Assignment 2

[Roll no 43212]

Title: To develop any distributed application using passing interface (MPI).

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

Theory:

Message Passing Interface:

Message passing is a popularly renowned mechanism to implement parallelism in applications. It is also called as MPI.

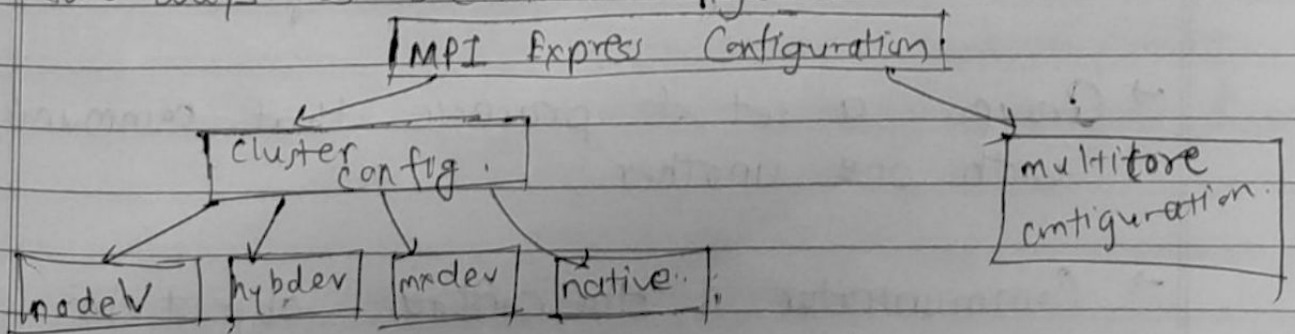
MPI is a message passing mechanism that can be used by application developers to execute their parallel java applications on compute clusters or network of computers.

MPI is familiar Java API for MPI implementation.

The programming model followed by MPI Express is single program multiple data (SPMD).

MPI with Java:

MPI express software can be configured in two ways as shown in figure.



Multicore configuration is used to execute MPI Express user programs on laptops & desktops.
The cluster configuration is used to execute MPI Express user programs on cluster or network of computer.

Installing MPI Express:

1) Download MPI express (mpi.jar) & unpack it.
2) Set environment variables: MPI-HOME & PATH.
   - Export MPI-HOME = /path/to/MPI.
   - Export PATH = $MPI-HOME/bin : $PATH
3) Create a new working directory for MPI express programs.
   e.g. /mpj - user directory.
4) Compile the MPI Express library: ~~cd~~
   cd $MPJ-HOME; ant.

MPI Environment:

→ MPI is for communication among processes, which have separate address space.

→ Group is a set of processes that communicate with one another.

→ Communicator is the central object for communication in MPI.

[Roll no 43212]

→ There is a default communicator where group contains all initial process called MPI_COMM_WORLD.

→ Every MPI Program must contain import mpi.MPI

→ MPI_Init initializes the execution enviroment for MPI

→ A process is identified by its rank in the group associated with a communicator.

→ MPI_finalize cleans up all the mess that was first put onto place by MPI_init.

- Steps for compilation & execution:-
  - Installing MPI - Express program in the multicore configuration.
  - 1.> Download MPI Express (mpj.jar) & unpack it.

2. Set MPJ_HOME and path environment variable

export : MPJ_HOME = /path/to/MPJ
export PATH = $MPJ_HOME/bin : $path

Roll no 43212

- compile
  javac -cp $MPJ_HOME/lib/mpj.jar program.java

- Execute
  $ MPJ_HOME/bin/mpjrun.sh -np 4 program

Conclusion:

In this assignment, I installed MPJ Express & created a program for sending & receiving prime numbers using MPJ.