파일시스템 변경에 따른 SSD RAID 임의 쓰기 분석

박찬현 ○ 원유집

한양대학교 컴퓨터·소프트웨어학과

parkch0708@hanyang.ac.kr, yjwon@hanyang.ac.kr

Analysis of SSD RAID random write following filesystem change

Chanhyun Park○ Youjip Won

Department of Computer and Software, Hanyang University

요 약

SSD는 HDD를 대체하여 컴퓨터의 I/O 성능의 병목을 제거할 장치로 각광받고 있다. 서버 환경에서는 HDD로 인한 병목 현상을 줄이기 위해 RAID를 사용하였다. SSD는 HDD보다 수 배 빠른 I/O 속도를 가지고 있기 때문에 HDD 대신 SSD로 RAID를 구성 할 경우 훨씬 더 높은 I/O 속도를 기대할 수 있다. 본 논문에서는 로그 기반 파일시스템이 SSD RAID에서 임의 쓰기 성능이 높을 것이라 생각하고, 이를 증명하기 위해 EXT4와 F2FS 파일시스템을 사용하여 SSD RAID의 임의 쓰기 성능을 측정하였다. 실험 결과 F2FS가 EXT4보다 임의 쓰기 성능이 최대 50배 이상 높다는 것을 확인하였다. 이러한 원인을 분석 하기 위해 블럭 트레이스 분석을 진행 하였다. 분석 결과 EXT4와 달리 F2FS에서는 임의 쓰기 임에도 순차 적으로 쓰기 작업이 진행되었고, 요청한 I/O 크기보다 큰 단위로 쓰기 작업이 진행되기 때문에 I/O 횟수가 EXT4에 비해 50배 이상 적었다. 이러한 현상이 성능 향상의 원인이라 추론하였다.

1. 서 론

저전력과 빠른 I/O 성능을 특징으로 하는 저장장치인 SSD는 HDD를 대체하여 컴퓨터의 I/O 성능의 병목을 제거할 장치로 각광 받고 있다.

안정성과 빠른 응답성의 필요성이 높은 서비스가 확산됨에 따라 보다 빠른 응답성과 처리 속도를 갖는 대안이 필요하다. 서버 환경에서는 이에 대한 대안으로 RAID[1]를 사용하였다. I/O 속도가 HDD보다 수 배 빠른 SSD로 RAID를 구성할 경우 훨씬 높은 I/O 속도를 기대할 수 있다.

최근의 SSD RAID 연구들을 보면, stripe 크기 변화에 따른 성능 변화를 분석하거나[2, 3], RAID 구성 변경에 따른 영향을 분석하였다[2, 4]. 하지만 RAID 구성에 대한 소프트웨어 계층의 영향에 대한 적절한 분석은 없었다.

우리는 로그 기반 파일시스템이 SSD에 적합하듯, SSD RAID에도 적합할 것이라 생각 하였다. 이를 증명하기 위하여 리눅스의 기본 파일시스템으로 사용되고 있는 저널링 파일시스템인 EXT4[5]와 로그 기반 파일시스템인 F2FS[6]를 사용하여 SSD RAID의 임의 쓰기 성능을 분석하였다. 이를 통하여 파일시스템의 선택이 RAID 구성에 대한 성능 차이를 최대 50배 이상을 이끌어 낼 수 있음을 확인하였다. 그리고 블럭 트레이스 분석을 통하여 성능 차이가 나는 원인을 분석하였다.

2. 실험에 사용된 파일 시스템들의 특징

저널링 파일 시스템은 모든 쓰기 요청에 대한 정보를 저널이라는 공간에 특정한 형식으로 저장을 한다. 저장 방식에 따라 모든 데이터와 메타데이터를 같이 저장하는 저널 모드와 메타데이터만 기록을 하되 데이터를 먼저 기록하도록 하는 ordered 모드, 그리고 메타데이터만 기록하도록 하는 write back 모드가 있다.

로그 기반의 파일 시스템의 경우 세그먼트라는 단위로 연이어서 디스크에 저장을 한다. 때문에 모든 쓰기 요청이 순차 쓰기로 기록된다. 이상 적인 로그 기반의 파일 시스템이라면 임의 쓰기 역시 순차 쓰기처럼 기록을 하여 다른 파일 시스템 보다 임의 쓰기의 성능이 높을 것이다.

3. 실험 설계

실험에 사용된 컴퓨터의 환경과 하드웨어의 스펙은 다음과 같다. CPU는 Intel의 Core i7-3770을 사용하였다. 해당 CPU는 3.5GHz로 동작하는 쿼드코어 CPU이다. 메모리는 삼성의 DDR3 SDRAM PC3 12800 4GB 메모리를 4개 사용하였다. 총 16GB의 메모리가 장착되어 있다. OS는 Ubuntu 13.04 64bits를 삼성 SSD 840 Pro 256GB에 설치하였다. 커널은 3.13 커널을 사용하였다.

RAID 구성을 위한 RAID 컨트롤러는 Dell의 PERC H710P를 사용하였다. 해당 RAID 컨트롤러는 1GB의 캐시 메모리를 가지고 있고, PCI-E 2.0 x8 lane 인터페이스를 가지고 있다. RAID구성시 사용한 SSD는 8개의 삼성 840 Pro 256GB 이다.

RAID 구성은 8개의 SSD로 RAID 0와 RAID 5를 구성하였다. Stripe 크기는 512KB로 설정하였다. EXT4와 F2FS 파일시스템을 사용하여 I/O 성능을 측정하였다. 단일 SSD 대비 성능 향상을 확인하기 위해 SATA 3.0 인터페이스에 연결한 SSD 840 Pro 256GB에서도 동일 실험을 진행하였다.

I/O 성능을 측정 하기 위해 MobiBench[7]라는 벤치마크 툴을 사용하였다. I/O 크기는 4KB로 설정하였다. 그리고 Buffered I/O를 사용하였다. I/O 성능 측정 시 파일의 크기는 단일 SSD의 경우 12.5GB, RAID의 경우 100GB로 설정하였다. 동일 실험을 5번 반복하여 평균을 구했다.

4. 실험 결과 및 분석

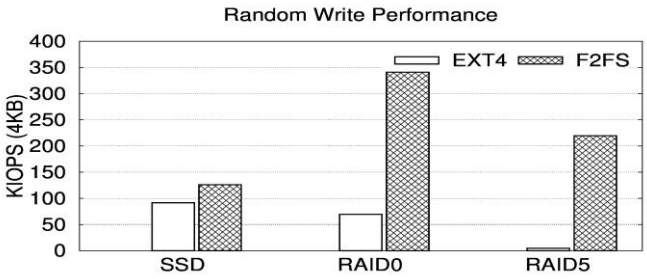


그림 1. EXT4, F2FS의 임의 쓰기 성능 비교

그림 1은 단일 SSD와 SSD 8개로 구성된 RAID 0, RAID 5에서 EXT4와 F2FS의 임의 쓰기 성능을 나타낸 그래프이다. 각각의 경우 5번 성능 측정을 진행하여 평균을 구했고, 모든 편차가 2% 미만이라 따로 표기하지 않았다.

모든 환경에서 F2FS가 EXT4보다 높은 성능을 보여주었다. F2FS에서 단일 SSD의 경우 125.9KIOPS 내외의 성능으로 EXT4보다 약 1.4배, RAID 0의 경우 340.8KIOPS 내외의 성능으로 EXT4보다 약 5배, RAID 5의 경우에는 219.5KIOPS 내외의 성능으로 EXT4보다 무려 50배 이상으로 높은 성능을 보여주었다.

이번 실험을 통하여 파일시스템이 SSD로 구성된 RAID의 성능에 큰 영향을 줄 수 있다는 점을 확인 할 수 있었다. 이러한 원인이 발생하는 이유를 파악하기 위하여 임의 쓰기를 할 때의 EXT4와 F2FS의 블럭 트레이스를 추출, 분석하였다.

그림 2는 단일 SSD, RAID 0, RAID 5에서 임의 쓰기 시의 EXT4와 F2FS의 블럭 트레이스 그래프이다. 그래프의 Y축은 Logical Block Address(LBA)를 나타낸다.

EXT4의 쓰기 패턴은 단일 SSD의 경우 LBA 0~12.5GB 영역을, RAID 0와 RAID 5의 경우 LBA 0~100GB 영역에 주기적으로 쓰기 작업을 하는 것을 확인할 수 있었다. 한 번의 주기의 크기는 분석 결과 약 2GB 내외였다. SSD의 경우 6.5번의 주기가, RAID 0와 RAID 5의 경우 55번의 주기를 확인 할 수 있었는데, 이 횟수를 2GB에 곱하면 설정한 파일 사이즈와 유사했다. 그래프 상에는 한번의 주기가 순차적으로 쓰인 것처럼 보이지만 실제로는 중간 중간 비어 있어, 임의 쓰기가 발생 한 것을 알 수 있다.

F2FS의 경우 임의 쓰기 임에도 불구하고 순차 쓰기 동작이 발생한 것을 알 수 있다. 이러한 쓰기 방식은 F2FS의 임의 쓰기 성능이 높은 이유가 된다. 데이터가 0~100GB 영역이 아니라 100~200GB 영역에 쓰인 것은 데이터 업데이트 시 덮어쓰기가 아닌 새로운 위치에 쓰는 로그 기반의 파일 시스템의 특성 때문이다.

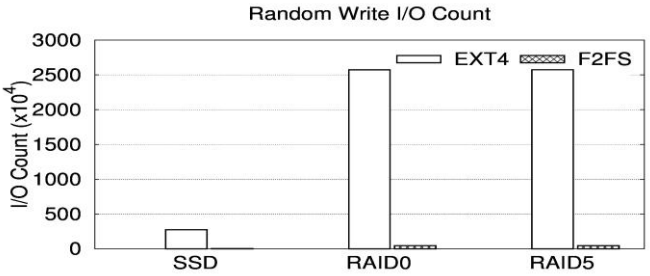


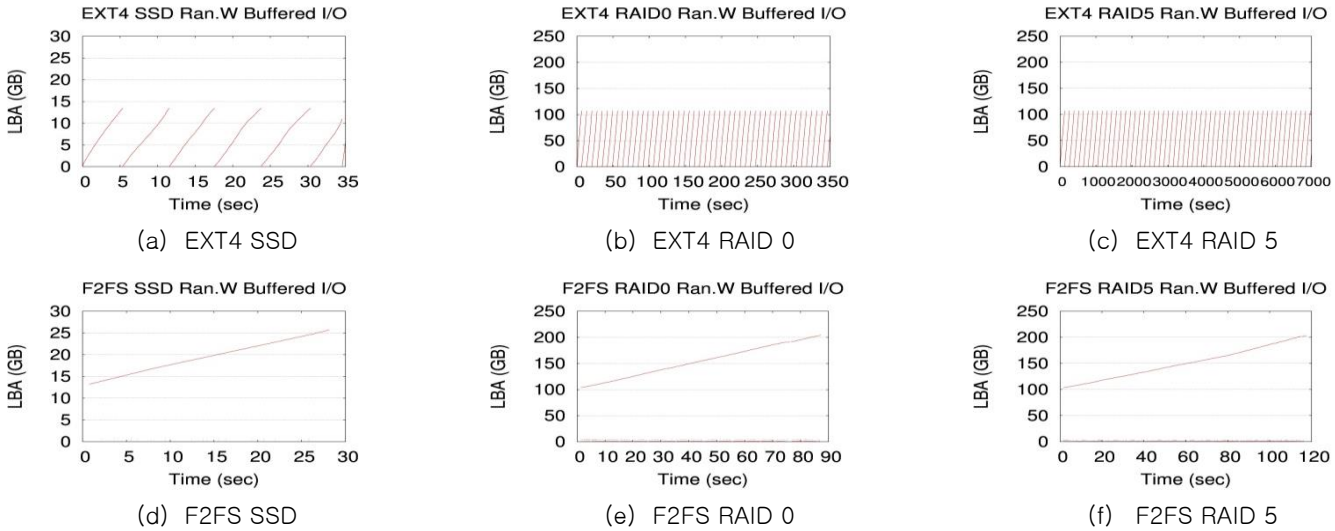
그림 3. EXT4, F2FS 임의 쓰기 시의 I/O 횟수

그림 3은 EXT4와 F2FS에서 임의 쓰기 시의 I/O 횟수를 나타낸 그래프이다. 그래프의 Y축은 발생한 I/O의 횟수를 나타낸다. (단위 - 만회)

EXT4의 경우 I/O가 SSD에서 약 272만회, RAID 0에서 약 2572만회, RAID 5에서 약 2573만회 발생하였다. 반면 F2FS에서는 I/O가 SSD에서 약 3만회, RAID 0에서 약 44만회, RAID 5에서 약 43만 회 발생하였다. 이는 EXT4와 비교 했을 때 각각 105배, 58배, 59배 적은 수치이다. I/O 횟수에서 차이가 나는 원인을 알아보기 위해 각각의 경우 I/O의 크기를 분석하였다.

표 1. EXT4, F2FS의 임의 쓰기 시 I/O의 크기 (단위 - KB)

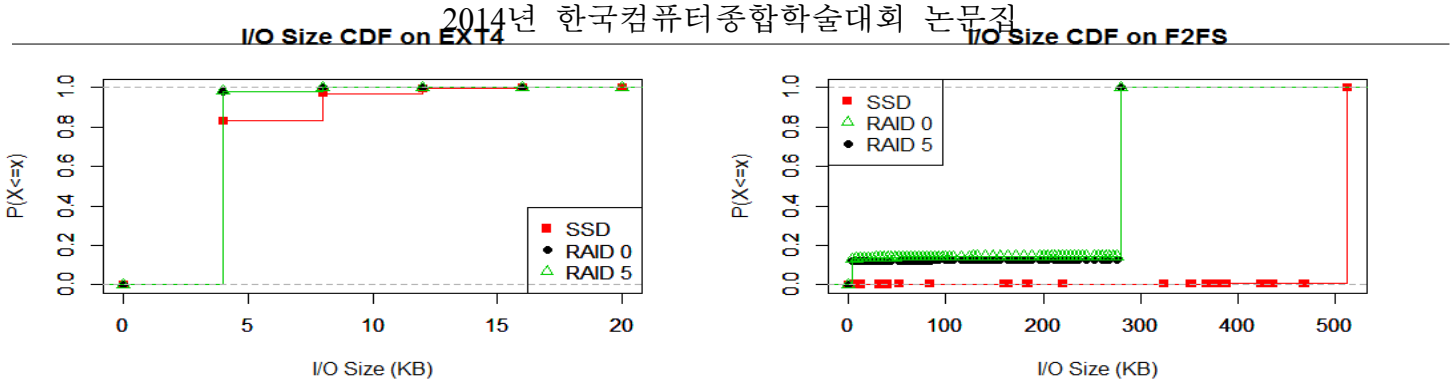
	EXT4			F2FS		
	SSD	RAID0	RAID5	SSD	RAID0	RAID5
평균	4.82	4.08	4.08	509.52	240.80	245.71
최소값	4	4	4	4	4	4
중앙값	4	4	4	512	280	280
최대값	36	16	20	512	280	280



(a) EXT4 SSD (b) EXT4 RAID 0 (c) EXT4 RAID 5

(d) F2FS SSD (e) F2FS RAID 0 (f) F2FS RAID 5

그림 2. RAID 5에서 EXT4, F2FS의 임의 쓰기시의 블럭 트레이스 그래프 (파일 크기 - 100GB)



(a) EXT4 (b) F2FS

그림 4. EXT4, F2FS의 임의 쓰기 시 I/O의 크기 (CDF)

표 1은 단일 SSD와 SSD 8개로 구성된 RAID 0, RAID 5에서 임의 쓰기 시 EXT4와 F2FS의 I/O 크기의 평균, 최소값, 중앙값, 최대값을 나타내고, 그림 4는 이를 CDF로 나타낸 그래프이다. EXT4의 경우 단일 SSD에서는 전체 I/O의 약 80%의 크기가 4KB, RAID 0와 RAID 5에서는 거의 모든 I/O의 크기가 4KB인 것을 확인할 수 있었다.

반면 F2FS의 경우 단일 SSD에서는 거의 모든 I/O의 크기가 512KB였고, RAID 0와 RAID 5에서는 전체 I/O의 약 80%의 크기가 280KB 인 것을 확인할 수 있었다. 나머지 20%의 I/O의 크기는 4KB인데, 이는 메타데이터 인 것으로 확인되었다.

블럭 트레이스 분석 결과 F2FS에서는 임의 쓰기 요청이 발생하여도 순차 적으로 쓰기 작업이 진행 되는 것을 알 수 있었다. 또 발생하는 대부분의 I/O의 크기가 EXT4보다 최소 70배에서 최대 128배 커 그만큼 발생하는 I/O의 횟수가 적은 것을 확인하였다.

이번 실험과 분석을 통하여 파일 시스템 변경에 따라 I/O 동작의 변화가 있음을 확인 하였다. 그리고 이러한 현상이 임의 쓰기 성능에 큰 영향을 줄 수 있음을 확인할 수 있었다.

5. 결 론

본 논문에서는 RAID 컨트롤러 Dell PERC H710P와 SSD 삼성 840 Pro 256GB 8개를 사용하여 RAID를 구성하였다. 그리고 EXT4와 F2FS 파일시스템을 사용하여 임의 쓰기 성능을 측정하였고, 성능 차이의 원인을 분석 하기 위해 블럭 트레이스를 분석하였다.

실험을 통하여 파일시스템의 변화로 임의 쓰기 성능이 차이가 나는 것을 확인할 수 있었다. 로그 기반 파일 시스템 중 하나인 F2FS의 임의 쓰기 성능은 RAID 0에서는 EXT4보다 약 5배의 I/O 성능을, RAID 5에서는 무려 50배의 I/O 성능을 보여주었다. 블럭 트레이스의 확인 결과 F2FS는 임의 쓰기 임에도 순차 적으로 쓰기 작업이 진행 되었다. 또, 작은 I/O 사이즈로 쓰기 요청을 보냈음에도 그 보다 큰 단위로 쓰기 작업이 진행 되었고, 이 때문에 발생하는 I/O 횟수가 적었다. 이러한 동작의 차이 때문에 F2FS가 EXT4보다 높은 임의 쓰기 성능을 보여주는 것으로 추론하였다.

한 가지 흥미로운 점은 F2FS의 I/O 크기가 단일 SSD에서는 대부분 512KB이었는데, RAID에서는 280KB로 줄어든다는 점이었다. 추가적인 연구가 진행되어 RAID에서도 단일 SSD와 같이 512KB 크기로 쓰기 작업을 진행할 수 있다면 훨씬 높은 성능을 확인 할 수 있을 것으로 기대된다.

Acknowledgment

본 연구는 지식경제부 및 한국산업기술평가관리원의 산업원천 기술개발사업(정보통신)의 일환으로 수행하였음. [No.10041608, 차세대 메모리 기반의 스마트 디바이스용 임베디드 시스템 소프트웨어]

참고문헌

[1] D. A. Patterson, G. Gibson, and R. H. Katz, "A case for redundant arrays of inexpensive disks (RAID)," presented at the Proceedings of the 1988 ACM SIGMOD international conference on Management of data, Chicago, Illinois, USA, 1988.

[2] I. Petrov, G. Almeida, A. Buchmann, and U. Graf, "Building large storage based on flash disks," *In Proceedings of ADMS*, 2010.

[3] S. Lee and D. Shin, "Performance Analysis on SSD RAID Device," in *KOREA INFORMATION SCIENCE SOCIETY 39*, 2012, pp. 367-369.

[4] N. Jeremic, G. Muhl, A. Busse, and J. Richling, "The pitfalls of deploying solid-state drive RAIDs," presented at the Proceedings of the 4th Annual International Conference on Systems and Storage, Haifa, Israel, 2011.

[5] A. Mathur, M. Cao, S. Bhattacharya, A. Dilger, A. Tomas, and L. Vivier, "The new ext4 filesystem: current status and future plans," in *Proceedings of the Linux Symposium*, 2007, pp. 21-33.

[6] J. Kim. (2012). *F2FS*. Available: http://www.kernel.org/doc/Documentation/filesystems/f2fs.txt

[7] S. Jeong, K. Lee, J. Hwang, S. Lee, and Y. Won, "Framework for Analyzing Android I/O Stack Behavior: From Generating the Workload to Analyzing the Trace," *Future Internet*, vol. 5, pp. 591-610, 2013.