

Distributed System

Practical Work 3: MPI File Transfer

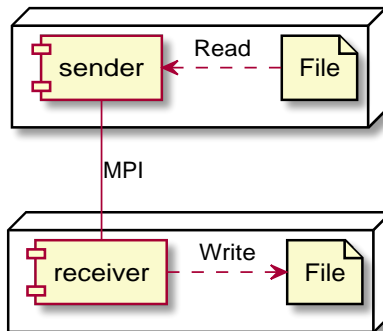
Group 1

March 23, 2021

1 Design

For this practical we used mpi4py as the Python binding for MPI, which is implementation independent. The group members choose to use MPICH to run the program because it is available on the majority of GNU/Linux distributions.

The service was designed as a simplified version of the RPC file transfer.



Unlike the RPC file transfer, file discovery was not considered due to the difficulties of setting up MPI across different machines for it to be useful.

The file system was organized as follows:

```
└─ logo-usth-pa3-01.png
└─ mpi
    └─ mpi.py
```

2 Implementation

The MPI communication was set up as follows:

```
from mpi4py import MPI

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

The process of rank 0 was then used for sending the file, which was solely passing the binary data to `send`:

```
if (rank == 0):
    f = open("../logo-usth-pa3-01.png", "rb")
    data = f.read()
    comm.send(data, dest=1)
```

The process of rank 1 was used for receiving the file, which was solely writing the binary data from `recv` to disk:

```
elif (rank == 1):
    data = comm.recv(source=0)
    f = open("logo.png", "wb")
    f.write(data)
```

3 Credits

The MPI file transfer was designed and implemented by Ngô Xuân Minh, and this article was written by Nguyễn Gia Phong.