**PERFORMANCE EVALUATION**

**Team**

Chaithanya Kumar Anichetty Murali(A20327683)

Shivakumar Vinayagam(A20341139)

Vinodh Kandaswami(A20334757)

**CPU BENCHMARK**

1. **A) B)**To measure the processor speed, in terms of floating point operations per second (Giga FLOPS) and integer operations per second(Giga IOPS)

|  |  |  |
| --- | --- | --- |
| **THREADS** | **GFLOPS** | **GIOPS** |
| 1 | 3.477 | 3.5 |
| 2 | 5.176 | 4.47 |
| 4 | 7.432877 | 7.37 |
| 8 | 7.39 | 7.41 |

**Graph:** GFLOPS and IOPS for varying Threads

**X-axis:** No of Threads

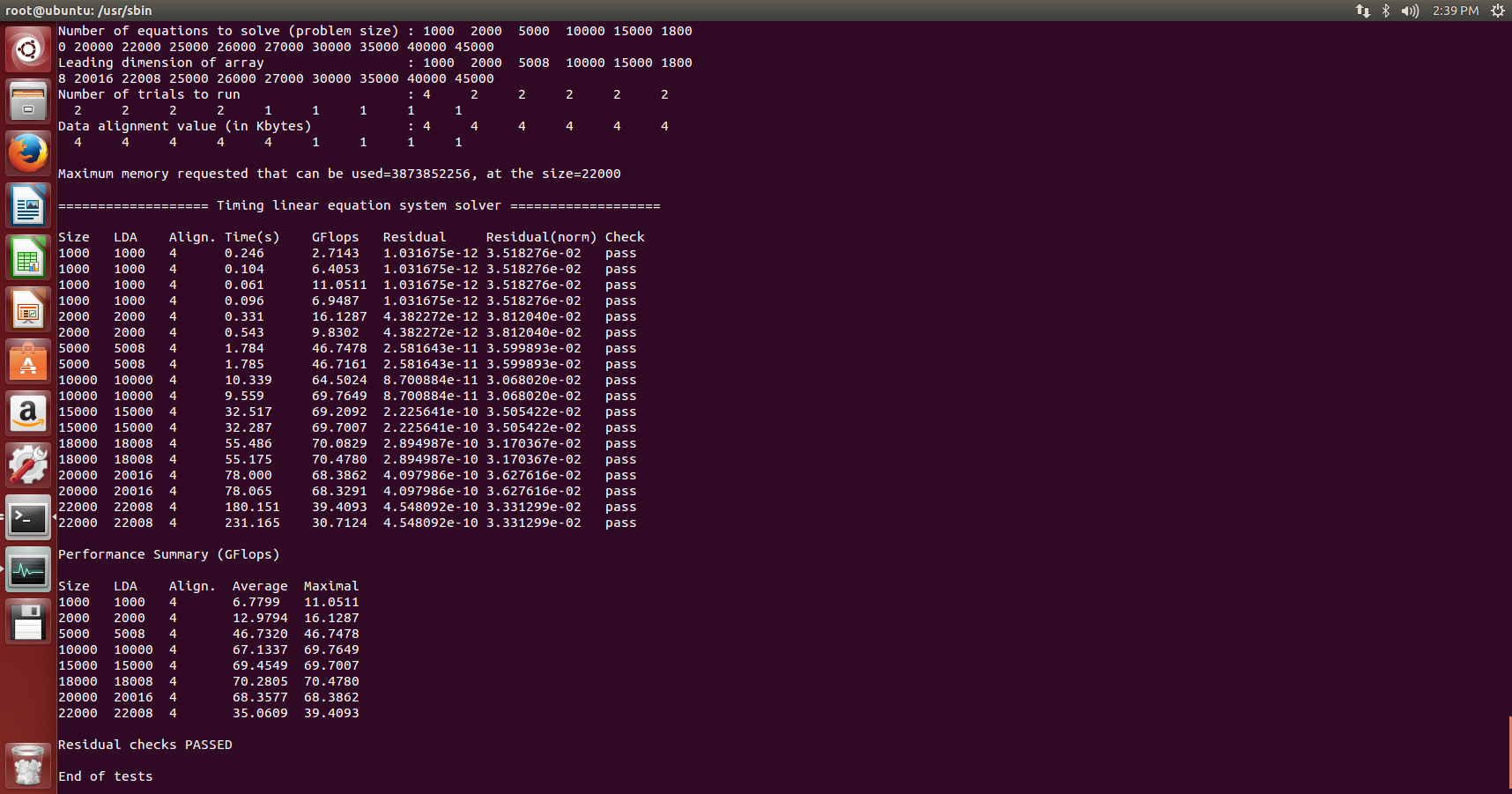
**Y-axis:** Giga operations per second

**1.C)** The **optimal number of threads to get the best performance is 4** as the number of cores in the tested computer is 4 .

1.D) The theoretical peak performance is 9.2 GFLOPS for the benchmarked computer

1.E) The efficiency achieved is about 80% of the peak theoretical value.

BENCHMARKING TOOL RESULTS LINPACK:



**GPU BENCHMARK**

**2.D)** Analyzing the GPU performance by calculating Throughput and Latency

**Throughput**

|  |  |
| --- | --- |
| **BLOCK SIZE** | **THROUGHPUT**  **(GB/S)** |
| 1B | 0.00017 |
| 1KB | 0.0181 |
| 1MB | 3.36 |

**GRAPH:** Shows the throughput

**X-axis:** Block size in Bytes

**Y-axis:** Throughput in GB/S

**Latency**

|  |  |
| --- | --- |
| **BLOCK SIZE** | **LATENCY**  **(ms/B)** |
| 1B | 58.191 |
| 1KB | 0.055 |
| 1MB | 0.0097 |

**GRAPH:** Shows the latency

**X-axis:** Block size in Bytes

**Y-axis:** Latency in Seconds

**2.A) B) C)Maximum GFLOPS and GIOPS of GPU with full concurrency**

**Practical values Theoretical Values**

GFLOPS 470.588 543.87

GIOPS 397.65 NIL

**2.C)The theoretical compute speed is calculated and the efficiency is about 85%. The value is low cause we are only executing about 5 operations for floating or integer per cycle or per thread.**

**DISK BENCHMARK**

1.A) B)Analyzing the Disk performance by calculating throughput and latency.

**THROUGHPUT**

**SEQUENTIAL READ**

|  |  |  |  |
| --- | --- | --- | --- |
| **BLOCK SIZE** | **1 Thread(Mb/s)** | **2 Threads(Mb/s)** | **4 Threads(Mb/s)** |
| 1B | 1.8 | 1.2 | 0.9 |
| 1KB | 697.3 | 340.68 | 294.74 |
| 1MB | 3874.2 | 2895.076 | 2158.092 |

**Graph:** Depicting for 1 and 2 threads

**X-axis:** Block size in Bytes

**Y-axis:** Throughput in Megabytes/second

**SEQUENTIAL WRITE**

|  |  |  |  |
| --- | --- | --- | --- |
| **BLOCK SIZE** | **1 Thread(Mb/s)** | **2 Threads(Mb/s)** | **4 Threads(Mb/s)** |
| 1B | 0.8 | 0.5 | 0.1 |
| 1KB | 225.8 | 157.39 | 114.59 |
| 1MB | 311.63 | 274.85 | 223.76 |

**Graph:** Plotting for 1,2 and 4 threads

**X-axis:** Block size in Bytes

**Y-axis:** Throughput in Megabytes/second

**RANDOM READ**

|  |  |  |  |
| --- | --- | --- | --- |
| **BLOCK SIZE** | **1 Thread(Mb/s)** | **2 Threads(Mb/s)** | **4 Threads(Mb/s)** |
| 1B | 1.2 | 1.09 | 0.951 |
| 1KB | 198.42 | 156.214 | 137.83 |
| 1MB | 2138.36 | 1792.053 | 1196.171 |

**Graph:** Plotting for 1,2 and 4 threads

**X-axis:** Block size in Bytes

**Y-axis:** Throughput in Megabytes/second

**RANDOM WRITE**

|  |  |  |  |
| --- | --- | --- | --- |
| **BLOCK SIZE** | **1 Thread(Mb/s)** | **2 Threads(Mb/s)** | **4 Threads(Mb/s)** |
| 1B | 0.35 | 0.27 | 0.245 |
| 1KB | 140.46 | 117.199 | 103.817 |
| 1MB | 303.65 | 167.791 | 127.815 |

**Graph:** Plotting for 1,2 and 4 threads

**X-axis:** Block size in Bytes

**Y-axis:** Throughput in Megabytes/second

**LATENCY**

**SEQUENTIAL READ:** 0**.**00190734863281 milliseconds

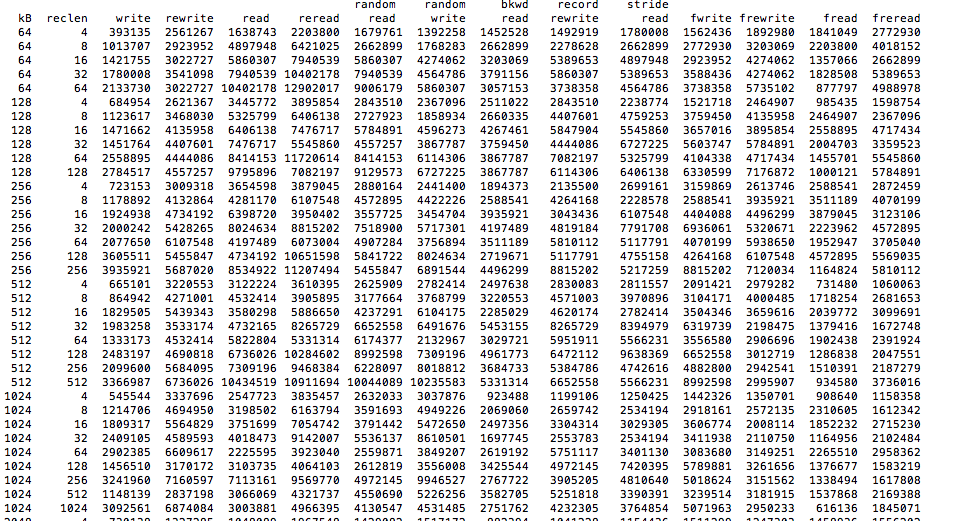
**SEQUENTIAL WRITE:** 0.0024576721191 milliseconds

**RANDOM READ:** 0.00405311584473 milliseconds

**RANDOM WRITE:** 0.0050067901611 milliseconds

**4 .D) The optimal number of concurrency to get the best performance is 1.**

BENCHMARKING TOOL RESULTS IOZONE



**MEMORY BENCHMARK**

**3.A) B) C)** Analyzing the memory performance by calculating throughput and latency.

**SEQUENTIAL THROUGHPUT(MB/s)**

|  |  |  |
| --- | --- | --- |
| **BLOCK SIZE** | **1 Thread** | **2 Threads** |
| 1B | 76 | 1.404 |
| 1KB | 412 | 343 |
| 1MB | 5619 | 1643 |

**Graph:** Depicting sequential throughput

**X-axis:** Block size in Bytes

**Y-axis:** Throughput in Megabytes per second

**SEQUENTIAL LATENCY(msec/B)**

|  |  |  |
| --- | --- | --- |
| **BLOCK SIZE** | **1 Thread** | **2 Threads** |
| 1B | 71 | 0.701 |
| 1KB | 398.4 | 163.29 |
| 1MB | 5371 | 930 |

**Graph:** Depicting sequential latency

**X-axis:** Block size in Bytes

**Y-axis:** latency in micro second/byte

**RANDOM THROUGHPUT(MB/s)**

|  |  |  |
| --- | --- | --- |
| **BLOCK SIZE** | **1 Thread** | **2 Threads** |
| 1B | 13 | 356 |
| 1KB | 2424 | 1489 |
| 1MB | 193712 | 319103 |

**Graph:** Depicting random throughput

**X-axis:** Block size in Bytes

**Y-axis:** Throughput in Megabytes per second

**RANDOM LATENCY(msec/B)**

|  |  |  |
| --- | --- | --- |
| **BLOCK SIZE** | **1 Thread** | **2 Threads** |
| 1B | 14 | 713 |
| 1KB | 2510 | 3062 |
| 1MB | 198723 | 56317 |

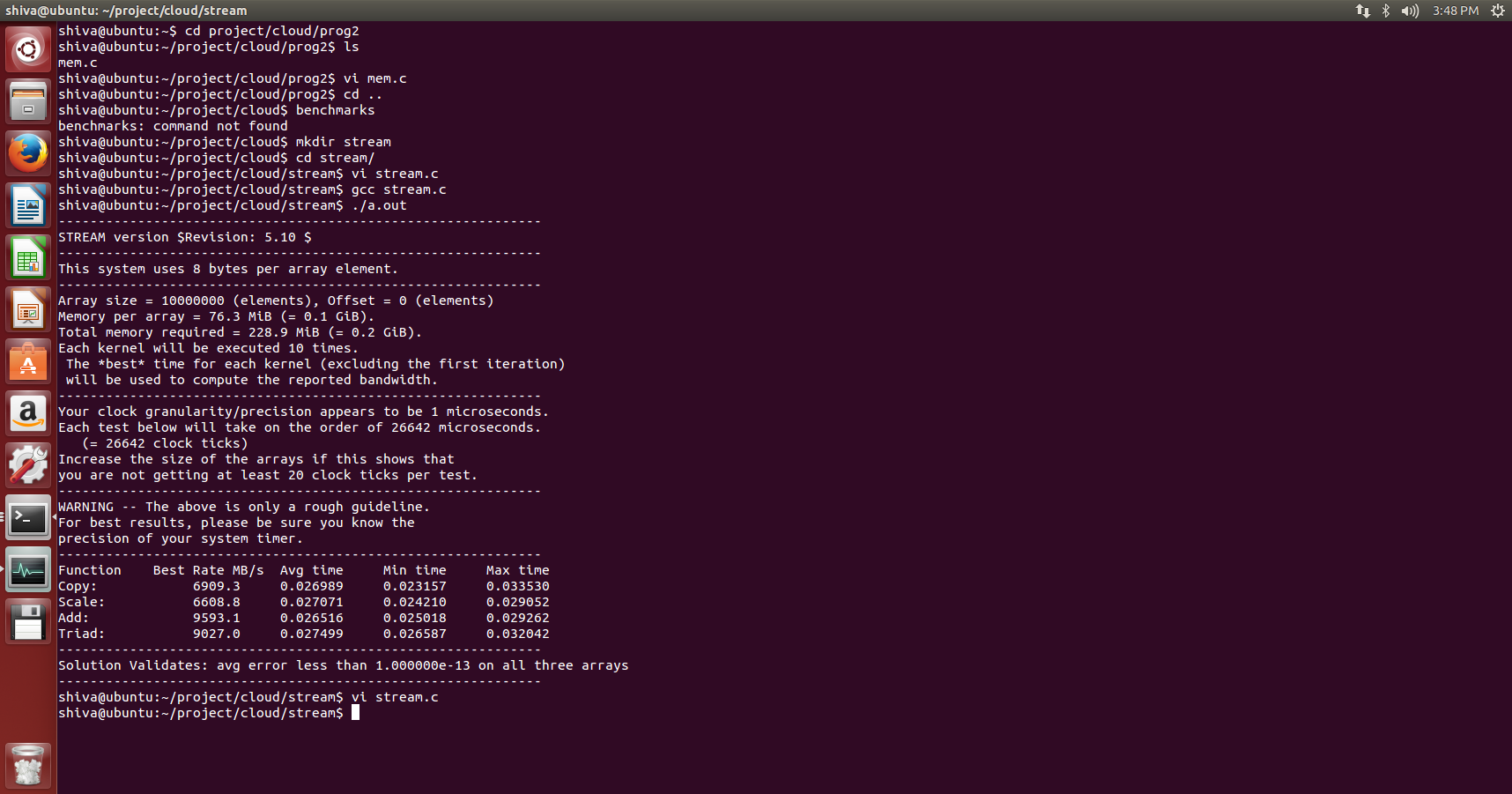
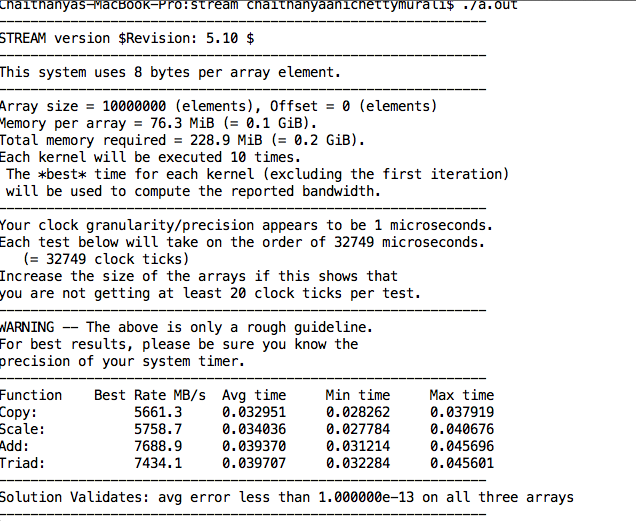
**Graph:** Depicting random latency

**X-axis:** Block size in Bytes

**Y-axis:** Latency in micro second/byte

**3 .D) The optimal number of concurrency to get the best performance is 1.**

STREAM BENCHMARKING TOOL RESULTS



**NETWORK BENCHMARK**

**5.A) B) C)**Analyzing the network performance by calculating throughput and latency using TCP and UDP protocol.

**UDP RESULT**

|  |  |  |
| --- | --- | --- |
| **Size** | **1 thread (Mb/s)** | **2 threads(Mb/s)** |
| 1B | 0.021 | 0.0206 |
| 1KB | 25 | 18 |
| 64KB | 521 | 493 |

* The size is given in bytes.
* The throughput is obtained in Mb/s

**GRAPH:** Shows the throughput for UDP protocol.

**X-axis** :Block size(bytes)

**Y-axis**: Throughput(Mb/s)

**TCP RESULT**

|  |  |  |
| --- | --- | --- |
| **SIZE** | **1 Thread(Mb/s)** | **2 Threads(Mb/s)** |
| 1B | 0.025 | 0.023 |
| 1KB | 29 | 22 |
| 64KB | 578 | 536 |

* The size is given in bytes.
* The throughput is obtained in Mb/s

**GRAPH:** Shows the throughput for TCP protocol.

**X-axis** : Block size(bytes)

**Y-axis**: Throughput(Mb/s)

**LATENCY**

**UDP Latency:** 0.00389999999868 milliseconds

**TCP Latency:** 0.00648498535156 milliseconds

**Theoretical value for optimal TCP window size**

The **theoretical value for optimal TCP window size** is

bandwidth delay product=Bottle neck bandwidth\*round trip time

bottleneck bandwidth=578 MB/s round trip time=105 ms

The optimal performance was achieved with a window size of 60 KB.

IPERF BENCHMARKING TOOL RESULTS

