MPI CheatSheet

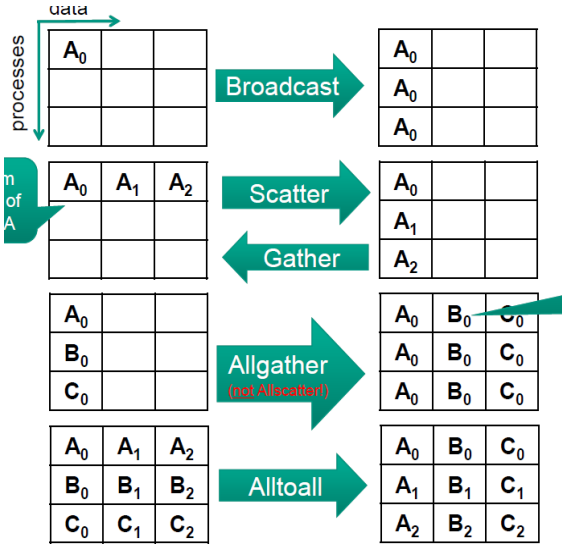
```
MPI_Comm_size(MPI_COMM_WORLD, &size);
MPI_Comm_rank(MPI_COMM_WORLD, &myrank);
```

- `MPI_Send` and `MPI_Recv`
  - both are *blocking* and *asynchronous*
    - i.e. no synchronous sending/receiving necessary
  - `MPI_Send` blocks until the message buffer can be reused
  - `MPI_Recv` blocks until message is received in the buffer *completely*

```
int MPI_Send(void* buf, int count,
             MPI_Datatype datatype, int dest,
             int tag, MPI_Comm comm)
```

Datatype of data needs to be specified explicitly.

- `buf`: the pointer to the sender's buffer
  - C/C++ uses `void*` for arguments with a "free choice" type
- `count`/`datatype`: number/type of buffer's elements
- `dest`: rank of the destination process
- `tag`: "context" of the message (e.g. a conversion ID)
- `comm`: communicator of the process group




```
int MPI_Barrier(MPI_Comm comm);
```

```
int MPI_Recv(void* buf, int count,
   MPI_Datatype datatype, int source, int tag,
   MPI_Comm comm, MPI_Status* status)
```

Wildcard possible: MPI_ANY_TAG

Wildcard possible:

```
int MPI_Bcast(void *buffer, int count, MPI_Datatype t,
              int root, MPI_Comm comm)
```

`root` is the rank of the message sender

root uses `buffer` to *provide* data

all others (i.e. receivers) use `buffer` for receiving data

- other parameters (`count`, `type`, `comm`) must be identical

`root` sends the data to itself, too

- into its part of *receive* buffer

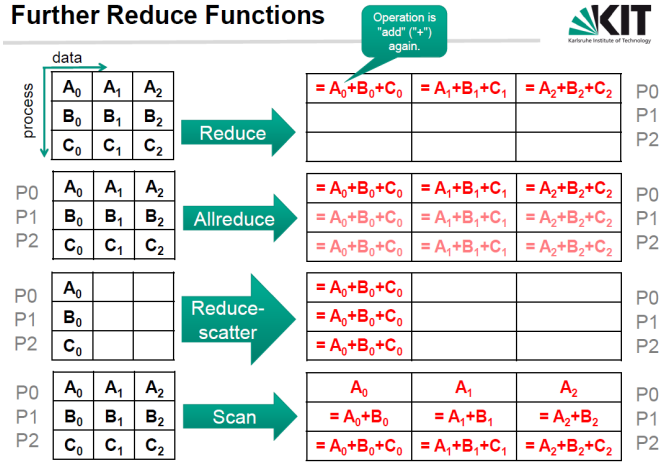
```
int MPI_Scatter (void *sendbuf,
int sendcount, MPI_Datatype sendtype, void *recvbuf,
int recvcount, MPI_Datatype recvtype, int root,
MPI_Comm comm)
```

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

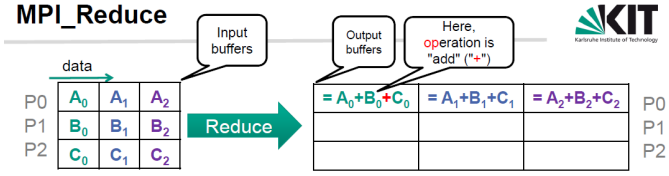
```
int MPI_Allgather (void *sendbuf,
int sendcount, MPI_Datatype sendtype, void *recvbuf,
int recvcount, MPI_Datatype recvtype, MPI_Comm comm)
```

```
int MPI_Alltoall (void *sendbuf,
int sendcount, MPI_Datatype sendtype, void *recvbuf,
int recvcount, MPI_Datatype recvtype, MPI_Comm comm)
```

Further Reduce Functions



MPI_Reduce




```
int MPI_Reduce (void *sendbuf, void *recvbuf,
                int count, MPI_Datatype type,
                MPI_Op op, int root, MPI_Comm comm)
```

- applies an operation to the data in `sendbuf` and stores the result in `recvbuf` of the root process
- `count`: number of columns in the output buffer
- `MPI_Op op` can be –
  - logical "AND" (`MPI_LAND`), bitwise "AND" (`MPI_BAND`), `MPI_LOR` / `BOR` / `LXOR` / `BXOR`, ...
  - `MPI_MAX` / `MIN` / `SUM` / `PROD` / ...
  - `MPI_MINLOC` resp. `MPI_MAXLOC` find local minimum resp. maximum and return the value of the "causing" rank
  - *own operations can also be defined*