

## Paralelismo - Práctica 2

Los puntos clave de la paralelización de esta práctica (cálculo del número de primos menor que un valor) son los siguientes:

- ♦ La inicialización de todos los procesos, el cálculo de números primos menores que el valor y la finalización de todos los procesos (incluyendo que el proceso 0 solicita el valor por teclado y muestra el resultado por pantalla) se hace exactamente igual que en la p1, por eso considero que sería redundante explicarlo, de todas formas, la explicación está en el informe de dicha práctica.
- ♦ El proceso 0 envía el valor obtenido al resto de procesos a través de una operación colectiva de difusión o broadcast, en el caso a), a través de una colectiva estándar (MPI\_Bcast) y en el caso b), una colectiva implementada por nosotros (MPI\_BinomialColectiva) (implementada en las líneas 38 a 59). Esta función imita el comportamiento de MPI\_Bcast, sin embargo, por razones de eficiencia, el proceso 0 no envía el valor a todos los procesos uno por uno, sino que el valor se va enviando en una estructura de árbol binomial, en cada paso  $i$  de la función los procesos con rango  $rank < 2^{i-1}$  envían el valor al proceso con rango igual a  $rank + 2^{i-1}$ , de esta forma, en cada paso hay el doble de procesos enviando, optimizando así la difusión en gran medida. Añadir que todos los procesos excepto el 0, obviamente, reciben el valor a través de MPI\_Recv (líneas 51 y 52) antes de proceder a su envío. (la colectiva se llama en la línea 24/77)
- ♦ El proceso 0 recolecta los valores obtenidos en el cálculo de números primos por el resto de procesos a través una operación colectiva de reducción, en el caso a), a través de una colectiva estándar (MPI\_Reduce) y en el caso b), una colectiva implementada por nosotros (MPI\_FlattreeColectiva) (implementada en las líneas 8 a 36). Esta última función imita el comportamiento de MPI\_Reduce aunque asumiendo que la operación siempre será una suma y trabajamos con enteros (el proceso root hace un MPI\_Recv por cada proceso y cada proceso hace un único MPI\_Send). (la colectiva se llama en la línea 43/97)

\*: Las líneas se muestran según la siguiente disposición: (líneas caso a/ líneas caso b)