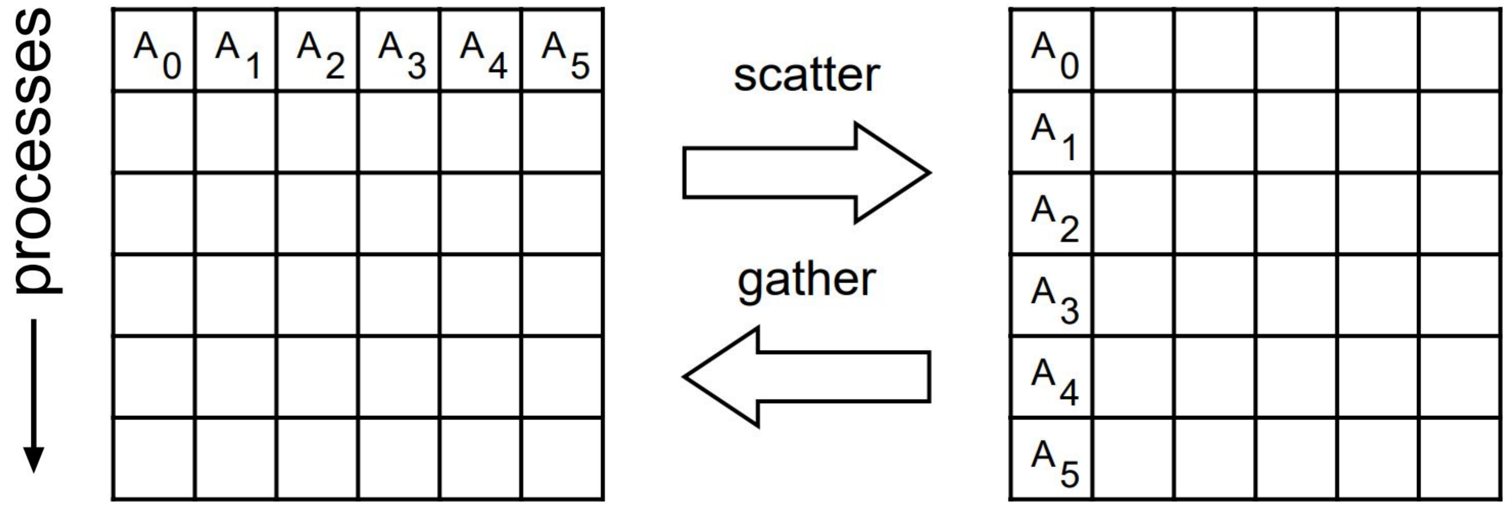
Gather and Scatter Operations

Gather and Scatter Operations



MPI_Gather

- int MPI_Gather(const void *sendbuf, int sendcount, MPI_Datatype sendtype, void *recvbuf, int recvcount, MPI_Datatype recvtype, int root, MPI_Comm comm);

MPI_Gather

- Each process (root process included) sends the contents of its send buffer to the root process.

- The root process receives the messages and stores them in rank order.

- The outcome is *as* if each of the n processes in the group including the root process had executed a call to MPI_Send and the root had executed n calls to MPI_Recv.

MPI_Gather Simple Example

```
int gsize, sendarray[100];
int root, myrank, *rbuf;
...
MPI_Comm_rank(MPI_COMM_WORLD,myrank);
if (myrank == root) {
    MPI_Comm_size(MPI_COMM_WORLD,&gsize);
    rbuf = (int *)malloc(gsize*100*sizeof(int));
    }
MPI_Gather(sendarray,100,MPI_Int,rbuf,100,MPI_Int,root,
MPI_COMM_WORLD);
```

MPI_Scatter

- int MPI_Scatter(const void *sendbuf, int sendcount, MPI_Datatype sendtype, void *recvbuf, int recvcount, MPI_Datatype recvtype, int root, MPI_Comm comm);

MPI_Scatter

- It is the inverse operation to MPI_Gather.

- The outcome is *as* if the root executed n send operations MPI_Send, and each process executed a receive MPI_Receive.

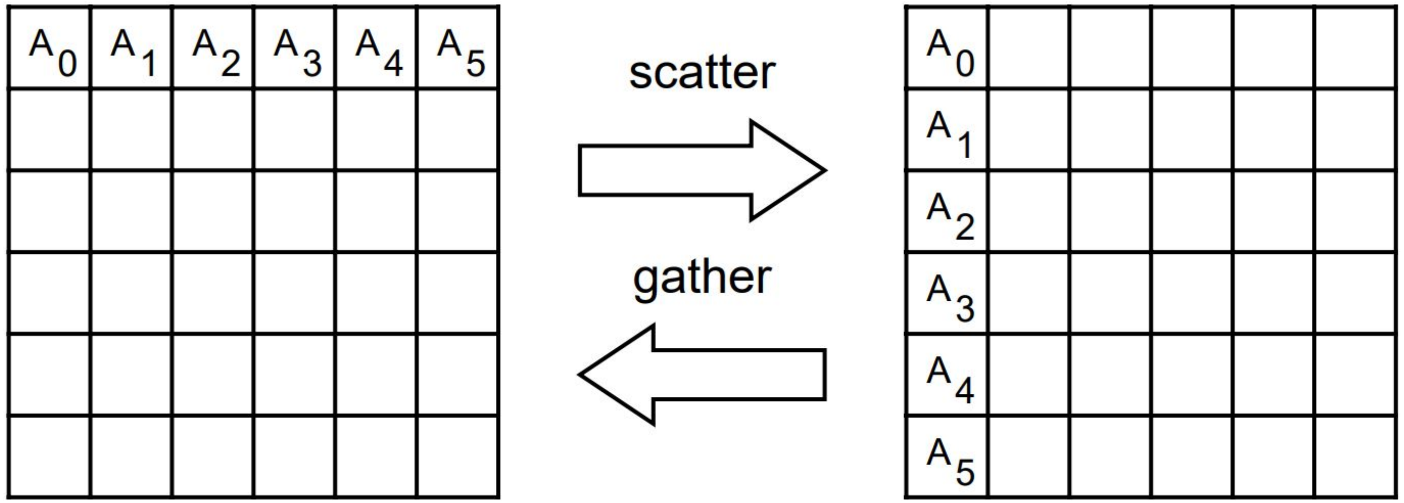
MPI_Scatter Simple Example

```c
int gsize, *sendbuf; // Size of sendbuf is 100
int root, rbuf[20];  // Assuming we have 5 processes
MPI_Comm_size(MPI_COMM_WORLD, &gsize);
sendbuf = (int *)malloc(100 * sizeof(int));

// Assuming we have 5 processes
int sendcount = 20;

MPI_Scatter(sendbuf, sendcount, MPI_INT, rbuf, sendcount,
MPI_INT, root, MPI_COMM_WORLD);
```

Gather and Scatter Operations



A_0 A_1 A_2 A_3 A_4 A_5
scatter
gather
A_0
A_1
A_2
A_3
A_4
A_5