Practical Work 3: MPI File Transfer

Your Name - ID

December 6, 2025

1 Choice of MPI Implementation

For this practical work, we selected **MPI (Message Passing Interface)** as the communication protocol. Specifically, we used the **`mpi4py`** library in Python.

Reasons for choice:

- **Standardization:** MPI is the de-facto standard for high-performance computing (HPC) and distributed systems.
- **Python Integration:** `mpi4py` provides Pythonic bindings for the MPI standard, allowing us to send complex Python objects (like strings) or raw buffers (like file chunks) easily.
- **SPMD Model:** It allows us to manage separate processes (Sender and Receiver) within a single codebase based on their Rank ID.

2 System Design & Organization

The system follows the **SPMD (Single Program, Multiple Data)** architecture. A single script is executed, but behavior diverges based on the process **Rank**.

- **Rank 0 (Sender):** Reads the file from disk and sends it to Rank 1.
- **Rank 1 (Receiver):** Listens for messages from Rank 0 and writes data to disk.

We use **MPI Tags** to distinguish message types in the stream:

- `Tag 1`: Filename metadata.
- `Tag 2`: File binary content (Chunks).
- `Tag 3`: End of transmission signal (EOF).

3 Implementation Code Snippets

3.1 Rank 0: Sender Logic

The sender reads the file and uses `comm.send` with specific tags.

```python
if rank == 0:
    # 1. Send Filename
    comm.send(filename, dest=1, tag=1)

    # 2. Send Data Chunks
    with open(filename, 'rb') as f:
        while True:
            chunk = f.read(4096)
            if not chunk: break
            comm.send(chunk, dest=1, tag=2)

    # 3. Send EOF
    comm.send(None, dest=1, tag=3)
```

Listing 1: Sender implementation

3.2 Rank 1: Receiver Logic

The receiver waits for any message (`MPI.ANY_TAG`) and switches behavior based on the received tag.

```python
elif rank == 1:
    while True:
        status = MPI.Status()
        # Receive data and get status
        data = comm.recv(source=0, tag=MPI.ANY_TAG, status=status)
        tag = status.Get_tag()

        if tag == 1:        # Filename Tag
            filename = data
            f = open("received_" + filename, "wb")
        elif tag == 2:      # Data Tag
            f.write(data)
        elif tag == 3:      # End Tag
            f.close()
            break
```

Listing 2: Receiver implementation