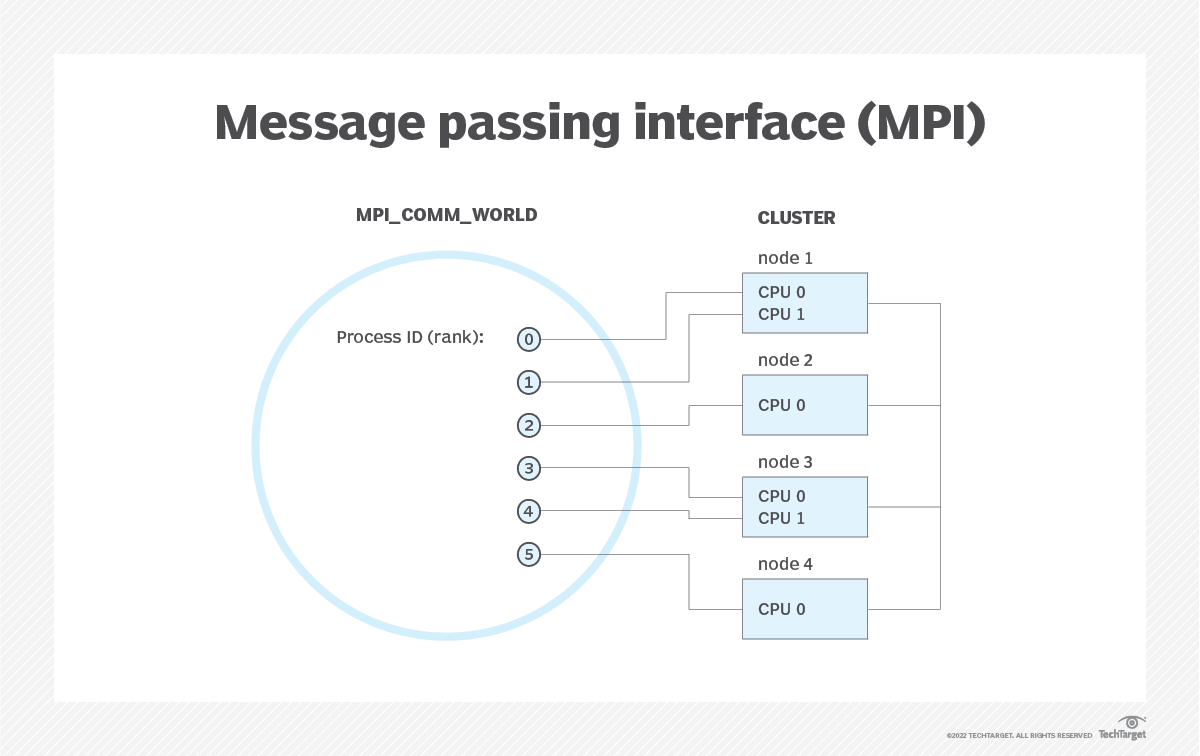
**EXPERIMENT 1**

**AIM**: Write Hello World using MPI

**THEORY**:

The Message Passing Interface (MPI) is a portable, standardized message-passing standard that functions on parallel computing architectures. The MPI system requires the syntax and semantics of library routines that can be used by a broad variety of users who are writing portable message-passing programs in C, C++, and Fortran.

There are many open-source MPI implementations that have aided in the development of the parallel software industry and the development of portable and scalable large-scale parallel applications.



Source: https://www.techtarget.com/searchenterprisedesktop/definition/message-passing-interface-MPI

**CPP IMPLEMENTATION** :

**CODE:**

| #include <mpi.h>  #include <stdio.h>  int main(int argc, char\*\* argv) {  // Initialize the MPI environment  MPI\_Init(NULL, NULL);  // Get the rank of the process  int my\_rank;  MPI\_Comm\_rank(MPI\_COMM\_WORLD, &my\_rank);  // Print the message  printf("Hello World! My rank is %d\n", my\_rank);  // Finalize the MPI environment.  MPI\_Finalize();  } |
| --- |

**OUTPUT:**

| Build started...  1>------ Build started: Project: Project1, Configuration: Debug x64 ------  1>MPI.cpp  ========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========  ========== Build started at 4:28 PM and took 04.288 seconds ========== |
| --- |

| C:\Users\JARVIS\source\repos\Project1\x64\Debug>mpiexec -n 4 Project1.exe  Hello World! My rank is 1  Hello World! My rank is 0  Hello World! My rank is 2  Hello World! My rank is 3 |
| --- |

**PYTHON IMPLEMENTATION:**

**CODE:**

| from mpi4py import MPI  comm = MPI.COMM\_WORLD  rank = comm.Get\_rank()  if rank == 0:  data = {'data' : 'Hello Junaid'}  else:  data = None  data = comm.bcast(data, root=0)  print(data) |
| --- |

**OUTPUT**:

| jarvis@jarvis-Inspiron-7591:~/Desktop$ mpiexec -n 4 python3 main.py  {'data': 'Hello Junaid'}  {'data': 'Hello Junaid'}  {'data': 'Hello Junaid'}  {'data': 'Hello Junaid'} |
| --- |

**CONCLUSION:** MPI is an important functionality that is used in distributed processing to achieve higher efficiency by sharing data between different processes. In this experiment, we have implemented the MPI mechanism using C++ and Python.