

# Comunicación colectiva

Carlos E. Alvarez<sup>1</sup>.

<sup>1</sup>Dep. de Matemáticas aplicadas y Ciencias de la Computación, Universidad del Rosario

2020-I

- En las comunicaciones punto a punto participan un par de procesos
- Las comunicaciones colectivas permiten que más de dos procesos participen

# Comunicaciones colectivas

- Participan todos los procesos en un comunicador
- Las operaciones de comunicación no generan bloqueo
- No se usan etiquetas

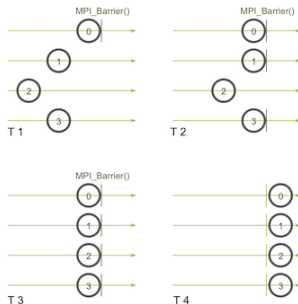
# Comunicaciones colectivas

Operaciones colectivas:

- Barrier
- Broadcast
- Scatter
- Gather
- Reduction

# Barrier

Los procesos detienen su ejecución hasta que todos lleguen a la barrera.



# Barrier

Sincroniza todos los procesos.

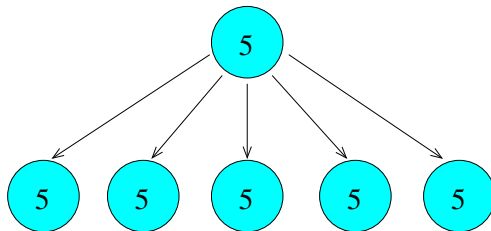
```
MPI_Barrier(comm)
```

# Barrier

Ejemplo: `barrier.cpp`

# Broadcast

Un proceso envía datos a todos los procesos (incluyéndolo).





# Broadcast

Copia los datos en `sendbuf` a todos.

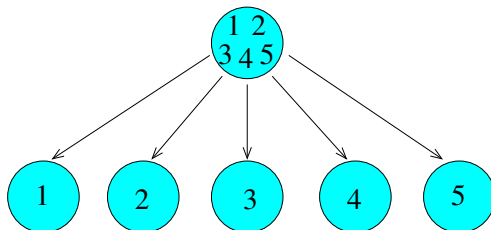
```
MPI_Bcast (sendbuf, count, datatype, source, comm)
```

# Broadcast

Ejemplo: `broadcast.cpp`

# Scatter

Un proceso particiona los datos y envía una parte a cada proceso (incluyéndolo).



# Scatter

Distribuye los datos en `sendbuf` entre todos, quienes guardan lo recibido en `recvbuf`.

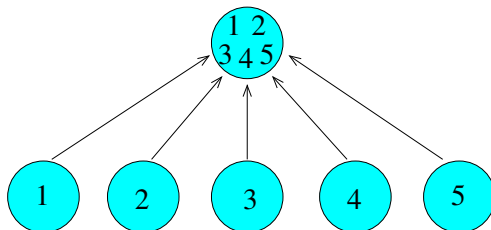
```
MPI_Scatter(sendbuf, sendcnt, datatype, recvbuf, rcvcnt,  
            datatype, source, comm)
```

# Scatter

Ejemplo: `scatter.cpp`

# Gather

Un proceso recolecta datos de cada uno de los procesos (incluyéndolo).



# Gather

Los datos en `sendbuf` de cada proceso son recolectados en `recvbuf` por un proceso.

```
MPI_Gather(sendbuf, sendcnt, datatype, recvbuf, rcvcnt,  
           datatype, destination, comm)
```

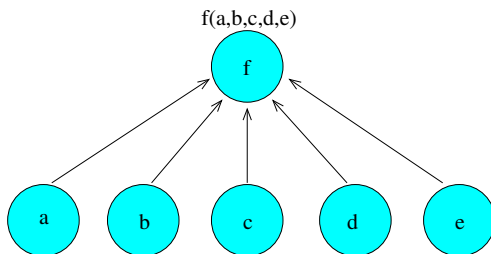
# Gather

Ejemplo: `gather.cpp`



# Reduce

Un proceso recolecta en `recvbuf` los datos enviados por todos los procesos, guardados en `sendbuf`, y realiza una operación de reducción sobre ellos.



# Reduce

Un proceso realiza la operación  $op$  sobre los datos enviados por todos los procesos.

```
MPI_Reduce(sendbuf,recvbuf,count,datatype,op,  
            destination,comm)
```

# Reduce

Operaciones:

Nombre	Operación
MPI_MAX	Maximum
MPI_MIN	Minimum
MPI_SUM	Addition
MPI_PROD	Product
MPI_LAND	Logical AND
MPI_LOR	Logical OR

## Reduce

Ejemplo: reduce\_sum.cpp