

Práctica 6: Tiempos CUDA VS MPI

Autor: Antonio Felipe Guzmán Martín

En esta última práctica comparamos la ejecución del algoritmo de cálculo de primos hasta un número tope M, en su versión CUDA y MPI. Los algoritmos se han ejecutado en diferentes máquinas por lo que compararemos solo los tiempos. En este caso CUDA consigue tener mejores tiempos a partir del número tope 262144.

La versión de CUDA calcula de forma diferente el número de primos. Ejecuta dos kernels. En el primero sitúa en un vector tamaño número "TOPE" si cada uno de los números hasta el número deseado es primo o no. En el caso de que sea primo el elemento i ésimo se da valor 1 y en caso contrario valor 0.

Posteriormente se llama al segundo kernel que hace reduce sobre los elementos del vector. Para ello usa el vector de primos con unos y ceros de entrada y se devuelve en la salida otro vector cuyos elementos contienen la suma de las hebras del bloque.

Para calcular el número de primos hasta tamaño tope 1048576. Hemos necesitado de 4096 bloques (vector de 4096 elementos) en los que cada elemento del vector se almacenaba la suma parcial de 256 hebras.
 $256 * 4096 = 1048576$

El cálculo de primos para MPI se hace de la misma forma que en la práctica anterior.

Dividiendo el trabajo entre hebras con **MPI_Bcast** y haciendo la suma de resultados parciales con **MPI_Reduce**. (Es la versión 1 prime.c). Se ha elegido la ejecución con 13 procesos por que es la que ha ofrecido tiempos más bajos entre todas las ejecutadas (desde $np=1$ hasta $np=13$).

Tiempo MPI	Tiempo cuda (s)	Tamaño
0,010104	0,286311	2048
0,019867	0,289682	4096
0,022989	0,336743	8192
0,01999	0,341601	16384
0,067758	0,359952	32768
0,188142	0,450027	65536
0,632428	0,823978	131072
2,292632	2,1895	262144
8,646306	7,423526	524288
30,441346	27,921961	1048576

