

MPI

-
-

Copyright (c) 2012 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice and Octave.

The Butterfly Swap Operations

Communicators and Groups defines collection of processes that may communicate with each other.

Need to specify a communicator as an argument.

MPI_COMM_WORLD - predefined communicator that includes all of your MPI processes.

Within a communicator, every process has its own unique, integer identifier, called rank or “task ID”.

Used to specify the source and destination.
Also can be used in conditional statements.

MPI_Alltoall

MPI_Alltoall - Sends data from all to all processes

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

INPUT PARAMETERS

sendbuf - starting address of send buffer (choice)

sendcounts - integer array equal to the group size specifying the number of elements to send to each processor

sendtype - data type of send buffer elements (handle)

recvcounts - integer array equal to the group size specifying the maximum number of elements that can be received from each processor

recvtype - data type of receive buffer elements (handle)

comm - communicator (handle)

OUTPUT PARAMETERS

recvbuf - address of receive buffer (choice)

MPI_Alltoallv

MPI_Alltoallv - Sends data from all to all processes, with a displacement

```
int MPI_Alltoallv (void *sendbuf, int *sendcnts, int *sdispls, MPI_Datatype sendtype,  
void *recvbuf, int *recvcnts, int *rdispls, MPI_Datatype recvtype, MPI_Comm comm )
```

INPUT PARAMETERS

sendbuf - starting address of send buffer (choice)

sendcounts - integer array equal to the group size specifying the number of elements to send to each processor

sdispls - integer array (of length group size). Entry *j* specifies the displacement (relative to sendbuf from which to take the outgoing data destined for process *j*)

sendtype - data type of send buffer elements (handle)

recvcounts - integer array equal to the group size specifying the maximum number of elements that can be received from each processor

rdispls - integer array (of length group size). Entry *i* specifies the displacement (relative to recvbuf at which to place the incoming data from process *i*)

recvtype - data type of receive buffer elements (handle)

comm - communicator (handle)

OUTPUT PARAMETERS

recvbuf - address of receive buffer (choice)

MPI_Alltoallv

Alltoallv

flexibility in that the location of send data is specified by `sdispls` and the location of the placement of receive data is specified by `rdispls`.

The ***j*th block** sent from **process *i*** is received by **process *j*** and is placed in the ***ith* block**.

Need not be all the same size block

`sendcount[j]`, sendtype at **process *i***
`recvcount[i]`, recvtype at **process *j***.

The amount of data sent must be equal to the amount of data received, pairwise between every pair of processes.

Distinct type maps between sender and receiver are still allowed.

MPI_Alltoallw

ALLTOALLW in MPI-2.

Can specify separately count, displacement, and datatype.

The displacement of blocks is specified in bytes.

Can be seen as a generalization several MPI functions depending on the input arguments.

References

- [1] <http://en.wikipedia.org/>
- [2] http://static.msi.umn.edu/tutorial/scicomp/general/MPI/mpi_coll_new.html
- [3] <https://computing.llnl.gov/tutorials/mpi/>
- [4] <https://computing.llnl.gov/tutorials/mpi/>