

Design Document

Programming assignment 1 deals with three individual modules of benchmarks.

They are:

1. CPU Benchmark
2. Disk Benchmark
3. Network Benchmark

The abstract of design and implementation of each benchmark as follows.

- Creation and initialization of threads is done in the main block along with the start time end time calculations and other necessary attributes.
- The main block contains code to create and initialize threads along with all the necessary attributes.
- Multiple threads can be implemented concurrently, not just one. Here along with the single thread we can implement 2 or 4 threads for the CPU benchmark which calls the function 2 and 4 times respectively.(CPU has 1,2,4 threads whereas Network and Disk and 1,2 Threads)
- The time taken to execute each thread in CPU is obtained by using `System.currentTimeMillis()` function which is later to be converted to seconds in order to compute FLOPS, IOPS of the CPU.
- The time taken to execute each thread in Disk and Network is obtained by using `System.nanoTime()` function which is later to be converted to seconds in order to compute latency and throughput of the disk and network.
- The random and sequential accesses are implemented in same function for each of the benchmarks.

The detailed design description of all benchmarks:

- **CPU Benchmark:** Three operations are to be performed. The first is to create a function that adds floating type values for 1 or 2 or 4 threads. Second is to create a function that adds integer type values for 1 or 2 or 4 threads. The third is to run the functions for adding floating type variables, integer type variables for 4 threads for 10 minutes respectively and take samples for every second on how many instructions per second were achieved during the experiment. Each of the above mentioned functions execute threads and outputs the benchmark.
- **Network Benchmark:** This benchmark consists of two programs for TCP and UDP i.e client and server. Blocks of packets of size 1 BYTE, 1KILOBYTE and 64 KILOBYTE are transferred from client to the server, a 1:1 communication is to be established which means 1 thread from Client can transfer data to 1 thread of the Server and reply is vice versa. The Server checks the data packets received and sends an acknowledgement to the Client. Client in turn calculates the round trip time which in turn can be used to determine the latency and Throughput.
- **Disk Benchmark:** Here three java files are created one each for 1 byte, 1KB, 1MB. Each of these files can run 1 to 2 threads in turn calling the Sequential read, write and Random read, write. The end result would be a file creation of size 1 byte, 1KB, 1MB based on the java run that is run. The time taken for the entire operation is calculated

which can be used to determine the latency read, write and Throughput read, write for Sequential and Random access for each java file.

Improvements and Extensions to program:

1. CPU benchmark:

- More than 4 Threads can be used to calculate the GFLOPS
- Better results of GFLOPS, GIOPS can be achieved by performing number of operations (such as additions, multiplication and so on).

2 . Disk Benchmark:

- More than 2 Threads can be used to calculate the latency and throughput.
- Having multicore architecture can be advantageous as it can be used to improve concurrency and a better benchmark.

3 . Network Benchmark:

- By properly tuning the code the program can be used for wireless networks.
- By using multiple Servers and Clients performance and efficiency can be improved.