Temporosa, Jullie Anne A.
J3A

1. Why is message passing required in distributed systems?
   - In distributed systems, multiple processes can run concurrently on computers or nodes, each with its own local memory. Processes must communicate explicitly with each other to coordinate, exchange, and share data and results.

     In this activity, worker processes compute sums of their data chunks, they send messages to the master using comm.send and the master receives messages to aggregate results using comm.recv.

     Message passing is the sending and receiving between processes, without it, distributed processes cannot exchange information, and the system would not function properly.

2. What happens if one process fails?
   - If a worker process fails, the master may hang waiting for a message, resulting to comm.recv never completing. Also, other workers may continue computing, but the final result is incomplete because there is no automatic fault tolerance.

3. How does this model differ from shared-memory programming?
   - Message passing or Distributed system's process has its own local memory and has explicit communication, white Shared-memory or Single machine's processes share a common memory space and has implicit read/write to shared variables.

     In simple terms, Message passing works across distributed nodes without shared memory but Shared-memory is simple but only works within a single system.

Trial 1: Error when using .ipynb only - Jupyter

distributed_task.ipynb > import ipyparallel as ipp

Generate   + Code   + Markdown   ▷ Run All   ↻ Restart   ✕ Clear All Outputs   Jupyter Variables   ☰ Outline   ···

```python
import ipyparallel as ipp
cluster = ipp.Client()
view = cluster[:] # Get a view of all engines

%%px
from mpi4py import MPI

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

if rank == 0:
    print(f"Master (Rank 0) reporting. Coordinating {size-1} workers.\n")

    results = []
    for i in range(1, size):
        data = comm.recv(source=i)
        results.append(data)
        print(f"[RECEIVED] Worker{data['rank']} finished Task {data['chunk_id']}.")
        print(f"            Result: {data['result']}\n")
    print(f"\n[FINAL] All tasks completed. Grand total: {sum(r['result'] for r in results)}")

else:
    start_val = rank * 100
    end_valu = start_val + 99

    computation = sum(range(start_val, end_valu + 1))
    payload ={
        "rank": rank
```

Result:

```
PS C:\Users\User\github-classroom\UPHSL-PDC2026\assignment-2-missanning> mpiexec -n 4 python distributed_task.ipynb
Traceback (most recent call last):
Traceback (most recent call last):
Traceback (most recent call last):
Traceback (most recent call last):
  File "C:\Users\User\github-classroom\UPHSL-PDC2026\assignment-2-missanning\distributed_task.ipynb", line 5, in <module>
    "execution_count": null,
                       ^^^^
NameError: name 'null' is not defined
  File "C:\Users\User\github-classroom\UPHSL-PDC2026\assignment-2-missanning\distributed_task.ipynb", line 5, in <module>
    "execution_count": null,
                       ^^^^
NameError: name 'null' is not defined
  File "C:\Users\User\github-classroom\UPHSL-PDC2026\assignment-2-missanning\distributed_task.ipynb", line 5, in <module>
    "execution_count": null,
                       ^^^^
NameError: name 'null' is not defined
  File "C:\Users\User\github-classroom\UPHSL-PDC2026\assignment-2-missanning\distributed_task.ipynb", line 5, in <module>
    "execution_count": null,
                       ^^^^
NameError: name 'null' is not defined
```

Trial 2: Converted the .ipynb file to .py file - Jupyter

```python
%pip install mpi4py ipyparallel
%load_ext ipyparallel
```
Python

UsageError: Line magic function `%%px` not found.

```python
import ipyparallel as ipp
cluster = await ipp.Cluster(engines="mpi", n=4).start_cluster()
client = cluster.connect_client()
view = client[:]
```
Python

distributed_task.ipynb > ⬩ import ipyparallel as ipp

✦ Generate ＋ Code ＋ Markdown ▷ Run All ↻ Restart ⤫ Clear All Outputs 🔲 Jupyter Variables ☰ Outline ⋯

```python
%%px
from mpi4py import MPI

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

if rank == 0:
    print(f"Master (Rank 0) reporting. Coordinating {size-1} workers.\n")

    results = []
    for i in range(1, size):
        data = comm.recv(source=i)
        results.append(data)
        print(f"[RECEIVED] Worker{data['rank']} finished Task {data['chunk_id']}.")
        print(f"           Result: {data['result']}\n")
    print(f"\n[FINAL] All tasks completed. Grand total: {sum(r['result'] for r in results)}")

else:
    start_val = rank * 100
    end_valu = start_val + 99

    computation = sum(range(start_val, end_valu + 1))
    payload ={
        "rank": rank,
        "chunk_id": f"Data-Range={start_val}-to-{end_valu}",
        "result": computation
```

PS C:\Users\User\github-classroom\UPHSL-PDC2026\assignment-2-missanning> jupyter nbconvert --to script mpi_master_worker.ipynb
>>
[NbConvertApp] Converting notebook mpi_master_worker.ipynb to script
[NbConvertApp] Writing 848 bytes to mpi_master_worker.txt
PS C:\Users\User\github-classroom\UPHSL-PDC2026\assignment-2-missanning>
[NbConvertApp] Converting notebook mpi_master_worker.ipynb to script
[NbConvertApp] Writing 848 bytes to mpi_master_worker.txt
[NbConvertApp] Converting notebook mpi_master_worker.ipynb to script
[NbConvertApp] Writing 848 bytes to mpi_master_worker.txt

It creates a new file:

Welcome | distributed_task.ipynb M ✕ | ⬩ mpi_master_worker.py U ✕ | ⬩ Assignment2.py

⬩ mpi_master_worker.py > ...

```python
import asyncio
import sys

if sys.platform.startswith("win"):
    asyncio.set_event_loop_policy(asyncio.WindowsSelectorEventLoopPolicy())

from mpi4py import MPI

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

print(f"Process {rank} of {size} started")

if rank == 0:
    # MASTER PROCESS
    print("\nMaster process running...\n")

    for i in range(1, size):
        message = comm.recv(source=i)
        print(f"Received from process {i}:")
        print(f"  Task: {message['task']}")
        print(f"  Result: {message['result']}\n")

else:
    # WORKER PROCESS
    data_chunk = list(range(rank * 10, rank * 10 + 10))
    result = sum(data_chunk)

    message = {
```

```python
if rank == 0:
    # MASTER PROCESS
    print("\nMaster process running...\n")

    for i in range(1, size):
        message = comm.recv(source=i)
        print(f"Received from process {i}:")
        print(f"  Task: {message['task']}")
        print(f"  Result: {message['result']}\n")

else:
    # WORKER PROCESS
    data_chunk = list(range(rank * 10, rank * 10 + 10))
    result = sum(data_chunk)

    message = {
        "rank": rank,
        "task": f"Sum of numbers {data_chunk[0]} to {data_chunk[-1]}",
        "result": result
    }

    comm.send(message, dest=0)
```

Result:

```
PS C:\Users\User\github-classroom\UPHSL-PDC2026\assignment-2-missanning> & C:/Users/User/AppData/Local/Programs/Python/Python313/python.exe c:
ser/github-classroom/UPHSL-PDC2026/assignment-2-missanning/mpi_master_worker.py
Process 0 of 1 started

Master process running...

PS C:\Users\User\github-classroom\UPHSL-PDC2026\assignment-2-missanning> mpiexec -n 4 python mpi_master_worker.py
Process 3 of 4 started
Process 2 of 4 started
Process 1 of 4 started
Process 0 of 4 started

Master process running...

Received from process 1:
  Task: Sum of numbers 10 to 19
  Result: 145

PS C:\Users\User\github-classroom\UPHSL-PDC2026\assignment-2-missanning> mpiexec -n 4 python mpi_master_worker.py
Process 3 of 4 started
Process 2 of 4 started
Process 1 of 4 started
Process 0 of 4 started

Master process running...

Received from process 1:
  Task: Sum of numbers 10 to 19
  Result: 145

Process 3 of 4 started
Process 2 of 4 started
Process 1 of 4 started
Process 0 of 4 started

Master process running...
```

```
Master process running...

Received from process 1:
  Task: Sum of numbers 10 to 19
  Result: 145

Process 2 of 4 started
Process 1 of 4 started
Process 0 of 4 started

Master process running...

Received from process 1:
  Task: Sum of numbers 10 to 19
  Result: 145

Process 0 of 4 started

Master process running...

Received from process 1:
  Task: Sum of numbers 10 to 19
  Result: 145

Master process running...

Received from process 1:
  Task: Sum of numbers 10 to 19
  Result: 145

  Task: Sum of numbers 10 to 19
  Result: 145

Received from process 2:
  Result: 145
```

```
Received from process 1:
  Task: Sum of numbers 10 to 19
  Result: 145

  Task: Sum of numbers 10 to 19
  Result: 145

Received from process 2:
  Result: 145

Received from process 2:
  Task: Sum of numbers 20 to 29
Received from process 2:
  Task: Sum of numbers 20 to 29
  Task: Sum of numbers 20 to 29
  Result: 245

Received from process 3:
  Task: Sum of numbers 30 to 39
  Result: 345

PS C:\Users\User\github-classroom\UPHSL-PDC2026\assignment-2-missanning>
  Result: 345

  Result: 345

PS C:\Users\User\github-classroom\UPHSL-PDC2026\assignment-2-missanning>

  Result: 345

PS C:\Users\User\github-classroom\UPHSL-PDC2026\assignment-2-missanning>

  Result: 345
```

Trial 3: FINAL RUN - Google Colab

```
!apt-get update
!apt-get install -y mpich
!pip install mpi4py
```

```
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 https://cloud.r-project.org/bin/linux/ubuntu jammy-cran40/ InRelease
Hit:3 https://cli.github.com/packages stable InRelease
Hit:4 https://r2u.stat.illinois.edu/ubuntu jammy InRelease
Hit:5 http://archive.ubuntu.com/ubuntu jammy InRelease
Hit:6 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:7 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:8 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu jammy InRelease
Hit:9 https://ppa.launchpadcontent.net/ubuntugis/ppa/ubuntu jammy InRelease
Reading package lists... Done
W: Skipping acquire of configured file 'main/source/Sources' as repository 'https://r2u.stat.illinois.edu/ubuntu jammy InRelease' does not seem to provide it (sources.list entry misspelt?)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
mpich is already the newest version (4.0-3).
0 upgraded, 0 newly installed, 0 to remove and 63 not upgraded.
Requirement already satisfied: mpi4py in /usr/local/lib/python3.12/dist-packages (4.1.1)
```

```python
%%writefile mpi_master_worker.py
from mpi4py import MPI

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

if rank == 0:
    # Master process
    print(f"Master process started. Expecting {size-1} worker messages...\n")

    for i in range(1, size):
        message = comm.recv(source=i)
        print(
            f"Received from process {i} | "
            f"Task: {message['task']} | "
            f"Result: {message['result']}"
        )

    print("\nAll worker results received.")

else:
    # Worker processes
    data_chunk = list(range(rank * 10, rank * 10 + 10))
    computed_sum = sum(data_chunk)

    message = {
        "rank": rank,
        "task": f"Data chunk {rank}",
        "result": computed_sum
    }

    comm.send(message, dest=0)
```

```
Overwriting mpi_master_worker.py
```

Result:

```
!mpiexec --allow-run-as-root --oversubscribe -n 4 python mpi_master_worker.py
```

Master process started. Expecting 3 worker messages...

Received from process 1 | Task: Data chunk 1 | Result: 145
Received from process 2 | Task: Data chunk 2 | Result: 245
Received from process 3 | Task: Data chunk 3 | Result: 345

All worker results received.