

CLOUD COMPUTING

Assignment 1 Manual

Initial Steps:

- ❖ Create instance on chameleon cloud
(<https://openstack.tacc.chameleoncloud.org/dashboard/project/instances/>)
- ❖ Instance Details : KVM virtual machine m1.medium((2 virtual processors with 4GB RAM and 40GB disk, CentOS 7 Linux for the OS)
- ❖ After creating the instance connect to the instance from any shell using the below command :
`ssh -i cloud.key cc@129.114.111.153`
Before that, you got accosiate a Floating IP to your instance and use them to connect to your machine.
- ❖ Now we can start running the benchmark programs on our instance. We have developed the CPU , Memory and Disk benchmarking on C and Network benchmarking is developed on Python

CPU Benchmark:

- ❖ Compile the CPU_FINAL.c as below
`gcc -pthread -o cpu CPU_FINAL.c`
- ❖ Next you can run the ouput file cpu using the below command
`./CPU`
- ❖ The program asks for the user to Enter their thread choice(1,2,4,8) and Operations to perform(Integer or Floating point operations)
- ❖ Once the choices are given by the user , the program generates the output showing the throughput and latency for the CPU operations.

Memory Benchmark:

- ❖ Compile the Memory.c as below
`gcc -pthread -o mem Memory.c`
- ❖ Next you can run the ouput file mem using the below command
`./mem`
- ❖ The program asks for the user to Enter their thread choice(1,2,4,8) and Block sizes(8B,8KB,8MB,80MB) to perform read+write operations on Memory.
- ❖ Once the choices are given by the user , the program generates the output showing the throughput and latency for the sequential read/write and random read/write performed on Memory

Disk Benchmark:

- ❖ Compile the DISK_FINAL.c as below
`gcc -pthread -o disk DISK_FINAL.c`
- ❖ Next you can run the disk file mem using the below command
`./mem`
- ❖ Place a 1GB test file on your machine to perform file operations on disk
- ❖ The program asks for the user to Enter their thread choice(1,2,4,8) and Block sizes(8B,8KB,8MB,80MB) to perform read+write operations on Disk.

- ❖ Once the choices are given by the user , the program generates the output showing the throughput and latency for the sequential read/write, sequential read and random read performed on Memory

Network Benchmark

- ❖ Goto the folder named "Network".
- ❖ Here we will require 2 instances to be opened.
- ❖ one of the instance will be acting as SERVER and another will be acting as CLIENT.
- ❖ Before execution, the files for Client and Server needs to be modified with the IPAddress.
- ❖ Change the serverIPAddress for server and client program for TCP.
- ❖ Change the clientIPAddress for server and serverIPAddress for client in UDP.
- ❖ Copy the entire "Network" folder on both the terminals.
- ❖ To execute command type `python *.py` on both the terminals.
- ❖ On Server side execute the code as:
`python networkServerTCP.py/ python networkServerUDP.py`
- ❖ On Client side execute the code as:
`python networkClientTCP.py/ python networkClientUDP.py`
- ❖ The desired output for TCP & UDP will be displayed on the Client Side terminal.