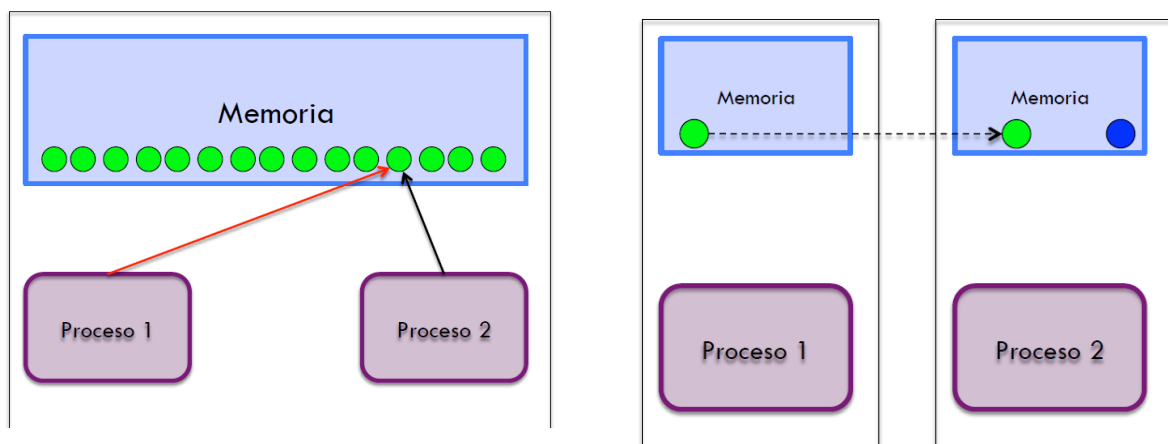


# Tema 4: MPI

Es un **estándar** para una biblioteca de paso de mensajes, cuyo objetivo es comunicar procesos en ordenadores remotos o procesos ejecutados en un mismo ordenador.

Hay que tener en cuenta que no siempre compensa paralelizar todos los programas, cuando el cálculo de los datos es menor que el coste del envío de los datos es mejor hacerlo en local, mientras que cuando el coste del envío es mejor que el coste del calculo en el ordenador local entonces compensa enviarlo a que lo realice otro ordenador.

- **Memoria compartida:** Todos los procesos acceden a una misma memoria, sin problemas en lectura pero si en escritura.
- **Memoria distribuida:** Cada proceso tiene su memoria "privada", la comunicación entre estos se realiza mediante paso de mensajes, por ejemplo sockets.



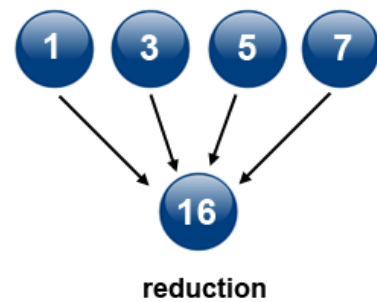
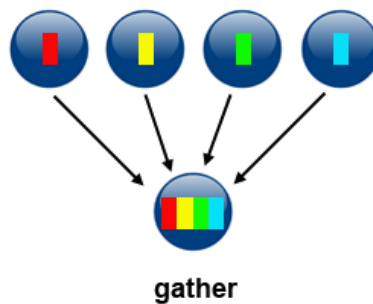
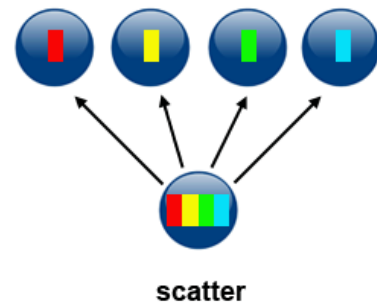
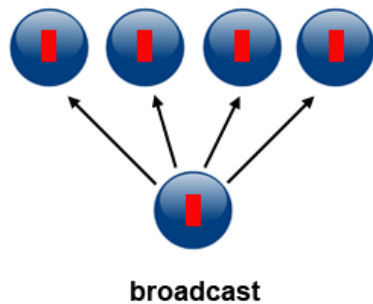
Nosotros vamos a utilizar el **modelo de ejecución SPMD** (Single Program Multiple Data)

Consiste en un sólo programa que se ejecuta en paralelo, de tal forma que este programa se copia en todos los nodos y se ejecutan "a la vez", sabemos qué nodo es cada uno mediante su rank(id).

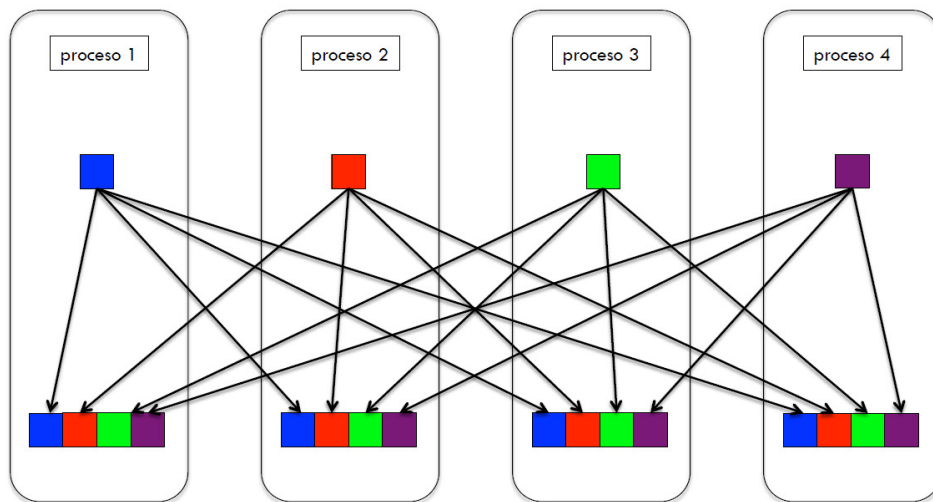
## Funciones

- **MPI\_COMM\_WORLD** es el comunicador que agrupa todos los procesos.

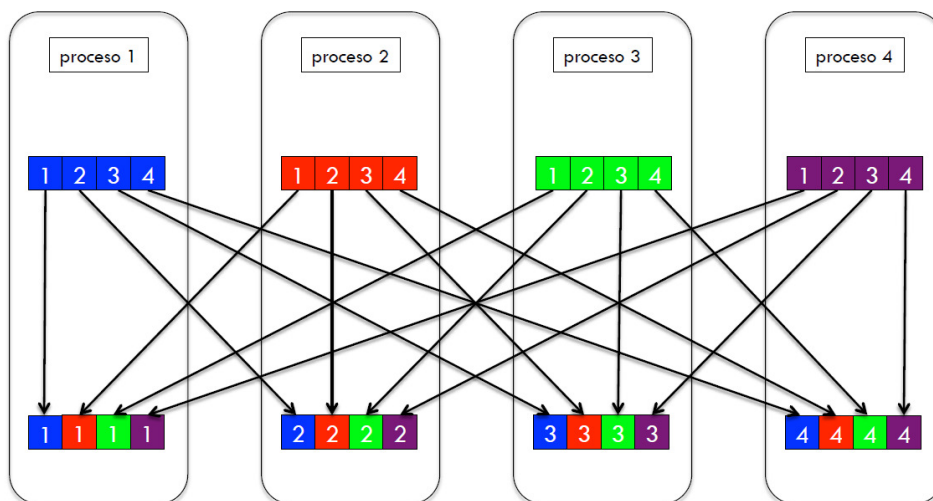
- `int MPI_Comm_size(MPI_Comm comm, int *size);` Nos indica el número de procesos sobre los que vamos a realizar el cálculo.
- `int MPI_Comm_rank(MPI_Comm comm, int *rank);` Nos indica el rank del proceso asociado al comunicador.



- **Scatter:** Divide de forma equitativa y manda a todos los procesos un cacho de datos para procesar (se puede quedar uno el master y procesarlo también, pero en la práctica esto no lo hacemos)
- **Gather:** Paso inverso al scatter, reunimos todos los cachos de todos los procesos.
- **AllGather:** Reunimos todos los cachos de todos los procesos y además se manda copias entre todos, de esta forma todos los procesos tienen todos los datos y no sólo lo tiene el master.



- **AlltoAll:** Se envía los datos y se organizan por "número de partición" , es decir en el proceso 1 se encontrarían todos los cachos\_1 de todos los procesos, en el proceso 2 se encontrarían todos los cachos\_2 y así sucesivamente.



- **Reduce:** Funciona como el scatter pero ya tiene funciones usuales implementadas como la suma, resta, producto, etc. Por lo que en una sola llamada tenemos el resultado de los datos que enviamos.