



ugr

Universidad  
de Granada

ARQUITECTURA Y COMPUTACIÓN DE ALTAS PRESTACIONES  
GRADO EN INGENIERÍA INFORMÁTICA

---

## PRÁCTICA 5

### SUMA DE VECTORES CON CUDA

---

**Autor**

Vladislav Nikolov Vasilev

**Rama**

Ingeniería de Computadores



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE  
TELECOMUNICACIÓN

CURSO 2019-2020

# Índice

1. Introducción	2
2. Especificaciones de la GPU	2
3. Versión secuencial	4
4. Versión paralela con CUDA	4
5. Resultados	5
6. Conclusiones	5
Referencias	6

## 1. Introducción

En esta práctica se pide implementar dos versiones de un mismo programa: una primera versión secuencial y una versión en **CUDA**. El objetivo del programa es, dados dos vectores **A** y **B**, ambos de tamaño  $n$ , obtener un vector de salida **C** del mismo tamaño en el que  $\forall \mathbf{C}_i \in \mathbf{C}$ , su valor venga determinado por la expresión:

$$\mathbf{C}_i = \left( \left( \frac{\log(5 \cdot \mathbf{A}_i \cdot 100 \cdot \mathbf{B}_i) + 7 \cdot \mathbf{A}_i}{0.33} \right)^3 \right)^7 \quad (1)$$

Además, se ha especificado que los vectores de entrada **A** y **B** se tienen que replicar un número de veces, de forma que su longitud se vea incrementada. En este caso, se ha decidido **que se van a tener 5 copias de los vectores originales**, es decir, que los vectores van a tener una longitud de  $5n$ , donde  $n$  es el tamaño original especificado en los archivos proporcionados.

Para cada uno de los nueve problemas proporcionados se van a tomar medidas de tiempo de lo que se tarda en hacer todas las operaciones con los dos programas. Con ellas se representará la evolución del tiempo de ejecución en cada caso en función del tamaño del problema y se hará un estudio de la ganancia.

Adicionalmente, se pide que se obtengan las especificaciones de la GPU que se está utilizando, de manera que se tenga más información sobre esta.

## 2. Especificaciones de la GPU

Primeramente vamos a ver cuáles son las especificaciones de la GPU que vamos a utilizar. Para ello se ha creado un programa que puede encontrarse en el archivo `deviceProperties.cu`, el cuál muestra la siguiente información:

- Nombre del dispositivo.
- Número de SMs (procesadores).
- Número de SPs por SM (cores por procesador).
- Número total de SPs.
- Memoria total de la gráfica en MB.
- Memoria compartida por SM en KB.

- Memoria compartida por bloque en KB.
- Número máximo de hebras por bloque.
- Número máximo de hebras por cada dimensión.

El código que muestra esta información es el siguiente:

```
1 int main()
2 {
3     int n_devices;
4
5     cudaGetDeviceCount(&n_devices);
6
7     int i;
8
9     for (i = 0; i < n_devices; i++)
10    {
11        cudaDeviceProp prop;
12        cudaGetDeviceProperties(&prop, i);
13        int numSPs = getSPcores(prop);
14
15        printf("Device number: %d\n", i);
16        printf(" Device name: %s\n", prop.name);
17        printf(" Number of SMs: %d\n", prop.multiProcessorCount);
18        printf(" Number of SPs per SM: %d\n", numSPs);
19        printf(" Total Number of SPs: %d\n", numSPs * prop.
20        multiProcessorCount);
21        printf(" Total Available Global Memory Size (MB): %zu\n",
22        prop.totalGlobalMem / (1<<20));
23        printf(" Shared Memory Per SM (KB): %zu\n", prop.
24        sharedMemPerMultiprocessor / (1<<10));
25        printf(" Shared Memory Per Block (KB): %zu\n", prop.
26        sharedMemPerBlock / (1<<10));
27        printf(" Max. Threads Per Block: %d\n", prop.
28        maxThreadsPerBlock);
29        printf(" Max. Threads Per Dimension: x: %d, y: %d, z: %d\n",
30        prop.maxThreadsDim[0], prop.maxThreadsDim[1], prop.
31        maxThreadsDim[2]);
32    }
33
34    return 0;
35 }
```

La función `getSPcores` se ha extraído de una respuesta a una pregunta en *StackOverflow* [1]. En caso de tener más de una tarjeta gráfica (que no es el caso), se mostraría la información para cada una de ellas. En este caso se ha obtenido la siguiente información:

```
Device number: 0
Device name: GeForce GTX 1050
```

Number of SMs: 5  
Number of SPs per SM: 128  
Total Number of SPs: 640  
Total Available Global Memory Size MB: 4040  
Shared Memory Per SM KB: 96  
Shared Memory Per Block KB: 48  
Max. Threads Per Block: 1024  
Max. Threads Per Dimension: x: 1024, y: 1024, z: 64

### 3. Versión secuencial

Una vez que hemos visto las especificaciones de nuestra gráfica, vamos a ver cómo sería una primera versión secuencial del programa.

La versión secuencial carga los ficheros de datos que se han especificado como parámetros, carga los valores, los copia un número de veces que se determina como parámetro y realiza la operación, obteniendo el vector **C** de salida. La función que realiza el cálculo que se puede ver en la ecuación (1) es la siguiente:

```
1 void add_vectors(float *A, float *B, float *C, int N)
2 {
3     int i;
4
5     for (i = 0; i < N; i++)
6     {
7         C[i] = pow(pow(log(5*A[i]*100*B[i] + 7*A[i]) / 0.33, 3), 7);
8     }
9 }
```

La función recibe como parámetro los dos vectores de entrada, el de salida y el tamaño de los vectores, y con un bucle recorre los elementos de **A** y **B** y calcula el correspondiente elemento de **C**.

El resto del código se encuentra en el archivo `vectorAdd.c` y se puede consultar de ser necesario. Como en general son operaciones sencillas, no se van a explicar ni a incluir aquí, ya que no se considera que estén relacionadas con la asignatura (lectura de ficheros, reserva de memoria, etc.).

### 4. Versión paralela con CUDA

Una vez implementada y probada la versión secuencial se ha procedido a desarrollar la versión paralela.

## 5. Resultados

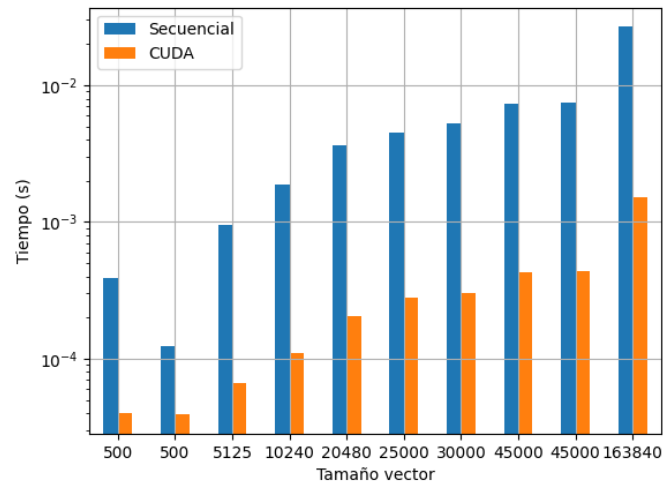


Figura 1: Evolución del tiempo de ejecución del programa secuencial y el programa en CUDA en función del tamaño del vector con escala logarítmica en el eje Y.

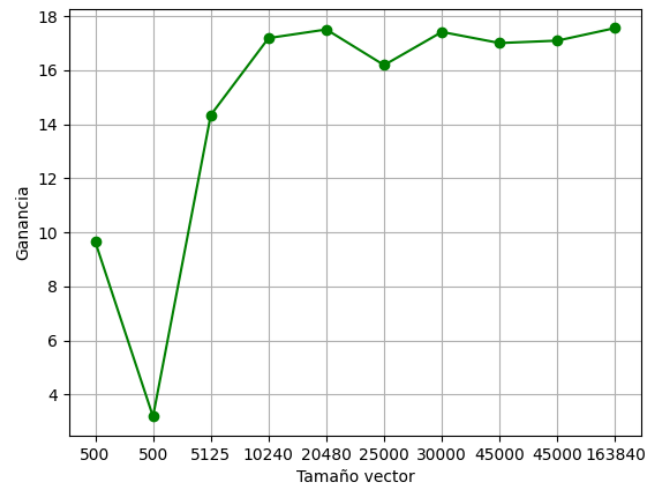


Figura 2: Ganancia obtenida en función del tamaño del vector.

## 6. Conclusiones

## Referencias

- [1] StackOverflow. *How can I get number of Cores in cuda device?*  
[https://stackoverflow.com/questions/32530604/  
how-can-i-get-number-of-cores-in-cuda-device](https://stackoverflow.com/questions/32530604/how-can-i-get-number-of-cores-in-cuda-device)