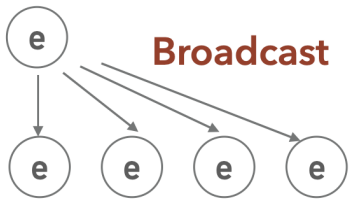
MPI Collective Communication

Michal A. Kopera

Broadcast

```
MPI_Bcast(
    void* data,
    int count,
    MPI_Datatype datatype,
    int root,
    MPI_Comm communicator)
```



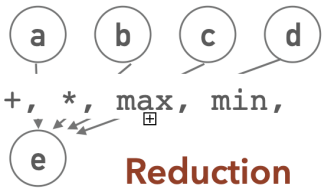
example:

```
MPI_Bcast(N, 1, MPI_INT, 0, MPI_COMM_WORLD);
```

will send 1 integer value N from rank 0 to all ranks in MPI_COMM_WORLD communicator

Reduction

```
MPI_Reduce(
    void* send_data,
    void* recv_data,
    int count,
    MPI_Datatype datatype,
    MPI_Op op,
    int root,
    MPI_Comm communicator)
```



example:

```
MPI_Reduce(&local_sum, &global_sum, 1, MPI_DOUBLE, MPI_SUM, 0,
           MPI_COMM_WORLD);
```

will sum (MPI_SUM) the values of variable local_sum from each rank to one variable global_sum stored only on rank 0. You can perform a range of reduction operations:

- MPI_MAX - Returns the maximum element.
- MPI_MIN - Returns the minimum element.

- MPI_SUM - Sums the elements.
- MPI_PROD - Multiplies all elements.

For a more complete list check online resources.

Allreduce

```
MPI_Allreduce(
    void* send_data,
    void* recv_data,
    int count,
    MPI_Datatype datatype,
    MPI_Op op,
    MPI_Comm communicator)
```

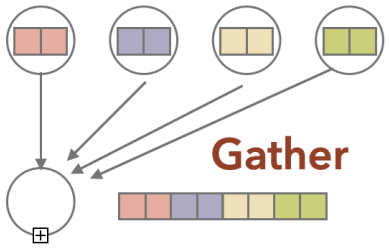
example

```
MPI_Allreduce(&local_sum, &global_sum, 1, MPI_DOUBLE, MPI_SUM,
              MPI_COMM_WORLD);
```

will perform the sum of local_sum variable, but the resulting variable global_sum will exist on all ranks in the MPI_COMM_WORLD communicator.

Gather

```
MPI_Gather(
    void* send_data,
    int send_count,
    MPI_Datatype send_datatype,
    void* recv_data,
    int recv_count,
    MPI_Datatype recv_datatype,
    int root,
    MPI_Comm communicator)
```

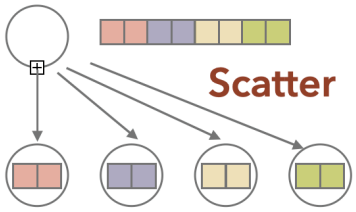


example

```
MPI_Gather(&local_a, 2, MPI_DOUBLE, global_a, 2, MPI_DOUBLE, 0,
           MPI_COMM_WORLD);
```

will gather the 2-element arrays local_a into one global array global_a on rank 0. The size of global_a should be nproc*2 for this example.

Scatter




```
MPI_Scatter(
    void* send_data,
    int send_count,
    MPI_Datatype send_datatype,
    void* recv_data,
    int recv_count,
    MPI_Datatype recv_datatype,
    int root,
    MPI_Comm communicator)
```

example

```
MPI_Scatter(big_array, 2, MPI_DOUBLE, small_chunks,
            2, MPI_DOUBLE, 0, MPI_COMM_WORLD);
```

will chop big_array on rank 0 into chunks, each 2-element large, and send each chunk to one process in the MPI_COMM_WORLD.