CS 553 Cloud Computing

Programming Assignment 1

Sujay Gunjal (CWID: A20351746)

Design Document

I have worked on the following three types of Benchmarks

- Disk benchmarking
- Network benchmarking
- CPU benchmarking

Disk Benchmarking:

- I have done disk benchmarking in java.
- The design of program includes declaration of four different functions for the four different operations they are sequential read, sequential write, random read and random write.
- Every time a new thread is created and makes call to run() method ,which will make call to other four methods by changing block sizes from (1B, 1KB, 1MB).
- HashMap data structure is used to store block of different length which will save the time while writing to the file. Using the clock start and end time is calculated.
- The difference of both the time further computes the latency and read or writes speed for both types of accesses i.e. random and sequential.

Network Benchmarking:

- I have done network benchmarking in java.
- The benchmark is divided into two different programs i.e. one for client and other for server.
- The basic flow of program is that, varying blocks of packets/ buffer size is transferred from client to server. While at server's end, numbers of bytes received are transferred back to the client.
- Further, at the client's end total time taken in this to and fro packet transfer is calculated and on this basis of this value, latency and throughput are calculated.

CPU Benchmarking:

- We have written code in C to measure the benchmarks more accurately since Java and other high level programming languages would skew the results due to its running environment and GC() processes. At first, I would calculate the time taken for running an empty loop. This would be subtracted from the total time taken to obtain the exact time for running the operations.
- For multi-threading, we have used the pthread library. Each of the threads would calculate the FLOPS and IOPS one after the other.
- There two files **'cpu600Performance.c'** and **'cpuPerformance.c'**.**'cpuPerformance.c'** is for normal execution of program.
- File named **'cpu600Performance.c'** is created for printing 600 samples for FLOPS and 600 samples of IOPS. Each sample is printing after 1-second. Output of this evaluation is store in 'flops.txt' and 'iops.txt'. It is not readable at console all 600 records, for that purpose all output data is saved in these files.

Improvements and Extensions to program:

The possible improvements and extensions to our program are as follows:

CPU benchmark:

- We would get more accurate benchmark results if we could isolate the effects of cache. The operations needs to be random enough so that the processor does not optimize it by serving the results from cache.
- We could differentiate the effect of FPU and CPU on the benchmarks.

Disk Benchmark:

- The level of concurrency may be increased.
- We can extend our project to find the rate of data transfer in virtual disk through a virtualization.
- As there are possibilities of bad sectors in any disk, due to which latency increases. We need to keep a check on these bad sectors

Network Benchmark

- We can extend the packet/ buffer size to 1GB. As we do not have 1GB Ethernet card, we could not test it for 1GB block of data.
- We can extend this program to wireless networks.
- We can test this benchmark from one node to another also.
- There is a possibility of data loss while transferring it from client to server. This may be avoided by using more rigid architecture.
- There can be delay in transfer due to slower connection, which might cause in increase in latency and decrease in throughput.