

MySQL Insert Buffer의 성능 평가

이황교^o 이상원
성균관대학교 컴퓨터공학과
gt369kr@skku.edu, swlee@skku.edu

Performance Evaluation for MySQL Insert Buffer

Hwanggyo Lee^o Sang-won Lee
Dept of Computer Engineering, Sungkyunkwan University

요 약

MySQL의 InnoDB에서는 비군집 보조 인덱스로 인해 발생하는 랜덤한 저장 장치 접근 및 이로 인한 데이터베이스 시스템의 성능 저하를 막기 위해 Insert Buffer라는 B+트리 구조를 사용한다. Insert Buffer를 사용하는 것으로 얻을 수 있는 성능 향상의 정도를 성능 평가 도구를 통해 확인하고, 하드디스크와 SSD라는 서로 다른 특성을 보이는 저장 장치에서 Insert Buffer의 사용 효과를 분석하였다. 더 나아가 MySQL 데이터베이스의 페이지 크기를 변화시키는 것이 페이지 단위로 동작하는 SSD에서 성능 상 어떤 변화를 가져오는지 실험을 통해 확인하였다.

1. 서 론

MySQL의 InnoDB 엔진(Engine)에서 사용하는 보조 인덱스(Secondary Index)는 비군집성(Non-clustered), 유일하지 않은(Non-unique) 탐색 키(Key) 사용이라는 두 가지 특성을 지닌다. 이러한 특성 때문에 InnoDB의 테이블(Table)에 레코드(Record)의 삽입, 삭제, 갱신이 발생하여 보조 인덱스가 변경되어야 하는 경우, 저장 장치에 대한 접근이 랜덤(Random)하게 발생한다. 저장 장치로 주로 쓰이는 하드디스크(Hard Disk)는 랜덤하게 읽기, 쓰기를 하는 경우 순차적인 접근의 경우보다 그 성능이 현저히 떨어지기 때문에, 보조 인덱스 갱신 과정은 데이터베이스 시스템(Database System)의 성능을 저하시키는 원인이 된다.

InnoDB에서는 랜덤한 저장 장치 접근으로 인해 발생하는 성능 저하를 해결할 목적으로 *Insert Buffer*, 또는 *Change Buffer*라고 부르는 B+트리(Tree) 형태의 자료 구조를 사용한다. 테이블 레코드의 삽입, 삭제, 갱신으로 인한 보조 인덱스 변경이 일어나면, Insert Buffer는 이들 연산으로 인한 랜덤한 저장 장치 접근을 회피, 지연시키는 방식으로 성능 향상을 꾀한다.

본 논문에서는 일반적인 하드디스크 환경에서 Insert Buffer를 사용하는 것이 사용하지 않은 경우에 비해 어느 정도의 성능 향상을 보이는지 성능 평가 도구를 통해 확인하였다. 또한, 하드디스크와는 다른 특성을 보이는 저장 장치인 SSD(Solid State Drive)에서 Insert Buffer의 사용 효과에 대해서도 실험을 통해 확인하였다.

2. Insert Buffer

그림 1은 테이블에 새로운 레코드가 삽입되었을 경우 Insert Buffer의 동작 예시이다. 예시 테이블에는 열

(Column) a1에 대해 보조 인덱스가 만들어져 있다. 새로 입력된 레코드의 a1 값들을 보조 인덱스에 직접 추가하려고 하는 경우, 보조 인덱스의 비군집성으로 인해 필요한 인덱스 페이지(Page)를 얻어오고 이를 갱신하기 위해서는 저장 장치에 랜덤하게 접근하여야 한다. InnoDB에서는 이를 막고자 보조 인덱스에 추가되어야 할 엔트리(Entry)를 바로 해당 인덱스에 저장하는 것이 아니라, 이를 테이블스페이스(Tablespace) 번호, 페이지 번호를 탐색 키로 사용하는 Insert Buffer B+트리에 저장한다. 그리고 해당 페이지에 대한 Insert Buffer 내용이 존재하는 것을 각 테이블스페이스의 Insert Buffer 비트맵(Bitmap)에 표시한다. 이후 인덱스 페이지가 메모리로 읽어져왔을 때, 비트맵을 확인하여 해당 페이지의 변경 정보가 Insert Buffer에 존재하는 경우, 그 내용을 페이지에 병합하여 적용시키고 Insert Buffer에 저장된 엔트리는 삭제한다. 이렇게 Insert Buffer를 사용하면 비군집 인덱스로 인해 발생하는 랜덤한 저장 장치 접근을 회피할 수 있기 때문에 데이터베이스 시스템의 성능 향상을 얻을 수 있다[1].

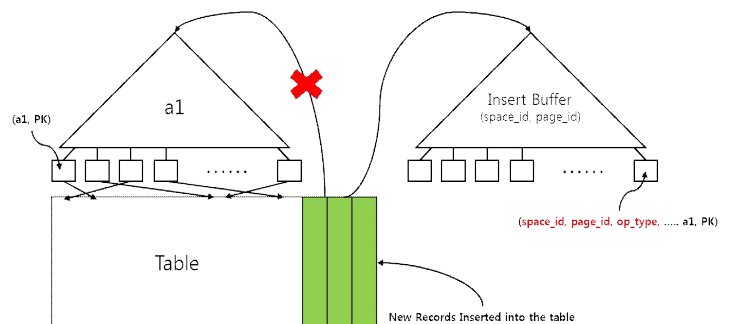


그림 1 Insert Buffer의 동작

Insert Buffer의 성능 향상 정도를 측정하기 위해 성능 평가를 실시하였다. 성능 평가는 MySQL의 평가 도구 중 하나인 Sysbench를 사용하였다.

(단위: 회)

| Insert Buffer | 사용 | 사용안함 |
|---------------|-----------|---------|
| 초당 트랜잭션 처리 수 | 10.43 | 4.79 |
| 읽기/쓰기 요청 수 | 1,355,148 | 624,258 |

표 1 Insert Buffer 사용 여부에 따른 성능 변화

실험은 우분투(Ubuntu) 14.04 운영체제 환경에서 진행하였다. 실험에 사용된 기계는 3.4GHz 쿼드코어(Quad-core) CPU, 8GB RAM이었으며, 하드디스크는 7200RPM, 1TB 크기의 상용 제품을 사용하였다. MySQL은 5.6.25 버전을 사용하였으며, InnoDB의 페이지 크기는 16KB, 버퍼 풀(Buffer Pool) 크기는 1GB로 설정하였다. Sysbench는 테이블 당 40,000,000 레코드, 10개의 테이블로 사용하였으며(16KB 페이지 기준 전체 크기 95GB), 64개 스레드(thread)로 2시간(7200초) 동안 실험을 진행하였다.

표 1은 위의 환경에서 진행한 성능 평가의 결과를 정리한 것이다. 실험 결과, Insert Buffer를 사용하는 것이 사용하지 않는 것 보다 트랜잭션(Transaction) 처리 성능이 약 2.2배 향상되는 것을 확인할 수 있었으며, 성능 평가 수행 과정에서 발생하는 읽기/쓰기 요청도 성능이 향상되는 만큼 더 많아지는 것을 확인할 수 있었다.

3. SSD에서의 Insert Buffer

하드디스크 외에, NAND 플래시 메모리(Flash Memory) 기반의 SSD 역시 저장 장치로 많이 사용되고 있다. NAND 플래시 SSD는 일반적으로 하드디스크에 비해 우수한 성능을 나타내며, 읽기/쓰기 성능의 비대칭성, 빠른 랜덤 읽기 성능, 덮어쓰기가 불가능한 점 등 하드디스크와 구별되는 몇 가지 특성을 지니고 있다[2].

SSD에서 Insert Buffer를 사용하는 경우에도, 하드디스크에서처럼 성능 향상을 기대할 수 있는지를 확인하기 위해 동일한 Sysbench 성능 평가를 SSD 상에서도 진행하였다. 다른 모든 환경은 하드디스크 실험과 동일하게 설정하였으며, 저장 장치만 최대 대역폭 550MB/s, 256GB 크기의 상용 SSD로 바꾸어 실험하였다. 실험 결과는 아래의 표 2에 정리하였다.

(단위: 회)

| Insert Buffer | 사용 | 사용안함 |
|---------------|------------|------------|
| 초당 트랜잭션 처리 수 | 281.76 | 136.42 |
| 읽기/쓰기 요청 수 | 36,517,788 | 17,681,688 |

표 2 SSD에서의 Insert Buffer 성능 평가

실험 전에는 Insert Buffer를 사용해 랜덤한 쓰기 연산을 줄이면, SSD의 빠른 읽기 성능을 바탕으로 하드디스크에 비해 더 크게 성능이 향상될 것으로 예상하였으나, 결과는 SSD와 하드디스크의 기본 성능 차이로 인해 절대적인 수치는 하드디스크 보다 훨씬 크게 나타났으나, 사용/미사용 여부에 따른 변화의 정도는 약 2.1배로 하드디스크에서와 거의 동일하게 나타났다. 하드디스크와 SSD에서의 실험 결과와 삽입, 삭제, 갱신, 즉 테이블에 쓰기를 유발하는 연산에서만 Insert Buffer가 동작한다는 두 가지 점을 토대로, Insert Buffer는 쓰기, 그 중에서도 랜덤한 쓰기로 인해 발생하는 성능 저하를 해결하는 것이 주된 기능이며, 읽기 연산의 성능과는 비교적 연관성이 떨어진다고 판단하였다.

SSD와 하드디스크가 구분되는 또 다른 특징 중 하나는, SSD는 페이지 단위로만 읽기/쓰기 연산이 수행된다는 것이다. 실험에 사용된 SSD를 포함한 일반적인 상용 SSD들은 4KB 페이지 단위로 읽기/쓰기 연산을 수행한다. MySQL InnoDB는 기본적으로 16KB 크기의 페이지를 사용하며, 그 크기를 4, 8, 16KB로 변경하여 사용할 수 있도록 되어있다. 페이지 크기는 실제 MySQL 데이터베이스 시스템의 성능에 영향을 끼칠 수 있는 요소이므로, 이것이 Insert Buffer의 성능에 주는 영향을 역시 Sysbench를 통해 측정해보았다.

실험은 InnoDB의 기본 페이지 크기인 16KB, SSD의 기본 페이지 크기인 4KB 두 가지 경우에 대해 Sysbench 성능 평가 결과의 차이, Insert Buffer 사용 여부에 따른 성능 변화 정도의 차이를 비교하였다. 다른 모든 환경은 지금까지의 성능 평가 실험과 동일하게 유지한 채, InnoDB의 페이지 크기만 4KB, 16KB로 바뀌며 실험을 진행하였다. 먼저, 아래의 표는 페이지 크기의 변화에 따른 성능 변화를 정리한 것이다.

(단위: 회)

| 페이지 크기 | | 16KB | 4KB |
|--------------|-------|--------|--------|
| 초당 트랜잭션 처리 수 | SSD | 281.76 | 320.17 |
| | 하드디스크 | 10.43 | 7.76 |

표 3 페이지 크기에 따른 성능 변화

표 3은 각각 SSD에서 페이지 크기를 변화시켰을 때의 성능 평가 결과와 대조군으로 하드디스크에서 동일한 실험을 진행한 결과이다. SSD에서는 4KB 페이지를 사용하는 경우, 16KB 페이지를 사용할 때보다 더 성능이 좋게 나타나지만, 반대로 하드디스크에서는 성능이 더 떨어지는 결과를 확인할 수 있었다. 하드디스크에서는 기계 장치가 동작하기 때문에 발생하는 지연을 감추기 위해 더 큰 페이지를 사용하여 좀 더 많은 데이터를 순차적으로 저장하도록 구성하는 것이 유리하지만, 기계적 지연이 없이 높은 입출력 속도를 보이는 SSD에서는 작은 크기의 페이지를 사용하는 것이 중복된 데이터의 읽기/쓰기 감소, 페이지 부재(Page Fault) 감소 등 이익이 더 많기 때문에 위와 같은 차이가 나타난 것이다[3].

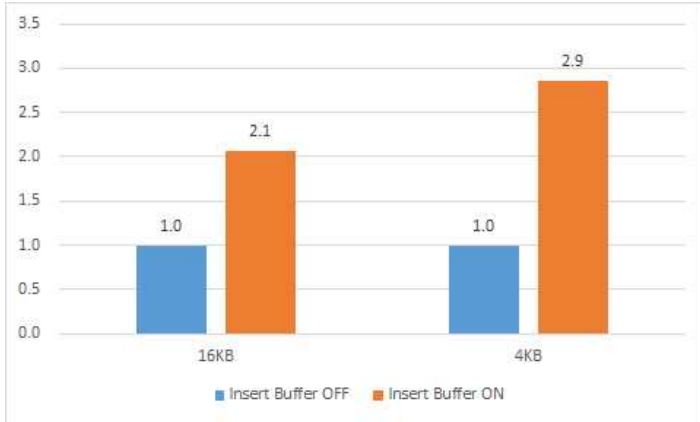


그림 2 SSD에서 페이지 크기 별 Insert Buffer 사용 여부에 따른 성능 변화 정도 비교

그림 2는 SSD에서 Insert Buffer를 사용하지 않은 경우와 비교하여, Insert Buffer를 사용한 경우 상대적 성능 변화 정도를 각 16KB, 4KB 페이지일 때로 구분하여 나타낸 그래프이다. 16KB 페이지를 사용한 경우에는 앞선 실험에서도 서술한대로 2.1배 정도의 성능 변화가 나타났으며, 4KB 페이지를 사용한 경우에는 16KB 페이지에서보다 더 큰 2.9배의 성능 변화가 나타난 것을 확인할 수 있었다. 이러한 차이가 나타난 원인을 찾기 위해 InnoDB에서 제공하는 통계 수치를 분석하였다. 분석한 결과 페이지 크기가 달라졌을 때 Insert Buffer를 사용하는 비율이 달라진 것을 확인할 수 있었는데, 그 결과는 아래의 표 4에 정리되어 있다.

(단위: 페이지)

| 페이지 크기 | 16KB | 4KB |
|----------------------|-----------|------------|
| 전체 입출력 | 6,855,844 | 14,775,383 |
| Insert Buffer 입출력 | 1,412,674 | 5,972,178 |
| Insert Buffer/전체 (%) | 20.61 | 40.42 |
| Insert Buffer 크기 | 16,291 | 65,274 |

표 4 InnoDB 통계 수치

위의 표 4에서, Insert Buffer 크기 항목의 값에 각 페이지 크기를 곱하면 바이트 단위로 나타나는 Insert Buffer의 크기를 구할 수 있다. Insert Buffer의 크기는 16KB, 4KB 페이지일 때 모두 255MB로 나타나는데, 이것은 InnoDB에서 기본적으로 Insert Buffer의 최대 크기를 버퍼 풀 크기의 25%로 정하고 있고, 본 논문의 실험에서는 버퍼 풀 크기를 1GB로 설정하였기 때문이다. Insert Buffer 입출력은, 앞서 서술한대로 Insert Buffer 역시 B+트리 구조를 가지기 때문에, Insert Buffer의 말단 노드(Node)들이 저장 장치에서 읽혀지고 써질 때 발생하는 페이지 입출력을 나타내는 값이다. Insert Buffer 입출력은 Insert Buffer 크기 대비 16KB 페이지의 경우

약 87배, 4KB 페이지의 경우 약 91배로 근소한 차이만 나타나고 있다. 차이가 크게 나타나는 값은 전체 입출력 대비 Insert Buffer 입출력의 비율로 16KB 페이지를 사용한 경우에는 전체의 20% 정도를 차지하던 Insert Buffer 입출력이 4KB 페이지를 사용하자 40% 까지 증가한 것을 확인할 수 있다. 이를 통해, Insert Buffer 페이지 입출력이 다소 증가하더라도, 전체 페이지의 접근 과정에서 발생하는 랜덤한 저장 장치 접근을 줄이는 것이 성능 향상에 더 큰 효과가 나타나는 것으로 판단할 수 있다.

4. 결 론

MySQL은 하드디스크가 가지는 단점을 상쇄하기 위해 비교적 큰 페이지 사용, Insert Buffer 사용 등의 방법을 제시해왔다. 이 중, Insert Buffer는 SSD에서도 마찬가지로 성능 향상의 효과를 기대할 수 있음을 성능 평가 실험을 통해 확인할 수 있었고, 페이지 크기의 경우 SSD에서는 하드디스크와 달리 더 작은 크기의 페이지를 사용하는 것이 더 좋은 성능을 나타낸다는 것을 확인할 수 있었다. 또한 InnoDB에서 제공하는 통계 수치를 통해 Insert Buffer의 사용 비율이 높아지더라도, 전체 페이지 접근에서 발생하는 랜덤한 입출력, 특히 쓰기를 줄이는 것이 성능을 향상시키는 데에 더 중요하다는 것도 판단할 수 있었다.

사 사

이 논문은 2015년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (10041244, 스마트TV 2.0 소프트웨어 플랫폼)

이 논문은 2016년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (R0126-16-1108, 비휘발성 메모리 기반 개방형 고성능 DBMS 개발)

참고문헌

- [1] The InnoDB Change Buffer, <http://mysqlserverteam.com/the-innodb-change-buffer/>
- [2] DongZhe Ma, JianHua Feng, GuoLiang Li, "A Survey of Address Translation Technologies for Flash Memories", ACM Computing Survey, 2014
- [3] Woon-Hak Kang, Sang-Won Lee, Bongki Moon, Yang-Suk Kee, Moonwook Oh, "Durable Write Cache in Flash Memory SSD for Relational and NoSQL Database