

Práctica 1

Introducción a la programación CUDA

Para todos los ejercicios se debe implementar una versión secuencial, calcular la aceleración obtenida y realizar en todos los casos la verificación de resultados.

Compilar en CUDA:

```
nvcc -o archivoEjecutable archivoFuente.cu
```

Consideraciones con respecto a los ejercicios de algebra lineal (involucran operaciones con matrices):

En el caso particular de la multiplicación de matrices de $N \times N$, una operación $C=AB$ necesita que las matrices C y A se encuentren almacenadas en memoria por filas, mientras que la matriz B debe estar almacenada en memoria por columnas. Esto favorece la localidad de caché y permite alcanzar mayor rendimiento.

Los lenguajes de programación permiten definir variables del tipo matriz pero no dan la posibilidad de seleccionar el orden que deben tener en memoria. Para controlar este orden, los programadores declaran las matrices como arreglos de dimensión $N \times N$.

El inconveniente está en encontrar la forma de acceder al elemento (i,j) de una matriz ordenada por filas o por columnas cuando éstas se almacenan como arreglos. La solución es la siguiente:

*Si M es un arreglo que representa una matriz de $N \times N$ almacenada por filas, se accede a la posición i,j como $M[i*N+j]$*

*Si M es un arreglo que representa una matriz de $N \times N$ almacenada por columnas, se accede a la posición i,j como $M[i+j*N]$*

Esta es la notación que utilizaremos durante todos los ejercicios que involucren matrices.

1. Realizar un programa CUDA que dado un vector V de N números enteros multiplique a cada número por una constante C. Realizar dos implementaciones:
 - a. C y N deben ser pasados como parámetros al kernel.
 - b. C y N deben estar almacenados en la memoria de constantes de la GPU.

Además, analizar las siguientes modificaciones sobre los tipos de datos:

 - c. Modificar el algoritmo para que la constante C y los números del vector V sean de tipo float.
 - d. Modificar el algoritmo de manera que C y los números del vector V sean de tipo double.
 - e. ¿Existen diferencias en el tiempo de ejecución al usar los distintos tipos de datos?
2. Calcular $C = A + B$ donde A, B y C son vectores de tamaño N:
 - a. Realizar el algoritmo haciendo que cada hilo calcule una posición.
 - b. Realizar el algoritmo haciendo que cada hilo calcule más de una posición.
3. Calcular $C = A + B$ donde A, B y C son matrices de NxN.
 - a. Realizar el algoritmo haciendo que cada hilo calcule una posición.
 - b. Realizar el algoritmo haciendo que cada hilo calcule más de una posición.
4. Calcular la transpuesta de una matriz A de NxN.
5. Calcular $C = AB$ donde A, B y C son matrices de NxN (Cada hilo calcula una posición de C).
6. Realizar un algoritmo que calcule la suma de los elementos de un vector de tamaño N.
7. Realizar un algoritmo que encuentre el máximo y el mínimo valor de los elementos de un vector de tamaño N.
8. Realizar un algoritmo que calcule la cantidad de ocurrencias de un elemento dado X en un vector de tamaño N.
9. Realizar un algoritmo que encuentre un string de tamaño S en una cadena de tamaño N y retorne la posición del vector donde se encuentra. El string S aparece solo una vez.
10. Realizar un algoritmo que resuelva la siguiente ecuación: $\sum_{i=0}^{N-1} V[i] - \bar{V}$ donde V es un vector de N elementos. \bar{V} es el promedio de los elementos del vector V.