

## Design Document of Benchmarking

Benchmarking is the process of measuring some units against its standard. To testify the results of the unit in terms of quality, time, efficiency, cost and etc. Here the benchmarking of following has been done.

Units under benchmarking:

- CPU
- Disk
- Network

Programming language used for benchmarking:

CPU – C Programming

Disk and Network – Java

Test Bed used:-

Amazon Free Tier EC2.T2 Micro

Processor: Intel Xeon Processor speed up to 3.3GHz

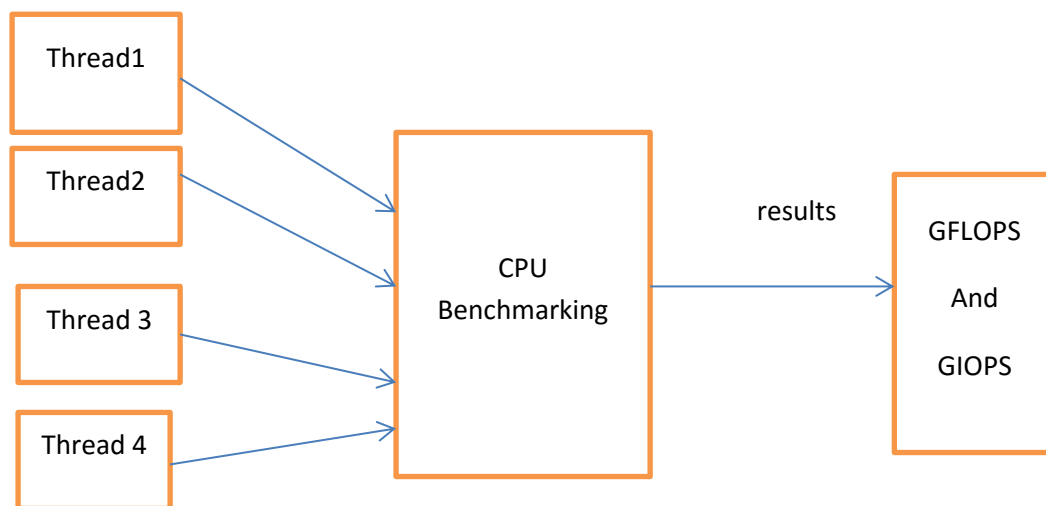
## CPU Benchmarking

CPU benchmarking is done using C Programming language

The idea of calculating the FLOPS and IOPS, of a CPU is achieved by executing the number of instructions inside the loop, with certain amount of time. Now to achieve the parallelism Pthreads are used with number of threads varying from 1, 2 and 4.

Calculation of FLOPS and IOPS are done on 1000000000 iteration of Floating point operations and Integer operations, GFLOPS are calculated from this results of FLOPS and IOPS.

The Execution of 10min of CPU experiment is implemented using Timer and clock which executes for 10min and takes the reading of number of instructions per second. To achieve the parallelism the synchronization has been maintained in order to have a better CPU benchmarking results.



To test the Results against the benchmark, Linpack is used.

To download and execute the Linkpack on machine download the Linpack from following site for Linux or windows:

<https://software.intel.com/en-us/articles/intel-mkl-benchmarks-suite>

## Disk Evaluation

Evaluation of Disk is implemented in Java.

Operations to be performed are:

- Random Read and Write
- Sequential Read and Write

Size under consideration:

1. 1 Byte
2. 1 KB
3. 1 MB

To execute the Disk benchmark, configure the evaluation=2 in ServerConfig.properties file. Then run the ServerController.java where main class resides. Now from there the flow of control will go to DiskEvaluation.java, where you the Number of threads is checked and based on that it will call method in ComputeDiskEvaluation.java where its determined what operation to be done, based on the configuration done in ServerConfig.properties file.

Firstly the Code executed for 1 Thread, on sequential. Then based on the parameter set into ServerConfig.properties file the program execution follows.

To ensure multithreading environment, to achieve the parallelism, CyclicBarrier is used, which make sure that threads starts at the same time to achieve the synchronization among threads.

For each respective read or write operation and based on sequential and random access, the log file will get generated per thread in backend of system in order to compute the benchmarking.

Now to testify the computed result with the standard results, **iozone3** is used which is a Disk Benchmarking Tool. By running this tool on each machine we will achieve the standard desired results of read/write of Random and sequential access of Disk.

## Network Evaluation

Benchmarking evaluation of Network is implemented in Java.

The Idea behind the network benchmarking was to test the network bandwidth and latency of the roundtrip time of a pack to send.

Protocols used:

- TCP
- UDP

Size of Packet under consideration:

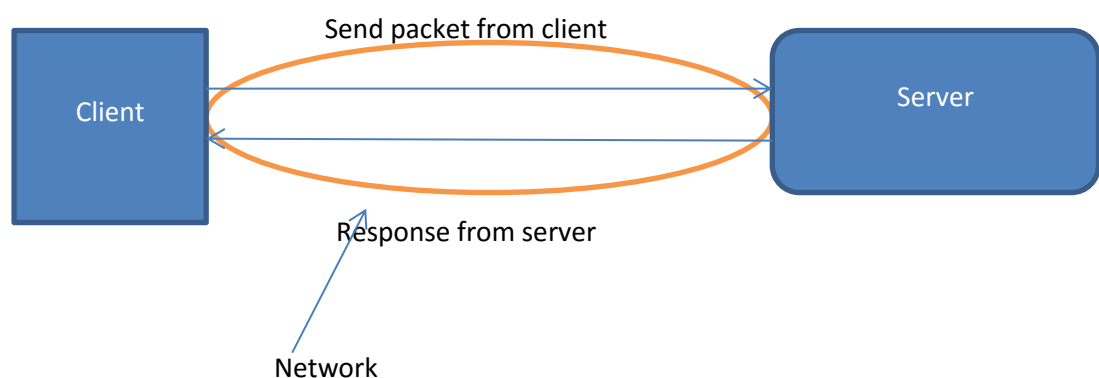
- 1 Byte
- 1 KB
- 64 KB

Here, 2 instances are required; one for client and other is for the server.

Configure the Client with the IP address and Port of server and Packet size (for UDP) in ClientConfig.properties and Configure the Server with Packet size (for UDP) in ServerConfig.properties and also so evaluation=3, and other required configuration.

Now the Java Sockets have been used for the implementation of TCP/UDP.

Multithreaded server is implemented which handles and responds to multiple client requests at the same time. Now here the idea is to measure the bandwidth and latency of the network, so Roundtrip time is calculated by measuring the time to send a packet from client to server, and sending of same packet from server to client.



## **Future Scope**

Better Multithreading can be achieved with the usage of tight synchronization protocols.

Throughput can be of Network system can be tested in two environments.

- 1) Ideal Environment where there is no traffic.
- 2) Environment with heavy traffic.

So it achieves better results and performance of the network under evaluation.

System can be automated with practical data as an input.

## Conclusion

Through the design and implementation of Benchmarking, we understood that how the performance of each system like CPU, Network and Disk have been analysed. Through the benchmarking we can analyse that, the CPU will take almost same time for all the floating point and integer instructions regardless of the number of threads executing in parallel. For network, TCP and UDP we can deduce that the UDP is slower than a TCP and network throughput increases as the size of data to be transferred increases. Disk benchmarking analysis conveys that the sequential access of disk will always be faster than the random access, and throughput will also increase by the same amount.