

Emilio Martínez Varela

DNI: **79344591L**

Grupo: 1.4

Esta práctica está compuesta por dos partes:

- Practica2a
- Practica2b

En el primer apartado debemos modificar la práctica 1 y utilizar operaciones colectivas MPI. Por una parte usamos MPI_Bcast que se encarga de enviar un mensaje desde un proceso origen a todos los procesos pertenecientes al mismo grupo, y MPI_Reduce que se encarga de reducir un valor de un grupo de procesos en un único proceso raíz.

La implementación de esta primera parte respecto a la práctica 1 es más sencilla ya que esta operación colectiva nos permite en una sola línea de código realizar lo mismo que los bucles de la práctica 1. Para ello usamos una variable root que define quién es el proceso root (cualquier proceso que esté comprendido dentro del número máximo de procesos asignados) el cual se encarga de enviar los datos y recibirlos de todos los demás procesos.

En el segundo apartado nos pide implementar dos funciones equivalentes a MPI_Recv y a MPI_Bcast, las cuales llamamos MPI_FlatticeColectiva y MPI_BinomialBcast respectivamente. El caso de MPI_BinomialBcast tiene como diferencia que el paso del mensaje se tiene que realizar como árbol binomial.

La implementación de MPI_FlatticeColectiva es similar al código de la práctica 1 donde en este caso obtenemos los datos de todos los procesos distintos al root, además realiza por defecto la operación suma de los valores.

La función MPI_BinomialBcast se diferencia a la Bcast normal en que en vez de enviar los datos en orden, estos se envían de forma de árbol binomial, asumiendo que el root es 0, donde en cada paso i , los procesos con el $\text{rank} < 2^{(i-1)}$ se comunican con el proceso $\text{rank} + 2^{(i-1)}$.

En esta función utilizamos el operador \ll que nos permite asignar con desplazamiento a la izquierda y esto hace que se reduzca el tamaño del código.