

# **AWS-EC2 Clouds Benchmarking**

**Programming assignment – 1**

**Design Document**

**SPRING -2015**

**by**

**Patel Darpan  
A20345898**

## **INTRODUCTION**

There are so many so many cloud available now a like a AWS – EC2 Clouds, Google-App Engine, Microsoft AZURE .... etc., They providing benchmark of CPU, DISK, GPU, NETWORK, KEYBOARD, MOUSE and other computer devices. Here, I do benchmark on AWS-EC2 clouds for CPU, DISK and MEMORY. It gave me a benchmarking result for multiple experiment. This programming assignment is purely based on AWS-EC2 cloud services. I made all the program for CPU, DISK, MEMORY

on my machine and then check it all on AWS-EC2 terminal. Then I made a performance evaluation of all the experiment in terms of table and graph form. It includes **Throughput** and **Latency** as well as Efficiency.

I also compare my result on various Benchmarking Tool like “linpack” for CPU, “iozone” for DISK and “stream” for MEMORY. It give me nearest and accurate result compare to my practical results. I faced so many issues during this programming assignment in different different categories. I include trade-off and future extensions of this programming assign ments.

## CPU BENCHMARK

File : Gflops.java Plateform : Eclipse, JAVA

- CPU Benchmarking parameter is FLOPS, IOPS, MIPS, etc.. In this assignment I used FLOPS and IOPS as my Benchmarking parameter. I used GFLOPS and GIOPS for more better and accurate results.
- CPU benchmarking program gives me a GFLOPS and GIOPS. I do millions of iterations in my program for better result so that it give me accurate result.
- Initially, for small no. of iterations it is increase gradually and then it is being stable after some points of iterations and that was 100000000 in my case.
- I used “linpack” benchmark for checking practical results.

ISSUES :

- Difficult to find theoretical peak performance of GFLOPS
- Troubleshooting is not that much easier.

## DISK BENCHMARK

File : Cloud\_disk\_thread.c Plateform : C (gedit/terminal)

- DISK performance parameter is measured in throughput(MB/s) and

latency (ms) for Sequential and Random Read/Write Operations on File.

- I used FILE SIZE is fixed 100 MB for each block size 1B, 1KB and 1MB because it gives me better result for small amount of block size like 1B.
- If I used small FILE SIZE then it gives me fluctuating results for all kind of Block size and all other experiments.
- I used “iozone” for checking practical performance.

ISSUES:

- a.) Implement programming for multi-threading in C and maintain concurrency for joining thread.
- b.) for 1B blocksize random read and write functions take so much of time and system is being hanged.

## MEMORY BENCHMARK

file : Memory\_benchmark.c Platform : C

- I used data structure for reading and writing from memory and to memory. I calculate throughput and latency for Sequential and Random Access.
- I used “stream” for checking my practical results.

ISSUES:

- how to find memory theoretical performance.
- Sequential Access is used cache reads/writes.

