Ivan Prskalo
CS351
Lab 3 - Benchmarking Storage

*All tests n on fourier

Workload	Concurrency	Record Size	FileIO Measured Throughput (MiB/sec)	IOZone Measured Throughput (MiB/sec)
WS	1	64K	295.59	176.03
WS	1	1M	228.74	203.40
WS	1	16M	251.47	235.88
WS	2	64K	250.19	136.27
WS	2	1M	274.40	267.20
WS	2	16M	250.64	279.91
WS	4	64K	225.23	136.06
WS	4	1M	194.47	97.34
WS	4	16M	204.37	222.05
WS	8	64K	208.63	173.84
WS	8	1M	211.59	176.45
WS	8	16M	214.75	164.45

Workload	Concurrency	Record Size	FileIO Measured Throughput (MiB/sec)	IOZone Measured Throughput (MiB/sec)
RS	1	64K	249.09	281.55
RS	1	1M	396.65	284.23
RS	1	16M	361.48	298.93
RS	2	64K	189.05	107.58
RS	2	1M	279.24	245.54
RS	2	16M	384.19	136.52
RS	4	64K	195.27	153.07

RS	4	1M	266.96	93.92
RS	4	16M	327.99	315.06
RS	8	64K	206.32	194.47
RS	8	1M	267.36	225.63
RS	8	16M	280.81	168.23

Workload	**Concurrency**	**Record Size**	**FileIO Measured Throughput (MiB/sec)**	**IOZone Measured Throughput (MiB/sec)**
WR	1	64K	819.39	45.04
WR	1	1M	1391.37	141.63
WR	1	16M	1526.46	179.03
WR	2	64K	1357.01	50.02
WR	2	1M	1619.64	138.55
WR	2	16M	1615.22	205.13
WR	4	64K	1619.98	60.37
WR	4	1M	1620.59	128.52
WR	4	16M	1559.77	162.11
WR	8	64K	1607.91	64.60
WR	8	1M	1605.74	81.97
WR	8	16M	1192.63	198.28

Workload	**Concurrency**	**Record Size**	**FileIO Measured Throughput (MiB/sec)**	**IOZone Measured Throughput (MiB/sec)**
RR	1	64K	254.42	19.05
RR	1	1M	810.46	38.45
RR	1	16M	826.25	369.39
RR	2	64K	270.09	21.15

RR	2	1M	1121.58	160.84
RR	2	16M	485.01	196.58
RR	4	64K	603.55	20.41
RR	4	1M	1178.41	83.59
RR	4	16M	277.86	300.59
RR	8	64K	773.17	86.19
RR	8	1M	1161.21	44.34
RR	8	16M	226.72	122.37

Workload	**Concurrency**	**Record Size**	**FileIO Measured Latency (Ops/sec)**	**IOZone Measured Latency (Ops/sec)**
WR	1	4K	28182.19	26150.02
WR	2	4K	39553.48	32350.21
WR	4	4K	61460.90	58400.58
WR	8	4K	74706.35	70116.36

Workload	**Concurrency**	**Record Size**	**FileIO Measured Latency (Ops/sec)**	**IOZone Measured Latency (Ops/sec)**
RR	1	4K	30420.06	29189.65
RR	2	4K	42037.85	41911.24
RR	4	4K	59324.49	56271.99
RR	8	4K	69235.97	65054.76

Latency Analysis:
In both my FileIO and IOZone's latency benchmarks, the lowest level of concurrency (1) caused the least amount of operations per second. Likewise, the highest level of concurrency (8) caused the most amount of operations per second. The range of values from write random was larger than the range from read random in both cases. Which suggests that latency performance benefits more from increased concurrency while writing than while reading files.
Here I'm just describing the data:
For write sequential, the highest throughput was achieved in my FileIO with the lowest record size of 64KiB and only once process running. In IOZone, the highest throughput was with two

processes running and the highest record size of 16M. For both FileIO and IOZone, the worst throughput was when record size was 1M and four processes running.
For read sequential, the highest throughput was achieved in my FileIO when one process was running and the record size was 1M. In IOZone, the highest throughput was when four processes were running with record size of 16M. The lowest throughput was the same for both with two processes running and a record size of 64K.
For write random, the highest throughput was achieved in my FileIO when four processes were running with record size of 1M. In IOZone, the highest throughput was when two processes were running with record size of 16M. Once again the lowest throughput was the same for my FileIO and IOZone this time with once process running with record size of 64K
For read random, the highest throughput was achieved in my FileIO when four processes were running with record size of 1M. In IOZone, the highest throughput was when one process was running at 16M. For my FileIO, the lowest throughput was when eight processes were running with record size of 16M. In IOZone, the lowest throughput was when one process was running with record size of 64K.

Here I look for patterns and reasoning:

For three out of the four workloads I got the same lowest throughput record size and concurrency for both FileIO and IOZone. For RR, the remaining one, the lowest were polar opposites with FileIO's being at the highest concurrency and record size and IOZone's being at the lowest concurrency and record size.
The actual values of throughput varied drastically between FileIO and IOZone. FileIO had a max throughput of 1620.59 and IOZone a max of 369.39. The same goes for the min throughputs with FileIO having a minimum of only 189.05 while IOZone had a min throughput of 19.05. I'm not too sure why there are such large differences, but one possible reason is that since the differences were most prominent in write random and read random, the performance was different because of the random element. In write sequential and read sequential the differences are much less prominent.
As concurrency increased once might expect that the throughput would increase as more processes are reading/writing over the same amount of total data, but this is not the case. In many cases this varies or even decreases as concurrency increases. I believe that this is due to the overhead that occurs when there are multiple processes.
It is interesting to note that IOZone's highest throughputs were consistently when record size was 16M which is the largest record size. For my FileIO the highest throughput was three out of four times when the record size was 1M and once when record size was 64K. These differences are probably due to the way my code is written compared to IOZone.