ECSE 4961 ACS Project 3 Report

Yilu Zhou

Yutong Xie

1. **Experiment Environment:**

OS	Windows 10 Home
Processor	i7 9750H @ 2.6Ghz 2.59Ghz
RAM	16GB
SSD	Intel 660p 512GB

2. **Experiment Setting:**

a) Memory:

The team used intel MLC to test memory sequential read/write performance under different setting of r/w proportion and stride length.

Random/sequential	Sequential r/w
r/w proportion	Read only, 3:1, 2:1, 1:1, write only
Stride length	64B/256B
Tool	mlc

b) SSD:

The team used Fio to test SSD random read/write performance given different r/w proportion and block size.

Random/sequential	Random r/w
r/w proportion	Read only, 70:30, write only
numjobs	5
size	1G
runtime	100
ioengine	windowsaio
Block size	4k/32k
Tool	fio

3. **Experiment Result**

a) Memory

a. Stride length=64B, Read only

Command= "--loaded_latency"

Inject Delay	latency (ns)	Bandwidth (MB/sec)
0	181.96	34581.3
2	174.54	34952
8	171.11	34912.9
15	169.29	34773.2
50	142.41	34631.6
100	103.89	32542.3
200	74.9	23358.4
300	65.9	16632.7

400	61.33	13104.3
500	59.75	10798.6
700	56.84	8175.3
1000	54.64	6140.3
1300	54.18	5025
1700	53.53	4149.4
2500	53.42	3212.9
3500	52.85	2653.2
5000	63.53	1983
9000	55.09	1720.3
20000	52.19	1479.6

b. Stride length=64B, r/w=3
Command= "--loaded_latency -W9"

Inject Delay	latency (ns)	Bandwidth (MB/sec)
0	277.62	29612.1
2	270.02	29667.8
8	259.47	29678.9
15	250.46	29877
50	203.48	30316.7
100	162.79	29769.5
200	89.44	23758.6
300	75.13	16648.4
400	69.3	13035.1
500	65.12	10775
700	60.89	8095.9
1000	57.66	6074
1300	56.41	4982.2
1700	54.97	4117.5
2500	54.35	3195.2
3500	53.81	2631.4
5000	53.09	2216.2
9000	53.21	1764.9
20000	52.69	1467.6

c. Stride length=64B, r/w=2
Command= "--loaded_latency -W7"

Inject Delay	latency (ns)	Bandwidth (MB/sec)
0	394.84	26746.2
2	358.73	29017.6

8	366.22	28247.5
15	337.7	27601.3
50	302.26	28099.1
100	234.98	29332.4
200	194.71	28562.1
300	96.92	23686.1
400	75.55	18899.3
500	69.84	15514.4
700	63.59	11588.3
1000	59.52	8534
1300	57.67	6887.3
1700	56.36	5563.2
2500	54.44	4197.2
3500	54.15	3346.8
5000	53.5	2705.9
9000	53.08	2050.5
20000	53	1588

d. Stride length=64B, r/w=1
Command= "--loaded_latency -W8"

Inject Delay	latency (ns)	Bandwidth (MB/sec)
0	355.14	26713.4
2	350.87	26699.8
8	350.33	26919.5
15	321.2	27426.2
50	271.43	27690.3
100	212.84	28230
200	112.64	22367.5
300	75.86	16628.3
400	78.69	12650.5
500	65.25	10728.8
700	61.39	8065.2
1000	57.48	6065.6
1300	56.33	4970.9
1700	55.59	4091.6
2500	56.85	3119.5
3500	56.02	2565.3
5000	60.22	2034.9
9000	53.06	1767.9
20000	52.21	1479

e. Stride length=64B, write only
Command = "--loaded_latency -W6"

Inject Delay	latency (ns)	Bandwidth (MB/sec)
0	322.91	32955.5
2	318.2	32958.6
8	328.83	32761.7
15	311.49	32994.3
50	85.85	22067.4
100	70.45	10771.3
200	66.62	6662.3
300	60.94	4991.2
400	59.35	4081.7
500	57.45	3540.1
700	54.89	2923.3
1000	54.26	2421
1300	53.11	2161.7
1700	52.91	1948
2500	52.76	1717.1
3500	52.49	1579.2
5000	52.9	1461.9
9000	51.85	1374.7
20000	51.86	1297.4

f. Stride length=256B, read only
Command = "--loaded_latency -l256"

Inject Delay	latency (ns)	Bandwidth (MB/sec)
0	118	28715.6
2	116.3	28819.8
8	118.27	27685.3
15	118.37	26175.9
50	101.8	24131.6
100	89.95	21992.1
200	79.97	17082.3
300	72.41	12874.5
400	76.4	10460.6
500	68.63	9026.7
700	64.97	7172.6
1000	64.26	5534.5
1300	62.06	4612.1
1700	61.81	3816
2500	59.83	3017.5
3500	58.45	2500.9
5000	58.83	2075.4
9000	57.12	1679
20000	57.54	1364.7

g. Stride length=256B, r/w = 3:1
Command = "--loaded_latency -W9 -l256"

Inject Delay	latency (ns)	Bandwidth (MB/sec)
0	186.96	25192.9
2	182.44	25445.6
8	175.88	25287.6
15	173.28	24755.2
50	145.93	21747.6
100	120.07	20984.2
200	107.54	17006.7
300	89.99	13338
400	80.85	10659.8
500	76.31	9083.1
700	70.88	7172
1000	66.66	5558.7
1300	62.88	4659.1
1700	62.19	3869.3
2500	60.64	3016.2
3500	58.84	2502.1
5000	58.48	2089.2
9000	57.15	1677.8
20000	57.31	1369.2

h. Stride length=256B, r/w = 2:1
Command = "--loaded_latency -W7 -l256"

Inject Delay	latency (ns)	Bandwidth (MB/sec)
0	316.14	20574.4
2	243.17	24112.3
8	195.87	25050.7
15	190.27	24476.2
50	157.66	23481.3
100	143.48	23235.2
200	121.6	21057.9
300	100.38	18011.3
400	87.94	14851.5
500	81.3	12699.4
700	77.47	9850.6
1000	70.11	7621.4
1300	66.31	6299.5
1700	65.8	5129.3
2500	69.27	3760.7
3500	60.44	3161.5

5000	62.55	2485.8
9000	58.51	1928.6
20000	57.59	1489.4

i. Stride length=256B, r/w = 1:1
Command = "--loaded_latency -W8 -l256"

Inject Delay	latency (ns)	Bandwidth (MB/sec)
0	250.36	26204.5
2	248.54	26100.7
8	234.82	26131.5
15	211.12	25678.4
50	159.91	23277.4
100	152.58	20634.1
200	130.17	17257
300	105.76	14016
400	87.86	11080.6
500	79.19	9541.6
700	73.66	7359.9
1000	67.19	5675
1300	64.65	4695
1700	61.49	3920.3
2500	60.07	3045.7
3500	58.25	2524.4
5000	57.71	2112.9
9000	57.23	1678.5
20000	56.38	1388.4

j. Stride length=256B, write only
Command = "--loaded_latency -W6 -l256"

Inject Delay	latency (ns)	Bandwidth (MB/sec)
0	330.53	8386.7
2	332.43	8299.1
8	332.7	8406.7
15	324.6	8360
50	88.86	6245
100	74.25	3294.3
200	69.09	2346.2
300	63.94	1986.3
400	62.22	1779.4
500	59.86	1676.8
700	58.04	1541.4
1000	56.8	1437.4
1300	56.63	1369.7

1700	56.74	1311.9
2500	55.82	1272.4
3500	55.87	1235.7
5000	55.44	1217.6
9000	56.48	1168.3
20000	56.54	1147.8

b) SSD

Command list:

queue depth	data access size	r:w	command
1	4k	r	fio.exe -filename="C:\Users\xieyu\Documents\ACS\p3\test.txt" -ioengine=windowsaio -direct=1 -iodepth=1 -thread -rw=randread -bs=4k -size=1G -numjobs=5 -runtime=100 -group_reporting -name=mytest
4	4k	r	fio.exe -filename="C:\Users\xieyu\Documents\ACS\p3\test.txt" -ioengine=windowsaio -direct=1 -iodepth=4 -thread -rw=randread -bs=4k -size=1G -numjobs=5 -runtime=100 -group_reporting -name=mytest
16	4k	r	fio.exe -filename="C:\Users\xieyu\Documents\ACS\p3\test.txt" -ioengine=windowsaio -direct=1 -iodepth=16 -thread -rw=randread -bs=4k -size=1G -numjobs=5 -runtime=100 -group_reporting -name=mytest
1	32k	r	fio.exe -filename="C:\Users\xieyu\Documents\ACS\p3\test.txt" -ioengine=windowsaio -direct=1 -iodepth=1 -thread -rw=randread -bs=32k -size=1G -numjobs=5 -runtime=100 -group_reporting -name=mytest
4	32k	r	fio.exe -filename="C:\Users\xieyu\Documents\ACS\p3\test.txt" -ioengine=windowsaio -direct=1 -iodepth=4 -thread -rw=randread -bs=32k -size=1G -numjobs=5 -runtime=100 -group_reporting -name=mytest
16	32k	r	fio.exe -filename="C:\Users\xieyu\Documents\ACS\p3\test.txt" -ioengine=windowsaio -direct=1 -iodepth=16 -thread -rw=randread -bs=32k -size=1G -numjobs=5 -runtime=100 -group_reporting -name=mytest
1	4k	70:30	fio.exe -filename="C:\Users\xieyu\Documents\ACS\p3\test.txt" -ioengine=windowsaio -direct=1 -iodepth=1 -thread -rw=randrw -rwmixwrite=30 -bs=4k -size=1G -numjobs=5 -runtime=100 -group_reporting -name=mytest
4	4k	70:30	fio.exe -filename="C:\Users\xieyu\Documents\ACS\p3\test.txt" -ioengine=windowsaio -direct=1 -iodepth=4 -thread -rw=randrw -rwmixwrite=30 -bs=4k -size=1G -numjobs=5 -runtime=100 -group_reporting -name=mytest
16	4k	70:30	fio.exe -filename="C:\Users\xieyu\Documents\ACS\p3\test.txt" -ioengine=windowsaio -direct=1 -iodepth=16 -thread -rw=randrw -rwmixwrite=30 -bs=4k -size=1G -numjobs=5 -runtime=100 -group_reporting -name=mytest
1	32k	70:30	fio.exe -filename="C:\Users\xieyu\Documents\ACS\p3\test.txt" -ioengine=windowsaio -direct=1 -iodepth=1 -thread -rw=randrw -

			rwmixwrite=30 -bs=32k -size=1G -numjobs=5 -runtime=100 -group_reporting -name=mytest
4	32k	70:30	fio.exe -filename="C:\Users\xieyu\Documents\ACS\p3\test.txt" -ioengine=windowsaio -direct=1 -iodepth=4 -thread -rw=randrw -rwmixwrite=30 -bs=32k -size=1G -numjobs=5 -runtime=100 -group_reporting -name=mytest
16	32k	70:30	fio.exe -filename="C:\Users\xieyu\Documents\ACS\p3\test.txt" -ioengine=windowsaio -direct=1 -iodepth=16 -thread -rw=randrw -rwmixwrite=30 -bs=32k -size=1G -numjobs=5 -runtime=100 -group_reporting -name=mytest
1	4k	w	fio.exe -filename="C:\Users\xieyu\Documents\ACS\p3\test.txt" -ioengine=windowsaio -direct=1 -iodepth=1 -thread -rw=randwrite -bs=4k -size=1G -numjobs=5 -runtime=100 -group_reporting -name=mytest
4	4k	w	fio.exe -filename="C:\Users\xieyu\Documents\ACS\p3\test.txt" -ioengine=windowsaio -direct=1 -iodepth=4 -thread -rw=randwrite -bs=4k -size=1G -numjobs=5 -runtime=100 -group_reporting -name=mytest
16	4k	w	fio.exe -filename="C:\Users\xieyu\Documents\ACS\p3\test.txt" -ioengine=windowsaio -direct=1 -iodepth=16 -thread -rw=randwrite -bs=4k -size=1G -numjobs=5 -runtime=100 -group_reporting -name=mytest
1	32k	w	fio.exe -filename="C:\Users\xieyu\Documents\ACS\p3\test.txt" -ioengine=windowsaio -direct=1 -iodepth=1 -thread -rw=randwrite -bs=32k -size=1G -numjobs=5 -runtime=100 -group_reporting -name=mytest
4	32k	w	fio.exe -filename="C:\Users\xieyu\Documents\ACS\p3\test.txt" -ioengine=windowsaio -direct=1 -iodepth=4 -thread -rw=randwrite -bs=32k -size=1G -numjobs=5 -runtime=100 -group_reporting -name=mytest
16	32k	w	fio.exe -filename="C:\Users\xieyu\Documents\ACS\p3\test.txt" -ioengine=windowsaio -direct=1 -iodepth=16 -thread -rw=randwrite -bs=32k -size=1G -numjobs=5 -runtime=100 -group_reporting -name=mytest

Result:

queue depth	data access size	r:w	read avg iops	read avg bw(KiB/s)	read avg lat(usec)	write avg iops	write avg bw(KiB/s)	write avg lat(usec)
1	4k	r	25597.63	102396.17	194.51			
4	4k	r	38582.21	154333.51	516.95			
16	4k	r	41993.68	167978.53	1903.94			
1	32k	r	11498.07	367978.18	433.93			
4	32k	r	13566.92	434193.58	1473.81			
16	32k	r	13554.79	433807.79	5903.73			
1	4k	70:30	23311.53	93250.28	198.06	10019.81	40083.4	34.68
4	4k	70:30	36858.96	147440.08	523.83	15847.33	63393.61	36.98
16	4k	70:30	35012.83	140055.98	2262.85	15054.65	60222.79	45.75
1	32k	70:30	6200.42	198461.56	766.97	2675.36	85651.42	76.72
4	32k	70:30	6608.97	211530.12	2652.49	2849.88	91252.79	852.65

16	32k	70:30	7088.44	226863.38	8404.36	3059.88	97964.25	6670.06
1	4k	w				109667	438672.39	44.73
4	4k	w				189443.23	757780.92	103.49
16	4k	w				194863.92	779461.51	382.89
1	32k	w				28463.91	910907	175.46
4	32k	w				28590.82	914974.18	699.98
16	32k	w				28768.09	920642.36	2787.31

4. Analysis

In memory experiment, the team used intel MLC to test the memory bandwidth and latency under different settings of stride length and read/write proportion. According to the result of the experiment, when stride length and r/w proportion fixed, increasing the injection delay reduces the latency, but decreases the bandwidth. When stride length fixed, increasing the r/w proportion causes the latency increase under similar bandwidth. This indicates the fact that write operation has larger latency than read operation in memory. In addition, increase the stride length from 64B to 256B also increase the latency under similar bandwidth.

In SSD experiment, the team test the random read/write performance with Fio given different setting of read/write proportion and data access length. According to the experiment, when the queue depth increases from 1 to 4 and 16, both bandwidth and latency increases, resulting in an increasing of iosp. When queue depth and read/write proportion are fixed, increasing data access size increases bandwidth and latency, and the iosp decreases. When queue depth and data access size are fixed, increasing the read/write proportion have floated affect on read bandwidth/latency and write bandwidth/latency, but both read and write iosp decrease. In addition, the write latency is smaller than the read latency, but the write bandwidth is smaller than the read bandwidth. Overall, the write iosp is smaller than the read iosp.

5. Conclusion

By working on the experiment with both memory and SSD, the team realize that there is a tradeoff between bandwidth and latency when changing the queue depth and data access size. In addition, different rate of read and write option also changes the performance of SSD and memory.

Comparing to D7-5600, the team's SSD have worse performance on read iosp, but the write iosp is better than D7-5600 when the queue depth increases to 16. This might because the D7-5600 has more functions on data security, error detection, and data duration, which decrease the write iops.