

# CS553 Cloud Computing

## Assignment -1

CWID – A20380536

## User Manual

---

### **CPU Benchmark:**

#### **Purpose:**

CPU Benchmark is done for measuring the speed of the processor. These are measured in terms of GFLOPS (Floating point operations) and GLOPS (Integer operations). Multi-threading is done using 1, 2, 4 and 8 threads.

#### **Execution:**

The source code for CPU Benchmark consists of two functions, one is for measuring speed in terms of GFLOPS and other in terms of IOPS using AVX instructions. For the file cpu.c, the command to be used are:

Compile: `$ gcc -mavx2 -pthread cpu.c`

Run: `./a.out 1 8` (first argument takes type of operation (1 or 2) and second argument takes number of threads (1, 2, 4 or 8))

While running the program, we need to pass two arguments i.e. first argument is for type of operation to be perform (1 or 2) and second argument is for number of threads (1, 2, 4 or 8)

For Running all source code benchmarks programs at a once, please use makefile file by using following commands:

`$ make experiment8 experiment9`

### **Memory Benchmark:**

#### **Purpose:**

Memory Benchmark is done for measuring speed of the memory. These are measured in terms of Throughput (MB/s) and Latency (us). Multi-threading is done using 1, 2, 4 and 8 threads. Block sizes are also varied (8B, 8KB, 8MB, 80MB).

#### **Execution:**

The source code for Memory Benchmark consists of one file “memory.c” for measuring the throughput and latency for Sequential and Random memory access operations with varying block sizes (8B, 8KB, 8MB, 80MB).

For memory.c file, the commands to be used are:

Compile: `$ gcc -pthread memory.c`

Run: `./a.out 1 1` (first argument takes type of function (1,2 or 3) and second argument takes number of threads (1, 2, 4 or 8))

While running the program, we need to pass two arguments i.e. first argument is for type of operation to be performed (1, 2 or 3) and second argument is for number of threads (1, 2, 4 and 8)

For Running all source code benchmarks programs at a once, please use makefile file by using following commands:

`$ make experiment5 experiment6 experiment7 experiment8`

### **Disk Benchmark:**

#### **Purpose:**

Disk Benchmark is done for measuring the speed of the disk for read and write operations for both sequential and random access. These are measured in terms of throughput (MB/s) and latency (ms). Multi-threading is done using 1, 2, 4 and 8 threads. Block sizes are also varied (8B, 8KB, 8MB, 80MB).

#### **Execution:**

The source code for Disk Benchmark consists of two files "create\_file.c", "disk.c" for measuring the throughput and latency for read+write operation, sequential read access and random read access operations with varying block sizes (8B, 8KB, 8MB, 80MB).

First run create\_file.c and then disk.c

For create\_file.c file, the commands to be used are:

Compile: `$ gcc create_file.c`

For disk.c file, the commands to be used are:

Compile: `$ gcc -pthread disk.c`

Run: `./a.out 2 8` (first argument takes type of function (1,2 or 3) and second argument takes number of threads (1, 2, 4 or 8))

While running the program, we need to pass two arguments i.e. first argument is for type of operation to be performed (1, 2 or 3) and second argument is for number of threads (1, 2, 4 and 8)

For Running all source code benchmarks programs at a once, please use makefile file by using following commands:

`$ make experiment1 experiment2 experiment3 experiment4`

### **Network:**

#### **Execution:**

The source code for Network consist of four files. Two files for TCP benchmark i.e. "tcp\_s.c and tcp\_c.c" and two for UDP benchmark i.e. "udp\_s.c and udp\_c.c" to calculate the throughput and the latency.

For tcp\_s.c file, the commands to be used are:

Compile: `$ gcc -pthread -o ts tcp_s.c`

For tcp\_c.c file, the commands to be used are:

Compile: `$ gcc -pthread -o tc tcp_c.c`

Open two terminals to run.

`$ ./ts`

`$ ./tc 1` (1, 2, 4 or 8) → number of threads

For udp\_s.c file, the commands to be used are:

Compile: `$ gcc -pthread -o us udp_s.c`

For udp\_c.c file, the commands to be used are:

Compile: `$ gcc -pthread -o uc udp_c.c`

Open two terminals to run.

`$ ./us`

`$ ./uc` (1, 2, 4 or 8) → number of threads

To run all the code at a time, execute the makefile using following command.

`make experiment1 experiment2 experiment3 experiment4 experiment5 experiment6 experiment7  
experiment8 experiment9`