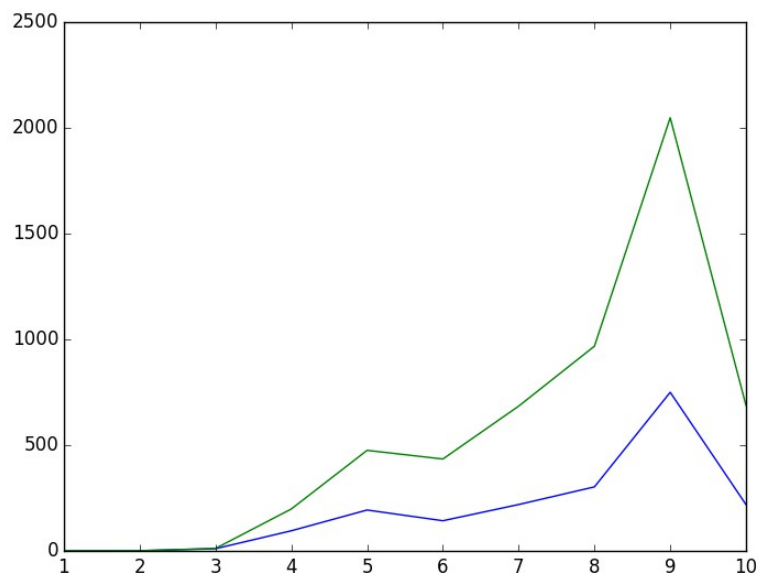
Gráfica 15: Enteros con tile 16x16 (aceleración)



Gráfica 16: Flotantes con tile 16x16 (aceleración)



Gráfica 17: Enteros con tile 32x32 (aceleración)



Gráfica 18: Flotantes con tile 32x32 (aceleración)

## Conclusiones

- Las transacciones de memoria entre el device y el host son muy lentas, lo cual no hace factible la aceleración en matrices pequeñas.
- Utilizar la memoria compartida genera muchos mejores tiempos que utilizar memoria global debido a que esta es mas rápida y esta más cerca de los threads, lo cual hace que el algoritmo paralelo con técnica de tiles tenga muchos mejores tiempos.
- Se deben realizar varias pruebas debido a que el tiempo puede variar debido al estrés de la máquina.
- Los tiempos tienden a estabilizarse con tamaños de matrices muy grandes debido al número de hilos del dispositivo, aunque la aceleración siempre sera exponencial.
- Se debe tener mucho cuidado cuando los tiles no son múltiplos del tamaño de la matriz debido a que esto puede utilizar basura de la memoria o ignorar partes sobrantes de la matriz.
- Los tiempos obtenidos de los algoritmos utilizando datos de tipo entero son mas rápidos que los utilizando datos de tipo float, debido a que las operaciones con estos últimos son mas complejas para las ALU.
- El algoritmo secuencial tiene comportamientos extraños con matrices cuadradas múltiplos de 32 ya que es mucho mas lento que matrices un poco mas grandes pero que no son cuadradas, lo cual genera picos extraños en los tiempos como se puede apreciar en las gráficas.