

## DESIGN DOCUMENT

### CPU BENCHMARKING

- C programming is used for benchmarking of CPU.
- For testing the performance of the CPU, GIOPS (Integer Operation) and GFLOPS (Floating Operation) are utilized.
- Arithmetic Operation are used to analyze the performance of the CPU
- Optimal Performance for CPU is measured using threads for both Integer and Floating Operations.
- We have used 1 thread, 2 threads, 4 threads and 8 threads for the calculating GIOPS and GFLOPS.
- INTOperation() functions is used for performing Integer operations and FLOATOperation() functions is used for performing Float operations. 22 arithmetic operations are looped with the result of the threads to calculate the GIOPS and GFLOPS for  $10^9$  times.
- Also for AVX instructions INTOperation() function is used for performing Integer operations and FloatOperation() function is used for performing Float operations. 6 arithmetic operations and 10 arithmetic operations are looped with result of the thread to calculate the GIOPS and GFLOPS for  $10^9$  times respectively.

### DISK BENCHMARKING

- C programming is used for benchmarking of Disk.
- Performance of disk is analysed on 4 different block sizes i.e. 8 bytes, 8KB, S, 80MB.
- Sequential read+write operations and Random read+write operations are performed for calculating the execution time along with its throughput and latency.
- We have used 1 thread, 2 threads, 4 threads and 8 threads for the calculating throughput and latency for Sequential and random read+write concurrency.
- Throughput is calculated in MB/sec and latency is calculated in ms.
- Sequential and Random read+write operations are done on 10GB file.

## MEMORY BENCHMARKING

- C programming is used for benchmarking of Memory.
- Performance of memory is analysed on 4 different block sizes i.e. 8 bytes, 8KB, 8MB, 80MB.
- Sequential read+write operations and Random read+write operations are performed for calculating the execution time along with its throughput and latency.
- Read+write operation uses memcpy() function and Sequential , Random write operation uses memset() function.
- We have used 1 thread, 2 threads, 4 threads and 8 threads for calculating throughput and latency of Sequential and random read+write concurrency.
- Throughput is calculated in MB/sec and latency is calculated in ms.
- Sequential and Random read+write operations are done on 1GB file.

## NETWORK BENCHMARKING

- JAVA programming is used for benchmarking of Network.
- TCP and UDP both protocol is used in this phase.
- Loop back interface consist of measuring the network speed for 1 node with 2 processes
- Packets are transferred from client to server and back to client.
- Performance of network is analyzed on 4 different block sizes i.e. 8 bytes, 8KB, 8MB, 80MB.
- Throughput is calculated in MB/sec and latency is calculated in ms.
- TCP supports 3-way handshake protocol as it is connection oriented.
- UDP is connection less hence there is no guarantee that the packet is received to receiver.

## Design Trade offs and improvements

### **CPU**

We can improve CPU benchmark by implementing complex operations like vectors and linear equations. Compute the benchmark for more cycles.

### **Memory**

We have run memory benchmark for 1GB memory. May be we can increase the memory size.

### **Disk**

We can use large machine instead of medium size machine where in we can read write to bigger files and test the results.

### **Network.**

For TCP we are using 100 MB file for transfer. Improving the code and making it more efficient to handle bigger file .