CS 553 Cloud Computing

Programming Assignment 1

RONAKKUMAR MAKADIYA (CWID: A20332994)
KAUSTUBH BOJEWAR (CWID: A20329244)
SOURABH CHOUGALE (CWID: A20326997)

Evaluation

CPU BENCHMARKING:

System Configuration:

- Operating System: UBUNTU (VMware)
- RAM: 1 GB
- No of Physical cores: 2
- No of threads: 4

Number of Threads	FLOPS (Giga)	IOPS(Giga)
1	0.88356	7.66497
2	0.45146	3.28748
4	0.22073	2.81893
8	0.10948	1.26383

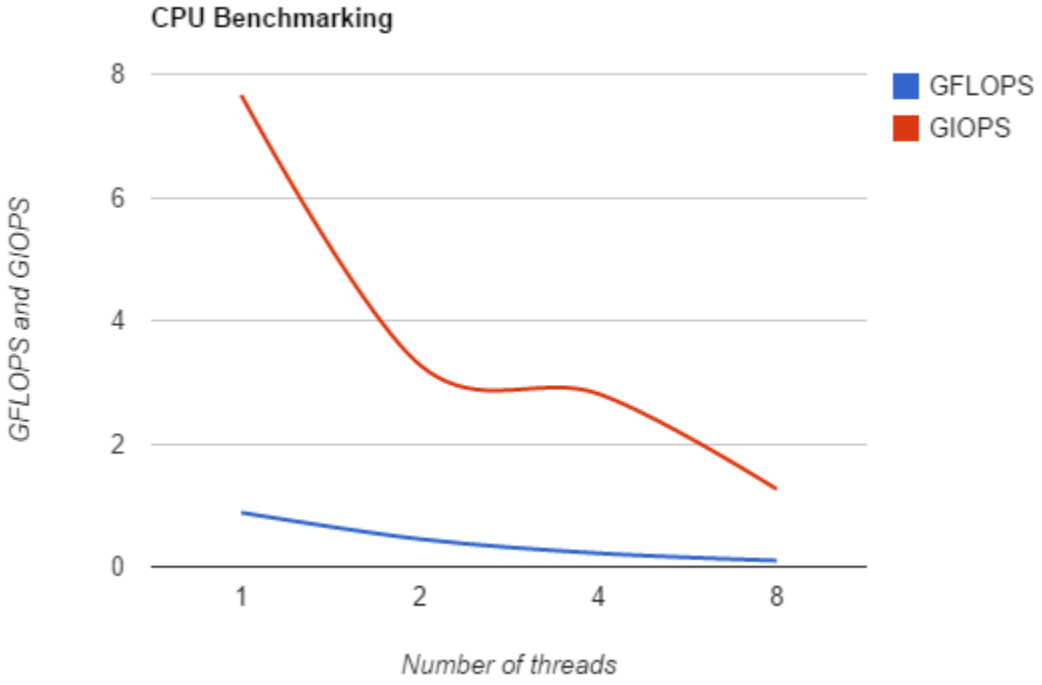
Table: CPU speed in terms of Giga Floating Point operation per sec (GFLOPS) and Giga Integer Operations per sec (GIOPS) using threads (1, 2, 4 and 8)

- The optimal number of concurrency for best performance is 1, since more threads brings in more overhead of thread maintenance, concurrency, switching etc.

GRAPH for CPU Benchmarking:

- **X-axis:** Number of Threads
- **Y-axis**: GFLOPS & GIOPS

This graph plots the GFLOPS and GIOPS versus the number of threads. I have tried with various number of threads (1, 2, 4 and 8) and it's been observed that the GFLOPS and GIOPS are highest when run on a single thread.



- 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

  = (0.88/13.6)* 100

  = 6.47 %

- **Extra Credit**: Average and Standard Deviation of all evaluations (Experiments were run 3 times to calculate)

Number of Threads	GFLOPS	GIOPS	Average (GFLOPS)	Average (GIOPS)
1	1. 0.8612 2. 0.8835 3. 0.8723	1. 7.6649 2. 7.5332 3. 7.7231	0.8724	7.6404
2	1. 0.4514 2. 0.4632 3. 0.4498	1. 3.2874 2. 3.2712 3. 3.4221	0.4548	3.3269
4	1. 0.2207 2. 0.2134 3. 0.2398	1. 2.8189 2. 2.7612 3. 2.9563	0.2246	2.8454
8	1. 0.1094 2. 0.1102 3. 0.1045	1. 1.2638 2. 1.3214 3. 1.1965	0.1080	1.2605

Table: Readings for 3 experiments and average is calculated.

Formula for Standard Deviation :

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

Standard Deviation for 1 Thread (GFLOPS) - 0.011
Standard Deviation for 2 Thread (GFLOPS) - 0.0073
Standard Deviation for 4 Thread (GFLOPS) - 0.013
Standard Deviation for 8 Thread (GFLOPS) – 0.003

Standard Deviation for 1 Thread (GIOPS) – 0.097
Standard Deviation for 2 Thread (GIOPS) – 0.082
Standard Deviation for 4 Thread (GIOPS) – 0.10
Standard Deviation for 8 Thread (GIOPS) – 0.06

GPU Benchmarking

This benchmarking is implemented in Jarvis.

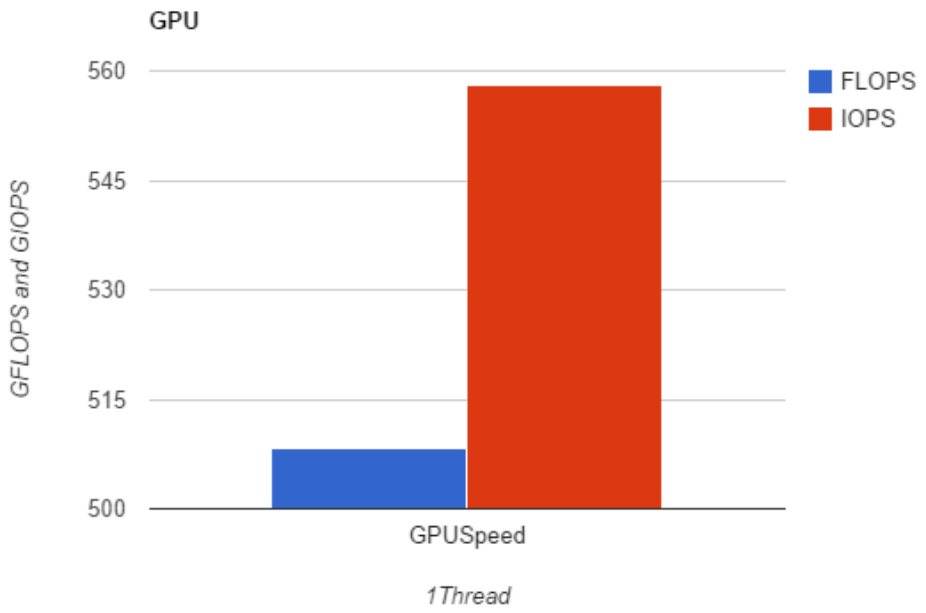
Number of Threads	FLOPS (Giga)	IOPS(Giga)
1	508.12	558.38

Extra Credit:

Number of Threads	GFLOPS	GIOPS	Average (GFLOPS)	Average (GIOPS)
1	1. 505 2. 508 3. 510	1. 558 2. 562 3. 570	507.6	563.33

Standard Deviation for GFLOPS – 2.51

Standard Deviation for GIOPS - 6.11



MEMORY BENCHMARKING:

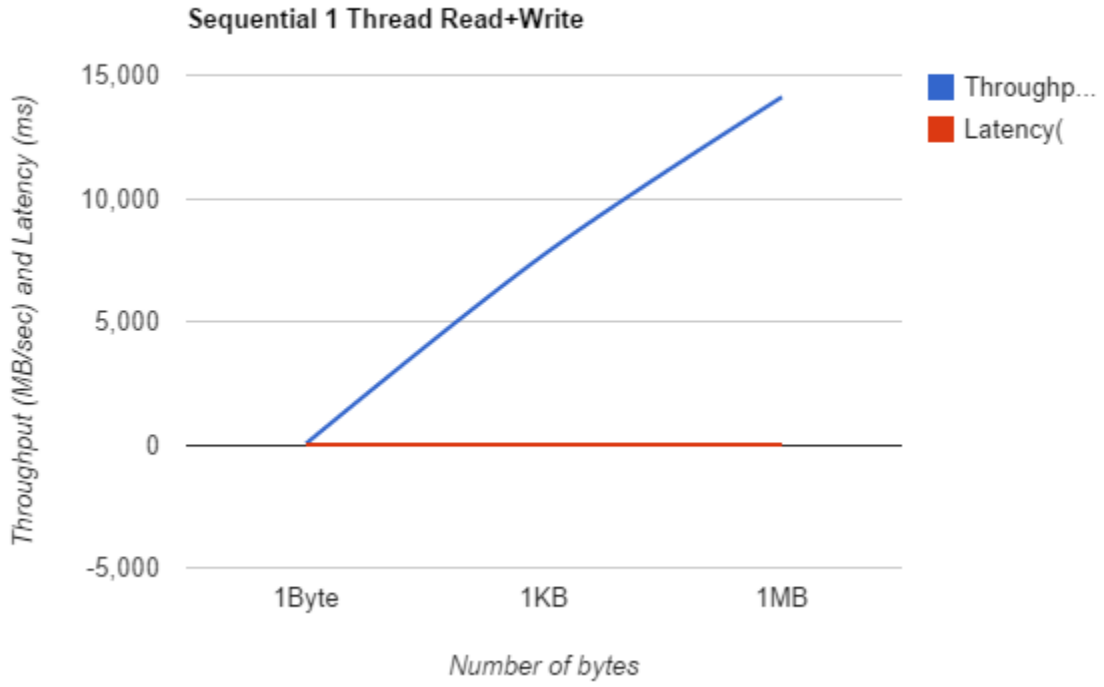
System Configuration:

- Operating System: UBUNTU (VMware)
- RAM: 1 GB
- No of Physical cores: 2
- No of threads: 4

Concurrency Level: 1 Thread

Access Type: Sequential (Read+Write)

Block Size	Throughput(MB/sec)	Latency(msec)
1 Byte	39.062	0.000024
1 Kilobyte	7716	0.0000070
1 Megabyte	14113.82	0.0000004

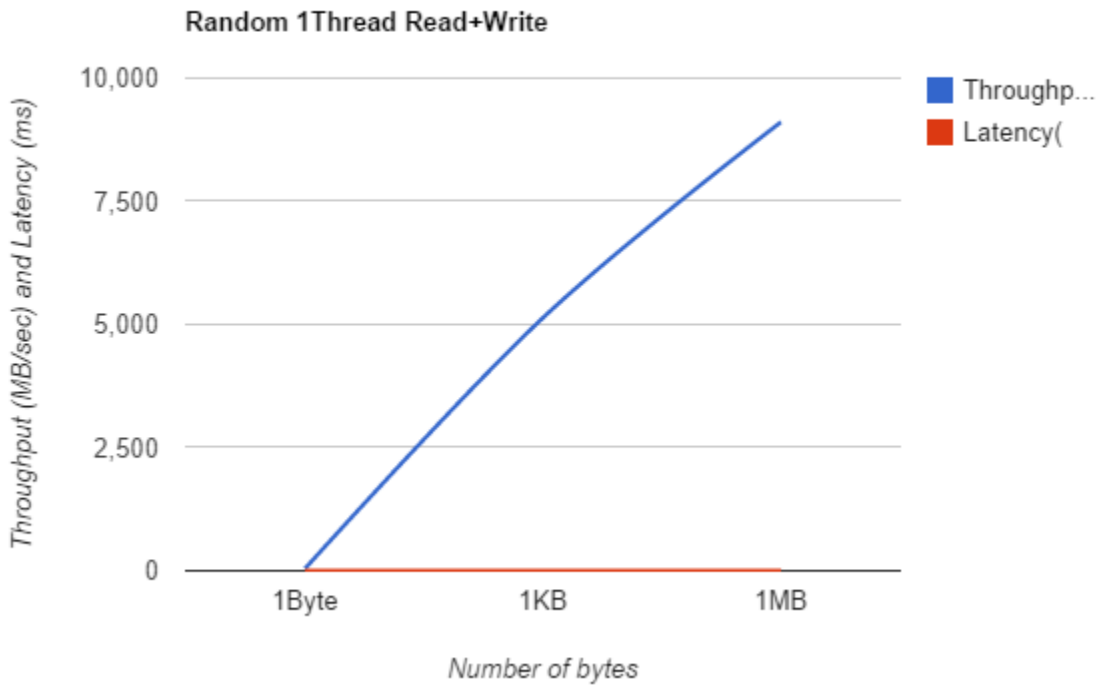


Concurrency Level: 1 Thread

Access Type: Random

Block Size	Throughput(MB/sec)	Latency(ms)
1 Byte	42.97	0.000022
1 Kilobyte	5136	0.0000077
1 Megabyte	9109.74	0.000000041

- **X-axis:** Number of Bytes
- **Y-axis**: Throughput (MB/sec) and Latency (ms)

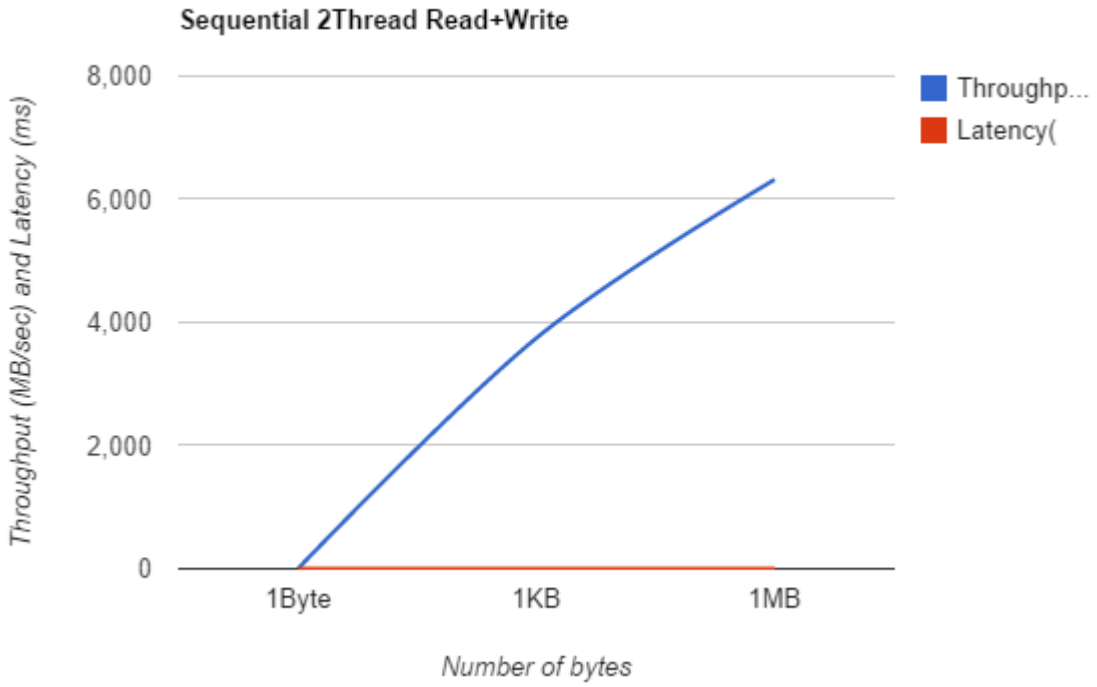


Concurrency Level: 2 Thread

Access Type: Sequential

Block Size	Throughput(MB/sec)	Latency(msec)
1 Byte	7.2865	0.000262
1 Kilobyte	3748.63	0.0000056
1 Megabyte	6321.08	0.000000014

- **X-axis:** Number of Bytes
- **Y-axis**: Throughput (MB/sec) and Latency (ms)

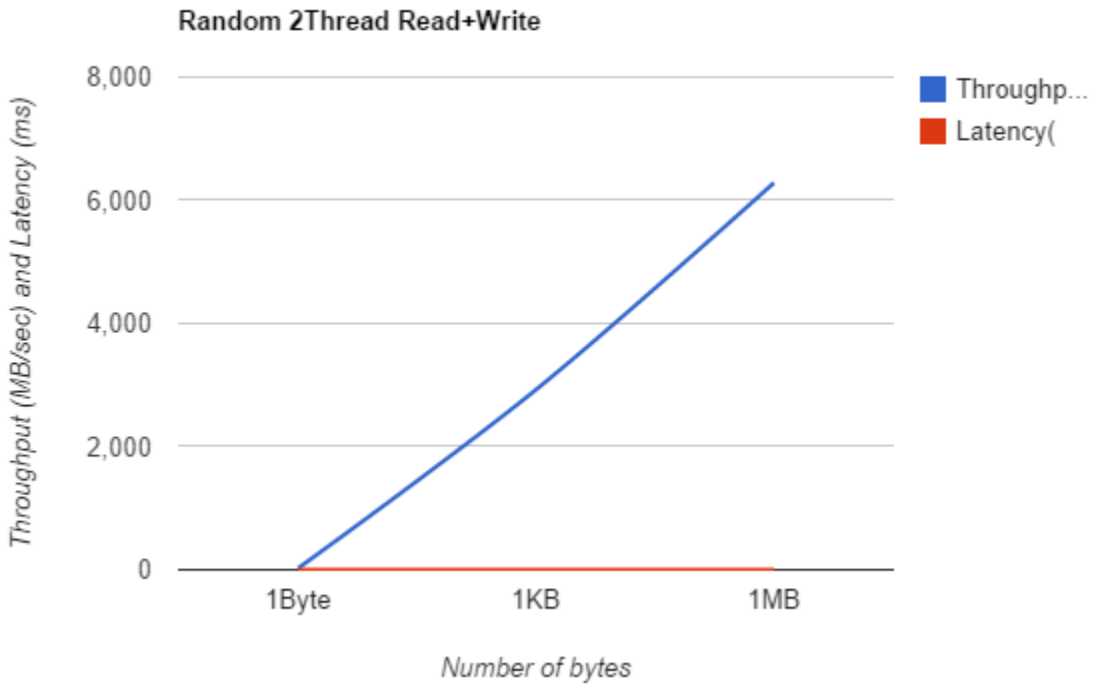


Concurrency Level: 2 Thread

Access Type: Random

Block Size	Throughput(MB/sec)	Latency(msec)
1 Byte	23.2495	0.00084
1 Kilobyte	2924.66	0.0000066
1 Megabyte	6285.12	0.00000093

- **X-axis:** Number of Bytes
- **Y-axis**: Throughput (MB/sec) and Latency (ms)



- **Extra Credit**: Average and Standard Deviation of all evaluations (Experiments were run 3 times to calculate)
  For 1 thread:

Access Type	Block Size	Throughput	Latency	Average Throughput	Average Latency
Sequential	1 byte	39.062 39.027 39.089	0.000024 0.000029 0.000021	39.057	2.47e-5
	1 Kilobyte	7716 7715.82 7715.56	0.000007 0.0000072 0.0000078	7715.79	7.33e-6
	1 Megabyte	14113.82 14113.02 14112.91	0.0000004 0.00000044 0.0000005	14113.25	4.47e-7
Random	1 byte	42.97 42.92 42.98	0.000022 0.000017 0.000019	42.96	0.00002
	1 Kilobyte	5136 5133 5131	0.0000077 0.0000069 0.0000022	5133.33	0.0000056
	1 Megabyte	9109.74 9109.68 9109.01	0.000000041 0.000000021 0.000000037	9109.48	0.000000033

Formula for Standard Deviation :

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

Standard Deviation for 1 Thread sequential read/write 1 byte –
Throughput: 0.03109
Latency: 0
Standard Deviation for 1 Thread sequential read/write 1Kilobyte-
Throughput: 0.22121
Latency:0
Standard Deviation for 1 Thread sequential read/write 1 Megabyte-
Throughput: 0.49669
Latency:0

Standard Deviation for 1 Thread random read/write 1 byte-
Throughput: 0.03215
Latency:0
Standard Deviation for 1 Thread random read/write 1Kilobyte-
Throughput: 2.51661
Latency:0
Standard Deviation for 1 Thread random read/write 1 Megabyte-
Throughput: 0.40526
Latency:0

For 2 threads:

Access Type	Block Size	Throughput	Latency	Average Throughput	Average Latency
Sequential	1 byte	7.2865 7.2931 7.2789	0.000262 0.000257 0.000267	7.28617	0.00026
	1 Kilobyte	3748.63 3748.09 3747.92	0.0000056 0.0000048 0.0000051	3748.21333	0.00001
	1 Megabyte	6321.08 6321.19 6320.97	0.000000014 0.000000021 0.000000007	6321.08	0.000000014
Random	1 byte	23.2495 23.2555 23.1903	0.00084 0.00079 0.00091	23.23177	0.00084
	1 Kilobyte	2924.66 2923.96 2924.79	0.0000066 0.0000087 0.0000042	2924.47	0.0000065
	1 Megabyte	6285.12 6285.78 6284.84	0.00000093 0.00000078 0.00000103	6285.24667	0.00000091

Standard Deviation for 2 Thread sequential read/write 1 byte –
Throughput: 0.03604
Latency: 0
Standard Deviation for 2 Thread sequential read/write 1Kilobyte-
Throughput: 0.37072
Latency:0

Standard Deviation for 2 Thread sequential read/write 1 Megabyte-
Throughput: 0.11
Latency:0

Standard Deviation for 2 Thread random read/write 1 byte-
Throughput: 0.03215
Latency:0
Standard Deviation for 2 Thread random read/write 1Kilobyte-
Throughput: 0.44643
Latency:0
Standard Deviation for 2 Thread random read/write 1 Megabyte-
Throughput: 0.48263
Latency:0

Disk Benchmarking

- Only 20GB hard disk is assigned to the VM, and we are performing the experiments on this disk space. All the read and write operations to and from the memory are performed using the file system of this virtual OS.
- The accuracy of the result could have been increased if we had a dual boot or the entire system running on Ubuntu. The following results are based on above configuration.

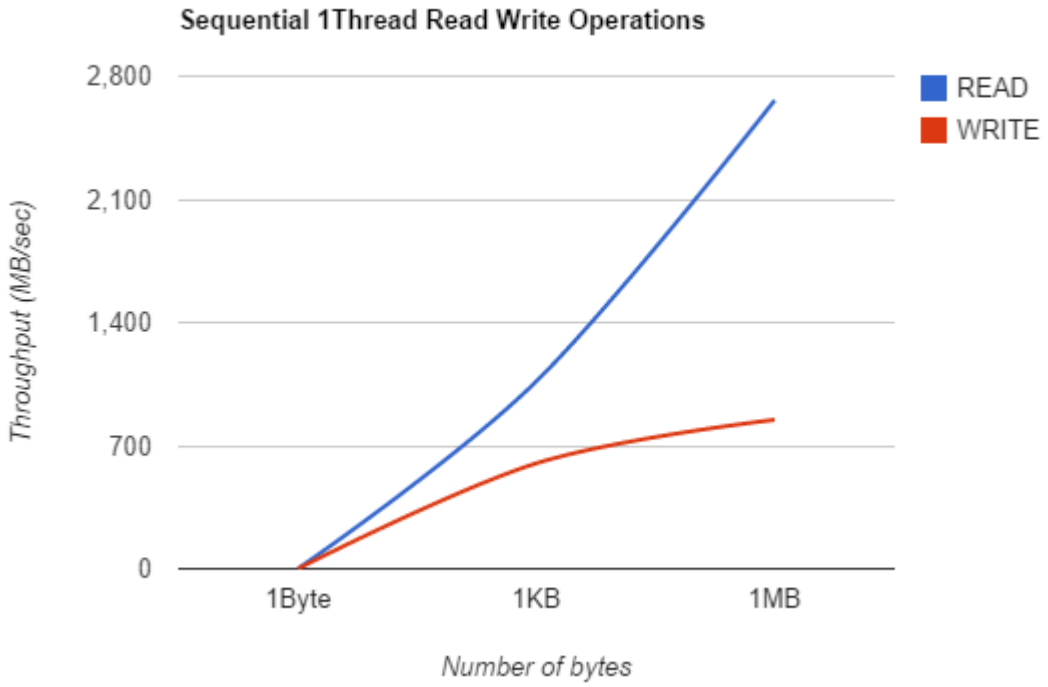
Concurrency level: 1 Thread for Read Operation + Sequential Access

Buffer Size	Latency (ms)	Throughput (MB/sec)
1 BYTE	0.000608	1.56
1 KILOBYTE	0.0009	1066.116
1 MEGABYTE	0.37	2663

Concurrency level: 1 Thread for Write Operation + Sequential Access

Buffer Size	Latency (ms)	Throughput (MB/sec)
1 BYTE	0.0006	1.48
1 KILOBYTE	0.0009	599.8
1 MEGABYTE	1.18	847

- **X-axis:** Number of Bytes
- **Y-axis**: Throughput (MB/sec)



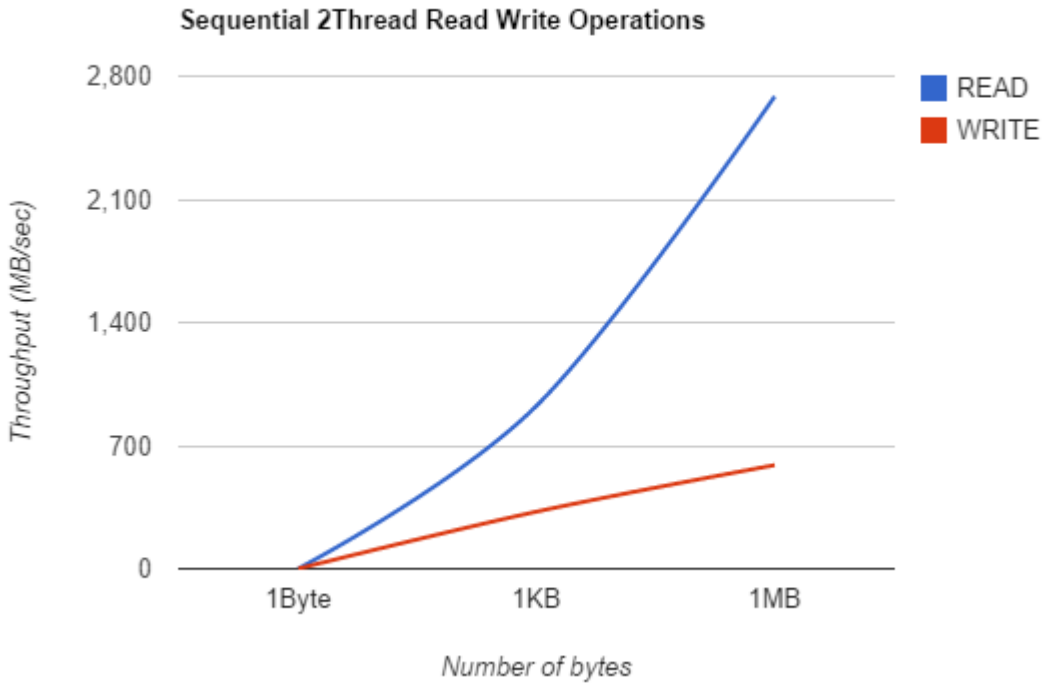
Concurrency level: 2 Thread for Read Operation + Sequential Access

Buffer Size	Latency (ms)	Throughput (MB/sec)
1 BYTE	0.00244	0.78
1 KILOBYTE	0.002	926
1 MEGABYTE	0.74	2685

Concurrency level: 2 Thread for Write Operation + Sequential Access

Buffer Size	Latency (ms)	Throughput (MB/sec)
1 BYTE	0.00218	0.87
1 KILOBYTE	0.00601	324.60
1 MEGABYTE	3.39	589.22

- **X-axis:** Number of Bytes
- **Y-axis**: Throughput (MB/sec)



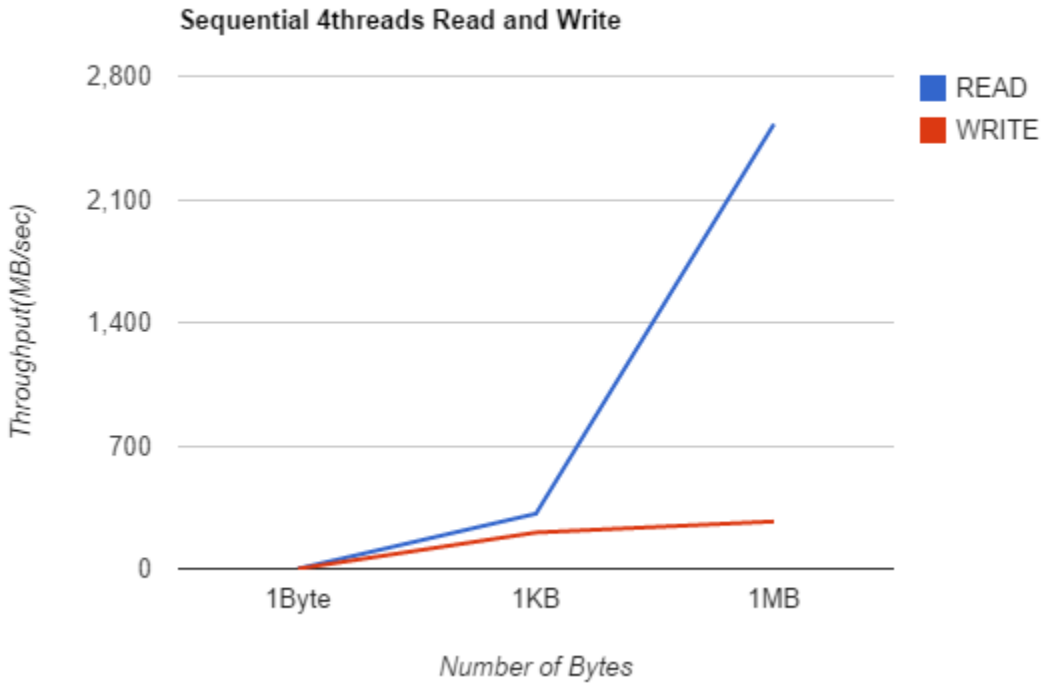
Concurrency level: 4 Thread for Read Operation + Sequential Access

Buffer Size	Latency (ms)	Throughput (MB/sec)
1 BYTE	0.01	0.36
1 KILOBYTE	0.12	313
1 MEGABYTE	1.5	2528

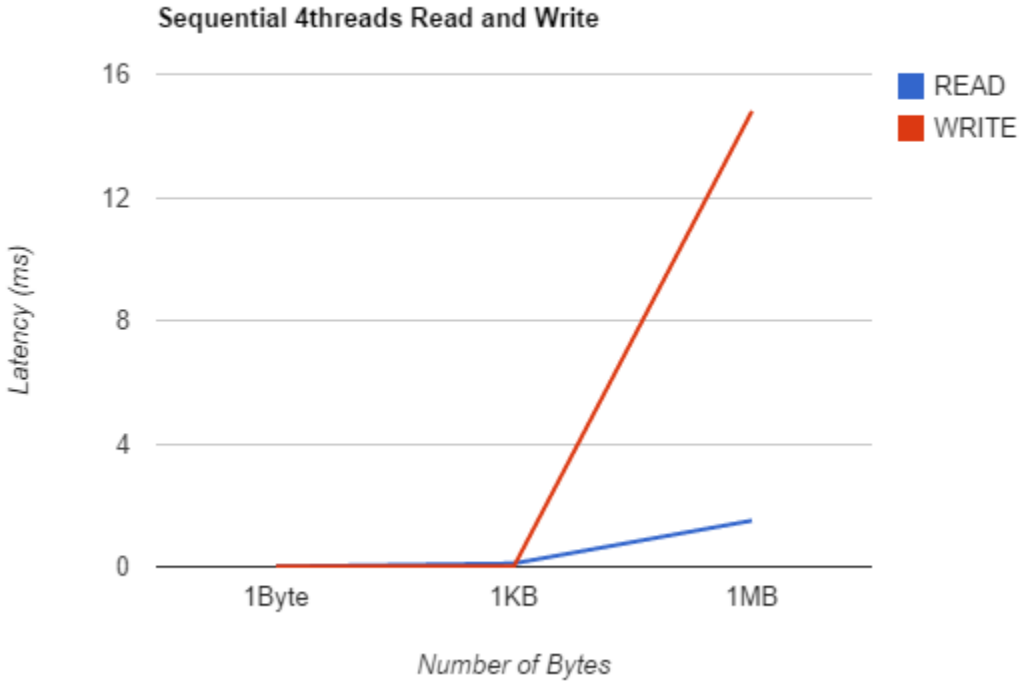
Concurrency level: 4 Thread for Write Operation + Sequential Access

Buffer Size	Latency (ms)	Throughput (MB/sec)
1 BYTE	0.009	0.41
1 KILOBYTE	0.018	206
1 MEGABYTE	14.8	269

- **X-axis:** Number of Bytes
- **Y-axis**: Throughput (MB/sec)



- **X-axis:** Number of Bytes
- **Y-axis**: Latency (ms)



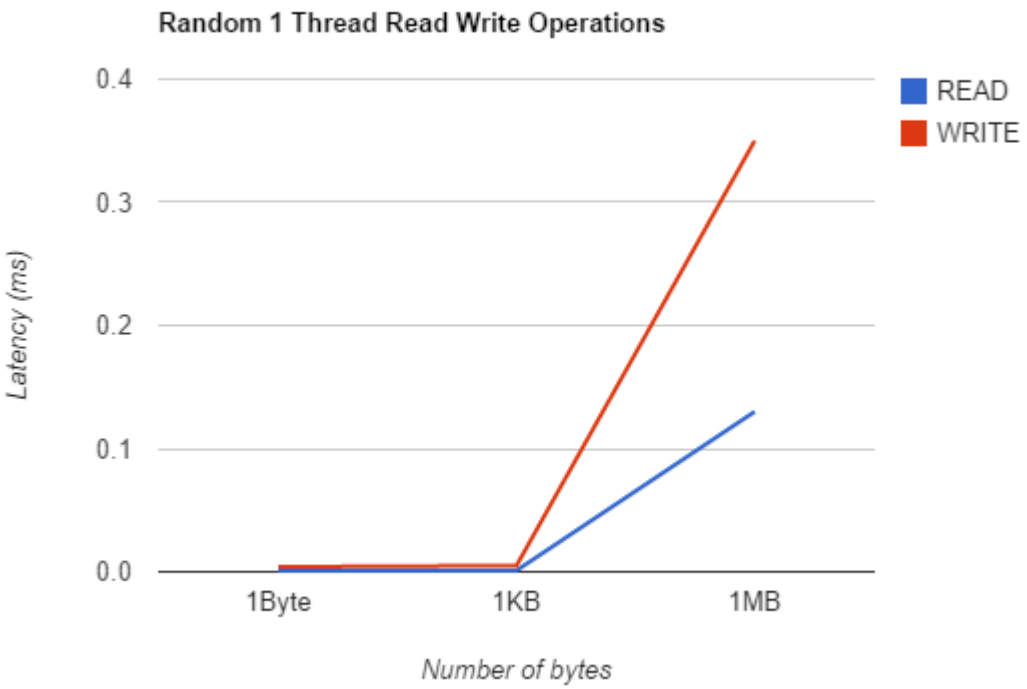
Concurrency level: 1 Thread for Read Operation + Random Access

Buffer Size	Latency (ms)	Throughput (MB/sec)
1 BYTE	0.001	0.879
1 KILOBYTE	0.001	5.4
1 MEGABYTE	0.13	7.4

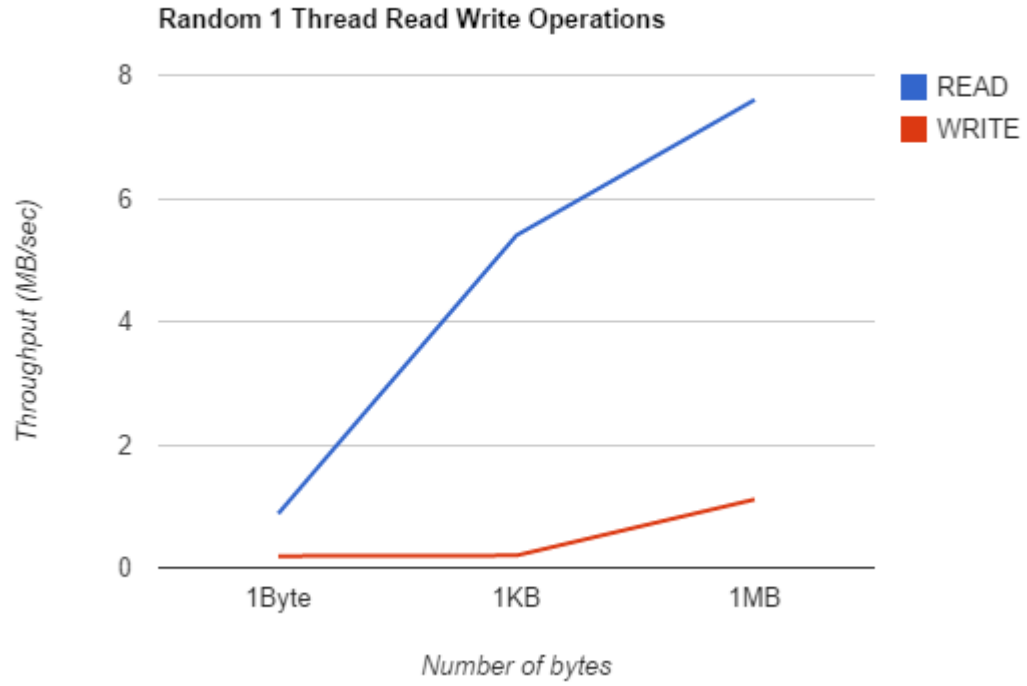
Concurrency level: 1 Thread for Write Operation + Random Access

Buffer Size	Latency (ms)	Throughput (MB/sec)
1 BYTE	0.004	0.19
1 KILOBYTE	0.01	0.2
1 MEGABYTE	0.35	1.11

- **X-axis:** Number of Bytes
- **Y-axis**: Latency (ms)



- **X-axis:** Number of Bytes
- **Y-axis**: Throughput (MB/sec)



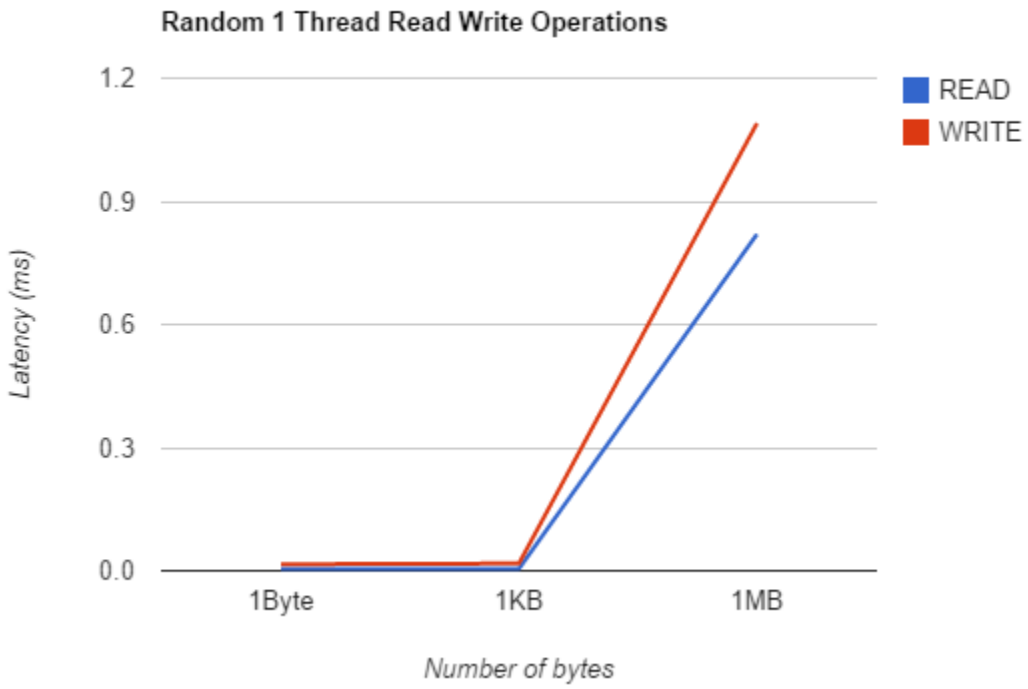
Concurrency level: 2 Thread for Read Operation + Random Access

Buffer Size	Latency (ms)	Throughput (MB/sec)
1 BYTE	0.005	0.3
1 KILOBYTE	0.005	0.38
1 MEGABYTE	0.82	2.3

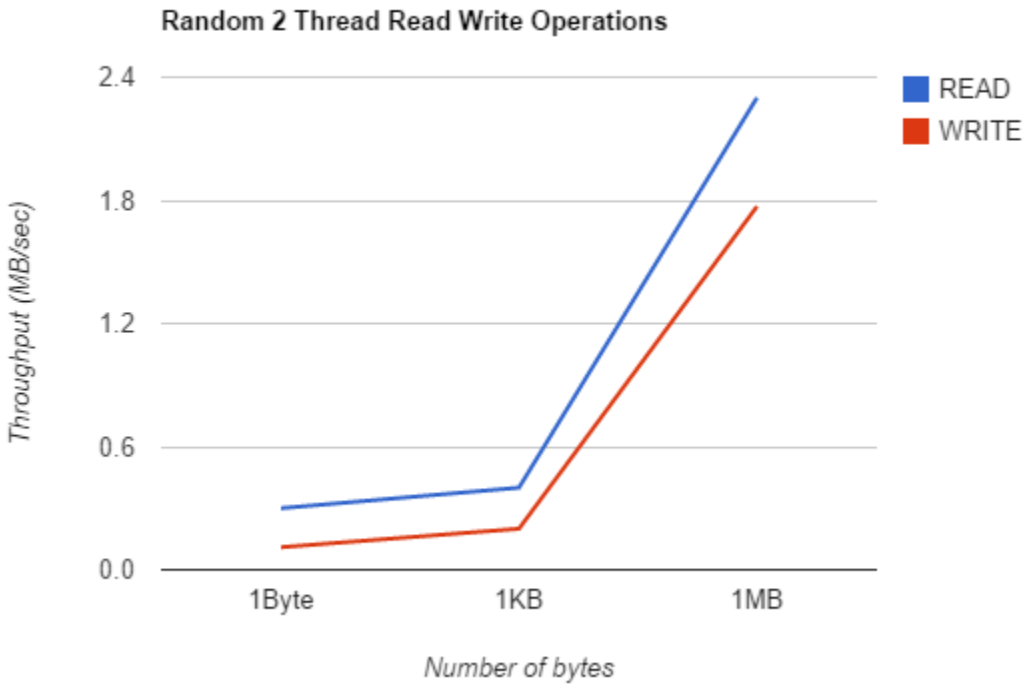
Concurrency level: 2 Thread for Write Operation + Random Access

Buffer Size	Latency (ms)	Throughput (MB/sec)
1 BYTE	0.016	0.11
1 KILOBYTE	0.018	0.2
1 MEGABYTE	1.09	1.77

- **X-axis:** Number of Bytes
- **Y-axis**: Latency (ms)



- **X-axis:** Number of Bytes
- **Y-axis**: Throghput (MB/sec)



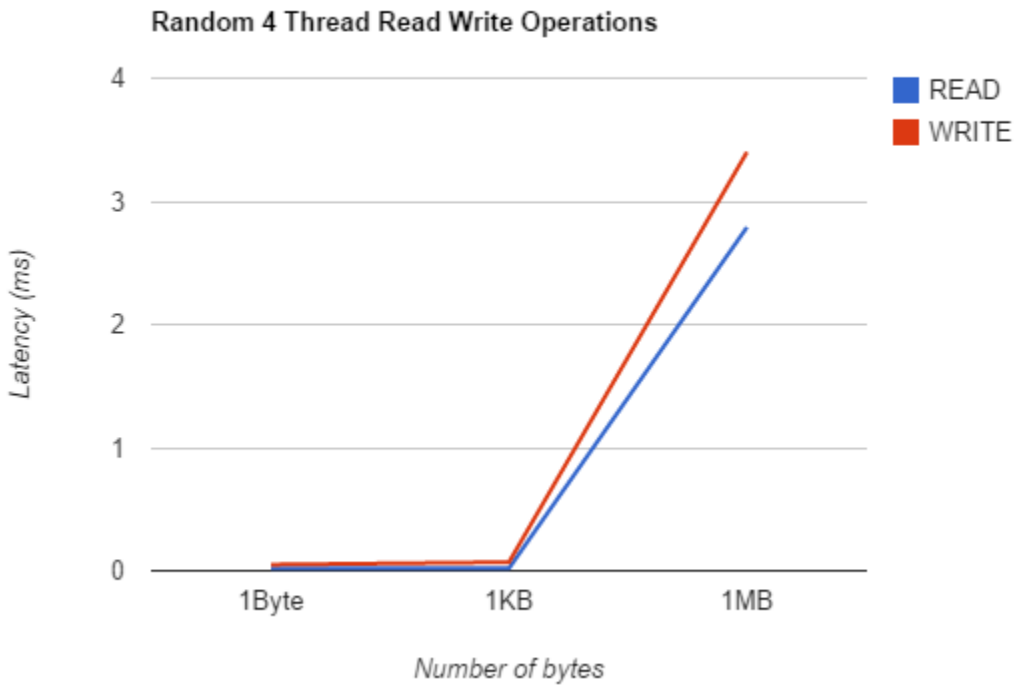
Concurrency level: 4 Thread for Read Operation + Random Access

Buffer Size	Latency (ms)	Throughput (MB/sec)
1 BYTE	0.02	0.18
1 KILOBYTE	0.3	1.25
1 MEGABYTE	2.79	1.4

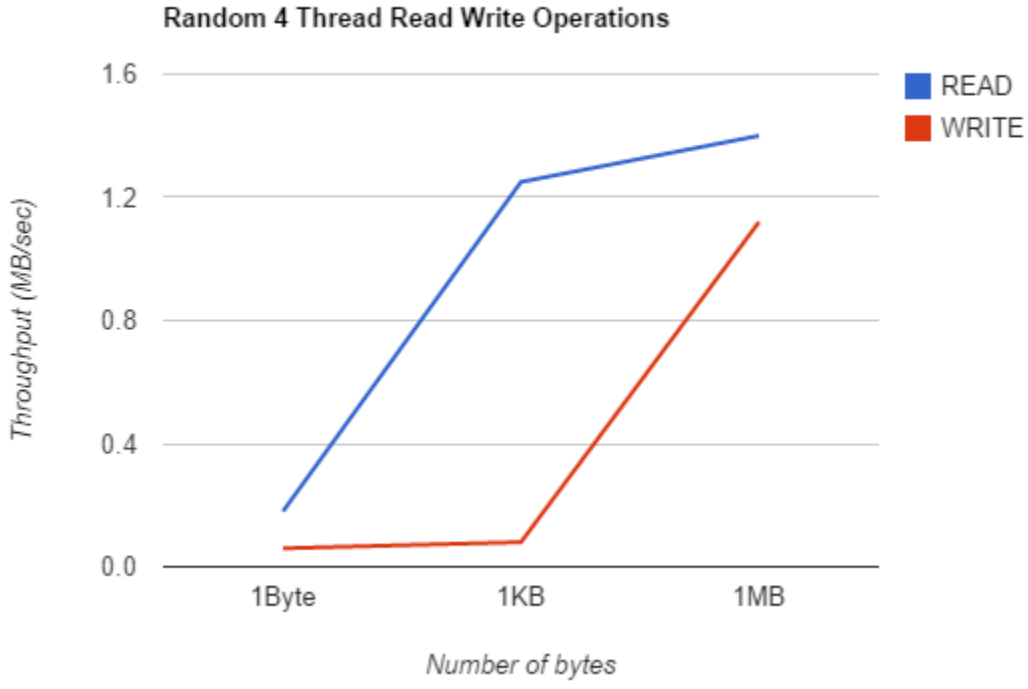
Concurrency level: 4 Thread for Write Operation + Random Access

Buffer Size	Latency (ms)	Throughput (MB/sec)
1 BYTE	0.05	0.06
1 KILOBYTE	0.07	0.08
1 MEGABYTE	3.4	1.12

- **X-axis:** Number of Bytes
- **Y-axis**: Latency (ms)



- **X-axis:** Number of Bytes
- **Y-axis**: Throghput (MB/sec)



- The optimal number of concurrency for best performance is 1.

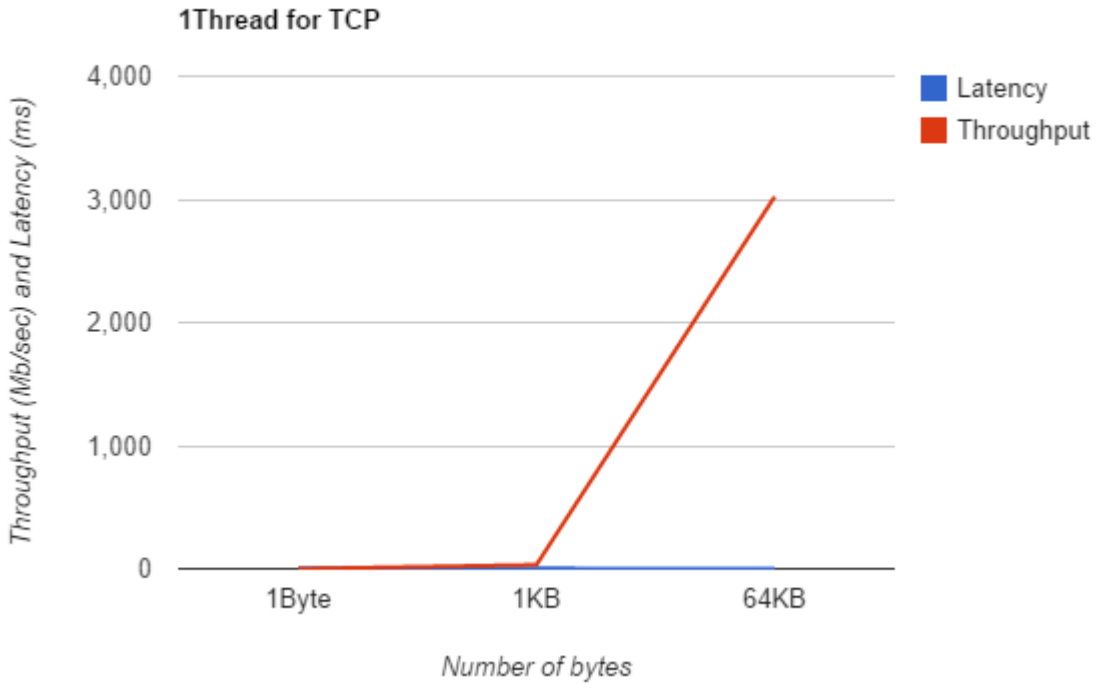
Network Benchmarking

We are testing the speed and latency in the VM ware player, which makes use of the network of the host machine. We have calculated results for TCP and UDP.

Concurrency level: 1 Thread for TCP

Buffer Size	Latency (ms)	Throughput (Mb/sec)
1 BYTE	6.01	0.013
1 KILOBYTE	2.49	32.1
64 KILOBYTE	0.004	3022.28

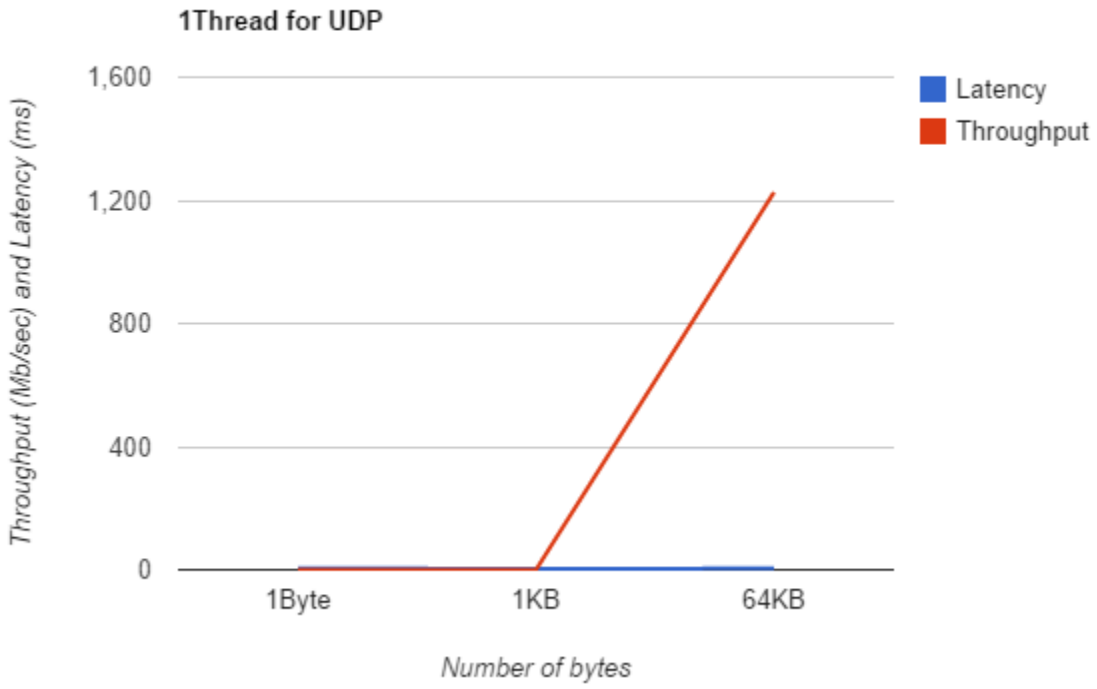
- **X-axis:** Number of Bytes
- **Y-axis**: Throughput (Mb/sec) and Latency (ms)



Concurrency level: 1 Thread for UDP

Buffer Size	Latency (ms)	Throughput (MB/sec)
1 BYTE	5.54	0.03
1 KILOBYTE	2.82	0.03
64 KILOBYTE	4.58	1226.17

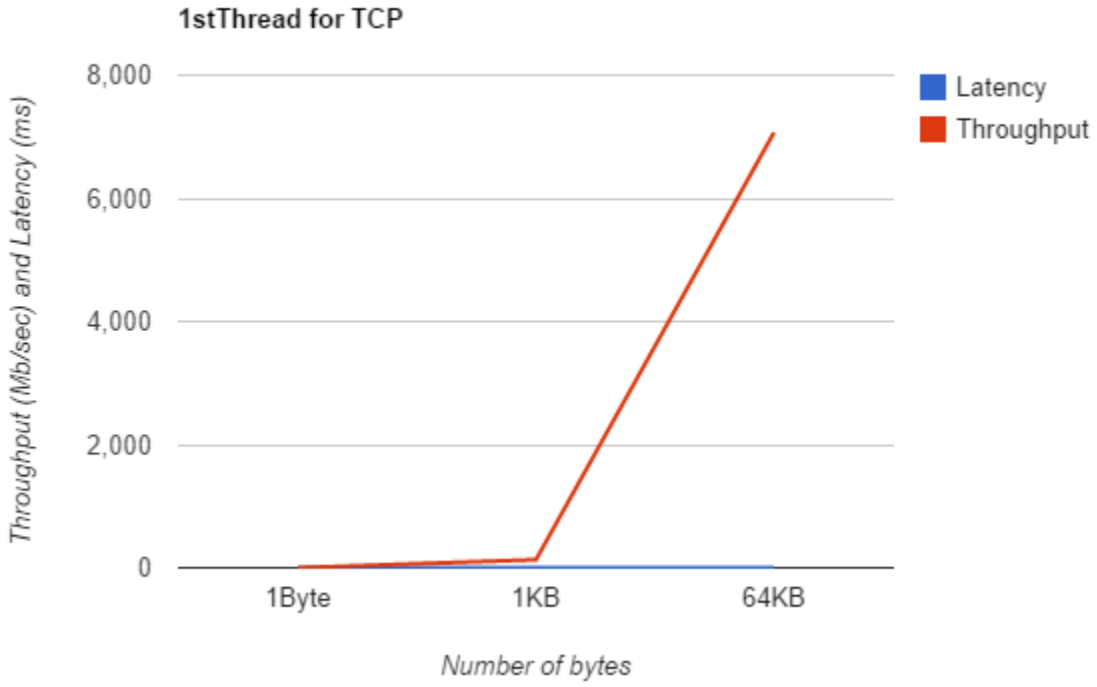
- **X-axis:** Number of Bytes
- ***Y-axis**: Throughput (Mb/sec) and Latency (ms)



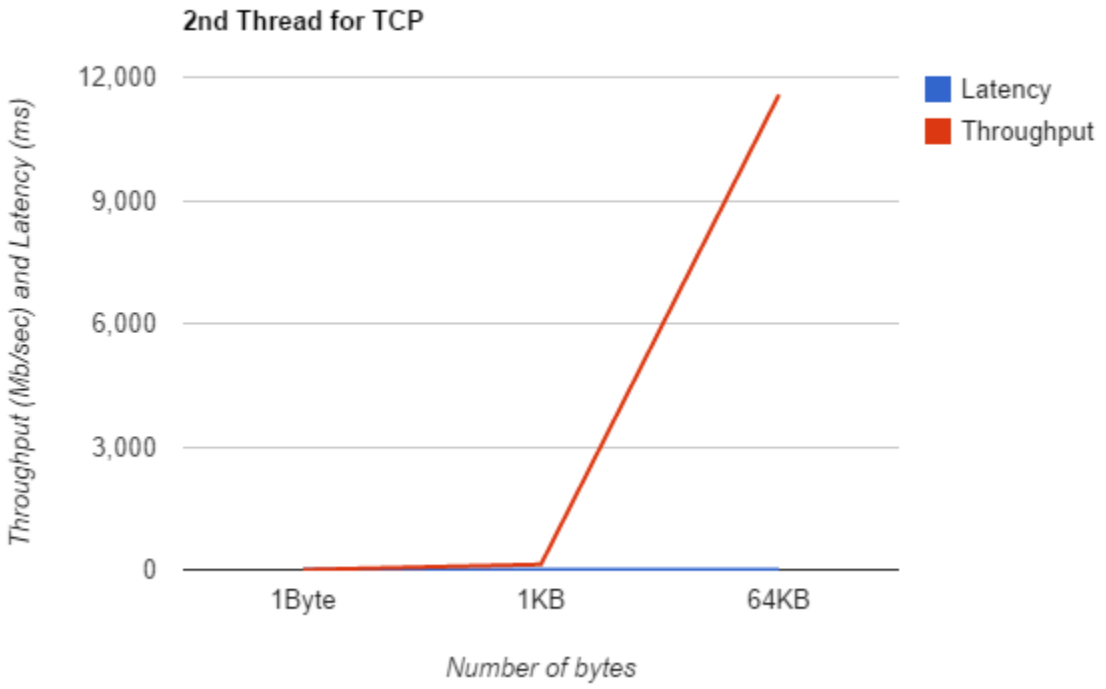
Concurrency level: 2 Thread for TCP

Buffer Size	Latency (ms)	Throughput (Mb/sec)
1 BYTE 1st Thread	8.52e-4	0.0090
1BYTE 2nd Thread	8.07e-4	0.0094
1 KILOBYTE 1st Thread	0.0060	131.49
1 KILOBYTE 2nd Thread	0.0039	131.24
64 KILOBYTE 1st Thread	0.0045	7068.64
64 KILOBYTE 2nd Thread	0.0035	11573.16

- **X-axis:** Number of Bytes
- **Y-axis**: Throughput (Mb/sec) and Latency (ms)



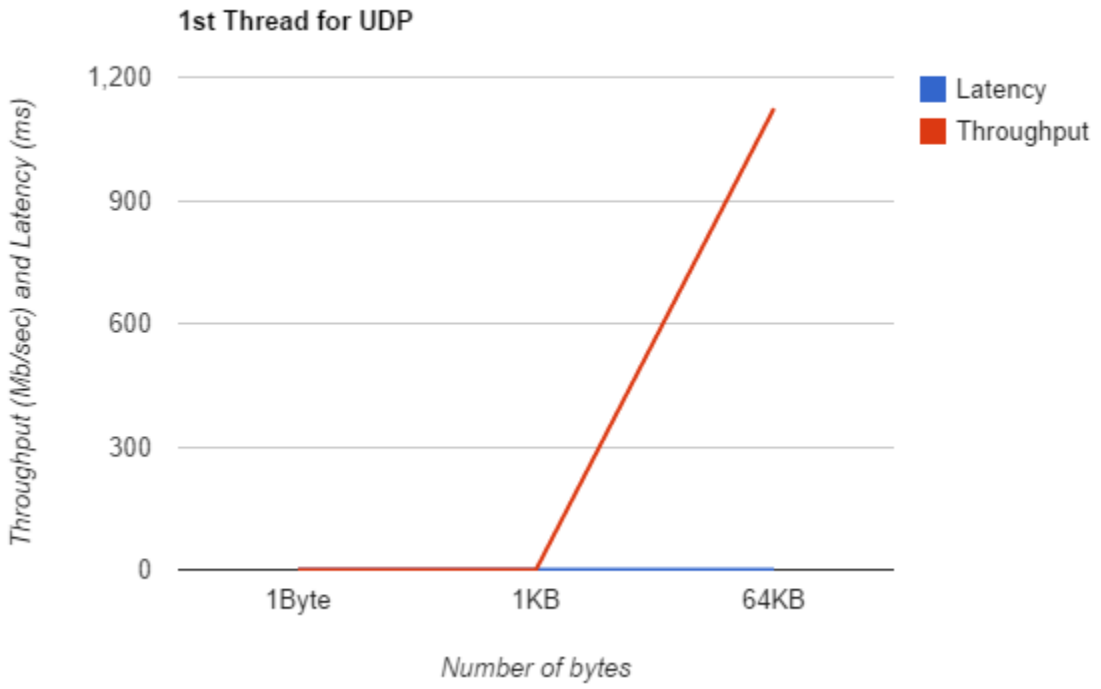
- **X-axis:** Number of Bytes
- **Y-axis**: Throughput (Mb/sec) and Latency (ms)



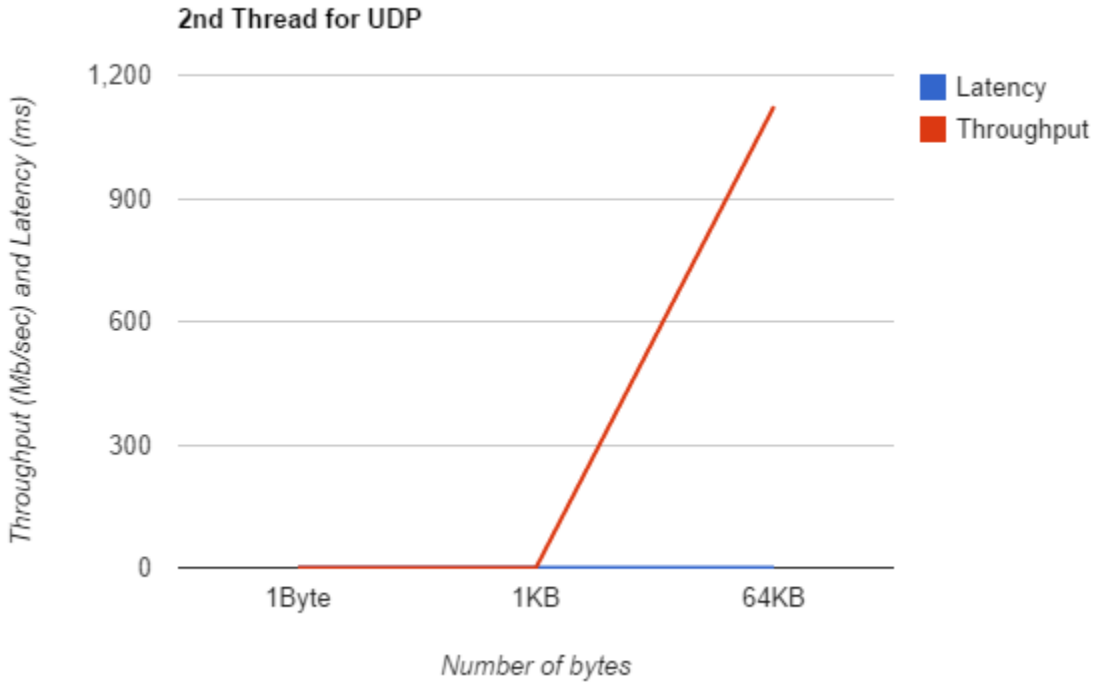
Concurrency level: 2 Thread for UDP

Buffer Size	Latency (ms)	Throughput (Mb/sec)
1 BYTE 1st Thread	0.0005	0.035
1BYTE 2nd Thread	0.0002	0.033
1 KILOBYTE 1st Thread	0.0005	0.039
1 KILOBYTE 2nd Thread	0.0005	0.033
64 KILOBYTE 1st Thread	0.0006	1123.77
64 KILOBYTE 2nd Thread	0.0006	1123.76

- **X-axis:** Number of Bytes
- **Y-axis**: Throughput (Mb/sec) and Latency (ms)



- **X-axis:** Number of Bytes
- **Y-axis**: Throughput (Mb/sec) and Latency (ms)



Extra credits:

Access Type	Block Size	Throughput	Latency	Average Throughput	Average Latency
TCP	1 byte	0.013 0.018 0.031	6.01 6.21 6.06	0.02067	6.09333
	1 Kilobyte	32.01 31.97 32.42	2.49 2.53 2.41	32.13333	2.47667
	64 Kilobyte	3022.28 3022.01 3021.87	0.004 0.012 0.001	3022.05333	0.00567
UDP	1 byte	0.030 0.039 0.041	5.54 5.97 5.21	0.03667	5.57333
	1 Kilobyte	0.030 0.045 0.009	2.82 2.71 2.09	0.028	2.54
	64 Kilobyte	1226.17 1226.61 1225.85	4.58 4.11 4.92	1226.21	4.53667

Formula for Standard Deviation :

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

Standard Deviation for 1 Thread and 1 byte using TCP –
Throughput: 0.00929
Latency: 0.10408
Standard Deviation for 1 Thread and 1Kilobyte using TCP -
Throughput: 0.24906
Latency: 0.0611
Standard Deviation for 1 Thread and 64 Kilobyte using TCP -
Throughput: 0.20841

Latency: 0.00569

Standard Deviation for 1 Thread and 1 byte using UDP -
Throughput: 0.00586
Latency: 0.38109

Standard Deviation for 1 Thread and 1 Kilobyte using UDP -
Throughput: 0.01808
Latency: 0.39357

Standard Deviation for 1 Thread and 64 Kilobyte using UDP -
Throughput: 0.38158
Latency: 0.40673

Extra Credit:

We have implemented IPERF. First we have installed it and all the following things are done after that.

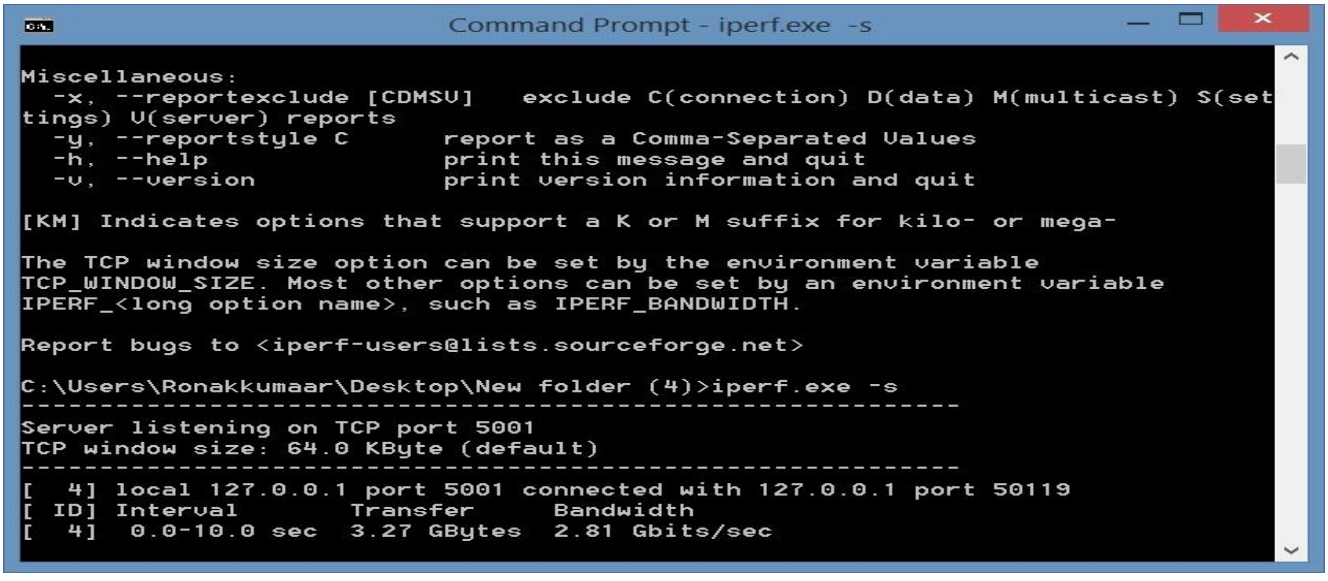
IPERF :
After installing iphere , set environment variables for iphere in your system. Now open the two command prompt for client and server.

TCP:

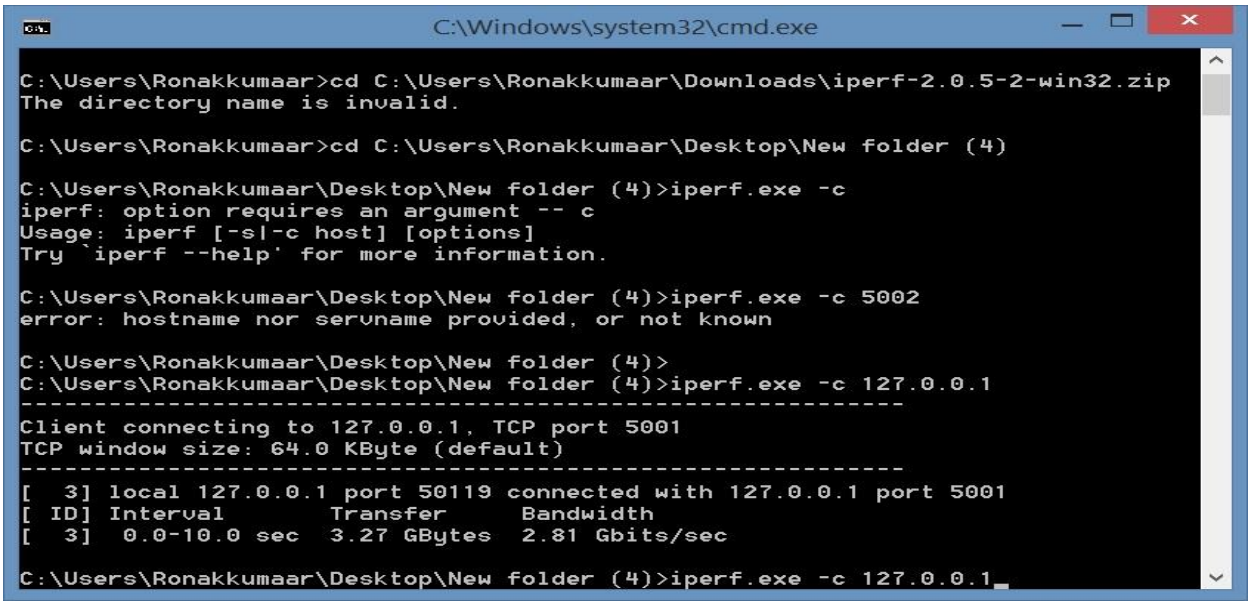
Client/Server (1 thread)

1) Client side: execute command : iphere.exe –c "ipaddress of server"
2) Server side: execute command : iphere.exe –s

Server :



Client:



Observation:

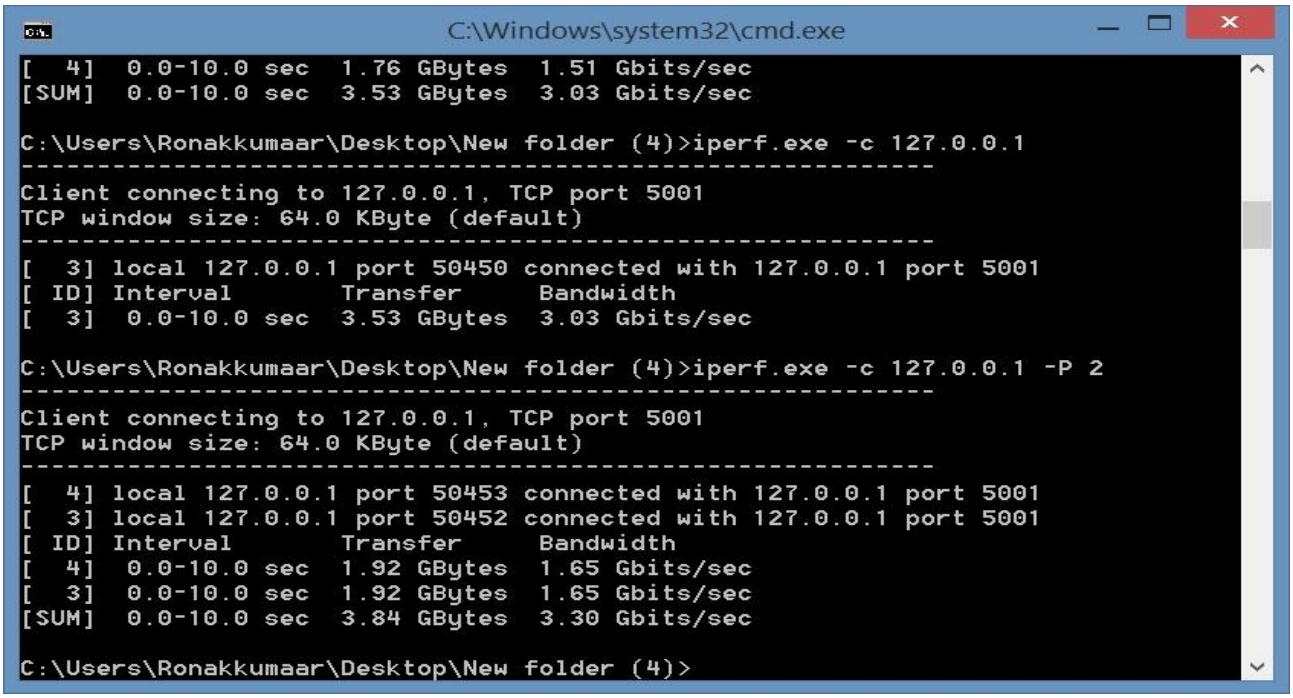
Data Transfer: 3.27 Gbytes
Bandwidth: 2.81 Gbits/sec

Client/Server (2 threads)

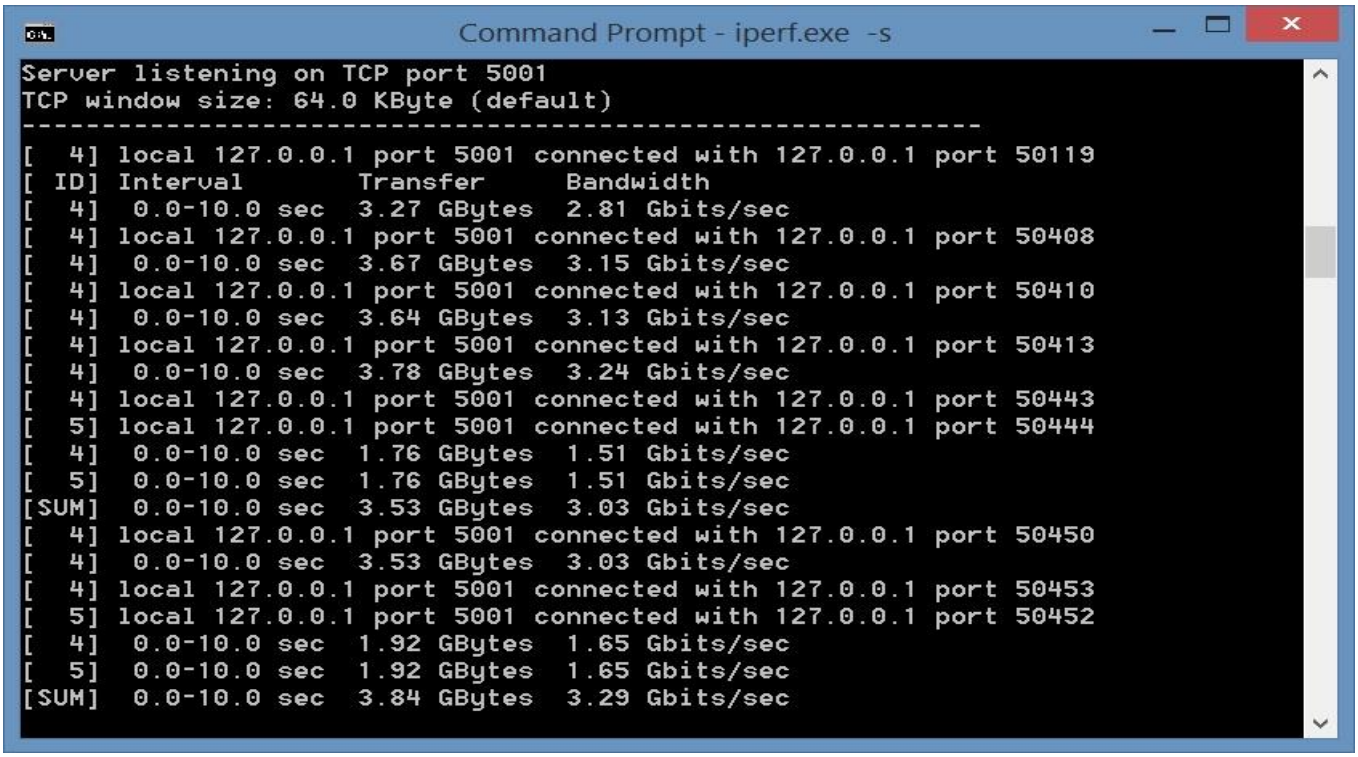
1 Client Side : iperf -c "ipaddress" – P n
-P -> It is for parallel execution using multithreading
n -> Number of Threads.

2. Server Side: iperf –s

Client :


C:\Windows\system32\cmd.exe
[4] 0.0-10.0 sec 1.76 GBytes 1.51 Gbits/sec
[SUM] 0.0-10.0 sec 3.53 GBytes 3.03 Gbits/sec
C:\Users\Ronakkumaar\Desktop\New folder (4)>iperf.exe -c 127.0.0.1
Client connecting to 127.0.0.1, TCP port 5001
TCP window size: 64.0 KByte (default)
[3] local 127.0.0.1 port 50450 connected with 127.0.0.1 port 5001
[ID] Interval   Transfer   Bandwidth
[3] 0.0-10.0 sec 3.53 GBytes 3.03 Gbits/sec
C:\Users\Ronakkumaar\Desktop\New folder (4)>iperf.exe -c 127.0.0.1 -P 2
Client connecting to 127.0.0.1, TCP port 5001
TCP window size: 64.0 KByte (default)
[4] local 127.0.0.1 port 50453 connected with 127.0.0.1 port 5001
[3] local 127.0.0.1 port 50452 connected with 127.0.0.1 port 5001
[ID] Interval   Transfer   Bandwidth
[4] 0.0-10.0 sec 1.92 GBytes 1.65 Gbits/sec
[3] 0.0-10.0 sec 1.92 GBytes 1.65 Gbits/sec
[SUM] 0.0-10.0 sec 3.84 GBytes 3.30 Gbits/sec
C:\Users\Ronakkumaar\Desktop\New folder (4)>

Server :


Command Prompt - iperf.exe -s
Server listening on TCP port 5001
TCP window size: 64.0 KByte (default)
[4] local 127.0.0.1 port 5001 connected with 127.0.0.1 port 50119
[ID] Interval   Transfer   Bandwidth
[4] 0.0-10.0 sec 3.27 GBytes 2.81 Gbits/sec
[4] local 127.0.0.1 port 5001 connected with 127.0.0.1 port 50408
[4] 0.0-10.0 sec 3.67 GBytes 3.15 Gbits/sec
[4] local 127.0.0.1 port 5001 connected with 127.0.0.1 port 50410
[4] 0.0-10.0 sec 3.64 GBytes 3.13 Gbits/sec
[4] local 127.0.0.1 port 5001 connected with 127.0.0.1 port 50413
[4] 0.0-10.0 sec 3.78 GBytes 3.24 Gbits/sec
[4] local 127.0.0.1 port 5001 connected with 127.0.0.1 port 50443
[5] local 127.0.0.1 port 5001 connected with 127.0.0.1 port 50444
[4] 0.0-10.0 sec 1.76 GBytes 1.51 Gbits/sec
[5] 0.0-10.0 sec 1.76 GBytes 1.51 Gbits/sec
[SUM] 0.0-10.0 sec 3.53 GBytes 3.03 Gbits/sec
[4] local 127.0.0.1 port 5001 connected with 127.0.0.1 port 50450
[4] 0.0-10.0 sec 3.53 GBytes 3.03 Gbits/sec
[4] local 127.0.0.1 port 5001 connected with 127.0.0.1 port 50453
[5] local 127.0.0.1 port 5001 connected with 127.0.0.1 port 50452
[4] 0.0-10.0 sec 1.92 GBytes 1.65 Gbits/sec
[5] 0.0-10.0 sec 1.92 GBytes 1.65 Gbits/sec
[SUM] 0.0-10.0 sec 3.84 GBytes 3.29 Gbits/sec

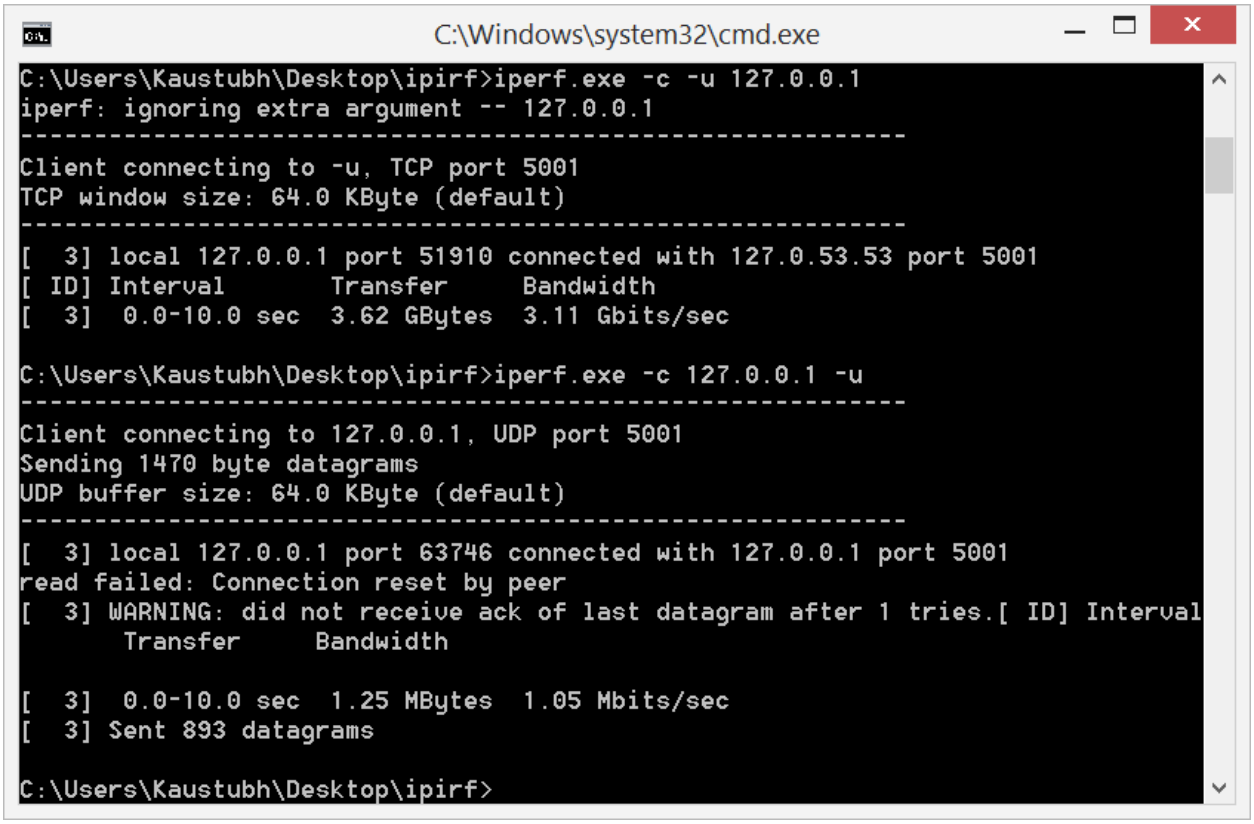
Observation:
Thread 1:
Data Transfer: 1.92 Gbytes
Bandwidth: 1.6 Gbits/sec
Thread 2:
Data Transfer: 1.92 Gbytes
Bandwidth: 1.6 Gbits/sec
Sum :
Data Transfer: 1.92 Gbytes
Bandwidth: 3.29 Gbits/sec

Efficiency: As per the observation we can conclude that 2 threads gives more efficient throughput as compare to 1 thread.

UDP:

1) Client side: execute command : iphere.exe –c "ipaddress of server" –u

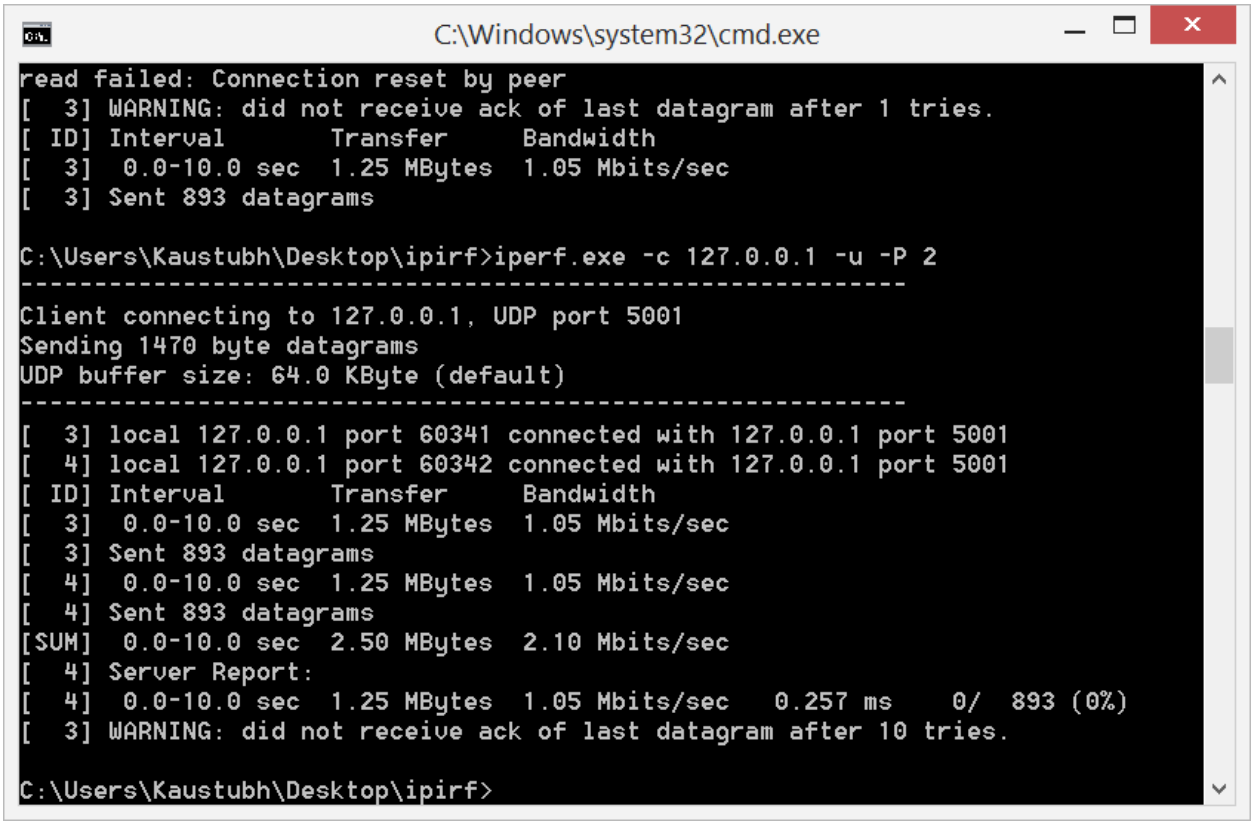
   -u for :'UDP'



Observation:
Data Transfer: 1.25 Mbytes
Bandwidth: 1.05 Mbits/sec

Client/Server (2 threads)

1 Client Side : iperf -c “ipaddress” – P n
-P -> It is for parallel execution using multithreading
n -> Number of Threads.



Observation:
Thread 1:
Data Transfer: 1.25 Mbytes
Bandwidth: 1.05 Mbits/sec
Thread 2:
Data Transfer: 1.25 Mbytes
Bandwidth: 1.05 Mbits/sec

Sum :
Data Transfer: 2.50 Mbytes
Bandwidth: 2.10 Mbits/sec

Efficiency: As per the observation we can conclude that there is no difference when executing with one thread or 2 threads.