

Introduction to Non-blocking Operations



Non-blocking Operations

- We can improve performance on many systems by overlapping communication and computation.
- **Non-blocking** operations in MPI refer to communication routines that allow a program to continue its execution without waiting for the completion of the communication operation.

Blocking Operations Review



- `int MPI_Send(const void *buf, int count, MPI_Datatype datatype, int dest, int tag, MPI_Comm comm);`
- `int MPI_Recv(void *buf, int count, MPI_Datatype datatype, int source, int tag, MPI_Comm comm, MPI_Status *status);`

Non-blocking Operations

- In contrast to blocking operations, where the program halts until the communication is finished, non-blocking operations enable overlap of computation and communication.
- The program can continue with other computations while the communication is in progress.

Non-blocking Operations

- They are **asynchronous**: Non-blocking operations initiate communication and then return control to the program immediately without waiting for the communication to complete.

Non-blocking Operations

- `int MPI_Isend(const void *buf, int count, MPI_Datatype datatype, int dest, int tag, MPI_Comm comm, MPI_Request *request);`
- `int MPI_Irecv(void *buf, int count, MPI_Datatype datatype, int source, int tag, MPI_Comm comm, MPI_Request *request);`

MPI_Isend

- Initiates the sending of a message but does not wait for the completion of the communication. It allows the program to continue with other computations or communication operations while the data is being sent in the background.

- `int MPI_Isend(const void *buf, int count, MPI_Datatype datatype, int dest, int tag, MPI_Comm comm, MPI_Request *request);`

MPI_Irecv

- Initiates the receiving of a message but does not wait for the completion of the communication. It allows the program to continue with other computations or communication operations while waiting for incoming data in the background.
- `int MPI_Irecv(void *buf, int count, MPI_Datatype datatype, int source, int tag, MPI_Comm comm, MPI_Request *request);`