

Operating Systems

Assignment #5

담당교수 : 김태석

강의 시간 : 수2

학부 : 컴퓨터정보공학부

학번 : 2017202088

이름 : 신해담

1. Introduction

- I/O Zone을 이용해서 리눅스의 I/O scheduler의 성능을 테스트한다. I/O scheduler는 Noop, CFQ, deadline을 사용하며, 측정 시간이 실행환경에 따라 변경될 수 있으므로 3번 이상 실행한 평균값을 취한다. 테스트는 sequential write, sequential read, random read/write 중 임의 선택해서 진행한다.

2. 실행결과 및 분석

- **Linux I/O scheduler**

- ✓ **Noop(No operation)**

가장 간단하게 구현된 scheduler이다. I/O에 대해 re-ordering을 하지 않고 FIFO로 동작하며 단순한 merge 기능만 제공한다. Scheduler가 별다른 기능을 수행하지 않으므로 device driver와 같은 다른 layer에서 ordering 및 scheduling 최적화가 수행된다고 가정하는 경우가 많다. 따라서 최적화가 잘 되어있는 driver를 포함하는 SSD 등에서 좋은 효과를 내기도 한다.

- ✓ **CFQ(Completely Fair Queueing)**










각 프로세스마다 queue를 제공하여 I/O request에 대해 공평하게 할당하는 것을 보장한다. 각 queue는 RR 방식으로 처리되며 정해진 time slice 내에서 request를 수행한다.

- ✓ **Deadline**

정렬 가능한 I/O request queue를 가지며, 각 request별로 start service time을 기억하고 deadline이 넘는 request를 먼저 수행하는 방식이다. Deadline queue(r/w), sector sorted queue(r/w) 등 총 4개의 queue를 사용한다. Sorted queue의 request를 처리하기 전에 deadline queue에 deadline을 넘긴 request가 있다면 먼저 수행한다. seek time이 감소하고, deadline 알고리즘으로 starvation을 방지하여 HDD같은 device에서 좋은 성능을 보인다.

- 테스트 환경

- ✓ VMware Hardware Spec

Device	Summary
 Memory	4 GB
 Processors	2
 Hard Disk (SCSI)	40 GB
 CD/DVD (SATA)	Auto detect
 Network Adapter	NAT
 USB Controller	Present
 Sound Card	Auto detect
 Printer	Present
 Display	Auto detect

- ✓ 저장장치

SK hynix PC601 HFS256GD9TNG-L2A0A

제조사	SK하이닉스	등록년월	2020년 04월
-----	--------	------	-----------

[기본사양]

제품분류	내장형SSD	폼팩터	M.2 (2280)
프로토콜	NVMe	용량	256GB

[성능]

순차읽기	3,100MB/s	순차쓰기	1,300MB/s
읽기IOPS	최대 200K	쓰기IOPS	최대 245K

- ✓ Stride read

I/O request가 순차적이거나 무작위로 발생할 때도 있지만, 일정 간격을 두고 발생하는 경우도 있다. 이런 상황에서의 I/O 성능을 확인하기 위해 stride read를 테스트 항목에 추가했다.

- 테스트 결과

- ✓ Noop

	Record size = 4 kBytes			평균
Initial write	20247.65	19768.67	20169.63	20061.98
Rewrite	19873.24	20355.88	20701.56	20310.23
Read	21270.94	20054.48	21519.68	20948.36
Re-read	21206.82	21457	21548.55	21404.12
Stride read	21141.77	20981.03	21366.57	21163.12
Random read	20394.22	20157.08	20594.8	20382.03
Random write	20263.27	20261.13	19549.59	20024.67
	Record size = 8 kBytes			평균
Initial write	39542.27	40125.41	39705.02	39790.9
Rewrite	41310.14	41154.95	40507.92	40991
Read	41806.43	41559.42	41901.22	41755.69
Re-read	41268.06	42126.39	42395.32	41929.92
Stride read	39702.17	41767.24	40393.52	40620.98
Random read	39403.64	40970.38	40667.24	40347.09
Random write	38641.42	39990.56	39525.08	39385.69
	Record size = 16 kBytes			평균
Initial write	77674	78210.48	76415.95	77433.48
Rewrite	78181.74	79758.54	78439.98	78793.42
Read	78848.96	80619.22	80750.9	80073.03
Re-read	80121.43	81918.23	82188.43	81409.36
Stride read	80530.33	80706.85	81996.79	81077.99
Random read	81408.05	81106.55	80602.94	81039.18
Random write	77171.56	77390.59	77582.94	77381.7
	Record size = 32 kBytes			평균
Initial write	150002.8	151664.9	149412.1	150359.9
Rewrite	151926.2	153151.9	151087.6	152055.2
Read	159368.6	154908.9	155652.2	156643.3
Re-read	155721.4	160925.9	151178.5	155941.9
Stride read	158318.6	143357.3	157600.9	153092.3
Random read	158202.8	147422	156067.4	153897.4
Random write	147218.5	147429.8	146060.9	146903.1

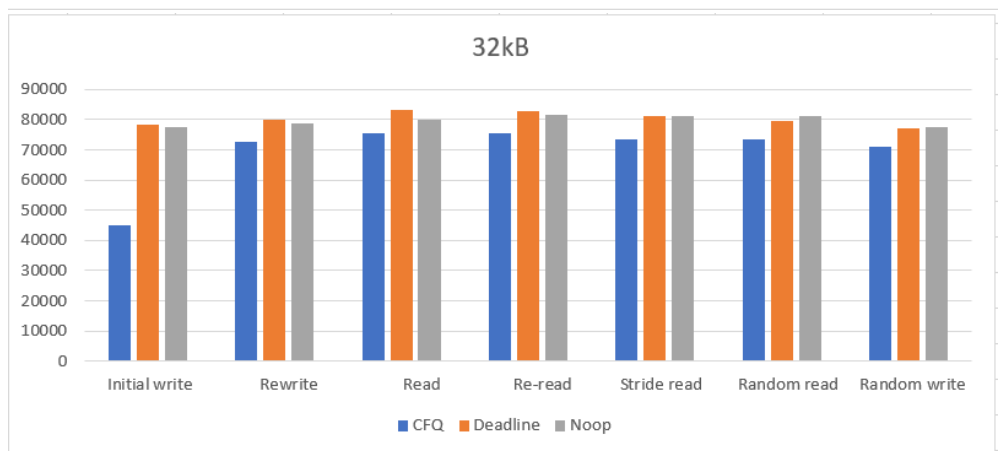
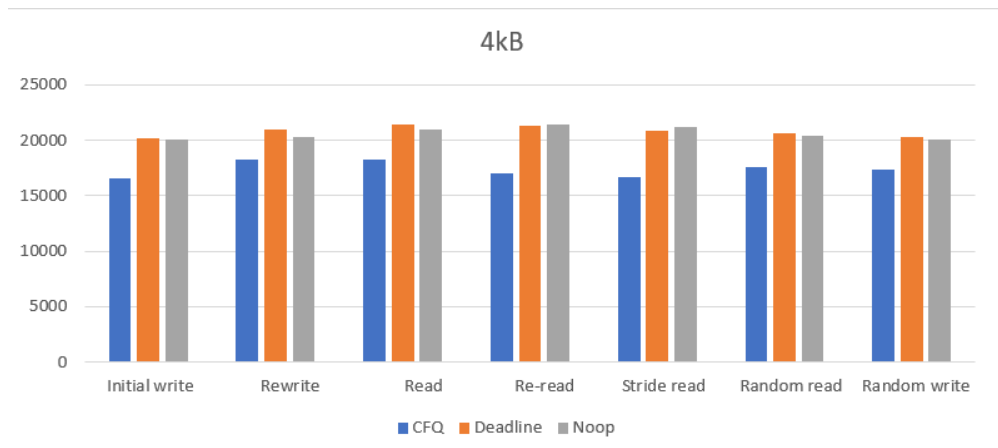
✓ CFQ

	Record size = 4 kBytes			평균
Initial write	17192.64	16423.27	16247.53	16621.15
Rewrite	18727.97	17905.67	18119.27	18250.97
Read	18910.36	17966.11	17907.79	18261.42
Re-read	14179.94	18522.39	18289.92	16997.42
Stride read	13740.86	18199.01	18141.01	16693.63
Random read	17445.24	18003.45	17451.54	17633.41
Random write	17112.4	17613.01	17310.86	17345.42
	Record size = 8 kBytes			평균
Initial write	28829	27834.18	29294.12	28652.43
Rewrite	38029.8	37607.73	36780.5	37472.68
Read	38500.38	38465.03	38744.22	38569.88
Re-read	38052.46	37681.22	38102.75	37945.48
Stride read	37823.72	37371.82	37403.6	37533.05
Random read	37456.72	37250.85	37312.42	37340
Random write	37028.59	36004.99	36200.58	36411.39
	Record size = 16 kBytes			평균
Initial write	47056.57	44072.21	43401.59	44843.46
Rewrite	72907.5	72857.19	72128.53	72631.07
Read	75266.55	75833.38	75665.53	75588.49
Re-read	76073.8	75310.89	74553.3	75312.67
Stride read	74727.86	73505.67	72546.57	73593.37
Random read	73822.76	74194.27	71939.04	73318.69
Random write	72980.84	70614.64	69719.48	71104.98
	Record size = 32 kBytes			평균
Initial write	68810.74	93119.61	106257.7	89396.02
Rewrite	138528.7	137672.1	132698	136299.6
Read	148045.2	147989.8	146610	147548.3
Re-read	148818.8	145903.8	135350.7	143357.8
Stride read	145441.3	142259.5	145882	144527.6
Random read	143524	141305.3	142511.8	142447.1
Random write	131868.3	132953.4	134753	133191.6

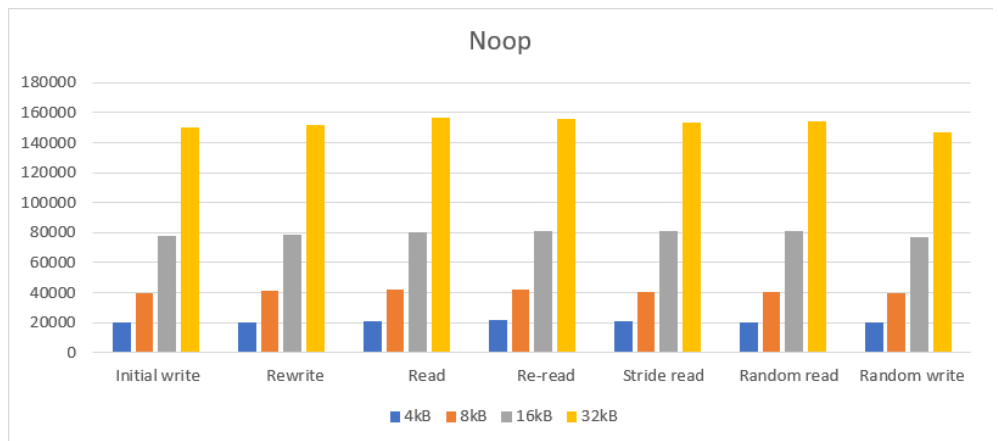
✓ **Deadline**

	Record size = 4 kBytes			평균
Initial write	20193.34	20009.53	20190.97	20131.28
Rewrite	20969.01	20934.92	21076.73	20993.55
Read	21231.88	21519.19	21427.38	21392.81
Re-read	21151.93	21503.64	21279.41	21311.66
Stride read	21361.98	21221.21	19913.92	20832.37
Random read	20896.21	20844.97	20162.03	20634.4
Random write	20319.95	20197.19	20211.71	20242.95
	Record size = 8 kBytes			평균
Initial write	39888.48	39741.88	39359.3	39663.22
Rewrite	41423.73	41035.17	41331.14	41263.35
Read	41887.46	41357.15	41165.75	41470.12
Re-read	42002.38	41768.04	41811.66	41860.69
Stride read	41758.77	41158.16	41194.5	41370.47
Random read	41053.88	40228.81	40968.32	40750.33
Random write	40253.81	40304.27	40090.36	40216.15
	Record size = 16 kBytes			평균
Initial write	77918.34	78719.23	78210.43	78282.67
Rewrite	79503.1	80100.58	79886.89	79830.19
Read	83655.88	83331.4	83126.77	83371.35
Re-read	84911.32	80345.35	83110.81	82789.16
Stride read	80396.09	81573.34	81134.66	81034.69
Random read	79849.31	78591.34	80633.02	79691.22
Random write	77518.61	76946.52	77162.76	77209.3
	Record size = 32 kBytes			평균
Initial write	153138.8	148702.3	149998.4	150613.2
Rewrite	153107	151969.3	154937.6	153338
Read	161672.3	158996.7	152891.1	157853.4
Re-read	154756.3	162680.5	147724.4	155053.7
Stride read	159159.5	157417.9	150583.9	155720.4
Random read	157256.4	155773.4	148733.4	153921
Random write	148389.5	146940.4	146545.2	147291.7

✓ Scheduler별 결과 비교 그래프



✓ Record size별 결과 비교 그래프



- **결과 분석**

- ✓ **Record size**

모든 scheduler에서 크기가 4k → 8k → 16k → 32k로 점점 증가함에 따라 읽기/쓰기 속도도 마찬가지로 2배씩 증가하는 경향을 보인다. 따라서 사용중인 SSD가 한 번에 저장할 수 있는 블록의 크기는 32kB보다 크다고 할 수 있을 것이다.

- ✓ **Scheduler**

deadline과 noop은 비슷한 성능을 보인다. deadline도 queue에서 request를 꺼내 처리하는 동작 외에 별다른 작업이 추가되지 않으므로, noop과 유사하게 SSD의 device driver에서 최적화가 수행되어 잘 작동되는 것으로 볼 수 있다.

반면에 cfq는 다른 둘에 비해 성능이 낮는데, I/O request를 queue에 할당해서 RR로 처리하면서 SSD에 request가 입력되기까지 지연이 발생하기 때문에 추정된다. 특히 sequential write에서 record size가 클수록 queue 병목현상이 심해지는 것으로 보인다.

- ✓ **Read**

Sequential / random / stride / re 각각의 읽기 성능이 편차가 크지 않고 거의 비슷하다. SSD 특성 상 HDD에서의 rotate/seek 딜레이가 없고 한 번에 원하는 sector를 read할 수 있기 때문일 것이다.

- ✓ **Write**

Sequential / re / random 각각에서의 쓰기 성능도 크게 차이가 나지 않는다. SSD에서 rewrite 시 쓰기 전에 erase하면서 공간이 확보되므로 상대적으로 성능이 약간 더 좋다.

3. Reference

- **I/O Scheduler**

<https://ji007.tistory.com/entry/IO-Schedulers>