from mpi4py import MPI

def calculate\_partial\_sum(arr):

partial\_sum = sum(arr)

return partial\_sum

def distribute\_array(arr):

"""

This function distributes an array to multiple processes, calculates the partial sum for each

process, gathers all the partial sums at the root process, and displays the intermediate and final

sums.

:param arr: The input array that needs to be distributed and processed in parallel

"""

comm = MPI.COMM\_WORLD

size = comm.Get\_size()

rank = comm.Get\_rank()

# Distribute the array to all processes

chunk\_size = len(arr) // size

remainder = len(arr) % size

start = rank \* chunk\_size

end = start + chunk\_size

if rank == size - 1:

# Include the remaining elements in the last chunk

end += remainder

chunk = arr[start:end]

# Calculate the partial sum for the chunk

partial\_sum = calculate\_partial\_sum(chunk)

# Gather all the partial sums at the root process (rank 0)

total\_sum = comm.reduce(partial\_sum, op=MPI.SUM, root=0)

if rank == 0:

# Display the intermediate and final sums in order of rank

for i in range(size):

print("Rank", i, "Partial Sum:", calculate\_partial\_sum(arr[i\*chunk\_size:(i+1)\*chunk\_size]))

print("Final Sum:", total\_sum)

if \_\_name\_\_ == '\_\_main\_\_':

arr = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] # Example array

distribute\_array(arr)

python3 mpi.py

pip install mpi4py

sudo apt install python-3-pip

pip install mpi4py

sudo apt update

sudo apt-get install libopenmpi-dev

pip install mpi4py

python3 mpi.py