Benchmarking

There are four sections to the assignment:

1)CPU

2)Memory

3)Disk

4)Network

CPU

CPU has two parts

- The GIOPS and FLOPS calculation:
  - This experiment is to compute GFLOPS and GIOPS
  - This is done by two programs. cpu_test_original.c and cpu_avx.c
  - This experiment can be run by running the following command

Optional 1:

```
$ ./cpu_1.sh
```

Optional 2:

```
gcc1 -pthread cpu_test_original.c

./a.out

gcc1 -mavx2 -pthread cpu_avx.c

./a.out
```

- The 600 sample program:
  - This experiment is to do integer and float operations for 10 minutes and produce 600 IOPS samples and 600 FLOPS samples
  - This is done by one program 600_sample_test.c
  - This experiment can be run by running the following command

Optional 1:

```
$ ./cpu_2.sh
```

Optional 2:

```
gcc1 -pthread 600_sample_test.c

./a.out>>sample.file
```

Memory

- This experiment consists of 3 functions that do sequential memory access, sequential write and random write using 8B,8KB, 8MB and 80MB block size
- This experiment is run by the program memory_test.c
- This can be run by the following command

Optional 1:

```
$./memory.sh
```

Optional 2:

```
gcc1 -pthread memory_test.c

./a.out
```

Disk

- This is experiment do sequential memory access, sequential write and random write using 8B,8KB, 8MB and 80MB block size
- This experiment is run by the program disk.c
- This can be run by the following command
- Since it will take very long time to execute diskpart if block size if small (due to 10GB file), especially for read+write. We set initial NUM_LOOPS as 1.0e9, you may change NUM_LOOPS(which is the read data size) and rewrite data size(in call_Read_Write() and Read_Write() functions) according to tables in performance file to get our reasonable run time.

To compile disk:

- find file location for two files: WriteFile.c disk.c
- In linux terminal, type:

gcc writeFile.c -o writeFile

./writeFile

***note: if it shows error message, that means you don’t have write permission yet. Then you should type:

sudo ./writeFile

- Then ype:

gcc disk.c -pthread -o disk

./disk Read_Write/Read_Seq/Read_Ran 8B/8KB/8MB/80MB 1/2/4/8

Network

To compile network:

- find file location for two files: network_server.c network_client.c
- In linux terminal, type:

gcc network_server.c -pthread -o server

gcc network_client.c -pthread -o client

- In one terminal, type:

./server tcp/udp 1/2/4/8

- In another terminal, type:

./client tcp/udp 1/2/4/8

***NOTE: If client side show connection failed, you can try repeat last two steps, and it will work. This might be caused by network traffic