

플래시 메모리가 2Q 구성 설정에 끼치는 영향

이재훈^o, 이상원

성균관대학교

{lhh0611, swlee} @ skku.edu

How Flash Memory Affects 2Q Configuration

Jae-Hun Lee^o, Sang-Won Lee

Sungkyunkwan University

요 약

플래시 메모리 저장장치 기술의 발달로 인해 플래시 메모리가 주류 저장장치로 자리 잡는 시대에 도달했다. 플래시 메모리는 기존 하드디스크와 다른 여러 가지 특성을 가져 DBMS의 변화를 야기하는 주요 쟁점으로 자리잡았다. 한편, 2Q는 DBMS의 버퍼 교체 관리 시스템의 정책 중 하나이며, 몇가지 기존 정책이 혼합되어 있는 방식이다. 따라서 각각의 방식을 얼마나 사용하는가에 따라 성능에 차이를 보인다. 따라서, 2Q가 처음 개발되었을 당시와 현재의 하드웨어 시스템의 변화로 인해 2Q의 구성 설정에도 영향이 있는지 확인할 필요성이 대두되었다. 본 논문에서는 플래시 메모리를 사용한 경우 2Q의 구성 비율을 바꾸어 성능을 측정하고, 결과를 도출하여 이전과의 차이점을 규명하여 새로운 구성 설정의 방향성을 제고한다.

1. 서 론

DBMS (데이터베이스 관리 시스템)은 다양한 분야의 하위 시스템들의 집합체로 구성 되어있다. 이 중, 버퍼 교체 관리 시스템은 실제적으로 사용하는 데이터 페이지를 외부 저장소와 메인 메모리 사이에서 관리하는 DBMS의 주요 하위 시스템 중 하나이다. 때문에, 컴퓨터과학계에서는 여러가지 버퍼 교체 관리 정책이 등장했다. 2Q는 이러한 버퍼 교체 관리 시스템의 정책 중 하나이다. 2Q는 1994년 처음 등장한 정책으로[1], 기존 LRU (Least Recently Used) 를 기반으로 하여 장점을 부각하고 단점을 보완한 방식이다. 2Q는 몇가지 버퍼 교체 정책이 혼합되어 있는 방식이며, 각각의 방식을 얼마나 사용하는가에 따라 성능의 차이를 보인다.

2Q의 첫 등장 후, 약 25년여동안 메인 메모리와 컴퓨터 저장장치의 변화가 있었다. 특히, 컴퓨터 주요 저장장치가 하드디스크(HDD)에서 플래시 메모리로 변화하면서 DBMS의 여러 하위 시스템에도 변화의 필요성이 대두되었다. 여러 논문들이 이러한 저장장치 특성의 변화에 맞는 DBMS 아이디어를 제시하고있다[2][3].

본 논문에서는 동일하게 사용해왔던 2Q의 설정 방식이 과거의 HDD를 사용했을 때와 비교하여 현재의 저장장치 변화에 의한 영향이 있는지 실험을 통해 분석, 확인해본다.

본 논문은 2장에서 2Q에 대해 구조와 각 구성 요소들의 최적화된 설정에 대해 알아본 뒤, 3장에서 플래시 메모리를 사용했을 때 2Q의 최적화 비율의 변화

여부가 발생하는지 확인하는 실험을 진행한다. 이후 4장에서는 실험의 결론을 언급하며 논문을 마친다.

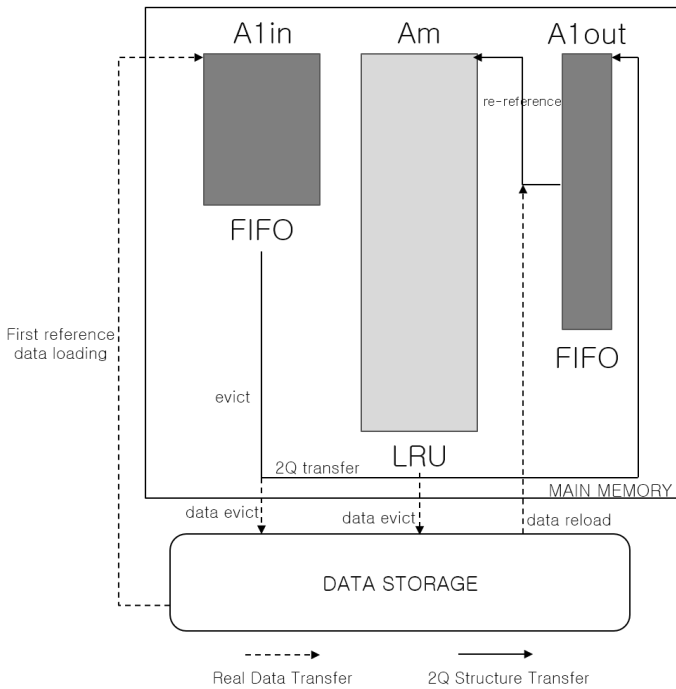
2. 2Q의 기본 구조 및 설정

2.1 2Q의 기본 구조

서론에서 설명한 것과 같이, 2Q는 LRU를 기반으로 LRU의 단점을 보완한 버퍼 교체 관리 정책이다. LRU는 page를 교체할 때, 메인 메모리에 존재하는 페이지 중 사용한 지 가장 오래된 페이지를 먼저 쫓아내는 방식이다. 이 방식은 한번 사용하고 사용되지 않을 페이지들과 일정 주기로 계속 사용하는 페이지들을 구분없이 관리하기 때문에, 자주 사용하는 페이지들이 쫓겨나는 경우가 발생하여 불필요한 read와 write를 초래하는 문제가 발생한다. 2Q는 이러한 문제를 적은 오버헤드로 해결하여 성능을 높였다.

아래의 [그림1]은 2Q의 구조를 표현한 그림이다. 크게 A1 과 Am으로 구성되어 있으며, A1은 A1in과 A1out으로 나뉘어 있다. 이 중 A1in 과 Am은 실제 메인 메모리에 위치한 페이지 정보를 가지고 있는 2Q 데이터이며, A1out은 최근에 victim으로 선택되어 쫓겨난 페이지 정보를 가지고 있는 2Q 데이터이다. A1의 두 큐는 FIFO(First In First Out) 정책으로 동작하며 Am은 LRU 정책으로 동작한다. 동작 방식은 다음과 같다. 페이지를 처음 참조할 때, 스토리지에서 메인 메모리로 실제 페이지를 가져오며 2Q 데이터를 만들어 A1in의 가장 앞부분에 추가한다. A1in 안에 있는 2Q 데이터는 자신이 가리키는 페이지가 다시 참조

되어도 변화가 없으며, 시간이 지남에 따라 다른 2Q 데이터들이 들어와 뒤로 밀려나면 페이지는 쫓겨나고 2Q 데이터는 A1out의 앞부분으로 옮겨진다. A1out에 2Q 데이터가 있는 동안 다시 참조된다면, 해당 페이지는 자주 사용된다 간주하여 다시 메인 메모리에 올라옴과 동시에 2Q 데이터는 Am으로 이동한다. A1out에서 다시 참조되지 않고 쫓겨나면 2Q 데이터는 사라지고 그 이후에 다시 참조되는 경우는 2Q 데이터가 새로 만들어지며 A1in 앞부분으로 다시 들어가게 된다.



[그림 1] 2Q 기본구조

2.2 2Q의 설정 환경

2Q는 A1in, A1out, 그리고 Am 각각의 크기에 따라 성능이 달라질 수 있다. 전체 크기가 동일할 때, Am이 크고 A1in이 작으면 Am으로 다시 참조되는 조건이 엄격해져 큰 Am을 온전히 다 사용하지 못하거나, 더 이상 쓰이지 않을 데이터가 계속 Am에 머물게 되는 등의 문제가 생겨 비효율적인 메인 메모리 사용으로 이어질 수 있다. 반대로, A1in이 크면 Am의 LRU 비율이 적어져 자주 사용되는 페이지를 상대적으로 많이 관리하지 못해 성능이 저하될 수 있다. 따라서 데이터 특성 및 저장장치의 특성에 따라 A1in, Am, 그리고 A1out의 크기 비율이 2Q 성능에 중요한 역할을 한다.

2Q 논문이 쓰여진 때는 HDD를 저장장치로 사용했을 때이며, 여러 실험의 결과 A1in, Am의 비율을 1:3으로 설정해야 한다고 언급 되어있다. 또한, A1out은 메인 메모리에 올라올 수 있는 페이지슬롯 개수의 50%를 담는 크기로 구성해야 한다고 제안한다.

그러나 A1out에 대한 부분은 수학적 치밀한 계산으로 도출된 답인 것에 반해, A1in과 Am의 비율에 대한 부분은 당시의 실험 환경에 맞춘 실험을 통해 도출된 것이다. 따라서 A1in과 Am의 비율의 최적점은 실험 환경에 따라 변할 수 있다. 다음 장에서는 이러한 문제가 스토리지를 플래시 메모리로 사용하였을 경우 어떻게 나타나는 지 확인하였다.

3. SSD 사용 시 2Q 설정에 따른 성능 측정 및 분석

2Q를 버퍼 교체 관리 시스템으로 사용한 PostgreSQL 8.0[4]의 2Q 또한 논문을 토대로 A1in과 Am은 1:3의 비율로, A1out은 이 둘의 페이지 합 50%를 관리하는 크기로 기본설정 되어있다. 차이점은 두가지가 존재한다. PostgreSQL은 멀티 트랜잭션을 지원하기 때문에 A1in에 있는 페이지라 하더라도 다른 트랜잭션에서 참조하면 Am으로 이동할 수 있다. 또한, 메인 메모리를 최대한 낭비하지 않기 위해 메인 메모리 버퍼를 모두 사용하지 않는 상황이라면 A1in, Am이 비율보다 더 많은 자원을 사용할 수 있도록 유연하게 시스템이 구축 되어있다.

실험은 PostgreSQL 8.0.2버전의 2Q에서 A1in, Am의 크기 비율을 변경하여 TPC-C 벤치마크를 수행하였을 때 tpmc로 성능을 측정하였다. 자세한 실험 환경은 [표 1] 과 같다.

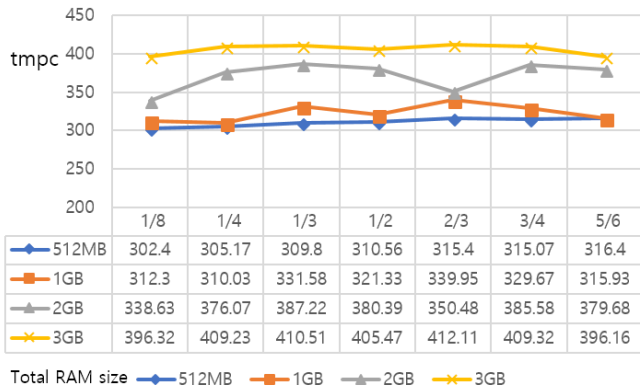
[표 1] 실험 환경

운영체제	Ubuntu 14.04LTS
프로세서	Intel(R) Xeon(R) CPU X5650 @ 2.67GHz (X24)
저장장치	Samsung SSD 850 PRO
데이터베이스	PostgreSQL 8.0.2
워크로드	TPC-C
벤치마크 도구	BenchmarkSQL 5.0 [5]

실험은 A1in의 비율을 기준으로 삼았으며, RAM 사이즈에 따라 차이가 존재하는지에 대한 유무도 추가하여 실험을 진행하였다. 실험은 TPC-C 10분 수행을 기준으로 각각 5번씩 수행하여 측정된 뒤, 평균값을 사용했다. 측정 결과는 [그림 2]와 같다.

실험 측정 결과, 플래시 메모리를 저장장치로 사용했을 때 각 큐의 크기 변화에 따른 차이를 확인할 수 있었다. 총 RAM 크기가 2GB 이상일 때, A1in의 비율이 1/4일 때는 플래시 메모리에서도 tpmc가 상대적으로 높게 나왔다. 그러나 A1in의 비율이 Am보다 높았던 3/4 지점 또한 1/4 일 때와 비슷하거나 미세하게 더 높은 tpmc를 보였다. RAM 크기가 작을수록 A1in이 상대적으로 더 낮은 수치를 보였지만,

그보다는 RAM 전체 크기에 더 많은 영향을 받음을 보인다. 이 실험의 결과는 TPC-C의 특성과 플래시 메모리의 특성에서 비롯된 결과임을 알 수 있다. TPC-C는 read비율이 많은 워크로드이며 플래시 메모리는 read에 대한 성능이 뛰어나기 때문에, 상대적으로 A1in 비율이 높을 때 나타나는 문제점인 자주 사용하는 데이터가 교체될 수 있다는 점이 적게 작용하게 된다. 이러한 특징 때문에, A1in의 비율을 높여 Am의 사용량을 줄여도 페이지 교체를 더 많이 하는 것보다는 전체 메인 메모리를 더 온전히 사용하게 되어 높은 성능을 낼 수 있게 된 것이다. 그러나 여전히 극단적으로 A1in의 비율이 높거나 낮은 경우는 성능이 상대적으로 좋지 않음을 보인다.



[그림 2] A1in의 비율 변화에 따른 tpmc 변화

4. 결 론

본 논문에서는 기존에 사용되어왔던 2Q의 구성 설정 환경이 시대가 변함에 따라 어떤 영향을 받는지에 대해 살펴보았다. 실험 측정 결과는 플래시 메모리를 사용하여 2Q를 버퍼 교체 관리 시스템으로 사용할 때 HDD와 동일한 결과가 나오지 않았다는 점에서 의의를 가진다. 2Q 논문에서는 A1in과 Am의 비율을 항상 1:3으로 유지해야한다고 언급되었다. 그러나 위의 실험 측정 결과는 시간이 지나 새로운 시스템이 출현함에 따라, 데이터베이스 특성과 스토리지의 특성을 함께 고려하여 다른 비율로 구성했을 경우 더 좋은 성능을 보일 수 있다는 점을 시사한다.

참 고 문 헌

[1] Johnson, Theodore, and Dennis Shasha. "2Q: A low overhead high performance buffer management replacement algorithm." Proceedings of the 20th VLDB Conference, 1994

[2] Nguyen, Trong-Dat, and Sang-Won Lee. "I/O characteristics of MongoDB and trim-based optimization in flash SSDs." Proceedings of the Sixth International Conference on Emerging Databases: Technologies, Applications, and Theory. ACM, 2016

[3] Woon-Hak Kang, Sang-Won Lee, Bongki Moon, Gi-Hwan Oh, Changwoo Min, "X-FTL: transactional FTL for SQLite databases." Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data. ACM, 2013

[4] PostgreSQL 8.0.2 source code, <https://ftp.postgresql.org/pub/source/v8.0.2/postgresql-8.0.2.tar.gz>

[5] benchmarksql-5.0 source code, <https://sourceforge.net/projects/benchmarksql/files/benchmarksql-5.0.zip/download>

사 사

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 SW 컴퓨팅산업원천기술개발사업 (SW 스타랩)의 연구결과로 수행되었음 (IITP-2015-0-00314)

이 논문은 2018년도 정부(과학기술정보통신부)의 재원으로 한국연구재단-차세대정보·컴퓨팅기술개발 사업의 지원을 받아 수행된 연구임 (No. NRF-2015M3C4A7065696)