

Ejercicios Tema 1

Ejercicio. Supongamos que tenemos un conjunto de hilos de 256 hilos y cada hilo utiliza 10 registros, ¿cuántos kernel idénticos al comentado puedo correr en el SM? ¿Y si tengo 11 registros?

a)

- Tenemos 256 hilos * 10 registros = 2560
- Si cada SM tiene 8192 registros -> $8192/2560 = 3$ kernel

b)

- Si tengo 11 registros por hilo -> 3120
- Si cada SM tiene 8192 registros -> $8192/3120 = 2$ kernel

Ejercicio 2: Si tenemos las siguientes matrices 8x8, 16x16, 24x24 y 32x32 y cada hilo consume 4 bytes de un SM, ¿cuántos bytes consume cada bloque? Indicar si es posible o no que corra en ese SM.

La memoria es de 16 K ¿?

- Matriz 8x8 = $8*8*4 = 256$ bytes -> entra en memoria y en cuanto hilos también porque se usan 64 hilos ($8*8$)
- Matriz 16x16 = $16*16*4 = 1024$ bytes -> entra en memoria y en cuanto a hilos también ($16*16 = 256$)
- Matriz 24x24 = $24*24*4 = 2304$ bytes -> entra en memoria y en cuanto a hilos también ($576 = 24*24$)
- Matriz 32x32 = $32*32*4 = 4096$ bytes -> cabe en memoria, pero no en cuanto a hilos porque se superan los 76 hilos que caben en un SM

Ejercicio 3: Suponer que tenemos tres programas que consumen 5K de memoria compartida. La cantidad de hilos que coge cada uno es de 256, ¿podría correr en un SM?, ¿y si fueran 7K de memoria?

- El primero si puede correr (15K frente a 16K disponibles) pero el segundo no, ya que se pasa de la memoria compartida (21K de memoria compartida y tenemos disponibles 16K)
- En cuanto a hilos si caben los dos programas ya que $256*3 = 768$ hilos

Ejercicio 4: Identifica las formas en que pueden ejecutarse en un SM las siguientes matrices de 8x8, 16x16 y 32x32.

- 8x8 = 64 -> puede correr en un SM porque no ocupa los 768 hilos por bloque. Lo más óptimo sería repartir 8 hilos en cada SP que forman el SM.
- 16x16 = 256 -> puede correr en un solo SM. Una forma óptima de repartir los hilos es asignar 32 hilos a cada uno de los 8 SP que forman el SM. Es más eficiente que asignar 1 SP con 256 hilos.

- $32 \times 32 = 1024 \rightarrow$ no lo puedo correr en un SM ya que supera el nº de hilos que hay en estos (768).

Ejercicio 5: Suponiendo 16 bloques y 64 hilos: dim3 dimGrid (,) y dim3 block(,)


- Dim3 dimGrid (4, 4) o (1, 16) o (2, 8)
- Dim3 block (1, 8, 8) o (2, 4, 8)

Ejercicio 6 ¿Cuanto es el coste temporal de una multiplicación float si en uno de los casos accedemos a memoria global y en otro a compartida?

- Float 7 = float x + float y \rightarrow una suma en global tiene 4 ciclos. La memoria global tiene entre 400 y 600 ciclos. Por tanto, la suma tiene un total de entre 404 y 604 ciclos máquina.
- Tengo que mover tres elementos de memoria global a compartida y esta operación cuesta 4 ciclos por elemento movido. Además, necesito una sincronización entre memoria global y compartida. El coste total que tengo es: 4 ciclos de operación + 3×4 ciclos de mover datos + 4 ciclos de sincronizar = 20 ciclos de coste total

Nota: la multiplicación ocupa 16 ciclos

Ejercicio 7: ¿Cuál de las siguientes matrices 8x8, 16x16, 24x24, 32x32 pueden correr en un SM?

- $8 \times 8 = 64 \rightarrow$ puede correr en un SM porque no ocupa los 768 hilos por bloque. Lo más óptimo sería repartir 8 hilos en cada uno de los 8 SP que forman el SM.
- $16 \times 16 = 256 \rightarrow$ puede correr en un solo SM. Una forma óptima de repartir los hilos es asignar 32 hilos a cada uno de los 8 SP que forman el SM. Es más eficiente que asignar 1 SP con 256 hilos. Usaremos tres bloques ($768/256 = 3$).
- $24 \times 24 = 576 \rightarrow$ puede correr en un solo SM. Podremos usar un solo bloque ($768/576 = 1$). 
- $32 \times 32 = 1024 \rightarrow$ no lo puedo correr en un SM ya que supera el nº de hilos que hay en estos (768).

Ejercicio 8 ¿Cuántos bloques de hilos puedo ejecutar en un SM si tengo matrices de 16x16 y consume cada uno 3K de memoria? ¿y si consumo 6K por bloque?

- En el primer caso consumo 3 K de memoria por bloque. Usaría 3 (ya que el máximo de hilos del SM es 768) bloques de memoria que me ocuparían 9 K, por tanto, entrarían en memoria. Podría ejecutar 3 matrices con este formato.
- En el segundo caso no entraría ya que si uso 3 bloques usaría 18 K de memoria, lo que supera la memoria compartida. Podría ejecutar 2 matrices con este formato.

Ejercicio 9: Supongamos que un Kernell tiene bloques de 256 hilos. Con cuatro instrucciones independientes de acceso a memoria global por hilo, y cada hilo usa 10 registros. ¿Cuántos bloques de hilos puedo ejecutar en un SM? ¿y si son 11 bloques?

a)

- Tenemos $256 \text{ hilos} \times 10 \text{ registros} = 2560$
- Si cada SM tiene 8192 registros $\rightarrow 8192/2560 = 3 \text{ kernel}$

b)

- Si tengo 11 registros por hilo -> 3120
- Si cada SM tiene 8192 registros -> $8192/3120 = 2$ kernel

DATOS GENERALES:

- 512 – SP
- 768 – SM
- 16K – Memoria compartida
- 8192 – Registros de 4 bit cada uno = 32 K de memoria
- 86.4 Gb/s
- 346 Gigaflops pico
- 367 pico Gigaflops

GTX 660

ANCHO DE BANDA

| GPU Engine Specs: | |
|---------------------------------|----------------------|
| CUDA Cores | 960 |
| Base Clock (MHz) | 980 |
| Boost Clock (MHz) | 1033 |
| Texture Fill Rate (billion/sec) | 78.4 |
| Memory Specs: | |
| Memory Clock | <u>6.0 Gbps</u> |
| Standard Memory Config | 2048 MB |
| Memory Interface | GDDR5 |
| Memory Interface Width | <u>192-bit</u> GDDR5 |
| Memory Bandwidth (GB/sec) | <u>144.2</u> |

$$\frac{6.0 \text{ Gbps} * 192 \text{ bits}}{8} = 144 \text{ GB/s}$$

GTX 280

ANCHO DE BANDA

| GPU Engine Specs | |
|-------------------------------------|----------------|
| CUDA Cores ¹ | 240 |
| Graphics Clock (MHz) | 602MHz |
| Processor Clock (MHz) | 1296MHz |
| Texture Fill Rate (billion/sec) | 48.2 |
| Graphics Performance | ?? |
| Memory Specs | |
| Memory Clock | <u>1107MHz</u> |
| Standard Memory Config | 1GB |
| Memory Interface Width ⁵ | <u>512-bit</u> |
| Memory Bandwidth (GB/sec) | <u>141.7</u> |

$$\frac{(2 * 1,107 \text{ GHz}) * 512 \text{ bits}}{8} = 141,7 \text{ GB/s}$$

*NOTA: El 2 viene de los flancos de subida y bajada. Se diferencian en que tenemos que pasar la frecuencia a GHz.

GTX 780

ANCHO DE BANDA

| GTX 780 GPU Engine Specs: | |
|---------------------------------|-----------------|
| CUDA Cores | 2304 |
| Base Clock (MHz) | 863 |
| Boost Clock (MHz) | 900 |
| Texture Fill Rate (billion/sec) | 160.5 |
| GTX 780 Memory Specs: | |
| Memory Speed | <u>6.0 Gbps</u> |
| Standard Memory Config | 3072 MB |
| Memory Interface | GDDR5 |
| Memory Interface Width | <u>384-bit</u> |
| Memory Bandwidth (GB/sec) | <u>288.4</u> |

$$\frac{6.0 * 384 \text{ bits}}{8} = 288 \text{ GB/s}$$

GTX 285

ANCHO DE BANDA

| GPU Engine Specs | |
|-------------------------------------|----------------|
| CUDA Cores ¹ | 240 |
| Graphics Clock (MHz) | 648MHz |
| Processor Clock (MHz) | 1476MHz |
| Texture Fill Rate (billion/sec) | 51.8 |
| Memory Specs | |
| Memory Clock | <u>1242MHz</u> |
| Standard Memory Config | 1 GB |
| Memory Interface | GDDR3 |
| Memory Interface Width ² | <u>512-bit</u> |
| Memory Bandwidth (GB/sec) | <u>159.0</u> |

$$\frac{(2 * 1,242 \text{ GHz}) * 512 \text{ bits}}{8} = 159 \text{ GB/s}$$

*NOTA: El 2 viene de los flancos de subida y bajada. Se diferencian en que tenemos que pasar la frecuencia a GHz.