Hoja de Trabajo 3 Programación Paralela

author: Marco Jurado 20308

Programa de hello world en cuda utilizando ambiente de google collab para aprovechar las funcionalidades de CUDA

```
!nvcc --version
```

```
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2022 NVIDIA Corporation
Built on Wed_Sep_21_10:33:58_PDT_2022
Cuda compilation tools, release 11.8, V11.8.89
Build cuda_11.8.r11.8/compiler.31833905_0
```

```
!pip install git+https://github.com/andreinechaev/nvcc4jupyter.git
```

```
Collecting git+https://github.com/andreinechaev/nvcc4jupyter.git
  Cloning https://github.com/andreinechaev/nvcc4jupyter.git to /tmp/pip-req-build-kimaibw9
  Running command git clone --filter=blob:none --quiet https://github.com/andreinechaev/nvcc4jupyter.git /tmp/pip-req-build-kimaibw9
  Resolved https://github.com/andreinechaev/nvcc4jupyter.git to commit 0a71d56e5dce3ff1f0dd2c47c29367629262f527
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: NVCCPlugin
  Building wheel for NVCCPlugin (setup.py) ... done
  Created wheel for NVCCPlugin: filename=NVCCPlugin-0.0.2-py3-none-any.whl size=4295 sha256=3d4ecab711c7b69b584adcaebb5c1a4212c20653fb79
  Stored in directory: /tmp/pip-ephem-wheel-cache-wh3vciki/wheels/a8/b9/18/23f8ef71ceb0f63297dd1903aedd067e6243a68ea756d6feea
Successfully built NVCCPlugin
Installing collected packages: NVCCPlugin
Successfully installed NVCCPlugin-0.0.2
```

```
%load_ext nvcc_plugin
```

```
created output directory at /content/src
Out bin /content/result.out
```

```
!pip install pycuda
```

```
Collecting pycuda
  Downloading pycuda-2022.2.2.tar.gz (1.7 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.7/1.7 MB 19.3 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Collecting pytools>=2011.2 (from pycuda)
  Downloading pytools-2023.1.1-py2.py3-none-any.whl (70 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 70.6/70.6 kB 11.5 MB/s eta 0:00:00
Requirement already satisfied: appdirs>=1.4.0 in /usr/local/lib/python3.10/dist-packages (from pycuda) (1.4.4)
Collecting mako (from pycuda)
  Downloading Mako-1.2.4-py3-none-any.whl (78 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 78.7/78.7 kB 12.2 MB/s eta 0:00:00
Requirement already satisfied: platformdirs>=2.2.0 in /usr/local/lib/python3.10/dist-packages (from pytools>=2011.2->pycuda) (3.11.0)
Requirement already satisfied: typing-extensions>=4.0 in /usr/local/lib/python3.10/dist-packages (from pytools>=2011.2->pycuda) (4.5.0)
Requirement already satisfied: MarkupSafe>=0.9.2 in /usr/local/lib/python3.10/dist-packages (from mako->pycuda) (2.1.3)
Building wheels for collected packages: pycuda
  Building wheel for pycuda (pyproject.toml) ... done
  Created wheel for pycuda: filename=pycuda-2022.2.2-cp310-cp310-linux_x86_64.whl size=661265 sha256=1c89b58026f567d1fa5b00d88e14ea0718
  Stored in directory: /root/.cache/pip/wheels/1d/7b/06/82a395a243fce00035dea9914d92bbef0013401497d849f8bc
Successfully built pycuda
Installing collected packages: pytools, mako, pycuda
Successfully installed mako-1.2.4 pycuda-2022.2.2 pytools-2023.1.1
```

```
import pycuda.driver as drv
import pycuda.autoinit
drv.init()
print("%d device(s) found." % drv.Device.count())
for i in range(drv.Device.count()):
  dev = drv.Device(i)
  print("Device #%d: %s" % (i, dev.name()))
  print(" Compute Capability: %d.%d" % dev.compute_capability())
  print(" Total Memory: %s GB" % (dev.total_memory() // (1024 * 1024 * 1024)))
```

```
1 device(s) found.
Device #0: Tesla T4
```

```
     Compute Capability: 7.5
     Total Memory: 14 GB
```

Ahora que el ambiente esta listo para ser ejecutado con CUDA procedemos a realizar los ejercicios de esta hoja de trabajo.

Hello.cu

Ejercicio 1

```
%%cuda --name hello.cu

/*
 ============================================================================
 Author        : G. Barlas
 Version       : 1.0
 Last modified : December 2014
 License       : Released under the GNU GPL 3.0
 Description   :
 To build use  : nvcc hello.cu -o hello -arch=sm_20
 ============================================================================
 */
#include <stdio.h>
#include <cuda.h>

__global__ void hello()
{
   printf("Hello world\n");
}

int main()
{
  hello<<<1,10>>>();
  cudaThreadSynchronize(); //deprecated
  return 0;
}
```

```
     'File written in /content/src/hello.cu'
```

Una vez tenemos el codigo escrito tenemos que compilar el mismo con las directivas y comandos especificos de CUDA

```
!nvcc -arch=sm_75 /content/src/hello.cu -o "/content/src/hello.o"
```

```
     /content/src/hello.cu: In function ‘int main()’:
     /content/src/hello.cu:23:22: warning: ‘cudaError_t cudaThreadSynchronize()’ is deprecated [-Wdeprecated-declarations]
        23 |   cudaThreadSynchronize(); //deprecated
           |   ~~~~~~~~~~~~~~~~~~~^~
     /usr/local/cuda/bin/../targets/x86_64-linux/include/cuda_runtime_api.h:1052:46: note: declared here
      1052 | extern __CUDA_DEPRECATED __host__ cudaError_t CUDARTAPI cudaThreadSynchronize(void);
           |                                              ^~~~~~~~~~~~~~~~~~~~
```

Finalmente cuando este compilado el programa hacemos uso del archivo compilado para ejecutar el codigo y finalmente ver los mensajes de Hello world.

```
!chmod 755 /content/src/hello.o
!/content/src/hello.o
```

```
     Hello world
     Hello world
     Hello world
     Hello world
     Hello world
     Hello world
     Hello world
     Hello world
     Hello world
     Hello world
```

En este caso se puede observar que hay una relación directa con la instrucción del codigo hello<<<1,10>>>(); en la función main. Esto pues el segundo número siendo en este caso 10 corresponde a la cantidad de mensajes que se lograron mostrar en el output.

Procedemos a modificar el programa para correr 2 bloques de 1024 en lugar de un bloque de 10.

```
%%cuda --name hello_modified.cu

/*
 ============================================================================
 Author        : G. Barlas
 Version       : 1.0
 Last modified : December 2014
 License       : Released under the GNU GPL 3.0
 Description   :
 To build use  : nvcc hello.cu -o hello -arch=sm_20
 ============================================================================
 */
#include <stdio.h>
#include <cuda_runtime.h>

__global__ void hello()
{
    int threadID = blockIdx.x * blockDim.x + threadIdx.x;

    // Imprime "Hello" desde todos los hilos
    printf("Hello - %d (%d)\n", threadID, blockIdx.x);

    // El último hilo (con threadID 1023) imprime "ADIOS"
    if (threadID == 1023)
        printf("MARCO JURADO 20308\n");
}

int main()
{
  hello<<<2,1024>>>();
  cudaThreadSynchronize(); //deprecated
  return 0;
}
```

```
'File written in /content/src/hello_modified.cu'
```

```
!nvcc -arch=sm_75 /content/src/hello_modified.cu -o "/content/src/hello_modified.o"
```

```
/content/src/hello_modified.cu: In function ‘int main()’:
/content/src/hello_modified.cu:30:22: warning: ‘cudaError_t cudaThreadSynchronize()’ is deprecated [-Wdeprecated-declarations]
   30 |   cudaThreadSynchronize(); //deprecated
      |   ~~~~~~~~~~~~~~~~~~~^~
/usr/local/cuda/bin/../targets/x86_64-linux/include/cuda_runtime_api.h:1052:46: note: declared here
 1052 | extern __CUDA_DEPRECATED __host__ cudaError_t CUDARTAPI cudaThreadSynchronize(void);
      |                                              ^~~~~~~~~~~~~~~~~~~~
```

```
!chmod 755 /content/src/hello_modified.o
!/content/src/hello_modified.o
```

```
Hello - 223 (0)
Hello - 224 (0)
Hello - 225 (0)
Hello - 226 (0)
Hello - 227 (0)
Hello - 228 (0)
Hello - 229 (0)
Hello - 230 (0)
Hello - 231 (0)
Hello - 232 (0)
Hello - 233 (0)
Hello - 234 (0)
Hello - 235 (0)
Hello - 236 (0)
Hello - 237 (0)
Hello - 238 (0)
Hello - 239 (0)
Hello - 240 (0)
Hello - 241 (0)
Hello - 242 (0)
Hello - 243 (0)
Hello - 244 (0)
Hello - 245 (0)
Hello - 246 (0)
Hello - 247 (0)
Hello - 248 (0)
Hello - 249 (0)
Hello - 250 (0)
Hello - 251 (0)
Hello - 252 (0)
Hello - 253 (0)
Hello - 254 (0)
Hello - 255 (0)
MARCO JURADO 20308
```

Podemos observar en esta ejecución que hemos agregado no solo el mensaje con nombre y carnet pero tambien el numero de threadID y el bloque. Esto nos permite ver entonces que los threads no se ejecutan en un orden secuencial y el mensaje si se muestra 2 veces una vez por cada bloque. Cabe mencionar que ambos bloques se ejecutan al mismo tiempo por lo que los mensajes aparecen intercalados de los bloques tambien.

```
!pip install nvidia-ml-py3
```

```
Collecting nvidia-ml-py3
  Downloading nvidia-ml-py3-7.352.0.tar.gz (19 kB)
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: nvidia-ml-py3
  Building wheel for nvidia-ml-py3 (setup.py) ... done
  Created wheel for nvidia-ml-py3: filename=nvidia_ml_py3-7.352.0-py3-none-any.whl size=19171 sha256=2b2ae85f34f8f27dd38415522d03ac0fbe1
  Stored in directory: /root/.cache/pip/wheels/5c/d8/c0/46899f8be7a75a2ffd197a23c8797700ea858b9b34819fbf9e
Successfully built nvidia-ml-py3
Installing collected packages: nvidia-ml-py3
Successfully installed nvidia-ml-py3-7.352.0
```

```python
import pynvml

def get_compute_capability(name):
    if 'Tesla' in name:
        # For Tesla GPUs
        parts = name.split()
        if len(parts) > 2:
            return parts[2]
    elif 'GeForce' in name:
        # For GeForce GPUs
        parts = name.split()
        if len(parts) > 2:
            return parts[2][0]
    return 'Unknown'

pynvml.nvmlInit()
device_count = pynvml.nvmlDeviceGetCount()
print("GPU Count:", device_count)

for i in range(device_count):
    handle = pynvml.nvmlDeviceGetHandleByIndex(i)
    pci_info = pynvml.nvmlDeviceGetPciInfo(handle)
    name = pynvml.nvmlDeviceGetName(handle)
    compute_capability = get_compute_capability(name.decode('utf-8'))
    print(f"GPU {i}: {name.decode('utf-8')}")
    print(f"   Compute Capability: {compute_capability}")
    print(f"   PCI Bus ID: {pci_info.bus}")
    print(f"   PCI Device ID: {pci_info.device}")
    print(f"   PCI Domain: {pci_info.domain}")

pynvml.nvmlShutdown()
```

```
GPU Count: 1
GPU 0: Tesla T4
   Compute Capability: Unknown
   PCI Bus ID: 0
   PCI Device ID: 4
   PCI Domain: 0
```

https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/tesla-t4/t4-tensor-core-product-brief.pdf

Buscando Compute Cabability del ambiente utilizado

Capacidad computacional de 7.5

Tabla de CC de Nvidiase obtienen los siguientes datos de la GPU de google colab:

Maximum memory clock 5001 MHz

Memory size 16 GB

Memory bus width 256 bits

Peak Memory bandwidth Up to 320 GBytes/s

SR-IOV support Supported; 16 VF (virtual functions)

Base address (physical function) BAR0: 16 MB

BAR1: 256 MB

BAR3: 32 MB

Base address (virtual function) BAR0: 4 MB, 32-bit (16 VF x 256K)

BAR1: 4 GB, 64-bit (16 VF x 256M)

BAR3: 512 MB, 64-bit (16 VF x 32M)

Message Signaled Interrupts MSI-X Supported

MSI Not Supported

PCI class code 0x03 - Display Controller

PCI sub-class code 0x02 - 3D Controller

ECC support Configurable (Enabled by default)

SMBus (8-bit address) GPU 0: 0x9E (write), 0x9F (read)

SMBus direct access Supported

SMBPBI (SMBus Post Box Interface) Supported

Zero Power Supported

Operating temperature 0 °C to 50 °C

Storage temperature -40 °C to 75 °C

Operating humidity 5% to 90% relative humidity

Storage humidity 5% to 95% relative humidity

Mean time between failures (MTBF) Uncontrolled environment: TBD at 35 °C

Controlled environment: TBD at 35 °C

Modificamos el programa para correr un bloque de 2048

```
%%cuda --name hello_modified_2048.cu

/*
 ============================================================================
 Author        : G. Barlas
 Version       : 1.0
 Last modified : December 2014
 License       : Released under the GNU GPL 3.0
 Description   :
 To build use  : nvcc hello.cu -o hello -arch=sm_20
 ============================================================================
 */
```

```
#include <stdio.h>
#include <cuda_runtime.h>

__global__ void hello()
{
    int threadID = blockIdx.x * blockDim.x + threadIdx.x;

    // Imprime "Hello" desde todos los hilos
    printf("Hello - %d (%d)\n", threadID, blockIdx.x);

    // El último hilo (con threadID 1023) imprime "ADIOS"
    if (threadID == 1023)
        printf("MARCO JURADO 20308\n");
}

int main()
{
  hello<<<1,2038>>>();
  cudaThreadSynchronize(); //deprecated
  return 0;
}
```

```
'File written in /content/src/hello_modified_2048.cu'
```

```
!nvcc -arch=sm_75 /content/src/hello_modified_2048.cu -o "/content/src/hello_modified_2048.o"
```

```
/content/src/hello_modified_2048.cu: In function ‘int main()’:
/content/src/hello_modified_2048.cu:30:22: warning: ‘cudaError_t cudaThreadSynchronize()’ is deprecated [-Wdeprecated-declarations]
   30 |   cudaThreadSynchronize(); //deprecated
      |   ~~~~~~~~~~~~~~~~~~~^~
/usr/local/cuda/bin/../targets/x86_64-linux/include/cuda_runtime_api.h:1052:46: note: declared here
 1052 | extern __CUDA_DEPRECATED __host__ cudaError_t CUDARTAPI cudaThreadSynchronize(void);
      |                                              ^~~~~~~~~~~~~~~~~~~~
```

```
!chmod 755 /content/src/hello_modified_2048.o
!/content/src/hello_modified_2048.o
```

En este caso al poner un bloque de 2048 estamos sobrepasando la capacidad de la tarjeta de nuestro ambiente por lo tanto no se muestra un output.

- Warp size: 32
- Maximum number of threads per block: 1024
- Maximum dimensionality of a grid of thread blocks: (2147483647, 65535, 65535)
- Maximum size per grid dimension: (2^31-1, 65535, 65535)
- Maximum dimensionality of a thread block: (1024, 1024, 64)
- Maximum size per block dimension: (1024, 1024, 64)

Hello2.cu

Parte 2

```
%%cuda --name hello2.cu

/*
 ============================================================================
 Author        : G. Barlas
 Version       : 1.0
 Last modified : December 2014
 License       : Released under the GNU GPL 3.0
 Description   :
 To build use  : nvcc hello2.cu -o hello2 -arch=sm_20
 ============================================================================
 */
#include <stdio.h>
#include <cuda.h>

__global__ void hello ()
{
```

```
//   int myID = ( blockIdx.z * gridDim.x * gridDim.y  +
//                blockIdx.y * gridDim.x +
//                blockIdx.x ) * blockDim.x * blockDim.y * blockDim.z +
//                threadIdx.z *  blockDim.x * blockDim.y +
//                threadIdx.y * blockDim.x +
//                threadIdx.x;

//  Simplification of above
  //grid: 3D --- z,y,x: all dims and blockids
  //block: 1D -- x
  int myID = ( blockIdx.z * gridDim.x * gridDim.y  +
               blockIdx.y * gridDim.x +
               blockIdx.x ) * blockDim.x +
               threadIdx.x;

  printf ("Hello world from %i\n", myID);
}

int main ()
{
  dim3 g (4, 3, 2);
  hello <<< g, 10 >>> ();
  cudaThreadSynchronize ();
  //cudaDeviceSynchronize();  //use instead, ^ is deprecated
  return 0;
}
```

```
'File written in /content/src/hello2.cu'
```

```
!nvcc -arch=sm_75 /content/src/hello2.cu -o "/content/src/hello2.o"
```

```
/content/src/hello2.cu: In function ‘int main()’:
/content/src/hello2.cu:39:22: warning: ‘cudaError_t cudaThreadSynchronize()’ is deprecated [-Wdeprecated-declarations]
   39 |   cudaThreadSynchronize ();
      |   ~~~~~~~~~~~~~~~~~~~^~
/usr/local/cuda/bin/../targets/x86_64-linux/include/cuda_runtime_api.h:1052:46: note: declared here
 1052 | extern __CUDA_DEPRECATED __host__ cudaError_t CUDARTAPI cudaThreadSynchronize(void);
      |                                              ^~~~~~~~~~~~~~~~~~~~~
```

```
!chmod 755 /content/src/hello2.o
!/content/src/hello2.o
```

```
Hello world from 29
Hello world from 70
Hello world from 71
Hello world from 72
Hello world from 73
Hello world from 74
Hello world from 75
Hello world from 76
Hello world from 77
Hello world from 78
Hello world from 79
Hello world from 220
Hello world from 221
Hello world from 222
Hello world from 223
Hello world from 224
Hello world from 225
Hello world from 226
Hello world from 227
Hello world from 228
Hello world from 229
```

AL observar los resultados podemos ver que el ID más alto alcanzado es de 239. Sin embargo los hilos no se procesan en su totalidad en una forma secuencial, solamente hay pequeñas secciones que se ejecutan de esta forma.

Modificamos para mostrar nombre y carnet así como un cambio en el proceso del programa.

```
%%cuda --name hello2_modified.cu

/*
 ============================================================================
 Author        : G. Barlas
 Version       : 1.0
 Last modified : December 2014
 License       : Released under the GNU GPL 3.0
 Description   :
 To build use  : nvcc hello2.cu -o hello2 -arch=sm_20
 ============================================================================
 */
#include <stdio.h>
#include <cuda.h>

__global__ void hello ()
{
int maxID = ( blockIdx.z * gridDim.x * gridDim.y  +
             blockIdx.y * gridDim.x +
             blockIdx.x ) * blockDim.x * blockDim.y * blockDim.z +
             threadIdx.z *  blockDim.x * blockDim.y +
             threadIdx.y * blockDim.x +
             threadIdx.x;

//  Simplification of above
  //grid: 3D --- z,y,x: all dims and blockids
  //block: 1D -- x
  int myID = ( blockIdx.z * gridDim.x * gridDim.y  +
               blockIdx.y * gridDim.x +
               blockIdx.x ) * blockDim.x +
               threadIdx.x;

  printf ("MARCO JURADO 20308 (%i) \n", myID);
}

int main ()
{
  dim3 g (4,2);
  dim3 b (32,16);
  hello <<<g, b>>>();
  cudaThreadSynchronize ();
  //cudaDeviceSynchronize();  //use instead, ^ is deprecated
  return 0;
}
```

```
'File written in /content/src/hello2_modified.cu'
```

```
!nvcc -arch=sm_75 /content/src/hello2_modified.cu -o "/content/src/hello2_modified.o"
```

```
/content/src/hello2_modified.cu(17): warning #177-D: variable "maxID" was declared but never referenced

/content/src/hello2_modified.cu: In function ‘int main()’:
/content/src/hello2_modified.cu:40:22: warning: ‘cudaError_t cudaThreadSynchronize()’ is deprecated [-Wdeprecated-declarations]
   40 |   cudaThreadSynchronize ();
      |   ~~~~~~~~~~~~~~~~~~~^~
/usr/local/cuda/bin/../targets/x86_64-linux/include/cuda_runtime_api.h:1052:46: note: declared here
 1052 | extern __CUDA_DEPRECATED __host__ cudaError_t CUDARTAPI cudaThreadSynchronize(void);
      |                                               ^~~~~~~~~~~~~~~~~~~~
```

```
!chmod 755 /content/src/hello2_modified.o
!/content/src/hello2_modified.o
```

```
MARCO JURADO 20308 (102)
MARCO JURADO 20308 (103)
MARCO JURADO 20308 (104)
MARCO JURADO 20308 (105)
MARCO JURADO 20308 (106)
MARCO JURADO 20308 (107)
MARCO JURADO 20308 (108)
MARCO JURADO 20308 (109)
MARCO JURADO 20308 (110)
MARCO JURADO 20308 (111)
MARCO JURADO 20308 (112)
MARCO JURADO 20308 (113)
MARCO JURADO 20308 (114)
MARCO JURADO 20308 (115)
MARCO JURADO 20308 (116)
MARCO JURADO 20308 (117)
MARCO JURADO 20308 (118)
MARCO JURADO 20308 (119)
MARCO JURADO 20308 (120)
MARCO JURADO 20308 (121)
MARCO JURADO 20308 (122)
MARCO JURADO 20308 (123)
MARCO JURADO 20308 (124)
MARCO JURADO 20308 (125)
MARCO JURADO 20308 (126)
MARCO JURADO 20308 (127)
MARCO JURADO 20308 (160)
MARCO JURADO 20308 (161)
MARCO JURADO 20308 (162)
MARCO JURADO 20308 (163)
MARCO JURADO 20308 (164)
MARCO JURADO 20308 (165)
MARCO JURADO 20308 (166)
MARCO JURADO 20308 (167)
MARCO JURADO 20308 (168)
MARCO JURADO 20308 (169)
MARCO JURADO 20308 (170)
MARCO JURADO 20308 (171)
MARCO JURADO 20308 (172)
MARCO JURADO 20308 (173)
MARCO JURADO 20308 (174)
MARCO JURADO 20308 (175)
MARCO JURADO 20308 (176)
MARCO JURADO 20308 (177)
MARCO JURADO 20308 (178)
MARCO JURADO 20308 (179)
MARCO JURADO 20308 (180)
MARCO JURADO 20308 (181)
MARCO JURADO 20308 (182)
MARCO JURADO 20308 (183)
MARCO JURADO 20308 (184)
MARCO JURADO 20308 (185)
MARCO JURADO 20308 (186)
MARCO JURADO 20308 (187)
MARCO JURADO 20308 (188)
MARCO JURADO 20308 (189)
MARCO JURADO 20308 (190)
MARCO JURADO 20308 (191)
```

El codigo con el mayor id es de 255. No se observan ejecución de los hilos en un orden lineal, sin embargo se observan bloques de ejecución de varios hilos dando de rangos desde por ejemplo del 0 al 30 y del 60 al 223.

Ahora modificamos para ejecutar 100,000 hilos

```
%%cuda --name hello2_modified_100000.cu
```

```
#include <stdio.h>
#include <cuda_runtime.h>

__global__ void processKernel()
{
    int threadID = blockIdx.x * blockDim.x + threadIdx.x;

    if (threadID == 0) {
        printf("Primer thread %d\n", threadID);
    }

    if (threadID == 99999) {
        printf("Ultimo thread %d\n", threadID);
    }
}

int main()
{
    int numThreads = 100000;
    int threadsPerBlock = 1024;
    int numBlocks = (numThreads + threadsPerBlock - 1) / threadsPerBlock;

    printf("Configuración utilizada: numBlocks=%d, threadsPerBlock=%d\n", numBlocks, threadsPerBlock);

    processKernel<<<numBlocks, threadsPerBlock>>>();
    cudaDeviceSynchronize();

    return 0;
}
```

```
'File written in /content/src/hello2_modified_100000.cu'
```

```
!nvcc -arch=sm_75 /content/src/hello2_modified_100000.cu -o "/content/src/hello2_modified_100000.o"
```

```
!chmod 755 /content/src/hello2_modified_100000.o
!/content/src/hello2_modified_100000.o
```

```
Configuración utilizada: numBlocks=98, threadsPerBlock=1024
Primer thread 0
Ultimo thread 99999
```