BENCHMARKING STORAGE

Abstract

Benchmarking storage over chameleon cloud with varying parameters of threads, record size and type of operation

Mohit Aggarwal
Siddhant Jain

AIM:

Develop an understanding of benchmarking storage systems using a 10GB file by varying the following parameters:

1. Record Size – 4KB, 64KB, 1MB, 16MB
2. Type of Operation – Sequential Read/ Write & Random Read/Write
3. No. of threads – 1,2,4,8,12,24 and48

Code Information:

The project was developed in C and compiled with gcc. Following functions or libraries were used to perform the required operations:

1. pthread – It was used to perform multithreading for various threads
2. read() – It was used to read data form the file
3. write() – It was used to write data to the file
4. fseek() – It was used to reach a certain position in a file
5. gettimeofday() – It was used to get the running time for a particular experiment to provide millisecond level accuracy

Design Decisions:

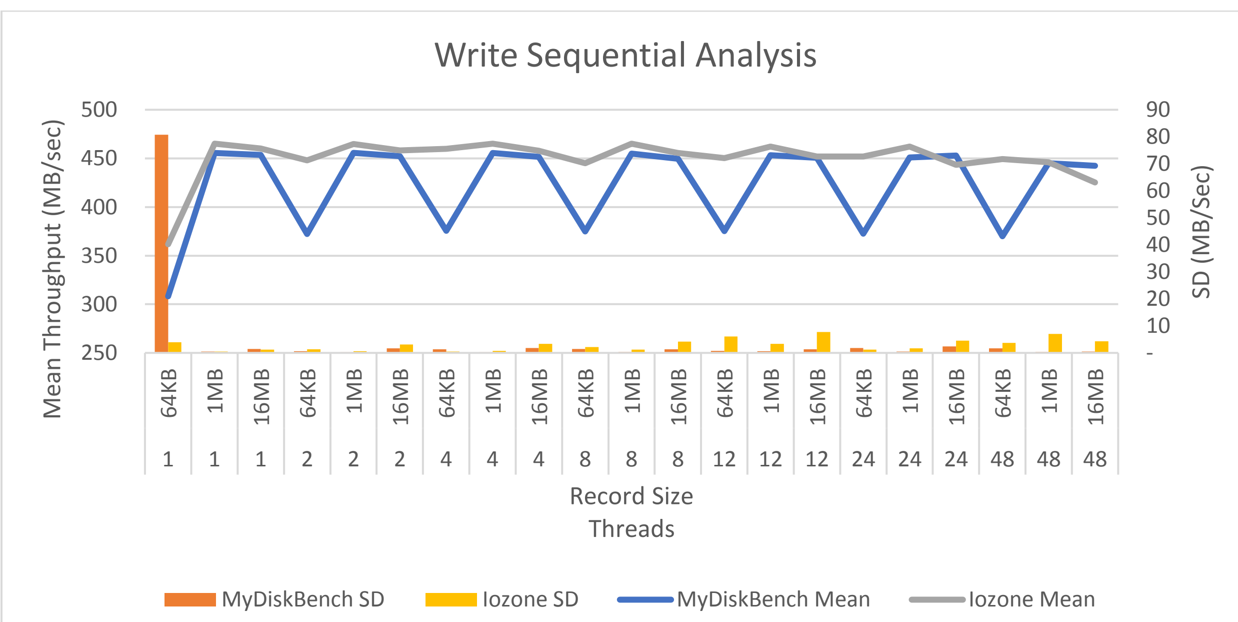
It was needed to made sure that the data is being stored or read form the disk and not from the memory and for this O_DIRECT flag was used while opening the file to make the data is read from the disk.

For random read and write, the rand() function to used to access the file at a random block, one drawback of this is that it's not necessary that the whole file will be read or written. It is possible that the rand() function generates the same number multiple times and the program ends up reading or writing the place in the file again and again.

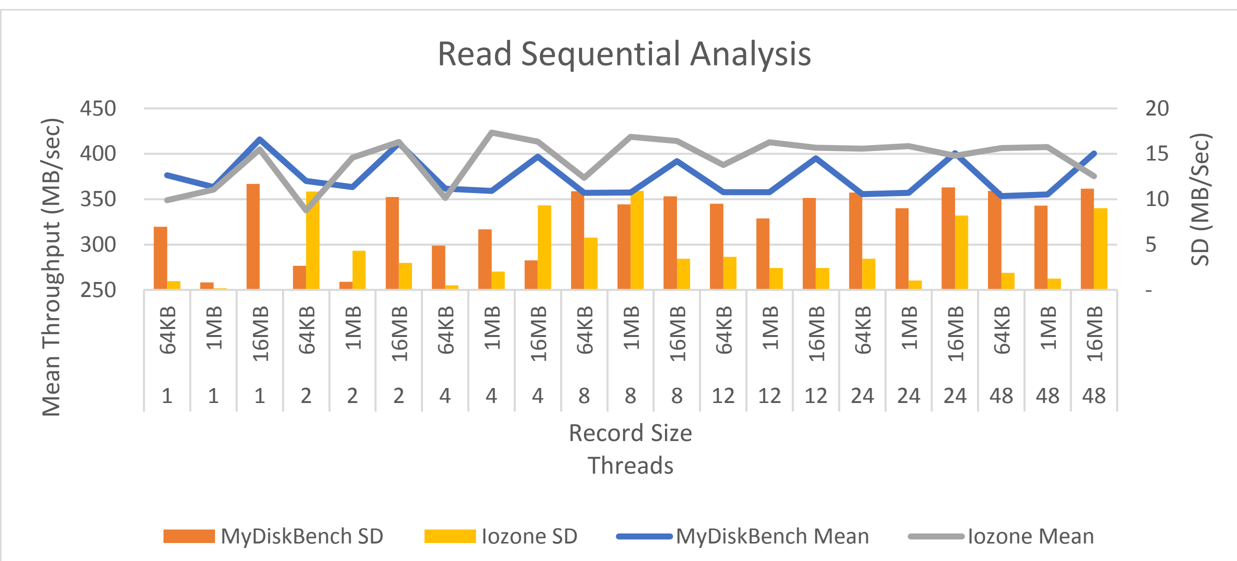
The code is completely automated to have minimal intervention from the user. The only task that the person who is running it is to make sure that the proper folder structure is present.

Results:

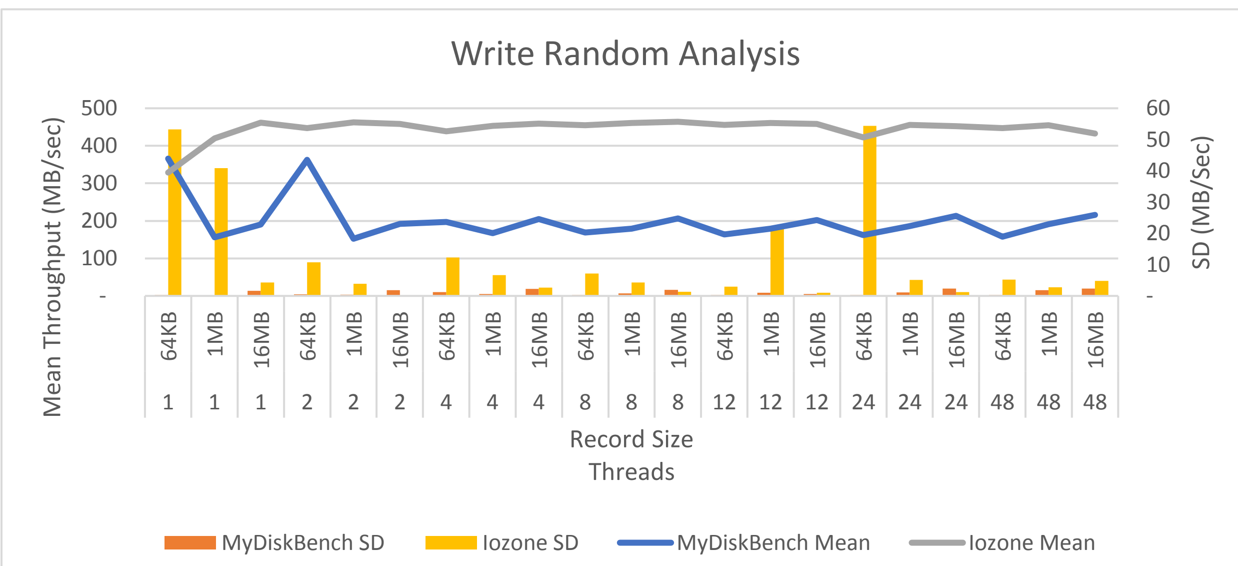
Workload	Concurrency	Record Size	MyDiskBench - Throughput (MB/sec)	IOZONE - Throughput (MB/sec)	Theoretical	MyDiskBench Efficiency	IOZone Efficiency
WS	1	64KB	308	362	485	63.54%	74.58%
WS	1	1MB	455	465	485	93.90%	95.90%
WS	1	16MB	454	460	485	93.55%	94.90%
WS	2	64KB	372	448	485	76.73%	92.39%
WS	2	1MB	456	465	485	93.95%	95.80%
WS	2	16MB	452	458	485	93.22%	94.49%
WS	4	64KB	375	460	485	77.42%	94.84%
WS	4	1MB	456	465	485	93.92%	95.89%
WS	4	16MB	452	458	485	93.12%	94.39%
WS	8	64KB	375	445	485	77.32%	91.76%
WS	8	1MB	455	465	485	93.80%	95.87%
WS	8	16MB	450	456	485	92.73%	93.93%
WS	12	64KB	375	450	485	77.36%	92.82%
WS	12	1MB	453	462	485	93.46%	95.29%
WS	12	16MB	451	452	485	92.91%	93.17%
WS	24	64KB	373	452	485	76.85%	93.16%
WS	24	1MB	451	462	485	92.96%	95.28%
WS	24	16MB	453	443	485	93.40%	91.42%
WS	48	64KB	370	449	485	76.29%	92.62%
WS	48	1MB	445	446	485	91.77%	91.98%
WS	48	16MB	442	425	485	91.20%	87.68%



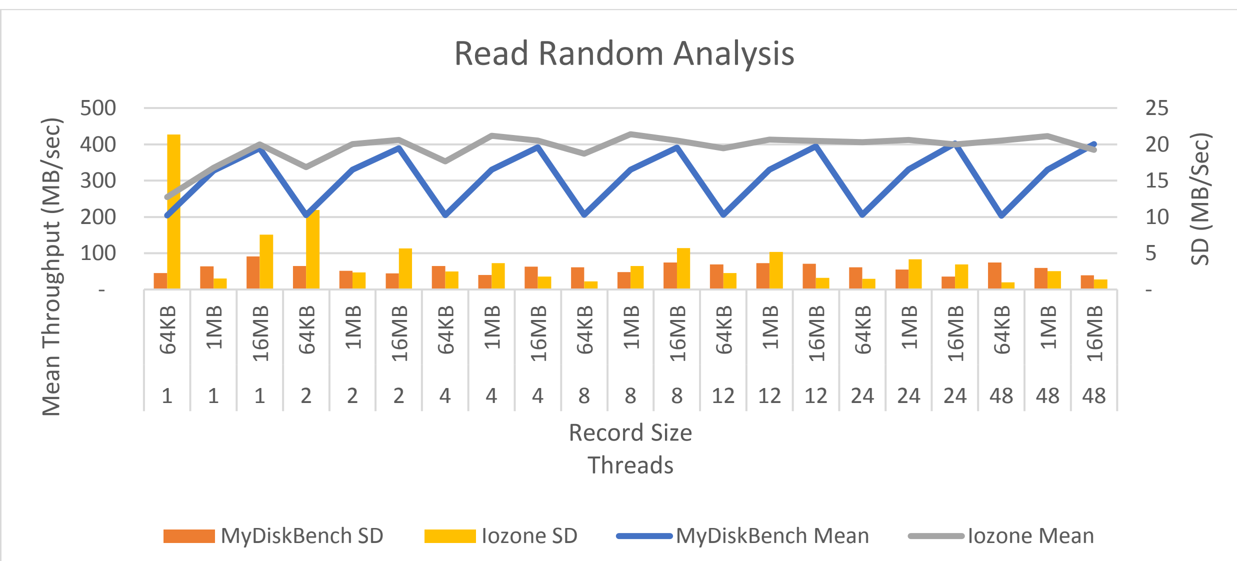
Workload	Concurrency	Record Size	MyDiskBench - Throughput (MB/sec)	IOZONE - Throughput (MB/sec)	Theoretical	MyDiskBench Efficiency	IOZone Efficiency
RS	1	64KB	376	349	485	77.58%	71.89%
RS	1	1MB	363	361	485	74.90%	74.36%
RS	1	16MB	416	405	485	85.74%	83.50%
RS	2	64KB	370	338	485	76.26%	69.62%
RS	2	1MB	363	396	485	74.89%	81.60%
RS	2	16MB	412	413	485	84.90%	85.19%
RS	4	64KB	361	351	485	74.53%	72.41%
RS	4	1MB	359	423	485	74.06%	87.29%
RS	4	16MB	397	413	485	81.84%	85.21%
RS	8	64KB	357	374	485	73.60%	77.05%
RS	8	1MB	357	419	485	73.68%	86.32%
RS	8	16MB	392	414	485	80.77%	85.36%
RS	12	64KB	358	388	485	73.73%	79.90%
RS	12	1MB	358	413	485	73.72%	85.06%
RS	12	16MB	395	407	485	81.44%	83.82%
RS	24	64KB	356	405	485	73.30%	83.61%
RS	24	1MB	357	409	485	73.59%	84.24%
RS	24	16MB	401	398	485	82.62%	82.01%
RS	48	64KB	353	406	485	72.87%	83.79%
RS	48	1MB	355	408	485	73.25%	84.03%
RS	48	16MB	400	375	485	82.57%	77.39%



Workload	Concurrency	Record Size	MyDiskBench - Throughput (MB/sec)	IOZONE - Throughput (MB/sec)	Theoretical	MyDiskBench Efficiency	IOZone Efficiency
WR	1	64KB	366	329	485	75.44%	67.79%
WR	1	1MB	156	420	485	32.15%	86.61%
WR	1	16MB	190	462	485	39.18%	95.20%
WR	2	64KB	363	447	485	74.79%	92.20%
WR	2	1MB	152	463	485	31.42%	95.43%
WR	2	16MB	192	459	485	39.63%	94.56%
WR	4	64KB	197	438	485	40.70%	90.35%
WR	4	1MB	168	453	485	34.56%	93.49%
WR	4	16MB	205	459	485	42.28%	94.67%
WR	8	64KB	169	455	485	34.82%	93.71%
WR	8	1MB	179	460	485	37.01%	94.93%
WR	8	16MB	207	464	485	42.61%	95.69%
WR	12	64KB	164	455	485	33.86%	93.91%
WR	12	1MB	179	460	485	36.98%	94.93%
WR	12	16MB	202	458	485	41.63%	94.52%
WR	24	64KB	162	422	485	33.35%	87.09%
WR	24	1MB	186	455	485	38.34%	93.88%
WR	24	16MB	214	452	485	44.04%	93.18%
WR	48	64KB	158	447	485	32.58%	92.21%
WR	48	1MB	191	455	485	39.46%	93.83%
WR	48	16MB	216	433	485	44.45%	89.29%

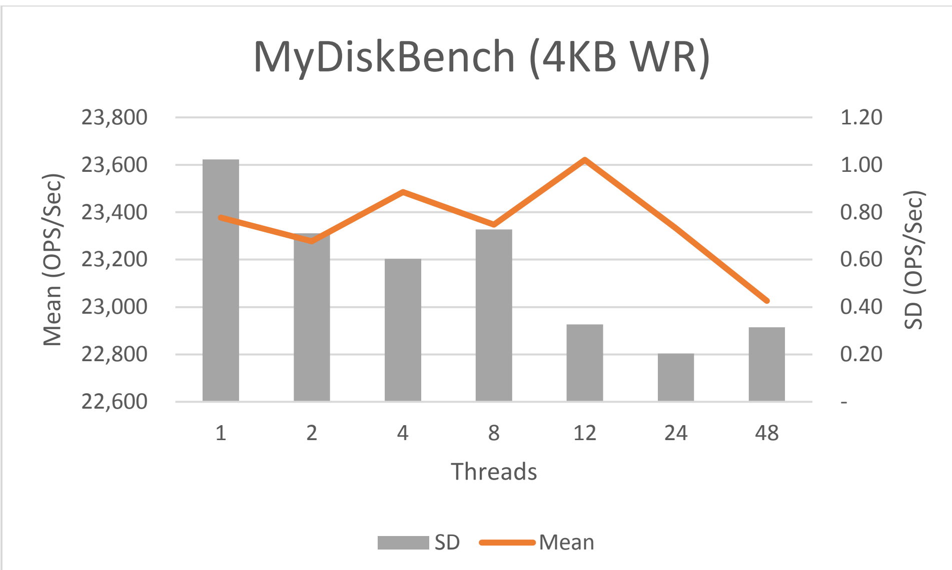


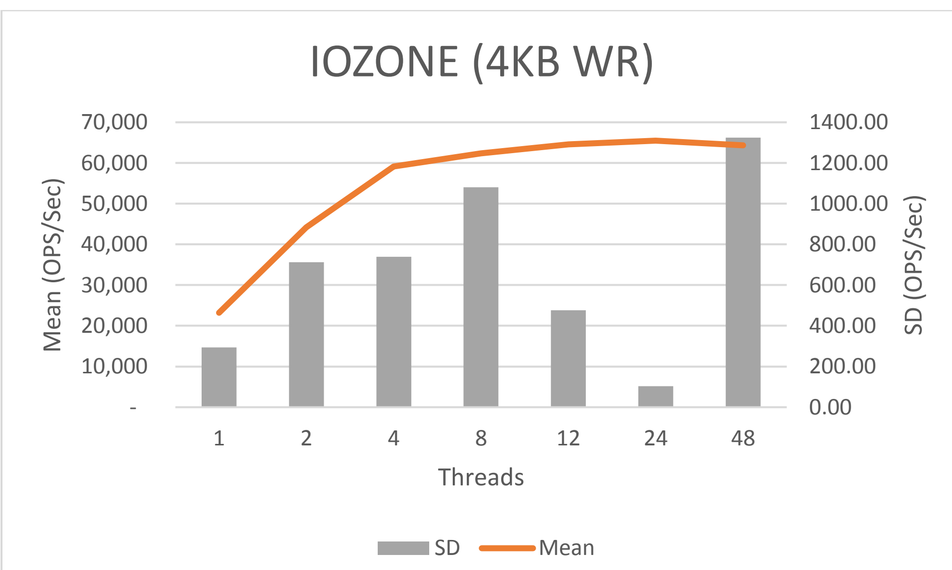
Workload	Concurrency	Record Size	MyDiskBench - Throughput (MB/sec)	IOZONE - Throughput (MB/sec)	Theoretical	MyDiskBench Efficiency	IOZone Efficiency
RR	1	64KB	204	255	485	42.01%	52.50%
RR	1	1MB	328	335	485	67.58%	69.13%
RR	1	16MB	389	400	485	80.19%	82.39%
RR	2	64KB	205	337	485	42.21%	69.52%
RR	2	1MB	330	400	485	68.11%	82.55%
RR	2	16MB	389	412	485	80.26%	85.04%
RR	4	64KB	205	353	485	42.30%	72.72%
RR	4	1MB	330	423	485	68.14%	87.29%
RR	4	16MB	392	410	485	80.82%	84.61%
RR	8	64KB	206	374	485	42.41%	77.06%
RR	8	1MB	330	428	485	68.04%	88.21%
RR	8	16MB	391	410	485	80.59%	84.63%
RR	12	64KB	205	389	485	42.34%	80.16%
RR	12	1MB	330	413	485	68.08%	85.12%
RR	12	16MB	394	409	485	81.25%	84.35%
RR	24	64KB	205	406	485	42.35%	83.70%
RR	24	1MB	331	412	485	68.28%	84.98%
RR	24	16MB	402	400	485	82.98%	82.49%
RR	48	64KB	203	410	485	41.90%	84.53%
RR	48	1MB	330	422	485	68.14%	87.06%
RR	48	16MB	401	385	485	82.60%	79.34%



Write Random

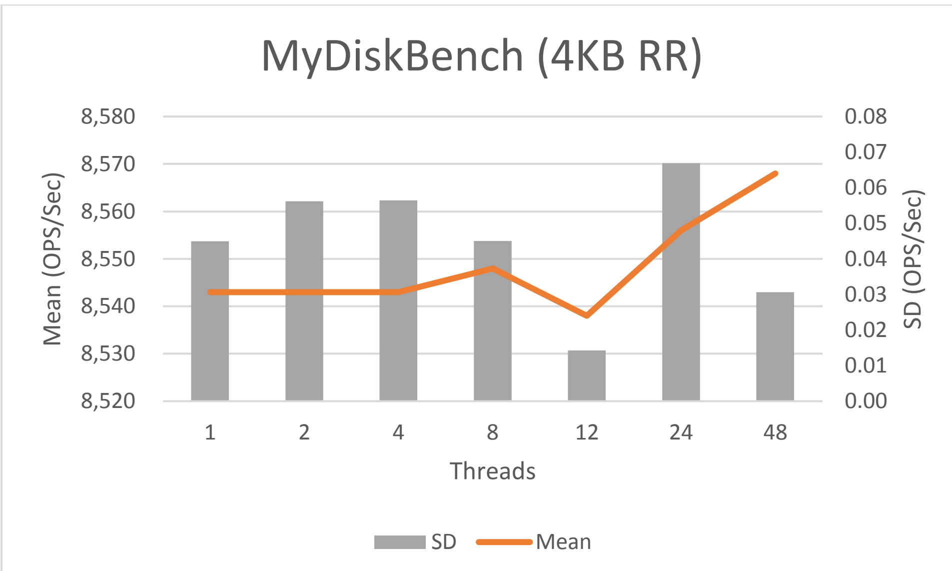
Concurrency	Record Size	MyDiskBench (OPS/sec)	IOZONE (OPS/sec)	Theoretical	MyDiskBench Efficiency	IOZone Efficiency
1	4KB	23,377	23,204	95,000	24.61%	24.43%
2	4KB	23,278	44,213	95,000	24.50%	46.54%
4	4KB	23,485	59,147	95,000	24.72%	62.26%
8	4KB	23,348	62,380	95,000	24.58%	65.66%
12	4KB	23,621	64,571	95,000	24.86%	67.97%
24	4KB	23,333	65,487	95,000	24.56%	68.93%
48	4KB	23,026	64,360	95,000	24.24%	67.75%

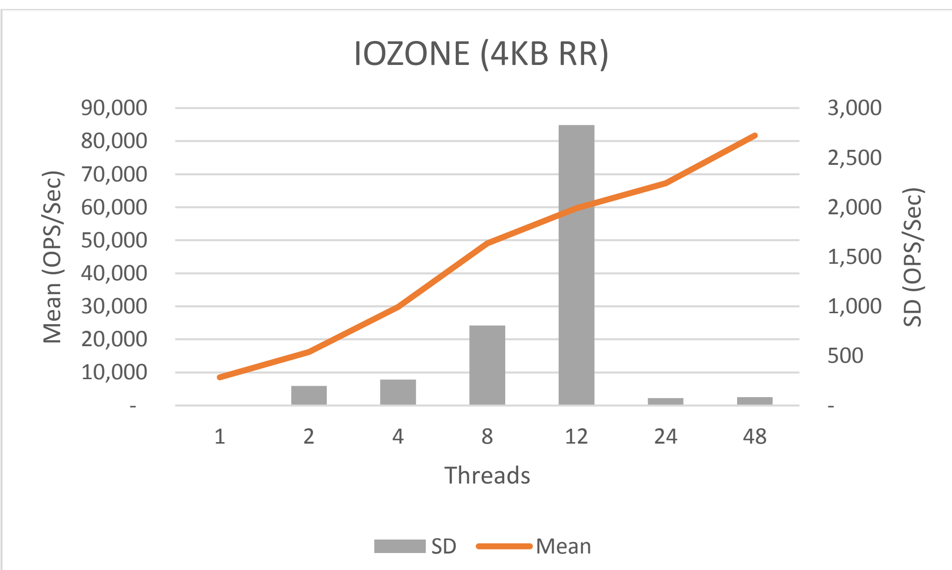




Read Random

Concurrency	Record Size	MyDiskBench (OPS/sec)	IOZONE (OPS/sec)	Theoretical	MyDiskBench Efficiency	IOZone Efficiency
1	4KB	8,543	8,527	95,000	8.99%	8.98%
2	4KB	8,543	16,170	95,000	8.99%	17.02%
4	4KB	8,543	29,776	95,000	8.99%	31.34%
8	4KB	8,548	49,100	95,000	9.00%	51.68%
12	4KB	8,538	59,708	95,000	8.99%	62.85%
24	4KB	8,556	67,214	95,000	9.01%	70.75%
48	4KB	8,568	81,702	95,000	9.02%	86.00%





We can see that there is an improvement in the reading and writing speeds as the block size increases but the same cannot be inferred with the increase in the number of threads. This is because there is no fixed pattern.

For the 4KB record size, the performance is constantly poor and changing the number of threads doesn't have any effect on it. It contrasts with the Iozone benchmarking results for 4KB, where even though the results are poor for 1 thread and 4KB record size, we can see a constant improvement as the number of threads increase.

Efficiency is very low for random writes as well which is to be expected since write is an expensive operation and when coupled with random, can result in poof performance

Also, the program is giving better efficiency for WR than RR for 4KB block size. This is an anomaly since we do not see similar behavior for any other record size while using the same program

Future Scope:

1. Deep dive and figure out why the WR is performing better than RR for 4KB record size
2. Make improvements in the program by further research since Iozone has better performance and thus we can perform better as well
3. Make improvements in the random read and write to make sure all the data is read or written and no same block is accessed again