CUDA: Práctica Final

Computación Paralela

Grado en Ingeniería Informática

Curso 2024/25

El trabajo propuesto consiste en escribir un código CUDA que permita realizar una multiplicación de matrices en una GPU. Esta operación matricial se llevará a cabo combinando un conjunto de operaciones algebraicas. Para ello, se proponen una serie de ejercicios que deben ser resueltos en el orden indicado con el fin de obtener correctamente la versión final del código.

Ejercicio 1: Producto Escalar ($R = V_1 \cdot V_2$)

Completa el código proporcionado en el directorio `scalarProd` de forma que, dados dos vectores, $V_1 \in \Re^n$ y $V_2 \in \Re^n$, permita realizar su producto escalar. La suma de los elementos obtenidos en R se debe realizar aplicando la operación de reducción implementada en el Ejercicio 5 (ítem 5) del Boletín de Ejercicios.

Ejercicio 2: Matriz Traspuesta (A^T)

Completa el código proporcionado en el directorio `transpose` de forma que permita obtener la matriz traspuesta, A^T, de una matriz cuadrada, $A \in \Re^{n \times n}$. Esta operación debe llevarse a cabo en la GPU haciendo uso de memoria *shared*.

Ejercicio 3: Multiplicación de Matrices ($C = A \times B$)

Completa el código proporcionado en directorio `matrixMul` de forma que, dadas dos matrices cuadradas, $A \in \Re^{n \times n}$ y $B \in \Re^{n \times n}$, permita obtener su producto en una matriz $C \in \Re^{n \times n}$. Esta operación se debe realizar multiplicando cada fila de A por todas las filas de B^T (traspuesta de B). Para obtener B^T se debe hacer uso del kernel implementado en el Ejercicio 2. Para calcular el producto de cada fila de A por cada fila de B^T se utilizará el kernel implementado en el Ejercicio 1.

Observaciones:

- Fecha de Entrega: **11 de Junio de 2025**.
- Se entregará únicamente el fichero matrixMul.cu con la implementación de todo el código necesario.
- El programa debe funcionar correctamente con diferentes configuraciones para el tamaño de problema y los bloques de hilos. Para comprobarlo se puede comparar la matriz C obtenida en la GPU con la que se obtendría al realizar el producto matricial en la CPU.
- Aquellos estudiantes que no entreguen esta práctica en la convocatoria ordinaria o no consigan una calificación ≥ 5, deberán modificar el código para que permita realizar el producto matricial de forma asíncrona en la GPU mediante el uso de *streams*, kernels concurrentes y memoria *pinned*.