REPORT -PA1

DESIGN

VIKRAM G AND MOUNA GIRI

2017

# **REPORT**

This programming assignment consists of four components:

1. CPU Benchmark
2. Memory Benchmark
3. Disk Benchmark
4. Network Benchmark

## CPU Benchmark:

1. VALUES FOR GIOPS AND GFOPS for 1,2,4 and 8 Thread

|  |  |  |
| --- | --- | --- |
| NO OF THREADS | Giga IOPS | Giga FLOPS |
| 1 | 1.578502 | 2.067615 |
| 2 | 3.825424 | 4.000017 |
| 4 | 3.773795 | 3.805590 |
| 8 | 2.863006 | 2.991353 |

GRAPH:

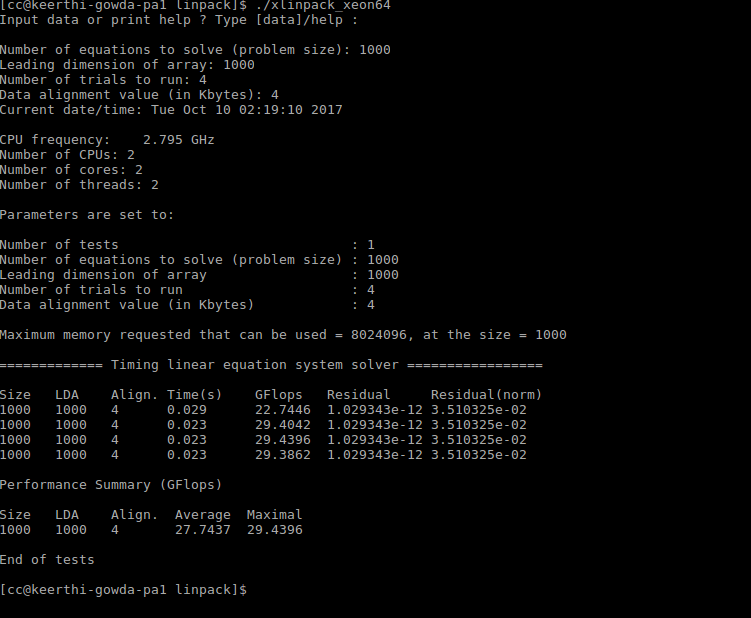
1. Compute the theoretical peak performance of your processor in flops/sec

Theoretical peak performance CPU:

CPU speed=number of cores\* IPC(instructions per cycle)\*clock speed;

Speed=2\*4\*2.795Ghz=22.36 GFlops

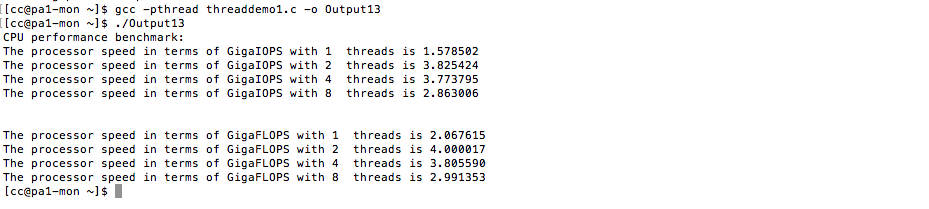
1. LINPACK:



CONCLUSION:

We can see that the GIOPS and GFLOPS are increasing with the increase in the number of threads and it decreases after a certain point of concurrency.

SCREENSHOT :



## 2. Memory Benchmark:

1. 8 B LATENCY

|  |  |  |  |
| --- | --- | --- | --- |
| **Latency** |  |  |  |
| **Number of threads** | **Seq RW** | **Seq W** | **Ran W** |
| **1** | 0.000002 | 0.000002 | 0.000015 |
| **2** | 0.000001 | 0.000001 | 0.000034 |
| **4** | 0.000001 | 0.000001 | 0.000036 |
| **8** | 0.000001 | 0.000001 | 0.000039 |

1. 8b – throughput

|  |  |  |  |
| --- | --- | --- | --- |
| **Throughput** |  |  |  |
| **Number of threads** | **Seq RW** | **Seq W** | **Ran W** |
| **1** | 420.214783 | 507.791595 | 63.108192 |
| **2** | 852.503784 | 1022.546387 | 28.441065 |
| **4** | 874.277954 | 1079.179321 | 26.628847 |
| **8** | 892.942627 | 1012.896057 | 24.193817 |

1. 8KB – THROUGHPUT

|  |  |  |  |
| --- | --- | --- | --- |
| **Number of Threads** | **Seq RW** | **Seq W** | **Ran W** |
| **1** | 2745.236084 | 3940.436035 | 3015.912842 |
| **2** | 6876.717773 | 9223.484375 | 7803.40918 |
| **4** | 8135.304199 | 11397.90332 | 9363.463867 |
| **8** | 6768.013184 | 13363.65332 | 8568.579102 |

1. 8MB THROUGHPUT

|  |  |  |  |
| --- | --- | --- | --- |
| **Number of Threads** | **Seq RW** | **Seq W** | **Ran W** |
| **1** | 2576.082031 | 4051.25415 | 4803.891602 |
| **2** | 6389.178711 | 9776.725586 | 9611.345703 |
| **4** | 8976.748047 | 13403.99219 | 12868.6543 |
| **8** | 10030.27148 | 11660.20801 | 12111.06445 |

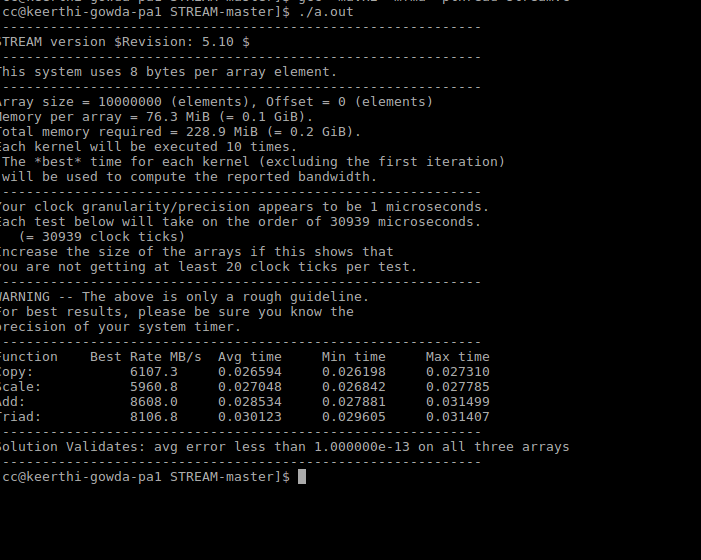
1. 80MB THROUGHPUT

|  |  |  |  |
| --- | --- | --- | --- |
| **Number of Threads** | **Seq RW** | **Seq W** | **Ran W** |
| **1** | 2688.454346 | 4159.290527 | 5471.248535 |
| **2** | 7169.160156 | 11011.5625 | 10397.96777 |
| **4** | 10624.94727 | 14264.97266 | 14478.45801 |
| **8** | 13222.74902 | 15960.06055 | 15960.06055 |

Theoretical Calculations:

Memory Bandwidth=Data transfer per clock \*Clock frequency \*memory bus width \* number of interfaces =1\*1200\*64\*2=19.2 GBPS

STREAM BENCHMARK:

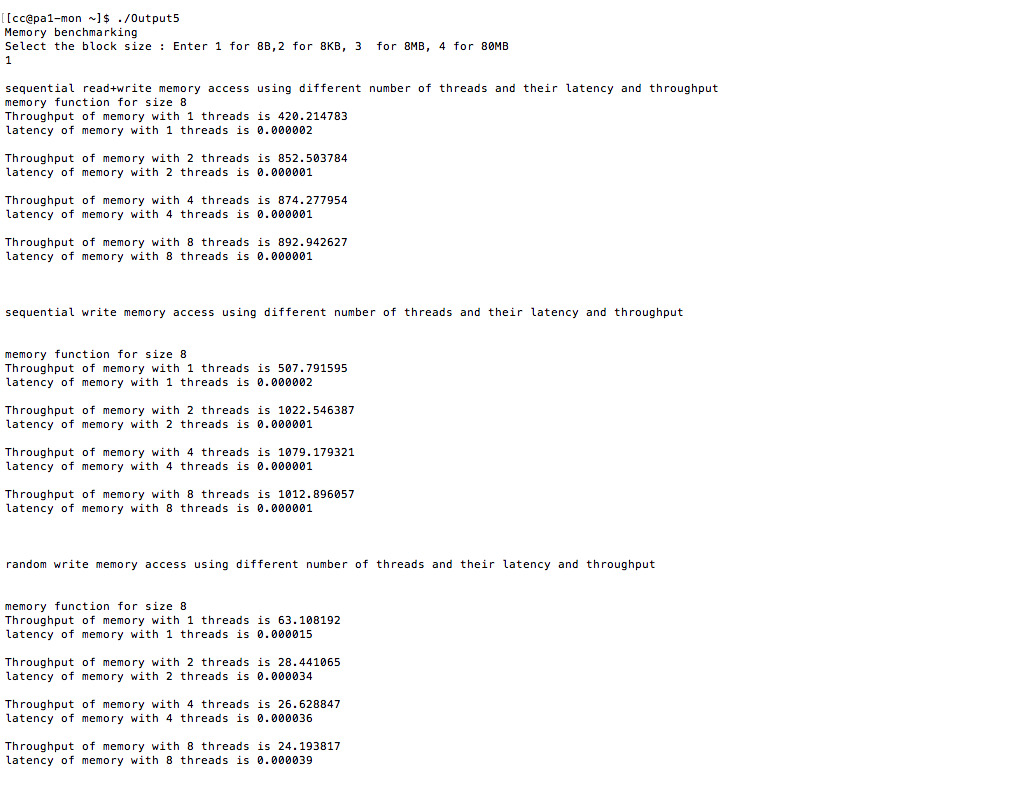


CONCLUSION:

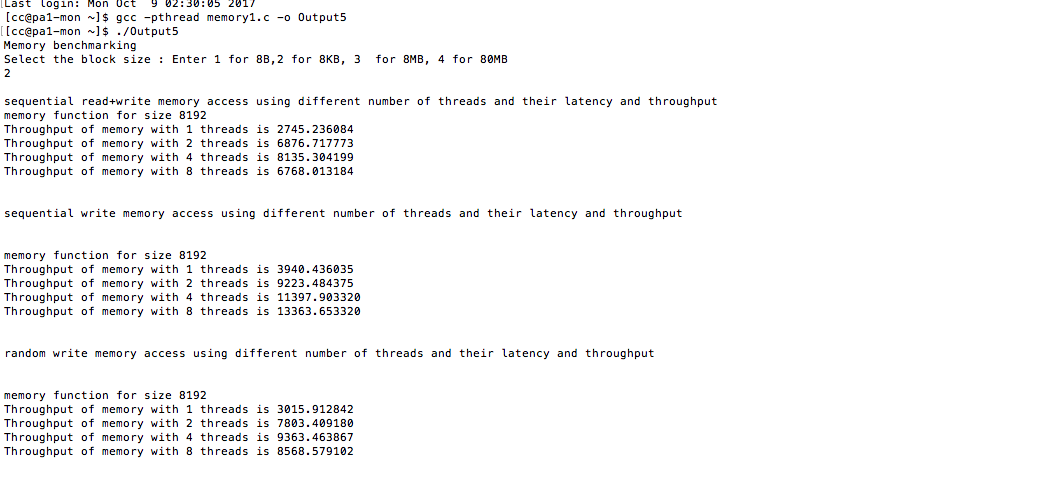
The Throughput increases with increase in the number of threads up to four threads. After 4 threads, it drops.

SCREENSHOT:

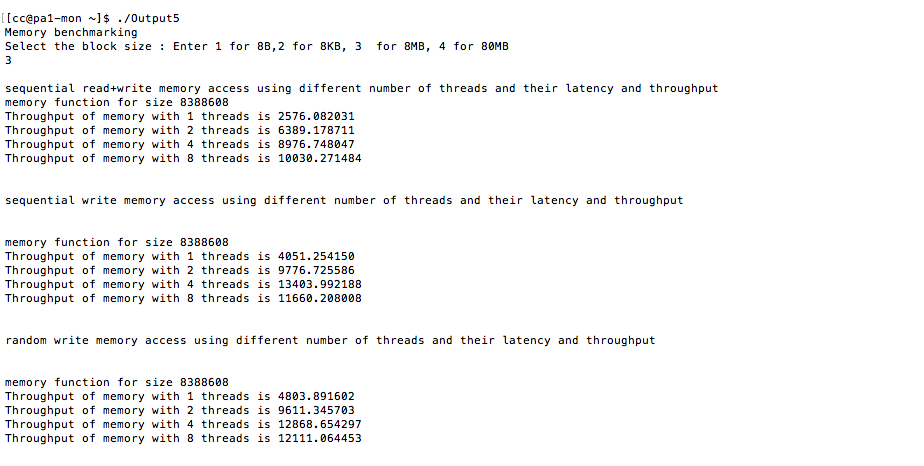
1. FOR 8b BLOCK

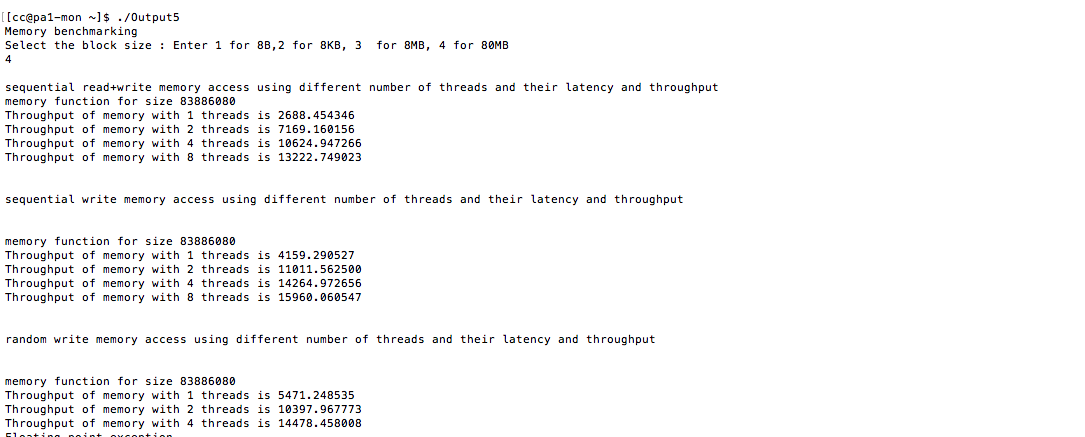


1. FOR 80kb BLOCK



1. FOR 8MB BLOCK



1. FOR 80M

## Disk Benchmark:

VALUES:

1. FOR DISK READ AND WRITE – THROUGHPUT IN MB/sec

|  |  |  |  |
| --- | --- | --- | --- |
| NO OF THREADS | 8KB | BMB | 80MB |
| 1 | 45.486497932431114 | 193.59412797853287 | 157.94600017455068 |
| 2 | 35.29965254335515 | 1054.411733345492 | 637.0114761866251 |
| 4 | 16.923403031713928 | 262.4263980126305 | 348.85840055660117 |
| 8 | 8.728010928973152 | 117.57765573492941 | 247.00810599409604 |

1. FOR DISK READ AND WRITE – LATENCY in ms

|  |  |
| --- | --- |
| NO OF THREADS | 8B |
| 1 | 2.071106208153069E-4 |
| 2 | 2.5440730676427485E-4 |
| 4 | 5.30984411100857E-4 |
| 8 | 0.0010806312774410472 |

1. SEQUENTIAL READ - THROUGHPUT IN MB/SEC

|  |  |  |  |
| --- | --- | --- | --- |
| **Number of Threads** | **80b** | **8MB** | **80MB** |
| **1** | 91.7936 | 3298.2699 | 4112.1067 |
| **2** | 85.2066 | 3355.378 | 4085.146 |
| **4** | 42.3547 | 1986.3583 | 2498.3119 |
| **8** | 20.999 | 2608.3442 | 3646.3499 |

1. SEQUENTIAL READ – LATENCY in ms

|  |  |
| --- | --- |
| **Number of threads** | **Seq R** |
| **1** | 1.0142 |
| **2** | 1.1079 |
| **4** | 2.2187 |
| **8** | 4.385 |

1. RANDOM READ - THROUGHPUT IN MB/SEC

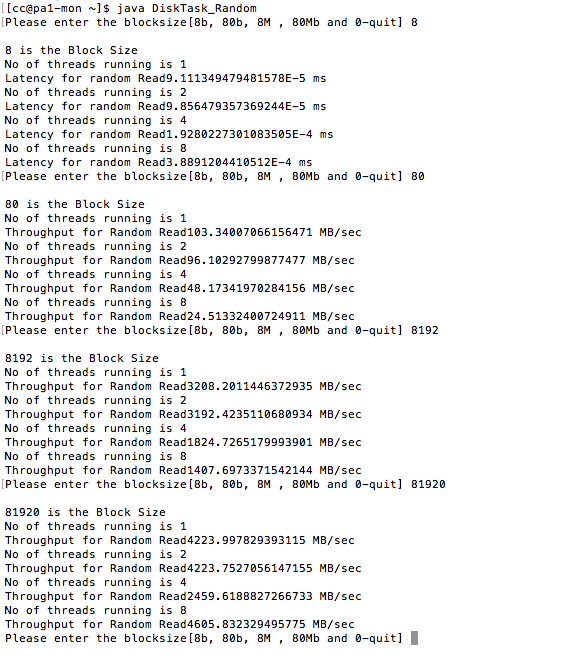
|  |  |  |  |
| --- | --- | --- | --- |
| **Number of Threads** | **80b** | **8MB** | **80MB** |
| **1** | 103.34 | 3208.2011 | 4223.9978 |
| **2** | 96.1029 | 3192.4235 | 4223.7527 |
| **4** | 48.1734 | 1824.7265 | 2459.6188 |
| **8** | 24.5133 | 1407.6973 | 4605.8323 |

1. RANDOM READ – LATENCY in ms

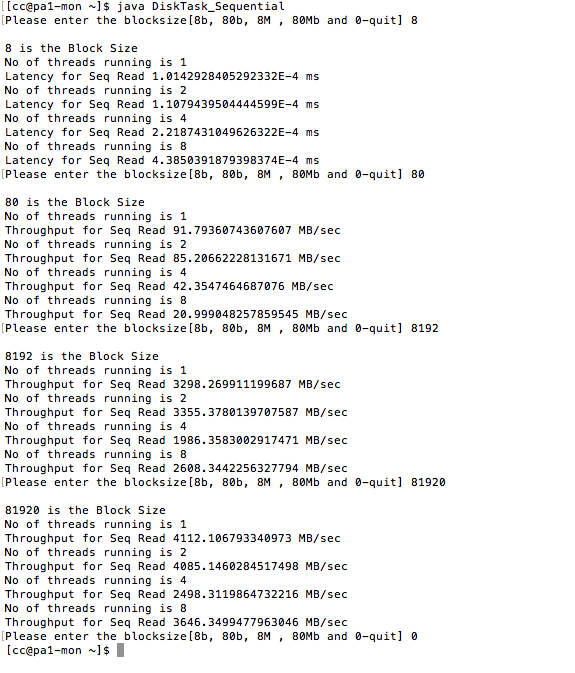
|  |  |
| --- | --- |
| **Number of Threads** | **8b** |
| **1** | 9.1113 |
| **2** | 9.8564 |
| **4** | 1.928 |
| **8** | 3.8891 |

SCREENSHOTS:

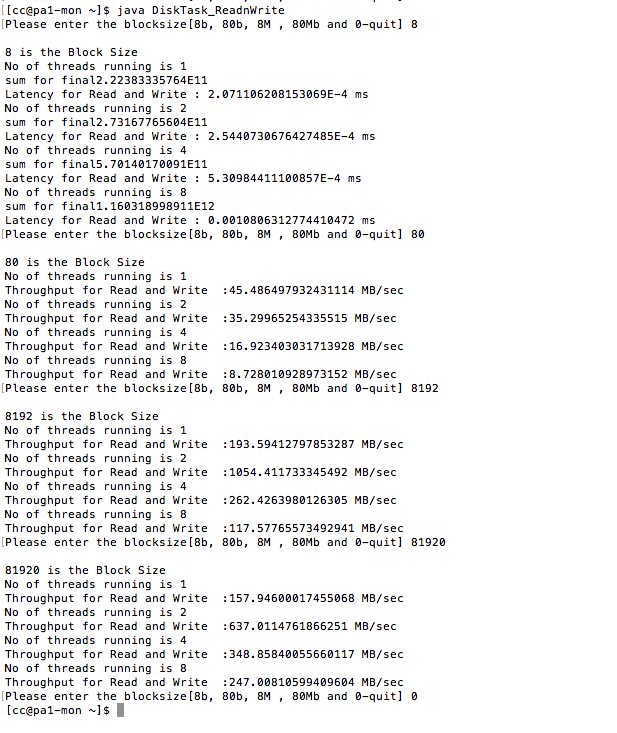
1. FOR RANDOM READ



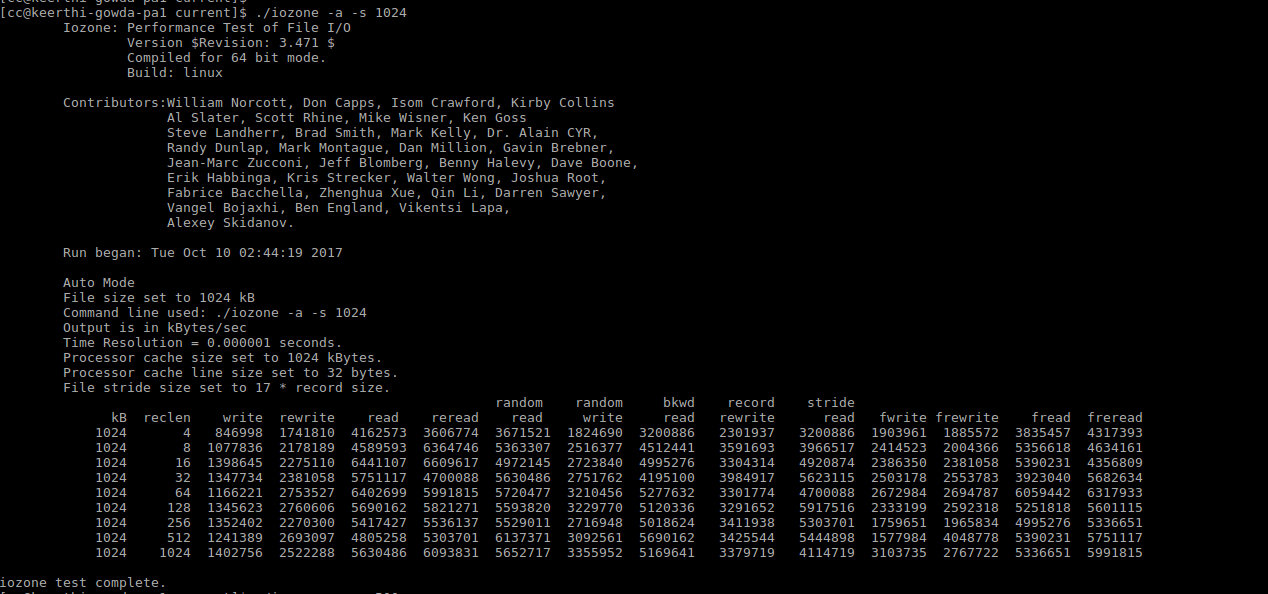
1. FOR SEQUENTIAL READ



1. FOR READ AND WRITE SEQUENTIAL



IOZONE:



CONCLUSION:

The sequential read is faster than random read.

And it is HDD Disk.

## NETWORK Benchmark:

|  |  |
| --- | --- |
| NO OF THREAD | THROUGHPUT |
| 1 | 472.03 |
| 2 | 458.48 |
| 4 | 722.93 |
| 8 | 508.74 |

SCREENSHOT:

