



운영체제
Assignment 5 과제

수업 명 : 운영체제
과제 이름 : assignment5
담당 교수님 : 최상호 교수님
학 번 : 2019202005
이 름 : 남종식

Introduction

이번 과제는 I/O Zone 을 이용한 세 가지 Linux I/O scheduler 성능 테스트를 하는 과제입니다. 세가지 Linux I/O scheduler 는 noop, deadline, CFQ 등이 있습니다.

noop 은 매우 간단한 I/O 스케줄링 알고리즘으로, 큐에 들어온 I/O 요청을 순서대로 처리합니다. 디스크 I/O 요청을 큐에 넣으면 가장 먼저 들어온 요청부터 처리하므로, 순차적인 디스크 I/O 에는 적합합니다.

Deadline 은 각 I/O 요청에 대해 deadline 을 부여하고, 이 마감 시간 안에 요청이 처리되도록 스케줄링합니다. 디스크 I/O 의 응답 시간을 개선하고, 실시간 응용 프로그램에 대한 보장된 응답 시간을 제공하는 데 중점을 둡니다.

CFQ 는 프로세스에 대한 공정성을 중시하는 I/O 스케줄러로, 각 프로세스에 대해 디스크 I/O 를 공정하게 분배하려고 노력합니다. 디스크 대역폭을 동적으로 조절하고, 여러 프로세스 간에 I/O 대역폭을 분배하여 응용 프로그램의 반응성을 향상시키기 위해 설계되었습니다.

실험 시에는 record size 를 변경해가면서 실행 후 이에 대한 결과를 비교하고 이를 분석해야합니다.

Conclusion & Analysis

```
os2019202005@ubuntu:~/Downloads/linux-4.19.67/os5$ sudo apt-get install iozone3
[sudo] password for os2019202005:
Reading package lists... Done
Building dependency tree
Reading state information... Done
iozone3 is already the newest version (429-3).
The following package was automatically installed and is no longer required:
  snapd-login-service
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 99 not upgraded.
```

IOZone 설치를 진행하였습니다.

```
os2019202005@ubuntu:~/Downloads/linux-4.19.67/os5$ cat /sys/block/sda/queue/scheduler
noop deadline [cfq]
os2019202005@ubuntu:~/Downloads/linux-4.19.67/os5$ echo noop | sudo tee /sys/block/sda/queue/scheduler
noop
os2019202005@ubuntu:~/Downloads/linux-4.19.67/os5$ cat /sys/block/sda/queue/scheduler
[noop] deadline cfq
os2019202005@ubuntu:~/Downloads/linux-4.19.67/os5$ echo cfq | sudo tee /sys/block/sda/queue/scheduler
cfq
os2019202005@ubuntu:~/Downloads/linux-4.19.67/os5$ cat /sys/block/sda/queue/scheduler
noop deadline [cfq]
os2019202005@ubuntu:~/Downloads/linux-4.19.67/os5$ echo deadline | sudo tee /sys/block/sda/queue/scheduler
deadline
os2019202005@ubuntu:~/Downloads/linux-4.19.67/os5$ cat /sys/block/sda/queue/scheduler
noop [deadline] cfq
```

Scheduler 를 확인하고 변경하는 과정을 진행하였습니다.

```
os2019202005@ubuntu:~$ make
rm -rf ~/iozone_test
sync
echo 3 | sudo tee /proc/sys/vm/drop_caches
3
```

캐시 및 버퍼를 비워서 실험에 영향을 주는 요소를 제거하기 위하여 이를 makefile 로 만들어 매 실험 전에 진행하였습니다.

```
os2019202005@ubuntu:~$ iozone -R -i 0 -i 1 -i 2 -i 5 -I -r 8k -s 1g -t 1 -F ~/iozone_test -b "./noop8k.xls"
```

```
os2019202005@ubuntu:~$ iozone -R -i 0 -i 1 -i 2 -i 5 -I -r 16k -s 1g -t 1 -F ~/iozone_test -b "./noop16k.xls"
```

```
os2019202005@ubuntu:~$ iozone -R -i 0 -i 1 -i 2 -i 5 -I -r 32k -s 1g -t 1 -F ~/iozone_test -b "./noop32k.xls"
```

```
os2019202005@ubuntu:~$ iozone -R -i 0 -i 1 -i 2 -i 5 -I -r 64k -s 1g -t 1 -F ~/iozone_test -b "./noop64k.xls"
```

```
os2019202005@ubuntu:~$ iozone -R -i 0 -i 1 -i 2 -i 5 -I -r 128k -s 1g -t 1 -F ~/iozone_test -b "./noop128k.xls"
```

```
os2019202005@ubuntu:~$ iozone -R -i 0 -i 1 -i 2 -i 5 -I -r 256k -s 1g -t 1 -F ~/iozone_test -b "./noop256k.xls"
```

```
iozone test complete.
os2019202005@ubuntu:~$ iozone -R -i 0 -i 1 -i 2 -i 5 -I -r 512k -s 1g -t 1 -F ~/iozone_test -b "./noop512k.xls"
```

```
os2019202005@ubuntu:~$ iozone -R -i 0 -i 1 -i 2 -i 5 -I -r 8m -s 1g -t 1 -F ~/iozone_test -b "./noop8m.xls"
```

```
os2019202005@ubuntu:~$ iozone -R -i 0 -i 1 -i 2 -i 5 -I -r 16m -s 1g -t 1 -F ~/iozone_test -b "./noop16m.xls"
```

먼저 -R 을 통해 excel report 를 생성하였고 -b[filename]을 통해 excel 과 호환 가능한 binary file 을 생성하였습니다. -i 를 통해 write/re-write read/re-read random-read/write 그리고 strid-read 를 따로 선택하여 진행했습니다. Stride -read 를 선택한 이유는 일반적인 write 와 read 를 제외하고 random read/write 측정이 있어 이와 대비되는 작업을 진행하고 싶어 선택하였습니다. Random read 는 임의의 위치에서 읽는 반면, stride read 는 일정한 간격으로 건너뛰어 데이터를 읽습니다. 이러한 이유로 stride read 를 선택하게 되었습니다.

-i 를 통해 파일에 대한 모든 operation 은 buffer cache 를 우회 및 disk 직접 이동하게 했습니다. -r 을 통해 record 를 직접 설정하여 이를 바꿔주며 테스트했습니다. -s 를 통해 file size 를 1 GB 로 설정하였고 -t 를 통해 thread 를 1 개만 사용하였습니다. -F 를 통해 thread or process 의 파일 경로를 ~/iozone_test 으로 설정하였습니다.

위 9 번의 과정을 5 번씩 총 3 개의 scheduler 에서 진행하였습니다.

	A	B	C	D	E	F	G	H	I
1	iozone -R -i 0 -i 1 -i 2 -i 5 -i r 256k -s 1g -t 1 -F /home/os2019202005/iozone_test -b ./deadline256k.xls								
2	Throughput report Y-axis is type of test X-axis is number of processes								
3	Record size = 256 kBytes								
4	Output is in kBytes/sec								
5	Initial write		1154275.375						
6	Rewrite		1202185.25						
7	Read		1455815.75						
8	Re-read		1484780.75						
9	Stride read		1430715.625						
10	Random read		1420967.125						
11	Random write		1120365.875						
12									
13									
14									

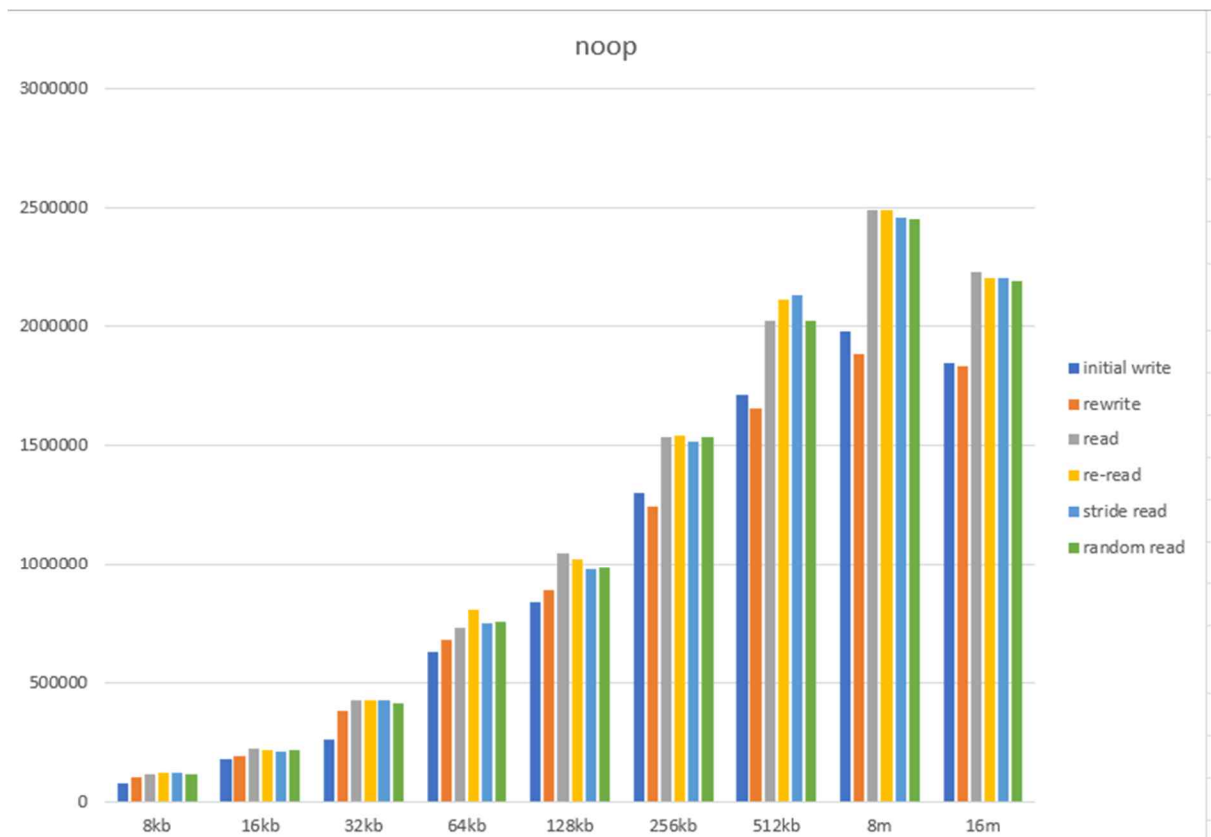
위 명령어를 진행하면 다음과 같은 excel 파일이 생성되는 것을 확인할 수 있으며 이 출력된 결과들을 다른 excel 파일에 정리하여 평균 값을 구했습니다.

Scheduler 별 성능 결과 분석

1 record크기	1차	2차	3차	4차	5차	평균
2 8k						
3 initial write	71160.44531	71406.46875	61932.44531	97056.85156	85793.92188	77470.02656
4 rewrite	103599.4375	110753.6641	101299.0938	103317.5469	103397.1641	104473.3813
5 read	110784.0703	118197.2422	114009.2813	119063.6875	113758.1016	115162.4766
6 re-read	110617.5703	119552.9141	112905.8984	126872.7109	128394.1328	119668.6453
7 stride read	110557.0938	124698.875	119308.2109	128432.2891	108484.9531	118296.2844
8 random read	106206.6016	123621.8203	106479.5156	123112.9453	117563.1016	115396.7969
9 random write	103389.6484	101623.9375	97429.27344	98394.13281	96589.26563	99485.25156
10 16k						
11 initial write	123013.5938	191103.84	184246.9688	184442.7188	193020.0938	175165.443
12 rewrite	194065.3125	155794.69	197521.6719	200310.7344	209954.7969	191529.4411
13 read	212063.8125	226315.36	239683	219074.2656	224934.0938	224414.1064
14 re-read	216456.75	216409.92	214729.6563	212459.4688	217034.5625	215418.0715
15 stride read	205296.2031	204616.02	207176.625	233373.625	210398.8438	212172.2634
16 random read	205430.7813	232557.02	204394.0156	226995.3125	207390.4063	215353.5071
17 random write	184428.0156	185063.39	184451.8281	222477.4063	190231.0625	193330.3405
18 32k						
19 initial write	221533.25	178180.6719	194541.9375	361183.5	359451.5938	262978.1906
20 rewrite	369360.0938	435263.9063	365507	366041.5938	358269	378888.3188
21 read	393615.6563	425862.4375	460986.2813	427340.9375	400453.8438	421651.8313
22 re-read	401287.8438	396727.4375	503590.7188	433640.9688	401274.375	427304.2688
23 stride read	388692.2813	482377.8125	387274.6875	460286.5313	393747.0625	422475.675
24 random read	384112.625	479010.9375	394174.8125	396799.4688	412456.9063	413310.95
25 random write	348808.625	344554.2188	362672.4375	360283.625	383428.3438	359949.45
26 64k						
27 initial write	633810.875	630133.6875	635368.5625	616932.75	625102.625	628269.7
28 rewrite	635220	664828.375	654105.3125	646584.625	782250.8125	676597.825
29 read	743298.75	739439.75	711400.0625	734114.9375	713327.625	728316.225
30 re-read	862001.6875	866795.8125	749669.25	739747.125	814880.4375	806618.8625
31 stride read	746398.125	733198.625	694027.25	723834.6875	842285.125	747948.7625
32 random read	838758.75	758226.125	763246.6875	713153.5625	707983	756273.625
33 random write	695022.25	741513.625	650849.625	653156.8125	662377.0625	680583.875
34 128k						
35 initial write	804062.3125	924261.6875	803360.5625	810145.9375	846013.4375	837568.7875
36 rewrite	857394.875	878695.3125	949379.1875	861035.375	880717.25	885444.4
37 read	972032.25	973506.75	1127569	984735.5625	1168346.875	1045238.088
38 re-read	962901.625	1049387.25	1019481.813	945416.75	1115283.5	1018494.188
39 stride read	948301.625	943616.5	1030487.75	899029.625	1067380.125	977763.125
40 random read	960025.25	966610.1875	1001729.125	955430.125	1026300.25	982018.9875
41 random write	836649.375	825021.5625	838818.375	735844.3125	837679.4375	814802.6125
42 256k						
43 initial write	1372144	1244321.5	1275121.75	1260795.5	1332217	1296919.95
44 rewrite	1284920.625	1196824	1214044	1259659.875	1265375.5	1244164.8
45 read	1438766.375	1536150.125	1482796.875	1577358.75	1633069.375	1533628.3
46 re-read	1560571.125	1541547.375	1437372.625	1591563.5	1576372.25	1541485.375
47 stride read	1530499.625	138843.75	1496032	1568086	1579856.375	1512563.55
48 random read	1641783.875	1448485.125	1512843.75	1505134.125	1553978	1532444.975

46	re-read	1560571.125	1541547.375	1437372.625	1591563.5	1576372.25	1541485.375
47	stride read	1530499.625	1388343.75	1496032	1568086	1579856.375	1512563.55
48	random read	1641783.875	1448485.125	1512843.75	1505134.125	1553978	1532444.975
49	random write	1223054.625	1230230.5	1228573	1230785.125	1238204.5	1230169.55
50	512k						
51	initial write	1662195.125	1675718.375	1749907.875	1744989.375	1707100.375	1707982.225
52	rewrite	1633049.375	1704888.125	1630035.875	1661479.25	1633250.125	1652540.55
53	read	2065842.75	1917189	2060306.25	2077235.25	1985462.25	2021207.1
54	re-read	2032750.625	2119037.25	2280733	2118177	1999995.875	2110138.75
55	stride read	2022439.25	2083451.375	2163908.75	2113629.75	2267044	2130094.625
56	random read	2066388.25	2065609.875	2029622.125	2041612.125	1897895.75	2020225.625
57	random write	1497927.5	1593665.625	1602195.875	1612524.25	1702173.5	1601697.35
58	8m						
59	initial write	1957116.125	1914441.75	2110537.25	1982492.625	1915798.5	1976077.25
60	rewrite	1852909.25	1858682.875	1895027.875	1921292.125	1878889	1881360.225
61	read	2462759.25	2467440.25	2522252.75	2564400.75	2444130.75	2492196.75
62	re-read	2443713.75	2422583.25	2543519	2558676.25	2491750.75	2492048.6
63	stride read	2465839.75	2413221	2383176.25	2598844	2429543.5	2458124.9
64	random read	2468904.5	2329196.5	2518950.5	2540222.5	2415417.25	2454538.25
65	random write	1901216	1892153.125	1937411.875	1945189.875	1846370.375	1904468.25
66	16m						
67	initial write	1797285.25	1842364.75	1854790.25	1869013.625	1865492.125	1845789.2
68	rewrite	1720469	1851247.25	1868523.75	1849020.5	1853734.75	1828599.05
69	read	2246677.75	2206725	2216141	2219725.75	2252376.75	2228329.25
70	re-read	2201670	2193201.75	2176905	2231977.5	2206557.75	2202062.4
71	stride read	2167894.75	2208263	2188837.75	2297625	2169706.5	2206465.4
72	random read	2201526.75	2187984	2158900.25	2214236.5	2195924	2191714.3
73	random write	1855286.25	1881882.875	1900017.75	1913749.5	1878491.75	1885885.625
74							
75							

위 화면은 noop scheduler 를 사용해 측정한 값입니다.



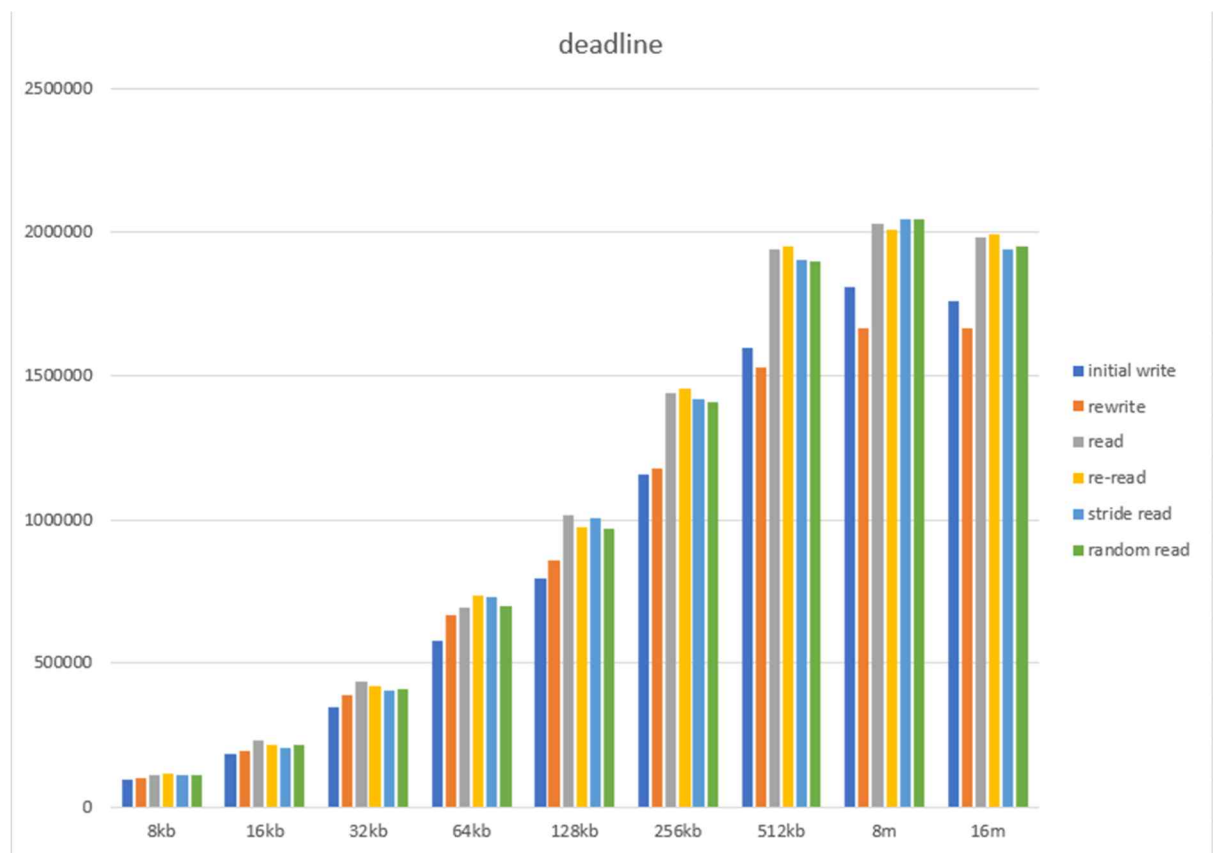
위 화면은 noop scheduler 의 실험을 5 회 이상 실시하여 얻은 평균값을 바탕으로 그래프를 나타낸 것입니다.

먼저 왼쪽부터 7 개씩 여백을 기준으로 8KB / 16KB / 32KB / 64KB / 128KB / 256KB / 512KB / 8MB / 16MB 의 record size 를 변경하여 진행 후 얻은 평균 값입니다. 여백으로 분리된 7 개의 그래프는 순서대로 initial write, rewrite, read, re-read, stride read, random read, random write 의 평균 값입니다. 먼저 record size 가 커질수록 대체로 성능이 좋은 모습을 확인할 수 있었으며 8M 일때는 16M 일 때 보다 좋은 성능을 보입니다. Noop scheduler 는 Random access 를 하는 device 를 위한 scheduler 이기 때문에 re-read 보다 더 좋은 결과를 내는 경우도 발생하는 모습을 볼 수 있습니다.

1	record크기	1차	2차	3차	4차	5차	평균			
2	8k									
3	initial write	94929.6875	93868.99219	94457.85156	93820.10156	99389.26563	95293.17969			
4	rewrite	100589.2813	99464.125	102408.0156	101997.4609	104427.1563	101777.2078			
5	read	109398.9922	111590.4141	110447.3594	111376.2422	118246.4609	112211.8938			
6	re-read	108541.4297	110430.3672	121903.6875	112218.8828	126317.375	115882.3484			
7	stride read	108222.1797	107426.5	105829.1719	107544.7734	122768.1875	110358.1625			
8	random read	124901.5547	104338.625	105246.7969	108045.3828	109834.3984	110473.3516			
9	random write	96506.0625	95708.13281	95108.34375	95607.52344	96909.25	95967.8625			
10	16k									
11	initial write	187906.6563	180736.8906	184412.4219	188789.2344	185581.4531	185485.3313			
12	rewrite	190146.4688	193355.8594	206542.7813	196785.6406	200498.5156	197465.8531			
13	read	211081.875	205940.1563	253472.3906	217609.6094	258254.1875	229271.6438			
14	re-read	215469.5	208788.6719	211579.125	215120.6719	218162.4219	213824.0781			
15	stride read	203299.5938	201356.7656	208009.0313	207801.9375	209705.5625	206034.5781			
16	random read	201489.4688	201822.0625	206746	206293.9375	257883.5156	214846.9969			
17	random write	204905.7031	182890.8594	189031.0156	187394.1875	189397.375	190723.8281			
18	32k									
19	initial write	353120.9063	339494.0625	353448.5	352888.3438	332906.3125	346371.625			
20	rewrite	447562.9688	366891.2813	379681.4063	378726	361841.3125	386940.5938			
21	read	408694.5	407265.0625	503370.6563	411444.4063	437267.5938	433608.4438			
22	re-read	437203.0313	408650.375	414095.375	435984.7188	399058	418998.3			
23	stride read	427213.1875	390279.7813	395539.0313	395048.4688	417172.0313	405050.5			
24	random read	427748.9063	373941.4375	382595.75	397936.875	456270.7813	407698.75			
25	random write	355458.2813	351906.7813	350392.4375	336999.1875	328217.6563	344594.8688			
26	64k									
27	initial write	581456.4375	581298.1875	567496.3125	580822	576962.6875	577607.125			
28	rewrite	654658.625	651826.25	696244.3125	700716.375	624262.375	665541.5875			
29	read	666500.5625	703786.1875	696735.6875	702625.0625	691684.25	692266.35			
30	re-read	733039.25	845581.4375	696892.6875	687869.5	712640.6875	735204.7125			
31	stride read	679527.875	753407.5	678800.25	670020.875	860490.0625	728449.3125			
32	random read	707746.5	803915.5	661798.4375	652263.6875	662620.625	697668.95			
33	random write	618756.9375	711638.0625	641093.625	604822.5625	603959.625	636054.1625			
34	128k									
35	initial write	802602.5	778234.4375	792052.9375	806882.4375	797136.5625	795381.775			
36	rewrite	844159.3125	847978.8125	824754.875	810709.0625	968115.8125	859143.575			
37	read	986178.5	1051423.25	959051.375	982116.125	1096633.875	1015080.625			
38	re-read	941227.9375	1007323.188	963819.5625	980643.875	970158.875	972634.6875			

39	stride read	1104954.125	978964.8125	919663.8125	974069.25	1037904.25	1003111.25												
40	random read	945452.6875	991114.1875	926765.75	903415.9375	1069146.375	967178.9875												
41	random write	786690.0625	841703.8125	790211.5625	801419.1875	884025.875	820810.1												
42	256k																		
43	initial write	1138005	1184321.5	1145382.5	1158771.875	1154275.375	1156151.25												
44	rewrite	1183282.125	1172682.375	1188832.25	1151682.625	1202185.25	1179732.925												
45	read	1435786.125	1409045.125	1445233.125	1452673.25	1455815.75	1439710.675												
46	re-read	1500944.125	1439204.625	1405302.375	1432371.625	1484780.75	1452520.7												
47	stride read	1423152.625	1448249.5	1383497	1403004.375	1430715.625	1417723.825												
48	random read	1420457.75	1426395.625	1362417	1400127	1420967.125	1406072.9												
49	random write	1164486.875	1123958.625	1124315.5	1143938.25	1120365.875	1135413.025												
50	512k																		
51	initial write	1622070.125	1576372.25	1587433.125	1599936.75	1601202.5	1597402.95												
52	rewrite	1599712.125	1555365.125	1472487.75	1510046	1513391.375	1530200.475												
53	read	1943257.75	1964932.25	1990462.75	1871217.125	1942929.75	1942559.925												
54	re-read	1981301.25	1957390.5	1935091.5	1949486.375	1934734.875	1951600.9												
55	stride read	1896745.75	1930791.25	1881880.5	1861127.375	1938923.625	1901893.7												
56	random read	1904736	1949598.625	1803374.75	1908941.625	1919526.875	1897235.575												
57	random write	1558639.375	1516945	1527434.25	1540892.75	1532649.625	1535312.2												
58	8m																		
59	initial write	1814870.75	1772219.125	1787813.75	1846230.875	1836616.75	1811550.25												
60	rewrite	1695010.375	1649649.125	1721036.5	1645710.25	1599426.5	1662166.55												
61	read	1911360	2086760	2049380.75	2142237	1959757.625	2029899.075												
62	re-read	1707016.875	2094748	2043009.125	2067601.625	2123612	2007197.525												
63	stride read	2023008.625	2138440.5	1921763.875	2073981.125	2071944.875	2045827.8												
64	random read	1912416.375	2156286	1999321.25	2061869	2097051.25	2045388.775												
65	random write	1677707.625	1720576	1705193.5	1686233.875	1707172.5	1699376.7												
66	16m																		
67	initial write	1753419.125	1757202.875	1779542.625	1766329.375	1727398.75	1756778.55												
68	rewrite	1594554.875	1686483.5	1674258.25	1693592.75	1665074.75	1662792.825												
69	read	1919653.5	1995326.625	2045770.75	1959926.25	1997839.125	1983703.25												
70	re-read	1923209.375	2019584.5	2020233.875	2003151.375	1993301.75	1991896.175												
71	stride read	1843248	2010122.25	1887618.375	1980354.75	1979461.625	1940161												
72	random read	1849066.375	1980683.875	1981862.875	1939290.5	1995319.375	1949244.6												
73	random write	1646731.875	1727108.375	1725331.875	1660860.625	1733057.25	1698618												
74																			

위 화면은 deadline scheduler 을 사용해 측정한 값입니다.



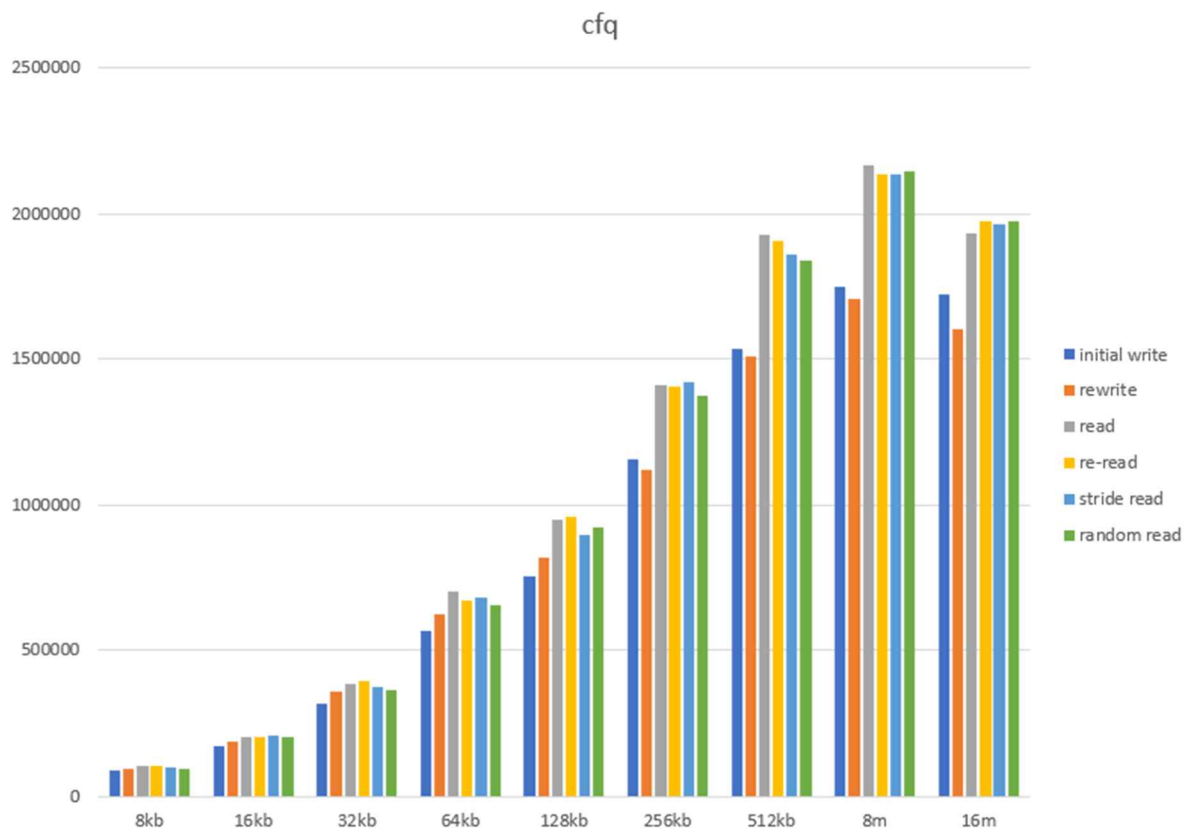
위 화면은 deadline scheduler 의 실험을 5 회 이상 실시하여 얻은 평균값을 바탕으로 그래프를 나타낸 것입니다. Record size 가 증가함에 따라 성능이 대체로

좋아지는 모습을 이 그래프에서도 확인할 수 있었으며 deadline scheduler 는 512K 부터 거의 성능이 크게 증가하지 않고 거의 비슷한 모습을 보였습니다.

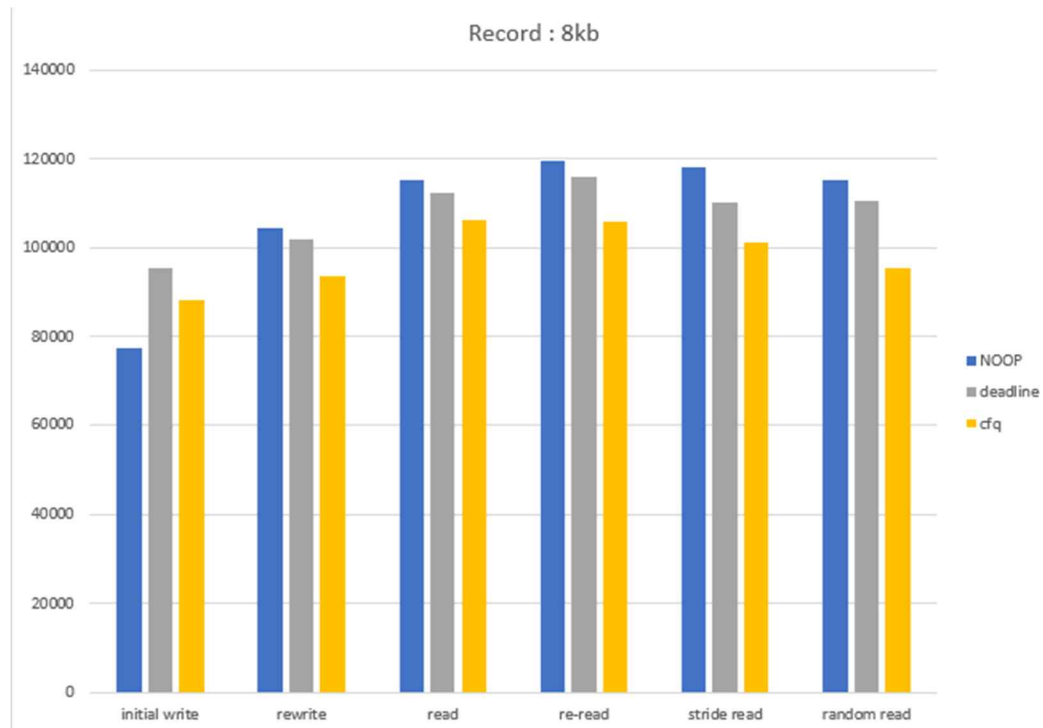
record크기	1차	2차	3차	4차	5차	평균				
8k										
initial write	84947.3125	87717.71094	88521.46094	86186.67969	92997.875	88074.20781				
rewrite	92422.92969	88901.29688	92632.15625	96844.04688	97846.65625	93729.41719				
read	101345.3984	97690.17969	110787.4766	106070.2656	115879.7969	106354.6234				
re-read	102314.3984	98187.44531	98422.85938	114608.1016	115652.0234	105836.9656				
stride read	100652.6016	99431.13281	99729.82031	96534.32813	109336.2891	101136.8344				
random read	95424.32813	94720.17188	95973.8125	94797.21875	96641.44531	95511.39531				
random write	85590.88281	86460.79688	87332.60156	94617.77344	88836.49219	88567.70938				
16k										
initial write	169489.7031	171603.9844	174532.3438	168098.6563	170440.25	170832.9875				
rewrite	189538.3281	176572.8125	221057.1563	178829.3906	181164.1719	189432.3719				
read	194933.6563	212415.1563	209166.7344	198740.2813	210813.1563	205213.7969				
re-read	193319.0781	217181.0313	209930.9219	198454.2656	212181	206213.2594				
stride read	226171.7813	218771.8438	200037.7344	190445.9219	212842.4063	209653.9375				
random read	189109.7188	232832.8438	199204.6406	197827.3281	189649.4844	201724.8031				
random write	172785.5625	180521.6406	176495.3906	171416.1094	170516.9375	174347.1281				
32k										
initial write	322303.1563	293769.1875	350951.5	294606.625	320872.1875	316500.5313				
rewrite	380315.7188	366361.5625	345364.7188	346029.75	346968.9688	357008.1438				
read	418207.0625	357069	380908.2188	371738.8125	397149.0625	385014.4313				
re-read	406665.4688	366141.9375	425906.7188	397805.2188	373974.5625	394098.7813				
stride read	400984.1563	355724.7813	380914.1875	359279.2813	381885	375757.4813				
random read	360013.2188	347363.8125	377616.7188	354606.8438	380916.9375	364103.5063				
random write	317154.4375	316644.25	325383.6875	318577.3438	329231.4063	321398.225				
64k										
initial write	598819.0625	552244.25	560578.75	563834.75	560976.4375	567290.65				
rewrite	605516.8125	697171.0625	595607.25	610298.9375	596673.0625	621053.425				
read	686321.1875	672504.625	710985.6875	805503.3125	639250.5	702913.0625				
re-read	674441.1875	659762.1875	689480.75	659453.9375	667342.1875	670096.05				
stride read	657627.625	708286.3125	702794.25	658447.25	664389.8125	678309.05				
random read	647367.8125	667968.25	650247	650686	657628.4375	654779.5				
random write	592755.625	587369.125	587502.375	593789.1875	591033.75	590490.0125				
128k										
initial write	734648.25	722849.3125	752051.1875	747132.6875	809028.9375	753142.075				
rewrite	918016.5625	790821.8125	794323.3125	794689.4375	815863.3125	822742.8875				
read	1022116.25	859651.3125	959352.375	937551.0625	962750.75	948284.35				
re-read	1028143.563	925468.3125	993912.875	907923.0625	933869.75	957863.5125				

37	read	1022116.25	859651.3125	959352.375	937551.0625	962750.75	948284.35	
38	re-read	1028143.563	925468.3125	993912.875	907923.0625	933869.75	957863.5125	
39	stride read	910085.5	861299.9375	879213.125	923295	916944.9375	898167.7	
40	random read	936435.1875	881326.625	950314.375	939907.75	907820.9375	923160.975	
41	random write	774925.6875	754018.625	768695.625	756472.375	780734.6875	766969.4	
42	256k							
43	initial write	1228218.875	119613.25	1123518.25	1214165	1110457.5	1159194.575	
44	rewrite	1029194.25	1151468.625	1143666.125	1174789.625	1127238.125	1121271.35	
45	read	1381039.375	1415538.875	1425255.625	1442864.875	1379616.25	1408863	
46	re-read	1385494.875	1389842.125	1433997.375	1444413	1387658.75	1408281.225	
47	stride read	1382741.875	1353077.875	1470967.375	1427125.5	1475944.75	1421971.475	
48	random read	1377841.625	1350417.25	1333897.875	1358640.875	1459824.5	1376124.425	
49	random write	1184798.25	1214592.5	1140154.25	1119554.5	1153551.5	1162530.2	
50	512k							
51	initial write	1602964.375	1516583	1589371	1483367.625	1490502.625	1536557.725	
52	rewrite	1555107.25	1491906.625	1481814.25	1517864.875	1497379.25	1508814.45	
53	read	1885643.625	1990316	1916610.875	1921898.25	1933729.375	1929639.625	
54	re-read	1971315	1868979.5	1925776.875	1853138	1914091.75	1906660.225	
55	stride read	1840271.75	1834878.125	1904036.75	1840462.75	1884660.25	1860861.925	
56	random read	1838596.875	1860427.5	1841881.75	1809206.375	1863077.875	1842638.075	
57	random write	1463077.75	1463169.75	1477960.5	1474064.5	1382049.25	1452064.35	
58	8m							
59	initial write	1816458.25	1665818.375	1750407.25	1729404.5	1783259.25	1749069.525	
60	rewrite	1775748.125	1746480	1690166.625	1672674	1655603.125	1708134.375	
61	read	2341820.75	2129914.25	2239828.75	2043386.875	2070794.75	2165149.075	
62	re-read	2320038	2124236.5	2124554.75	2069140.5	2051204.5	2137834.85	
63	stride read	2245845	2175478.25	2119380.5	2077713.125	2052095.5	2134102.475	
64	random read	2295290.25	2199702.25	2124068.25	2053953.75	2055684.75	2145739.85	
65	random write	1810130.5	1650502.75	1746933.625	1707508.625	1697821.5	1722579.4	
66	16m							
67	initial write	1774039.125	1825830.125	1702466	1655380.125	1642637.5	1720070.575	
68	rewrite	1701203.625	1619670	1579796.875	1523397.5	1584394.75	1601692.55	
69	read	1983591.625	1949457	1856064.5	1943596.125	1945091.75	1935560.2	
70	re-read	1990656.5	1940797.5	1938851.875	1997264.75	2000607.25	1973635.575	
71	stride read	2034213.5	1956780	1954049.25	1876646.625	1998738	1964085.475	
72	random read	1993824.125	1955623.625	2010765.625	1930407.375	1995132.875	1977150.725	
73	random write	1741322.875	1560000.5	1596599.375	1544278.625	1672439.25	1622928.125	

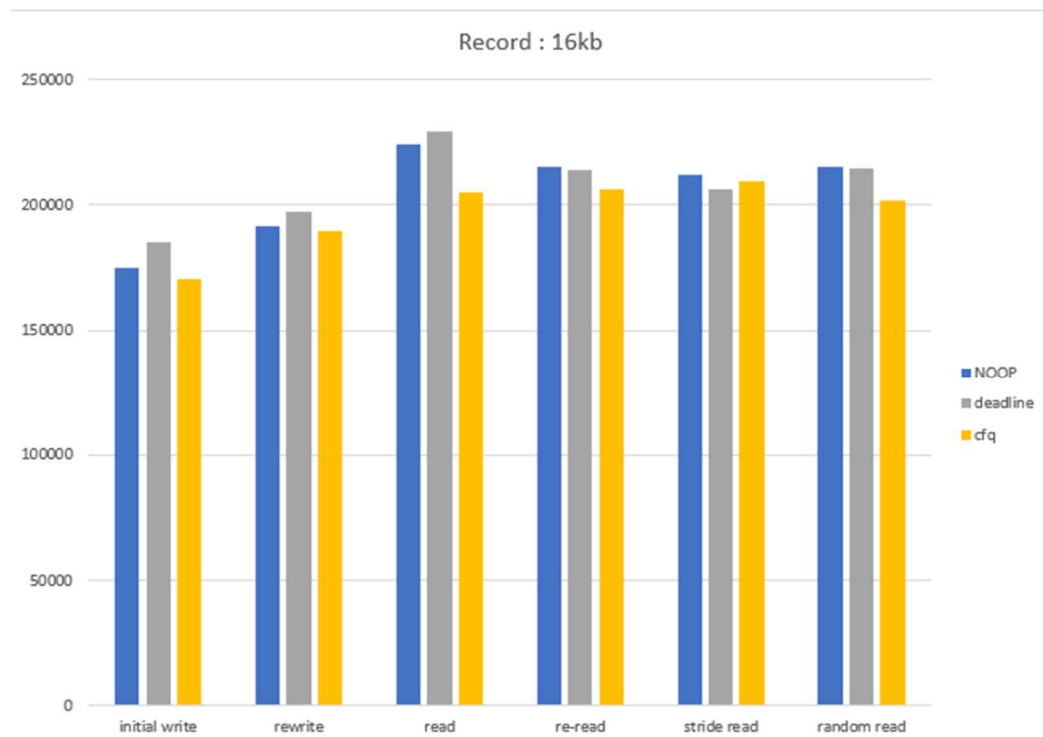
위 화면은 cfq scheduler 을 사용해 측정한 값입니다.



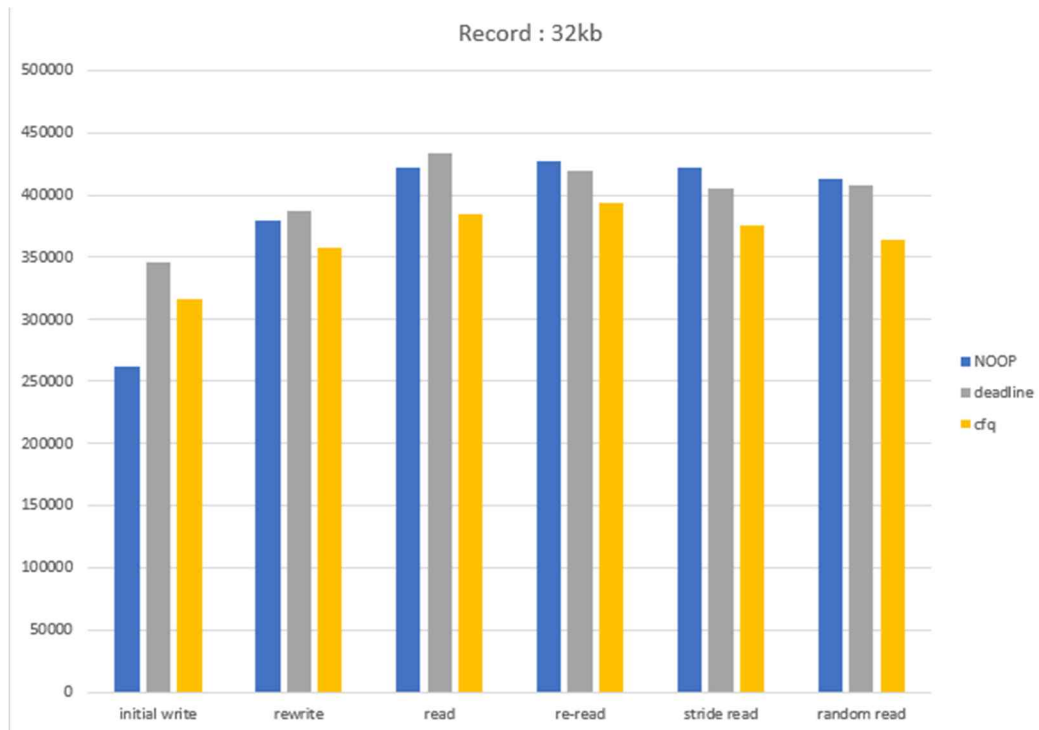
위 화면은 cfq scheduler 의 실험을 5 회 이상 실시하여 얻은 평균값을 바탕으로 그래프를 나타낸 것입니다. Cfq scheduler 또한 record size 가 증가함에 따라 성능이 대체로 좋아지며 이는 입출력을 요청하는 모든 프로세스들에 대해 디스크 I/O 대역폭을 공평하게 할당하는 것을 보장하는 기법이므로 record size 가 작을 때에는 성능 면에서 크게 차이 나지 않는 모습을 확인할 수 있습니다. 하지만 record size 가 커질수록 차이가 조금씩 나는 모습 또한 확인할 수 있습니다. 기존에 cfq scheduler 의 이론적 정의와는 다른 모습을 보이는 측면도 있었습니다. 공평하게 할당하여 오차범위가 비슷할 것 같다고 생각했지만 오차가 발생하는 모습이었습니다. 그리고 8M 일 때 16M 일 때 보다 성능이 좀 더 좋은 모습도 확인할 수 있습니다.



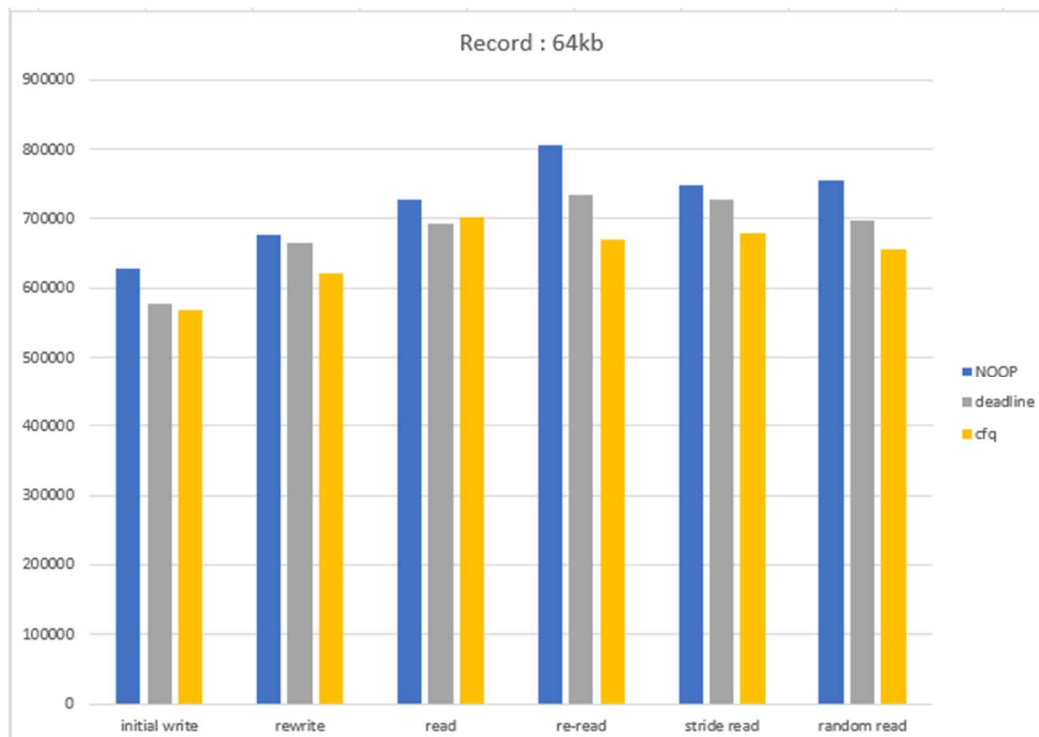
8KB : 대부분의 테스트 연산에서 noop 이 제일 좋은 성능을 보입니다.



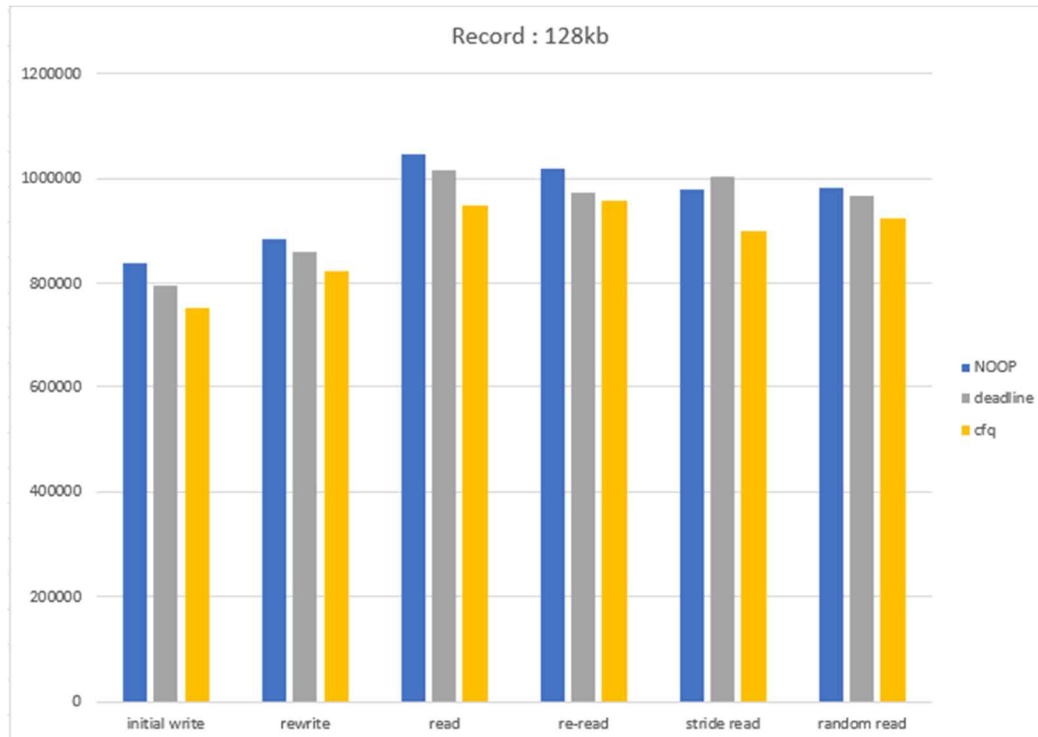
16 KB : 이전보다 오차가 크게 차이 나지 않는 것을 확인할 수 있습니다.



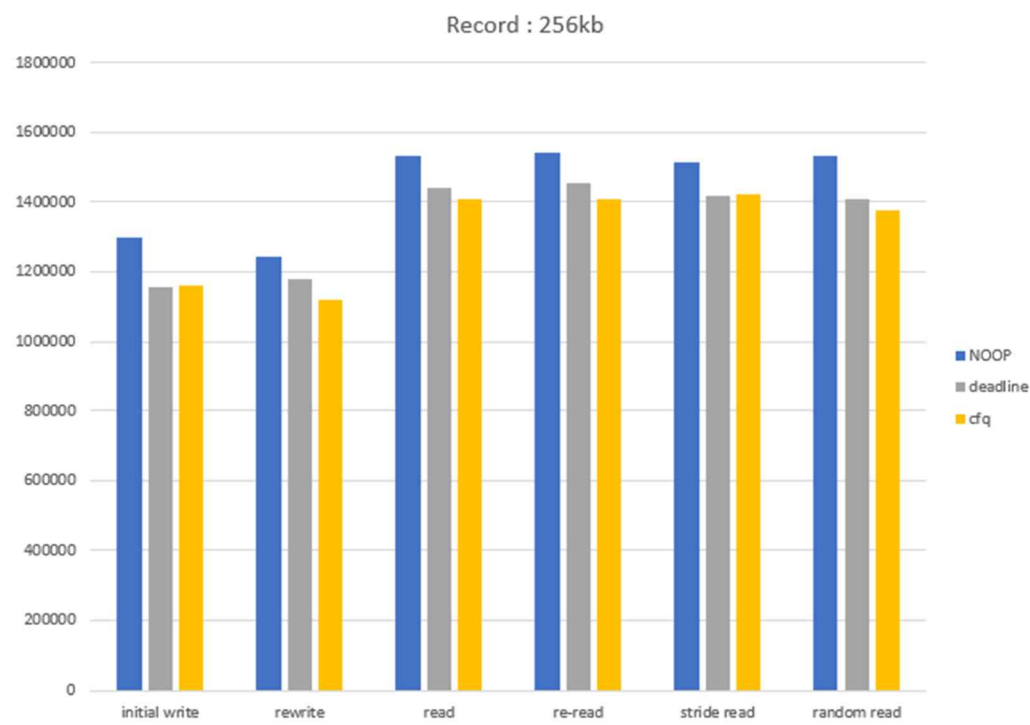
32KB : record size 가 커질수록 deadline 의 성능이 좋아지는 모습을 확인할 수 있습니다.



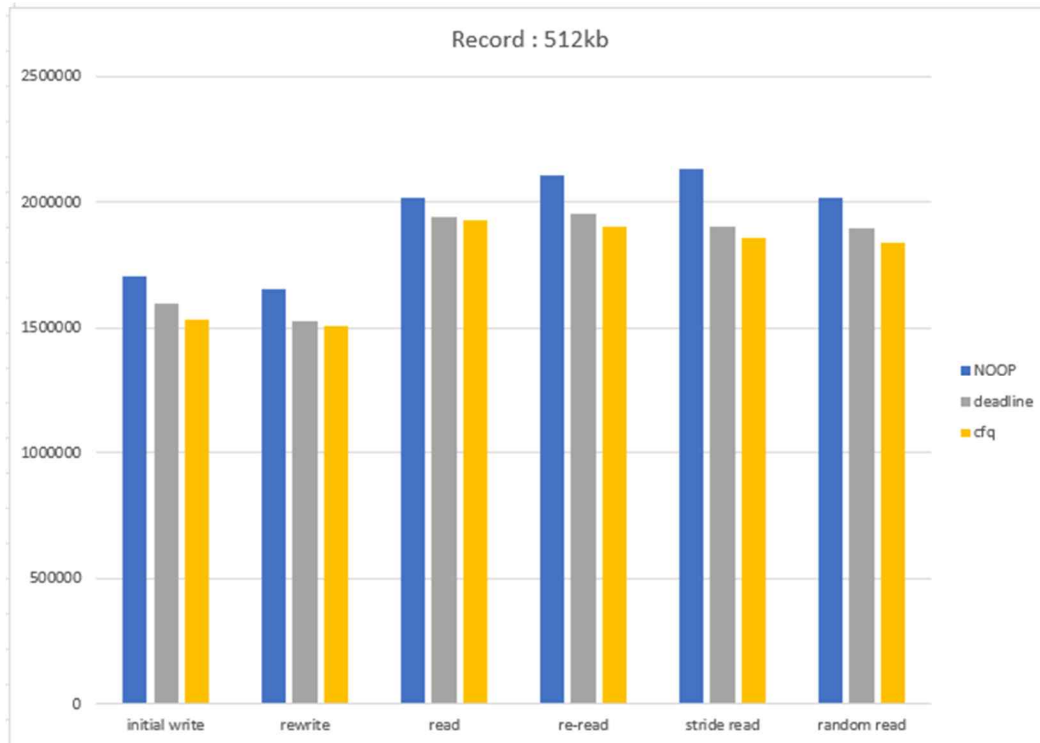
64KB : noop 에서 random access 에서의 성능이 점점 좋아지는 것을 확인할 수 있으면 cfq 가 deadline 보다 성능이 더 좋은 경우도 볼 수 있습니다.



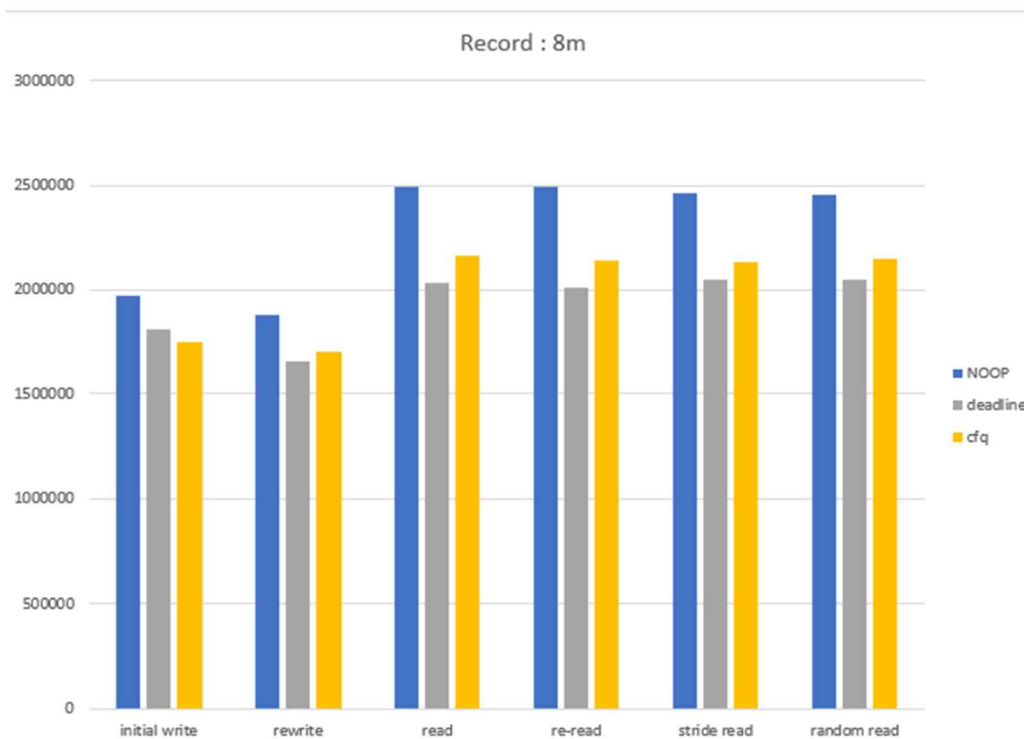
128KB : 큰 차이 없이 이전과 비슷한 성능을 확인할 수 있습니다.



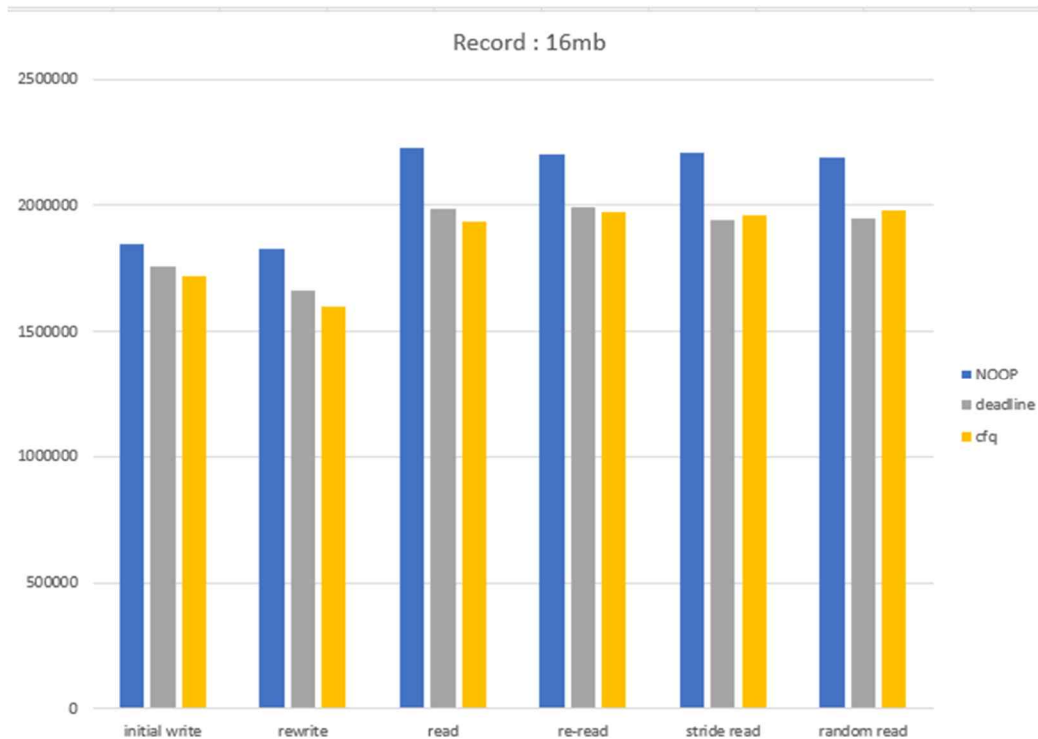
256KB : deadline 과 cfq 의 성능이 좋아지는 모습을 확인할 수 있습니다.



512KB : 세개의 scheduler 의 성능이 비슷해지는 모습을 확인할 수 있습니다.
 하지만 noop 이 가장 성능이 좋은 모습을 볼 수 있습니다.



8MB : 대부분 cfq 가 deadline 보다 성능이 좋은 모습을 확인할 수 있습니다.



16MB : deadline 과 cfq 가 성능이 비슷하고 noop 이 이들보다 성능이 좋은 모습을 확인할 수 있습니다.

고찰

이번 과제에서는 I/O Zone 을 이용한 세 가지 Linux I/O scheduler 성능 테스트를 진행하였는데 scheduler 3 개를 가지고 9 개의 record size 그리고 이를 5 번씩 진행하여 135 번 진행하였습니다. 처음에 과제를 시작할 때는 조금 막막하였지만 하다 보니 생각보다 빠르게 진행할 수 있었습니다. 과제를 하면서는 굳이 이렇게 많이 테스트할 이유가 있나 라는 생각도 들었지만 한편으로는 어떠한 실험을 하여 결과를 얻기 위해서는 이렇게 많은 데이터가 필요하구나 라는 생각을 했습니다. 더 나아가 이런 과제 수준의 실험이 아니라 정말 연구실에서 진행하는 실험에는 얼마나 많은 데이터 셋이 필요할까 라는 생각도 들었습니다.

실험적인 부분에서는 이번 실습 강의자료에서 처음 본거라 낯설었지만 이론적인 부분에서는 운영체제 강의시간에 다 배운 내용이었어서 크게 어려운 부분은 없었습니다. 이번 실험에서는 특히 캐시 버퍼를 지우는 것이 굉장히 중요한 것 같았다. 이 과정을 진행하지 않으면 실험 결과가 크게 달라질 것이고 이러한 결과는 의미가 없기 때문에 캐시 버퍼를 지우는 과정을 빼먹지 않으려 노력했습니다. 하지만 오차를 발생시키는 요소가 다른 부분에서 발생한 것

같았습니다. 예상과 다른 결과가 좀 있었는데 이러한 요소들을 완전히 제거하지 못한 부분들이 조금 아쉬웠습니다.

Reference

2023 년 2 학기 운영체제실습 13 주차 I/O Scheduler

2023 년 2 학기 운영체제 & 운영체제실습 Assignment 5