CS553 Cloud Computing

Assignment -1

CWID – A20380536

Design Document

The programming assignment# 1 consists of four benchmarks which are mentioned below. I have used C language for the benchmarking code.

- CPU Benchmarking
- Memory Benchmarking
- Disk Benchmarking
- Network Benchmarking

The program design, and design tradeoffs considered for benchmarking are as follows:

- Clock () function is used for measuring the processor, memory and disk speed. More specifically, clock () function is used to determine the execution time of the threads used in the programs.
- For CPU benchmark, multiple addition operations are performed multiple times over a loop for calculating FLOPS and IOPS.
- For memory benchmarking, read+write operations, sequential write access and random write access methods are implemented in the same program.
- For disk benchmarking, read+write operations, sequential read access and random read access methods are implemented in the same program.
- The main () method of the programs are used for creating and joining the threads with all the important attributes.
- CPU Benchmarking, memory benchmarking, disk benchmarking and network benchmarking is implemented for 1, 2, 4 and 8 threads.
- As the name suggests, a single thread is calling a function for once. However, multiple threads like 2,4 or threads are calling functions more than once, here 2, 4 or 8 times respectively.

The detailed benchmarking description are as follows:

CPU Benchmark: Two functions are written for CPU benchmarking with AVX instructions, one for IOPS and other for FLOPS. Both functions have addition operations functionality implemented, one of which adds integer values and other adds floating values. The processor speed is measured at varying levels of concurrency (1, 2, 4 and 8 threads).

Memory Benchmark: In memory benchmark three operations are implemented i.e. read+write operations, sequential write access and random write access. The program is implemented for varying block sizes (8B, 8KB, 8MB, 80MB). Function memcpy() is used to implement read and write operations. This function copies the data from source location to the memory block of destination location. The

memory speed is measured at varying levels of concurrency (1, 2, 4 and 8 threads) for calculating throughput (in MB/s) and latency (in microseconds, us) on 1.28GB data.

Disk Benchmark: In memory benchmark three operations are implemented i.e. read+write operations, sequential read access and random read access. The program is implemented for varying block sizes (8B, 8KB, 8MB, 80MB). Function fread() and fwrite() is used to implement read and write data into the file respectively. The start and end times of threads are measured using clock () function. The disk speed is measured at varying levels of concurrency (1, 2, 4 and 8 threads) for calculating throughput (in MB/s) and latency (in milliseconds, ms) on 5GB file.

Network Benchmark:

Improvements and Extensions to the programs can be discussed as follows:

CPU Benchmark:

- One of the possible extension to CPU benchmarking is to increase the level of concurrency i.e., instead of restricting to 8 threads, we can use even more than 8 threads.
- In the program, have implemented addition operations using AVX instructions. However, for better and efficient results, we can perform other implementations also like sorting, searching etc. It will increase the complexity of the program and helps us to get more refined results.
- We can also use modern CPU architecture like multi-core architecture.

Memory Benchmark:

- Frequently got segmentation fault error during memory access while performing memory benchmarking operations. This can be avoided by allocating a block of MB or KB memory which will return a pointer to the starting point of the memory block.
- Same as for CPU, the level of concurrency can be increased for memory speed measurement.

Disk Benchmark:

- The disk benchmark which has been performed can also be performed to measure the performance of both SSD and HDD. This can be implemented by setting disks as RAID arrays.
- Multi-core architecture can be used to obtain better and more reliable results.
- Same as for CPU, the level of concurrency can be increased for disk speed measurements.

Network Benchmark:

For Network benchmark, client-server application has been built to send and receive data using TCP and UDP connections.

TCP: In TCP the client establishes the connection with the server using sockets. The data is transferred using the connection. TCP benchmark has been implemented using the loopback interface. Latency (Round trip time for the data to send and receive) and the throughput (time when the entire data is transferred) for the connection has been calculated to get the network speed.

UDP: In UDP there is no connection established between client and server. The client creates sockets and sends the packets using the send to and receive to function. The packet includes the data and the address to the server.

Latency: (execution time*1000)/iteration_count;

Throughput: (((512*iteration)/execution time) *thread count)/1024