CS-553- PA1

PERFORMANCE REPORT

Krishna Bharadwaj

A20398222

Contents

1 Introduction 3

1.1 Theoretical Values 3

2 Performance Analysis 4

2.1 CPU Benchmark 4
2.2 Memory Benchmark 5
2.3 Disk Benchmark 8
2.4 Network Benchmark 13

1 Introduction

This document gives a detailed performance analysis of the benchmark tools and comparison with tools such as linpack, pmbw, IOZone and Iperf.

1.1 Theoretical Values

a. CPU: (Assuming Haswell processors with 2 cores and 2 sockets per core)

   I. QP: 2 * 2.3 * 2 * 64 = 588.8 Gops

   II. HP: 2 * 2.3 * 2 * 32 = 294.4 GIOPS

   III. SP: 2 * 2.3 * 2 * 16 = 147.2 GIOPS

   IV. DP: 2 * 2.3 * 2 * 8 = 73.6 GFLOPS

b. Memory:

   I. Throughput: 2133,000,000 * 2 *64*2/(8*1000*1000*1000) = 68.25 GB/s

   II. Latency: 0.0104 micros

c. Disk:

Sequential Read (up to):	540MB/s
Sequential Write (up to):	410MB/s
Random Read (up to):	93000 IOPS
Random Write (up to):	43000 IOPS
Latency - Read:	500 µs
Latency - Write:	500 µs

d. Network:

   I. 56000 Gb/s

   II. 0.7 micros

2 Performance Analysis

This section should briefly introduce the system context and design, and discuss the background to the project.

2.1 CPU Benchmark

Workload	Concurrency	MyCPUBench Measured Ops/Sec (GigaOPS)	HPL Measured Ops/Sec (GigaOPS)	Theoretical Ops/Sec (GigaOPS)	MyCPUBench Efficiency (%)	HPL Efficiency (%)
QP	1	3.614	N/A	588.544	0.614057742	N/A
QP	2	7.18063	N/A	588.544	1.220066809	N/A
QP	4	7.147113	N/A	588.544	1.214371908	N/A
HP	1	4.105	N/A	294.272	1.394967921	N/A
HP	2	7.857641	N/A	294.272	2.670196621	N/A
HP	4	7.791421	N/A	294.272	2.64769363	N/A
SP	1	3.808	N/A	147.136	2.588081775	N/A
SP	2	7.493206	N/A	147.136	5.092707427	N/A
SP	4	7.642508	N/A	147.136	5.194179535	N/A
DP	1	3.761962	33.4721	73.568	5.113584711	45.49818
DP	2	7.057217	57.0881	73.568	9.592780829	77.59909
DP	4	7.127054	68.7276	73.568	9.68770933	93.42051

Table 1: CPU Performance Table

The above table summarizes the values obtained for the CPU speeds for each of the precisions measured in Giga Ops/s of my benchmark measured over 1t trillion operations as well the linpack values obtained for Double precision operations.

Analysis:

- As we can see, the speed increases as we increase the number of threads and peaks at 4 threads
- The efficiency though when compared with the theoretical value is at 10%.
- Efficiency of linpack compared to theoretical values is at 77% for 2 threads and 93% for 4 threads

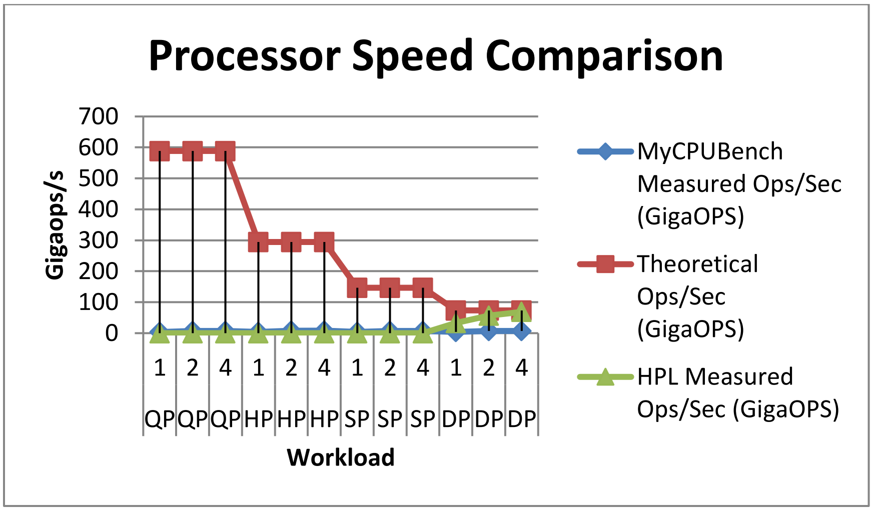

Chart 1

The above chart further visualizes the values obtained in the table above in comparison with the theoretical values. As we can see, our benchmark has an efficiency of 10% compared to the theoretical values and linpack as an efficiency of 77%.

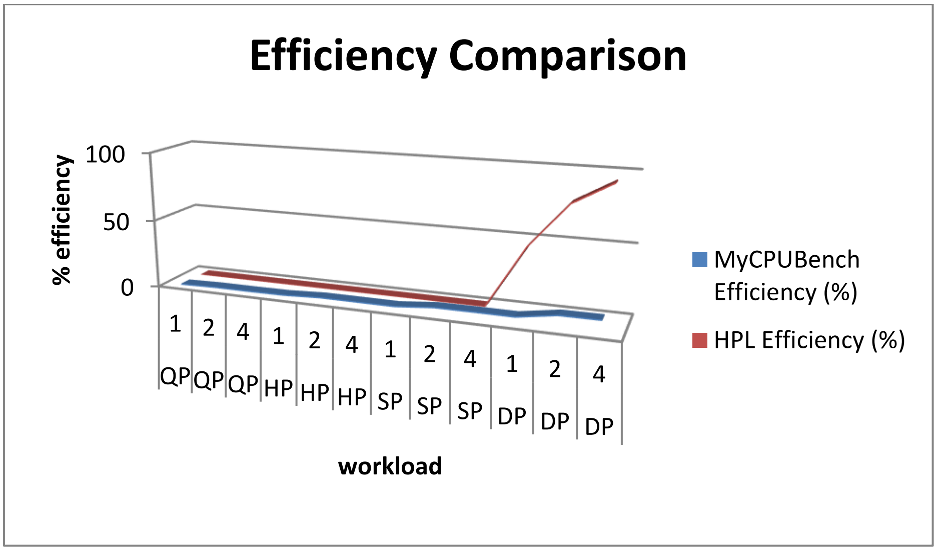

Chart 2

The above chart depicts the efficiency comparison of my benchmark when compared to Linpack.

2.2 Memory Benchmark

Implemented a strong scaling experiment to copy 1gb of memory from source to destination, iterated 100 times.

Compared the values obtained with the theoretical values as well as pmbw benchmark values.

Throughput:

Workload	Concurrency	Block Size	MyRAMBench Measured Throughput (GB/sec)	pmbw Measured Throughput (GB/sec)	Theoretical Throughput (GB/sec)	MyRAMBench Efficiency (%)	pmbw Efficiency (%)
RWS	1	1KB	4.348816	17.0934	68.256	6.371331	25.04307
RWS	1	1MB	4.400497	16.2943	68.256	6.447048	23.87233
RWS	1	10MB	6.320323	16.6644	68.256	9.259732	24.41456
RWS	2	1KB	14.940697	30.0942	68.256	21.88921	44.09019
RWS	2	1MB	16.118664	32.9942	68.256	23.61501	48.3389
RWS	2	10MB	20.595506	29.8839	68.256	30.17391	43.78209
RWS	4	1KB	31.646622	38.3133	68.256	46.3646	56.13177
RWS	4	1MB	29.830328	35.3751	68.256	43.7036	51.82709
RWS	4	10MB	43.673061	32.3993	68.256	63.98421	47.46733
RWR	1	1KB	1.801291	5.2105	68.256	2.639022	7.633761
RWR	1	1MB	4.398093	0.4836	68.256	6.443526	0.708509
RWR	1	10MB	7.411847	0.3138	68.256	10.85889	0.45974
RWR	2	1KB	2.150343	9.6309	68.256	3.150409	14.10997
RWR	2	1MB	8.898959	1.1654	68.256	13.03762	1.707396
RWR	2	10MB	14.778386	0.6789	68.256	21.65141	0.994638
RWR	4	1KB	2.086764	24.9305	68.256	3.057261	36.52499
RWR	4	1MB	8.881235	2.1791	68.256	13.01165	3.19254
RWR	4	10MB	14.828312	0.8075	68.256	21.72455	1.183046

Table 2: Memory Throughput Table

Analysis:

- **As we increase the block size from 1kb to 1MB, we see an increase in throughput.**
- **The throughput also increases as the number of threads increase**
- **The best performance is obtained at 4 threads**

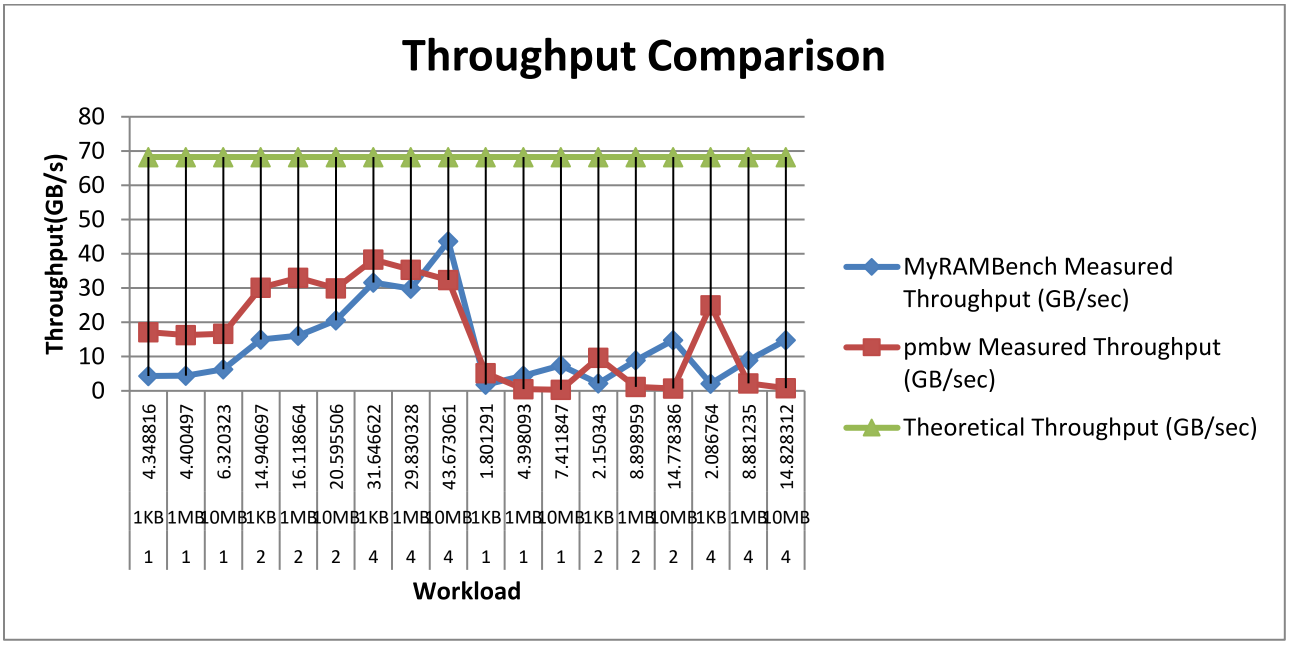

Chart 3

Chart 3 compares the throughputs obtained in my benchmark with pmbw and the theoretical values. As we can see, I am getting similar values in most of the experiments when compared with pmbw.

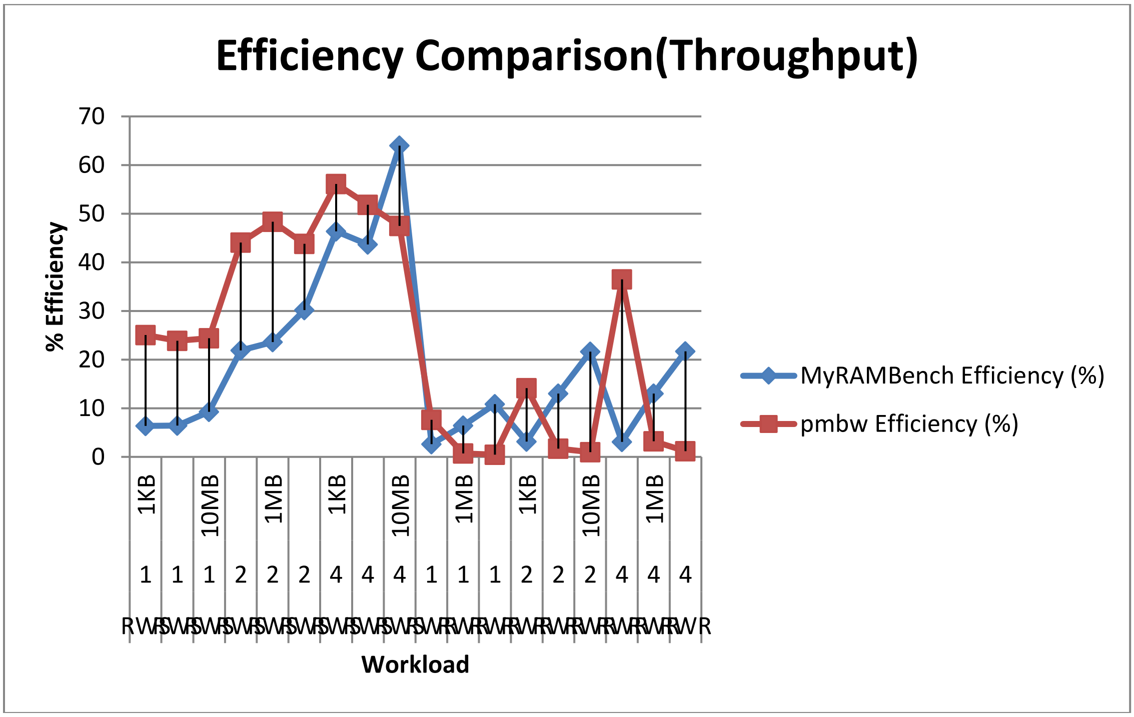

Chart 4

Chart 4 visualizes the efficiencies obtained from my benchmark and pmbw. It can be seen that in most places, there is a decrease of about 10% in efficiency in my benchmark when compared to pmbw.

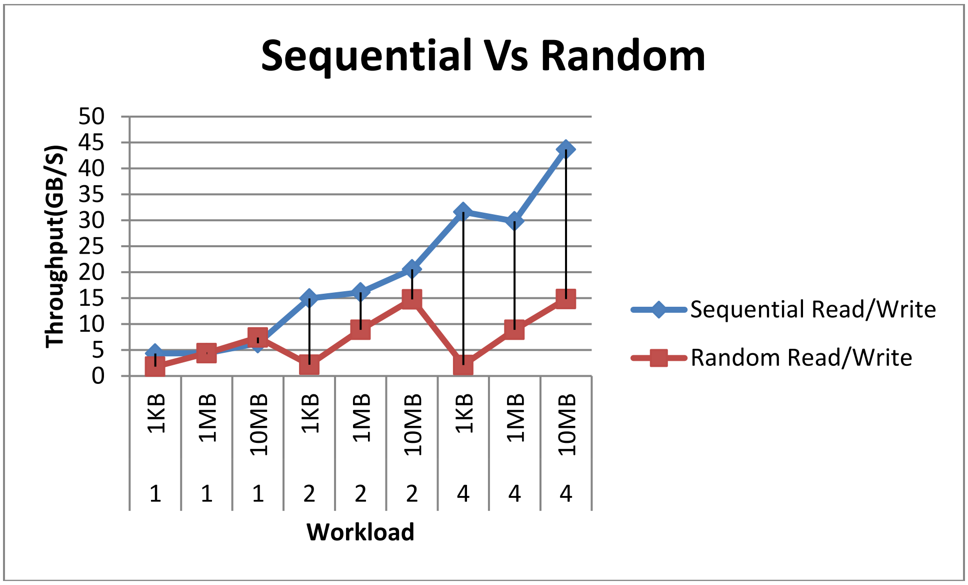

Chart 5

Analysis:

- **From the above chart, it can be seen that for lesser block sizes, sequential and random access patterns perform almost equally**
- **At higher block sizes, sequential access pattern has a much higher throughput when compared to random access pattern.**

Latency:

Workload	Concurrency	Block Size	MyRAMBench Measured Latency (microsec)	pmbw Measured Latency (microsec)	Theoretical Latency (microsec)	MyRAMBench Efficiency (%)	pmbw Efficiency (%)
RWS	1	1	0.013177	0.0004201	0.01406	93.71977	2.987909
RWS	2	1	0.006952	0.00029109	0.01406	49.44523	2.070341
RWS	4	1	0.004819	0.00022483	0.01406	34.27454	1.599075
RWR	1	1	0.125083	0.0015563	0.01406	889.6373	11.06899
RWR	2	1	0.292478	0.00082712	0.01406	2080.213	5.882788
RWR	4	1	0.302988	0.00030633	0.01406	2154.964	2.178734

Table 3

Analysis:

- The above table depicts the latency values obtained by my benchmark and pmbw benchmark and a comparison with the theoretical values
- It can be seen that the latency reduces with the increase in the numbr of threads.

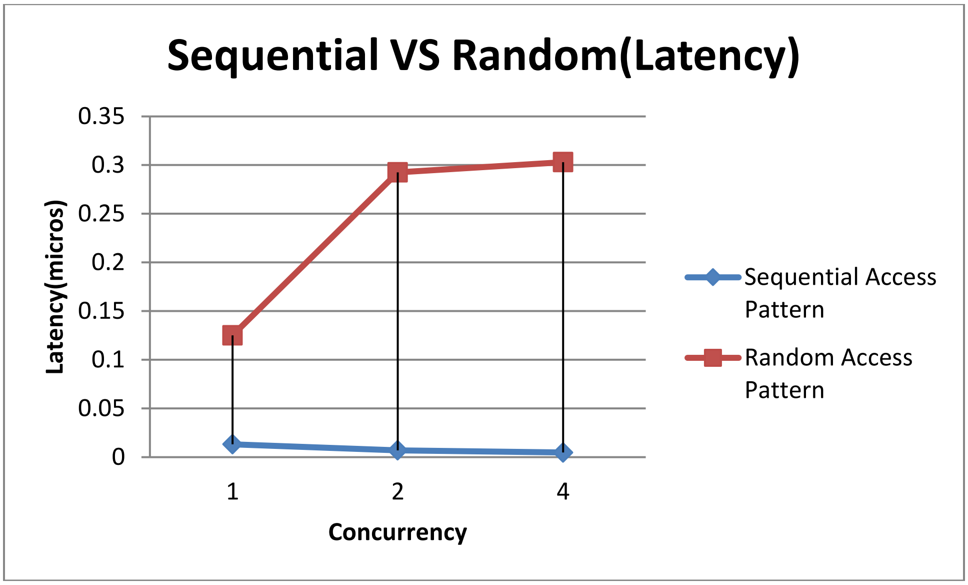

Chart 6

Chart 6 depicts the comparison of latency values obtained for random and sequential access patterns. It can be observed that random access pattern has a higher latency.

2.3 Disk Benchmark

Implemented a strong scaling experiment to read/write 10gb of memory from a file using various access patterns. Compared the values obtained with the theoretical values as well as pmbw benchmark values.

Throughput:

The below table depicts the throughput values obtained by my disk benchmark and IOZone and a comparison with the theoretical values.

Work-load	Con-currency	Block Size	MyDisk Bench Measured Throughput(MB/sec)	IOZoneMeasured Throughput (MB/sec)	Theoretical Throughput (MB/sec)	MyDiskBench Efficiency(%)	IOZone Efficiency(%)
RS	1	1MB	319.162394	432.62	540	59.10415	80.11481
RS	1	10MB	330.548687	380.17	540	61.21272	70.40185
RS	1	100MB	336.06262	364.78	540	62.23382	67.55185
RS	2	1MB	421.996204	496.23	1080	39.07372	45.94722
RS	2	10MB	475.603383	764.25	1080	44.03735	70.76389
RS	2	100MB	328.374656	843.89	1080	30.40506	78.13796
RS	4	1MB	1056.994432	1022.67	2160	48.93493	47.34583
RS	4	10MB	333.903077	1206.21	2160	15.45848	55.84306
RS	4	100MB	369.838725	1147.82	2160	17.12216	53.13981
WS	1	1MB	330.687228	346.58	410	80.65542	84.53171
WS	1	10MB	316.033056	338.91	410	77.08123	82.66098
WS	1	100MB	134.560701	265.84	410	32.81968	64.83902
WS	2	1MB	422.72529	423.83	820	51.55186	51.68659
WS	2	10MB	437.932963	360.24	820	53.40646	43.93171
WS	2	100MB	319.230157	331.57	820	38.93051	40.43537

WS	4	1MB	346.929101	364.28	1640	21.15421	22.2122
WS	4	10MB	252.970432	214.69	1640	15.42503	13.09085
WS	4	100MB	593.60114	445.37	1640	36.19519	27.15671
RR	1	1MB	308.371438	319.56	372	82.89555	85.90323
RR	1	10MB	333.448556	356.45	372	89.63671	95.81989
RR	1	100MB	94.65405	153.79	372	25.44464	41.3414
RR	2	1MB	228.703241	340.24	744	30.73968	45.73118
RR	2	10MB	634.55101	615.65	744	85.28911	82.74866
RR	2	100MB	107.795803	541.62	744	14.48868	72.79839
RR	4	1MB	421.384647	913.25	1488	28.31886	61.37433
RR	4	10MB	1151.136633	1257.61	1488	77.36133	84.5168
RR	4	100MB	1205.944941	961.53	1488	81.04469	64.61895
WR	1	1MB	97.93379	127.43	172	56.93825	74.08721
WR	1	10MB	105.931273	137.2	172	61.58795	79.76744
WR	1	100MB	120.48215	143.25	172	70.04776	83.28488
WR	2	1MB	186.668627	273.58	344	54.26414	79.52907
WR	2	10MB	216.876828	294.26	344	63.04559	85.5407
WR	2	100MB	248.747672	313.52	344	72.31037	91.13953
WR	4	1MB	387.373588	423.18	688	56.3043	61.50872
WR	4	10MB	406.836626	467.29	688	59.13323	67.92006
WR	4	100MB	555.204978	513.72	688	80.6984	74.6686

Table 4

Analysis:

- **We see a steady increase in throughput as we increase the number of threads and block size. This is in concurrence with the values obtained from IOZone**
- **The value peaks at 4 threads and 100MB block size.**
- **The performance of sequential reads/writes is better when compared with random reads/writes.**

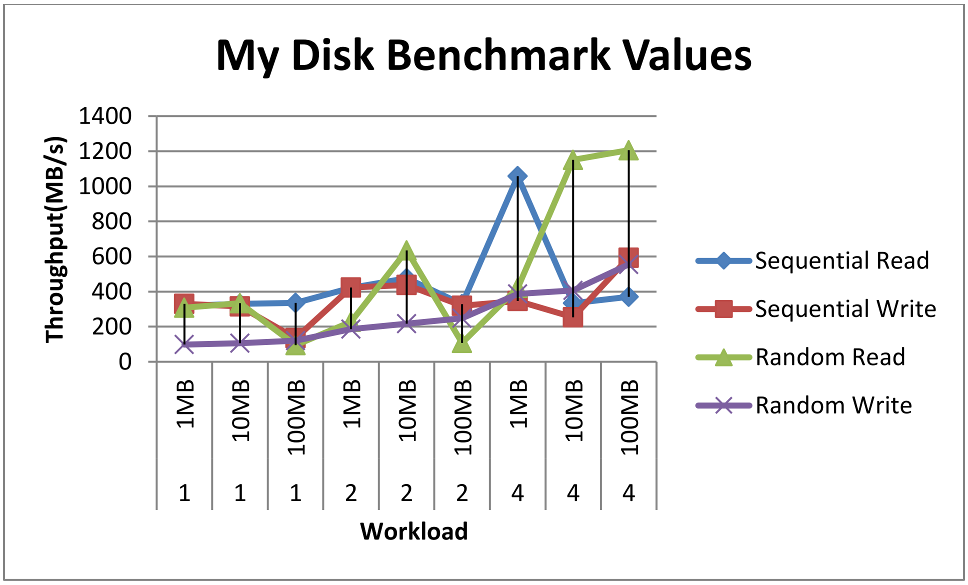

Chart 7

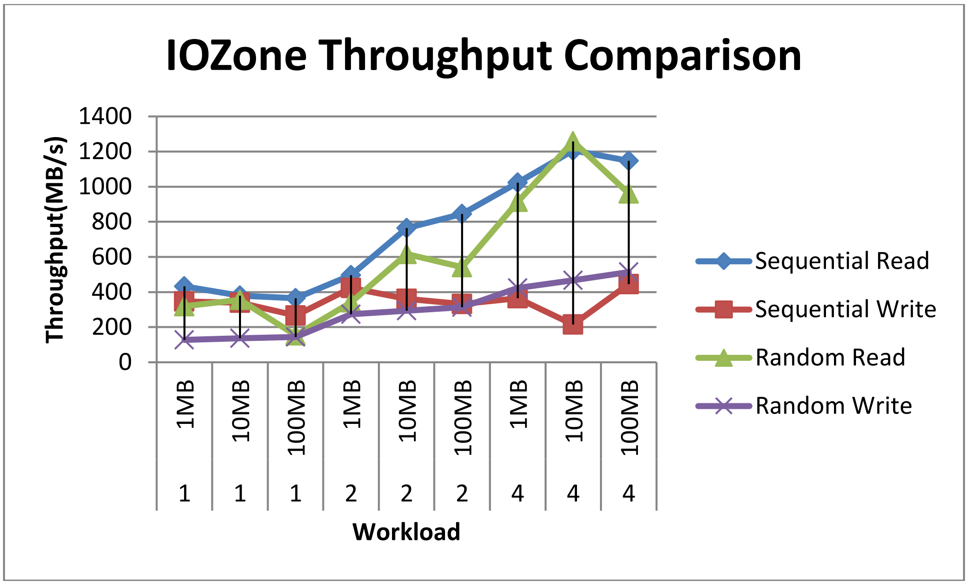

Chart 8

The charts above further visualize the throughputs obtained from my disk benchmark and IOZone for each of the 4 operations.

Latency:

Work-load	Con-currency	Block Size	MyDiskBench Measured Latency(ms)	IOZone Measured Latency(ms)	Theoretical Latency(ms)	MyDiskBench Efficiency(%)	IOZone Efficiency(%)
RR	1	1KB	0.005515	0.05	0.5	1.103	10
RR	2	1KB	0.005222	0.07	0.5	1.0444	14
RR	4	1KB	0.004366	0.032	0.5	0.8732	6.4
RR	8	1KB	0.004543	0.035	0.5	0.9086	7
RR	16	1KB	0.005619	0.015	0.5	1.1238	3
RR	32	1KB	0.00394	0.01	0.5	0.788	2
RR	64	1KB	0.004963	0.002	0.5	0.9926	0.4
RR	128	1KB	0.012807	0.045	0.5	2.5614	9
WR	1	1KB	0.046454	0.3	0.5	9.2908	60
WR	2	1KB	0.040017	0.02	0.5	8.0034	4
WR	4	1KB	0.035688	0.045	0.5	7.1376	9
WR	8	1KB	0.033028	0.027	0.5	6.6056	5.4
WR	16	1KB	0.033037	0.025	0.5	6.6074	5
WR	32	1KB	0.038176	0.013	0.5	7.6352	2.6
WR	64	1KB	0.018741	0.007	0.5	3.7482	1.4
WR	128	1KB	0.010193	0.003	0.5	2.0386	0.6

Table 5

Analysis:

- We see that the latency decreases as we increase the number of threads until we reach 32 threads. As we increase the threads after 32, we see an increase in latency
- Latency of Writes seems to be higher than Reads.
- Write random latencies decrease monotonically as we increase the number of threads.

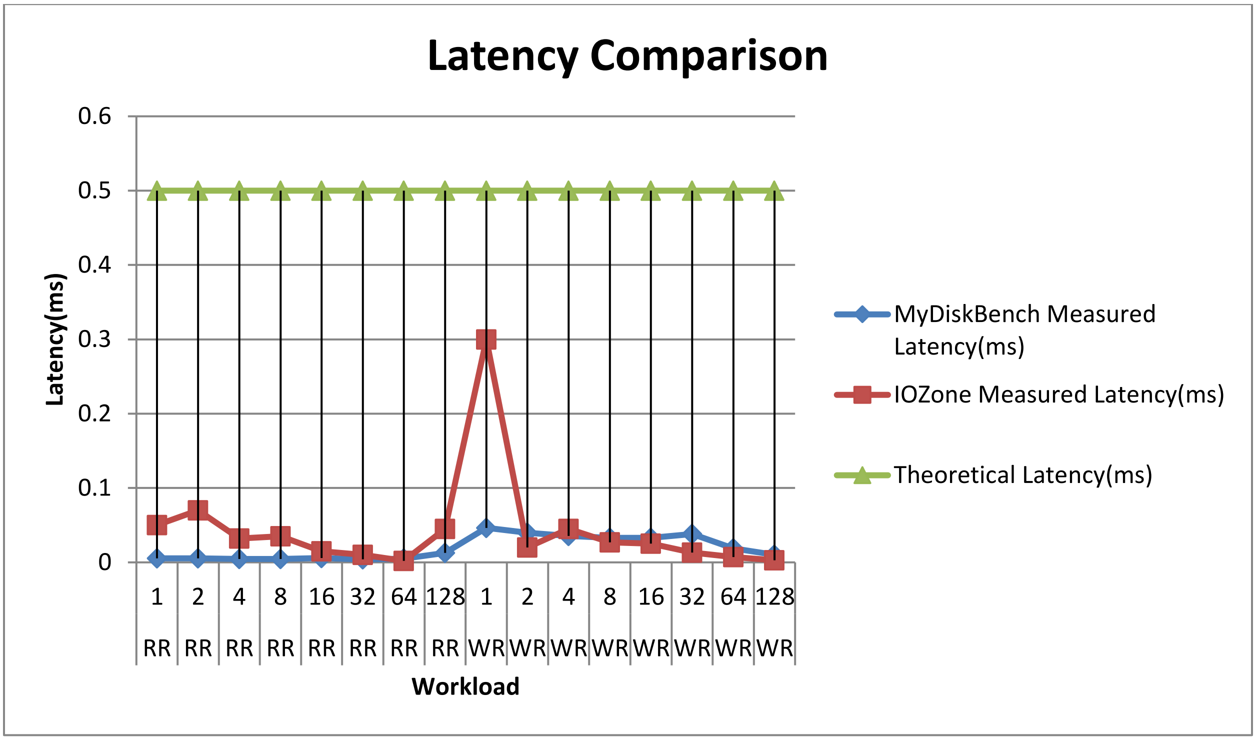

Chart 9

Chart 9 depicts the latency values obtained in my benchmark, IOZone and Theoretical values.

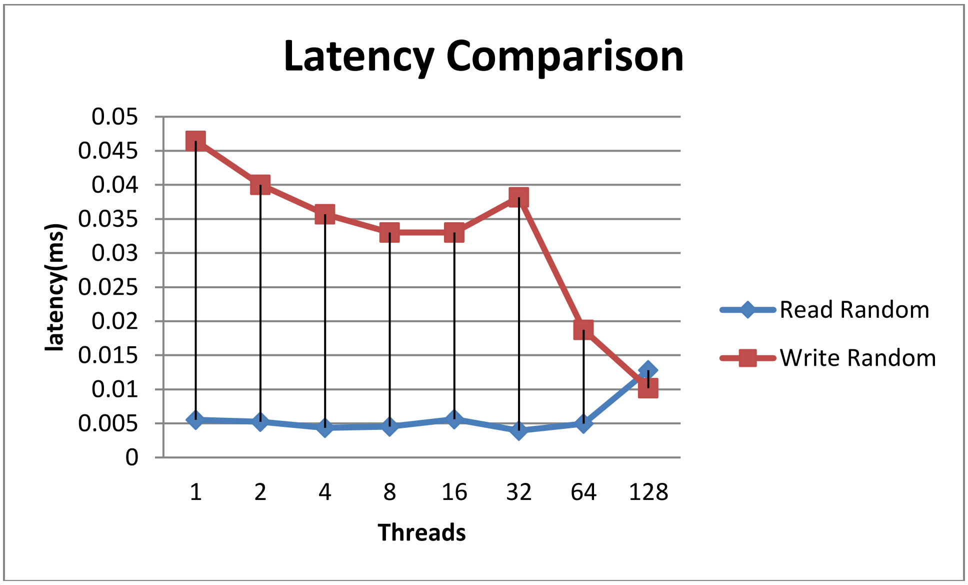

Chart 10

Work-load	Con-currency	Block Size	MyDisk Bench Measured IOPS	IOZone Measured IOPS	Theoretical IOPS	MyDiskBench Efficiency(%)	IOZone Efficiency(%)
RR	1	1KB	36265	42681	93000	38.99462	45.89355
RR	2	1KB	38300	51024	93000	41.1828	54.86452
RR	4	1KB	45809	59418	93000	49.25699	63.89032
RR	8	1KB	44024	56281	93000	47.33763	60.5172
RR	16	1KB	35594	54928	93000	38.27312	59.06237
RR	32	1KB	50761	57597	93000	54.58172	61.93226
RR	64	1KB	40298	64876	93000	43.33118	69.75914
RR	128	1KB	15616	68126	93000	16.7914	73.25376
WR	1	1KB	4305	16475	43000	10.01163	38.31395
WR	2	1KB	4998	21541	43000	11.62326	50.09535
WR	4	1KB	5604	23012	43000	13.03256	53.51628
WR	8	1KB	6055	29412	43000	14.0814	68.4
WR	16	1KB	6054	31602	43000	14.07907	73.49302
WR	32	1KB	5239	33572	43000	12.18372	78.07442
WR	64	1KB	10672	35934	43000	24.8186	83.56744
WR	128	1KB	19621	36835	43000	45.63023	85.66279

Table 6

Table 6 summarizes the IOPS values obtained for my disk benchmark and IOZone and a comparison with the theoretical values.

Analysis:

- Similar to table 5, we see that the IOPS steadily increases for read random as we increase the number of threads until 32 and then we see a decrease as we go all the way to 128 threads. This is depicted in chart 11.
- Write random IOPS increases as we increase the number of threads

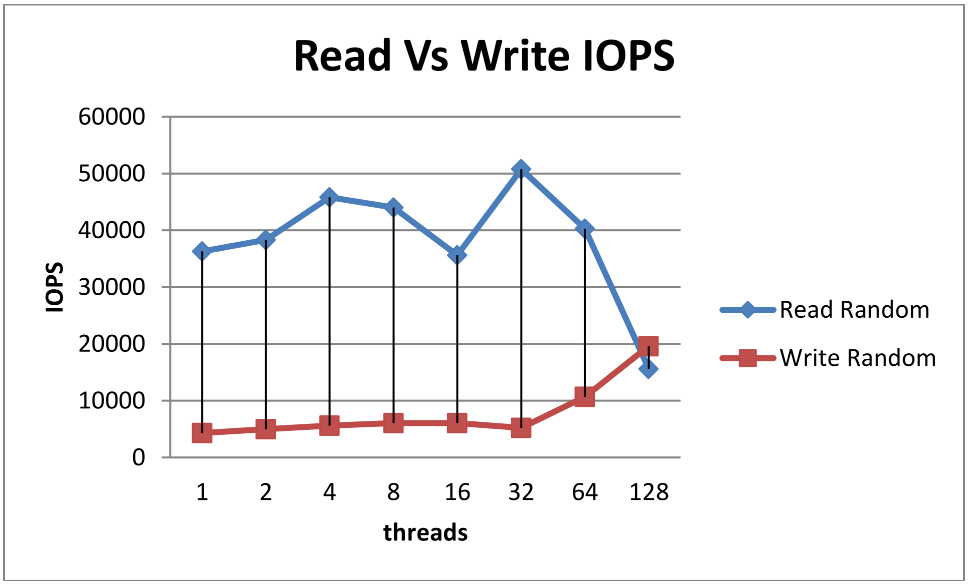

Chart 11

2.4 Network Benchmark

Concurrency	Block Size	MyNETBenchMeasured Throughput (Mb/sec)	iperf Measured Throughput(Mb/sec)	Theoretical Throughput (Mb/sec)	MyNETBench Efficiency (%)	iperf Efficiency (%)
1	1KB	2047.342451	1723.34	56000	3.655968663	3.0773929
1	32KB	1888.372209	2567.56	56000	3.37209323	4.5849286
2	1KB	2043.993613	2345.47	56000	3.649988595	4.1883393
2	32KB	6627.132173	3456.61	56000	11.83416459	6.1725179
4	1KB	4105.147744	6845.46	56000	7.330620971	12.224036
4	32KB	9119.878529	7973.23	56000	16.28549737	14.237911
8	1KB	6706.657665	5134.44	56000	11.9761744	9.1686429
8	32KB	9242.065422	8302.56	56000	16.50368825	14.826
1	1KB	301.44555	2972.21	56000	0.538295625	5.3075179
1	32KB	1017.488	2874.64	56000	1.816942857	5.1332857
2	1KB	350.45656	3572.1	56000	0.625815286	6.37875
2	32KB	1636.758422	4284.3	56000	2.922782896	7.6505357
4	1KB	371.279332	3560.99	56000	0.662998807	6.3589107
4	32KB	1917.812308	5430.23	56000	3.424664836	9.6968393
8	1KB	445.24422	5793.46	56000	0.795078964	10.345464
8	32KB	1670.649274	5189.09	56000	2.983302275	9.2662321

Table 7

Analysis:

- **We see that the throughput increases with the increase in the number of threads**
- **Throughput for 32KB block size is higher than 1KB**
- **TCP seems to be performing better than UDP in My Network Benchmark**

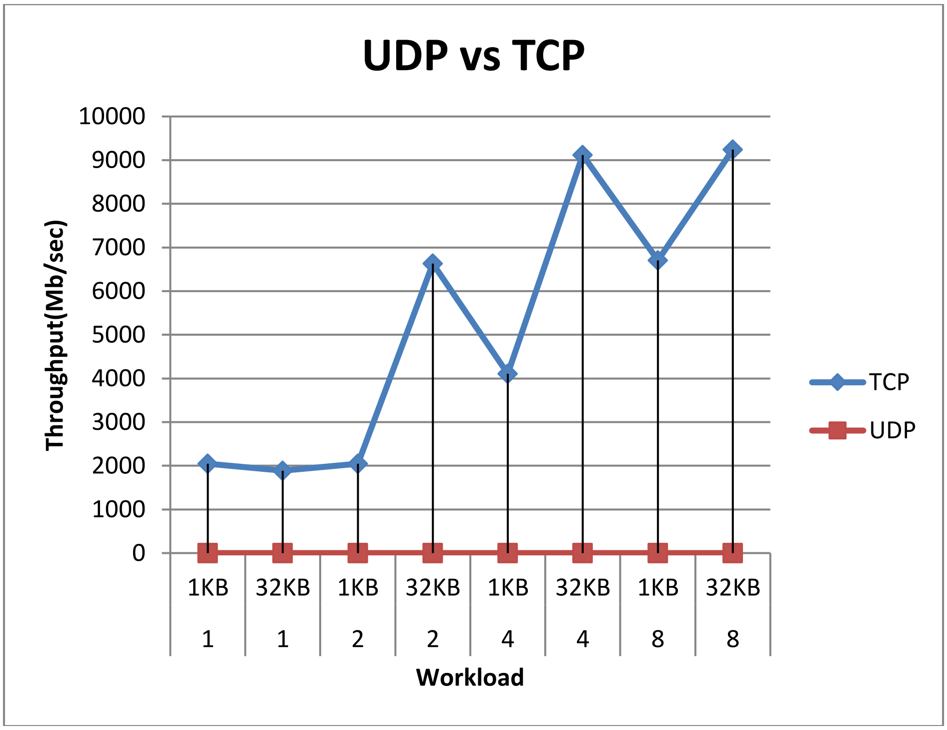

Chart 12

Chart 12 depicts the performance of TCP compared to UDP.

Latency:

Protocol	Con-currency	Message Size	MyNETBench Measured Latency(ms)	ping Measured Latency (ms)	Theoretical Latency (ms)	MyNETBench Efficiency(%)	iperf Efficicency(%)
TCP	1	1B	0.194956	1.82	0.0007	0.359055377	260000
TCP	2	1B	0.093806	1.82	0.0007	0.746220924	260000
TCP	4	1B	0.052827	1.82	0.0007	1.325079978	260000
TCP	8	1B	0.036502	1.82	0.0007	1.91770314	260000
UDP	1	1B	0.13358	1.82	0.0007	0.524030543	260000
UDP	2	1B	0.068405	1.82	0.0007	1.023317009	260000
UDP	4	1B	0.036664	1.82	0.0007	1.909229762	260000
UDP	8	1B	0.025087	1.82	0.0007	2.790289792	260000

Analysis:

- **Latency decreases with the number of threads**
- **UDP latency is comparatively lesser**

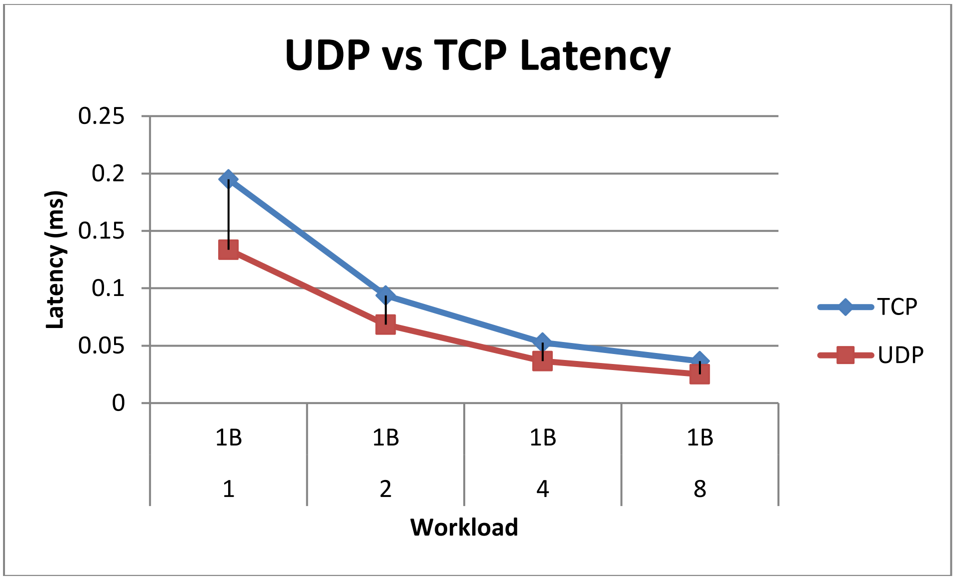

Chart 13