In-Page 로깅 기법을 위한 보조 로그 영역

반재광*, 김일택**, 김성수**, 정태선**
*아주대학교 소프트웨어학과
**아주대학교 컴퓨터공학과
tschung@ajou.ac.kr

An Auxiliary Log Area for In-Page Logging Scheme

Jae-Kwang Van*, Rize Jin**, Sungsoo Kim**, Tae-Sun Chung**

*Dept of Software, Ajou University
**Dept of Computer Engineering, Ajou University

요 약

플래시 메모리에서 B-tree 데이터를 저장하고 관리[4, 5]할 때 빈번한 수정과 구조변동으로 인해 발생하는 블록에 대한 쓰기와 지우기 연산의 비용으로 인해 플래시 메모리의 사용 수명을 단축시키는 문제를 해결하기 위해 현재 많이 쓰이고 있는 로그 저장방식을 검토하고 이를 효율적으로 B-tree에 저장하고 관리하도록 동적 블록 그룹핑과 순환 순서 기반의 저장 알고리즘으로 제안 된 GRR (Ground Round Robin) 기법을 보조 로그 블록을 할당하여 머지횟수를 줄일 수 있는 알고리즘을 제안한다.

1. 서론

플래시 메모리는 이동성, 내구성, 빠른 속도와 소비 전력 절감 등의 이점을 갖고 있기 때문에 모바일 플랫폼과 같은 임베디드 기기의 저장 장치로 사용되고 있다. 현재 가격 인하 및 용량의 증가로 인해 더욱 주목받고 있으며 플래시 메모리가 탑재된 SSD가 HDD의 대체 저장매체로서 상용화되고 있다. 이에 따라 고성능 DBMS의 구현이 가능하게 되었다. 그러나 플래시 메모리는 쓰기 전 지우기, 읽기/쓰기/지우기 연산 성능의 불일치 및 한정된 지우기 횟수 등의 단점을 갖고 있다. 이러한 단점을 극복하기 위해 효율적으로 데이터를 관리하기 위한 기법을 제시한다.

2. 연구의 필요성

플래시 메모리 [2]는 하드 디스크(읽기: 12.7ms, 쓰기: 13.7ms)에 비해 빠른 연산속도(읽기: 80μs, 쓰기: 200μs, 지우기, 1.5ms)를 갖지만 비대칭적 접근 비용을 가지고 있다. 게다가 특정 블록에 대한 지우기 연산 횟수도 제한 (대략 10,000 ~ 100,000번)이 따른다. 이런 특성 때문에 플래시 메모리 상에서 정보를 관리할 때 새로운 디자인 원칙을 적용해야 한다. 예를 들면 읽기 연산을 특정 범위 내에서 수용하면서 쓰기/지우기 연산을 줄이는 접근 방법[3]이 있다.

<표 1> 접근 패턴의 성능비교: 랜덤과 순차적인 접근

매체	랜덤-순차 연산의 비	
	읽기 작업	쓰기 작업
하드 디스크[1]	4.3 ~ 12.3	4.5 ~ 10.0
플래시 메모리[2]	1.1 ~ 1.2	2.4 ~ 14.2

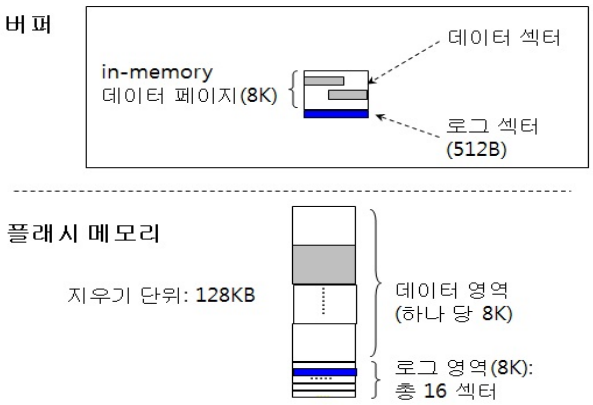
<표 1>에서 보는 바와 같이 플래시 메모리는 하드 디스크와 다르게 기계적 접근이 아닌 전자회로적 접근으로, 랜덤과 순차 접근의 비용차이가 적다. 하지만 플래시 메모리의 쓰기 전 지우기 특성으로 인해 플래시 메모리의 쓰기 연산이 읽기 연산보다 접근 패턴에 영향을 많이 받는 편이기 때문에 연산을 순차적으로 한다면 랜덤하게 하는 것보다 지우기 연산을 줄일 수 있다.

3. 관련 연구

3.1 IPL (In-page Logging) Scheme

플래시 메모리의 쓰기 전 지우기 특성으로 인해 발생하는 빈번한 쓰기와 지우기 연산을 줄이기 위해 IPL (In-page Logging) 구조[3]가 제시되었다. IPL구조는 데이터베이스에서 발생하는 정보 갱신에 있어 실질적으로 변경된 부분만 페이지 단위로 저장하므로 전체 정보를 수정하는 것 보다 더 나은 쓰기 성능을 보여준다.

IPL은 저장 관리자와 버퍼 관리자로 구성되는데, 저장 관리자는 지우기 단위인 블록을 두 영역으로 나눈다. 데



<그림 1> IPL의 버퍼와 저장 구조 [3]

이터 영역(Data Area)에는 실제 데이터가 저장되고, 로그 영역(Log Area)에는 해당 블록에 있는 데이터 영역의 로그가 저장된다.

3.2 GRR (Group Round Robin) Based Storage Scheme[6]

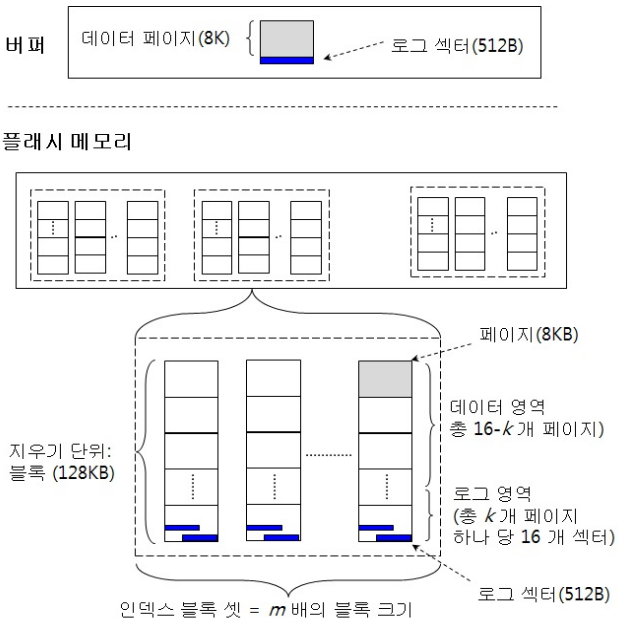
GRR은 B-tree 구조의 빈번한 변경으로 인해 발생하는 쓰기와 지우기 연산을 줄이기 위해 업데이트된 정보를 실질적으로 변경된 부분만 섹터단위로 로그 영역에 저장하며, 작업 부하(Workload)에 대한 분석을 통해 동적 그룹핑(Block Grouping)과 그룹 내의 순환 순서(Round Robin) 기반의 저장 방식을 사용한다. GRR의 전체적인 구조는 <그림 2>와 같다.

3.2.1 동적 블록 그룹핑

동적 블록 그룹핑은 <그림 3>과 같은 용어와 변수를 사용한다. B-tree를 플래시 메모리에 저장할 시 해당 테이블의 크기와 작업 부하를 고려하여 m개의 플래시 블록을 항당하여 한 인덱스 그룹을 만든다. 인덱스 블록 내에서 데이터 영역은 블록의 위부터 아래롤 증가하고, 로그 영역은 밑에서부터 위로 증가한다. 로그 정보가 많이 생성됨에 따라 k의 값은 커지고 최대는 p까지 증가한다. 이는 로그 페이지가 많아지면서 데이터 페이지의 데이터를 재구성 할 때 부하도 커지므로 제한(limit)을 정한다.

3.2.2 순환 순서 기반 저장 기법

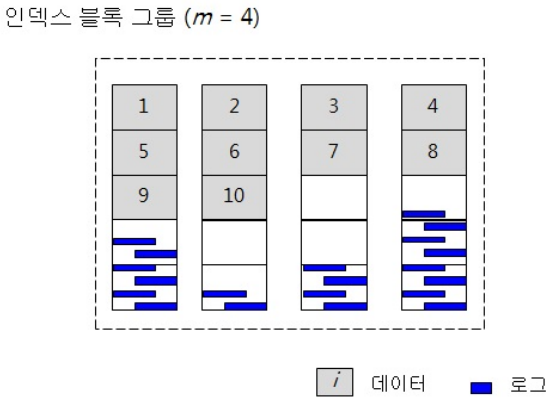
데이터를 저장할 때 IPL처럼 한 블록을 다 사용 후 다음 블록에 저장하는 것과는 다르게 인덱스 블록 그룹에서 순환 순서로 데이터를 저장한다. 그림3에서와 같이 B-tree 노드의 번호를 생성된 순서대로 부여하고 해당 노드에 대한 로그는 그 노드가 속한 블록에 저장한다. 이러한 방식으로 In-Page Logging의 장점을 살리면서 데이터를 분산하여 저장 가능하므로 특정 블록의 빠른



<그림 2> GRR 저장 구조

인덱스 블록: B-트리 노드를 저장하는 플래시 메모리 블록.
m: 하나의 그룹에 있는 인덱스 블록들의 개수
k: 인덱스 블록의 로그 페이지의 수
p: 하나 인덱스 블록에서 로그 영역이 최대 할당할 수 있는 페이지의 수

<그림 3>동적 블록 그룹핑에 사용하는 용어와 변수

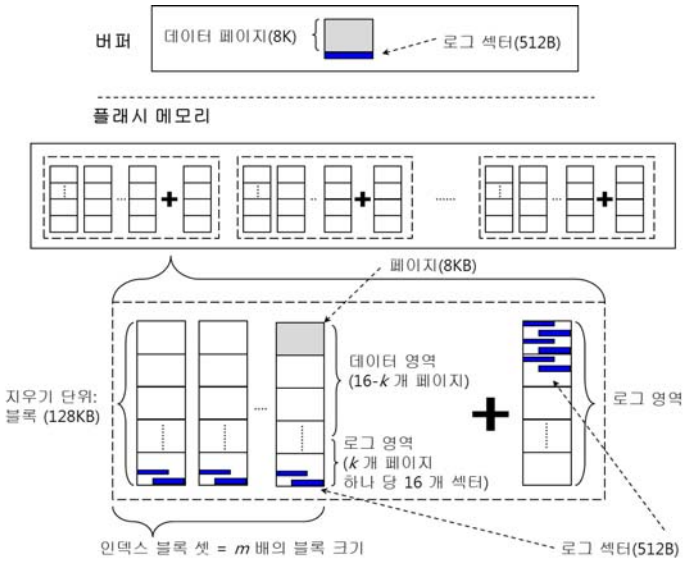


<그림 4> 순환 순서 기반의 저장 구조

마모를 방지할 수 있다. 또한 동적으로 로그 영역의 크기가 조절 가능하므로 B-tree의 빈번한 구조 변경으로 인한 지우기 연산 횟수를 줄일 수 있다.

4. ALA (An Auxiliary Log Area) Storage Scheme

ALA는 플래시 메모리에 데이터를 저장 할 시 기본적으로 GRR의 동적 블록 그룹핑 방식과 순환 순서 기반



<그림 5> 듀얼 로그 블록 저장 구조

저장 방식을 사용한다. GRR은 블록셋의 어느 한 블록의 빈 공간이 없을 때 그 블록을 머지(merge)한다. 하지만 제안한 기법은 블록셋의 한 블록의 빈 공간이 없을 때 새로운 블록을 할당하여 해당 블록 셋의 로그 블록으로 사용한다. 할당된 로그 블록은 해당 블록셋 안에 있는 빈 공간이 없는 블록들의 공용 로그 영역으로 사용하고, 이 로그 블록이 전부 찼을 시에 머지 작업을 실행한다. 이는 보조 로그영역을 할당하면서 빈 공간이 없을 때 실행하는 머지 작업횟수를 줄일 수 있다. ALA의 구조는 <그림 5>에서 보여준다.

5. 결론 및 후속 연구

B-트리는 작은 랜덤 수정과 빈번한 구조의 변동 때문에 플래시 메모리에 저장하고 관리할 때 많은 문제를 일으킨다. 현재까지 플래시 메모리 기반의 B-트리 구현과 저장 관리자 구현에 대한 연구가 많이 진행해 왔다. 본 논문에서 현재 주목받는 IPL 구조를 B-트리의 저장 및 관리에 효율적으로 적용할 수 있는 기법에 블록의 머지 연산의 횟수를 감소시키기 위한 기법을 제안했다. 구체적으로는 해당 테이블 크기와 그의 작업 부하를 고려하여 블록 그룹핑하고 동적으로 로그 영역의 크기를 조절하므로 B-트리의 빈번한 변동으로 인한 블록 지우기 연산을 줄인다. 그리고 그룹 내에서 순환 순서 기반의 저장 방식을 사용하므로 블록들의 마모도를 평준화시킬 수 있다. 보조 로그 영역을 사용함으로 기존 연구 GRR보다 머지 성능이 향상되었다.

6. 참고 문헌

[1] Seagate Barracuda 7200.7 ST380011A

[2] Samsung Electronics. “NAND based solid state drive (MMDPE56G8DXP-0VB) data sheet,” 2009.

[3] Sang-Won Lee and Bongki Moon. “Design of flash-based dbms: An in-page logging approach,” In Proceeding soft the 2007 ACMSIGMOD international conference on management of data, pp55 - 66.

[4] Gap-Joo Na, Bongki Moon and Sang-Won Lee. IPL B+-tree for Flash Memory Database Systems. Journal of Information Science and Engineering 27(1), pp:111-147, 2011.

[5] Rize Jin, Se Jin Kwon, and Tae-sun Chung, “FlashB-tree: A Novel B-tree Index Scheme for Solid State Drives,” In Proceedings of The 8th ACM Research on Applied Computing Symposium(ACMRACS'11), 2011. Miami, FL, USA.

[6] 권선형, 김일택, 정태선, “GRR: 플래시 메모리를 위한 그룹 순환 순서 기반의 B-트리 인덱스 저장 기법,” 한국정보과학회 학술발표논문집, 여수, 한국, 2013 pp.371~373.