Lab Course Distributed Data Analytics
Exercise Sheet 1

Yusef Wurf Morales Jaén - 39226

Exercise 0 Explain your system

Laptop	Lenovo Legion 5 15ARH05
CPU	4
Logical Processors	8
Base Speed	1.7 GHz
Cores	4
Sockets	1
RAM	16 GB
Operating Sys	Windows
programming language used	Python 3.10.4
IDE	Spyder 5.2.2

Exercise 1 Basic Parallel Vector Operations with MPI

A) Add two vectors and store results in a third vector

In this exercise, the main process (rank 0) initializes two vectors with a size of 100 with 0, initializes the vectors and divide it into small chunks and send to other workers for parallel execution. Master also add one chunk of vector itself and wait for other processes. When all processes finish, master receives the results of other chunks and add the results to the vector which will contain the summed vector.

```
mpiexec -n 4 -ppn 1 -hosts MPI1,MPI2,MPI3,MPI4 ^
python -u ^
C:^Users^yusef^Desktop^Cloud^Exercise_Sheet_1^Ex_1_A.py
```

```
# Ex_1_A.py
# Author: Yusef Morales
# Date: 2022-04-20
# Version: 1.0
# Description: Add two vectors and store results in a third vector
# Usage: mpiexec -n 4 -ppn 1 -hosts MPI1,MPI2,MPI3,MPI4 ^
# python -u ^
# C:^Users^yusef^Desktop^Cloud^Exercise_Sheet_1^Ex_1_A.py

from mpi4py import MPI
import sys
import numpy as np

comm = MPI.COMM_WORLD
comm_size = comm.Get_size()
rank = comm.Get_rank()
processor_name = MPI.Get_processor_name()

# Initialize vectors
vector_size = 100
if rank == 0:
    print("Vector size: ", vector_size)

# Initialize vectors
vector1 = np.zeros(vector_size, dtype=int)
vector2 = np.zeros(vector_size, dtype=int)
if rank == 0:
    vector1 = np.ones(vector_size, dtype=int)
    vector2 = np.ones(vector_size, dtype=int)

# Calculate chunk size
chunk_size = int(vector_size / comm_size)
if rank == 0:
    print("Chunk size: ", chunk_size)

# Scatter vectors
if rank == 0:
    print("Scattering vectors to other processes")
chunk1 = np.zeros(chunk_size, dtype=int)
chunk2 = np.zeros(chunk_size, dtype=int)
comm.Scatter(vector1, chunk1, root=0)
comm.Scatter(vector2, chunk2, root=0)

# Add chunks
if rank == 0:
    print("Adding chunks")
chunk = chunk1 + chunk2

# Gather results
if rank == 0:
    print("Gathering results")
result = np.zeros(vector_size, dtype=int)
comm.Gather(chunk, result, root=0)

# Print results
if rank == 0:
    print("Result: ", result)

# Print processor name
if rank == 0:
    print("Processor name: ", processor_name)

# Print rank
if rank == 0:
    print("Rank: ", rank)

# Print size
if rank == 0:
    print("Size: ", comm_size)

# Print chunk
if rank == 0:
    print("Chunk: ", chunk)

# Print chunk1
if rank == 0:
    print("Chunk1: ", chunk1)

# Print chunk2
if rank == 0:
    print("Chunk2: ", chunk2)

# Print vector1
if rank == 0:
    print("Vector1: ", vector1)

# Print vector2
if rank == 0:
    print("Vector2: ", vector2)

# Print vector_size
if rank == 0:
    print("Vector_size: ", vector_size)

# Print comm_size
if rank == 0:
    print("Comm_size: ", comm_size)

# Print chunk_size
if rank == 0:
    print("Chunk_size: ", chunk_size)

# Print processor_name
if rank == 0:
    print("Processor_name: ", processor_name)

# Print comm
if rank == 0:
    print("Comm: ", comm)

# Print MPI.COMM_WORLD
if rank == 0:
    print("MPI.COMM_WORLD: ", MPI.COMM_WORLD)

# Print MPI.Get_processor_name()
if rank == 0:
    print("MPI.Get_processor_name(): ", MPI.Get_processor_name())

# Print MPI.Get_rank()
if rank == 0:
    print("MPI.Get_rank(): ", MPI.Get_rank())

# Print MPI.Get_size()
if rank == 0:
    print("MPI.Get_size(): ", MPI.Get_size())

# Print MPI.Get_version()
if rank == 0:
    print("MPI.Get_version(): ", MPI.Get_version())

# Print MPI.Get_library_version()
if rank == 0:
    print("MPI.Get_library_version(): ", MPI.Get_library_version())

# Print MPI.Get_vendor()
if rank == 0:
    print("MPI.Get_vendor(): ", MPI.Get_vendor())

# Print MPI.Wtime()
if rank == 0:
    print("MPI.Wtime(): ", MPI.Wtime())

# Print MPI.Wtick()
if rank == 0:
    print("MPI.Wtick(): ", MPI.Wtick())

# Print MPI.Is_initialized()
if rank == 0:
    print("MPI.Is_initialized(): ", MPI.Is_initialized())

# Print MPI.Is_finalized()
if rank == 0:
    print("MPI.Is_finalized(): ", MPI.Is_finalized())
```

```
C:\Users\Sachin>mpiexec.exe -n 2 -wdir C:\Users\Sachin -hosts 2 DESKTOP-7FIJ0JS 192.168.1.2 -mapall -pwd *****
Enter password for user sachin:
Sorry, 0 + 1 = 1
Sorry, 0 + 2 = 2
Sorry, 0 + 3 = 3
Sorry, 0 + 4 = 4
Sorry, 0 + 5 = 5
Sorry, 0 + 6 = 6
Sorry, 0 +
```

```
Mon Mar 3 12:12:12 2025
NVSMI LOG
    Timestamp                                 : Mon Mar 3 12:12:12 2025
    Driver Version                            : 550.90.07
    CUDA Version                              : 12.4

    Attached GPUs                             : 1
    GPU 00000000:3B:00.0
        Product Name                          : NVIDIA A100-SXM4-40GB
        Product Brand                         : NVIDIA
        Product Architecture                  : Ampere
        Display Mode                          : Enabled
        Display Active                        : Disabled
        Persistence Mode                      : Enabled
        Addressing Mode                       : HMM
        Bus                                   : 0x3B
        Device                                : 0x0
        Domain                                : 0x0000
        Device Id                             : 0x20B010DE
        Bus Id                                : 00000000:3B:00.0
        Compute M.C.P. Id                     : N/A
        Chassis Serial                        : 1320420000164
        Slot Number                           : N/A
        Tray Index                            : N/A
        Node Index                            : N/A
        Peer Type                             : N/A
        Module Id                             : 1
        GPU Module Type                       : Baseboard
        Fabric
            State                             : Completed
            Status                            : Success
        GPU UUID                              : GPU-0a0a0a0a-0a0a-0a0a-0a0a-0a0a0a0a0a0a
        Minor Number                          : 0
        Bridge Chip
            Type                              : N/A
            Firmware                          : N/A
        Multi-Instance Mode                   : Disabled
        MIG Mode
            Current                           : Disabled
            Pending                           : Disabled
        Mig-enabled                           : 0
        Multi-instance GPU                    : Disabled
        Multi-instance GPU Profiles           : N/A
        Multi-instance GPU Partitions         : N/A
        Accounting Mode                       : Enabled
        Accounting Mode Buffer                : 4000
        Accounting Pids                       : No running processes found
        Accounting Processes
            GPU                               : 0
            GPU Instance                      : N/A
            Compute Instance                  : N/A
            Process ID                        : 1
            Command                           : /sbin/init
            Start Time                        : Mon Mar 3 12:0 1:00
            GPU Utilization                   : 0 %
            Memory Utilization                : 0 %
            Max memory                        : 0 MiB
            Time                              : 0 ms
```

```
Linux gpu-a100-7-2 5.15.0-105-generic #115-Ubuntu SMP Mon Apr 15 19:11:56 UTC 2024 x86_64 x86_64 x86_64 GNU/Linux
```