

휴대저장장치에서 고정그리드파일의 구현

김동현*, 반재훈**

동서대학교 컴퓨터정보공학부*, 고신대학교 인터넷비즈니스학과**

The Implementation of a Fixed Grid File on the Hand-held Storage

Dong Hyun Kim*, Chae Hoon Ban**

Dongseo University, Division of Computer Information&Engineering*

Kosin University, Dept. of Internet Business*

E-mail : pusrover@dongseo.ac.kr, chban@kosin.ac.kr

요 약

스마트폰과 같은 휴대용기기에서 연산에 필요한 데이터를 저장하기 위하여 플래쉬메모리를 이용한 저장장치를 사용한다. 플래쉬메모리는 비휘발성 저장장치로 대용량의 데이터를 저장할 수 있기 때문에 질의 연산 처리를 위하여 색인 구조를 사용해야 한다. 그러나 플래쉬메모리는 덮어쓰기 연산을 지원하지 않기 때문에 동일 위치에 쓰기 연산을 수행하면 수행 속도가 매우 저하되는 단점이 있다. 이 논문에서는 플래쉬메모리에서 다차원 공간 색인인 고정그리드파일을 구현하고 다양한 조건에서 연산의 수행 속도를 측정한다.

ABSTRACT

Hand-held devices such as smart phones exploit flash memory based storages to store data for processing jobs. Since the flash memory, non-volatile memory, is able to store mass data, it is required to use the index for processing queries. However, the flash memory has the shortcomings that it does not support the overwrite operation and its write operation is very slow. In this paper, we build the fixed grid file, one of the multi-dimensional spatial index, on a flash memory and evaluate the performance test.

키워드

fixed grid file, index, flash memory, write, overwrite

1. 서 론

NAND 플래쉬메모리는 비휘발성 저장장치로서 전원이 꺼져도 저장하고 있는 데이터를 그대로 유지할 수 있다. 플래쉬메모리는 작은 크기로 많은 양의 데이터를 저장할 수 있기 때문에 최근 들어 스마트폰, USB, MP3 등의 휴대용 기기에 많이 사용되고 있다. 또한 플래쉬메모리 칩의 대량 생산과 저장 용량의 확대에 따라 저비용으로 대용량의 데이터를 저장할 수 있어서 워크스테이션 또는 PC와 같은 고정 기기의 저장장치로 주로 사용되었던 자기디스크를 점차 대체하고 있다.

플래쉬메모리는 자기디스크와는 달리 전기 신호를 이용하여 데이터에 접근하기 때문에 저장위치와는 상관없이 균일한 시간에 데이터에 접근할

수 있는 장점이 있다. 그러나 쓰기 연산 속도가 읽기 연산 속도보다 느리고, 특히 덮어쓰기 기능을 지원하지 않기 때문에 동일한 위치에 데이터를 쓰기 위하여 지움 연산을 먼저 수행해야 하는 단점이 있다. 하지만 대용량의 데이터를 효과적으로 처리하기 위한 색인은 구조상 쓰기 연산을 많이 수행해야 하며 특히 데이터의 수정에 따라 동일한 위치에 색인 데이터를 써야 하는 연산이 다수 발생한다. 따라서 기존의 자기디스크에서 구축하고 운용하였던 색인을 플래쉬메모리 기반의 저장장치에서 사용할 때 발생하는 문제점을 살펴볼 필요가 있다.

이 논문에서는 플래쉬메모리에서 데이터 공간 분할 기법의 하나인 고정그리드파일의 성능을 살펴본다. 이를 위하여 고정그리드파일을 플래쉬메

모리 기반의 저장장치에서 구현한다. 그리고 다양한 저장장치와 스케일크기, 버킷의 크기에 따른 색인의 성능을 파악한다.

이 논문은 다음과 같이 구성된다. 2장에서는 플래쉬메모리 기반 색인의 관련 연구를 기술한다. 그리고 3장에서는 고정그리드화일의 성능 비교를 제시한다. 그리고 마지막으로 4장에서 결론을 기술한다.

II. 관련 연구

기존의 플래쉬메모리 기반의 색인은 주로 데이터 유도 방식의 색인인 트리 구조 색인에 대하여 연구되어졌다. [1]에서는 b-tree를 플래쉬메모리에 적용한 BFTL 기법을 제안하였다. BFTL 기법은 새로운 키값이 삽입/삭제되면 삽입 또는 삭제될 키값을 포함한 색인유닛을 구성하고 메인메모리 버퍼인 유보버퍼에 저장한다. 유보버퍼가 가득차면 비로소 플래쉬메모리에 쓰기 연산을 수행한다. 유보버퍼의 색인유닛을 한 번에 쓰기 때문에 동일 노드의 엔트리들이 여러 페이지에 나누어 저장될 수 있으므로 노드변환테이블을 사용하여 분산된 엔트리들을 관리한다.

[2]에서는 r-tree를 플래쉬메모리에 구축하기 위한 기법을 제시하였다. BFTL 기법과 유사하게 삽입/삭제되는 키에 대하여 유보버퍼에 저장하고 유보버퍼가 가득차면 색인유닛으로 변경한 후에 플래쉬메모리에 저장한다. [3]에서는 MR-tree를 플래쉬메모리에서 효율적으로 운영하기 위한 실험을 수행하였다. MR-tree에서도 노드의 삽입/삭제 연산을 메인메모리 버퍼에 저장한 다음에 플래쉬메모리의 페이지 크기가 되면 쓰기 연산을 수행한다.

[4]에서는 환경에 따라 두 가지 모드로 수행하는 self-tuning b+-tree를 제안하였다. 첫 번째 모드는 자기디스크 환경의 트리처럼 쓰기 연산을 수행하면서 동일노드의 엔트리를 동일 페이지에 저장한다. 두 번째 모드는 BFTL 기법처럼 유보버퍼에 쓰기 연산을 저장하다가 한꺼번에 쓰기 연산을 수행한다. 대신에 동일 노드의 엔트리들이 서로 다른 페이지에 저장된다. 연산트리의 각 노드는 두 가지 중 하나의 모드로 수행되면서 수행 비용을 계산하여 너무 많은 비용이 발생하면 다른 모드로 전환된다.

III. 고정그리드화일 성능 비교

동일한 기준으로 색인의 성능을 평가하기 위하여 통상적으로 저장장치로의 접근 횟수를 이용한다. 그러나 자기디스크와 플래쉬메모리는 저장 데이터에 대한 접근 방식이 상이하기 때문에 평균 수행시간을 이용하여 성능 비교를 수행하였다.

실험을 위한 공간데이터는 그림 1과 같이 고정그리드화일에 효율적인 균등분포형의 점 데이터를 사용하였다. 그리고 데이터집합의 크기는 3가

지 종류를 사용하였다. 성능비교를 위한 저장장치는 자기디스크와 SD메모리 그리고 USB메모리를 사용하였다.

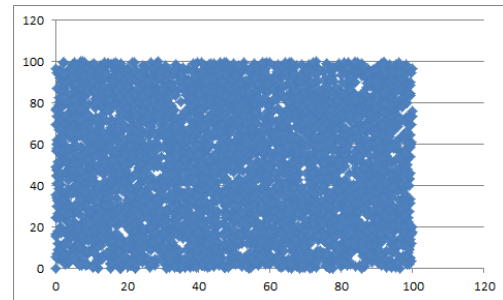


그림 1. 실험 데이터(5,000개)

그림 2는 32X32 스케일이고 데이터 버킷의 크기가 248바이트인 경우에 색인의 구축 시간을 비교한 그래프이다. 그림 2에서 보듯이 자기디스크와 비교하여 SD메모리는 약 3~4%, 그리고 USB메모리는 약 17~18%의 성능만을 보여주고 있다.

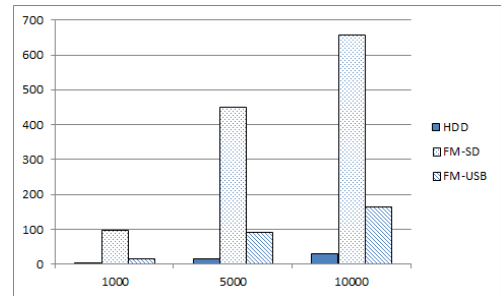


그림 2. 32X32, 버킷 248바이트, 구축시간비교

그림 3은 512X512스케일, 데이터 버킷의 크기가 512바이트인 색인에서 검색질의와 변경질의를 1:9의 비율로 수행한 시간을 비교한 그래프이다. 그림 3에서 보듯이 SD메모리는 약 4~5%, 그리고 USB메모리는 약 16~17%의 성능만을 보여준다.

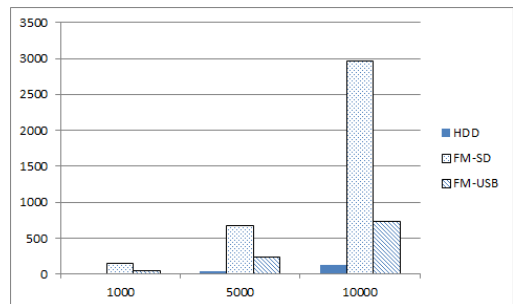


그림 3. 512X512, 버킷 512바이트, 검색질의:변경질의 = 1:9

그림 4는 4X4스케일, 버킷의 크기가 512바이트인 색인에서 덮어쓰기 연산의 횟수에 따른 수행시간의 누적도를 보여준다. 그림 4에서 보듯이 자기디스크는 덮어쓰기가 가능하지만 SD메모리나 USB메모리는 덮어쓰기를 지원하지 않기 때문에

지움 연산을 수행하기 위하여 연산의 수행시간이 자기디스크와 비교할 때 매우 증가하고 있는 것을 보여준다.

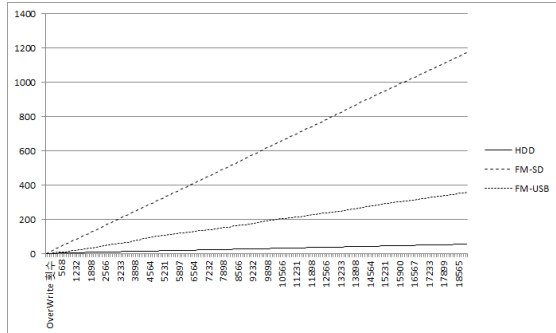


그림 4. 4X4, 버킷 512바이트, 덮어쓰기 연산에 따른 수행시간

IV. 결론

플래쉬메모리는 작은 크기와 적은 비용으로 많은 데이터를 저장할 수 있기 때문에 휴대용 기기에서의 사용량이 증가하고 있다. 그러나 쓰기 연산의 속도가 느리고 특히 덮어쓰기를 지원하지 않기 때문에 동일 위치에 데이터를 쓰기 위해서 지움 연산을 먼저 수행해야하는 단점이 있다. 따라서 플래쉬메모리의 대용량 데이터를 효율적으로 수행하기 위하여 색인을 구축할 때 발생하는 문제점을 살펴볼 필요가 있다.

이 논문에서는 자기디스크와 플래쉬메모리에서 고정그리드화일을 구축한 후에 다양한 환경에서 성능을 실험하였다. 실험 결과를 보면 데이터 공간 분할 방식의 색인이라도 쓰기 연산의 횟수가 증가하면 색인의 성능이 저하되며 특히 덮어쓰기 연산을 수행할 때 자기디스크와 비교하여 매우 성능이 떨어짐을 알 수 있다. 향후 연구로는 플래쉬메모리의 단점을 적용한 고정그리드화일을 설계하는 것이다.

참고문헌

[1] Chil-Hsien Wu et al., "An Efficient B-Tree Layer for Flash Memory Storage System", Real-Time and Embedded Computing Systems and Applications, LNCS 2968, Springer, pp.409-430, 2003.

[2] Chin-Hsien Wu et al., "An Efficient R-Tree Implementation over Flash-Memory Storage Systems", ACM Intl. Symp. on Advances in Geographic Information Systems, ACM, pp.17-24, 2003.

[3] Hyun Seung Lee et al., "Performance of Index trees on Flash Memory", Intl. MultiConf. on Computer Science and Information Technology, pp.725-734, 2007.

[4] Suman Nath and Aman Kansal, "FlashDB: Dynamic Self-tuning Database for NAND Flash", Intl. Conf. on Information Processing in Sensor Networks, pp.410-419, 2007.