

2017

REPORT -PA1

DESIGN

VIKRAM G AND MOUNA GIRI

REPORT

This programming assignment consists of four components:

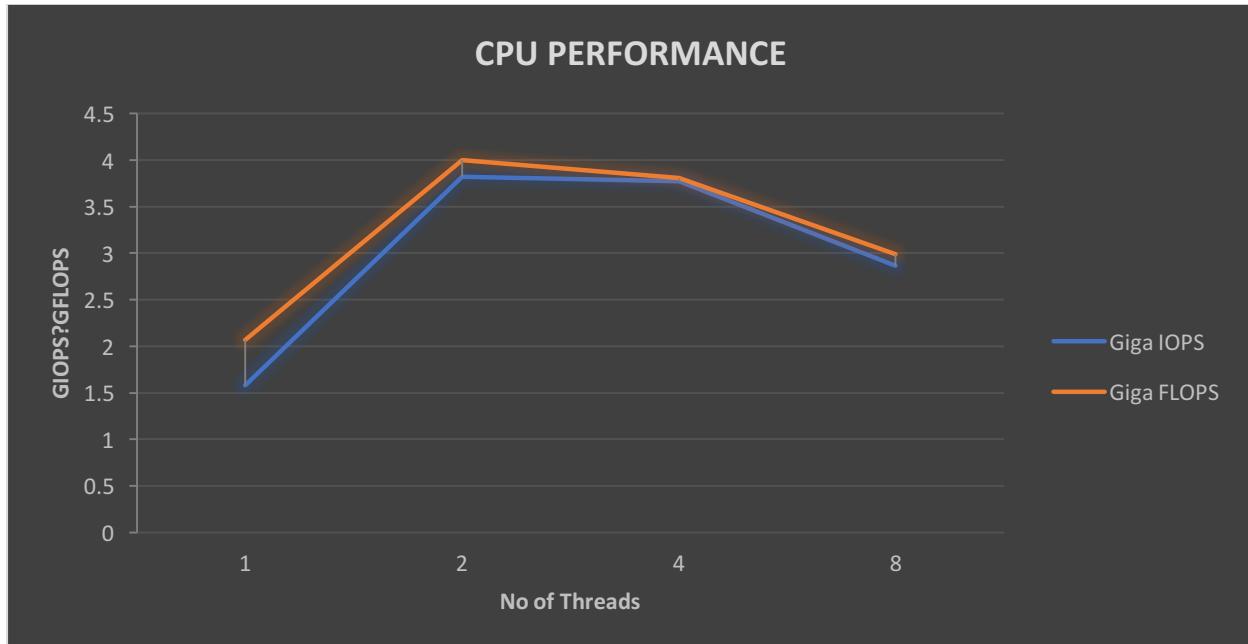
1. CPU Benchmark
2. Memory Benchmark
3. Disk Benchmark
4. Network Benchmark

CPU Benchmark:

- a. VALUES FOR GIOPS AND GFLOPS 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:



b. 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

c. LINPACK:

```
Input data or print help ? Type [data]/help :

Number of equations to solve (problem size): 1000
Leading dimension of array: 1000
Number of trials to run: 4
Data alignment value (in Kbytes): 4
Current date/time: Tue Oct 10 02:19:10 2017

CPU frequency: 2.795 GHz
Number of CPUs: 2
Number of cores: 2
Number of threads: 2

Parameters are set to:

Number of tests : 1
Number of equations to solve (problem size) : 1000
Leading dimension of array : 1000
Number of trials to run : 4
Data alignment value (in Kbytes) : 4

Maximum memory requested that can be used = 8024096, at the size = 1000

===== Timing linear equation system solver =====

Size   LDA   Align. Time(s)    GFlops   Residual   Residual(norm)
1000   1000   4      0.029     22.7446  1.029343e-12 3.510325e-02
1000   1000   4      0.023     29.4042  1.029343e-12 3.510325e-02
1000   1000   4      0.023     29.4396  1.029343e-12 3.510325e-02
1000   1000   4      0.023     29.3862  1.029343e-12 3.510325e-02

Performance Summary (GFlops)

Size   LDA   Align. Average Maximal
1000   1000   4      27.7437  29.4396

End of tests
```

CONCLUSION:

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

SCREENSHOT :

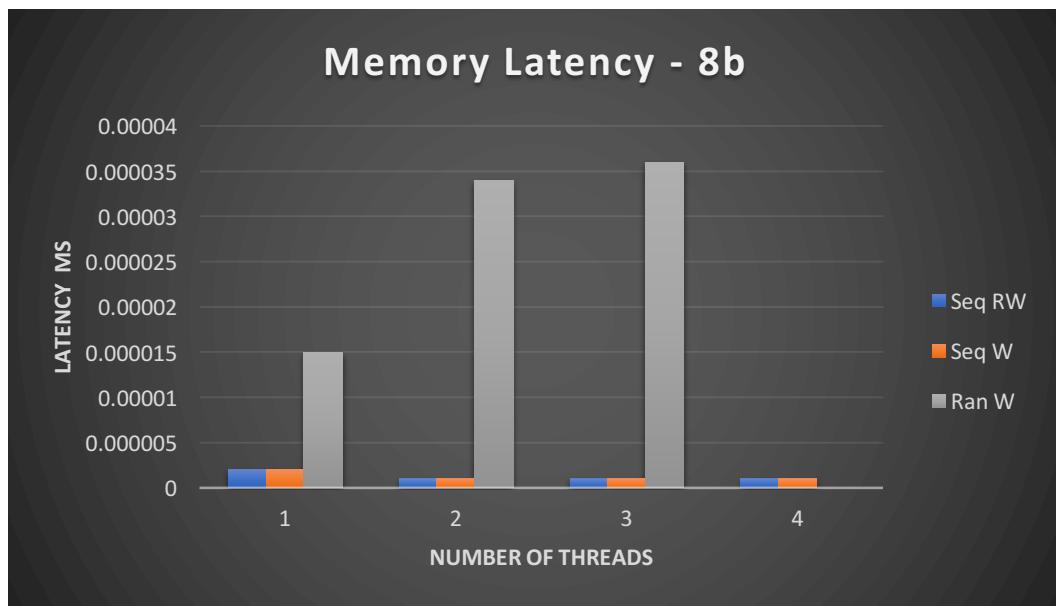
```
[cc@pa1-mon ~]$ gcc -pthread threaddemo1.c -o Output13
[cc@pa1-mon ~]$ ./Output13
CPU performance benchmark:
The processor speed in terms of GigaIOPS with 1 threads is 1.578502
The processor speed in terms of GigaIOPS with 2 threads is 3.825424
The processor speed in terms of GigaIOPS with 4 threads is 3.773795
The processor speed in terms of GigaIOPS with 8 threads is 2.863006

The processor speed in terms of GigaFLOPS with 1 threads is 2.067615
The processor speed in terms of GigaFLOPS with 2 threads is 4.000017
The processor speed in terms of GigaFLOPS with 4 threads is 3.805590
The processor speed in terms of GigaFLOPS with 8 threads is 2.991353
[cc@pa1-mon ~]$
```

2. Memory Benchmark:

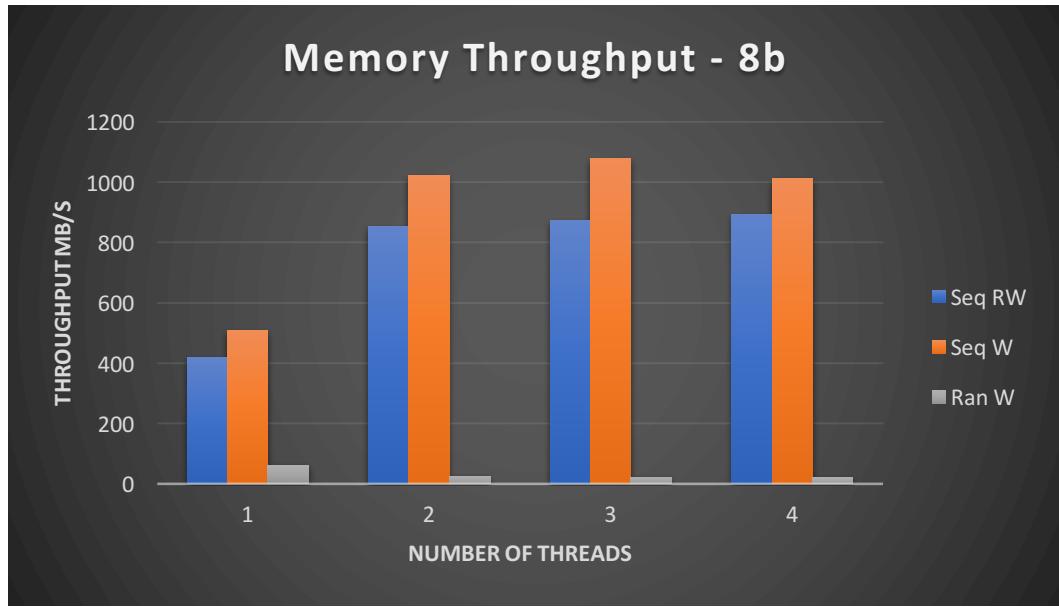
a. 8 B LATENCY

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



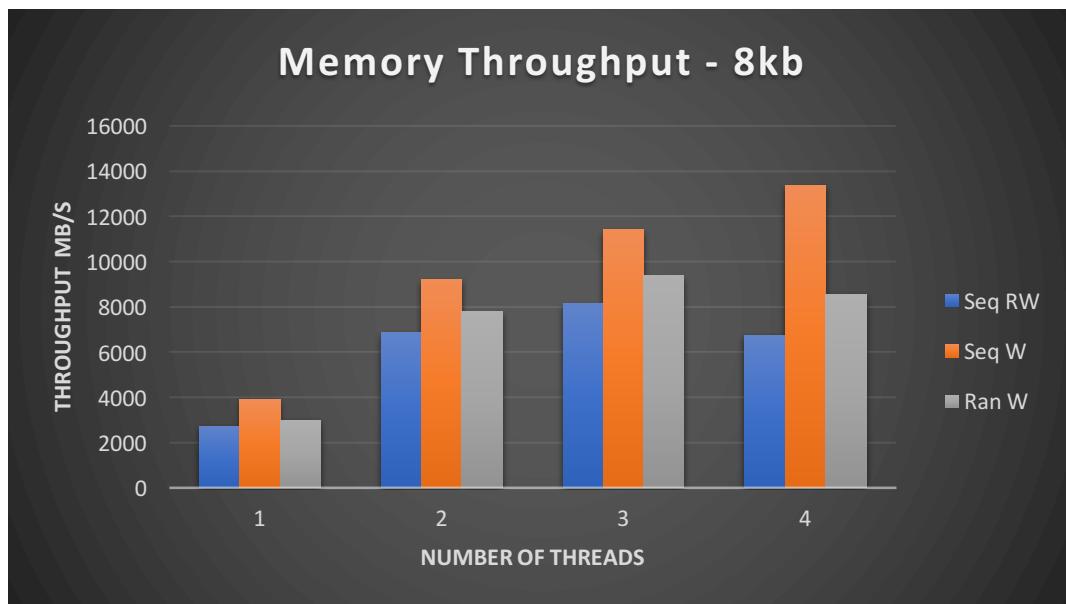
b. 8b – throughput

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



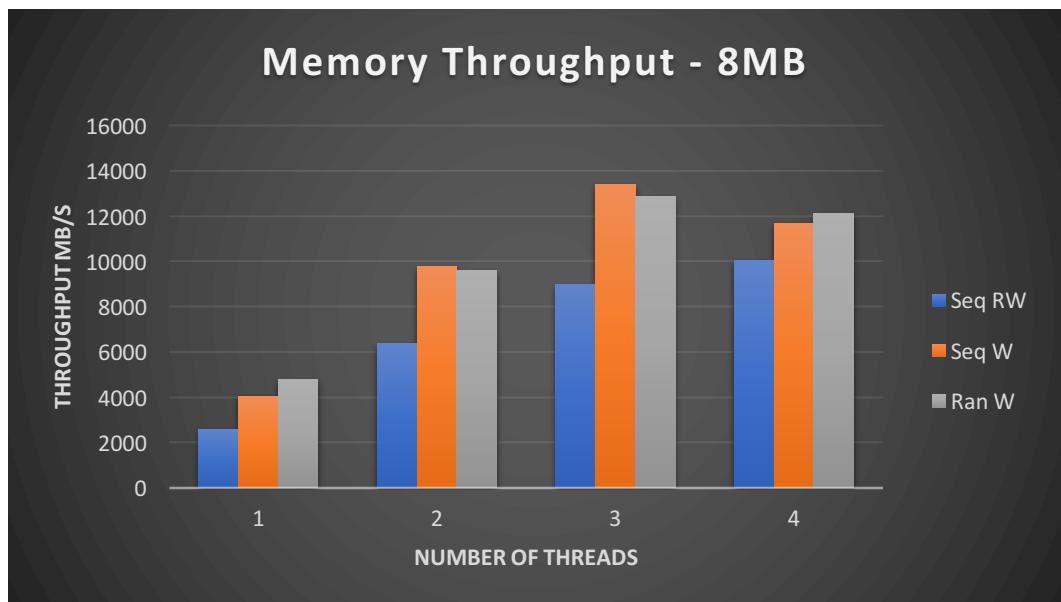
c. 8KB – THROUGHPUT

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



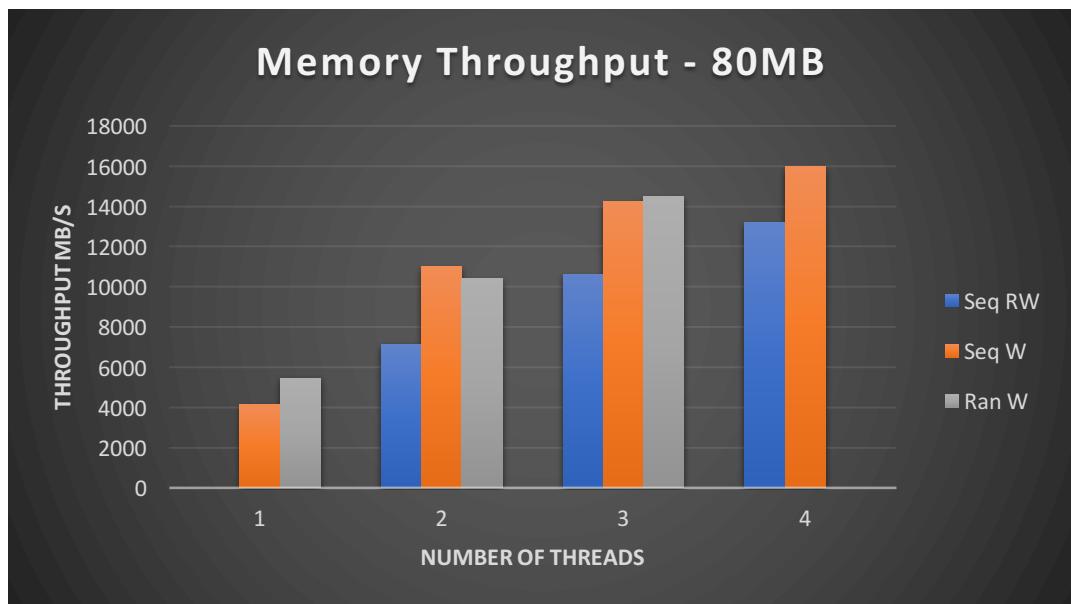
d. 8MB THROUGHPUT

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



e. 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 = $1 * 1200 * 64 * 2 = 19.2 \text{ GBPS}$

STREAM BENCHMARK:

```

STREAM version $Revision: 5.10 $
-----
This system uses 8 bytes per array element.
-----
array size = 10000000 (elements), Offset = 0 (elements)
memory per array = 76.3 MiB (= 0.1 GiB).
total memory required = 228.9 MiB (= 0.2 GiB).
each kernel will be executed 10 times.
The *best* time for each kernel (excluding the first iteration)
will be used to compute the reported bandwidth.
-----
our clock granularity/precision appears to be 1 microseconds.
each test below will take on the order of 30939 microseconds.
(= 30939 clock ticks)
increase the size of the arrays if this shows that
you are not getting at least 20 clock ticks per test.
-----
WARNING -- The above is only a rough guideline.
For best results, please be sure you know the
precision of your system timer.
-----
Function      Best Rate MB/s    Avg time      Min time      Max time
Copy:          6107.3        0.026594     0.026198     0.027310
Scale:         5960.8        0.027048     0.026842     0.027785
Add:           8608.0        0.028534     0.027881     0.031499
Triad:         8106.8        0.030123     0.029605     0.031407
-----
Solution Validates: avg error less than 1.000000e-13 on all three arrays

```

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

```
[cc@pa1-mon ~]$ ./Output5
Memory benchmarking
Select the block size : Enter 1 for 8B,2 for 8KB, 3 for 8MB, 4 for 80MB
1

sequential read+write memory access using different number of threads and their latency and throughput
memory function for size 8
Throughput of memory with 1 threads is 420.214783
latency of memory with 1 threads is 0.000002

Throughput of memory with 2 threads is 852.503784
latency of memory with 2 threads is 0.000001

Throughput of memory with 4 threads is 874.277954
latency of memory with 4 threads is 0.000001

Throughput of memory with 8 threads is 892.942627
latency of memory with 8 threads is 0.000001

sequential write memory access using different number of threads and their latency and throughput

memory function for size 8
Throughput of memory with 1 threads is 507.791595
latency of memory with 1 threads is 0.000002

Throughput of memory with 2 threads is 1022.546387
latency of memory with 2 threads is 0.000001

Throughput of memory with 4 threads is 1079.179321
latency of memory with 4 threads is 0.000001

Throughput of memory with 8 threads is 1012.896057
latency of memory with 8 threads is 0.000001

random write memory access using different number of threads and their latency and throughput

memory function for size 8
Throughput of memory with 1 threads is 63.108192
latency of memory with 1 threads is 0.000015

Throughput of memory with 2 threads is 28.441065
latency of memory with 2 threads is 0.000034

Throughput of memory with 4 threads is 26.628847
latency of memory with 4 threads is 0.000036

Throughput of memory with 8 threads is 24.193817
latency of memory with 8 threads is 0.000039
```

2. FOR 80kb BLOCK

```
Last login: Mon Oct  9 02:30:05 2017
[cc@pa1-mon ~]$ gcc -pthread memory1.c -o Output5
[cc@pa1-mon ~]$ ./Output5
Memory benchmarking
Select the block size : Enter 1 for 8B,2 for 8KB, 3 for 8MB, 4 for 80MB
2

sequential read+write memory access using different number of threads and their latency and throughput
memory function for size 8192
Throughput of memory with 1 threads is 2745.236084
Throughput of memory with 2 threads is 6876.717773
Throughput of memory with 4 threads is 8135.304199
Throughput of memory with 8 threads is 6768.013184

sequential write memory access using different number of threads and their latency and throughput

memory function for size 8192
Throughput of memory with 1 threads is 3940.436035
Throughput of memory with 2 threads is 9223.484375
Throughput of memory with 4 threads is 11397.903320
Throughput of memory with 8 threads is 13363.653320

random write memory access using different number of threads and their latency and throughput

memory function for size 8192
Throughput of memory with 1 threads is 3015.912842
Throughput of memory with 2 threads is 7803.409180
Throughput of memory with 4 threads is 9363.463867
Throughput of memory with 8 threads is 8568.579102
```

3. FOR 8MB BLOCK

```
[[cc@pa1-mon ~]$ ./Output5
Memory benchmarking
Select the block size : Enter 1 for 8B,2 for 8KB, 3 for 8MB, 4 for 80MB
3

sequential read+write memory access using different number of threads and their latency and throughput
memory function for size 8388608
Throughput of memory with 1 threads is 2576.082031
Throughput of memory with 2 threads is 6389.178711
Throughput of memory with 4 threads is 8976.748047
Throughput of memory with 8 threads is 10030.271484

sequential write memory access using different number of threads and their latency and throughput

memory function for size 8388608
Throughput of memory with 1 threads is 4051.254150
Throughput of memory with 2 threads is 9776.725586
Throughput of memory with 4 threads is 13403.992188
Throughput of memory with 8 threads is 11660.208008

random write memory access using different number of threads and their latency and throughput

memory function for size 8388608
Throughput of memory with 1 threads is 4803.891602
Throughput of memory with 2 threads is 9611.345703
Throughput of memory with 4 threads is 12868.654297
Throughput of memory with 8 threads is 12111.064453
```

4. FOR 80M

```
[cc@pa1-mon ~]$ ./Output5
Memory benchmarking
Select the block size : Enter 1 for 8B,2 for 8KB, 3 for 8MB, 4 for 80MB
4

sequential read+write memory access using different number of threads and their latency and throughput
memory function for size 83886080
Throughput of memory with 1 threads is 2688.454346
Throughput of memory with 2 threads is 7169.160156
Throughput of memory with 4 threads is 10624.947266
Throughput of memory with 8 threads is 13222.749023

sequential write memory access using different number of threads and their latency and throughput

memory function for size 83886080
Throughput of memory with 1 threads is 4159.290527
Throughput of memory with 2 threads is 11011.562500
Throughput of memory with 4 threads is 14264.972656
Throughput of memory with 8 threads is 15960.060547

random write memory access using different number of threads and their latency and throughput

memory function for size 83886080
Throughput of memory with 1 threads is 5471.248535
Throughput of memory with 2 threads is 10397.967773
Throughput of memory with 4 threads is 14478.458008
Execution point execution
```

Disk Benchmark:

VALUES:

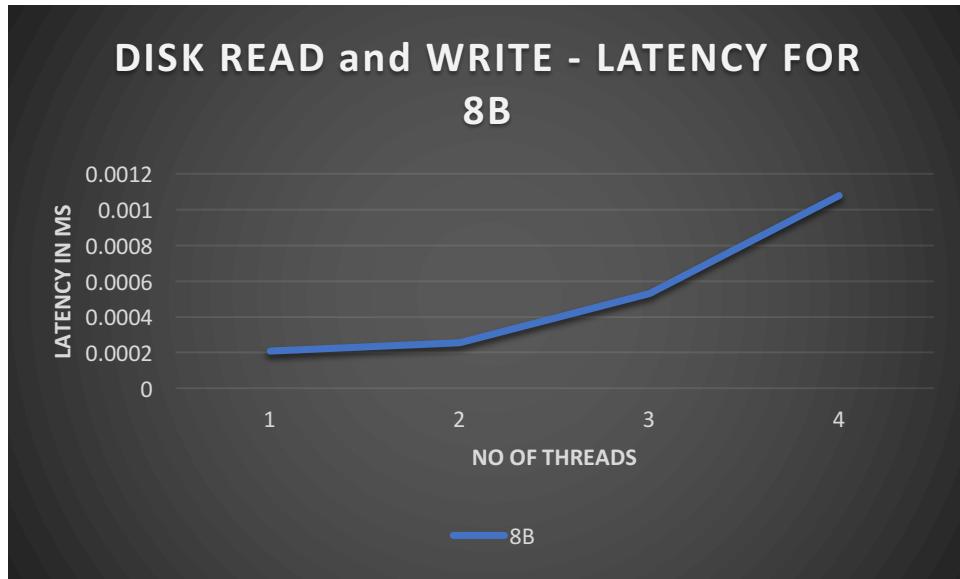
1. FOR DISK READ AND WRITE – THROUGHPUT IN MB/sec

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



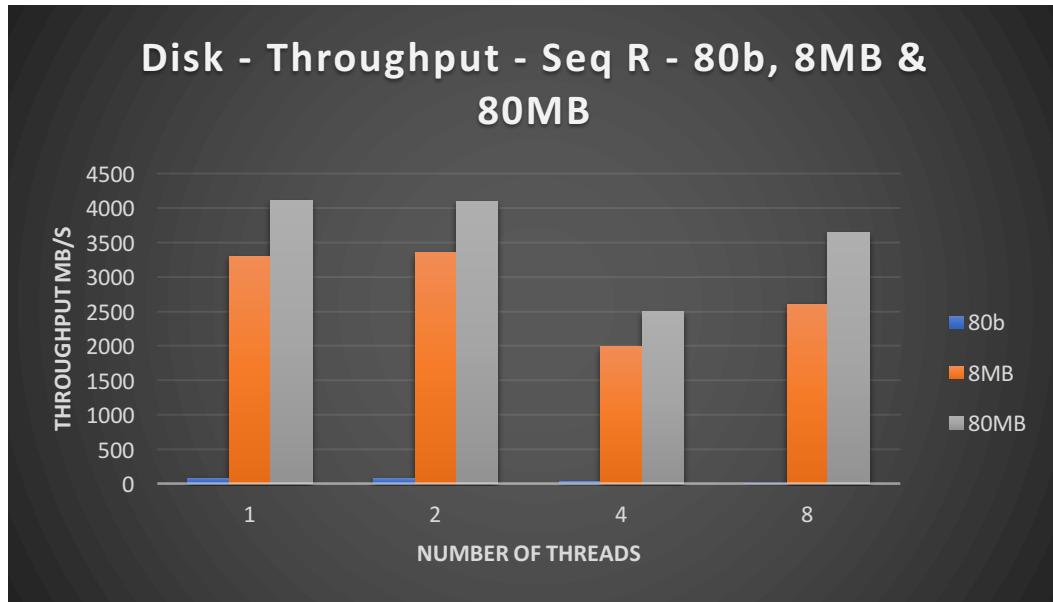
2. FOR DISK READ AND WRITE – LATENCY in ms

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



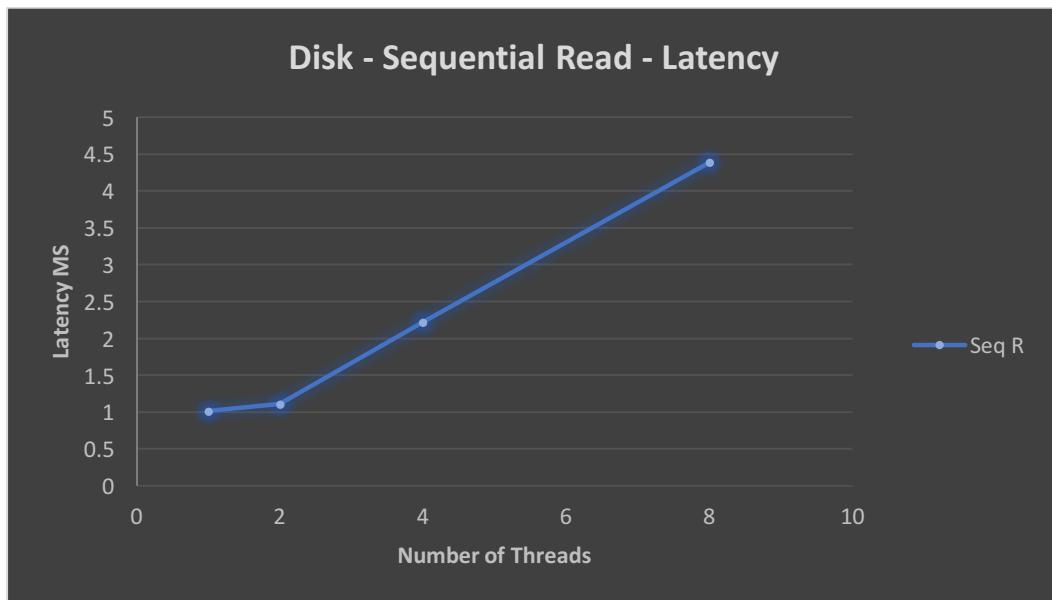
3. SEQUENTIAL READ - THROUGHPUT IN MB/SEC

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



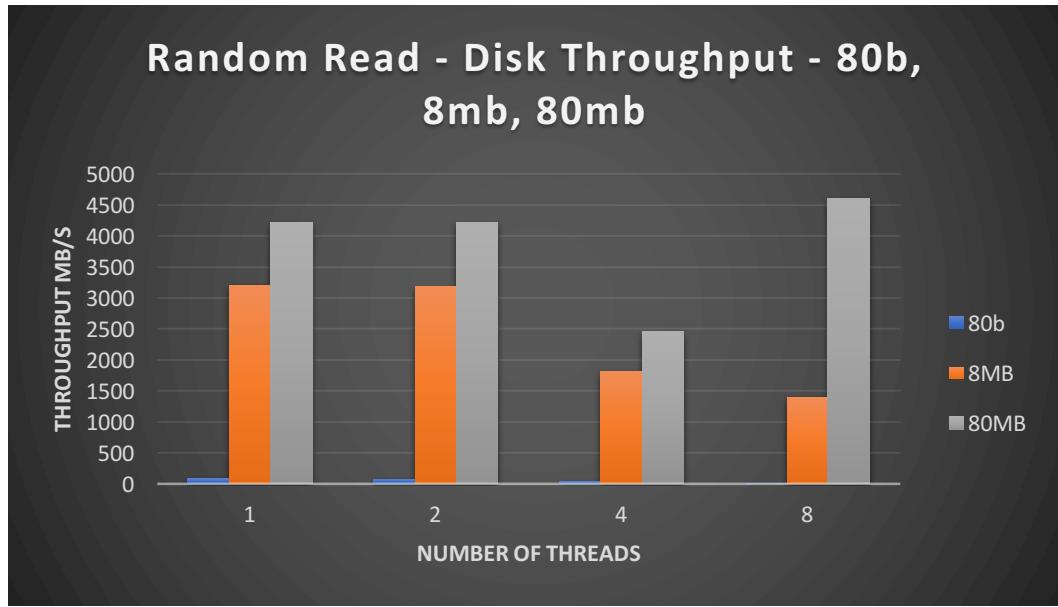
4. SEQUENTIAL READ – LATENCY in ms

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



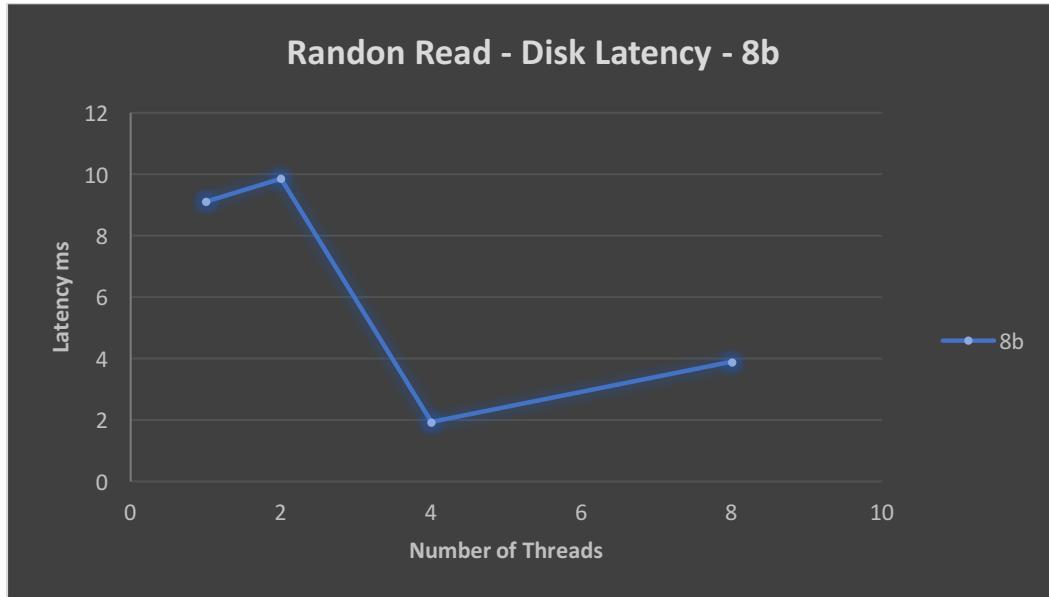
5. RANDOM READ - THROUGHPUT IN MB/SEC

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



6. 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

```
[cc@pa1-mon ~]$ java DiskTask_Random
[Please enter the blocksize[8b, 80b, 8M , 80Mb and 0-quit] 8

8 is the Block Size
No of threads running is 1
Latency for random Read9.111349479481578E-5 ms
No of threads running is 2
Latency for random Read9.856479357369244E-5 ms
No of threads running is 4
Latency for random Read1.9280227301083505E-4 ms
No of threads running is 8
Latency for random Read3.8891204410512E-4 ms
[Please enter the blocksize[8b, 80b, 8M , 80Mb and 0-quit] 80

80 is the Block Size
No of threads running is 1
Throughput for Random Read103.34007066156471 MB/sec
No of threads running is 2
Throughput for Random Read96.10292799877477 MB/sec
No of threads running is 4
Throughput for Random Read48.17341970284156 MB/sec
No of threads running is 8
Throughput for Random Read24.51332400724911 MB/sec
[Please enter the blocksize[8b, 80b, 8M , 80Mb and 0-quit] 8192

8192 is the Block Size
No of threads running is 1
Throughput for Random Read3208.2011446372935 MB/sec
No of threads running is 2
Throughput for Random Read3192.4235110680934 MB/sec
No of threads running is 4
Throughput for Random Read1824.7265179993901 MB/sec
No of threads running is 8
Throughput for Random Read1407.6973371542144 MB/sec
[Please enter the blocksize[8b, 80b, 8M , 80Mb and 0-quit] 81920

81920 is the Block Size
No of threads running is 1
Throughput for Random Read4223.997829393115 MB/sec
No of threads running is 2
Throughput for Random Read4223.7527056147155 MB/sec
No of threads running is 4
Throughput for Random Read2459.6188827266733 MB/sec
No of threads running is 8
Throughput for Random Read4605.832329495775 MB/sec
[Please enter the blocksize[8b, 80b, 8M , 80Mb and 0-quit]
```

2. FOR SEQUENTIAL READ

```
[[cc@pa1-mon ~]$ java DiskTask_Sequential
[Please enter the blocksize[8b, 80b, 8M , 80Mb and 0-quit] 8

8 is the Block Size
No of threads running is 1
Latency for Seq Read 1.0142928405292332E-4 ms
No of threads running is 2
Latency for Seq Read 1.1079439504444599E-4 ms
No of threads running is 4
Latency for Seq Read 2.2187431049626322E-4 ms
No of threads running is 8
Latency for Seq Read 4.3850391879398374E-4 ms
[Please enter the blocksize[8b, 80b, 8M , 80Mb and 0-quit] 80

80 is the Block Size
No of threads running is 1
Throughput for Seq Read 91.79360743607607 MB/sec
No of threads running is 2
Throughput for Seq Read 85.20662228131671 MB/sec
No of threads running is 4
Throughput for Seq Read 42.3547464687076 MB/sec
No of threads running is 8
Throughput for Seq Read 20.999048257859545 MB/sec
[Please enter the blocksize[8b, 80b, 8M , 80Mb and 0-quit] 8192

8192 is the Block Size
No of threads running is 1
Throughput for Seq Read 3298.269911199687 MB/sec
No of threads running is 2
Throughput for Seq Read 3355.3780139707587 MB/sec
No of threads running is 4
Throughput for Seq Read 1986.3583002917471 MB/sec
No of threads running is 8
Throughput for Seq Read 2608.3442256327794 MB/sec
[Please enter the blocksize[8b, 80b, 8M , 80Mb and 0-quit] 81920

81920 is the Block Size
No of threads running is 1
Throughput for Seq Read 4112.106793340973 MB/sec
No of threads running is 2
Throughput for Seq Read 4085.1460284517498 MB/sec
No of threads running is 4
Throughput for Seq Read 2498.3119864732216 MB/sec
No of threads running is 8
Throughput for Seq Read 3646.3499477963046 MB/sec
[Please enter the blocksize[8b, 80b, 8M , 80Mb and 0-quit] 0
[cc@pa1-mon ~]$ ]
```

3. FOR READ AND WRITE SEQUENTIAL

```
[cc@pa1-mon ~]$ java DiskTask_ReadnWrite
[Please enter the blocksize[8b, 80b, 8M , 80Mb and 0-quit] 8

8 is the Block Size
No of threads running is 1
sum for final2.22383335764E11
Latency for Read and Write : 2.071106208153069E-4 ms
No of threads running is 2
sum for final2.73167765604E11
Latency for Read and Write : 2.5440730676427485E-4 ms
No of threads running is 4
sum for final5.70140170091E11
Latency for Read and Write : 5.30984411100857E-4 ms
No of threads running is 8
sum for final1.160318998911E12
Latency for Read and Write : 0.0010806312774410472 ms
[Please enter the blocksize[8b, 80b, 8M , 80Mb and 0-quit] 80

80 is the Block Size
No of threads running is 1
Throughput for Read and Write :45.486497932431114 MB/sec
No of threads running is 2
Throughput for Read and Write :35.29965254335515 MB/sec
No of threads running is 4
Throughput for Read and Write :16.923403031713928 MB/sec
No of threads running is 8
Throughput for Read and Write :8.728010928973152 MB/sec
[Please enter the blocksize[8b, 80b, 8M , 80Mb and 0-quit] 8192

8192 is the Block Size
No of threads running is 1
Throughput for Read and Write :193.59412797853287 MB/sec
No of threads running is 2
Throughput for Read and Write :1054.411733345492 MB/sec
No of threads running is 4
Throughput for Read and Write :262.4263980126305 MB/sec
No of threads running is 8
Throughput for Read and Write :117.57765573492941 MB/sec
[Please enter the blocksize[8b, 80b, 8M , 80Mb and 0-quit] 81920

81920 is the Block Size
No of threads running is 1
Throughput for Read and Write :157.94600017455068 MB/sec
No of threads running is 2
Throughput for Read and Write :637.0114761866251 MB/sec
No of threads running is 4
Throughput for Read and Write :348.85840055660117 MB/sec
No of threads running is 8
Throughput for Read and Write :247.00810599409604 MB/sec
[Please enter the blocksize[8b, 80b, 8M , 80Mb and 0-quit] 0
[cc@pa1-mon ~]$
```

IOZONE:

```

I ozone: Performance Test of File I/O
Version $Revision: 3.471 $
Compiled for 64 bit mode.
Build: linux

Contributors: William Norcott, Don Capps, Isom Crawford, Kirby Collins
Al Slater, Scott Rhine, Mike Wisner, Ken Goss
Steve Landherr, Brad Smith, Mark Kelly, Dr. Alain CYR,
Randy Dunlap, Mark Montague, Dan Million, Gavin Brebner,
Jean-Marc Zucconi, Jeff Blomberg, Benny Halevy, Dave Boone,
Erik Habbinga, Kris Strecker, Walter Wong, Joshua Root,
Fabrice Bacchella, Zhenghua Xue, Qin Li, Darren Sawyer,
Vangel Bojaxhi, Ben England, Vikentsi Lapa,
Alexey Skidanov.

Run began: Tue Oct 10 02:44:19 2017

Auto Mode
File size set to 1024 kB
Command line used: ./iozone -a -s 1024
Output is in kBBytes/sec
Time Resolution = 0.000001 seconds.
Processor cache size set to 1024 kBytes.
Processor cache line size set to 32 bytes.
File stride size set to 17 * record size.

      kB  reclen   write  rewrite   read  reread   random   random    bkwd   record   stride
1024       4   846998  1741810  4162573  3606774  3671521  1824690  3200886  2301937  3200886  1903961  1885572  3835457  4317393
1024       8   1077836  2178189  4589593  6364746  5363307  2516377  4512441  3591693  3966517  2414523  2004366  5356618  4634161
1024      16   1398645  2275110  6441107  6689617  4972145  2723840  4995276  3304314  4920874  2386350  2381058  5390231  4356809
1024      32   1347734  2381058  5751117  4700088  5630486  2751762  4195100  3984917  5623115  2503178  2553783  3923040  5682634
1024      64   1166221  2753527  6402699  5991815  5720477  3210456  5277632  3301774  4700088  2672984  2694787  6059442  6317933
1024     128   1345623  2760606  5690162  5821271  5593820  3229770  5120336  3291652  5917516  2333199  2592318  5251818  5601115
1024     256   1352402  2270300  5417427  5536137  5529011  2716948  5018624  3411938  5303701  1759651  1965834  4995276  5336651
1024     512   1241389  2693897  4805258  5303701  6137371  3092561  5690162  3425544  5444898  1577984  4048778  5390231  5751117
1024    1024   1402756  2522288  5630486  6093831  5652717  3355952  5169641  3379719  4114719  3103735  2767722  5336651  5991815

iozone test complete.

```

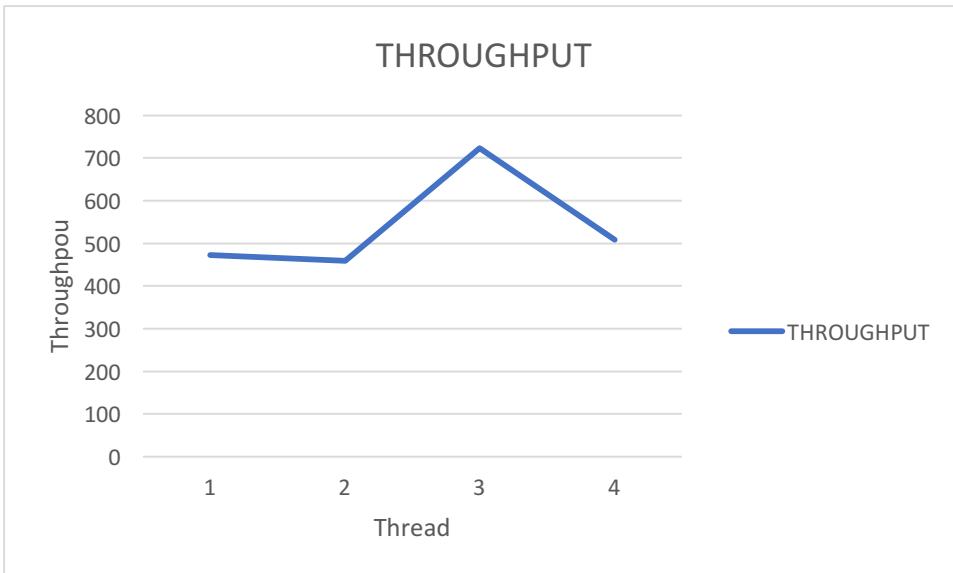
CONCLUSION:

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:

```
[cc@fin ~]$ ./client
connected to server
please enter the number of threads you wish to run in between[1,2,4,8] :
4

Throughput of client for uploading file with 4 threads is 458.480072
latency of client for uploading with 4 threads is 0.000002
```

```
[cc@fin ~]$ ls
```

```
[cc@fin ~]$ ./client
connected to server
please enter the number of threads you wish to run in between[1,2,4,8] :
8

Throughput of client for uploading file with 8 threads is 472.030243
latency of client for uploading with 8 threads is 0.000002

[cc@fin ~]$
```

```
[cc@fin ~]$ ./client
connected to server
please enter the number of threads you wish to run in between[1,2,4,8] :
1
```

```
Throughput of client for uploading file with 1 threads is 722.931030
```

```
[cc@fin ~]$ ./client
connected to server
please enter the number of threads you wish to run in between[1,2,4,8] :
2
```

```
Throughput of client for uploading file with 2 threads is 508.742737
latency of client for uploading with 2 threads is 0.000002
```

```
[cc@fin ~]$
```