Project Report 3

Linsen Ma, Yuyang Zhang

Overview

In this project, in order to get a deeper understanding on the performance of modern memory and storage devices, the team use Intel Memory Latency Checker(MLC) and Flexible IO tester(FIO) to experiment on measuring read/write latency of our memory and storage devices. After the experiments, the team find out an obvious trade-off between access latency and throughput.

Testing Platform

CPU: i7-6700HQ

Memory: 16GB

Storage: WD Black M.2 SSD

OS: Ubuntu 20.04 LTS

Experiment results

To get a comprehensive result, the team conduct experiments for both memory and storage with various command settings.

Memory

For Memory Experiments, the team test 64B Read only, Read:Write 3:1, Read:Write 2:1, Read:Write 1:1, Write only, 256B Read only, Read:Write 3:1, Read:Write 2:1, Read:Write 1:1, Write only.

a. 64B, read only
   Command "sudo ./mlc --loaded_latency"

Inject Delay	latency (ns)	Bandwidth (MB/sec)
00000	204.11	17783.5
00002	203.71	17836.0
00008	180.08	19852.9
00015	155.17	22026.3
00050	106.63	25930.6
00100	70.81	15999.3
00200	65.25	9989.8
00300	62.65	7325.2
00400	62.12	5874.8
00500	61.44	4991.5
00700	60.17	3997.9
01000	63.54	3085.0

01300	67.95	2518.4
01700	67.39	2128.9
02500	63.14	1861.7
03500	61.08	1660.8
05000	61.98	1455.0
09000	59.64	1313.1
20000	59.32	1187.2

b. 64B, read : write = 3 : 1
Command "sudo ./mlc --loaded_latency -W9"

Inject Delay	latency (ns)	Bandwidth (MB/sec)
00000	208.72	21243.8
00002	198.55	22215.7
00008	184.85	23265.6
00015	157.59	25972.2
00050	116.57	27944.1
00100	75.25	16745.0
00200	67.35	10336.0
00300	64.09	7479.2
00400	64.57	5977.3
00500	61.70	5137.1
00700	60.53	4041.7
01000	59.54	3180.7
01300	59.07	2713.7
01700	58.46	2357.9
02500	58.19	1954.1
03500	57.92	1718.2
05000	57.80	1538.7
09000	57.44	1360.8
20000	57.32	1227.6

c. 64B, read : write = 2 : 1
Command "sudo ./mlc --loaded_latency -W7"

Inject Delay	latency (ns)	Bandwidth (MB/sec)
00000	225.82	22119.2
00002	235.27	21127.5
00008	194.30	24157.1
00015	179.88	25000.4
00050	139.14	28041.8
00100	84.75	23870.5

00200	71.56	14655.7
00300	66.62	10513.8
00400	64.29	8316.9
00500	62.58	7159.1
00700	61.38	5490.8
01000	60.20	4209.4
01300	62.71	3415.6
01700	59.76	2937.8
02500	59.29	2359.1
03500	58.95	2000.6
05000	60.90	1686.3
09000	58.52	1452.1
20000	59.20	1242.2

d. 64B, read : write = 1 : 1
Command= "sudo ./mlc --loaded_latency -W8"

Inject Delay	latency (ns)	Bandwidth (MB/sec)
00000	231.03	24959.6
00002	240.58	23735.5
00008	213.18	26268.9
00015	171.63	27516.5
00050	92.76	18252.4
00100	102.53	8959.6
00200	65.29	5968.3
00300	82.25	3751.9
00400	75.37	3217.2
00500	79.61	2734.1
00700	63.20	2512.6
01000	69.89	1923.7
01300	59.04	1899.3
01700	58.52	1722.3
02500	57.94	1535.2
03500	58.43	1401.8
05000	58.16	1314.9
09000	57.64	1229.5
20000	57.96	1158.3

e. 64B, write only
Command = "sudo ./mlc --loaded_latency -W6"

Inject Delay	latency (ns)	Bandwidth (MB/sec)
00000	366.94	29218.4
00002	387.52	27741.5

00008	355.39	30592.7
00015	93.86	26377.6
00050	67.59	10410.0
00100	65.44	5033.6
00200	65.16	3387.4
00300	59.69	2753.3
00400	59.36	2357.8
00500	58.33	2130.4
00700	58.46	1840.9
01000	58.53	1620.1
01300	57.79	1514.2
01700	58.16	1412.8
02500	57.92	1318.2
03500	57.41	1267.3
05000	57.91	1212.2
09000	57.80	1166.9
20000	57.85	1133.1

f. 256B, read only
Command = "sudo ./mlc --loaded_latency -l256"

Inject Delay	latency (ns)	Bandwidth (MB/sec)
00000	143.98	10475.7
00002	145.21	10417.0
00008	130.70	11712.8
00015	129.30	11565.6
00050	100.45	12828.7
00100	81.89	11065.3
00200	73.85	8568.7
00300	70.05	6615.0
00400	68.50	5506.8
00500	67.47	4697.5
00700	66.13	3725.3
01000	65.87	2939.8
01300	65.22	2513.2
01700	64.82	2169.7
02500	64.57	1801.8
03500	64.11	1586.5
05000	63.56	1424.2
09000	63.31	1245.7
20000	62.76	1127.0

g. 256B, read : write = 3:1
Command = "sudo ./mlc --loaded_latency -l256 -W9 "

Inject Delay	latency (ns)	Bandwidth (MB/sec)
00000	142.48	12717.1
00002	141.78	12763.4
00008	131.71	14071.8
00015	114.85	16864.7
00050	102.05	15176.6
00100	86.55	12081.9
00200	79.82	8580.0
00300	76.81	6447.1
00400	73.15	5369.9
00500	68.66	4727.9
00700	66.53	3751.8
01000	65.25	2970.9
01300	64.21	2542.9
01700	63.77	2195.1
02500	63.09	1854.8
03500	62.91	1615.5
05000	62.40	1451.8
09000	62.10	1268.6
20000	61.72	1144.1

h. 256B, read / write = 2:1
Command = "sudo ./mlc --loaded_latency -l256 -W7 "

Inject Delay	latency (ns)	Bandwidth (MB/sec)
00000	148.96	13167.7
00002	150.93	13115.4
00008	134.85	14769.8
00015	126.69	15902.7
00050	106.37	17057.9
00100	89.35	16104.2
00200	80.43	12302.3
00300	75.21	9548.7
00400	71.88	7740.5
00500	69.63	6553.6
00700	67.70	5080.7
01000	66.14	3924.0
01300	65.50	3278.2
01700	64.92	2761.1
02500	64.41	2211.1
03500	63.61	1895.7
05000	63.35	1633.8
09000	62.36	1382.8

20000	62.18	1190.6

i. 256B, read : write = 1:1
Command = "sudo ./mlc --loaded_latency -l256 -W8"

Inject Delay	latency (ns)	Bandwidth (MB/sec)
00000	148.11	15740.5
00002	148.04	15877.4
00008	127.14	19146.2
00015	118.00	20446.7
00050	98.04	16969.2
00100	81.79	9920.1
00200	71.61	5801.8
00300	68.05	4272.2
00400	66.30	3484.9
00500	64.90	3017.7
00700	64.58	2464.5
01000	64.05	2035.3
01300	63.31	1819.1
01700	63.06	1641.1
02500	62.82	1445.4
03500	62.38	1331.8
05000	62.71	1234.5
09000	62.45	1144.0
20000	62.03	1085.5

j. 256B, write only
Command = "sudo ./mlc --loaded_latency -l256 -W6"

Inject Delay	latency (ns)	Bandwidth (MB/sec)
00000	370.94	29854.1
00002	370.81	29862.0
00008	370.51	29869.6
00015	93.95	26334.6
00050	69.58	10291.2
00100	66.73	5103.6
00200	63.27	3455.6
00300	62.42	2704.9
00400	62.23	2307.1
00500	66.79	1963.9
00700	67.14	1683.9
01000	64.12	1519.2
01300	63.37	1406.7
01700	64.63	1299.1

02500	62.69	1232.3
03500	65.64	1123.7
05000	65.85	1075.3
09000	64.68	1048.0
20000	63.99	1026.4

Storage

To get a comprehensive understanding on the tradeoff between throughput and latency, the team use FIO to test the SSD with: 4k/32k Bs, 512MB size, 8 Numjobs, 20 Runtime. Moreover, the team test the SSD with Randread, Randwrite, Randrw with read : write 7 : 3 settings.

a. 4k Bs Randread
Command "sudo fio --name =read --ioengine=libaio --filename=/dev/nvme0n1 --iodepth=1/4/8/32/128 --rw=randread --bs=4k --direct=1 --numjobs=8 --runtime=20 --group_reporting"

Iodepth	Bs	Size	Numjobs	Runtime	Iops min	Iops max	Iops avg	Utility(%)	Latency(usec)
1	4k	512M	8	20	68822	74167	72613	99.41	93.3
4	4k	512M	8	20	129872	134632	132017	98.8	194.04
8	4k	512M	8	20	142714	148214	145645	98.63	344.06
32	4k	512M	8	20	145646	154292	150072	98.56	1329.96
128	4k	512M	8	20	133978	154130	149215	97.96	5318.36

b. 4k Bs Randwrite
Command "sudo fio --name =write --ioengine=libaio --filename=/dev/nvme0n1 --iodepth=1/4/8/32/128 --rw=randwrite --bs=4k --direct=1 --numjobs=8 --runtime=20 --group_reporting"

Iodepth	Bs	Size	Numjobs	Runtime	Iops min	Iops max	Iops avg	Utility(%)	Latency(usec)
1	4k	512M	8	20	203365	349486	301050	97.68	22.01
4	4k	512M	8	20	353688	366434	361466	97.3	86.94
8	4k	512M	8	20	349218	371110	360683	97.03	176.63
32	4k	512M	8	20	359376	371170	366716	96.18	933.9
128	4k	512M	8	20	107636	356124	204425	98.7	4733.39

c. 4k Bs Randrw read : write = 7 : 3
Command "sudo fio --name =readwrite --ioengine=libaio --filename=/dev/nvme0n1 --iodepth=1/4/8/32/128 --rw=randrw --rwmixwrite=30 --bs=4k --direct=1 --numjobs=8 --runtime=20 --group_reporting"

Iodepth	Bs	Size	Numjobs	Runtime	Iops avg write	Write latency(usec)	Iops avg read	Read latency(usec)	Utility(%)
1	4k	512M	8	20	25119	34.48	58587	116.49	99.38
4	4k	512M	8	20	55047	21.72	128382	237.65	98.29
8	4k	512M	8	20	56681	18.34	132212	473.73	98.38
32	4k	512M	8	20	56156	26.08	130928	1934.55	98.33
128	4k	512M	8	20	55856	4124.82	130231	6073.34	98.4

d. 32k Bs Randread
Command “sudo fio --name =read --ioengine=libaio --filename=/dev/nvme0n1 --iodepth=1/4/8/32/128 --rw=randread --bs=32k --direct=1 --numjobs=8 --runtime=20 --group_reporting”

Iodepth	Bs	Size	Numjobs	Runtime	Iops min	Iops max	Iops avg	Utility(%)	Latency(usec)
1	32k	512M	8	20	41710	43574	42617	96.76	186.09
4	32k	512M	8	20	80906	82348	81579	93.77	389.51
8	32k	512M	8	20	98962	99406	99184	92.63	640.65
32	32k	512M	8	20	91994	104122	98058	91.99	2633.3
128	32k	512M	8	20	90416	102616	96516	91.91	10611

e. 32k Bs Randwrite
Command “sudo fio --name =write --ioengine=libaio --filename=/dev/nvme0n1 --iodepth=1/4/8/32/128 --rw=randwrite --bs=4k --direct=1 --numjobs=8 --runtime=20 --group_reporting”

Iodepth	Bs	Size	Numjobs	Runtime	Iops min	Iops max	Iops avg	Utility(%)	Latency(usec)
1	32k	512M	8	20	49142	49470	49296	96.54	157.92
4	32k	512M	8	20	48138	49964	49208	96.4	642.53
8	32k	512M	8	20	48198	49110	48640	96.54	1312.07
32	32k	512M	8	20	14004	49514	38897	97.2	6412.73
128	32k	512M	8	20	12766	50186	27465	97.89	35208.54

f. 32k Bs Randrw read : write = 7 : 3
Command “sudo fio --name =readwrite --ioengine=libaio --filename=/dev/nvme0n1 --iodepth=1/4/8/32/128 --rw=randrw --rwmixwrite=30 --bs=32k --direct=1 --numjobs=8 --runtime=20 --group_reporting”

Iodepth	Bs	Size	Numjobs	Runtime	Iops avg write	Write latency(usec)	Iops avg read	Read latency(usec)	Utility(%)
1	32k	512M	8	20	12862	38.91	30002	246.57	96.8
4	32k	512M	8	20	17923	55.88	41670	740.31	95.59
8	32k	512M	8	20	17912	1053.05	41672	1070.96	95.55
32	32k	512M	8	20	17622	11667.18	41178	1206.47	95.59
128	32k	512M	8	20	17755	24802.93	41512	14077.67	95.42

Analysis

Memory

From the experiment results, the team figure out with fixed stride length:64B/256B, increasing the injection delay will reduce the latency but bandwidth will decrease, too. For fixed stride length, with increase of ratio of write operation, the latency will increase a little with increased bandwidth in small injection delay and similar bandwidth in large injection delay. From this result, the team think it is because the write operation has a higher latency than the read operation in memory. And from the testing result, the team find out that when the stride length increase from 64B to 256B the latency under the same injection delay and other setting will increase and bandwidth will decrease.

Storage

From the experiment results, the team figure out with fixed Bs, Numjobs and other factors, for every testing scenario, increased Iodepth will cause increase in latency and an increase-flat-decrease trend for the IOPs. For fixed Iodepth, Numjobs and other factors, if we increase the Block size from 4k to 32k, IOPS will decrease and latency will increase(the total bandwidth will increase because each block increase from 4k to 32k but the IOPS do not reduce to less than 12.5%). For fixed Bs, Iodepth and other settings, the write operation will have lower IOPS than the read operation(lower Bandwidth) but in most cases, the write operation will have smaller latency than the read operation.(not 100%)

Conclusion

After conducting the comprehensive experiment with both SSD and memory, the team get to know the tradeoff between the bandwidth(IOPS) and latency if we vary the queue depth(Iodepth) and data access size. Moreover, the team get to know the performance differences for writing/reading operations in memory and SSD. And compared with the enterprise-grade D7-5600 SSD, the team’s consumer level SSD have poor performance in both read and write operation which matches with the team’s expectation about the SSD.