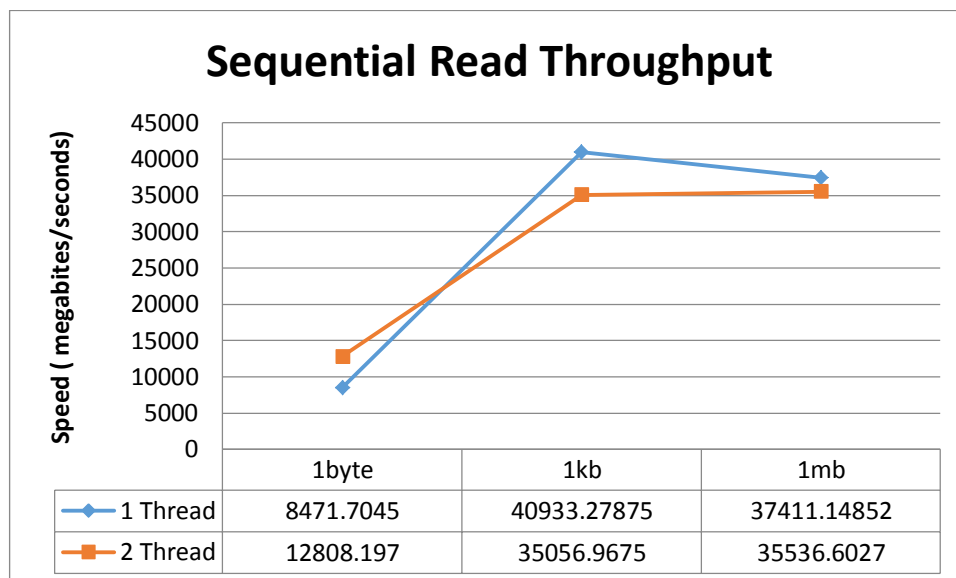
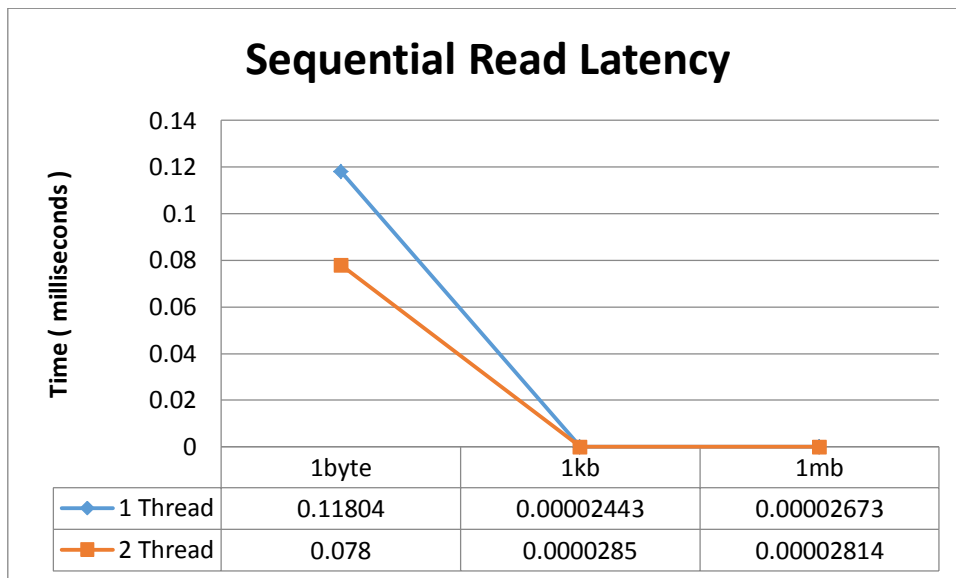


I. Disk Benchmarking:

Following parameters are used for the evaluation:

- 1.Total size of data : 100 MB
- 2.Amazon T2.Micro instance

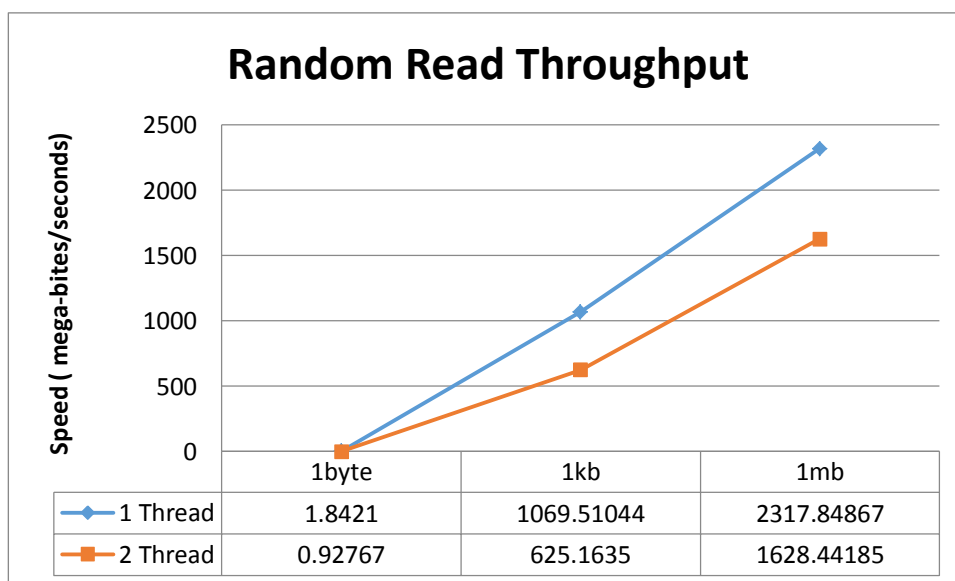
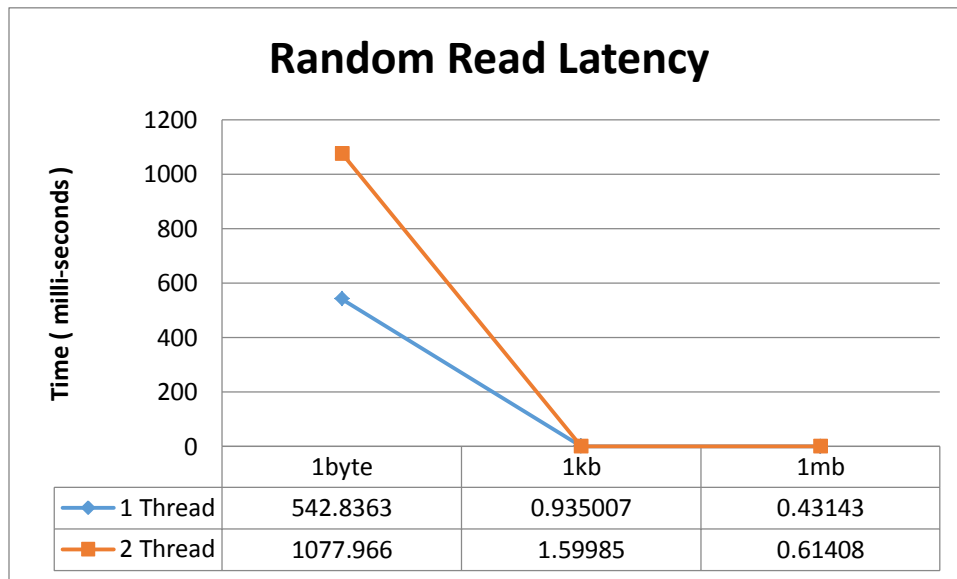
1. Sequential Read operation :



Analysis :

Above graph shows, there is increase in transfer rate of data as number of block-size increases from 1Byte to 1MByte.

2. Random Read Operation:

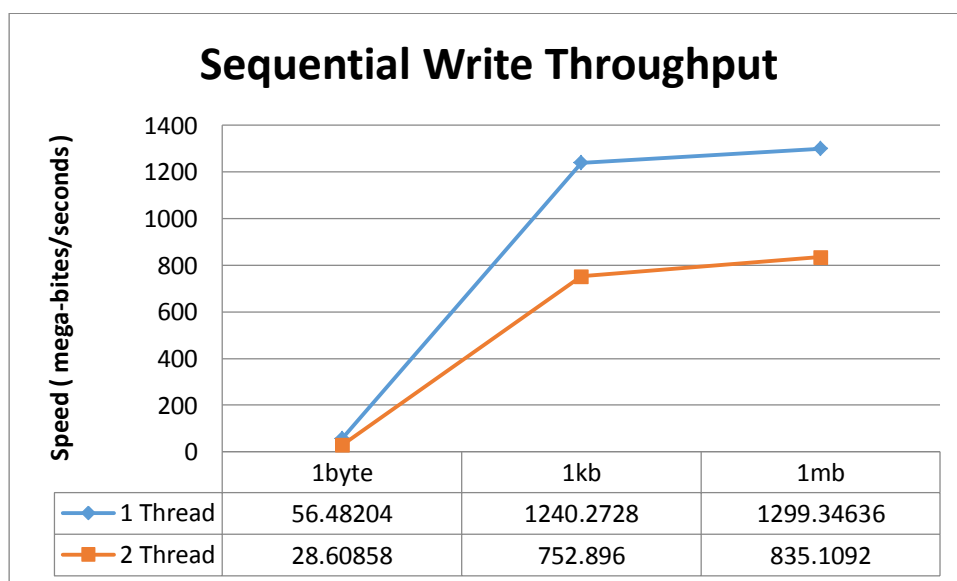
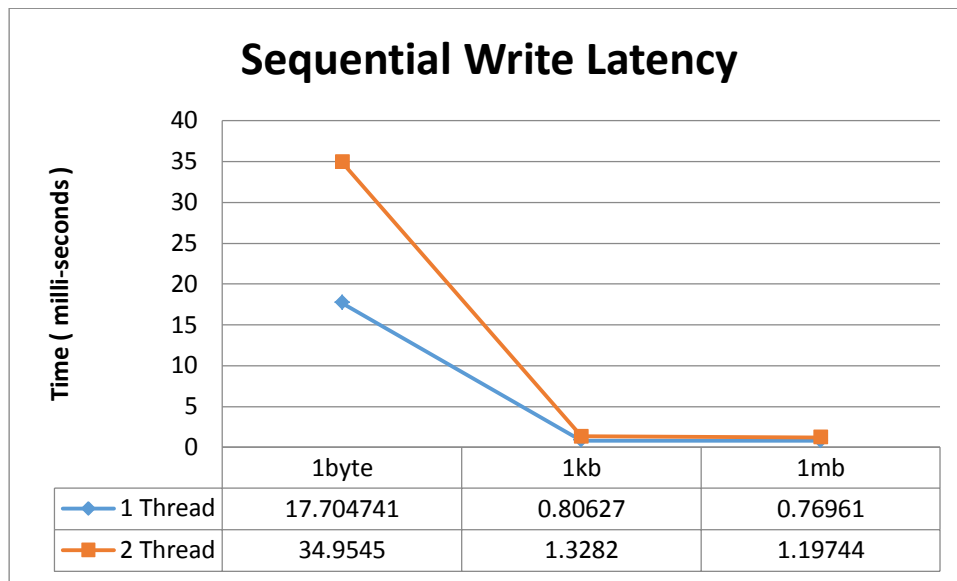


Analysis:

Random read operations gives very bad performance as compared sequential read operation. Following table compares throughput of sequential and random read operation for 1-Thread.

Block Size	Sequential Read	Random Read
1 Byte	8471.7	1.84
1 KByte	40933.27	1069.51
1 Mbyte	37411.14	2317.85

3. sequential Write Operation:



Analysis:

There is drastic increase in transfer rate when block size of data increases from 1Byte to 1Kbyte. There is slightly increase in performance when block size increases from 1Kbyte to 1MByte.

Theoretical Performance of a disk:

Amazon has given t2.micro instance theoretical value for disk is given as 160Mb/S.

IOZONE Benchmarking system (Extra Credit)

- Compared evaluation of self implemented benchmarking with IOZONE benchmarking.
- Both systems (mine & IOZONE) are evaluated on amazon's t2.micro instance.
- lmerf has been evaluated for packet size 100MBytes.

```

Auto Mode
Command line used: ./iozone -a
Output is in Kbytes/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  read  random  random  bkwd  record  stride
64   4 1330167 3363612 1183548 9006179 8182586 2923952 4897948 4564786 10821524 3203069 3541098 6421025 7100397
64   8 1232453 3057153 15972885 15972885 7100397 3022727 5735102 5389653 12902017 4207076 4988978 7940539 12902017
64  16 1255511 4274062 12902017 15972885 10821524 3958892 4564786 4897948 12902017 2892445 4564786 5389653 12902017
64  32 1599680 4274062 15972885 15972885 12902017 4897948 6421025 5283570 12310336 3958892 5860307 7940539 15972885
64  64 1562436 3588436 12902017 30484297 15972885 5735102 7940539 4564786 12902017 5283570 4564786 7940539 12902017
128  4 1406136 3560017 8548124 10779307 9129573 3759450 6406138 4934216 9129573 4012317 3982553 3560017 8036304
128  8 1618028 4759253 14200794 15881078 12842051 4104338 8036304 6114306 12842051 2905056 4407601 9129573 14200794
128 16 1642783 4759253 14200794 16365173 12842051 4135958 8036304 5847904 14200794 4557257 4717434 9795896 9795896
128 32 1802755 4717434 14200794 11720614 11470204 4407601 8036304 5603747 9795896 4407601 5545860 7082197 10779307
128 64 1663139 4759253 15881078 18012359 14200794 5545860 7917784 5545860 14200794 3867787 7082197 6406138 10567140
128 128 1705404 4717434 14200794 15881078 14200794 5122535 6727225 4135958 7082197 5122535 5545860 6406138 12842051
256  4 1479379 3245838 11207494 11091721 9114515 3410808 7735574 4819184 3756894 3187509 3823789 8534922 7120034
256  8 1740810 4672389 12090911 14164395 12228612 4404088 9434868 6398720 10758322 4496299 3756894 11569783 14164395
256 16 2152625 3879045 11091721 14164395 12228612 4929815 9868433 5569035 17096249 4264168 2325094 7518900 14353744
256 32 1985448 5217259 8815202 16072608 13454450 5117791 8534922 5347168 10651598 3935921 5320671 7735574 12228612
256 64 2065659 5320671 14953435 15835569 15164624 5347168 10245071 5841722 11091721 4840911 5810112 12228612 14164395
256 128 1911233 5455847 4496299 15164624 13454450 5347168 10651598 4907284 11091721 2366082 8271916 7571923 14164395
256 256 1600674 5242734 9518507 12812277 12228612 3935921 8208677 4907284 10651598 5022044 4496299 6719046 12228612
512  4 1689859 4163357 8528336 11620224 9304292 3710197 8528336 5227492 6931711 3409755 3993047 7871843 9638369
512  8 1940253 4701087 8665997 13872122 12499490 4237281 9814569 6652558 11138071 4660280 4784885 8844453 13522711
512 16 2350044 4660280 10235583 14240069 13109944 4650188 4871723 6652558 10641343 3913012 4871723 12427157 13438092
512 32 1999881 5177083 13190469 14146265 13438092 5624545 11620224 6472112 13109944 3905895 2285029 5278892 6551087
512 64 2297251 5509113 14146265 15037801 13872122 5684095 12146009 6736026 14628067 5019764 5439343 13438092 15037801
512 128 2256220 4099771 9107004 10641343 13522711 4784885 11138071 6319739 11620224 4494470 4732165 10856530 12797441
512 256 2225819 4237291 10641343 10641343 12215097 4742616 10434519 5624545 10641343 3849877 7309196 10856530 11943357
512 512 1163439 5227492 11620224 12797441 9145790 5886650 11138071 5760329 7513788 4973263 5684095 11620224 12797441
1024  4 1615374 3037876 10134284 11645609 8914314 4146499 6787181 5303701 8326715 3791442 4029784 9765601 10454984
1024  8 2056183 4674510 13102174 10454984 11614118 3631168 10906310 6963241 11773301 4810640 3984917 7420395 13998981
1024 16 2115949 5175871 11249091 14230902 12208350 5536137 11018226 7160597 8199542 4110780 4321737 10134284 13998981
1024 32 2295784 4142499 8610501 12983353 12944224 5720477 9485232 6787181 8822754 3999762 5144870 12639479 14422045
1024 64 2124322 5690162 10662628 14618393 13264026 5277632 8988941 6787181 14230902 4287225 5065980 7577494 14668318
1024 128 1874871 4531485 10039528 13643232 12639479 6093831 11614118 7020149 15080341 5593820 4270175 7883484 12071103
1024 256 2011877 4178773 8262639 10557785 11614118 4832290 6280976 5991815 11903823 4876180 4805258 8895850 12313351
1024 512 2216407 3458646 11018226 12639479 12348754 5169641 7160597 5303701 11278631 3804877 5593820 9220512 11903823
1024 1024 1896395 5444898 13142265 12790037 12639479 5720477 12037272 5169641 9142007 5623115 4396951 8262639 11489839

```

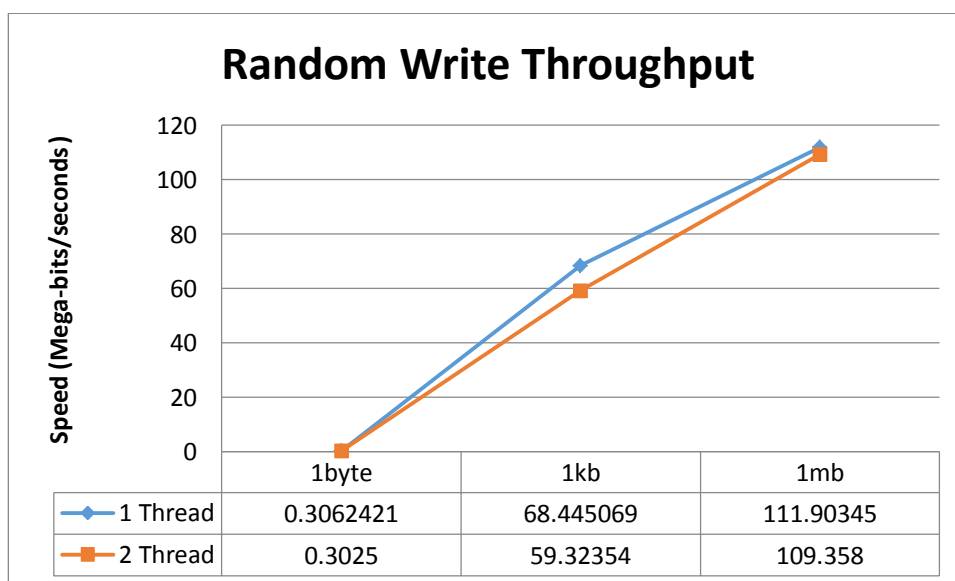
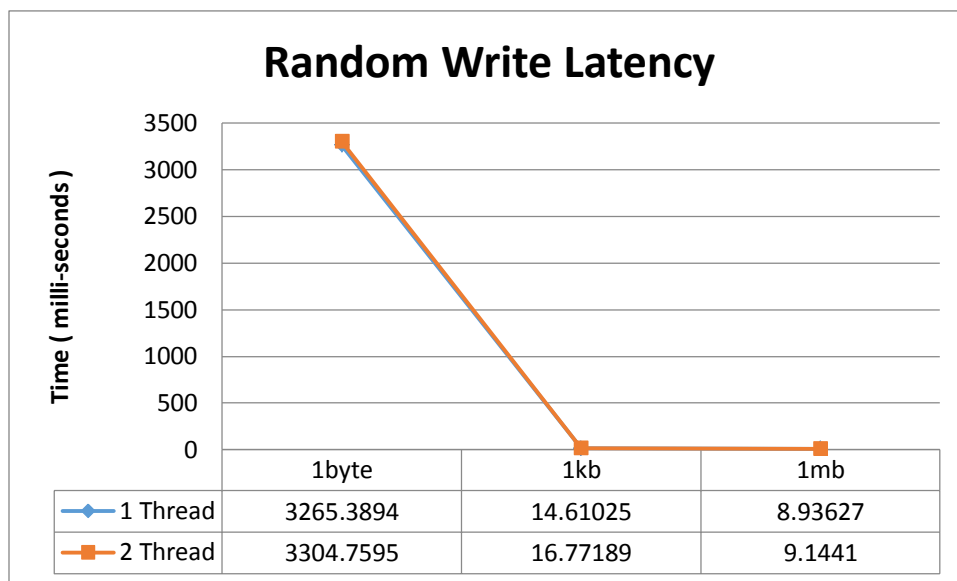
Following are the results for 1024KB :

Operations	Throughput in MB/S
Sequential Read	9402.175
Sequential Write	2403.712
Random Read	9142.007
Random Write	3908.759

- Comparing this values with my system values.

Operations	My System Throughput	IOZONE Throughput
Sequential Read	40933.27	10134284
Sequential Write	1240.2728	1615374
Random Read	1069.51	8914314
Random Write	68.44	4146499

4. Random Write Operation:



Analysis:

There is huge drop in performance when compared to sequential write operations. Following table compares throughput of sequential and random write operation for 1-Thread.

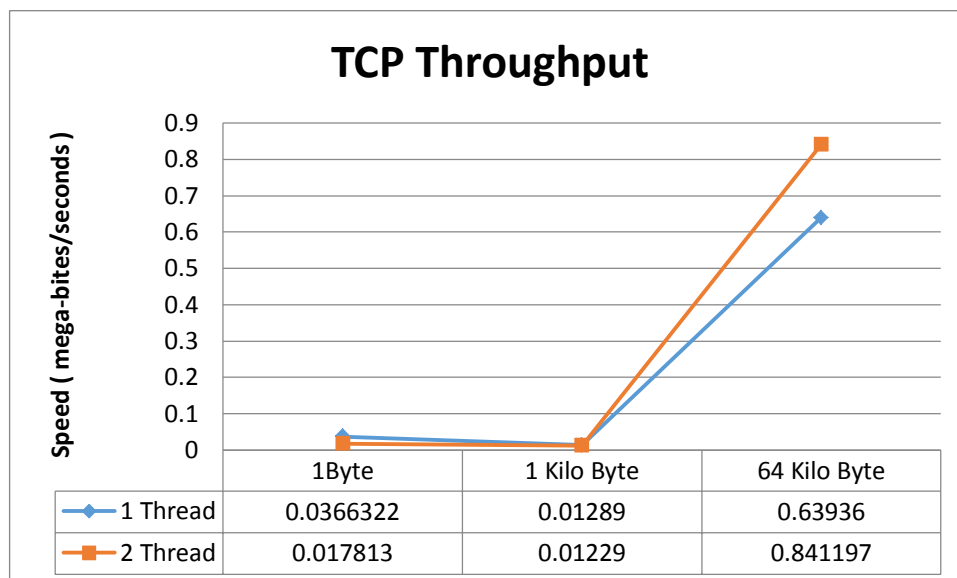
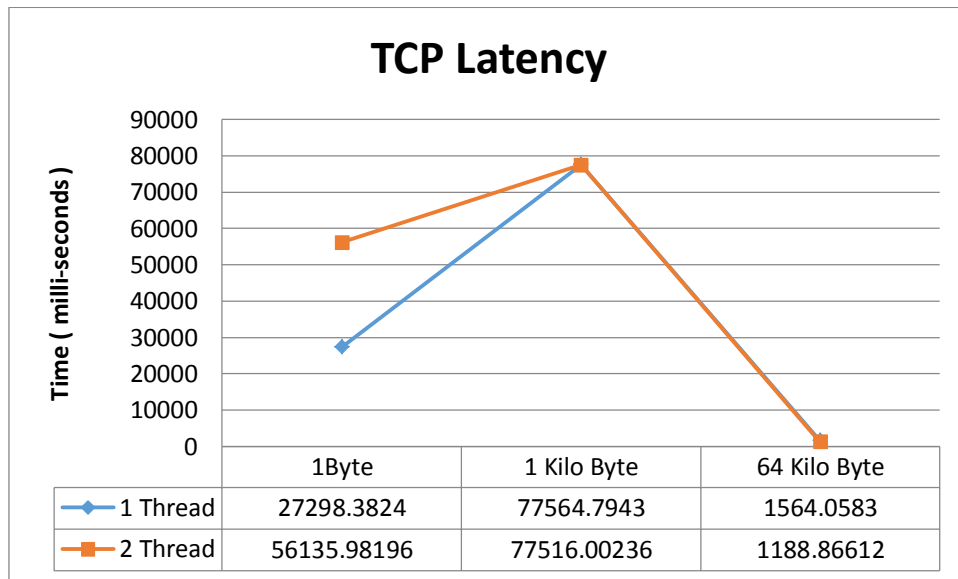
Block Size	Sequential Write	Random Write
1 Byte	56.48	0.30
1 KByte	1240.27	68.44
1 Mbyte	1299.34	111.90

III. Network Benchmarking

Following parameters are used for the evaluation:

1. Total size of data : 64000 KBytes
2. Amazon T2.Micro instance

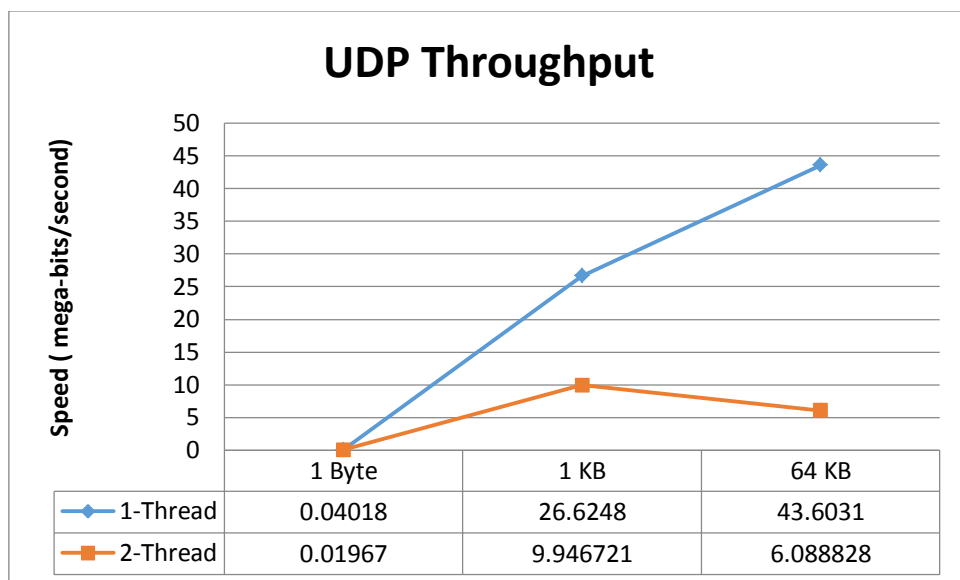
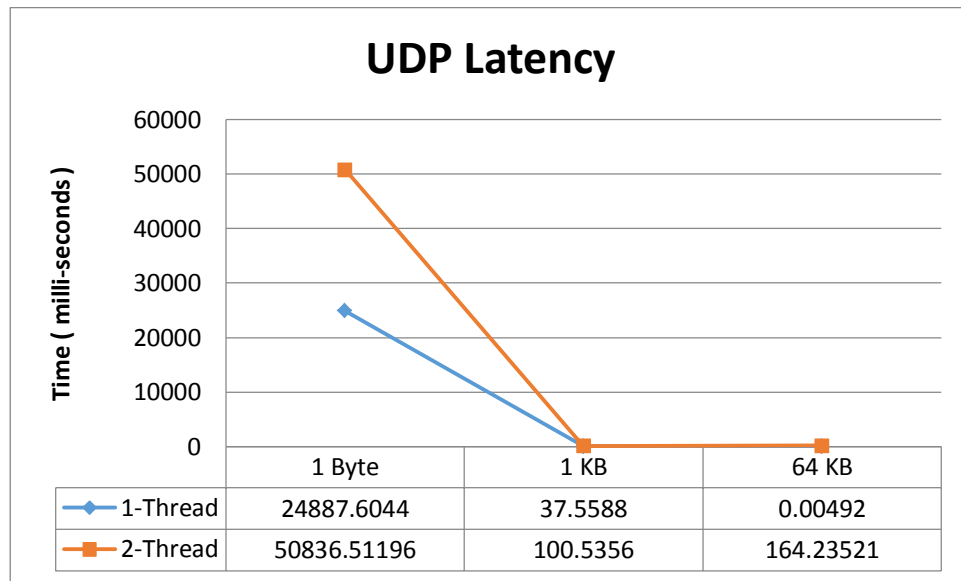
1. TCP Operation:



Analysis:

Throughput of TCP increases drastically when size of packet is increases from 1KByte to 64KByte There is slightly increase in performance when uses multithreading .

2. UDP Operation:



Analysis:

There is increase in performance when the data packets send over network using UDP protocols. Following table compares throughput of UDP and TCP performed on 1-Thread.

Block Size	Sequential Write	Random Write
1 Byte	0.036	0.04
1 KByte	0.012	26.62
1 Mbyte	0.639	43.60

Theoretical Performance of a Network:

- Amazon has given t2.micro instance theoretical value for Network's Bandwidth is given as 2.7MB/S.

Extra Credit for Network Benchmarking:

- Iperf Benchmarking system**
 - Compared evaluation of self implemented benchmarking with Iperf benchmarking.
 - Both systems (mine & Iperf) are evaluated on amazon's t2.micro instance.
 - Iperf has been evaluated for packet size 64KB.

Iperf TCP :

```
ubuntu@ip-172-31-35-59: ~
-----
Accepted connection from 52.36.110.95, port 60646
[ S] local 172.31.35.59 port 5201 connected to 52.36.110.95 port 60647
[ ID] Interval      Transfer    Bandwidth
[ 5] 0.00-1.00 sec  39.6 MBytes 332 Mbits/sec
[ 5] 1.00-2.00 sec  41.8 MBytes 351 Mbits/sec
[ 5] 2.00-3.00 sec  41.3 MBytes 346 Mbits/sec
[ 5] 3.00-4.00 sec  40.2 MBytes 337 Mbits/sec
[ 5] 4.00-5.00 sec  37.2 MBytes 312 Mbits/sec
[ 5] 5.00-6.00 sec  41.1 MBytes 345 Mbits/sec
[ 5] 6.00-7.00 sec  42.1 MBytes 353 Mbits/sec
[ 5] 7.00-8.00 sec  41.6 MBytes 349 Mbits/sec
[ 5] 8.00-9.00 sec  39.9 MBytes 335 Mbits/sec
[ 5] 9.00-10.00 sec 38.4 MBytes 322 Mbits/sec
[ 5] 10.00-10.04 sec 1.48 MBytes 332 Mbits/sec
-----
[ ID] Interval      Transfer    Bandwidth  Retr
[ 5] 0.00-10.04 sec 405 MBytes 338 Mbits/sec 21      sender
[ 5] 0.00-10.04 sec 405 MBytes 338 Mbits/sec                receiver
Server listening on 5201
-----
Accepted connection from 52.36.110.95, port 60648
[ S] local 172.31.35.59 port 5201 connected to 52.36.110.95 port 60649
[ ID] Interval      Transfer    Bandwidth
[ 5] 0.00-1.00 sec  39.3 MBytes 330 Mbits/sec
[ 5] 1.00-2.00 sec  37.6 MBytes 316 Mbits/sec
[ 5] 2.00-3.00 sec  32.2 MBytes 270 Mbits/sec
[ 5] 3.00-4.00 sec  35.2 MBytes 295 Mbits/sec
[ 5] 4.00-5.00 sec  38.3 MBytes 321 Mbits/sec
[ 5] 5.00-6.00 sec  31.1 MBytes 261 Mbits/sec
[ 5] 6.00-7.00 sec  32.5 MBytes 273 Mbits/sec
[ 5] 7.00-8.00 sec  35.0 MBytes 294 Mbits/sec
[ 5] 8.00-9.00 sec  39.9 MBytes 335 Mbits/sec
[ 5] 9.00-10.00 sec 39.4 MBytes 330 Mbits/sec
[ 5] 10.00-10.04 sec 1.51 MBytes 338 Mbits/sec
-----
[ ID] Interval      Transfer    Bandwidth  Retr
[ 5] 0.00-10.04 sec 362 MBytes 303 Mbits/sec 0      sender
[ 5] 0.00-10.04 sec 362 MBytes 303 Mbits/sec                receiver
Server listening on 5201

ubuntu@ip-172-31-28-228: ~
[ 4] 1.00-2.00 sec 41.9 MBytes 351 Mbits/sec 0 83.4 KBytes
[ 4] 2.00-3.00 sec 41.2 MBytes 346 Mbits/sec 0 83.4 KBytes
[ 4] 3.00-4.00 sec 40.1 MBytes 336 Mbits/sec 0 83.4 KBytes
[ 4] 4.00-5.00 sec 37.2 MBytes 312 Mbits/sec 0 83.4 KBytes
[ 4] 5.00-6.00 sec 41.2 MBytes 346 Mbits/sec 0 83.4 KBytes
[ 4] 6.00-7.00 sec 42.1 MBytes 353 Mbits/sec 0 83.4 KBytes
[ 4] 7.00-8.00 sec 41.7 MBytes 350 Mbits/sec 0 83.4 KBytes
[ 4] 8.00-9.00 sec 39.8 MBytes 334 Mbits/sec 0 83.4 KBytes
[ 4] 9.00-10.00 sec 38.4 MBytes 322 Mbits/sec 0 83.4 KBytes
-----
[ ID] Interval      Transfer    Bandwidth  Retr
[ 4] 0.00-10.00 sec 405 MBytes 340 Mbits/sec 21      sender
[ 4] 0.00-10.00 sec 405 MBytes 340 Mbits/sec                receiver
iperf Done.
ubuntu@ip-172-31-28-228:~$ iperf3 -c 52.25.40.18 -f m -w 64k
Connecting to host 52.25.40.18, port 5201
[ 4] local 172.31.28.228 port 60649 connected to 52.25.40.18 port 5201
[ ID] Interval      Transfer    Bandwidth  Retr Cwnd
[ 4] 0.00-1.00 sec 41.0 MBytes 344 Mbits/sec 0 83.4 KBytes
[ 4] 1.00-2.00 sec 37.5 MBytes 315 Mbits/sec 0 83.4 KBytes
[ 4] 2.00-3.00 sec 31.8 MBytes 267 Mbits/sec 0 83.4 KBytes
[ 4] 3.00-4.00 sec 35.5 MBytes 298 Mbits/sec 0 83.4 KBytes
[ 4] 4.00-5.00 sec 38.0 MBytes 319 Mbits/sec 0 83.4 KBytes
[ 4] 5.00-6.00 sec 31.0 MBytes 260 Mbits/sec 0 83.4 KBytes
[ 4] 6.00-7.00 sec 32.8 MBytes 275 Mbits/sec 0 83.4 KBytes
[ 4] 7.00-8.00 sec 35.0 MBytes 294 Mbits/sec 0 83.4 KBytes
[ 4] 8.00-9.00 sec 40.1 MBytes 336 Mbits/sec 0 83.4 KBytes
[ 4] 9.00-10.00 sec 39.4 MBytes 330 Mbits/sec 0 83.4 KBytes
-----
[ ID] Interval      Transfer    Bandwidth  Retr
[ 4] 0.00-10.00 sec 362 MBytes 304 Mbits/sec 0      sender
[ 4] 0.00-10.00 sec 362 MBytes 304 Mbits/sec                receiver
iperf Done.
```

Iperf UDP:

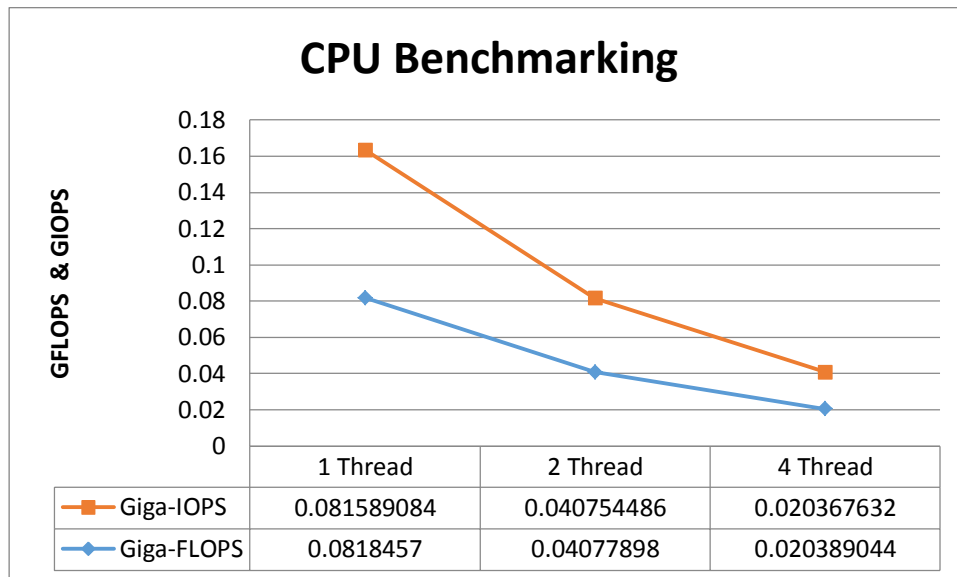
```
ubuntu@ip-172-31-35-59: ~
[ S] 0.00-10.04 sec 685 MBytes 572 Mbits/sec                receiver
-----
Server listening on 5201
-----
Accepted connection from 52.36.110.95, port 60654
[ S] local 172.31.35.59 port 5201 connected to 52.36.110.95 port 54828
[ ID] Interval      Transfer    Bandwidth  Jitter  Lost/Total Datagrams
[ 5] 0.00-1.00 sec  120 KBytes 983 Kbits/sec 23.753 ms 0/15 (0%)
[ 5] 1.00-2.00 sec  128 KBytes 1.05 Mbits/sec 8.509 ms 0/16 (0%)
[ 5] 2.00-3.00 sec  128 KBytes 1.05 Mbits/sec 3.076 ms 0/16 (0%)
[ 5] 3.00-4.00 sec  128 KBytes 1.05 Mbits/sec 1.139 ms 0/16 (0%)
[ 5] 4.00-5.00 sec  128 KBytes 1.05 Mbits/sec 0.480 ms 0/16 (0%)
[ 5] 5.00-6.00 sec  128 KBytes 1.05 Mbits/sec 0.232 ms 0/16 (0%)
[ 5] 6.00-7.00 sec  128 KBytes 1.05 Mbits/sec 0.116 ms 0/16 (0%)
[ 5] 7.00-8.00 sec  128 KBytes 1.05 Mbits/sec 0.084 ms 0/16 (0%)
[ 5] 8.00-9.00 sec  128 KBytes 1.05 Mbits/sec 0.054 ms 0/16 (0%)
[ 5] 9.00-10.00 sec 128 KBytes 1.05 Mbits/sec 0.072 ms 0/16 (0%)
[ 5] 10.00-10.04 sec 0.00 Bytes 0.00 bits/sec 0.072 ms 0/0 (-nan%)
-----
[ ID] Interval      Transfer    Bandwidth  Jitter  Lost/Total Datagrams
[ 5] 0.00-10.04 sec 1.24 MBytes 1.04 Mbits/sec 0.072 ms 0/159 (0%)
Server listening on 5201

ubuntu@ip-172-31-28-228: ~
[ 4] 0.00-10.00 sec 685 MBytes 575 Mbits/sec 1      sender
[ 4] 0.00-10.00 sec 685 MBytes 575 Mbits/sec                receiver
iperf Done.
ubuntu@ip-172-31-28-228:~$ iperf3 -c 52.25.40.18 -u -f m -w 64k
Connecting to host 52.25.40.18, port 5201
[ 4] local 172.31.28.228 port 54828 connected to 52.25.40.18 port 5201
[ ID] Interval      Transfer    Bandwidth  Total Datagrams
[ 4] 0.00-1.00 sec  120 KBytes 0.98 Mbits/sec 15
[ 4] 1.00-2.00 sec  128 KBytes 1.05 Mbits/sec 16
[ 4] 2.00-3.00 sec  128 KBytes 1.05 Mbits/sec 16
[ 4] 3.00-4.00 sec  128 KBytes 1.05 Mbits/sec 16
[ 4] 4.00-5.00 sec  128 KBytes 1.05 Mbits/sec 16
[ 4] 5.00-6.00 sec  128 KBytes 1.05 Mbits/sec 16
[ 4] 6.00-7.00 sec  128 KBytes 1.05 Mbits/sec 16
[ 4] 7.00-8.00 sec  128 KBytes 1.05 Mbits/sec 16
[ 4] 8.00-9.00 sec  128 KBytes 1.05 Mbits/sec 16
[ 4] 9.00-10.00 sec 128 KBytes 1.05 Mbits/sec 16
-----
[ ID] Interval      Transfer    Bandwidth  Jitter  Lost/Total Datagrams
[ 4] 0.00-10.00 sec 1.24 MBytes 1.04 Mbits/sec 0.072 ms 0/159 (0%)
[ 4] Sent 159 datagrams
iperf Done.
```


Comparing this values with my system values, efficiency for my system is calculated and found to be high as compared to the Iperf performance.

- Instruction to TCP Iperf:
 1. Install Iperf benchmark in local :
`Sudo apt-get install iperf3`
 2. Run TCP server
`Iperf3 -s`
 3. Run TCP client :
`Iperf3 -c @"public_ip"-f m -w 64k`
- Instruction to UDP Iperf:
 1. Install Iperf benchmark in local :
`Sudo apt-get install iperf3`
 2. Run TCP server
`Iperf3 -s`
 3. Run TCP client :
`Iperf3 -c @"public_ip"-f m -w 64k`
- For UDP:
 1. For running a Server :
`Iperf3 -s`
 2. Run UDP client:
`Iperf3 -c @"public_ip" -u -f m -w 64k .`

III. CPU Benchmarking:



X-axis : GFLOPS and GIOPS

Y-axis : Number of threads

- Above Table shows CPU speed in terms of GIGA FLOPS and GIGA IOPS i.e. Giga Floating Point Operation per Second and Giga Integer Operations per second respectively.
- This benchmark is performed in three different threads (1,2 & 4).
- As more threads put more overhead of thread maintenance, concurrency and switching we can conclude that, the optimal number of concurrency for best performance is achieved using 4 threads.
- **Theoretical Peak Performance** = number of cores* clock cycle * FLOPs/cycle
= $2 * 1.7 * 4$
= 13.6 GFLOPS
- **Efficiency** = (FLOPS for 1 thread /Theoretical Peak Performance)*100
= $(0.081589/13.6) * 100$
= 0.5 %