RollNo.:43159

Name:Sakshi Tantak

Batch:P11

Assignment2:Todevelopanydistributedapplication(dateandtime)usingMessagePassing Interface (MPI).

**Problem statement:** get the date and time string by passing it to multiple processes using MPI.

**Objectives:** Students will be able to implement any distributed application based on MPI. **Tools:** JDK v 1.8+, MPJ Express (v0.44)

**Theory:**

**Message Passing Interface**

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

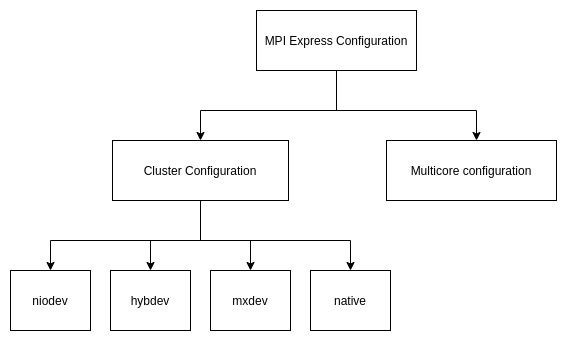
* A basic prerequisite for message passing is a good communication API.
* MPJ Express is a message passing library that can be used by application developers to execute their parallel Java applications on compute clusters or a network of computers.
* MPJ is a familiar Java API for MPI implementation.
* 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 a 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.
* As we do not have access to a network or computers, similar simulation can be made on local machines by making use of multiprogramming processors.

**MPJ: MPI using Java**

**MPJ Express configuration:**

The MPJ Express software can be configured in two ways as shown in Figure.

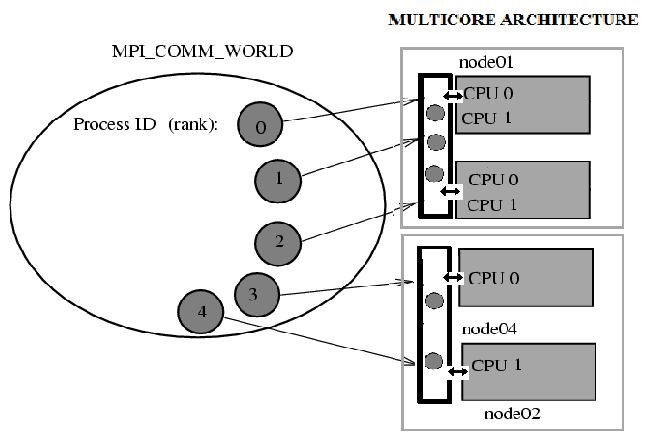
1. Multicore configuration—is used to execute MPJ Express user programs on laptops and desktops ( Which we will be using )
2. The cluster configuration—is used to execute MPJ Express user programs on clusters or a network of computers.

**MPI Multi**

**Core**

**configuration:**

* Meant for the users who plan to execute the parallel Java applications using MPI in their local machines, like laptops or desktops - typically the hardware that contains shared memory and multicore processors.
* In this configuration, users can write their application which passes messages parallel using MPJ express and it will be ported automatically on the multicore processors.



**How to install MPJ Express for use?**

1. Download MPJ Express from its site
2. Untar the tarball using tar -xvfz mpj-v0\_44.tar.gz
3. Copy the content to a new folder (Optional )
4. Set environment variables $MPJ\_HOME and PATH
5. export MPJ\_HOME=/path/to/mpj/
6. export PATH=$MPJ\_HOME/bin:$PATH

5 . Compile the MPJ Express Library using ant (Need to install ant using SNAP )

a. Cd $MPJ\_HOME; ant

**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 is there to cleanup all the MPI calls and variables.

**Communication collective**

**Collective Operations in MPI:**

●Collective communications refer to the set of MPI functions that transmit data

among all processes specified by a given communicator.

* Send-Recv: Ability to Send data items in a message into multiple memory

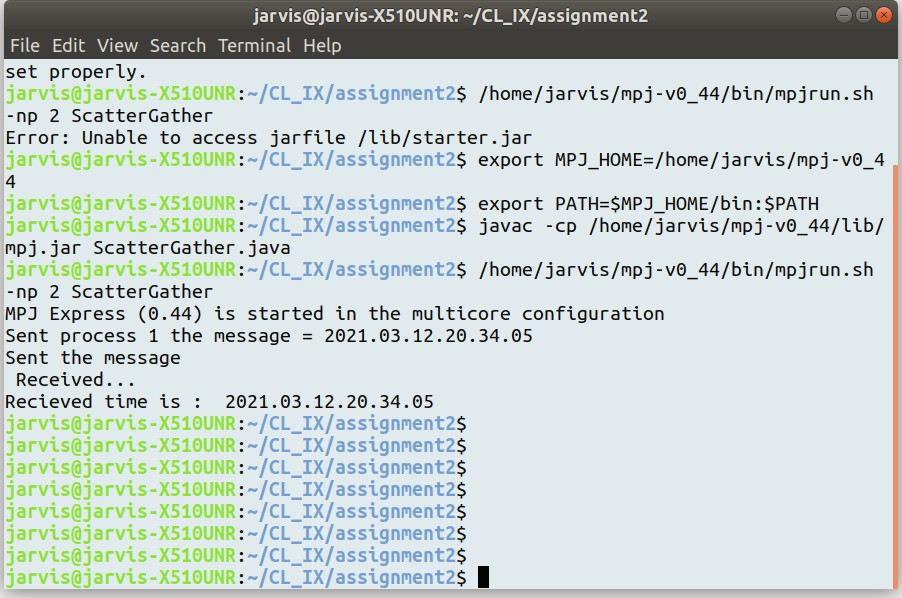
locations and “Recv” (Receive) data items from multiple memory locations into one message.

* In MPI\_Gather, each process sends the content of “send” buffer to the root process. Root receives and stores in rank order.
* IN sendbuf starting address of send buffer.
* OUT recvbuf (address of receive buffer).

**Implementation of Date and time service using MPI:**

1. Create DateTime.java file and create public class DateTime having main function in it.
2. Import mpi.MPI at the beginning of the file.
3. Initially call Init method of MPI.
4. We can get rank and size using the internal futputunction of MPI.
5. Here we send the message in the form of date and time string and receive it later and finally display it.
6. Finally finalize method is used to terminate the program.

**Output**



**Conclusion:**

Thus, in this assignment, I learned how to use MPI in Java using MPJ Express for distributed computing systems and parallel processing also implemented date and time application using it.