

Assignment 2

Title

To develop any distributed application using Message Passing Interface (MPI).

Objectives

By the end of this assignment, the student will be able to implement any distributed applications based on MPI.

Tools

MPJ Express Software

Theory

Message Passing Interface

- Message passing is a popularly renowned mechanism to implement parallelism in applications; it is also called MPI.
- A basic prerequisite for message passing is a good communication API.
- MPJ Express is a message passing library that can be used by application developers to execute their parallel Java applications on compute clusters or networks of computers.
- MPJ is a familiar Java API for MPI implementation.
- MPJ Express is essentially a middleware that supports communication between individual processors of clusters.
- The programming model followed by MPJ Express is Single Program Multiple Data (SPMD).
- MPJ Express is designed for distributed memory machines like network of computers or clusters, it is possible to efficiently execute parallel user applications on desktops or laptops that contain shared memory or multicore processors.

MPJ Express Configuration

- The MPJ Express software can be configured in two ways as shown in Figure.
- Multicore configuration—is used to execute MPJ Express user programs on laptops and desktops.
- The cluster configuration—is used to execute MPJ Express user programs on clusters or networks of computers.

Multicore configuration

- The multicore configuration is meant for users who plan to write and execute parallel Java applications using MPJ Express on their desktops or laptops—typically such hardware contains shared memory and multicore processors.
- In this configuration, users can write their message passing parallel applications using MPJ Express and it will be ported automatically on multicore processors. Also this configuration is preferred for teaching purposes since students can execute message passing code on their personal laptops and desktops. The user applications stay the same when executing the code in multicore or cluster configuration.

MPI Environment

- MPI is for communication among processes, which have separate address spaces.
- Group is the set of processes that communicate with one another.
- Communicator is the central object for communication in MPI.
- There is a default communicator whose group contains all initial processes, called `MPI_COMM_WORLD`.
- Every MPI program must contain `import mpi.MPI`
- `MPI_Init` initializes the execution environment for MPI.
- A process is identified by its rank in the group associated with a communicator.
- How many processes are participating in this computation?
 - `MPI_Comm_size` function reports the number of processes.
 - `MPI_Comm_rank` function reports the rank, a number between 0 and size-1, identifying the calling process.

- MPI_Finalize cleans up all the extraneous mess that was first put into place by MPI_Init.

Conclusion

Thus in this assignment we've learnt about MPI and implemented a program in JAVA using it.