USER MANUAL

Contribution :

- CPU, Disk, GPU, Memory, Network – **Vivek Pabani A20332117**
- Memory – **Sanket Nitin Wagh A20330391**
- Network – **Chaitanya Reddy Chatla A20330053**

Environment Details

- Programming Language : C , Python, CUDA
- Operating System : Ubunu on Docker Container, Jarvis for GPU

Steps for Benchmark :

1) CPU

- **Programs :**

1. cpubenchmark.c

- **Logic :**

1. This program finds the FLOPS and IOPS of CPU.
2. It works on four leves of concurrency : 1 Thread, 2 Threads, 4 Threads and 8 Threads.
3. The main function decides the number of threads with count variable, type of operation with choice variable. The intFun and floatFun functions perform integer and floating point operations for each choice of threads.

- **Instructions :**

1. Go to directory– cpu
2. To compile the program, run command: make
3. To run the program, run command: make run

2) MEMORY

- **Programs** :

1. memorybenchmark.c

- **Logic** :

1. This program finds the throughput and latency for different four operations : Sequential Write and Read, Random Write and Read.
2. It works on three leves of concurrency : 1 Thread, 2 Threads and 4 Threads.
3. The main function decides the number of threads with choice variable, and size of block with typeOfBlocks variable.
4. The dataTransfer function performs all 4 mentioned operations for each combination of threads and blocksize.

- **Instructions** :

1. Go to directory – memory
2. To compile the program, run command: make
3. To run the program, run command: make run

3) GPU

- **Programs** :

1. gpuflopsbench.cu
2. gpumembench.cu

- **Logic** :

1. The memory program finds Read and Write Memory Bandwidth of GPU.
2. The main function creates memory on CPU and copies to GPU, and then read memory from GPU into CPU.
3. This operation is done for three types of bloxk sizes : 1B, 1KB, 1MB.
4. The flops program finds the FLOPS and IOPS of GPU.
5. The gpuFun is the function which performs the operation on GPU environment.
6. The host variables are first declared and initialized. Same number of variables are allocated memory on GPU, and data is copied fro host to device variables.
7. The result variable is the copied back to host.

- **Instructions** :

1. Go to directory – gpu
2. To compile the program, run command: make
3. To run the program, run command: make run

4) DISK

- **Programs** :

1. Diskbenchmark.c

- **Logic** :

1 This program finds the throughput and latency for different four operations : Sequential Write and Read, Random Write and Read.
2 It works on three leves of concurrency : 1 Thread, 2 Threads and 4 Threads.
3 The main function decides the number of threads with choice variable, and size of block with typeOfBlocks variable.
4 The dataTransfer function performs all 4 mentioned operations for each combination of threads and blocksize.

- **Instructions** :

1 Go to directory – cpu
2 To compile the program, run command: make
3 To run the program, run command: make run

5) NETWORK

- **Programs** :

1 tcpserver.py
2 threadtcpclient.py
3 udpserver.py
4 udpclient.py

- **Logic** :

1 The Network benchmark generates the throughput and latency for a TCP and UDP connection.
2. The throughput is calculated by dividing the receive window size with the round trip time taken by the packet.
3. The throughput and latency are calculated for single and two threads and by varying the size of the packets being transmitted over the network.

- **Instructions** :

1 Go to directory – network

2 To run the programs, run command: ./script.sh