SSD와 파일 시스템의 NoSQL 워크로드 성능평가

안미진○, 오기환, 이상원
성균관대학교
{meeeejin, wurikiji, swlee}@skku.edu

SSD and File System Performance Tests on NoSQL Workload

Mijin Ahn, Gihwan Oh, Sang-Won Lee
Sungkyunkwan University

요 약

대용량 데이터를 처리하기 위해 전 세계적으로 많은 기업들이 NoSQL과 SSD를 사용하고 있다. 본 논문에서는 이를 효과적으로 지원하기 위해 다양한 파일시스템의 성능을 SSD와 NoSQL 워크로드 상에서 측정하였다. 본 논문에서 실험에 사용한 NoSQL은 Couchbase 사에서 발표한 ForestDB로 HB+-Trie를 기반으로 해 기존의 B+ Tree 기반의 NoSQL보다 더 나은 읽기/쓰기 성능을 보여준다. 이러한 ForestDB와 SSD를 이용하여, 기존에 널리 사용되던 파일시스템인 EXT4와 XFS, 그리고 SSD의 특성을 고려하여 개발된 파일시스템인 F2FS가 NoSQL 워크로드 상에서 얼마나 효과적으로 동작하는 지에 대해 성능을 비교하였다.

1. 서 론

최근 급격히 늘어난 소셜 네트워크 서비스의 사용자 수와 여러 분야에 걸쳐 생성된 방대한 데이터의 양은 기존의 관계형 데이터베이스로 관리하기에는 무리가 있다. 관계형 데이터베이스는 대용량 데이터를 처리하는 데 있어 연산 수행 속도와 데이터 관리의 효율성 측면에서 문제가 있다. 이러한 문제를 해결하고 대용량 데이터를 유연하게 처리하고자 관계형 데이터베이스와는 다른 새로운 형태의 데이터베이스가 등장했고 이는 NoSQL이라고 불리며 전세계적으로 발표되고 있다.

또한 많은 업체들은 대용량 데이터를 빠르게 처리하기 위해 플래시 메모리 저장장치인 Solid State Drive(SSD)를 주 저장장치로 사용하고 있다. 기존의 저장장치인 Hard Disk Drive(HDD)는 기계적 장치로 움직이는 데에 비해 SSD는 전기 신호로 움직이기 때문에 매우 빠른 읽기/쓰기 성능을 보여주고 IO 처리량을 크게 증가시킨다. 이에 따라 대용량 데이터를 다루는 많은 업체들은 NoSQL과 고속 저장장치인 SSD를 함께 사용하고 있다.

하지만 기존의 운영체제와 파일시스템들은 HDD만을 대상으로 최적화되어 있었기 때문에 SSD의 최대 성능을 효과적으로 사용하지 못했다. 이러한 문제를 극복하고자 시스템 분야와 DBMS 분야에서 다양한 연구들이 진행되었으며, 현재도 많은 연구가 진행되고 있다[1].

파일시스템 분야에서는 SSD의 특성을 고려한 BtrFS, F2FS와 같은 새로운 파일시스템이 개발되었다. 또한 SSD는 HDD와 달리 바로 덮어쓰기를 할 수 없고 블록 단위로 삭제(erase)를 하고 쓰기를 수행하므로, 파일시스템에서 블록 단위로 데이터를 삭제하고 쓰기 위해 많은 노력을 기울였다. 하지만 이러한 파일시스템을 SSD와 NoSQL 워크로드에서 상에서 사용하였을 때, 어떤 파일시스템을 선택하는 것이 좋은 지에 대한 고려는 아직까지 이루어지지 않았다.

따라서 본 논문에서는 기존에 널리 사용되는 파일시스템인 EXT4[2]와 XFS[3], 그리고 SSD의 특성을 고려하여 개발된 파일시스템인 F2FS[4]가 Couchbase 사에서 발표한 ForestDB의 워크로드 상에서 어떠한 성능을 보이는지 알아보았다. 이를 통해 어떤 파일시스템이 ForestDB에 더 적합한지 파악하고 플래시메모리 SSD에 최적화된 F2FS가 다른 파일시스템보다 실제로 더 나은 성능을 보이는지 알아보았다.

본 논문은 다음과 같이 구성되어 있다. 먼저 2장에서는 ForestDB에 대해 설명한다. 그리고 3장에서는 플래시 메모리 기반 SSD의 특성을 간단히 설명하고, 이를 고려하여 개발된 F2FS에 대해서 간단히 소개한다. 4장에서는 본 논문에서 수행한 성능 평가 환경과 그 결과를 보이며, 5장에서 결론으로 논문을 마무리한다.

2. ForestDB

ForestDB[5]는 Couchbase 사에서 개발한 Key-Value 데이터베이스로 대용량 데이터 처리에 적합하고 Hierarchical B+ Tree를 기반으로 한 Trie를 사용해 기존의 B+ Tree 기반의 데이터베이스보다 더 나은 읽기/쓰기 성능을 보여준다. 또한 기존의 관계형

데이터베이스 에서 취하는 In-Place Update 방식과는 다르게 모든 데이터 쓰기 및 업데이트 작업을 Append-only 방식으로 취하여 모든 쓰기 요청을 순차 쓰기로 처리하는 특징을 가지고 있다.

본 논문에서는 성능 측정을 위해 ForestDB에서 제공하는 ForestDB-Benchmark[6]를 사용했다. 이 때 사용한 워크로드는 덮어쓰기로 데이터베이스의 임의의 키(Key)를 선택하여 해당하는 문서(document) 값을 업데이트하는 작업이다. 이 워크로드는 야후 사에서 개발한 YCSB 워크로드 중 F 워크로드이다[7].

3. SSD와 SSD의 특성을 고려해 개발된 F2FS

플래시 메모리 SSD는 전기 신호로만 움직이는 저장장치로 HDD와 달리 지연 시간이 존재하지 않는다. 하지만 플래시메모리는 덮어쓰기가 되지 않는 특징으로 인해, 읽기/쓰기 연산 외에 삭제 연산이 필요한데 이는 읽기/쓰기 연산에 비해 수행 시간이 훨씬 오래 걸린다. 이러한 문제점을 극복하기 위해 쓰기 연산에 대해 덮어쓰기 대신 COW(copy-on-write)를 해 out-of-place로 데이터를 갱신하는 기법이 등장했다. 또한 SSD는 다수의 플래시 메모리 칩으로 구성되어 채널이라고 불리는 단위로 동시에 접근이 가능하다. 따라서 HDD에 비해 높은 IO 동시성(concurrency)을 제공한다.

F2FS는 “Flash Friendly File System”의 약자로, 플래시 메모리 기반의 저장장치에 최적화된 파일 시스템이다. F2FS는 로그 기반(Log-structured) 파일 시스템으로 COW를 적용해 out-of-place로 데이터를 갱신한다. 또한 가비지 소거(garbage collection) 문제를 플래시메모리의 특성을 고려해 해결하여 소거 비용을 줄였다.

본 논문에서는 SSD에 최적화된 파일시스템인 F2FS와 기존의 파일시스템인 EXT4와 XFS을 대상으로 SSD 상에서 NoSQL 성능 평가를 수행하였다.

4. 성능 평가

[표 1] 실험 환경

운영체제	Ubuntu 14.04.3 LTS
프로세서	Intel® Core™ i7-4770 CPU @ 3.40GHz
메모리(RAM)	32.00GB
저장장치	Samsung 840 Pro SSD
데이터베이스	ForestDB
성능 평가 시 사용한 툴	ForestDB-Benchmark

본 논문에서 성능 측정은 Samsung 840 Pro SSD를 기반으로 측정하였고 자세한 실험 환경은 [표 1]과 같다.

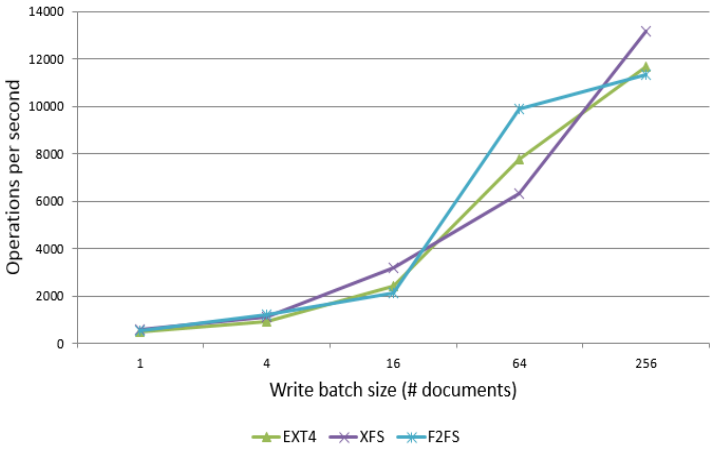
성능 평가 대상이 되는 NoSQL 데이터베이스는 ForestDB이고 성능 평가에 사용한 파일시스템은 EXT4, XFS, F2FS다. 그리고 2장에서 언급했듯이 ForestDB-Benchmark의 덮어쓰기 워크로드를 사용했고, 100,000,000개의 문서를 대상으로 성능을 측정했다. 또한 NoSQL의 일반적인 환경을 조성하기 위하여 데이터베이스 파일의 개수를 4개로 설정해 4개의 데이터베이스 파일에 동시에 접근할 수 있도록 했다.

본 논문에서는 파일시스템 종류와 문서의 개수인 쓰기 배치 크기(write batch size)를 달리해 실험을 진행했고 이에 따른 실험 결과는 아래의 [표 2]와 같다.

[표 2] 각 파일 시스템의 성능 측정 결과 (ops)

파일 시스템	쓰기 배치 크기(write batch size)				
	1	4	16	64	256
EXT4	501	909	2425	7744	11694
XFS	554	1109	3166	6304	13172
F2FS	529	1202	2127	9911	11346

[표 2]를 그래프로 나타내면 아래의 [그림 1]과 같다.



[그림 1] 각 파일 시스템의 성능 측정 결과 (ops)

성능 측정 결과, 쓰기 배치 크기에 따라 가장 높은 ops를 보이는 파일시스템이 달랐다. 쓰기 배치 크기가 1일 때는 XFS가 554 ops로 가장 좋은 성능을 보였고, EXT4가 501 ops로 가장 낮은 ops 값을 보였다, 쓰기 배치 크기가 4일 때는 F2FS가 1202 ops로 가장 좋은 성능을 보였고, EXT4가 909 ops로 가장 낮은 성능을 보였다. 쓰기 배치 크기가 16일 때는 XFS가 3166 ops로 가장 높은 ops 값을 보였고, F2FS가 2127 ops로 가장 낮은 성능을 보였다. 쓰기 배치 크기가 64일 때는 F2FS가 9911 ops로 가장 좋은 성능을 보였고, XFS4가 6304 ops로 가장 낮은 ops 값을 보였다. 마지막으로 쓰기 배치 크기가 256일 때는 XFS가 13172 ops로 가장 좋은 성능을 보였고, F2FS가 11346 ops로 가장 낮은 ops 값을 보였다. 즉, 쓰기 배치 크기가 1, 16,

256일 때는 XFS가 가장 높은 ops를 보였고 쓰기 배치 크기가 4, 64일 때는 F2FS가 가장 높은 ops를 보였다.

또한 각 파일 시스템간 ops 차이는 쓰기 배치 크기가 1일 때 4.5%, 4일 때 24.3%, 16일 때 32.8%였다. 그리고 쓰기 배치 크기가 64일 땐 36.4%였고 마지막으로 쓰기 배치 크기가 256일 땐 14.0%였다. 즉, 각 파일 시스템간 ops 차이가 쓰기 배치 크기가 4일 때 4.5%로 가장 작았고, 쓰기 배치 크기가 64일 때 36.4%로 가장 컸다.

5. 결론

본 논문에서는 SSD 상에서 NoSQL 워크로드를 이용해 기존에 널리 사용되는 파일시스템들과 플래시메모리에 최적화된 파일시스템의 성능을 측정하였다.

성능 측정 결과, 쓰기 배치 크기에 따라 가장 좋은 성능을 보이는 파일 시스템이 달랐다. 쓰기 배치 크기가 1, 16, 256일 때는 XFS가 가장 높은 ops 값을 보였고, 쓰기 배치 크기가 4, 64일 때는 F2FS가 가장 높은 ops 값을 보였다. 플래시메모리의 특성을 고려해 만들어진 F2FS는 앞서 말했듯이 쓰기 배치 크기가 4, 64일 때는 가장 좋은 성능을 보였으나, 쓰기 배치 크기가 16, 256일 때는 오히려 가장 나쁜 성능을 보였다. 또한 각 파일 시스템간 ops 차이는 쓰기 배치 크기가 1일 때 4.5%로 가장 작았고, 쓰기 배치 크기가 64일 때 36.4%로 가장 컸다.

사사

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 SW중심대학-ICT/SW창의연구과정 지원 사업의 연구 결과로 수행되었음 (IITP-2015-R2215-15-1005)

이 논문은 2015년도 정부 (미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (R0126-15-1108, FlashSQL:비휘발성 메모리 기반 개방형 고성능 DBMS 개발)

본 연구는 2015년 한국연구재단과 한국여성과학기술인지원센터의 지원을 받아 수행된 연구임 (2014H1C3A1000274, 1711018159)

6. 참고 문헌

[1] Sang-Won Lee, Bongki Moon, Chanik Park, “Advances in Flash Memory SSD Technology for Enterprise Database Applications”, ACM SIGMOD, June 2009

[2] Ext4 (and Ext2/Ext3) Wiki, https://ext4.wiki.kernel.org

[3] XFS, http://xfs.org

[4] Changman Lee, Dongho Sim, Joo-Young Hwang, and Sangyeun Cho, “F2FS: A New File System for Flash Storage”, 13th USENIX Conference on File and Storage Technologies, February 2015

[5] ForestDB, https://github.com/couchbase/forestdb

[6] ForestDB-Benchmark, https://github.com/couchbaselabs/ForestDB-Benchmark

[7] YCSB, https://github.com/brianfrankcooper/YCSB