

Alejandra Guzmán Domínguez 20262
Jorge Caballeros Pérez 20009
Eduardo Ramírez 19



Universidad del Valle de Guatemala	Programación Paralela y Distribuida
Hoja de Trabajo 3	10/11/2023

2. hello.cu (50 pts)

El programa hello.cu ilustra la forma básica del modelo de ejecución para CUDA. Realice las siguientes acciones para comprender el efecto de la configuración del kernel y su relación con el Compute Capability de una tarjeta.

1. Compile el programa (ignore la advertencia sobre código deprecado en caso le salga):

```
$ nvcc hello.cu -o hello
```

2. Ejecute el programa. Observe cuántas veces se imprime el mensaje y su conexión con la configuración de la llamada al kernel – hello<<<g,b>>>():

```
$ ./hello
```

3. Modifique el programa para correr 2 bloques de 1024 hilos. Modificarlo también para que imprima su nombre y carnet. Busque en el despliegue de consola el mensaje del último hilo de la serie (1023).

(10 PTS) CAPTURA DE PANTALLA DE LA EJECUCIÓN CON 2 BLOQUES DE 1024 HILOS

```
jorge@UNKNOWN: ~/hdt3
y Eduardo Ramirez
Hello, World! I'm thread 108. Nuestro nombre es Jorge Caballeros, Ale Guzman
y Eduardo Ramirez
Hello, World! I'm thread 109. Nuestro nombre es Jorge Caballeros, Ale Guzman
y Eduardo Ramirez
Hello, World! I'm thread 110. Nuestro nombre es Jorge Caballeros, Ale Guzman
y Eduardo Ramirez
Hello, World! I'm thread 111. Nuestro nombre es Jorge Caballeros, Ale Guzman
y Eduardo Ramirez
Hello, World! I'm thread 112. Nuestro nombre es Jorge Caballeros, Ale Guzman
y Eduardo Ramirez
Hello, World! I'm thread 113. Nuestro nombre es Jorge Caballeros, Ale Guzman
y Eduardo Ramirez
Hello, World! I'm thread 114. Nuestro nombre es Jorge Caballeros, Ale Guzman
y Eduardo Ramirez
Hello, World! I'm thread 115. Nuestro nombre es Jorge Caballeros, Ale Guzman
y Eduardo Ramirez
Hello, World! I'm thread 116. Nuestro nombre es Jorge Caballeros, Ale Guzman
y Eduardo Ramirez
Hello, World! I'm thread 117. Nuestro nombre es Jorge Caballeros, Ale Guzman
y Eduardo Ramirez
Hello, World! I'm thread 118. Nuestro nombre es Jorge Caballeros, Ale Guzman
y Eduardo Ramirez
Hello, World! I'm thread 119. Nuestro nombre es Jorge Caballeros, Ale Guzman
y Eduardo Ramirez
Hello, World! I'm thread 120. Nuestro nombre es Jorge Caballeros, Ale Guzman
y Eduardo Ramirez
Hello, World! I'm thread 121. Nuestro nombre es Jorge Caballeros, Ale Guzman
y Eduardo Ramirez
Hello, World! I'm thread 122. Nuestro nombre es Jorge Caballeros, Ale Guzman
y Eduardo Ramirez
Hello, World! I'm thread 123. Nuestro nombre es Jorge Caballeros, Ale Guzman
y Eduardo Ramirez
Hello, World! I'm thread 124. Nuestro nombre es Jorge Caballeros, Ale Guzman
y Eduardo Ramirez
Hello, World! I'm thread 125. Nuestro nombre es Jorge Caballeros, Ale Guzman
y Eduardo Ramirez
Hello, World! I'm thread 126. Nuestro nombre es Jorge Caballeros, Ale Guzman
y Eduardo Ramirez
Hello, World! I'm thread 127. Nuestro nombre es Jorge Caballeros, Ale Guzman
y Eduardo Ramirez
jorge@UNKNOWN:~/hdt3$
```

output muy largo como para mostrar completo

4. Busque en el sitio de Nvidia el Compute Capability de la tarjeta que poseen las máquinas del Laboratorio (o de su computador, en caso tenga tarjeta NVIDIA y estén haciendo todo esto en su compu). Escriba acá el valor de CC y busque la tabla resumen con las características del CC:

(10 PTS) Compute Capability:

5. Modifique el programa para correr 1 bloque de 2048 hilos. Coloque una captura de



Alejandra Guzmán Domínguez 20262

Jorge Caballeros Perez 20009

Eduardo Ramirez 19

pantalla de la salida y busque en la tabla de CC el siguiente dato:

(10 PTS) Maximum number of threads per block:

(10 PTS) CAPTURA DE PANTALLA DE LA EJECUCIÓN CON 1 BLOQUE DE 2048 HILOS

```
jorge@UNKNOWN: ~/hdt3
jorge@UNKNOWN:~/hdt3$ ./hello2
Hello world from the thread with the maximum global ID: 99999 (global ID: 39 137439)
jorge@UNKNOWN:~/hdt3$ |
```

(10 PTS) EXPLIQUE EN POCAS PALABRAS EL RESULTADO

3. *hello2.cu* (50 pts)

El programa *hello2.cu* ilustra la forma para calcular un identificador global al momento de usar hilos que pertenecen a bloques diferentes. Realice las siguientes acciones para comprender el efecto de la configuración del kernel y su relación con la forma de calcular el ID único de los hilos.

1. Descargue, compile y ejecute *hello2.cu*. Observe la relación de la configuración de la llamada al kernel con la geometría de los hilos y el resultado. Escriba la respuesta a los dos enunciados:

i. Máximo ID de los hilos:

ii. Ejecución de los hilos en orden:

2. Observe que la fórmula genérica para cálculo del ID global está en los comentarios. Modifique el programa para que imprima también su nombre y carnet. Luego, realice la siguiente modificación al programa (al inicio del main) y use la fórmula genérica para derivar el nuevo cálculo de ID:

dim3 g (4,2);

dim3 b (32,16);

hello <<<g, b>>>();

(10 PTS) FÓRMULA PARA CALCULAR EL ID GLOBAL Y SALIDA DE PANTALLA

(10 PTS) CAPTURA DE PANTALLA DE LA NUEVA CONFIGURACIÓN (buscar el mensaje impreso por el hilo con el máximo ID global)

3. Revise nuevamente la información del Compute Capability respecto a las dimensiones máximas de hilos-bloque en x, y, & z para una grilla. Cree una configuración para lanzar



Alejandra Guzmán Domínguez 20262

Jorge Caballeros Perez 20009

Eduardo Ramirez 19

exitosamente el kernel para procesar 100,000 datos. (Sugerencia: busque una configuración que lance como mínimo 100,000 hilos. Modifique el kernel para que imprima el mensaje únicamente si es el ID global máximo)

(10 PTS) MOSTRAR SU CONFIGURACIÓN USADA

```
int main()
{
    int totalThreads = 100000; // Número total de hilos
    int threadsPerBlock = 256; // Número de hilos por bloque
    int numBlocks = (totalThreads + threadsPerBlock - 1) / threadsPerBlock; // Asegúrese de lanzar suficientes bloques

    // llama al kernel con la configuración de bloques y hilos
    hello<<<numBlocks, threadsPerBlock>>>(totalThreads);
    cudaDeviceSynchronize(); // Espera a que finalicen todos los hilos

    return 0;
}
```

(10 PTS) FÓRMULA PARA CALCULAR EL ID GLOBAL Y SALIDA DE PANTALLA

```
__global__ void hello(int totalThreads)
{
    int myID = blockIdx.x * blockDim.x + threadIdx.x;
    int globalID = myID + blockIdx.x * blockDim.x * gridDim.x;

    if (myID < totalThreads)
    {
        if (myID == totalThreads - 1)
        {
            printf("Hello world from the thread with the maximum global ID: %i (global ID: %i)\n", myID, globalID);
        }
    }
}
```

Alejandra Guzmán Domínguez 20262

Jorge Caballeros Perez 20009

Eduardo Ramirez 19



(10 PTS) CAPTURA DE PANTALLA CORRIENDO EL CODIGO CON LA NUEVA CONFIGURACIÓN (con el mensaje impreso por el hilo con el máximo ID global)

The screenshot shows a Visual Studio Code editor with a file named `hello2.cu` open. The code is written in CUDA C++ and includes `<stdio.h>` and `<cuda.h>`. It defines a `hello` function that takes `totalThreads` as an argument. Inside the function, it calculates `myID` and `globalID` for each thread. A conditional statement checks if `myID` is less than `totalThreads`, and if so, it prints a message. The `main` function sets `totalThreads` to 100000, `threadsPerBlock` to 256, and calculates `numBlocks`. It then calls the `hello` kernel with these parameters.

```
1  #include <stdio.h>
2  #include <cuda.h>
3
4  __global__ void hello(int totalThreads)
5  {
6      int myID = blockIdx.x * blockDim.x + threadIdx.x;
7      int globalID = myID + blockIdx.x * blockDim.x * gridDim.x;
8
9      if (myID < totalThreads)
10     {
11         if (myID == totalThreads - 1)
12         {
13             printf("Hello world from the thread with the maximum global ID: %d (global ID: %d)\n", myID, globalID);
14         }
15     }
16 }
17
18 int main()
19 {
20     int totalThreads = 100000; // Número total de hilos
21     int threadsPerBlock = 256; // Número de hilos por bloque
22     int numBlocks = (totalThreads + threadsPerBlock - 1) / threadsPerBlock;
23
24     // Llama al kernel con la configuración de bloques y hilos
25     hello<<numBlocks, threadsPerBlock>>>(totalThreads);
26 }
```

The terminal window below the editor shows the command `./hello2` being executed, resulting in the output: `Hello world from the thread with the maximum global ID: 99999 (global ID: 39)`.