Exp 10a INSTALLATION OF MPI AND EXECUTION OF SAMPLE PROGRAMS
23/09/24

Aim:

To install Message Passing Interface(MPI) in linux and execute the sample programs.

Procedure:

Step1: Run update and upgrade commands on ubuntu.

```
tce@tce-VirtualBox:~$ sudo apt update
[sudo] password for tce:
Get:1 https://dl.google.com/linux/chrome/deb stable InRelease [1,825 B]
Hit:2 http://in.archive.ubuntu.com/ubuntu bionic InRelease
Err:1 https://dl.google.com/linux/chrome/deb stable InRelease
  The following signatures couldn't be verified because the public key is not a
vailable: NO_PUBKEY E88979FB9B30ACF2
Hit:3 http://in.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:4 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:5 http://in.archive.ubuntu.com/ubuntu bionic-backports InRelease
```

```
tce@tce-VirtualBox:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  gir1.2-goa-1.0 gir1.2-snapd-1
Use 'sudo apt autoremove' to remove them.

Enable UA Infra: ESM to receive additional future security updates.
See https://ubuntu.com/18-04 or run: sudo ua status

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

The following NEW packages will be installed:
  linux-headers-5.4.0-150-generic linux-hwe-5.4-headers-5.4.0-150
  linux-image-5.4.0-150-generic linux-modules-5.4.0-150-generic
```

Step2: Install MPI Environment

```
tce@tce-VirtualBox:~$ sudo apt install -y openmpi-bin openmpi-common libopenmpi
-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  gir1.2-goa-1.0 gir1.2-snapd-1
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  autotools-dev ibverbs-providers libfabric1 libhwloc-dev libibverbs-dev
  libibverbs1 libltdl-dev libnl-route-3-200 libnuma-dev libopenmpi2
  libpsm-infinipath1 librdmacm1 libtool
Suggested packages:
  libtool-doc openmpi-doc autoconf automaken gcj-jdk
The following NEW packages will be installed:
  autotools-dev ibverbs-providers libfabric1 libhwloc-dev libibverbs-dev
  libibverbs1 libltdl-dev libnl-route-3-200 libnuma-dev libopenmpi-dev
  libopenmpi2 libpsm-infinipath1 librdmacm1 libtool openmpi-bin
```

```
tce@tce-VirtualBox:~$ mpirun --version
mpirun (Open MPI) 2.1.1

Report bugs to http://www.open-mpi.org/community/help/
tce@tce-VirtualBox:~$
```

Step3: Install python3 and pip: *sudo apt install python3 python3-pip*

```
tce@tce-VirtualBox:~$ sudo apt install -y python3 python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3 is already the newest version (3.6.7-1~18.04).
python3-pip is already the newest version (9.0.1-2.3~ubuntu1.18.04.8).
The following packages were automatically installed and are no longer required:
  gir1.2-goa-1.0 gir1.2-snapd-1
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
tce@tce-VirtualBox:~$ python3 --version
Python 3.6.9
tce@tce-VirtualBox:~$
```

Step4: Install mpi4py: *pip install mpi4py*

```
tce@tce-VirtualBox:~$ pip3 install mpi4py
WARNING: pip is being invoked by an old script wrapper. This will fail in a fut
ure version of pip.
Please see https://github.com/pypa/pip/issues/5599 for advice on fixing the und
erlying issue.
To avoid this problem you can invoke Python with '-m pip' instead of running pi
p directly.
Defaulting to user installation because normal site-packages is not writeable
Collecting mpi4py
  Using cached mpi4py-4.0.0.tar.gz (464 kB)
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Installing backend dependencies ... done
  Preparing metadata (pyproject.toml) ... done
Building wheels for collected packages: mpi4py
```

```
tce@tce-VirtualBox:~$ python3 -m mpi4py --version
mpi4py 4.0.0
tce@tce-VirtualBox:~$
```

Step5: Once the environment is set, Open a editor and write a parallel python scripts.

PROGRAM1:

Open ▾ b.py ~/mpi Save

```
from mpi4py import MPI

comm = MPI.COMM_WORLD
rank = comm.Get_rank()

if rank == 0:
    data = {'key1': 7, 'key2': 3.14}
else:
    data = None

# Broadcast data from process 0 to all other processes
data = comm.bcast(data, root=0)
print(f"Process {rank} received data: {data}")
```

OUTPUT1:

```
tce@tce-VirtualBox:~/mpi$ mpirun -np 4 python3 b.py
Process 0 received data: {'key1': 7, 'key2': 3.14}
Process 2 received data: {'key1': 7, 'key2': 3.14}
Process 1 received data: {'key1': 7, 'key2': 3.14}
Process 3 received data: {'key1': 7, 'key2': 3.14}
```

PROGRAM2:

Open ▾ g.py ~/mpi Save

```
from mpi4py import MPI

comm = MPI.COMM_WORLD
rank = comm.Get_rank()

# Each process has its own data
data = f"Process {rank} data"

# Allgather: every process sends its data to every other process
gathered_data = comm.allgather(data)
print(f"Process {rank} received: {gathered_data}")
```

OUTPUT2:

```
tce@tce-VirtualBox:~/mpi$ mpirun -np 4 python3 g.py
Process 3 received: ['Process 0 data', 'Process 1 data', 'Process 2 data', 'Pro
cess 3 data']
Process 0 received: ['Process 0 data', 'Process 1 data', 'Process 2 data', 'Pro
cess 3 data']
Process 1 received: ['Process 0 data', 'Process 1 data', 'Process 2 data', 'Pro
cess 3 data']
Process 2 received: ['Process 0 data', 'Process 1 data', 'Process 2 data', 'Pro
cess 3 data']
tce@tce-VirtualBox:~/mpi$
```

PROGRAM3:

Open ▾ s.py ~/mpi Save

```
from mpi4py import MPI

comm = MPI.COMM_WORLD
rank = comm.Get_rank()
size = comm.Get_size()

if rank == 0:
    data = list(range(size))  # Data to be scattered: [0, 1, 2, ...]
else:
    data = None

# Scatter the data to all processes
recv_data = comm.scatter(data, root=0)
print(f"Process {rank} received {recv_data}")
```

OUTPUT3:

```
tce@tce-VirtualBox:~/mpi$ mpirun -np 4 python3 s.py
Process 0 received 0
Process 3 received 3
Process 1 received 1
Process 2 received 2
```

Result:

Thus the installation of Message Passing Interface(MPI) in linux and execution of sample programs has been executed successfully and output has been verified.