KV SSD (키-밸류 SSD) 상 LevelDB의 성능 평가

Performance Evaluation of LevelDB on KV SSD (Key Value SSD)

요 약

최근 다양한 응용분야에서 키-밸류 형식의 데이터를 관리하기 위한 용도로 키-밸류 스토어(key-value store)가 사용되고 있다. 기존 키-밸류 스토어는 소프트웨어에서 키-밸류 데이터를 효과적으로 관리하지만, 하드웨어에서는 블록 데이터 형태의 SSD 저장장치를 활용한다. 키-밸류 데이터를 블록 데이터로 전환하는 오버헤드를 줄이기 위해, 최근 제안된 KV SSD(key-value SSD)는 하드웨어에서도 키-밸류 데이터를 저장한다. KV SSD는 KV SSD를 사용하는 자체 소프트웨어(KV 스택, KV Stacks)를 지원하지만, 호스트 소프웨어를 최대한 가볍게 만들어 키-밸류 데이터와 하드웨어 사이의 불필요한 오버헤드를 줄이는 데 목적이 있다. 그러므로 기존 키-밸류 스토어에 비교했을 때 KV SSD 소프트웨어에는 분명 한계점이 있다. 따라서 이 논문에서는 기존 키-밸류 스토어 LevelDB를 KV SSD에 맞게 변형한 키-밸류 스토어 Level-KV를 KV SSD 상으로 이식하고 그 성능을 KV 스택, LevelDB와 비교하여 평가해보고자 한다. Level-KV는 KV 스택와 달리 기존 키-밸류 스토어 LevelDB와 호환이 가능하면서도 KV SSD를 활용하는, KV 스택와 LevelDB 중간 지점의 역할을 한다는 데 의의가 있다.

1. 서 론

키-밸류 스토어(key-value store)는 대규모의 데이터-집약적인 어플리케이션과 서비스를 구축하고 운영하는 데 활용하는 가장 인기있는 소프트웨어 서비스 중 하나가 되었다. 웹 인덱싱, 전자 상거래, 사진 저장소, 클라우드 데이터, 소셜 네트워크, 온라인 게임 등 다양한 분야에서 이미 대규모의 키-밸류 데이터를 사용하고 있기 때문에 키-밸류 스토어를 데이터베이스로 채택하여 활용하고 있다[1].

그러나 기존 키-밸류 스토어는 소프트웨어 상에서는 키-밸류 데이터를 효과적으로 관리하지만, 하드웨어에서는 블록 데이터 형태의 SSD 저장장치를 활용한다. 따라서 키-밸류 데이터를 블록 데이터로 변환하는 큰 비용을 치러야 한다. 소프트웨어의 오버헤드를 줄이기 위해, 최근 제안된 KV SSD는 저장장치 상에서 키-밸류 스토어 서비스 제공을 시도한다. KV SSD는 호스트 소프트웨어의 최소한의 개입으로 어플리케이션의 데이터 요청에 대응할 수 있다는 장점이 있다[2].

KV SSD는 KV SSD 인터페이스를 포함한 호스트 소프트웨어를 제공한다 (KV SSD에서 제공하는 메뉴얼에서 사용된 명칭에 따라 이하 KV 스택(KV Stacks)로 총칭하기로 한다.)[4]. 그러나 KV 스택가 기존의 키-밸류 스토어를 대체할 수 있는 것은 아니다. KV 스택는 store/retrieve/delete/exist 등 기본적인 기능 외의 키-밸류 스토어의 기능(레인지 쿼리, 스냅샷 등)은 지원하지 않으며, KV API는 기존 키-밸류 스토어와 호환이 가능한 것도 아니다.

따라서 이 논문에서는 기존의 인기 있는 키-밸류 스토어 LevelDB에서 파생된, KV SSD를 의식한 키-밸류 스토어 Level-KV를 제안한다. 기존의 LevelDB와 달리 Level-KV는 키-밸류 형태의 데이터를 저장하는 KV SSD를 사용할 수 있게 설계하였으면서도 LevelDB와 호환이 가능하다. 따라서 가벼운 소프트웨어 KV 스택와 오버헤드가 큰 소프트웨어 LevelDB의 중간 지점의 역할을 한다는 데 의의가 있다.

본 연구에서는 Level-KV를 KV SSD 상으로 이식하여 기존의 LevelDB, KV 스택와 성능을 비교한다. 키-밸류 스토어의 성능 측정은 어떻게 세부적인 사항을 조정하느냐에 따라 성능이 천차만별로 달라잘 수 있다는 점에서 어려움을 겪는다. 따라서 이 연구에서 put, get에 대해서 3가지 소프트웨어의 성능을 최대한 정확히 측정하고 비교할 수 있도록 벤치마크를 설계하고 실험 환경을 조정하였다.

2. 연구 배경

2.1 LevelDB

LevelDB는 LSM 트리(LSM-tree, Log-Structured Merge tree)를 기반으로 한 키-밸류 스토어이다. 최신 어플리케이션에 유용한 레인지 쿼리(range query), 스냅샷(snapshot) 등의 기능을 지원하기 때문에 널리 사용되고 있다[1].

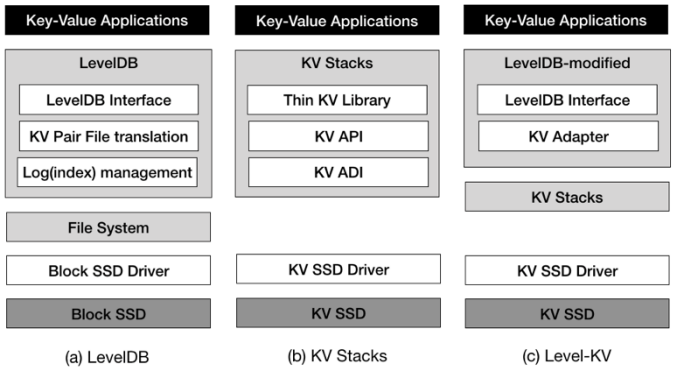
LevelDB의 작동 원리는 그림 2의 (a)와 같다. 주요 구조는 디스크의 로그 파일, 메모리의 정렬된 스킵리스트(멤테이블과 불변의 멤테이블), 그리고 디스크의 레벨 0부터 레벨 6까지의(L0부터 L6) SSTable 파일들이다. LevelDB는 처음 키-밸류 쌍을 로그 파일과 멤테이블(memtable)에 저장한다. 멤테이블이 차면, LevelDB는 새로운 멤테이블과 로그 파일을 제공하는 한편, 가득 찬 멤테이블을 불변의 멤테이블(immutable memtable)로 전환한다. 가득 찬 로그 파일은 제거되고, 불변의 멤테이블은 디스크로 내려가 정렬 스트링 테이블(SSTable, Sorted String Table)로 다시 태어나 레벨 0에 위치하게 된다. 각 레벨마다 파일의 사이즈가 제한되어 있으므로, 각 레벨에서 정해진 파일 사이즈를 넘어가게 되면 그 레벨의 일부 파일들이 다음 레벨의 파일들과 병합(compaction)되어 새로운 다음 레벨의 파일들이 탄생한다[1, 3].

LevelDB에서 키-밸류 쌍을 찾을 때는 멤테이블, 불변의 멤테이블, 레벨 0의 SSTable부터 레벨 6의 SSTable 순서로 찾는다[1, 3].

2.2 KV SSD

대부분의 스토리지 장치는 블록 인터페이스를 기반으로 한다. 그러나 구조화되지 않은 데이터의 급속한 증가는 키-밸류 등의 다양한 데이터 포맷의 출현으로 이어졌다. 키-밸류 데이터가 블록 형식의 스토리지를 사용하는 경우, 소프트웨어에서 키-밸류 데이터를 블록 데이터로 변환해야 하는 오버헤드가 발생한다. (LevelDB의 경우 SSTable 파일로 변환한다.) 변환 계층의 추가는 성능 저하로 이어진다. 또한 유지, 보수 차원에서 호스트 소프트웨어의 복잡성을 증가시킨다. 따라서 KV SSD는 키-밸류 형태의 데이터 형식을 저장장치 차원에서 직접 지원한다[2,4].

KV SSD는 기존 블록 SSD와는 다르기 때문에 KV SSD에 맞춘 빠르고 효율적인 호스트 소프트웨어가 따로 개발되었다[4]. 그 구조는 그림 1의 (b)의 형태와 같다.



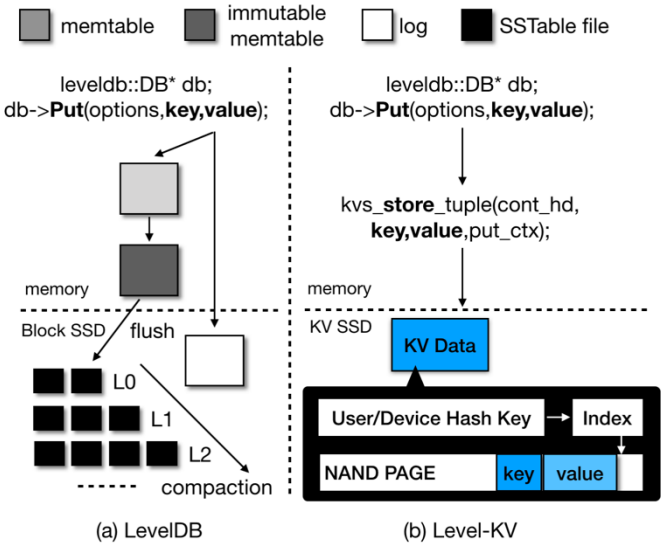
[그림 1] LevelDB, KV 스택와 비교한 Level-KV 구조

3. 시스템 모델 및 성능 측정

3.1 Level-KV

본 연구는 KV SSD 상으로 기존 키-밸류 스토어의 이식을 목표로 한다. 기존 키-밸류 스토어는 블록 SSD를 사용하게끔 설계되어 있으므로 기존 키-밸류 스토어를 KV SSD에 맞춘 형태로 변환해야 한다. Level-KV는 기존의 LevelDB를 KV SSD를 사용하게끔 변형시킨 형태를 띄고 있다. LevelDB와 동일한 인터페이스를 제공하지만, 현재는 Put(key, value), Get(key) 기능밖에 KV SSD 위에서 지원하지 않는다. 나머지 API는 기존 LevelDB와 동일하게 블록 SSD에서 동작한다.

Put(key, value)의 경우 동작 원리는 그림 2의 (b)와 같다. 기존의 LevelDB에서 디스크의 로그 파일에 데이터가 적히는 대신 Level-KV는 KV API를 이용해 키-밸류 데이터를 KV SSD의 키-밸류 데이터로 저장한다. Get(key)의 경우 KV 스택와 동일한 순서로 KV SSD에서 키-밸류 쌍을 찾는다.



[그림 2] LevelDB, Level-KV 작동 원리

3.2 벤치마크

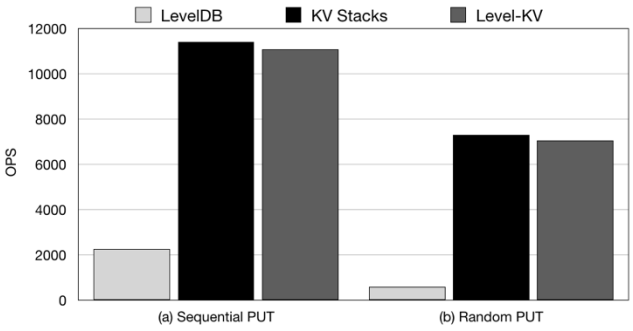
KV SSD에서 제공되는 KVbench는 키-밸류 저장소의 성능 평가를 위한 워크로드를 제공한다[4]. KVbench는 ForestDB 벤치마크를 KV SSD API를 지원하도록 확장시킨 것으로, 기본적으로 RocksDB, KV 스택, Aerospike의 키-밸류 엔진을 지원한다. 본 연구에서는 LevelDB와 Level-KV를 KV 스택와 비교하여 평가하기 위해서 db_bench(LevelDB의 디폴트 마이크로벤치마크)[3]를 KVbench에 탑재하였다.

3.3 성능 평가

LevelDB 성능은 하드웨어 시스템, 워크로드 등에 의해서 크게 달라질 수 있다[5]. 실험은 2개의 Intel® Xeon® Silver 4116 CPU @ 2.10GHz (24 cores per CPU) 프로세서와 64GB 메모리를 가진 머신에서 진행되었다.

사용된 운영체제는 64 bit의 Ubuntu 18.04.3 LTS이며, 커널 환경은 4.15.0-65-generic, 파일 시스템은 ext4이다. 사용된 SSD 저장장치는 3.84TB의 SAMSUNG PM983 모델로, LevelDB의 성능을 평가할 때는 block 모드(block interface)를, Level-KV와 KV 스택의 성능을 평가할 때는 KV 모드(KV interface)를 사용하였다. LevelDB에서 소프트웨어에서 진행되는 데이터 압축 기능(data compression)을 비활성화하여 성능 이해와 분석에 용이하게 하였다.

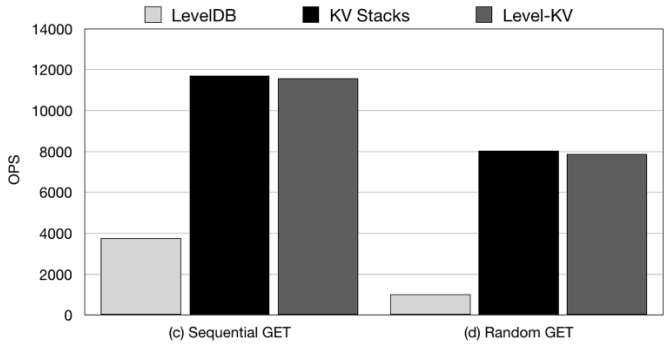
키 사이즈는 16B, 밸류 사이즈는 4KB로 하였고, 한 워크로드에는 십만 개의 오퍼레이션이 포함된다. 모든 워크로드는 단 하나의 PM983과 KV-PM983에서 실행되었다.



[그림 3] LevelDB, KV 스택과 비교한 Level-KV 쓰기 성능

3.3.1 쓰기 성능

그림 3의 (a)는 100% 순차 쓰기(sequential put), (b)는 100% 무작위 쓰기(random put)의 워크로드에 대한 결과이다. 순차 쓰기, 무작위 쓰기에 대해서 Level-KV의 OPS(Operation Per Second)는 LevelDB보다 약 5배에서 12배 정도까지 성능 향상을 보였다. KV 스택과 Level-KV는 거의 동일한 성능을 보였다. Level-KV와 비교했을 때 LevelDB가 무작위 쓰기에서 순차 쓰기보다 성능이 더 떨어지는 이유는 순차 쓰기에서는 병합(compaction)이 일어나지 않지만 무작위 쓰기에서는 병합이 발생하기 때문이다.



[그림 4] LevelDB, KV 스택과 비교한 Level-KV 읽기 성능

3.3.1 읽기 성능

LevelDB의 읽기(get) 성능은 블록과 페이지의 캐시 방식 때문에 DRAM 크기에 의해 크게 좌우된다[5]. 따라서 쓰기(put) 성능에 주목하는 것이 본 연구의 목적에 있어 더 정확하다. 그림 3의 (c)는 100% 순차 쓰기(sequential get), (d)는 100% 무작위 쓰기(random get)의 워크로드에 대한 결과이다. Level-KV가 LevelDB보다 대략 3배에서 8배 정도까지 성능이 향상되었음을 확인해볼 수 있다.

4. 결론

키-밸류 스토어는 데이터 집약적인 어플리케이션의 기본 구성 요소가 되었다. 위 논문에서는 KV SSD에 최적화된 키-밸류 스토어 Level-KV를 제안한다. Level-KV는 블록 SSD를 활용하는 기존의 LevelDB에 비해 성능과 복잡도 면에서 이점이 있고, KV 스택에 비해 기존 키-밸류 스토어와 호환이 가능하다는 이점이 있다.

그러나 현재 개발된 Level-KV는 LevelDB의 API 중 Put(key, value), Get(key)밖에 개발되지 않은 상태이다. Level-KV가 KV 스택보다 완벽한 이점을 가지려면 KV 스택가 지원하지 않는 LevelDB의 기능(레인지 쿼리 등)도 KV SSD 상에서 지원해야 할 것이다. 따라서 LevelDB의 모든 기능을 KV SSD에 최적화하고 LevelDB와 성능을 비교해보는 것이 최종 과제라고 할 수 있겠다.

더불어, 키-밸류 스토어의 성능이 하드웨어 시스템과 소프트웨어 환경 설정에 큰 영향을 받는 만큼 매개변수가 변화함에 따라 소프트웨어의 성능이 어떻게 변화하는지 보다 정확하게 분석할 필요가 있다. KV SSD는 효율적인 확장성(scalability)에 강점을 가지고 있는 저장장치인 만큼 키-밸류 크기를 키울수록 Level-KV의 성능에 긍정적인 영향이 있는지 알아보는 것은 보다 정확한 성능 측정의 출발점일 것이다.

참고문헌

[1] Lanyue Lu, Thanumalayan S. Pillai, Andrea C. Arpaci-Dusseau, and Remzi H. Arpaci-Dusseau. Wiskey: Separating keys from values in ssd-conscious storage. In *Proceedings of 14th USENIX Conference on File and Storage Technologies (FAST)*, Santa Clara, CA, February 2016.

[2] Samsung Electronics. Samsung key value SSD enables high performance scaling. https://www.samsung.com/semiconductor/global.semi.static/Samsung_Key_Value_SSD_enables_High_Performance_Scaling-0.pdf, 2017.

[3] Sanjay Ghemawat and Jeff Dean. LevelDB. http://code.google.com/p/leveldb, 2011.

[4] Samsung Electronics. KV SSD. https://github.com/OpenMPDK/KVSSD, 2018.

[5] Google. LevelDB Benchmarks. http://www.lmdb.tech/bench/microbench/benchmark.html, 2011.