운영체제실습

assignment 5

학　　과: 컴퓨터정보공학부

담당교수: 김태석 교수님

학　　번: 2019202050

성　　명: 이강현

1. Introduction

이번 과제에서는 linux I/O scheduler 의 성능을 비교해본다. 비교해볼 scheduler 는 cfq,noop,deadline 세가지이고 벤치마크 iozone 을 이용해서 여러가지 테스트를 진행한 후 데이터의 전송속도를 비교해보며 성능을 비교할 수 있다.

2. Result

먼저 벤치마크를 이용해서 성능을 평가하기 위해서 scheduler 를 변경해본다.

```
os2019202050@ubuntu:~$ cat /sys/block/sda/queue/scheduler
noop deadline [cfq]
```

먼저 현재 scheduler 가 무엇인지 확인해본 결과 cfq 로 되어있었다.

이후 평가할 noop 와 deadline 으로 scheduler 를 변경하기 위해서 다음과 같은 명령어를 이용했다.

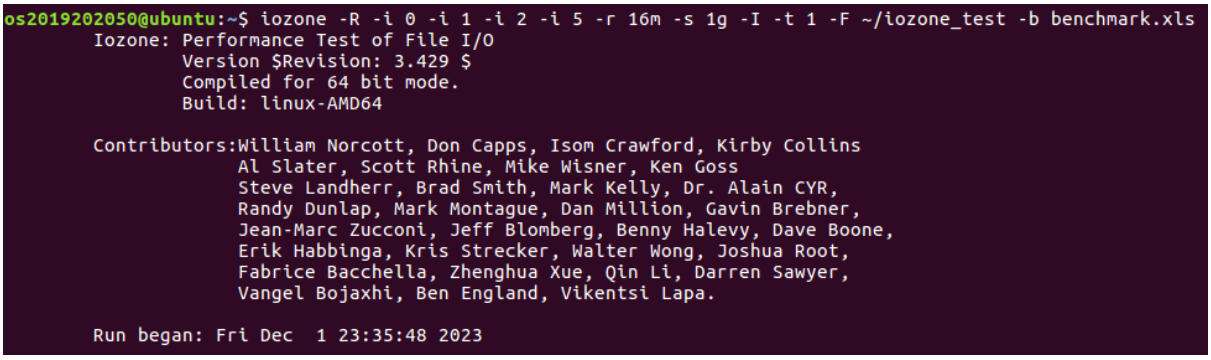
<noop 로 변경시>

```
os2019202050@ubuntu:~$ echo noop | sudo tee /sys/block/sda/queue/scheduler
[sudo] password for os2019202050:
noop
os2019202050@ubuntu:~$ cat  /sys/block/sda/queue/scheduler
[noop] deadline cfq
os2019202050@ubuntu:~$
```

<deadline 으로 변경시>

```
os2019202050@ubuntu:~$ echo deadline | sudo tee /sys/block/sda/queue/scheduler
deadline
os2019202050@ubuntu:~$ cat  /sys/block/sda/queue/scheduler
noop [deadline] cfq
os2019202050@ubuntu:~$
```

iozone 벤치마크를 실행시키기 위해 다음과 같은 명령어를 사용하였다.



-R 로 excel report 를 생성하게끔 하고 -i 옵션과 뒤에 인자로 어떤 테스트를 진행할지를 결정한다. 모든 scheduler 모두 0,1,2,5 총 4 가지의 테스트를 진행하게끔 하였다. 0 은 write/re-write 연산 1 은 read/re-read 2 는 random-read/write 5 는 strid-read 이다. 0,1,2 는 과제 조건에 따라 선택하였고 strid-read 를 택한 이유는 데이터는 랜덤하게 저장되어 있는 것보다 일반적으로 메모리 참조를 할때는 locality 라는 특성이 있기 때문에 데이터 순차접근을 많이 하게 될 것을 고려해서 strid-read 의 성능을 확인하고 싶어서 추가하였다.

이후 -r 옵션은 record size 를 변경하는 것으로 전체 파일 사이즈를 이 옵션에 의해 설정된 크기로 분할하여 테스트를 수행한다. 매 스케쥴러마다 테스트를 수행할 때 record size 를 8kb/16kb/32kb/64kb/128kb/256kb/512kb/8M/16M 으로 바꾸어가며 테스트한다.

-s 는 전체 파일 사이즈를 말하는 것으로 1g, 1 기가로 고정하였다.

-I 는 buffer cache 를 이용하지 않고 연산을 수행한다는 의미이다. 버퍼캐시를 이용한다면 버퍼에 의한 성능의 개선인지 스케쥴러의 성능이 좋은 것인지 알 수 없을 수 있으니 이를 배제한다.

-t 옵션과 인자 1 로 테스트에 사용할 thread or process 의 개수를 1 로 한다.

-F 옵션으로 thread or process 파일 이름을 정하고 -b 옵션과 결과가 저장되어 추출되는 엑셀파일의 이름을 명시하였다.

또한 iozone 을 실행하기 전 매 순간 버퍼에 의한 성능차이를 배제하기 위해 아래와 같은 작업을 반복하였다.

```
os2019202050@ubuntu:~$ rm -rf ~/iozone_test
os2019202050@ubuntu:~$ sync
os2019202050@ubuntu:~$ echo 3|sudo tee /proc/sys/vm/drop_caches
3
```

CFQ

8 kBytes

execute count	1	2	3	4	5	Average
Initial write	45532.6211	44596.1992	41290.5234	44671.8008	41790.9688	**43576.42**
Rewrite	45058.9531	45106.5	44840.6094	43935.9492	44801.7109	**44748.74**
Read	45420.2578	45248.1836	44926.5586	45527.4023	43977.3906	**45019.96**
Re-read	45431.1211	46573.6914	46833.5859	44906.875	45864.3125	**45921.92**
Random read	44907.3242	44978.9805	45591.1875	44215.7266	44686.1289	**44875.87**
Random write	44929.2344	45040.4375	44876.1602	46147.2188	46804.5156	**45559.51**
stride read	46744.9453	48686.9922	48504.9648	48392.8164	45483.5898	**47562.66**

16 kBytes

execute count	1	2	3	4	5	Average
Initial write	88307.1016	90380.5313	79440.2891	80763.9219	75665.9844	**82911.57**
Rewrite	87916.6641	88119.6953	85922.8203	85633.0938	84307.9063	**86380.04**
Read	88323.4453	86939.0234	89126.5469	87117.4609	88015.2578	**87904.35**
Re-read	88261.8359	86528.125	87923.0469	91043.875	91176.9375	**88986.76**
Random read	89140.2344	88810.3281	87464.5938	86957.1641	80959.0391	**86666.27**
Random write	90768.8125	93909.9453	93953.4609	89861.7656	88769.3672	**91452.67**
stride read	97473.7734	99392.2109	101295.398	98618.4453	101279.469	**99611.86**

32 kBytes

execute count	1	2	3	4	5	Average
Initial write	170695.734	131943.656	134547.25	100288.953	117492.852	**130993.7**
Rewrite	169882.938	164780.641	157244.672	165890.859	168025.875	**165165**
Read	168513.063	164438.328	167938.313	161734.453	175710.031	**167666.8**
Re-read	175615.219	160231.031	164848.406	164053.328	163510.047	**165651.6**
Random read	170080.266	171822.844	161323.297	169909.031	173464.453	**169320**
Random write	178249.469	163838.438	183948.5	166743	167566.281	**172069.1**
stride read	175011.547	182635.453	204240.047	181487.156	207252.094	**190125.3**

64 kBytes

execute count	1	2	3	4	5	Average
Initial write	261681.594	281599.281	270791.813	246500.047	238642.438	**259843**
Rewrite	322360.656	328401.156	315533.531	309303.5	311476.5	**317415.1**
Read	308539.188	304724.25	312756.438	316112.938	313183.063	**311063.2**
Re-read	303358.875	302705.313	288004.5	324612.031	301009.531	**303938.1**
Random read	304277.656	319762.781	299664.563	290284.719	312999.375	**305397.8**
Random write	316406.438	320272.031	338897.5	322560.781	342189.25	**328065.2**
stride read	310696.688	310631.031	288154.406	310399.594	301648.875	**304306.1**

Noop

Record size = 8 kBytes

execute count	1	2	3	4	5	Average
Initial write	84508.34	78923.7	81604.1	73334.52	87417.6	**81157.65**
Rewrite	76938.82	80977.65	85540.99	79123.34	80854.46	**80687.05**
Read	75682.41	80011.44	77761.32	78630.7	82554.63	**78928.1**
Re-read	78913.68	81057.87	79534.92	73412.19	80197.5	**78623.23**
Random read	76198.61	74739.38	74548.71	75274.73	76703.55	**75492.99**
Random write	77294.42	74444.32	73974.55	72768.31	76861.95	**75068.71**
stride read	74750.15	74352.53	71892.34	73464.43	74006.13	**73693.11**

Record size = 16 kBytes

execute count	1	2	3	4	5	Average
Initial write	143852.6	147624.7	167837.1	147436.6	165319.2	**154414**
Rewrite	133319.9	141431.2	161832.9	147918.6	162458.6	**149392.3**
Read	163371.6	148818.5	161730.2	142693.9	156367.5	**154596.3**
Re-read	136696.4	143131.1	142515.7	145982.5	150692.5	**143803.6**
Random read	142198.3	134457.4	139687.1	138024.9	162387.8	**143351.1**
Random write	153705.1	137302	144203.3	158180.1	128440.3	**144366.2**
stride read	150446.2	134305.2	132585.6	149438.6	135249.4	**140405**

Record size = 32 kBytes

execute count	1	2	3	4	5	Average
Initial write	332180.5	293085.2	282646.6	282071.6	283395.1	**294675.8**
Rewrite	301526.9	285719.4	301751.3	295022.5	288025.6	**294409.1**
Read	300612.9	302602.8	297270.2	293509.8	277641.7	**294327.5**
Re-read	286509.9	297457.5	285986.9	280013	277876.1	**285568.7**
Random read	261121	279027.8	255320	243220.1	252789.2	**258295.6**
Random write	305547.6	269699.8	240603.3	264264.1	274700.4	**270963**
stride read	271067.9	273717.4	241183.7	250556.1	260843.2	**259473.6**

Record size = 64 kBytes

execute count	1	2	3	4	5	Average
Initial write	499184.9	490045.5	493448	486239.1	490272.6	**491838**
Rewrite	518767.8	569022.2	530913.3	537035.1	521618.1	**535471.3**
Read	548786.1	518259.3	514497.6	509859.8	555457.3	**529372**
Re-read	512569.8	502658.1	536608	566658.2	525771.8	**528853.2**
Random read	501204.8	564020.9	468278.2	489435.1	548166.4	**514221.1**
Random write	481448.7	467734.8	512884.1	537824.2	469783.3	**493935**
stride read	492568.3	504407.7	502576.7	538367.6	513262.8	**510236.6**

Deadline

Record size = 8 kBytes

execute count	1	2	3	4	5	Average
Initial write	75632.02	79792.96	79027.69	79074.11	79330.12	**78571.38**
Rewrite	84644.48	78733.16	81796.64	78293.76	79957.47	**80685.1**
Read	78826.56	78414.19	84318.4	75301.51	77904.51	**78953.03**
Re-read	79785.67	80567.33	77297.95	71846.58	78487.39	**77596.98**
Random read	80105.91	83585.78	67913.79	81485.53	82886	**79195.4**
Random write	74422.56	80024.64	79754.32	80467.75	82155.35	**79364.93**
stride read	77032.48	81973.28	73847.88	79053.73	82469.6	**78875.4**

Record size = 16 kBytes

execute count	1	2	3	4	5	Average
Initial write	152892.1	152403.8	149397.9	153497.4	149767.1	**151591.7**
Rewrite	162329.6	150464	154962.4	156335.3	156904.7	**156199.2**
Read	152117.3	146999.1	145619.1	147073.5	148873.8	**153536.5**
Re-read	159626.5	157600.5	157676.5	147609.9	144219.1	**153346.5**
Random read	144721.1	151447.8	143227.4	141358.2	139447.4	**144040.4**
Random write	139814.4	144939.4	144385.1	144784.5	151259.8	**145036.7**
stride read	140846.4	148265.5	144390.2	141447.2	143455.7	**143681**

Record size = 32 kBytes

execute count	1	2	3	4	5	Average
Initial write	278003.1	291509.8	273952.1	276172.3	284404.9	**280808.4**
Rewrite	321200.6	291238.1	281266.8	295087.2	295802.7	**296919.1**
Read	291703.7	283351.8	278312.3	267790	263728.2	**276977.2**
Re-read	302922.9	299446.3	277848.3	276930.9	309235.1	**293276.7**
Random read	276016	327347.5	264435.3	260512.3	271358.2	**279933.9**
Random write	270169.3	260721.6	284302.3	265503.8	266144.6	**269368.3**
stride read	273652.3	289317.3	276765.4	258873.4	269866.1	**273694.9**

Record size = 64 kBytes

execute count	1	2	3	4	5	Average
Initial write	497547.3	493122.4	578760.6	485385.1	505433.3	**512049.7**
Rewrite	572374.9	506226.8	553487.8	514854.8	540742.5	**537537.4**
Read	591023.7	531981.2	548579.4	564891.9	476827.2	**542660.7**
Re-read	513916.8	505604.1	500503.8	532615.9	508866.5	**512301.5**
Random read	539948.4	545006.1	515101.9	480855.9	540346	**524251.7**
Random write	503714.1	539720.8	486424.9	499675.5	437612.9	**493429.6**
stride read	522065.6	552572.2	467985.2	506764.5	470812.8	**504040.1**

128 kBytes

execute count	1	2	3	4	5	Average
Initial write	557519.813	715346.188	706139.375	716815.25	706647.125	**680493.6**
Rewrite	561844.5	720826.25	687470.375	744309.063	730733.375	**689036.7**
Read	566859.125	736386.438	838347.688	775918.75	756058.438	**734714.1**
Re-read	556286.438	753366.875	734068.75	756513.375	681864.5	**696420**
Random read	525301.625	727502.75	756469.688	788293.375	697560.188	**699025.5**
Random write	479218.031	722574.75	665571.625	735198.875	745973.875	**669707.4**
stride read	626979	633668.813	613472.813	640461.938	689004.938	**640717.5**

Record size = 128 kBytes

execute count	1	2	3	4	5	Average
Initial write	789856.8	761830.6	725483.9	730614.8	677345.3	**737026.3**
Rewrite	724798.9	741717.1	737394.1	754116.8	709954.1	**733596.2**
Read	765568.4	748814.9	732302.1	777513.7	826616.3	**770163.1**
Re-read	723368.9	867527.4	805020.3	765369.5	731834.4	**778624.1**
Random read	812679.4	755083.4	751186.4	718948.7	794451.6	**766469.9**
Random write	692292	727925.6	716242.9	706942.9	753257.6	**719332.2**
stride read	724789.2	766078.5	751394	718650.3	802737	**752729.8**

Record size = 128 kBytes

execute count	1	2	3	4	5	Average
Initial write	741349.6	746347.8	757257	689244.1	691767.3	**725193.1**
Rewrite	739336.8	727307.9	726392.1	759050.5	749863.3	**740390.1**
Read	791067.1	749621.6	803813.8	782051.4	816206.9	**788552.2**
Re-read	795622.1	728535.4	758319.4	790207.4	756635.8	**765864**
Random read	737467.3	818083.5	718359.2	731966.6	729308.2	**747037**
Random write	787879.9	731794.1	749702.6	727288.8	710623.8	**741457.9**
stride read	788613.7	768128.9	741570.9	758347.6	739705.5	**759273.3**

256 kBytes

execute count	1	2	3	4	5	Average
Initial write	1108703.13	1095876.13	1154086.13	1033162.19	1066467.25	**1091659**
Rewrite	1152480	872304.125	1082855	1024211.81	1101876	**1046745**
Read	1258337.13	1201442.75	1215612.13	1077284.38	1160882	**1182712**
Re-read	1166862.13	1215427.25	1166673.88	1188014.38	1176315.75	**1182659**
Random read	1201597	1115776.88	1197657	1057674.13	1052991.13	**1125139**
Random write	1100083.25	1014022.19	1020158.38	1029334.69	1084178.13	**1049555**
stride read	1080008.5	1161363.13	1066880.44	1063716.81	1108393.81	**1096073**

Record size = 256 kBytes

execute count	1	2	3	4	5	Average
Initial write	1235829	1148626	1126207	1165508	1105821	**1156398**
Rewrite	974547.4	1139597	1161956	1102396	1174112	**1110522**
Read	1235696	1145265	1236871	1227389	1303810	**1229806**
Re-read	1174403	1167249	1221580	1195932	1191062	**1190045**
Random read	1182111	1113752	1159318	1164480	1126319	**1149196**
Random write	1062546	1106001	1032917	1142740	1052616	**1079364**
stride read	1113687	1109880	1107509	1170503	1106741	**1121664**

Record size = 256 kBytes

execute count	1	2	3	4	5	Average
Initial write	1111858	1083364	1132284	1085786	1137818	**1110222**
Rewrite	1101932	1155810	1088116	1189338	1109784	**1128996**
Read	1264509	1250936	1203670	1263156	1302160	**1256886**
Re-read	1341594	1166587	1282941	1162166	1301942	**1251046**
Random read	1179475	1311988	1160209	1257002	1231238	**1227982**
Random write	1108121	1109560	1102280	1142879	1098562	**1112280**
stride read	1213685	1231224	1176137	1222134	1213703	**1211376**

512 kBytes

execute count	1	2	3	4	5	Average
Initial write	1519129.5	1531096.38	1582145.13	1577591.25	1528875.38	**1547768**
Rewrite	1567153.5	1622657.88	1484651	1565971.63	1480049	**1544097**
Read	1579526.13	1473708.38	1769298.13	1669650.75	1537731.88	**1605983**
Re-read	1586468.25	1577004.75	1589985.25	1612179.63	1710330	**1615194**
Random read	1195627.63	1479518.25	1656991.63	1784542.88	1615985.63	**1546533**
Random write	1473710.38	1464528.13	1377405.25	1457183.63	1474025.5	**1449371**
stride read	1559591.25	1545286.88	1370107.25	1500714.25	1393689.75	**1473878**

Record size = 512 kBytes

execute count	1	2	3	4	5	Average
Initial write	1568424	1623577	1549120	1484631	1528633	**1550877**
Rewrite	1498415	1678605	1386908	1439938	1492765	**1499326**
Read	1775514	1760091	1688283	1729636	1706814	**1732068**
Re-read	1653090	1786151	1694096	1812218	1624093	**1713929**
Random read	1701571	1644116	1657167	1737964	1695463	**1687256**
Random write	1562511	1464930	1432471	1436144	1493749	**1477961**
stride read	1672335	1572514	1591342	1680177	1625348	**1628343**

Record size = 512 kBytes

execute count	1	2	3	4	5	Average
Initial write	1607540	1559189	1560650	1452583	1549646	**1545922**
Rewrite	1566727	1467889	1560263	1506101	1581512	**1536498**
Read	1682727	1759243	1753838	1602619	1707023	**1701090**
Re-read	1682498	1717913	1667434	1574460	1727191	**1673899**
Random read	1694655	1678356	1690385	1525534	1622226	**1642231**
Random write	1645721	1485637	1505470	1425511	1456295	**1503727**
stride read	1704090	1633602	1625812	1503946	1574871	**1608464**

8192 kBytes

execute count	1	2	3	4	5	Average
Initial write	1954493.88	1867674.63	2004906.5	1905078.5	1931321.13	**1932695**
Rewrite	2025248.13	1885412.5	1996804.13	1920398.63	1940650.13	**1953703**
Read	2099058	2028542.75	2027965.63	1902824.13	2031408	**2017960**
Re-read	2082966.88	2078886.75	2145257	1822114.5	2021709.5	**2030187**
Random read	2153049.5	1976998.63	2140689.75	1635043.13	1890555.75	**1959267**
Random write	2003278.13	2038702	2019584.5	1937176.38	1853944.25	**1970537**
stride read	2049222.38	2044168.75	1948138.5	2042290	2000390.25	**2016842**

Record size = 8192 kBytes

execute count	1	2	3	4	5	Average
Initial write	1988550	2009868	2057024	2148167	2141314	**2068984**
Rewrite	1990259	2047549	2133351	1901630	2109348	**2036427**
Read	2233331	2605716	2408205	2153779	2513148	**2382836**
Re-read	2446663	2470493	2311196	2230638	2473890	**2386576**
Random read	2284306	2056862	2320141	2372559	2318827	**2270539**
Random write	2021421	2011386	2099840	2025784	2192279	**2070142**
stride read	2286738	2190343	2289370	2226532	2375868	**2273770**

Record size = 8192 kBytes

execute count	1	2	3	4	5	Average
Initial write	2163740	1981669	2076618	1874623	2057786	**2030887**
Rewrite	2138341	2208049	2167599	2164870	2254155	**2186602**
Read	2503304	2551964	2357373	2482942	2655579	**2510232**
Re-read	2211192	2495813	2414611	2092553	2116885	**2266211**
Random read	2151468	2169074	2306490	2287285	2343161	**2251495**
Random write	2060329	2004163	2036477	2077528	2121919	**2060083**
stride read	2127751	2229061	2253848	2262438	2263146	**2227249**

16384 kBytes

execute count	1	2	3	4	5	Average
Initial write	1893498.88	1996550.38	1891489.88	2035445.88	1898586.38	**1943114**
Rewrite	2005401	1914574.25	1903007	2095208	2099276.5	**2003493**
Read	2184743	2092668.88	1937598.75	2232475	2124476.75	**2114392**
Re-read	1754979.5	2042868.63	1947835.5	1809409.63	2120122	**1935043**
Random read	2081888.13	1448919.88	1997911.75	2131845.75	2174674.75	**1967048**
Random write	1954671.88	1999161.38	1987041.88	2100559.75	2074322.5	**2023151**
stride read	1959963.75	1929990.63	1891423.13	1947937.75	1904677	**1926798**

Record size = 16384 kBytes

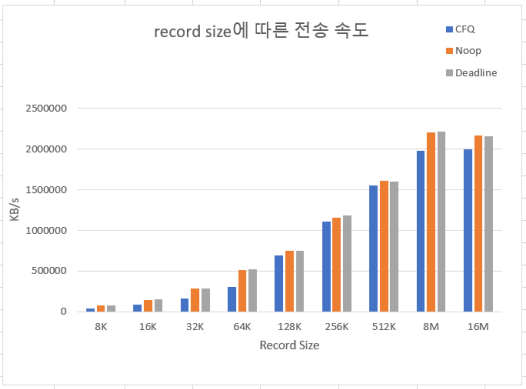
execute count	1	2	3	4	5	Average
Initial write	2067011	1960414	2003501	1993547	1912909	**1987476**
Rewrite	2054831	2026399	2182706	2026802	2134302	**2085008**
Read	2382889	2511144	2276381	2289607	2282814	**2348567**
Re-read	2095535	2397415	2120130	2253654	2099504	**2193248**
Random read	2275423	2267486	2129954	2330884	2312379	**2263225**
Random write	2169612	2099046	2097765	2142632	2150042	**2131819**
stride read	2262960	2340586	2150754	2335026	2302317	**2278329**

Record size = 16384 kBytes

execute count	1	2	3	4	5	Average
Initial write	2051084	2160770	2070087	1914060	1578872	**1954975**
Rewrite	2060577	2166135	2078282	2058806	2257377	**2124235**
Read	2397573	2279290	2486332	2343554	2511023	**2403554**
Re-read	2190782	2003906	2104920	2149616	2003321	**2090509**
Random read	2350539	2158487	2283023	2311533	2273553	**2275427**
Random write	2106226	2142789	2049934	2146052	2028959	**2094792**
stride read	2286678	2113073	2191967	2326357	2287977	**2241210**

위는 전체 테스트 결과이다 cfq,noop,deadline 세 가지 스케쥴러에 대해 각각 테스트들을 5 번씩 실행하여 평균을 내었다. 아래는 record size 에 따라 각 scheduler 들이 어떤 성능을 보이는지, 또 테스트별로 각 scheduler 들의 성능이 어떠한 지를 나타낸 그래프이다.

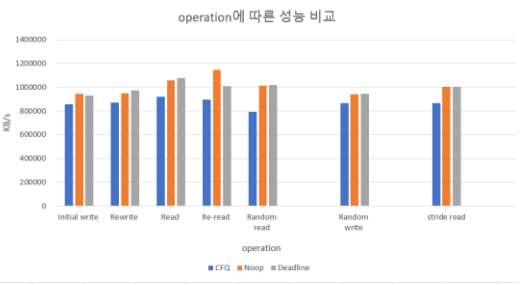
size	8K	16K	32K	64K	128K	256K	512K	8M	16M
CFQ	**44950.4**	**87383.61**	**161811.04**	**304287.1**	**694899.55**	**1113078**	**1551491**	**1977391**	**1997707**
Noop	**78326.29**	148320.6	283039.96	515615.1	750868.62	1152555	1610236	2202584	2168224
Deadline	79061.14	150625.2	282880.6	520371.8	751415.71	1181235	1600561	2217585	2157249



record size 를 비교할 때는 각 테스트들의 평균값을 record size 마다 뽑아내어 비교하였다. record size 가 높아질수록 모든 scheduler 에서 더 좋은 성능을 보였다. 이러한 현상은 8M 에서 가장 높았다가 그보다 더 record size 가 커진다면 오히려 낮아졌는데 이는 1G 에 해당하는 파일 사이즈를 8M 로 분할하여 처리할 때 가장 효율적임을 나타낸다.

record size 가 너무 작다면 한번에 처리할 수 있는 크기보다 더 작으니 그만큼 분할된 task 가 더 많아지기 때문에 성능이 낮은 것 같고 너무 크다면 한번에 파일을 분할할 때 하나의 record size 가 여러 블록에 걸쳐 저장되어 읽고 쓰는데 오히려 I/O 작업을 증가시킬 수 있고 또한 디스크 I/O 대역폭의 차이 때문에 성능이 감소할 수도 있다고 예상한다.

test	Initial write	Rewrite	Read	Re-read	Random read	Random write	stride read	평균
CFQ	857006	872309	918602	896000	790327.3288	866607.5975	866213	**866724**
Noop	946983	947204	1057852	1144433	1014227	940328	1004294	**1007903**
Deadline	932247	976451	1079160	1009339	1019066	944393	1005318	**995139**



여러가지 테스트를 진행한 결과 각 테스트들의 결과는 우위를 가리기 힘들었다. 캐시를 사용하지 않기에 write 와 re-write, read 와 re-read 의 성능차이가 그렇게 크게 나지 않은 것이라고 생각하고 확실히 random read-write 는 SSD 환경에서도 순차접근보다 랜덤 접근이 매핑된 데이터 테이블을 더 많이 확인해야 하기 때문에 더 오래 걸린 것이라고 예상한다. 추가적으로 진행한 strid-read 는 random-read 보다 압도적으로 높은 성능을 보여줄 것으로 예상했으나 cfq 에서 소폭 상승한 것을 제외하고 나머지 스케쥴러에서는 효과가 없었다.

전체적인 스케쥴러의 성능 비교시 차이가 많이 나지는 않지만 Noop>deadline>cfq 라는 결과를 얻을 수 있었다.

3. 고찰

1 개의 프로세스를 이용하여 테스트를 진행했기에 프로세스간 디스크 I/O 대역폭을 공평하게 할당하는 CFQ 방식은 오히려 프로세스를 위한 큐를 할당하고 정렬하는 것 자체가 overhead 가 아니었을까 라는 생각이 들 정도로 전체적인 성능이 낮았고 deadline 은 읽기요청을 우선으로 하는 scheduler 답게 read 관련 테스트에서는 write 보다 좋은 성능을 보였으나 요청 큐를 정렬하고 다양한 큐를 사용하는데 있어서 벤치마크를 하나 실행하는 환경에서는 아무것도 하지않고 요청을 병합하기만 하는 Noop 에 성능적으로 낮을 수 있다고 생각했다. Noop 는 정렬도 하지 않고 탐색에 대한 부담이 없기에 가장 성능이 좋았던 것으로 예상된다.

다양한 scheduler 를 iozone 이라는 벤치마크를 이용해서 성능을 비교해보고 그 이유를 생각해보는 과정에서 다양한 실행환경에서 알맞은 I/O scheduler 를 사용하는 것이 성능향상에 도움이 될 수 있다는 점을 알게 되었다.

4. Reference

운영체제실습 강의자료 참조