Ejercicios Sesión 2
Hilos en CUDA

Albert García García <agarcia@dtic.ua.es>

Sergio Orts Escolano <sorts@dtic.ua.es>

José García Rodríguez <jgarcia@dtic.ua.es>

Universidad de Alicante

Departamento de tecnología informática y computación

Ejercicio 1

- Un estudiante ha mencionado que es capaz de multiplicar dos matrices de 1024x1024 utilizando un código basado en tiling y utilizando 1024 hilos por bloque en la arquitectura G80. Además ha mencionado que cada hilo es capaz de calcular un elemento de la matriz resultante. ¿Cuál sería tu reacción y por qué?

Ejercicio 1. Solución

La primera arquitectura G80 presenta la limitación de solo poder computar 512 hilos por bloque como máximo. Un ejemplo de ello es el modelo Gforce 8600

```
There is 1 device supporting CUDA

Device 0: "GeForce 8600 GTS"
  Major revision number:                         1
  Minor revision number:                         1
  Total amount of global memory:                 268173312 bytes
  Total amount of constant memory:               65536 bytes
  Total amount of shared memory per block:       16384 bytes
  Total number of registers available per block: 8192
  Warp size:                                     32
  Maximum number of threads per block:           512
  Maximum sizes of each dimension of a block:    512 x 512 x 64
  Maximum sizes of each dimension of a grid:     65535 x 65535 x 1
  Maximum memory pitch:                          262144 bytes
  Texture alignment:                             256 bytes
  Clock rate:                                    1458000 kilohertz

Test PASSED
```

Ejercicio 2

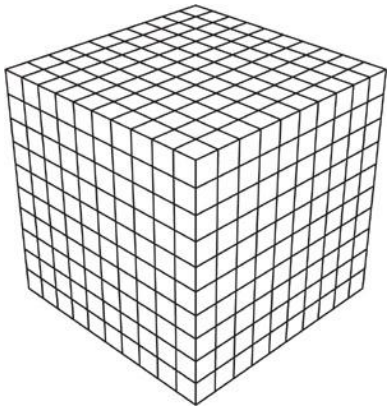
- Relacionado con el ejercicio 1, para la multiplicación de matrices utilizando tiling deberíamos utilizar bloques de hilos de tamaño 8x8, 16x16, o 32x32 suponiendo que tenemos disponible una tarjeta con arquitectura GT200 ?

Ejercicio 2. Solución

- Para resolver esta pregunta debemos analizar las distintas opciones disponibles y :
  - 8x8 → Cada bloque estaría compuesto de 64 hilos y dado que el número máximo de hilos por SM es de 1024, tendríamos 12 bloques para ejecutar sobre el SM. Dado que el número máximo de bloques por SM es de 8 solo vamos a poder ejecutar 64 x 8 = 512 hilos. Esto significa que los recursos de un SM van a ser utilizados por debajo de sus posibilidades.
  - 16x16 → Cada bloque estaría compuesto de 256 hilos. Esto significa que cada SM puede ejecutar 1024/256 = 4 bloques. Estamos por debajo de la limitación de los 8 bloques. De esta forma no solo conseguimos ocupar todos los hilos de ejecución sino también conseguir el número máximo de warps.
  - 32x32 → Para esta arquitectura excedería el tamaño máximo de hilos por bloque. En caso de que se pudiese. Tendríamos un único bloque que consigue total ocupación respecto al número de hilos que puede ejecutar un SM.

Ejercicio 3

- Imagina que subdivimos el espacio 3D en voxels. Queremos calcular de forma paralela una primitiva sobre cada uno de los vóxels que componen el espacio 3D. ¿Cómo organizarías los hilos para llevar a cabo la ejecución? ¿Por qué? Si guardamos el resultado de cada uno de los hilos en un vector unidimensional, ¿Cómo calcularías el índice global?



Cada hilo computará un cálculo en cada uno de los vóxeles que componen el grid 3D.

Ejercicio 3. Solución

- La ejecución se llevaría a cabo organizando los hilos como un grid 3D aprovechándose de las 3 dimensiones que nos ofrece CUDA para organizar la ejecución de los hilos.

- `x = threadIdx.x + blockIdx.x * blockDim.x;`
- `y = threadIdx.y + blockIdx.y * blockDim.y;`
- `z = threadIdx.z + blockIdx.z * blockDim.z;`

-
```
global_index = z * blockDim.x * gridDim.x * blockDim.y * gridDim.y +
               y * blockDim.y * gridDim.y +
               x;
```

¿Preguntas?