

Paralelismo - Práctica 3

Los puntos clave de la paralelización de esta práctica son los siguientes:

- ♦ Si el tamaño de la matriz no es divisible entre el número de procesos, añadimos a esta posiciones extra (que no serán usadas) hasta que su tamaño sea múltiplo del número de procesos, esta técnica se conoce como "*padding*" y la usamos con el objetivo de facilitar la asignación de tareas. (líneas 66 a 74)
- ♦ Reservamos memoria para las matrices globales y locales con las que vamos a trabajar, los vectores que almacenarán los tiempos de comunicación y computación de cada proceso y los vectores de resultados global y local. (líneas 80 a 84)
- ♦ El proceso 0 inicializa las matrices de datos y asigna un bloque de esta matriz a cada proceso a través de colectivas *scatter*, para que cada proceso trabaje con su bloque correspondiente (incluido él mismo), esto se conoce como descomposición de dominio, en concreto, distribución de matrices por bloques. Cada proceso guarda este bloque en su matriz local. El tamaño de este bloque será equitativo gracias al "*padding*" realizado previamente y a que todas las secuencias de ADN tengan el mismo número de bases (constante N). (líneas 86 a 100)
- ♦ Todos los procesos miden lo similares que son las secuencias de ADN de los dos conjuntos (bloques) que le fueron asignados, estos cálculos son realizados en paralelo por todos los procesos. Cada proceso almacena los resultados obtenidos en su vector local. (líneas 108 a 114)
- ♦ El proceso 0 recoge los resultados locales de cada proceso en un vector global (variable *result*), así como sus tiempos de computación (variable *computingTime*) y comunicación (variable *communicationTime*) que fueron medidos a lo largo de la ejecución del programa, todo esto se hace mediante colectivas *gather*. (líneas 119 a 127)
- ♦ Solo el proceso 0 imprime por pantalla el vector resultado o los tiempos de computación y comunicación de cada proceso, dependiendo del valor de la constante *DEBUG*. (líneas 129 a 140)
- ♦ Todos los procesos liberan la memoria reservada previamente y finalizan.