

Práctica 4y5: Suma de vectores en cuda

Autor: Antonio Guzmán Martín

En esta práctica se ha estudiado las capacidades de cálculo de la GPU y se ha comparado con las de la CPU. El problema escogido ha sido 0_Simple/vectorAdd. Es uno de los ejemplos para aprender CUDA que suministra el instalador de NVIDIA CUDA. Se han aplicado modificaciones que permiten

Sumar un conjunto de vectores cuyas componentes se leen desde un fichero raw.

Cambios en función vectorADD para añadir carga de cálculo y poder comparar la potencia de cálculo en igualdad de condiciones (me refiero a meter un cálculo suficientemente pesado para que sea más importante que el trasladar datos y recibirlos de la GPU).

Las características de la máquina son:

Device Number: 0

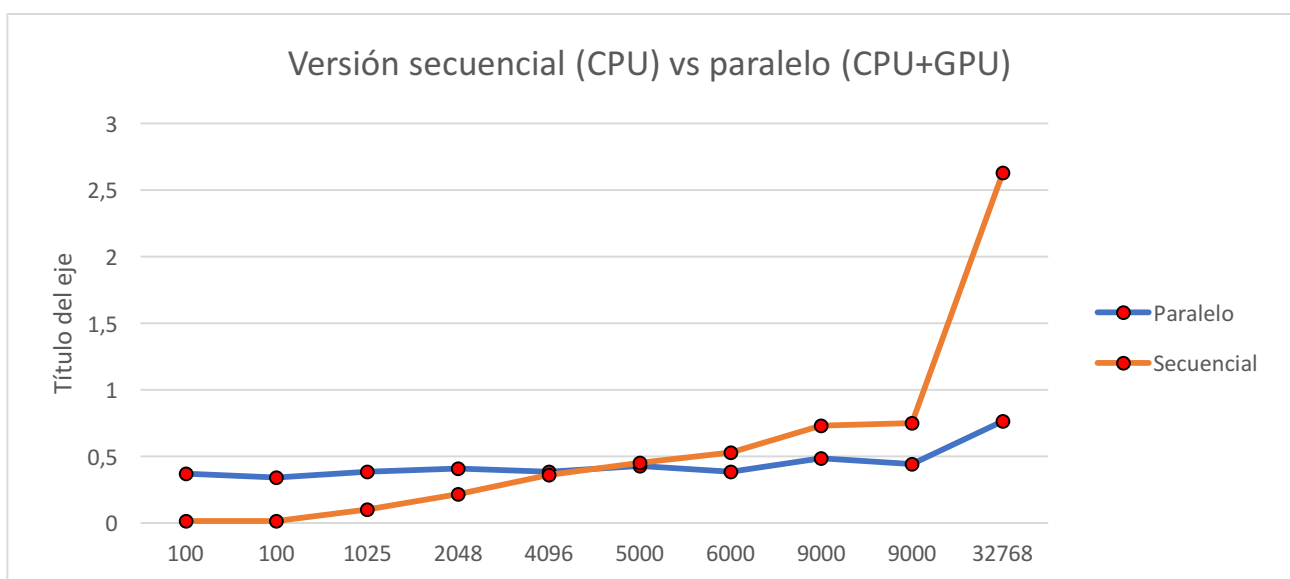
Device name: GeForce GTX 660 Ti

Memory Clock Rate (KHz): 3004000

Memory Bus Width (bits): 192

Peak Memory Bandwidth (GB/s): 144.192000

En esta gráfica podemos observar como se produce la evolución en los tiempos para la versión secuencial, usando solo la CPU y la paralela usando GPU+CPU.



Conclusiones

Para tamaños pequeños como 100. Estamos desperdiciando hebras que podrían hacer trabajo ya que

$\text{blocksPerGrid} = (100 + 255) / 256 = 1,38$ que aprox. son 2 bloques.

Teniendo 2 bloques de 256 hebras cada uno, ni siquiera alcanza para dar trabajo a todas las hebras del primer bloque. Esto sumado al hecho del tiempo gastado en hacer `cudaMemcpy` desde la cpu a la gpu en dos ocasiones empeora claramente los tiempos.

El tamaño de problema para el cual se equiparan tiempos es en 4096. A partir de este momento la versión paralela ofrecerá mejores tiempos que la versión secuencial.

Tipo	Tiempo (s)	Tamaño
secuencial	0,012046	100
paralelo	0,369271999	100
secuencial	0,012086	100
paralelo	0,339102	100
secuencial	0,100401	1025
paralelo	0,384577003	1025
secuencial	0,214396	2048
paralelo	0,407017003	2048
secuencial	0,357802	4096
paralelo	0,385093	4096
secuencial	0,452215	5000
paralelo	0,429086002	5000
secuencial	0,530567	6000
paralelo	0,385378999	6000
secuencial	0,732619	9000
paralelo	0,485984995	9000
secuencial	0,747334	9000
paralelo	0,443362995	9000
secuencial	2,631001	32768
paralelo	0,765615014	32768