Github link : <https://github.com/ramaraosm/mpi_matrix_multiplication.git>

Below document explains steps to execute matrix multiplication on two different ec2 machines on which mpi4py is installed. The time taken when run using mpi and serial mode are captured.

Step1:launch 2 ec2 instances with ubuntu 22 , sg -ssh 22 and custom tcp with port range and a keypair

Step2 :connect to the instances and update and install commands

sudo apt update

sudo apt install -y python3-numpy python3-matplotlib python3-pandas python3-pip

sudo apt install -y mpich python3-mpi4py

step3: ssh-keygen -t rsa

cat ~/.ssh/id\_rsa.pub

copy the pub content into ~/.ssh/authorized\_keys of node2

step4:check connection

ssh [ubuntu@3.110.49.184](mailto:ubuntu@3.110.49.184)

step 5:

on node1 add the ip address to /etc/hosts file

step6:create a file matrix\_mpi\_benchmark.py

from mpi4py import MPI

import numpy as np

import time

import csv

import os

comm = MPI.COMM\_WORLD

rank = comm.Get\_rank()

size = comm.Get\_size()

# Configurable matrix size

N = 512 # Adjust based on memory & benchmarking goals

# File to log performance metrics

CSV\_FILE = "mpi\_benchmark\_results.csv"

os.makedirs("logs", exist\_ok=True)

CSV\_PATH = os.path.join("logs", CSV\_FILE)

# Root initializes matrices

A, B = None, None

if rank == 0:

A = np.random.randint(0, 10, (N, N), dtype=np.int32)

B = np.random.randint(0, 10, (N, N), dtype=np.int32)

# Broadcast B

B = comm.bcast(B if rank == 0 else None, root=0)

# Divide rows across processes

rows\_per\_proc = N // size

assert N % size == 0, "Matrix size must be divisible by number of processes"

local\_A = np.zeros((rows\_per\_proc, N), dtype=np.int32)

comm.Scatter(A, local\_A, root=0)

# Local matrix multiplication and timing

comm.Barrier() # sync processes

start = time.time()

local\_C = np.matmul(local\_A, B)

end = time.time()

elapsed\_time = end - start

# Gather final result matrix

C = None

if rank == 0:

C = np.zeros((N, N), dtype=np.int32)

comm.Gather(local\_C, C, root=0)

# Save metrics to CSV (only rank 0)

if rank == 0:

print(f"✅ Done: Matrix {N}x{N}, Processes: {size}, Time: {elapsed\_time:.4f} sec")

headers = ["Matrix Size (N)", "Processes", "Time (s)"]

new\_row = [N, size, round(elapsed\_time, 6)]

# Write or append to CSV

file\_exists = os.path.isfile(CSV\_PATH)

with open(CSV\_PATH, "a", newline="") as f:

writer = csv.writer(f)

if not file\_exists:

writer.writerow(headers)

writer.writerow(new\_row)

step7:

create matrix\_serial.py

import numpy as np

import time

N = 512

A = np.random.randint(0, 10, (N, N), dtype=np.int32)

B = np.random.randint(0, 10, (N, N), dtype=np.int32)

start = time.time()

C = np.matmul(A, B)

end = time.time()

print(f"✅ Serial Execution: Matrix {N}x{N}, Time: {end - start:.4f} sec")

step8:

copy matrix\_mpi\_benchmark.py onto node2

scp -i mymumbaikp.pem matrix\_mpi\_benchmark.py [ubuntu@172.31.6.90:/home/ubuntu/](mailto:ubuntu@172.31.6.90:/home/ubuntu/)

step9:create a hosts file with the following entries

mpi-node1 slots=2

mpi-node2 slots=2

step 10:

mpirun -np 1 python3 matrix\_mpi\_benchmark.py

step11:

python3 matrix\_serial.py

step12: logs after running below commands

mpirun -np 1 python3 matrix\_mpi\_benchmark.py with 512 and 1024 N size

Matrix Size (N),Processes,Time (s)

512,1,0.157555

1024,1,4.040942

Same when in serail way took below time :

Serial Execution: Matrix 512x512, Time: 0.2901 sec

Serial Execution: Matrix 1024x1024, Time: 4.1459 sec





