CS553 Programming Assignment #1

Benchmarking

Design Doc

Harsh C. Parikh
A20338453

Introduction:

This assignment is about benchmarking of different parts of computer system such as CPU, memory, disk and network benchmarking and these are performed on t2.micro instances of Amazon AWS EC2. I am going to discuss the program design, design trade offs made and possible improvements and extensions in this documents.

Program Design:

The benchmarking system is developed in Python and bash scripting is also used. For concurrency, multi-processing is used. And for other operations Python inbuilt library is used for different purposes.

CPU Benchmarking:

The CPU Benchmarking aims to measure how fast the CPU can perform mathematical operations. We can get the idea of CPU performance from GIOPS and GFLOPS terms. The required number of threads are created to perform concurrent operations. The total time elapsed and the number of operations & iterations are used to calculate the GIOPS and GFLOPS. Moreover, number of iteration is important because it affects the result as well. At last, Lin-pack benchmarking is performed and the results are compared.

Disk Benchmarking:

Disk Benchmarking is used to check the disk speed while reading and writing. Disk benchmarking is performed by changing the block sizes from 1B/1KB/1MB. Sequential and random access is also performed here. The required number of threads are created to parallelize the operations. For every operation the file size is fixed to 30 MB. File Write operation is performed and is followed by Read operation otherwise there might come an exception. IOzone benchmarking is performed and the results are compared.

Network Benchmarking:

Network benchmarking is used to check the network performance on different protocols. In this, we are changing the block sizes to 1B/1KB/64KB and also varying the number of threads. Sometimes few problems are arose if two threads try to access file at a same time then it's not allowed. Only one can access it and other will be in sleep or doing other activity. Iperf benchmarking is performed and results are compared in document.

Design Trade offs:

- Compiler optimizations are turned off. I haven't used any compiler to compile the program.
- Garbage collector is turned off.
- No other processes are performed while calculating timing function.

Possible Improvements and Extensions:

• Nowadays Modern processors have multiple cores with different clock frequencies. So the clock frequencies is considered when performing benchmarking. It is also an important factor.

• Memory and disk cache can be disabled somehow to get more accuracy in the results.

• Hardware should be considered when performing memory benchmark as RAM with different clock frequencies might affect the performance benchmarking.

• The file contents should be verified somehow before reading and writing to the file. Because while doing sequential or random access operations, it may affect efficiency.