Performance Evaluation

1. CPU:

   Floating point operations:

   Observation 1:

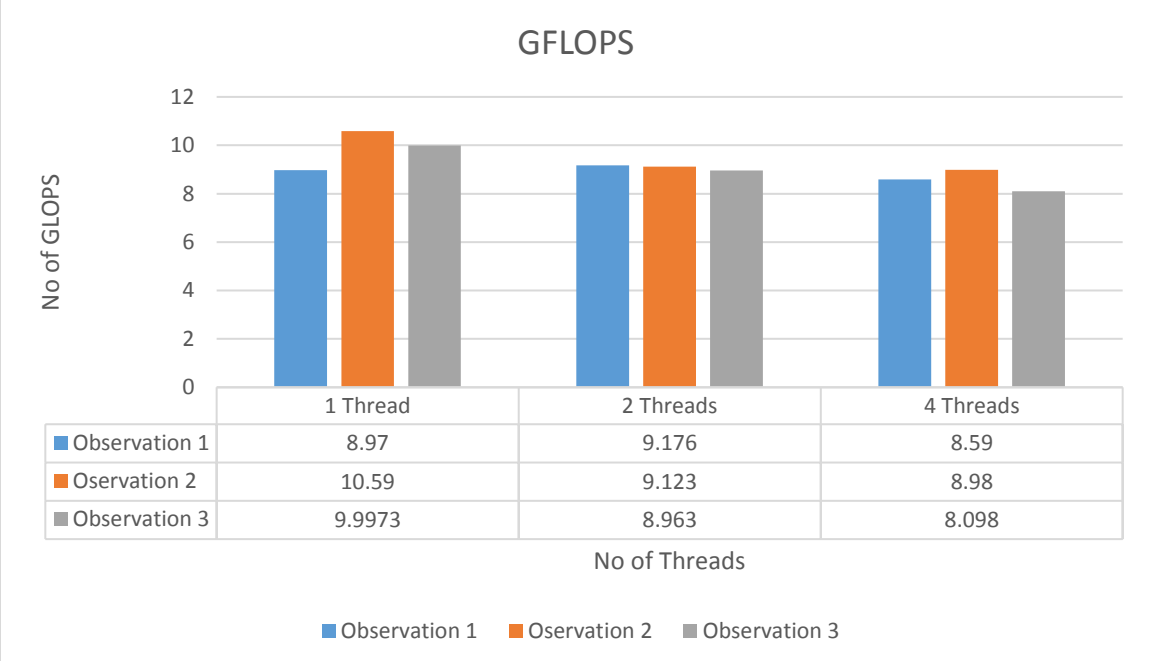
No of GFlops	No of threads
8.97	1
9.176	2
8.590	4

   Observation 2:

No of GFlops	No of threads
10.59	1
9.123	2
8.98	4

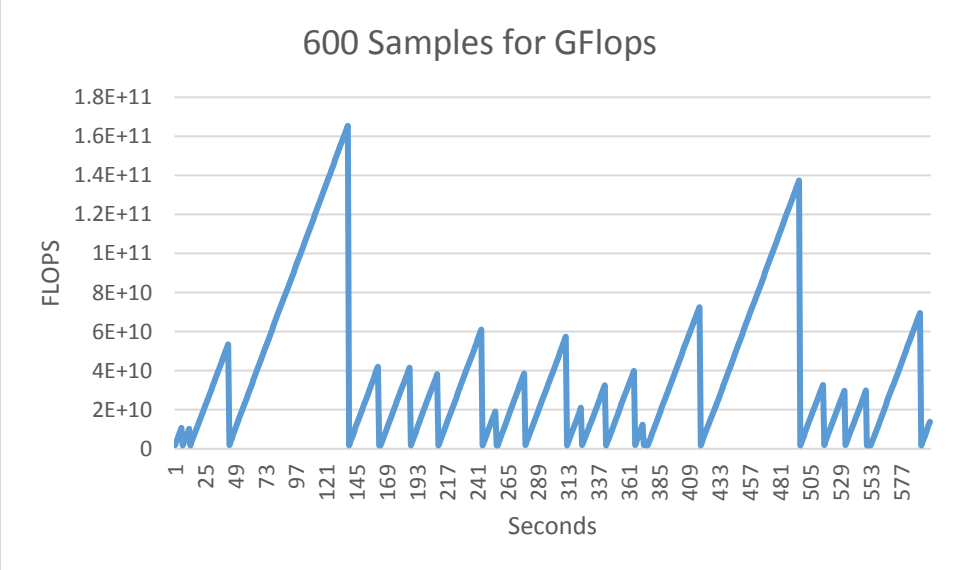
   Observation 3:

No of GFlops	No of threads
9.997	1
8.96	2
8.098	4



Explanation:

Here graphs shows no of GFLOPs of CPU on y-axis and no of threads for calculating GFLOPs on X-axis. About 3 observations were carried out for measuring the no of GFLOPs of the CPU. Each observation by taken for each 1, 2 and 4 threads running concurrently. Here as it can be seen from graph the average GFLOPs remains around 9.12 with 1, 2 or 4 threads running concurrently. From the consistency of the system it can be said that the GFLOPs for Amazon EC2 t2.micro instance should be around 9-10.



Explanation:

Graph shows plot of Flops taken per second for a time period of 10 minutes.

Integer Operations:

Observation 1:

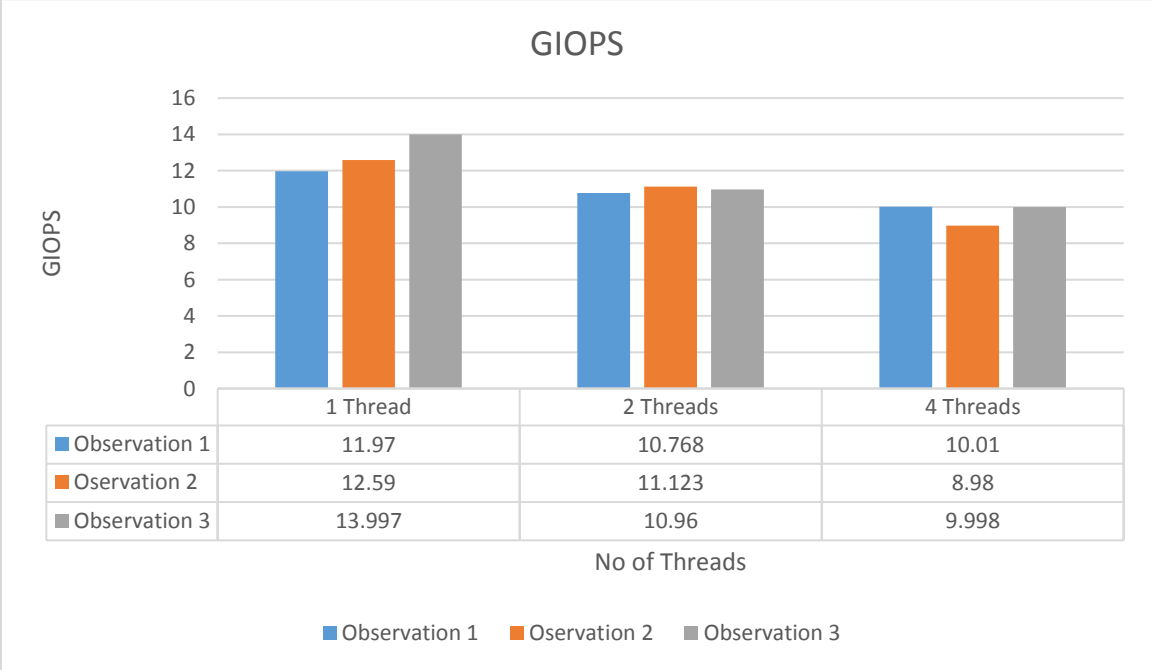
No of GIops	No of threads
11.97	1
10.768	2
10.010	4

Observation 2:

No of GIops	No of threads
12.59	1
11.123	2
8.98	4

Observation 3:

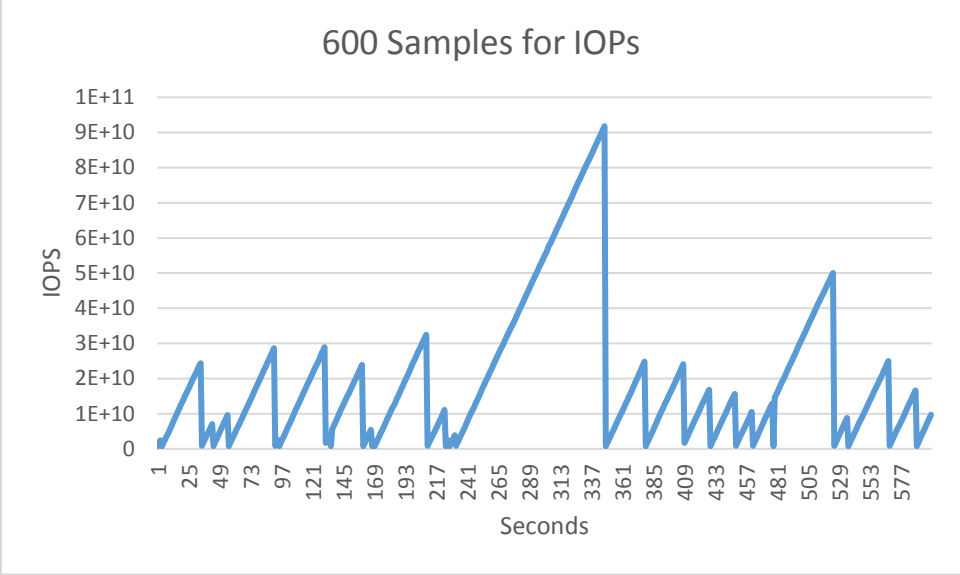
No of GIops	No of threads
13.997	1
10.96	2
9.998	4



Explanation:

Here graphs shows no of GIOPs of CPU on y-axis and no of threads for calculating GIOPs on X-axis. About 3 observations were carried out for measuring the no of GIOPs of the CPU. Each observation by taken for each 1, 2 and 4 threads running concurrently.

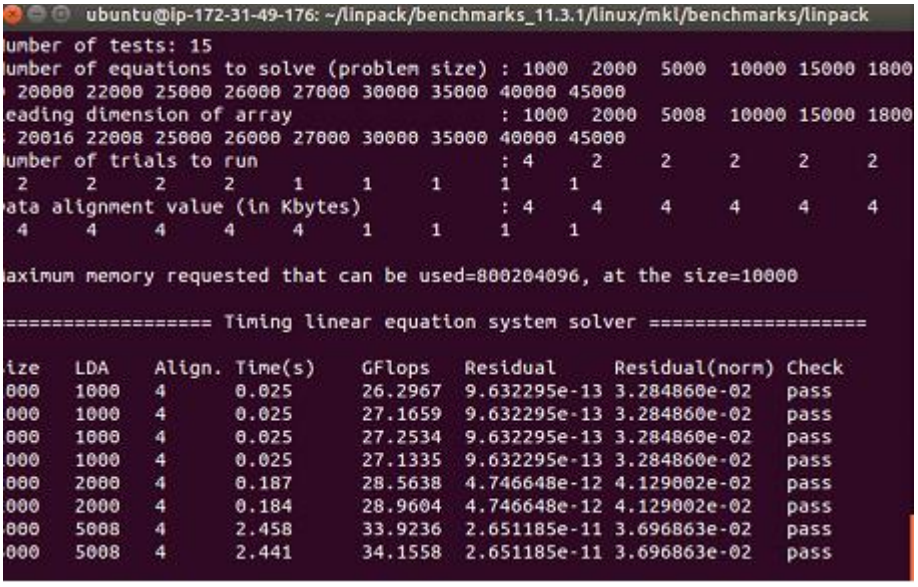
Here as it can be seen from graph the average GIOPs remains around 11 with 1, 2 or 4 threads running concurrently. From the consistency of the evaluation results it can be said that the GIOPs for the system should be around 11-12



Explanation:

Graph shows plot of lops taken per second for a time period of 10 minutes.

LINPACK:

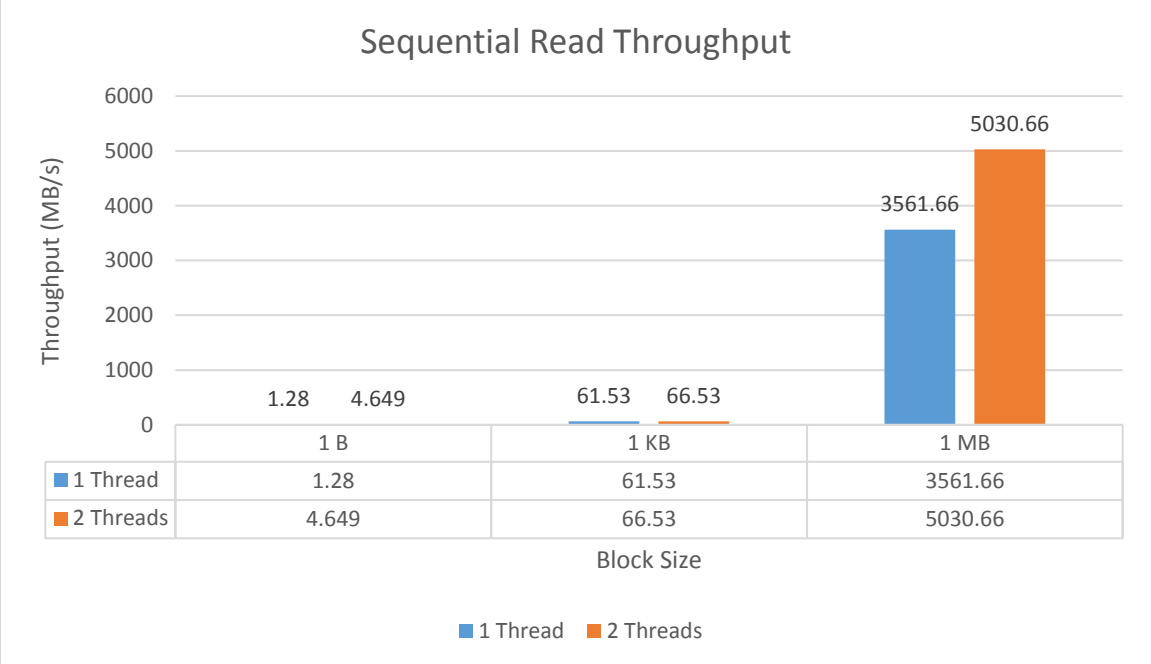


Explanations

2. DISK:

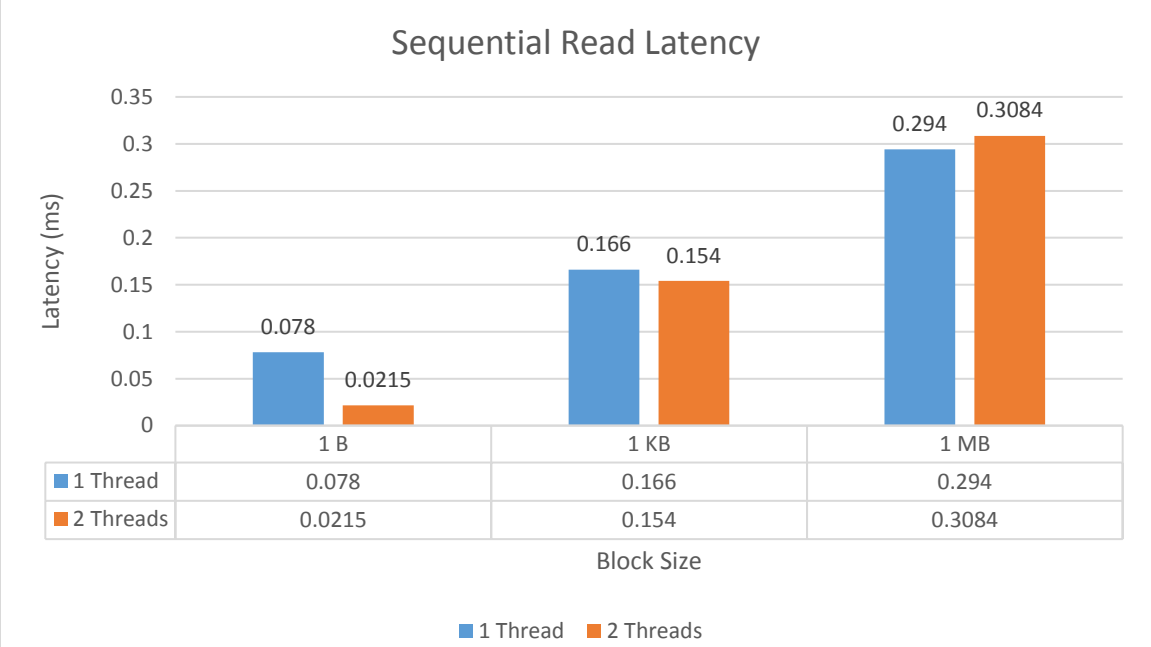
Sequential Read:

Data Size	Throughput (MB/s)	Latency (ms)	Threads
1 B	1.280	0.078	1
1 KB	61.53	0.166	1
1 MB	3561.66	0.294	1
1 B	4.649	0.0215	2
1 KB	66.53	0.154	2
1 MB	5030.66	0.3084	2



Explanation:

Here y-axis represents the throughput in MB/s and x-axis represents the Data size read sequentially. Throughput increases as size of data read increases. This is because throughput is calculated by data read/ time taken. So as data size increases throughput increases. Also when running 2 threads concurrently, throughput obtained is similar. Apart from this on comparing results with IOZONE read similar results were obtained.

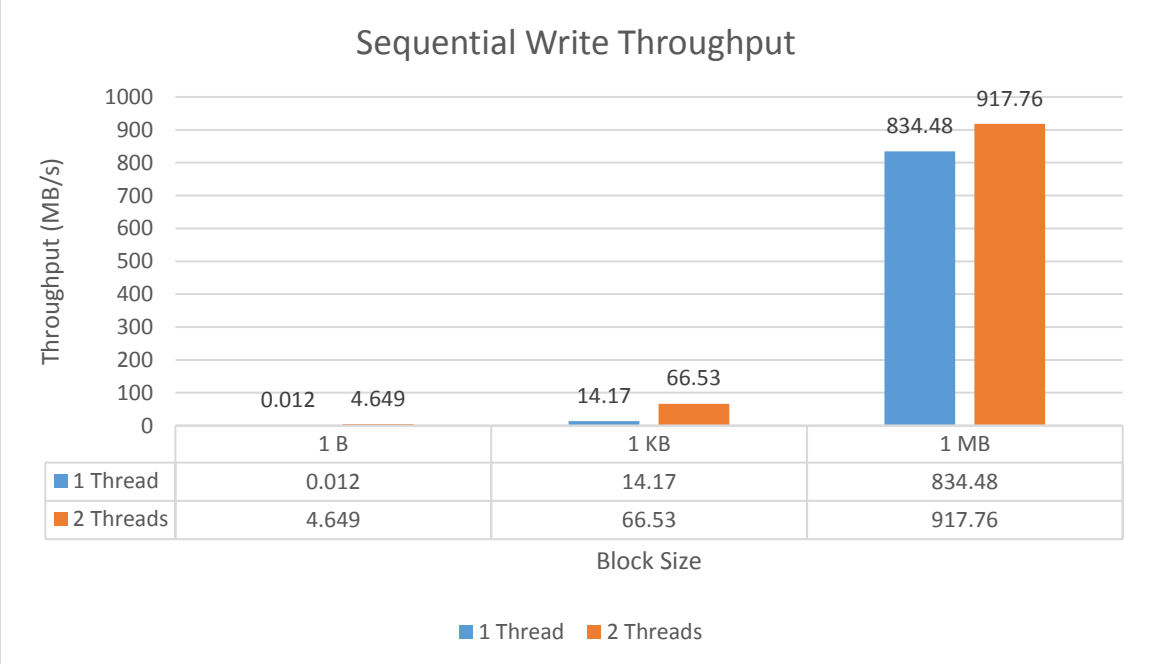


Explanation:

Here Y-axis represent Latency (ms) and X-axis represent data size that is being read sequentially. Latency is the time taken for the disk to read and reply back. So as data increases the latency increases. Hence the graph makes sense. Also on comparing the results with that of IOZONE, implied the correctness of the results.

Sequential Write:

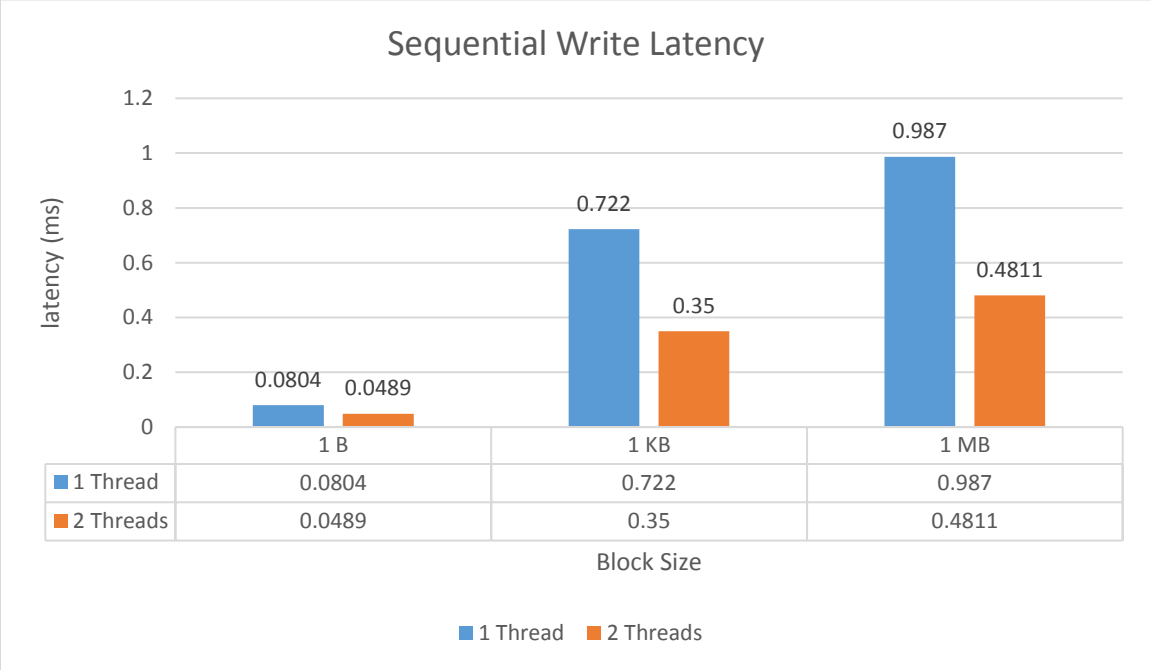
Data Size	Throughput (MB/s)	Latency (ms)	Threads
1 B	0.012	0.0804	1
1 KB	14.17	0.722	1
1 MB	834.48	0.987	1
1 B	4.649	0.0489	2
1 KB	66.53	0.350	2
1 MB	917.76	0.4811	2



Explanation:

Here y-axis represents the throughput in MB/s and x-axis represents the Data size written sequentially. Throughput increases as size of data read increases. This is because throughput is calculated by data read/ time taken. So as data size increases throughput increases. Also when running 2 threads concurrently, throughput obtained is similar.

Sequential write takes some time more than sequential read it is quite obvious and implies the correctness of the results.Apart from this on comparing results with IOZONE write similar results were obtained.

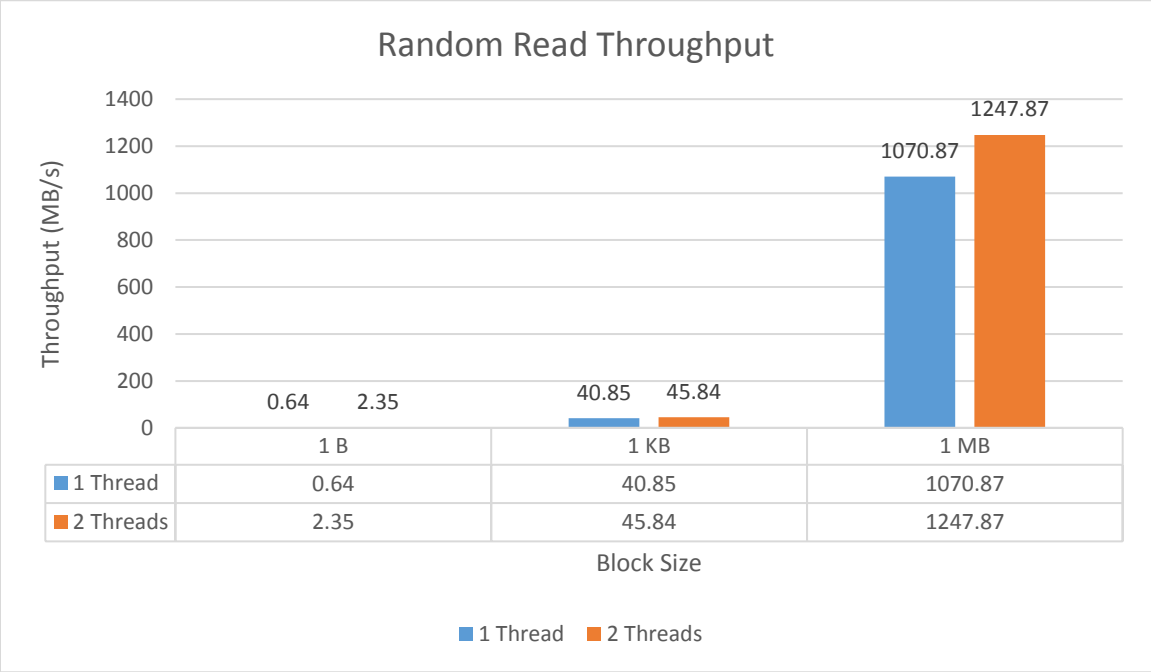


Explanation:

Here Y-axis represent Latency (ms) and X-axis represent data size that is being written sequentially. Latency is the time taken for the disk to read and reply back. So as data increases the latency increases. Hence the graph makes sense. Also on comparing the results with that of IOZONE, implied the correctness of the results.

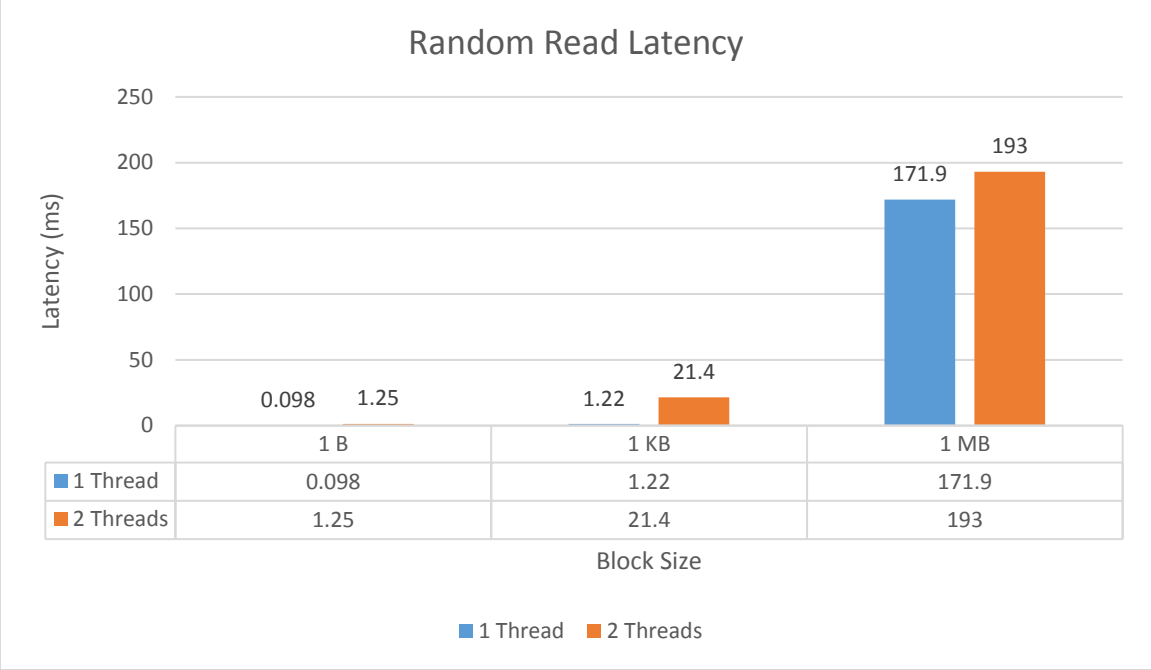
Random Read:

Data Size	Throughput (MB/s)	Latency (ms)	Threads
1 B	0.64	0.098	1
1 KB	40.85	1.22	1
1 MB	1070.87	171.9	1
1 B	2.35	122.5	2
1 KB	45.84	21.4	2
1 MB	1247.87	193	2



Explanation:

Here y-axis represents the throughput in MB/s and x-axis represents the Data size read randomly. Throughput increases as size of data read increases. This is because throughput is calculated by data read/ time taken. So as data size increases throughput increases.
Also when running 2 threads concurrently, throughput obtained is similar.
Apart from this on comparing results with IOZONE read similar results were obtained.
Random read takes some time more than sequential read. This is because in random read a Math.random() function is defined which defines the position from which to read the data. Hence, random read takes more time compared to sequential read and hence lower throughput.

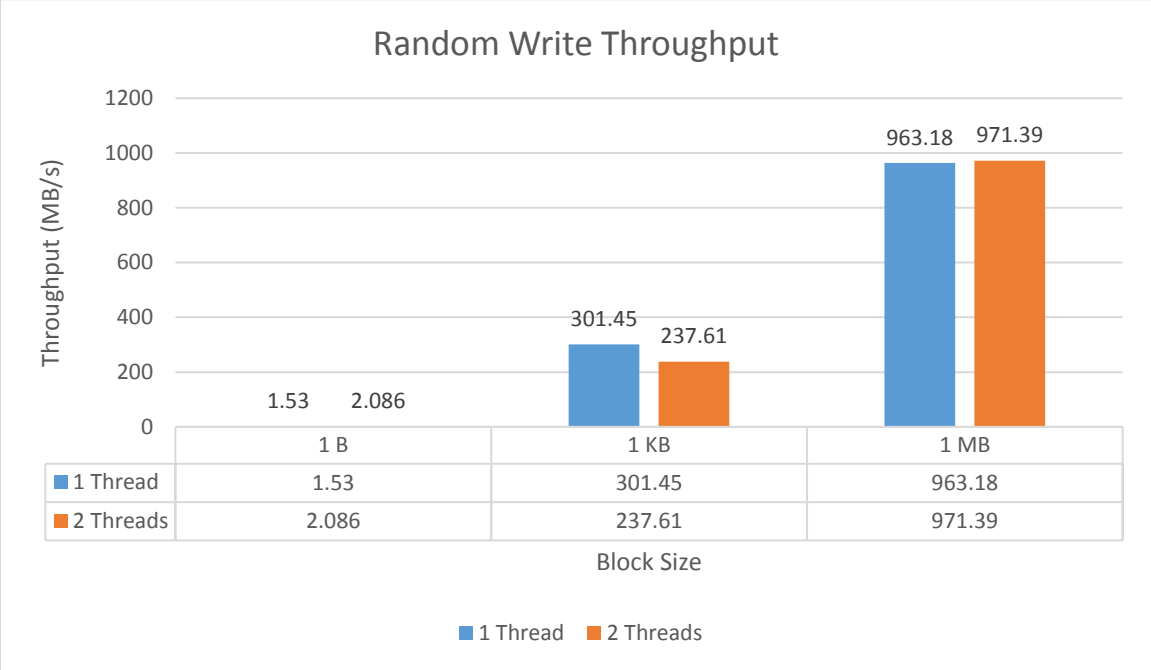


Explanation:

Here Y-axis represent Latency (ms) and X-axis represent data size that is being read randomly. Latency is the time taken for the disk to read and reply back. So as data increases the latency increases. Hence the graph makes sense. Also on comparing the results with that of IOZONE, implied the correctness of the results.

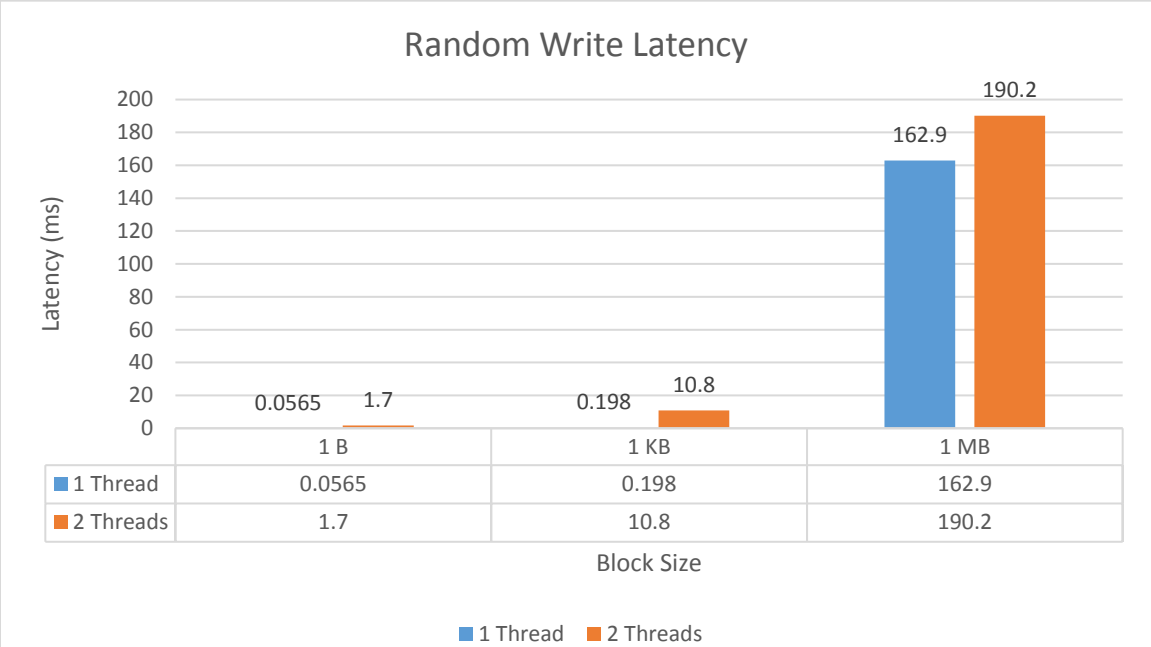
Random Write:

Data Size	Throughput (MB/s)	Latency (ms)	Threads
1 B	1.53	0.0565	1
1 KB	301.45	0.198	1
1 MB	963.18	162.9	1
1 B	2.086	180.7	2
1 KB	237.61	10.8	2
1 MB	971.39	190.2	2



Explanation:

Here y-axis represents the throughput in MB/s and x-axis represents the Data size written randomly. Throughput increases as size of data read increases. This is because throughput is calculated by data read/ time taken. So as data size increases throughput increases.
Also when running 2 threads concurrently, throughput obtained is similar.
Apart from this on comparing results with IOZONE random write similar results were obtained. Random write takes some time more than sequential write. This is because in random read a Math.random() function is defined which defines the position from which to read the data. Hence, random write takes more time compared to sequential write and hence lower throughput.



Explanation:

Here Y-axis represent Latency (ms) and X-axis represent data size that is being written randomly. Latency is the time taken for the disk to read and reply back. So as data increases the latency increases. Hence the graph makes sense. Also on comparing the results with that of IOZONE, implied the correctness of the results.

IOZONE:

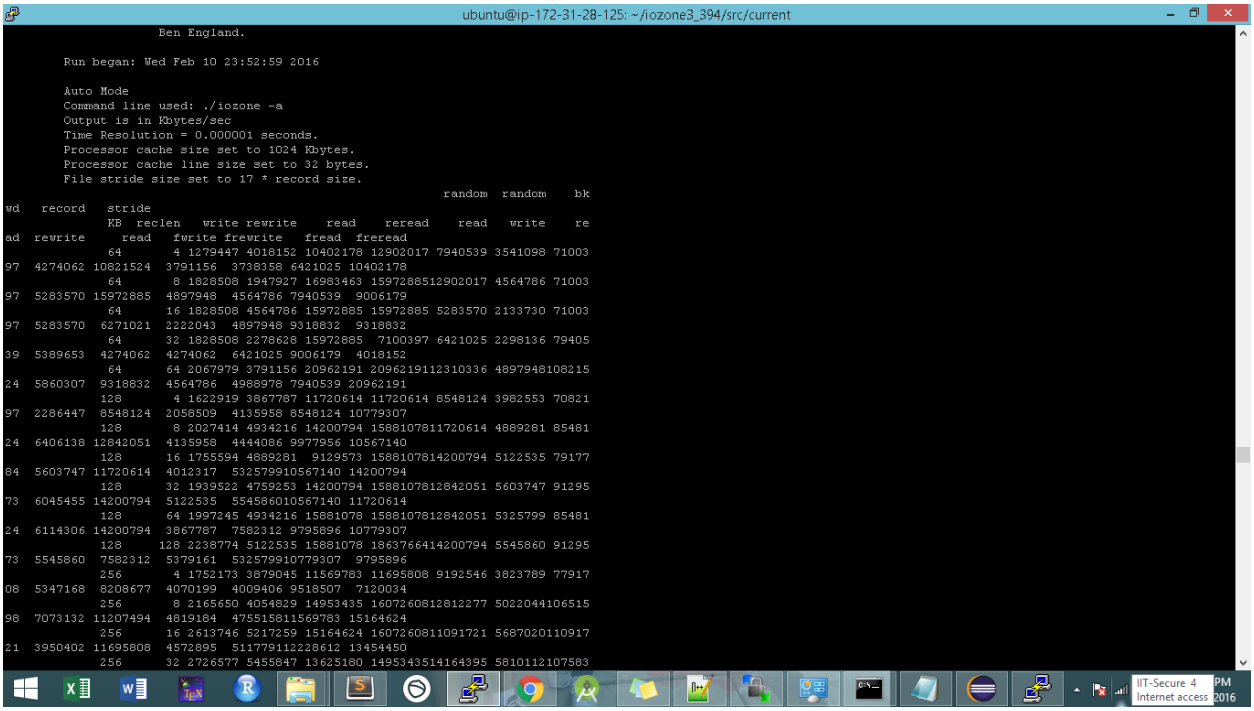
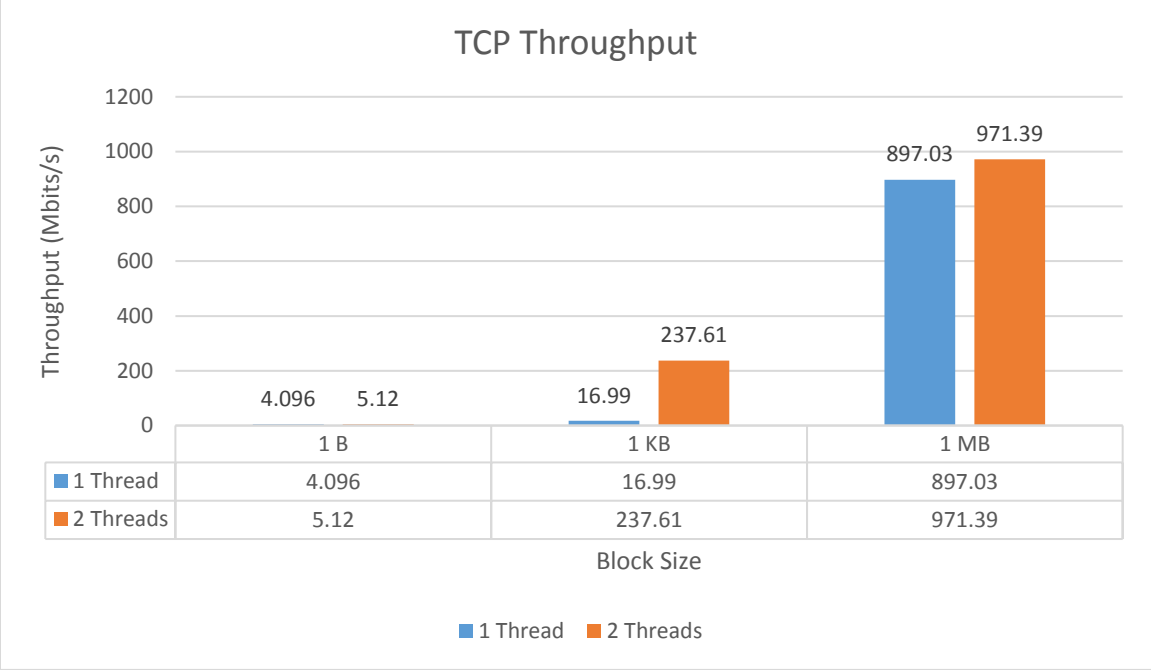


Figure shows the results obtained on running IOZONE on Amazon EC2 t2.micro instance.

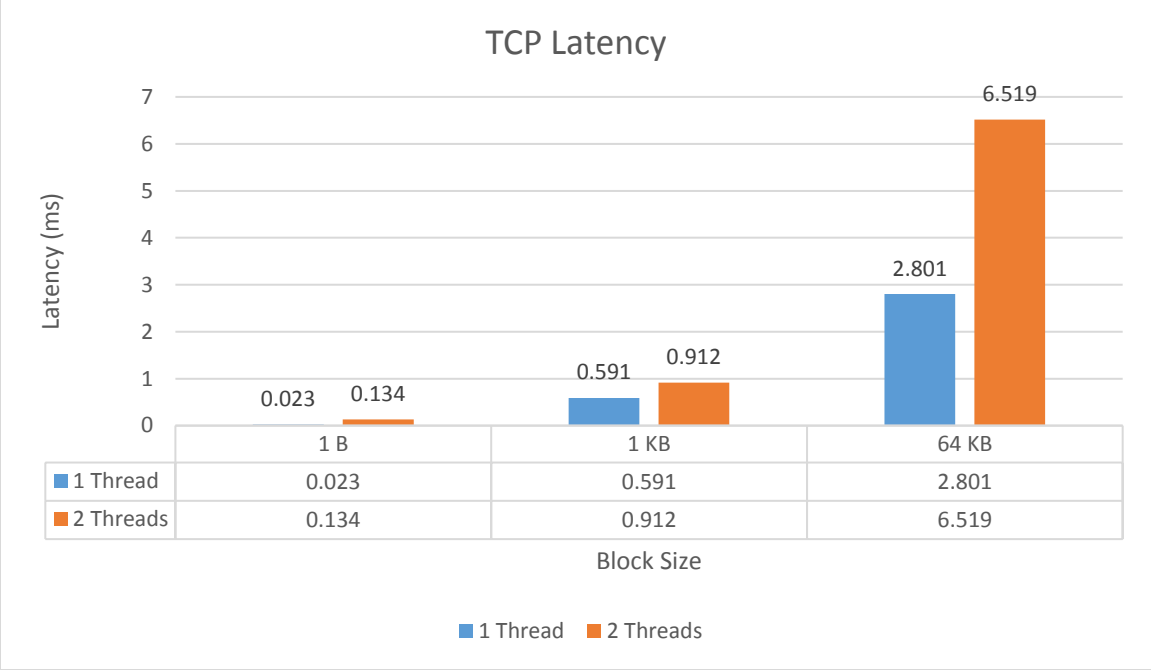
3. Network:
   TCP:

Data Size	Throughput (Mbits/s)	Latency (ms)	Threads
1 B	4.096	0.023	1
1 KB	16.99	0.591	1
64 KB	897.03	2.801	1
1 B	5.12	0.134	2
1 KB	237.61	0.912	2
64 KB	971.39	6.519	2



Explanation:

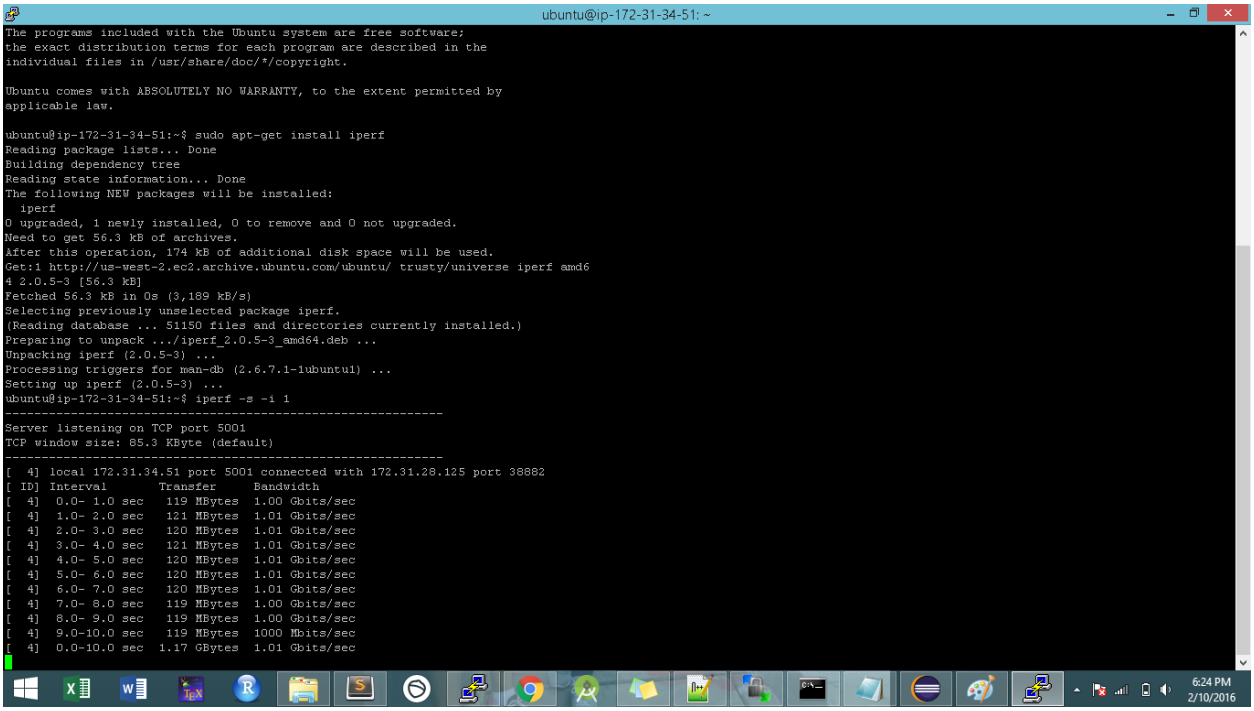
Here y-axis represents the throughput in Mbits/s and x-axis represents the Data size sent and received. Throughput increases as size of data increases. This is because throughput is calculated by data / time taken. So as data size increases throughput increases. Also when running 2 threads concurrently, throughput obtained is similar. Apart from this on comparing results with Iperf for tcp similar results were obtained. the data.



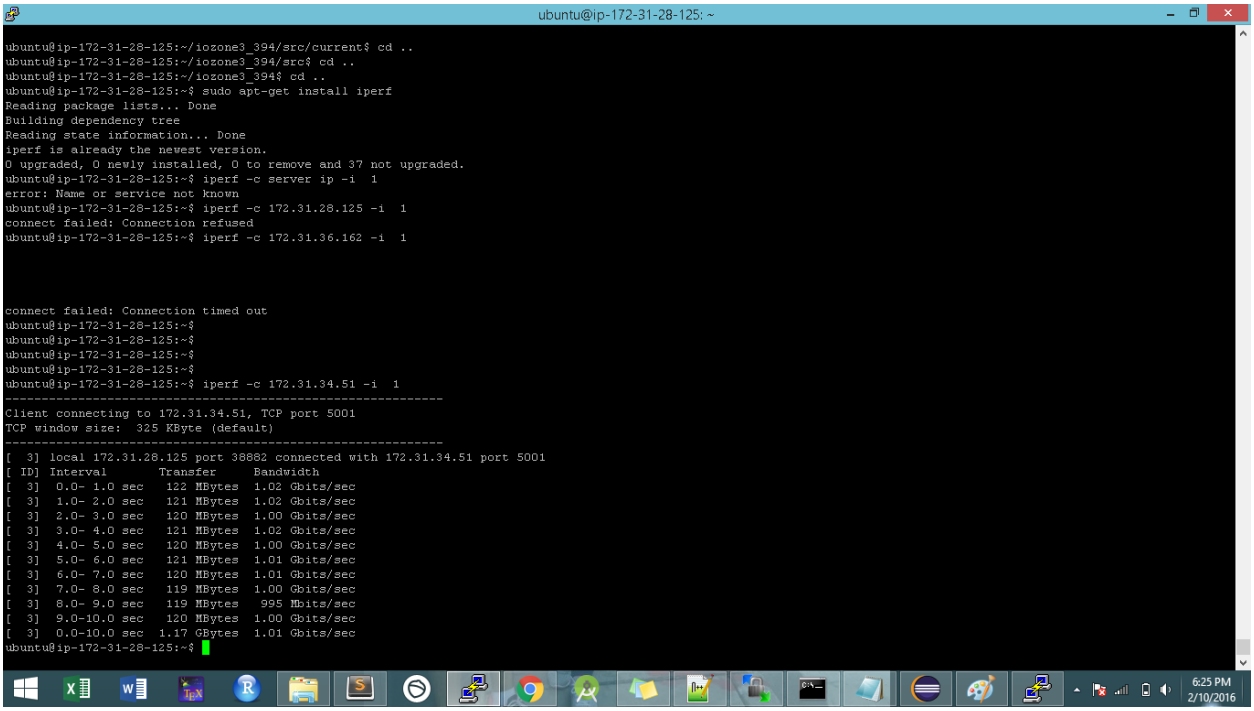
Explanation:

Here Y-axis represent Latency (ms) and X-axis represent data size that is being sent and received. Latency is the time taken for the network to read and reply back. So as data increases the latency increases. Hence the graph makes sense. Also on comparing the results with that of Iperf tcp, implied the correctness of the results

Iperf:



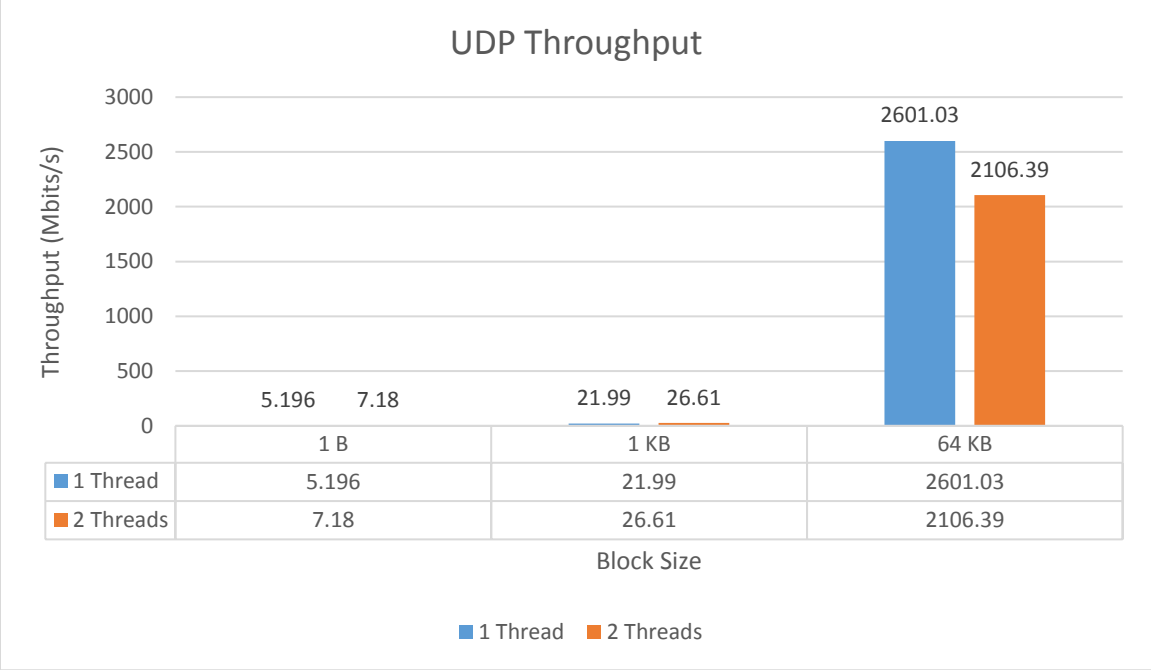
Iperf TCP Server



Iperf TCP Client

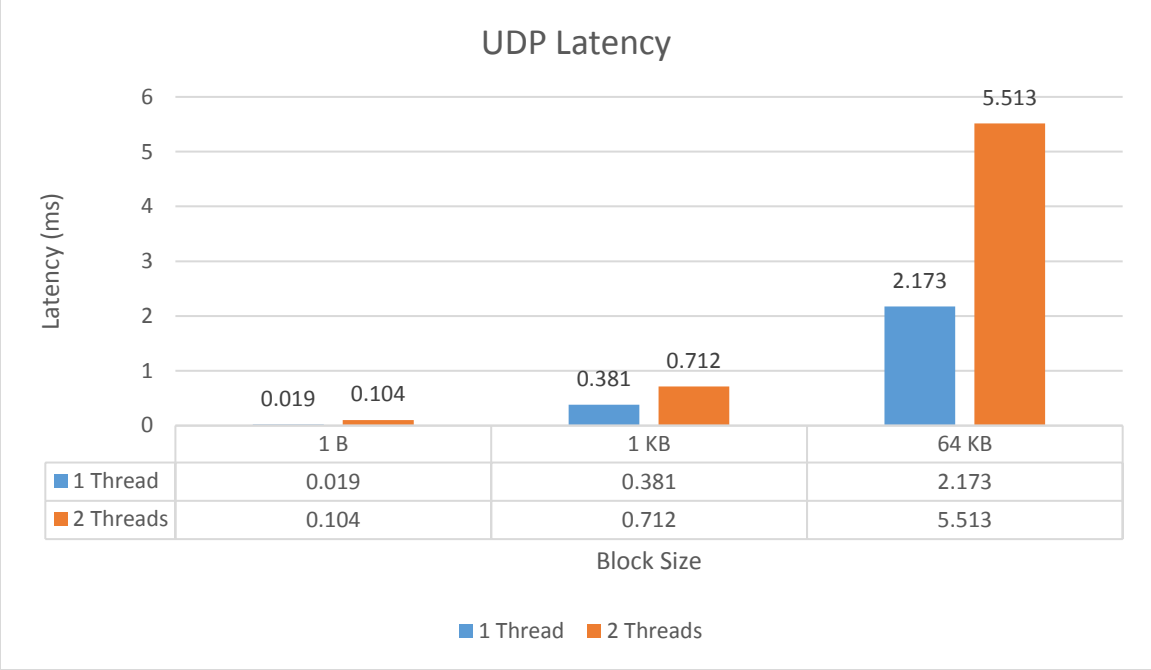
UDP:

Data Size	Throughput (Mbits/s)	Latency (ms)	Threads
1 B	5.196	0.019	1
1 KB	21.99	0.381	1
64 KB	2601.03	2.173	1
1 B	7.18	0.104	2
1 KB	26.61	0.712	2
64 KB	2106.39	5.513	2



Explanation:

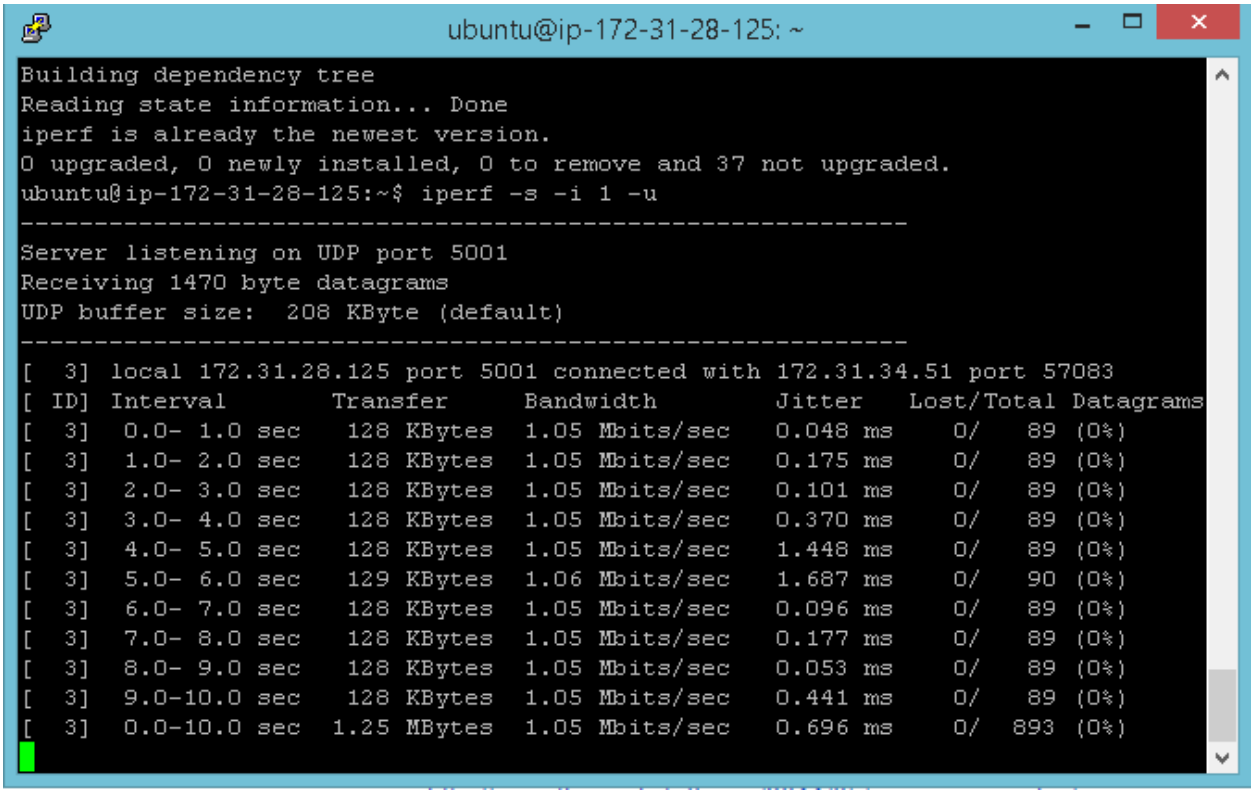
Here y-axis represents the throughput in Mbits/s and x-axis represents the Data size sent and received. Throughput increases as size of data increases. This is because throughput is calculated by data / time taken. So as data size increases throughput increases. Also when running 2 threads concurrently, throughput obtained is similar. Apart from this on comparing results with Iperf for udp similar results were obtained. the data.



Explanation:

Here Y-axis represent Latency (ms) and X-axis represent data size that is being sent and received. Latency is the time taken for the network to read and reply back. So as data increases the latency increases. Hence the graph makes sense. Also on comparing the results with that of Iperf udp, implied the correctness of the results

IPerf UDP Server:



IPerf udp client